



Cisco Elastic Services Controller 6.0 Install and Upgrade Guide

First Published: 2023-10-26

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2023 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE

About This Guide	ix
Audience	ix
Terms and Definitions	ix
Related Documentation	xi

CHAPTER 1

Elastic Services Controller Overview	1
Elastic Services Controller Overview	1

PART I

Installing Cisco Elastic Services Controller on OpenStack	3
--	----------

CHAPTER 2

Prerequisites	5
Virtual Resource Requirements	5
Software Requirements	6
Preparing for the Installation	6

CHAPTER 3

Installing Cisco Elastic Services Controller on OpenStack	9
Installation Scenarios	9
Main Components of Cisco Elastic Services Controller Setup	10
Installing Cisco Elastic Services Controller Using the QCOW Image	10
Additional Installation Options	14
Managing Root Certificates in Cisco Elastic Services Controller	18
Enabling/Disabling the Root Certificate Validation	18
Adding a Root Certificate	18
Removing a Root Certificate	19
Managing Root Certificates During the Upgrade	20
Managing Keystone in Cisco Elastic Services Controller	20

ESCADM Keystore Commands 20
 Using a Bootable Volume in ESC Installation 22

CHAPTER 4

Installing High Availability Active/Standby 23
 High Availability Active/Standby Overview 23
 ESC Active/Standby Architecture 24
 How High Availability Works 24
 Deploying ESC High Availability Active/Standby 25
 Deploying ESC in High Availability Active/Standby Mode on Internal Storage 26
 Deploying ESC in High Availability Active/Standby Mode on Replicate External Storage 27
 Configuring the Northbound Interface Access 28
 Configuring ESC HA Active/Standby with Multiple Interfaces 28
 Configuring the ESC HA Active/Standby Virtual IP Address 29
 Configuring the ESC L3 HA Active/Standby With BGP 30
 Important Notes 33
 Troubleshooting High Availability Active/Standby 34

CHAPTER 5

Cisco Elastic Services Controller Active/Active High Availability Overview 35
 Cisco Elastic Services Controller Active/Active HA Overview 35
 ESC Active/Active Architecture 36

CHAPTER 6

Installing Active/Active High Availability Cluster 37
 Installing Active/Active High Availability Cluster 37
 Setting up the User Configuration 37
 Validating Active/Active High Availability Cluster Post Installation 39
 Adding Default VIM Connector to the Active/Active High Available Cluster 40
 Adding BGP in an Active/Active Cluster 40

CHAPTER 7

Managing Cluster in ESC Active/Active High Availability 43
 Managing Cluster in ESC Active/Active High Availability 43

CHAPTER 8

Configuring GEO in Active/Active High Availability 45
 Configuring GEO in Active/Active High Availability 45

	Transition Conditions	45
	Verifying GEO Services	47
	Active/Active GEO HA Failure Injection Limitations	48
<hr/>		
CHAPTER 9	DRBD Encryption for ESC Active/Standby and Active/Active HA Data Replication	51
	DRBD Encryption for ESC HA Data Replication	51
	ESC HA with DRBD Encryption	51
<hr/>		
CHAPTER 10	Upgrading ESC Active/Active High Availability	53
	Upgrading ESC Active/Active High Availability	53
	Backing up the Database	53
	Removing the Old VMs	54
	Installing a New ESC Active/Active VM	54
	Restoring the ESC Database	55
<hr/>		
CHAPTER 11	Redistributing Workload After Active /Active Leader Failover	57
	Redistributing Workload After Active/Active Leader Failover	57
	Rebalancing API to Redistribute Ownership	57
<hr/>		
PART II	Installing Cisco Elastic Services Controller on VMware vCenter	59
<hr/>		
CHAPTER 12	Prerequisites	61
	Virtual Resource and Hypervisor Requirements	61
	vCenter Resources	61
	Important Notes	62
<hr/>		
CHAPTER 13	Installing Cisco Elastic Services Controller on VMware vCenter	65
	Installing Cisco Elastic Services Controller on VMware vCenter	65
	Preparing to Install Cisco Elastic Services Controller	65
	Installing the Elastic Services Controller Using the OVA Image	66
	Installing Elastic Services Controller Using OVF Tool	68
	Powering on Cisco Elastic Services Controller Virtual Machine	70
	Next Steps: Cisco Elastic Services Controller Virtual Machine	71
	Logging in to Cisco Elastic Services Controller Portal	71

Configuring the Virtual Machine to Automatically Power Up 71

CHAPTER 14

Installing High Availability 73

- High Availability Active/Standby Overview 73
- How High Availability Active/Standby Works 73
- Deploying ESC High Availability Active/Standby with User Data (HA Active/Standby Pair) 74
- Deploying ESC High Availability Active/Standby (Standalone Instances) 78
- Important Notes for ESC HA Active/Standby 79
- Troubleshooting High Availability Active/Standby 79

PART III

Installing Cisco Elastic Services Controller on VMware vCloud Director 81

CHAPTER 15

Installing Cisco Elastic Services Controller on VMware vCloud Director 83

- Uploading the ESC OVA file into a Catalog in VMware vCloud Director 83
- Deploying ESC vApp on vCD Web Client 83
- Adding the VIM Connectors 84
- Adding Self-Signed Certificate 84

PART IV

Installing Cisco Elastic Services Controller on a Kernel-based Virtual Machine (KVM) 85

CHAPTER 16

Installing Cisco Elastic Services Controller on a Kernel-based Virtual Machine 87

- Installing Cisco Elastic Services Controller in a Kernel-based Virtual Machine 87
 - Preparing to Install Cisco Elastic Services Controller on a Kernel-based Virtual Machine 87
 - Installing Elastic Services Controller on a Kernel-Based Virtual Machine 89
- Next Steps: Cisco Elastic Services Controller Kernel-based Virtual Machine 90
 - Logging in to Cisco Elastic Services Controller Portal 90
 - Verifying ESC installation for a Kernel-based Virtual Machine (KVM) 90
 - Troubleshooting Tips 90

PART V

Upgrading D-MONA 91

CHAPTER 17

Upgrading D-MONA 93

- Upgrading D-MONA 93

PART VI**Post Installation Tasks 97****CHAPTER 18****Post Installation Tasks 99**

ESC Lifecycle Notifications During Login	99
Changing the ESC Password	100
Changing the ConfD Netconf/CLI Administrator Password Using the Command Line Interface	100
Creating Readonly User Group for ConfD in ESC	101
Restricting ESC ConfD CLI access	102
Changing Linux Account Password	103
Changing the ESC Portal Password	104
Configuring Pluggable Authentication Module (PAM) Support for Cisco Elastic Services Controller	105
PAM Authentication Service Configurations and User Groups	105
Adding PAM User to an ESC Service/Component	106
Configuring Cisco Elastic Services Controller as Identity Management Client	107
Configuring Cisco Elastic Services Controller as the Identity Policy and Audit Client	107
Authenticating REST Requests	109
REST Authentication	109
Enabling ETSI REST Authentication	109
Changing the REST Interface Password	110
Changing the ETSI REST Interface Password	110
Sending an Authorized REST Request	111
Sending an Authorized ETSI REST Request	111
Configuring Openstack Credentials	112
Enabling Barbican Client in ESC	117
Staging VPC Volume	117
Running MONA in a Root Jail	118
Installing the MONA Root Jail	118
Reconfiguring ESC Virtual Machine	119
Reconfiguring Rsyslog	119
Reconfiguring NTP	120
Reconfiguring DNS	121
Reconfiguring Hosts	122
Reconfiguring Timezone	122

Verifying ESC Configurations and Other Post-Install Operations 122
 Logging in to the ESC Portal 124

PART VII **Upgrading Cisco Elastic Services Controller 127**

CHAPTER 19 **ESC in Maintenance Mode 129**
 Setting ESC in a Maintenance Mode 129
 Using the escadm Tool 129
 Setting ESC in an Operation Mode 130
 Backup the Database from the ESC Standalone Instances 131
 Backup the Database from the ESC HA Active/Standby Instances 132
 Restoring ESC Database 134

CHAPTER 20 **Upgrading Cisco Elastic Services Controller 137**
 Upgrading Standalone ESC Instance 138
 Deploy the ESC for Upgrade 139
 Restoring the ESC Database 139
 Important Notes: 139
 Upgrading ESC HA Active/Standby Instances 140
 Deploying the ESC HA Active/Standby nodes for Upgrade 140
 Restoring the ESC Database on New Active and Standby ESC Instances 140
 Upgrading VNF Monitoring Rules 142

PART VIII **Troubleshooting Cisco Elastic Services Controller Installation 143**

CHAPTER 21 **Troubleshooting ESC Issues 145**
 Viewing ESC Log Messages 145
 General Installation Errors 150
 ESC Failover Scenarios 153

APPENDIX A **Cisco Elastic Services Controller Installer Arguments 155**



About This Guide

This guide describes the installation requirements, procedures, and upgrade procedures for Cisco Elastic Services Controller.

- [Audience, on page ix](#)

Audience

This guide is designed for network administrators responsible for provisioning, configuring, and monitoring VNFs. Cisco Elastic Services Controller (ESC) and the VNFs whose lifecycle it manages are deployed in a Virtual Infrastructure Manager (VIM). Currently OpenStack, VMware vCenter, VMware vCloud Director, and VMware NSX-T are the supported VIMs. The administrator must be familiar with the VIM layer, vCenter, OpenStack, and the commands used.

Cisco ESC is targeted for Service Providers (SPs) and Large Enterprises. ESC helps SPs reduce cost of operating the networks by providing effective and optimal resource usage. For Large Enterprises, ESC automates provisioning, configuring and monitoring of network functions.

Terms and Definitions

The below table defines the terms used in this guide.

Table 1: Terms and Definitions

Terms	Definitions
ESC	Elastic Services Controller (ESC) is a Virtual Network Function Manager (VNFM), performing lifecycle management of Virtual Network Functions.
ETSI	European Telecommunications Standards Institute (ETSI) is an independent standardization organization that has been instrumental in developing standards for information and communications technologies (ICT) within Europe.
ETSI Deployment Flavour	A deployment flavour definition contains information about affinity relationships, scaling, min/max VDU instances, and other policies and constraints to be applied to the VNF instance. The deployment flavour defined in the VNF Descriptor (VNFD) must be selected by passing the <i>flavour_id</i> attribute in the InstantiateVNFRequest payload during the instantiate VNF LCM operation.

Terms	Definitions
HA	ESC High Availability (HA) is a solution for preventing single points of ESC failure and achieving minimum ESC downtime.
KPI	Key Performance Indicator (KPI) measures performance management. KPIs specify what, how and when parameters are measured. KPI incorporates information about source, definitions, measures, calculations for specific parameters.
MSX	Cisco Managed Services Accelerator (MSX) is a service creation and delivery platform that enables fast deployment of cloud-based networking services for both Enterprises and Service Providers customers.
NFV	Network Function Virtualization (NFV) is the principle of separating network functions from the hardware they run on by using virtual hardware abstraction.
NFVO	NFV Orchestrator (NFVO) is a functional block that manages the Network Service (NS) lifecycle and coordinates the management of NS lifecycle, VNF lifecycle (supported by the VNFM) and NFVI resources (supported by the VIM) to ensure an optimized allocation of the necessary resources and connectivity.
NSO	Cisco Network Services Orchestrator (NSO) is an orchestrator for service activation which supports pure physical networks, hybrid networks (physical and virtual) and NFV use cases.
OpenStack Compute Flavor	Flavors define the compute, memory, and storage capacity of nova computing instances. A flavor is an available hardware configuration for a server. It defines the <i>size</i> of a virtual server that can be launched.
Service	A service consists of a single or multiple VNFs.
VDU	The Virtualisation Deployment Unit (VDU) is a construct that can be used in an information model, supporting the description of the deployment and operational behaviour of a subset of a VNF, or the entire VNF if it was not componentized in subsets.
VIM	The Virtualized Infrastructure Manager (VIM) adds a management layer for the data center hardware. Its northbound APIs are consumed by other layers to manage the physical and virtual resources for instantiation, termination, scale in and out procedures, and fault & performance alarms.
VM	A Virtual Machine (VM) is an operating system OS or an application installed on a software, which imitates a dedicated hardware. The end user has the same experience on a virtual machine as they would have on dedicated hardware.
VNF	A Virtual Network Function (VNF) consists of a single or a group of VMs with different software and processes that can be deployed on a Network Function Virtualization (NFV) Infrastructure.
VNFC	A Virtual Network Function Component is (VNFC) a composite part of the VNF, synonymous with a VDU, which could be implemented as a VM or a container.
VNFM	Virtual Network Function Manager (VNFM) manages the life cycle of a VNF.

Related Documentation

The Cisco ESC doc set comprises of the following guides to help you perform installation, configuration; the lifecycle management operations, healing, scaling, monitoring and maintenance of the VNFs using different APIs.

Guide	Information Provided in This Guide
Cisco Elastic Services Controller Release Notes	Includes new features and bugs, known issues.
Cisco Elastic Services Controller Install and Upgrade Guide	Includes procedure for new installation and upgrade scenarios, pre and post installation tasks, and procedure for ESC High Availability (HA) deployment.
Cisco Elastic Services Controller User Guide	Includes lifecycle management operations, monitoring, healing and scaling of the VNFs.
Cisco Elastic Services Controller ETSI NFV MANO User Guide	Includes lifecycle management operations, monitoring, healing and scaling of the VNFs using the ETSI APIs.
Cisco Elastic Services Controller Administration Guide	Includes maintenance, monitoring the health of ESC, and information on system logs generated by ESC.
Cisco Elastic Services Controller NETCONF API Guide	Information on the Cisco Elastic Services Controller NETCONF northbound API, and how to use them.
Cisco Elastic Services Controller REST API Guide	Information on the Cisco Elastic Services Controller RESTful northbound API, and how to use them.
Cisco Elastic Services Controller ETSI REST API Guide	Includes information on the Cisco Elastic Services Controller ETSI APIs, and how to use them.
Cisco Elastic Services Controller Deployment Attributes	Includes information about deployment attributes used in a deployment datamodel.
Cisco Elastic Services Controller Open Source	Includes information on licenses and notices for open source software used in Cisco Elastic Services Controller.

Obtaining Documentation Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation*, at: <http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html>.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.



CHAPTER 1

Elastic Services Controller Overview

- [Elastic Services Controller Overview, on page 1](#)

Elastic Services Controller Overview

Cisco Elastic Services Controller (ESC) is a Virtual Network Functions Manager (VNFM), which performs life cycle management of Virtual Network Functions (VNFs). ESC provides agent-less and multi vendor VNF management by provisioning virtual services, and monitoring their health and load. ESC provides the flexibility to define monitoring rules, and associate actions to be triggered based on the outcome of these rules. As a VNFM, in addition to the typical lifecycle management operations, ESC also supports automatic VM recovery when a VM fails and performs automatic scaling in and out functions. ESC fully integrates with Cisco and other third party applications.

- As part of the Cisco Orchestration Suite, ESC is packaged with Cisco Network Services Orchestrator (NSO), and available within Cisco Solution, Managed Services Accelerator Solution (MSX).
- As a Specialized Virtual Network Function Manager (SVNFM), ESC tightly integrates with the Cisco Mobility VNFs.
- ESC can also be utilized as a Generic Virtual Network Function Manager (GVNFM) to provide lifecycle management for both Cisco and third-party VNFs.

ESC is deployed in a virtual machine within OpenStack, VMware vCenter, KVM and manages its VNFs in a Virtual Infrastructure Manager (VIM).

Elastic Services Controller as a VNF Manager targets the virtually managed services and all service provider NFV deployments such as virtual video, WiFi, authentication and others.

ESC can manage both basic and complex VNFs . Basic VNFs include a single VM such as a vFW, vRouter and others.

Complex VNFs include multiple VMs that are orchestrated as a single entity with dependencies between them.

IPv6 Support

Elastic Services Controller provides IPv6 support on OpenStack for:

- VNF Management
- HA— ESC manages VNFs on IPv4 and IPv6 (OpenStack and KVM only).

Elastic Services Controller provides IPv6 support for northbound interface (for example, NFVO to VNFM), and southbound interface (for example, VNFM to VNF). In order to support both northbound and southbound IPv6 concurrently, the following pre-requisites must be met:

- OpenStack cloud computing is set up and configured for ipv6, including the endpoints (that are ipv6 based).
- The OpenStack cloud computing must contain a Controller, endpoints, and a few Compute hosts, with an ipv6 management and os_api based networks.
- The ESC default security group rules support the IPv6 traffic.



Note When you are deploying a VM, you can attach an out-of-band port of an IPv6 subnet to a VM. However, if you are deleting this VM, you cannot attach the same IPv6 address to another VM due to a known OpenStack issue.



PART I

Installing Cisco Elastic Services Controller on OpenStack

- [Prerequisites, on page 5](#)
- [Installing Cisco Elastic Services Controller on OpenStack, on page 9](#)
- [Installing High Availability Active/Standby, on page 23](#)
- [Cisco Elastic Services Controller Active/Active High Availability Overview, on page 35](#)
- [Installing Active/Active High Availability Cluster, on page 37](#)
- [Managing Cluster in ESC Active/Active High Availability, on page 43](#)
- [Configuring GEO in Active/Active High Availability, on page 45](#)
- [DRBD Encryption for ESC Active/Standby and Active/Active HA Data Replication , on page 51](#)
- [Upgrading ESC Active/Active High Availability, on page 53](#)
- [Redistributing Workload After Active /Active Leader Failover, on page 57](#)



CHAPTER 2

Prerequisites

The following sections detail the prerequisites for installing Cisco Elastic Services Controller:

- [Virtual Resource Requirements, on page 5](#)
- [Software Requirements, on page 6](#)
- [Preparing for the Installation, on page 6](#)

Virtual Resource Requirements

The following table lists the virtual resource requirements for Cisco Elastic Services Controller:

ESC Deployment	Number of ESC VMs	Virtual CPU per ESC VM	Virtual Memory (GB) per ESC VM	Virtual Hard Disk (GB) per ESC VM	Number of total VMs Supported	VIM
ESC Standalone	1	4	8	40	1000	All supported VIMs
ESC Active/Standby HA	2	4	8	40	2500	All supported VIMs
ESC Active/Active HA	3	8	16	40*	5000	OpenStack Only
ESC GEO HA	6	8	16	40*	5000	OpenStack Only



Note *The Active/Active HA and Geo HA deployment can accommodate up to 60 GB of hard disk while provisioning the NFVI resources.

Software Requirements

The following table lists the software requirements on OpenStack:

Requirements	Description
Supported Browsers	Google Chrome 44.x or later
OpenStack Version	Any of the following: <ul style="list-style-type: none"> • Train • Queens (Keystone V3) • Ocata (Keystone V2 and V3)

Preparing for the Installation

Before you perform the installation, ensure that you are prepared by reviewing this checklist:

Requirements	Your Information/ Notes
For Pre-installation Configuration	
QCOW image location	
QCOW image	
VM per Instance	
IP address	
Subnet mask	
Hostname	
Domain name	
Gateway IP address	
Admin password	

Table 2: ESC Release Package

Requirements	Your Information/ Notes
ESC.qcow2	An image file for booting up the ESC instance
bootvm.py	The installation script is compatible with python 2.7.6 and Python 3.4

Pre-requisites for Installing ESC 6.0 on OpenStack:

While glancing the ESC 6.0 image on OpenStack, make sure that the following property is configured for that image, only then the ESC 6.0 image will boot.

```
--property hw_firmware_type=uefi
```

If Cisco Virtual Infrastructure Manager (CVIM) is being used as Virtual Infrastructure platform, then UEFI is NOT SUPPORTED by CVIM 3.x versions. For more details on UEFI support on CVIM platform, please refer to the respective product user guide.

While glancing the ESC 6.0 image on CVIM 4.x platform, ensure that the following properties are configured for that image, only then the ESC 6.0 image will boot.

```
--property hw_firmware_type=uefi--property hw_machine_type=q35
```



Note For the compute host to support UEFI, user may require to install "ovmf" package on the host. Command to install ovmf package differs based on the Host Operating System.



CHAPTER 3

Installing Cisco Elastic Services Controller on OpenStack

This chapter describes how to install Cisco Elastic Services Controller on OpenStack and includes the following sections:

- [Installation Scenarios, on page 9](#)
- [Main Components of Cisco Elastic Services Controller Setup, on page 10](#)
- [Installing Cisco Elastic Services Controller Using the QCOW Image, on page 10](#)
- [Managing Root Certificates in Cisco Elastic Services Controller, on page 18](#)
- [Managing Keystore in Cisco Elastic Services Controller, on page 20](#)
- [Using a Bootable Volume in ESC Installation, on page 22](#)

Installation Scenarios

The following sections briefly describe some of the common deployment scenarios that are addressed by ESC.

Cisco Elastic Services Controller can be installed in different modes as per the requirement. These different modes are configured during installation. The following sections briefly describe some of the common deployment scenarios that are addressed by ESC.

ESC Standalone

In the standalone scenario, a single active VM is deployed for ESC.

ESC with HA

ESC supports High Availability (HA) in the form of Active/Standby and Active/Active models. For Active/Standby model, two ESC instances are deployed in the network to prevent ESC failure and to provide ESC service with minimum interruption. If the active ESC instance fails, the standby instance automatically takes over the ESC services. ESC HA resolves the following single point failures of ESC:

- Network failures
- Power failures
- Dead VM instance
- Scheduled downtime

- Hardware issues
- Internal application failures



Note Installing or upgrading software on the ESC VM is not supported. For further assistance, contact Cisco TAC.

For more information on Installing ESC HA Active/Standby, see [Installing Cisco Elastic Services Controller Using the QCOW Image, on page 10](#) and Installing Cisco Elastic Services Controller on VMware vCenter.

For more information on the Active/Active model, see ESC HA Active/Active Overview Chapter.

Main Components of Cisco Elastic Services Controller Setup

The Cisco Elastic Service Controller (ESC) setup has the following components:

- **Virtual Infrastructure Manager**—Elastic Services Controller (ESC) and its VNFs are deployed in a Virtual Infrastructure Manager (VIM). This might have one or more underlying physical nodes.
- **ESC Virtual Machine**—The ESC VM is a VM that contains all the services and processes used to register and deploy services. This includes the ESC Manager and all other services. ESC provides Netconf API, ETSI API, REST API, and Portal as north bound interfaces to communicate with ESC. ESC VM contains CLI to interact with ESC VM. There are two CLI, one uses the REST API and the other uses Netconf API.

Installing Cisco Elastic Services Controller Using the QCOW Image

You can install Cisco Elastic Services Controller (ESC) on OpenStack by using a QCOW image. ESC will be deployed in the OpenStack as running VM instance to manage VNFs. Therefore, ESC need OpenStack environment parameters to be installed in OpenStack. The installation time varies from 10 to 20 minutes, depending on the host and the storage area network load. This procedure describes how to create ESC Virtual Machine (VM) in OpenStack.

Before you begin

- All system requirements are met as specified in [Prerequisites, on page 5](#).
- You have the information identified in [Preparing for the Installation, on page 6](#).
- Copy the ESC image file on the system where you want to install ESC.
- This system must be accessible by OpenStack.

Procedure

Step 1 Log in to the system where you want to install ESC.

Step 2 Check the compatibility of the bootvm.py and the ESC image:

```
./bootvm.py --version
```

For more information on the ESC installer arguments, see **Appendix: A Cisco Elastic Services Controller Installer Arguments**.

Step 3 In a text editor, create a file named PROJECT-openrc.sh file and add the following authentication information. The following example shows the information for a project called admin, where the OpenStack username is also admin, and the identity host is located at controller node.

Note To set the required environment variables for the OpenStack command-line clients, you must create an environment file called an OpenStack rc file, or openrc.sh file. This project-specific environment file contains the credentials that all OpenStack services use. The ESC installation script requires these OpenStack environment parameters to perform authentication and installation on OpenStack. If all the OpenStack credentials are passed through its own arguments, the bootvm.py script doesn't require these parameters.

```
export OS_NO_CACHE=true
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=admin_pass
export OS_AUTH_URL=http://controller node:35357/v2.0
```

The other OpenStack parameters required for installation are: --os_auth_url, --os_username, --os_password, --os_tenant_name, --bs_os_user_domain_name, --bs_os_project_domain_name, --bs_os_identity_api_version, --bs_os_auth_url, --bs_os_username, --bs_os_password, --bs_os_tenant_name, --bs_os_user_domain_name, --bs_os_project_domain_name, --bs_os_identity_api_version.

For OpenStack V2 API, you need following items to be defined in your global environment variables: --os_password, --os_auth_url, --os_username, --os_tenant_name.

For OpenStack V3 API, set --os_identity_api_version=3 . Other parameters required for OpenStack V3 API are: --os_user_domain_name, --os_project_domain_name, --os_project_name, --os_password, --os_auth_url, --os_username, --os_identity_api_version, --os_ca_cert, --requests_ca_bundle.

Note The arguments, --os_tenant_name, --os_username, --os_password, --os_auth_url will also by default configure the VIM connector. If you want to skip configuring the VIM connector, pass the parameter (--no_vim_credentials) with the bootvm.py. When no_vim_credentials parameter is provided, the bootvm.py arguments (os_tenant_name, os_username, os_password, os_auth_url) are ignored. For more information on configuring VIM connectors after installation, and managing VIM connectors, see Managing VIM Connectors in the *Cisco Elastic Services Controller User Guide*.

Note --os_ca_cert and --requests_ca_bundle arguments are only required for https connection.

Step 4 On any shell from which you want to run OpenStack commands, source the PROJECT-openrc.sh file for the respective project. In this example, you source the admin-openrc.sh file for the admin project .

```
$ source admin-openrc.sh
```

Step 5 Check the environment variables.

```
$ env | grep OS_
```

Step 6 Register ESC image file in the OpenStack image using the glance command:

```
$ glance image-create \
--name <image_name> \
--disk-format <disk_format> \
--container-format <container_format> \
--file <file>\
--progress
```

An example configuration is shown below:

```
$ glance image-create \
--name esc-1_0_01_11_2011-01-01 \
--disk-format qcow2 \
--container-format bare \
--file esc-1_0_01_11_2011-01-01.qcow2 \
--progress
```

The **glance image-create** command is used to create a new image.

Note For ESC 6.0 and above, set the following properties for the ESC image,

For OpenStack:

```
openstack image set --property hw_firmware_type=uefi <Image Name>
```

For CVIM:

```
openstack image set --property hw_firmware_type=uefi --property
hw_machine_type='q35' <Image Name>
```

Verify the image in OpenStack/CVIM,

- a) Using OpenStack Controller dashboard:
 1. Log into OpenStack using your credentials.
 2. Navigate to Admin > Image.
 3. Verify if the image appears in the list.
- b) Using nova CLI:
 1. \$ nova image-show <image_name>

Step 7 The standard resource requirement of ESC is 4vCPU, 8G RAM, and 40GB disk space. ESC installation script takes the pre-defined "m1.large" flavor which has the definition of 4vCPU, 8G RAM and 80G disk space. To use 40GB disk space, create a flavor with the minimum disk space requirement.

```
$ nova flavor-create ESC_FLAVOR_NAME ESC_FLAVOR_ID 8192 40 4
```

Step 8 To deploy ESC VM, do the following:

- a. Ensure that the existing network have connectivity to OpenStack controller. To verify the network connectivity using the nova CLI use:

```
$ nova net-list
```
- b. Record the ID of the network that ESC connects to boot the ESC VM by with image and flavor created earlier. The bootvm.py command requires at least one --user_pass argument to create an admin account for linux (ssh/console access) and at least one --user_confid_pass to create an admin account for ConfD (netconf/cli access). The following are the syntax for these mandatory user credential arguments:


```
--user_pass admin:'PASSWORD-OR-HASH'[:OPTIONAL-PUBLIC-KEY-FILE][:OPTIONAL-ROLE]
--user_confid_pass admin:'PASSWORD-OR-HASH'[:OPTIONAL-PUBLIC-KEY-FILE]
```

Generating hashed password is optional. You can select a plain password as and when required.

To generate a hashed password on Ubuntu OS, use the following command:

```
mkpasswd --method=SHA-512 --salt Xyz123 <<< <Password>
```

Invoking the `esc_nc_cli` command without username and password fails with the following message:

```
admin@esc$ esc_nc_cli --user <username> --password <password> get
esc_datamodel/opdata/status
ERROR Cannot find file /home/admin/.ssh/confd_id_rsa to support public key authentication
with esc-nc-admin
* Check your arguments or installation.
* Usage without --user --password or --privKeyFile arguments requires ssh keys and
configuration.
* Use the following command to generate new SSH Keys and update to authorized_keys:
sudo escadm confd keygen --user admin
* NOTE: Consult with site security policies for use of public key authentication.
```

You must run the `esc_nc_cli` command with a username and password:

```
admin@esc$ esc_nc_cli --user admin --password ADMIN-CONFID-PASSWORD get
esc_datamodel/opdata/status
Operational Data
/opt/cisco/esc/confd/bin/netconf-console --port=830 --host=127.0.0.1 --user admin
--password=***** --get -x "esc_datamodel/opdata/status"
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <esc_datamodel xmlns="http://www.cisco.com/esc/esc">
      <opdata>
        <status>OPER_UP</status>
      </opdata>
    </esc_datamodel>
  </data>
</rpc-reply>
```

The following is an example to install ESC with an authorized public key. In the following example, single quotes are used to avoid conflict with shell reserved characters:

```
--user_pass admin:'$algorithm$salt$hash-of-salt-password':$HOME/.ssh/esc_rsa.pub
--user_confid_pass admin:'$algorithm$salt$hash-of-salt-password':$HOME/.ssh/esc_rsa.pub
```

The public keys are generated as part of a key pair, such as:

```
ssh-keygen -t rsa -b 1024 -C "esc" -N "" -f ~/.ssh/esc_rsa
```

Your public key and identification key are saved in `/home/username/.ssh/esc_rsa` and `esc_rsa.pub` files. For more examples on the user credential arguments, see Appendix A: Cisco Elastic Services Controller Installer Arguments.

- c. To check the details of ESC VM and get the information including the IP address(es) of the ESC VM, use the following command:

```
$ nova show <esc_vm_name>
```

Additional Installation Options

- **Creating ConfD SSH Keys:** You can invoke the following command to create SSH keys and configure the public key authentication:

```
sudo escadm confd keygen --user USERNAME
```

Run the above command on each user account on the ESC VM that requires authentication without password. In an Active/Standby deployment, run the command on both the nodes.

- **Enabling ConfD SSH keys using bootvm.py:** Enabling the following bootvm.py command restores the functionality prior to ESC Release 5.4:

```
--confd_keygen
```

For information on ConfD cli, see Changing the ConfD NetConf CLI Administrator Password Using the Command Line Interface.

- **Deploying ESC in an OpenStack IPv6 environment:** Before deploying ESC instance in IPv6, make sure to source openrc that supports ipv6 addresses. To deploy ESC in IPv6 environment, use the following bootvm arguments:

```
./bootvm.py <esc_vm_name-ipv6> --poll --user_rest_pass <username>:<password> --image <image_name>
--net <ipv6_network> --ipaddr <ipv6_ip_address> --enable-http-rest --user_pass <username>:<password>
--user_confid_pass <username>:<password> --etc_hosts_file <hosts-file-name> --route <default_routing_configuration>
```

- **Deploying ESC in DHCP mode:** If you use the bootvm.py argument without --ipaddr, then the ESC instance will be deployed in a DHCP mode. To deploy ESC in a DHCP network, use the following configuration:

```
./bootvm.py <esc_vm_name> --image <image_name> --net <IPv6 network> <IPv4 network>
--flavor <flavor_name>
--user_pass <username>:<password>
--user_confid_pass <username>:<password>
```



Note By default, ESC only support DHCP in IPv4 networks. If IPv6 is used, you need to log in the ESC VM and run "dhclient -6 ethX" (ethX is the V6 interface name) manually to enable V6 DHCP.

When deploying ESC with multiple network interfaces, with one or more type of DHCP, use `bootvm.py --defroute N` to specify the interface index to be assigned default route and gateway IP.

By default, `N = 0`. So the first network interface on the `bootvm.py` command line is default.

For Example:

```
--defroute 1 will assign <network2> with <default_gateway_ip_address>
./bootvm.py <esc_vm_name> --image <image_id> --net <network1>
<network2> --defroute 1 --gateway_ip <default_gateway_ip_address>
```

- **Using a Bootable Volume in ESC Installation:** You can attach a volume to an ESC instance and launch an instance from inside the volume. For more information, see the section [Using a Bootable Volume in ESC Installation](#).
- **Assigning Floating IP to the ESC:** If you want to associate a floating IP with the ESC instance, do the following:
 1. Check for an available floating IP address and assign it to the ESC VM:


```
$ nova floating-ip-list
$ nova floating-ip-associate esc_vm_name <ip_address>
```
 2. Or create a new floating IP address and assign it to the ESC VM:


```
$ nova floating-ip-create <FLOATING_NETWORK - ID>
$ nova floating-ip-associate esc_vm_name <ip_address>

or
neutron floatingip-create FLOATING_NETWORK
neutron floatingip-associate floating-ip-ID port-ID
```
- **Deploying ESC with static IPs:** To use ESC in a specific network with static IPs, for example, 192.168.0.112 at network1, specify **--ipaddr** and **--gateway_ip** to the bootvm command line, as shown below:



Note Before assigning static IP, make sure the static IP is available and is not being used on other machine.

```
./bootvm.py <esc_vm_name> --image <image_id> --net <network>
--ipaddr <ip_address> --gateway_ip <default_gateway_ip_address> --user_pass
<username>:<password>
--user_confid_pass <username>:<password>
```

- **Deploying ESC with multiple network interfaces:** To use multiple networks for ESC, for example, 192.168.0.112 at network1 and 10.20.0.112 at network2, specify both the IP addresses and the network names of the interfaces in the **--net** and **--ipaddr** arguments in the following command line. In addition, also choose the default gateway for ESC from the gateways of these networks. Specify the default gateway for ESC through the **--gateway_ip** argument.

```
./bootvm.py <esc_vm_name> --image <image_id> --net <network1>
<network2> --ipaddr <ip_address1> <ip_address2> --gateway_ip <default_gateway_ip_address>

--user_pass <username>:<password>
--user_confid_pass <username>:<password>
```



Note If **--flavor** is not specified, bootvm.py will use the default flavor "m1.large" in OpenStack.

Ensure that m1.large exists in OpenStack if no flavor is specified; also ensure that the flavor suits the ESC deployment requirements.

- **Deploy ESC with log forwarding options:** To forward ESC logs to an rsyslog server, specify the IP address of the rsyslog server while creating an ESC VM. Optionally, you can also specify the port and protocol to use.

For example, if the IP address of the rsyslog server is 172.16.0.0 the port on the server to forward logs is 514, and the protocol used is UDP, the ESC installation could be

```
./bootvm.py <esc_vm_name> --image <image_id> --net network1 --rsyslog_server 172.16.0.0
--rsyslog_server_port 514 --rsyslog_server_protocol udp --user_pass <username>:<password>
--user_confid_pass <username>:<password>
```

- **Disabling the ESC GUI:** To boot up ESC VM with the graphical user interface disabled, modify the `--esc_ui_startup` argument value, as shown in the command line below:

```
./bootvm.py <esc_vm_name> --image <image_id> --net <network> --user_pass
<username>:<password>
--user_confid_pass <username>:<password>
--esc_portal_startup=False
```

- **Enabling REST interface for ESC:** To support the REST interface, specify `--enable-https-rest` argument. You can activate REST interface on both https or http:

```
./bootvm.py <esc_vm_name> --image <image_id> --net <network> --user_pass
<username>:<password>
--user_confid_pass <username>:<password> --enable-https-rest
```

OR

```
./bootvm.py <esc_vm_name> --image <image_id> --net <network> --user_pass
<username>:<password>
--user_confid_pass <username>:<password> --enable-http-rest
```

- **Enabling REST interface for ETSI:** To support the ETSI REST interface, specify `--enable-http-etsi` to activate the interface over http, or `--enable-https-etsi` to activate the interface over https.

```
./bootvm.py <esc_vm_name> --image <image_id> --net <network> --user_etsi_pass
<username>:<password> --enable-https-etsi . . .
```

OR

```
./bootvm.py <esc_vm_name> --image <image_id> --net <network> --user_etsi_pass
<username>:<password>
--user_confid_pass <username>:<password> --enable-http-etsi...
```



Note Only the https REST and ETSI interfaces should be enabled in a production environment.

- **Deploying ESC with ETSI OAuth2 Clients:** To add ETSI OAuth2 clients during the installation, use the arguments as shown below. The OAuth2 clients can also be added using the `escadm` commands after the installation.

```
./bootvm.py <esc_vm_name> --image <image_id> --net <network> --
user_etsi_pass <username>:<password> --etsi_oauth2_pass <clientId>:<clientSecret>
--enable-https-etsi ...
```

- **Deploying ESC with global parameters:** To set the global configurations through the `esc_params_file` during the installation, use the arguments as shown below. These global configurations can also be changed through REST API after the installation.



Note The default security group is applied to the tenant during tenant creation. By default, the ESC configuration parameter for the security group, `openstack.DEFAULT_SECURITY_GROUP_TO_TENANT` is set to true. The configuration parameter must be set at the time of installation. You can query or update the parameter on ESC VM through the REST API. If the parameter is set to true, you can create and assign default security group during tenant creation. If the parameter is set to false, you cannot create or assign default security group during tenant creation. For details on the parameters that can be configured through `esc_params_file`, see Appendix A: Cisco Elastic Services Controller Installer Arguments.

```
./bootvm.py <esc_vm_name> --image <image_id> --net <network> --flavor <flavor_name>
--user_pass <username>:<password>:<public key file> --user_confid_pass
<username>:<password>
--esc_params_file <esc parameter configuration file>
```

- **Deploying ESC with ETSI properties:** To set the ETSI properties through the `etsi_params_file` during installation, use the arguments as shown below. These properties can also be changed in the `etsi-production.properties` file after the installation.



Note For details on the properties that can be configured through the `etsi_params_file`, see the ETSI Production Properties chapter in the *ETSI NFV MANO User Guide*.

```
./bootvm.py <esc_vm_name> --image <image_id> --net <network> --flavor <flavor_name>
--user_pass <username>:<password>:<public key file> --user_confid_pass
<username>:<password>
--etsi_params_file <etsi properties file>
```

- **Deploying two instances of ESC to build an ESC HA (Active/Standby) pair:** For more information on deploying ESC HA (Active/Standby), see Configuring High Availability in Installing ESC on OpenStack and Installing ESC on VMware chapters.
- **Adding a Dynamic Mapping File:** In Cisco ESC Release 2.1 and earlier, mapping the actions and metrics defined in the datamodel to the valid actions and metrics available in the monitoring agent was enabled using the `dynamic_mappings.xml` file. The file was stored in the ESC VM and was modified using a text editor. ESC 2.2 and later do not have an `esc-dynamic-mapping` directory and `dynamic_mappings.xml` file. If you want to add an existing `dynamic_mapping.xml` file to the ESC VM, do the following:

1. Backup this file to a location outside of ESC, such as, your home directory.
2. Create `esc-dynamic-mapping` directory on your ESC VM. Ensure that the read permissions are set.
3. Install on your ESC VM using the following bootvm argument:

```
--file
root:root:/opt/cisco/esc/esc-dynamic-mapping/dynamic_mappings.xml:<path-to-local-copy-of-dynamic-mapping.xml>
```

The CRUD operations for mapping the actions and the metrics is available through REST API. To update an existing mapping, delete and add a new mapping through the REST API.

- **Changing the confd password on an ESC VM :** As an administrator, you can configure the confd password through `bootvm.py`, during the installation time:

```
./bootvm.py --user_pass <username>:<password> --user_confid_pass
admin:'PASSWORD-OR-HASH':OPTIONAL-PUBLIC-KEY
```

To reconfigure this password after the installation, execute the following commands:

```
$ /opt/cisco/esc/confd/bin/ssh admin@localhost -p 2024
$ configure
$ set aaa authentication users user admin password <your_password>
$ commit
$ exit
```



Note For the ease of future upgrades, make sure that you keep a copy of all the commands and arguments that are used while installing ESC using the `bootvm.py` file.

Managing Root Certificates in Cisco Elastic Services Controller

Cisco Elastic Services Controller (ESC) provides a mechanism to enable verification of SSL certificates. Currently, this feature is supported only on OpenStack. Certificate validation is enabled by default during the initial ESC boot up. However, ESC also allows you to configure these SSL certificates. This section describes how to enable/disable certificate validation, add/remove, or list the certificates for Cisco Elastic Service Controller on OpenStack. You can add a root certificate during the ESC bootup or even after ESC bootup is completed.

Enabling/Disabling the Root Certificate Validation

Cisco Elastic Services Controller by default enables certificate validation. You can also enable or disable by modifying the parameter, `DISABLE_CERT_VALIDATION`, available under the `Openstack` category in the `esc_params.conf` file, or through the REST interface, or using the `escadm` tool.

On ESC active node, use the command, `sudo escadm enable-certificate` or `sudo escadm disable-certificate` to enable and disable the certificate validation, respectively.

Adding a Root Certificate

You can add a root certificate during the ESC bootup or even after ESC bootup is completed. Before adding certificates, ensure the OpenStack environment file, OpenStack RC file has parameters to perform authentication and installation on OpenStack. The `--os_auth_url` must be specified while passing the parameters. `--os_auth_url` specifies the secure (https) or unsecured (http) keystone URL used by OpenStack for authentication.

- Add certificate for standalone (only) during the bootup time, i.e, during the ESC VM installation:

```
./bootvm.py test-vm --image <image_name> --net <network> [--cert_file CERT_FILE]
[--confd_aes_key CONFD_AES_KEY]
/home/cisco/openstack.crt
--user_pass <username>:<password> --user_confid_pass <username>:<password>
```



Note Currently, ESC does not support adding a certificate for HA Active/Standby during the installation as the keepalived service is not running when a certificate is added.

- Add certificate for standalone/HA Active/Standby after booting up the ESC instance. The escadm tool has an **truststore add** option which has the following arguments: The `--file` argument refers to the CA certificate file. Using this argument you can import any file format supported by the java keytool: X.509 v1, v2, and v3 certificates, and PKCS#7. The `--alias` argument is unique and refers to the name this specific CA certificate is given.

1. Copy/Transfer CA Certificate file to ESC active VM .
2. Add certificate to ESC truststore. To do this, execute the following command:

```
sudo escadm escadm truststore add --alias [ca cert alias] --file [file path]
```

or

```
sudo escadm truststore escadm truststore --alias [ca cert alias] --file [file path]
```

3. Verify the certificate is added.

```
sudo escadm truststore show
```

or

```
sudo escadm truststore show
```

Removing a Root Certificate

The escadm tool has a 'truststore delete' option which only takes `--alias` argument. The `--alias` argument refers to the name of the CA certificate to be deleted. Use this argument on the standalone/HA Active/Standby ESC VM:

Procedure

- Step 1** On the active ESC use escadm to delete certificate from ESC truststore.

```
sudo escadm truststore delete --alias [ca cert alias]
```

or

```
sudo escadm truststore truststore delete --alias [ca cert alias]
```

- Step 2** Verify the certificate is removed.

```
sudo escadm truststore show
```

or

```
sudo escadm truststore show
```

Managing Root Certificates During the Upgrade

- **Image Upgrade:** If you are backing up the ESC DB for upgrade, then no other action is required, the ESC truststore will be restored once the ESC DB is restored. If you are not backing up the ESC DB for upgrade, then each CA certificate needs to be added again to the ESC truststore.
- **RPM Upgrade:** This upgrade method keeps the ESC truststore as is, i.e. all ca certificates in the ESC truststore should remain there after upgrade.

Managing Keystore in Cisco Elastic Services Controller

The keystore is a storage for certificates and keys that an application uses to authenticate itself with different clients.

ESC Keystore holds only one certificate, that is used by all the applications for example, ESCManager, VIMManager, MONA, and so on.

ESCADM provides multiple commands to manage the keystore.

When deploying ESC for the first time, a self signed certificate is created and stored in the keystore by default.

important Notes:

- In an Active/Active mode, the default certificate Common Name (CN) is `db.service.consul`. If you want to set a new certificate, you must set it with the same CN, otherwise, you must add the following configuration to the heat template of all the nodes to disable the certificate validation in MONA when deploying the ESC:

```
mona:
    certificate_validation: false
```

- All the escadm keystore commands require root privileges to be executed on the ESC.

ESCADM Keystore Commands

- `escadm keystore show`

The `escadm keystore show` command displays information about the certificate currently stored in the keystore. It shows informations like, the creation date, Alias, and certificate fingerprint.

For example:

```
$ sudo escadm keystore show
Keystore type: PKCS12
Keystore provider: SUN
Your keystore contains 1 entry
esc, Apr 13, 2020, PrivateKeyEntry,
Certificate fingerprint (SHA1):
FF:11:66:3E:93:DD:3A:0B:9A:72:40:16:35:34:D2:22:E1:25:07:80
```

- `escadm keystore export [--out <file path>]`

The export command displays the full content of the certificate. If the `out` option is specified, the content is saved in a file whose path is specified in the previous option.

- `escadm keystore set-- file <file path>`

The set command replaces the current certificate with a new one residing inside the file whose path is specified in the option `file`.

Ensure that the file is in PEM format, containing both the certificate and a private key.

The following example shows how to generate a new self-signed certificate and set it to the keystore:

1. Use the following command to generate the certificate:

```
openssl req -newkey rsa:2048 -new -nodes -x509 -days 3650 -keyout key.pem -out cert.pem
```

The previous commands gives the following two files:

`key.pem` and `cert.pem`

2. Use the following commands to combine the two files:

```
cat key.pem > server.pem
cat cert.pem >> server.pem
```

3. To set the new certificate to the keystore, use the following command:

```
$ sudo escadm keystore set --file server.pem
```

```
Service "keystore" successfully updated ESC keystore and will take effect once ESC
services are restarted by running "sudo escadm restart"
```



Note Ensure that you delete any left over files containing private keys and certificate after the setting is done.

It is mandatory to restart your system for the change to take effect. Use the following command to restart in standalone mode or H/A mode:

```
sudo escadm restart
```

If the ESC is running in the Active/Active mode, you must restart all the nodes in the same cluster. However, if the ESC is running in an Active/Active GEO mode, you must stop the GEO service on all the clusters, to make sure a GEO switchover does not take place.

Only in GEO Active/Active mode, to stop GEO service, log in to any node in each cluster and use the following command:

```
$ sudo escadm geo stop --cluster
```

You can login to any node in the cluster where the certificate is originally set. Use the following command:

```
$ sudo escadm stop --cluster
$ sudo escadm start --cluster
```

If you are on a GEO Active/Active mode, once all the node are up and running in passing state, login again to any node to start the GEO service. Use the following command to start the service:

```
$ sudo escadm geo start --cluster
```

Using a Bootable Volume in ESC Installation

A volume in OpenStack is a detachable block storage device that can be attached to an ESC instance. You can store and also run ESC instances from a volume.



-
- Note**
- Only one ESC instance can be launched from one volume at a time.
 - ESC installation with a combination of bootable volume and high availability (Active/Standby and Active/Active) on cinder is not supported.
-

To launch an ESC instance from a bootable volume, do the following:

Procedure

- Step 1** Create a bootable volume in OpenStack based on an ESC image or from a bootable volume. The bootable volume must be at least of 30 GB disk size. For more information, see OpenStack documentation.
- Step 2** Deploy ESC VM using the `bootvm.py` command and choose the `--boot_volume` argument instead of the `--image` argument, as shown below:

```
./bootvm.py <esc_vm_name> --boot_volume <volume_name_or_id> --net <network> --user_pass
<username>:<password>
--user_confid_pass <username>:<password> --flavor <flavor_name>
```

- Note**
- Only one of these arguments, `--image` or `--boot_volume` must be passed to the `bootvm.py` command. The installation will fail, if both or none of the arguments are used.
 - When launching an ESC instance from a bootable volume, volume disk size is considered over the flavor disk size.
 - If an ESC instance is deleted, the volume attached to it will not be deleted, as the volume was created out-of-band.
-



CHAPTER 4

Installing High Availability Active/Standby

This chapter contains the following sections:

- [High Availability Active/Standby Overview, on page 23](#)
- [How High Availability Works, on page 24](#)
- [Deploying ESC High Availability Active/Standby, on page 25](#)
- [Configuring the Northbound Interface Access, on page 28](#)
- [Important Notes, on page 33](#)
- [Troubleshooting High Availability Active/Standby, on page 34](#)

High Availability Active/Standby Overview

ESC supports High Availability (HA) in the form of Active/Standby and Active/Active models. For Active/Standby model, two ESC instances are deployed in the network to prevent ESC failure and provide ESC service with minimum service interruption. If the active ESC instance fails, the standby instance automatically takes over the ESC services. ESC HA Active/Standby resolves the following single point failures:

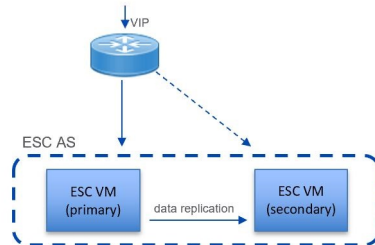
- Network failures
- Power failures
- Dead VM instance
- Scheduled downtime
- Hardware issues
- Internal application failures

ESC Active/Standby Architecture

Figure 1: Cisco Elastic Services Controller Active/Standby Architecture

Local AS Architecture

Active-Standby for all ESC services



Northbound access via Virtual IP (VIP):

- Option 1: VIP as a 2nd ip address on an ESC interface
- Option 2: VIP as an ESC BGP Anycast ip address

Primary:

- One ESC is configured to start up with Primary role
- Primary owns the VIP, receives all northbound requests

Secondary:

- One ESC is configured to start up with Secondary role
- Secondary does not run ESC services
- Secondary receives replicated data from primary
- On primary failure, secondary is promoted to primary role

How High Availability Works

ESC HA Active/Standby network can be either set up as a single installation of a ESC HA Active/Standby pair or deployed as two standalone ESC nodes that are converted into HA pair after re-configuring these nodes post deployment. The HA deployment consists of two ESC instances: an active and a standby. Under normal circumstances, the active ESC instance provides the service. The corresponding standby instance is passive. The standby instance is in constant communication with the active instance and monitors the active instances' status. If the active ESC instance fails, the standby instance automatically takes over the ESC services to provide ESC service with minimum interruption.

The standby also has a complete copy of the database of the active, but it does not actively manage the network until the active instance fails. The KeepAliveD service monitors both active and standby instances activity status. When the active instance fails, the standby takes over automatically. The standby instance takes over active instance to manage the services while active instance restoration is taking place.

When the failed instance is restored, if required you can manually initiate a switch-over and resume network management via the active instance.

Both active and standby ESC instances are connected to the northbound orchestration system through an IPv4 or IPv6 network. For the northbound system, a unique virtual IP address is assigned to access the current active ESC High Availability Active/Standby instance. The deployed VNFs are connected to both ESC active and standby instances through another IPv6 network.

ESC HA Active/Standby nodes are managed by KeepAliveD and DRBD (Replication tool to keep the ESC database synchronized) sync network services. While the KeepAliveD service monitors both active and standby instances status, the DRBD service monitors active instance DB and sync the changes to the standby instance DB. These two services can be co-located on same VIP network or in two separate networks. VM handshake between ESC instances occurs through the KeepAliveD over the IPv4 or IPv6 network.

Deploying ESC High Availability Active/Standby

To deploy Cisco Elastic Services Controller (ESC) High Availability (HA) Active/Standby, ESC standalone instances can be installed on two separate nodes - Active and Standby. For more information see, [How High Availability Works, on page 24](#). You can connect the active and standby instances to either a Cinder volume or Replication based volume (DRBD).

The following deployment mechanisms can be used to deploy ESC HA Active/Standby:

- **Internal Storage**—When ESC HA Active/Standby is configured with Internal storage, the active and the standby instances have individual databases which are always synchronized. In this solution, ESC HA Active/Standby is designed with database replication and DRBD is used as the tool for disk-level replication. The database in the active instance simultaneously propagates the data to the database in the standby instance thus requiring no external storage. In the event of an active instance failing, the standby instance get assigned the role of the active instance along with its own synchronized database.

ESC HA Active/Standby is deployed using internal storage, the ESC instances reply on the virtual IP address (that is `kad_vip` argument), and the interface of `vrrp` instance (that is `kad_vif` argument) to select the active ESC instance. To establish a reliable heartbeat network, it is recommended that the active and standby ESC instances are on different physical hosts. The reliability of the physical links between the ESC instances (such as, network interface bonding) can also be taken into consideration.

- **Replicate External-Storages** — In this type of architecture, ESC HA Active/Standby is configured with DRBD and both active and standby instances store their data in two external storages (OpenStack Cinder volumes). Each ESC node is attached by a Cinder volume and ESC data files are stored in the cinder volume. The data in two ESC nodes are synchronized through the database replication mechanism provided by DRBD.

The table lists the differences between the HA Active/Standby options :

	Internal Storage Based ESC HA Active/Standby	Replicate External Storage Based ESC HA Active/Standby
Data sharing method	Data replication between HA Active/Standby nodes	Data replication between two external storages (cinder volume)
Installation Method	Post-installation Configuration Bootvm Installation	Bootvm Installation
VIM Support	OpenStack, VMware, KVM	OpenStack only
Dependency	VIM independent	Rely on OpenStack cinder
Advantages	<ul style="list-style-type: none"> • No dependency on specific VIM components. • Flexible to build of HA Active/Standby clusters from commodity hardware, without the requirement for shared-storage. 	<ul style="list-style-type: none"> • Use database replication mechanism for data synchronization • Two cinder volumes are used as external storage and are attached to ESC node.

Limitations	The data consistency may be affected in a double fault condition (occurs when both ESC nodes have problems).	The data consistency may be affected in a double fault condition (occurs when both ESC nodes have problems).
--------------------	--	--

Deploying ESC in High Availability Active/Standby Mode on Internal Storage

When you boot ESC instances on active and standby instances, you need to specify the following *bootvm.py* command arguments to deploy ESC HA Active/Standby on an internal storage:

- `kad_vip`



Note When ESC HA Active/Standby is deployed, the `kad_vip` argument allows end users to access the active ESC instance.

- `kad_vif`
- `ha_node_list`

These arguments enable the *bootvm.py* command to automatically set up the internal storage on the OpenStack. For more information on using the *bootvm.py* command arguments, see Appendix A: Cisco Elastic Services Controller Installer Arguments.

To deploy ESC HA Active/Standby instances, use the *bootvm* script on both the nodes with the following arguments:

ON HA NODE 1:

```
$ ./bootvm.py <ESC_HA_Node1>\
--user_pass <username>:<password>\
--user_confid_pass <username>:<password>\
--gateway_ip <default gateway IP address>\
--net <network name1>\
--ipaddr <static ip address>
--image <image_name>\
--avail_zone nova:<openstack zone>\
--ha_node_list=<ESC_HA_NODE1_IP> <ESC_HA_NODE2_IP>\
--db_volume_id <cinder volume id>\
--kad_vip <virtual IP address>\
--kad_vif <VRRP_Interface_Instance>\
--ha_mode drbd
```

ON HA NODE 2:

```
$ ./bootvm.py <ESC_HA_Node2>\
--user_pass <username>:<password>\
--user_confid_pass <username>:<password>\
--gateway_ip <default gateway IP address>\
--net <network name1>\
--ipaddr <static ip addresses>\
--image <image_name>\
--avail_zone nova:<openstack zone>\
--ha_node_list=<ESC_HA_NODE1_IP> <ESC_HA_NODE2_IP>\
--db_volume_id <cinder volume id>\
--kad_vip <virtual IP address>\
```

```
--kad_vif <VRRP_Interface_Instance>\
--ha_mode drbd
```

OR

You can also use **escadm** tool to re-configure ESC HA Active/Standby parameters on each of the standalone ESC VMs. Three parameters "--ha_node_list , --kad_vip, --kad_vif" are all required to configure ESC HA Active/Standby.



Note You should make sure that both the standalone ESC VMs health check is passed before running the below commands to perform HA Active/Standby configuration.

For example:

```
$ sudo bash
$ escadm ha set --ha_node_list='<ESC_HA_NODE1_IP> <ESC_HA_NODE2_IP>' --kad_vip <virtual IP
  address> --kad_vif <VRRP_Interface_Instance>
$ sudo escadm reload
$ sudo escadm restart
```

Deploying ESC in High Availability Active/Standby Mode on Replicate External Storage

Replicate external storage ESC HA Active/Standby requires two cinder volumes for database storage.

Before you begin

- Networks and IP addresses that both ESC instances will connect to
- Keepalived interface and virtual IP for HA Active/Standby switchover

Procedure

Step 1 Create two cinder volumes in OpenStack. The configured cinder volume size should be 3GB.

```
$ cinder create --display-name cindervolume_name_a[SIZE]
$ cinder create --display-name cindervolume_name_b[SIZE]
```

Step 2 Check the status of the created cinder volume and find the uuids for deployment.

```
$ cinder list
```

Step 3 Deploy ESC HA Active/Standby instances. Use the bootvm script on both the nodes with the following arguments:

ON HA NODE 1:

```
$ ./bootvm.py <ESC_HA_Node1>\
--user_pass <username>:<password>\
--user_confid_pass <username>:<password>\
--gateway_ip <default gateway IP address>\
--net <network name>\
--ipaddr <static ip address>\
```

```

--image <image_name>\
--avail_zone nova:<openstack zone>\
--kad_vip <virtual IP address>\
--kad_vif <VRRP_Interface_Instance>\
--ha_node_list=<ESC_HA_NODE1_IP> <ESC_HA_NODE2_IP>\
--db_volume_id <cinder volume id>\
--ha_mode drbd_on_cinder

ON HA NODE 2:

$ ./bootvm.py <ESC_HA_Node2>\
--user_pass <username>:<password>\
--user_confid_pass <username>:<password>\
--gateway_ip <default gateway IP address>\
--net <network name1>\
--ipaddr <static ip address>\
--image <image_name>\
--avail_zone nova:<openstack zone>\
--kad_vip <virtual IP address>\
--kad_vif <VRRP_Interface_Instance>\
--ha_node_list=<ESC_HA_NODE1_IP> <ESC_HA_NODE2_IP>\
--db_volume_id <cinder volume id>\
--ha_mode drbd_on_cinder

```

Step 4 After both VMs are rebooted; one of ESC VM should be in ACTIVE state and the other one should be in STANDBY state. You can check ESC HA Active/Standby state by using following command: `$ sudo escadm status --v`.

Configuring the Northbound Interface Access

When you configure ESC HA Active/Standby, you can also specify a virtual Anycast IP address to the HA Active/Standby pair. The northbound interface as well as the service portal uses virtual Anycast IP address to access the ESC active HA Active/Standby instance. When deploying ESC HA Active/Standby, use the following arguments with the `./bootvm.py` script.

- `--ha_node_list`
- `--kad_vip`
- `--kad_vif`

For more details on these arguments, see section **Appendix A: Cisco Elastic Services Controller Installer Arguments**.

The following section explains how to configure ESC HA Active/Standby with multiple interfaces and to configure the virtual Anycast IP address.

Configuring ESC HA Active/Standby with Multiple Interfaces

You can configure ESC HA Active/Standby with DRDB synchronization and VRRP heartbeat broadcasting on a network interface for data synchronization and VNF monitoring. You can use an additional network interface to allocate Virtual IP for the northbound access. To configure the multiple interfaces on ESC HA Active/Standby nodes, use `--ha_node_list`, `--kad_vip`, `--kad_vif` arguments to specify these multiple network interfaces configuration. For details on these arguments, see section **Appendix A: Cisco Elastic Services Controller Installer Arguments**.



Note KeepAlived doesn't support single IPv4 VIP address with a IPv6 VRRP instance.

Example configuration steps are shown below:

```
./bootvm.py <esc_ha1> \
--user_pass <username>:<password>
--user_confid_pass <username>:<password>
--image <image_id> \
--net <net-name> \
--gateway_ip <default_gateway_ip_address> \
--ipaddr <ip_address1> <ip_address2> \
--ha_node_list < IP addresses HA nodes1> < IP addresses for HA nodes2> \
--kad_vip <keepalived VIP of the HA nodes and the interface for keepalived VIP> \ (for
example: --kad_vip 192.0.2.254:eth2)
--kad_vri <virtual router id of vrrp instance>
--kad_vif <virtual IP of the HA nodes or the interface of the keepalived VRRP> \ (for
example: --kad_vif eth1 )
--ha_mode <HA installation mode> \
--route <routing configuration> \ (for example:192.0.2.254/24:192.168.0.1:eth1 )
--avail_zone nova:<openstack zone> \
```

Similarly, a three network interface can be configured for ESC HA Active/Standby nodes. An example three interfaces configuration is shown below with the following assumptions :

- Network 1 is an IPv6 network used for northbound connection. ESC VIP is allocated in this network and the Orchestrator send requests to ESC through ESC VIP.
- Network 2 is an IPv4 network used for ESC sync traffic (DRDB synchronization) and VRRP heartbeat. This network is also used for OpenStack connection and VNF monitoring.
- Network 3 is another IPv4 network used for management. The SA, rsyslog, etc. can use this network to manage ESC.

```
./bootvm.py esc-ha-0 --image ESC-2_2_x_yyy --net network-v6 network --gateway_ip 192.168.0.1 --ipaddr
2001:cc0:2020::fa 192.168.0.239 192.168.5.239 --ha_node_list 192.168.0.239 192.168.0.243 --kad_vip
[2001:cc0:2020::fc/48]:eth0 --kad_vif eth1 --ha_mode drbd --route 172.16.0.0:eth1 --avail_zone nova: zone
name
```

```
./bootvm.py esc-ha-1 --image ESC-2_2_x_yyy --net network-v6 network lab-net-0 --gateway_ip 192.168.0.1
--ipaddr 2001:cc0:2020::fa 192.168.0.239 192.168.5.239 --ha_node_list 192.168.0.239 192.168.0.243 --kad_vip
[2001:cc0:2020::fc/48]:eth0 --kad_vif eth1 --ha_mode drbd --route 172.16.0.0:eth1 --avail_zone nova: zone
name
```

Configuring the ESC HA Active/Standby Virtual IP Address

In this option, the value of `kad_vip` argument should be a virtual IP, which allows the service portal and the northbound to access the active ESC and send requests to ESC HA Active/Standby service through virtual IP (VIP).

If northbound and both ESC HA Active/Standby nodes are located in the same network, you can connect directly through the virtual IP (VIP). If northbound doesn't sit on the same network as ESC HA Active/Standby, assign a floating IP to ESC HA Active/Standby VIP using the procedure below:

1. Create a port with the VIP address (`kad_vip`) in the same network as ESC's `kad_vip` connects.

```
neutron port-create network --name network_vip --fixed-ip
subnet_id=network-subnet,ip_address=192.168.0.87
```

2. Deploy ESC HA Active/Standby . See **Configuring High-Availability Active/Standby** section in Installing ESC on OpenStack.



Note Make sure the *kad_vip* using the same IP address as the port created above.

3. Associate a floating IP with the port created above. The first uuid is the floating ip id and the second one is the port id.

```
neutron floatingip-associate <floating IP> <port ID>
```

Access ESC HA Active/Standby through the floating IP and it will connect to the ESC active node.

4. For the portal access, make sure the keepalive network is accessible by your browser and the virtual IP is the IP address to access the portal of the active node.

For example, if the VIP is 192.0.2.254, access ESC HA Active/Standby portal with `https://192.0.2.254:9001/`.

Configuring the ESC L3 HA Active/Standby With BGP

To configure BGP for ESC HA Active/Standby, there are two options:

1. Directly booting ESC HA Active/Standby L3 with BGP
2. Using post configuration from existing ESC HA Active/Standby pair

To configure BGP for ESC HA Active/Standby, the following network parameters are required:

- BGP remote IP
- IP of the interface for BGP anycast routing
- BGP local AS number for routing configuration
- BGP remote AS number for routing configuration
- BGP routing configuration
- `--bgp_local_ip`
- `--bgp_local_router_id`



Note You must configure BGP router with neighbors, and restart it. Verify that the router is able to ping the AnyCast IP.

On the BGP router, set two neighbors. The below BGP configuration is designed for Bird router. The configuration is router specific. For each types of router, the procedure is different:

The below configurations are given according to the bootvm command :

```
protocol bgp E3 from EXABGP {
  neighbor 198.18.42.222 as 65012;
}
```

```
protocol bgp E4 from EXABGP {
    neighbor 198.18.61.222 as 65011;
}
```

Booting an ESC VM with BGP options

```
[admin@na-test-52-1 ~]$ health.sh
===== ESC HA (ACTIVE) with DRBD =====
vimmanager (pgid 41908) is running
monitor (pgid 42067) is running
mona (pgid 42130) is running
drbd (pgid 38488) is active
snmp (pgid 2121) is running
etsi (pgid 43247) is running
pgsql (pgid 42566) is running
keepalived (pgid 40281) is running
portal (pgid 43307) is running
confd (pgid 25644) is running
filesystem (pgid 0) is running
escmanager (pgid 42775) is running
=====
ESC HEALTH PASSED
```

```
[admin@na-test-52-2 ~]$ health.sh
===== ESC HA (Standby) with DRBD =====
pgsql is stopped
vimmanager is stopped
monitor is stopped
mona is stopped
drbd (pgid 2471) is standby
etsi is disabled at startup
filesystem is stopped
snmp is disabled at startup
bgp is stopped
keepalived (pgid 2787) is running
portal is stopped
confd is stopped
escmanager is stopped
=====
ESC HEALTH PASSED
```

Use below values for BGP post configuration:

```
./bootvm.sh <NETWORK_VM_name> \
--image <ESC_image> \
--ipaddr <static_IP_address1> <IP_address2> <IP_address_3>\
--gateway_ip <gateway IP address of NETWORK> \
--net <net_id1> <net_id2> <net_id3> \
--network_params_file <network_params_file> \
--host_mapping_file <host_mapping_file> \
--avail_zone <openStack_zone> \
--ha_node_list <IP_address_ha_node_1> <IP_address_ha_node_2> \
--user_portal_pass <user>:<password> \
--user_rest_pass <user>:<password> \
--confd_aes_key <password> \
--kad_vif <interface> \
--user_confd_pass <user>:<password> \
--user_pass <user>:<password> \
--kad_vip <vip address> \
--bgp_remote_ip <BGP_remote_IP_address> \
--bgp_local_ip <BGP_local_IP_address> \
--bgp_local_as <BGP_local_AS_#> \
--bgp_remote_as <BGP_remote_AS_#>\
--bgp_local_router_id <local_BGP_reouter_id> \
--bgp_anycast_ip <BGP_anycast_IP> \
```

```
--bgp_md5 <BGP_MD5>
```

Where,

```
--ip_addr: ----> the local IP address of the ESC VM
--net: ----> the network id(s) in OpenStack that ESC will connect to.
--bgp_anycast_ip: ----> the IP address that NCS will communicate with
--bgp_remote_ip: ----> this IP address of the external router that ESC will peer with
--bgp_local_as: ----> local AS for the ESC "router"
--bgp_remote_as: ----> AS number for the external router ESC will peer with
--bgp_local_router_id: ----> id for the esc "router"
--bgp_md5: ----> optional - md5 to be used to pair with external router
```

Configuring BGP HA Active/Standby Post Configuration

1. For each HA Active/Standby instance, create the network interface file:

```
# cat /etc/sysconfig/network-scripts/ifcfg-lo:2
IPV6INIT='no'
IPADDR='10.0.124.124' <----- bgp anycast IP
BROADCAST='10.0.124.255'
NETWORK='10.0.124.0'
NETMASK='255.255.255.0'
DEVICE='lo:2'
ONBOOT='yes'
NAME='loopback'
```

2. For each HA Active/Standby instance:

```
Bring lo:2 up
# ifup lo:2
```

To configure BGP for ESC HA Active/Standby, use the escadm tool in ESC Virtual Machine, as shown below:

```
$ sudo bash
# escadm bgp set --local_ip LOCAL_IP --anycast_ip ANYCAST_IP --remote_ip REMOTE_IP --local_as
LOCAL_AS --remote_as REMOTE_AS
--local_router_id LOCAL_ROUTER_ID
# escadm reload
# reboot
```

Example:

```
[root@bgp-001 admin]# escadm bgp set --local_ip 198.18.42.124 --anycast_ip 10.0.124.124
--remote_ip 192.168.0.2 --local_as 65124 --remote_as 65000 --local_router_id 198.18.42.124

[root@bgp-002 admin]# escadm bgp set --local_ip 198.18.42.125 --anycast_ip 10.0.124.124
--remote_ip 192.168.0.2 --local_as 65114 --remote_as 65000 --local_router_id 198.18.42.125
```

Configuring a BGP Router

To configure a BGP router, log in to the BGP router to configure BGP Anycast routing. The required parameters are:

```
<Router_AS_#>same as--bgp_remote_asabove
<Esc_ip_address>must be the ESC VM's IP address configured for BGP advertisement.
<ESC_AS_#>same as--bgp_local_asshown above
configure

router bgp <Router_AS_#>

neighbor <ESC_IP_address>
```

```
remote-as <ESC_AS_#>
  address-family ipv6 unicast
  route-policy anycast-in in
  route-policy anycast-out out

route-policy anycast-in
  pass
end-policy

route-policy anycast-out
  drop
end-policy

commit
```

Important Notes

• ESC HA Active/Standby

- An HA Active/Standby failover takes about 2 to 5 minutes. The ESC service will not be available during the switchover time.
- When the switchover is triggered during transactions, all incomplete transactions will be dropped. The requests should be re-sent by northbound if it does not receive any response from ESC.

• External Storage

- If the active ESC instance is suspended by OpenStack command, the switch over will be triggered but the cinder volume won't be attached to the new active ESC instance. This is not a valid use case for ESC HA Active/Standby.

• Internal Storage

- Two ESC instances have to be deployed to establish the HA Active/Standby solution. The ESC HA Active/Standby will start to work when both ESC instances are successfully deployed and are able to connect to each other. If you just deploy one ESC instance with HA Active/Standby parameters, the ESC instance keeps Switching-to-Active state and will not be able to provide any service until it reaches its peer.
- Split-brain scenario can still happen in this ESC HA Active/Standby solution, although the chance is very low.

• ETSI-specific Notes

ESC supports ETSI MANO northbound API defined by the European Telecommunications Standards Institute (ETSI) for NFV Management and Orchestration. The ETSI MANO API is another programmatic interface based on the REST architecture. For more information, see ETSI MANO Compliant Lifecycle Operations in the [Cisco Elastic Services Controller User Guide](#). Consider the following notes while enabling ETSI service on ESC which is in HA Active/Standby mode:

- The `server.address` value in the `etsi-vnfm.properties` file must be set to a Virtual IP (VIP) address. This IP address can be used to communicate back to the ETSI services using API callbacks. If the virtual IP address is not specified, the ETSI service startup may fail.

- The ETSI VNFM service and the `escadm` script generate and maintain the `security.user.name` and `security.user.password` property values. You should not change it manually. The `security.user.password` is encoded.



Note Do not execute the `escadm start|stop|restart` commands from the DRBD/filesystem mount point directory called `/opt/cisco/esc/esc_database/`

Troubleshooting High Availability Active/Standby

- Check for network failures. If a network problem occurs, you must check the following details:
 - The IP address assigned is correct, and is based on the OpenStack configuration.
 - The gateway for each network interface must be pingable.
- Check the logs for troubleshooting:
 - The ESC Admin logs at `/var/log/esc/escadm.log`
 - The ESC manager log at `/var/log/esc/escmanager.log`
 - The AA elector log at `/var/log/esc/elector-{pid}.log`
- Check for DRBD (Replication based ESC HA Active/Standby) for Internal Storage solution:
 - Check the DRBD configuration file at `/etc/drbd.d/esc.res`
 - Access the DRBD log
 - `/var/log/messages|grep drbd`
- To collect log files through CLI, use the following command on all ESC nodes:


```
sudo escadm log collect
```



CHAPTER 5

Cisco Elastic Services Controller Active/Active High Availability Overview

This chapter contains the following sections:

- [Cisco Elastic Services Controller Active/Active HA Overview, on page 35](#)
- [ESC Active/Active Architecture, on page 36](#)

Cisco Elastic Services Controller Active/Active HA Overview

ESC supports High Availability (HA) in the form of an Active/Active model. ESC Active/Active HA has three VMs as a cluster in one datacenter. There are two datacenters. Between the two datacenters, one datacenter acts as active, another acts as standby. ESC Active/Active HA uses Openstack heat template to deploy the three VM cluster in a datacenter.

In a datacenter, ESC service runs on each VM; however, there is only one ESC on a datacenter that runs as cluster leader. DB service runs only on the leader. The ESC services on the other two ESC VMs run as cluster follower. The DB service is only active on ESC services leader VM.

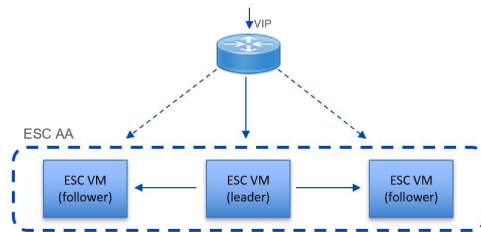
DRBD synchronises the data among ESC VMs. The ESC service on the three ESC VMs connect to the active DB service. When a leader switchover happens, all the ESC service connects to the newly active DB service.

ESC Active/Active Architecture

Figure 2: Cisco Elastic Services Controller Active/Active Architecture

Local AA Architecture

Active-Active LCM core services, Active-Standby support services



Northbound access via Virtual IP (VIP):

- Option 1: VIP as a 2nd ip address on an ESC interface
- Option 2: VIP as an ESC BGP Anycast ip address

Cluster Leader Elections:

- Elect leader on startup and when the leader fails
- Leader owns the VIP, receives all northbound requests

Internal Load Balancing:

- Northbound requests are internally distributed across leader and follower nodes for processing

Active-Standby support services:

- Some microservices only run on the leader node
- For example, a single database on the leader is used by all nodes
- On failure, a new ESC leader is elected, starts leader-only services
- Data is replicated from leader to one or more follower nodes

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential 2





CHAPTER 6

Installing Active/Active High Availability Cluster

This chapter contains the following sections:

- [Installing Active/Active High Availability Cluster](#) , on page 37
- [Validating Active/Active High Availability Cluster Post Installation](#), on page 39
- [Adding Default VIM Connector to the Active/Active High Available Cluster](#), on page 40
- [Adding BGP in an Active/Active Cluster](#), on page 40

Installing Active/Active High Availability Cluster

To configure the Active/Active HA cluster, you can run the following commands on OpenStack, where `openrc` is `openrc` for the OpenStack, and `test` is the stack name.

```
source ~/elastic-services-controller/esc-bootvm-scripts/openrc my-server-42
openstack stack create test -t aa.yaml
```

To check the status of the stack, use the following commands:

- `openstack stack list`
- `openstack stack show test`
- `openstack stack event list test`

Once the stack is in `CREATE_COMPLETE` status, you can `ssh` into the VMs. The `openstack stack show test` command lists the IP addresses for the 3 VMs, you can use them to access to the VMs.



Note The Active/Active HA Cluster is deployed on OpenStack only.

Setting up the User Configuration

You can configure some of the parameters based on your settings, such as network, subnet, using either static IP or DHCP to allocate IP address, flavor, image, password, etc. The configurable parameters are available in `aa-param.yaml`, Openstack heat environment file, when the ESC cluster is instantiated via heat template (`aa.yaml`).

To instantiate ESC cluster, use the following example:

```
openstack stack create name -t aa.yaml -e aa-params.yaml
```

The following example shows how to use a static IP address to configure the port while using environment template:

```
sample@my-server-39:~/aa4.5/apr15$ more aa-params.yaml
parameters:
  network_1_name: network
  subnet_name: subnetwork
  esc_1_ip: 172.23.0.228
  esc_2_ip: 172.23.0.229
  esc_3_ip: 172.23.0.230
```

Following are the user configurable parameters available in aa.yaml :

```
parameters:
  network_1_name:
    type: string
    description: Name of the image
    default: network
  subnet_name:
    type: string
    description: subnet name
  esc_1_ip:
    type: string
    description: static IP address of esc-1 VM.

  esc_2_ip:
    type: string
    description: static IP address of esc-2 VM.

  esc_3_ip:
    type: string
    description: static IP address of esc-3 VM.

resources:
  esc_1_port:
    type: OS::Neutron::Port
    properties:
      network_id: { get_param: network_1_name }
      fixed_ips: [ { "subnet": { get_param: subnet_name }, "ip_address": { get_param: esc_1_ip } } ]

  esc_2_port:
    type: OS::Neutron::Port
    properties:
      network_id: { get_param: network_1_name }
      fixed_ips: [ { "subnet": { get_param: subnet_name }, "ip_address": { get_param: esc_2_ip } } ]

  esc_3_port:
    type: OS::Neutron::Port
    properties:
      network_id: { get_param: network_1_name }
      fixed_ips: [ { "subnet": { get_param: subnet_name }, "ip_address": { get_param: esc_3_ip } } ]

...omitting...
```

The following example shows how to use configurable image, flavor, and vm name prefix from environment template. Ensure that the flavor used suits the ESC deployment requirement.

```
sample@my-server-39:~/aa4.5/apr15$ more aa-params.yaml
parameters:
```

```

nameprefix: abc
image_name: ESC-5_0_DEV_4
flavor_name: m1.large
sample@my-server-39:~/aa4.5/apr15$

```

The following example shows how to use configurable image, flavor, and vmnameprefix from heat template:

```

parameters:
  nameprefix:
    type: string
    description: Name prefix of vm
    default: helen
  image_name:
    type: string
    description: Name of the image
    default: ESC-5_0_DEV_4
  flavor_name:
    type: string
    description: Name of the image
    default: m1.large

esc-1:
  type: OS::Nova::Server
  properties:
    name:
      str_replace:
        template: $nameprefix-esc-1
        params:
          $nameprefix : { get_param: nameprefix }
    image: { get_param: image_name }
    flavor: { get_param: flavor_name }
    ... omitting...

```

Validating Active/Active High Availability Cluster Post Installation

To verify all the ESC nodes use the following commands. Here, all ESC nodes implies each VMs.

```

sample@my-server-39:~$ openstack --insecure server list | grep abc
| 5ea6fc79-2b2a-4064-9c6a-a83d6b06c225 | abc-test-esc-3
| ACTIVE | network=172.23.7.203
| ESC-5_0_DEV_13 | m1.large |
| 10e165d9-5015-4b64-88fe-19e874e6e7c1 | abc-test-esc-1
| ACTIVE | network=172.23.7.205
| ESC-5_0_DEV_13 | m1.large |
| 35f6bad1-865f-4155-8411-d37e2616e079 | abc-test-esc-2
| ACTIVE | network=172.23.7.204
| ESC-5_0_DEV_13 | m1.large |

```

To find out the leader node, ssh into one of the nodes/VMs and run the following:

```

[admin@sample-test-esc-1 ~]$ sudo escadm elector dump
{
  "13078@sample-test-esc-3.novalocal:42143": {
    "state": "FOLLOWER",
    "location": "13078@test1-test-esc-3.novalocal:42143",
    "service": "esc_service"
  },
  "13053@sample-test-esc-2.novalocal:50474": {
    "state": "FOLLOWER",

```

```

        "location": "13053@sample-test-esc-2.novalocal:50474",
        "service": "esc_service"
    },
    "13187@sample-test-esc-1.novalocal:59514": {
        "state": "LEADER",
        "location": "13187@sample-test-esc-1.novalocal:59514",
        "service": "esc_service"
    }
}

```

Adding Default VIM Connector to the Active/Active High Available Cluster

You can add a default vim connector to the 3 ESC VM cluster in the following two different ways:

1. Once the 3 ESC VM cluster boots up, use netconf interface to add a default vim connector by using the following command. The `vim.xml` is the default vim connector deployment file.

```
[admin@name-esc-1 ~]$ esc_nc_cli --host db.service.consul --user admin --password
<admin_password> edit-config vim.xml
```

2. To configure a default vim connector, you must add the default vim connector configuration inside the heat template day0 file. Add the following block in the `write_files` section under the cloud-config in `aa-day0.yaml` file. Once the 3 ESC VM cluster boots up, it creates a default vim connector by its own.

The following example shows how to configure default vim connector in heat template day0 file:

```
- path: /opt/cisco/esc/esc-config/esc_params.conf
  content: |
    openstack.os_auth_url=http://172.16.103.38:35357/v3
    openstack.os_project_name=admin
    openstack.os_tenant_name=admin
    openstack.os_user_domain_name=default
    openstack.os_project_domain_name=default
    openstack.os_identity_api_version=3
    openstack.os_image_api_version=2
    openstack.os_username=admin
    openstack.os_password=password1
```

Adding BGP in an Active/Active Cluster

To initiate the BGP process, add the anycast IP to the lo device. You can configure it in `sys-cfg.yaml`.

For example:

```
#cloud-config
write_files:
- path: /etc/cloud/cloud.cfg.d/sys-cfg.yaml
  content: |
    network:
      version: 1
      config:
      - type: physical
        name: lo
        subnets:
        - type: static
```

```
address: 172.23.188.188/23
```

You must specify the advertise IP for consul. In `esc-config.yaml`, add the following:

```
consul:  
  advertise_addr: 172.23.1.149
```

Following is the example for adding the BGP section:

```
bgp:  
  depend_on: elector:leader  
  anycast_ip: 172.23.188.188/23  
  local_as: '65001'  
  local_ip: 192.168.1.11  
  local_router_id: 192.168.1.11  
  remote_as: '65000'  
  remote_ip: 192.168.1.12
```




CHAPTER 7

Managing Cluster in ESC Active/Active High Availability

This chapter contains the following section:

- [Managing Cluster in ESC Active/Active High Availability, on page 43](#)

Managing Cluster in ESC Active/Active High Availability

To do cluster management in ESC Active/Active HA, you can call some `escadm` commands on any ESC node and execute it on all nodes in Active/Active cluster.

The following are the supported commands for cluster level calling:

- `escadm start`
- `escadm stop`
- `escadm geo start/stop`
- `escadm vim show`

Add the option `--cluster` to execute any of the previous commands to be executed on all nodes in Active/Active local cluster:

For example:

```
sudo escadm geo start --v --cluster
```

The output of the command and exit code of each node are displayed with the IP address of that node to distinguish the execution output, and the output on the local node.

Example for `escadm geo start --cluster`:

```
[root@name-geo-2-1 admin]# escadm geo start --cluster
192.168.1.13 # remote host
exit status : 0
Starting geo service: [OK]
192.168.1.12 # remote host
exit status : 0
Starting geo service: [OK]
Starting geo service: [OK] # output of the local node
```




CHAPTER 8

Configuring GEO in Active/Active High Availability

- [Configuring GEO in Active/Active High Availability](#), on page 45
- [Verifying GEO Services](#), on page 47
- [Active/Active GEO HA Failure Injection Limitations](#), on page 48

Configuring GEO in Active/Active High Availability

ESC Active/Active HA has three VMs as a cluster in one datacenter. The second datacenter consists of GEO-HA.

Following are the 6 pre-defined roles in GEO:

1. `init`: initial role of geo service
2. `pre_primary`
3. `primary`
4. `pre_secondary`
5. `secondary`
6. `unknown`: used when consul is not reachable

GEO can change one role to another. Transitions are defined in `esc-config.yaml`. Each transition is divided into the following 3 parts:

- `from`: current role
- `goto`: destination role
- `condition`: when GEO changes the role

Transition Conditions

When A/A HA Geo comes up, the primary datacenter has to go through `init`, `pre_primary`, and `primary` states; meanwhile, the secondary datacenter has to go through `init`, `pre-secondary`, and `secondary` states. When

all ESC VMs' health check pass on both primary and secondary datacenters, the ESC A/A HA GEO is up and running. It is ready for use.

Condition Functions

The following are all supported condition functions:

1. `return`: do nothing but return the argument
2. `and`: return true if all arguments are true
3. `or`: return true if any of the argument is true
4. `len`: return the length of the argument
5. `equals`: return true if all arguments are equal
6. `true`: return true if args can be tested for truth value in python
7. `false`: opposite to 'true'

The following are samples for GEO configurations on primary datacenter:

```

on_init: consul start
on_primary: start
on_secondary: stop
on_stop: consul stop
startup: manual
transitions:
- condition:
  return:
    and:
      - equals:
        - len: service1
        - 3
      - equals:
        - len: service2
        - 3
  rise: 3
service1:
  dc: dcl
  name: consul_agent
  passing: true
  type: service
service2:
  dc: dc2
  name: geo
  passing: true
  type: service
from: init
goto: primary
- condition:
  fall: 2
  return:
    equals:
      - len: service
      - 3
  service:
    dc: dcl
    name: consul_agent
  from: primary
  goto: secondary

```

The following are samples for GEO configurations on secondary datacenter:

```

on_init: consul start
on_primary: start
on_secondary: stop
on_stop: consul stop
startup: manual
transitions:
- condition:
  return:
    and:
      - equals:
        - len: service1
        - 3
      - equals:
        - len: service2
        - 3
  rise: 3
service1:
  dc: dc1
  name: consul_agent
  passing: true
  type: service
service2:
  dc: dc2
  name: geo
  passing: true
  type: service
from: init
goto: secondary
- condition:
  fall: 2
  return:
    equals:
      - len: service
      - 3
  service:
    dc: dc1
    name: consul_agent
from: secondary
goto: primary

```

Verifying GEO Services

To start Active/Active GEO-HA, run the following command:

```
escadm geo start
```

To verify the GEO status, use the following command:

```
[root@test-geo3-ha-1 esc-scripts]# escadm geo status
geo (pgid 3745) is primary
```

To verify the GEO services in current datacenter, use the following command:

```
[root@test-geo3-ha-1 esc-scripts]# escadm geo dump
{
  "37410@test-geo3-ha-2.novalocal:44793": {
    "role": "primary",
    "location": "37410@test-geo3-ha-2.novalocal:44793",
    "service": "geo"
  },
  "43391@test-geo3-ha-3.novalocal:52459": {
    "role": "primary",
    "location": "43391@test-geo3-ha-3.novalocal:52459",

```

```

        "service": "geo"
    },
    "37898@test-geo3-ha-1.novalocal:38841": {
        "role": "primary",
        "location": "37898@test-geo3-ha-1.novalocal:38841",
        "service": "geo"
    }
}

```

To verify all the GEO services in the datacenters, use the following command:

```

[root@test-geo4-ha-1 admin]# escadm geo dump --all
{
  "3745@test-geo4-ha-1.novalocal:36760": {
    "role": "primary",
    "location": "3745@test-geo4-ha-1.novalocal:36760",
    "service": "geo"
  },
  "3742@test-geo4-ha-6.novalocal:42362": {
    "role": "secondary",
    "location": "3742@test-geo4-ha-6.novalocal:42362",
    "service": "geo"
  },
  "3738@test-geo4-ha-3.novalocal:51936": {
    "role": "primary",
    "location": "3738@test-geo4-ha-3.novalocal:51936",
    "service": "geo"
  },
  "3713@test-geo4-ha-4.novalocal:37604": {
    "role": "secondary",
    "location": "3713@test-geo4-ha-4.novalocal:37604",
    "service": "geo"
  },
  "3710@test-geo4-ha-2.novalocal:44450": {
    "role": "primary",
    "location": "3710@test-geo4-ha-2.novalocal:44450",
    "service": "geo"
  },
  "3714@test-geo4-ha-5.novalocal:34875": {
    "role": "secondary",
    "location": "3714@test-geo4-ha-5.novalocal:34875",
    "service": "geo"
  }
}

```

Active/Active GEO HA Failure Injection Limitations

The ESC Active/Active GEO HA is enhanced to support a one-way GEO HA failover with a maintenance window to move it back to healthy state.

If a GEO failover happens and the ESC VMs move to the unhealthy state, use the following steps to bring ESC A/A GEO HA back to the healthy state through manual intervention:

Procedure

-
- Step 1** Resolve any issues in Datacenter1 (DC1), that may have caused the failures and enabled the GEO switch to Datacenter2 (DC2).

- Step 2** Ensure that the consul is running in at least 2 nodes in DC1 and DC2.
- Step 3** Run the `sudo escadm geo replicate -all` command on DC2 when DC1 has at least two nodes with consul in the running state.
- Step 4** Run the `sudo escadm stop` command on all the 6 ESC VMs.
- Step 5** Run the `sudo escadm geo restart` command on all the 6 ESC VMs.
-

What to do next



Note ESC does not support any operations on ESC VMs in DC1 after the GEO HA fails over to DC2.



CHAPTER 9

DRBD Encryption for ESC Active/Standby and Active/Active HA Data Replication

This chapter contains the following sections:

- [DRBD Encryption for ESC HA Data Replication](#) , on page 51
- [ESC HA with DRBD Encryption](#), on page 51

DRBD Encryption for ESC HA Data Replication

ESC uses DRBD for data replication across different nodes in an HA cluster environment. DRBD layers logical block device over existing local block devices on cluster nodes.

The written data to the active node is transferred to the lower-level block device and simultaneously propagated to the secondary node(s). Currently ESC mounts DRBD device directly on `/opt/cisco/esc/esc_database`.

For Example:

```
# df
Filesystem            1K-blocks    Used Available Use% Mounted on
devtmpfs              2961760      0   2961760  0% /dev
tmpfs                 2972164      4   2972160  1% /dev/shm
tmpfs                 2972164    8748  2963416  1% /run
...
tmpfs                 594436       0    594436  0% /run/user/1004
/dev/mapper/esc_crypt 3028620    57212  2797848  3% /opt/cisco/esc/esc_database
```

Block device encryption encrypts or decrypts the data transparently as it is written/read from block devices, the underlying block device sees only encrypted data.

Security is enhanced with the `dm-crypt/LUKS` layer to encrypt the data in DRBD partition, between filesystem and DRBD device. LUKS (Linux Unified Key Setup) is a specification for block device encryption.

ESC HA with DRBD Encryption

The following bootvm commands boot ESC HA with DRBD encrypted:

Select DRBD LUKS encryption through `bootvm.py`. There are 4 variations that result in the equivalent ESC day-0 `user-data/esc-config.yaml` when passed to the ESC VM instance.

```
bootvm.py --fs_encryption_type luks --fs_luks_key_prompt
bootvm.py --fs_encryption_type luks --fs_luks_key 'LuksKeyValue'
=> injects the luks key into default file location /opt/cisco/esc/esc-config/luks_key
```

```
bootvm.py --fs_encryption_type luks --file
root:0400:/opt/cisco/esc/esc-config/luks_key:path-to-local-luks-key-file
=> injects a local file containing the luks key
```

The following command shows an advanced usage to manage the luks key file at a different path on the ESC VM filesystem:

```
bootvm.py --fs_encryption_type luks --fs_luks_key_file path-on-esc-vm-luks-key-file
--fs_luks_key_prompt
bootvm.py --fs_encryption_type luks --fs_luks_key_file path-on-esc-vm-luks-key-file
--fs_luks_key 'LuksKeyValue'
=> injects the luks key into a different file location
```

```
bootvm.py --fs_encryption_type luks --fs_luks_key_file path-on-esc-vm-luks-key-file --file
root:0400:path-on-esc-vm-luks-key-file:path-to-local-luks-key-file
=> injects the luks key as read from a local file into a different file location
```

Use the following commands if you are installing ESC with a custom user-data, such as ESC Active/Active deployment with heat templates:

The luks key is specified as a day-0 file and an attribute under `esc-config.yaml / filesystem`.

Encode the luks key as base64:

```
base64 <<<'LuksKeyValue'
THVrc0tleVZhbHVlCg==
```

Then, insert the previous luks key into the user-data / cloud-config file:

```
write_files:
- path: /opt/cisco/esc/esc-config/luks_key
  owner: root:root
  permissions: '400'
  encoding: b64
  content: THVrc0tleVZhbHVlCg==

- path: /opt/cisco/esc/esc-config/esc-config.yaml
  owner: root:esc-user
  permissions: '0640'
  content: |
    resources:
      filesystem:
        depend_on: drbd:active
        encryption_type: luks
        luks_key_file: /opt/cisco/esc/esc-config/luks_key
```




CHAPTER 10

Upgrading ESC Active/Active High Availability

This chapter contains the following sections:

- [Upgrading ESC Active/Active High Availability, on page 53](#)

Upgrading ESC Active/Active High Availability

Cisco Elastic Service Controller Active/Active HA supports local Active/Active to Active/Active simple upgrade.

Local Active/Active to Active/Active Simple Upgrade

Procedure

- Step 1** Back up the database. For more information, see the section, [Backing up the Database, on page 53](#).
 - Step 2** Remove old VMs. For more information, see the section, [Removing the Old VMs, on page 54](#) [Removing Old VMs](#).
 - Step 3** Install new ESC Active/Active VMs. For more information, see the section, [Installing a New ESC Active/Active VM, on page 54](#).
 - Step 4** Restore the ESC database. For more information, see the section, [Restoring the ESC Database, on page 55](#).
-

Backing up the Database

Before upgrade, take the back up of database using the following procedure.

Procedure

- Step 1** Put the ESC leader VM in the maintenance mode, by running the following command:

```
escadm op_mode set --mode=maintenance
```
- Step 2** Wait until all the ESC VMs stop processing transactions. To verify, run the following command:

```
escadm ip_trans
```

Step 3 Create a backup of database on the ESC leader by running the following command:

```
escadm backup --file dbback.tar, scp <dbback.tar>
```

Step 4 Collect logs from all the ESC VMs by running the following command:

```
escadm log collect
scp
```

Removing the Old VMs

Procedure

Step 1 Shutdown all the ESC follower VMs and the ESC leader VM running the following command:

```
nova stop
```

Step 2 Remove old ESC Active/Active VMs from the OpenStack by running the following command:

```
openstack stack delete {stack name}
```

Installing a New ESC Active/Active VM

After backing up the database and shutting down of the old ESC Active/Active VMs, a new/upgraded (based on new ESC package) Active/Active ESC VM must be installed.

Procedure

Step 1 For OpenStack, register a new image by running the following command:

```
glance image-create
```

Step 2 Install the new ESC Active/Active VMs by running the following command:

```
openstack stack create {stack name} --template {location of the template file}
```

Step 3 Check all the ESC VMs health, and stop the escadm service in the follower VMs by running the following command:

```
sudo escadm stop for all followers VMs
```

Step 4 Once the escadm service is stopped in all the follower VMs, stop the escadm service in the leader VM by running the following command:

```
sudo escadm stop
```

Restoring the ESC Database

Restore the ESC database on the new ESC instance, using the following procedure:

Procedure

Step 1 Copy the backup file to the new leader by running the following command:

```
scp
```

Step 2 Restore the database on the ESC leader by running the following command:

```
sudo escadm restore --file <dbback.tar>
```

After restoring, the restore process starts the escadm service in the leader VM. However, the escadm service remains to be stopped in all the follower VMs.

Step 3 Verify that the ESC leader VM is running all the services without any interruption.

Step 4 Put the ESC leader VM into the operation mode by running the following command:

```
sudo escadm op_mode set --mode=operation
```

Step 5 Start the ESC services on the follower VMs by running the following command:

```
sudo escadm start
```



CHAPTER 11

Redistributing Workload After Active /Active Leader Failover

- [Redistributing Workload After Active/Active Leader Failover, on page 57](#)

Redistributing Workload After Active/Active Leader Failover

When one of the nodes(leader/follower) in the ESC Active/Active setup fails, ESC equally distributes the ownership of the loads from the failed node to the available healthy nodes. When the failed node is available again, ESC does not redistribute the configurations automatically from the overloaded nodes to the nodes that are back from failed state to active state.

With a large number of deployments, the nodes with loads are likely to experience performance issues. ESC provides a new REST API that can be called on demand to redistribute the configurations from the overloaded nodes to the available nodes based on ownership transfer and ownership policies.

When the rebalancing API is triggered, ESC performs the following functions to transfer the deployments from one node to another:

- Unloading the deployment with VMs in memory, unset monitor from the current owner
- Assigning new owner based on the ownership policy
- Updating ownership record
- Notifying the new owner

Rebalancing API to Redistribute Ownership

When the failed node is available again in the Active/Active setup, ESC does not redistribute the ownership from the overloaded nodes to the healthy node. You can manually trigger rebalancing API to redistribute the ownerships among available nodes.

Use the following URL to trigger the rebalancing API:

```
http://localhost:8080/ESCManager/internal/ownership/rebalance
```

When rebalancing API is triggered, ESC uses an optimal distribution approach to find the overloaded nodes. ESC identifies the configurations to be transferred and calls remotely to unload the configurations in the current owner and reload remotely in the new owners of the earmarked configurations.

The following example shows how rebalancing is done:

URL: `http://localhost:8080/ESCManager/internal/ownership/rebalance`

Method: PUT

Input: None

```
{
  "esc1-uuid": "--> esc node uuid
  {
    "previous_configs_count": 300, --> number of configs before rebalance
    "requested_unload_count": 100, --> number of configs earmarked for unload
    "accepted_unload_count": 100, --> number of configs actually unloaded
    "current_configs_count": 200 --> number of configs after rebalance
  },
  "esc2-uuid":
  {
    "previous_configs_count": 300,
    "requested_unload_count": 100,
    "accepted_unload_count": 90,
    "current_configs_count": 210
  },
  "esc3-uuid":
  {
    "previous_configs_count": 0, --> new esc node, no configs yet
    "requested_unload_count": 0, --> previous config count is less than optimal count
per esc, so no unload requested
    "accepted_unload_count": 0,
    "current_configs_count": 190--> 100 from esc1 and 90 from esc2 got loaded
  }
}
Exception: {"error": {"error_code":500, "error_message": "Internal Error"}}
```

You can trigger the rebalancing API multiple times until you receive the following message:

No configs found for rebalance.

You will receive an error code/message when one of the following instances occurs.

- During the ongoing transfer of ownership due to a failover.
- If there are less than two healthy ESC nodes.
- If the pause ownership is enabled.
- If ESC is standalone.



PART II

Installing Cisco Elastic Services Controller on VMware vCenter

- [Prerequisites, on page 61](#)
- [Installing Cisco Elastic Services Controller on VMware vCenter, on page 65](#)
- [Installing High Availability, on page 73](#)



CHAPTER 12

Prerequisites

The following sections detail the prerequisites for installing Cisco Elastic Services Controller:

- [Virtual Resource and Hypervisor Requirements, on page 61](#)
- [vCenter Resources, on page 61](#)
- [Important Notes, on page 62](#)

Virtual Resource and Hypervisor Requirements

The following table lists the prerequisites to install Cisco Elastic Services Controller on VMware vCenter or vSphere:

See the [VMware Compatibility Guide](#) to confirm that VMware supports your hardware platform.

Requirement	Description
System Requirements	
Virtual CPUs	4 VCPUs
Memory	8 GB RAM
Disk Space	40 GB
Hypervisor Requirements	
VMware vCenter	ESC supports VMware vCenter version 6.5, 6.7, 7.0
VMware vCloud Director (vCD)	ESC supports VMware vCD version 10.3(vCloud API v35.0).

vCenter Resources

Resources to be created/installed on vCenter:

- **Datacenters:** At least one datacenter. For more details, see the **Important Notes** below.
- **Hosts:** Host configuration based on your targeted performance objectives. Each Host under the single vDS must have at least two physical Network Interface Card (NIC) connected, (one for vCenter

Management Interface by default, and the other used to assign to VDS's uplink portgroup). This setup is required for data access across hosts.

- **Compute Clusters:** Clusters can be created to group several hosts together.
- **Datastores:** Shared datastore is required if user wants to leverage DRS.
- **Distributed Switches:** At least one distributed switch that will contains all the VNF supporting networks.

Important Notes

Keep in mind the following important notes while installing ESC on a VMware:

- A single ESC instance will only support:
 - Multiple Datacenter supported deployment, network, image, subnet creation
 - One vSphere Distributed Switch (VDS)
- DPM, HA Active/Standby, and vMotion must be off.
- If DRS is enabled, it has to be in the "Manual Mode".
- Fault Tolerance is not supported.
- Datastore Cluster is not supported, only flat datastore(s) structure under the cluster or under the datacenter are supported.
- ESC only supports a default resource pool. Adding and creating resource pools are not supported.
- Image (Template) created through ESC are stored under `/esc-ovas` folder .
- Day-0, smart license, and other supported files are packed into a ISO file, and uploaded to the same folder where the VM rest, then mount it as a CD-ROM to the VM.
- ESC/VIM does not respond for the name and file content passed in for generating ISO file. They have to be provided according to each template's requirements. e.g. for ASAv, the day-0 config has to be named as "day0-config", and smart license token has to be named as "idtoken".
- 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. See the [Troubleshooting guide](#) , page 91 to increase the timeout for rollback.
- The following VM features and operations are not supported in all versions of the Cisco CSR 1000V. If still these operations are used or performed, there may be risk of encountering dropped packets, dropped connections, and other error statistics.
 1. DRS
 2. Suspend
 3. Snapshot
 4. Resume

- Although deployments can be processed without shared storage, ESC does not guarantee optimized computing resource. Shared storage(s) should associate with as many as possible hosts, which will give more opportunity to DRS to balance resources.
- Every time a redeploy happens as part of recovery on VMware, VM's interface(s) will have different MAC addresses.
- All the VM group defined in a datamodel must accompany with a "zone-host" placement policy, meaning the deployment has to be either host-targeted or cluster-targeted.
- Recovery may fail, if a VM has PCI/PCIe passthrough device(s) attached, when it's recovered to a computing-host (picked based on ESC placement algorithm) which does not have any PCI/PCIe passthrough enabled device available .
- For PCI/PCIe passthrough working, DRS has to be off .
- If you experience a PowerOn error on a VM that has PCI/PCIe passthrough device(s) attached to it, update the VM or the image (template) the VM is cloned from, using the solution described [here](#).

Pre-requisites for Installing ESC 6.0 on VMware vCenter:

While booting ESC 6.0 OVA using vCenter GUI set the Firmware as EFI as shown in the picture:





CHAPTER 13

Installing Cisco Elastic Services Controller on VMware vCenter

This chapter describes how to install Cisco Elastic Services Controller on VMware vCenter and includes the following sections:

- [Installing Cisco Elastic Services Controller on VMware vCenter, on page 65](#)
- [Next Steps: Cisco Elastic Services Controller Virtual Machine, on page 71](#)

Installing Cisco Elastic Services Controller on VMware vCenter

Cisco Elastic Services Controller can be installed in a VMware ESXi hypervisor and can be accessed or managed using vSphere client of VMware. You can install Cisco Elastic Services Controller in a VMware environment using an Open Virtual Appliance (OVA) package.

The VMware vSphere client can be connected directly to your ESXi installation, or it can be connected to a vCenter server which in turn is connected to your vSphere installation. Connecting through vCenter provides a number of capabilities that connecting directly to ESXi does not. If a vCenter server is available and associated with the ESXi installation, it should be used.

Preparing to Install Cisco Elastic Services Controller

In order to install Cisco Elastic Services Controller and configure its network connection, you have to answer several questions. Some of these questions concern the networking environment in which the virtual machine is being installed, and some of them concern values which are unique to the particular virtual machine being installed.

Before you perform the installation, ensure that you are prepared by reviewing this checklist:

Requirements	Your Information/ Notes
OVA image location	
OVA image	
VSphere Web Client	
Hostname	

Requirements	Your Information/ Notes
IP address	
Subnet mask	
Network	
vCenter IP	
vCenter Port	
vCenter credentials	
Datacenter Name	
Datastore Host	
Computer Cluster Name	

Installing the Elastic Services Controller Using the OVA Image

To install Cisco Elastic Services Controller, you must first download the correct installation file.

Using vSphere, connect directly to the ESXi installation or the vCenter server, and select the ESXi installation where the OVA is to be deployed.

This procedure describes how to deploy the Elastic Services Controller OVA image on VMware.

Before you begin

- Set your keyboard to United States English.
- Confirm that the Elastic Services Controller OVA image is available from the VMware vSphere Client.
- Make sure that all system requirements are met as specified in the Chapter 6: Prerequisites.
- Gather the information identified in Preparing to Install Cisco Elastic Services Controller.

Procedure

- Step 1** Using the VMware vSphere Client, log in to the vCenter server.
- Step 2** Choose **vCenterHome > Hosts and Clusters**. Right click the host where you want to deploy ESC, and then choose **Deploy OVF Template**.
- Step 3** In the wizard, provide the information as described in the following table:

Screen	Action
Select source	Select the Elastic Services Controller OVA.
Review details	Review OVF Template details.
Select name and folder	Enter a name and select a folder for the VM.

Screen	Action
Select configuration	Select any one of the following deployment configurations: <ul style="list-style-type: none"> • Large, 1 Network • Large, 2 Networks • Large, 3 Networks
Select a resource	Select the host or cluster to run the ESC template.
Select storage	Select a location to store the files for the VM and a provisioning type. The storage can be local or shared remote, such as NFS or SAN. You can choose either Thin provisioned format or Thick provisioned format to store the VM virtual disks.
Select networks	Based on the Network configuration for the deployment selected in the Select configuration option, you can allocate the pre-configured networks in vCenter to ESC network interfaces.
Customize template	
Bootstrap Properties	
Username	Administrator username for remote login.
Password	Administrator password.
Host name	VM Hostname.
Network IP	VM IP address.
Network Gateway	Gateway IP address.
Enable Https Rest	Enable external REST interface over HTTPS on port 8443.
Enable Portal startup	Enable Portal startup at port 9001 (for https).
VIM Settings of vCenter Server	
vCenter IP	IP address of vCenter server for VNF deployment .
vCenter Port	Port of the vCenter server.
vCenter Username	Username to access vCenter server.
vCenter Password	Password to access vCenter server.

Screen	Action
Datacenter Name	Name of the Datacenter in target vCenter for VNF deployment (Default VDC after Multi-VDC supported)
Datastore Name	The destination datastore of all the image (template) create through ESC .
Datastore Host	The destination computing-host of all the image (template) create through ESC.
Ready to Complete	Review the deployment settings. Caution Any discrepancies can cause VM booting issues. Carefully review the IP address, subnet mask, and gateway information for accuracy.
Public Key	Administrator authorized public key for remote login.
ConfD Username	Administrator username for netconf and ConfD CLI.
ConfD Password	Administrator password for netconf and ConfD CLI.
ConfD Public Key	Administrator authorized public key for netconf and ConfD CLI.

Step 4 Check the **Power on after deployment** check box to power on the VM after deployment.

Step 5 Click **Finish**.
A progress indicator shows the task progress until Elastic Services Controller is deployed.

Step 6 After Elastic Services Controller is successfully deployed, click **Close**.

Step 7 Power on the Elastic Services Controller VM.

Installing Elastic Services Controller Using OVF Tool

In addition to installing the Elastic Services Controller using the OVA image, you can use the VMware OVF Tool, a command-line client, to install Elastic Services Controller on VMware vCenter or vSphere.

To install Elastic Services Controller (ESC) from the command line, do the following:

Procedure

Step 1 Use the probe mode to learn the properties of the OVA package. The probe mode allows you to investigate the contents of a source.

To invoke the probe mode, use the **ovftool** command with only a source and no target.

```
>ovftool <source locator>
```

The following example shows the result of probing the ESC OVA.


```

NETWORK_OVA=(Path to the OVA Package)

NETWORK_HOSTNAME="$(User Name)"
NETWORK_GATEWAY="192.0.2.1"
NETWORK_NET1_IP="192.0.2.0.xx/24" #
NETWORK_NET2_IP="192.51.100.xx/24"
ADMIN_USERNAME="(admin name)"
ADMIN_PASSWORD="(password)"
HTTPS_REST="True"

VMWARE_VCENTER_PORT='80'
VMWARE_VCENTER_IP='192.0.2.0.xx'
VMWARE_DATASTORE_HOST='192.0.2.0.xx'
VMWARE_DATACENTER_NAME='DC-NETWORK-1'
VMWARE_DATASTORE_NAME='cluster-datastore1'
VMWARE_COMPUTE_CLUSTER_NAME='DC-CLUSTER-1'
VMWARE_VCENTER_USERNAME='root'
VMWARE_VCENTER_PASSWORD='password'
VMWARE_VCENTER_FOLDER="$USER"

# All valid deployment options:
#       4CPU-8GB (default)
#       4CPU-8GB-2Net
#       4CPU-8GB-3Net
DEPLOYMENT_OPTION="4CPU-8GB-2Net"

```

Step 2

Before you deploy the ESC OVA, configure the properties of the OVA packages. Ensure the following OVA package properties are updated for the ESC OVA: NETWORK_OVA, NETWORK_HOSTNAME, VMWARE_VCENTER_FOLDER, NETWORK_NET1_IP, NETWORK_NET2_IP, and VMWARE_VCENTER_FOLDER.

The OVA descriptors contain configuration properties for the OVA package. You can set only one property at a time, but you can have multiple instances of the option per command. For multiple property mappings, repeat the option, separating them with a blank, for example `--prop:p1=v1 --prop:p2=v2 --prop:p3=v3`.

```

>.ovftool/ovftool\
--powerOn \
--acceptAllEulas \
--noSSLVerify \
--datastore=$VMWARE_DATASTORE_NAME \
--diskMode=thin \
--name=$NETWORK_HOSTNAME \
--deploymentOption=$DEPLOYMENT_OPTION \
--vmFolder=$VMWARE_VCENTER_FOLDER \
--prop:admin_username=$ADMIN_USERNAME --prop:admin_password=$ADMIN_PASSWORD \
--prop:admin_username=admin \
--prop:admin_password='Strong4Security!' \
--prop:confd_admin_username=admin \
--prop:confd_admin_password='Strong4Security!' \
--prop:network_hostname=$NETWORK_HOSTNAME \
--prop:vmware_vcenter_port=$VMWARE_VCENTER_PORT \
--prop:vmware_vcenter_ip=$VMWARE_VCENTER_IP \
--prop:vmware_datastore_host=$VMWARE_DATASTORE_HOST \
--prop:vmware_datacenter_name=$VMWARE_DATACENTER_NAME \
--prop:vmware_vcenter_username=$VMWARE_VCENTER_USERNAME \
--prop:vmware_datastore_name=$VMWARE_DATASTORE_NAME \
--prop:vmware_compute_cluster_name=$VMWARE_COMPUTE_CLUSTER_NAME \
--prop:vmware_vcenter_password=$VMWARE_VCENTER_PASSWORD \
--prop:net1_ip=$NETWORK_NET1_IP \
--prop:net2_ip=$NETWORK_NET2_IP \
--prop:gateway=$NETWORK_GATEWAY \
--prop:https_rest=$HTTPS_REST \
--net:"Network1=VM Network" --net:"Network2=MgtNetwork" --net:"Network3=VNFNetwork" \

```

```

$NETWORK_OVA
vi://$VMWARE_VCENTER_USERNAME:$VMWARE_VCENTER_PASSWORD@$VMWARE_VCENTER_IP/$VMWARE_DATACENTER_NAME/
host/$VMWARE_COMPUTE_CLUSTER_NAME

```

Following are some advanced examples of passing user credentials through properties.

Advanced usage with password hash:

```

--prop:admin_username=admin \
--prop:admin_password='$6$wnOi$UDQnkKm2tQtr2jDNhoo4wS42ffYfYmzxMKLDugfzTbTXmMQDw146VzpxQvMureaa125.agYHZUqQ8L.sdm2v0'
\
--prop:confd_admin_username=admin \
--prop:confd_admin_password='$6$wnOi$UDQnkKm2tQtr2jDNhoo4wS42ffYfYmzxMKLDugfzTbTXmMQDw146VzpxQvMureaa125.agYHZUqQ8L.sdm2v0'
\

```

Advanced usage with password hash and authorized public key:

```

--prop:admin_username=admin \
--prop:admin_password='$6$wnOi$UDQnkKm2tQtr2jDNhoo4wS42ffYfYmzxMKLDugfzTbTXmMQDw146VzpxQvMureaa125.agYHZUqQ8L.sdm2v0'
\
--prop:admin_public_key='ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAu+nkTtu2pShVbTYL+mmKxtmz5dNXfy8IeX/1H5fXsODH1EAyS1zHGfXq36RT5vIG/
+c2uV8fRsWaY7xXDrdGICxfkPuEj2UQH2MQx2yFjMFcaSAT56hsqE= admin@net' \
--prop:confd_admin_username=admin \
--prop:confd_admin_password='$6$wnOi$UDQnkKm2tQtr2jDNhoo4wS42ffYfYmzxMKLDugfzTbTXmMQDw146VzpxQvMureaa125.agYHZUqQ8L.sdm2v0'
\
--prop:confd_admin_public_key='ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAu+nkTtu2pShVbTYL+mmKxtmz5dNXfy8IeX+XxU6TT2sTsxkKtVy8u0AeBp1qKzKp+
c2uV8fRsWaY7xXDrdGICxfkPuEj2UQH2MQx2yFjMFcaSAT56hsqE= admin@net' \

```

Note You need to replace the variables (IP addresses, root password, VM names, and so on) in the examples above with values from your own system.

Step 3 To deploy the OVA package with the VMware OVF Tool, use the following command syntax:

```
>ovftool <source locator> <target locator>
```

where *<source locator>* is the path to the OVA package and *<target locator>* is the path target for the virtual machine, OVA package or VI. A VI location refers to any location on a VMware product, such as vSphere, VMware Server or ESXi. For more information on the VMware OVF Tool, see the VMware OVF Tool user documentation.

The ESC VM is deployed on VMware and powered on automatically.


Powering on Cisco Elastic Services Controller Virtual Machine

To power on the Cisco Elastic Services Controller virtual machine (VM):



Note You must set the memory and CPUs based on the requirements prior to clicking the power on. Once you start the VM you cannot change the memory or CPU settings until you shut down.

Procedure

- Step 1** After deploying the VM, select the virtual machine name in vSphere, right-click on it and select **Open Console**.
- Step 2** Click the **Power on** button (). During the initial boot of the newly deployed machine, you will be prompted to enter a root (system) password, which is not the Cisco Elastic Services Controller portal password. Initialization may vary depending on the setup.

Note This is the root password for the underlying Linux operating system on which the Cisco Elastic Services Controller portal is installed. You will be asked to enter this password twice. You will need root access to the underlying Linux operating system at various times in the future, so make sure that you remember this password.

The End User License Agreement window appears on the first boot. Read the license agreement in its entirety, and only if you understand and accept the license terms, enter y (Yes).

Next Steps: Cisco Elastic Services Controller Virtual Machine

Logging in to Cisco Elastic Services Controller Portal

Configuring the Virtual Machine to Automatically Power Up

You can configure the ESXi hypervisor to automatically power up the ESC VM when power is restored to the ESXi hypervisor layer.



Note You must manually power up the VM.

Procedure

- Step 1** In the vSphere client, select the ESXi machine to which you are connected. It is not a specific VM that you have to select but the ESXi hypervisor on which they reside.
- Step 2** Select the **Configuration** tab.
- Step 3** Click the **Virtual Machine Startup/Shutdown** link under the **Software** area. You should see the VM in the list shown in window.
- Step 4** Click the **Properties...** link present at the top right corner of the page. If you do not see that, resize the window until you do.
- The Virtual Machine Startup and Shutdown page is displayed.
- Step 5** Check the **Allow virtual machines to start and stop automatically with the system** check box.

Step 6 Select the virtual machine running ESC and use the **Move Up** button on the right to move it up into the group labeled **Automatic Startup**

Step 7 Click **OK**

This ensures that whenever power is restored to the ESXi hypervisor, the ESC VM powers up automatically.



CHAPTER 14

Installing High Availability

This chapter contains the following sections:

- [High Availability Active/Standby Overview](#), on page 73
- [How High Availability Active/Standby Works](#), on page 73
- [Deploying ESC High Availability Active/Standby with User Data \(HA Active/Standby Pair\)](#), on page 74
- [Deploying ESC High Availability Active/Standby \(Standalone Instances\)](#), on page 78
- [Important Notes for ESC HA Active/Standby](#), on page 79
- [Troubleshooting High Availability Active/Standby](#), on page 79

High Availability Active/Standby Overview

ESC supports High Availability (HA) in the form of Active/Standby and Active/Active models. For Active/Standby model, two ESC instances are deployed in the network to prevent ESC failure and provide ESC service with minimum service interruption. If the active ESC instance fails, the standby instance automatically takes over the ESC services. ESC HA Active/Standby resolves the following single point failures:

- Network failures
- Power failures
- Dead VM instance
- Scheduled downtime
- Hardware issues
- Internal application failures

How High Availability Active/Standby Works

A High Availability Active/Standby deployment consists of two ESC instances: an active and a standby. Under normal circumstances, the active ESC instance provides the services. The corresponding standby instance is passive. The standby instance is in constant communication with the active instance and monitors the active instances' status. If the active ESC instance fails, the standby instance automatically takes over the ESC services to provide ESC service with minimum interruption.

The standby also has a complete copy of the database of the active, but it does not actively manage the network until the active instance fails. When the active instance fails, the standby takes over automatically. Standby instance takes over active instance to manage the services while active instance restoration taken place.

When the failed instance is restored, failback operations can be initiated to resume network management via the original active instance.

ESC instances are managed by using KeepAliveD service. The VM handshake between ESC instances occurs through the KeepAliveD over the IPv4 network.

Deploying ESC High Availability Active/Standby with User Data (HA Active/Standby Pair)

Before you begin:

- Cisco Elastic Services Controller (ESC) High Availability (HA) Active/Standby requires a network to keep alive and replicate database between active and standby nodes. Both ESC VMs must have at least one network interface connecting to the same network and must be able to communicate to each other through the network.
- Ensure the two ESC VMs are located in different hosts and datastores so that single point failures can be prevented.

You can deploy ESC HA Active/Standby on VMware vCenter or vSphere in either of two ways:

- Deploying ESC HA Active/Standby with user data as a High Availability Active/Standby pair (Supported from ESC 4.2)
- Deploying ESC HA Active/Standby as two standalone instances and then using post configuration to set them as a High Availability pair. For more information, see the section on "Deploying ESC High Availability Active/Standby (Standalone Instances)".

To deploy ESC HA Active/Standby on VMware vCenter or vSphere with user data as a High Availability Active/Standby pair, define the user data file for each HA Active/Standby instance and then point the user data for each instance via ovftool. The encoding of user data is done via a set of commands in the ovftool script, and the result of this is set as a variable to the `-prop:user-data=` property in the ovftool.



Note The admin user/password and confd user/password properties are mandatory OVF properties. These properties cannot be defined in the user-data files.

- Define the two VMs for ESC HA Active/Standby.

User Data 1

```
#cloud-config
ssh_pwauth: True
write_files:
- path: /etc/cloud/cloud.cfg.d/sys-cfg.yaml
  content: |
    network:
      version: 1
      config:
        - type: nameserver
          address:
```

```

- 161.44.124.122
- type: physical
  name: eth0
  subnets:
  - type: static
    address: 172.16.0.0
    netmask: 255.255.255.0
    routes:
    - gateway: 172.16.0.0
      network: 0.0.0.0
      netmask: 0.0.0.0
- path: /opt/cisco/esc/esc-config/esc-config.yaml
  content: |
    resources:
      confd:
        option: start-phase0
      drbd:
        nodes:
        - 172.16.0.0
        - 172.16.1.0
        run_forever: true
      esc_service:
        depend_on: filesystem
        type: group
      escmanager:
        depend_on:
        - pgsql
        - mona
        - vimmanager
      etsi:
        depend_on: pgsql
        startup: false
      filesystem:
        depend_on: drbd:active
      keepalived:
        vip: 172.16.2.0
      portal:
        depend_on: escmanager
        startup: false
      snmp:
        startup: false
runcmd:
- [ cloud-init-per, once, escadm_ovf_merge, sh, -c, "/usr/bin/escadm ovf merge"]
- [ cloud-init-per, once, escservicestart, sh, -c, "chkconfig esc_service on && service esc_service start"]

```

User data 2

```

#cloud-config
ssh_pwauth: True
write_files:
- path: /etc/cloud/cloud.cfg.d/sys-cfg.yaml
  content: |
    network:
      version: 1
      config:
      - type: nameserver
        address:
        - 161.44.124.122
      - type: physical
        name: eth0
        subnets:
        - type: static
          address: 172.16.1.0
          netmask: 255.255.255.0

```

```

        routes:
        - gateway: 172.16.0.0
          network: 0.0.0.0
          netmask: 0.0.0.0
- path: /opt/cisco/esc/esc-config/esc-config.yaml
  content: |
    resources:
      confd:
        option: start-phase0
      drbd:
        nodes:
        - 172.16.0.0
        - 172.16.1.0
        run_forever: true
      esc_service:
        depend_on: filesystem
        type: group
      escmanager:
        depend_on:
        - pgsqldb
        - mona
        - vimmanager
      etsi:
        depend_on: pgsqldb
        startup: false
      filesystem:
        depend_on: drbd:active
      keepalived:
        vip: 172.16.2.0
      portal:
        depend_on: escmanager
        startup: false
      snmp:
        startup: false
  runcmd:
  - [ cloud-init-per, once, escadm_ovf_merge, sh, -c, "/usr/bin/escadm ovf merge"]
  - [ cloud-init-per, once, escservicestart, sh, -c, "chkconfig esc_service on && service esc_service start"]

```

- OVFTool should be called twice - once for each VM instance. Each instance needs to provide a "--prop:user-data" property to point to its hashed user-data.
- Here is an example to boot a pair of HA Active/Standby instances that use 172.16.0.0 and 172.16.1.0 (floating) IPs to its instances, and 172.16.2.0 as a KAD_VIP.

```

user_data_1=`cat ./user-data-1`
user_data_2=`cat ./user-data-2`
dec_user_data_1=`echo "$user_data_1" | base64 | tr -d '[:space:]'`
dec_user_data_2=`echo "$user_data_2" | base64 | tr -d '[:space:]'`
# vcenter-16 is the developer lab for vmware5
ESC_OVA=/scratch/BUILD-${ESC_IMAGE}/BUILD-${ESC_IMAGE}/ESC-${ESC_IMAGE}.ova
# All valid deployment options:
#       2CPU-4GB
#       4CPU-8GB (default)
#       4CPU-8GB-2Net
#       4CPU-8GB-3Net
DEPLOYMENT_OPTION="4CPU-8GB-2Net"
deploy_vmware_vml() {
  /usr/bin/ovftool \
  --powerOn \
  --acceptAllEulas \
  --noSSLVerify \
  --datastore=$VM_WARE_DATASTORE_NAME \
  --diskMode=thin \

```



```

--name=$INSTANCE_NAME"-0" \
--deploymentOption=$DEPLOYMENT_OPTION \
--vmFolder=$FOLDER \
--prop:admin_username=$ESC_VM_USERNAME --prop:admin_password=$ESC_VM_PASSWORD \
--prop:esc_hostname=$INSTANCE_NAME"-0" \
--prop:rest_username=$REST_USERNAME \
--prop:rest_password=$REST_PASSWORD \
--prop:portal_username=$PORTAL_USERNAME \
--prop:portal_password=$PORTAL_PASSWORD \
--prop:confd_admin_username=$CONFD_USERNAME \
--prop:confd_admin_password=$CONFD_PASSWORD \
--prop:vmware_vcenter_port=$VMWARE_VCENTER_PORT \
--prop:vmware_vcenter_ip=$VM_WARE_VCENTER_IP \
--prop:vmware_datastore_host=$VM_WARE_DATASTORE_HOST \
--prop:vmware_datacenter_name=$VM_WARE_DATACENTER_NAME \
--prop:vmware_vcenter_username=$VM_WARE_VCENTER_USERNAME \
--prop:vmware_datastore_name=$VM_WARE_DATASTORE_NAME \
--prop:vmware_vcenter_password=$VM_WARE_VCENTER_PASSWORD \
--prop:net1_ip=$NET1_IP1 \
--prop:net2_ip=$NET2_IP1 \
--prop:gateway=$ESC_GATEWAY \
--prop:https_rest=$HTTPS_REST \
--prop:user-data=$dec_user_data_1 \
--net:"Network1=VM Network" --net:"Network2=MgtNetwork" --net:"Network3=VNFNetwork" \
  $ESC_OVA vi://$VM_WARE_VCENTER_USERNAME:$VM_WARE_VCENTER_PASSWORD@$VM_WARE_VCENTER_IP/
$VM_WARE_DATACENTER_NAME/host/$VM_WARE_DATASTORE_CLUSTER
}
deploy_vmware_vm2() {
/usr/bin/ovftool \
--powerOn \
--acceptAllEulas \
--noSSLVerify \
--datastore=$VM_WARE_DATASTORE_NAME \
--diskMode=thin \
--name=$INSTANCE_NAME"-1" \
--deploymentOption=$DEPLOYMENT_OPTION \
--vmFolder=$FOLDER \
--prop:admin_username=$ESC_VM_USERNAME --prop:admin_password=$ESC_VM_PASSWORD \
--prop:esc_hostname=$INSTANCE_NAME"-1" \
--prop:rest_username=$REST_USERNAME \
--prop:rest_password=$REST_PASSWORD \
--prop:portal_username=$PORTAL_USERNAME \
--prop:portal_password=$PORTAL_PASSWORD \
--prop:confd_admin_username=$CONFD_USERNAME \
--prop:confd_admin_password=$CONFD_PASSWORD \
--prop:vmware_vcenter_port=$VMWARE_VCENTER_PORT \
--prop:vmware_vcenter_ip=$VM_WARE_VCENTER_IP \
--prop:vmware_datastore_host=$VM_WARE_DATASTORE_HOST \
--prop:vmware_datacenter_name=$VM_WARE_DATACENTER_NAME \
--prop:vmware_vcenter_username=$VM_WARE_VCENTER_USERNAME \
--prop:vmware_datastore_name=$VM_WARE_DATASTORE_NAME \
--prop:vmware_vcenter_password=$VM_WARE_VCENTER_PASSWORD \
--prop:net1_ip=$NET1_IP2 \
--prop:net2_ip=$NET2_IP2 \
--prop:gateway=$ESC_GATEWAY \
--prop:https_rest=$HTTPS_REST \
--prop:user-data=$dec_user_data_2 \
--net:"Network1=VM Network" --net:"Network2=MgtNetwork" --net:"Network3=VNFNetwork" \
  $ESC_OVA vi://$VM_WARE_VCENTER_USERNAME:$VM_WARE_VCENTER_PASSWORD@$VM_WARE_VCENTER_IP/
$VM_WARE_DATACENTER_NAME/host/$VM_WARE_DATASTORE_CLUSTER
}
deploy_vmware_vm1
deploy_vmware_vm2

```

- Once the VMs are deployed successfully, you can check the status of ESC HA Active/Standby. You will find that one VM instance is booted as ACTIVE while the other VM instance is a STANDBY.

Deploying ESC High Availability Active/Standby (Standalone Instances)

To deploy ESC HA Active/Standby on VMware vCenter or vSphere, two separate standalone nodes need to be installed first. After the standalone ESC instances are installed, reconfigure these nodes to turn them into active and standby using the following:

- `kad_vip`
- `kad_vif`
- `ha_node_list`



Note

- On each ESC VM, we need to run `escadm` tool to configure ESC HA Active/Standby parameters and then reload and restart the `escadm` service.
- When you are deploying ESC HA Active/Standby, the `kad_vip` argument allows end users to access the active ESC instance.

Procedure

Step 1 Log in to the ESC Standalone instances.

Step 2 As an admin user, run the `escadm` tool on both the active and standby instances and provide the corresponding arguments.

- **`kad_vip`**— Specifies the IP address for Keepalived VIP (virtual IP) plus the interface of Keepalived VIP [ESC-HA Active/Standby]
- **`kad_vif`**— Specifies the interface for Keepalived virtual IP and keepalived VRRP [ESC-HA Active/Standby]. You can also use this argument to only specify the interface for keepalived VRRP, if the VIP interface is already specified using the `kad_vip` argument.
- **`ha_node_list`**— Specifies list of IP addresses for HA Active/Standby nodes in the active/standby cluster for DRDB synchronization. This argument is utilized for replication-based HA Active/Standby solution only. For ESC instances with multiple network interfaces, the IP addresses should be within the network that `--kad_vif` argument specifies .

```
$ sudo escadm ha set --kad_vip= <ESC_HA_VIP> --kad_vif= <ESC_KEEPLIVE_IF> --ha_node_list=
<ESC_NODE_1_IP> <ESC_NODE_2_IP>
$ sudo escadm reload
$ sudo escadm restart
```

Step 3 After the restart, one ESC VM should be in active state and the other one should be in standby state.

Step 4 Add the VIP to the allowed address pairs for both VMs so that the VIP is reachable from outside.

Step 5 Verify the status of each ESC instance.

```
# sudo escadm status
```

The following table lists few other command to check the status:

Status	CLI Commands
ESC HA Active/Standby Role	<code>cat /opt/cisco/esc/keepalived_state</code>
ESC Health	<code>sudo escadm health</code>
ESC Service Status	<p>If you want to see more details (such as status of the VIM manager, SNMP, portal, ESC manager, keepalived status and so on), add '-v':</p> <pre>sudo escadm status --v</pre> <p>To check the detailed status, check the <code>/var/log/esc/escadm.log</code></p>

Important Notes for ESC HA Active/Standby

- The HA Active/Standby failover takes about 2 to 5 minutes based on the number of managed VNFs to be operational. ESC service will not be available during the switchover time.
- When the switchover is triggered during transactions, all incomplete transactions will be dropped. The requests should be re-sent by Northbound interface if it does not receive any response from ESC.

Troubleshooting High Availability Active/Standby

- Check for network failures. If a network problem occurs, you must check the following details:
 - The IP address assigned is correct, and is based on the OpenStack configuration.
 - The gateway for each network interface must be pinged.
- Check the logs for troubleshooting:
 - The ESC manager log at `/var/log/esc/escmanager.log`
 - The KeepAliveD log at `/var/log/messages` by `grep keepalived`
 - The ESC service status log at `/var/log/esc/escadm.log`



PART **III**

Installing Cisco Elastic Services Controller on VMware vCloud Director

- [Installing Cisco Elastic Services Controller on VMware vCloud Director, on page 83](#)



CHAPTER 15

Installing Cisco Elastic Services Controller on VMware vCloud Director

This chapter describes how to install Cisco Elastic Services Controller on VMware vCloud Director and includes the following sections:

- [Uploading the ESC OVA file into a Catalog in VMware vCloud Director, on page 83](#)
- [Deploying ESC vApp on vCD Web Client, on page 83](#)
- [Adding the VIM Connectors, on page 84](#)
- [Adding Self-Signed Certificate, on page 84](#)

Uploading the ESC OVA file into a Catalog in VMware vCloud Director

Procedure

- Step 1** Log in to a vCD organization through a vCD web client.
- Step 2** From **Catalogs > My Organization's Catalogs**, select a catalog.
- Step 3** Under the vApp Templates tab, click **Upload**
- The **Upload OVF package as a vApp Template** window appears.
- Step 4** In the OVF package field, click **Local file**. Click **Browse** and choose the ESC OVA file from the local system.
- Step 5** Click **Upload** to transfer the ESC OVA file into the catalog.

After uploading the ESC OVA file, you can access the ESC vApp template from the catalog.

Deploying ESC vApp on vCD Web Client

This section describes the deployment scenario for Elastic Services Controller (ESC) and the procedure to deploy ESC vApp on VMware vCloud Director (vCD).

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

To deploy ESC vApp on vCD web client, you must:

Procedure

-
- Step 1** Log in to a vCD organization through a vCD web client.
- Step 2** From **Catalogs > My Organization's Catalogs**, select a Catalog.
- Step 3** Select the ESC vApp template, right click and select **Add to My Cloud**.
- Step 4** To complete the deployment, you must specify the following in the **Add to My Cloud** wizard:
- Select Name and Location**- Specify **ESC** as the name of the vApp and select its virtual datacenter.
 - Configure Resources**- Specify **ESC** as the VM name and **SharedOnly** for storage policy.
 - Network Mapping**- Specify interface(s) network type as **MgtNetwork**, **SvcNetwork**, and **VnfNetwork**. IP allocation is **Static-Manual** for all types of network in an IPv4.
 - Custom Properties**- Fill in the custom properties for day(0) operation. **Note:** Do not fill or change anything in the **VM Settings of vCenter Server** section.
 - Customize Hardware**- Specify CPU use, memory consumption, and other core parameters.
 - Ready to Complete**- Check the **Power on vApp after this wizard is finished** on the bottom right corner of the wizard and click **Finish**.
-

Adding the VIM Connectors

For more information on configuring VIM connectors after installation and managing VIM connectors, see the Managing VIM Connectors chapter in the Cisco Elastic Services Controller User Guide.

Adding Self-Signed Certificate

If the vCD instance is using a self-signed certificate, use the following command to add the certificate into ESC, so that it qualifies the SSL verification:

```
$ escadm truststore add --help
usage: escadm truststore add [-h] [-v] --alias ALIAS --file FILE
```

optional arguments:

```
-h, --help          show this help message and exit
-v, --v, --verbose  show verbose output
--alias ALIAS
--file FILE
```




PART **IV**

Installing Cisco Elastic Services Controller on a Kernel-based Virtual Machine (KVM)

- [Installing Cisco Elastic Services Controller on a Kernel-based Virtual Machine, on page 87](#)



CHAPTER 16

Installing Cisco Elastic Services Controller on a Kernel-based Virtual Machine

This chapter describes how to install Cisco Elastic Services Controller on a Kernel-based Virtual Machine and includes the following sections:

- [Installing Cisco Elastic Services Controller in a Kernel-based Virtual Machine, on page 87](#)
- [Next Steps: Cisco Elastic Services Controller Kernel-based Virtual Machine, on page 90](#)

Installing Cisco Elastic Services Controller in a Kernel-based Virtual Machine

Cisco Elastic Services Controller can be installed in a Kernel-based Virtual Machine. You can install Cisco Elastic services controller in a Kernel-based Virtual Machine using libvirt.



Note For the KVM host to support UEFI, user may require to install "ovmf" package on the host. Command to install ovmf package differs based on the host operating system.

Preparing to Install Cisco Elastic Services Controller on a Kernel-based Virtual Machine

If you plan to run Cisco Elastic Services Controller on a kernel-based virtual machine, make sure the following are setup:

	Notes
Python 2.7 or 3.x	Installed by default on Linux
python-setuptools	Installed by default on Linux

	Notes
pip	<p>On RHEL 6.x/7.x:</p> <pre># easy_install pip</pre> <p>Since the installation using pip compiles source files, the gcc and python development packages are also required on RHEL. To install these packages on RHEL:</p> <pre># yum install gcc python-devel</pre> <p>On RHEL8.x:</p> <pre># yum install gcc</pre> <p>On Ubuntu: Installed by default. Since the installation using pip compiles source files, the gcc and python development packages are also required on Ubuntu. To install these packages on Ubuntu:</p> <pre># apt-get install python-dev</pre>
OpenStack clients	<pre># pip install python-keystoneclient # pip install python-cinderclient # pip install python-novaclient # pip install python-neutronclient</pre>
genisoimage	<p>On RHEL:</p> <pre># yum install genisoimage</pre> <p>On Ubuntu:</p> <pre># apt-get install genisoimage</pre>
libvirt and virtinst	<p>For Python 2</p> <p>On RHEL 6.x:</p> <pre># yum install libvirt-python python-virtinst</pre> <p>On RHEL 7.x:</p> <pre># yum install libvirt-python virt-install</pre> <p>On Ubuntu:</p> <pre># apt-get install libvirt-dev # pip install libvirt-python</pre> <p>For Python 3:</p> <p>On Ubuntu</p> <pre># sudo apt-get install -y python3-libvirt</pre> <p>On RHEL</p> <pre># yum install -y python3-libvirt</pre> <p>Virt-Manager</p> <pre>git clone https://github.com/virt-manager/virt-manager.git /usr/share/virt-manager-py3</pre>

	Notes
Other	<p>On RHEL 8.x</p> <pre># yum install libcurl-devel # yum install openssl-devel # pip3 install pycurl # pip3 install urlgrabber</pre>



Note libvirt will create the default network automatically.

Installing Elastic Services Controller on a Kernel-Based Virtual Machine

To install standalone Elastic Services Controller (ESC) on a kernel-based virtual machine, do the following:

Procedure

- Step 1** Load the variables from the openrc file that contains OpenStack credentials:
- ```
cat ./openrc.sh
export OS_TENANT_NAME='<OS tenant username>'
export OS_USERNAME='<OS username>'
export OS_PASSWORD='<OS password>'
export OS_AUTH_URL='http://<Openstack Host>:5000/v2.0/'

source ./openrc.sh
```
- Step 2** Copy the ESC qcow2 image and the bootvm.py into the kernel-based VM.
- Step 3** Boot ESC on a kernel-based VM on the default network that was created when libvirt was installed, use one of the following command:
- ```
./bootvm.py --user_pass <username>:<password> --user_confid_pass <username>:<password>
--libvirt --image <image_name> esc-vm --net <default network>
```
- Step 4** Boot ESC on a kernel-based VM on the default network with static IP, using the following command:
- ```
./bootvm.py --user_pass <username>:<password> --user_confid_pass <username>:<password>
--libvirt --image <image_name> esc-vm --net <network> --ipaddr <ip_address>
```
- Step 5** Get a list of used IP addresses in your network. Use IP addresses that are not in the list for both HA Active/Standby bootvm.py command and for kad\_vip. Determine the first 3 octets of your network (i.e. 192.168.122) and pass it in the below command :
- ```
arp -an | grep 192.168.122
```
- Step 6** To install ESC on a kernel-based VM in high availability, use the following command twice for both the HA nodes:
- Note** For the second bootvm.py command, use the other HA instance name.

```
./bootvm.py --user_pass <username>:<password> --user_confd_pass <username>:<password>
--libvirt --image <image_name> --ha_mode drbd --gateway_ip <default_gateway_ip_address>
--ipaddr <ip_address>
--ha_node_list <ha peer ip addresses separated by comma> --kad_vip <vip address> esc-ha-1
--net <network>
```

Next Steps: Cisco Elastic Services Controller Kernel-based Virtual Machine

Logging in to Cisco Elastic Services Controller Portal

Verifying ESC installation for a Kernel-based Virtual Machine (KVM)

After deploying ESC on a Kernel-based virtual machine, use the following procedure to verify the deployment.

Procedure

Step 1 Check that the ESC VMs have booted using the following command:

```
$ virsh list
```

Step 2 Get the IP address of the ESC VM, using the following command:

```
$ arp -an | grep <ip_address>
```

Step 3 Connect to ESC using SSH and verify the processes are running:

```
$ ssh USERNAME@ESC_IP
```

Troubleshooting Tips

When SSH access is not available, due to network conditions or ESC startup failures, you can connect to ESC through console(if enabled in ESC VM image) or VNC access. To access ESC VM through VNC, do the following:

1. Identify the vnc port.

```
virsh dumpxml 10 | fgrep vnc
```

2. Create a ssh tunnel to the local vnc port to allow connection from your remote VNC client.



PART **V**

Upgrading D-MONA

- [Upgrading D-MONA, on page 93](#)



CHAPTER 17

Upgrading D-MONA

- [Upgrading D-MONA, on page 93](#)

Upgrading D-MONA

ESC allows upgrading the D-MONA instance to a newer version starting from ESC 5.7 release. Use the following procedure to upgrade the D-MONA:

- Ensure the `<image>` tag in the XML file refers to the newer version of the D-MONA image.
- Ensure the respective D-MONA image is registered in the Virtual Infrastructure Manager (OpenStack).

Now, execute following command:

```
esc_nc_cli edit-config <dmona.xml>
```

Upon successful upgrade, VM_UPDATED and SERVICE_UPDATED notifications are generated by ESC. For example:

```
2022-03-02 15:47:24.948 INFO ===== SEND NOTIFICATION STARTS =====
2022-03-02 15:47:24.948 INFO Type: VM_UPDATED
2022-03-02 15:47:24.948 INFO Status: SUCCESS
2022-03-02 15:47:24.948 INFO Status Code: 200
2022-03-02 15:47:24.948 INFO Status Msg: VM image Updated. VM name:
[dmona2_Grp1_0_6c14ce01-82d4-40ca-a8bb-dce97f99eeae], new Image: [ESC-5_7_0_64]
2022-03-02 15:47:24.948 INFO Tenant: admin
2022-03-02 15:47:24.948 INFO Deployment ID: 47831c9f-2323-431f-b7ea-748891d23430
2022-03-02 15:47:24.949 INFO Deployment name: dmona2
2022-03-02 15:47:24.949 INFO VM group name: Grp1
2022-03-02 15:47:24.949 INFO User configs: 1
2022-03-02 15:47:24.949 INFO VM Source:
2022-03-02 15:47:24.949 INFO VM ID: 0f5b4ba3-bf05-4cd8-bde8-1604102ef261
2022-03-02 15:47:24.949 INFO VM Name: dmona2_Grp1_0_6c14ce01-82d4-40ca-a8bb-dce97f99eeae
2022-03-02 15:47:24.949 INFO VM Name (Generated):
dmona2_Grp1_0_6c14ce01-82d4-40ca-a8bb-dce97f99eeae
2022-03-02 15:47:24.949 INFO Host ID: dab2a418b071996c15dc91ec9f11b925890028e65ce7314aae40640a
2022-03-02 15:47:24.950 INFO Host Name: my-ucs-54
2022-03-02 15:47:24.950 INFO ===== SEND NOTIFICATION ENDS =====
2022-03-02 15:47:25.113 INFO
2022-03-02 15:47:25.113 INFO ===== SEND NOTIFICATION STARTS =====
2022-03-02 15:47:25.113 INFO Type: SERVICE_UPDATED
2022-03-02 15:47:25.114 INFO Status: SUCCESS
2022-03-02 15:47:25.114 INFO Status Code: 200
2022-03-02 15:47:25.114 INFO Status Msg: Service group update completed successfully
2022-03-02 15:47:25.114 INFO Tenant: admin
```

```
2022-03-02 15:47:25.114 INFO Deployment ID: 47831c9f-2323-431f-b7ea-748891d23430
2022-03-02 15:47:25.114 INFO Deployment name: dmona2
2022-03-02 15:47:25.115 INFO ===== SEND NOTIFICATION ENDS =====
```

On Success:

```
VM_UPDATED -200 - success message
SERVICE_UPDATED -200 - success message
```

On Failure:

```
VM_UPDATED -500 - failure message
SERVICE_UPDATED -500 - failure message
```

If the upgrade fails, then repeat the same steps mentioned previously to downgrade D-MONA to previous version.



Note

- D-MONA upgrade works only with OpenStack as Virtual Infrastructure Manager.
- The upgrade option is only applicable for D-MONA and not applicable for other VNFs.

For example, to upgrade D-MONA version to ESC-a_b_c_d image, update the image name in the XML file and use the following command to upgrade the D-MONA:

```
esc_nc_cli edit-config <dmona.xml>
```

```
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
  <tenants>
    <tenant>
      <name>admin</name>
      <deployments>
        <deployment>
          <name>dmona2</name>
          <vm_group>
            <name>Grp1</name>
            <image>ESC-a_b_c_d</image>
            <flavor>m1.medium</flavor>
            <bootup_time>600</bootup_time>
            <recovery_wait_time>0</recovery_wait_time>
            <recovery_policy>
              <recovery_type>AUTO</recovery_type>
              <max_retries>1</max_retries>
              <action_on_recovery>REDEPLOY_ONLY</action_on_recovery>
            </recovery_policy>
          </vm_group>
          <interfaces>
            <interface>
              <nicid>0</nicid>
              <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>
                <type>HTTPGET</type>
                <nicid>0</nicid>
              </metric_collector>
            </kpi>
          </kpi_data>
        </deployment>
      </deployments>
    </tenant>
  </tenants>
</esc_datamodel>
```

```

    <poll_frequency>5</poll_frequency>
    <polling_unit>seconds</polling_unit>
    <continuous_alarm>false</continuous_alarm>
    <properties>
      <property>
        <name>protocol</name>
        <value>https</value>
      </property>
      <property>
        <name>port</name>
        <value>8443</value>
      </property>
      <property>
        <name>path</name>
        <value>mona/v1/health/status</value>
      </property>
    </properties>
  </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>--user-data</dst>
    <file>file:///opt/cisco/esc/esc-config/dmona/user-data.template</file>
    <variable>
      <name>vm_credentials</name>
      <val>$$rounds=4096$LRNEIVZu5o$HcPKqABon6RvnS705M7J212xuFWCL7xOjUY5ST.0e7oIhvfPqicNoVEMZYkw06GRIBVYRnc/wGdp2Z5KtgX/</val>
    </variable>
  </configuration>
  <configuration>
    <dst>/opt/cisco/esc/mona/dmona.crt</dst>
    <data>$DMONA_CERT</data>
  </configuration>
  <configuration>
    <dst>/opt/cisco/esc/mona/config/application-dmona.properties</dst>
    <file>file:///opt/cisco/esc/esc-config/dmona/application-dmona.template</file>
  </configuration>
  <variable>
    <name>monitoring.agent</name>
    <val>true</val>
  </variable>
  <variable>
    <name>security_basic_enabled</name>
    <val>true</val>
  </variable>
  <variable>
    <name>security_user_name</name>
    <val>dmona</val>
  </variable>
  <variable>
    <name>security_user_password</name>
    <val>defaultUser</val>
  </variable>

```

```

    <variable>
      <name>monitoring.agent.instance.name</name>
      <val>D-MONA-OTTAWA</val>
    </variable>
    <variable>
      <name>monitoring.agent.instance.id</name>
      <val>D-MONA-123</val>
    </variable>
    <variable>
      <name>monitoring.agent.api.key</name>
      <val>233</val>
    </variable>
    <variable>
      <name>notification.url</name>
      <val> https://<ESC_VM_IP:8443/ESCManager/v0/api/events/notif</val>
    </variable>
    <variable>
      <name>monitoring.agent.vim.mapping</name>
      <val>>false</val>
    </variable>
  </configuration>
</config_data>
</vm_group>
</deployment>
</deployments>
</tenant>
</tenants>
</esc_datamodel>

```

Upon completion of the D-MONA upgrade, that is, after receiving the SERVICE_UPDATED successful notification, the agent is rebuilt (*using the Compute API up to microservice version 2.56*) with the new image and the central ESC repopulates it with all its monitoring rules so that monitoring of VNFs continues where it was prior to the upgrade.



Note Whilst the D-MONA agent upgrade is in progress, the VNFs it is responsible from monitoring is not monitored. If you need to continue monitoring for these VNFs then consider migrating monitoring to another monitoring agent (D-MONA or central MONA).



PART VI

Post Installation Tasks

- [Post Installation Tasks, on page 99](#)



CHAPTER 18

Post Installation Tasks

This chapter contains the following sections:



Note It is recommended that you ignore the `do not edit` message, if you want to modify the `cloud-init day-0` configuration, file.

- [ESC Lifecycle Notifications During Login, on page 99](#)
- [Changing the ESC Password , on page 100](#)
- [Configuring Pluggable Authentication Module \(PAM\) Support for Cisco Elastic Services Controller, on page 105](#)
- [Configuring Cisco Elastic Services Controller as Identity Management Client, on page 107](#)
- [Authenticating REST Requests, on page 109](#)
- [Configuring Openstack Credentials , on page 112](#)
- [Enabling Barbican Client in ESC, on page 117](#)
- [Staging VPC Volume , on page 117](#)
- [Running MONA in a Root Jail, on page 118](#)
- [Installing the MONA Root Jail, on page 118](#)
- [Reconfiguring ESC Virtual Machine, on page 119](#)
- [Verifying ESC Configurations and Other Post-Install Operations , on page 122](#)
- [Logging in to the ESC Portal, on page 124](#)

ESC Lifecycle Notifications During Login

When logging in to ESC through SSH, the message of the day may include an End-of-Life (EoL) or End-of-Support (EoS) notice. Note that each version of ESC is EoL on the release date of the next version, however this does not mean that the software is now defunct but rather that Cisco are providing a newer version. In fact, as the message advises, support and releases of software maintenance patches continue as normal. For example:

```
$ ssh admin@ESC_IP
```

```
*** ESC NOTICE: ESC 5.7 EOL on 29 July 2022 ***
```

```
*** Please note that this version continues to be supported until 29 January 2026 ***
```

```
admin@ESC_IP$
```

A further notice appears when the EoS date is 30 days away:

```
*** ESC NOTICE: Support for ESC 5.7 ends on 29 January 2026 ***
*** Please contact your Cisco Account Manager for details on how to upgrade ***
```

These notices ensure that you upgrade in a timely manner and have the latest feature set and security patches installed.

Changing the ESC Password

You will be forced to change the default password on the first time login. Portal will not let you bypass this step and will keep returning you to this page until you change the default password. After the first time password change, you can change your password using the procedures described in this section. Also, if the user has multiple browsers or tabs or the SAME user is logged on by 2 or more computers and one of the user changes the password then everyone will be logged off and asked to re-enter the new password. The user session has an expiry of 1 hour so if the user is inactive on the portal for an hour then portal will expire the session and the user will have to re-login. If you forgot your password, you can also update or randomly generate the password.

This section discusses how to change the passwords.

Example for REST:

```
sudo escadm rest set --username {USERNAME} --password {PASSWORD}
```

Example for ETSI:

```
sudo escadm etsi set --rest_user {USERNAME:PASSWORD}
```

Changing the ConfD Netconf/CLI Administrator Password Using the Command Line Interface

After you install ESC, to change the ConfD admin password, do the following:

You cannot execute the confd commands, such as `confd_cli`. The `confd_cli -u admin` is replaced with the `ssh admin@localhost -p 2024` command.

For information on Installing ESC, see [Installing Cisco Elastic Services Controller Using the QCOW Image](#).

To access the confD cli for an admin account:

```
admin@esc$ ssh admin@localhost -p 2024
admin@localhost's password: *****

admin connected from 127.0.0.1 using ssh on esc
admin@esc>
```

Procedure

Step 1 Log in to the ESC VM.

```
$ ssh USERNAME@ESC_IP
```

Step 2 Switch to the admin user.

```
[admin@esc-ha-0 esc]$ sudo bash
[sudo] password for admin:
```


Step 3 Load the ConfD CLI:

```
$ /opt/cisco/esc/confd/bin/ssh admin@localhost -p 2024
```

Step 4 Set the new admin password:

```
$ configure
$ set aaa authentication users user admin password <new password>
```

Step 5 Save the changes.

```
$ commit
```

Creating Readonly User Group for ConfD in ESC

ConfD in ESC is enhanced with the introduction of a new group named `readonly`. If you are a member of `readonly` group, you can only retrieve the information and you cannot modify the permissions.

You can use `'readonly'` as the role name with `bootvm`. The following example shows how to create two users in ConfD. One is `admin` and the other is `readonly`:

```
# bootvm.py name=500-105-100 --user_confid_pass admin:admin --user_confid_pass
readonly:readonly::readonly --user_pass admin:admin --image ESC-5_0_0_105 --net network
```

For HA A/A, you can use `'readonly'` as the group name in `aa-day0.yaml` Following is the example:

```
confd:
  init_aaa_users:
    - group: readonly
      name: admin
      passwd: $6$rounds=4096$Ps1JIjKihRTF$fo8XPBxwEHJWWfNiXDnO269rlhAxAhWBc
PBfGnZxy1gM3QMxcN8jJ6guWt9Bu.ZkWdPt3hr0Ogh073Wr3iDHb0
```

You can also create a `confd` `readonly` user after ESC vm is deployed. The following steps create a `confd` `readonly` user named `'test'` with password `'test'`:

```
[root@name-500-155 admin]# /opt/cisco/esc/confd/bin/ssh admin@localhost -p 2024
admin connected from 127.0.0.1 using console on name-500-155
admin@name-500-155> configure
Entering configuration mode private
[ok][2019-12-06 18:17:39]
[edit]
admin@name-500-155% set aaa authentication users user test uid 9000 gid 9000 password $0$test
  homedir /var/confd/homes/test ssh_keydir /var/confd/homes/test/.ssh
[ok][2019-12-06 18:19:15]
[edit]
admin@name-500-155% set nacm groups group readonly user-name test
[ok][2019-12-06 18:19:41]
[edit]
admin@name-500-155% commit
Commit complete.
[ok][2019-12-06 18:19:47]
[edit]
admin@name-500-155%
```

As a `readonly` user, you can also access ConfD remotely:

```
name@my-server-39:~$ ssh -p 2024 readonly@172.29.0.57
readonly@172.29.0.57's password:
readonly connected from 172.16.103.46 using ssh on name-500-156
readonly@name-500-156> configure
Entering configuration mode private
```

```
[ok][2019-12-13 16:15:33]
[edit]
readonly@name-500-156% show esc_datamodel
tenants {
    tenant admin {
        description      "Built-in Admin Tenant";
        managed_resource false;
        vim_mapping       true;
    }
}
[ok][2019-12-13 16:15:38]
[edit]
```

ESC in ConfD sends access-denied error if you fall under readonly ConfD group, and require modify permissions. Following is the example of the access-denied error message:

```
$ esc_nc_cli --user readonly --password ***** edit-config dep.xml
Configure
/opt/cisco/esc/confd/bin/netconf-console --port=830 --host=127.0.0.1 --user=readonly
--password=***** --edit-config=/tmp/d.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <rpc-error>
    <error-type>application</error-type>
    <error-tag>access-denied</error-tag>
    <error-severity>error</error-severity>
  </rpc-error>
</rpc-reply>
```

If ESC is configured to use PAM/IDM. The groups in IDM servers are directly mapped to the groups in ConfD. Hence, the readonly user must be mapped in the IDM group 'readonly'.

For example:

```
$ ipa group-find --all readonly
-----
1 group matched
-----
dn: cn=readonly,cn=groups,cn=accounts,dc=linuxsysadmins,dc=local
Group name: readonly
GID: 5003
Member users: readonly
ipantsecurityidentifier: S-1-5-21-2222126199-2113948134-574478857-1003
ipauniqueid: 858b8cda-0d34-11ea-bca8-525400b29c19
objectclass: top, groupofnames, nestedgroup, ipausergroup, ipaobject, posixgroup,
ipantgroupattrs
-----
Number of entries returned 1
-----
```

Restricting ESC ConfD CLI access

Access to the ESC ConfD CLI is disabled by default for new ESC deployments starting from ESC 5.6 release, and also to the upgrades from earlier version of ESC to ESC 5.6 version. This restriction is introduced as a level of protection from user using the ConfD CLI command without knowing its implications.

When ConfD CLI access is enabled, you can login to the ConfD CLI using one of the following ways:

- `esc_nc_cli`

```
[admin@esc-test1606-confd-instance ~]$ esc_nc_cli cli
ssh -o StrictHostKeyChecking=no -p 2024 admin@127.0.0.1
admin@127.0.0.1's password:

admin connected from 127.0.0.1 using ssh on esc-test1606-confd-instance.novalocal
admin@esc-test1606-confd-instance>
```

- `confd_cli`

```
[admin@esc-test1606-confd-instance ~]$ sudo -i

#####
#           ESC on esc-test1606-confd-instance.novalocal
#####

[root@esc-test1606-confd-instance ~]#
[root@esc-test1606-confd-instance ~]# source /opt/cisco/esc/confd/confdrc
[root@esc-test1606-confd-instance ~]# confd_cli -u admin -C

admin connected from 127.0.0.1 using console on esc-test1606-confd-instance.novalocal
esc-test1606-confd-instance#
```

Access to the ESC ConfD CLI using `esc_nc_cli` or `confd_cli` commands is disabled by default starting ESC 5.6 release. Use the following commands to enable or disable the access to ConfD CLI:

1. Command to enable ConfD CLI access

```
esc_nc_cli cli enable
```



Note Once the ConfD CLI access is enabled, access privilege will be purely based on Network Configuration Access Control Model(NACM) rules list defined in ESC. Refer RFC8341 for more details on NACM.

2. Command to disable ConfD CLI access

```
esc_nc_cli cli disable
```

Changing Linux Account Password

Procedure

Step 1 Log in to ESC VM.

```
$ ssh USERNAME@ESC_IP
```

Step 2 To update or generate a random password, use the following command:

```
/usr/bin/pwqcheck
/usr/bin/pwqgen
```

Changing the ESC Portal Password

The user can update or reset the default admin password.

Procedure

Step 1 Log in to ESC VM.

Step 2 Switch to the root user.

Step 3 To update the default admin password or randomly generate a password, use one of the following method:

- Using escadm utility:

To update the default admin password (admin/*****):

```
[root@anyname-v44-52 admin]# sudo escadm portal set --username admin --password *****
Successfully updated password for username admin
```

To generate a random password:

```
[root@anyname-v44-52 admin]# sudo escadm portal set --username admin
Would you like to use the generated password: "Accent5omit&Wide"?[y|n]y
Successfully updated password for username admin
```

The `--must_change` variable will ask the user to change their password at the next login.

The `--must_change` variable is not applicable for REST users.

```
[root@anyname-v44-52 admin]# sudo escadm portal set --username admin --must_change
Would you like to use the generated password: "Rainy4Dozen&Behave"?[y|n]y
Successfully reset password for username admin. User must change the password at the
next login.
```

- To reset to a specific password:

```
[root@anyname-v44-52 admin]# sudo escadm portal set --username admin --password *****
--must_change
Successfully reset password for username admin. User must change the password at the
next login.
```

- Using the bootvm command line:

```
--user_portal_pass admin:<new password>
```

- Using the ESC Portal:

- Log in to ESC portal using your username and password.
- Choose **Accounts Setting** on the Navigation menu.
- Enter the old password in the Old password field, then enter a new password in the New Password and Confirm Password fields.
- Click **Update Password**.

Configuring Pluggable Authentication Module (PAM) Support for Cisco Elastic Services Controller

You can configure the ESC services to use Pluggable Authentication Modules (PAM) for user authentication in ESC. With Cisco Elastic Services supporting PAM, you can also enable LDAP authentication in ESC. If PAM is not configured, ESC will continue to use the default authentication method for each ESC service. The following table lists the commands to enable PAM authentication for each ESC service.

Table 3: Configuring PAM for ESC Services

ESC Service/Component	Command to configure PAM authentication
ESCManager (REST interface)	<code>sudo escadm escmanager set --auth PAM:<pam_service_name></code>
ESC Monitor (Health API)	<code>sudo escadm monitor set --auth PAM:<pam_service_name></code>
Confd	<code>sudo escadm confd set --auth PAM:<pam_service_name></code>
Portal	<code>sudo escadm portal set --auth PAM:<pam_service_name></code>
ETSI	<code>sudo escadm etsi set --pam_service <pam_service_name></code>



Note

- The SSHD service that runs inside the ESC VM already uses PAM authentication by default.
- If any component sets PAM authentication without specifying the PAM service, ESC defaults to the PAM service 'system-auth'.

PAM Authentication Service Configurations and User Groups

Each ESC service (listed above) has an associated PAM authentication configuration and a user group to provide specific access control. The user group is defined in the `/etc/group` file. The admin user is a member of all the groups.

Table 4: PAM Authentication Service Configurations and User Groups

<code>/etc/group</code>	<code>/etc/pam.d</code>
<code>portal-user:x:1002:admin</code>	<code>portal-auth</code>
<code>rest-user:x:1003:admin</code>	<code>rest-auth</code>
<code>confd-user:x:1004:admin</code>	<code>confd-auth</code>

/etc/group	/etc/pam.d
etsi-user:x:1005:admin	etsi-auth
health-user:x:1006:admin	health-auth

For example, to configure PAM authentication for the health API and restrict access only to the health user group, run the following command:

```
$ sudo escadm monitor set --auth PAM:health-auth
ESC configuration was changed and saved automatically. They will take effect once you restart
ESC service by running "sudo escadm restart"
```

For adding PAM user to ESC component, see [Adding PAM User to an ESC Service/Component, on page 106](#).

Adding PAM User to an ESC Service/Component

You can add the PAM user to the following groups of ESC service:

- rest-user
- confd-user
- portal-user
- etsi-user
- health-user

Perform the following steps to add the PAM user to an ESC service/component:

Procedure

-
- Step 1** Log in to the ESC VM.
- Step 2** Add a PAM user, by using the following command:
- ```
sudo passwd pamuser
Changing password for user pamuser.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```
- Step 3** Add a PAM user to an ESC service/component group, by using the following command:
- ```
sudo usermod -a -G <ESC Service Group> pamuser
```

Note The PAM user must be added to the admin or readonly group for the Confd service

Configuring Cisco Elastic Services Controller as Identity Management Client

Prerequisites

- Ensure that the Identity Management Client (IDM) server is up and running.
- Ensure that the DNS server in ESC is in working state. So that, ESC interacts with the IDM server using the host name.

The following example shows how ESC (esc-client-500.linuxsysadmins.local) reaches the IDM server (idmns.linuxsysadmins.local).

```
[root@esc-client-500 admin]# ping idmns
PING idmns.linuxsysadmins.local (192.168.222.176) 56(84) bytes of data.
64 bytes from idmns.linuxsysadmins.local (192.168.221.176): icmp_seq=1 ttl=64 time=0.492 ms
64 bytes from idmns.linuxsysadmins.local (192.168.221.176): icmp_seq=2 ttl=64 time=0.457 ms
64 bytes from idmns.linuxsysadmins.local (192.168.221.176): icmp_seq=3 ttl=64 time=0.645 ms
```

You can configure IDM through sssd. ESC provides a PAM configuration file, point to /etc/pam.d/system-auth to system-auth-esc-sssd to start configuring services in ESC to work with the IDM server.

```
# cd /etc/pam.d
# ln -sf system-auth-esc-sssd system-auth
# ls -al /etc/pam.d/system-auth
lrwxrwxrwx. 1 root root 20 Nov 13 00:39 /etc/pam.d/system-auth -> system-auth-esc-sssd
```

The following table lists the commands to enable IDM authentication for each ESC service.

Table 5: Configuring IDM for ESC Services

ESC Service/Component Command	Command
ESCManager	# sudo escadm escmanager set --auth PAM:system-auth-esc-sssd
ETSI	# sudo escadm etsi set --pam_service system-auth-esc-sssd
ConfD	# sudo escadm confd set --auth PAM:system-auth-esc-sssd

Configuring Cisco Elastic Services Controller as the Identity Policy and Audit Client

To Configure the ESC as an Identity Policy and Audit Client (IPA) client, run the following command:

```
ipa-client-install
```

Following is the example to configure ESC as the IPA client:

```
[root@esc-client-500 admin]# ipa-client-install --domain linuxsysadmins.local --server
idmns.linuxsysadmins.local --realm LINUXSYSADMINS.LOCAL
WARNING: ntpd time&date synchronization service will not be configured as
conflicting service (chronyd) is enabled
Use --force-ntpd option to disable it and force configuration of ntpd

Autodiscovery of servers for failover cannot work with this configuration.
If you proceed with the installation, services will be configured to always access the
discovered server for all operations and will not fail over to other servers in case of
failure.
Proceed with fixed values and no DNS discovery? [no]: yes
Client hostname: esc-client-500.linuxsysadmins.local
Realm: LINUXSYSADMINS.LOCAL
DNS Domain: linuxsysadmins.local
IPA Server: idmns.linuxsysadmins.local
BaseDN: dc=linuxsysadmins,dc=local

Continue to configure the system with these values? [no]: yes
Skipping synchronizing time with NTP server.
User authorized to enroll computers: admin
Password for admin@LINUXSYSADMINS.LOCAL:
Successfully retrieved CA cert
    Subject:      CN=Certificate Authority,O=LINUXSYSADMINS.LOCAL
    Issuer:       CN=Certificate Authority,O=LINUXSYSADMINS.LOCAL
    Valid From:   2019-11-12 23:23:32
    Valid Until:  2039-11-12 23:23:32

Enrolled in IPA realm LINUXSYSADMINS.LOCAL
Created /etc/ipa/default.conf
Configured sudoers in /etc/nsswitch.conf
Configured /etc/sss/sss.conf
Configured /etc/krb5.conf for IPA realm LINUXSYSADMINS.LOCAL
trying https://idmns.linuxsysadmins.local/ipa/json
[try 1]: Forwarding 'schema' to json server 'https://idmns.linuxsysadmins.local/ipa/json'
trying https://idmns.linuxsysadmins.local/ipa/session/json
[try 1]: Forwarding 'ping' to json server
'https://idmns.linuxsysadmins.local/ipa/session/json'
[try 1]: Forwarding 'ca_is_enabled' to json server
'https://idmns.linuxsysadmins.local/ipa/session/json'
Systemwide CA database updated.
Adding SSH public key from /etc/ssh/ssh_host_ecdsa_521_key.pub
Adding SSH public key from /etc/ssh/ssh_host_ecdsa_384_key.pub
Adding SSH public key from /etc/ssh/ssh_host_rsa_key.pub
Adding SSH public key from /etc/ssh/ssh_host_ed25519_key.pub
Adding SSH public key from /etc/ssh/ssh_host_ecdsa_key.pub
[try 1]: Forwarding 'host_mod' to json server
'https://idmns.linuxsysadmins.local/ipa/session/json'
Could not update DNS SSHFP records.
SSSD enabled
Configured /etc/openldap/ldap.conf
Configured /etc/ssh/ssh_config
Configured /etc/ssh/sshd_config
Configuring linuxsysadmins.local as NIS domain.
Client configuration complete.
The ipa-client-install command was successful
```


Authenticating REST Requests

ESC REST and ETSI REST APIs use http basic access authentication where the ESC client will have to provide a username and password when making REST requests. The user name and password will be encoded with Base64 in transit, but not encrypted or hashed. HTTPS will be used in conjunction with Basic Authentication to provide the encryption.

This section discusses ESC REST and ETSI REST authentications, how to change the default password of the interfaces, and how to send an authorized requests from the ESC client.

REST Authentication

By default, the REST authentication is enabled. To disable REST authentication, you can pass the argument **--disable-rest-auth** to `bootvm`. Cisco does not recommend you to use this in a production environment.

ESC also supports https communication over port 8443. ESC will generate a self-signed certificate that the client will need to trust to get the https communication going. By default, REST is enabled as HTTP and restricted to localhost.

ESC can be installed with external access to REST on HTTPS or HTTP enabled with additional `bootvm.py` arguments: **--enable-https-rest** or **--enable-http-rest**.

It is recommended to use only enabled external REST API when required. When enabled, it is recommended to use **bootvm.py --enable-https-rest --user_rest_pass USERNAME:PASSWORD**.



Note Make sure to pass either **--enable-https-rest** or **--enable-http-etsi-rest** or both the arguments to the `bootvm.py` script to enable http and https interfaces to the REST API. You must pass **--user_rest_pass** while using either **--enable-https-rest** or **--enable-http-etsi-rest** when REST authentication is not disabled. To enable https or http after ESC VM is booted, use the `escadm` command specified below.

```
sudo escadm escmanager set --url
http://127.0.0.1:8080/ESCManager,https://0.0.0.0:8443/ESCManager
```

You must change the configuration of the peer instance if ESC is in HA Active/Standby mode.

Enabling ETSI REST Authentication

If the ETSI REST http or https interfaces are enabled, then all requests to an ETSI API must contain authentication data. You can use the **--enable-http-etsi-rest** or **--enable-https-etsi** arguments respectively to enable http and https interfaces to the ESC `bootvm.py` installation script.

You can enable both interfaces simultaneously, but only the https interface should be enabled in a production environment.



Note To enable http or https after the ESC VM has booted, use the escadm command specified below:

```
sudo escadm etsi enable_http_rest
```

OR

```
sudo escadm etsi enable_https_rest
```

Then restart the ETSI service.

Changing the REST Interface Password

The REST interface has only one default username/password (admin/<default_password>). The password can be updated after the bootup using escadm tool from the ESC VM CLI. You can also update the password through the REST API.

Procedure

Step 1 Log in to ESC VM.

Step 2 To replace the existing password with a new one, use one of the below options:

- Using the escadm tool from the ESC VM CLI, you can generate a random password:

```
[root@test-v44-52 admin]# escadm rest set --help
usage: escadm rest set [-h] [-v] --username USERNAME [--password PASSWORD]

optional arguments:
  -h, --help            show this help message and exit
  -v, --v, --verbose    show verbose output
  --username USERNAME   new password or use randomly generated password if no
  --password PASSWORD   password provided
```

- Using the REST API:

```
http://[ESCVM_IP]:8080/ESCManager/v0/authentication/setpassword?userName=admin&password=yourPassword
or
https://[ESCVM_IP]:8443/ESCManager/v0/authentication/setpassword?userName=admin&password=yourPassword
```

Changing the ETSI REST Interface Password

The ETSI REST interface has only one default username/password (admin/<default_password>). The password can be updated after the bootup using escadm tool from the ESC VM CLI.

Procedure

Step 1 Log in to ESC VM.

Step 2 To set the default ETSI REST username and password, use the following command:

```
sudo escadm etsi set --rest_user username:password
```

or

```
[admin@xyz-esc-4-4-0-59-keep ~]$ escadm etsi set --help
usage: escadm etsi set [-h] [-v] [--startup {0,1,true,false,manual,auto}]
[--rest_user REST_USER] [--pam_service PAM_SERVICE]
```

optional arguments:

```
-h, --help show this help message and exit
-v, --v, --verbose show verbose output
--startup {0,1,true,false,manual,auto}
set to false|0|manual to disable etsi at startup.
--rest_user REST_USER
Set the user for rest. Format username:password
--pam_service PAM_SERVICE
Specify a PAM service to use for authentication. This
will override the rest user. To revert to the using
the rest user for authentication, supply an empty
string.
```

Sending an Authorized REST Request

To send an authorized request, the ESC client should send the request with the following header:

```
Authorization: Basic YWRtaW46Y2lzY28xMjM=
```

where *YWRtaW46Y2lzY28xMjM=* is the Base64 encoded string of the default username/password.

Most libraries and web clients have an interface for providing the username/password and the application will encode the username/password and add the HTTP Basic Auth header.

Example using the default credentials:

For HTTP:

```
http://[ESCV_M_IP]:8080/ESCManager/v0/tenants/
```

For HTTPS:

```
https://[ESCV_M_IP]:8443/ESCManager/v0/tenants/
```

Sending an Authorized ETSI REST Request

To send an authorized request, the ESC client should send the request with the following header:

```
Authorization: Basic YWRtaW46Y2lzY28xMjM=
```

where *YWRtaW46Y2lzY28xMjM=* is the Base64 encoded string of the default username/password.

Most libraries and web clients have an interface for providing the username/password and the application will encode the username/password and add the HTTP Basic Auth header.

Example using the default credentials:

For HTTP:

```
http://[ESCV_M_IP]:8250/vnflcm/v1/vnf_lcm_op_occs
```

For HTTPS:

```
https://[ESCV_M_IP]:8251/vnflcm/v1/vnf_lcm_op_occs
```

Configuring Openstack Credentials

If ESC was deployed without passing VIM credentials, you can set VIM credentials through ESC VIM and through VIM User APIs (REST or Netconf API).



Note ESC will accept the northbound configuration request only if the following conditions are met:

- ESC has VIM or a VIM user configured through APIs (REST/Netconf).
- ESC has VIM or a VIM user configured, and ESC is able to reach the VIM.
- ESC has VIM or a VIM user configured, and ESC is able to authenticate the user.

Configuring using Netconf API

• Passing VIM credential using Netconf :

```
<esc_system_config xmlns="http://www.cisco.com/esc/esc">
  <vim_connectors>
    <!--represents a vim-->
    <vim_connector>
      <!--unique id for each vim-->
      <id>my-server-30</id>
      <!--vim type [OPENSTACK|VMWARE_VSPHERE|LIBVIRT]-->
      <type>OPENSTACK</type>
      <properties>
        <property>
          <name>os_auth_url</name>
          <value>http://<os_ip:port>/v3</value>
        </property>
        <!-- The project name for openstack authentication and authorization -->
        <property>
          <name>os_project_name</name>
          <value>vimProject</value>
        </property>
        <!-- The project domain name is needed for openstack v3 identity api -->
        <property>
          <name>os_project_domain_name</name>
          <value>default</value>
        </property>
      </properties>
    </vim_connector>
  </vim_connectors>
  <users>
    <user>
      <id>admin</id>
      <credentials>
        <properties>
          <property>
            <name>os_password</name>
            <value>*****</value>
          </property>
          <!-- The user domain name is needed for openstack v3 identity api -->
          <property>
            <name>os_user_domain_name</name>
            <value>default</value>
          </property>
        </properties>
      </credentials>
    </user>
  </users>
</esc_system_config>
```

```

        </property>
      </properties>
    </credentials>
  </user>
</users>
</vim_connector>
</vim_connectors>
</esc_system_config>

```

**Note**

- From ESC 3.0 onwards, multiple VIM connectors are supported but within one ESC, only one type of VIM are supported. For example, all the Vim Connector(s) has to be for OpenStack only. One ESC VIM cannot have two VIM connector, one points to OpenStack, one points to VMware.
- One VIM is chosen as the default VIM which supports all pre 3.0 config requests and datamodels.
- Deployments can be done on the VIM that is not the default VIM. The deployment to a non default VIM has to have all out-of-band resources (except ephemeral volumes). No other configurations like image, flavor, network, and so on can be done on the VIM that is not the default VIM.
- The default VIM connector will be auto provisioned and does not need to be configured in the following scenarios:
 - If VIM credentials have been passed during ESC boot up.
 - If upgrading from 2.3.x to 3.0.
- The change in the datamodel for Openstack create VIM connector would be handled during upgrade by migration. The 'os_tenant_name' and 'os_project_domain_name' properties would be moved to the VIM Connector properties and 'os_tenant_name' will be renamed to 'os_project_name'.
- For the default VIM Connector, once it is properly authenticated, its properties cannot be updated.
- VIM user can be deleted, recreated, or its properties can be updated at anytime.

• Updating VIM Connector using Netconf:

```

<esc_system_config xmlns="http://www.cisco.com/esc/esc">
  <vim_connectors>
    <vim_connector nc:operation="replace">
      <id>example_vim</id>
      <type>OPENSTACK</type>
      <properties>
        <property>
          <name>os_auth_url</name>
          <value>{auth_url}</value>
        </property>
        <property>
          <name>os_project_name</name>
          <value>vimProject</value>
        </property>
      </properties>
    </vim_connector>
  </vim_connectors>
</esc_system_config>

```

```

    </property>
    <!-- The project domain name is only needed for openstack v3 identity api -->
    <property>
      <name>os_project_domain_name</name>
      <value>default</value>
    </property>
    <property>
      <name>os_identity_api_version</name>
      <value>3</value>
    </property>
  </properties>
</vim_connector>
</vim_connectors>
</esc_system_config>

```

- Updating VIM user using Netconf:

```

<esc_system_config xmlns="http://www.cisco.com/esc/esc">
  <vim_connectors>
    <vim_connector>
      <id>example_vim</id>
      <users>
        <user nc:operation="replace">
          <id>my_user</id>
          <credentials>
            <properties>
              <property>
                <name>os_password</name>
                <value>*****</value>
              </property>
            <!-- The user domain name is only needed for openstack v3 identity api
-->
            <property>
              <name>os_user_domain_name</name>
              <value>default</value>
            </property>
          </properties>
          </credentials>
        </user>
      </users>
    </vim_connector>
  </vim_connectors>
</esc_system_config>

```

- Deleting VIM connector using Netconf:

```

<esc_system_config xmlns="http://www.cisco.com/esc/esc"> <vim_connectors>
  <vim_connector nc:operation="delete">
    <id>example_vim</id>
  </vim_connector>
</vim_connectors>
</esc_system_config>

```

- Deleting VIM Connector using command:

```

$ esc_nc_cli --user <username> --password <password> delete-vim-connector <vim connector id>

```

- Deleting VIM user using command:

```

$ esc_nc_cli --user <username> --password <password> delete-vim-user <vim connector id>
  <vim user id>

```

Configuring using REST API

- Adding VIM using REST:

```
POST /ESCManager/v0/vims/
HEADER: content-type, callback

<?xml version="1.0"?>
<vim_connector xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <id>example_vim</id>
  <type>OPENSTACK</type>
  <properties>
    <property>
      <name>os_auth_url</name>
      <value>{auth_url}</value>
    </property>
    <property>
      <name>os_project_name</name>
      <value>vimProject</value>
    </property>
    <!-- The project domain name is only needed for openstack v3 identity api -->
    <property>
      <name>os_project_domain_name</name>
      <value>default</value>
    </property>
    <property>
      <name>os_identity_api_version</name>
      <value>3</value>
    </property>
  </properties>
</vim_connector>
```

- Adding VIM user using REST:

```
POST /ESCManager/v0/vims/{vim_id}/vim_users
HEADER: content-type, callback

<?xml version="1.0"?>
<user xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <id>my_user</id>
  <credentials>
    <properties>
      <property>
        <name>os_password</name>
        <value>*****</value>
      </property>
      <!-- The user domain name is only needed for openstack v3 identity api -->
      <property>
        <name>os_user_domain_name</name>
        <value>default</value>
      </property>
    </properties>
  </credentials>
</user>
```

- Update VIM using REST:

```
PUT /ESCManager/v0/vims/{vim_id}
HEADER: content-type, callback

<?xml version="1.0"?>
<vim_connector xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <!--unique id for each vim-->
  <id>example_vim</id>
```

```

<type>OPENSTACK</type>
<properties>
  <property>
    <name>os_auth_url</name>
    <value>{auth_url}</value>
  </property>
  <property>
    <name>os_project_name</name>
    <value>vimProject</value>
  </property>
  <!-- The project domain name is only needed for openstack v3 identity api -->
  <property>
    <name>os_project_domain_name</name>
    <value>default</value>
  </property>
  <property>
    <name>os_identity_api_version</name>
    <value>3</value>
  </property>
</properties>
</vim_connector>

```

- Update VIM user using REST:

```

PUT /ESCManager/v0/vims/{vim_id}/vim_users/{vim_user_id}
HEADER: content-type, callback

```

```

<?xml version="1.0"?>
<user xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <id>my_user</id>
  <credentials>
    <properties>
      <property>
        <name>os_password</name>
        <value>*****</value>
      </property>
      <!-- The user domain name is only needed for openstack v3 identity api -->
      <property>
        <name>os_user_domain_name</name>
        <value>default</value>
      </property>
    </properties>
  </credentials>
</user>

```

- Delete VIM using REST:

```

DELETE /ESCManager/v0/vims/{vim_id}

```

- Delete VIM user using REST:

```

DELETE /ESCManager/v0/vims/{vim_id}/vim_users/{user_id}

```

- Notification example after each VIM or VIM user configuration is complete:

```

<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-10-06T16:24:05.856+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Created vim connector successfully</status_message>
    <vim_connector_id>my-server-30</vim_connector_id>
  </escEvent>
</notification>

```



```

        <type>CREATE_VIM_CONNECTOR</type>
    </event>
</escEvent>
</notification>

```

Important Notes:

- In ESC 3.0, you can add multiple VIM Connector for Openstack VIM. Each VIM Connector can have only one VIM User.
- VIM username and password can be updated at anytime. VIM endpoint will not be able to update after a resource is created through ESC.
- After VIM is connected and VIM user is authenticated, VIM can no longer be deleted or updated, only VIM user can be deleted or updated.
- The name of a VIM property or VIM user credentials property is not case sensitive, e.g. OS_AUTH_URL and os_auth_url is the same to ESC.

Enabling Barbican Client in ESC

OpenStack Barbican provides secure storage, provisioning, and management of secret, for example, passwords, encryption keys, and X.509 Certificates.

Enable the Barbican client to manage the secret used to encrypt an OpenStack volume before mounting it on a VM. You can access the OpenStack Barbican APIs through a Python 3 environment. The OpenStack Barbican client python-barbicanclient 5.0.1 is integrated in the ESC.

Enable the virtual environment by using the following command:

```
source /opt/esc_custom_python3_venv/bin/activate
```

Staging VPC Volume

During VPC deployment, additional disk permission is required to create a volume. For security purpose, the permission is disabled from ESC 5.6 release, making it unavailable for arbitrary use by LCM scripts. To enable volume creation, temporary permission is granted to the script using the sudo program.

Setting up VPC Volume Staging

Use the following steps to set up VPC Volume Staging.

- Volume creation is performed by the `esc_stage_content_via_volume.sh` script. Edit the `/etc/sudoers.d/50-esc-sudoers` file. Add the following line and replace `/path/to/` with the real path to the script from the root directory.

```
mona-user ALL = (root) NOPASSWD: /path/to/esc_stage_content_via_volume.sh
```

- In the VPC deployment payload, update the deployment policy to use `sudo-esc_stage_content_via_volume.sh`.

```

<policy>
<name>1</name>

```

```

<conditions>
<condition>
<name>LCS::PRE_DEPLOY</name>
</condition>
</conditions>
<actions>
<action>
<name>GEN_VPC_ISO</name>
<type>script</type>
<properties>
<property>
<name>script_filename</name>
<value>/path/to/sudo-esc_stage_content_via_volume.sh</value>
</property>
...
</action>
</actions>
</policy>

```

- The `sudo-esc_stage_content_via_volume.sh` script simply calls `esc_stage_content_via_volume.sh` using `sudo`. Edit the file by setting the `STAGING_FILE` variable, replacing `/path/to/` with the real path from the root directory.

```
STAGING_FILE=/path/to/esc_stage_content_via_volume.sh
```

Running MONA in a Root Jail

The ESC MONA service normally runs as the `mona-user`, sharing the same filesystem as other ESC services. For additional security, it is possible to run MONA in its own filesystem called a root jail. This effectively sandboxes MONA so that the process (and any LCM scripts called by MONA) cannot access the wider filesystem. Interacting with MONA remains the same. Stopping/starting, health queries, log location and service jar updates are not affected by the root-jail, or the network connectivity.

- When the MONA process starts, it checks for the existence of the jail directory `/opt/cisco/esc/mona-jail`. If it exists, ESC will start (chroot) MONA into this environment.

Installing the MONA Root Jail

Perform the following steps:

- Stop the ESC service using the command `sudo service esc_service stop`
- Untar the MONA jail archive using the command `sudo tar -C /opt/cisco/esc -xvzf mona-jail.tar.gz`. The archive is located in the ESC repository `elastic-services-controller/esc-mona/jail/mona-jail.tar.gz`.
- Start the ESC service using the command `sudo service esc_service start`.

Customising the Root Jail Environment

The jail contains a minimal set of binaries for MONA to operate. Some custom CLM scripts may need additional binaries (plus dependent libraries) added to the jail. If for example you needed the `strings` command available to MONA scripts, you could use the `ldd` command to find the dependencies.

```
$ ldd /usr/bin/strings
linux-vdso.so.1 => (0x00007ffd9ef96000)
libbfd-2.27-44.base.el7_9.1.so => /lib64/libbfd-2.27-44.base.el7_9.1.so (0x00007fa950415000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007fa950211000)
libc.so.6 => /lib64/libc.so.6 (0x00007fa94fe43000)
/lib64/ld-linux-x86-64.so.2 (0x00007fa95075e000)
$
```

Copying `/usr/bin/strings` into `/opt/cisco/esc/mona-jail/bin`, and shared libraries into `/opt/cisco/esc/mona-jail/lib64`. The `strings` program will then be accessible to MONA.

An easier way to achieve the same is by using JailKit from the Centos RPM repository.

```
sudo jk_cp -j /opt/cisco/esc/mona-jail /usr/bin/strings
```

Reconfiguring ESC Virtual Machine

This section covers the following:

- Reconfiguring Rsyslog
- Reconfiguring NTP
- Reconfiguring DNS
- Reconfiguring Hosts
- Reconfiguring Timezone

Reconfiguring Rsyslog

Rsyslog parameters are optional. If there is a need for customization after booting an ESC VM, you can edit the files in ESC VM (`/etc/rsyslog.d`).

Procedure

Step 1 Editing the Rsyslog file:

- If you haven't specified the log forwarding configuration at the bootup time, you may create a file under `/etc/rsyslog.d/` like `/etc/rsyslog.d/log-forwarding.conf`.
- If you have specified the log forwarding through installation, you may just need to edit the file. The file could be `/etc/rsyslog.d/20-cloud-config.conf`. In this file, to forward logs to multiple rsyslog servers, edit the following line:

```
*.* @[server_ip]:port
```

Note

- Use '@@' before specifying server ip address (if TCP is the protocol used to forward logs to the rsyslog server).
- Use '@' before specifying server ip address (if UDP is the protocol used to forward logs to the rsyslog server).
- `server_ip` can either be ipv4/ipv6 address of the rsyslog server.
- '[' around the `server_ip` is required to separate it from ':port#', if an ipv6 server address is specified.

For further information on Rsyslog configuration, see the Red Hat documentation.

- Step 2** **Configuring the ESC log file:** Configure which ESC log files you want to forward to the rsyslog server:
- Navigate to `/etc/rsyslog.d/` Create or modify a configuration file, such as **log-esc.conf**. Make a copy of `sample-log-esc.conf`.
 - Specify the following block for every file you want to forward to rsyslog server.

```
$InputFileName /var/log/esc/escmanager.log
$InputFileTag esc-manager:
$InputFileStateFile stat-esc-manager
$InputFileSeverity info
$InputRunFileMonitor
```

For example:

```
$InputFileName /var/log/esc/file1.log
$InputFileTag file1:
$InputFileStateFile stat-file1
$InputFileSeverity info
$InputRunFileMonitor

$InputFileName /var/log/esc/file2.log
$InputFileTag file2:
$InputFileStateFile stat-file2
$InputFileSeverity info
$InputRunFileMonitor
```

- Step 3** Restart the rsyslog service

```
# service rsyslog restart
```

- Step 4** Configure the server side to receive forwarded logs.
- On a designated server, go to `/etc/rsyslog.conf`, and uncomment the lines shown below, depending on if you want to listen to logs from clients based on TCP or UDP:

```
#$ModLoad imudp
#$UDPServerRun 514
```

- Exit the file. Run this command as the last step.

```
sudo service rsyslog restart
```

Now, the server is listening for logs on port 514, using TCP/UDP.

Reconfiguring NTP

Procedure

- Step 1** Open the NTP configuration file `/etc/ntp.conf` in a text editor such as `vi`, or create a new one if it does not already exist:

```
# vi /etc/ntp.conf
```

- Step 2** Add or edit the list of public NTP servers. If you don't specify the NTP server through the installation, the file should contain the following default lines, but feel free to change or expand these according to your needs:

```
server 0.rhel.pool.ntp.org iburst
server 1.rhel.pool.ntp.org iburst
server 2.rhel.pool.ntp.org iburst
server 3.rhel.pool.ntp.org iburst
server <your_ntp_server_ip> iburst
```

The `iburst` directive at the end of each line speeds up the initial synchronization.

- Step 3** Once you have the list of servers complete, in the same file, set the proper permissions, giving the unrestricted access to localhost only. Make sure those lines are there in your configure file.

```
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict -6 ::1
```

- Step 4** Save all changes, exit the editor, and restart the NTP daemon:

```
# service ntpd restart
```

- Step 5** Make sure that `ntpd` is started at boot time:

```
# chkconfig ntpd on
```

Reconfiguring DNS

Procedure

- Step 1** The `/etc/resolv.conf` file contains the configuration for the DNS client (resolver). It typically looks something like this:

```
search domain.com
nameserver 8.8.4.4
```

This results in a `/etc/resolv.conf`:

```
Created by cloud-init on instance boot automatically, do not edit.
;
#Generated by esc-cloud
domain cisco.com
search cisco.com
nameserver 8.8.4.4
```

Note It is recommended that you ignore the `do not edit` message, if you want to modify the file.

- Step 2** You may modify the IP address of the "nameserver" item or add new nameserver records.

```
search domain.com
nameserver <your_first_dns_ip>
nameserver <your_second_dns_ip>
```

- Step 3** Restart Network Service.

```
service network restart
```

Reconfiguring Hosts

The `/etc/hosts` file allows you to add, edit, or remove hosts. This file contains IP addresses and their corresponding hostnames. If your network contains computers whose IP addresses are not listed in DNS, it is recommended that you add them to the `/etc/hosts` file.

Procedure

- Step 1** Add the IP addresses that are not listed in DNS to the `/etc/hosts` file.
- Step 2** Restart your network for the changes to take effect.

```
service network restart
```

Reconfiguring Timezone

For ESC VM, in `/etc` the file "localtime" is a link to or copy of a file containing information about your time zone. Access your zone information files from `/usr/share/zoneinfo`. To change the time zone, find your country, your city or a city in the same time zone from zone information files in `/usr/share/zoneinfo` and link it to the localtime in the `/etc` file.

```
$ ln -sf /usr/share/zoneinfo/America/Los_Angeles /etc/localtime
```

Verifying ESC Configurations and Other Post-Install Operations

This section covers various post-install checks and operations using the `escadm` tool.

Verifying Existing ESC Configurations

You can use `escadm dump` command for displaying current ESC configurations in yaml format. The output will show the various services in ESC.

```
$ sudo escadm dump
```

```
resources:
  confd:
    init_aaa_users:
      - name: admin
        passwd:
    option: start-phase0
  esc_service:
    group:
      - confd
      - mona
      - vimmanager
      - pgsq1
      - escmanager
```

```

- portal
- monitor
- snmp
  type: group
escmanager: {}
mona: {}
monitor: {}
pgsql: {}
portal: {}
snmp:
  run_forever: true
vimmanager: {}

```

Verifying VIM configurations

You can use `escadm vim show` command to verify the vim settings are correctly populated:

```

$ sudo escadm vim show

[
  {
    "status": "CONNECTION_SUCCESSFUL",
    "status_message": "Successfully connected to VIM",
    "type": "OPENSTACK",
    "id": "default_openstack_vim",
    "properties": {
      "property": [
        {
          "name": "os_auth_url",
          "value": "http://172.16.103.143:35357/v3"
        }
      ]
    }
  }
]

```

Troubleshooting ESC Services Startup Issues

Problem: Issues encountered while verifying ESC services status at the installation time using `sudo escadm status`.

Causes: Some services take time to start or have trouble starting.

Solution:

1. Identify the issues using one of the following method:

- Check the log `/var/log/esc/escadm.log`

```

$ cat /var/log/esc/escadm.log
2017-06-01 20:35:02,925: escadm.py(2565): INFO: promote drbd to primary...
2017-06-01 20:35:02,934: escadm.py(2605): INFO: Waiting for at least one drbd to be
  UptoDate...
2017-06-01 20:35:02,942: escadm.py(2616): INFO: Waiting for peer drbd node to be
  demoted...
2017-06-01 20:35:14,008: escadm.py(2423): INFO: mount: /dev/drbd1
  /opt/cisco/esc/esc_database
2017-06-01 20:35:14,017: escadm.py(1755): INFO: Starting filesystem service: [OK]
2017-06-01 20:35:15,039: escadm.py(1755): INFO: Starting vimmanager service: [OK]
2017-06-01 20:35:16,116: escadm.py(1755): INFO: Starting monitor service: [OK]
2017-06-01 20:35:17,163: escadm.py(1755): INFO: Starting mona service: [OK]
2017-06-01 20:35:18,440: escadm.py(1755): INFO: Starting snmp service: [OK]

```

```
2017-06-01 20:35:21,397: escadm.py(1770): INFO: Starting confd service:[FAILED]
2017-06-01 20:35:28,304: escadm.py(1755): INFO: Starting pgsqldb service: [OK]
2017-06-01 20:35:29,331: escadm.py(1755): INFO: Starting escmanager service: [OK]
2017-06-01 20:35:30,354: escadm.py(1755): INFO: Starting portal service: [OK]
2017-06-01 20:35:31,523: escadm.py(1755): INFO: Starting esc_service service: [OK]
```

- Add '-v' to escadm status to show the verbose output of the ESC services.

```
$ sudo escadm status --v
0 ESC status=0 ESC HA Active Healthy
pgsqldb (pgid 61397) is running
vimmanager (pgid 61138) is running
monitor (pgid 61162) is running
mona (pgid 61190) is running
drbd is active
snmp (pgid 61541) is running
filesystem (pgid 0) is running
<<service>> is dead
keepalived (pgid 60838) is running
portal (pgid 61524) is running
confd (pgid 61263) is running
escmanager (pgid 61491) is running
```

2. Confirm the status of the identified services that has issues and manually start these services.

```
$ sudo escadm <<service>> status// If the status is stopped or dead, manually start the
services using the next command.
```

```
$ sudo escadm <<service>> start --v
```

Logging in to the ESC Portal



Note

- The ESC portal is enabled by default. You must ensure that the ESC portal is not disabled during installation. For more information on enabling or disabling the ESC portal, see [Installing Cisco Elastic Services Controller Using the QCOW Image, on page 10](#).
- When you log in to the ESC portal for the first time you are prompted to change the default password.

To log in to the ESC portal, do the following:

Before you begin

- Register an instance of ESC. For more information on registering the ESC instance see, [Installing Cisco Elastic Services Controller Using the QCOW Image, on page 10](#)
- Ensure that you have the username and password.

Procedure

- Step 1** Using your web browser, enter the IP address of ESC and port 443.

Example:

For example, if the IP address of ESC is 192.0.2.254, enter:

https://192.0.2.254: 443 [login via https]

A Security Alert message is displayed.

Step 2 Click **Yes** to accept the security certificate. The Login page is displayed.

Step 3 Enter the username and password and click **Login** .

If you are logging in for the first time, the login page reappears, prompting you to change your password.

Step 4 Enter the old password in the Old Password field, then enter a new password in the New Password and Confirm Password fields.

Step 5 Click **Update Password** or press **Enter**.

Note

- If the UI becomes unresponsive, restart the UI by executing the **sudo escadm portal restart** from the ESC shell prompt.
 - ESC Portal only supports one user.
 - Currently, a pre-installed self-signed certificate supports HTTPS. The user must confirm the self-signed certificate before proceeding with the ESC Portal.
 - In HTTPS communication mode, if the URL protocol type returned by OpenStack is not HTTPS, the access to the VNF Console may be disabled. For security reasons, while running in HTTPS more non-secure communication will be rejected.
-



PART **VII**

Upgrading Cisco Elastic Services Controller

- [ESC in Maintenance Mode, on page 129](#)
- [Upgrading Cisco Elastic Services Controller, on page 137](#)



CHAPTER 19

ESC in Maintenance Mode

This chapter contains the following chapters:

- [Setting ESC in a Maintenance Mode, on page 129](#)
- [Backup the Database from the ESC Standalone Instances, on page 131](#)
- [Backup the Database from the ESC HA Active/Standby Instances, on page 132](#)
- [Restoring ESC Database, on page 134](#)

Setting ESC in a Maintenance Mode

ESC must be put to maintenance mode to backup and restore ESC database. To do so, use the `escadm` tool as specified in the below section.

Before you begin



Note From ESC release 4.4, ESC continues to be in maintenance mode after HA Active/Standby switch over or DB restoration, if ESC was in maintenance mode before HA Active/Standby switch over or DB restoration.

During maintenance mode,

- Northbound requests are blocked by ESC and ESC responds with maintenance mode notification.
- Only REST requests receive response that ESC is unavailable temporarily. ConfD requests get the maintenance mode rejection message, or an OK message for all idempotent request such as create tenant request when the tenant already exists.
- Monitoring actions are paused.
- All ongoing requests and transactions continue to progress.

Using the `escadm` Tool

ESC can be put to maintenance mode using the `escadm` tool.

Procedure

Step 1 Put ESC to maintenance mode from the VM shell:

```
sudo escadm op_mode set --mode=maintenance
```

```
Set mode to MAINTENANCE
Operation Mode = MAINTENANCE
```

For releases before ESC 5.6 release if you have D-MONA deployed, do the following to each D-MONA:

```
curl -i -k --user {REST username}:{REST password} -X POST https://{DMONA
IP}:8443/mona/v1/rules/pause_all
```

Note Starting from ESC 5.6 release, when ESC is set to maintenance mode the corresponding D-MONA instances will also be set to maintenance mode.

If D-MONA is not reachable while setting ESC to maintenance mode, you will receive the following warning:

```
Set mode to MAINTENANCE
Operation Mode = MAINTENANCE
Warning: There is at least one suspend monitoring operation failure.
Please check the logs for more details.
```

Step 2 To query operation mode at any time,

```
sudo escadm op_mode show
```

Example:

```
Operation Mode = OPERATION
```

Step 3 Set maintenance mode when there is no in-flight transaction. Using the `ipt_check` flag with the `escadm` tool, you can choose to set ESC in the maintenance mode only if there are no ongoing transactions in ESC. Set the flag to true, if you do not want ESC to set in the maintenance mode, if there are ongoing transactions in ESC.

```
sudo escadm op_mode set --mode=maintenance --ipt_check=true
```

With the `ipt_check` option set to true, `escadm` tool checks if there is any on going operation, if so, the `escadm` tool will not set ESC to maintenance mode.

Note ESC with D-MONA deployment, requires to do extra calls when going to maintenance mode.

Setting ESC in an Operation Mode

Put ESC in operation mode using the `escadm` tool:

```
sudo escadm op_mode set --mode=operation
```

Response is as follows:

```
Set mode to OPERATION
Operation Mode = OPERATION
```

For releases before ESC 5.6 release if you have D-MONA deployed, do the following to each D-MONA:

```
curl -i -k --user {REST username}:{REST password} -X POST https://{DMONA
IP}:8443/mona/v1/rules/resume_all
```



Note Starting from ESC 5.6 release, when ESC is set to operational mode the corresponding D-MONA instances will also be set to Operational mode.

If D-MONA is not reachable while setting ESC to operation mode, you will receive the following warning:

```
Set mode to OPERATION
Operation Mode = OPERATION
Warning: There is at least one resume monitoring operation failure.
Please check the logs for more details.
```

Verify ESC's operation mode at any time using the following command:

```
sudo escadm op_mode show
```

Backup the Database from the ESC Standalone Instances

- The following assumptions should be taken into consideration:
 - A third machine is required to store the database and log backups.
 - ESC does not support database schema downgrade. Restoring database to the older ESC version could cause unexpected problems.
- Before you start the backup process, ensure you have an external storage space (could be in the OpenStack controller or any system accessible by ESC). The backup/restore could be expressed in a generic format which will be used by the escadm tool: `scp://<username>:<password>@<backup_ip>:<filename>`. In this format, the credentials, IP address and file storage path of the third machine are required. You may also use localhost IP as the backup IP to backup database in a location of ESC VM and then copy the files to the external storage

To backup the ESC database from a standalone ESC or a HA Active/Standby (active node):

Procedure

Step 1 Log in to ESC VM and set it to maintenance mode, run:

```
$ sudo escadm op_mode set --mode=maintenance
```

Step 2 To make sure ESC is in maintenance mode, run:

```
$ sudo escadm op_mode show
```

Step 3 Backup database. Execute the commands below as a root user:

```
# sudo escadm backup --file /tmp/db_file_name.tar.bz2
scp://<username>:<password>@<backup_vm_ip>:<filename>
```

Step 4 To put ESC back to operation mode, run:

```
$ sudo escadm op_mode set --mode=operation
$ sudo escadm op_mode show
```

Step 5 Collect all the logs from the old ESC VM and back it up. Execute the below command as a root user.

```
# sudo escadm log collect
```

A timestamped log file will be generated in: /var/tmp/esc_log <timestamp>.tar.bz2

Note If a dynamic mapping file is used by ESC service, the dynamic mapping file should be backed up at the same time with ESC logs. The default path of the dynamic mapping file is */opt/cisco/esc/esc-dynamic-mapping/dynamic_mappings.xml*.

Step 6 After a successful database back-up, shut down the old ESC VM using Horizon/Kilo or Nova commands. For ESC VM instances based in VMware vSphere, shutdown the active instance through VMware client dashboard. An example of shutting down a VM in OpenStack is shown below :

```
$ nova stop OLD_ESC_ID
```

Step 7 Detach the old port from the old VM and rename the old ESC node. Examples of detaching and renaming the VM in OpenStack is shown below:

```
nova interface-detach ESC_NAME port-id-of-ESC_NAME
nova rename ESC_NAME ESC_NAME.old
```

In VMWare, assign a different IP address to the old VM and then rename the old VM.

Backup the Database from the ESC HA Active/Standby Instances

- The following assumptions should be taken into consideration:
 - A third machine is required to store the database and log backups.
 - ESC does not support database schema downgrade. Restoring database to the older ESC version could cause unexpected problems.
- Before you start the backup process, ensure you have an external storage space available (could be in the OpenStack controller or any system accessible by ESC). The backup/restore could be expressed in a generic format which will be used by the escadm tool:


```
scp://<username>:<password>@<backup_ip>:<filename>
```

 . In this format, the credentials, IP address and file storage path of the third machine are required. You may also use localhost IP as the backup IP to backup database in a location of ESC VM and then copy the files to the external storage.

To backup the ESC database from a standalone ESC or a HA Active/Standby (active node):

Procedure

Step 1 Perform the following steps on the Standby ESC node.

a) Connect to the standby ESC instance using SSH:

```
$ ssh <username>@<standby_vm_ip>
```

b) Verify that the ESC instance is standby and note the name of the standby ESC HA Active/Standby instance:


```
$ sudo escadm status --v
```

If the output value shows "STANDBY", the node is the standby ESC node.

- c) Change access to an admin user.

```
sudo bash
```

- d) Collect all the logs from the standby ESC VM and back it up.

```
$ sudo escadm log collect
```

A timestamped log file will be generated in: /var/tmp/esc_log <timestamp>.tar.bz2

- e) Shutdown the standby ESC instance through OpenStack Kilo/Horizon using Nova command or VMware client. An example of shutting down the VM on OpenStack is shown below:

```
$ nova stop OLD_ESC_STANDBY_ID
```

Step 2 Perform the following steps on the Active ESC node.

- a) Connect to the active ESC instance using SSH:

```
$ ssh <username>@<active_vm_ip>
```

- b) Change access to an admin user.

```
$ sudo bash
```

- c) Verify that the ESC instance is Active and note the name of the Active ESC HA Active/Standby instance

```
$ sudo escadm status --v
```

If the output value shows "ACTIVE", the node is the active ESC node.

- d) Back up the database files from the active node of ESC HA Active/Standby:

```
$ sudo escadm backup --file /tmp/db_file_name.tar.bz2
scp://<username>:<password>@<standby_vm_ip>:<filename>
```

- e) Collect the logs from the active ESC VM and back it up.

```
$ sudo escadm log collect
```

A timestamped log file will be generated in: /var/tmp/esc_log <timestamp>.tar.bz2

Note If a dynamic mapping file is used by ESC service, the dynamic mapping file should be backed up at the same time with ESC logs. The default path of the dynamic mapping file is */opt/cisco/esc/esc-dynamic-mapping/dynamic_mappings.xml*.

- Step 3** Shutdown the active ESC instance through OpenStack Kilo/Horizon using Nova command. For ESC VM instances based in VMware vSphere, shutdown the active instance through VMware client dashboard. An example of shutting down the VM on OpenStack is shown below:

```
$ nova stop OLD_ESC_ACTIVE
```

Use the **nova list** command or the **nova show OLD_ESC_STANDBY** command, to verify if the ESC HA Active/Standby instances have been successfully shut down.

- Step 4** (Only for OpenStack) Detach the port from the old ESC VM and rename the old VM.

If upgraded ESC VM needs to operate with same IP addresses and same instance names as old instances, detach the ports from each instance, shutdown the old ESC VMs and then rename the old ESC instances.

If you intend to use old VMware active instance, assign a different IP address and rename the VM name. If not, you can delete the old VM and use the same IP address for the new upgraded VMware active instance. After deleting the old VM, you can continue with the old instance name and the IP address.

OpenStack commands for detaching the ports and renaming the old VMs are shown below:

```
nova interface-list ESC_NAME
nova interface-detach ESC_NAME port-id-of-ESC_NAME
nova rename ESC_NAME ESC_NAME.old
```

Restoring ESC Database

Before you begin

To restore the database,

- In standalone ESC instance, stop ESC services. Run # `sudo escadm stop`.
- In HA Active/Standby type instances, stop escadm on the standby first, and later on the Active ESC HA Active/Standby instance. Run # `sudo escadm stop`.
- All the services must be stopped. To check the status, run # `sudo escadm status --v`.

Procedure

Step 1 Restore the database. Execute the following command from the ESC VM:

```
$ scp <username>@<server_ip>:/path/db.tar.bz2 /tmp/
$ sudo escadm restore -file /tmp/db.tar.bz2
```

Step 2 Enter the ESC password in the URL, or manually enter it after executing the above command.

Step 3 Restart the ESC service to complete the database restore by running the following command:

```
$ sudo escadm restart
```

Note ESC maintenance mode blocks the northbound request and VNF monitoring. However, if there are some ongoing transactions because of northbound requests before ESC entering maintenance mode, those transactions may have following restriction with backup and restore:

- ESC reports an error for the deployment, network creation, and subnet creation requests, if these transactions are interrupted by backup and restore. Northbound handles these error messages but it may cause network or subnet leakage in some cases (For example, ESC is interrupted before getting the UUID from OpenStack).
 - ESC reports an error for service chain upgrade, and requires service chain undeployment and deployment (rather than downgrade and upgrade) to re-create the service.
-



CHAPTER 20

Upgrading Cisco Elastic Services Controller

Cisco Elastic Service Controller supports two type of upgrades:

- **Backup and Restore Upgrade:** This upgrade process involves stopping the ESC keepalive daemon (for ESC HA), backing up the database, stopping and renaming (or deleting) the ESC instances, re-installing the ESC instances, and restore database. For information on the supported ESC versions for ESC 6.0 upgrade, see the table below.

You can upgrade the ESC instance as a standalone instance or as a high availability pair. The upgrade procedure is different for standalone and high availability pair.

This chapter lists separate procedures on how to upgrade ESC standalone and ESC High Availability instance. You must review these instructions before you decide to upgrade the ESC instance. See the [Installation Scenarios, on page 9](#) for more information on the installation scenarios.

- ESC only support direct upgrade from previous two minor releases. For example, ESC 6.0 will support direct upgrade from ESC 5.10 and ESC 5.9. For any release older than the supported versions for direct upgrade, you need to perform the staged upgrade.

Note Example of an ESC major release, ESC 3.x, ESC 4.x, and ESC 5.x
Example of an ESC minor release, ESC 5.1.x and ESC 5.4.x
Example of an ESC maintenance release, ESC 5.4.0.x and 5.4.1.x;
Example of an ESC patch release, ESC 5.4.0.88, ESC 5.4.0.89, and ESC 5.4.0.100.

- For ESC upgrade, you should be familiar with ESC installation process.
 - For OpenStack, refer to the OpenStack installation procedures, see Chapter 4: Installing Cisco Elastic Services Controller in OpenStack.
 - For VMware, refer to the VMware Installation installation procedures, see Chapter 7: Installing Cisco Elastic Services Controller in VMware vCenter.

- For ESC HA, please refer to the ESC HA installation procedures, see Chapter 5 : Configuring High Availability for OpenStack and Chapter 8: Configuring High Availability for VMware.

Table 6: Supported ESC Versions for Upgrading to ESC 6.0

Virtual Infrastructure Manager	Supported Versions for Backup and Restore Upgrade	Supported Versions for In-Service Upgrade
OpenStack, KVM	5.10, 5.9	In Service Upgrade is not supported from 5.x to 6.0 due to Base Operating System change.
VMware, vCD	5.10, 5.9	In Service Upgrade is not supported from 5.x to 6.0 due to Base Operating System change.

IMPORTANT NOTES

- After upgrading to the new ESC version, ESC service will manage the life cycle of all VNFs deployed in the previous release. To apply any new features (with new data models) to the existing VNFs, you must undeploy and redeploy these VNFs.
- Upgrade is supported for Active/Active HA only.



Note Do not change the VIP in ESC upgrade in ETSI deployment.

If you change ETSI's REST schema, from http to https in-flight, ESC stops sending recovery notification from ESC core for any existing deployment.

- [Upgrading Standalone ESC Instance, on page 138](#)
- [Upgrading ESC HA Active/Standby Instances, on page 140](#)

Upgrading Standalone ESC Instance

To upgrade standalone ESC instance, perform the following tasks:

1. Back up the ESC database. For more information, see [Backup the Database from the ESC Standalone Instances](#).



Note To backup any custom scripts used by the deployments, and for restoring them before restoring the database, you must re-install the scripts on the backup as well.

2. Redeploy the ESC instance. For more information, see the below section, **Deploy the ESC for Upgrade**.
3. Restore the ESC database on the new ESC instance. For more information, see the below section, **Restoring the ESC Database**.

Deploy the ESC for Upgrade

After backing up and shutting down of the old ESC VM, a new/upgraded (based on new ESC package) ESC VM should be installed. All parameters for ESC installation should be the same as the old ESC VM deployment.

- For OpenStack, you need to register the new ESC qcow2 image using the Glance command with a new image name and then use new bootvm.py script and new image name to install ESC VM.



Note In OpenStack, if an old ESC VM was assigned with floating IP, the new ESC VM should be associated with the same floating IP after the installation.

- For VMWare, you need to use the new ESC OVA file to install ESC VM. All other configurations and property values should be the same as the old VM.

Restoring the ESC Database

Restore the ESC database on the new ESC instance , using the following procedure:

Procedure

Step 1 Connect to the new ESC instance using SSH.

```
$ ssh USERNAME@NEW_ESC_IP
```

Step 2 Stop the ESC service.

```
$ sudo escadm stop
```

Step 3 Check ESC service status to make sure all the services are stopped.

```
$ sudo escadm status
```

Step 4 Restore the database files.

```
$ scp://<username>:<password>@<standby_ip>:<filename>
$ sudo escadm restore --file /path/where/file/scp-ed/to/db.tar.bz2
```

Step 5 Start the ESC service:

```
$ sudo escadm start
```

After ESC service is started, the standalone ESC upgrade is complete. You can check the health of the new ESC service by running `$ sudo escadm status` in the new ESC VM.

Step 6 In Openstack, after restoring the database successfully, delete the old ESC instance:

```
$ nova delete OLD_ESC_ID
```

Important Notes:

After upgrading to the new ESC version, ESC service will keep doing life cycle management of all VNFs deployed by the old version. However, to apply any new features (with new data models) to the VNFs deployed

by the ESC with old version is not guaranteed. If you want to apply any new feature of the new ESC version to existing VNFs, you have to undeploy and redeploy those VNFs.

Upgrading ESC HA Active/Standby Instances

To upgrade ESC HA Active/Standby nodes, perform the following tasks:

1. Back up the database from an old ESC HA Active/Standby active instance. For more information, see [Backup the Database from the ESC HA Active/Standby Instances](#).



Note To backup any custom scripts used by the deployments, and for restoring them before restoring the database, you must copy the scripts on the backup as well.

2. Deploy new ESC HA Active/Standby nodes based on new ESC version. For more information, see the below section, **Deploy the ESC HA Active/Standby nodes for Upgrade**.
3. Restore the database on active ESC instance (Standby ESC instance will sync with the active ESC instance). For more information, see the below section, **Restoring the ESC Database on New Active and Standby Instances**.

Deploying the ESC HA Active/Standby nodes for Upgrade

After backing up and shutting down the two old ESC VMs, based on new ESC package install the new ESC VMs.

- For OpenStack, you need to register the new ESC qcow2 image using the Glance command with a new image name and then to use new bootvm.py script and new image name to install ESC VM. All other bootvm.py arguments should be the same as used to setup an old VMs.
- For VMWare, there are two steps to bring up HA Active/Standby pair in VMware: 1) setup two standalone instances 2) reconfigure each instance with HA Active/Standby info. All other configurations and property values should be the same as the old VMs.
- If VIP is used for Northbound access, keep VIP the same for the new deployment as used to reconfigure the old HA Active/Standby pair.

Restoring the ESC Database on New Active and Standby ESC Instances

Procedure

Shut down the Standby ESC instance.

Step 1 Connect to the standby ESC instance using SSH.

```
$ ssh USERNAME@ESC_STANDBY_IP
```

Step 2 Verify that the ESC instance is standby and note the name of the standby ESC HA Active/Standby instance :

```
$ sudo escadm status
```

If the output value shows "STANDBY", the node is the standby ESC node.

Note If a dynamic mapping file (dynamic_mapping.xml) is used by ESC service, the dynamic mapping file should be restored into the standby ESC VM. Before power off the standby ESC node, you need to copy the backup dynamic mapping file (dynamic_mapping.xml) to the path `/opt/cisco/esc/esc-dynamic-mapping/`.

Step 3 Shutdown the standby ESC instance through OpenStack Kilo/Horizon using Nova command. For ESC VM instances based in VMware vSphere, shutdown the active instance through VMware client dashboard. An example of shutting down the standby ESC instance in OpenStack is shown below:

```
$ nova stop NEW_ESC_STANDBY_ID
```

Restore the database on the new Active ESC instance.

Step 4 Connect to the active ESC instance using SSH.

```
$ ssh USERNAME@ESC_ACTIVE_IP
```

Step 5 Verify that the ESC instance is active.

```
$ sudo escadm status
```

If the output value shows 'ACTIVE', the node is the active ESC node.

Step 6 Stop the ESC services on the active node and verify the status to ensure the services are stopped.

```
$ sudo escadm stop
$ sudo escadm status
```

Step 7 Restore the database files.

```
$ sudo escadm restore --file /tmp/db.tar.bz2
$ scp://<username>:<password>@<standby_ip>:<filename>
```

Note If a dynamic mapping file (dynamic_mapping.xml) is used by ESC service, the dynamic mapping file should be restored into the ESC VM. Before starting the ESC node, you need to copy the backup dynamic mapping file (dynamic_mapping.xml) to the path `/opt/cisco/esc-dynamic-mapping/`.

Step 8 Reboot the VM to restart the full ESC service:

```
$ sudo escadm restart
```

Step 9 Use the `$ sudo escadm status` to check the status of the ESC service.

Step 10 Start the standby ESC node.

Power on the standby ESC node through OpenStack Nova/Horizon or VMware client. After starting the standby node, ESC HA Active/Standby upgrade process should be complete.

Step 11 Delete the old HA Active/Standby instance through OpenStack Nova/Horizon or VMware client. An example of deleting the VM on OpenStack is shown below:

```
$ nova delete OLD_ESC_ACTIVE_RENAMED OLD_ESC_STANDBY_RENAMED
```

Upgrading VNF Monitoring Rules

To upgrade the VNF monitoring rules, you must back up the *dynamic_mappings.xml* file and then restore the file in the upgraded ESC VM. For more information, see the backup and restore procedures. For upgrade of HA Active/Standby instance, see [Upgrading ESC HA Active/Standby Instances](#). For upgrade of the standalone instance, see [Upgrading Standalone ESC Instance](#).



PART **VIII**

Troubleshooting Cisco Elastic Services Controller Installation

- [Troubleshooting ESC Issues, on page 145](#)



CHAPTER 21

Troubleshooting ESC Issues

This chapter contains the following sections:

- [Viewing ESC Log Messages, on page 145](#)
- [General Installation Errors, on page 150](#)
- [ESC Failover Scenarios, on page 153](#)

Viewing ESC Log Messages

Log messages are created for ESC events throughout the VNF lifecycle. These can be external messages, messages from ESC to other external systems, error messages, warnings, events, failures and so on. The log file can be found at `/var/log/esc/escmanager_tagged.log`.

The log message format is as follows:

```
date=<time-date>] [loglevel=<loglevel>] [tid=<transactionid>] [cl=<classifications>]
[tags=<tags>] [msg=<message>
```

Sample log is as follows:

```
date=15:43:58,46022-Nov-2016]
[loglevel=ERROR ] [tid=0793b5c9-8255-47f3-81e6-fbb59f6571f7] [cl=OS ]
[tags=wf:create_vm,eventType:VM_DEPLOY_EVENT,tenant:CSCvd94541,depName:test-dep,vmGrpName:test-VNF,
vmName:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0]
[tags=wf:create_vm,eventType:VM_DEPLOY_EVENT,tenant:test,depName:test-dep,vmGrpName:test-VNF,
vmName:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0] [msg=sleepingfor5seconds
to allow vm to become ACTIVE instance id:
162344f7-78f9-4e45-9f23-34cf87377fa7
name:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0
```

When a request is received, a RequestDetails object is created which autogenerates a unique transaction id. This value is carried forward across all threads. Classifications and tags are optional. These are prefixes added to the log messages to enhance readability, and help in debugging. With classifications and tags, the log messages can be easily parsed and filtered by the log analysis tools.

The following classifications are supported:

NBI	"com.cisco.esc.rest""com.cisco.esc.filter"(North Bound Interface - Clientinterface)
SBI	"com.cisco.esc.rest"- source is a callback handler or"EventsResource"(South Bound Interface - i.e. between ESC and the VIM)

SM	"com.cisco.esc.statemachines". stands for StateMachine. This classification indicates logs in the StateMachine category.
MONITORING	"com.cisco.esc.monitoring""com.cisco.esc.paadaptor"(MONA related logs)
DYNAMIC_MAPPING	"com.cisco.esc.dynamicmapping""com.cisco.esc.db.dynamicmapping"(MONA related logs)
CONFD	"com.cisco.esc.confid"
CONFD_NOTIFICATION	"com.cisco.esc.confid.notif""com.cisco.esc.confid.ConfdNBIAadapter"
OS	"com.cisco.esc.vim.openstack"
LIBVIRT	"com.cisco.esc.vim.vagrant"
VIM	"com.cisco.esc.vim"
REST_EVENT	"ESCManager_Event""com.cisco.esc.util.RestUtils". indicates REST notifications in logs.
WD	"com.cisco.esc.watchdog"
DM	"com.cisco.esc.datamodel""com.cisco.esc.jaxb.parameters"(Datamodel and resource objects)
DB	"com.cisco.esc.db"(Database related logs)
GW	"com.cisco.esc.gateway"
LC	"com.cisco.esc.ESCManager"(Start up related logs)
SEC	"com.cisco.esc.jaas"
MOCONFIG	"com.cisco.esc.moconfig"(MOCONFIG object related logs --this is internal for ESC developers)
POLICY	"com.cisco.esc.policy"(Service/VM Policy related logs)
TP	"com.cisco.esc.threadpool"
ESC	"com.cisco.esc" Any other packages not mentioned above

The following tags are supported:

- **Workflow [wf:]**—Generated using action and resource from RequestDetails object. Example "wf:create_network"
- **Event type [eventType:]**—Event that triggered the current action. Example: "eventType:VM_DEPLOY_EVENT"
- **Resource based**—These values are generated based on the type of parameter used by the event. The hierarchy, that is, the tenant, the vm group and so on is added to the log.

Tenant	[tenant:<tenant name>]
--------	------------------------

Network	[tenant:<tenant id>, network:<network name>] Note The tenant appears only if applicable.
Subnet	[tenant:<tenant name or id>, network:<network name or id>, subnet:<subnet name>] Note The tenant appears only if applicable.
User	[tenant:<tenant name>, user:<user name or id>] Note The tenant appears only if applicable.
Image	[image:<image name>]
Flavor	[flavor:<flavor name>]
Deployment	[tenant:<tenant name or id>, depName:<deployment name>]
DeploymentDetails	[tenant:<tenant name or id>, depName:<deployment name>, vmGroup:<vm group name>, vmName:<vm name>]
Switch	[tenant:<tenant name or id>, switch:<switch name>]
Volume	[volume:<volume name>]
Service	[svcName:<Service Registration name>]

Further, ESC logs can also be forwarded to an rsyslog server for further analysis and log management.

Filtering Logs Using Confd APIs

You can query and retrieve logs (for example, deployment logs, or error logs) in ESC using log filters introduced in the confd APIs. New filters for Tenant, Deployment Name, and VM Name are introduced. This enables you to query the ESC logs further for most recent error logs using the log filters in Confd APIs. You can also retrieve ESC logs related to the communication between ESC and the OS (by setting the classification tag to "OS").

The log format to retrieve confd API logs:

```
date=<time-date>] [loglevel=<loglevel>] [tid=<transactionid>] [cl=<classifications>]
[tags=<tags>] [msg=<message>
```

The sample log is as follows:

```
date=15:43:58,46022-Nov-2016] [loglevel=ERROR ] [tid=0793b5c9-8255-47f3-81e6-fbb59f6571f7]
[cl=OS ]
[tags=wf:create_vm,eventType:VM_DEPLOY_EVENT,tenant:test,depName:test-dep,vmGrpName:test-VNF,
vmName:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0]
[msg=sleepingfor5seconds to allow vm to become ACTIVE instance id:
162344f7-78f9-4e45-9f23-34cf87377fa7 name:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0
```

The parameters for log level, classification and tags are dependent on each other to retrieve the logs. You can successfully retrieve the logs with the following combination.

- log_level=ERROR, classifications=OS, tags=(depName:test-dep)
- log_level=ERROR, classifications=OS, tags=(tenant: test)

The log filter returns a value when all the following conditions are met:

- Log level
- Classifications (if provided)
- Tags (if provided)



Note If there are more than one classification listed, it has to match at least one of the classifications. The same applies to the tags as well.

For example, the following log filter criteria does not return the log sample mentioned earlier:

```
log_level=ERROR, classifications=VIM, tags=(depName:test-dep)
```

It does not return any value though the log level and tags match, the classification VIM does not match.

The data model is as follows:

```
rpc filterLog {
  description "Query and filter escmanager logs using given parameters";
  tailf:actionpoint escrpc;
  input {
    leaf log_level {
      mandatory false;
      description "One of DEBUG / INFO / WARNING / ERROR / TRACE / FATAL. Results will
include all logs at and
      above the level specified";
      type types:log_level_types;
      default ERROR;
    }
    leaf log_count {
      mandatory false;
      description "Number of logs to return";
      type uint32;
      default 10;
    }
    container classifications {
      leaf-list classification {
        description "Classification values to be used for the log filtering. For example:
'OS', 'SM'.
          Logs containing any of the provided classification values will be
returned.";
        type types:log_classification_types;
      }
    }
    container tags {
      list tag {
        key "name";
        leaf name {
          mandatory true;
          description "Tag name to be used for the log filtering. For example: 'tenant',
'depName'.
          Logs containing any of the provided tag name plus the tag values
will be returned.";
          type types:log_tag_types;
        }
        leaf value {
          mandatory true;
          description "Tag value pairs to be used for the log filtering. For example:
'adminTenant', 'CSRDeployment'";
          type string;
        }
      }
    }
  }
}
```



```

    <tag>
      <name>depName</name>
      <value>CSR_ap1</value>
    </tag>
    <tag>
      <name>tenant</name>
      <value>admin</value>
    </tag>
  </tags>
</filterLog>

```

The response is as follows:

```

<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <filterLogResults xmlns="https://www.cisco.com/esc/esc">
    <log_level>INFO</log_level>
    <logs>
      <classifications>
        <classification>OS</classification>
        <classification>SM</classification>
      </classifications>
      <tags>
        <tag>
          <name>depName</name>
          <value>CSR_ap1</value>
        </tag>
        <tag>
          <name>tenant</name>
          <value>admin</value>
        </tag>
      </tags>
      <log_date_time>13:06:07,575 31-Oct-2016</log_date_time>
      <log_message> No pending work flow to start.</log_message>
    </logs>
  </filterLogResults>
</rpc-reply>

```



Note The logging API responses are in XML format. If the log messages contain any XML characters, then the characters will be escaped so not to break the XML conformance.

General Installation Errors

This section cover some of the common installation problems and how to troubleshoot them.

Problem/Error	Possible Reason	User Action
<p>Issues encountered while verifying ESC services status at the installation time using <code>sudo escadm status</code></p>	<p>Some services take time to start or have trouble starting.</p>	<p>1. Identify the issues using one of the following method:</p> <ul style="list-style-type: none"> • Check the log <code>/var/log/esc/escadm.log</code> <pre>\$ cat /var/log/esc/escadm.log</pre> <ul style="list-style-type: none"> • Add '-v' to escadm status to show the verbose output of the ESC services and see the services are up and running. <p>2. Confirm the status of the identified services that has issues and manually start these services.</p> <pre>\$ sudo escadm <<service>> status// If the status is stopped or dead, manually start the services using the next command. \$ sudo escadm <<service>> start --v</pre>
<p>Authentication related error during the ESC installation</p>	<p>OpenStack credential-related arguments are missing.</p>	<p>Source your OpenStack RC file and verify the OpenStack clients work properly</p>
<p>ESC HA-related (Active/Standby) Issues</p>		
<p>Network Problem</p>		<p>Check for the following:</p> <ul style="list-style-type: none"> • The static IP addresses for both ESC nodes used are based on the OpenStack configuration. • The gateway for each network interface is accessible.

Problem/Error	Possible Reason	User Action
<p>ESC active node stays in the "switching-to-active" state</p>	<p>This could be because of the following issues:</p> <ul style="list-style-type: none"> • ESC HA Active/Standby node can not reach its peer during the initial installation. • ESC service (tomcat) can't start properly due to database problems (etc. database migration, database file corruption). • Confd can't start due to the CDB file corruption. • Postgresql can't start or init due to the file system issues. • The connection between ESC nodes is too slow. 	<p>Check for the:</p> <ul style="list-style-type: none"> • Connectivity between ESC active node and standby node. For initial installation, ESC active service won't be up if it can't reach the standby node. You need to make sure you have both ESC nodes successfully deployed and they can reach each other. • ESC logs at /var/log/esc/esc_haagent.log (ESC 2.X) or /var/log/esc/escadm.log (ESC 3.X) to identify the services that has issues. • Log at /var/log/esc/escmanager.log for esc_service and postgresql issues.
<p>MTU issues for ESC HA Active/Standby</p>		<p>For the ESC VMs, reduce the MTU for network interfaces from 1500 to 1450. Follow the below steps to reduce the MTU value:</p> <ol style="list-style-type: none"> 1. Identify the interface you would like to change. Access the interface from the location, /etc/sysconfig/network-scripts/ifcfg-ethX, where X represent the interface number you want to change. 2. Use a text editor like VIM to add or edit the MTU items for the interface, for instance, set MTU = 1450 3. Restart the interface. <pre>network service restart</pre>

Problem/Error	Possible Reason	User Action
Other Issues		<p>There are several useful logs to check for ESC HA Active/Standby troubleshooting.</p> <ul style="list-style-type: none"> • ESC manager log is located at <code>/var/log/esc/escmanager.log</code> • ESC HA Active/Standby log about esc service startup/stop is located at <code>/var/log/esc/esc_haagent.log</code> for ESC 2.X and <code>/var/log/esc/escadm.log</code> for ESC 3.X • For Keepalived configuration: <ul style="list-style-type: none"> • Check the configuration file at <code>/etc/keepalived/keepalived.conf</code> • Keepalived log is located at <code>/var/log/messages</code> by <code>grep keepalived</code> or <code>vrpp</code> • For DRBD configuration: <ul style="list-style-type: none"> • DRBD configuration can be verified by checking the file at <code>/etc/drbd.d/esc.res</code> • The DRBD log is located at <code>/var/log/messages</code> by <code>grep drbd</code>
ESC Services Startup Issues		
Service status reports ETSI service is dead at installation time using <code>sudo escadm status --v</code>	<p>If the host has multiple IP addresses the ETSI service is unable to determine which IP address to use when generating callback URLs.</p> <p>This cause can be verified by examining the log file <code>/var/log/esc/etsi-vrpp/etsi-vrpp.log</code> for an exception including:</p> <p>Configure <code>server.host</code> property in configuration file.</p>	<ol style="list-style-type: none"> 1. Set the <code>server.host</code> property: <pre>sudo escadm etsi set -server_host <IP></pre> 2. Start the ETSI service: <pre>sudo escadm etsi start</pre>

ESC Failover Scenarios

- When customer deploys many VNFs

- When customer does a DB backup
- One or more of the VMs in the VNFs is recovered via redeploy – new VM ID
- ESC Fails – It is a permanent failure
- Customer Restores DB – When there is a mismatch between 1 or more VM IDs
- Same VNF fails, but when ESC tries to recover, it fails because it is not found on the VIM by VM ID and redeploy fails because the ports are in use.
- The workaround is to delete the VM out of band and recover – if the VM is deleted it is not found by ESC and the port is available for redeployment.



APPENDIX **A**

Cisco Elastic Services Controller Installer Arguments

You need to specify the following *bootvm.py* script arguments to boot ESC instances.

Arguments	Description
esc_hostname	Specifies the host name of the ESC VM instance.
--image	Specifies the image id used in the OpenStack glance to boot up the ESC instance.
--boot_volume	Specify the volume name or id of the external bootable volume from where you want to launch ESC instance.
--ignore-ssl-errors	It sets the "ignoreSslErrors" to "yes". It helps in deploying to a development or test environment when you don't have a root trusted certificate installed.
--managers	It is a comma delimited list of locations where SNMP traps is delivered. It must be supplied in the following format: <code>udp:ipv4/port</code> or <code>udp:[ipv6]/port</code>
--net	Specifies the Network IDs or names in OpenStack that ESC connects to.
--ipaddr	(Optional) Specifies the IP addresses that ESC will be assigned in the network. Note The IP address must correspond to the net_id in the --net argument.
--gateway_ip	(Optional) Specifies the default gateway IP address of ESC.
--os_auth_url	(Optional) Specifies the OpenStack keystone url used by os_auth_url for authentication.
--os_username	(Optional) Specifies the OpenStack keystone username used by os_username for authentication.

Arguments	Description
--os_password	(Optional) Specifies the OpenStack keystone password used by os_password for authentication.
--os_tenant_name	(Optional) Specifies the OpenStack tenant name used by os_tenant_name for ESC deployment.
--bs_os_auth_url	(Optional) Specifies the OpenStack keystone url used by bs_os_auth_url for authentication.
--bs_os_username	(Optional) Specifies the OpenStack keystone username used by bs_os_username for authentication.
--bs_os_password	(Optional) Specifies the OpenStack keystone password used by bs_os_password for authentication.
--bs_os_tenant_name	(Optional) Specifies the OpenStack tenant name used by bs_os_tenant_name for ESC deployment.
--flavor	(Optional) Specifies the OpenStack flavor id to boot the ESC VM.
--security_rules_file	(Optional) Specifies the file to define security rules (IP, Port security) for ESC VM.
--etc_hosts_file	(Optional) Specifies the file for adding more entries to the ESC vm's hosts file (/etc/hosts).
--avail_zone	(Optional) Specifies the OpenStack zone used for ESC deployment.
--esc_params_file	(Optional) Specifies the default parameter file for ESC deployment.
--etsi_params_file	(Optional) Specifies the ETSI properties file for ESC deployment.
--db_volume_id	(Optional) Specifies the cinder volume id to mount for database storage in ESC HA Active/Standby [ESC-HA Active/Standby].
--ha_node_list	(Optional) Specifies list of IP addresses for HA Active/Standby nodes in the Active/Standby cluster. For ESC nodes with multiple network interfaces, these IPs should be the addresses in the network used for data synchronization. Note This argument is utilized for replication-based HA Active/Standby solution only.
--kad_vip	(Optional) Specifies the IP address for Keepalived VIP (virtual IP) plus the interface of Keepalived VIP [ESC-HA Active/Standby]. An example format for specifying the interface of VIP is --kad_vip 192.0.2.1:eth2 or --kad_vip [2001:cc0:2020::fc]:eth2
--kad_vif	(Optional) Specifies the interface for Keepalived virtual IP and keepalived VRRP [ESC-HA Active/Standby]. You can also use this argument to only specify the interface for Keepalived VRRP, if the VIP interface is already specified using the <i>kad_vip</i> argument.

Arguments	Description
--kad_vri	Specifies the virtual router id of vrrp instance. Accepted values for kad_vri are 0 to 254. ESC VMs in the same HA Active/Standby should use the same kad_vri number. If kad_vip is not used for L3 HA Active/Standby, the kad_vir has to be used, otherwise, you can skip kad_vri argument.
--route	Specifies the routing configuration for ESC VM.
--ntp_server	(Optional) Specifies the NTP server address.
--rsyslog_server	(Optional) Specifies the IP address of rsyslog server that ESC sends the log to
--rsyslog_server_port	(Optional) Specifies the port of rsyslog server that ESC sends the log to.
--rsyslog_server_protocol	(Optional) Specifies the protocol to be used by the ESC to forward logs to the server.
--secure	(Optional) Enables secure configuration. You can specify the following values: <ul style="list-style-type: none"> • A—Root is completely locked out. You cannot login as a root even from the console. • B—SELinux runs in the enforcing mode. • C—IPv4/IPv6 tables are started. • D—SSH password authentication is disabled. You need the private key to ssh into ESC vm. • E—host keys for confd will be re-created.
--host_mapping_file	(Optional) Specifies the host mapping file for VNF deployment.
--version	(Optional) Prints the version of bootvm.py and exits.
--rng_virtio	Enables installing and deploying the ESC VM on Libvirt/KVM with the RNG Virtio device. The default values are: device=/dev/random rate_period=1000 rate_bytes=1024

Arguments	Description
--user_pass	<p>This along with --user_confid_pass are mandatory arguments from 3.0 onwards.</p> <p>This argument adds a user to access the ESC VM. Use this argument to specify a user without administrative privileges, i.e, a non-admin/non-root user. Use the following format: user_name:password. The bootvm.py command requires at least one --user_pass argument to create an admin account for linux (ssh/console access) . The following is the syntax for the mandatory user credential argument:</p> <pre>--user_pass admin: 'PASSWORD-OR-HASH' [:OPTIONAL-PUBLIC-KEY-FILE] [:OPTIONAL-ROLE]</pre> <p>This user can only do the following:</p> <ul style="list-style-type: none"> • Login to ESC through SSH. • Access and drive the Netconf CLI, such as, esc_nc_cli, netconf-console, and so on. • Read ESC -related logs from /var/logs/esc • Access REST interface through localhost <p>This user cannot:</p> <ul style="list-style-type: none"> • Access the ESC DB and reconfigure ESC system. • Access the system-level logs • Configure the system level components, such as: Rsyslog, Keepalived, DRDB, and so on. • Access the encryption keys and values from REST interface or ESC logs. <p>Following is an example of --user_pass for admin account and stronger clear text passwords. Use single quotes to avoid conflict with shell reserved characters:</p> <pre>-user_pass admin:'Strong4Security!'</pre> <p>Another example to install ESC using a password hash for both admin accounts. Use single quotes to avoid conflict with shell reserved characters:</p> <pre>--user_pass admin:'\$algorithm\$salt\$hash-of-salt-password'</pre> <p>ESC 2.1 and later, accepts the public key for this attribute. For example, the following will generate 'admin321' as the password for user 'admin' and use /tmp/abc.pub as the key file to inject the public key for it:</p> <pre>--user_pass admin:admin321:/tmp/abc.pub</pre>

Arguments	Description
--user_confid_pass	<p>Used to change confd users. The bootvm.py command requires at least one --user_confid_pass to create an admin account for ConfD (netconf/cli access). The following is the syntax for the mandatory user credential argument:</p> <pre>--user_confid_pass admin: 'PASSWORD-OR-HASH' [:OPTIONAL-PUBLIC-KEY-FILE]</pre> <p>Following is an example of --user_confid_pass for admin account and stronger clear text passwords. Use single quotes to avoid conflict with shell reserved characters:</p> <pre>--user_confid_pass:'Strong4Security!'</pre> <p>Another example, to install ESC using a password hash for both admin accounts. Use single quotes to avoid conflict with shell reserved characters:</p> <pre>--user_confid_pass:'\$algorithm\$salt\$hash-of-salt-password'</pre> <p>ESC 2.1 and later, accepts the public key for this attribute. For example, the following will generate 'admin321' as the password for user 'admin' and use /tmp/abc.pub as the key file to inject the public key for it:</p> <pre>--user_confid_pass:admin321:/tmp/abc.pub</pre>
--esc_portal_startup	(Optional) Starts the ESC portal.
--log	(Optional) Specifies the log file. By default, logs to stdout.
--esc_monitor_check_ips	(Optional) Specifies the IP addresses that must be monitored by esc_monitor (for HA Active/Standby failover).
--enable-https-rest	(Optional) Enables a secure REST Interface for the created ESC VM.
--enable-http-rest	(Optional) Enables an unsecured REST Interface for the created ESC VM.
--disable-rest-auth	(Optional) Disables REST API authentication. Note REST authentication should not be disabled in a production environment.
--enable-snmp-agent	(Optional) Enables automatic start-up of the SNMP service. The default value is False.
--ha_mode	Specifies the ESC HA Active/Standby mode for HA Active/Standby installation. Specify one of the following available options for HA Active/Standby: no_ha : No HA, cinder : Shared Cinder Volume, drbd : Built-in DRBD, drbd_on_cinder : DRBD over Cinder Volume
--enable-https-etsi	(Optional) Enables a secure ETSI REST Interface for the created ESC VM.

Arguments	Description
--enable-http-etsi	(Optional) Enables an unsecured ETSI REST Interface for the created ESC VM. Enabling this interface is not recommended in a production environment.
--encrypt_key	Specifies the key for encryption.
--proxy	Uses the proxy on a given port.
--noproxy	Lists the hosts which do not use proxy.
--kad_unicast_src_ip	Specifies the source IP address of unicast. Should be the IP address of interface that ESC VM uses for unicast (L3) VRRP communication. Example: --kad_unicast_src_ip 10.0.0.1
--kad_unicast_peer	Specified the peer IP addresses of unicast. Should be the ip address of interface that ESC peer VM uses for unicast (L3) VRRP communication. Example: --kad_unicast_peer 10.0.0.1
--placement_hint	Use this argument to specify the placement of ESC HA Active/Standby virtual machines using the server group, samehost, differenthost filters. Example: <ul style="list-style-type: none"> • --placement_hint different_host=2b299428-e7a7-4528-8566-9a4970183c6a [ID should be the VM uuid] • --placement_hint same_host=2b299428-e7a7-4528-8566-9a4970183c6a [ID should be the VM uuid] • --pacement_hint group=4c7758ab-e9cb-4cf0-8f02-344ec666365b [ID should be the server group uuid]
--format {json}	Use this argument to capture the success and failure message in the output. Example: \$./bootvm.py --image ESC-2_3_0_8 --net network --format json --test-0 { "status" : "Success" , "vm_uuid" : "UUID" }
--user_rest_pass	Adds a user to access the Rest API. Format is username: password. This option can be repeated.
--user_portal_pass	Add a portal user. Format username: password. This option can be repeated.
--user_etsi_pass	Adds a user to access the ETSI REST API. Format username:password. Only a single user can be supplied.
--etsi_oauth2_pass	Add an OAuth2 client for access to the ETSI REST API. Format clientId:clientSecret. This option can be repeated.

Arguments	Description
--no_vim_credentials	<p>Use this argument to deploy ESC without passing the VIM credential. If this argument is used, following parameters will not be passed during the installation:</p> <ul style="list-style-type: none"> • --os_auth_url • --os_username • --os_password • --os_tenant_name <p>After the deployment is complete, the user can set these VIM credential through ESC's VIM/VIM User APIs (REST/Netconf). For more information on configuring through REST APIs and Netconf, see <i>Configuring VIM credentials after installing ESC</i> in the Post Installation Tasks chapter.</p>

Cisco Elastic Services Controller Installer File Reference

File	Description
security_rules_file	<p>The file contains the following:</p> <ul style="list-style-type: none"> • Security rules to create a security group for the tenant. • Configurations to allow traffic for the tenant.
etc_hosts_file	The file contains one or more entries that you want to include in the /etc/hosts file.
esc_params_file	The file contains information to configure various parameters of ESC. For details on parameters that can be configured in the esc_params_file are described in table below.
host_mapping_file	The file contains information to map a network based on the hosts.

ESC Configuration Parameters

Using this file, you can configure various ESC parameters during the installation. The parameters that can be configured are shown in the table.

Below is an example configuration using this file:

```
openstack.endpoint=adminURL
affinity.filter=ServerGroupAffinity
```

Table 7: ESC Configuration Parameters

esc_param.conf	Type	Default Value	Description
default.vn_recovery_retries_max	Int	3	Number of recovery attempts allowed per VM.

esc_param.conf	Type	Default Value	Description
openstack.endpoint	String	publicURL	<p>The parameter to set up the keystone endpoint value of ESC. Options: adminURL, publicURL</p> <p>You can change the default value using CLI or REST services.</p> <p>Using CLI:</p> <pre>\$ sudo escadm escmanager config set --key openstack.endpoint --value publicURL { "category": "OPENSTACK", "type": "STRING", "value": "publicURL", "key": "ENDPOINT" }</pre> <p>Using REST:</p> <pre>\$ curl -X PUT http://172.16.0.1:8080/ESCManager/v0/config /openstack/endpoint/publicURL</pre>
log.level	String	INFO	Level of logging. Options: INFO, Trace, DEBUG
affinity.filter	String	SameHostFilter	<p>A constant string used to build PolicyEngine and initializing VM policy table.</p> <p>Options: SameHostFilter, ServerGroupAffinity</p>
anti_affinity.filter	String	DifferentHostFilter	<p>A constant string used to build PolicyEngine and initializing VM policy table.</p> <p>Options: DifferentHostFilter</p>



Note ESC uses SameHostFilter and DifferentHostFilter for ESC policy engine by default but OpenStack may not configure those filters by default. You may need to add SameHostFilter and DifferentHostFilter to the following scheduler options in the **/etc/nova/nova.conf** file of the nova service in your OpenStack.

```
scheduler_default_filters = RetryFilter, AvailabilityZoneFilter, RamFilter, ComputeFilter,
ComputeCapabilitiesFilter,
ImagePropertiesFilter, ServerGroupAntiAffinityFilter, ServerGroupAffinityFilter,
DifferentHostFilter, SameHostFilter
```

ServerGroupAntiAffinityFilter for Openstack

ESC adapts to use ServerGroupAntiAffinityFilter for Openstack.

REST

```
PUThttp://localhost:8080/ESCManager/v0/config/anti_affinity/filter/ServerGroupAntiAffinity
```

```
PUThttp://localhost:8080/ESCManager/v0/config/affinity/filter/ServerGroupAffinity
```

CLI

```
sudo escadm escmanager config set --key ANTI_AFFINITY.FILTER --value ServerGroupAntiAffinity
sudo escadm escmanager config set --key AFFINITY.FILTER --value ServerGroupAffinity
```

Important Points

ServerGroupAntiAffinityFilter from Openstack doesn't support inter-dep anti-affinit, scaling, and mix use of ServerGroup and default (SameHost/DifferentHost) filter. If you are using ServerGroupAntiAffinity filter, Intra vm group placement is not allowed. You can only use <placement_group> for the VM based placement policy, one VM per vm_group. You can not add a single vm group in two different placement_groups.

ESC Services, Ports, and Security Group Overview

Table 8: External Services

	Service	Visibility	Optional?	Interface	Protocol	Port
1	sshd	External (Orchestration)	No	0.0.0.0	TCP	22
2	ESC Web UI/Portal (HTTPS)	External (Orchestration)	Yes (REST and/or Netconf can be used instead)	0.0.0.0	TCP	443 (Previously 9001)
3	ESC Netconf API	External (Orchestration)	Yes (REST and/or Portal can be used instead)	0.0.0.0	TCP	830
4	ESC SNMP	External (Orchestration)	Yes (only configurable through custom user-data/esc-config.yaml)	0.0.0.0	TCP	2001
5	ESC DRBD (HA Active/Standby Replication)	External (Orchestration)	No. Required for HA Active/Standby setup.	0.0.0.0	TCP	7789
6	ESC ConfD CLI	External	Yes Configurable by modifying and reloading esc_production_confidconf	0.0.0.0	SSH	2024
7	ESC REST API (HTTPS)	External (Orchestration)	Yes (Portal and/or Netconf can be used instead)	0.0.0.0	TCP	8443

	Service	Visibility	Optional?	Interface	Protocol	Port
8	ESC Keepalived	External (Orchestration)	No. Required for HA Active/Standby setup.	0.0.0.0	Multicast VRRP	N/A
9	ETSI-VNFM (HTTP)	External	Yes (configurable through etsi-production.properties)	0.0.0.0	TCP	8250
10	ETSI-VNFM (HTTPS)	External	Yes (configurable through etsi-production.properties)	0.0.0.0	TCP	8251
11	ETSI-VNFM (Prometheus)	External	Yes (configurable through etsi-production.properties)	0.0.0.0	TCP	8252
12	ETSI-VNFM (Additional HTTPS Port for artifacts download)	External	Yes (configurable through etsi-production.properties)	0.0.0.0	TCP	8253
13	ESC Health API	External (Orchestration)	No	0.0.0.0	TCP	For ESC 5.2 and later, 8060 For ESC 5.1 and earlier, 60000
14	ETSI Health API	Internal				Local only
15	D-MONA REST API	External	No	0.0.0.0	TCP	8443
16	Consul Service ¹	External	No	0.0.0.0	TCP	8300, 8301, 8302
17	Consul Service ²		No	0.0.0.0	TCP and UDP	8301 8302 8600
18	ConfD	External ³	No for A/A set	Limited to ESC node IPs ⁴	TCP	4565
19	PostgreSQL	External ⁵	No for A/A set	Limited to ESC node IPs ⁶	TCP	7878

	Service	Visibility	Optional?	Interface	Protocol	Port
20	ESCManager RMI Registry ⁷	External	No for A/A set	Limited to ESC node IPs	TCP	8679
21	ESCManager RMI Service ⁸	External	No for A/A set	Limited to ESC node IPs	TCP	8680
22	ESCManager Prometheus Metrics	External ⁹	Yes. (application.properties)	0.0.0.0	TCP	8088
23	MONA Prometheus	External ¹⁰	Yes. (application.properties)	0.0.0.0	TCP	8092
24	VIMManager Prometheus Metrics	External ¹¹	Yes. (application.properties)	0.0.0.0	TCP	8097
25	DHCP	External DHCP is limited to resolving IP addresses for ESC interfaces.	No	0.0.0.0	UDP	66,67

¹ Only needed for A/A ESC set. Otherwise, the port will not listen.

² Only needed for A/A ESC set. Otherwise, the port will not listen.

³ Introduced only since ESC 5.0

⁴ ESC A/A set (3 VMs)

⁵ Introduced only since ESC 5.0

⁶ ESC A/A set (3 VMs)

⁷ Only needed for A/A ESC set. Otherwise, the port is not listened.

⁸ Only needed for A/A ESC set. Otherwise, the port is not listened.

⁹ ESC 5.2 and later

¹⁰ ESC 5.2 and later

¹¹ ESC 5.2 and later

