Configuring High Availability

This chapter contains the following sections:

High Availability Overview

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

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

How High Availability Works

ESC HA network can be either set up as a single installation of a ESC HA pair or deployed as two standalone ESC nodes that are converted into HA pair after re-configuring these nodes post deployment. A HA deployment consists of two ESC instances: a primary and a standby. Under normal circumstances, the primary ESC instance provides the service. The corresponding standby instance is passive. The standby instance is in constant communication with the primary instance and monitors the primary instances' status. If the primary 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 primary, but it does not actively manage the network until the primary instance fails. The KeepAliveD service monitors both primary and standby instances activity status. When the primary instance fails, the standby takes over automatically. The standby instance takes over primary instance to manage the services while primary 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 primary instance.

Both primary 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 primary ESC High Availability instance. The deployed VNFs are connected to both ESC primary and standby instances through another IPv6 network.

ESC HA nodes are managed by KeepAliveD and DRBD (Replication tool to keep the ESC database synchronized) sync network services. While the KeepAliveD service monitors both primary and standby instances status, the DRBD service monitors primary 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

To deploy Cisco Elastic Services Controller (ESC) High Availability (HA), ESC standalone instances can be installed on two separate nodes - Primary and Standby. For more information see, How High Availability Works, on page 1. You can connect the Primary and Standby instances to either a Cinder volume or Replication based volume (DRBD).

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

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

  ESC HA 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 Primary ESC instance. To establish a reliable heartbeat network, it is recommended that the Primary 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 is configured with DRBD and both Primary and Standby instance 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 node are synchronized through the database replication mechanism provided by DRBD.

The table lists the differences between the HA options :

	Internal Storage Based ESC HA	**Replicate External Storage Based ESC HA**
Data sharing method	Data replication between HA 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	• No dependency on specific VIM components. • Flexible to build of HA clusters from commodity hardware, without the requirement for shared-storage.	• 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 Mode on Internal Storage

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

- kad_vip

Note When ESC HA is deployed, the *kad_vip* argument allows end users to access the Primary 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 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_confd_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_confd_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 parameters on each of the standalone ESC VMs. Three parameters "--ha_node_list , --kad_vip, --kad_vif" are all required to configure ESC HA. 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 restart
```

Deploying ESC in High Availability Mode on Replicate External Storage

Replicate external storage ESC HA 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 switchover

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 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_confd_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

ON HA NODE 2:

$ ./bootvm.py <ESC_HA_Node2>\
--user_pass <username>:<password>\
--user_confd_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; the keepalived state on one of ESC VM should be one of ESC VM should be in MASTER state and the other one should be in BACKUP state. You can check ESC HA state by using following command: *$ sudo escadm status --v*.

Configuring the Northbound Interface Access

When you configure ESC HA, you can also specify a virtual Anycast IP address to the HA pair. The northbound interface as well as the service portal uses virtual Anycast IP address to access the ESC Primary HA instance. When deploying ESC HA, 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 with multiple interfaces and to configure the virtual Anycast IP address.

Configuring ESC HA with Multiple Interfaces

You can configure ESC HA 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 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**.

Example configuration steps are shown below:

```
./bootvm.py <esc_ha1> \
--user_pass <username>:<password>
--user_confd_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 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 esc-v6 esc-net --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 10.85.103.0/24:192.168.0.1:eth1 --avail_zone nova: zone name

./bootvm.py esc-ha-1 --image ESC-2_2_x_yyy --net esc-v6 esc-net 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 10.85.103.0/24:192.168.0.1:eth1 --avail_zone nova: zone name

Configuring the ESC HA 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 Primary ESC and send requests to ESC HA service through virtual IP (VIP).

If northbound and both ESC HA 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, assign a floating IP to ESC HA 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 esc-net --name esc_vip --fixed-ip
   subnet_id=esc-subnet,ip_address=192.168.0.87
   ```

2. Deploy ESC HA . See **Configuring High-Availability** 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 through the floating IP and it will connect to the ESC Primary 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 Primary node.

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

Configuring the ESC L3 HA With BGP

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

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

The following procedure is for direct BGP L3 HA boot:

```
/scratch/BUILD-4_3_0_78/BUILD-4_3_0_78/bootvm.py bgp-ha-0 --ha_node_list 10.0.42.222
10.0.61.222 --image ESC-4_3_0_78 --ipaddr 198.18.42.222 10.0.42.222 --user_confd_pass abc:abc
 --gateway_ip 10.0.42.1 --user_pass abc:A@22p0le! --flavor m1.medium --net provider
aaa_service --kad_vif eth1 --kad_unicast_src_ip 10.0.42.222 --kad_unicast_peer 10.0.61.222
 --kad_vri 112 --bgp_local_ip 198.18.42.222 --bgp_anycast_ip 10.0.199.199  --bgp_remote_ip
 198.18.0.2 --bgp_local_as 65012 --bgp_remote_as 65000 --bgp_local_router_id 198.18.42.222

/scratch/BUILD-4_3_0_78/BUILD-4_3_0_78/bootvm.py bgp-ha-1 --ha_node_list 10.0.42.222
10.0.61.222 --image ESC-4_3_0_78 --ipaddr 198.18.61.222 10.0.61.222 --user_confd_pass abc:abc
 --gateway_ip 10.0.61.1 --user_pass abc:A@22p0le! --flavor m1.medium --net provider
aaa_service --kad_vif eth1 --kad_unicast_src_ip 10.0.61.222 --kad_unicast_peer 10.0.42.222
 --kad_vri 112 --bgp_local_ip 198.18.61.222 --bgp_anycast_ip 10.0.199.199  --bgp_remote_ip
 198.18.0.2 --bgp_local_as 65011 --bgp_remote_as 65000 --
bgp_local_router_id 198.18.61.222
```

To configure BGP for ESC HA, 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

```
   ######################################################################
   #   ESC on bgp-001.novalocal is in MASTER state.
   ######################################################################

[admin@bgp-001 ~]$ health.sh
============== ESC HA (MASTER) with DRBD =================
vimmanager (pgid 4007) is running
monitor (pgid 4135) is running
mona (pgid 4167) is running
drbd (pgid 0) is master
snmp (pgid 5375) is running
etsi is disabled at startup
pgsql (pgid 4586) is running
keepalived (pgid 3068) is running
portal (pgid 5315) is running
confd (pgid 4417) is running
filesystem (pgid 0) is running
escmanager (pgid 4615) is running
=======================================
ESC HEALTH PASSED
[admin@bgp-001 ~]$


   ######################################################################
   #   ESC on bgp-002.novalocal is in BACKUP state.
   ######################################################################

[admin@bgp-002 ~]$ health.sh
============== ESC HA (BACKUP) with DRBD =================
vimmanager is stopped
monitor is stopped
mona is stopped
drbd (pgid 0) is backup
snmp is stopped
etsi is disabled at startup
pgsql is stopped
keepalived (pgid 3069) is running
portal is stopped
confd is stopped
filesystem is stopped
escmanager is stopped
=======================================
```

```
ESC HEALTH PASSED
[admin@bgp-002 ~]$
```

Use below values for BGP post configuration:

```
./bootvm.sh <ESC_VM_name> \
--image <ESC_image> \
--ipaddr <static_IP_address1> <IP_address2> <IP_address_3>\
--gateway_ip <gateway IP address of ESC> \
--net <net_id1> <net_id2> <net_id3> \
--esc_params_file <esc_params_file> \
--host_mapping_file <host_mapping_file> \
--avail_zone <openStack_zone> \
--bgp_remote_ip <BGP_remote_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 Post Configuration

1. For each HA 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 instance:

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

To configure BGP for ESC HA, 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 198.18.0.2 --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 198.18.0.2 --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_as*above

`<Esc_ip_address>`must be the ESC VM's IP address configured for BGP advertisement.

`<ESC_AS_#>`same as--*bgp_local_as*shown 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**
  - An HA 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 Primary ESC instance is suspended by OpenStack command, the switch over will be triggered but the cinder volume won't be attached to the new Primary ESC instance. This is not a valid use case for ESC HA.
- **Internal Storage**
  - Two ESC instances have to be deployed to establish the HA solution. The ESC HA 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 parameters, the ESC instance keeps Switching-to-Master 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 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 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.

Troubleshooting High Availability

- 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 ESC HA log at */var/log//esc/esc_haagent.log*
  - The KeepAliveD log at */var/log/messages* by grep keepalived
- Check for DRBD (Replication based ESC HA) for Internal Storage solution:
  - Check the DRBD configuration file at

    ```
    /etc/drbd.d/esc.res
    ```

  - Access the DRBD log

```
/var/log/messages|grep drbd
```