Installing High Availability

This chapter contains the following sections:

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:
          - 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"]
```

- 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-${ESC_IMAGE}.ova
# All valid deployment options:
#          2CPU-4GB
#          4CPU-8GB  (default)
#          4CPU-8GB-2Net
#          4CPU-8GB-3Net
DEPLOYMENT_OPTION="4CPU-8GB-2Net"
deploy_vmware_vm1() {
/usr/bin/ovftool \
--powerOn \
--acceptAllEulas \
--noSSLVerify \
--datastore=$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_KEEPALIVE_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	`cat /opt/cisco/esc/keepalived_state`
ESC Health	`sudo escadm health`
ESC Service Status	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': `sudo escadm status --v` To check the detailed status, check the /var/log/esc/escadm.log

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*