高可用性のインストール

この章は、次の項で構成されています。

- 高可用性アクティブ/スタンバイの概要 （1 ページ）
- 高可用性アクティブ/スタンバイの仕組み （2 ページ）
- ユーザデータを使用した ESC 高可用性アクティブ/スタンバイの展開（HA アクティブ/スタンバイペア） （2 ページ）
- ESC 高可用性アクティブ/スタンバイの展開（スタンドアロンインスタンス） （6 ページ）
- ESC HA アクティブ/スタンバイに関する特記 （8 ページ）
- 高可用性アクティブ/スタンバイのトラブルシューティング （8 ページ）

高可用性アクティブ/スタンバイの概要

ESC は、アクティブ/スタンバイおよびアクティブ/アクティブモデルの形式で高可用性（HA）をサポートします。アクティブ/スタンバイモデルでは、ESC 障害を防止し、サービスの中断を最小限に抑えて ESC サービスを提供するために、ネットワークに 2 つの ESC インスタンスが展開されます。アクティブ ESC インスタンスで障害が発生しても、スタンバイインスタンスが自動的に ESC サービスを引き継ぎます。ESC HA アクティブ/スタンバイ は、次のシングルポイント障害を解決します。

- ネットワーク障害
- 停電
- VM インスタンスのダウン
- スケジュールされたダウンタイム
- ハードウェアに関する問題
- 内部アプリケーションの障害

高可用性アクティブ/スタンバイの仕組み

高可用性アクティブ/スタンバイの展開は、アクティブとスタンバイの 2 つの ESC インスタンスで構成されます。通常の状況では、アクティブ ESC インスタンスによってサービスが提供されます。対応するスタンバイインスタンスはパッシブ状態になります。スタンバイインスタンスは、アクティブインスタンスと常時通信して、アクティブインスタンスのステータスをモニタします。アクティブ ESC インスタンスに障害が発生すると、スタンバイインスタンスが ESC サービスを自動的に引き継ぎ、最小限の中断で ESC サービスの提供を継続します。

スタンバイインスタンスにもアクティブインスタンスのデータベースの完全なコピーが存在しますが、アクティブインスタンスに障害が発生しない限り、スタンバイインスタンスがアクティブにネットワークを管理することはありません。アクティブインスタンスに障害が発生すると、スタンバイが自動的に引き継ぎます。アクティブインスタンスの復元中、スタンバイインスタンスがアクティブインスタンスを引き継ぎ、サービスを管理します。

障害が発生したインスタンスが復元されると、元のアクティブインスタンスを使用してネットワーク管理を再開するためのフェールバック操作を開始できます。

ESCインスタンスは、キープアライブサービスを使用して管理されます。ESCインスタンス間の VM ハンドシェイクは、IPv4 ネットワーク上で キープアライブサービスを介して行われます。

ユーザデータを使用したESC高可用性アクティブ/スタンバイの展開（HA アクティブ/スタンバイペア）

始める前に

- Cisco Elastic Services Controller（ESC）高可用性（HA）アクティブ/スタンバイでは、キープアライブを維持し、アクティブノードとスタンバイノード間でデータベースを複製するためのネットワークが必要です。両方のESC VMには、同じネットワークに接続する少なくとも1つのネットワーク インターフェイスが必要であり、ネットワークを介して相互に通信できる必要があります。
- 2 つの ESC VM が異なるホストとデータストアに配置されることを確認し、シングルポイント障害を防止できるようにします。

ESC HA アクティブ/スタンバイを、次のいずれかの方法で VMware vCenter または vSphere に展開できます。

- ESC HA アクティブ/スタンバイを高可用性アクティブ/スタンバイペアとしてユーザデータを使用して展開する（ESC 4.2 でサポート）
- ESC HA アクティブ/スタンバイを、2 つのスタンドアロンインスタンスとして展開し、POST 設定を使用してそれらを高可用性ペアとして設定します。詳細については、「ESC 高可用性アクティブ/スタンバイの展開（スタンドアロンインスタンス）」のセクションを参照してください。

ESC HA アクティブ/スタンバイを、高可用性アクティブ/スタンバイペアとしてユーザデータを使用して VMware vCenter または vSphere に展開するには、ユーザデータファイルを HA アクティブ/スタンバイインスタンスごとに定義し、次に、各インスタンスのユーザデータを ovftool を介して指定します。ユーザデータのエンコードは、ovftool スクリプトの一連のコマンドを介して行われ、その結果は、ovftool の「–prop:user-data = 」プロパティの変数として設定されます。

（注） 「admin user/password」および「confd user/password」プロパティは、必須の OVF プロパティです。これらのプロパティは、ユーザデータファイルでは定義できません。

- ESC HA アクティブ/スタンバイの 2 つの VM を定義します。

ユーザデータ 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"]
```

ユーザデータ 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"]
```

- 各 VM インスタンスについて、OVFtool を 2 回呼び出す必要があります。各インスタンスは、ハッシュ化されたユーザデータを指す「--prop:user-data」プロパティを提供する必要があります。
- ここでは、172.16.0.0 および 172.16.1.0（フローティング）IP をインスタンスに、172.16.2.0 を KAD_VIP として使用する HA アクティブ/スタンバイインスタンスのペアをブートする例を示しています。

```
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
```

- VM が正常に展開された後に、ESC HA アクティブ/スタンバイのステータスを確認できます。1 つの VM インスタンスがマスターとして起動され、他の VM インスタンスが STANDBY であることがわかります。

ESC高可用性アクティブ/スタンバイの展開（スタンドアロンインスタンス）

VMware vCenter または vSphere で ESC HA アクティブ/スタンバイを展開するには、2 つの別個のスタンドアロンノードを最初にインストールする必要があります。スタンドアロン ESC インスタンスがインストールされた後、次を使用して、これらのノードがアクティブとスタンバイになるように再設定します。

- kad_vip
- kad_vif
- ha_node_list

（注）
- ESC VM ごとに、*escadm* ツールを実行して ESC HA アクティブ/スタンバイパラメータを設定した後、escadm サービスをリロードして再起動する必要があります。
- ESC HA アクティブ/スタンバイを展開する際、kad_vip 引数を使用すると、エンドユーザがアクティブ ESC インスタンスにアクセスできるようになります。

手順

ステップ 1 ESC スタンドアロンインスタンスにログインします。

ステップ 2 管理者ユーザとして、アクティブインスタンスとスタンバイインスタンスの両方で *escadm* ツールを実行し、対応する引数を指定します。

- **kad_vip**：keepalived VIP（仮想 IP）の IP アドレスと keepalived VIP のインターフェイスを指定します（ESC-HA アクティブ/スタンバイ）。
- **kad_vif**：keepalived 仮想 IP と keepalived VRRP のインターフェイスを指定します（ESC-HA アクティブ/スタンバイ）。VIP インターフェイスが引数 *kad_vip* を使用してすでに指定されている場合は、引数 kad_vip を使用して keepalived VRRP のインターフェイスのみを指定することもできます。
- **ha_node_list**：DRDB 同期のため、アクティブ/スタンバイクラスタに含まれる HA アクティブ/スタンバイノードの IP アドレスのリストを指定します。この引数は、レプリケーションベースの HA アクティブ/スタンバイソリューションのみに使用されます。複数のネットワークインターフェイスを持つ ESC インスタンスの場合、IP アドレスは、引数 *--kad_vif* で指定されたネットワーク内にある必要があります。

```
$ 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
```

ステップ 3 再起動後、1 つの ESC VM はアクティブ状態になり、もう 1 つはスタンバイ状態になる必要があります。

ステップ 4 VIP が外部から到達可能になるように、両方の VM で許可されたアドレスペアに VIP を追加します。

ステップ 5 各 ESC インスタンスのステータスを確認します。

```
# sudo escadm status
```

次の表に、ステータスを確認するための他のコマンドをいくつか示します。

ステータス	CLI コマンド
ESC HA アクティブ/スタンバイのロール	`cat /opt/cisco/esc/keepalived_state`

ESC の正常性	`sudo escadm health`
ESC サービスのステータス	詳細情報（VIM マネージャ、SNMP、ポータル、ESC マネージャ、keepalived のステータスなど）を表示するには、「–v」を追加します。 `sudo escadm status --v` 詳細なステータスを確認するには、/var/log/esc/escadm.log をチェックします。

ESC HA アクティブ/スタンバイに関する特記

- HA アクティブ/スタンバイフェールオーバーには、動作可能な管理対象 VNF の数に基づいて約 2 ー 5 分かかります。ESC サービスは、スイッチオーバー時間中は使用できません。
- トランザクション中にスイッチオーバーがトリガーされると、すべての未完了のトランザクションがドロップされます。要求は、ESC からの応答を受信しない場合、ノースバウンド インターフェイスによって再送信される必要があります。

高可用性アクティブ/スタンバイのトラブルシューティング

- ネットワーク障害をチェックします。ネットワークに問題が発生している場合は、次の詳細情報をチェックする必要があります。
  - 割り当てられている IP アドレスは正しいもので、OpenStack 設定に基づいている必要があります。
  - 各ネットワークインターフェイスのゲートウェイで ping が応答する必要があります。
- トラブルシューティングの際には、次のログをチェックします。
  - ESC マネージャログ：*/var/log/esc/escmanager.log*
  - キープアライブログ：*/var/log/messages* （grep keepalived を実行）
  - ESC サービスステータスログ：*/Var/log/esc/escadm.log*