高可用性アクティブ/スタンバイのインストール

この章は、次の項で構成されています。

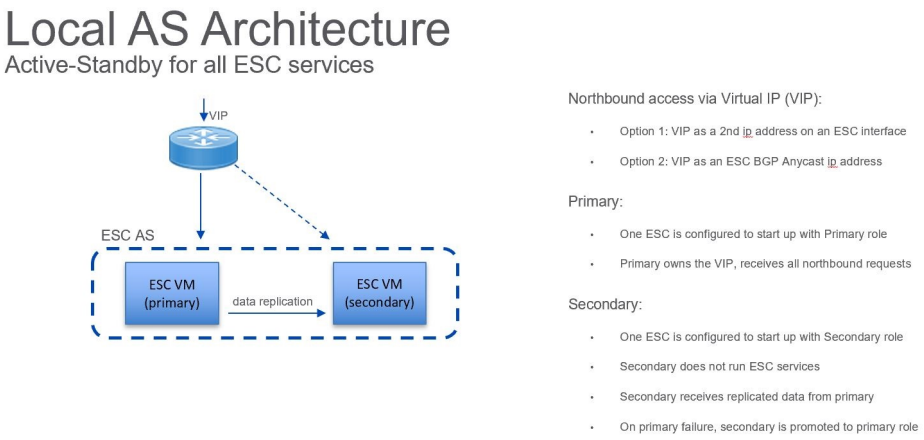
高可用性アクティブ/スタンバイの概要

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

ESC アクティブ/スタンバイのアーキテクチャ

図 *1 : Cisco Elastic Services Controller* アクティブ/スタンバイのアーキテクチャ



ハイ アベイラビリティの仕組み

ESC HA アクティブ/スタンバイネットワークは、ESC HA アクティブ/スタンバイペアの単一のインストールとして設定するか、2 つのスタンドアロン ESC ノードとして展開することができます。2 つのノードは、展開後、再設定を経て HA ペアに変換されます。HA 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。KeepAliveD サービスは、アクティブとスタンバイ両方のインスタンスのアクティビティステータスをモニタします。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たら、必要に応じて手動でスイッチオーバーを開始し、アクティブインスタンスによるネットワーク管理を再開できます。

アクティブとスタンバイ両方の ESC インスタンスは、IPv4 または IPv6 ネットワーク経由でノースバウンド オーケストレーション システムに接続されます。ノースバウンドシステムに

は、現在のアクティブ ESC 高可用性アクティブ/スタンバイインスタンスにアクセスするための一意の仮想 IP アドレスが割り当てられます。展開された VNF は、別の IPv6 ネットワーク経由で ESC のアクティブとスタンバイ両方のインスタンスに接続されます。

ESC HA アクティブ/スタンバイノードは、keepalived および DRBD（ESC データベースの同期を維持するためのレプリケーションツール）同期ネットワークサービスによって管理されます。KeepAliveD サービスは、アクティブとスタンバイ両方のインスタンスのステータスをモニタしますが、DRBD サービスは、アクティブインスタンス DB をモニタして変更内容をスタンバイインスタンス DB に同期します。これら 2 つのサービスは、同じ VIP ネットワークに配置することも、2 つの異なるネットワークに配置することもできます。ESC インスタンス間の VM ハンドシェイクは、IPv4 または IPv6 ネットワーク上の keepalived を使用して行われます。

ESC 高可用性アクティブ/スタンバイの展開

Cisco Elastic Services Controller（ESC）の高可用性（HA）アクティブ/スタンバイを展開するには、2 つの独立したノード（アクティブおよびスタンバイ）に ESC スタンドアロンインスタンスをインストールします。詳細については、ハイ アベイラビリティの仕組み （2 ページ）を参照してください。アクティブインスタンスとスタンバイインスタンスは、Cinder ボリュームまたはレプリケーションベースのボリューム（DRBD）に接続できます。

ESC HA アクティブ/スタンバイを展開する際、次の展開メカニズムを利用できます。

- 内部ストレージ：ESC HA アクティブ/スタンバイが内部ストレージを使用して構成されている場合、アクティブインスタンスとスタンバイインスタンスは個別のデータベースを備え、常に同期されます。このソリューションでは、ESC HA アクティブ/スタンバイはデータベースレプリケーションを利用して設計され、DRBD はディスクレベルのレプリケーション用ツールとして使用されます。アクティブインスタンスのデータベースは、スタンバイインスタンスのデータベースにも同時にデータを伝送するため、外部ストレージを必要としません。アクティブインスタンスで障害が発生した場合、スタンバイインスタンスには、アクティブインスタンスのロールと、同期された固有のデータベースが割り当てられます。

  ESC HA アクティブ/スタンバイは、内部ストレージを利用して展開されます。ESC インスタンスは仮想 IP アドレス（kad_vip 引数）および vrrp インスタンスのインターフェイス（kad_vif 引数）で応答し、アクティブ ESC インスタンスを選択します。信頼性の高いハートビートネットワークを確立するには、アクティブおよびスタンバイ ESC インスタンスを異なる物理ホスト上に配置することが推奨されます。ESC インスタンス間の物理リンクの信頼性（ネットワークインターフェイスの結合など）を考慮することもできます。

- 外部ストレージレプリケーション：このタイプのアーキテクチャでは、ESC HA アクティブ/スタンバイは DRBD を利用して構成されます。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が、2 つの外部ストレージ （OpenStack Cinder ボリューム）にデータを保存します。各 ESC ノードは Cinder ボリュームによって接続され、ESC データファイルは Cinder ボリュームに保存されます。2 つの ESC ノードのデータは、DRBD で提供されるデータベース レプリケーション メカニズムを利用して同期されます。

HA アクティブ/スタンバイのオプションの違いを次の表に示します。

	内部ストレージベースの ESC HA アクティブ/スタンバイ	外部ストレージ レプリケーション ベースの ESC HA アクティブ/スタンバイ
データ共有方法	HA アクティブ/スタンバイ ノード間のデータレプリケーション	2 つの外部ストレージ（Cinder ボリューム）間のデータレプリケーション
インストール方法	インストール後の設定 bootvm のインストール	bootvm のインストール
VIM 対応	OpenStack、VMware、KVM	OpenStack のみ
依存条件	VIM に依存しない	OpenStack Cinder に依存
利点	• 特定の VIM コンポーネントへの依存関係がありません。 • 共有ストレージを必要としないため、市販のハードウェアから HA アクティブ/スタンバイクラスタを柔軟に構築できます。	• データベース レプリケーション メカニズムを利用してデータが同期されます。 • 2 つの Cinder ボリュームが外部ストレージとして使用され、ESC ノードに接続されます。
制限事項	二重障害が発生すると（両方の ESC ノードで問題が発生した場合）、データの一貫性に影響を与える可能性があります。	二重障害が発生すると（両方の ESC ノードで問題が発生した場合）、データの一貫性に影響を与える可能性があります。

内部ストレージを利用した高可用性アクティブ/スタンバイモードの ESC の展開

アクティブおよびスタンバイインスタンスで ESC インスタンスを起動する場合は、次の *bootvm.py* コマンド引数を指定して、内部ストレージに ESC HA アクティブ/スタンバイを展開する必要があります。

- kad_vip

（注） ESC HA アクティブ/スタンバイが展開されると、エンドユーザは *kad_vip* 引数を使用してアクティブ ESC インスタンスにアクセスできます。

- kad_vif

- ha_node_list

これらの引数を指定すると、*bootvm.py* コマンドを使用して、OpenStack で内部ストレージを自動設定できます。*bootvm.py* コマンド引数の使用に関する詳細は、「付録 A：Cisco Elastic Service Controller のインストーラ引数」を参照してください。

ESC HA アクティブ/スタンバイインスタンスを展開するには、両方のノードの bootvm スクリプトで次の引数を指定します。

```
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
```

または

escadm ツールを使用して、各スタンドアロン型 ESC VM で ESC HA アクティブ/スタンバイのパラメータを再設定することもできます。ESC HA アクティブ/スタンバイを構成するには、「--ha_node_list、--kad_vip、--kad_vif」の 3 つのパラメータが必要です。

（注） 次のコマンドを実行して HA アクティブ/スタンバイを構成する前に、両方のインスタンスがスタンドアロン ESC VM の正常性チェックに合格していることを確認してください。

次に例を示します。

```
$ 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
```

外部ストレージレプリケーションを利用した高可用性アクティブ/スタンバイモードの ESC の展開

外部ストレージレプリケーションを利用したESC HAアクティブ/スタンバイでは、データベースストレージに 2 つの Cinder ボリュームが必要です。

始める前に

- 両方の ESC インスタンスが接続するネットワークおよび IP アドレス
- HAアクティブ/スタンバイのスイッチオーバー用キープアライブインターフェイスおよび仮想 IP

手順

ステップ 1 OpenStack に 2 つの Cinder ボリュームを作成します。Cinder ボリュームサイズは 3 GB に設定する必要があります。

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

ステップ 2 作成した Cinder ボリュームのステータスを確認し、展開用の UUID を見つけます。

```
$ cinder list
```

ステップ 3 ESC HAアクティブ/スタンバイインスタンスを展開します。両方のノードで、次の引数を指定した bootvm スクリプトを使用します。

```
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
```

ステップ 4 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。ESC HA アクティブ/スタンバイの状態を確認するには、*$ sudo escadm status --v* コマンドを使用します。

ノースバウンド インターフェイス アクセスの設定

ESC HA アクティブ/スタンバイを設定する場合は、HA アクティブ/スタンバイペアに仮想エニーキャスト IP アドレスを指定することもできます。ノースバウンドインターフェイスおよびサービスポータルは、仮想エニーキャスト IP アドレスを使用して ESC アクティブ HA アクティブ/スタンバイインスタンスにアクセスします。ESC HA アクティブ/スタンバイを展開する場合は、*./bootvm.py* スクリプトで次の引数を使用します。

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

これらの引数のさらなる詳細については、「**付録 A**：**Cisco Elastic Services Controller インストーラの引数**」のセクションを参照してください。

ここでは、複数のインターフェイスを使用して ESC HA アクティブ/スタンバイを設定し、仮想エニーキャスト IP アドレスを設定する方法について説明します。

複数のインターフェイスを使用した ESC HA アクティブ/スタンバイの設定

データ同期と VNF モニタリングのために、ネットワーク インターフェイスで DRDB 同期と VRRP ハートビート ブロードキャストを使用して ESC HA アクティブ/スタンバイを設定できます。追加のネットワーク インターフェイスを使用して、ノースバウンドアクセス用の仮想 IP を割り当てることができます。ESC HA アクティブ/スタンバイノードで複数のインターフェイスを設定するには、--ha_node_list、--kad_vip、--kad_vif 引数を使用して、これらの複数のネットワーク インターフェイス設定を指定します。これらの引数の詳細については、「**付録 A**：**Cisco Elastic Services Controller インストーラの引数**」のセクションを参照してください。

（注） KeepAlived は、IPv6 VRRP インスタンスで単一の IPv4 VIP アドレスをサポートしていません。

設定手順の例を次に示します。

```
./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> \
```

同様に、3 つのネットワーク インターフェイスを ESC HA アクティブ/スタンバイノードに設定できます。次に、3 つのインターフェイス設定の例と前提条件を示します。

- ネットワーク 1 は、ノースバウンド接続に使用される IPv6 ネットワークです。ESC VIP はこのネットワークに割り当てられ、Orchestrator は ESC VIP を使用して要求を ESC に送信します。
- ネットワーク 2 は、ESC 同期トラフィック（DRDB 同期）と VRRP ハートビートに使用される IPv4 ネットワークです。このネットワークは、OpenStack 接続および VNF モニタリングにも使用されます。
- ネットワーク 3 は、管理に使用される別の IPv4 ネットワークです。SA、rsyslog などでは、このネットワークを使用して ESC を管理できます。

./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

ESC HA アクティブ/スタンバイ仮想 IP アドレスの設定

このオプションでは、kad_vip 引数の値は仮想 IP である必要があります。これにより、サービスポータルとノースバウンドがアクティブ ESC にアクセスし、仮想 IP（VIP）を介して ESC HA アクティブ/スタンバイサービスに要求を送信できます。

ノースバウンドと両方の ESC HA アクティブ/スタンバイノードが同じネットワークにある場合は、仮想 IP（VIP）を介して直接接続できます。ノースバウンドが ESC HA アクティブ/スタンバイと同じネットワーク上にない場合は、次の手順を使用して、フローティング IP を ESC HA アクティブ/スタンバイ VIP に割り当てます。

1. ESC の kad_vip の接続先と同じネットワークに VIP アドレス（kad_vip）を使用してポートを作成します。

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

2. ESC HA アクティブ/スタンバイを展開します。「OpenStack で ESC をインストール」の「**高可用性アクティブ/スタンバイの設定**」のセクションを参照してください。

（注） *kad_vip* が上記で作成したポートと同じ IP アドレスであることを確認してください。

3. 上記で作成したポートにフローティング IP を関連付けます。最初の uuid はフローティング IP ID で、2 つ目の uuid はポート ID です。

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

   フローティング IP を介して HA アクティブ/スタンバイにアクセスすると、ESC アクティブノードに接続されます。

4. ポータルアクセスの場合、ブラウザからキープアライブネットワークにアクセスできること、および仮想 IP がアクティブノードのポータルにアクセスするための IP アドレスであることを確認してください。

   たとえば、VIP が 192.0.2.254 の場合、https://192.0.2.254:9001/ で ESC HA アクティブ/スタンバイポータルにアクセスします。

BGP を使用した ESC L3 HA アクティブ/スタンバイの設定

ESC HA アクティブ/スタンバイの BGP を設定するには、次の 2 つのオプションがあります。

1. BGP を使用した ESC HA アクティブ/スタンバイ L3 の直接起動
2. 既存 ESC HA アクティブ/スタンバイペアからの POST 設定の使用

ESC HA アクティブ/スタンバイの BGP を設定するには、次のネットワークパラメータが必要です。

- BGP リモート IP
- BGP エニーキャストルーティングのインターフェイスの IP
- ルーティング設定の BGP ローカル AS 番号
- ルーティング設定用の BGP リモート AS 番号
- BGP ルーティングの設定

- --bgp_local_ip
- --bgp_local_router_id

（注） ネイバーを使用して BGP ルータを設定し、再起動する必要があります。ルータがエニーキャスト IP に ping できることを確認します。

BGP ルータで、2 つのネイバーを設定します。次の BGP 設定は、Bird ルータ向けに設計されています。この設定は、ルータ固有です。ルータのタイプごとに、手順は異なります。

次の設定は、bootvm コマンドによって指定されます。

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

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

BGP オプションを使用した ESC VM の起動

```
[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
```

BGP POST 設定には次の値を使用します。

```
./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>
```

それぞれの説明は次のとおりです。

```
--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
```

BGP HA アクティブ/スタンバイ ポスト コンフィギュレーションの設定

1. HA のアクティブ/スタンバイインスタンスごとに、ネットワーク インターフェイス ファイルを作成します。

   ```
   # 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. HA のアクティブ/スタンバイインスタンスごとに、次のようにします。

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

ESC HA アクティブ/スタンバイの BGP を設定するには、次に示すように、ESC 仮想マシンに escadm ツールを使用します。

```
$ 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
```

例：

```
[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
```

BGP ルータの設定

BGP ルータを設定するには、BGP ルータにログインして BGP エニーキャストルーティングを設定します。次のパラメータが必要です。

<Router_AS_#> は上記の *--bgp_remote_as* と同様です

<Esc_ip_address> は、BGP アドバタイズメント用に設定された ESC VM の IP アドレスである必要があります。

<ESC_AS_#> は上記の *--bgp_local_as* と同様です

```
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
```

特記事項

- **ESC HA アクティブ/スタンバイ**
  - HA アクティブ/スタンバイのフェールオーバーには 2 ～ 5 分ほどかかります。スイッチオーバーの間、ESC サービスは使用できなくなります。
  - 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。要求に対する ESC からの応答が受信されない場合、ノースバウンドから要求が再送信される必要があります。
- **外部ストレージ**

- アクティブ ESC インスタンスが OpenStack コマンドによって中断された場合は、スイッチオーバーがトリガーされますが、Cinder ボリュームは新しいアクティブ ESC インスタンスに接続されません。これは ESC HA アクティブ/スタンバイの有効な使用例ではありません。

- **内部ストレージ**
  - HA アクティブ/スタンバイソリューションを確立するには、2 つの ESC インスタンスを展開する必要があります。両方の ESC インスタンスが正常に展開され、相互に接続できる場合に、ESC HA アクティブ/スタンバイは動作を開始します。HA アクティブ/スタンバイのパラメータを使用して 1 つの ESC インスタンスのみを展開した場合、ESC インスタンスの状態は「アクティブ状態に切り替え中」のままになり、ピアに到達するまでサービスを提供できなくなります。
  - スプリットブレインのシナリオは、可能性は非常に低いとはいえ、ESC HA アクティブ/スタンバイソリューションでも発生する場合があります。

- **ETSI 固有の注意事項**

  ESC は、欧州電気通信標準化機構（ETSI）によって定義された ETSI MANO ノースバウンド API を NFV 管理およびオーケストレーションに対してサポートします。ETSI MANO API は、REST アーキテクチャに基づく別のプログラマチック インターフェイスです。詳細については、『*Cisco Elastic Services Controller User Guide*』の「ETSI MANO Compliant Lifecycle Operations」を参照してください。HA アクティブ/スタンバイモードの ESC で ETSI サービスを有効にする場合は、次の注意事項を考慮してください。
  - *etsi-vnfm.properties* ファイル内の server.address の値は、仮想 IP（VIP）アドレスに設定する必要があります。この IP アドレスは、API コールバックを使用した ETSI サービスへの応答に使用できます。仮想 IP アドレスが指定されていない場合、ETSI サービスの起動に失敗することがあります。
  - ETSI VNFM サービスと escadm スクリプトは、security.user.name プロパティと security.user.password プロパティの値を生成して保持します。手動で変更しないでください。security.user.password がエンコードされます。

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は、ピン可能である必要があります。

- 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。
  - */var/log/esc/escadm.log* の ESC 管理ログ
  - */var/log/esc/escmanager.log* の ESC マネージャのログ
  - */var/log/esc/elector-{pid}.log* の AA エレクタのログ
- 内部ストレージソリューションの DRBD（レプリケーションベース ESC HA アクティブ/スタンバイ）を確認します。
  - 次の DRBD 設定ファイルを確認します。

    ```
    /etc/drbd.d/esc.res
    ```
  - DRBD ログへのアクセス
  - `/var/log/messages|grep drbd`
- CLIを使用してログファイルを収集するには、すべてのESCノードで次のコマンドを使用します。

  ```
  sudo escadm log collect
  ```