



Cisco Elastic Services Controller の高可用性に関するトラブルシューティング

- [Cisco Elastic Services Controller の高可用性に関するトラブルシューティング \(1 ページ\)](#)
- [Cisco Elastic Services Controller の高可用性に関するトラブルシューティングの概要 \(1 ページ\)](#)
- [高可用性アクティブノードが Switching-to-Active 状態のままになる \(3 ページ\)](#)
- [両方の HA VM インスタンスの Keepalived サービス状態がバックアップ状態のままになる \(3 ページ\)](#)
- [Cisco Elastic Services Controller HA の動作が遅い \(4 ページ\)](#)
- [VIP で Cisco Elastic Services Controller HA にアクセスできない \(5 ページ\)](#)
- [バックアップVMのステータスを表示しないアクティブなVMのステータスチェック \(10 ページ\)](#)

Cisco Elastic Services Controller の高可用性に関するトラブルシューティング

ESCHA は多くのコンポーネント/サービスで構成されており、セルフヘルスチェックのモニタリングを続けます。ESC マイクロサービスに障害が発生すると、HA 同期やその他の関連する問題が発生します。

Cisco Elastic Services Controller の高可用性に関するトラブルシューティングの概要

ESC HA の一般的なトラブルシューティング項目を次に示します。

問題： ネットワークの問題

ソリューション： ネットワークに問題がある場合は、次の項目を確認してください。

- 両方の ESC ノードの静的 IP アドレスが、OpenStack 構成に基づいて正しく設定されていて、各ノードが他のノードにアクセスできること。
- 各ネットワークインターフェイスのゲートウェイに各インスタンスからアクセスできること。
- 仮想 IP アドレス (kad_vip) が、マスターノードから ping 可能なこと (kad_vip を見つけるには、「sed -n '/virtual_ipaddress/{n;p;}' /etc/keepalived/keepalived.conf」を実行します)。

ログの確認 :

ESC HA に関するトラブルシューティングの際に確認する一部のログとログの場所を次に示します。

- ESC マネージャログ : /var/log/esc/escmanager.log
- ESC サービスの起動/停止に関する ESC HA ログ : /var/log/esc/esc_haagent.log (ESC 2.X) および /var/log/esc/escadm.log (ESC 3.X)
- exabgp ログ : /var/log/exabgp.log

Keepalived の構成とログの確認 :

次のパスで Keepalived の構成を確認します。

- /etc/keepalived/keepalived.conf にある構成ファイルをチェックして、Keepalived 構成を確認できます。
- Keepalived のログは、grep keepalived または vrrp を実行すると /var/log/messages に格納されます。

DRBD の構成とログの確認 :

次のパスで DRBD の構成を確認します。

- DRBD の構成を確認するには、/etc/drbd.d/esc.res にあるファイルをチェックします。
- DRBD のログは、grep drbd を実行すると /var/log/messages に格納されます。

BGP の構成の確認 :

BGP の構成を確認します。

- BGP の構成は、インストール引数および ASR の構成と同じである必要があります。
- BGP の構成は、/opt/cisco/esc/esc-scripts/bgp-sa/exabgp/neighbor_init.conf にあるファイルをチェックすることで確認できます。

高可用性アクティブノードが **Switching-to-Active** 状態のままになる

ESC 高可用性 (HA) クラスタは、起動時に問題が発生する場合があります。考えられる問題を次に示します。

問題：

- ESC HA ノードが、初回インストール中にピアに到達できない。初めてアクティブに切り替えるときに、ESC HA がピアに到達できることを確認します。
- データベースの問題（データベースの移行、データベースファイルの破損など）が原因で、ESC サービス (tomcat/escmanager) を正しく起動できない。
- CDB ファイルが破損しているため、confd を開始できない。
- ファイルシステムの問題（ディスク容量が 100% フル状態）により、PostgreSQL を開始または初期化できない。
- ESC ノード間の接続が低速である（MTU の問題）。

確認：

前述の問題を解決するには、次の項目を確認してください。

- ESC アクティブノードとスタンバイノード間の接続。初回インストールでは、ESC アクティブ (escadm) サービスは、スタンバイノードに到達できない場合は起動しません。両方の ESC ノードが正常に展開され、相互に到達できることを確認します。
- /var/log/esc/esc_haagent.log (ESC 2.X) または /var/log/esc/escadm.log (ESC 3.X 以降) で ESC ログを確認します。ほとんどの場合、ESC サービスがブロックされた理由と、うまく起動しなかったステップまたはサービスが表示されます。
- esc_service/escadm および PostgreSQL が開始されている場合は、/var/log/esc/escmanager.log のログでエラーメッセージを確認してください。

両方の HA VM インスタンスの **Keepalived** サービス状態がバックアップ状態のままになる

問題：

ESC HA には、アクティブ、バックアップ、障害、および停止の 4 つの異なる状態があります。バックアップ状態は、停止からアクティブへ、または障害からアクティブへの遷移状態です。両方の ESC VM がバックアップ状態に留まる可能性があります。通常は長くは続きませんが、両方の ESC HA VM の Keepalived 状態が 2 分以上バックアップ状態になっている場合は、

問題が発生している可能性があります。ただし、ネットワークで VRRP ブロードキャスト干渉が生じる可能性があります。

ソリューション：

いずれかの ESC VM で以下のコマンドを実行して、この問題を診断します。

```
$ sudo tcpdump -vvv -n -i ethX host ff02::12 (for IPv6 network)
$ sudo tcpdump -vvv -n -i ethX host 224.0.0.18 (for IPv4 Network)
```

最初の tcpdump コマンドは、ESC のハートビートネットワークで VRRP ブロードキャストパケットをリッスンします。ハートビート ネットワーク インターフェイスを使用して、前述のコマンドの ethX を (eth0 など) に置き換えます。これで、サブネット内の任意のノードによって生成される VRRP ブロードキャストを ESC VM でリッスンできるかどうかの情報を得られるので、ネットワーク内で VRRP ブロードキャストを実行しているユーザーを確認できます。次に例を示します。

```
# sudo tcpdump -vvv -n -i eth0 host 224.0.0.18
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
21:40:37.269728 IP (tos 0xc0, ttl 255, id 16606, offset 0, flags [none], proto VRRP (112), length 40)
    152.16.3.76 > 224.0.0.18: vrrp 152.16.3.76 > 224.0.0.18: VRRPv2, Advertisement, vrid 78, prio 101, authtype none, intvl 5s, length 20, addrs: 152.16.3.78
21:40:37.271332 IP (tos 0xc0, ttl 255, id 63866, offset 0, flags [none], proto VRRP (112), length 40)
    152.16.7.228 > 224.0.0.18: vrrp 152.16.7.228 > 224.0.0.18: VRRPv2, Advertisement, vrid 230, prio 101, authtype none, intvl 5s, length 20, addrs: 152.16.7.230
21:40:38.269976 IP (tos 0xc0, ttl 255, id 49799, offset 0, flags [none], proto VRRP (112), length 40)
    152.16.3.61 > 224.0.0.18: vrrp 152.16.3.61 > 224.0.0.18: VRRPv2, Advertisement, vrid 74, prio 101, authtype none, intvl 5s, length 20, addrs: 152.16.3.74
21:40:39.271020 IP (tos 0xc0, ttl 255, id 20946, offset 0, flags [none], proto VRRP (112), length 40)
    152.16.1.195 > 224.0.0.18: vrrp 152.16.1.195 > 224.0.0.18: VRRPv2, Advertisement, vrid 193, prio 101, authtype none, intvl 5s, length 20, addrs: 152.16.1.193
21:40:42.270541 IP (tos 0xc0, ttl 255, id 16607, offset 0, flags [none], proto VRRP (112), length 40)
```

ソリューション：

他の VM やマシンが ESC HA 構成と同じ VRID でブロードキャストを実行していないことを確認してください。実行されている場合、ESC HA ハートビートに干渉が発生し、両方の ESC HA VM がバックアップ状態のままになります。次のコマンドを実行して、ESC HA の VRID 値を見つけます。

```
$ cat /etc/keepalived/keepalived.conf | grep virtual_router_id
```

ESC HA の VRID がサブネット内の他のシステムによって使用されている場合は、bootvm.py 引数に --kad_vri の値を指定します。

Cisco Elastic Services Controller HA の動作が遅い

問題：

一部の OpenStack 環境では、Neutron の設定が異なり、ネットワークのスループットが非常に遅くなります。そのような場合、ESC VM では、ネットワーク インターフェイスの MTU を 1500 から 1450 に減らす必要があります。



- (注) ESC のネットワーク インターフェイスの MTU 値は、VIM、NFVO、管理ジャンプボックスなど、ESC が直接通信するコンポーネントを管理する VM の他のネットワーク インターフェイスの MTU と一致している必要があります。

ソリューション：

MTU 値を減らすには、次の手順を実行します。

- 変更するインターフェイスを特定し、`/etc/sysconfig/network-scripts/ifcfg-ethX` に移動します。X は、変更するインターフェイス番号を表します。

- VIM などのテキストエディタを使用して、MTU 項目を追加または編集します。

```
mtu=1450
```

- 次のコマンドを使用して、ネットワーク インターフェイスを再起動します。

```
# network service restart
i.e: sudo ifdown eth0 && sudo ifup eth0
```

VIP で Cisco Elastic Services Controller HA にアクセスできない

VIP が ESC インスタンスのポートの `allowed_address_pairs` にあることを確認します。

始める前に

問題 1：

VIP で ESC HA にアクセスできない

ESC VIP は ESC HA インスタンス間でフロートし、接続を ESC マスターにリダイレクトします。

検証およびトラブルシューティング：

VIP が OpenStack 環境で機能しない場合は、次の 2 つの項目を確認してください。

- ESC インスタンスの元のインターフェイスに、許可されたアドレスペアとして VIP を割り当てる必要があります。
- ESC のインターフェイスのポートをチェックし、許可されたアドレスペアの設定が正しいことを確認します。

手順

- ステップ 1** VIP フェールオーバー用の ESC インターフェイスのポート UUID を見つけます。次の例では、152.16.3.76 が IP です。

```
$ neutron port-list | grep 152.16.3.76
| 80d7e031-04cd-4fb7-8f48-dcbcd88bd8685 | | fa:16:3e:87:c9:e5 | {"subnet_id":
"7b2ce63b-eb20-4ff8-8d49-e46ee8dde0f5", "ip_address": "152.16.3.76"}
```

- ステップ 2** ポートの許可されたアドレスペアを確認し、ポートの許可されたアドレスペアに VIP を追加します。

次に例を示します。

```
$ neutron port-show 80d7e031-04cd-4fb7-8f48-dcbcd88bd8685
```

Field	Value
admin_state_up	True
allowed_address_pairs	
binding:host_id	my-ucs-64
binding:profile	{}
binding:vif_details	{"port_filter": true, "ovs_hybrid_plug": false}
binding:vif_type	ovs
binding:vnic_type	normal
created_at	2017-12-13T21:16:56
description	
device_id	b895cd19-2491-4ac0-b4b5-087a4f76b701
device_owner	compute:None
extra_dhcp_opts	
fixed_ips	{"subnet_id": "7b2ce63b-eb20-4ff8-8d49-e46ee8dde0f5", "ip_address": "152.16.3.76"}
id	80d7e031-04cd-4fb7-8f48-dcbcd88bd8685
mac_address	fa:16:3e:87:c9:e5
name	
network_id	c7fafeca-aa53-4349-9b60-1f4b92605420
port_security_enabled	True
security_groups	e8e9e10c-0e73-4e01-b364-115f785f787d
status	ACTIVE
tenant_id	d972982b511d4caa973f2ab71b58c2fe

```

| updated_at          | 2017-12-13T21:17:20
+-----+-----+
$ neutron port-update <your_esc_port_id> --allowed-address-pairs type=dict list=true
ip_address=<your_vip_address>
For Example:
$ neutron port-update 80d7e031-04cd-4fb7-8f48-dcbcd8685 --allowed-address-pairs
type=dict list=true ip_address=152.16.3.78
Updated port: 80d7e031-04cd-4fb7-8f48-dcbcd8685

$ neutron port-show 80d7e031-04cd-4fb7-8f48-dcbcd8685
+-----+-----+
| Field                | Value
+-----+-----+
| admin_state_up      | True
| allowed_address_pairs | {"ip_address": "152.16.3.78", "mac_address": "fa:16:3e:87:c9:e5"}
| binding:host_id     | my-ucs-64
| binding:profile     | {}
| binding:vif_details | {"port_filter": true, "ovs_hybrid_plug": false}
| binding:vif_type    | ovs
| binding:vnic_type   | normal
| created_at          | 2017-12-13T21:16:56
| description         |
| device_id           | b895cd19-2491-4ac0-b4b5-087a4f76b701
| device_owner        | compute:None
| extra_dhcp_opts     |
| fixed_ips           | {"subnet_id": "7b2ce63b-eb20-4ff8-8d49-e46ee8dde0f5",
"ip_address": "152.16.3.76"}
| id                  | 80d7e031-04cd-4fb7-8f48-dcbcd8685
| mac_address         | fa:16:3e:87:c9:e5
| name                |
| network_id          | c7fafeca-aa53-4349-9b60-1f4b92605420
| port_security_enabled | True
| security_groups     | e8e9e10c-0e73-4e01-b364-115f785f787d
| status              | ACTIVE
| tenant_id           | d972982b511d4caa973f2ab71b58c2fe
| updated_at          | 2018-01-29T21:35:17
+-----+-----+

```

次のタスク

他の VM が VIP IP アドレスを引き継ぐ：

このようなシナリオでは、VIP IP アドレスを引き継いだユーザーを調べる必要があります。ユーザーがわかったら、IP アドレスをリリースするか、HA VIP 用に別の IP アドレスを選択します。使用している VIP が安全で、誰にも引き継ぎされないように、VIP を占有するポートを作成できます。VIP アドレスを予約するには、以下のコマンドを実行します。

```
$ neutron port-create <network_name> --fixed-ip ip_address=<your_vip_address> --name kad-vip
```

For example:

```
$ neutron port-create esc-net --fixed-ip ip_address=152.16.3.78 --name kad-vip
Created a new port:
```

Field	Value
admin_state_up	True
allowed_address_pairs	
binding:host_id	
binding:profile	{}
binding:vif_details	{}
binding:vif_type	unbound
binding:vnic_type	normal
created_at	2018-01-29T21:53:33
description	
device_id	
device_owner	
extra_dhcp_opts	
fixed_ips	{ "subnet_id": "7b2ce63b-eb20-4ff8-8d49-e46ee8dde0f5", "ip_address": "152.16.3.78" }
id	3c037a4b-4245-4554-adf5-56ca6bbffa98
mac_address	fa:16:3e:4e:f2:96
name	kad-vip
network_id	c7fafeca-aa53-4349-9b60-1f4b92605420
port_security_enabled	True
security_groups	e8e9e10c-0e73-4e01-b364-115f785f787d
status	DOWN
tenant_id	d972982b511d4caa973f2ab71b58c2fe
updated_at	2018-01-29T21:53:33

VIP が管理ネットワークとは別のネットワークにある :

ESC HA 構成では、次の 3 つの構成パラメータ (bootvm.py 引数) が提供されます。

- **--ha_node_list** : アクティブ/スタンバイクラスタ内の HA ノードの IP アドレスのリスト。複数のネットワーク インターフェイスを持つ ESC ノードの場合、これらの IP は、データ同期に使用されるネットワーク内のアドレスである必要があります。この引数は、レプリケーションベースの HA ソリューションのみに使用されます。次に例を示します。

```
--ha_node_list 192.168.0.12 192.168.0.22
```

- **--kad_vip** : keepalived VIP (仮想 IP) の IP アドレスと keepalived VIP (ESC 2.2) のインターフェイス。次に例を示します。

```
-kad_vip 10.20.0.194
```

ESC 2.2 以降、VIP のインターフェイスは次の形式で指定されます。

```
--kad_vip 10.20.0.194:eth2 or --kad_vip [2001:cc0:2020::fc]:eth2;
```

- **--kad_vif** :

- keepalived VRRP および VIP (ESC 1.0 ~ ESC 2.1) のインターフェイス。
- keepalived VRRP のインターフェイス (VIP インターフェイスが kad_vip (ESC 2.2) で指定されている場合のみ) 。次に例を示します。

```
--kad_vif eth0
```

別のインターフェイスでVIPを使用します。つまり、同期インターフェイス (kad_vif) のネットワーク/インターフェイス、--ha_node_list、および--kad_vifを1つのネットワーク/インターフェイス (eth1) で設定し、--kad_vipを別のネットワーク/インターフェイス (eth0) で設定する場合とは違うインターフェイスで使用します。

たとえば、次の bootvm.py コマンドの場合、ESC HA はデータ同期とハートビートには eth1 (192.168.0.0/24) を使用し、VIP アクセスには eth0 (192.168.5.0/24) を使用します。VIP 192.168.5.200 は、ネットワーク (192.168.5.0/24) 内の ESC ノード間でフロートします。

```
./bootvm.py esc-ha-1 --image ESC-2_2_8_106 --net lab-net-0 esc-net --gateway_ip 192.168.0.1 --ipaddr 192.168.5.239 192.168.0.239 --ha_node_list 192.168.0.239 192.168.0.243 --kad_vip 192.168.5.200/24:eth0 --kad_vif eth1 --ha_mode drbd --route 10.85.103.0/24:192.168.0.1:eth1 --avail_zone nova:my-ucs-26
./bootvm.py esc-ha-0 --image ESC-2_2_8_106 --net lab-net-0 esc-net --gateway_ip 192.168.0.1 --ipaddr 192.168.5.243 192.168.0.243 --ha_node_list 192.168.0.239 192.168.0.243 --kad_vip 192.168.5.200/24:eth0 --kad_vif eth1 --ha_mode drbd --route 10.85.103.0/24:192.168.0.1:eth1 --avail_zone nova:my-ucs-27
```

バックアップ VM のステータスを表示しないアクティブな VM のステータスチェック

ESC HA のハートビートは VRRP プロトコルに基づいています。VRRP プロトコルに基づき、ESC のアクティブな VM はバックアップ VM インスタンスのステータスを認識しないため、アクティブな VM が動作している限り、ESC サービスは正常に機能するため、ステータスチェックにはバックアップ VM ステータスは含まれません。

バックアップ VM のステータスを確認する場合は、ESC のアクティブな VM で次のコマンドを実行します。

```
$ sudo cat /proc/drbd
version: 8.4.10-1 (api:1/proto:86-101)
GIT-hash: a4d5de01fffd7e4cde48a080e2c686f9e8cebf4c build by abcbuild@, 2017-09-15 14:23:22

1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
   ns:5883476 nr:3012 dw:5886500 dr:378689 al:26 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f
   oos:0
```

ro に Primary/Secondary と表示され、ds に UpToDate/UpToDate を表示されていることを確認します。これは、バックアップがアクティブな VM に接続されていて、アクティブとバックアップ間の同期が良好であることを意味しています。次の例は、バックアップ VM が切断された日時を示しています。

```
$ sudo cat /proc/drbd
version: 8.4.10-1 (api:1/proto:86-101)
GIT-hash: a4d5de01fffd7e4cde48a080e2c686f9e8cebf4c build by abcbuild@, 2017-09-15 14:23:22

1: cs:WFConnection ro:Primary/Unknown ds:UpToDate/DUnknown C r-----
   ns:5888880 nr:3012 dw:5891912 dr:378689 al:26 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f
   oos:84
```

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。