Cisco Elastic Services Controller トラブルシューティング ガイド

初版：2022 年 11 月 25 日

最終更新：2020 年 3 月 11 日

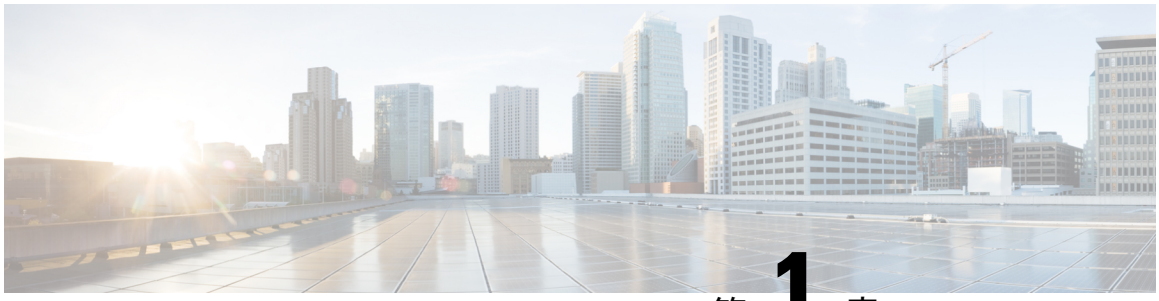
シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（ **www.cisco.com/jp/go/safety_warning/** ）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

目 次

第 I 部

Cisco Elastic Services Controller に関するトラブルシューティング



第 1 章

Cisco Elastic Services Controller のインストールに関するトラブルシューティング

Cisco Elastic Services Controller のインストールに関するトラブルシューティングの概要

ESC では、OpenStack および Libvirt（KVM）環境でのインストールに bootvm.py と呼ばれる Python ベースのスクリプトが使用されます。boovtm.py スクリプトの全引数については、Cisco Elastic Services Controller インストールおよびアップグレードガイド [英語] を参照してください。ESC イメージとともにリリースされた特定の bootvm.py を使用することが重要です。

bootvm.py は、ESC OpenStack インストール用の Python および OpenStack クライアントに依存しています。bootvm.py を実行する予定の環境に、Python および OpenStack クライアントがインストールされていることを確認します。

Cisco Elastic Services Controller のインストールの OpenStack ログイン情報が機能しない

問題に関する説明：

bootvm.py を実行して ESC をインストール中にエラーが発生する場合があります。一般的なエラーの 1 つは次のとおりです。

- OpenStack ログイン情報のエラー

説明：

bootvm.py の実行後に長い Python スタックトレース情報が表示された場合、エラーメッセージの下部にある数行を確認する必要があります。次に例を示します。

```
Unauthorized: The request you have made requires authentication. (HTTP 401) (Request-ID:
 req-e93d90b0-aced-4b88-b4ca-bcc3d88e8bc0)
The request you have made requires authentication. (HTTP 401) (Request-ID:
req-e93d90b0-aced-4b88-b4ca-bcc3d88e8bc0) -- Booting up ESC VM has failed.
```

ソリューション：

このようなシナリオでは、bootvm.py 引数またはグローバル環境（bootvm.py 引数で指定していない場合）で OpenStack ログイン情報を確認する必要があります。

以下に、グローバル環境を介して OpenStack ログイン情報パラメータを確認する例を示します。

```
$ env | grep OS_
OS_USER_DOMAIN_NAME=default
OS_IMAGE_API_VERSION=2
OS_PROJECT_NAME=admin
OS_IDENTITY_API_VERSION=3
OS_PASSWORD=cisco123
OS_AUTH_TYPE=password
OS_AUTH_URL=http://10.85.103.145:35357/v3
OS_USERNAME=admin
OS_TENANT_NAME=admin
OS_PROJECT_DOMAIN_NAME=default
```

他の OpenStack クライアント（OpenStack、Nova、Neutron など）と同様、bootvm.py は、OpenStack に ESC をインストールするために使用されます。bootvm.py の次の引数を使用して、OpenStack ログイン情報を ESC インストーラに渡すことができます。

```
--os_auth_url
--os_username
--os_password
--os_tenant_name
--os_project_name
--os_user_domain_name
--os_project_domain_name
--os_identity_api_version

--bs_os_auth_url
--bs_os_username
--bs_os_password
--bs_os_tenant_name
--bs_os_project_name
--bs_os_user_domain_name
--bs_os_project_domain_name
--bs_os_identity_api_version
```

`bs_` で始まるブートストラップ引数は、OpenStack での ESC インストールにのみ使用され、`os_` で始まる引数は、（ESC 3.x のデフォルトの VIM コネクタとして）ESC が VNF ライフサイクル管理を実行するために使用されます。

これらの引数を指定しない場合、ESC は、ESC のインストールと VNF ライフサイクル管理の両方に対して、Linux のグローバル環境変数から同じ OpenStack ログイン情報を使用します。OpenStack クライアントと同様に、OpenRC ファイルを作成し、そのファイルをソースにしてグローバル環境変数を追加できます。

OpenStack V2 API の場合、次の項目をグローバル環境変数にエクスポートする必要があります。

```
OS_PASSWORD
OS_AUTH_URL
OS_USERNAME
OS_TENANT_NAME
```

OpenStack V3 API の場合、OpenStack V3 API を使用するには OS_IDENTITY_API_VERSION=3 を設定する必要があります。次の項目をグローバル環境変数にエクスポートする必要があります。

```
OS_USER_DOMAIN_NAME
OS_PROJECT_DOMAIN_NAME
OS_PROJECT_NAME
OS_TENANT_NAME
OS_PASSWORD
OS_AUTH_URL
OS_USERNAME
OS_IDENTITY_API_VERSION
```

Cisco Elastic Services Controller のインストール中に証明書の検証が失敗する

問題に関する説明：

OpenStack が自己署名証明書を使用して設定されているが、ESC インストール用の ca_cert ファイルを提供していない場合、次のエラーが発生する可能性があります。

```
SSLError: SSL exception connecting to https://10.85.103.49:35357/v3: [SSL:
CERTIFICATE_VERIFY_FAILED] certificate verify failed (_ssl.c:590)
SSL exception connecting to https://10.85.103.49:35357/v3: [SSL: CERTIFICATE_VERIFY_FAILED]
 certificate verify failed (_ssl.c:590) -- Booting up ESC VM has failed.
```

ソリューション：

bootvm.py は、特定の CA 証明書の ESC インストールに対するコマンドラインで渡される引数を提供しません。OpenStack エンドポイントが https（OS_AUTH_URL を確認）と自己署名証明書で設定されている場合は、次の 2 つの環境変数をエクスポートし、グローバル環境を介して CA 証明書ファイルを設定する必要があります。

```
export OS_CACERT=<path_to_ca_cert_file>
export REQUESTS_CA_BUNDLE=<path_to_ca_cert_file>
```

（注） 以前のアプローチでは、VNF ライフサイクル管理ではなく、ESC インストール用の CA 証明書を指定していました。

VNF ライフサイクル管理に関する CA 証明書を渡す場合は、ESC の bootvm.py コマンドで次の引数を指定します。

```
--cert_file <path_to_ca_cert_file>
```

bootvm.py スクリプトがエラーで失敗する

問題に関する説明：

bootvm.py スクリプトを実行して ESC VM を作成しているときに、次のエラーが発生し、bootvm.py スクリプトが異常終了する場合があります。

```
bootvm script fails with error "object of type 'NoneType' has no len()"
```

説明：

詳細が指定されていないか、部分的に指定されているため、bootvm.py スクリプトは OpenStack ログイン情報と接続の詳細の認証に失敗します。

ソリューション：

最新の値をソースとする OpenRC ファイルがあることを確認してください。

次に例を示します。

```
export OS_USERNAME=admin
export OS_PASSWORD=<HIDDEN>
export OS_REGION_NAME=RegionOne
export OS_AUTH_URL=http://172.29.91.77:5000/v3
export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_IDENTITY_API_VERSION=3
```

bootvm.py を再度実行する前に、OpenRC ファイルを入手します。

`os_<variable_name>`、つまり `--os_auth_url=http://172.29.91.77:5000/v3` を使用して、bootvm.py に直接値を渡すこともできます。

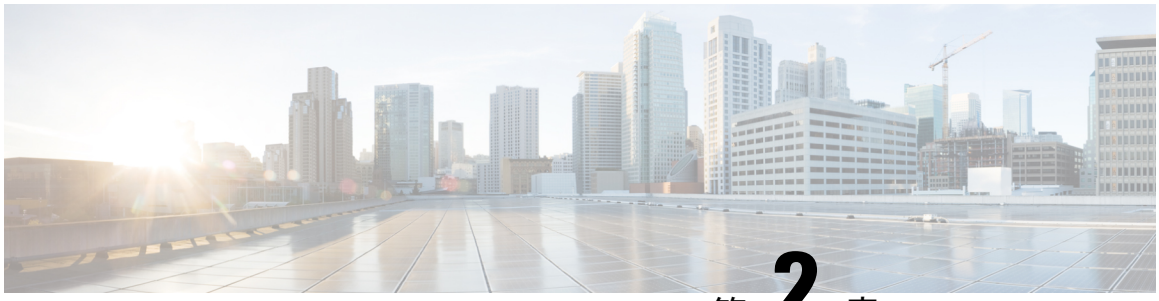
同じエラーが引き続き発生する場合は、デバッグオプションを指定して bootvm.py スクリプトを実行し、出力をファイルにリダイレクトします。このアクションを実行するには、次のコマンドライン引数を追加します。

```
--loglevel DEBUG --log /tmp/esc-install.log
```

テクニカルサポートに連絡する際は、結果の esc-install.log ファイルを添付してください。

第 II 部

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



第 2 章

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

ESC HA 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が、通常は長くは続きません。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-dcbcdbbd8685 |    | 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-dcbcdbbd8685
+-----------------------+-------------------------------------------------------------------------------------+
| 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-dcbcdbbd8685
                      |
| 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-dcbcdbbd8685 --allowed-address-pairs
type=dict list=true ip_address=152.16.3.78
Updated port: 80d7e031-04cd-4fb7-8f48-dcbcdbbd8685

$ neutron port-show 80d7e031-04cd-4fb7-8f48-dcbcdbbd8685
+---------------------+-------------------------------------------------------------------------------------+
| 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-dcbcdbbd8685
                      |
| 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
```

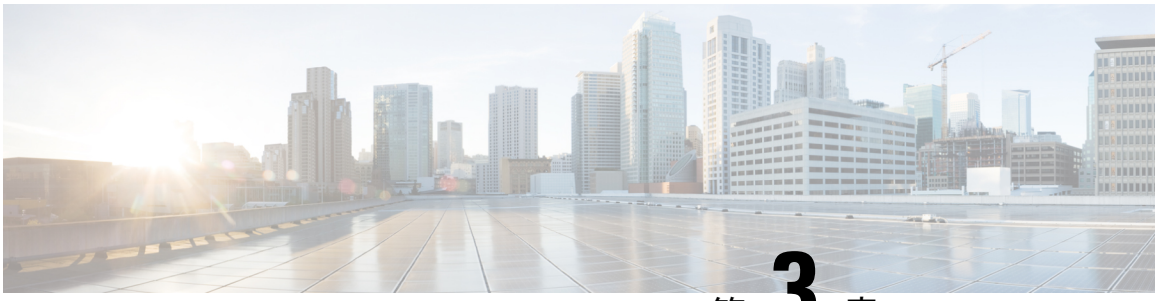
ro に Primary/Secondary と表示され、ds に UpToDate/UpToDate を表示されていることを確認します。これは、バックアップがアクティブなVMに接続されていて、アクティブとバックアップ間の同期が良好であることを意味しています。次の例は、バックアップVMが切断された日時を示しています。

```
$ 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
```

第 III 部

Cisco Elastic Services Controller マイクロサービスに関するトラブルシューティング



第 3 章

Cisco Elastic Services Controller マイクロサービスに関するトラブルシューティング

マイクロサービス ソフトウェア アーキテクチャに基づいて設計されている ESC は、マイクロサービス ソフトウェア アーキテクチャに統合された多くのコンポーネントやアプリケーションを備えており、ベンダーに依存しない、カスタマイズ可能でプログラム可能なプラットフォームを提供します。

Cisco Elastic Services Controller マイクロサービスの概要

ESC は、ベンダーにさまざまなサービスを提供するマイクロサービス ソフトウェア アーキテクチャに基づいています。すべてのマイクロサービスが正常な状態で実行されていることを確認します。ESC のマイクロサービスの正常性状態とサービス全体のステータスを確認する方法は次のとおりです。

ESC の全体的なステータスを確認するには、ESC VM または ESC HA のアクティブな VM で次のコマンドを実行します。

出力の 1 行目には ESC サービスの全体的なステータスが示され、後続の行には各マイクロサービスまたはコンポーネントのステータスが示されます。

```
# escadm status --v
0 ESC status=0 ESC HA Master Healthy
vimmanager (pgid 6432) is running
monitor (pgid 6450) is running
mona (pgid 6529) is running
drbd (pgid 0) is master
confd (pgid 6656) is running
keepalived (pgid 1374) is running
pgsql (pgid 6760) is running
filesystem (pgid 0) is running
snmp (pgid 6598) is running
escmanager (pgid 6924) is running
```

Cisco Elastic Services Controller ステータスが正常ではない

問題：

一部のシナリオでは、ESC ステータスを確認すると、ESC 正常性ステータスの出力に、ESC のステータスが重大なエラーステータスであり、ESC のコンポーネントまたはマイクロサービスの 1 つまたは複数が停止/デッド状態または未実行状態になっていることが表示されます。次に例を示します。

```
# escadm status --v
2 ESC status=0 ESC HA Master Critical
vimmanager (pgid 6432) is running
monitor (pgid 6450) is running
mona is stopped
drbd (pgid 0) is master
confd (pgid 6656) is running
keepalived (pgid 1374) is running
pgsql (pgid 6760) is running
filesystem (pgid 0) is running
snmp (pgid 6598) is running
escmanager (pgid 6924) is running
```

ソリューション：

次のアクションを実行して、ESC サービスを回復します。

- ESC サービスの再起動

  以下のコマンドを使用して、スタンドアロン ESC で ESC サービスを再起動します。

  ESC 2.X の場合：

  ```
  $ sudo service esc_service stop
  $ sudo service esc_service status (make sure ESC service is stopped)
  $ sudo service esc_service start
  ```

  ESC 3.X 以降のリリースの場合：

  ```
  $ sudo escadm stop
  $ sudo escadm status (make sure ESC service is stopped)
  $ sudo escadm start
  ```

  ESC HA の場合、以下のコマンドを実行して ESC サービスを再起動します。

  ESC 2.X の場合：

  ```
  $ sudo service keepalived stop
  $ sudo service keepalived status (make sure ESC service is stopped)
  $ sudo service keepalived start
  ```

  ESC 3.X 以降のリリースの場合：

  ```
  $ sudo escadm stop
  $ sudo escadm status (make sure ESC service is stopped)
  $ sudo escadm start
  ```

  ESC サービスを再起動すると、ESC HA フェールオーバーがトリガーされることに注意してください。前述のコマンドを実行後、バックアップ VM が HA マスターノードとして実

行状態に切り替わります。スイッチオーバーをトリガーしない場合は、以下で説明する 2 つの追加手順を実行する必要があります。

（注） ESC サービスを再起動すると、ESC HA フェールオーバーがトリガーされます。前述のコマンドを実行すると、バックアップ VM が切り替わり、HA のアクティブな VM として実行が開始されます。

スイッチオーバーをトリガーしない場合は、次の手順を実行します。

アクティブな VM でサービス再起動コマンドを実行する前に、まずバックアップ VM にログインして、以下のコマンドを実行します。

ESC 2.X の場合：

```
$ sudo service keepalived stop
$ sudo service keepalived status (make sure ESC service is stopped)
```

ESC 3.X 以降のリリースの場合：

```
$ sudo escadm stop
$ sudo escadm status (make sure ESC service is stopped)
```

アクティブな VM で ESC サービスを再起動したら、バックアップ VM に再度ログインして、以下のコマンドを実行します。

ESC 2.X の場合：

```
$ sudo service keepalived start
```

ESC 3.X 以降のリリースの場合：

```
$ sudo escadm start
```

- ESC VM の再起動

  ESC サービスを再起動しても解決しない場合は、以下のコマンドを実行して ESC VM を再起動します。

  ```
  $ sudo reboot
  ```

（注） ESC のアクティブな VM を再起動すると、ESC HA スイッチオーバーがトリガーされます。バックアップ VM が新しいアクティブな VM になり、すべてのサービスが開始されます。

- ESC の起動ログを確認します。

ESC サービスが起動時に引き続きスタックする場合は、ESC ログを確認して詳細を調べます。以下のログファイルを確認する必要があります。

- `/var/log/esc/escadm.log`：問題を引き起こしているマイクロサービスを確認するための ESC サービスの開始/停止ログ。

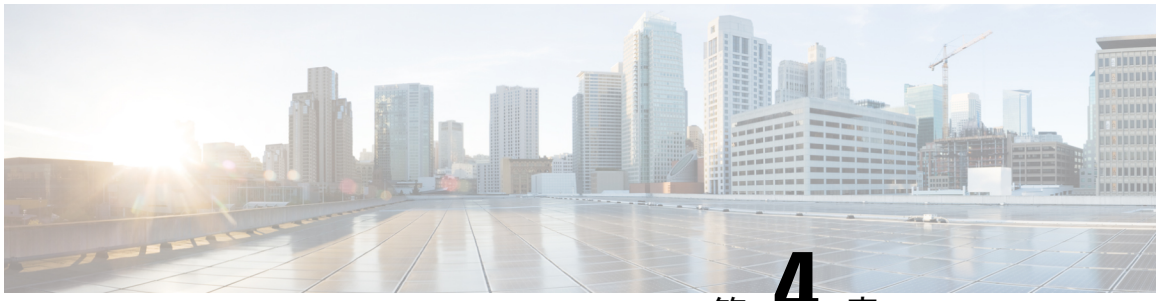
- `/var/log/esc/escmanager.log`：ESCManager サービスの起動/停止に関する ESCManager のログ。
- `/var/log/messages`：OS メッセージログファイルには、システムレベルでの致命的な起動エラーも含まれています。

問題が見つからない場合は、ESC ログを収集し、ESC VM（HA 内の 2 つの VM）のログファイルをテクニカルサポートチームに送信します。ログを収集するには、次のコマンドを使用します。

```
$ sudo escadm log collect
```

第 IV 部

Cisco Elastic Services Controller のアップグレードに関するトラブルシューティング



第 4 章

Cisco Elastic Services Controller のアップグレードに関するトラブルシューティング

Cisco Elastic Services Controller のアップグレードに関するトラブルシューティング

問題に関する説明：

ESC アップグレードアプローチの選択の仕方がわかりません。

説明：

ESC アップグレードには 3 つのアップグレードアプローチがあります。以下は、ESC セットアップをアップグレードするための最良のアプローチを選択するための参照基準です。

ソリューション：

- アップグレードパスが ESC パッチビルド間（たとえば、ESC 3.1.0.116 から ESC 3.1.0.145 へのアップグレード）の場合は、RPM アップグレードを最初の選択肢として検討してください。
- アップグレードパスが ESC 公式 FCS リリース間（たとえば、ESC 3.0.0 から ESC 3.1.0）の場合、RPM アップグレードはその状況では機能せず、イメージアップグレードのみを実行できます。
  - 正常な ESC HA の場合は、インサービスアップグレードを検討してください。
  - マスター VM のみを使用するスタンドアロン ESC または ESC HA の場合、データベースのバックアップまたは復元アプローチを採用します。

Cisco Elastic Services Controller のアップグレードのロールバック

問題に関する説明：

アップグレードがうまくいかず、ロールバックしたい。

説明：

ESC アップグレードには 3 つのアップグレードアプローチがあります。以下は、ESC セットアップをアップグレードするための最良のアプローチを選択するための参照基準です。

ソリューション：

ESC のインストールとアップグレードに関する 2 つの重要なポイントは次のとおりです。

- ESC または ESC HA を正常にインストールしたら、完全な bootvm.py コマンドラインを記録に残します。その bootvm.py コマンドラインは、次にイメージアップグレードアプローチを使用して ESC アップグレードを実行するときに使用できます。
- ESC のアップグレード中に、ESC データベースのバックアップを取得してください。データベースをバックアップする前に、ESC サービスを停止し、サービスが停止状態になるまで待ちます。

  例：

```
#ESC backup DB ESC 2.X:
$ sudo /opt/cisco/esc/esc-scripts/esc_dbtool.py backup --file
scp://<dest_user>:<dest_password>@<dest_host>:/tmp/db.tar.bz2

#ESC backup DB ESC 3.X and up:
sudo escadm backup --file /tmp/db.tar.bz
scp /tmp/db.tar.bz <dest_user>@<dest_host>:/tmp/db.tar.bz
```

- データベースのバックアップを取得すると、ESC のアップグレード手順中または後に問題が発生した場合に、ESC のアップグレードを簡単にロールバックできます。

データベース バックアップ ファイルに bootvm.py コマンドが記録されている場合は、次の方法で ESC アップグレードをロールバックできます。

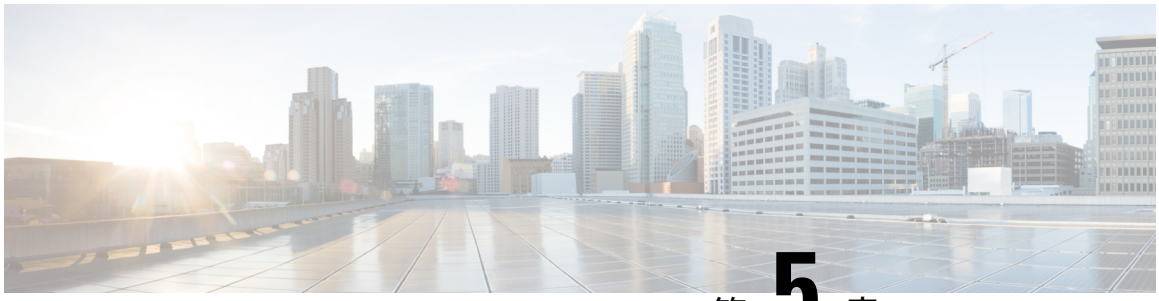
- インサービスアップグレードを実行していて、古いプライマリ ESC VM を削除していない（または古いプライマリ ESC VM の名前のみ変更している）場合は、まずアップグレードされた VM インスタンスを削除してから、古い（名前が変更された）プライマリ ESC を有効にできます。この場合、データベースは古いプライマリ VM に変更を加えることなく保持されます。古いリリースイメージがあるレコードから完全な bootvm.py コマンドラインを使用して、バックアップ ESC VM を再インストールします。再展開されたバックアップ VM のデータベースは、プライマリ VM から自動的に同期されます。データベースを復元する必要はありません。
- 古いバージョンのプライマリ ESC VM とバックアップ ESC VM の両方を削除した場合、またはスタンドアロン ESC の DB バックアップ/復元アップグレードを実行している場合

は、古いリリースイメージがあるレコードから完全な bootvm.py コマンドラインを使用して、ESC VM を再展開します。次に、データベースの復元ガイドに従って、ロールバックのためにデータベースを復元します。DB を復元する前に、ESC サービスを停止し、サービスが停止状態になるまで待ちます。

例：

```
#ESC Restore DB ESC 2.X:
$ sudo /opt/cisco/esc/esc-scripts/esc_dbtool.py restore --file
scp://<dest_user>:<dest_password>@<dest_host>:/tmp/db.tar.bz2

#ESC Restore DB ESC 3.X and up:
scp <src_user>@<src_host>:/tmp/db.tar.bz /tmp/db.tar.bz
sudo escadm restore --file /tmp/db.tar.bz
```



第 5 章

Cisco Elastic Services Controller のバックアップと復元に関するトラブルシューティング

破損したデータベース バックアップ ファイル

問題に関する説明：

破損したデータベース バックアップ ファイルがあります。

ソリューション：

データベース バックアップ コマンドを実行すると、すべてのデータベース バックアップ ファイルを含むtarファイルが作成されます。DBバックアップファイルが破損していないことを確認するには、次のコマンドを実行して検証し、echoコマンドの出力が0であることを確認します。

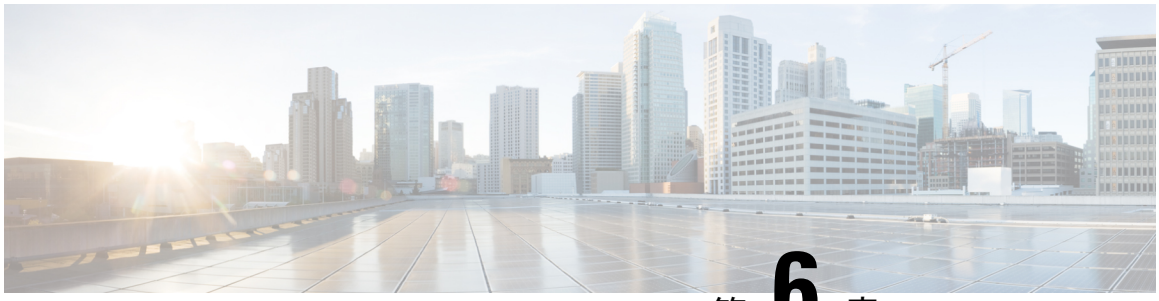
```
$tar -tvf <your_backup_tar_file>
$echo $?
0
```

バックアップ tar ファイルを他の VM にコピーするには、チェックサムアプローチを使用してデータが統合されていることを確認します。次に例を示します。

```
$ sha256sum /tmp/db.tar.bz
d4a831983d0cafddf1c734eed5ad7f39904f948f86ffd64c675107d94ca15a4f  /tmp/db.tar.bz
```

第 V 部

ConfD および NETCONF API に関するトラブルシューティング



第 6 章

ConfD および NETCONF API に関するトラブルシューティング

ConfD および NETCONF API に関するトラブルシューティング

セキュリティ上の理由から、ESC のデフォルトでは ConfD の開発者ログと NETCONF トレースログが無効化されます。

デバッグには次の 2 つのログを使用します。

2 つのログを手動で有効にするには、ESC VM で次の ConfD の構成ファイルを開きます。

```
$ sudo vim /opt/cisco/esc/esc_database/esc_production_confd.conf
```

次のログで以下のように手動で変更します。

enabled を true に変更。

```
<netconfTraceLog>
    <enabled>true</enabled>
    <filename>/var/log/esc/confd/netconf.trace</filename>
    <format>pretty</format>
</netconfTraceLog>
```

このセクションを変更して、開発者ログを有効にします。

enabled と file.enabled の両方を true に変更。

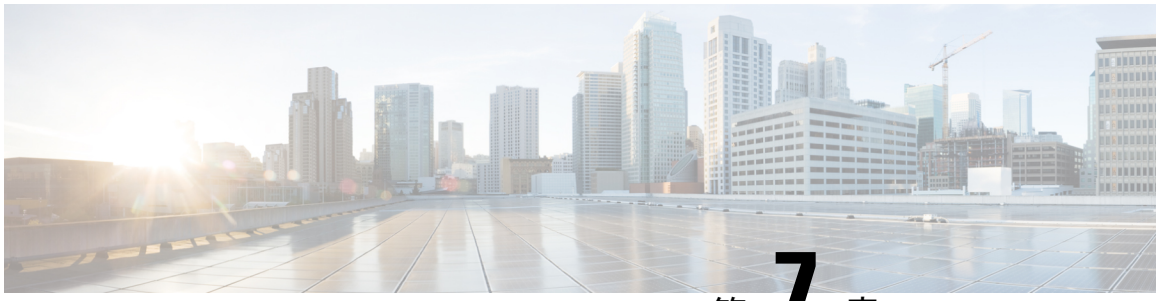
```
<developerLog>
    <enabled>true</enabled>
    <file>
        <enabled>true</enabled>
        <name>/var/log/esc/confd/devel.log</name>
    </file>
    <syslog>
        <enabled>false</enabled>
```

ConfD および NETCONF API に関するトラブルシューティング

```
    </syslog>
</developerLog>
```

第 VI 部

VNF展開に関するトラブルシューティング



第 7 章

VNF展開に関するトラブルシューティング

概要

このガイドでは、ESCによって展開および管理されているVNF間の問題をトラブルシューティングする方法について、順を追って説明します。

トラブルシューティング用のログ

VNF 展開の ESC トラブルシューティングを実行する前の最初の手順は、ログの収集および確認です。ESC ログを収集するには、次の手順を実行します。

ESC リリース 2.3.2 の場合：

```
sudo /opt/cisco/esc/esc-scripts/collect_esc_log.sh
```

ESC リリース 3.0 以降の場合：

```
sudo escadm log collect
```

有用なエラーメッセージを含む重要なログがいくつかあります。

- YangESC ログには、着信要求と通知が含まれています。

  ```
  /var/log/esc/yangesc.log
  ```

- ESCManager ログには、ESC 処理の詳細が含まれています。

  ```
  /var/log/esc/escmanager.log
  ```

- VimManager ログには、VimManager 処理の詳細が含まれています。

  ```
  /var/log/esc/vimmanager/vimmanager.log
  ```

- Vim_VimManager ログには、VimManager と VIM 間の未処理の要求と応答が含まれています。

  ```
  /var/log/esc/vimmanager/vim_vimmanager.log
  ```

- Mona ログには、処理の詳細とスクリプト実行のモニタリング情報が含まれています。

  ```
  /var/log/esc/mona/mona.log
  ```

展開された VNF VM が VIM 関連エラーで失敗する

問題

ステータスコードが 200 以外の VM_DEPLOYED 通知を（NETCONF、REST、ポータルを介して）受け取った場合、エラーが原因で展開が失敗したことを意味します。

/var/log/esc/yangesc.log に通知が見つかりました。

```
02:19:25,758 23-Jan-2018 WARN  ===== SEND NOTIFICATION STARTS =====
02:19:25,758 23-Jan-2018 WARN  Type: VM_DEPLOYED
02:19:25,758 23-Jan-2018 WARN  Status: FAILURE
02:19:25,758 23-Jan-2018 WARN  Status Code: 500
02:19:25,759 23-Jan-2018 WARN  Status Msg: VIM Driver: Exception while creating VM: Error
 creating VM from template, the host [10.67.103.255] does not exist
02:19:25,759 23-Jan-2018 WARN  Tenant: admin
02:19:25,759 23-Jan-2018 WARN  Deployment ID: 169384d7-c67b-40a4-bcaa-dd3294305ba3
02:19:25,759 23-Jan-2018 WARN  Deployment name:
Vmware-NetConf-Intra-Affinity-Anti-Affinity-With-InvalidCluster-InvalidHost
02:19:25,759 23-Jan-2018 WARN  VM group name:
Group2-uLinux-Intra-Anti-Affinity-With-InvalidHost
02:19:25,759 23-Jan-2018 WARN  User configs: 1
02:19:25,759 23-Jan-2018 WARN      VM Name:
Vmware-NetConf-I_Group2_0_65aa6ca8-3b53-4eb3-a39f-a3f12394a190
02:19:25,759 23-Jan-2018 WARN      Host ID:
02:19:25,759 23-Jan-2018 WARN      Host Name:
02:19:25,760 23-Jan-2018 WARN  =====  SEND NOTIFICATION ENDS  =====
```

解決方法

失敗の理由は、ステータスメッセージ自体に記載されています。前述の例は、展開対象のホストが存在しないことを示しています。次に、失敗した VM 展開の別の例を示します。

/var/log/esc/yangesc.log に通知が見つかりました。

```
07:20:56,164 25-Jan-2018 WARN  Status Msg: Failed to create VM ports for instance :
jenkins-ErrHandl_ErrorG_2_cc0f8c28-8900-4977-90d9-b9f996c8ca71. Create port operation
failed: Exception during processing:
com.cisco.esc.vimmanager.exceptions.CreatePortException: Create port failed:
ClientResponseException{message=No more IP addresses available on network
0b7965b4-c604-444c-8cbb-7c2399e912d4., status=409, status-code=CONFLICT}.
```

前述の例は、VIMからの直接応答（エラーメッセージとステータスコードあり）を含む展開失敗メッセージを示しています。このような VIM 関連の問題など、ロールアクセスの問題が原因で展開が失敗した場合は、VIMインスタンスで必要なアクションを実行するか、適切な構成で ESC 展開データモデルを調整します。一般的な VIM 関連の問題を次に示します。

1. クォータ不足エラー
    1. 該当テナント/プロジェクト/ユーザーの下で ESC を介して問題のリソースを削除します。または、
    2. テナント/プロジェクト/ユーザーごとに VIM のリソース制限を設定します。
2. 使用中エラー
    1. リソース名または設定を変更します。あるいは、VIM で許可されない制限があり、設定可能な場合があります。
    2. 問題のリソースを削除します。

展開された VNF VM が LCM で失敗する

問題

VNF 展開データモデルに LCM アクション（ステージングスクリプトなど）が含まれる場合、アクションが完了しなかったために展開が失敗した可能性があります。この場合、/var/log/esc/escmanager.log に次のエラーメッセージが表示されます。

```
22:12:11,912 25-Jan-2018
VM_STATE_MACHINE-ab-auto-test-vnf_ab-aut_0_31ebad33-e12f-4772-a89c-3bdc239acf69 ERROR
[StateMachineCloudUtils.java:setupPersonalities():1081]
[tid=fffae7af-a321-4ea5-a1bc-3b30c903f3a5]
com.cisco.esc.datamodel.exceptions.ESCException: Action [GEN_VPC_ISO] failed
    at
com.cisco.esc.statemachines.utils.StateMachineCloudUtils.setupPersonalities(StateMachineCloudUtils.java:1069)
```

説明

ステージングスクリプトの失敗の詳細を確認するには、/var/log/esc/mona/mona.log ログで次のようなエントリを探します。

/var/log/esc/mona/mona.log

```
2018-01-25 19:34:45.751 [http-nio-127.0.0.1-8090-exec-5] Script:
[/opt/cisco/esc/esc-scripts/esc_volume_em_staging.sh] execution in progress
2018-01-25 19:34:45.751 [http-nio-127.0.0.1-8090-exec-5] Use the original script path
and skip downloading: no protocol: /opt/cisco/esc/esc-scripts/esc_volume_em_staging.sh
2018-01-25 19:49:45.772 [http-nio-127.0.0.1-8090-exec-5] Script execution failed, timer
 expired for script: /opt/cisco/esc/esc-scripts/esc_volume_em_staging.sh
```

```
2018-01-25 19:49:45.805 [http-nio-127.0.0.1-8090-exec-5] Script execution failed
com.cisco.esc.mona.exceptions.ActionExecutionException: Script execution failed, timer
expired for script:/opt/cisco/esc/esc-scripts/esc_volume_em_staging.sh
```

解決方法

一般的なエラーには、権限の問題や、スクリプト実行のタイムアウトなどがあります。ESC VM でスクリプトのリハーサルを実行して、スクリプトが機能することを確認します。

ロールアクセスの問題により、展開された VNF VM が失敗する（ESC リリース 3.1 以降）

問題

管理者以外のユーザーとして OpenStack に VNF を展開すると、展開時に次のようなロールアクセスエラーが発生する場合があります。

```
02:19:25,758 23-Jan-2018 WARN  ===== SEND NOTIFICATION STARTS =====
02:19:25,758 23-Jan-2018 WARN  Type: VM_DEPLOYED
02:19:25,758 23-Jan-2018 WARN  Status: FAILURE
02:19:25,758 23-Jan-2018 WARN  Status Code: 500
02:19:25,759 23-Jan-2018 WARN  Status Msg: VIM Driver: Exception while creating VM:
{"message": "You are not authorized to perform the requested action:
identity:create_project", "code": 403, "title": "Forbidden"}}
02:19:25,759 23-Jan-2018 WARN  Tenant: admin
02:19:25,759 23-Jan-2018 WARN  Deployment ID: 169384d7-c67b-40a4-bcaa-dd3294305ba3
02:19:25,759 23-Jan-2018 WARN  Deployment name:
Vmware-NetConf-Intra-Affinity-Anti-Affinity-With-InvalidCluster-InvalidHost
02:19:25,759 23-Jan-2018 WARN  VM group name:
Group2-uLinux-Intra-Anti-Affinity-With-InvalidHost
02:19:25,759 23-Jan-2018 WARN  User configs: 1
02:19:25,759 23-Jan-2018 WARN      VM Name:
Vmware-NetConf-I_Group2_0_65aa6ca8-3b53-4eb3-a39f-a3f12394a190
02:19:25,759 23-Jan-2018 WARN      Host ID:
02:19:25,759 23-Jan-2018 WARN      Host Name:
02:19:25,760 23-Jan-2018 WARN  =====  SEND NOTIFICATION ENDS  =====
```

解決方法

ESC リリース 3.1 以降では、Neutron で 2 つの権限を付与する必要があります。

```
create_port:fixed_ips
create_port:mac_address
```

1. ESC の OpenStack に新しいロールを作成します。OpenStack Horizon（[アイデンティティ（Identity）]->[ロール（Roles）]）に移動し、**vnfm** または他の任意の名前を指定して新しいロールを作成します。
2. OpenStack Horizon の ESC によって管理されるプロジェクトに vnfm ロールを持つユーザーを割り当てます（[アイデンティティ（Identity）] → [プロジェクト（Projects）]）。[メンバーの管理（Manage members）] をクリックし、ESC のユーザーが vnfm ロールを持っていることを確認します。

3. デフォルト値に「or role:vnfm」を追加して、OpenStack コントローラの以下の項目を変更します。policy.json ファイルへの変更はすぐに有効になり、サービスを再起動する必要はありません。

/etc/neutron/policy.json	"create_port:fixed_ips": "rule:context_is_advsvc または rule:admin_or_network_owner",	"create_port:fixed_ips": "rule:context_is_advsvc または rule:admin_or_network_owner または role:vnfm",
/etc/neutron/policy.json	"create_port:mac_address": "rule:context_is_advsvc または rule:admin_or_network_owner"	"create_port:mac_address": "rule:context_is_advsvc または rule:admin_or_network_owner または role:vnfm"

VNF VM は展開されたが、ブートループに入る

問題

VNF が展開され、ステータスコードが 200 の VM_DEPLOYED 通知を受信し、VM_ALIVE はまだ受信していないとします。VIM UI（OpenStack Horizon、VMware vCenter など）を介して VNF のコンソールをチェックしているときに、VNF は再起動サイクルまたはループに入ります。ほとんどの場合は、障害のデイゼロデータが渡されたことを示しています。渡されたデイゼロデータを確認するには、OpenStack の場合、/var/log/esc/vimmanger/vim_vimmanager.log をチェックして、サーバーを作成するために OpenStack に送信された POST リクエストを探します。

```
2018-01-26 16:02:55.648 INFO  os -   1 * Sending client request on thread
http-nio-127.0.0.1-8095-exec-4
1 > POST http://ocata1-external-controller:8774/v2/d6aee06abdbe42edaade348280199a64/servers
1 > Accept: application/json
1 > Content-Type: application/json
1 > User-Agent: OpenStack4j / OpenStack Client
1 > X-Auth-Token: ***masked***
{
  "server" : {
    "name" : "jenkins-jenkinsy_MAKULA_0_bbc61ba6-6c63-4fb9-b9cd-5ae92a898943",
    "imageRef" : "67fc9890-230e-406c-bd01-f2e1ffa2437f",
    "flavorRef" : "cc12dec2-411a-46bd-b8c2-4ff8738ddb02",
    "personality" : [ {
      "path" : "iosxe_config.txt",
      "contents" :
[redacted]

    } ],
    "config_drive" : true,
    "networks" : [ {
      "port" : "01b3b168-8fab-4da4-b195-f9652d36674e"
    }, {
      "port" : "dfe709c9-4ca4-400f-a765-2dbc88828585"
    } ],
    "block_device_mapping_v2" : [ {
      "source_type" : "image",
      "destination_type" : "local",
      "uuid" : "67fc9890-230e-406c-bd01-f2e1ffa2437f",
```

```
      "boot_index" : 0,
      "delete_on_termination" : true
    } ]
  }
}
```

解決方法

base64 で符号化されたパーソナリティコンテンツの文字列値を復号化します。

```
hostname csr
!
platform console serial
!
ip subnet-zero
no ip domain-lookup
ip domain name cisco.com
!
enable password cisco123
username admin password cisco123
username admin privilege 15
!
interface GigabitEthernet1
 description management network
 ip address dhcp
 no shut
!
interface GigabitEthernet2
 description service network
 ip address dhcp
 no shut
!
interface GigabitEthernet3
 description service network
 ip address dhcp
 no shut
!
crypto key generate rsa modulus 1024
ip ssh version 2
ip ssh authentication-retries 5
ip scp server enable
file prompt quiet
!
line con 0
 stopbits 1
line vty 0 4
 login local
 privilege level 15
 transport input ssh telnet
 transport output ssh telnet
!
snmp-server community public RO
!
end
```

コンテンツに正しいデイゼロ設定が含まれているか確認します。デイゼロ設定がボリュームを介して渡される場合は、そのボリュームを VNF から切り離し、別の VM に接続して内容を確認します。

VMware の場合、デイゼロ設定が OVF 設定を介して渡される場合は、vCenter から確認します。

1. [VM設定（VM settings）] を開きます。

2. [オプション（Options）] で、[OVF設定（OVF Settings）] を選択します。
3. [OVF環境（OVF Enviroment）] の下にある [表示（View）] をクリックします。

デイゼロ設定が CDROM を介して渡される場合、CDROM にマウントされた ISO ファイルから確認できます。

特定のデータストアから ISO ファイルをダウンロードし、ISO ファイルをローカルにマウントしてその内容を確認します。

VNF VM は展開されたが、稼働状態にならない

問題

VNF が正常に展開されたが、VM_ALIVE 通知を受信しなかった場合、ESC は新しく展開された VNF に到達できなかったと推測されます。主な原因はネットワークの問題です。VNF 展開データモデルの KPI セクションを確認します。

```
<kpi_data>
  <kpi>
    <event_name>VM_ALIVE</event_name>
    <metric_value>1</metric_value>
    <metric_cond>GT</metric_cond>
    <metric_type>UINT32</metric_type>
    <metric_occurrences_true>1</metric_occurrences_true>
    <metric_occurrences_false>30</metric_occurrences_false>
    <metric_collector>
      <type>ICMPPing</type>
      <nicid>2</nicid>
      <poll_frequency>10</poll_frequency>
      <polling_unit>seconds</polling_unit>
      <continuous_alarm>false</continuous_alarm>
    </metric_collector>
  </kpi>
</kpi_data>
```

解決方法

VNF が稼働しているか確認するには、次に示されている特定の IP を使用して、ESC VM から VNF に ICMP ping を実行します。

```
<nicid>2</nicid>
```

ここで、nicid 2 は、ESC が ping しようとしている nicid が 2 のインターフェイスの IP を指しており、以下の内容を指しています。

```
<interface>
  <nicid>2</nicid>
  <network>NVPGW100-UAS-uas-orchestration</network>
  <allowed_address_pairs>
    <address>
      <ip_address>172.168.11.0</ip_address>
      <netmask>255.255.255.0</netmask>
    </address>
  </allowed_address_pairs>
</interface>
```

ここで 172.168.11.0 は IP です。インターフェイスが ESC と同じネットワークを共有していることを確認してください。前述の例では、ネットワークはNVPGW100-UAS-uas-orchestrationです。ping が失敗した場合、ゲートウェイまたはサブネットで使用可能な別の IP に ping を実行して、問題がネットワークにあるのか確認できます。

VNF リカバリの失敗

一般的なリカバリの問題を次に示します。

1. リカバリ動作が期待どおりに動作しない。再起動に失敗後、ESCが再展開を試みていません。
   1. XML ファイルのリカバリポリシーがREBOOT_THEN_REDEPLOYに設定されていて、「再起動のみ」に設定されていないことを確認します。リカバリマニュアルを読み、リカバリオプションと期待される動作を理解します。

2. ESC がリカバリを 1 回のみ試行しているか、何度も試行しています。
   1. 設定パラメータ「VM_RECOVERY_RETRIES_MAX」を再確認します。デフォルト値は 3 回です。この値を確認するには、ESC VM 内で REST 呼び出しを実行します。

      ```
      curl -H "accept: Application/json"
      http://127.0.0.1:8080/ESCManager/v0/config/default/VM_RECOVERY_RETRIES_MAX |
      python -mjson.tool
      ```

   2. 正しく設定されている場合は、ESC がリカバリ時に正常であり、スイッチオーバーが発生していないことを確認します。また、2 番目の ESC VM でリカバリの試行を継続している可能性があります。

再起動の失敗により VNF VM リカバリが失敗する

問題

VM の再起動に失敗したため、VNF VM のリカバリに失敗しました。失敗するかどうかは、VM リカバリポリシーの定義によって異なります。

```
<recovery_policy>
  <recovery_type>AUTO</recovery_type>
  <action_on_recovery>REBOOT_ONLY</action_on_recovery>
  <max_retries>3</max_retries>
</recovery_policy>
```

説明

ESCは、エラー状態のRECOVERY_COMPLETEDイベントを受信する前に、未成功状態でVNF VM の再起動を 3 回試行します。再起動操作は、他の 2 つのシステム全体の設定パラメータにも依存します。

```
VM_STATUS_POLLING_VM_OPERATION_RETRIES
VM_STATUS_POLLING_WAIT_TIME_SEC
```

ESC は、VM の再起動を VIM に要求後、VM ステータスのポーリングを継続します。VM_STATUS_POLLING_VM_OPERATION_RETRIES では、ESC のポーリング試行回数を定義し、VM_STATUS_POLLING_WAIT_TIME_SEC では、ポーリング間での ESC の待機時間を定義します。以下はそれぞれのデフォルト値です。

```
VM_STATUS_POLLING_VM_OPERATION_RETRIES=10
VM_STATUS_POLLING_WAIT_TIME_SEC=5
```

解決方法

OpenStack で VNF VM が再起動状態からアクティブ状態に移行するのに 50 秒以上かかる場合は、ESC REST API を使用して VM_STATUS_POLLING_WAIT_TIME_SEC をより大きな数値に変更します。

```
curl -X PUT -H "accept:application/json"
http://localhost:8080/ESCManager/v0/config/openstack/vm_status_polling_wait_time_sec/20
 -k | python -mjson.tool
```

成功の応答を受信したら、VM の手動リカバリを再度実行します。

エラー状態の VNF VM の回復

VNF VM が ESC でエラー状態になっている場合、エラー状態の原因となった外部の問題（VIM の問題など）が解決した場合に稼働状態に戻すための 2 つのオプションがあります。次の 2 つのオプションのいずれかを実行する前に、同じ展開環境で実行中の操作がないことを確認してください。操作は /var/log/esc/yangesc.log で確認します。 完了通知（成功または失敗）のない、以前に開始されたアクションを探します。進行中の操作が見つかった場合は、その操作が完了するのを待ってから、次のアクションを実行します。

VNF VM の手動リカバリ

makecall ディレクトリで、次のコマンドを実行します。

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO {ESC generated VM Name}
```

VM モニタリングの手動設定解除または設定（ESC リリース 3.1 以降）

次のコマンドを実行して、モニタリングの設定を解除します。

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli vm-action DISABLE_MONITOR {ESC generated VM
 Name}
```

次に、再度有効にします。

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli vm-action ENABLE_MONITOR {ESC generated VM
Name}
```

サービス更新による VNF の削除または追加

スクリプトを使用したデータモデルの準備

ESC VM でスクリプトを実行してください。スクリプトにより 2 つのデータモデル xml ファイルが生成されます。1 つは VM グループを削除するためのファイルで、もう 1 つは VM グループを追加し直すためのファイルです。

```
[admin@abc-test-232 ~]$ ./genVMGroupDeletionDM.py -h
usage: genVMGroupDeletionDM.py [-h] vm_group_name [vm_group_name ...]

************************************************************
Utility tool for generating VM group removing datamodel for ESC
Check the following wiki for details
https://confluence-eng-sjc1.cisco.com/conf/display/ESCWIKI/How+to+Use+Service+Update+to+Remove+a+VM+Group


positional arguments:
  vm_group_name  <Required> VM group name(s) separate by space

optional arguments:
  -h, --help     show this help message and exit
```

例

```
[admin@abc-test-232 ~]$ ./genVMGroupDeletionDM.py g1 g2

Datamodel is generated:
[/home/admin/delete_g1_g2.xml]
[/home/admin/add_g1_g2.xml]
** Use on your own risk! **

[admin@abc-test-232 ~]$
```

手動でのデータモデルの準備

1. 現在の ESC データモデルを取得します。

   ```
   /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli get esc_datamodel/tenants > {file name}
   ```

2. ステップ 1 で取得した元のファイル（および <xml> タブの前にあるすべての CLI 出力）から余分なラッパー <data> および <rpc-reply> を削除します。最終結果は次のようになります。

   ステップ 2 後のデータモデルの例

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
     <data>
       <esc_datamodel xmlns="http://www.cisco.com/esc/esc">
   ```

（注） データモデルに複数の展開が含まれている場合は、データモデルの編集時に他の展開をそのまま維持してください。テキストの書式設定は変更しないでください。他の展開セクションをデータモデルから完全に削除します。削除すると、サービス更新の発生時に、削除された展開は変更されません。たとえば、削除する VM グループが c3 の場合、データモデル編集時に EM の展開部分をデータモデルから削除できます。

3. 次に、VM グループ c3 を削除する例を示します。まず、VM グループ c3 が含まれる <policies> の下に定義済みの <placement_group> または <placement> policy(ies) が存在するか確認します。ポリシーをマークして削除します。次の例には、8 つの配置ポリシーがあります。

   ```
   ...
                   <placement>
                     <target_vm_group_ref>c3</target_vm_group_ref>
                     <type>anti_affinity</type>
   ```

```
      <enforcement>strict</enforcement>
      <vm_group_ref>c1</vm_group_ref>
      <vm_group_ref>s2</vm_group_ref>
    </placement>
    <placement>
      <target_vm_group_ref>s10</target_vm_group_ref>
      <type>anti_affinity</type>
      <enforcement>strict</enforcement>
      <vm_group_ref>c1</vm_group_ref>
      <vm_group_ref>c3</vm_group_ref>
      <vm_group_ref>s2</vm_group_ref>
      <vm_group_ref>s4</vm_group_ref>
      <vm_group_ref>s5</vm_group_ref>
      <vm_group_ref>s6</vm_group_ref>
      <vm_group_ref>s7</vm_group_ref>
      <vm_group_ref>s8</vm_group_ref>
      <vm_group_ref>s9</vm_group_ref>
    </placement>
...
    <placement>
      <target_vm_group_ref>s4</target_vm_group_ref>
      <type>anti_affinity</type>
      <enforcement>strict</enforcement>
      <vm_group_ref>c1</vm_group_ref>
      <vm_group_ref>c3</vm_group_ref>
      <vm_group_ref>s2</vm_group_ref>
    </placement>
    <placement>
      <target_vm_group_ref>s5</target_vm_group_ref>
      <type>anti_affinity</type>
      <enforcement>strict</enforcement>
      <vm_group_ref>c1</vm_group_ref>
      <vm_group_ref nc:operation='delete'>c3</vm_group_ref>
      <vm_group_ref>s2</vm_group_ref>
      <vm_group_ref>s4</vm_group_ref>
    </placement>
    <placement>
      <target_vm_group_ref>s6</target_vm_group_ref>
      <type>anti_affinity</type>
      <enforcement>strict</enforcement>
      <vm_group_ref>c1</vm_group_ref>
      <vm_group_ref>c3</vm_group_ref>
      <vm_group_ref>s2</vm_group_ref>
      <vm_group_ref>s4</vm_group_ref>
      <vm_group_ref>s5</vm_group_ref>
    </placement>
    <placement>
      <target_vm_group_ref>s7</target_vm_group_ref>
      <type>anti_affinity</type>
      <enforcement>strict</enforcement>
      <vm_group_ref>c1</vm_group_ref>
      <vm_group_ref>c3</vm_group_ref>
      <vm_group_ref>s2</vm_group_ref>
      <vm_group_ref>s4</vm_group_ref>
      <vm_group_ref>s5</vm_group_ref>
      <vm_group_ref>s6</vm_group_ref>
    </placement>
    <placement>
      <target_vm_group_ref>s8</target_vm_group_ref>
      <type>anti_affinity</type>
      <enforcement>strict</enforcement>
      <vm_group_ref>c1</vm_group_ref>
      <vm_group_ref>c3</vm_group_ref>
      <vm_group_ref>s2</vm_group_ref>
```

```
    <vm_group_ref>s4</vm_group_ref>
    <vm_group_ref>s5</vm_group_ref>
    <vm_group_ref>s6</vm_group_ref>
    <vm_group_ref>s7</vm_group_ref>
  </placement>
  <placement>
    <target_vm_group_ref>s9</target_vm_group_ref>
    <type>anti_affinity</type>
    <enforcement>strict</enforcement>
    <vm_group_ref>c1</vm_group_ref>
    <vm_group_ref>c3</vm_group_ref>
    <vm_group_ref>s2</vm_group_ref>
    <vm_group_ref>s4</vm_group_ref>
    <vm_group_ref>s5</vm_group_ref>
    <vm_group_ref>s6</vm_group_ref>
    <vm_group_ref>s7</vm_group_ref>
    <vm_group_ref>s8</vm_group_ref>
  </placement>
```

target_vm_group が c3 の場合、XML 要素に属性 `nc:operation='delete'` を追加して、配置ポリシー全体を削除します。

.\

```
<placement nc:operation='delete'>
  <target_vm_group_ref>c3</target_vm_group_ref>
  <type>anti_affinity</type>
  <enforcement>strict</enforcement>
  <vm_group_ref>c1</vm_group_ref>
  <vm_group_ref>s2</vm_group_ref>
</placement>
```

vm_group_ref が c3 の場合、vm_group_ref エントリ自体を削除し、他の関係はそのままにします。

```
<placement>
  <target_vm_group_ref>s10</target_vm_group_ref>
  <type>anti_affinity</type>
  <enforcement>strict</enforcement>
  <vm_group_ref>c1</vm_group_ref>
  <vm_group_ref nc:operation='delete'>c3</vm_group_ref>
  <vm_group_ref>s2</vm_group_ref>
  <vm_group_ref>s4</vm_group_ref>
  <vm_group_ref>s5</vm_group_ref>
  <vm_group_ref>s6</vm_group_ref>
  <vm_group_ref>s7</vm_group_ref>
  <vm_group_ref>s8</vm_group_ref>
  <vm_group_ref>s9</vm_group_ref>
</placement>
```

vm_group_ref 要素が 1 つだけある配置ポリシーの場合、vm_group_ref が c3 であるか、target_vm_group が c3 である場合は、ポリシー全体を削除します。これは、c3 が削除されると、このポリシーに意味がなくなるためです。

```
<placement nc:operation='delete'>
  <target_vm_group_ref>c11</target_vm_group_ref>
  <type>anti_affinity</type>
  <enforcement>strict</enforcement>
  <vm_group_ref>c3</vm_group_ref>
</placement>
```

4. 最後のステップでは、属性 nc:operation='delete' を XML 要素に追加して、VM グループ自体を削除対象としてマークします。

```
<vm_group nc:operation='delete'>
                <name>c3</name>
                <flavor>SFPCF101-DEPLOYMENT-control-function</flavor>
                <bootup_time>1800</bootup_time>
                <recovery_wait_time>1</recovery_wait_time>
...
```

5. 同じ VM グループを再び追加するためにデータモデルを準備するには、削除データモデルを取得し、すべての nc:operation='delete' をすべての場所から削除します。

2 つのデータモデルファイルの準備ができたら、次のコマンドを使用してサービス更新を実行します。

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli edit-config {deleting datamodel file}
```

サービス更新が完了するまで待ちます。次に、VNF を追加し直します。

警告 VNF の追加し直す前に、サービスは稼働状態になっている必要があります。稼働状態になっていない場合は、該当 VM をリカバリしてください。

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli edit-config {adding datamodel file}
```

非アクティブ状態の VNF サービス（展開）の回復

ESC リリース 3.1 以降のリリースでは、VM の停止操作が失敗すると、サービスが非アクティブ状態でスタックし、リカバリがトリガーされない場合があります。1 つの VM はエラー状態ですが、サービスは非アクティブ状態になります。ENABLE MONITOR を使用して、VM とサービスを稼働状態またはエラー状態に戻すことができます。

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli vm-action ENABLE_MONITOR {VM Name}
```

この操作により、VM のモニタリングを有効にできます。VM が VIM で実行されている場合、VM ALIVE イベントは VM ステートマシンに戻す必要があります。VM は最終的に稼働状態に遷移します。VM が VIM で実行されていない場合、タイマーが期限切れになり、リカバリ手順で VM を戻すことができます。その間、サービスはアクティブまたはエラー状態に遷移します。

サービスステータスが正しくないために VNF リカバリが拒否される

問題

VNF VM の手動リカバリを実行すると、誤ったサービスステータス（ESC 3.1 以降のリリース）が原因でリクエストが拒否されます（以下を参照）。

```
$ /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO vm-name

Recovery VM Action
```

```
/opt/cisco/esc/confd/bin/netconf-console --port=830 --host=127.0.0.1 --user=admin
--privKeyFile=/home/admin/.ssh/confd_id_dsa --privKeyType=dsa
--rpc=/tmp/esc_nc_cli.L1WdqyIE7r
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <rpc-error>
    <error-type>application</error-type>
    <error-tag>operation-failed</error-tag>
    <error-severity>error</error-severity>
    <error-path xmlns:esc="http://www.cisco.com/esc/esc"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
    /nc:rpc/esc:recoveryVmAction
  </error-path>
    <error-message xml:lang="en">Exception from action callback: Recovery VM Operation:
 recovery_do is not applicable since the service is in [SERVICE_INERT_STATE]
state.</error-message>
    <error-info>
      <bad-element>recoveryVmAction</bad-element>
    </error-info>
  </rpc-error>
</rpc-reply>
```

解決方法

この時点で、opdata をチェックして、サービスの状態と VM の状態を調べます。

```
$ /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli get esc_datamodel/opdata

  <state_machine>
    <state>SERVICE_INERT_STATE</state>
    <vm_state_machines>
        <vm_state_machine>
            <vm_name>depz_g1_0_b6d19896-bc3b-400a-ad50-6d84c522067d</vm_name>
            <state>VM_MONITOR_UNSET_STATE</state>
        </vm_state_machine>
        <vm_state_machine>
            <vm_name>depz_g1_1_f8445a8a-29ba-457d-9224-c46eeaa97f72</vm_name>
            <state>VM_ALIVE_STATE</state>
        </vm_state_machine>
     </vm_state_machines>
  </state_machine>
```

未設定監視状態の VM の監視を有効にします。

```
$ /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli vm-action ENABLE_MONITOR vm-name
```

しばらくしてから、opdata を再度確認します。サービスはアクティブ状態またはエラー状態に遷移する必要があります。

```
$ /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli get esc_datamodel/opdata

        <state_machine>
            <state>SERVICE_ACTIVE_STATE</state>
            <vm_state_machines>
                <vm_state_machine>
                    <vm_name>depz_g1_0_b6d19896-bc3b-400a-ad50-6d84c522067d</vm_name>
                    <state>VM_ALIVE_STATE</state>
                </vm_state_machine>
                <vm_state_machine>
                    <vm_name>depz_g1_1_f8445a8a-29ba-457d-9224-c46eeaa97f72</vm_name>
                    <state>VM_ALIVE_STATE</state>
                </vm_state_machine>
            </vm_state_machines>
        </state_machine>
```

エラー状態の VM を手動でリカバリします。

```
$ /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO vm-name
```

VIM コネクタの問題により VNF 操作が拒否される

問題

VNF 操作（展開、リカバリなど）が次の理由で拒否される：

```
Default VIM Connector is not set up, or is unreachable. Please check your VIM Connector
 credentials and VIM status.
```

説明

ESC が複数 VIM 用に設定されている場合は、少なくとも 1 つの VIM コネクタが「デフォルト」としてマークされていることを確認してください。それ以外の場合は、ESC VIM コネクタのステータスを確認してください。

```
[admin@leishi-test ~]$ escadm vim show
[
    {
        "status": "CONNECTION_SUCCESSFUL",
        "status_message": "Successfully connected to VIM",
        "type": "OPENSTACK",
        "id": "default_openstack_vim",
        "properties": {
            "property": [
                {
                    "name": "os_project_domain_name",
                    "value": "default"
                },
                {
                    "name": "os_auth_url",
                    "value": "http://10.85.103.143:35357/v3"
                },
                {
                    "name": "os_project_name",
                    "value": "admin"
                }
            ]
        }
    }
]
{
    "user": [
        {
            "credentials": {
                "properties": {
                    "property": [
                        {
                            "name": "os_password",
                            "value": "cisco123"
                        },
                        {
                            "name": "os_user_domain_name",
                            "value": "default"
                        }
                    ]
                }
            },
```

```
            "vim_id": "default_openstack_vim",
            "id": "admin"
        }
    ]
}
```

解決方法

VIM コネクタが返されない場合は、追加します。1 つの VIM コネクタが返され、ステータスが「CONNECTION_SUCCESSFUL」でない場合は、/var/log/esc/vimmanager/vimmanager.log で次のエントリを確認します。

```
2017-12-07 23:11:49.760 [http-nio-127.0.0.1-8095-exec-5] INFO
c.c.e.v.c.VimConnectionManagerService - Registering an user.
```

エントリの後に例外またはエラーが表示されている場合は、根本原因を示しています。たとえば、SSL に関連するエラーがある場合、証明書が見つからないか、間違っていることを意味します。

```
2017-12-07 23:11:49.818 [http-nio-127.0.0.1-8095-exec-5] ERROR
c.c.e.v.p.i.o.OpenStackProvider - Failed to register a user
org.openstack4j.api.exceptions.ConnectionException: javax.net.ssl.SSLHandshakeException:
 sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid
certification path to requested target
    at
org.openstack4j.connectors.jersey2.HttpExecutorServiceImpl.invoke(HttpExecutorServiceImpl.java:58)
```

接続タイムアウトがある場合、またはホスト名が例外の名前でない場合は、提供された AuthUrl に対して CURL get 呼び出しを試し、OpenStack が ESC VM から到達可能であることを確認します。

```
curl -k https://www.xxxxx.com:5000/
```

「Register an user」エントリの後にエラーも例外も表示されない場合は、提供されたログイン情報が正しくないことを意味します。この場合、/var/log/esc/vimmanager/vim_vimmanager.log を確認してください。最初の認証が行われたログファイルの先頭を確認します。

```
2017-12-07 23:11:49.748 INFO  os -   1 * Sending client request on thread
http-nio-127.0.0.1-8095-exec-4
1 > POST https://10.85.103.49:35357/v3/auth/tokens
1 > Accept: application/json
1 > Content-Type: application/json
1 > OS4J-Auth-Command: Tokens
{
  "auth" : {
    "identity" : {
      "password" : {
        "user" : {
          "name" : "admin",
          "domain" : {
            "name" : "default"
          },
          "password" : "****"
        }
      },
      "methods" : [ "password" ]
    },
    "scope" : {
      "project" : {
        "name" : "admin",
        "domain" : {
```

```
          "name" : "default"
        }
      }
    }
  }
}
```

authUrl、ユーザー、プロジェクトまたはテナントを再確認します。V3 認証の場合、authUrl が実際の V3 エンドポイントであることを確認してください。そうでない場合は、404 が返されます。V3 認証の場合も、ユーザードメインとプロジェクトドメインが提供されていることを確認します。Horizon の OpenRC ファイルを使用して ESC VM を起動し、OpenRC にプロジェクトドメインまたはユーザードメインが含まれていない場合は、明示的に宣言します。

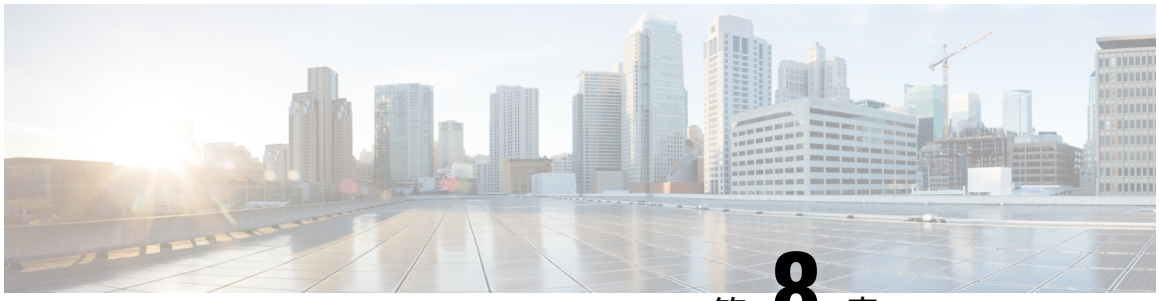
```
OS_PROJECT_DOMAIN_NAME=default
OS_USER_DOMAIN_NAME=default
```

ESC が bootvm を使用して、デフォルトの VIM コネクタの正しいパスワードを取得するか確認するには、次の手順を実行します。

```
admin@leishi-test ~]$ sudo escadm reload
[sudo] password for admin:
[admin@leishi-test ~]$ cat /opt/cisco/esc/esc-config/esc_params.conf
openstack.os_auth_url= http://10.85.103.153:35357/v3
openstack.os_project_name= admin
openstack.os_tenant_name= admin
openstack.os_user_domain_name= default
openstack.os_project_domain_name= default
openstack.os_identity_api_version= 3
openstack.os_username = admin
openstack.os_password = cisco123
```

第 VII 部

Cisco Elastic Services Controller のインストール後に関するトラブルシューティング



第 8 章

Cisco Elastic Services Controller のインストール後に関するトラブルシューティング

Cisco Elastic Services Controller のインストール後に関するトラブルシューティング

直面する可能性のある問題の例を次に示します。

始める前に

問題：

ESC と管理およびオーケストレーション スタック内の他のコンポーネント間の通信に問題がある場合、いくつかの理由が考えられます。

手順

ステップ 1 クライアントがリクエストを行い、サーバーを呼び出した後にリクエストがハングします。

ステップ 2 通知はコンポーネント間で正常に送信されていません。以下に、mona.log から抜粋した例を示します。この例では、D-MONA エージェントが中央の ESC に通知を送信できていません。

```
2022-03-01 16:44:31.355 [QuartzScheduler_Worker-2] [ruleName] :
rule-VM_ALIVE-1-esc-etsi-sol3-vnf-info-001__319bb3d8-71c4-4957-8fe3-34240296d8e9__0:PostVmEventAction
 execution, Post to url:https://x.x.x.x:8443/ESCManager/dmona/api/events/notif,
payload:rule-VM_ALIVE-1-esc-etsi-sol3-vnf-info-001__319bb3d8-71c4-4957-8fe3-34240296d8e9__0
2022-03-01 16:44:41.488 [QuartzScheduler_Worker-2] Error during esc post execution:I/O
error on POST request for "https://x.x.x.x:8443/ESCManager/dmona/api/events/notif": Read
 timed out; nested exception is java.net.SocketTimeoutException: Read timed out
```

Cisco Elastic Services Controller のインストール後に関するトラブルシューティング

次のタスク

ソリューション：

クライアントからサーバーにcurlコマンドを発行して、ルーティングの問題を確認します。小さいペイロードまたは空のペイロードを送信し、サーバーがペイロードを確実に受信できるようにします。ルーティングに問題がないことを確認したら、クライアントが試みている実際の呼び出しを行い、呼び出しが失敗するのを待ちます。

システムがインターフェイスを介してデータを送信しようとすると、インターフェイスで設定された最大伝送ユニット（MTU）サイズに基づいて、データはパケットにフラグメント化されます。

たとえば、D-MONAはオーケストレーション ネットワーク上のVNFと通信しますが、ESCは管理ネットワークを介して通信します。管理ネットワークが通知を正常に受信しない場合は、MTUサイズを確認してください。

多くの場合、MTUサイズが大きいとイベントで機能しないため、パケットを宛先に送信する中間システムがあります。中間システムは、よりサイズが小さいMTUで構成されます（パケットは入力および出力インターフェイスを介して転送されるため）。

インターフェイスのMTUサイズを確認するには、次のように入力します。

```
$ ifconfig eth0
ether: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9000
        inet 172.29.0.33  netmask 255.255.240.0  broadcast 172.29.15.255
```

安全な値1500に更新するには、次のように入力します。

```
$ sudo ifconfig eth0 mtu 1500
```

ここで、クライアントが失敗した実際の呼び出しを再試行し、サーバーが正常に受信したことを確認します。

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。