



ESC 正常性のモニタリング

ESC とそのサービスの正常性を監視するには、次のいずれかを使用します。

- ヘルス API のモニタリング
- SNMP トラップ
- [REST API を使用した ESC の正常性のモニタリング \(1 ページ\)](#)
- [SNMP トラップ通知を使用した ESC の正常性のモニタリング \(8 ページ\)](#)

REST API を使用した ESC の正常性のモニタリング

ESC は、ESC およびそのサービスの正常性を監視するためのサードパーティ製ソフトウェアに REST API を提供します。サードパーティ製ソフトウェアは API を使用して ESC が正常な状態であるかを定期的に照会し、ESC が稼働中であるかどうかを確認できます。クエリへの応答として、API はステータスコードとメッセージを提供します。詳細については、[表 1: スタンドアロンおよびアクティブ/スタンバイ ハイアベイラビリティにおける ESC ヘルス API のステータスコードとメッセージ \(3 ページ\)](#) を参照してください。HA セットアップでは、仮想 IP (VIP) をモニタリング IP として使用する必要があります。戻り値で、ESC HA ペアの全体的な状態が示されます。詳細については、[表 3: スタンドアロン ESC と HA のヘルス API ステータスメッセージ \(6 ページ\)](#) を参照してください。

ESC の正常性を監視する REST API は次のとおりです。

```
GET to https://<esc_vm_ip>:8060/esc/health
```



- (注)
- ヘルス API のモニタリングは、既存の REST の基本的な HTTP 認証を使用して保護されません。ユーザは ESC REST API クレデンシャルを使用してレポートを取得できます。
 - ESC ヘルス API ポート番号が 60000 から 8060 に変更されました。

次に、エラー状態のヘルス API のモニタリングの応答を示します。

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```

<esc_health_report>
<status_code>{error status code}</status_code>
<message>{error message}</message>
</esc_health_report>

```

ローカルアクティブ/アクティブのヘルス API のモニタリングの応答は次のとおりです。

```

<?xml version="1.0" encoding="UTF-8" ?>
<esc_health_report>
  <status_code>2010</status_code>
  <message>ESC service is being provided. ESC AA cluster one or mode node(s) not
healthy</message>
  <nodes>
    <node>
      <name>aa-esc-1.novalocal</name>
      <status>HEALTHY</status>
      <datacenter>dcl</datacenter>
      <services>
        <service>
          <name>escmanager</name>
          <status>running</status>
          <is_expected>True</is_expected>
        </service>
        <service>
          <name>elector</name>
          <status>leader</status>
          <is_expected>True</is_expected>
        </service>
        <service>
          <name>drbd</name>
          <status>master</status>
          <is_expected>True</is_expected>
        </service>
        <service>
          <name>pgsql</name>
          <status>running</status>
          <is_expected>True</is_expected>
        </service>
        ...
      </services>
    </node>
    <node>
      <name>aa-esc-2.novalocal</name>
      <status>HEALTHY</status>
      <datacenter>dcl</datacenter>
      <services>
        <service>
          <name>escmanager</name>
          <status>running</status>
          <is_expected>True</is_expected>
        </service>
        <service>
          <name>elector</name>
          <status>follower</status>
          <is_expected>True</is_expected>
        </service>
        <service>
          <name>drbd</name>
          <status>backup</status>
          <is_expected>True</is_expected>
        </service>
        <service>
          <name>pgsql</name>
          <status>stopped</status>
          <is_expected>True</is_expected>
        </service>
      </services>
    </node>
  </nodes>
</esc_health_report>

```

```

        </service>
        ...
    </services>
</node>
<node>
  <name>aa-esc-3.novalocal</name>
  <status>NOT_HEALTHY</status>
  <datacenter>dc1</datacenter>
  <services>
    <service>
      <name>escmanager</name>
      <status>stopped</status>
      <is_expected>False</is_expected>
    </service>
    <service>
      <name>elector</name>
      <status>follower</status>
      <is_expected>True</is_expected>
    </service>
    <service>
      <name>vimmanager</name>
      <status>running</status>
      <is_expected>True</is_expected>
    </service>
    ...
  </services>
</node>
</nodes>
</esc_health_report>

```

XML 応答と JSON 応答は、ヘルス API のモニタリングでもサポートされています。

API 応答が成功すると、*stage* という追加のフィールドが導入されます。

```

<?xml version="1.0" encoding="UTF-8" ?>
<esc_health_report>
<status_code>{success status code}</status_code>
<stage>{Either INIT or READY}</stage>
<message>{success message}</message>
</esc_health_report>

```

stage フィールドには、INIT パラメータまたは READY パラメータが含まれています。

INIT : INIT パラメータは ESC が設定パラメータの設定や VIM コネクタの登録などの事前プロビジョニング要求を受け入れる初期段階のものであります。

READY : ESC は、このパラメータを使用した展開、展開解除などのあらゆるプロビジョニング要求に対応できます。

ESC の正常性の状態が次のステータスコードとメッセージで示されます。2000 シリーズのステータスコードは、ESC が動作していることを意味します。5000 シリーズのステータスコードは、1 つ以上の ESC コンポーネントが稼働していないことを意味します。

(注) ESC のヘルス API は VIM のステータスを確認しません。これは、複数 VIM の展開が ESC リリース 3.0 で導入されたためです。

表 1: スタンドアロンおよびアクティブ/スタンバイ ハイアベイラビリティにおける ESC ヘルス API のステータスコードとメッセージ

ステータスコード	メッセージ
2000	ESC サービスが実行されています。(ESC services are running.)
2010	ESC サービスが提供されています。(ESC services are being provided.) ESC AA クラスターの 1 つまたは複数のノードが正常ではありません。(ESC AA cluster one or more node(s) not healthy.)
2020	ESC サービスが実行されています。(ESC services are running.) 1 つ以上の VIM サービス (Keystone や Nova) に到達できません。(One or more VIM services (for example, keystone and nova) not reachable.) (注) ESC リリース 3.0 以降はサポートされていません。
2030	ESC サービスは実行されていますが、VIM クレデンシャルが指定されていません。(ESC services are running, but VIM credentials are not provided.) (注) ESC リリース 3.0 以降はサポートされていません。
2040	ESC サービスが実行されています。VIM が設定されており、ESC が VIM への接続を初期化しています。(ESC services running. VIM is configured, ESC initializing connection to VIM.)

ステータス コード	メッセージ
2100	ESC サービスは実行されていますが、ESC ハイアベイラビリティノードに到達できません。1つ以上の VIM サービス (Nova など) に到達できません。(ESC services are running, but ESC High-Availability node is not reachable. One or more VIM services (for example, nova) are not reachable.) (注) ESC リリース 3.0 以降はサポートされていません。
5010	ESC サービス、ESC_MANAGER が実行されていません。(ESC service, ESC_MANAGER is not running.)
5020	ESC サービス、CONFD が実行されていません。(ESC service, CONFD is not running.)
5030	ESC サービス、MONA が実行されていません。(ESC service, MONA is not running.)
5040	ESC サービス、VIM_MANAGER が実行されていません。(ESC service, VIM_MANAGER is not running.)
5090	複数の ESC サービス (ConfD や Mona など) が実行されていません。(More than one ESC service (for example, confd and mona) are not running.)

表 2: アクティブ/アクティブハイアベイラビリティにおける ESC ヘルス API のステータスコードとメッセージ

ステータス コード	メッセージ
2000	ESC サービスが実行されています (アクティブ/アクティブセットアップ)。(ESC services are running (Active-Active setup).)
2010	ESC サービスが提供されています。(ESC services are provided.) ESC アクティブ/アクティブクラスタの 1 つまたは複数のノードが正常ではありません。(In ESC Active/Active cluster one or more node(s) are not healthy.)

ステータスコード	メッセージ
5000	ESC サービスが提供されていません。ESC AA クラスタが正常ではありません (ESC services not being provided, ESC AA cluster not healthy)



(注) ESC HA モードでは、DRBD セットアップでのみ ESC HA を参照します。ESC HA セットアップの詳細については、『[Cisco Elastic Services Controller Install Guide](#)』を参照してください。

次の表では、スタンドアロン ESC のステータスメッセージと、成功シナリオと障害シナリオの HA について説明します。ESC のスタンドアロンおよび HA のセットアップの詳細については、『[Cisco Elastic Services Controller Install Guide](#)』を参照してください。

表 3: スタンドアロン ESC と HA のヘルス API ステータスメッセージ

	Success	Partial Success	Failure
スタンドアロン Esc	応答はヘルス API のモニタリングから収集され、ステータスコードは 2000 になります。	NA	<ul style="list-style-type: none"> モニタは、ヘルス API のモニタリングからの応答を取得できません。 応答はヘルス API のモニタリングから収集され、ステータスコードは 5000 シリーズで返されます。
HA の ESC (アクティブ/スタンバイ)	応答はヘルス API のモニタリングから収集され、ステータスコードは 2000 になります。	応答はヘルス API のモニタリングから収集され、ステータスコードは 2010 になります。これは、ESC スタンバイノードが ESC HA の ESC マスターノードに接続できないことを示します。ただし、これはノースバウンドへの ESC サービスには影響しません。	<ul style="list-style-type: none"> モニタは、2 分以上にわたってヘルス API のモニタリングの応答を取得できません。 <ul style="list-style-type: none"> (注) HA スイッチオーバー時の特定の期間は ESC のヘルス API のモニタリングが使用できない場合があります。モニタリングソフトウェアは、このシナリオでサービス障害を報告するように適切なしきい値を設定する必要があります。 応答はヘルス API のモニタリングから収集され、ステータスコードは 5000 シリーズで返されます。

	Success	Partial Success	Failure
HA (アクティブ/アクティブ) のESC	<p>応答はヘルス API のモニタリングから収集され、ステータスコードは 2000 になります。</p>	<p>応答はヘルス API のモニタリングから収集され、ステータスコードは 2010 になります。この状態は、ESC サービスは提供されているが、ESC AA クラスタ内の 1 つまたは複数のノードが正常ではないことを示します。ただし、これはノースバウンドへの ESC サービスには影響しません。</p>	<ul style="list-style-type: none"> ローカルアクティブ/アクティブでは、モニタが 2 分以上にわたってヘルス API のモニタリングの応答を取得できない場合です。 <p>アクティブ/アクティブ GEO では、モニタが 7 分以上にわたってヘルス API のモニタリングの応答を取得できない場合です (Heat テンプレートの設定によって異なります)。</p> <p>(注) ローカルおよび GEO スイッチオーバー時の特定の期間は ESC のヘルス API のモニタリングが使用できない場合があります。モニタリングソフトウェアは、このシナリオでサービス障害を報告するように適切なしきい値を設定する必要があります。</p> <ul style="list-style-type: none"> GEO スイッチオーバー期間は、Heat テンプレートの設定によって異なります。デフォルトでは、スイッチオーバーはプライマリデータセンターの障害発生から 5 分後に開始されます。 <p>応答はヘルス API のモニタリングから収集され、ステータスコードは 5000 で返されます。</p> <p>(注) スイッチオーバー中は、新しいリーダーが正常になるまで、ステータスコードは一時的に 5000 で返されます。</p>

SNMP トラップ通知を使用した ESC の正常性のモニタリング

また、SNMP エージェントを使用し、SNMP トラップを介してさまざまな ESC コンポーネントの正常性に関する通知を設定することもできます。このエージェントは、標準の ESC インストールの一部としてインストールされ、SNMP バージョン 2c プロトコルをサポートしています。SNMP トラップは現在、ESC で管理されている VNF ではなく、ESC 製品の状態のみをサポートしています。この項では、ESC SNMP エージェントを設定するために必要な手順について説明します。また、通知の一部としてトリガーされるイベントについても説明します。

始める前に

- **CISCO-ESC-MIB** ファイルと **CISCO-SMI MIB** ファイルがシステムで使用できることを確認します。これらのファイルは /opt/cisco/esc/snmp/mibs ディレクトリにあります。これらのファイルを SNMP マネージャマシンにダウンロードし、\$HOME/.snmp/mibs ディレクトリに配置します。
- SNMP エージェントを設定します。SNMP エージェントを設定するには、次の 3 つの方法があります。これらの方法については、次の項で詳しく説明します。

SNMP エージェントの設定

SNMP トラップを受信するには、SNMP エージェントパラメータを設定します。エージェントは、この項で説明する 3 つの異なる方法を使用して設定できます。使用する最良または最適な方法は、用途によって異なります。

1. ESC のインストール時の SNMP エージェントの有効化および設定 :

• BootVM によるスタンドアロンまたはアクティブ/スタンバイ HA セットアップ

ESC のインストール中に、次の追加パラメータを使用して SNMP エージェントを設定します。

```
% bootvm.py <esc_vm_name> --image <image-name> --net <net-name> --enable-snmp-agent
--ignore-ssl-errors
--managers "udp:ipv4/port,udp:[ipv6]/port"
```



(注) マネージャの値は、SNMP トラップが「udp:ipv4/port」または「udp:[ipv6]/port」形式で配信される場所のカンマ区切りリストです。IP とポートは実際の値に置き換える必要があります。

• アクティブ/アクティブ HA 設定

アクティブ/アクティブインストール中に SNMP エージェントを有効にできます。設定パラメータ ignore_ssl_errors および managers リストを渡して、インストール時に

エージェントを設定できます。aa-params.yaml で定義するか、次のコマンドラインで渡すことができます。

```
openstack stack create name-aa --template aa.yaml -e aa-params.yaml \
--parameter nameprefix=ESC_AA \
--parameter image_name=ESC-5_2_0_43 \
--parameter flavor_name=m1.large \
...
--parameter snmp_agent_startup: auto \
--parameter snmp_agent_ignore_ssl_errors: true \
--parameter snmp_agent_managers: [ "udp:ipv4/port,udp:[ipv6]/port" ]
```

2. ESCADM による有効化と設定

• スタンドアロンまたはアクティブ/スタンバイ HA 設定

ESCADM ツールを使用して、マネージャや ignoreSslErrors プロパティなどの SNMP エージェント設定パラメータを変更できます。

```
sudo escadm snmp set --ignore_ssl_errors=true
--managers="udp:ipv4/port,udp:[ipv6]/port"
```

• アクティブ/アクティブ HA 設定

ESC ノード 1、ノード 2、ノード 4、およびノード 5 のすべてのリーダー対応ノードで、次のコマンドを実行します。

```
sudo escadm snmp set --startup=auto
```



(注) スタック更新によってノードが削除され、再作成された場合は、以前のコマンドを再実行する必要があります。

プライマリデータセンターの SNMP 対応ノード（ノード 1 および 2）でのみ、ESC サービスを再起動します。一度に 1 つのノードです。

```
sudo escadm stop
sudo escadm start
```

リーダーノードが正常になり、SNMP エージェントが実行されたら、リーダーノードに SNMP エージェント設定を次のように追加できます。

```
sudo escadm snmp set --ignore_ssl_errors=true
--managers="udp:ipv4/port,udp:[ipv6]/port"
```



(注) ignore-ssl-errors パラメータは主に、ESC VM で自己署名証明書が使用される SSL エラーを防止する開発者環境用です。

マネージャの値は、SNMP トラップが「udp:ipv4/port」または「udp:[ipv6]/port」形式で配信される場所のカンマ区切りリストです。IP とポートは実際の値に置き換える必要があります。

3. 設定ファイルの更新

この設定の更新を有効にするには、SNMP エージェントがすでに有効になっている必要があります。

設定は、`/opt/cisco/esc/esc_database/snmp.conf` ファイルにあります。このファイルは JSON 形式です。次に例を示します。

```
{
  "sysDescr": "ESC SNMP Agent",
  "listeningPort": "2001",
  "managers": [
    "udp:[ipv4]/port",
    "udp:[ipv6]/port"
  ],
  "ignoreSslErrors": "yes",
  "logLevel": "INFO",
  "sysLocation": "Unspecified",
  "sysName": "system name",
  "pollSeconds": "15",
  "listeningAddress": "0.0.0.0",
  "healthUrl": "https://<esc_vm_ip>:8060/esc/health",
  "sysContact": "root@localhost"
}
```

ESC SNMP MIB の定義

次の表で、ESC MIB の内容について説明します。これらの値は、`snmp.conf` ファイルで設定できます。

変数	Simple IOD	説明
sysName	SNMPv2-MIB::sysName.0	ESC マシンの名前を指定します。デフォルトでは、ホスト名が取得されます。
sysDescr	SNMPv2-MIB::sysDescr.0	SNMP エージェントの名前を指定します。
sysLocation	SNMPv2-MIB::sysLocation.0	ESC マシンが配置されている場所を指定します。
sysContact	SNMPv2-MIB::sysContact.0	管理者の連絡先を指定します。

SNMP トラップ通知の有効化

ESCADM ツールを使用して、SNMP サービスを開始します。

```
sudo escadm snmp start
```

また、ESCADM ツールを使用してステータスの取得を停止したり、SNMP エージェントの設定を変更できます。

```
sudo escadm snmp stop
sudo escadm snmp status
sudo escadm snmp restart
```

ESC での SNMP トラップの管理

この項の内容は、次のとおりです。

- ESC での SNMP 通知タイプについて
- ネットワークからの SNMP トラップメッセージの直接受信
- トラップエンドポイントの管理 (SNMP マネージャ)
- HA 環境での ESC SNMP の管理
- アクティブ/アクティブ環境での ESC SNMP エージェントの管理
- ESC での自己署名証明書の管理

手順

- **ESC での SNMP 通知タイプについて** : 次の表に、このバージョンの SNMP エージェントでサポートされているすべてのイベントを示します。これらのステータスコードとメッセージは、ESC の状態が変更された場合にのみ、登録されたマネージャに SNMP トラップを介して返されます。2000 シリーズのステータスコードは、ESC が動作していることを意味します。5000 シリーズのステータスコードは、1 つ以上の ESC コンポーネントが稼働していないことを意味します。2000 シリーズおよび 5000 シリーズのステータスコードの詳細については、「REST API を使用した ESC の正常性のモニタリング」の項を参照してください。

ステータスコード	SNMP エージェント固有のメッセージ
5100	ESC モニタ API の使用時に HTTP エラーが発生しました。
5101	ESC モニタは応答しましたが、データを理解できませんでした。
5102	エージェントは ESC モニタ API へのネットワーク接続を作成できませんでした。
5199	未処理のエラーが発生しました (詳細はメッセージに示されます)。

ステータスコード	SNMP エージェント固有のメッセージ
5210	「AA リーダーノードが変更されました」。 ("AA LEADER node change".) ノードが リーダーになった AA 環境では、ノード上 のエージェントがこの通知を送信します。 ローカルリーダーの変更のみの場合です。
5200	「HA マスターノードが変更されました」。 ("HA MASTER node change".) ノードが マスターノードになった A/SHA 環境では、 エージェントがこの通知を送信します。
5220	「GEO AA プライマリデータセンターが変 更されました」 ("Geo AA Primary datacenter change") GEO AA 環境では、 GEO スイッチオーバー後にノードがリー ダーになったとき、ノード上のエージェン トがこの通知を送信します。GEO リーダー の変更のみの場合です。

- ネットワークからの SNMP トラップメッセージの直接受信 : snmpget snmpwalk や snmptrapd などの基本的な SNMP UNIX ツールを使用して、ネットワークから SNMP トラップメッセージを直接受信します。使用例 :

```
snmptrapd -m ALL -f -Lo -c snmptrapd.conf <port>
```

これにより、ポート 12113 で SNMP トラップデーモンが開始されます。シスコおよび ESC MIB が ~/.snmp/mibs に存在することを確認します。参照された snmptrapd.conf は次のようになります。

```
disableAuthorization yes
authCommunity log,execute,net public
# traphandle default /Users/ahanniga/bin/notify.sh esc

createUser myuser MD5 mypassword DES myotherpassword
```

```
format2 %V\n% Agent Address: %A \n Agent Hostname: %B \n Enterprise OID: %N \n Trap
Sub-Type: %q \n Community/Infosec Context: %P \n Uptime: %T \n PDU Attribute/Value
Pair Array:\n%v \n ----- \n
```

トラップには、statusCode と statusMessage の 2 つのエントリが含まれます。このトラップは、ステータスが変化したときに送信されます。

```
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (3971) 0:00:39.71
SNMPv2-MIB::snmpTrapOID.0 = OID: CISCO-ESC-MIB::statusNotif
SNMPv2-MIB::sysDescr.0 = STRING: ESC SNMP Server
CISCO-ESC-MIB::escStatusCode.0 = STRING: "2000"
CISCO-ESC-MIB::escStatusMessage.0 = STRING: "ESC services are running."
```

- **トラップエンドポイントの管理 (SNMP マネージャ)** : SNMP エージェントは、変更の設定ファイルを監視し、変更が行われたときにリロードします。コンフィギュレーション

ファイルに対してマネージャのエンドポイントを追加または削除し、以降のトラップでは新しい設定が使用されます。

- **HA 環境での ESC SNMP エージェントの管理** : 2 つ以上の ESC ノードを HA 設定で展開し、SNMP エージェントがこの設定をサポートします。ただし、HA 展開では次の点を考慮してください。
 - 1 つの ESC ノード (マスターノード) のみが SNMP トラップを送信できます。
 - バックアップノードがマスターになった場合は、SNMP エージェントが稼働している必要があります。
 - マスター設定に加えた変更は、バックアップノードにも適用する必要があります。
 - フェールオーバーが原因でノードがマスターノードになると、トラップが生成されません。
- **AA 環境での ESC SNMP エージェントの管理** : SNMP エージェントサービスは、ローカルまたは GEO ESC アクティブ/アクティブセットアップでもサポートされます。アクティブ/アクティブ展開における考慮事項は次のとおりです。
 - SNMP エージェントはリーダーノードでのみトラップを実行し、送信します。
 - トラップは次のシナリオで送信されます。
 - ESC ヘルス API のステータスコードの変更時。SNMP エージェントは、AA のヘルスマニタ API をポーリングし、返されたステータスコードに変更がある場合は、トラップとしてサブスクライバに送信されます。
 - ローカルスイッチオーバーを示す新しいリーダーになるノードによる、ローカルスイッチオーバーの後。
 - 新しい GEO プライマリデータセンターのリーダーになるノードによる、GEO スイッチオーバーの後。
 - リーダーノードの設定に対する変更は、スイッチオーバー後に新しいリーダーによって引き継がれます。
- **自己署名証明書の管理** : ESC が展開されて SNMP エージェントが ESC のヘルス API を使用する場合は、サーバにルート信頼証明書をインストールしておくことを推奨します。環境が既知であり、信頼できるものである場合、設定パラメータ「ignoreSslErrors」を使用してこれらのエラーを無視することができます。ただし、この設定をよりセキュアなデフォルトに維持する場合は、ESC 証明書を JVM 信頼ストアにインポートすることによって、自己署名証明書をインストールできます。次の項では、これを実行する手順について説明します。
 - a) localhost の代替名として esc を追加します。ファイル「/etc/hosts:」で次のように追加します (または、「esc」が最後に追加されていることを確認します)。

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
esc
```

- b) SNMP エージェント コンフィギュレーション ファイル
「/opt/cisco/esc/esc_database/snmp.conf」では、healthUrl が ESC を指している必要があります。

```
"healthUrl": "https://esc:8060:/esc/health"
```

- c) 証明書をトラストストアにインポートします。次に、\$JAVA_HOME
is/usr/lib/jvm/jre-1.8.0-openjdk.x86_64 を想定し、証明書をインポートする例を示します。

```
cd /opt/cisco/esc/esc-config  
sudo openssl x509 -inform PEM -in server.pem -outform DER -out server.cer  
sudo keytool -importcert -alias esc -keystore $JAVA_HOME/lib/security/cacerts  
-storepass changeit -file server.cer
```