ESCに関する問題のトラブルシューティング

この章は、次の項で構成されています。

ESC ログメッセージの表示

ログメッセージは、VNF ライフサイクル全体にわたって ESC イベント用に作成されます。これらには、外部メッセージ、ESCから他の外部システムへのメッセージ、エラーメッセージ、警告、イベント、障害などがあります。ログファイルは、`/var/log/esc/escmanager_tagged.log` にあります。

次に、ログメッセージの形式を示します。

```
date=<time-date>] [loglevel=<loglevel>] [tid=<transactionid>] [cl=<classifications>]
[tags=<tags>] [msg=<message>
```

次に、ログの例を示します。

```
date=15:43:58,46022-Nov-2016]
[loglevel=ERROR ] [tid=0793b5c9-8255-47f3-81e6-fbb59f6571f7] [cl=OS ]
[tags=wf:create_vm,eventType:VM_DEPLOY_EVENT,tenant:CSCvd94541,depName:test-dep,vmGrpName:test-VNF,
vmName:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0]
[tags=wf:create_vm,eventType:VM_DEPLOY_EVENT,tenant:test,depName:test-dep,vmGrpName:test-VNF,
vmName:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0] [msg=sleepingfor5seconds
to allow vm to become ACTIVE instance id:
162344f7-78f9-4e45-9f23-34cf87377fa7
name:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0
```

要求を受信すると、一意のトランザクション ID を自動生成する RequestDetails オブジェクトが作成されます。この値は、すべてのスレッドで転送されます。分類とタグは任意です。これらは、読みやすくするためにログメッセージに追加されたプレフィックスであり、デバッグに役立ちます。分類とタグを使用すると、ログメッセージを簡単に解析し、ログ分析ツールでフィルタリングすることができます。

次に、サポートされている分類を示します。

NBI	「com.cisco.esc.rest」「com.cisco.esc.filter」（ノース バウンド インターフェイス：証明書）
SBI	「com.cisco.esc.rest」：ソースはコールバックハンドラまたは「EventsResource」（サウスバウンドインターフェイス、ESC と VIM 間）
SM	「com.cisco.esc.statemachines」は StateMachine を意味します。この分類は、StateMachine カテゴリのログを示します。
MONITORING	「com.cisco.esc.monitoring」「com.cisco.esc.paadaptor」（MONA 関連ログ）
DYNAMIC_MAPPING	「com.cisco.esc.dynamicmapping」「com.cisco.esc.db.dynamicmapping」（MONA 関連ログ）
CONFD	「com.cisco.esc.confd」
CONFD_NOTIFICATION	「com.cisco.esc.confd.notif」「com.cisco.esc.confd.ConfdNBIAdapter」
OS	「com.cisco.esc.vim.openstack」
LIBVIRT	「com.cisco.esc.vim.vagrant
VIM	「com. esc. vim」
REST_EVENT	「ESCManager_Event」「com.cisco.esc.util.RestUtils」。ログ内の REST 通知を示します。
WD	「com.cisco.esc.watchdog」
DM	「com.cisco.esc.datamodel」「com.cisco.esc.jaxb.parameters」（データモデルとリソースオブジェクト）
DB	「com.cisco.esc.db」（データベース関連ログ）
GW	「com.cisco.esc.gateway」
LC	「com.cisco.esc.ESCManager」（スタートアップ関連ログ）
SEC	「com.cisco.esc.jaas」
MOCONFIG	「com.cisco.esc.moconfig」（MOCONFIG オブジェクト関連ログ。これは ESC 開発者用の内部ログです）
POLICY	「com.cisco.esc.policy」（サービス/VM ポリシー関連ログ）
TP	「com.cisco.esc.threadpool」

ESC	「com.cisco.esc」上記にないその他のパッケージ

次に、サポートされているタグを示します。

- **ワークフロー [wf:]**：RequestDetails オブジェクトのアクションとリソースを使用して生成されます。例：「wf: create_network」
- **イベントタイプ [eventType:]**：現在のアクションをトリガーしたイベント。例：「eventType:VM_DEPLOY_EVENT」
- **リソースベース**：これらの値は、イベントで使用されるパラメータのタイプに基づいて生成されます。階層 （テナント、VM グループなど）がログに追加されます。

テナント	[tenant:<tenant name>]
ネットワーク	[tenant:<tenant id>, network:<network name>] （注） テナントは、該当する場合にのみ表示されます。
サブネット	[tenant:<tenant name or id>, network:<network name or id>, subnet:<subnet name>] （注） テナントは、該当する場合にのみ表示されます。
ユーザ	[tenant:<tenant name>, user:<user name or id>] （注） テナントは、該当する場合にのみ表示されます。
イメージ	[image:<image name>]
フレーバ	[flavor:<flavor name>]
配置	[tenant:<tenant name or id>, depName:<deployment name>]
DeploymentDetails	[tenant:<tenant name or id>, depName:<deployment name>, vmGroup:<vm group name>, vmName:<vm name>]
スイッチ	[tenant:<tenant name or id>, switch:<switch name>]
音量	[volume:<volume name>]
サービス	[svcName:<Service Registration name>]

さらに、分析とログの管理を促進するため、ESC ログを rsyslog サーバに転送することもできます。

ConfD API を使用したログのフィルタリング

ConfD API に導入されたログフィルタを使用して、ESC でログ（展開ログやエラーログなど）を照会および取得できます。テナント、展開名、およびVM名の新しいフィルタが導入されました。これにより、ConfD APIのログフィルタを使用して、最新のエラーログのESCログをさらに照会することができます。ESC と OS 間の通信に関連する ESC ログを取得することもできます（分類タグを「OS」に設定します）。

次に、ConfD API ログを取得するためのログ形式を示します。

```
date=<time-date>] [loglevel=<loglevel>] [tid=<transactionid>] [cl=<classifications>]
[tags=<tags>] [msg=<message>
```

次に、サンプルログの例を示します。

```
date=15:43:58,46022-Nov-2016] [loglevel=ERROR ] [tid=0793b5c9-8255-47f3-81e6-fbb59f6571f7]
 [cl=OS ]
[tags=wf:create_vm,eventType:VM_DEPLOY_EVENT,tenant:test,depName:test-dep,vmGrpName:test-VNF,
vmName:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0]
[msg=sleepingfor5seconds to allow vm to become ACTIVE instance id:
162344f7-78f9-4e45-9f23-34cf87377fa7
name:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0
```

ログレベル、分類、およびタグのパラメータは、ログを取得するために相互に依存します。次の組み合わせを使用してログを正常に取得できます。

- log_level=ERROR, classifications=OS, tags=(depName:test-dep)
- log_level=ERROR, classifications=OS, tags=(tenant: test)

ログフィルタは、次の条件がすべて満たされたときに値を返します。

- ログ レベル
- 分類（指定されている場合）
- タグ（指定されている場合）

（注） 複数の分類がリストされている場合は、1 つ以上の分類に一致する必要があります。同じことが、タグにも適用されます。

たとえば、次のログフィルタ条件では、前述のログサンプルを返しません。

```
log_level=ERROR, classifications=VIM, tags=(depName:test-dep)
```

ログレベルとタグが一致していても、分類の VIM が一致していないので値は返されません。

次に、データモデルを示します。

```
rpc filterLog {
    description "Query and filter escmanager logs using given parameters";
    tailf:actionpoint escrpc;
    input {
      leaf log_level {
        mandatory false;
       description "One of DEBUG / INFO / WARNING / ERROR / TRACE / FATAL. Results will
 include all logs at and
                     above the level specified";
        type types:log_level_types;
        default ERROR;
      }
      leaf log_count {
        mandatory false;
        description "Number of logs to return";
        type uint32;
        default 10;
      }
```

```
      container classifications {
        leaf-list classification {
          description "Classification values to be used for the log filtering.  For
example: 'OS', 'SM'.
                      Logs containing any of the provided classification values will be
 returned.";
          type types:log_classification_types;
        }
      }
      container tags {
        list tag {
          key "name";
          leaf name {
            mandatory true;
          description "Tag name to be used for the log filtering.  For example: 'tenant',
 'depName'.
                       Logs containing any of the provided tag name plus the tag values
 will be returned.";
            type types:log_tag_types;
          }
          leaf value {
            mandatory true;
           description "Tag value pairs to be used for the log filtering.  For example:
 'adminTenant', 'CSRDeployment'";
            type string;
          }
        }
      }
    }
    output {
      container filterLogResults {
        leaf log_level {
          description "Log level used to filter for the logs.";
          type types:log_level_types;
        }
        list logs {
          container classifications {
            leaf-list classification {
              description "Classifications used to filter for the logs.";
              type types:log_classification_types;
            }
          }
          container tags {
            list tag {
              key "name";
              leaf name {
                mandatory true;
                description "Tag name used to filter for the logs.";
                type types:log_tag_types;
              }
              leaf value {
                mandatory true;
                description "Tag value used to filter for the logs.";
                type string;
              }
            }
          }
          leaf log_date_time {
            description "Timestamp of the log.";
            type string;
          }
          leaf log_message {
            description "The log message.";
            type string;
```

```
        }
      }
    }
  }
}
```

NETCONF コンソールまたは esc_nc_cli を使用して、ConfD API ログを照会できます。

- NETCONF コンソールを使用して、次のクエリを実行します。

```
/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=log.xml
```

- esc_nc_cli を使用して、次のクエリを実行します。

```
esc_nc_cli --user <username> --password <password> filter-log log.xml
```

次に、log.xml の例を示します。

```
<filterLog xmlns="https://www.cisco.com/esc/esc">
  <log_level>INFO</log_level>
  <log_count>1</log_count>
  <classifications>
    <classification>OS</classification>
    <classification>SM</classification>
  </classifications>
  <tags>
    <tag>
      <name>depName</name>
      <value>CSR_ap1</value>
    </tag>
    <tag>
      <name>tenant</name>
      <value>admin</value>
    </tag>
  </tags>
</filterLog>
```

応答は次のとおりです。

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <filterLogResults xmlns="https://www.cisco.com/esc/esc">
    <log_level>INFO</log_level>
    <logs>
      <classifications>
        <classification>OS</classification>
        <classification>SM</classification>
      </classifications>
      <tags>
        <tag>
          <name>depName</name>
          <value>CSR_ap1</value>
        </tag>
        <tag>
          <name>tenant</name>
          <value>admin</value>
        </tag>
      </tags>
      <log_date_time>13:06:07,575 31-Oct-2016</log_date_time>
      <log_message> No pending work flow to start.</log_message>
    </logs>
  </filterLogResults>
</rpc-reply>
```

（注） ロギング API の応答は XML 形式です。ログメッセージに XML 文字が含まれている場合はその文字がエスケープされるため、XML 準拠は解除されません。

一般的なインストールエラー

ここでは、一般的なインストールの問題とそのトラブルシューティング方法について説明します。

問題/エラー	考えられる理由	ユーザのアクション

インストール時に `sudo escadm status` を使用して ESC サービスのステータスを確認する際に発生する問題	サービスの中には、開始に時間がかかるものや、開始時に問題が発生するものがあります。	**1.** 次のいずれかの方法で、問題を特定します。
		• ログ /var/log/esc/escadm.log の確認
		`$ cat /var/log/esc/escadm.log`
		• 「-V」を escadm ステータスに追加して、ESC サービスの詳細出力を表示し、サービスが稼働中であることを確認します。
		2. 問題のあることが特定されたサービスのステータスを確認し、これらのサービスを手動で開始します。
		`$ sudo escadm <<service>> status//`
		`If the status is stopped or dead, manually start the services`
		`using the next command.`
		`$ sudo escadm <<service>> start --v`

ESC のインストール中に発生する認証に関するエラー	OpenStack のログイン情報に関する引数がありません。	OpenStack RC ファイルを取得し、OpenStack クライアントが正常に動作していることを確認します。
ESC HA 関連（アクティブ/スタンバイ）の問題		
ネットワークの問題		次の状態がないかどうかを確認します。 • 使用される両方の ESC ノードの静的 IP アドレスが、OpenStack の設定に基づいている。 • 各ネットワークインターフェイスのゲートウェイがアクセス可能である。

問題/エラー	考えられる理由	ユーザのアクション
ESC マスターノードが「マスターに切り替え中」の状態のままである。	これは、次の問題が原因である可能性があります。 • ESC HA アクティブ/スタンバイノードが、初回インストール中にピアに到達できない。 • 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）が適切に起動できない。 • CDB ファイルが破損しているため、confd を開始できない。 • ファイルシステムに問題があるため、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。	次のことを確認します。 • ESC マスター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 マスターサービスが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、相互に到達可能であることを確認する必要があります。 • ESC では /var/log/esc/esc_haagent.log（ESC 2.x）または /var/log/esc/escadm.log（ESC 3.x）にログが記録されており、問題のあるサービスを特定できます。 • esc_service および postgresql の問題については、/var/log/esc/escmanager.log にログが記録されます。

問題/エラー	考えられる理由	ユーザのアクション
ESC HA アクティブ/スタンバイの MTU 問題		ESC VM の場合は、ネットワークインターフェイスの MTU を 1500 から 1450 に減らします。MTU 値を減らすには、次の手順を実行します。 **1.** 変更するインターフェイスを特定します。/etc/sysconfig/network-scripts/ifcfg-ethX からインターフェイスにアクセスします。 X は変更するインターフェイス番号を表します。 **2.** VIM などのテキストエディタを使用して、インターフェイスの MTU 項目を追加または編集します（例：set MTU = 1450） **3.** インターフェイスを再起動します。 `network service restart`

その他の問題		ESC HA アクティブ/スタンバイをトラブルシューティングする際に役立つログがいくつかあります。
		• ESC マネージャのログは /var/log/esc/escmanager.log に格納されています。
		• ESC サービスのスタートアップ/停止に関する ESC HA アクティブ/スタンバイログは、/var/log/esc/esc_haagent.log（ESC 2.X の場合）および /var/log/esc/escadm.log（ESC 3.X の場合）に格納されています。
		• キープアライブ構成の場合：
		  • /etc/keepalived/keepalived.conf でコンフィギュレーション ファイルを確認します。
		  • キープアライブのログは、grep keepalived または vrrp を実行すると、/var/log/messages に格納されます。
		• DRBD 構成の場合：
		  • DRBD 構成は、/etc/drbd.d/esc.res のコンフィギュレーション ファイルで確認できます。
		  • DRBD のログは、grep drbd を実行すると /var/log/messages に格納されます。

問題/エラー	考えられる理由	ユーザのアクション
ESC サービスのスタートアップに関する問題		

インストール時に sudo escadm status --v を使用すると、サービスステータスによって ETSI サービスが停止していることが示されます。	ホストに複数の IP アドレスがある場合、ETSI サービスは、コールバック URL の生成時に使用する IP アドレスを特定できません。
	この原因を検証するには、ログファイル /var/log/esc/etsi-vnfm/etsi-vnfm.log に次のような例外が記録されていないかを確認します。
	コンフィギュレーションファイルで、server.host プロパティを設定します。
		1. server.host プロパティは、次のように設定します。
		`sudo escadm etsi set -server_host <IP>`
		2. ETSI サービスを開始します。
		`sudo escadm etsi start`

ESC のフェールオーバーシナリオ

- お客様が多数の VNF を展開する場合
- お客様が DB をバックアップする場合
- VNF 内の 1 つ以上の VM が再展開によって回復（新しい VM ID）
- ESC エラー：永続的な障害が発生
- お客様の DB 復元：1 つ以上の VM ID で不一致がある場合
- 同じ VNF は失敗しますが、ESC が回復しようとして失敗するのは、VIM 上で VM ID によって検出されないためです。また、ポートが使用中のために再展開に失敗します。
- これを回避するには、VM をアウトオブバンドで削除して復旧します。 VM が削除されると、ESC で検出されなくなり、ポートが再展開で使用可能になります。