



仮想ネットワーク機能の修復

- [修復の概要](#) (1 ページ)
- [VM の修復](#) (1 ページ)
- [リカバリポリシーと再展開ポリシー](#) (10 ページ)
- [ホストの有効化と無効化](#) (18 ページ)
- [通知とイベント](#) (20 ページ)

修復の概要

ライフサイクル管理の一環として、ESC は障害発生時に VNF を修復します。修復パラメータは、データモデルの KPI セクションで設定されます。展開中に指定したリカバリポリシーがリカバリを制御します。ESC はポリシー駆動型フレームワークを使用したリカバリをサポートしています。

ESC は KPI を使用して VM をモニターします。イベントは KPI 条件に基づいてトリガーされます。トリガーされるすべてのイベントに対して実行されるアクションは、展開時にルールセクションで設定されます。

VM の修復

各 VM グループは、修復を有効にするように設定されます。修復は、データモデルで定義されたリカバリポリシーを使用して、サービスの動作前と動作後の2つの段階で実行されます (VNF レベル、必要に応じて、VM レベルでオーバーライドされます)。

VM は展開され、モニタされています。ESC が VM Alive イベントを受信後、VM Down イベントを受信すると、設定済みのリカバリポリシーを使用して、修復ワークフローによる VM のリカバリが試みられます。

ESC は展開後に VM Alive を受信しない場合、タイムアウト発生時にリカバリポリシーを使用して VM を回復します。リカバリ手順はすべて、リカバリポリシーの設定、および以下で説明されているポリシー名によって異なります。追加のポリシーを使用して、VM を手動でリカバリするときの動作をオーバーライドできます。

ESCは、YANGベースのデータモデルに、修復を定義するために必要なすべてのパラメータと説明の包括的な詳細情報を提供します。ESCは、イベントとルールを定義するデータモデルXMLファイル内の2つのセクションを使用します。

- `<kpi>` セクションでは、モニタリングのタイプ、イベント、ポーリング間隔、およびその他のパラメータを定義します。
- `<rule>` セクションでは、KPI モニタリングイベントがトリガーされたときのアクションを定義します。

KPI、ルール、およびデータモデルの詳細については、[KPI](#)、[ルール](#)、および[メトリック](#)を参照してください。

設定には、次の手順が含まれます。

1. KPI の定義
2. ルールの定義

次に、データモデルで KPI を設定する例を示します。

```
<kpi>
<event_name>VM_ALIVE</event_name>
<metric_value>1</metric_value>
<metric_cond>GT</metric_cond>
<metric_type>UINT32</metric_type>
<metric_collector>
<type>ICMPPing</type>
<nicid>0</nicid>
<poll_frequency>3</poll_frequency>
<polling_unit>seconds</polling_unit>
<continuous_alarm>>false</continuous_alarm>
</metric_collector>
</kpi>
```

次の例は、すべてのイベントのルールを設定する方法を示しています。

```
<rules>
<admin_rules>
<rule>
<event_name>VM_ALIVE</event_name>
<action>ALWAYS log</action>
<action>FALSE recover autohealing</action>
<action>TRUE servicebooted.sh</action>
</rule>
</admin_rules>
</rules>
```

前述の例では、`nicid 0` で ICMP Ping をモニタする KPI を定義しています。また、属性メトリック条件とポーリングを定義しています。KPIに基づいて、`VM_ALIVE` イベントが適切な値でトリガーされます。対応するルールのアクションでは、次のステップを定義します。

- `FALSE` : VM のリカバリをトリガーします。
- `TRUE` : 定義されたアクションをトリガーします。

リカバリポリシーで設定された再起動、および再展開オプションを使用してVMでリカバリがトリガーされた場合、ESCはVMリカバリの最初のステップとしてVMを再起動します。失敗した場合、VMは展開解除され、同じデイズロ設定の新しいVMが展開されます。ESCは、以前のVMと同じネットワーク設定（MACやIPアドレスなど）を再利用しようとします。

通常、VMが到達不能な場合、ESCは到達不能なすべてのVMでVMリカバリを開始します。ESCはネットワークの停止中はVMリカバリを一時停止するため、ネットワークの停止中はVMリカバリが遅延します。ESCは到達不能なVMを検出し、最初にゲートウェイの到達可能性を評価して、ネットワーク障害の存在を検出します。

ESCがゲートウェイにpingを実行できない場合、VMの回復アクションは実行されません。ゲートウェイが到達可能になると、VMリカバリが再開されます。

二重障害状態の場合、つまり、ネットワークゲートウェイとVMの障害が同時に発生した場合、ゲートウェイが再度到達可能になった後、ESCは自動的にVMモニタリングを実行します。

ETSI APIを使用したVNFの修復の詳細については、Cisco Elastic Services Controller NFV MANO ガイド [英語] を参照してください。

リカバリポリシー

ESCには、VNFの展開時に指定できる次のVMリカバリタイプがあります。

- 自動回復
- 手動回復

ESCは、ポリシー主導型フレームワークを使用したリカバリをサポートしています。詳細については、「[リカバリポリシー（ポリシーフレームワークを使用）](#)」を参照してください。

展開データモデルで指定できるVMリカバリには、次の3種類のアクションがあります。

- **REBOOT_THEN_REDEPLOY（デフォルト）**：VMダウンイベントを受信するか、タイマーが期限切れになると、修復ワークフローは最初にVMの再起動を試行し、再起動に失敗すると、同じホストでVMの再展開を試行します。
- **REBOOT_ONLY**：VMダウンイベントを受信するか、タイマーが期限切れになると、修復ワークフローはVMの再起動のみを試行します。
- **REDEPLOY_ONLY**：VMダウンイベントを受信するか、タイマーが期限切れになると、修復ワークフローはVMの再展開のみを試行します。



- (注) VMを再展開するREBOOT_THEN_REDEPLOYおよびREDEPLOY_ONLYがポリシーに含まれ、配置ポリシーが適用されていない場合、VIMはVMを再展開するホストを決定します。



(注) ESC は、vCloud Director の手動リカバリと自動リカバリの両方をサポートしています。3種類のリカバリアクションはすべて、vCloud Director に適用されます。REBOOT_THEN_REDEPLOY がデフォルトのリカバリアクションです。vCD の展開については、[VMware vCloud Director \(vCD\) での仮想ネットワーク機能の展開](#)を参照してください。

VM の再展開を伴うリカバリアクションでは、障害があるか、削除された ESC 管理対象のエフェメラルポートとボリュームが自動的に再作成および接続され、リカバリが成功することを確認します。

自動回復

自動回復では、リカバリタイプパラメータは[自動 (Auto)]に設定されます。ESC は、リカバリポリシーで指定された <action-on-recovery> 値により、VM を自動的に回復させます。ユーザーがリカバリタイプを選択しない場合、リカバリタイプはデフォルトで自動になります。

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

手動回復

VM の手動回復

手動回復では、ESC は VM_MANUAL_RECOVERY_NEEDED 通知をノースバウンド (NB) に送信し、NB から回復のための指示を待ちます。ESC は、NB からリカバリ指示を受信すると、リカバリを実行します。展開全体の手動回復については、以下を参照してください。[展開の手動回復 \(8 ページ\)](#)

ESC は、リカバリポリシーの action-on-recovery パラメータを使用して、単一のリクエストベースでのアクションのオーバーライドもサポートします。前述の3つのリカバリアクションに加えて、次の2つのリカバリアクションを利用できます。

- **RESET_STATE_THEN_REBOOT** : VM を再起動する前に、VM の状態がリセットされ、リカバリのために VIM が VM を再起動できるようになります。これは OpenStack にのみ適用されます。
- **DISASTER_RECOVERY** : VNF が展開されている VIM が使用できなくなり、サービスを継続するために VNF を新しい VIM に移動する必要がある場合、このアクションを呼び出して VNF (個々の VM ではなくサービス全体) を新しい VIM に再展開できます。

このアクションを使用するには、VIM ロケータを更新するモデル専用サービス更新を先に実行する必要があります。この手順を実行しないと、リカバリリクエストが失敗します。このタイプのサービス更新を実行する方法の詳細については、以下を参照してください (REST API 経由のみ)。

元の VNF は削除されません。このリカバリアクションの使用は、オーケストレーションスタックから VNF に到達できないことを意味し、VIM 自体がリカバリされたときに、古い展開を手動でクリーンアップする**必要がある**と想定されるためです。

手動リカバリポリシーのデータモデルは次のとおりです。

```
<vm_group>
...
  <recovery_policy>
    <recovery_type>MANUAL</recovery_type>
    <action_on_recovery>REBOOT_THEN_REDEPLOY</action_on_recovery>
    <max_retries>3</max_retries>
  </recovery_policy>
...
</vm_group>
```

データモデルのリカバリポリシーパラメータの詳細については、[Elastic Services Controller 展開属性 \[英語\]](#) を参照してください。ESC ポータル (VMware のみ) でのリカバリポリシーの設定の詳細については、「ESCポータルを使用した VMware vCenter での VNF の展開」を参照してください。

VM_MANUAL_RECOVERY_NEEDED 通知は次のとおりです。

```
===== SEND NOTIFICATION STARTS =====
WARN  Type: VM_MANUAL_RECOVERY_NEEDED
WARN  Status: SUCCESS
WARN  Status Code: 200
WARN  Status Msg: Recovery event for VM
[manual-recover_error-g1_0_7d96ad0b-4f27-4a5a-bdf7-ec830e93d07e] triggered.
WARN  Tenant: manual-recovery-tenant
WARN  Service ID: NULL
WARN  Deployment ID: 08491863-846a-4294-b305-c0002b9e8daf
WARN  Deployment name: dep-error
WARN  VM group name: error-g1
WARN  VM Source:
WARN    VM ID: ffea079d-0ea2-4d47-ba31-26a08e6dff22
WARN    Host ID: 3a5351dc4bb7df0ee25e238a8ebbd6c6fcdf225aebcb9dff6ba10249
WARN    Host Name: my-server-27
WARN    [DEBUG-ONLY] VM IP: 192.168.0.3;
WARN  ===== SEND NOTIFICATION ENDS =====
```

VM の手動回復用 API

Confd API と REST API を使用して手動リカバリを実行できます。手動回復要求は、事前定義されたリカバリアクションを任意のアクションに上書きするように設定できます。

Netconf API recovery-vm-action DO generated vm name [xmlfile]

API を使用してリカバリを実行するには、`esc_nc_cli` にログインし、次のコマンドを実行します。

```
$ esc_nc_cli --user <username> --password <password> recovery-vm-action DO [xmlfile]
```

リカバリが実行され、リカバリ通知が NB に送信されます。



- (注) リカバリ (`recovery-vm-action DO<VM-NAME>`) は、VMが動作し、サービスがアクティブになった後に実行できます。展開が不完全な場合は、リカバリを実行する前に展開を完了する必要があります。

設定可能な手動回復中にフェールオーバーが発生した場合、手動回復は事前定義されたリカバリアクションで再開されます。

展開の移行では、常にデフォルトのリカバリポリシーを使用する必要があります。LCSベースのリカバリでは、VM/VNF手動回復のリカバリアクションを指定しないでください。モニタの有効化オプションと設定可能な手動回復オプションを同時に使用することはできません。

REST API

`http://ip:8080/ESCAPI/#!/Recovery_VM_Operations/handleOperation`

`POST /v0/{internal_tenant_id}/deployments/recovery-vm/{vm_name}`

リカバリ VM 操作ペイロード:

```
{
  "operation": "recovery_do",
  "properties": {
    "property": [
      {
        "name": "action",
        "value": "REDEPLOY_ONLY"
      }
    ]
  }
}
```

モデル専用サービス更新を実行するには、新しいパラメータを `edit-config` API に提供して、VIM でアクションが実行されないようにし、更新を ESC データモデルのみに制限します。これにより、VIM で展開を更新する準備が整うまでに、データモデルの準備を完了できます。

`http://ip:8080/ESCManager/v0/conf/edit-config?modelOnly=true`

たとえば、`<action-on-recovery>` として `DISASTER_RECOVERY` を使用してリカバリ API を呼び出す前に、VIM ロケータを更新します。

```
<?xml version="1.0" encoding="UTF-8"?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc"
  xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns:ns1="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <tenants>
    <tenant>
      <name>admin-tenant</name>
      <deployments>
        <deployment>
          <name>test-deploy</name>
          <networks>
            <network>
              <name>test-network</name>
              <locator>
                <vim_id>my-ucs-59</vim_id>
              </locator>
            </network>
          </networks>
        </deployment>
      </deployments>
    </tenant>
  </tenants>
</esc_datamodel>
```

```

        <vim_project>admin</vim_project>
      </locator>
    </network>
  </networks>
  <vm_group>
    <name>g1</name>
    <locator>
      <vim_id>my-ucs-59</vim_id>
      <vim_project>admin</vim_project>
    </locator>
    <bootup_time>120</bootup_time>
  </vm_group>
</deployment>
</deployments>
</tenant>
</tenants>
</esc_datamodel>

```



- (注) VIM が再び使用可能になったら、ディザスタリカバリシナリオで古い展開を削除する必要がありますことを忘れないでください。

この API のさらなる用途は、前述のリカバリ API を介した VM の再起動前に実行する永続ボリュームの UUID の更新です。これには、以前のバージョンの ESC のように、VM グループを削除して再度追加する必要がないという利点があります。ペイロードの例を次に示します。

```

<esc_datamodel xmlns="http://www.cisco.com/esc/esc"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:ns1="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <tenants>
    <tenant>
      <name>my-tenant</name>
      <deployments>
        <deployment>
          <name>my-dep</name>
          <vm_group>
            <name>my-vm</name>
            <bootup_time>1800</bootup_time>
            <volumes>
              <volume>
                <name>new-volume</name>
                <valid>1</valid>
                <bus>ide</bus>
                <type>lvm</type>
              </volume>
              <volume nc:operation="delete">
                <name>old-volume</name>
                <valid>1</valid>
              </volume>
            </volumes>
          </vm_group>
        </deployment>
      </deployments>
    </tenant>
  </tenants>
</esc_datamodel>

```

VM の手動回復でサポートされる VM の状態とサービスの組み合わせ

`recovery-vm-action` の API は、自動と手動の両方のリカバリタイプに適用されますが、特定の VM の状態とサービスに限ります。次のテーブルに詳細を示します。一般に、展開、サービス更新、展開解除、およびリカバリの間は、ESC は手動リカバリアクションを拒否します。

VM 状態	サービス ステート	<code>recovery-vm-action</code>
動作中 (Alive)	ACTIVE	サポート対象
接続中 (Alive)	ERROR	サポート対象
ERROR	ERROR	サポート対象

展開の手動回復

モニタリングパラメータを使用しないリカバリ

ESC は、サービスレベルでの VM の手動回復、つまり展開全体の回復をサポートします。サービスが正常に展開された後、VM の障害が原因でサービスがエラー状態に移行することがあります。ESC は、障害が発生したこれらの VM を手動で回復することも、展開回復要求によって展開全体を回復することもできます。VM 単独の手動回復については、[手動回復 \(4 ページ\)](#) を参照してください。

展開の手動回復用 API

NETCONF API と REST API を使用して手動リカバリを実行できます。

手動回復要求は、事前定義されたリカバリアクションを任意のアクションに上書きするように設定できます。



(注) 展開回復後のサービスアクティブ通知はありません。展開のサービス状態がアクティブかどうかを確認するには、`esc_nc_cli --user <username> --password <password> get esc_datamodel` などのクエリを実行する必要があります。

設定可能な手動回復中にフェールオーバーが発生した場合、手動回復は事前定義されたリカバリアクションで再開されます。

展開の移行では、常にデフォルトのリカバリポリシーを使用する必要があります。LCS ベースのリカバリでは、VM/VNF 手動リカバリのリカバリアクションを指定しないでください。モニターの有効化オプションと設定可能な手動リカバリオプションは同時に使用できません。

NETCONF API

```
svc-action RECOVER tenant-name deployment-name [xmlfile]
```

API を使用してリカバリを実行するには、`esc_nc_cli` にログインします。

REST API

```
POST /v0/{internal_tenant_id}/deployments/service/{internal_deployment_id}
Content-Type: application/xml
```



```
Accept: application/json
Callback: http://172.16.0.1:9010/
Callback-ESC-Events: http://172.16.0.1:9010/
<service_operation xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <operation>recover</operation>
</service_operation>
```

値は次のとおりです。

`internal_tenant_id` : システム管理者のテナント ID またはテナント名。

`internal_deployment_id` : 展開名。

展開の手動回復でサポートされる VM の状態とサービスの組み合わせ

`svc-action RECOVER` の API は、自動と手動の両方のリカバリタイプに適用されますが、特定の VM の状態とサービスに限ります。次のテーブルに詳細を示します。一般に、展開、サービスの更新、展開解除、およびリカバリの間は、ESC は手動回復アクションを拒否します。



- (注) サービスがアクティブまたはエラー状態の場合、ESC は VM レベルのリカバリ要求を受け入れます。

サービスリカバリ要求後にすべての VM が動作状態になっている場合、NB に通知は送信されません。

VM 状態	サービス ステート	<code>svc-action RECOVER</code>
ERROR	ERROR	サポート対象
ERROR	ERROR	サポート対象

モニタリングパラメータによるリカバリの有効化

手動回復では、モニタリングパラメータに応じて VM をリカバリできます。VM がエラー状態の場合は、エラー状態の VM を動作状態に戻すためのモニタリングパラメータを設定します。VM が回復すると、ESC は `RECOVERY_CANCELLED` 通知を送信します。VM が動作状態に復帰しない場合、リカバリプロセスがトリガーされます。詳細については、「手動回復」を参照してください。

NETCONF API

```
svc-action SET_MONITOR_AND_RECOVER <tenant-name> <dep-name>
```

リカバリ通知 :

```
===== SEND NOTIFICATION STARTS =====
WARN Type: VM_RECOVERY_INIT
WARN Status: SUCCESS
WARN Status Code: 200
WARN Status Msg: Recovery with enabling monitor first event for VM Generated ID
[dep-resource_g1_0_74132737-d0a4-4ef0-bd9e-86465c1017bf] triggered.
```



(注) モニタリングパラメータで有効化されるリカバリは、サービスレベルでの手動回復専用です。

`monitor_on_error` パラメータにより、エラー状態にある VM の継続的なモニタリングが設定されます。

```
<recovery_policy>
    <recovery_type>AUTO</recovery_type>
    <action_on_recovery>REBOOT_ONLY</action_on_recovery>
    <max_retries>1</max_retries>
    <monitor_on_error>true</monitor_on_error>
</recovery_policy>
```

デフォルト値は `false` です。

`false` を指定すると、エラー状態にある VM のモニタリングは設定解除されます。

`true` を指定すると、エラー状態にある VM のモニタリングは設定されます。後から VM 稼働イベントが発生した場合 (`VM_RECOVERY_COMPLETE` の後)、VM は稼働状態に戻ります。

リカバリポリシーと再展開ポリシー

ESCは、ポリシー駆動型フレームワークを使用して、展開のライフサイクルステージに基づいてアクションを実行します。展開は、そのライフサイクルを通じて複数のステージで構成されます。各ライフサイクルステージ (LCS) は、条件に関連付けられています。条件は、定義済みのアクションまたはカスタムスクリプトに関連付けられています。それらの条件とアクションは、データモデルの `policy` タグ内で指定されます。ポリシー駆動型フレームワークの詳細については、[ポリシー駆動型データモデル](#)を参照してください。

ESCのリカバリおよび再展開のワークフローはポリシー駆動型です。VNFが展開されると、リカバリおよび再展開のポリシーが展開データモデルで指定されます。これらのポリシーは、VMまたはVNFのライフサイクルステージに基づいており、アクションが関連付けられています。

展開データモデルの作成時に、次のポリシーを指定できます。

- **リカバリポリシー**：リカバリポリシーは、VMライフサイクル、つまり単一のVMのリカバリ用です。事前定義されたアクションに基づいて、VMが再起動または再展開されます。ユーザは、ポリシーフレームワークを使用せずにリカバリを実行できます。[リカバリポリシー \(3 ページ\)](#)を参照してください。
- **再展開ポリシー**：再展開ポリシーは、展開ライフサイクル全体、つまり展開内のすべてのVMグループに適用されます。事前定義された一連のアクションに基づいて、ホストが無効になり、VMが展開内で回復されます。

最大試行回数の後にVMリカバリが失敗すると、ESCはホストを無効にし、展開内のすべてのVMの再展開をトリガーします。すべてのVMが古いホストから展開解除され、新しいホストに再展開されます。

ESCは、最初に障害が発生したVMの再展開をサポートします。再展開中は、障害が発生したVMが最初に回復され、障害が発生していないVMは再展開のためにキューに入れられます。

リカバリポリシー（ポリシーフレームワークを使用）

ESCはポリシー主導型フレームワークのデータモデルを使用したVMのリカバリをサポートしています。リカバリは、VM展開のライフサイクルステージと事前定義されたアクションに基づいています。

自動回復および手動回復については、[リカバリポリシー（3 ページ）](#) を参照してください。

次の表に、さまざまなライフサイクルステージで実行される事前定義されたアクションを示します。

事前定義されたアクション名	範囲	説明
SET_RECOVERY::REBOOT_ONLY	展開	すべての VM グループ（展開内）または VM（VM グループ内）のリカバリアクションを REBOOT_ONLY に設定します。
SET_RECOVERY::REBOOT_THEN_REDEPLOY	展開	すべての VM グループ（展開内）または VM（VM グループ内）のリカバリアクションを REBOOT_THEN_REDEPLOY に設定します。
SET_RECOVERY::REDEPLOY_ONLY	展開	すべての VM グループ（展開内）または VM（VM グループ内）のリカバリアクションを REDEPLOY_ONLY に設定します。
SET_RECOVERY::RESET_STATE_THEN_REBOOT	展開	すべての VM グループ（展開内）または VM（VM グループ内）のリカバリアクションを RESET_STATE_THEN_REBOOT に設定します（OpenStack のみ）。

サポートされる条件と事前定義されたアクションの組み合わせ

次の表に、ポリシーフレームワークを使用したリカバリおよび再展開ポリシーでサポートされる LCS 条件とそのアクションを示します。ポリシー主導型フレームワークの詳細については、[リカバリポリシーと再展開ポリシー（10 ページ）](#) を参照してください。

条件	事前定義されたアクション	説明
<p>LCS::PRE_DEPLOY</p> <p>: 展開で VM を展開する直前に発生します。</p> <p>LCS::POST_DEPLOY_ALIVE</p> <p>: 展開がアクティブになった直後に発生します。</p>	<ul style="list-style-type: none"> • SET_RECOVERY::REBOOT_ONLY <ul style="list-style-type: none"> : すべての VM グループ（展開内）または VM（VM グループ内）のリカバリアクションを REBOOT_ONLY に設定します。 • SET_RECOVERY::REBOOT_THEN_REDEPLOY <ul style="list-style-type: none"> : すべての VM グループ（展開内）または VM（VM グループ内）のリカバリアクションを REBOOT_THEN_REDEPLOY に設定します。 • SET_RECOVERY::REDEPLOY_ONLY <ul style="list-style-type: none"> : すべての VM グループ（展開内）または VM（VM グループ内）のリカバリアクションを REDEPLOY_ONLY に設定します。 • SET_RECOVERY_REDEPLOY::SERIALIZED <ul style="list-style-type: none"> : 展開のリカバリをキューに入れます。つまり、現在進行中のリカバリが完了するまで、新しいリカバリは開始されません。 	<p>リカバリ用に事前定義されたアクションのいずれかを選択します。</p> <p>SET_RECOVERY_REDEPLOY::SERIALIZED は、</p> <p>DROP_RECOVERIES アクションが使用される場合に選択します。これは、再展開が失敗した場合、元のホストに VM を保持する必要があることを意味します。選択しない場合、DROP_RECOVERIES アクションは使用できません。</p>
<p>LCS::DEPLOY_ERR</p> <p>: 展開が失敗した直後に発生します。</p>	<p>DISABLE_HOST</p> <p>: 展開または VM が使用しているホストを無効にします。</p>	-

条件	事前定義されたアクション	説明
LCS::POST_ DEPLOY::VM_RECOVERY _ERR : 1つのVMのリカバリが失敗した直後に発生します	DISABLE_HOST : 展開または VM が使用しているホストを無効にします。 REDEPLOY_ALL:: DISABLE_HOST : VM が使用しているホストを無効にしてから、(展開内の)すべてのVMまたはそのホスト上のすべてのVMの再展開をトリガーします。	必要に応じて、DISABLE_HOSTを選択します。 REDEPLOY_ALL:: DISABLE_HOST ホストの無効化後に再展開が必要かどうかを選択します。 DISABLE_HOST と REDEPLOY_ALL:DISABLE_HOST は重複するため、一緒にすることはできません。
LCS::POST_ DEPLOY::VM_RECOVERY_ REDEPLOY_ERR : 1つのVMの再展開が失敗した直後に発生します。	<ul style="list-style-type: none"> • DISABLE_HOST : 展開または VM が使用しているホストを無効にします。 • DROP_RECOVERIES : 展開内で保留中のすべてのリカバリをドロップします。 	DISABLE_HOST DISABLE_HOSTが必要かどうかを選択します。 再展開が失敗した後に VM を元のホストに保持する必要がある場合は、DROP_RECOVERIESを選択します。 DROP_RECOVERIESを選択する場合は、SET_RECOVERY_REDEPLOY::SERIALIZED アクションが完了していることを確認します。

再展開ポリシー

再展開ポリシーは、ポリシー駆動型フレームワークの一部です。このフレームワークを使用して、特定のライフサイクル条件用に事前定義されたアクションを指定できます。ESCポリシー駆動型フレームワークの詳細については、[ポリシー駆動型データモデル](#)を参照してください。

再展開ポリシーは、最大試行回数後にVMリカバリが失敗したときに呼び出されます。ESCはホストを無効にし、展開内のすべてのVMの再展開をトリガーします。すべてのVMが古いホストから展開解除され、新しいホストに再展開されます。ライフサイクルステージ (LCS) と事前定義されたアクションの組み合わせに基づいて、VMが再展開されます。再展開ポリシーは、展開全体に適用されます。

ポリシーデータモデルでは、次のライフサイクル条件とアクションの組み合わせを使用できません。



- (注) ESC は、何も選択されていない場合、デフォルトのリカバリアクション REBOOT_THEN_REDEPLOY を使用します。

再展開ポリシーのデータモデルの例を次に示します。

```
<tenants>
  <tenant>
    <name>xyz-redeploy-ten-0502</name>
    <deployments>
      <deployment>
        <name>dep</name>
        <policies>
          <policy>
            <name>1</name>
            <conditions>
              <condition>
                <name>LCS::PRE_DEPLOY</name>
              </condition>
            </conditions>
            <actions>
              <action>
                <name>SET_RECOVERY::REBOOT_THEN_REDEPLOY</name>
                <type>pre-defined</type>
              </action>
              <action>
                <name>SET_RECOVERY_REDEPLOY::SERIALIZED</name>
                <type>pre-defined</type>
              </action>
            </actions>
          </policy>
          <policy>
            <name>2</name>
            <conditions>
              <condition>
                <name>LCS::POST_DEPLOY_ALIVE</name>
              </condition>
            </conditions>
            <actions>
              <action>
                <name>SET_RECOVERY::REBOOT_ONLY</name>
                <type>pre-defined</type>
              </action>
            </actions>
          </policy>
          <policy>
            <name>3</name>
            <conditions>
              <condition>
                <name>LCS::DEPLOY_ERR</name>
              </condition>
            </conditions>
            <actions>
              <action>
                <name>DISABLE_HOST</name>
                <type>pre-defined</type>
              </action>
            </actions>
          </policy>
        </policies>
      </deployment>
    </deployments>
  </tenant>
</tenants>
```

```

        </actions>
    </policy>
    <policy>
        <name>4</name>
        <conditions>
            <condition>
                <name>LCS::POST_DEPLOY::VM_RECOVERY_ERR</name>
            </condition>
        </conditions>
        <actions>
            <action>
                <name>REDEPLOY_ALL::DISABLE_HOST</name>
                <type>pre-defined</type>
            </action>
        </actions>
    </policy>
    <policy>
        <name>5</name>
        <conditions>
            <condition>
                <name>LCS::POST_DEPLOY::VM_RECOVERY_REDEPLOY_ERR</name>
            </condition>
        </conditions>
        <actions>
            <action>
                <name>DISABLE_HOST</name>
                <type>pre-defined</type>
            </action>
            <action>
                <name>DROP_RECOVERIES</name>
                <type>pre-defined</type>
            </action>
        </actions>
    </policy>
</policies>
<vm_group>
    <name>Group1</name>
    <image>xyz-redeploy-img-0502</image>
    <flavor>xyz-redeploy-flv-0502</flavor>
    <recovery_policy>
        <max_retries>1</max_retries>
    </recovery_policy>
    .....
    .....
</deployment>
</deployments>
</tenant>
</tenants>

```

サポート対象のライフサイクルステージ (LCS)

条件名	範囲	説明
LCS::PRE_DEPLOY	展開	展開のVMを展開する直前に発生します。
LCS::POST_DEPLOY_ALIVE	展開	展開がアクティブになった直後に発生します。
LCS::DEPLOY_ERR	展開	展開が失敗した直後に発生します。

LCS::POST_DEPLOY:: VM_RECOVERY_ERR	展開	1つのVMのリカバリが失敗した直後に発生します (これは展開レベルで指定され、すべてのVMグループに適用されます)。
LCS::POST_DEPLOY:: VM_RECOVERY_REDEPLOY_ERR	展開	1つのVMの再展開が失敗した直後に発生します (これは展開レベルで指定され、すべてのVMグループに適用されます)。

サポートされている定義済みアクション

事前定義されたアクション名	範囲	説明
DISABLE_HOST	展開	展開またはVMが使用しているホストを無効にします。
REDEPLOY_ALL::DISABLE_HOST	展開	VMが使用しているホストを無効にしてから、(展開内の)すべてのVMまたはそのホスト上のすべてのVMの再展開をトリガーします。
DROP_RECOVERIES	展開	展開内で保留中のすべてのリカバリをドロップします。
SET_RECOVERY_REDEPLOY::SERIALIZED	展開	展開のリカバリをキューに入れます。つまり、現在進行中のリカバリが完了するまで、新しいリカバリは開始されません。

再展開回数の制限

Cisco Elastic Services Controller (ESC) は、次のパラメータを使用して再展開の回数を制限します。

- `max_redep` : 再展開の最大数を制限します。デフォルトでは、`max_redep` の値は -1 です。これは再展開の最大数に制限がないことを示します。この値は、`bootvm.py` 引数または REST API を使用して変更できます。
- `redep_count` : 現在の再展開の数で構成されます。`redep_count` は、再展開の成功または失敗に関係なく、再展開後に 1 ずつ自動的に増加します。



(注) 再展開の制限は次のとおりです。

- REDEPLOY_ALL::DISABLE_HOST ポリシーによってトリガーされる再展開。
- 単一の VIM 設定のみの展開。

次の場合、Cisco Elastic Services Controller (ESC) が再展開を実行します。

- 再展開の最大数がデフォルト値の -1 に設定されている場合 (`max_redep = -1`) 。
- 現在の再展開の数が再展開の最大数よりも少ない場合 (`redep_count < max_redep`)、ESC が再展開を実行し、再展開の完了後に再展開数を 1 増やします。

再展開の回数が再展開の最大数以上の場合 (`redep_count >= max_redep`)、ESC は再展開を実行しません。

各値は、`bootvm.py` パラメータと REST API を使用して設定できます。

bootvm.py パラメータの使用

次の行を含む `esc_params.conf` ファイルで `max_redep` 値を指定します。 `default.max_redep = 3`

コマンド `bootvm.py ... --esc_params_file <path_to_file>/esc_params.conf ...` を実行します。

REST API の使用

次の API を使用して、`redep_count` パラメータを取得およびリセットできます。

- `redep_count` の現在の値を取得するには、次の手順を実行します。
`GET http://<ESC IP>:8080/ESCManager/v0/systemstate/redep_count`
- `redep_count` をリセットするには、次の手順を実行します。
`POST http://<ESC IP>:8080/ESCManager/v0/systemstate/redep_count/reset`

REST API を使用して `max_redep` 値を取得および変更することもできます。

- `max_redep` の現在の値を取得するには、次の手順を実行します。
`GET http://<ESC IP>:8080/ESCManager/v0/config/default/max_redep`
- `max_redep` 値を変更するには、次の手順を実行します。
`PUT http://<ESC IP>:8080/ESCManager/v0/config/default/max_redep/<value>`

ここで、`<value>` は次のいずれかです。

-1 : 制限なしのデフォルト値。

0 : 再展開を許可しない場合。

1 以上 (> 0) : 許可される再展開の最大数を指定します。

これらの値は、ESCADM ツールを使用して設定することもできます。ESCADM ツールの詳細については、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。

再展開ポリシーの詳細については、[再展開ポリシー \(13 ページ\)](#) を参照してください。

再展開の制限により再展開されない VM は、エラー状態に移行します。ESC では、各 VM でモニタリング操作を有効にすることで、エラー状態にある VM を手動で回復します。

エラー状態にある単一の VM でモニタリング操作を有効にするには、次の手順を実行します。

```
POST http://<ESC IP>:8080/ESCManager/v0/<internal-tenant-id>/deployments/vm/<vm-name> {
  "operation" : "enable_monitoring" }
```

esc_nc_cli コマンドを使用してモニタリングを有効にすることもできます。

```
esc_nc_cli --user <username> --password <password> vm-action ENABLE_MONITOR <generated
vm name>
```

手動リカバリプロセスの一環として、モニタリング操作の有効化により VM がエラー状態から稼働状態に移行します。VM の手動リカバリが失敗した場合、自動リカバリがトリガーされません。

展開内の VM（エラー状態）のモニタリング操作を有効にするには、次の手順を実行します。

```
POST http://<ESC
IP>:8080/ESCManager/v0/<internal-tenant-id>/deployments/service/<internal-deployment-id>
{ "operation" : "enable_monitoring" }
```

esc_nc_cli コマンドを使用してモニタリングを有効にすることもできます。

```
esc_nc_cli --user <username> --password <password> svc-action ENABLE_MONITOR <tenant>
<dep name>
```

手動リカバリプロセスの一環として、モニタリング操作の有効化により展開内のすべての VM がエラー状態から稼働状態に移行します。手動リカバリが失敗した場合、展開内のすべての VM に対して自動リカバリがトリガーされます。

詳細については、[モニタリング操作](#)、および「[リカバリポリシー](#)」を参照してください。

ホストの有効化と無効化

NETCONF API および REST API を使用して、OpenStack でホストを有効または無効にできます。ホストは、VNF のリカバリまたは再展開のシナリオ中に無効にすることもできます。



(注) VMware vCenter でのホストの有効化と無効化はサポートされていません。

複数の OpenStack VIM がある ESC で NETCONF API および REST API を使用して、デフォルト以外の VIM でホストを有効または無効にすることはできません。

NETCONF の使用

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli --user <username> --password <password>
host-action < ENABLE | DISABLE > <host-name>
```

ペイロードは次のとおりです。

```
<hostAction xmlns="http://www.cisco.com/esc/esc">
  <actionType>ENABLE/DISABLE</actionType>
  <hostName>my-server</hostName>
</hostAction>
```

値は次のとおりです。

- actionType は ENABLE または DISABLE です

- `hostName` はターゲットホストのホスト名または UUID です

REST の使用

```
POST /v0/hosts/{hostName}/disable
POST /v0/hosts/{hostName}/enable
GET /v0/hosts/{hostName}/status
```

ホストの有効化

ホストを有効にすることで、無効化されたホストを OpenStack に戻し、新しい VM インスタンスをそのホストに展開します。

NETCONF 通知の例は次のとおりです。

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-03-30T15:04:05.95+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Host action successful</status_message>
    <vm_source>
      <hostname>my-server</hostname>
    </vm_source>
    <vm_target>
</vm_target>
    <event>
      <type>HOST_ENABLE</type>
    </event>
  </escEvent>
</notification>
```

サンプル REST 通知は次のとおりです。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <host_action_event xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <event_type>HOST_ENABLE</event_type>
    <host_name>my-server</host_name>
    <message>Host action successful</message>
  </host_action_event>
```

ホストの無効化

VNF の再展開中にホストを無効にし、その展開内のすべての VM に対してホストベースの再展開をトリガーします。これにより、再展開された VM が別のホストにあることが保証されます。ホストが正常に動作していない場合は、ホストを無効にすることもできます。無効になったホストは OpenStack から削除されるため、新しいインスタンスは展開されません。

NETCONF 通知の例は次のとおりです。

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-03-30T15:03:48.121+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Host action successful</status_message>
    <vm_source>
      <hostname>my-server</hostname>
    </vm_source>
    <vm_target>
</vm_target>
  </escEvent>
</notification>
```

```
<event>
  <type>HOST_DISABLE</type>
</event>
</escEvent>
</notification>
```

サンプル REST 通知は次のとおりです。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<host_action_event xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <event_type>HOST_DISABLE</event_type>
  <host_name>my-server</host_name>
  <message>Host action successful</message>
</host_action_event>
```

通知とイベント

修復中に ESC によって次の通知が生成されます。

- VM_RECOVERY_INIT
- VM_RECOVERY_DEPLOYED
- VM_RECOVERY_UNDEPLOYED
- VM_RECOVERY_COMPLETE
- VM_RECOVERY_CANCELLED
- VM_RECOVERY_REBOOT

これらの通知は、ワークフローに基づいて生成されます。各通知には、通知がトリガーされる展開に関する詳細情報が含まれます。すべてのリカバリは VM_RECOVERY_INIT で始まり、VM_RECOVERY_COMPLETE で終わります。

VM のリカバリ中、リカバリ待機時間内に VM が正常に戻ると、実行するリカバリアクションがないため、VM_RECOVERY_CANCELLED 通知が送信されます。リカバリ待機時間が経過すると、リカバリアクションがトリガーされます。リカバリが完了すると、ESC は成功または失敗の通知 (VM_RECOVERY_REBOOT 通知など) を送信します。

次の表に、さまざまなシナリオと、イベントごとに生成される通知を示します。

シナリオ	通知
VM Alive 後の ESC-NORTHBOUND リカバ リコールフロー：再起動	<p>ノースバウンドから ESC に展開要求が送信されると、ESC は VM を展開し、受信したすべての VM Alive をモニタするように KPI を設定します。次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="781 485 1114 531"><type>SERVICE_ALIVE</type> <status>SUCCESS</status></pre> <p>ESC が VM Down イベントを受信すると、次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="781 663 1151 709"><type>VM_RECOVERY_INIT</type> <status>SUCCESS</status></pre> <p>ESC は VM でハード再起動を実行し、VM Alive イベントをブート時間内に受信します。</p> <pre data-bbox="781 842 1203 888"><type>VM_RECOVERY_COMPLETE</type> <status>SUCCESS</status></pre> <p>ESC は、再起動によるリカバリの試行中にエラーを受信します。次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="781 999 1203 1066"><type>VM_RECOVERY_COMPLETE</type> <status>FAILURE</status></pre>

シナリオ	通知
VM Alive 後の ESC-NORTHBOUND リカバリ コールフロー：展開解除/ 再展開	<p>ノースバウンドから ESC に展開要求が送信されると、ESC は VM を展開し、受信したすべての VM Alive をモニタするように KPI を設定します。次の NETCONF 通知がトリガーされません。</p> <pre data-bbox="743 485 1073 531"><type>SERVICE_ALIVE</type> <status>SUCCESS</status></pre> <p>ESC が VM Down イベントを受信すると、次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="743 663 1110 709"><type>VM_RECOVERY_INIT</type> <status>SUCCESS</status></pre> <p>ESC は再起動による VM の回復に失敗し、展開解除して再展開することで回復を進めます。</p> <p>モニタリングの設定を解除し、VM の展開を解除します。</p> <p>次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="743 951 1187 997"><type>VM_RECOVERY_UNDEPLOYED</type> <status>SUCCESS</status></pre> <p>ESC は VM を展開し、VM Alive イベントをモニタするように KPI を設定し、次の NETCONF 通知をトリガーします。</p> <pre data-bbox="743 1129 1162 1176"><type>VM_RECOVERY_DEPLOYED</type> <status>SUCCESS</status></pre> <p>ESC は VM Alive イベントを受信し、次の NETCONF 通知をトリガーします。</p> <pre data-bbox="743 1308 1162 1354"><type>VM_RECOVERY_COMPLETE</type> <status>SUCCESS</status></pre>

シナリオ	通知
<p>ESC-NORTHBOUND リカバリコールフローによる複数回のリカバリの試行</p>	<p>ノースバウンドから ESC に展開要求が送信されると、ESC は VM を展開し、受信したすべての VM Alive をモニタするように KPI を設定します。次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="781 485 1114 531"><type>SERVICE_ALIVE</type> <status>SUCCESS</status></pre> <p>ESC が VM Down イベントを受信すると、次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="781 663 1149 709"><type>VM_RECOVERY_INIT</type> <status>SUCCESS</status></pre> <p>ESC は、VM Alive イベントを受信するまで、展開解除と再展開によって VM を回復できません。リカバリの最大試行回数に達するまで、指定されたブート時間リカバリを試行し続けます。</p> <p>モニタリングの設定を解除し、VM の展開を解除します。</p> <p>次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="781 1024 1227 1071"><type>VM_RECOVERY_UNDEPLOYED</type> <status>SUCCESS</status></pre> <p>ESC は VM を展開し、VM Alive イベントをモニタするように KPI を設定します。</p> <p>次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="781 1255 1203 1302"><type>VM_RECOVERY_DEPLOYED</type> <status>SUCCESS</status></pre> <p>ESC は VM Alive イベントを受信し、次の NETCONF 通知をトリガーします。</p> <pre data-bbox="781 1434 1203 1480"><type>VM_RECOVERY_COMPLETE</type> <status>SUCCESS</status></pre>

シナリオ	通知
VM Alive 前の ESC-NORTHBOUND リカバリ リコールフロー：展開解除/ 再展開	<p>ノースバウンドから ESC に展開要求が送信されると、ESC は VM を展開し、受信したすべての VM Alive をモニタするように KPI を設定します。</p> <p>ESC は、展開後に VM Alive イベントを受信しません。リカバリは、VM の展開と再展開によって実行されます。</p> <p>次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="743 583 1112 636"><type>VM_RECOVERY_INIT</type> <status>SUCCESS</status></pre> <p>ESC はモニタリングの設定を解除し、VM の展開を解除します。</p> <p>次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="743 821 1188 873"><type>VM_RECOVERY_UNDEPLOYED</type> <status>SUCCESS</status></pre> <p>ESC は VM を展開し、VM Alive イベントをモニタするように KPI を設定し、次の NETCONF 通知をトリガーします。</p> <pre data-bbox="743 999 1162 1052"><type>VM_RECOVERY_DEPLOYED</type> <status>SUCCESS</status></pre> <p>ESC は VM Alive イベントを受信し、次の NETCONF 通知をトリガーします。</p> <pre data-bbox="743 1178 1162 1230"><type>VM_RECOVERY_COMPLETE</type> <status>SUCCESS</status></pre>

シナリオ	通知
VM Alive 後の ESC-NORTHBOUND リカバリ コールフローのエラーパ ス：展開解除/再展開	<p>ノースバウンドから ESC に展開要求が送信されると、ESC は VM を展開し、受信したすべての VM Alive をモニタするように KPI を設定します。次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="781 485 1114 531"><type>SERVICE_ALIVE</type> <status>SUCCESS</status></pre> <p>ESC が VM Down イベントを受信すると、次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="781 663 1151 709"><type>VM_RECOVERY_INIT</type> <status>SUCCESS</status></pre> <p>ESC は再起動による VM の回復に失敗し、展開解除して再展開することで回復を進めます。</p> <p>モニタリングの設定を解除し、VM の展開を解除します。</p> <p>次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="781 951 1227 997"><type>VM_RECOVERY_UNDEPLOYED</type> <status>SUCCESS</status></pre> <p>ESC がエラーを受信した場合、またはリカバリの最大試行回数に達した場合。</p> <p>次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="781 1182 1203 1228"><type>VM_RECOVERY_COMPLETE</type> <status>FAILURE</status></pre>

シナリオ	通知
<p>VM Alive 前の ESC-NORTHBOUND リカバリ コールフローのエラーパ ス：展開解除/再展開</p>	<p>ノースバウンドから ESC に展開要求が送信されると、ESC は VM を展開し、受信したすべての VM Alive をモニタするように KPI を設定します。次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="743 485 1073 531"><type>SERVICE_ALIVE</type> <status>SUCCESS</status></pre> <p>ESC が VM Down イベントを受信すると、次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="743 663 1110 709"><type>VM_RECOVERY_INIT</type> <status>SUCCESS</status></pre> <p>ESC はモニタリングの設定を解除し、VM の展開を解除します。リカバリは、展開解除してから再展開することで実行されます。</p> <p>次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="743 932 1187 978"><type>VM_RECOVERY_UNDEPLOYED</type> <status>SUCCESS</status></pre> <p>ESC がエラーを受信した場合、またはリカバリの最大試行回数に達した場合。</p> <p>次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="743 1163 1162 1209"><type>VM_RECOVERY_COMPLETE</type> <status>FAILURE</status></pre> <pre data-bbox="743 1257 1073 1304"><type>SERVICE_ALIVE</type> <status>FAILURE</status></pre>

シナリオ	通知
VM Alive 後の ESC-NORTHBOUND リカバリ リコールフロー： VM_RECOVERY_CANCELLED	<p>ノースバウンドから ESC に展開要求が送信されると、ESC は VM を展開し、受信したすべての VM Alive 通知をモニタするように KPI を設定します。次の NETCONF 通知がトリガーされます。</p> <pre><type>SERVICE_ALIVE</type> <status>SUCCESS</status></pre> <p>ESC が VM Down イベントを受信すると、次の NETCONF 通知がトリガーされます。</p> <pre><type>VM_RECOVERY_INIT</type> <status>SUCCESS</status></pre> <p>リカバリ待機時間中に VM が正常に戻ると、VM_RECOVERY_CANCELLED 通知が送信されます。リカバリアクションは実行されません。</p> <pre><type>VM_RECOVERY_CANCELLED</type> <status>SUCCESS</status></pre>
VM Alive 後の ESC-NORTHBOUND リカバリ リコールフロー：再起動	<p>ノースバウンドから ESC に展開要求が送信されると、ESC は VM を展開し、受信したすべての VM Alive 通知をモニタするように KPI を設定します。次の NETCONF 通知がトリガーされます。</p> <pre><type>SERVICE_ALIVE</type> <status>SUCCESS</status></pre> <p>ESC が VM Down イベントを受信すると、次の NETCONF 通知がトリガーされます。</p> <pre><type>VM_RECOVERY_INIT</type> <status>SUCCESS</status></pre> <p>ESC は VM でハード再起動を実行し、再起動通知を送信します。</p> <pre><type>VM_RECOVERY_REBOOT</type> <status>SUCCESS</status></pre> <p>VM Alive イベントは、ブート時間内に受信されます。</p> <pre><type>VM_RECOVERY_COMPLETE</type> <status>SUCCESS</status></pre>

シナリオ	通知
VM Alive 後の ESC-NORTHBOUND リカバリ リコールフローのエラーパ ス：再起動	<p>ノースバウンドから ESC に展開要求が送信されると、ESC は VM を展開し、受信したすべての VM Alive 通知をモニタするように KPI を設定します。次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="743 457 1073 506"><type>SERVICE_ALIVE</type> <status>SUCCESS</status></pre> <p>ESC が VM Down イベントを受信すると、次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="743 615 1110 663"><type>VM_RECOVERY_INIT</type> <status>SUCCESS</status></pre> <p>次に、ESC が再起動通知を送信します。</p> <pre data-bbox="743 730 1138 779"><type>VM_RECOVERY_REBOOT</type> <status>FAILURE</status></pre> <p>ESC は、再起動によるリカバリの試行中にエラーを受信しません。</p> <p>次の NETCONF 通知がトリガーされます。</p> <pre data-bbox="743 940 1162 989"><type>VM_RECOVERY_COMPLETE</type> <status>FAILURE</status></pre>

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。