



# 仮想ネットワーク機能のモニタリング

- [VNF のモニタリング \(1 ページ\)](#)
- [モニタリング方式 \(8 ページ\)](#)
- [VM のモニタリング \(9 ページ\)](#)
- [VM モニタリングステータスの通知 \(11 ページ\)](#)
- [モニタリング操作 \(12 ページ\)](#)

## VNF のモニタリング

VNFは展開後、正常性とワークロードを確認するために定期的にモニタされます。モニタリングは、展開データモデルのKPIセクション内のメトリックの定義に基づいています。KPIセクションで説明されているように、メトリックタイプによって、モニタする変数だけでなく、実行するコレクタアクションも決まります。ESCでは、モニタ対象のメトリックと、条件を満たしたときに実行する必要があるアクションを定義できます。それらのメトリックとアクションは、展開データモデルで定義されます。VNFをモニタするために、複数のモニタリング方法が使用されます。次の内容をモニタできます。

- VM の稼働状態
- ディスク使用率、メモリ、CPU、ネットワークスループットの VM 変数
- VM モニタリング インターフェイスの ICMP メッセージ

### モニタリングの前提条件

ESCでVMをモニタするには、次の前提条件を満たしている必要があります。

- 正常に展開されたVMのモニタリングが有効になっている。展開されたVMは稼働している必要があります。
- KPIは、モニタリングパラメータを使用してデータモデル内で設定されている必要がある。

## モニタリングおよびアクション実行エンジン

モニタリングは、展開データモデルの KPI セクション内のメトリックの定義に基づいています。KPI セクションで説明されているように、メトリックタイプによって、モニタする変数だけでなく、実行するコレクタアクションも決まります。モニタリングエンジンは、メトリックとアクションで構成されます。

1. メトリック
2. アクション

メトリックとアクション<metadata>セクションでは、エンジンのプログラム可能な側面を制御するプロパティまたはエントリについて記述します。

### メトリックセクション

メトリックセクションは次のとおりです。

```
<metrics>
  <metric>
    <name>{metric name}name>
      <type>{metric type}type>
      <metaData>
        <type>{monitoring engine action type}</type>
      <properties>
        <property>
          <name></name>
          <value></value>
        </property>
        : : : : :
      </properties/>
    </metaData>
  </metric or action>
  : : : : :
</metrics>
```

表 1: メトリックセクションの説明

タグ名	説明	値
名前	ユーザ定義のメトリック名。 メトリック名は一意である必要があります。	
タイプ	ダイナミックマッピングがサポートされるタイプ。	MONITOR_SUCCESS_FAILURE MONITOR_THRESHOLD MONITOR_COMPUTE_THRESHOLD

### メトリック メタデータ セクション

メタデータセクションの目的は、モニタリングソリューションに固有の情報を提供することです。

表 2: メトリック メタデータ セクション

タグ名	説明	値
タイプ	アクションタイプ、値は、MONA でサポートされるアクションと 1 対 1 でマッピングされます。	custom_script custom_script_threshold snmp_get_threshold
プロパティ	選択したアクションに渡されるプロパティ (名前/値) のリストのコンテナ。プロパティは、予期されるモニタリングおよびアクション属性のリストによって定義されます。	プロパティは、選択したアクションタイプに基づきます。

### アクションセクション

アクションセクションは次のとおりです。

```
<actions>
  <action>
    <name>{action name}name>
    <type>{action type}type>
    <metaData>
      <type>{monitoring engine action type}</type>
      <properties>
        <property>
          <name></name>
          <value></value>
        </property>
        : : : : :
      </properties>
    </metaData>
  </action>
  : : : : :
</actions>
```

表 3: アクション

タグ名	説明	値
名前	ユーザ定義のアクション名。アクション名は一意である必要があります。	主な要件の 1 つは、選択された名前の先頭に TRUE または FALSE を付けて、MONITOR_SUCCESS_FAILURE のためだけに、ESC データモデルルールと動的アクション間のマッピングを許可することです。

タグ名	説明	値
タイプ	サポート対象タイプ。	ESC_POST_EVENT スクリプト CUSTOM_SCRIPT

#### アクションメタデータ セクション

メタデータセクションの目的は、モニタリングソリューションに固有の情報を提供することです。

表 4: アクションメタデータ セクション

タグ名	説明	値
タイプ	アクションタイプ、値は、モニタリングおよびアクションエンジンでサポートされるアクションと1対1でマッピングされます。	icmp_ping icmp4_ping icmp6_ping esc_post_event スクリプト custom_script snmp_get snmp_get_threshold
プロパティ	選択したアクションに渡されるプロパティ（名前/値）のリストのコンテナ。プロパティは、予期されるモニタリングおよびアクション属性のリストによって定義されます。	プロパティは、選択したアクションタイプに基づきます。

詳細については、「KPI、ルール、およびダイナミックマッピング API」セクションを参照してください。

表 5: サポートされているアクションタイプ

タイプ	プロパティと各プロパティの説明
icmp_ping	<ul style="list-style-type: none"> <li>• ip_address</li> <li>• enable_events_after_success : MONA がイベント通知の転送を開始するタイミングを制御するブール値。true に設定すると、初めて成功に移行した後にのみ通知が転送されます。</li> <li>• timeOut : デフォルトで 5 秒に設定</li> </ul>
icmpv4_ping	<ul style="list-style-type: none"> <li>• ip_address</li> <li>• enable_events_after_success : MONA がイベント通知の転送を開始するタイミングを制御するブール値。true に設定すると、初めて成功に移行した後にのみ通知が転送されます。</li> <li>• timeOut : デフォルトで 5 秒に設定</li> </ul>
icmpv6_ping	<ul style="list-style-type: none"> <li>• ip_address</li> <li>• enable_events_after_success : MONA がイベント通知の転送を開始するタイミングを制御するブール値。true に設定すると、初めて成功に移行した後にのみ通知が転送されます。</li> <li>• timeOut : デフォルトで 5 秒に設定</li> </ul>
スクリプト	<ul style="list-style-type: none"> <li>• script_filename : 実行するスクリプトへのフルパス (スクリプトは ESC VM に配置する必要があります)。</li> <li>• wait_for_script : アクションがスクリプトの完了を待機しているかどうかを制御するブール値 (実際には実行されません)。</li> </ul>
custom_script	script_filename : 実行するスクリプトへのフルパス (スクリプトは ESC Manager VM に配置する必要があります)。

タイプ	プロパティと各プロパティの説明
custom_script_threshold	<ul style="list-style-type: none"> <li>• script_filename : 実行するスクリプトへのフルパス (スクリプトは ESC Manager VM に配置する必要があります)。</li> <li>• しきい値</li> </ul>
post_esc_event	<ul style="list-style-type: none"> <li>• esc_url</li> <li>• vm_external_id</li> <li>• vm_name</li> <li>• esc_event</li> <li>• event_name</li> </ul>
snmp_get	<ul style="list-style-type: none"> <li>• target_oid、agent_address、SNMP エージェントの IP アドレス (IPV4/IPV6 がサポートされます)。</li> <li>• agent_port : SNMP エージェントで使用されるポート。</li> <li>• agent_protocol : SNMP エージェントで使用されるプロトコル (tcp/udp)。</li> <li>• Community : SNMP エージェントで使用される SNMPv2c コミュニティストリング。</li> </ul>
snmp_get_threshold	<ul style="list-style-type: none"> <li>• target_oid : しきい値の比較に使用されるオブジェクト識別子。</li> <li>• agent_address : SNMP エージェントの IP アドレス (IPV4/IPV6 がサポートされます)。</li> <li>• agent_port : SNMP エージェントで使用されるポート。</li> <li>• agent_protocol : SNMP エージェントで使用されるプロトコル (tcp/udp)。</li> <li>• community : SNMP エージェントで使用される SNMPv2c コミュニティストリング。</li> </ul>

タイプ	プロパティと各プロパティの説明
snmp_get_threshold_ratio	<ul style="list-style-type: none"> <li>• <b>oid_total_value</b> : 比率/パーセンテージ計算の経過を表すために使用されるオブジェクト識別子。</li> <li>• <b>oid_current_value</b> : 比率/パーセンテージ計算の経過を表すために使用されるオブジェクト識別子。パーセンテージ/比率の計算に使用されるアルゴリズム。現在、<b>COMPUTE_TOTAL_CURRENT_BASED</b> と <b>COMPUTE_TOTAL_AVAILABILITY_BASED</b> の 2 つのアルゴリズムがサポートされています。</li> <li>• <b>agent_address</b> : SNMP エージェントの IP アドレス (IPV4/IPV6 がサポートされます)。</li> <li>• <b>agent_port</b> : SNMP エージェントで使用されるポート。</li> <li>• <b>agent_protocol</b> : SNMP エージェントで使用されるプロトコル (tcp/udp)。</li> <li>• <b>community</b> : SNMP エージェントで使用される SNMP v2c コミュニティストリング。</li> </ul>

### プロパティとランタイムパラメータ インジェクション

選択したアクションタイプに渡されるプロパティリストは、選択した一部のパラメータのランタイム値を自動的に挿入する機能をサポートします。たとえば、仮想マシン `ip_address` のランタイム値や仮想マシンの名前を、選択したアクションに引数として自動的に渡すことができます。

次に、実行時にスクリプトに渡すことができるパラメータの一部を示します。パラメータ値は、次の場合にのみ実行時に設定されます。

- パラメータがサポート対象のパラメータである。
- パラメータの値が、`dynamic-mappings.xml` ファイル内で空である。

それ以外の場合、スクリプト内で定義された値がそのまま渡されます。

次の表に、実行時に渡されるパラメータを示します。

esc_url	Elastic Services Controller の URL。
vm_external_id	管理対象 VM の外部 ID。

vm_name	管理対象 VM の名前。
vm_mac_address	管理対象 VM の MAC アドレス。
vm_external_host_id	VM 外部ホスト識別子。
vm_external_host_name	VM 外部ホスト名。
vm_group_name	VM グループ名。
ip_address	VM IP アドレス。
event_name	ESC イベント名。



(注) 選択したアクションに渡されるプロパティリストは、アクションタイプのパラメータによってバインドされていません。スクリプト設計者は、独自のパラメータを定義できます。ただし、値を指定する必要があります。

## モニタリング方式

ESC は、いくつかのモニタリング方法を使用して VNF をモニタします。モニタリング方式の KPI データモデルを設定する必要があります。

### ICMP ping モニタリング

ping モニタリングは、VNF の動作状態または到達可能性を評価します。

VM が到達不能の場合、VM の修復がトリガーされます。定義された間隔ごとに、ESC はメトリック値をポーリングし、必要に応じてアラームを送信します。ポーリング数、メトリック値、およびその他の設定は、KPI データモデルで設定されます。

### SNMP モニタリング

SNMP モニタリングでは、特定の期間におけるメモリ使用率や CPU などの VM の負荷がモニタされます。SNMP Get 操作は、VNF の動作状態または到達可能性を評価するために使用されます。このモニタリング方式では、成功または失敗のみがモニタされます。

### SNMP しきい値モニタリング

SNMP しきい値モニタリングでは、データモデルの KPI セクションで上限および下限しきい値レベルを設定できます。アクションは、しきい値の上限と下限に基づいて実行されます。

### カスタムモニタリング

ESC2.1 以前では、データモデルで定義されたアクションとメトリックをモニタリングエージェントで使用可能な有効なアクションとメトリックにマッピングするために、ダイナミックマッ



ピング XML が必要です。ファイルは ESC VM に保存され、テキストエディタを使用して変更されます。この方法はエラーが発生しやすく、アクティブ VM とスタンバイ VM の両方で HA ペアを変更する必要があります。ESC 2.2 以降には、`esc-dynamic-mapping` ディレクトリと `dynamic_mappings.xml` ファイルはありません。アクションとメトリックをマッピングするための CRUD 操作が、ESC の REST API を介して実行できるようになりました。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。

## VM のモニタリング

Cisco Elastic Services Controller は、VM をモニタしてエラー状態を検出します。ESC はそのモニタリング方式の 1 つを使用して VM のアクションを検出し、この情報をルールサービスに渡して処理します。モニタリング要求は、VNF 展開要求とともにノースバウンドクライアントから送信されます。

データモデル xml ファイルには、イベントとルール (KPI とルール) を定義する 2 つのセクションがあります。

モニタとアクションに基づいて、ルールがトリガーされます。

```
<kpi>
  <event_name>VM_ALIVE</event_name>
  <metric_value>50</metric_value>
  <metric_cond>GT</metric_cond>
  <metric_type>UIN32</metric_type>
  <metric_occurrences_true>3</metric_occurrences_true>
  <metric_occurrences_false>3</metric_occurrences_false>
  <metric_collector>
    <type>ICMPPing</type>
    <nicid>0</nicid>
    <poll_frequency>15</poll_frequency>
    <polling_unit>seconds</polling_unit>
    <continuous_alarm>>false</continuous_alarm>
  </metric_collector>
</kpi>
```

上記の例では、VM が動作しているかどうかを確認するイベントが送信されます。VM は定期的に ping され、その結果に基づいて VM\_ALIVE イベントが VM の詳細とともにルールエンジンに送信されます。

ルールエンジンは、モニタリングエンジンからイベントを受信します。ルールエンジンは、単純なイベントから複雑なイベントまで処理できます。受信したイベントに基づいて、アクションがトリガーされます。

VM が動作していない場合、イベントに基づいて、`<rule>` セクションで定義されたアクションがトリガーされます。これは `dep.xml` データモデルで確認できます。

```
<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>
```

```

    </rule>
  </admin_rules>
</rules>

```

ルールセクションでは、モニタリングイベントが検出されたときに実行されるアクションについて説明します。ダイナミックマッピング API は、キーワードに基づいてルールを駆動します。

上記の例では、指定された条件に基づいて次のアクションが実行されます。

- **ALWAYS log** : イベントが ping 可能かどうかにかかわらず、詳細がログに記録されます。
- **TRUE servicebooted.sh** : ダイナミックマッピング API でこのキーワードによって識別されるアクションは、VM が ping 不能状態から ping 可能状態に移行したときにトリガーされます。serviceboot スクリプトは、VM が動作していることを ESC に通知し、VM の状態を移行できるようにします。
- **FALSE recover autohealing** : このキーワードによって識別されるアクションがトリガーされ、管理者の介入なしで VM が回復されます。

トラブルシューティング用のモニタリングログファイルは、/var/log/mona にあります。

### VM ネットワークステータスのモニタリング

ICMP ping モニタリングを使用する場合、ESC が VM ダウンイベントを受信すると、修復ワークフローは回復ポリシーを使用して VM の回復を試みます。ESC から VNF へのネットワークインターフェイスまたは IP ルートに問題がある場合。たとえば、ゲートウェイがダウンしている場合は、VM ダウンイベントが誤ってトリガーされ、不要なリカバリが発生する可能性があります。

インターフェイスチェック機能は、すべてのネットワークインターフェイスのヘルスステータスとゲートウェイの動作状態をチェックすることによって、ネットワークルートをさらにスキャンします。ネットワーク環境に問題がある場合は、VNF が動作していると見なされます。

ESC がネットワークの問題を検出した場合、または既存の問題が修正された場合（自動修復）、VM\_NETWORK\_STATE イベントがノースバウンドに送信されます。

次の障害通知がノースバウンドに送信されます。

```

16:13:15,567 14-Mar-2018 WARN ===== SEND NOTIFICATION STARTS =====
16:13:15,567 14-Mar-2018 WARN Type: VM_NETWORK_STATE
16:13:15,567 14-Mar-2018 WARN Status: FAILURE
16:13:15,567 14-Mar-2018 WARN Status Code: 500
16:13:15,567 14-Mar-2018 WARN Status Msg: Warning: VM
[NG_G1_0_46fdcf70-f4ea-4289-ae79-08674e7d6f42] has a network problem: Network interface
not healthy, please check.
16:13:15,567 14-Mar-2018 WARN Tenant: tenant2
16:13:15,567 14-Mar-2018 WARN Deployment ID: 455d2407-9dda-4203-95b0-724c4a651720
16:13:15,567 14-Mar-2018 WARN Deployment name: NG
16:13:15,567 14-Mar-2018 WARN VM group name: G1
16:13:15,567 14-Mar-2018 WARN VM Source:
16:13:15,567 14-Mar-2018 WARN VM ID: 4bee016a-6b30-43ff-a249-157a07d9b4db
16:13:15,567 14-Mar-2018 WARN VM Name: NG_G1_0_46fdcf70-f4ea-4289-ae79-08674e7d6f42
16:13:15,568 14-Mar-2018 WARN VM Name (Generated):
NG_G1_0_46fdcf70-f4ea-4289-ae79-08674e7d6f42

```

```

16:13:15,568 14-Mar-2018 WARN      VIM ID: default_openstack_vim
16:13:15,568 14-Mar-2018 WARN      VIM Project: tenant2
16:13:15,568 14-Mar-2018 WARN      VIM Project ID: 62afb63cd28647a7b526123cac1ba605
16:13:15,568 14-Mar-2018 WARN      Host ID:
b83004159a46c20bc8383927c2231067bb0c1905b4b4c28475653190
16:13:15,568 14-Mar-2018 WARN      Host Name: my-server-50
16:13:15,568 14-Mar-2018 WARN      ===== SEND NOTIFICATION ENDS =====

```

ネットワークの問題が修正されると、次の成功通知がノースバウンドに送信されます。

```

16:13:19,141 14-Mar-2018 INFO      ===== SEND NOTIFICATION STARTS =====
16:13:19,141 14-Mar-2018 INFO      Type: VM_NETWORK_STATE
16:13:19,142 14-Mar-2018 INFO      Status: SUCCESS
16:13:19,142 14-Mar-2018 INFO      Status Code: 200
16:13:19,142 14-Mar-2018 INFO      Status Msg: Network of VM
[NG_G1_0_46fdcf70-f4ea-4289-ae79-08674e7d6f42] has been restored.
16:13:19,142 14-Mar-2018 INFO      Tenant: tenant2
16:13:19,142 14-Mar-2018 INFO      Deployment ID: 455d2407-9dda-4203-95b0-724c4a651720
16:13:19,142 14-Mar-2018 INFO      Deployment name: NG
16:13:19,142 14-Mar-2018 INFO      VM group name: G1
16:13:19,142 14-Mar-2018 INFO      VM Source:
16:13:19,142 14-Mar-2018 INFO      VM ID: 4bee016a-6b30-43ff-a249-157a07d9b4db
16:13:19,142 14-Mar-2018 INFO      VM Name: NG_G1_0_46fdcf70-f4ea-4289-ae79-08674e7d6f42
16:13:19,142 14-Mar-2018 INFO      VM Name (Generated):
NG_G1_0_46fdcf70-f4ea-4289-ae79-08674e7d6f42
16:13:19,142 14-Mar-2018 INFO      VIM ID: default_openstack_vim
16:13:19,143 14-Mar-2018 INFO      VIM Project: tenant2
16:13:19,143 14-Mar-2018 INFO      VIM Project ID: 62afb63cd28647a7b526123cac1ba605
16:13:19,143 14-Mar-2018 INFO      Host ID:
b83004159a46c20bc8383927c2231067bb0c1905b4b4c28475653190
16:13:19,143 14-Mar-2018 INFO      Host Name: my-server-50
16:13:19,143 14-Mar-2018 INFO      ===== SEND NOTIFICATION ENDS =====

```

ETSI API を使用した VNF のモニタリングについては、『Cisco Elastic Services Controller ETSI NFV MANO Guide』を参照してください。

## VM モニタリングステータスの通知

ESC は、次の条件で VM\_MONITORING\_STATUS を送信します。

モニタリングスクリプトが欠落している場合、または ESC スイッチオーバー後にモニタが停止しているときにモニタリングタイマーが期限切れになった場合、モニタリングの設定または設定解除操作中にエラーが発生します。

VM\_MONITOR\_STATUS 通知が NB に送信されます。ESC は VM をモニタせず、リカバリプロセスを開始できません。障害発生後にモニタリングを有効にするには、モニタリングを無効にしてから有効にする必要があります。

### 通知

```

WARN ===== SEND NOTIFICATION STARTS =====
WARN Type: VM_MONITORING_STATUS
WARN Status: FAILURE
WARN Status Code: 500
WARN Status Msg: No response from the monitor
WARN Tenant: tenant
WARN Deployment ID: 02cc4018-e4e3-4974-884a-f9fee17d7040
WARN Deployment name: dep

```

```
WARN VM group name: g1
WARN VM Source:
WARN VM ID: 6aa98b79-9d35-442a-9abb-f611e6316083
WARN VM Name: dep_g1_0_7fdae2a6-5095-4071-9c50-fb80c0e6b80e
WARN VM Name (Generated): dep_g1_0_7fdae2a6-5095-4071-9c50-fb80c0e6b80e
WARN VIM ID: default_openstack_vim
WARN VIM Project: tenant
WARN VIM Project ID: 33bf6768e45445da87feed838b248849
WARN Host ID: 79e4104d1d33de80aab13205b1e3c61d64aa4b61230c8b7b064b2891
WARN Host Name: my-ucs-62
WARN ===== SEND NOTIFICATION ENDS =====
```

## モニタリング操作

RESTful インターフェイスを使用して VM のモニタリングを設定および設定解除できます。

VM のモニタリングにはペイロードが必要です。

```
POST ESCManager/v0/{internal_tenant_id}/deployments/vm/{vm_name}
```

例：

```
<?xml version="1.0" encoding="UTF-8"?>
<vm_operation xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <operation>enable_monitoring</operation>
  <force>>false</force>
</vm_operation>
```

VM モニタリングを設定するには `enable_monitoring` を、VM モニタリングを設定解除するには `disable_monitoring` を、それぞれ `operation` フィールドに指定する必要があります。



---

(注) ユーザーが ESC ポータルから VM を再起動すると、モニタリングが自動的に有効になります。

---

## 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。