



# 仮想ネットワーク機能のスケールリング

- [スケールリングの概要](#) (1 ページ)
- [VM のスケールインとスケールアウト](#) (1 ページ)
- [スケールリングのためのリソースの一貫した順序付け](#) (3 ページ)
- [スケールリング通知とイベント](#) (4 ページ)

## スケールリングの概要

ESC では、サービスを柔軟に拡張でき、スケールインとスケールアウトの両方を自動的に実行するように設定できます。スケールリングは、KPI、ルール、およびアクションを使用して実現されます。これらは展開時に設定されます。KPI では、イベント名としきい値を定義します。ルールでは、スケールアウトとスケールインをトリガーするアクションを定義します。

KPI、ルール、およびメトリックの詳細については、[KPI](#)、[ルール](#)、および[メトリック](#)を参照してください。

## VM のスケールインとスケールアウト

スケールリングワークフローは、VNF の展開が成功した後に開始されます。VM は、データモデルの KPI データを形成する CPU 負荷、メモリ使用率などの属性をモニタするように設定されます。いずれかの属性について、定義されたアクションに基づいて KPI がしきい値に達すると、スケールインとスケールアウトが実行されます。

- スケールアウト中に、VM の数がアクティブな最大数を下回ると、新しい VM の展開がトリガーされます。
- スケールイン中に、VM の数がアクティブな最小数を超えると、VM は展開解除されます。



(注) VM が展開され、VMAlive イベントを受信しなかった場合、リカバリがトリガーされます。展開解除中のエラーは、ノースバウンドユーザに通知されます。

データモデルのスケーリングセクションでは、最小値と最大値が設定されます。`min_active`では、展開された VM の数を定義します。`max_active`では、展開可能な VM の最大数を定義します。たとえば、最小で 2、最大で 100 の VM を指定して VNF を展開する場合、以下の XML で各 VM グループのスケーリングを定義します。

スタティック IP アドレスを使用してアクティブ VM を設定した場合、スケールアウトされた VM にスタティック IP アドレスを割り当てる必要があります。展開時に、スタティック IP アドレスのリストを指定する必要があります。次に、スタティック IP プールを作成する例を示します。

```
<scaling>
  <min_active>1</min_active>
  <max_active>2</max_active>
  <elastic>true</elastic>
  <static_ip_address_pool>
    <network>1234-5678-9123</network>
    <gateway>10.86.22.1</gateway>
    <netmask>255.255.255.0</netmask>
    <ip_address>10.86.22.227</ip_address>
    <ip_address>10.86.22.228</ip_address>
  </static_ip_address_pool>
</scaling>
```

次に、KPI データセクションで CPU 負荷を検出する方法の例を示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<kpi>
  <event_name>VM_OVERLOADED</event_name>
  <metric_value>70</metric_value>
  <metric_cond>GT</metric_cond>
  <metric_type>UINT32</metric_type>
  <metric_occurrences_true>2</metric_occurrences_true>
  <metric_occurrences_false>4</metric_occurrences_false>
  <metric_collector>
    <type>CPU_LOAD_1</type>
    <nicid>0</nicid>
    <poll_frequency>3</poll_frequency>
    <polling_unit>seconds</polling_unit>
    <continuous_alarm>>false</continuous_alarm>
  </metric_collector>
</kpi>
<kpi>
  <event_name>VM_UNDERLOADED</event_name>
  <metric_value>40</metric_value>
  <metric_cond>LT</metric_cond>
  <metric_type>UINT32</metric_type>
  <metric_occurrences_true>2</metric_occurrences_true>
  <metric_occurrences_false>4</metric_occurrences_false>
  <metric_collector>
    <type>CPU_LOAD_1</type>
    <nicid>0</nicid>
    <poll_frequency>3</poll_frequency>
    <polling_unit>seconds</polling_unit>
    <continuous_alarm>>false</continuous_alarm>
  </metric_collector>
</kpi>
```

KPI ルールは次のとおりです。

```
<rule>
  <event_name>VM_OVERLOADED</event_name>
  <action>ALWAYS log</action>
  <action>TRUE servicescaleup.sh</action>
</rule>
<rule>
  <event_name>VM_UNDERLOADED</event_name>
  <action>ALWAYS log</action>
  <action>TRUE servicescaledown.sh</action>
</rule>
```

ETSI API を使用した VNF のスケーリングについては、Cisco Elastic Services Controller NFV MANO ガイド [英語] を参照してください。

## スケーリングのためのリソースの一貫した順序付け

ESC では、IP アドレス、MAC アドレス、デイズロ設定変数などのリソースを展開データモデルに一貫した方法で指定できます。

ESC は、手動および自動スケーリング中に、一貫した方法で展開データモデル内の静的 IP アドレスプールを割り当ておよび割り当て解除します。

次に例を示します。

```
<scaling>
  <min_active>3</min_active>
  <max_active>6</max_active>
  <static_ip_address_pool>
    <network>jenkins-internal-vnf-net-1</network>
    <ip_address>192.168.15.3</ip_address>
    <ip_address>192.168.15.111</ip_address>
    <ip_address>192.168.15.22</ip_address>
    <ip_address>192.168.15.5</ip_address>
    <ip_address>192.168.15.4</ip_address>
    <ip_address>192.168.15.222</ip_address>
  </static_ip_address_pool>
</scaling>
```

- **手動スケーリング**：ESC は、スケールアウト時に静的 IP プールで使用可能な順序で IP アドレスを割り当てます。スケールインの間、IP アドレスはラストインファーストアウトの順序でリリースされます。
- **自動スケーリング**：自動スケーリングは、SNMP イベントを使用して VNF のオーバーロードとアンダーロードを示します。オーバーロードイベントによって ESC がスケールアウトし、展開データモデルにリストされている順序から、静的 IP プールの最初の空き IP アドレスを割り当てます。スケールインの間、ESC は IP アドレスの割り当てを解除し、IP アドレスは将来のスケーリングイベントのために解放されます。

デイズロ設定、展開データモデルの IP アドレスの詳細については、[導入パラメータ](#)を参照してください。

## スケーリング通知とイベント

スケーリング通知は、ノースバウンドユーザに送信されます。通知には、スケーリング中のサービスを特定するためのステータスメッセージとその他の詳細情報が含まれます。通知のリストは次のとおりです。

```
VM_SCALE_OUT_INIT
VM_SCALE_OUT_DEPLOYED
VM_SCALE_OUT_COMPLETE
VM_SCALE_IN_INIT
VM_SCALE_IN_COMPLETE
```

次の表に、スケーリングシナリオと生成される通知を示します。

シナリオ	通知
スケールアウト	<p>ESCがVMを展開し、KPI Monitorと受信したすべてのVM Aliveが設定されます。次のNETCONF通知がトリガーされます。</p> <pre>&lt;type&gt;SERVICE_ALIVE&lt;/type&gt; &lt;status&gt;SUCCESS&lt;/status&gt;</pre> <p>ESCがVM_OVERLOADED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>&lt;type&gt; VM_SCALE_OUT_INIT&lt;/type&gt; &lt;status&gt;SUCCESS&lt;/status&gt;</pre> <p>ESCが最大値の制限に達したかチェックし、達していない場合は新しいVMが展開されます。</p> <pre>&lt;type&gt; VM_SCALE_OUT_DEPLOYED&lt;/type&gt; &lt;status&gt;SUCCESS&lt;/status&gt;</pre> <p>展開が完了すると、次のNETCONF通知が送信されます。</p> <pre>&lt;type&gt;VM_SCALE_OUT_COMPLETE&lt;/type&gt; &lt;status&gt;SUCCESS&lt;/status&gt;</pre>

シナリオ	通知
スケールイン	<p>ESCがVMを展開し、KPIMonitorと受信したすべてのVM Aliveが設定されます。</p> <p>NETCONF通知が送信されます。</p> <pre data-bbox="980 436 1312 485">&lt;type&gt;SERVICE_ALIVE&lt;/type&gt; &lt;status&gt;SUCCESS&lt;/status&gt;</pre> <p>ESCがVM_UNDERLOADED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="980 630 1365 678">&lt;type&gt; VM_SCALE_IN_INIT&lt;/type&gt; &lt;status&gt;SUCCESS&lt;/status&gt;</pre> <p>ESCがVMの数が最小アクティブ制限を超えているかチェックします。超えている場合は、展開解除の完了後に、いずれかのVMの展開が解除され、NETCONF通知が送信されます。</p> <pre data-bbox="980 856 1403 905">&lt;type&gt;VM_SCALE_IN_COMPLETE&lt;/type&gt; &lt;status&gt;SUCCESS&lt;/status&gt;</pre>

すべてのエラーシナリオで、通知はFAILUREステータスで送信されます。また、ステータスメッセージには、対応する障害の詳細が表示されます。

