



ETSI LCM 操作のアラームと通知

- [ETSI アラーム \(1 ページ\)](#)
- [通知への登録 \(4 ページ\)](#)
- [VNF の ETSI 障害および負荷の通知 \(6 ページ\)](#)

ETSI アラーム

ESC は、NFVO にアラームと通知を提供します。NFVO はこれらのアラームと通知に登録し、要求を ESC に送信する必要があります。

NFVO は、次の方法でアラームに関する情報を受信できます。

すべてのアラームをクエリする

NFVO は、アラームリソースからすべてのアラームのリストを取得できます。

メソッドタイプ :

GET

VNFM エンドポイント :

/vnfm/v1/alarms

HTTP 要求ヘッダー :

Accept:application/json

たとえば、イベントタイプが ENVIRONMENTAL_ALARM のすべてのアラームをクエリする場合

メソッドタイプ :

GET

VNFM エンドポイント :

http://localhost:8250/vnfm/v1/alarms?eventType="ENVIRONMENTAL_ALARM"

HTTP 要求ヘッダー :

Accept:application/json

複数アラームのクエリ中に、NFVOはURIクエリパラメータを使用して結果をフィルタ処理できます。アラームのURIクエリでは、次の属性名がサポートされています。

- id
- managedObjectId
- rootCauseFaultyResource.faultyResourceType
- eventType
- perceivedSeverity
- probableCause



(注) URIクエリパラメータは、複数のアラームのクエリ専用です。

個々のアラームのクエリ

NFVOは、*alarmId* リソースから特定のアラームをクエリできます。

メソッドタイプ：

GET

VNFM エンドポイント

/vnffm/v1/alarms/{alarmId}

HTTP 要求ヘッダー：

Accept:application/json

個々のアラームの変更

アラームを変更するには、NFVOが *AlarmModifications* リソースに PATCH 要求を送信する必要があります。

メソッドタイプ：

PATCH

VNFM エンドポイント：

/vnffm/v1/alarms/{alarmId}

HTTP 要求ヘッダー：

Content-Type: application/merge-patch+json

If-Match: ETag value



(注) **If-Match:** はオプションです。指定した場合、その値は VNF に保存された ETag 値に対して検証されます (1 つの VNF クエリから返されます)。

サポートされる属性は `ackState` で、サポートされる属性値は `ACKNOWLEDGE` です。他のすべての変更ペイロードは拒否されます。

VNF 障害および負荷アラーム

次のアラームは、ETSI VNF 障害および負荷通知用に作成されます。

- 障害アラーム：ESC は、VFND の `VM_ALIVE` KPI 設定に基づいて VNF 内のコンピューティングリソースの1つが到達不能になると、障害アラームを生成します。詳細については、「[VNF の ETSI 障害および負荷の通知](#)」を参照してください。

例：

メソッドタイプ

POST

VNFM エンドポイント

`/vnffm/v1/extension/alarms`

HTTP 要求ヘッダー

`Content-Type:application/json`

要求ペイロード：

```
{
  "externalAlarmId" : "26bf1e3d-cefa-4f59-88ea-210a29358a5c", #generated value
  "alarmSource" : "MONA", #hard-coded
  "managedObjectId" : "08733ef2-319b-46ce-9d8d-95730306bd1a", #external_deployment_id

  "rootCauseFaultyResource" : "chrimann-dep_g1_0_212da327-0573-421b-ae37-057f6b1a6aef",
  #vm_name
  "alarmRaisedTime" : "$timestamp", #generated value
  "ackState" : "UNACKNOWLEDGED", #hard-coded
  "perceivedSeverity" : "CRITICAL", #hard-coded
  "eventTime" : "2018-05-08T00:59:32.571+00:00", #do we have the eventTime?
  "eventType" : "EQUIPMENT_ALARM", #hard-coded
  "faultType" : "COMPUTE", #hard-coded
  "probableCause" : "VM_MANUAL_RECOVERY_NEEDED", #event_name
  "isRootCause" : "TRUE", #hard-coded
  "links" : {
    "objectInstance" :
    "{(http_scheme)://{api_root}/vnflcm/v1/vnf_instances/08733ef2-319b-46ce-9d8d-95730306bd1a"}"
  }
}
```

- 負荷アラーム：ESCは、VFNDの関連KPI設定に基づいて、VNF内のコンピューティングリソースの1つが過負荷または過小負荷になると、負荷アラームを生成します。ESCは、NFVOから通知を受信した後にこれらのアラームを作成します。詳細については、「[VNF の ETSI 障害および負荷の通知](#)」を参照してください。

例：

メソッドタイプ

POST

VNFM エンドポイント

```

/vnffm/v1/extension/alarms

HTTP 要求ヘッダー

Content-Type:application/json

要求ペイロード

```

アラーム拡張

ETSI は、サードパーティツールとやり取りするアラームの拡張機能を提供します。アラームを作成するには、POST 要求を送信する必要があります。

メソッドタイプ

POST

VNFM エンドポイント

```
/vnffm/v1/extension/alarms
```

HTTP 要求ヘッダー

```
Content-Type:application/json
```

要求ペイロード

```

[admin@davwebst-esc-4-2-0-49-keep ETSI]$ cat CreateAlarm.json
{
  "id": "alm87032",
  "externalAlarmId": "ext-id-xx11214",
  "managedObjectId": "930fb087-c1b9-4660-bec8-2a8d97dc1df5",
  "rootCauseFaultyResource": {
    "id": "fres7629",
    "faultyResource": {
      "resourceId": "res7727"
    },
    "faultyResourceType": "NETWORK"
  },
  "alarmRaisedTime": "2018-05-30T13:55:15.645000+00",
  "ackState": "UNACKNOWLEDGED",
  "perceivedSeverity": "MAJOR",
  "eventTime": "2018-05-30T13:55:15.645000+00",
  "eventType": "ENVIRONMENTAL_ALARM",
  "probableCause": "Server room overheating",
  "isRootCause": "false",
  "vnfInstanceIds": [
    "res-a3023a03-fc73-430a-a983-5e9439011e45"
  ]
}

```

通知への登録

NFVO は、ESC からの障害管理に関連した ETSI 通知に登録できます。

サブスクリプションの作成

NFVO は、通知に登録するための POST 要求を送信します。

メソッドタイプ :

POST

VNFM エンドポイント :

/vnffm/v1/subscriptions

応答ペイロード :

```
{
  "filter" : {
    "notificationTypes" : [
      "AlarmNotification",
      "AlarmClearedNotification",
      "AlarmListRebuiltNotification"
    ],
    "perceivedSeverities" : [
      "CRITICAL",
      "MAJOR"
    ]
  },
  "callbackUri" : "https://nfvo.endpoint.listener",
  "authentication" : {
    "authType" : "BASIC",
    "paramsBasic" : {
      "userName" : "admin",
      "password" : "pass123"
    }
  }
}
```

これにより、新しい登録リソースと新しい ID が作成されます。必須パラメータは `callbackUri` だけです。その他はすべてオプションです。GET 要求を送信することで、`callbackuri` が有効で到達可能かどうかを確認できます。

すべてのサブスクリプションのクエリ

NFVO は、サブスクリプションリソースに GET 要求を送信することで、そのサブスクリプションに関する情報をクエリできます。

メソッドタイプ :

GET

VNFM エンドポイント :

/vnffm/v1/subscriptions

HTTP 要求ヘッダー :

Accept:application/json

たとえば、`callbackUri` が以下の場合、すべてのアラートサブスクリプションをクエリするには

`http://10.10.1.44:9202/alerts/subscriptions/callback`

GET

VNFM エンドポイント

`http://localhost:8250/vnffm/v1/subscriptions?callbackUri="http://10.10.1.44:9202/alerts/subscriptions/callback"`

HTTP 要求ヘッダー

```
Accept:application/json
```

NFVO は URI クエリパラメータを使用して、結果をフィルタ処理できます。サブスクリプションの URI クエリでは、次の属性名がサポートされています。

- id
- filter
- callbackUri



(注) URI クエリパラメータは、複数のサブスクリプションのクエリ専用です。

個々のサブスクリプションのクエリ

個々のサブスクリプションをクエリするには、サブスクリプション ID を知っている必要があります。

メソッドタイプ :

```
GET
```

VNFM エンドポイント :

```
/vnffm/v1/subscriptions/{subscriptionId}
```

HTTP 要求ヘッダー :

```
Accept:application/json
```

サブスクリプションの削除

NFVO が必要としていないサブスクリプションを削除できます。個々のサブスクリプションに削除要求を送信します。

メソッドタイプ :

```
DELETE
```

VNFM エンドポイント :

```
/vnffm/v1/subscriptions/{subscriptionId}
```

HTTP 要求ヘッダー :

```
http://localhost:8250/vnffm/v1/subscriptions/682791f8-34ad-487e-811a-553036bf49b2
```

VNF の ETSI 障害および負荷の通知

ESC は次の通知を生成します。

- VM の障害

展開された VNF 内の VM に障害が発生すると、NFVO は ESC から障害通知を受信します。通知を受信すると、アラームが生成されます。アラームの詳細については、[ETSI アラーム \(1 ページ\)](#) を参照してください。

NFVO は通知のために ESC に登録する必要があります。

例 :

```
<?xml version="1.0" encoding="UTF-8"?>
<esc_event xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <deployment_name>sample-dep</deployment_name>
  <event_name>MY_VM_UNDERLOADED</event_name>
  <event_type>VM_UNDERLOADED</event_type>
  <external_deployment_id>e911ecef-5f3f-456c-9c80-d99aca2416da</external_deployment_id>

  <external_tenant_id>etsi_tenant</external_tenant_id>
  <internal_deployment_id>99f7629f-98d3-40f5-ad68-7addcfe07006</internal_deployment_id>

  <internal_tenant_id>etsi_tenant</internal_tenant_id>
  <vm_source>

  <generated_vm_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</generated_vm_name>

  <interfaces>
    <addresses>
      <address>
        <address_id>0</address_id>
        <gateway>172.16.0.1</gateway>
        <ip_address>172.16.0.0</ip_address>
        <dhcp_enabled>>true</dhcp_enabled>
        <prefix>20</prefix>
        <subnet>365a0884-fdb3-424c-afe9-2deb3b39baae</subnet>
      </address>
    </addresses>
    <network_uuid>c7fafeca-aa53-4349-9b60-1f4b92605420</network_uuid>
    <mac_address>fa:16:3e:38:1d:6c</mac_address>
    <nic_id>0</nic_id>
    <port_forwarding/>
    <port_uuid>0aeb9585-5190-4f3b-b1aa-495e09c56b7d</port_uuid>
    <security_groups/>
    <subnet_uuid>none</subnet_uuid>
    <type>virtual</type>

  <vim_interface_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</vim_interface_name>

</interfaces>
<vim_id>default_openstack_vim</vim_id>
<vim_project>admin</vim_project>
<vim_project_id>c12f013306d849e5b1bbf257c54d5891</vim_project_id>
<host_uuid>6b8cf361c5ff08a5a886e26f591b8087dadcf2d2b34fb3b5d2772a8d</host_uuid>

<host_name>my-server</host_name>
<vm_uuid>9fea3fe7-9417-4734-b962-b24340941ef3</vm_uuid>
<vm_group_name>vm1</vm_group_name>
<vm_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</vm_name>
</vm_source>
</esc_event>
```

• VM のオーバーロードとアンダーロード

同様に、NFVO は VM のオーバーロードまたはアンダーロードの通知を受信します。

スケーリングが自動的に有効になっていない場合、ESCはVMの状態に応じて通知を生成します。

次に例を示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<esc_event xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <deployment_name>sample-dep</deployment_name>
  <event_name>MY_VM_UNDERLOADED</event_name>
  <event_type>VM_UNDERLOADED</event_type>
  <external_deployment_id>e911eecf-5f3f-456c-9c80-d99aca2416da</external_deployment_id>

  <external_tenant_id>etsi_tenant</external_tenant_id>
  <internal_deployment_id>99f7629f-98d3-40f5-ad68-7addcfe07006</internal_deployment_id>

  <internal_tenant_id>etsi_tenant</internal_tenant_id>
  <vm_source>

  <generated_vm_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</generated_vm_name>

  <interfaces>
    <addresses>
      <address>
        <address_id>0</address_id>
        <gateway>172.16.0.1</gateway>
        <ip_address>172.16.0.0</ip_address>
        <dhcp_enabled>true</dhcp_enabled>
        <prefix>20</prefix>
        <subnet>365a0884-fdb3-424c-afe9-2deb3b39baae</subnet>
      </address>
    </addresses>
    <network_uuid>c7fafeca-aa53-4349-9b60-1f4b92605420</network_uuid>
    <mac_address>fa:16:3e:38:1d:6c</mac_address>
    <nic_id>0</nic_id>
    <port_forwarding/>
    <port_uuid>0aeb9585-5190-4f3b-b1aa-495e09c56b7d</port_uuid>
    <security_groups/>
    <subnet_uuid>none</subnet_uuid>
    <type>virtual</type>

  <vim_interface_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</vim_interface_name>

  </interfaces>
  <vim_id>default_openstack_vim</vim_id>
  <vim_project>admin</vim_project>
  <vim_project_id>c12f013306d849e5b1bbf257c54d5891</vim_project_id>
  <host_uuid>6b8cf361c5ff08a5a886e26f591b8087dadcf2d2b34fb3b5d2772a8d</host_uuid>

  <host_name>my-server</host_name>
  <vm_uuid>9fea3fe7-9417-4734-b962-b24340941ef3</vm_uuid>
  <vm_group_name>vm1</vm_group_name>
  <vm_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</vm_name>
  </vm_source>
</esc_event>
```

VM のアンダーロードの例 :

```
<?xml version="1.0" encoding="UTF-8"?>
<esc_event xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <deployment_name>sample-dep</deployment_name>
  <event_name>MY_VM_OVERLOADED</event_name>
  <event_type>VM_OVERLOADED</event_type>
  <external_deployment_id>e911eecf-5f3f-456c-9c80-d99aca2416da</external_deployment_id>
```



```

<external_tenant_id>etsi_tenant</external_tenant_id>
<internal_deployment_id>99f7629f-98d3-40f5-ad68-7addcfe07006</internal_deployment_id>

<internal_tenant_id>etsi_tenant</internal_tenant_id>
<vm_source>

<generated_vm_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</generated_vm_name>

<interfaces>
  <addresses>
    <address>
      <address_id>0</address_id>
      <gateway>172.16.0.1</gateway>
      <ip_address>172.16.0.0</ip_address>
      <dhcp_enabled>true</dhcp_enabled>
      <prefix>20</prefix>
      <subnet>365a0884-fdb3-424c-afe9-2deb3b39baae</subnet>
    </address>
  </addresses>
  <network_uuid>c7fafeca-aa53-4349-9b60-1f4b92605420</network_uuid>
  <mac_address>fa:16:3e:38:1d:6c</mac_address>
  <nic_id>0</nic_id>
  <port_forwarding/>
  <port_uuid>0aeb9585-5190-4f3b-b1aa-495e09c56b7d</port_uuid>
  <security_groups/>
  <subnet_uuid>none</subnet_uuid>
  <type>virtual</type>

<vim_interface_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</vim_interface_name>

</interfaces>
<vim_id>default_openstack_vim</vim_id>
<vim_project>admin</vim_project>
<vim_project_id>c12f013306d849e5b1bbf257c54d5891</vim_project_id>
<host_uuid>6b8cf361c5ff08a5a886e26f591b8087dadcf2d2b34fb3b5d2772a8d</host_uuid>

<host_name>my-server</host_name>
<vm_uuid>9fea3fe7-9417-4734-b962-b24340941ef3</vm_uuid>
<vm_group_name>vm1</vm_group_name>
<vm_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</vm_name>
</vm_source>
</esc_event>

```



- (注) ETSI は、ESC からの通知を受信したときに、`instantiatedVnfInfo.vnfcResourceInfo` に存在する VNFC のアラームのみを生成します。

KPI 手順を使用した VNF の自動スケーリング

ESC は、KPI 手順を使用して VM を自動スケーリングできます。スケーリングワークフローは、VNF インスタンスがインスタンス化された状態のときに開始されます。NFVO は、VNF の `isAutoscaleEnabled` 設定可能プロパティを変更しながら、自動スケーリングを有効または無効にできます。

次に、ETSI 準拠の自動スケールをトリガーするイベントを示します。これには、`ScaleVnfToLevelRequest`: 機能の導入が必要です。

• オーバーロードとアンダーロード

VM の状態が変化し、VM がオーバーロードまたはアンダーロードの場合、ESC はスケーリングが自動的に有効になっているかどうかを判断する通知を受け取ります。そうでない場合、ESC は VNF の状態を確認するために ETSI-VNFM コンポーネントへの通知を生成します。

次の例は、ESC からのアンダーロード通知を示しています。

```

Headers:
  esc-status-code = 200
  esc-status-message = VM [sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de]
  underloaded.
Body:
<?xml version="1.0" encoding="UTF-8"?>
<esc_event xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <deployment_name>sample-dep</deployment_name>
  <event_name>MY_VM_UNDERLOADED</event_name>
  <event_type>VM_UNDERLOADED</event_type>
  <external_deployment_id>e911ecef-5f3f-456c-9c80-d99aca2416da</external_deployment_id>

  <external_tenant_id>etsi_tenant</external_tenant_id>
  <internal_deployment_id>99f7629f-98d3-40f5-ad68-7addcfe07006</internal_deployment_id>

  <internal_tenant_id>etsi_tenant</internal_tenant_id>
  <vm_source>

  <generated_vm_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</generated_vm_name>

  <interfaces>
    <addresses>
      <address>
        <address_id>0</address_id>
        <gateway>172.24.0.1</gateway>
        <ip_address>172.24.0.37</ip_address>
        <dhcp_enabled>>true</dhcp_enabled>
        <prefix>20</prefix>
        <subnet>365a0884-fdb3-424c-afe9-2deb3b39baae</subnet>
      </address>
    </addresses>
    <network_uuid>c7fafeca-aa53-4349-9b60-1f4b92605420</network_uuid>
    <mac_address>fa:16:3e:38:1d:6c</mac_address>
    <nic_id>0</nic_id>
    <port_forwarding/>
    <port_uuid>0aeb9585-5190-4f3b-b1aa-495e09c56b7d</port_uuid>
    <security_groups/>
    <subnet_uuid>none</subnet_uuid>
    <type>virtual</type>

  <vim_interface_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</vim_interface_name>

  </interfaces>
  <vim_id>default_openstack_vim</vim_id>
  <vim_project>admin</vim_project>
  <vim_project_id>c12f013306d849e5b1bbf257c54d5891</vim_project_id>
  <host_uuid>6b8cf361c5ff08a5a886e26f591b8087dadcf2d2b34fb3b5d2772a8d</host_uuid>

  <host_name>my-server-65</host_name>
  <vm_uuid>9fea3fe7-9417-4734-b962-b24340941ef3</vm_uuid>
  <vm_group_name>vm1</vm_group_name>
  <vm_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</vm_name>
  </vm_source>
</esc_event>

```

次の例は、ESC からのオーバーロード通知を示しています。

```

Headers:
  esc-status-code = 200
  esc-status-message = VM [sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de]
  overloaded.
Body:
<?xml version="1.0" encoding="UTF-8"?>
<esc_event xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <deployment_name>sample-dep</deployment_name>
  <event_name>MY_VM_OVERLOADED</event_name>
  <event_type>VM_OVERLOADED</event_type>
  <external_deployment_id>e911eecf-5f3f-456c-9c80-d99aca2416da</external_deployment_id>

  <external_tenant_id>etsi_tenant</external_tenant_id>
  <internal_deployment_id>99f7629f-98d3-40f5-ad68-7addcfe07006</internal_deployment_id>

  <internal_tenant_id>etsi_tenant</internal_tenant_id>
  <vm_source>

  <generated_vm_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</generated_vm_name>

  <interfaces>
    <addresses>
      <address>
        <address_id>0</address_id>
        <gateway>172.24.0.1</gateway>
        <ip_address>172.24.0.37</ip_address>
        <dhcp_enabled>true</dhcp_enabled>
        <prefix>20</prefix>
        <subnet>365a0884-fdb3-424c-afe9-2deb3b39baae</subnet>
      </address>
    </addresses>
    <network_uuid>c7fafeca-aa53-4349-9b60-1f4b92605420</network_uuid>
    <mac_address>fa:16:3e:38:1d:6c</mac_address>
    <nic_id>0</nic_id>
    <port_forwarding/>
    <port_uuid>0aeb9585-5190-4f3b-b1aa-495e09c56b7d</port_uuid>
    <security_groups/>
    <subnet_uuid>none</subnet_uuid>
    <type>virtual</type>

  </vim_interface_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</vim_interface_name>

  </interfaces>
  <vim_id>default_openstack_vim</vim_id>
  <vim_project>admin</vim_project>
  <vim_project_id>c12f013306d849e5b1bbf257c54d5891</vim_project_id>
  <host_uuid>6b8cf361c5ff08a5a886e26f591b8087dadcf2d2b34fb3b5d2772a8d</host_uuid>

  <host_name>my-server-65</host_name>
  <vm_uuid>9fea3fe7-9417-4734-b962-b24340941ef3</vm_uuid>
  <vm_group_name>vm1</vm_group_name>
  <vm_name>sample-dep_vm1_0_fbc3da46-e0c6-40dc-91c8-70b1a88857de</vm_name>
</vm_source>
</esc_event>

```

• VNFD

VNFD 通知には、VNF 操作フローの *isAutoscaleEnabled* 設定可能プロパティに必要なスケールアクションの手順が含まれています。

スケーリングが自動的に有効になっていない場合は、KPI 手順を使用して手動 LCM 操作を実行できます。これは、ESC 通知ストリームを処理することで実行されます。KPI イベントを受信したら、通知を検証する必要があります。

次のアクションを実行する必要があります。

- 一致する VNF インスタンスを検索する
- 適切な設定プロパティが自動操作を有効にするよう設定されていることを確認する

検証に合格した場合、適切な操作の実行および関連する通知を生成するための操作フローを開始するよう要求できます。スケーリングでは、指定した KPI データによってスケーリングパラメータが決定されます。プロパティファイルには、次の新しい属性が含まれます。

```
external.scaling.decision = 1
#external.scaling.window = 120
external.healing.decision = 1
#external.healing.window = 120
```

• VnfInstance リソース

VNFD は、現在の `scaleStatus` を使用してスケールレベルを決定します。要求の処理によって、ESCManager から要求する VM の数が決まります。要求は、増分の相対数 (`SCALE_IN` または `SCALE_OUT`) のみを指定します。

スケーリングする VNF の `vnfInstance` リソースから、`vnfInstanceId` を使用して、次のペイロードで `ScaleVnfToLevel` エンドポイントを呼び出すことができます。

`VnfLcmOpOcc.isAutomaticInvocation` に `true` が設定されていることを確認します。

次の例は JSON ペイロードを示しています。

```
{
  /* "instantiationLevelId":"id111", */
  "scaleInfo": [
    { "aspectId":"processing", "scaleLevel":"3" },
    { "aspectId":"database", "scaleLevel":"2" }
  ]
  "additionalParams": {
    "password": "pass1234",
    "username": "admin"
  },
  "action": "scale_to_level"
}
```

KPI 手順を使用した VNF の修復

ESC は、KPI 手順を使用して VM を自動修復できます。NFVO は、VNF の `isAutohealEnabled` 設定可能プロパティの変更中に自動修復を有効または無効にします。

`isAutohealEnabled` プロパティは、自動修復機能を有効 (TRUE) または無効 (FALSE) にすることを許可します。

•