



## 仮想ネットワーク機能の修復

- [ETSI API を使用した仮想ネットワーク機能の修復 \(1 ページ\)](#)
- [修復中の既存の展開の更新 \(3 ページ\)](#)

### ETSI API を使用した仮想ネットワーク機能の修復

ESC は、ライフサイクル管理の一環として、障害が発生すると VNF を修復します。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、ポリシー主導型の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。『[Cisco Elastic Services Controller User Guide](#)』の「[Configuring a Recovery Policy Using the Policy-driven Framework](#)」を参照してください。

修復パラメータは、VNF を修復する通知をトリガーするためにモニタする動作を定義します。これらのパラメータは、ルールとともに VNFD の各コンピューティングノードの KPI セクションで設定されます。ルールでは、VNF を修復するためにこれらの KPI 条件の結果として実行されるアクション（トリガーされるイベントを含む）を定義します。

ESC ETSI は、次の 2 つのセクションを使用してモニタリングを設定します。

- `kpi_data` : 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。
- `admin_rules` : 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。

例 :

```
vdu1:
  type: cisco.nodes.nfv.Vdu.Compute
  properties:
    name: Example VDU1
    description: Example VDU
    ...
  kpi_data:
    VM_ALIVE-1:
      event_name: 'VM_ALIVE-1'
      metric_value: 1
      metric_cond: 'GT'
      metric_type: 'UINT32'
      metric_occurrences_true: 1
```

```

metric_occurrences_false: 30
metric_collector:
  type: 'ICMPping'
  nicid: 1
  poll_frequency: 10
  polling_unit: 'seconds'
  continuous_alarm: false
admin_rules:
  VM_ALIVE-1:
    event_name: 'VM_ALIVE-1'
    action:
      - 'ALWAYS log'
      - 'FALSE recover autohealing'
      - 'TRUE esc_vm_alive_notification'
...

```

この例は、デフォルトのKPIと、ESCでの展開を完了するために必要なサービスライブ通知をサポートするルールを示しています。VNFDで公開されるKPI、ルール、および基盤となるデータモデルの詳細については、『[Cisco Elastic Services Controller User Guide](#)』の「KPIs, Rules and Metrics」を参照してください。

インスタンスに注意が必要なことを示すイベントを受信した場合、タイマーが期限切れになった、または手動のリカバリ要求を受信した場合のリカバリには、3種類のアクションがあります。修復のワークフローは次のようになります。

- **REBOOT\_THEN\_REDEPLOY** : 最初に、影響を受けたVNFCの再起動を試みます。これが失敗した場合、影響を受けたVNFCの再展開（同じホスト上で）を試みます
- **REBOOT\_ONLY** : VMの再起動のみを試みます
- **REDEPLOY\_ONLY** : VMの再展開のみを試みます

リカバリポリシーはVNFレベルで設定され、そこに含まれるVNFCごとに適用されます。モニタリングエージェントが各VNFCをモニタし、リカバリ状況になると、メッセージがアラームに変換され、登録されたコンシューマ（NFVOまたはElement Manager）に送信されます。

VNFインスタンスで自動修復が有効になっている場合、ESCは展開時に設定されたリカバリポリシーに基づいてVNFのリカバリを自動的に試みます。これは、VNFDで設定、またはインスタンス化の前にVNFインスタンスにおいて変更できます。

VNFのリカバリは、影響を受けたVNFCに対するアクションを要求することです。初期展開操作がタイムアウトした後、ESCが定義されたポリシーを使用してサービスを回復できない場合、サービスが展開に失敗すると、ライフサイクル管理操作は失敗します。

自動修復フラグ (*isAutohealEnable*) VNFインスタンスリソースを変更するには、[仮想ネットワーク機能の変更](#)を参照してください。

自動修復が有効でない場合、アラームのみがすべてのサブスクリバにディスパッチされます。サブスクリバは手動のHealVnfRequestを開始できます。データ構造は、あらゆるVNF固有のアクションに使用できます。必須パラメータはありません。

*SOL003* の例 :

```

Request Payload (ETSI data structure: HealVNFRequest)
POST /vnf_instances/{vnfInstanceId}/heal
{

```

```
    "cause": "b9909dde-e21e-45ec-9cc0-9e9ae413eee0",
  }
}
```

*SOL002* の例 :

```
POST /vnf_instance/{vnfInstanceId}/heal
{
  "vnfcInstanceId": ["b9909dde-e21e-45ec-9cc0-9e9ae413eee0"],
  "cause": "b9909dde-e21e-45ec-9cc0-9e9ae413eee0",
  "healScript": "REBOOT_ONLY"
}
```

`healScript` は有効なリカバリポリシー名の列挙として実装されます。これにより、展開データモデルで設定されたポリシーを上書きできます。`vnfcInstanceId` のリストは、必要な VNFC が影響を受けることを許可しますが、このリストがない場合、要求は VNF 全体に適用されます。

追加のパラメータを使用して、展開時に設定されたポリシーに関係なく、上書きするリカバリポリシーを指定できます。

リカバリポリシーは、追加のパラメータを使用して VNFC レベルで指定できます。これにより、VNF レベルで設定された値が上書きされます。リカバリポリシーが VNFC レベルで指定されていない場合、ESC は VNF レベルのリカバリポリシーからプロパティを継承します。

オプションの追加パラメータが `cisco.datatypes.nfv.VnfcAdditionalConfigurableProperties` データタイプに追加され、VNFC レベルのリカバリをサポートします。

```
cisco.datatypes.nfv.VnfcAdditionalConfigurableProperties:
  derived_from: toasca.datatypes.nfv.VnfcAdditionalConfigurableProperties
  properties:
    ...
    is_vnfc_autoheal_enabled:
      type: boolean
      description: It permits to enable (TRUE)/disable (FALSE) the auto-healing
        functionality. If the properties is not present for configuring, then VNF-level property
        is used instead
      required: false
      recovery_action:
        type: string
        required: false
      constraints:
        - valid_values: [ REBOOT_THEN_REDEPLOY, REDEPLOY_ONLY, REBOOT_ONLY ]
```

モニタリングの詳細については、[ETSI API を使用した仮想ネットワーク機能のモニタリング](#)を参照してください

## 修復中の既存の展開の更新

展開が正常に作成されたら、その中のリソースを更新できます。展開管理の一環として、リソースを追加または削除したり、既存のリソースの設定を更新したりできます。これらの更新は、実行中の展開で実行できます。リソースは、リカバリプロセスの一環として更新されます。

修復ワークフロー中に、（ETSI NFV MANO API を介してプロビジョニングされた）既存の展開を更新できます。修復要求中に、既存のイメージと Day-0 パラメータが比較され、後続の修復要求の一部として提供される新しいパラメータに更新されます。

ヒーリングワークフローでは、次のことが可能です。

- 展開モデルを新しいイメージと Day-0 設定で更新する
- アップグレードされたイメージによる修復時に、新規または既存の設定データを VNFC に再適用する



(注) 変更が VIM で直接実行されない場合、データモデルの更新後に VNF を再展開する必要があります。

HealVnfRequest を介して新しい *additionalParams* を指定した後、（NFVO からの）付与応答も新しいイメージまたは新しい *additionalParams* を指定する場合、これもサービス更新をトリガーします。

展開を再展開の一環として移動させる必要があると NFVO が判断した場合、付与はリソースの新しい配置を反映するための新しい *zoneId* を提供します。

リカバリアクションは、サービスの更新が完了した後に実行されます。再展開の場合は、最新の展開モデルを考慮して、展開された更新が元に戻されないようにします。

次の例は、新しい *additionalParams* や新しい *vimSoftwareImageId* でサービス更新をトリガーするために、NFVO が付与に返す詳細を示しています。

例：

```
{
  "headers" : {
    "Content-Type" : [ "application/json" ],
    "Location" : [
"http://{nfvoApiRoot}/sol003/default/grant/v1/grants/38ba2103-dab3-450e-992b-ee85aad6c899"
    ],
    "Content-Length" : [ "22935" ],
  },
  "body" : {
    "id" : "38ba2103-dab3-450e-992b-ee85aad6c899",
    "vnfInstanceId" : "6aaf527c-0093-49c3-ba2e-49fc6d8a4f71",
    "vnfLcmOpOccId" : "cdc5d9b3-81a0-400b-a4d9-97d1b3e117d9",
    "_links" : {
      "self" : {
        "href" :
"http://{nfvoApiRoot}/sol003default/grant/v1/grants/38ba2103-dab3-450e-992b-ee85aad6c899"
      },
      "vnfLcmOpOcc" : {
        "href" :
"https://{vnfmApiRoot}/vnflcm/v1/vnf_lcm_op_occs/cdc5d9b3-81a0-400b-a4d9-97d1b3e117d9"
      },
      "vnfInstance" : {
        "href" :
"https://{vnfmApiRoot}/vnflcm/v1/vnf_instances/6aaf527c-0093-49c3-ba2e-49fc6d8a4f71"
      }
    }
  }
}
```

```
    },
    "vimConnections" : [ {
      "id" : "myVimConnection",
      "vimType" : "OPENSTACK_V3",
      "vimId" : "595b0bc2-8dad-4087-abdf-ebe3b0b14d96",
      "interfaceInfo" : {
        "endpoint" : "https://{vimApiRoot}/v3"
      },
      "accessInfo" : {
        "password" : "*****",
        "project" : "cisco",
        "projectDomain" : "demo",
        "region" : "RegionOne",
        "userDomain" : "demo",
        "username" : "*****"
      }
    }
  ],
  "zones" : [{
    "id" : "1773873a-ab15-4a7b-b024-bc338425ed24",
    "zoneId" : "nova"
  }, {
    "id" : "1773873a-ab15-4a7b-b024-bc555555ed55",
    "zoneId" : "nova2"
  } ],
  "addResources" : [{
    "resourceDefinitionId" : "res-a6252dbf-b418-4f88-b8a9-14d8f3942938",
    "vimConnectionId" : "myVimConnection",
    "zoneId" : "1773873a-ab15-4a7b-b024-bc555555ed55"
  } ],
  "vimAssets" : {
    "softwareImages" : [ {
      "vnfdSoftwareImageId" : "s3",
      "vimSoftwareImageId" : "3a609da7-e2b2-4e27-91b6-7bcabe902820",
      "vimConnectionId" : "myVimConnection"
    }, {
      "vnfdSoftwareImageId" : "s4",
      "vimSoftwareImageId" : "3a609da7-e2b2-4e27-91b6-7bcabe902820",
      "vimConnectionId" : "myVimConnection"
    } ]
  }
},
"additionalParams": [
  ...
  /* changed additionalParams */
  "CF_VIP_ADDR": "10.123.23.4",
  "SF_VIP_ADDR": "10.123.24.4",
  ...
],
"statusCode" : "CREATED",
"statusCodeValue" : 201
}
```

修復の詳細については、[ETSI API を使用した仮想ネットワーク機能の修復 \(1 ページ\)](#) を参照してください。

