



# Scaling and Healing VNFs Using ETSI API

- [Scaling VNFs Using ETSI API, on page 1](#)
- [Healing VNFs Using ETSI API, on page 3](#)

## Scaling VNFs Using ETSI API

ESC can scale VMs using the ETSI API. The scaling workflow begins when the VNF instance is in the instantiated state. The scaling details are defined in the VNFD by the NFVO.

Scaling of a VNF instance can be achieved in two ways:

- Scale
- Scale to level

As part of scaling,

- Add VM instances to the VNF to scale out.
- Remove VM instances from the VNF to scale in.



---

**Note** Currently, ETSI supports only manual scaling.

---

### Scale

ESC can scale a VNF incrementally. The *numberOfSteps* attribute determines the increment by which you can scale the VNF instance. By default, the number of steps is set to 1.

Method type:

POST

VNFM Endpoint:

`/vnf_instances/{vnfInstanceId}/scale`

HTTP Request Header:

Content-Type:application/json

Request Payload (ETSI data structure: ScaleVnfRequest)

```
{
  "type":"SCALE_OUT",
  "aspectId":"processing",
  "numberOfSteps":"1",
}
```

Response Headers:

not applicable.

Response Body:

not applicable.

### Scale to Level

Allows ESC to scale the VNF to a specified level.

Method type:

POST

VNFM Endpoint:

/vnf\_instances/{vnfInstanceId}/scale\_to\_level

HTTP Request Header:

Content-Type:application/json

Request Payload (ETSI data structure: ScaleVnfRequest)

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

Response Headers:

not applicable.

Response Body:

not applicable.

# Healing VNFs Using ETSI API

As part of the VNF lifecycle management, ESC heals the VNFs when there is a failure. ETSI API heals a VNF instance based on the instructions from the NFVO.



**Note** Currently, ETSI supports only manual healing.

Method Type:

POST

VNFM Endpoint:

/vnf\_instances/{vnfInstanceId}/heal

HTTP Request Headers:

Content-Type:application/json

Request Payload (ETSI data structure: HealVnfRequest):

```
{
  "cause": "b9909dde-e21e-45ec-9cc0-9e9ae413eee0",
  "additionalParams": {
    "password": "pass1234",
    "username": "admin"
  }
}
```

Response Headers:

not applicable.

Response Body:

not applicable.

## Manual Recovery of a Deployment

Currently, ETSI supports recovery of a complete deployment.

REST API to heal a complete deployment is as follows:

```
{http_scheme}://{restapi_root}:8080/ESCAPI/#!/Recovery_VM_Operations/handleOperation
/v0/{internal_tenant_id}/deployments/service/{internal_deployment_id}
```

payload:

```
Content-Type: application/xml
Accept: application/json
Callback: http://{etsi-vnfm-callback-endpoint}:{etsi-vnfm-callback-port}/
Callback-ESC-Events: http://127.0.0.1:9010/
  <service_operation xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <operation>recover</operation>
  </service_operation>
```

where,

internal\_tenant\_id—is the system admin tenant ID or the tenant name.

internal\_deployment\_id—is the deployment name.

Both ids are encoded into the URL.

Manual healing of the complete deployment is supported when the deployment (service) is in the following states:

**Table 1: VM and Deployment States**

<b>VM state</b>	<b>Deployment or service state</b>	<b>recovery-vm-action</b>
error	error	supported
error	active	supported