

# ESC REST API

## ESC REST Northbound API Documentation

### Table of Contents

- [Resources managed by ESC REST](#)
- [Behavior of ESC REST operations](#)
- [ESC REST API Headers/Path/Body Parameters](#)
  - [Callbacks](#)
  - [Media Type](#)
  - [Internal IDs](#)
  - [Body](#)
  - [String Restrictions in Body/Internal IDs](#)
- [REST API Security](#)
  - [REST Authentication](#)
  - [REST Https Support](#)
- [REST API List](#)

### Resources managed by ESC REST

- Tenants
- Networks
- Subnets
- Flavors
- Images

- Volumes
- Deployments

## Behavior of ESC REST operations

GET operations are synchronous the request/response parameters are described with an example of a request/response

POST, DELETE, and PUT operations are asynchronous. The asynchronous design use the “web hooks” approach using two independent "one-way" invocations - one to start a long-running operation (Client to ESC) and the other one to notify a requester that it is completed (ESC to client)

## ESC REST API Headers/Path/Body Parameters

### Callbacks

One of the header parameters of the operation request will contain a callback field, whose value is a URI where the client expects a callback. If this field is not present, no callback will be performed.

When the operation has completed (either successfully or unsuccessfully), ESC will perform a POST request to the callback URI, with an entity body containing the resource for this operation. The status will be returned in the HTTP header called `esc-status-code` and the status message will be returned in the HTTP header called `esc-status-message`.

ESC Clients can match a completion report back to the original request by comparing the value of the HTTP header called `esc-transaction-id` to the one returned in the initial status response.

ESC Clients are expected to provide a REST implementation for all the callbacks.

### Media Type

All POST, DELETE, and PUT operations that require request and/or response body must provide a header parameter for media type. Note, that ESC is currently tested to accept and return XML.

### Internal IDs

In some of the ESC REST API, there is a path parameter called `"*_internal_id"` where the asterisk is replaced by the resource type. ESC

Clients must provide an internal id for its resource it wants to create. The internal ids are enforced so the ESC Clients are able to refer to them in future operations such as GET, PUT, or DELETE requests.

It is recommended that the ESC Client generated internal ids within the scope of the ESC.

NOTE: Reusing (recycling) of internal ids is not recommended. This practice can lead to confusion in trouble shooting.

## Body

In some of the POST, PUT, and DELETE requests you are required to provide a JSON/XML body. Depending on the resource, you will need to provide certain fields in the body to successfully execute a request. Check the body section for each API (bottom of page) to see what fields are supported.

## String Restrictions in Body/Internal IDs

### Body

Some string parameters/fields, such as id, name, are restricted by the underlying cloud provider, such as Openstack. The allowable range for these parameters is included in the Create section for each resource, in this notation: String minimum-length . . maximum length, for example, for Tenant name: String 1 . . 64

Generally, names are either 1..64 (for Tenant/User) or 1..255 (for Network/Subnet/Flavor/Image/VM).

### Internal ID

Ids are restricted to 72 characters such as internal\_id 1..72 and external\_id 1..72.

Internal id naming must conform the standards stated in RFC 3986 Uniform Resource Identifier (URI): Generic Syntax. See sections related to URI encoding, sections 2.2 Reserved Characters and 2.3 Unreserved Characters. ESC will accept an internal id specified in the request URL form the unreserved character set: unreserved = ALPHA / DIGIT / "-" / "." / "\_"

NOTE: the tilde “~” is not supported in the internal id character set.

The internal id received in a request must be unique within the scope of all the domain resources (tenant, network, subnet, service catalog, deployment ) For example, you cannot create 2 networks with the same internal\_network\_id, even if the networks are created under different tenants.

## REST API Security

## REST Authentication

### Overview

ESC REST API uses http basic access authentication where the ESC client will have to provide a username and password when making ESC REST requests. The user name and password will be encoded with Base64 in transit, but not encrypted or hashed. HTTPS will be used in conjunction with Basic Authentication to provide the encryption.

### Setup

By default, the REST authentication is disabled. To enable it the pass the argument `--enable-auth` to ESC `bootvm.py`, ESC installation script.

### Username and Password

The REST interface has only one default username/password (admin credentials).

The REST password can be updated using `escadm` tool from the ESC VM CLI:

```
$ escadm rest set --username admin --password test123
```

The REST password can be reset using `escadm` tool from the ESC VM CLI:

```
$ escadm rest set
```

You can also update the password through the REST API:

[http://\[ESCVM\\_IP\]:8080/ESCManager/v0/authentication/setpassword?userName=yourUsername&password=yourPassword](http://[ESCVM_IP]:8080/ESCManager/v0/authentication/setpassword?userName=yourUsername&password=yourPassword)

### Sending an Authorized Request

To send an authorized request an ESC client should send the request with the following header:

```
Authorization: Basic <hashed-pass>
```

where is the Base64 encoded string of the default username/password

Most libraries and web clients have an interface for simply providing the username/password and the app will encode the username/password and add the HTTP Basic Auth header.

Example using the default admin credentials:

```
http://[ESCVM_IP]:8080/ESCManager/v0/tenants/
```

Headers:

Content-Type      application/xml

Authorization     Basic <hashed-pass>

## REST Https Support

### Overview

ESC supports https communication over port 8443. ESC will generate a self-signed certificate that the client will need to trust to get the https communication going.

### Setup

By default, the REST https is disabled and restricted to localhost. To enable it pass the argument --enable-https-rest to ESC bootvm.py, ESC installation script.

### HA Considerations

In an HA setup every ESC HA VM will have its own self-signed generated certificate. Upon an HA switchover, ESC north bound client will be required to re-accept the new certificate it will be served.

## REST API List

### 01-Tenants : Manage ESC tenants

GET /v0/tenants

[Get all tenants](#)

#### Implementation Notes

All tenants in ESC will be returned as a List

## Response Class (Status 200)

Model | Model Schema

```
[
  {
    "internal_tenant_id": "string",
    "external_tenant_id": "string",
    "name": "string",
    "enabled": true,
    "vim_mapping": true,
    "extensions": {
      "extension": [
        {
          "name": "string",
          "properties": {
            "property": [
              {
                "name": "string",
                "value": "string"
              }
            ]
          }
        }
      ],
      "containers": {
        "container": [
          {
            "name": "string",
            "properties": {
              "property": [
                {
                  "name": "string",
                  "value": "string"
                }
              ]
            }
          }
        ]
      }
    }
  }
]
```

```
    }
  }
]
},
"event_type": "CREATE_TENANT",
"managed_resource": true
}
]
```

Response Content Type

Try it out!

**DELETE** /v0/tenants/{internal\_tenant\_id}

Delete a tenant from ESC

### Implementation Notes

Delete a tenant from ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Tenant deletion request accepted		
400	Invalid Request		
404	Invalid Tenant Id		

Try it out!

**GET** /v0/tenants/{internal\_tenant\_id} [Get a tenant by its name](#)

### Implementation Notes

Only tenants created using ESC will be checked

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Get request accepted		
404	Invalid Request - Invalid Tenant ID		

Try it out!

**POST** /v0/tenants/{internal\_tenant\_id} [Create a tenant in ESC](#)



## Implementation Notes

Creates a tenant which will be managed by ESC and can be used for creating other resources under it. Example request:

```
POST /v0/tenants/tentest HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Callback: http://127.0.0.1:9010/
Body:
<tenant xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <name>example-tenant</name>
  <managed_resource>true</managed_resource>
</tenant>
```

After the tenant creation operation is done, ECS will send a callback to the client with the result. Example callback:

```
HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: Tenant successfully created
<tenant xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <enabled>true</enabled>
  <event_type>CREATE_TENANT</event_type>
  <external_tenant_id>484c620285ea4f588227ff3396215980</external_tenant_id>
  <internal_tenant_id>tentest</internal_tenant_id>
  <name>example-tenant</name>
  <managed_resource>true</managed_resource>
</tenant>
```

## Parameters

Parameter	Value	Description	Parameter	Data Type
-----------	-------	-------------	-----------	-----------

Type

body

(required)

Tenant configuration  
payload

body

Model | Model Schema

Parameter content type:

application/xml



```
{
  "internal_tenant_id": "string",
  "external_tenant_id": "string",
  "name": "string",
  "enabled": true,
  "vim_mapping": true,
  "extensions": {
    "extension": [
      {
        "name": "string",
        "properties": {
          "property": [
            {
              "name": "string",
              "value": "string"
            }
          ]
        }
      }
    ]
  },
  "containers": {
    "container": [
      {
        "name": "string",
        "properties": {
          "property": [
            {
              "name": "string",
              "value": "string"
            }
          ]
        }
      }
    ]
  }
}
```

```

    ]
  },
  "event_type": "CREATE_TENANT",
  "managed_resource": true
}

```

Click to set as parameter value

<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Tenant creation request accepted		
400	Invalid Request		
409	Conflicting Request		

Try it out!

**PUT** /v0/tenants/{internal\_tenant\_id}

Update a tenant from ESC

### Implementation Notes

Update a tenant from ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid black; padding: 5px; width: fit-content;">(required)</div> <p>Parameter content type:</p> <input type="text" value="application/xml"/>	<b>Tenant payload</b>	body	<div style="display: flex; border-bottom: 1px solid gray;"> <span style="border-right: 1px solid gray; padding-right: 5px;">Model</span> <span>Model Schema</span> </div> <pre style="background-color: #fff9c4; padding: 10px;"> {   "internal_tenant_id": "string",   "external_tenant_id": "string",   "name": "string",   "enabled": true,   "vim_mapping": true,   "extensions": {     "extension": [       {         "name": "string",         "properties": {           "property": [             {               "name": "string",               "value": "string"             }           ]         }       },       {         "containers": {           "container": [             {               "name": "string",               "properties": {                 "property": [                   {                     "name": "string",                     "value": "string"                   }                 ]               }             }           ]         }       }     ]   } } </pre>

Parameter	Value	Description	Parameter Type	Data Type
				<pre> } ] }, "event_type": "CREATE_TENANT", "managed_resource": true } </pre> <p>Click to set as parameter value</p>
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Tenant update request accepted		
400	Invalid Request		
404	Invalid Tenant Id		
500	Internal Server Error		

[Try it out!](#)

## 02-Networks : Manage ESC networks

<b>GET</b>	/v0/{internalTenantId}/networks	Get all networks under a tenant
------------	---------------------------------	---------------------------------

## Implementation Notes

All networks under that tenant in ESC will be returned as a List

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network (max length 72 characters)</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

Try it out!

DELETE

/v0/{internalTenantId}/networks/{internalNetworkId}

Delete a network from an existing tenant in ESC

## Implementation Notes

Delete a network from ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalNetworkId</b>	<input type="text" value="(required)"/>	<b>Internal network ID (max length 72 characters)</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network (max length 72 characters)</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid request		
500	Internal server error		

[Try it out!](#)

**GET** `/v0/{internalTenantId}/networks/{internalNetworkId}` [Get a network by its tenant and internal network ID](#)

**Implementation Notes**  
Only networks created using ESC will be checked

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network (max length 72 characters)</b>	path	string
<b>internalNetworkId</b>	<input type="text" value="(required)"/>	<b>Internal network ID (max length 72 characters)</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Network Id not found		

Try it out!

POST

/v0/{internalTenantId}/networks/{internalNetworkId}

Create a network under an existing tenant in ESC

## Implementation Notes

Create a network in ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network (max length 72 characters)</b>	path	string
<b>internalNetworkId</b>	<input type="text" value="(required)"/>	<b>Internal network ID (max length 72 characters)</b>	path	string



Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">(required)</div> <p>Parameter content type:</p> <input type="text" value="application/xml"/>	<b>Network configuration payload</b>	body	Model   Model Schema <pre> {   "locator": {     "vim_id": "string",     "vim_type": "OPENSTACK",     "vim_project": "string",     "vim_project_id": "string",     "vim_region": "string",     "vim_vdc": "string"   },   "vimNetName": "string",   "internal_tenant_id": "string",   "internal_network_id": "string",   "internal_deployment_id": "string",   "name": "string",   "shared": true,   "external_tenant_id": "string",   "tenant_name": "string",   "external_network_id": "string",   "subnet": [     "string"   ],   "admin_state_up": true,   "locators": {     "switch_name": "string",     "datacenter": "string"   },   "event_type": "CREATE_NETWORK",   "provider_segmentation_id": "string",   "router_external": true,   "provider_physical_network": "string",   "provider_network_type": "string",   "vlan_id": 0,   "switch_name": "string", </pre>

Parameter	Value	Description	Parameter Type	Data Type
				<pre>"number_of_ports": 0 }</pre> <p>Click to set as parameter value</p>
<b>Response Messages</b>				
HTTP Status Code	Reason	Response Model	Headers	
200	Request accepted			
400	Invalid request			
404	Network Id not found			
409	Conflicting request			
503	Unable to query VIM			

Try it out!

### 03-Subnets : Manage ESC subnets

DELETE	/v0/{internalTenantId}/subnets/{internalSubnetId}	Delete a subnet from an existing tenant in ESC		
<b>Implementation Notes</b>				
Delete a subnet from ESC				
<b>Parameters</b>				
Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the subnet (max length 72 characters)</b>	path	string
<b>internalSubnetId</b>	<input type="text" value="(required)"/>	<b>Internal subnet ID (max length 72 characters)</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	Not found		
500	Internal server error		

[Try it out!](#)

**GET** /v0/{internalTenantId}/subnets/{internalSubnetId} [Get a subnet by its tenant and internal subnet ID](#)

### Implementation Notes

Only subnets created using ESC will be checked

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the subnet (max length 72 characters)</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>internalSubnetId</b>	<input type="text" value="(required)"/>	<b>Internal subnet ID (max length 72 characters)</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Not found		

[Try it out!](#)

**POST** `/v0/{internalTenantId}/subnets/{internalSubnetId}` Create a subnet under an existing tenant in ESC

**Implementation Notes**  
Create a subnet in ESC

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the subnet (max length 72 characters)</b>	path	string
<b>internalSubnetId</b>	<input type="text" value="(required)"/>	<b>Internal subnet ID (max length 72 characters)</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">(required)</div> <p>Parameter content type:  <input type="text" value="application/xml"/> </p>	<b>Subnet configuration payload</b>	body	<div style="display: flex; justify-content: space-between;"> <span>Model</span> <span>Model Schema</span> </div> <pre style="background-color: #fff9c4; padding: 10px; border: 1px solid #ccc;"> {   "networkFromCdg": {     "locator": {       "vim_id": "string",       "vim_type": "OPENSTACK",       "vim_project": "string",       "vim_project_id": "string",       "vim_region": "string",       "vim_vdc": "string"     },     },     "vimNetName": "string",     "internal_tenant_id": "string",     "internal_network_id": "string",     "internal_deployment_id": "string",     "name": "string",     "shared": true,     "external_tenant_id": "string",     "tenant_name": "string",     "external_network_id": "string",     "subnet": [       "string"     ],     },     "admin_state_up": true,     "locators": {       "switch_name": "string",       "datacenter": "string"     },     },     "event_type": "CREATE_NETWORK",     "provider_segmentation_id": "string",     "router_external": true,     "provider_physical_network": "string",     "provider_network_type": "string",     "vlan_id": 0,   } </pre>

Parameter	Value	Description	Parameter Type	Data Type
				<pre> "switch_name": "string", "number_of_ports": 0 }, "internal_tenant_id": "string", "external_tenant_id": "string", "internal_network_id": "string", "external_network_id": "string", "internal_subnet_id": "string", "external_subnet_id": "string", "name": "string", "allocation_pools": [ { "start": "string", "end": "string" } ], "gateway_ip": "string", "ip_version": 0, "cidr": "string", "enable_dhcp": true, "event_type": "CREATE_SUBNET" } </pre>

[Click to set as parameter value](#)

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	Not found		
409	Conflicting Request		
500	Internal server error		

GET

/v0/{internalTenantId}/subnets

Get all subnets under a network

### Implementation Notes

All subnet under that network in ESC will be returned as a List

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
internalNetworkId	<input type="text"/>	Internal network ID of the subnet (max length 72 characters)	header	string
internalTenantId	<input type="text" value="(required)"/>	Internal tenant ID of the subnet (max length 72 characters)	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Not found		

[Try it out!](#)

## 04-Images : Manage ESC images

DELETE

/v0/images/{internalImageId}

Delete an image from ESC

### Implementation Notes

Delete an image from ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalImageId</b>	<input type="text" value="(required)"/>	<b>Internal image ID (max length 72 characters)</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
500	Internal server error		

Try it out!

GET

/v0/images/{internalImageId}

Get the image by the internal image ID

## Implementation Notes

Images not created by ESC will not be checked

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalImageId</b>	<input type="text" value="(required)"/>	<b>Internal image ID (max length 72 characters)</b>	path	string

## Response Messages



HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	Image Id not found		

Try it out!

**POST** /v0/images/{internalImageId} [Create an image in ESC](#)

### Implementation Notes

Creates an Image(Template) which will be managed by ESC and can be used for creating instance VMs.

Example request:

```
POST /v0/images/my_image_id HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Callback: http://127.0.0.1:9010/
Body:

<?xml version="1.0"?>
<image xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <name>uLinuxImg</name>
  <src>http://VAR_FILE_SERVER_IP/share/images/myULinux.qcow2</src>
  <disk_bus>virtio</disk_bus>
</image>
```

After the Image creation operation is done, ESC will send a callback to the client with the result.

Example callback:

```
HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: Image successfully created
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<image xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <disk_bus>virtio</disk_bus>
  <e1000_net>>false</e1000_net>
  <event_type>CREATE_IMAGE</event_type>
  <external_image_id>5021843b-fcc5-2c1b-8fc4-935147f95872</external_image_id>
  <external_tenant_id>SystemAdminTenantId</external_tenant_id>
  <imageisenabled>>true</imageisenabled>
  <name>uLinuxImg</name>
  <internal_image_id>my_image_id</internal_image_id>
  <internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
  <visibility>public</visibility>
  <src>http://VAR_FILE_SERVER_IP/share/images/myULinux.qcow2</src>
</image>
```

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalImageId</b>	<input type="text" value="(required)"/>	<b>Internal image ID (max length 72 characters)</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	(required)	<b>ESC datamodel with the image to create</b>	body	string

Parameter content type:

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
409	Conflicting Request		
500	Internal server error		

[Try it out!](#)

**GET** /v0/images [Get all images](#)

### Implementation Notes

All images created in ESC will be returned as a List

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

[Try it out!](#)

## 05-Flavors : Manage ESC flavors

**GET** /v0/flavors Get all flavors

**Implementation Notes**  
All flavors created in ESC will be returned as a List

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

[Try it out!](#)

**DELETE** /v0/flavors/{internalFlavorId} Delete a flavor from ESC

**Implementation Notes**  
Delete a flavor from ESC

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalFlavorId</b>	<input type="text" value="(required)"/>	<b>Internal flavor ID (max length 72 characters)</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

HTTP Status Code	Reason	Response Model	Headers
404	Flavor Id not found		
500	Internal server error		

Try it out!

**GET** /v0/flavors/{internalFlavorId} Get the flavor by the internal flavor ID

**Implementation Notes**  
 Flavors not created by ESC will not be checked

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>internalFlavorId</b>	<input type="text" value="(required)"/>	<b>Internal flavor ID (max length 72 characters)</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Flavor Id not found		
406	Operation not supported on VIM/Flavor part of service registration		

Try it out!

**POST** /v0/flavors/{internalFlavorId} Create a flavor in ESC

## Implementation Notes

Creates a Flavor which could be used for creating instance VMs.

Example request:

```
POST /v0/flavors/my_flavor_id HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Callback: http://127.0.0.1:9010/
Body:
<?xml version="1.0"?>
<flavor xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <name>lightVMFlavor</name>
  <vcpus>2</vcpus>
  <memory_mb>2048</memory_mb>
</flavor>
```

After the Flavor creation operation is done, ESC will send a callback to the client with the result.


Example callback:

```
HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: Flavor creation completed successfully created
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flavor xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ephemeral_disk_mb>0</ephemeral_disk_mb>
  <event_type>CREATE_FLAVOR</event_type>
  <external_flavor_id>fb63c114-de1a-40d8-abae-fa6011abd150</external_flavor_id>
  <name>lightVMFlavor</name>
  <internal_flavor_id>my_flavor_id</internal_flavor_id>
```

```
<memory_mb>2048</memory_mb>
<vcpus>2</vcpus>
</flavor>
```

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalFlavorId</b>	<input type="text" value="(required)"/>	<b>Internal flavor ID (max length 72 characters)</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> <div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div>	<b>ESC datamodel with the flavor to create</b>	body	string

Parameter content type:  

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	Flavor Id not found		
406	Operation not supported on VIM/Flavor part of service registration		
409	Conflicting Request		
500	Internal server error		

## 06-Deployments : Manage ESC deployments

**GET** /v0/deployments Find all deployments

**Implementation Notes**  
All deployments created in ESC will be returned as a List

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

[Try it out!](#)

**GET** /v0/deployments/config/{internal\_deployment\_id} Find deployment configuration data by its internal ID

**Implementation Notes**  
Deployment configuration data will be returned if found in CDB

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal ID of the deployment (max length 1024 characters)</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		



HTTP Status Code	Reason	Response Model	Headers
404	Deployment Id not found		

Try it out!

**DELETE** /v0/deployments/{internal\_deployment\_id} Delete a deployment from ESC

### Implementation Notes

Delete a deployment from ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal ID of the deployment (max length 72 characters)</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	Deployment Id not found		

Try it out!

GET /v0/deployments/{internal\_deployment\_id}

Find a deployment by its internal ID

### Implementation Notes

A deployment will be returned if found in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal ID of the deployment (max length 1024 characters)</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Deployment Id not found		

Try it out!

POST /v0/deployments/{internal\_deployment\_id}

Creates a deployment

### Implementation Notes

Creates a Simple deployment of Instance VMs.

**NOTE:** As deployment requirements vary a lot, please see ESC User guide for datamodel excerpts of specific functionality you may desire that's not displayed in the simple example below.

Example request:

```
POST /v0/deployments/my_dep_id HTTP/1.1
Content-Type: application/xml
Accept: application/xml
```

Callback: http://127.0.0.1:9010/

Callback-ESC-Events: http://127.0.0.1:9010/

Body:

```
<esc_datamodel xmlns="http://www.cisco.com/esc/esc" xmlns:ns2="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <tenants>
    <tenant>
      <name>admin</name>
      <deployments>
        <deployment>
          <name>MyDemoDepName</name>
          <vm_group>
            <name>ASA-1</name>
            <bootup_time>300</bootup_time>
            <reboot_time>100</reboot_time>
            <recovery_wait_time>10</recovery_wait_time>
            <image>ASA</image>
            <interfaces>
              <interface>
                <nicid>1</nicid>
                <network>MgtNetwork</network>
                <ip_address>18.0.0.16</ip_address>
              </interface>
            </interfaces>
            <scaling>
              <min_active>1</min_active>
              <max_active>1</max_active>
              <elastic>true</elastic>
              <static_ip_address_pool>
                <network>MgtNetwork</network>
                <ip_address>18.0.0.16</ip_address>
              </static_ip_address_pool>
            </scaling>
          <vm_group>
            <kpi_data>
```

```
<kpi>
  <event_name>VM_ALIVE</event_name>
  <metric_value>1</metric_value>
  <metric_cond>GT</metric_cond>
  <metric_type>UINT32</metric_type>
  <metric_collector>
    <type>ICMPPing</type>
    <nicid>1</nicid>
    <poll_frequency>3</poll_frequency>
    <polling_unit>seconds</polling_unit>
    <continuous_alarm>>false</continuous_alarm>
  </metric_collector></kpi>
</kpi_data>
<rules>
  <admin_rules>
    <rule>
      <event_name>VM_ALIVE</event_name>
      <action>"ALWAYS log"</action>
      <action>"TRUE esc_vm_alive_notification"</action>
    </rule>
  </admin_rules>
</rules>
<config_data>
  <configuration>
    <dst>day0-config</dst>
    <file>http://MY_FILE_SERVER_IP:/share/images/day0-configASAv</file>
  </configuration>
  <configuration>
    <dst>idtoken</dst>
    <file>http://MY_FILE_SERVER_IP:/share/images/asa-smart.config</file>
  </configuration>
</config_data>
<placement>
  <type>zone_host</type>
  <zone>MY_CLUSTER1</zone>
```

```

    </placement>
  <volumes>
    <volume>
      <name>MY_DATASTORE1</name>
      <volid>1</volid>
    </volume>
  </volumes>
</vm_group>
</deployment>
</deployments>
</tenant>
</tenants>
</esc_datamodel>

```

While the Deployment operation is being processed successfully, ESC will send a VM\_DEPLOYED callback for each VM created on the VIM as part of this deployment

```

HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: VM Deployed

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <datacenter>
    <default>false</default>
  </datacenter>
  <deployment_details>
    <host_uuid>host-14882</host_uuid>
    <host_name>10.85.103.13</host_name>
    <vm_uuid>50217d2d-8afb-41c2-cffd-d76b1239f51d</vm_uuid>
    <interfaces>
      <interface>

```

```

<network_uuid>MgtNetwork</network_uuid>
<ip_address>18.0.0.16</ip_address>
<mac_address>00:50:56:a1:6a:12</mac_address>
<nic_id>1</nic_id>
<port_forwarding></port_forwarding>
<port_uuid></port_uuid>
<security_groups></security_groups>
<subnet_uuid></subnet_uuid>
<type>virtual</type>
</interface>
</interfaces>
<vm_group_name>ASA-1</vm_group_name>
<vm_name>Sanity-vmware-de_ASA-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d</vm_name>
<vm_state_machine_state>VM_INERT_STATE</vm_state_machine_state>
</deployment_details>
<deployment_name>MyDemoDepName</deployment_name>
<deployment_stage>VM_DEPLOYED</deployment_stage>
<external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
<external_tenant_id>SystemAdminTenantId</external_tenant_id>
<internal_deployment_id>my_dep_id</internal_deployment_id>
<internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
<service_state_machine_state>SERVICE_DEPLOYING_STATE</service_state_machine_state>
</deployment>

```

While the Deployment operation is being processed successfully, ESC will send a VM\_ALIVE callback for each monitorable/pingable VM created on the VIM as part of this deployment

```

HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: VM_Alive event received, VM ID: [Sanity-vmware-de_ASA-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d]

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

```
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <datacenter>
    <default>>false</default>
  </datacenter>
  <deployment_details>
    <host_uuid>host-14882</host_uuid>
    <host_name>10.85.103.13</host_name>
    <vm_uuid>50217d2d-8afb-41c2-cffd-d76b1239f51d</vm_uuid>
    <interfaces>
      <interface>
        <network_uuid>MgtNetwork</network_uuid>
        <ip_address>18.0.0.16</ip_address>
        <mac_address>00:50:56:a1:6a:12</mac_address>
        <nic_id>1</nic_id>
        <port_forwarding></port_forwarding>
        <port_uuid></port_uuid>
        <security_groups></security_groups>
        <subnet_uuid></subnet_uuid>
        <type>virtual</type>
      </interface>
    </interfaces>
    <vm_group_name>ASA-1</vm_group_name>
    <vm_name>Sanity-vmware-de_ASA-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d</vm_name>
    <vm_state_machine_state>VM_ALIVE_STATE</vm_state_machine_state>
  </deployment_details>
  <deployment_name>MyDemoDepName</deployment_name>
  <deployment_stage>VM_ALIVE</deployment_stage>
  <external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
  <external_tenant_id>SystemAdminTenantId</external_tenant_id>
  <internal_deployment_id>my_dep_id</internal_deployment_id>
  <internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
  <service_state_machine_state>SERVICE_INERT_STATE</service_state_machine_state>
</deployment>
```

After the Deployment operation is done, ESC will send a SERVICE\_ALIVE callback to the client with the result.

Example callback:

```
HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: Service group deployment completed successfully!

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <datacenter>
    <default>>false</default>
  </datacenter>
  <deployment_name>MyDemoDepName</deployment_name>
  <deployment_stage>SERVICE_ALIVE</deployment_stage>
  <external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
  <external_tenant_id>SystemAdminTenantId</external_tenant_id>
  <internal_deployment_id>my_dep_id</internal_deployment_id>
  <internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
  <service_state_machine_state>SERVICE_ACTIVE_STATE</service_state_machine_state>
</deployment>
```

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC event notifications</b>	header	string



Parameter	Value	Description	Parameter Type	Data Type
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal ID of the deployment (max length 72 characters)</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/>	<b>ESC datamodel to deploy</b>	body	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	An associated resource not found		
406	Not Acceptable		
409	Conflicting Request		
500	Internal server error		
503	Failed to query VIM		

Try it out!

## Implementation Notes

Updates an existing deployment. Updates could be addition or removal of VM groups, interfaces, networks, KPIs and Rules, Day 0 config.

After the service update operation is done, ESC will send a callback to the client with the result.

Example request of adding VM group:

```
PUT /v0/deployments/my_dep_id HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Callback: http://127.0.0.1:9010/
Callback-ESC-Events: http://127.0.0.1:9010/
```

Body:

```
<esc_datamodel xmlns:ns2="urn:ietf:params:xml:ns:netconf:notification:1.0" xmlns="http://www.cisco.com/esc/esc">
  <tenants>
    <tenant>
      <name>admin</name>
      <deployments>
        <deployment>
          <name>MyDemoDepName</name>
          <vm_group>
            <name>ASA-1</name>
            <bootup_time>300</bootup_time>
            <recovery_wait_time>0</recovery_wait_time>
            <image>ASA</image>
            <interfaces>
              <interface>
                <nicid>1</nicid>
                <network>MgtNetwork</network>
                <ip_address>18.0.0.16</ip_address>
              </interface>
            </interfaces>
          </vm_group>
        </deployment>
      </deployments>
    </tenant>
  </tenants>
</esc_datamodel>
```

```
<scaling>
  <min_active>1</min_active>
  <max_active>1</max_active>
  <elastic>>true</elastic>
  <static_ip_address_pool>
    <network>MgtNetwork</network>
    <ip_address>18.0.0.16</ip_address>
  </static_ip_address_pool>
</scaling>
<kpi_data>
  <kpi>
    <event_name>VM_ALIVE</event_name>
    <metric_value>1</metric_value>
    <metric_cond>GT</metric_cond>
    <metric_type>UINT32</metric_type>
    <metric_collector>
      <type>ICMPPing</type>
      <nicid>1</nicid>
      <poll_frequency>3</poll_frequency>
      <polling_unit>seconds</polling_unit>
      <continuous_alarm>>false</continuous_alarm>
    </metric_collector></kpi>
</kpi_data>
<rules>
  <admin_rules>
    <rule>
      <event_name>VM_ALIVE</event_name>
      <action>"ALWAYS log"</action>
      <action>"TRUE esc_vm_alive_notification"</action>
    </rule>
  </admin_rules>
</rules>
<config_data>
  <configuration>
    <dst>day0-config</dst>
```

```
<file>http://MY_FILE_SERVER_IP:/share/images/day0-configASAv</file>
</configuration>
<configuration>
  <dst>idtoken</dst>
  <file>http://MY_FILE_SERVER_IP:/share/images/asa-smart.config</file>
</configuration>
</config_data>
<placement>
  <type>zone_host</type>
  <zone>MY_CLUSTER1</zone>
</placement>
<volumes>
<volume>
  <name>MY_DATASTORE1</name>
  <valid>1</valid>
</volume>
</volumes>
</vm_group>
<name>CSR-1</name>
<bootup_time>300</bootup_time>
<recovery_wait_time>0</recovery_wait_time>
<image>ASA</image>
<interfaces>
<interface>
  <nicid>1</nicid>
  <network>MgtNetwork</network>
  <ip_address>18.0.0.16</ip_address>
</interface>
</interfaces>
<scaling>
  <min_active>1</min_active>
  <max_active>1</max_active>
  <elastic>true</elastic>
  <static_ip_address_pool>
    <network>MgtNetwork</network>
```

```
<ip_address>18.0.0.16</ip_address>
</static_ip_address_pool>
</scaling>
<kpi_data>
  <kpi>
    <event_name>VM_ALIVE</event_name>
    <metric_value>1</metric_value>
    <metric_cond>GT</metric_cond>
    <metric_type>UINT32</metric_type>
    <metric_collector>
      <type>ICMPPing</type>
      <nicid>1</nicid>
      <poll_frequency>3</poll_frequency>
      <polling_unit>seconds</polling_unit>
      <continuous_alarm>>false</continuous_alarm>
    </metric_collector></kpi>
  </kpi_data>
  <rules>
    <admin_rules>
      <rule>
        <event_name>VM_ALIVE</event_name>
        <action>"ALWAYS log"</action>
        <action>"TRUE esc_vm_alive_notification"</action>
      </rule>
    </admin_rules>
  </rules>
  <config_data>
    <configuration>
      <dst>day0-config</dst>
      <file>http://MY_FILE_SERVER_IP:/share/images/day0-configASAv</file>
    </configuration>
    <configuration>
      <dst>idtoken</dst>
      <file>http://MY_FILE_SERVER_IP:/share/images/asa-smart.config</file>
    </configuration>
  </config_data>

```

```
</config_data>
<placement>
  <type>zone_host</type>
  <zone>MY_CLUSTER1</zone>
</placement>
<volumes>
<volume>
  <name>MY_DATASTORE1</name>
  <volid>1</volid>
</volume>
</volumes>

</deployment>
</deployments>
</tenant>
</tenants>
</esc_datamodel>
```

While the Deployment Update operation is being processed successfully, ESC will send a VM\_DEPLOYED callback for each VM created on the VIM as part of this update request.

```
HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: VM Deployed
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <datacenter><default>false</default></datacenter>
  <deployment_details>
    <host_uuid>host-14882</host_uuid>
    <host_name>10.85.103.13</host_name>
    <vm_uuid>50217d2d-8afb-41c2-cffd-d76b1239f51e</vm_uuid>
```

```

<interfaces>
<interface>
<network_uuid>MgtNetwork</network_uuid>
<ip_address>18.0.0.17</ip_address>
<mac_address>00:50:56:a1:6a:13</mac_address>
<nic_id>1</nic_id>
<port_forwarding></port_forwarding>
<port_uuid></port_uuid>
<security_groups></security_groups>
<subnet_uuid></subnet_uuid>
<type>virtual</type>
</interface>
</interfaces>
<vm_group_name>CSR-1</vm_group_name>
<vm_name>Sanity-vmware-de_CSR-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d</vm_name>
<vm_state_machine_state>VM_INERT_STATE</vm_state_machine_state>
</deployment_details>
<deployment_name>MyDemoDepName</deployment_name>
<deployment_stage>VM_DEPLOYED</deployment_stage>
<external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ad</external_deployment_id>
<external_tenant_id>SystemAdminTenantId</external_tenant_id>
<internal_deployment_id>my_dep_id</internal_deployment_id>
<internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
<service_state_machine_state>SERVICE_DEPLOYING_STATE</service_state_machine_state>
</deployment>

```

While the Deployment Update operation is being processed successfully, ESC will send a VM\_ALIVE callback for each monitorable/pingable VM on the VIM created as part of this update request.

```

HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: VM_Alive event received, VM ID: [Sanity-vmware-de_CSR-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d]

```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <datacenter><default>false</default></datacenter>
  <deployment_details>
    <host_uuid>host-14882</host_uuid>
    <host_name>10.85.103.13</host_name>
    <vm_uuid>50217d2d-8afb-41c2-cffd-d76b1239f51e</vm_uuid>
    <interfaces>
      <interface>
        <network_uuid>MgtNetwork</network_uuid>
        <ip_address>18.0.0.17</ip_address>
        <mac_address>00:50:56:a1:6a:13</mac_address>
        <nic_id>1</nic_id>
        <port_forwarding></port_forwarding>
        <port_uuid></port_uuid>
        <security_groups></security_groups>
        <subnet_uuid></subnet_uuid>
        <type>virtual</type>
      </interface>
    </interfaces>
    <vm_group_name>CSR-1</vm_group_name>
    <vm_name>Sanity-vmware-de_CSR-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d</vm_name>
    <vm_state_machine_state>VM_ALIVE_STATE</vm_state_machine_state>
  </deployment_details>
  <deployment_name>MyDemoDepName</deployment_name>
  <deployment_stage>VM_ALIVE</deployment_stage>
  <external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
  <external_tenant_id>SystemAdminTenantId</external_tenant_id>
  <internal_deployment_id>my_dep_id</internal_deployment_id>
  <internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
  <service_state_machine_state>SERVICE_INERT_STATE</service_state_machine_state>
</deployment>
```



After the Deployment Update operation is done, ESC will send a SERVICE\_UPDATED callback to the client with the result.

Example callback:

```
HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: Service group update completed successfully

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<datacenter><default>>false</default></datacenter>
<deployment_name>MyDemoDepName</deployment_name>
<deployment_stage>SERVICE_UPDATED</deployment_stage>
<external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
<external_tenant_id>SystemAdminTenantId</external_tenant_id>
<internal_deployment_id>my_dep_id</internal_deployment_id>
<internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
<service_state_machine_state>SERVICE_ACTIVE_STATE</service_state_machine_state>
</deployment>
```

If a VM specific resource like interface, is updated, then a VM\_UPDATED followed by a SERVICE\_UPDATED notification is received.

Example callback:

```
HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: VM has been updated successfully. vm: VM ID: [Sanity-vmware-de_CSR-1_0_b323effd-7f70-4055-b11e-0ef66f0a75

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<datacenter><default>>false</default></datacenter>
```

```
<deployment_details>
<host_uuid>host-14882</host_uuid>
<host_name>10.85.103.13</host_name>
<vm_uuid>50217d2d-8afb-41c2-cffd-d76b1239f51e</vm_uuid>
<interfaces>
<interface>
<network_uuid>MgtNetwork</network_uuid>
<ip_address>18.0.0.17</ip_address>
<mac_address>00:50:56:a1:6a:13</mac_address>
<nic_id>1</nic_id>
<port_forwarding></port_forwarding>
<port_uuid></port_uuid>
<security_groups></security_groups>
<subnet_uuid></subnet_uuid>
<type>virtual</type>
</interface>
<interface>
<network_uuid>MgtNetwork2</network_uuid>
<ip_address>18.0.1.18</ip_address>
<mac_address>00:50:56:a1:6b:14</mac_address>
<nic_id>2</nic_id>
<port_forwarding></port_forwarding>
<port_uuid></port_uuid>
<security_groups></security_groups>
<subnet_uuid></subnet_uuid>
<type>virtual</type>
</interface>
</interfaces>
<vm_group_name>CSR-1</vm_group_name>
<vm_name>Sanity-vmware-de_CSR-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d</vm_name>
<vm_state_machine_state>VM_ALIVE_STATE</vm_state_machine_state>
</deployment_details>
<deployment_name>MyDemoDepName</deployment_name>
<deployment_stage>VM_ALIVE</deployment_stage>
<external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
```

```
<external_tenant_id>SystemAdminTenantId</external_tenant_id>
<internal_deployment_id>my_dep_id</internal_deployment_id>
<internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
<service_state_machine_state>SERVICE_INERT_STATE</service_state_machine_state>
</deployment>
```

Followed by:

```
HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: Service group update completed successfully
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<datacenter><default>>false</default></datacenter>
<deployment_name>MyDemoDepName</deployment_name>
<deployment_stage>SERVICE_UPDATED</deployment_stage>
<external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
<external_tenant_id>SystemAdminTenantId</external_tenant_id>
<internal_deployment_id>my_dep_id</internal_deployment_id>
<internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
<service_state_machine_state>SERVICE_ACTIVE_STATE</service_state_machine_state>
</deployment>
```

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC event notifications</b>	header	string
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal ID of the deployment (max length 72 characters)</b>	path	string
body	<div style="border: 1px solid #ccc; width: 100px; height: 100px; margin: 0 auto;"></div> <p>Parameter content type: <input type="text" value="application/xml"/></p>		body	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	An associated resource not found		
406	Not Acceptable		
409	Conflicting Request		
500	Internal server error		

[Try it out!](#)

**POST** /v0/{internal\_tenant\_id}/deployments/service/{internal\_deployment\_id}

## 07-Deployment Operations : Operations to a deployment

### Implementation Notes

Performs actions like stop, start, reboot, disable\_monitoring or enable\_monitoring on a deployment. Example request:

```
POST /v0/tenants/test-tenant1/deployments/vm/test-dep1-cirros_overid_0_cfeefcee-a1fe-4e94-9ffc-81792b265788 HTTP/1.1
Content-Type: application/xml
Accept: application/json
Callback: http://127.0.0.1:9010/

Body:
<service_operation xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <operation>stop</operation>
</service_operation>
```

While the stop deployment operation is being performed, ESC will send a VM\_STOPPED callback to the client for each VM stopped as part of this request. Example callback:

```
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: 5e47b5ff-bc59-4ece-b15b-8941910def74
ESC-Status-Code: 200
ESC-Status-Message: VM successfully stopped

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <datacenter>
    <default>>false</default>
  </datacenter>
  <deployment_details>
    <host_uuid>17be6eae40795ec068ce77a05fcb1c47f7ac42dfba160261719d4087</host_uuid>
    <host_name>my-ucs-3</host_name>
    <vm_uuid>cb86a952-c030-46f6-96f6-e81069bcea3a</vm_uuid>
```

```

<interfaces>
  <interface>
    <network_uuid>943fda9e-79f8-400c-b442-3506f102721a</network_uuid>
    <gateway>192.168.0.1</gateway>
    <ip_address>192.168.0.135</ip_address>
    <mac_address>fa:16:3e:3c:17:b0</mac_address>
    <netmask>255.255.255.0</netmask>
    <nic_id>0</nic_id>
    <port_forwarding></port_forwarding>
    <port_uuid>263eaf20-2662-4a6b-a1f1-ed76001a1b5c</port_uuid>
    <security_groups></security_groups>
    <subnet_uuid>e313b95c-ca1f-4c81-8d60-c9e721a85d0b</subnet_uuid>
    <type>virtual</type>
  </interface>
</interfaces>
<vm_group_name>overridegrp2</vm_group_name>
<vm_name>test-dep1-cirros_overid_0_cfefcece-a1fe-4e94-9ffc-81792b265788</vm_name>
<vm_state_machine_state>VM_SHUTDOWN_STATE</vm_state_machine_state>
</deployment_details>
<deployment_name>test-dep1-cirros</deployment_name>
<deployment_stage>VM_STOPPED</deployment_stage>
<external_deployment_id>5c278010-ad0f-4e29-bdce-f4b6c6946c38</external_deployment_id>
<external_tenant_id>29f2cd7934dc4311ad3a34d72d1ab7b8</external_tenant_id>
<internal_deployment_id>test-tenant1test-dep1-cirros</internal_deployment_id>
<internal_tenant_id>test-tenant1</internal_tenant_id>
<service_state_machine_state>SERVICE_STOPPED_STATE</service_state_machine_state>
</deployment>

```

After the stop deployment operation is done successfully, ESC will send a SERVICE\_STOPPED callback to the client with the result. Example callback:

```

HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: 5e47b5ff-bc59-4ece-b15b-8941910def74

```

ESC-Status-Code: 200

ESC-Status-Message: Service group suspension completed successfully

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <datacenter>
    <default>>false</default>
  </datacenter>
  <deployment_details>
    <host_uuid>17be6eae40795ec068ce77a05fcb1c47f7ac42dfba160261719d4087</host_uuid>
    <host_name>my-ucs-3</host_name>
    <vm_uuid>cb86a952-c030-46f6-96f6-e81069bcea3a</vm_uuid>
    <interfaces>
      <interface>
        <network_uuid>943fda9e-79f8-400c-b442-3506f102721a</network_uuid>
        <gateway>192.168.0.1</gateway>
        <ip_address>192.168.0.135</ip_address>
        <mac_address>fa:16:3e:3c:17:b0</mac_address>
        <netmask>255.255.255.0</netmask>
        <nic_id>0</nic_id>
        <port_forwarding></port_forwarding>
        <port_uuid>263eaf20-2662-4a6b-a1f1-ed76001a1b5c</port_uuid>
        <security_groups></security_groups>
        <subnet_uuid>e313b95c-ca1f-4c81-8d60-c9e721a85d0b</subnet_uuid>
        <type>virtual</type>
      </interface>
    </interfaces>
    <vm_group_name>overridegrp2</vm_group_name>
    <vm_name>test-dep1-cirros_overid_0_cfeefcee-a1fe-4e94-9ffc-81792b265788</vm_name>
    <vm_state_machine_state>VM_SHUTDOWN_STATE</vm_state_machine_state>
  </deployment_details>
  <deployment_name>test-dep1-cirros</deployment_name>
  <deployment_stage>SERVICE_STOPPED</deployment_stage>
  <external_deployment_id>5c278010-ad0f-4e29-bdce-f4b6c6946c38</external_deployment_id>
  <external_tenant_id>29f2cd7934dc4311ad3a34d72d1ab7b8</external_tenant_id>
</deployment>
```

```

<internal_deployment_id>test-tenant1test-dep1-cirros</internal_deployment_id>
<internal_tenant_id>test-tenant1</internal_tenant_id>
<service_state_machine_state>SERVICE_STOPPED_STATE</service_state_machine_state>
</deployment>

```

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC event notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal deployment ID (max length 72 characters)</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/>	<b>Deployment operation payload specifying the type of operation</b>	body	Model   Model Schema <pre> {   "operation": "stop",   "properties": {} } </pre> Click to set as parameter value

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Deployment operation request accepted		



HTTP Status Code	Reason	Response Model	Headers
400	Missing callback or, Deployment operation is not applicable		
404	Deployment or tenant not found		

Try it out!

## 08-VM Operations : Operations to a VM

**POST** /v0/{internal\_tenant\_id}/deployments/recovery-vm/{vm\_name}

Handle recovery VM operation request with the operation type as payload

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
Callback	<input type="text"/>	Callback URL for notifications	header	string
Callback-ESC-Events	<input type="text"/>	Callback URL for ESC event notifications	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string
<b>vm_name</b>	<input type="text" value="(required)"/>	<b>VM name</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">(required)</div> <p>Parameter content type:  <input type="text" value="application/xml"/></p>	<b>Recovery VM operation payload specifying the type of operation: recovery_do</b>	body	Model   Model Schema <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> <pre>{   "operation": "string",   "checkVmBeforeRecovery": true,   "properties": {} }</pre> </div> <p>Click to set as parameter value</p>

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Recovery VM operation request accepted		
400	Recovery VM operation is not applicable		
404	VM or tenant not found		

[Try it out!](#)

**POST** /v0/{internal\_tenant\_id}/deployments/snapshot-vm/{vm\_name}

Handle VM snapshot operation request with the operation type as payload

### Implementation Notes

vm\_operation\_notes

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for</b>	header	string

Parameter	Value	Description	Parameter Type	Data Type
<b>notifications</b>				
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC event notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string
<b>vm_name</b>	<input type="text" value="(required)"/>	<b>VM name</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/>	<b>VM operation payload specifying the type of operation</b>	body	Model   Model Schema <pre>{   "operation": "snapshot",   "name": "string",   "metadata": {} }</pre> Click to set as parameter value

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	VM snapshot operation request accepted		
400	Missing callback or, VM operation is not applicable		
404	VM or tenant not found		

Try it out!

**POST** /v0/{internal\_tenant\_id}/deployments/vm/{vm\_name} Handle VM operation request with the operation type as payload

## Implementation Notes

Performs actions like stop, start, reboot, disable\_monitoring or enable\_monitoring on an existing VM managed by ESC. Example request:

```
POST /v0/tenants/test-tenant1/deployments/vm/test-dep1-cirros_overid_0_cfeefcee-a1fe-4e94-9ffc-81792b265788 HTTP/1.1
Content-Type: application/xml
Accept: application/json
Callback: http://127.0.0.1:9010/
Callback-ESC-Events: http://127.0.0.1:9010/

Body:
<vm_operation xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <operation>stop</operation>
</vm_operation>
```

After the VM stop operation is done, ESC will send a VM\_STOP\_COMPLETE callback to the client with the result. Example callback:

```
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: 4546c40a-d6e3-4abe-a7e7-51afd4fc5362
ESC-Status-Code: 200
ESC-Status-Message: Successfully stopped VM [test-dep1-cirros_overid_0_cfeefcee-a1fe-4e94-9ffc-81792b265788].

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<esc_event xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <deployment_name>test-dep1-cirros</deployment_name>
  <event_details></event_details>
  <event_type>VM_STOP_COMPLETE</event_type>
  <external_deployment_id>5c278010-ad0f-4e29-bdce-f4b6c6946c38</external_deployment_id>
  <external_tenant_id>29f2cd7934dc4311ad3a34d72d1ab7b8</external_tenant_id>
  <internal_deployment_id>test-tenant1test-dep1-cirros</internal_deployment_id>
  <internal_tenant_id>test-tenant1</internal_tenant_id>
  <vm_source>
```

```

<interfaces>
  <network_uuid>943fda9e-79f8-400c-b442-3506f102721a</network_uuid>
  <gateway>192.168.0.1</gateway>
  <ip_address>192.168.0.135</ip_address>
  <mac_address>fa:16:3e:3c:17:b0</mac_address>
  <netmask>255.255.255.0</netmask>
  <nic_id>0</nic_id>
  <port_forwarding></port_forwarding>
  <port_uuid>263eaf20-2662-4a6b-a1f1-ed76001a1b5c</port_uuid>
  <security_groups></security_groups>
  <subnet_uuid>e313b95c-ca1f-4c81-8d60-c9e721a85d0b</subnet_uuid>
  <type>virtual</type>
</interfaces>
<host_uuid>17be6eae40795ec068ce77a05fcb1c47f7ac42dfba160261719d4087</host_uuid>
<host_name>my-ucs-3</host_name>
<vm_uuid>cb86a952-c030-46f6-96f6-e81069bcea3a</vm_uuid>
<vm_group_name>overridegrp2</vm_group_name>
<vm_name>test-dep1-cirros_overid_0_cfeefcee-a1fe-4e94-9ffc-81792b265788</vm_name>
</vm_source>
</esc_event>

```

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for notifications</b>	header	string
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC event notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>vm_name</b>	<input type="text" value="(required)"/>	<b>VM name</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/>	<b>VM operation payload specifying the type of operation</b>	body	Model   Model Schema <pre>{   "operation": "stop" }</pre> Click to set as parameter value

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	VM operation request accepted		
400	Missing callback or, VM operation is not applicable		
404	VM or tenant not found		

[Try it out!](#)

## 09-Configuration Parameters

**GET** `/v0/config/{category}` Retrieve all ESC config parameters under a category

**Implementation Notes**  
 Retrieve all ESC config parameters under a category

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

Parameter	Value	Description	Parameter Type	Data Type
<b>category</b>	<input type="text" value="(required)"/>	<b>ESC config parameter category</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Get category request accepted.		
400	Invalid category name provided.		
404	Category not found.		

**GET** `/v0/config/{category}/{key}` Retrieve a ESC config parameters under a category, with a given key

**Implementation Notes**  
 Retrieve a ESC config parameters under a category, with a given key

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>category</b>	<input type="text" value="(required)"/>	<b>ESC config parameter category</b>	path	string
<b>key</b>	<input type="text" value="(required)"/>	<b>ESC config parameter key</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Get key under category request accepted.		

HTTP Status Code	Reason	Response Model	Headers
400	Invalid key or category name provided.		
404	Key or category not found.		

Try it out!

**POST** /v0/config/{category}/{key} Create a new key under a category.


### Implementation Notes

Create a new key under a category.

### Response Class (Status 200)

Model | Model Schema

```
{
  "category": "DEFAULT",
  "key": "MAX_LOGS",
  "value": {},
  "type": "STRING"
}
```

Response Content Type  

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>category</b>	<input type="text" value="(required)"/>	<b>ESC config parameter category</b>	path	string
<b>key</b>	<input type="text" value="(required)"/>	<b>ESC config parameter key</b>	path	string



Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	(required)	<b>ESC config parameter value</b>	body	string

Parameter content type:

[Try it out!](#)

**GET** `/v0/config` Retrieve all ESC config parameters

**Implementation Notes**  
 Retrieve all ESC config parameters

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Get all parameters request accepted		

[Try it out!](#)

**PUT** `/v0/config/{category}/{key}/{value}` Change an ESC config parameter under a category, with a given key

**Implementation Notes**  
 Change an ESC config parameter under a category, with a given key.  
 Format: PUT `/v0/config/{category}/{key}/{value}`  
 Example request:

```
curl -X PUT -H "accept: Application/xml" http://127.0.0.1:8080/ESCManager/v0/config/log/level/debug
```

Example response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<escconfigparameter>
  <category>LOG</category>
  <key>LEVEL</key>
  <type>STRING</type>
  <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:string">
</escconfigparameter>
```

Details of Config Parameters: Please refer to installation and user guide for more details.

Category: DEFAULT Key: VM\_RECOVERY\_RETRIES\_MAX Default Value: 3 Type: Int Description: Number of maximum recovery attempts allowed per VM.

Category: OPENSTACK Key: ENDPOINT Default Value: adminURL Type: String Description: The parameter to set up the keystone endpoint value of ESC. Options : adminURL, publicURL

Category: LOG Key: LEVEL Default Value: INFO Type: String Description: Level of logging of ESCManager. Options: INFO, TRACE, DEBUG.

Category: AFFINITY Key: FILTER Default Value: SameHostFilter Type: String Description: A constant string used to build PolicyEngine and initializing VM policy table. Options: SameHostFilter, ServerGroupAffinity

Category: ANTI-AFFINITY Key: FILTER Default Value: DifferentHostFilter Type: String Description: A constant string used to build PolicyEngine and initializing VM policy table. Options: DifferentHostFilter, ServerGroupAffinity

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>category</b>	<input type="text" value="(required)"/>	<b>ESC config parameter category</b>	path	string
<b>key</b>	<input type="text" value="(required)"/>	<b>ESC config parameter key</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>value</b>	<input type="text" value="(required)"/>	<b>ESC config parameter value</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Update value request accepted.		
400	Invalid request.		
404	Associated resource not found.		
406	Not acceptable.		
409	Conflicting resource.		
500	Internal server error.		

[Try it out!](#)

## 10-Volumes

GET	/v0/{internalTenantId}/volumes	Get all volumes under a tenant		
<b>Implementation Notes</b>				
All volumes under that tenant in ESC will be returned as a List				
<b>Parameters</b>				
Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the volume (max length 72 characters)</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

Try it out!

DELETE

/v0/{internalTenantId}/volumes/{internalVolumeId}

Delete a volume from an existing tenant in ESC

## Implementation Notes

Delete a volume from ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalVolumeId</b>	<input type="text" value="(required)"/>	<b>Internal volume ID (max length 72 characters)</b>	path	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the volume (max length 72 characters)</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid request		
500	Internal server error		

Try it out!

**GET** /v0/{internalTenantId}/volumes/{internalVolumeId}

Get a volume by its tenant and internal volume ID

## Implementation Notes

Only volumes created using ESC will be checked

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the volume (max length 72 characters)</b>	path	string
<b>internalVolumeId</b>	<input type="text" value="(required)"/>	<b>Internal volume ID (max length 72 characters)</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Volume Id not found		

Try it out!

POST

/v0/{internalTenantId}/volumes/{internalVolumeId}

Create a volume under an existing tenant in ESC

## Implementation Notes

Create a volume in ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
		<b>volume (max length 72 characters)</b>		
<b>internalVolumeId</b>	<input type="text" value="(required)"/>	<b>Internal volume ID (max length 72 characters)</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/>	<b>Volume configuration payload</b>	body	Model   Model Schema <pre> {   "external_volume_id": "string",   "size": "string",   "sizeunit": "string",   "bus": "string",   "type": "string",   "outOfBand": true,   "bootIndex": "string",   "name": "string",   "format": "string",   "storageLocation": "string",   "external_tenant_id": "string",   "internal_tenant_id": "string",   "internal_volume_id": "string",   "valid": 0,   "event_type": "CREATE_VOLUME",   "image": "string" } </pre>
<a href="#">Click to set as parameter value</a>				
<b>Response Messages</b>				
HTTP Status Code	Reason	Response Model	Headers	
200	Request accepted			
400	Invalid request			

HTTP Status Code	Reason	Response Model	Headers
404	Volume Id not found		
409	Conflicting request		
503	Unable to query VIM		

Try it out!

## 11-ESC Operation Mode

**POST** /v0/operationmode/{escOpMode} Set operation mode in ESC

**Implementation Notes**  
Set operation mode in ESC

**Response Class (Status 200)**

Model | Model Schema

```
{
  "mode": "string"
}
```

Response Content Type

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>escOpMode</b>	<input type="text" value="[MAINTENANCE]"/>	<b>ESC maintenace mode.</b>	path	string

Try it out!



GET

/v0/operationmode

Get operation mode in ESC

### Implementation Notes

Set operation mode in ESC

### Response Class (Status 200)

Model | Model Schema

```
{
  "mode": "string"
}
```

Response Content Type

Try it out!

## 12-Host Actions

GET

/v0/hosts/{hostName}/status

Get the current status of the host

### Implementation Notes

This is to see if the host can be schedule to deploy instances

### Response Class (Status 200)

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

Parameter	Value	Description	Parameter Type	Data Type
Client-Transaction-Id	<input type="text"/>		header	string
<b>hostName</b>	<input type="text" value="(required)"/>		path	string

[Try it out!](#)

**POST** /v0/hosts/{hostName}/disable Disable a host

**Implementation Notes**  
To disable a host

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>CALLBACK-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for host action notifications</b>	header	string
<b>hostName</b>	<input type="text" value="(required)"/>	<b>Target host name</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

[Try it out!](#)

**POST** /v0/hosts/{hostName}/enable Enable a host

### Implementation Notes

To enable a host

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for host action notifications</b>	header	string
<b>hostName</b>	<input type="text" value="(required)"/>	<b>Target host name</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

Try it out!

## 13-ESC VIM Connectors : Manage ESC VIM Connectors

GET /v0/vims

[Get all VIM Connectors](#)

### Implementation Notes

All VIM Connectors in ESC will be returned as a List

### Response Class (Status 200)

Model | Model Schema

```
[  
  {
```

```
"id": "string",
"type": "OPENSTACK",
"properties": {
  "property": [
    {
      "name": "string",
      "value": "string",
      "encrypted_value": "string"
    }
  ]
},
"event_type": "string"
}
```

Response Content Type

Try it out!

POST

/v0/vims

Create a VIM connector in ESC

### Implementation Notes

Create a VIM connector in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">(required)</div> <p>Parameter content type:  <input type="text" value="application/xml"/></p>	<b>VIM Connector configuration payload</b>	body	Model   Model Schema <pre style="background-color: #fff9c4; padding: 10px; border: 1px solid #ccc;"> {   "id": "string",   "type": "OPENSTACK",   "properties": {     "property": [       {         "name": "string",         "value": "string",         "encrypted_value": "string"       }     ]   },   "event_type": "string" }</pre> <p>Click to set as parameter value</p>

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
500	Internal Server Error		

Try it out!

**GET** /v0/vims/default\_vim\_connector
Get default VIM Connector name

Response Class (Status 200)

Model | Model Schema

```
{  
  "defaultVmConnectorId": "string",  
  "event_type": "string"  
}
```

Response Content Type

Try it out!

POST /v0/vims/default\_vim\_connector

Create a Default VIM connector in ESC

### Implementation Notes

Create a Defaukt VIM connector in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>body</b>	<input type="text" value="(required)"/>  Parameter content type: <input type="text" value="application/xml"/>	<b>Default VIM Connector configuration payload</b>	body	Model   Model Schema <pre>{   "defaultVmConnectorId": "string",   "event_type": "string" }</pre> Click to set as parameter value

### Response Messages

HTTP Status Code

Reason

Response Model

Headers

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
500	Internal Server Error		

Try it out!

**DELETE** /v0/vims/{vim\_id} Delete a VIM Connector from ESC

**Implementation Notes**  
Delete a VIM Connector from ESC

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>VIM ID</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	VIM Connector deletion request accepted		
400	Invalid Request		
404	VIM Id not found		
500	Internal Server Error		

Try it out!

**GET****/v0/vims/{vim\_id}**

Get a VIM Connector by its id

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>VIM ID</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Get request accepted		
404	VIM Id not found		

[Try it out!](#)**PUT****/v0/vims/{vim\_id}**

Update an existing VIM connector in ESC

### Implementation Notes

Update an existing VIM connector in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>VIM ID</b>	path	string



Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">(required)</div> <p>Parameter content type:  <input type="text" value="application/xml"/></p>	<b>Update VIM Connector configuration payload</b>	body	Model   Model Schema <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <pre> {   "id": "string",   "type": "OPENSTACK",   "properties": {     "property": [       {         "name": "string",         "value": "string",         "encrypted_value": "string"       }     ]   },   "event_type": "string" } </pre> </div> <p style="text-align: right; font-size: small;">Click to set as parameter value</p>

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
500	Internal Server Error		

Try it out!

**PUT** /v0/vims/{vim\_id}/check\_status Trigger VIM connector status checking

### Implementation Notes

Trigger VIM connector status checking

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>VIM ID</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	Not Found		
500	Internal Server Error		

Try it out!

## 14-ESC VIM Users : Manage ESC VIM Users

GET /v0/vims/{vim\_id}/vim\_users

[Get all VIM Users](#)

### Implementation Notes

All VIM Users in ESC will be returned as a List

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>vim_id</b>	<input type="text" value="(required)"/>		path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Get request accepted		
404	Unknown VIM Id provided		

Try it out!

**POST** /v0/vims/{vim\_id}/vim\_users Create a VIM user under an existing VIM in ESC

**Implementation Notes**  
Create a VIM user in ESC

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>Vim id</b>	path	string

body	(required)	Vim user configuration payload	body	Model   Model Schema
	<input type="text" value="(required)"/>			<pre> {   "credentials": {     "properties": {       "property": [         {           "name": "string",           "value": "string",           "encrypted_value": "string"         }       ]     }   } } </pre>

Parameter content type:

Parameter	Value	Description	Parameter Type	Data Type
				<pre> } }, "id": "string", "vim_id": "string", "event_type": "string" } </pre> <p>Click to set as parameter value</p>
<b>Response Messages</b>				
HTTP Status Code	Reason	Response Model	Headers	
200	Request accepted			
400	Invalid Request			
500	Internal Server Error			
<input type="button" value="Try it out!"/>				

DELETE /v0/vims/{vim_id}/vim_users/{vim_user_id}		Delete a VIM User from ESC		
<b>Implementation Notes</b>				
Delete a VIM User from ESC				
<b>Parameters</b>				
Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>Vim id</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>vim_user_id</b>	<input type="text" value="(required)"/>	<b>Vim user id</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	VIM User Id not found		
500	Internal Server Error		

[Try it out!](#)

**GET** /v0/vims/{vim\_id}/vim\_users/{vim\_user\_id} [Get a VIM User by its id](#)

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>Vim id</b>	path	string
<b>vim_user_id</b>	<input type="text" value="(required)"/>	<b>Vim user id</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Get request accepted		
404	VIM User Id not found		

[Try it out!](#)

PUT

/v0/vims/{vim\_id}/vim\_users/{vim\_user\_id}

Update a VIM user under an existing VIM in ESC

## Implementation Notes

Update a VIM user in ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>Vim id</b>	path	string
<b>vim_user_id</b>	<input type="text" value="(required)"/>	<b>Vim user id</b>	path	string

**body****Vim user configuration payload**

body

[Model](#) | [Model Schema](#)

Parameter content type:



```
{
  "credentials": {
    "properties": {
      "property": [
        {
          "name": "string",
          "value": "string",
          "encrypted_value": "string"
        }
      ]
    }
  },
  "id": "string",
  "vim_id": "string",
  "event_type": "string"
}
```

Parameter	Value	Description	Parameter Type	Data Type
				Click to set as parameter value
<b>Response Messages</b>				
HTTP Status Code	Reason	Response Model		Headers
200	Request accepted			
400	Invalid Request			
500	Internal Server Error			

Try it out!

## 15-System States

<b>POST</b>	<code>/v0/systemstate/{name}/reset</code>	Reset system state of a given name
-------------	---	------------------------------------

## Implementation Notes

Reset system state of a given name

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>name</b>	<input type="text" value="(required)"/>	<b>System state name</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Reset system state of specified name request accepted.		
404	Name not found.		

[Try it out!](#)

**GET** /v0/systemstate/{name}

Retrieve system state of a given name

## Implementation Notes

Retrieve system state of a given name

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>name</b>	<input type="text" value="(required)"/>	<b>System state name</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Get system state of specified name		



HTTP Status Code	Reason	Response Model	Headers
	request accepted.		
404	Name not found.		

[Try it out!](#)

## 16-File Servers

GET	/v0/fileserver	Get all File Servers								
<p><b>Implementation Notes</b></p> <p>All file servers created will be returned as a List</p>										
<p><b>Response Messages</b></p> <table border="1"> <thead> <tr> <th>HTTP Status Code</th> <th>Reason</th> <th>Response Model</th> <th>Headers</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>Request accepted</td> <td></td> <td></td> </tr> </tbody> </table> <p><a href="#">Try it out!</a></p>			HTTP Status Code	Reason	Response Model	Headers	200	Request accepted		
HTTP Status Code	Reason	Response Model	Headers							
200	Request accepted									

DELETE	/v0/fileserver/{internalServerId}	Delete a File Server										
<p><b>Implementation Notes</b></p> <p>Delete a File Server</p>												
<p><b>Parameters</b></p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Value</th> <th>Description</th> <th>Parameter Type</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td><b>Callback</b></td> <td><input type="text" value="(required)"/></td> <td><b>Callback URL for ESC</b></td> <td>header</td> <td>string</td> </tr> </tbody> </table>			Parameter	Value	Description	Parameter Type	Data Type	<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC</b>	header	string
Parameter	Value	Description	Parameter Type	Data Type								
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC</b>	header	string								

Parameter	Value	Description	Parameter Type	Data Type
<b>notifications</b>				
<b>internalServerId</b>	<input type="text" value="(required)"/>	<b>File Server ID</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Not found		
500	Internal server error		

GET	/v0/fileserver/{internalServerId}	<a href="#">Get specific File Server</a>		
<b>Implementation Notes</b>				
Matching File server will be returned if exists				
<b>Parameters</b>				
Parameter	Value	Description	Parameter Type	Data Type
<b>internalServerId</b>	<input type="text" value="(required)"/>	<b>File Server Id</b>	path	string
<b>Response Messages</b>				
HTTP Status Code	Reason	Response Model	Headers	
200	Request accepted			
404	Not found			

POST

/v0/fileserver/{internalServerId}


Create a File Server

### Implementation Notes

Create a File Server

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalServerId</b>	<input type="text" value="(required)"/>	<b>File Server ID</b>	path	string
<b>body</b>	<div style="border: 1px solid #ccc; height: 100px; width: 100%; display: flex; align-items: center; justify-content: center;"><p>(required)</p></div>	<b>ESC datamodel with the file server payload</b>	body	string

Parameter content type:  

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
204	No content		
400	Invalid Request		
409	Conflicting Request		

Try it out!

PUT

/v0/fileserver/{internalServerId}


Update a File Server

### Implementation Notes

Update a File Server

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalServerId</b>	<input type="text" value="(required)"/>	<b>File Server ID</b>	path	string
<b>body</b>	<div style="border: 1px solid #ccc; height: 100px; width: 100%; display: flex; align-items: center; justify-content: center;">(required)</div>	<b>ESC datamodel with the file server payload</b>	body	string

Parameter content type:  

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
204	No content		
400	Invalid Request		
404	Not Found		

Try it out!

## SSL Management

PUT

/internal/ssl/reload

Reload ESC Truststore from file system

### Implementation Notes

Reload ESC Truststore from file system

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	ESC Truststore reloaded successfully		
500	Failed to reload ESC truststore		

Try it out!

[ BASE URL: /ESCManager , API VERSION: 2.2.0 ]