



Managing ESC Resources

- [Managing VIM Connectors, on page 1](#)

Managing VIM Connectors

A VIM connector contains details such as URL and authentication credentials, which enables ESC to connect and communicate with the VIM. ESC connects to more than one VIM if the VIM connectors are configured. You can configure the VIM connector and its credentials in two ways:

- At the time of installation using the `bootvm.py` parameters—Only a single VIM connector can be configured using `bootvm.py`, which becomes the default VIM connector.
- Using the VIM Connector APIs—The VIM connector API allows you to add multiple VIM connectors. You can configure a default VIM connector (if it is not already configured using the `bootvm.py` parameters), and additional VIM connectors.

The default VIM connector connects ESC to the default VIM. Each VIM in a multi VIM deployment is configured with a VIM connector. These VIMs are non-default VIMs. ESC creates and manages resources on a default VIM. Only deployments are supported on a non-default VIM.

For a single VIM deployment, a single configured VIM connector becomes the default VIM connector. For a multiple VIM deployment, you need to add multiple connectors, and specify one connector as default using the default VIM connector API. For more information, see [Deploying VNFs on Multiple OpenStack VIMs](#).



Note ESC accepts the northbound configuration request to create, update, or delete a resource, or a deployment only if the following conditions are met:

- ESC has the target VIM/VIMs and corresponding VIM user configured.
 - ESC is able to reach the target VIM/VIMs.
 - ESC is able to authenticate the VIM user.
-

Configuring the VIM Connector

You can configure the VIM Connector during or after installation.

Configuring the VIM Connector During Installation

To configure the VIM Connector during installation, the following parameter must be provided to `bootvm.py`:

Environment variables	bootvm.py arguments
OS_TENANT_NAME	--os_tenant_name
OS_USERNAME	--os_username
OS_PASSWORD	--os_password
OS_AUTH_URL	--os_auth_url

Configuring the VIM Connector After Installation

To configure the VIM Connector after installation, the following parameter must be provided to `bootvm.py`:

`--no_vim_credentials`

When the `no_vim_credentials` parameter is provided, the following `bootvm.py` arguments are ignored:

- `os_tenant_name`
- `os_username`
- `os_password`
- `os_auth_url`

For details on Installation, see the [Cisco Elastic Services Controller Install and Upgrade Guide](#). You can configure the same using the VIM Connector APIs post installation, for more details, see [Managing VIM Connector Using the VIM Connector APIs, on page 3](#).

Default VIM Connector

The default VIM connector API allows you to specify a default VIM connector when multiple connectors are available in a deployment.

For a Single VIM deployment, ESC supports a single VIM connector. This single VIM connector becomes the default VIM connector. ESC supports multiple VIM connectors for multi VIM deployments. You can configure the default VIM connector using the new locator attribute. If you are using the ESC Release 2.x datamodel for deployments and creating resources, then configure the default VIM connector explicitly in ESC.

The locator attribute is introduced in the data model for deploying VMs on non-default VIMs. For more details, see [Deploying VNFs on Multiple OpenStack VIMs](#).

While deploying, if the VIM connectors are available, but the default connector is not yet configured, then it is mandatory that you specify the locator attribute else the request is rejected.

The data model prior to ESC Release 3.0 cannot be used if the default VIM connector is not configured. While upgrading from ESC Release 2.x to ESC Release 3.0 and later, the existing VIM connector is provisioned as the default VIM connector.



Note You cannot change or delete the default VIM connector to a different one once configured.

You must specify the default connector at the top level (or beginning) of the data model. The data model is as follows:

```
<esc_system_config>
  <vim_connectors>
    <default_vim_connector>vim1</default_vim_connector>
    <vim_connector>
      <id>vim1</id>
    ...
  </vim_connector>
    <vim_connector>
      <id>vim2</id>
    ...
  </vim_connector>
</vim_connectors>
</esc_system_config>
```

To add the default VIM connector using the REST API,

```
<?xml version="1.0"?>
<default_vim_connector xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <defaultVimConnectorId>tb3_v3</defaultVimConnectorId>
</default_vim_connector>
```

To add a VIM connector at the time of installation, see [Configuring the VIM Connector During Installation in *Configuring the VIM Connector*, on page 1](#). The VIM connectors allow multiple VIMs to connect to ESC. For more details on multi VIM deployment, see [Deploying VNFs on Multiple OpenStack VIMs](#).

Deleting VIM Connector

ESC creates SystemAdminTenant automatically when the default VIM connector is created and configured. The SystemAdminTenant cannot be deleted. The VIM is connected and the VIM user is authenticated to the system admin tenant. Hence, the default VIM cannot be deleted or updated. However, the VIM user and its properties can be deleted or updated. You can update and delete the non-default VIM connectors if there are no resources created on the VIM from ESC. If there are resources created on the VIM through ESC, then you must first delete the resources, and then the VIM user to delete the VIM connector.

Managing VIM Connector Using the VIM Connector APIs

If ESC was deployed without passing VIM credentials, you can set the VIM credentials through ESC using the VIM connector and VIM User APIs (REST or Netconf API). Even if the default VIM connector is configured during installation, the additional VIM connectors can be configured using the VIM connector APIs.

Managing using Netconf API

- Passing VIM credential using Netconf:

```
<esc_system_config xmlns="http://www.cisco.com/esc/esc">
```

```

<vim_connectors>
  <!--represents a vim-->
  <vim_connector>
    <!--unique id for each vim-->
    <id>my-server</id>
    <!--vim type [OPENSTACK|VMWARE_VSPHERE|LIBVIRT|AWS|CSP]-->
    <type>OPENSTACK</type>
    <properties>
      <property>
        <name>os_auth_url</name>
        <value>http://{os_ip:port}/v3</value>
      </property>
      <!-- The project name for openstack authentication and authorization -->
      <property>
        <name>os_project_name</name>
        <value>vimProject</value>
      </property>
      <!-- The project domain name is only needed for openstack v3 identity api -->
      <property>
        <name>os_project_domain_name</name>
        <value>default</value>
      </property>
      <property>
        <name>os_identity_api_version</name>
        <value>3</value>
      </property>
    </properties>
    <users>
      <user>
        <id>admin</id>
        <credentials>
          <properties>
            <property>
              <name>os_password</name>
              <value>*****</value>
            </property>
          </properties>
          <!-- The user domain name is only needed for openstack v3 identity api -->
          <property>
            <name>os_user_domain_name</name>
            <value>default</value>
          </property>
        </properties>
      </credentials>
    </user>
  </users>
</vim_connector>
</vim_connectors>
</esc_system_config>

```

- Updating VIM Connector using Netconf:

```

<esc_system_config xmlns="http://www.cisco.com/esc/esc">
  <vim_connectors>
    <vim_connector nc:operation="replace">
      <id>example_vim</id>
      <type>OPENSTACK</type>
      <properties>
        <property>
          <name>os_auth_url</name>
          <value>{auth_url}</value>
        </property>
      </properties>
    </vim_connector>
  </vim_connectors>
</esc_system_config>

```

```

        <name>os_project_name</name>
        <value>vimProject</value>
    </property>
    <!-- The project domain name is only needed for openstack v3 identity api -->
    <property>
        <name>os_project_domain_name</name>
        <value>default</value>
    </property>
    <property>
        <name>os_identity_api_version</name>
        <value>3</value>
    </property>
</properties>
</vim_connector>
</vim_connectors>
</esc_system_config>

```

- Updating VIM user using Netconf:

```

<esc_system_config xmlns="http://www.cisco.com/esc/esc">
  <vim_connectors>
    <vim_connector>
      <id>example_vim</id>
      <users>
        <user nc:operation="replace">
          <id>my_user</id>
          <credentials>
            <properties>
              <property>
                <name>os_password</name>
                <value>*****</value>
              </property>
            <!-- The user domain name is only needed for openstack v3 identity api -->
            <property>
              <name>os_user_domain_name</name>
              <value>default</value>
            </property>
          </properties>
          </credentials>
        </user>
      </users>
    </vim_connector>
  </vim_connectors>
</esc_system_config>

```

- Deleting VIM connector using Netconf:

```

<esc_system_config xmlns="http://www.cisco.com/esc/esc"> <vim_connectors>
  <vim_connector nc:operation="delete">
    <id>example_vim</id>
  </vim_connector>
</vim_connectors>
</esc_system_config>

```

- Deleting VIM User using Netconf:

```

<esc_system_config xmlns="http://www.cisco.com/esc/esc">
  <vim_connectors>
    <vim_connector>
      <id>example_vim</id>
      <users>

```

```

        <user nc:operation="delete">
          <id>my_user</id>
        </user>
      </users>
    </vim_connector>
  </vim_connectors>
</esc_system_config>

```

- Deleting VIM Connector using command:

```
$/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli delete-vim-connector <vim connector id>
```

- Deleting VIM user using command:

```
$/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli delete-vim-user <vim connector id> <vim
user id>
```

Managing using REST API

- Adding VIM using REST:

```

POST /ESCManager/v0/vims/
HEADER: content-type, callback

<?xml version="1.0"?>
<vim_connector xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <id>example_vim</id>
  <type>OPENSTACK</type>
  <properties>
    <property>
      <name>os_auth_url</name>
      <value>{auth_url}</value>
    </property>
    <property>
      <name>os_project_name</name>
      <value>vimProject</value>
    </property>
    <!-- The project domain name is only needed for openstack v3 identity api -->
    <property>
      <name>os_project_domain_name</name>
      <value>default</value>
    </property>
    <property>
      <name>os_identity_api_version</name>
      <value>3</value>
    </property>
  </properties>
</vim_connector>

```

- Adding VIM user using REST:

```

POST /ESCManager/v0/vims/{vim_id}/vim_users
HEADER: content-type, callback

<?xml version="1.0"?>
<user xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <id>my_user</id>
  <credentials>
    <properties>
      <property>
        <name>os_password</name>
        <value>*****</value>
      </property>
    </properties>
  </credentials>
</user>

```

```

    </property>
    <!-- The user domain name is only needed for openstack v3 identity api -->
    <property>
      <name>os_user_domain_name</name>
      <value>default</value>
    </property>
  </properties>
</credentials>
</user>

```

- Updating VIM using REST:

```

PUT /ESCManager/v0/vims/{vim_id}
HEADER: content-type, callback

<?xml version="1.0"?>
<vim_connector xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <!--unique id for each vim-->
  <id>example_vim</id>
  <type>OPENSTACK</type>
  <properties>
    <property>
      <name>os_auth_url</name>
      <value>{auth_url}</value>
    </property>
    <property>
      <name>os_project_name</name>
      <value>vimProject</value>
    </property>
    <!-- The project domain name is only needed for openstack v3 identity api -->
    <property>
      <name>os_project_domain_name</name>
      <value>default</value>
    </property>
    <property>
      <name>os_identity_api_version</name>
      <value>3</value>
    </property>
  </properties>
</vim_connector>

```

- Updating VIM user using REST:

```

PUT /ESCManager/v0/vims/{vim_id}/vim_users/{vim_user_id}
HEADER: content-type, callback

<?xml version="1.0"?>
<user xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <id>my_user</id>
  <credentials>
    <properties>
      <property>
        <name>os_password</name>
        <value>*****</value>
      </property>
      <!-- The user domain name is only needed for openstack v3 identity api -->
      <property>
        <name>os_user_domain_name</name>
        <value>default</value>
      </property>
    </properties>
  </credentials>
</user>

```

- Deleting VIM using REST:

```
DELETE /ESCManager/v0/vims/{vim_id}
```

- Deleting VIM user using REST:

```
DELETE /ESCManager/v0/vims/{vim_id}/vim_users/{vim_user_id}
```

- Notification example after each VIM or VIM user configuration is done:

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-10-06T16:24:05.856+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Created vim connector successfully</status_message>
    <vim_connector_id>my-server</vim_connector_id>
    <event>
      <type>CREATE_VIM_CONNECTOR</type>
    </event>
  </escEvent>
</notification>
```

For more information on the APIs, see [Cisco Elastic Services Controller API Guides](#).

Important Notes:

- You can add more than one VIM connector, but all the VIM connectors must have the same VIM type. Multiple VIM connectors can be added for OpenStack VIM only. However, only one VIM user can be configured per VIM connector.
- `os_project_name` and `os_project_domain_name` properties specify the OpenStack project details for authentication and authorization under the VIM connector properties. If the `os_tenant_name` property exists under the Vim User, it will be ignored.
- The VIM connector properties `os_auth_url` and `os_project_name` and VIM User property `os_password` are mandatory properties for the OpenStack VIM. If these properties are not provided, then the request to create the VIM connector is rejected.
- VIM username and password can be updated anytime. VIM endpoint cannot be updated while resources created through ESC exist.
- The name of a VIM property or VIM user credentials property are not case sensitive, e.g. `OS_AUTH_URL` and `os_auth_url` is the same to ESC.

You can encrypt the VIM connector credentials by replacing the existing `<value>` field with `<encrypted_value>`.

For example,

```
<credentials>
  <properties>
    <property>
      <name>os_password</name>
      <encrypted_value>*****</encrypted_value>
    </property>
    <property>
      <name>os_user_domain_name</name>
      <value>default</value>
    </property>
```



```
</properties>
</credentials>
```

This stores the `os_value` password as an aes-cfb-128-encrypted-string in the CFB using the keys contained in `/opt/cisco/esc/esc_database/esc_production_conf.d.conf`.



Note The existing value must be replaced with encrypted value only within the credentials specified.

For more information, see [Encrypting Configuration Data](#).

VIM Connector Status API

The table below shows the VIM connector status and a status message for each VIM connector. The status shows ESC connection and authentication status of the VIM.

VIM Reachability	User Authentication	Status (by ESC)	Status Message
NOT REACHABLE	-	CONNECTION_FAILED	Unable to establish VIM connection
REACHABLE	VIM user is not configured	NO_CREDENTIALS	No VIM user credentials found
REACHABLE	Authentication failed	AUTHENTICATION_FAILED	VIM authentication failed
REACHABLE	Authentication successful	CONNECTION_SUCCESSFUL	Successfully connected to VIM

Status using the REST API

HTTP Operation: GET

Path: `ESCManager/v0/vims`, `ESCManager/v0/vims/<specific_vim_id>`

Sample REST Response is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vim_connector xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <properties>
    <property>
      <name>os_auth_url</name>
      <value>http://172.16.0.0:5000/v2.0</value>
    </property>
  </properties>
  <id>default_openstack_vim</id>
  <status>CONNECTION_SUCCESSFUL</status>
  <status_message>Successfully connected to VIM</status_message>
  <type>OPENSTACK</type>
</vim_connector>
```

Status using the NETCONF API

The opdata shows the status. The VIM connector status is within the vim connector container.

Sample opdata is as follows:

```
<system_config>
  <active_vim>OPENSTACK</active_vim>
  <openstack_config>
```

```

    <os_auth_url>http://172.16.0.0:5000/v2.0/</os_auth_url>
    <admin_role>admin</admin_role>
    <os_tenant_name>admin</os_tenant_name>
    <os_username>admin</os_username>
    <member_role> member_</member_role>
  </openstack_config>
  <vim_connectors>
    <vim_connector>
      <id>my-server</id>
      <status>CONNECTION_FAILED</status>
      <status_message>Unable to establish VIM connection</status_message>
    </vim_connector>
    <vim_connector>
      <id>Openstack-Liberty</id>
      <status>NO_CREDENTIALS</status>
      <status_message>No VIM user credentials found</status_message>
    </vim_connector>
  </vim_connectors>
</system_config>

```

VIM Connector Operation Status

The VIM_CONNECTION_STATE notification notifies the status of each VIM connector and user added to ESC through REST and NETCONF. For more details about the VIM connectors, see [Managing VIM Connectors, on page 1](#).

The notification shows:

- Event Type: VIM_CONNECTION_STATE
- Status: Success or Failure
- Status message
- vim_connector_id

Notifications are sent for monitoring the VIM connector, adding or deleting the VIM user, and updating the VIM connector. The success and failure notification examples are as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2017-06-27T14:50:40.823+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
    <status>FAILURE</status>
    <status_code>0</status_code>
    <status_message>VIM Connection State Down</status_message>
    <vim_connector_id>my-server</vim_connector_id>
    <event>
      <type>VIM_CONNECTION_STATE</type>
    </event>
  </escEvent>
</notification>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2017-06-27T14:51:55.862+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
    <status>SUCCESS</status>
    <status_code>0</status_code>
    <status_message>VIM Connection State Up</status_message>
    <vim_connector_id>my-server</vim_connector_id>
    <event>

```

```
        <type>VIM_CONNECTION_STATE</type>
    </event>
</escEvent>
</notification>
```

