Post Installation Tasks

This chapter contains the following sections:

> **Note** It is recommended that you ignore the `do not edit` message, if you want to modify the cloud-init day-0 configuration, file.

Changing the ESC Password

You will be forced to change the default password on the first time login. Portal will not let you bypass this step and will keep returning you to this page until you change the default password. After the first time password change, you can change your password using the procedures described in this section. Also, if the user has multiple browsers or tabs or the SAME user is logged on by 2 or more computers and one of the user changes the password then everyone will be logged off and asked to re-enter the new password. The user session has an expiry of 1 hour so if the user is inactive on the portal for an hour then portal will expire the session and the user will have to re-login. If you forgot your password, you can also update or randomly generate the password.

This section discusses how to change the passwords.

Example for REST:

```
sudo escadm rest set --username {USERNAME} --password {PASSWORD}
```

Example for ETSI:

```
sudo escadm etsi set --rest_user {USERNAME:PASSWORD}
```

Changing the ConfD Netconf/CLI Administrator Password Using the Command Line Interface

After you install ESC, to change the Confd admin password, do the following:

Procedure

Step 1 Log in to the ESC VM.

```
$ ssh USERNAME@ESC_IP
```

Step 2 Switch to the admin user.

```
[admin@esc-ha-0 esc]$ sudo bash
[sudo] password for admin:
```

Step 3 Load the ConfD CLI:

```
$ /opt/cisco/esc/confd/bin/confd_cli  -u admin
```

Step 4 Set the new admin password:

```
$ configure
$ set aaa authentication users user admin password <new password>
```

Step 5 Save the changes.

```
$ commit
```

Changing Linux Account Password

Procedure

Step 1 Log in to ESC VM.

```
$ ssh USERNAME@ESC_IP
```

Step 2 To update or generate a random password, use the following command:

```
/usr/bin/pwqcheck
/usr/bin/pwqgen
```

Changing the ESC Portal Password

The user can update or reset the default admin password.

Procedure

Step 1   Log in to ESC VM.

Step 2   Switch to the root user.

Step 3   To update the default admin password or randomly generate a password, use one of the following method:

- Using escadm utility:

  To update the default admin password (admin/********):

  ```
  [root@anyname-v44-52 admin]# escadm portal set --username admin --password ********
  Successfully updated password for username admin
  ```

  To generate a random password:

  ```
  [root@anyname-v44-52 admin]# escadm portal set --username admin
  Would you like to use the generated password: "Accent5omit&Wide"?[y|n]y
  Successfully updated password for username admin
  ```

  The *--must_change* variable will ask the user to change their password at the next login.

  The *--must_change* variable is not applicable for REST users.

  ```
  [root@anyname-v44-52 admin]# escadm portal set --username admin --must_change
  Would you like to use the generated password: "Rainy4Dozen&Behave"?[y|n]y
  Successfully reset password for username admin. User must change the password at the
  next login.
  ```

- To reset to a specific password:

  ```
  [root@anyname-v44-52 admin]# escadm portal set --username admin --password P@55w0rd!
  --must_change
  Successfully reset password for username admin. User must change the password at the
  next login.
  ```

- Using the bootvm command line:

  ```
  --user_portal_pass admin:<new password>
  ```

- Using the ESC Portal:

  a. Log in to ESC portal using your username and password.

  b. Choose **Accounts Setting** on the Navigation menu.

  c. Enter the old password in the Old password field, then enter a new password in the New Password and Confirm Password fields.

  d. Click **Update Password**.

Configuring Pluggable Authentication Module (PAM) Support for Cisco Elastic Services Controller

You can configure ESC services to use Pluggable Authentication Modules (PAM) for user authentication in ESC. With Cisco Elastic Services supporting PAM, you can also enable LDAP authentication in ESC. If PAM is not configured, ESC will continue to use the default authentication method for each ESC service. The following table lists the commands to enable PAM authentication for each ESC service.

Table 1: Configuring PAM for ESC Services

ESC Service/Component	Command
ESCManager (REST interface)	`sudo escadm escmanager set --auth PAM:[:<pam_service_name>]`
ESC Monitor (Health API)	`sudo escadm monitor set --auth PAM:[:<pam_service_name>]`
Confd	`sudo escadm confd set --auth PAM:[:<pam_service_name>]`
Portal	`sudo escadm portal set --auth PAM[:<pam_service_name>]`
ETSI	`sudo escadm etsi set --pam_service <pam_service_name>`

Note

- The SSHD service that runs inside the ESC VM already uses PAM authentication by default.

- If any component sets PAM authentication without specifying the PAM service, ESC defaults to the PAM service 'system-auth'.

Authenticating REST Requests

ESC REST and ETSI REST APIs use http basic access authentication where the ESC client will have to provide a username and password when making 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.

This section discusses ESC REST and ETSI REST authentications, how to change the default password of the interfaces, and how to send an authorized requests from the ESC client.

REST Authentication

By default, the REST authentication is enabled. To disable REST authentication, you can pass the argument **--disable-rest-auth** to bootvm. Cisco does not recommend you to use this in a production environment.

ESC also 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. By default, REST is enabled as HTTP and restricted to localhost.

ESC can be installed with external access to REST on HTTPS or HTTP enabled with additional bootvm.py arguments: **--enable-https-rest** or **--enable-http-rest**.

It is recommended to use only enabled external REST API when required. When enabled, it is recommended to use **bootvm.py --enable-https-rest --user_rest_pass USERNAME:PASSWORD**.

Note   Make sure to pass either **--enable-https-rest** or **--enable-http-etsi-rest** or both the arguments to the bootvm.py script to enable http and https interfaces to the REST API. You must pass **--user_rest_pass** while using either **--enable-https-rest** or **--enable-http-etsi-rest** when REST authentication is not disabled. To enable https or http after ESC VM is booted, use the escadm command specified below.

```
sudo escadm escmanager set --url
http://127.0.0.1:8080/ESCManager,https://0.0.0.0:8443/ESCManager
```

You must change the configuration of the peer instance if ESC is in HA Active/Standby mode.

Enabling ETSI REST Authentication

If the ETSI REST http or https interfaces are enabled, then all requests to an ETSI API must contain authentication data. You can use the **--enable-http-etsi-rest** or **--enable-https-etsi** arguments respectively to enable http and https interfaces to the ESC bootvm.py installation script.

You can enable both interfaces simultaneously, but only the https interface should be enabled in a production environment.

Note   To enable http or https after the ESC VM has booted, use the escadm command specified below:

```
sudo escadm etsi enable_http_rest
```

OR

```
sudo escadm etsi enable_https_rest
```

Then restart the ETSI service.

Changing the REST Interface Password

The REST interface has only one default username/password (admin/<default_password>). The password can be updated after the bootup using escadm tool from the ESC VM CLI. You can also update the password through the REST API.

Procedure

Step 1   Log in to ESC VM.

Step 2   To replace the existing password with a new one, use one of the below options:

- Using the escadm tool from the ESC VM CLI, you can generate a random password:

```
[root@test-v44-52 admin]# escadm rest set --help
usage: escadm rest set [-h] [-v] --username USERNAME [--password PASSWORD]

optional arguments:
  -h, --help          show this help message and exit
  -v, --v, --verbose  show verbose output
  --username USERNAME
  --password PASSWORD  new password or use randomly generated password if no
                       password provided
```

- Using the REST API:

```
http://[ESCVM_IP]:8080/ESCManager/v0/authentication/setpassword?userName=admin&password=yourPassword
```

or

```
https://[ESCVM_IP]:8443/ESCManager/v0/authentication/setpassword?userName=admin&password=yourPassword
```

Changing the ETSI REST Interface Password

The ETSI REST interface has only one default username/password (admin/<default_password>). The password can be updated after the bootup using escadm tool from the ESC VM CLI.

Procedure

Step 1   Log in to ESC VM.

Step 2   To set the default ETSI REST username and password, use the following command:

```
sudo escadm etsi set --rest_user username:password
```

or

```
[admin@xyz-esc-4-4-0-59-keep ~]$ escadm etsi set --help
usage: escadm etsi set [-h] [-v] [--startup {0,1,true,false,manual,auto}]
[--rest_user REST_USER] [--pam_service PAM_SERVICE]

optional arguments:
-h, --help show this help message and exit
-v, --v, --verbose show verbose output
--startup {0,1,true,false,manual,auto}
set to false|0|manual to disable etsi at startup.
--rest_user REST_USER
Set the user for rest. Format username:password
--pam_service PAM_SERVICE
Specify a PAM service to use for authentication. This
will override the rest user. To revert to the using
```

```
the rest user for authentication, supply an empty
string.
```

Sending an Authorized REST Request

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

`Authorization: Basic YWRtaW46Y2lzY28xMjM=`

where *YWRtaW46Y2lzY28xMjM=* is the Base64 encoded string of the default username/password.

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

Example using the default credentials:

For HTTP:

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

For HTTPS:

https://[ESCVM_IP]:8443/ESCManager/v0/tenants/

Sending an Authorized ETSI REST Request

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

`Authorization: Basic YWRtaW46Y2lzY28xMjM=`

where *YWRtaW46Y2lzY28xMjM=* is the Base64 encoded string of the default username/password.

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

Example using the default credentials:

For HTTP:

http://[ESCVM_IP]:8250/vnflcm/v1/vnf_lcm_op_occs

For HTTPS:

http://[ESCVM_IP]:8251/vnflcm/v1/vnf_lcm_op_occs

Configuring Openstack Credentials

If ESC was deployed without passing VIM credentials, you can set VIM credentials through ESC VIM and through VIM User APIs (REST or Netconf API).

> ✎
>
> **Note** ESC will accept the northbound configuration request only if the following conditions are met:
>
> • ESC has VIM or a VIM user configured through APIs(REST/Netconf).
>
> • ESC has VIM or a VIM user configured, and ESC is able to reach the VIM.
>
> • ESC has VIM or a VIM user configured, and ESC is able to authenticate the user.

Configuring 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-30</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 needed for openstack v3 identity api -->
        <property>
          <name>os_project_domain_name</name>
          <value>default</value>
        </property>
      </properties>
      <users>
        <user>
          <id>admin</id>
          <credentials>
            <properties>
              <property>
                <name>os_password</name>
                <value>********</value>
              </property>
              <!-- The user domain name is 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>
```

Note
- From ESC 3.0 onwards, multiple VIM connectors are supported but within one ESC, only one type of VIM are supported. For example, all the Vim Connector(s) has to be for OpenStack only. One ESC VIM cannot have two VIM connector, one points to OpenStack, one points to VMware.

- One VIM is chosen as the default VIM which supports all pre 3.0 config requests and datamodels.

- Deployments can be done on the VIM that is not the default VIM. The deployment to a non default VIM has to have all out-of-band resources (except ephemeral volumes). No other configurations like image, flavor, network, and so on can be done on the VIM that is not the default VIM.

- The default VIM connector will be auto provisioned and does not need to be configured in the following scenarios:
  - If VIM credentials have been passed during ESC boot up.
  - If upgrading from 2.3.x to 3.0.

- The change in the datamodel for Openstack create VIM connector would be handled during upgrade by migration. The 'os_tenant_name' and 'os_project_domain_name' properties would be moved to the VIM Connector properties and 'os_ternant_name' will be renamed to 'os_project_name'.

- For the default VIM Connector, once it is properly authenticated, its properties cannot be updated.

- VIM user can be deleted, recreated, or its properties can be updated at anytime.

- **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>
        <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>
    </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 Connector using command:

```
$/esc_nc_cli delete-vim-connector <vim connector id>
```

- Deleting VIM user using command:

```
$/esc_nc_cli delete-vim-user <vim connector id> <vim user id>
```

Configuring using REST API

- Adding VIM using REST:

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

```xml
<?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
<?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>
```

• Update VIM using REST:

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

```xml
<?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>
```

- Update 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>
```

- Delete VIM using REST:

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

- Delete VIM user using REST:

```
DELETE /ESCManager/v0/vims/{vim_id}/vim_users/{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-30</vim_connector_id>
    <event>
      <type>CREATE_VIM_CONNECTOR</type>
    </event>
  </escEvent>
</notification>
```

Important Notes:

- In ESC 3.0, you can add multiple VIM Connector for Openstack VIM. Each VIM Connector can have only one VIM User.

- VIM username and password can be updated at anytime. VIM endpoint will not be able to update after a resource is created through ESC.

- After VIM is connected and VIM user is authenticated, VIM can no longer be deleted or updated, only VIM user can be deleted or updated.

- The name of a VIM property or VIM user credentials property is not case sensitive, e.g. OS_AUTH_URL and os_auth_url is the same to ESC.

Reconfiguring ESC Virtual Machine

This section covers the following:

- Reconfiguring Rsyslog

- Reconfiguring NTP

- Reconfiguring DNS

- Reconfiguring Hosts

- Reconfiguring Timezone

Reconfiguring Rsyslog

Rsyslog parameters are optional. If there is a need for customization after booting an ESC VM, you can edit the files in ESC VM (/etc/rsyslog.d/).

Procedure

Step 1   **Editing the Rsyslog file**:

- If you haven't specified the log forwarding configuration at the bootup time, you may create a file under /etc/rsyslog.d/ like /etc/rsyslog.d/log-forwarding.conf.
- If you have specified the log forwarding through installation, you may just need to edit the file. The file could be /etc/rsyslog.d/20-cloud-config.conf. In this file, to forward logs to multiple rsyslog servers, edit the following line:

```
*.* @[server_ip]:port
```

Note	• Use '@@' before specifying server ip address (if TCP is the protocol used to forward logs to the rsyslog server).
	• Use '@' before specifying server ip address (if UDP is the protocol used to forward logs to the rsyslog server).
	• server_ip can either be ipv4/ipv6 address of the rsyslog server.
	• '[ ]' around the server_ip is required to separate it from ' :port# ', if an ipv6 server address is specified.

For further information on Rsyslog configuration, see the Red Hat documentation.

Step 2 **Configuring the ESC log file:** Configure which ESC log files you want to forward to the rsyslog server:

a) Navigate to /etc/rsyslog.d/ Create or modify a configuration file, such as **log-esc.conf**. Make a copy of sample log-esc.conf .

b) Specify the following block for every file you want to forward to rsyslog server.

```
$InputFileName /var/log/esc/escmanager.log
$InputFileTag esc-manager:
$InputFileStateFile stat-esc-manager
$InputFileSeverity info
$InputRunFileMonitor
```

For example:

```
$InputFileName /var/log/esc/file1.log
$InputFileTag file1:
$InputFileStateFile stat-file1
$InputFileSeverity info
$InputRunFileMonitor

$InputFileName /var/log/esc/file2.log
$InputFileTag file2:
$InputFileStateFile stat-file2
$InputFileSeverity info
$InputRunFileMonitor
```

Step 3 Restart the rsyslog service

```
# service rsyslog restart
```

Step 4 Configure the server side to receive forwarded logs.

a) On a designated server, go to /etc/rsyslog.conf, and uncomment the lines shown below, depending on if you want to listen to logs from clients based on TCP or UDP:

```
#$ModLoad imudp
#$UDPServerRun 514
```

b) Exit the file. Run this command as the last step.

```
sudo service rsyslog restart
```

Now, the server is listening for logs on port 514, using TCP/UDP.

Reconfiguring NTP

Procedure

Step 1 Open the NTP configuration file /etc/ntp.conf in a text editor such as vi, or create a new one if it does not already exist:

```
# vi /etc/ntp.conf
```

Step 2 Add or edit the list of public NTP servers. If you don't specify the NTP server through the installation, the file should contain the following default lines, but feel free to change or expand these according to your needs:

```
server 0.rhel.pool.ntp.org iburst
server 1.rhel.pool.ntp.org iburst
server 2.rhel.pool.ntp.org iburst
server 3.rhel.pool.ntp.org iburst
server <your_ntp_server_ip> iburst
```

The iburst directive at the end of each line speeds up the initial synchronization.

Step 3 Once you have the list of servers complete, in the same file, set the proper permissions, giving the unrestricted access to localhost only. Make sure those lines are there in your configure file.

```
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict -6 ::1
```

Step 4 Save all changes, exit the editor, and restart the NTP daemon:

```
# service ntpd restart
```

Step 5 Make sure that ntpd is started at boot time:

```
# chkconfig ntpd on
```

Reconfiguring DNS

Procedure

Step 1 The /etc/resolv.conf file contains the configuration for the DNS client (resolver). It typically looks something like this:

```
search domain.com
nameserver 8.8.4.4
```

This results in a `/etc/resolv.conf`:

```
Created by cloud-init on instance boot automatically, do not edit.
;
#Generated by esc-cloud
domain cisco.com
```

```
search cisco.com
nameserver 8.8.4.4
```

Note   It is recommended that you ignore the `do not edit` message, if you want to modify the file.

Step 2   You may modify the IP address of the "nameserver" item or add new nameserver records.

```
search domain.com
nameserver <your_first_dns_ip>
nameserver <your_second_dns_ip>
```

Reconfiguring Hosts

The /etc/hosts file allows you to add, edit, or remove hosts . This file contains IP addresses and their corresponding hostnames. If your network contains computers whose IP addresses are not listed in DNS, it is recommended that you add them to the /etc/hosts file.

Procedure

Step 1   Add the IP addresses that are not listed in DNS to the /etc/hosts file.

Step 2   Restart your network for the changes to take effect.

```
service network restart
```

Reconfiguring Timezone

For ESC VM, in /etc the file "localtime" is a link to or copy of a file containing information about your time zone. Access your zone information files from /usr/share/zoneinfo. To change the time zone, find your country, your city or a city in the same time zone from zone information files in /usr/share/zoneinfo and link it to the localtime in the /etc file.

```
$ ln \-sf /usr/share/zoneinfo/America/Los_Angeles /etc/localtime
```

Verifying ESC Configurations and Other Post-Install Operations

This section covers various post-install checks and operations using the escadm tool.

Verifying Existing ESC Configurations

You can use `escadm dump` command for displaying current ESC configurations in yaml format. The output will show the various services in ESC.

```
$ sudo escadm dump

resources:
  confd:
```

```
     init_aaa_users:
     - name: admin
       passwd:
     option: start-phase0
  esc_service:
    group:
    - confd
    - mona
    - vimmanager
    - pgsql
    - escmanager
    - portal
    - monitor
    - snmp
    type: group
  escmanager: {}
  mona: {}
  monitor: {}
  pgsql: {}
  portal: {}
  snmp:
    run_forever: true
  vimmanager: {}
```

Verifying VIM configurations

You can use `escadm vim show` command to verify the vim settings are correctly populated:

```
$ sudo escadm vim show

[
    {
        "status": "CONNECTION_SUCCESSFUL",
        "status_message": "Successfully connected to VIM",
        "type": "OPENSTACK",
        "id": "default_openstack_vim",
        "properties": {
            "property": [
                {
                    "name": "os_auth_url",
                    "value": "http://10.85.103.143:35357/v3"
                }
            ]
        }
    }
]
```

Troubleshooting ESC Services Startup Issues

Problem: Issues encountered while verifying ESC services status at the installation time using `sudo escadm status`.

Causes: Some services take time to start or have trouble starting.

Solution:

1. Identify the issues using one of the following method:

    • Check the log /var/log/esc/escadm.log

    ```
    $ cat /var/log/esc/escadm.log
    2017-06-01 20:35:02,925: escadm.py(2565): INFO: promote drbd to master...
    ```

```
2017-06-01 20:35:02,934: escadm.py(2605): INFO: Waiting for at least one drbd to be
 UptoDate...
2017-06-01 20:35:02,942: escadm.py(2616): INFO: Waiting for peer drbd node to be
demoted...
2017-06-01 20:35:14,008: escadm.py(2423): INFO: mount: /dev/drbd1
/opt/cisco/esc/esc_database
2017-06-01 20:35:14,017: escadm.py(1755): INFO: Starting filesystem service: [OK]
2017-06-01 20:35:15,039: escadm.py(1755): INFO: Starting vimmanager service: [OK]
2017-06-01 20:35:16,116: escadm.py(1755): INFO: Starting monitor service: [OK]
2017-06-01 20:35:17,163: escadm.py(1755): INFO: Starting mona service: [OK]
2017-06-01 20:35:18,440: escadm.py(1755): INFO: Starting snmp service: [OK]
2017-06-01 20:35:21,397: escadm.py(1770): INFO: Starting confd service:[FAILED]
2017-06-01 20:35:28,304: escadm.py(1755): INFO: Starting pgsql service: [OK]
2017-06-01 20:35:29,331: escadm.py(1755): INFO: Starting escmanager service: [OK]
2017-06-01 20:35:30,354: escadm.py(1755): INFO: Starting portal service: [OK]
2017-06-01 20:35:31,523: escadm.py(1755): INFO: Starting esc_service service: [OK]
```

- Add '-v' to escadm status to show the verbose output of the ESC services.

```
$ sudo escadm status --v
0 ESC status=0 ESC HA Master Healthy
pgsql (pgid 61397) is running
vimmanager (pgid 61138) is running
monitor (pgid 61162) is running
mona (pgid 61190) is running
drbd is master
snmp (pgid 61541) is running
filesystem (pgid 0) is running
<<service>> is dead
keepalived (pgid 60838) is running
portal (pgid 61524) is running
confd (pgid 61263) is running
escmanager (pgid 61491) is running
```

2. Confirm the status of the identified services that has issues and manually start these services.

```
$ sudo escadm <<service>> status// If the status is stopped or dead, manually start the
 services using the next command.

$ sudo escadm <<service>> start --v
```

Logging in to the ESC Portal

✎

Note

- The ESC portal is enabled by default. You must ensure that the ESC portal is not disabled during installation. For more information on enabling or disabling the ESC portal, see Installing Cisco Elastic Services Controller Using the QCOW Image.

- When you log in to the ESC portal for the first time you are prompted to change the default password.

To log in to the ESC portal, do the following:

Before you begin

- Register an instance of ESC. For more information on registering the ESC instance see, Installing Cisco Elastic Services Controller Using the QCOW Image

- Ensure that you have the username and password.

Procedure

Step 1 Using your web browser, enter the IP address of ESC and port 8443.

Example:

For example, if the IP address of ESC is 192.0.2.254, enter:

https://192.0.2.254: 8443 [login via https]

A Security Alert message is displayed.

Step 2 Click **Yes** to accept the security certificate. The Login page is displayed.

Step 3 Enter the username and password and click **Login** .

If you are logging in for the first time, the login page reappears, prompting you to change your password.

Step 4 Enter the old password in the Old Password field, then enter a new password in the New Password and Confirm Password fields.

Step 5 Click **Update Password** or press **Enter**.

Note
- If the UI becomes unresponsive, restart the UI by executing the **sudo escadm portal restart** from the ESC shell prompt.

- ESC Portal only supports one user.

- Currently, a pre-installed self-signed certificate supports HTTPS. The user must confirm the self-signed certificate before proceeding with the ESC Portal.

- In HTTPS communication mode, if the URL protocol type returned by OpenStack is not HTTPS, the access to the VNF Console may be disabled. For security reasons, while running in HTTPS more non-secure communication will be rejected.