インストール後のタスク

この章は、次の項で構成されています。

（注） cloud-init day-0 コンフィギュレーション ファイルを変更する場合は、[編集しない（do not edit）] メッセージを無視することをお勧めします。

ESC パスワードの変更

初回ログイン時には、デフォルトのパスワードを強制的に変更する必要があります。ポータルでは、この手順をバイパスすることはできず、デフォルトのパスワードを変更するまでこのページに戻ります。パスワードを初めて変更した後、このセクションで説明されている手順を使用してパスワードを変更できます。また、ユーザが複数のブラウザまたはタブを持っている場合、または同じユーザが 2 台以上のコンピュータからログインしている場合、ユーザの 1 人がパスワードを変更すると、全員がログオフされ、新しいパスワードを再入力するように求められます。ユーザセッションの有効期限は 1 時間であるため、ユーザがポータルで 1 時間アクティブでない場合、ポータルはセッションを期限切れにし、ユーザは再ログインする必要があります。パスワードを忘れた場合は、パスワードを更新したり、ランダムに生成したりすることもできます。

ここでは、パスワードを変更する方法について説明します。

REST の例：

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

ETSI の例：

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

コマンドライン インターフェイスを使用した ConfD Netconf/CLI 管理者パスワードの変更

ESC をインストールした後、Confd 管理者パスワードを変更するには、次の手順を実行します。

ESC リリース 5.4 以降では、confd_cli などの confd コマンドを実行できません。*confd_cli -u admin* は、*ssh admin@localhost -p 2024* コマンドに置き換えられます。

ESC のインストールの詳細については、『QCOW イメージを使用した Cisco Elastic Services Controller のインストール』を参照してください。

管理者アカウントの confD cli にアクセスするには、次の手順を実行します。

```
admin@esc$ ssh admin@localhost -p 2024
admin@localhost's password: *******

admin connected from 127.0.0.1 using ssh on esc
admin@esc>
```

手順

ステップ 1 ESC VM にログインします。

```
$ ssh USERNAME@ESC_IP
```

ステップ 2 管理者ユーザに切り替えます。

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

ステップ 3 ConfD CLI をロードします。

```
$ /opt/cisco/esc/confd/bin/ssh admin@localhost -p 2024
```

ステップ 4 新しい管理者パスワードを設定します。

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

ステップ 5 変更内容を保存します。

```
$ commit
```

ESC における ConfD の読み取り専用ユーザグループの作成

ESC の ConfD は、readonly という名前の新しいグループを導入することで強化されています。読み取り専用グループのメンバーの場合は、情報を取得するだけで、権限を変更することはできません。

Bootvm のロール名として「readonly」を使用できます。次の例は、ConfD で 2 人のユーザを作成する方法を示しています。1 つは管理者専用で、もう 1 つは読み取り専用です。

```
# bootvm.py name-500-105-100 --user_confd_pass admin:admin --user_confd_pass
readonly:readonly::readonly --user_pass admin:admin --image ESC-5_0_0_105 --net network
```

HA A/A では、aa-day0.yaml のグループ名として「readonly」を使用できます。次が例になります。

```
confd:
  init_aaa_users:
  - group: readonly
    name: admin
    passwd: $6$rounds=4096$Ps1JIjKihRTF$fo8XPBxwEHJWWfNiXDnO269rlhAxAhWBc
PBfGnZxy1gM3QMxcN8jJ6guWt9Bu.ZkWdPt3hr0OghO73Wr3iDHb0
```

ESC vm が展開された後に、confd の読み取り専用ユーザを作成することもできます。次の手順では、「test」という名前の confd 読み取り専用ユーザと「test」というパスワードを作成します。

```
[root@name-500-155 admin]#  /opt/cisco/esc/confd/bin/ssh admin@localhost -p 2024
admin connected from 127.0.0.1 using console on name-500-155
admin@name-500-155> configure
Entering configuration mode private
[ok][2019-12-06 18:17:39]
[edit]
admin@name-500-155% set aaa authentication users user test uid 9000 gid 9000 password
$0$test homedir /var/confd/homes/test ssh_keydir /var/confd/homes/test/.ssh
[ok][2019-12-06 18:19:15]
[edit]
admin@name-500-155% set nacm groups group readonly user-name test
[ok][2019-12-06 18:19:41]
[edit]
admin@name-500-155% commit
Commit complete.
[ok][2019-12-06 18:19:47]
[edit]
admin@name-500-155%
```

読み取り専用ユーザとして、リモートで ConfD にアクセスすることもできます。

```
name@my-server-39:~$ ssh -p 2024 readonly@172.29.0.57
readonly@172.29.0.57's password:
readonly connected from 172.16.103.46 using ssh on name-500-156
readonly@name-500-156> configure
Entering configuration mode private
[ok][2019-12-13 16:15:33]
[edit]
readonly@name-500-156% show esc_datamodel
tenants {
    tenant admin {
        description      "Built-in Admin Tenant";
        managed_resource false;
        vim_mapping      true;
    }
}
```

```
[ok][2019-12-13 16:15:38]
[edit]
```

読み取り専用の ConfD グループに分類され、変更権限を必要とする場合、ConfD の ESC からアクセス拒否エラーが送信されます。次に、アクセス拒否エラーメッセージの例を示します。

```
$ esc_nc_cli --user readonly --password ****** edit-config dep.xml
Configure
/opt/cisco/esc/confd/bin/netconf-console --port=830 --host=127.0.0.1 --user=readonly
--password=****** --edit-config=/tmp/d.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <rpc-error>
    <error-type>application</error-type>
    <error-tag>access-denied</error-tag>
    <error-severity>error</error-severity>
  </rpc-error>
</rpc-reply>
```

ESC が PAM/IDM を使用するように設定されている場合は、次のようにします。IDM サーバのグループは、ConfD のグループに直接マッピングされます。したがって、読み取り専用ユーザは、IDM グループ「readonly」にマッピングする必要があります。

次に例を示します。

```
$ ipa group-find  --all readonly
---------------
1 group matched
---------------
  dn: cn=readonly,cn=groups,cn=accounts,dc=linuxsysadmins,dc=local
  Group name: readonly
  GID: 5003
  Member users: readonly
  ipantsecurityidentifier: S-1-5-21-2222126199-2113948134-574478857-1003
  ipauniqueid: 858b8cda-0d34-11ea-bca8-525400b29c19
  objectclass: top, groupofnames, nestedgroup, ipausergroup, ipaobject, posixgroup,
ipantgroupattrs
----------------------------
Number of entries returned 1
----------------------------
```

Linux アカウントのパスワードの変更

手順

ステップ 1 ESC VM にログインします。

```
$ ssh USERNAME@ESC_IP
```

ステップ 2 ランダムなパスワードを更新または生成するには、次のコマンドを使用します。

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

ESC ポータルパスワードの変更

ユーザは、デフォルトの管理者パスワードを更新またはリセットできます。

手順

ステップ 1 ESC VM にログインします。

ステップ 2 ルートユーザに切り替えます。

ステップ 3 デフォルトの管理者パスワードを更新するか、またはランダムにパスワードを生成するには、次のいずれかの方法を使用します。

- Escadm ユーティリティを使用：

  デフォルトの管理者パスワード（admin/********）を更新する場合：

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

  ランダムなパスワードを生成する場合：

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

  --must_change 変数は、次のログイン時にパスワードを変更するようユーザに要求します。

  --must_change 変数は、REST ユーザには適用されません。

  ```
  [root@anyname-v44-52 admin]# sudo 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.
  ```

- 特定のパスワードにリセット：

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

- bootvm コマンドラインを使用：

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

- ESC ポータルを使用：

  1. ユーザ名とパスワードを使用して ESC ポータルにログインします。

2. ナビゲーションメニューの [アカウントの設定（Accounts Settings）] を選択します。
3. [古いパスワード（Old Password）] フィールドに古いパスワードを入力し、[新しいパスワード（New Password）] および [パスワードの確認（Confirm Password）] フィールドに新しいパスワードを入力します。
4. [パスワードを更新（Update Password）] をクリックします。

Cisco Elastic Services Controller での着脱可能な認証モジュール（PAM）サポートの設定

ESC サービスを設定して、ESC のユーザ認証に着脱可能な認証モジュール（PAM）を使用できます。PAM をサポートする Cisco Elastic Services を使用すると、ESC で LDAP 認証を有効にすることもできます。PAM が設定されていない場合、ESC は ESC サービスごとにデフォルトの認証方式を引き続き使用します。次の表に、各 ESC サービスに対して PAM 認証を有効にするコマンドを示します。

表 *1 : ESC* サービス用の *PAM* の設定

ESC サービス/コンポーネント	PAM 認証を設定するコマンド
ESCManager（REST インターフェイス）	`sudo escadm escmanager set --auth PAM:<pam_service_name>`
ESC Monitor（ヘルス API）	`sudo escadm monitor set --auth PAM:<pam_service_name>`
Confd	`sudo escadm confd set --auth PAM:<pam_service_name>`
ポータル	`sudo escadm portal set --auth PAM:<pam_service_name>`
ETSI	`sudo escadm etsi set --pam_service <pam_service_name>`

（注）
- ESC VM 内で実行される SSHD サービスは、すでに PAM 認証をデフォルトで使用しています。
- いずれかのコンポーネントが PAM サービスを指定せずに PAM 認証を設定した場合、ESC はデフォルトで PAM サービス「system-auth」になります。

PAM 認証サービスの設定とユーザグループ

各 ESC サービス（上記）には、関連付けられた PAM 認証設定と、特定のアクセス制御を提供するユーザグループがあります。ユーザグループは `/etc/group` ファイルで定義されます。管理者ユーザは、すべてのグループのメンバーです。

表 *2 : PAM* 認証サービスの設定とユーザグループ

/etc/group	/etc/pam.d
portal-user:x:1002:admin	portal-auth
rest-user:x:1003:admin	rest-auth
confd-user:x:1004:admin	confd-auth
etsi-user:x:1005:admin	etsi-auth
health-user:x:1006:admin	health-auth

たとえば、ヘルス API の PAM 認証を設定し、ヘルスユーザグループのみにアクセスを制限するには、次のコマンドを実行します。

```
$ sudo escadm monitor set --auth PAM:health-auth
ESC configuration was changed and saved automatically. They will take effect once you
restart ESC service by running "sudo escadm restart"
```

ESC コンポーネントへの PAM ユーザの追加については、ESC サービス/コンポーネントへの PAM ユーザの追加 （7 ページ）を参照してください。

ESC サービス/コンポーネントへの PAM ユーザの追加

次の ESC サービスグループに PAM ユーザを追加できます。

- rest-user
- confd-user
- portal-user
- etsi-user
- health-user

次の手順を実行して、PAM ユーザを ESC サービス/コンポーネントに追加します。

手順

ステップ 1 ESC VM にログインします。

ステップ 2 次のコマンドを使用して、PAM ユーザを追加します。

```
sudo passwd pamuser
            Changing password for user pamuser.
            New password:
            Retype new password:
            passwd: all authentication tokens updated successfully.
```

ステップ 3 次のコマンドを使用して、PAM ユーザを ESC サービス/コンポーネントグループに追加します。

```
sudo usermod -a -G <ESC Service Group> pamuser
```

（注） PAMユーザは、Confdサービスの管理者または読み取り専用グループに追加する必要があります。

Cisco Elastic Services Controller を ID 管理クライアントとして 設定

前提条件

- ID 管理クライアント（IDM）サーバが起動して稼働中であることを確認します。
- ESC で DNS サーバが稼働状態にあることを確認します。DNS サーバが稼働中、ESC インスタンスはホスト名を使用して IDM サーバと対話します。

次の例は、ESC（esc-client-500.linuxsysadmins.local）が IDM サーバ（idmns.linuxsysadmins.local）に到達する様子を示しています。

```
[root@esc-client-500 admin]# ping idmns
PING idmns.linuxsysadmins.local (192.168.222.176) 56(84) bytes of data.
64 bytes from idmns.linuxsysadmins.local (192.168.221.176): icmp_seq=1 ttl=64 time=0.492
 ms
64 bytes from idmns.linuxsysadmins.local (192.168.221.176): icmp_seq=2 ttl=64 time=0.457
 ms
64 bytes from idmns.linuxsysadmins.local (192.168.221.176): icmp_seq=3 ttl=64 time=0.645
 ms
```

IDM は sssd を使用して設定できます。IDM サーバと連携するために ESC サービスの設定を開始するには、ESC の PAM 設定ファイルで、`/etc/pam.d/syste-auth`、`system-auth-esc-sssd` を指定します。

```
# cd /etc/pam.d
# ln -sf system-auth-esc-sssd  system-auth
# ls -al /etc/pam.d/system-auth
lrwxrwxrwx. 1 root root 20 Nov 13 00:39 /etc/pam.d/system-auth -> system-auth-esc-sssd
```

各 ESC サービスに対して IDM 認証を有効にするためのコマンド一覧を次の表に示します。

表 *3 : ESC* サービスに対する *IDM* の設定

ESC サービス/コンポーネントコマンド	コマンド
ESCManager	`# sudo escadm escmanager set --auth PAM:system-auth-esc-sssd`
ETSI	`# sudo escadm etsi set --pam_service system-auth-esc-sssd`
ConfD	`# sudo escadm confd set --auth PAM:system-auth-esc-sssd`

ID ポリシーおよび監査クライアントとしての Cisco Elastic Services Controller の設定

ESC をアイデンティティポリシーおよび監査クライアント（IPA）クライアントとして設定するには、次のコマンドを実行します。

```
ipa-client-install
```

次に、IPA クライアントとして ESC を設定する例を示します。

```
[root@esc-client-500 admin]# ipa-client-install --domain linuxsysadmins.local --server
idmns.linuxsysadmins.local --realm LINUXSYSADMINS.LOCAL
WARNING: ntpd time&date synchronization service will not be configured as
conflicting service (chronyd) is enabled
Use --force-ntpd option to disable it and force configuration of ntpd

Autodiscovery of servers for failover cannot work with this configuration.
If you proceed with the installation, services will be configured to always access the
discovered server for all operations and will not fail over to other servers in case of
 failure.
Proceed with fixed values and no DNS discovery? [no]: yes
Client hostname: esc-client-500.linuxsysadmins.local
Realm: LINUXSYSADMINS.LOCAL
DNS Domain: linuxsysadmins.local
IPA Server: idmns.linuxsysadmins.local
BaseDN: dc=linuxsysadmins,dc=local

Continue to configure the system with these values? [no]: yes
Skipping synchronizing time with NTP server.
User authorized to enroll computers: admin
Password for admin@LINUXSYSADMINS.LOCAL:
Successfully retrieved CA cert
    Subject:     CN=Certificate Authority,O=LINUXSYSADMINS.LOCAL
    Issuer:      CN=Certificate Authority,O=LINUXSYSADMINS.LOCAL
    Valid From:  2019-11-12 23:23:32
    Valid Until: 2039-11-12 23:23:32

Enrolled in IPA realm LINUXSYSADMINS.LOCAL
Created /etc/ipa/default.conf
Configured sudoers in /etc/nsswitch.conf
Configured /etc/sssd/sssd.conf
Configured /etc/krb5.conf for IPA realm LINUXSYSADMINS.LOCAL
trying https://idmns.linuxsysadmins.local/ipa/json
[try 1]: Forwarding 'schema' to json server 'https://idmns.linuxsysadmins.local/ipa/json'
trying https://idmns.linuxsysadmins.local/ipa/session/json
```

```
[try 1]: Forwarding 'ping' to json server
'https://idmns.linuxsysadmins.local/ipa/session/json'
[try 1]: Forwarding 'ca_is_enabled' to json server
'https://idmns.linuxsysadmins.local/ipa/session/json'
Systemwide CA database updated.
Adding SSH public key from /etc/ssh/ssh_host_ecdsa_521_key.pub
Adding SSH public key from /etc/ssh/ssh_host_ecdsa_384_key.pub
Adding SSH public key from /etc/ssh/ssh_host_rsa_key.pub
Adding SSH public key from /etc/ssh/ssh_host_ed25519_key.pub
Adding SSH public key from /etc/ssh/ssh_host_ecdsa_key.pub
[try 1]: Forwarding 'host_mod' to json server
'https://idmns.linuxsysadmins.local/ipa/session/json'
Could not update DNS SSHFP records.
SSSD enabled
Configured /etc/openldap/ldap.conf
Configured /etc/ssh/ssh_config
Configured /etc/ssh/sshd_config
Configuring linuxsysadmins.local as NIS domain.
Client configuration complete.
The ipa-client-install command was successful
```

REST 要求の認証

ESC REST および ETSI REST API は、HTTP 基本アクセス認証を使用します。この場合、ESC クライアントは、REST 要求を行うときにユーザ名とパスワードを提供する必要があります。ユーザ名とパスワードは、送信中に Base64 でエンコードされますが、暗号化もハッシュ化もされません。HTTPS は基本認証と組み合わせて使用され、暗号化を提供します。

ここでは、ESC REST および ETSI REST 認証について、インターフェイスのデフォルトパスワードを変更する方法、および ESC クライアントから許可された要求を送信する方法について説明します。

REST 認証

デフォルトでは、REST 認証は有効に設定されています。REST 認証を無効にするには、引数 **--disable-rest-auth** を bootvm に渡すことができます。シスコでは、実稼働環境でこれを使用することは推奨していません。

ESC は、ポート 8443 経由の https 通信もサポートしています。ESC は、クライアントが https 通信を開始するために信頼する必要がある自己署名証明書を生成します。デフォルトでは、REST は HTTP として有効になっており、localhost に制限されています。

ESCは、追加の bootvm.py 引数（**--enable-https-rest** または **--enable-http rest**）を使用して HTTPS または HTTP 上の REST への外部アクセスを有効にしてインストールできます。

必要に応じて、有効になっている外部 REST API のみを使用することをお勧めします。有効にした場合、**bootvm.py --enable-https-rest --user_rest_pass USERNAME:PASSWORD** を使用することを推奨します。

（注） REST API への http および https インターフェイスを有効にするには、**--enable-https-rest** または **--enable-http-etsi-rest** を渡すか、もしくは bootvm.py スクリプトへの引数の両方を渡すようにしてください。REST 認証が無効になっていない場合は、**--user_rest_pass** または **--enable-https-rest** を使用しているときに、**--user_rest_pass** を渡す必要があります。ESC VM が起動した後に https または http を有効にするには、以下に指定された escadm コマンドを使用します。

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

ESC が HA アクティブ/スタンバイモードの場合は、ピアインスタンスの設定を変更する必要があります。

ETSI REST 認証の有効化

ETSI REST http または https インターフェイスが有効になっている場合は、ETSI API へのすべての要求に認証データが含まれている必要があります。**--enable-http-etsi-rest** または **--enable-https-etsi** 引数をそれぞれ使用して、http および https インターフェイスを ESC bootvm.py インストールスクリプトに対して有効にすることができます。

両方のインターフェイスを同時に有効にすることはできますが、実稼働環境では https インターフェイスのみを有効にする必要があります。

（注） ESC VM が起動した後に http または https を有効にするには、次に指定された escadm コマンドを使用します。

```
sudo escadm etsi enable_http_rest
```

または

```
sudo escadm etsi enable_https_rest
```

その後、ETSI サービスを再起動します。

REST インターフェイスパスワードの変更

REST インターフェイスには、デフォルトのユーザ名/パスワード（admin/<default_password>）が 1 つしかありません。パスワードは、起動後に ESC VM CLI から escadm tool を使用して更新できます。REST API を使用してパスワードを更新することもできます。

手順

ステップ 1 ESC VM にログインします。

ステップ 2 既存のパスワードを新しいものに置き換えるには、次のいずれかのオプションを使用します。

- ESC VM CLI から escadm ツールを使用すると、ランダムなパスワードを生成できます。

```
[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
```

- REST API の使用

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

または

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

ETSI REST インターフェイスのパスワードの変更

ETSI REST インターフェイスには、デフォルトのユーザ名/パスワード（admin/<default_password>）が 1 つしかありません。パスワードは、起動後に ESC VM CLI から escadm tool を使用して更新できます。

手順

ステップ 1 ESC VM にログインします。

ステップ 2 デフォルトの ETSI REST ユーザ名とパスワードを設定するには、次のコマンドを使用します。

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

または

```
[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.
```

承認済み REST 要求の送信

許可された要求を送信するには、ESC クライアントが次のヘッダーを使用して要求を送信する必要があります。

```
Authorization: Basic YWRtaW46Y2lzY28xMjM=
```

ここで、*YWRtaW46Y2lzY28xMjM=* は、デフォルトのユーザ名/パスワードの Base64 でエンコードされた文字列です。

ほとんどのライブラリと Web クライアントは、ユーザ名/パスワードを提供するためのインターフェイスを備えており、アプリケーションはユーザ名/パスワードをエンコードし、HTTP 基本認証ヘッダーを追加します。

デフォルトのクレデンシャルを使用する例：

HTTP の場合：

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

HTTPS の場合：

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

承認済みの ETSI REST 要求の送信

許可された要求を送信するには、ESC クライアントが次のヘッダーを使用して要求を送信する必要があります。

```
Authorization: Basic YWRtaW46Y2lzY28xMjM=
```

ここで、*YWRtaW46Y2lzY28xMjM=* は、デフォルトのユーザ名/パスワードの Base64 でエンコードされた文字列です。

ほとんどのライブラリと Web クライアントは、ユーザ名/パスワードを提供するためのインターフェイスを備えており、アプリケーションはユーザ名/パスワードをエンコードし、HTTP 基本認証ヘッダーを追加します。

デフォルトのクレデンシャルを使用する例：

HTTP の場合：

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

HTTPS の場合：

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

OpenStack ログイン情報の設定

VIM クレデンシャルを渡さずに ESC が展開された場合、ESC VIM および VIM ユーザ API（REST または Netconf API）を介して VIM クレデンシャルを設定できます。

（注） ESC は、次の条件を満たしている場合にのみノースバウンド設定要求を受け入れます。

- ESC には、API（REST/Netconf）を介して設定された VIM または VIM ユーザが含まれています。
- ESC には VIM または VIM ユーザが設定されており、ESC は VIM に到達できます。
- ESC には VIM または VIM ユーザが設定されており、ESC はユーザを認証できます。

Netconf API を使用した設定

- **Netconf を使用した VIM クレデンシャルの提供**：

```
<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>
```

（注）

- ESC 3.0 以降では、複数の VIM コネクタがサポートされていますが、1 つの ESC 内では 1 つのタイプの VIM のみがサポートされています。たとえば、すべての VIM コネクタが OpenStack 専用である必要があります。1 つの ESC VIM には 2 つの VIM コネクタを設定できません。1 つは OpenStack、1 つは VMware を指します。
- 1 つの VIM がデフォルトの VIM として選択されます。これは、すべての pre 3.0 設定要求とデータモデルをサポートしています。
- 展開はデフォルトの VIM ではない VIM で行うことができます。デフォルト以外の VIM への展開では、すべてのアウトオブバンドリソース（一時ボリュームを除く）を持つ必要があります。イメージ、フレーバ、ネットワークなどのその他の設定は、デフォルトの VIM ではない VIM で実行できます。
- デフォルトの VIM コネクタは自動プロビジョニングされ、次のシナリオで設定する必要はありません。
  - ESC 起動中に VIM クレデンシャルが渡された場合。
  - 2.3.x から 3.0 にアップグレードする場合。
- Openstack create VIM コネクタのデータモデルの変更は、移行によるアップグレード中に処理されます。「os_tenant_name」および「os_project_domain_name」プロパティは、VIM コネクタのプロパティに移動され、「os_ternant_name」は「os_project_name」に変更されます。
- デフォルトの VIM コネクタでは、正常に認証されると、それらのプロパティを更新できなくなります。
- VIM ユーザは、いつでも削除、再作成、またはそのプロパティを更新できます。

- **Netconf を使用した VIM コネクタの更新**：

```
<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>
```

- Netconf を使用した VIM ユーザの更新：

```
<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>
```

- Netconf を使用した VIM コネクタの削除：

```
<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>
```

- コマンドを使用した VIM コネクタの削除：

```
$ esc_nc_cli --user <username> --password <password> delete-vim-connector <vim
connector id>
```

- コマンドを使用した VIM ユーザの削除：

```
$ esc_nc_cli --user <username> --password <password> delete-vim-user <vim connector
 id> <vim user id>
```

REST API を使用して設定

- REST を使用した VIM の追加：

```
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>
```

- REST を使用した VIM ユーザの追加：

```
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>
      <!-- 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>
```

- REST を使用した VIM の更新：

```
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>
```

- REST を使用して VIM ユーザの更新：

```
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>
```

- REST を使用した VIM の削除：

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

- REST を使用した VIM ユーザの削除：

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

- 各 VIM または VIM ユーザの設定が完了した後の通知の例：

```
<?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>
```

特記事項：

- ESC 3.0 では、Openstack VIM の複数の VIM コネクタを追加できます。各 VIM コネクタでは、1 つの VIM ユーザのみを持つことができます。
- VIM ユーザ名とパスワードはいつでも更新できます。VIM エンドポイントは、ESC を使用してリソースを作成した後は更新できません。
- VIM が接続され、VIM ユーザが認証されると、VIM を削除または更新することができなくなります。また、VIM ユーザのみを削除または更新できます。
- VIM プロパティまたは VIM ユーザログイン情報プロパティの名前は大文字と小文字が区別されません。たとえば、OS_AUTH_URL と os_auth_url は ESC にとっては同じです。

ESC での Barbican クライアントの有効化

OpenStack Barbican は、パスワード、暗号キー、X.509 証明書などの秘密のセキュアなストレージ、プロビジョニング、および管理を提供します。

VM にマウントする前に、OpenStack ボリュームの暗号化に使用するシークレットを Barbican クライアントで管理できるようにします。Python 3 環境を介して OpenStack Barbican API にアクセスできます。OpenStack Barbican クライアントの python-barbicanclient 5.0.1 は ESC に統合されています。

次のコマンドを使用して、仮想環境を有効にします。

```
source /opt/esc_custom_python3_venv/bin/activate
```

.

ESC 仮想マシンの再設定

ここでは、次のトピックについて取り上げます。

- rsyslog の再設定
- NTP の再設定
- DNS の再設定
- ホストの再設定
- タイムゾーンの再設定

rsyslog の再設定

rsyslogパラメータはオプションです。ESC VMを起動した後にカスタマイズが必要になった場合は、ESC VM（/etc/rsyslog.d/）内のファイルを編集できます。

手順

ステップ 1 **rsyslog ファイルの編集**：

- ブートアップ時のログ転送の設定を指定しなかった場合は、/etc/rsyslog.d/ に /etc/rsyslog.d/log-forwarding.conf のようなファイルを作成できます。
- インストール時にログ転送を指定した場合は、ファイルを編集するだけで済みます。ファイルは /etc/rsyslog.d/20-cloud-config.conf である可能性があります。複数の rsyslog サーバにログを転送するには、このファイルで次の行を編集します。

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

（注）
- サーバの IP アドレスを指定する前に「@@」を入力してください（rsyslog サーバへのログの転送に使用されるプロトコルが TCP である場合）。
- サーバのIPアドレスを指定する前に「@」を入力してください（rsyslogサーバへのログの転送に使用されるプロトコルが UDP である場合）。
- server_ip には、rsyslog サーバの IPv4 アドレスと IPv6 アドレスのいずれかを使用できます。
- IPv6 サーバアドレスが指定されている場合は、server_ip を囲む「[ ]」を「:port#」から分離する必要があります。

rsyslog の設定の詳細については、Red Hat のマニュアルを参照してください。

ステップ 2 **ESC ログファイルの設定**：rsyslogサーバにどのESCログファイルを転送するかを設定します。

a) /etc/rsyslog.d/ に移動して、**log-esc.conf** などの設定ファイルを作成または変更します。サンプルとして log-esc.conf のコピーを作成します。

b) rsyslog サーバに転送するすべてのファイルに対して、次のブロックを指定します。

```
$InputFileName /var/log/esc/escmanager.log
$InputFileTag esc-manager:
```

```
$InputFileStateFile stat-esc-manager
$InputFileSeverity info
$InputRunFileMonitor
```

次に例を示します。

```
$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
```

ステップ 3 rsyslog サービスを再起動します。

```
# service rsyslog restart
```

ステップ 4 転送されたログを受信するようにサーバ側を設定します。

a) 指定されたサーバで、/etc/rsyslog.conf に移動し、TCP または UDP に基づいてクライアントからのログをリッスンするかどうかに応じて、以下に示す行をコメント解除します。

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

b) ファイルを終了します。最後の手順として、このコマンドを実行します。

```
sudo service rsyslog restart
```

サーバは、TCP/UDP を使用してポート 514 でログをリッスンするようになりました。

NTP の再設定

手順

ステップ 1 vi などのテキストエディタで NTP 設定ファイル /etc/ntp.conf を開きます。ファイルがまだ存在しない場合は、新しいファイルを作成します。

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

ステップ 2 パブリック NTP サーバのリストを追加または編集します。インストール時に NTP サーバを指定しない場合、ファイルに次のデフォルト行が含められますが、必要に応じて自由に変更または拡張できます。

```
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
```

各行の最後にある iburst ディレクティブは、初期同期を高速化します。

ステップ 3 サーバのリストアップが完了したら、同じファイルで適切な権限を設定し、localhost のみに無制限のアクセス権を付与します。該当する行が設定ファイルに含まれていることを確認してください。

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

ステップ 4 すべての変更を保存し、エディタを終了して、NTP デーモンを再起動します。

```
# service ntpd restart
```

ステップ 5 起動時に ntpd が開始されていることを確認してください。

```
# chkconfig ntpd on
```

DNS の再設定

手順

ステップ 1 /etc/resolv.conf ファイルには、DNS クライアント（リゾルバ）の設定が含まれています。通常、次のように表示されます。

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

その結果、`/etc/resolv.conf` に次のような記述が含められます。

```
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
```

（注） ファイルを変更する場合は、`do not edit` メッセージを無視することをお勧めします。

ステップ 2 「nameserver」項目の IP アドレスを変更するか、または新しいネームサーバレコードを追加できます。

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

ステップ 3 ネットワークサービスを再起動します。

```
service network restart
```

ホストの再設定

/etc/hosts ファイルを使用して、ホストを追加、編集、または削除できます。このファイルには、IP アドレスと対応するホスト名が含まれています。IP アドレスが DNS にリストされていないコンピュータがネットワークに含まれている場合は、それらのコンピュータを /etc/hosts ファイルに追加することをお勧めします。

手順

ステップ 1 DNS にリストされていない IP アドレスを/etc/hosts ファイルに追加します。

ステップ 2 ネットワークを再起動して、変更内容を有効にします。

```
service network restart
```

タイムゾーンの再設定

ESC VM の場合、/etc の「localtime」ファイルは、タイムゾーンに関する情報が含まれているファイルのリンクまたはコピーです。/usr/share/zoneinfo からゾーン情報ファイルにアクセスします。タイムゾーンを変更するには、/usr/share/zoneinfo のゾーン情報ファイルで、現在の国や都市、または同じタイムゾーンにある都市を検索し、/etc ファイル内の localtime にリンクします。

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

ESC 設定と他のインストール後操作の確認

ここでは、escadm ツールを使用したインストール後の各種チェックおよび操作について説明します。

既存の ESC 設定の確認

`escadm dump` コマンドを使用すると、現在の ESC 設定を yaml 形式で表示できます。出力結果には、 ESC のさまざまなサービスが表示されます。

```
$ 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: {}
```

VIM 設定の確認

`escadm vim show` コマンドを使用すると、VI Mの設定が正しく入力されているかを確認できます。

```
$ 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://172.16.103.143:35357/v3"
                }
            ]
        }
    }
]
```

ESC サービスのスタートアップ問題に関するトラブルシューティング

問題：インストール時に `sudo escadm status` を使用して ESC サービスのステータスを確認する際に発生する問題

原因：サービスの中には、開始に時間がかかるものや、開始時に問題が発生するものがあります。

解決策：

1. 次のいずれかの方法で、問題を特定します。
    - ログ /var/log/esc/escadm.log の確認

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

- ESC サービスの詳細出力を表示するには、「-v」を escadm ステータスに追加します。

```
$ sudo escadm status --v
0 ESC status=0 ESC HA Active Healthy
pgsql (pgid 61397) is running
vimmanager (pgid 61138) is running
monitor (pgid 61162) is running
mona (pgid 61190) is running
drbd is active
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. 問題のあることが特定されたサービスのステータスを確認し、これらのサービスを手動で開始します。

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

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

ESC ポータルへのログイン

（注）
- ESC ポータルはデフォルトで有効になっています。インストール時に ESC ポータルが無効になっていないことを確認する必要があります。ESC ポータルの有効化または無効化の詳細については、「QCOW イメージを使用した Cisco Elastic Services Controller のインストール」を参照してください。
- ESC ポータルへの初回ログイン時に、デフォルトパスワードの変更を求められます。

ESC ポータルにログインするには、次の手順を実行します。

始める前に

- ESC のインスタンスを登録します。ESC のインスタンスの登録における詳細については、次を参照してください。 QCOW イメージを使用した Cisco Elastic Services Controller のインストール
- ユーザ名とパスワードを取得していることを確認します。

手順

ステップ 1 Web ブラウザを使用して、ESC とポート 443 の IP アドレスを入力します。

例：

たとえば、ESC の IP アドレスが 192.0.2.254 の場合は、次のように入力します。

https://192.0.2.254: 443 [login via https]

セキュリティ アラート メッセージが表示されます。

ステップ 2 [はい（Yes）] をクリックしてセキュリティ証明書を受け入れます。ログイン ページが表示されます。

ステップ 3 ユーザ名とパスワードを入力して、[ログイン（Login）] をクリックします。

初回ログイン時には、ログインページが再表示され、パスワードの変更を求められます。

ステップ 4 [古いパスワード（Old Password）] フィールドに古いパスワードを入力し、[新しいパスワード（New Password）] および [パスワードの確認（Confirm Password）] フィールドに新しいパスワードを入力します。

ステップ 5 [パスワードの更新（Update Password）] をクリックするか、Enter を押します。

（注）
- UI が応答しなくなった場合は、ESC シェルプロンプトから **sudo escadm portal restart** を実行して UI を再起動します。
- ESC ポータルは 1 人のユーザのみをサポートします。
- 現在、事前インストールされた自己署名証明書はHTTPSをサポートしています。ESCポータルの処理を進める前に、ユーザは自己署名証明書を確認する必要があります。
- HTTPS 通信モードでは、OpenStack によって返される URL プロトコルタイプが HTTPS ではない場合、VNF コンソールへのアクセスが無効になることがあります。セキュリティ上の理由から、HTTPS で実行している間は、安全性の低い通信は拒否されます。