



Cisco 動的属性コネクタのインストール

この章では、サポートされているすべてのオペレーティングシステムに Cisco 動的属性コネクタをインストールする方法について説明します。

- [サポートされているオペレーティングシステムとサードパーティソフトウェア \(1 ページ\)](#)
- [前提条件ソフトウェアのインストール \(2 ページ\)](#)
- [Cisco Secure 動的属性コネクタ のインストール \(7 ページ\)](#)
- [Cisco Secure 動的属性コネクタのアップグレード \(9 ページ\)](#)

サポートされているオペレーティングシステムとサードパーティソフトウェア

dynamic attributes connector の前提条件は次のとおりです。

- Ubuntu 18.04 または 20.04
- CentOS 7 Linux または 8
- Red Hat Enterprise Linux (RHEL) 7 または 8
- Python 3.6.x
- Ansible 2.9 以降

すべてのオペレーティングシステムの最小要件：

- 4 個の CPU
- 8 GB RAM
- 100 GB 使用可能なディスクスペース

vCenter 属性を使用する場合は、次も必要です。

- vCenter 6.7

- 仮想マシンに、VMware ツールがインストールされている必要があります。

前提条件ソフトウェアのインストール

始める前に

物理的または仮想的な設定があり、システムがFirepower Management Center と通信できることを確認してください。

ステップ 1 (オプション) テキストエディタを使用して `/etc/environment` を編集し、次の変数をエクスポートして、Ubuntu マシンがインターネットプロキシの背後にある場合にインターネットと通信できるようにします。

変数	値
<code>export http_proxy</code>	HTTP プロキシで使します。 <code>user:pass@host-or-ip:port</code>
<code>export https_proxy</code>	HTTPS プロキシでこれを使します。 <code>user:pass@host-or-ip:port</code>
<code>export no_proxy</code>	プロキシ構成を削除します。 <code>export no_proxy="localhost,127.0.0.1"</code>

例 :

認証なしの HTTP プロキシ :

```
vi /etc/environment
export http_proxy="myproxy.example.com:8181"
```

認証付き HTTPS プロキシ :

```
vi /etc/environment
export https_proxy="ben.smith:bens-password@myproxy.example.com:8181"
```

ステップ 2 別のコマンドウィンドウを使用して設定を確認します。

```
env grep | proxy
```

結果の例 :

```
http_proxy=myproxy.example.com:8181
```

ステップ 3 次のいずれかのセクションに進みます。

関連トピック

[前提条件ソフトウェアのインストール : Ubuntu](#) (5 ページ)

[前提条件ソフトウェアのインストール : CentOS](#) (3 ページ)

[前提条件ソフトウェアのインストール : RHEL](#) (4 ページ)

前提条件ソフトウェアのインストール : CentOS

始める前に

次のことをすべて行います。

- システムがサポートされているオペレーティングシステムとサードパーティソフトウェア (1 ページ) で説明した前提条件を満たしていることを確認します。
- (オプション) dynamic attributes connector へのプロキシアクセスが必要な場合は、前提条件ソフトウェアのインストール (2 ページ) を参照してください。

ステップ 1 Docker がインストールされていないことを確認し、インストールされている場合はアンインストールします。

```
docker --version
```

Docker がインストールされている場合は、[Ubuntu での Docker エンジンのアンインストール](#)の説明に従ってアンインストールします。

ステップ 2 リポジトリを更新およびアップグレードします。

CentOS 7 :

```
sudo yum -y update && sudo yum -y upgrade
```

ステップ 3 epel リポジトリをインストールします。

CentOS 7 :

```
sudo yum -y install epel-release
```

ステップ 4 (CentOS 7 のみ。) Python3 をインストールします。

```
sudo yum install -y python3 libselenium-python3
```

ステップ 5 Ansible をインストールします。

CentOS 7 および CentOS 8 :

```
sudo yum install -y ansible
```

ステップ 6 Ansible のバージョンが 2.9 以降であることを確認します。

CentOS 7 :

```
ansible --version
ansible 2.9.24
config file = /etc/ansible/ansible.cfg
configured module search path = [u'/home/admin/.ansible/plugins/modules',
u'/usr/share/ansible/plugins/modules']
ansible python module location = /usr/lib/python2.7/site-packages/ansible
executable location = /usr/bin/ansible
python version = 2.7.5 (default, Apr 2 2020, 13:16:51) [GCC 4.8.5 20150623 (Red Hat 4.8.5-39)]
```

(注) 前出の出力が示すように、Ansible が Python 2.x を参照するのは正常です。コネクタは引き続き Python 3 を使用します。

次のタスク

[Cisco Secure 動的属性コネクタ のインストール \(7 ページ\)](#) で説明されているように、コネクタをインストールします。

オプションで `dynamic attributes connector` でのプロキシの使用を停止するには、`/etc/environment` を編集してプロキシ構成を削除します。

前提条件ソフトウェアのインストール : RHEL

始める前に

次のことをすべて行います。

- システムがサポートされているオペレーティングシステムとサードパーティソフトウェア ([1 ページ](#)) で説明した前提条件を満たしていることを確認します。
- (オプション) `dynamic attributes connector` へのプロキシアクセスが必要な場合は、[前提条件ソフトウェアのインストール \(2 ページ\)](#) を参照してください。

ステップ 1 Docker がインストールされていないことを確認し、インストールされている場合はアンインストールします。

```
docker --version
```

Docker がインストールされている場合は、[Ubuntu](#) での [Docker エンジンのアンインストール](#) の説明に従ってアンインストールします。

ステップ 2 リポジトリを更新します。

RHEL 7 :

```
sudo yum -y update && sudo yum -y upgrade
```

RHEL 8 :

```
sudo dnf -y update && sudo dnf -y upgrade
```

ステップ 3 epel リポジトリをインストールします。

RHEL 7 :

```
sudo yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

RHEL 8 :

```
sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

ステップ 4 (RHEL 7 のみ。) Python3 をインストールします。

```
sudo yum install -y python3 libselenium-python3
```

ステップ 5 Ansible をインストールします。

RHEL 7 :

```
sudo yum -y install ansible
```

RHEL 8 :

```
sudo dnf install -y ansible
```

ステップ 6 Ansible のバージョンを確認します。

```
ansible --version
```

次に例を示します。

RHEL 7 :

```
ansible 2.9.24
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/stevej/.ansible/plugins/modules',
u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.5 (default, Mar 20 2020, 17:08:22) [GCC 4.8.5 20150623 (Red Hat 4.8.5-39)]
```

(注) 前出の出力が示すように、Ansible が Python 2.x を参照するのは正常です。コネクタは引き続き Python 3 を使用します。

RHEL 8 :

```
ansible 2.9.24
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/stevej/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.6/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.6.8 (default, Mar 18 2021, 08:58:41) [GCC 8.4.1 20200928 (Red Hat 8.4.1-1)]
```

次のタスク

[Cisco Secure 動的属性コネクタのインストール \(7 ページ\)](#) で説明されているように、コネクタをインストールします。

オプションで `dynamic attributes connector` でのプロキシの使用を停止するには、`/etc/environment` を編集してプロキシ構成を削除します。

前提条件ソフトウェアのインストール : Ubuntu

ステップ 1 Docker がインストールされていないことを確認し、インストールされている場合はアンインストールします。

```
docker --version
```

Docker がインストールされている場合は、[Ubuntu](#) での [Docker エンジン](#) のアンインストールの説明に従ってアンインストールします。

ステップ 2 リポジトリを更新します。

```
sudo apt -y update && sudo apt -y upgrade
```

ステップ 3 Python のバージョンを確認します。

```
/usr/bin/python3 --version
```

バージョンが 3.6 より前の場合は、バージョン 3.6.x をインストールする必要があります。

ステップ 4 Python 3.6 をインストールします。

```
sudo apt -y install python3.6
```

ステップ 5 共通ライブラリをインストールします。

```
sudo apt -y install software-properties-common
```

ステップ 6 Ansible をインストールします。

```
sudo apt-add-repository -y -u ppa:ansible/ansible && sudo apt -y install ansible
```

ステップ 7 Ansible のバージョンを確認します。

```
ansible --version
```

次に例を示します。

```
ansible --version
ansible 2.9.19
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/admin/.ansible/plugins/modules',
u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.17 (default, Feb 27 2021, 15:10:58) [GCC 7.5.0]
```

(注) 前出の出力が示すように、Ansible が Python 2.x を参照するのは正常です。コネクタは引き続き Python 3.6 を使用します。

次のタスク

[Cisco Secure 動的属性コネクタ のインストール \(7 ページ\)](#) で説明されているように、コネクタをインストールします。

オプションで `dynamic attributes connector` でのプロキシの使用を停止するには、`/etc/environment` を編集してプロキシ構成を削除します。

Cisco Secure 動的属性コネクタ のインストール

インストールについて

このトピックでは、Cisco Secure 動的属性コネクタ のインストールについて説明します。sudo 権限を持つユーザーとしてコネクタをインストールする必要がありますが、非権限ユーザーとしてコネクタを実行できます。

はじめる前に

システムに次の前提条件ソフトウェアがインストールされていることを確認してください。

- Ubuntu 18.04 または 20.04
- CentOS 7 Linux または 8
- Red Hat Enterprise Linux (RHEL) 7 または 8
- Python 3.6.x
- Ansible 2.9 以降

すべてのオペレーティングシステムの最小要件：

- 4 個の CPU
- 8 GB RAM
- 100 GB 使用可能なディスクスペース

vCenter 属性を使用する場合は、次も必要です。

- vCenter 6.7
- 仮想マシンに、VMware ツールがインストールされている必要があります。

前提条件ソフトウェアをインストールするには、[前提条件ソフトウェアのインストール \(2 ページ\)](#) を参照してください。

Readme とリリース ノートの表示

最新のインストール情報については、以下を参照してください。

Readme : <https://galaxy.ansible.com/cisco/csdac>

リリースノート : [Cisco Secure 動的属性コネクタ リリースノート](#)

動的属性コネクタソフトウェアの取得

動的属性コネクタ ソフトウェアの最新バージョンを取得するには、次のコマンドを実行します。

```
ansible-galaxy collection install cisco.csdac
```

ムスター（収集）サービスのインストール

ムスター（収集）サービスは、動的属性コネクタの別名です。

~/ansible/collections/ansible_collections/cisco/csdac ディレクトリから次のコマンドを実行します。

```
ansible-playbook default_playbook.yml [--ask-become-pass] [--extra-vars " vars " ]
```

構文の説明

--ask-become-pass **sudo** パスワードを入力するように求められます。マシンで **sudo** が有効になっている場合は必須です。

--extra-vars 次のオプションの追加変数により、動的属性コネクタがプロキシを使用できるようになります。使用する値は、[前提条件ソフトウェアのインストール（2 ページ）](#)の説明に従って構成した `/etc/environment` の値と一致する必要があります。

- **csdac_proxy_enabled=true**
- **csdac_http_proxy_url=http://PROXY_URL**
csdac_https_proxy_url=PROXY_URL

次のオプションの追加変数は、動的属性コネクタに安全に接続するために使用できる自己署名証明書を作成します。これらのパラメータを省略すると、動的属性コネクタはデフォルトの証明書を使用します。

- **csdac_certificate_domain**
自動生成された証明書のドメイン名。デフォルト値は、ホストの自動検出されたホスト名です (ansible によって検出される)
- **csdac_certificate_country_name**
Two-letter country code. (デフォルト値は us)
- **csdac_certificate_organization_name**
組織名。 (デフォルト値は cisco)
- **csdac_certificate_organization_unit_name**
組織単位名 (デフォルト値は Cisco)

デフォルトの証明書を使用したインストール例

たとえば、デフォルトのオプションでソフトウェアをインストールするには：

```
ansible-galaxy collection install cisco.csdac
cd ~/.ansible/collections/ansible_collections/cisco/csdac
ansible-playbook default_playbook.yml --ask-become-pass
```


オプションの証明書を使用したインストール例

たとえば、オプションの証明書を使用してソフトウェアをインストールするには：

```
ansible-galaxy collection install cisco.csdac
cd ~/.ansible/collections/ansible_collections/cisco/csdac
ansible-playbook default_playbook.yml --ask-become-pass --extra-vars
"csdac_certificate_domain=domain.example.com csdac_certificate_country_name=US
csdac_certificate_organization_name=Cisco
csdac_certificate_organization_unit_name=Engineering"
```

証明書を作成したら、コネクタへのアクセスに使用する Web ブラウザに証明書をインポートします。証明書は ~/csdac/app/config/certs ディレクトリに作成されます。

インストールログの表示

インストールログは次の場所にあります。

```
~/.ansible/collections/ansible_collections/cisco/csdac/logs/csdac.log
```

証明書を使用した **dynamic attributes connector** への接続

証明書とキーがある場合は、それらを仮想マシンの ~/csdac/app/config/certs ディレクトリに配置します。

前のタスクを実行した後、次のコマンドを入力して、dynamic attributes connector の Docker コンテナを再起動します。

```
docker restart muster-ui
```

コネクタへのログイン

1. `https://ip-address` で 動的属性コネクタ にアクセスします。
2. ログインします。

初回ログインのユーザー名は `admin`、パスワードは `admin` です。初めてログインしたときに、パスワードを変更するよう求められます。

Cisco Secure 動的属性コネクタのアップグレード

このトピックでは、Cisco Secure 動的属性コネクタ バージョン 1.0 からバージョン 1.1 にアップグレードする方法について説明します。バージョン 1.0 は Ubuntu 18.04 でのみ利用可能だったため、これらの手順が適用されるオペレーティングシステムは Ubuntu 18.04 だけです。

このトピックでは、以前の Cisco Secure 動的属性コネクタ を現在のバージョンにアップグレードする方法について説明します。これらのタスクは、Cisco Secure 動的属性コネクタ のバージョンやオペレーティングシステムに関係なく実行できます。

ステップ 1 アップグレードするマシンにログインします。

ステップ 2 次のコマンドを入力します。

```
ansible-galaxy collection install cisco.csdac --force
ansible-playbook default_playbook.yml --ask-become-pass
ansible-playbook default_playbook.yml --ask-become-pass [--extra-vars vars]
```

構文の説明

--ask-become-pass **sudo** パスワードを入力するように求められます。マシンで **sudo** が有効になっている場合は必須です。

--extra-vars 次のオプションの追加変数により、動的属性コネクタがプロキシを使用できるようになります。使用する値は、[前提条件ソフトウェアのインストール \(2 ページ\)](#) の説明に従って構成した `/etc/environment` の値と一致する必要があります。

- **csdac_proxy_enabled=true**
- **csdac_http_proxy_url=http://PROXY_URL**
csdac_https_proxy_url=PROXY_URL

次のオプションの追加変数は、動的属性コネクタに安全に接続するために使用できる自己署名証明書を作成します。これらのパラメータを省略すると、動的属性コネクタはデフォルトの証明書を使用します。

- **csdac_certificate_domain**
自動生成された証明書のドメイン名。デフォルト値は、ホストの自動検出されたホスト名です (ansible によって検出される)
- **csdac_certificate_country_name**
Two-letter country code. (デフォルト値は us)
- **csdac_certificate_organization_name**
組織名。 (デフォルト値は cisco)
- **csdac_certificate_organization_unit_name**
組織単位名 (デフォルト値は Cisco)

ステップ 3 アップグレードが完了するまで待ちます。

ステップ 4 アップグレードのログは次の場所にあります。

```
~/ansible/collections/ansible_collections/cisco/csdac/logs/csdac.log
```

次のタスク

「[コネクタの作成](#)」を参照してください。

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。