



# Cisco Secure Dynamic Attributes Connector 설치

이 장에서는 지원되는 모든 운영 체제에 Cisco Dynamic Attributes Connector를 설치하는 방법을 설명합니다.

- [지원되는 운영 체제 및 서드파티 소프트웨어, 1 페이지](#)
- [사전 요건 소프트웨어 설치, 2 페이지](#)
- [Cisco Secure Dynamic Attributes Connector 설치, 7 페이지](#)
- [Cisco Secure Dynamic Attributes Connector 업그레이드, 9 페이지](#)

## 지원되는 운영 체제 및 서드파티 소프트웨어

동적 속성 커넥터에는 다음 요구 사항이 적용됩니다.

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

모든 운영 체제에 대한 최소 요구 사항:

- CPU 4개
- 8GB RAM
- 100GB 사용 가능한 디스크 공간

다음과 같이 가상 머신의 크기를 조정하는 것이 좋습니다.

- 커넥터 50개, 커넥터당 필터 5개 및 워크로드 20,000개 가정: CPU 4개, 8GB RAM; 100GB의 사용 가능한 디스크 공간

- 커넥터 125개, 커넥터당 필터 5개 및 워크로드 50,000개 가정: CPU 8개, 16GBRAM, 사용 가능한 디스크 공간 100GB



참고 가상 머신의 크기를 적절하게 조정하지 않으면 동적 속성 커넥터가 실패하거나 시작되지 않을 수 있습니다.

vCenter 속성을 사용하려면 다음 항목도 필요합니다.

- vCenter 6.7
- 가상 머신에 VMware Tools를 설치해야 합니다.

## 사전 요건 소프트웨어 설치

시작하기 전에

물리적 또는 가상 설정이 있고 시스템이 온프레미스 Firewall Management Center 또는 클라우드 사용 Firewall Management Center와 통신할 수 있는지 확인합니다.

단계 1 (선택 사항). Ubuntu 시스템이 인터넷 프록시 뒤에 있는 경우 텍스트 편집기를 사용하여 `/etc/environment`를 편집하여 다음 변수를 내보내 인터넷과의 통신을 활성화합니다.

| 변수                              | 값  |
|---------------------------------|--|
| <code>export http_proxy</code>  | HTTP 프록시와 함께 사용합니다.<br><code>user:pass@host-or-ip:port</code>        |
| <code>export https_proxy</code> | HTTPS 프록시와 함께 사용합니다.<br><code>user:pass@host-or-ip:port</code>       |
| <code>export no_proxy</code>    | 프록시 구성을 제거합니다.<br><code>export no_proxy="localhost,127.0.0.1"</code> |

예:

다음 인증을 사용하지 않는 HTTP 프록시

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

다음 인증을 사용하는 HTTPS 프록시:

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

단계 2 다른 명령 창을 사용하여 설정을 확인합니다.

```
env grep | proxy
```

결과 예:

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

단계 3 다음 섹션 중 하나를 계속 진행합니다.

#### 관련 항목

[사전 요건 소프트웨어 설치 - Ubuntu](#), 5 페이지

[사전 요건 소프트웨어 설치 - CentOS](#), 3 페이지

[사전 요건 소프트웨어 설치 - RHEL](#), 4 페이지

## 사전 요건 소프트웨어 설치 - CentOS

#### 시작하기 전에

다음은 모두 수행합니다.

- 시스템이 지원되는 운영 체제 및 서드파티 소프트웨어, 1 페이지에 설명된 사전 요건을 충족하는지 확인합니다.
- (선택 사항). 동적 속성 커넥터에 대한 프록시 액세스가 필요한 경우 [사전 요건 소프트웨어 설치](#), 2 페이지의 내용을 참조하십시오.

단계 1 Docker가 설치되어 있지 않은지 확인하고 설치되어 있으면 제거합니다.

```
docker --version
```

Docker가 설치되어 있으면 [Ubuntu에서 Docker 엔진 제거](#)에 설명된 대로 제거합니다.

단계 2 저장소를 업데이트하고 업그레이드합니다.

CentOS 7:

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

단계 3 epel 저장소를 설치합니다.

CentOS 7:

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

단계 4 (CentOS 7만 해당) Python 3을 설치합니다.

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

단계 5 Ansible을 설치합니다.

CentOS 7:

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

단계 6 Ansible 버전이 2.9 이상인지 확인합니다.

**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)]
```

참고 위의 출력에 나와 있는 것처럼 Ansible에서 Python 2.x를 참조하는 것은 정상입니다. 커넥터는 여전히 Python 3을 사용합니다.

다음에 수행할 작업

[Cisco Secure Dynamic Attributes Connector 설치, 7 페이지](#)에 설명된 대로 커넥터를 설치합니다.

필요에 따라 동적 속성 커넥터에서 프록시 사용을 중지하려면 `/etc/environment`를 편집하고 프록시 구성을 제거합니다.

## 사전 요건 소프트웨어 설치 - RHEL

시작하기 전에

다음은 모두 수행합니다.

- 시스템이 [지원되는 운영 체제 및 서드파티 소프트웨어, 1 페이지](#)에 설명된 사전 요건을 충족하는지 확인합니다.
- (선택 사항). 동적 속성 커넥터에 대한 프록시 액세스가 필요한 경우 [사전 요건 소프트웨어 설치, 2 페이지](#)의 내용을 참조하십시오.

**단계 1** Docker가 설치되어 있지 않은지 확인하고 설치되어 있으면 제거합니다.

```
docker --version
```

Docker가 설치되어 있으면 [Ubuntu에서 Docker 엔진 제거](#)에 설명된 대로 제거합니다.

**단계 2** 저장소를 업데이트합니다.

RHEL 7:

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

RHEL 8:

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

**단계 3** epel 저장소를 설치합니다.

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
```

단계 4 (RHEL 7만 해당) Python 3을 설치합니다.

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

단계 5 Ansible을 설치합니다.

RHEL 7:

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

RHEL 8:

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

단계 6 Ansible 버전을 확인합니다.

```
ansible --version
```

예는 다음과 같습니다.

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)]
```

참고 위의 출력에 나와 있는 것처럼 Ansible에서 Python 2.x를 참조하는 것은 정상입니다. 커넥터는 여전히 Python 3을 사용합니다.

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)]
```

다음에 수행할 작업

[Cisco Secure Dynamic Attributes Connector 설치, 7 페이지](#)에 설명된 대로 커넥터를 설치합니다.

필요에 따라 동적 속성 커넥터에서 프록시 사용을 중지하려면 `/etc/environment`를 편집하고 프록시 구성을 제거합니다.

## 사전 요건 소프트웨어 설치 - Ubuntu

단계 1 Docker가 설치되어 있지 않은지 확인하고 설치되어 있으면 제거합니다.

```
docker --version
```

Docker가 설치되어 있으면 [Ubuntu에서 Docker 엔진 제거](#)에 설명된 대로 제거합니다.

**단계 2** 저장소를 업데이트합니다.

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

**단계 3** Python 버전을 확인합니다.

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

버전이 3.6 이전인 경우 버전 3.6.x를 설치해야 합니다.

**단계 4** Python 3.6을 설치합니다.

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

**단계 5** 공통 라이브러리를 설치합니다.

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

**단계 6** Ansible을 설치합니다.

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

**단계 7** Ansible 버전을 확인합니다.

```
ansible --version
```

예는 다음과 같습니다.

```
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]
```

**참고** 위의 출력에 나와 있는 것처럼 Ansible에서 Python 2.x를 참조하는 것은 정상입니다. 커넥터는 여전히 Python 3.6을 사용합니다.

다음에 수행할 작업

[Cisco Secure Dynamic Attributes Connector 설치](#), 7 페이지에 설명된 대로 커넥터를 설치합니다.

필요에 따라 동적 속성 커넥터에서 프록시 사용을 중지하려면 `/etc/environment`를 편집하고 프록시 구성을 제거합니다.

# Cisco Secure Dynamic Attributes Connector 설치

## 설치 정보

이 항목에서는 Cisco Secure Dynamic Attributes Connector 설치에 대해 설명합니다. `sudo` 권한이 있는 사용자로 커넥터를 설치해야 하지만 권한이 없는 사용자로도 실행할 수 있습니다.

## 시작하기 전에

시스템에 다음 사전 요건 소프트웨어가 있는지 확인합니다.

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

모든 운영 체제에 대한 최소 요구 사항:

- CPU 4개
- 8GB RAM
- 100GB 사용 가능한 디스크 공간

다음과 같이 가상 머신의 크기를 조정하는 것이 좋습니다.

- 커넥터 50개, 커넥터당 필터 5개 및 워크로드 20,000개 가정: CPU 4개, 8GB RAM; 100GB의 사용 가능한 디스크 공간
- 커넥터 125개, 커넥터당 필터 5개 및 워크로드 50,000개 가정: CPU 8개, 16GBRAM, 사용 가능한 디스크 공간 100GB



**참고** 가상 머신의 크기를 적절하게 조정하지 않으면 동적 속성 커넥터가 실패하거나 시작되지 않을 수 있습니다.

vCenter 속성을 사용하려면 다음 항목도 필요합니다.

- vCenter 6.7
- 가상 머신에 VMware Tools를 설치해야 합니다.

사전 요건 소프트웨어를 설치하려면 [사전 요건 소프트웨어 설치, 2 페이지](#) 섹션을 참조하십시오.

**Readme**와 릴리스 노트 보기

최신 설치 정보는 다음을 참조하십시오.

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

릴리스 노트: [Cisco Secure Dynamic Attributes Connector 릴리스 노트](#)

**Dynamic Attributes Connector** 소프트웨어 다운로드

최신 버전의 동적 속성 커넥터 소프트웨어를 다운로드하려면 다음 명령을 실행합니다.

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

## 소집 서비스 설치

소집 서비스는 동적 속성 커넥터의 또 다른 이름입니다.

~/ansible/collections/ansible\_collections/cisco/csdac 디렉터리에서 다음 명령을 실행합니다.

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

**Syntax Description**

**--ask-become-pass** sudo 비밀번호를 입력하라는 프롬프트가 표시됩니다. 시스템에서 sudo가 활성화된 경우 필요합니다.

**--extra-vars** 다음의 선택적 추가 변수는 동적 속성 커넥터가 프록시를 사용할 수 있도록 합니다. 사용하는 값은 [사전 요건 소프트웨어 설치, 2 페이지](#)에서 설명한 대로 구성된 /etc/environment의 값과 일치해야 합니다.

- **csdac\_proxy\_enabled=true**
- **csdac\_http\_proxy\_url=http://PROXY\_URL**  
**csdac\_https\_proxy\_url=PROXY\_URL**

다음 추가 변수는 동적 속성 커넥터에 안전하게 연결하는 데 사용할 수 있는 자체 서명 인증서를 생성합니다. 이러한 매개 변수를 생략하면 동적 속성 커넥터는 기본 인증서를 사용합니다.

- **csdac\_certificate\_domain**  
자동 생성된 인증서의 도메인 이름입니다. 기본값은 호스트의 자동 탐지 호스트 이름(ansible에 의해 탐지됨)입니다.
- **csdac\_certificate\_country\_name**  
2자로 된 국가 코드. (기본값은 US)
- **csdac\_certificate\_organization\_name**  
조직의 이름입니다. (기본값은 Cisco)
- **csdac\_certificate\_organization\_unit\_name**  
조직 단위 이름(기본값은 Cisco)



### 기본 인증서를 사용한 설치 예

예를 들어 기본 옵션으로 소프트웨어를 설치하려면 다음을 수행합니다.

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

### 선택적 인증서를 사용하는 설치 예

예를 들어, 선택적 인증서를 사용하여 소프트웨어를 설치하려면 다음을 수행합니다.

```
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"
```

인증서를 생성한 후 커넥터에 액세스하는 데 사용할 웹 브라우저로 가져옵니다. 인증서는 `~/csdac/app/config/certs` 디렉터리에 생성됩니다.

### 설치 로그 보기

설치 로그는 다음 위치에 있습니다.

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

인증서를 사용하여 동적 속성 커넥터에 연결합니다.

인증서와 키가 있는 경우 가상 머신의 `~/csdac/app/config/certs` 디렉터리에 저장합니다.

이전 작업을 수행한 후 다음 명령을 입력하여 동적 속성 커넥터의 Docker 컨테이너를 다시 시작합니다.

```
docker restart muster-ui
```

### 커넥터에 로그인

1. `https://ip-address`에서 동적 속성 커넥터에 액세스합니다.
2. 로그인합니다.

초기 로그인은 사용자 이름 `admin`, 비밀번호는 `admin`입니다. 처음 로그인할 때 비밀번호를 변경해야 합니다.

## Cisco Secure Dynamic Attributes Connector 업그레이드

이 항목에서는 이전 Cisco Secure Dynamic Attributes Connector에서 현재 버전으로 업그레이드하는 방법을 설명합니다. 이러한 작업은 Cisco Secure Dynamic Attributes Connector 버전 또는 운영 체제에 관계없이 수행할 수 있습니다.

단계 1 업그레이드할 시스템에 로그인합니다.

단계 2 다음 명령을 입력합니다.

```
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]
```

**Syntax Description** **--ask-become-pass** **sudo** 비밀번호를 입력하라는 프롬프트가 표시됩니다. 시스템에서 **sudo**가 활성화된 경우 필요합니다.

**--extra-vars** 다음의 선택적 추가 변수는 동적 속성 커넥터가 프록시를 사용할 수 있도록 합니다. 사용하는 값은 [사전 요건 소프트웨어 설치, 2 페이지](#)에서 설명한 대로 구성된 `/etc/environment`의 값과 일치해야 합니다.

- **csdac\_proxy\_enabled=true**
- **csdac\_http\_proxy\_url=http://PROXY\_URL**  
**csdac\_https\_proxy\_url=PROXY\_URL**

다음 추가 변수는 동적 속성 커넥터에 안전하게 연결하는 데 사용할 수 있는 자체 서명 인증서를 생성합니다. 이러한 매개 변수를 생략하면 동적 속성 커넥터는 기본 인증서를 사용합니다.

- **csdac\_certificate\_domain**  
자동 생성된 인증서의 도메인 이름입니다. 기본값은 호스트의 자동 탐지 호스트 이름(ansible에 의해 탐지됨)입니다.
- **csdac\_certificate\_country\_name**  
2자로 된 국가 코드. (기본값은 `us`)
- **csdac\_certificate\_organization\_name**  
조직의 이름입니다. (기본값은 `Cisco`)
- **csdac\_certificate\_organization\_unit\_name**  
조직 단위 이름(기본값은 `Cisco`)

단계 3 업그레이드가 완료될 때까지 기다립니다.

단계 4 업그레이드 로그는 다음 위치에서 사용할 수 있습니다.

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

다음에 수행할 작업

[커넥터 생성의 내용을 참조하십시오.](#)

## 번역에 관하여

Cisco는 일부 지역에서 본 콘텐츠의 현지 언어 번역을 제공할 수 있습니다. 이러한 번역은 정보 제공의 목적으로만 제공되며, 불일치가 있는 경우 본 콘텐츠의 영어 버전이 우선합니다.