



Ansible

Ansibleは、クラウドプロビジョニング、構成管理、アプリケーションの展開、サービス内オーケストレーション、およびその他の IT ニーズを自動化するオープンソースの IT 自動化エンジンです。Puppet や Chef と同様に、Ansible を使用すると、管理者はさまざまなタイプのサーバー環境の管理、自動化、およびオーケストレーションを行えます。Ansible はエージェントレスであり、デバイスを自動化するためにターゲットノード（サーバーまたはスイッチ）にソフトウェア エージェントをインストールする必要はありません。通常、Ansible は管理対象のターゲット サーバーで SSH と Python をサポートする必要がありますが、MDS スイッチの場合、Ansible は SSH と NX-API の両方を使用するように拡張されているため、スイッチ上の Python サポートは必要ありません。Ansible プレイブックは YAML で記述されているため、自動化ジョブを読みやすい形式で記述することができます。各 Ansible プレイブック内では、さまざまな Ansible モジュールを使用できます。

次に、cisco.nxos コレクション内の MDS 固有のモジュールを示します：

- nxos_vsan
(https://docs.ansible.com/ansible/latest/collections/cisco/nxos/nxos_vsan_module.html#ansible-collections-cisco-nxos-nxos-nsan-module)
- nxos_zone_zoneset
(https://docs.ansible.com/ansible/latest/collections/cisco/nxos/nxos_zone_zoneset_module.html#ansible-collections-cisco-nxos-nxos-zone-zoneset-module)
- nxos_devicealias
(https://docs.ansible.com/ansible/latest/collections/cisco/nxos/nxos_devicealias_module.html#ansible-collections-cisco-nxos-nxos-devicealias-module)
- nxos_fc_interfaces
(https://docs.ansible.com/ansible/latest/collections/cisco/nxos/nxos_fc_interfaces_module.html#ansible-collections-cisco-nxos-nxos-fc-interfaces-module)

任意のコマンドを実行するには、nxos_command モジュール

(https://docs.ansible.com/ansible/latest/collections/cisco/nxos/nxos_command_module.html#ansible-collections-cisco-nxos-nxos-command-module)

を使用します。

Cisco MDS のサポートが制限されている他のモジュールもあります。Cisco MDS との互換性については、個々のモジュールのマニュアルを参照してください：<https://docs.ansible.com/ansible/latest/collections/cisco/nxos/index.html#modules>

Ansible モジュールは、SSH 接続または NX-API コールを行い、リアルタイムの状態データを収集し、Cisco MDS デバイスの構成を変更します。Ansible の詳細については、[Ansible の公式ドキュメント](#)を参照してください。



(注) Cisco MDS Ansible モジュールの場合、ターゲット ノードに Python インタープリタは必要ありません。

Ansible でサポートされる Cisco MDS モジュールの詳細については、[Ansible モジュール](#)を参照してください。

- [はじめに \(2 ページ\)](#)
- [ホスト ファイル \(2 ページ\)](#)
- [資料 \(3 ページ\)](#)
- [プレイブックの例 \(3 ページ\)](#)

はじめに

Ansible のインストールについては、公式の [Ansible インストールガイド](#)、https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html を参照してください。

ホスト ファイル

ホスト ファイルには、管理対象のデバイスがリストされます。1つのデバイスを1つだけのグループに含めることも、複数のグループに含めることもできます。次のホストファイルでは、2つのデバイス `mds1` と `mds2` を持つ `edge` と呼ばれる単一のグループが使用されています。接続は、HTTPS 接続を使用してターゲット デバイスに接続する NX-API に設定されます。ユーザー名とパスワードはホスト ファイルに保存されます。セキュリティを強化するために、このホストファイルは **Ansible Vault** を使用して暗号化できますが、ここでは使用しません。



(注) Ansible 2.5 から、`ansible_connection: local` は廃止されました。代わりに `ansible_connection: ansible.netcommon.network_cli` または `ansible_connection: ansible.netcommon.httpapi` を使用します。`httpapi` 接続プラグインは、さまざまなトグルを提供します。使用可能なすべてのオプションは、[Ansible のドキュメント](#)で指定されています。`network_cli` 接続プラグインは、さまざまなトグルを提供します。使用可能なすべてのオプションは、[Ansible のドキュメント](#)で指定されています。

```
$ cat /etc/ansible/hosts
[all:vars]
ansible_connection = ansible.netcommon.httpapi
ansible_httpapi_use_ssl=True
ansible_httpapi_port=8443
ansible_user=username
ansible_password=password

[edge]
```

```
mds1
mds2
```

資料

すべての Cisco MDS NX-OS モジュールのマニュアルは、<https://docs.ansible.com/ansible/latest/collections/cisco/nxos/> から入手できます。または、組み込みのマニュアル ツールを使用して端末で表示できます。

```
$ansible-doc
```

プレイブックの例

この最初のプレイブックでは、いくつかの VSAN をプロビジョニングし、VSAN を削除します。**nxos_vsan** という Ansible モジュールを使用して、このタスクを自動化します。

以下に示すように、プレイブックは YAML で定義されています。

```
---
- name: Test that vsan module works
  gather_facts: no
  hosts:
    - mds1
  cisco.nxos.nxos_vsan:
    vsan:
      - id: 922
        interface:
          - fc1/1
          - fc1/2
          - port-channel 1
        name: vsan-SAN-A
        remove: false
        suspend: false
      - id: 923
        interface:
          - fc1/11
          - fc1/21
          - port-channel 2
        name: vsan-SAN-B
        remove: false
        suspend: true
      - id: 1923
        name: vsan-SAN-Old
        remove: true
    register: result
  - debug: var=result
```

上記のプレイブックが **vsan.yml** と呼ばれると仮定すると、このタスクはターミナルから以下に示すように実行できます。

```
$ ansible-playbook vsan.yml
```

```
PLAY [VSAN TEST (NXOS)] *****

TASK [Test that vsan module works] *****
changed: [mds1]
```

```

TASK [debug] *****
ok: [mds1] => {
  "result": {
    "changed": true,
    "cmds": [
      "terminal dont-ask",
      "vsan database",
      "vsan 922",
      "vsan 922 name vsan-SAN-A",
      "no vsan 922 suspend",
      "vsan database",
      "vsan 922 interface fc1/1",
      "vsan 922 interface fc1/2",
      "vsan 922 interface port-channel 1",
      "vsan database",
      "vsan 923",
      "vsan 923 name vsan-SAN-B",
      "vsan 923 suspend",
      "vsan database",
      "vsan 923 interface fc1/11",
      "vsan 923 interface fc1/21",
      "vsan 923 interface port-channel 2",
      "vsan database",
      "no vsan 1923",
      "no terminal dont-ask"
    ],
    "failed": false,
    "messages": [
      "creating vsan 922",
      "setting vsan name to vsan-SAN-A for vsan 922",
      "no suspending the vsan 922",
      "adding interface fc1/1 to vsan 922",
      "adding interface fc1/2 to vsan 922",
      "adding interface port-channel 1 to vsan 922",
      "creating vsan 923",
      "setting vsan name to vsan-SAN-B for vsan 923",
      "suspending the vsan 923",
      "adding interface fc1/11 to vsan 923",
      "adding interface fc1/21 to vsan 923",
      "adding interface port-channel 2 to vsan 923",
      "deleting the vsan 1923"
    ]
  }
}

PLAY RECAP *****
mds1                : ok=2    changed=1    unreachable=0    failed=0

```

まとめ

Ansible を使用して VSAN を設定/設定解除する方法を見てきました。これは単純な例ですが、自動化が必要なさまざまなタスクでモジュールを使用できます。

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。