

A10のAnsibleモジュールと スタートアップガイド

A10ネットワークス株式会社
2020年1月



Reliable Security Always™

AnsibleによるThunderの構成管理

- **Ansible : エージェントレスな構成管理ツール**
 - Playbook : 実行主体・実行タスクと実行後の結果を記述したもの
 - モジュール : Playbookから呼び出されて実行されるスクリプト
 - 冪等性 : Playbookの実行回数に関わらず結果が変わらない特性
- **AnsibleによるThunderの構成管理方法**
 - 方法① : a10_*モジュールを利用してPlaybookを記述
 - 方法② : uriモジュールを利用したPlaybook内でaXAPI*を実行



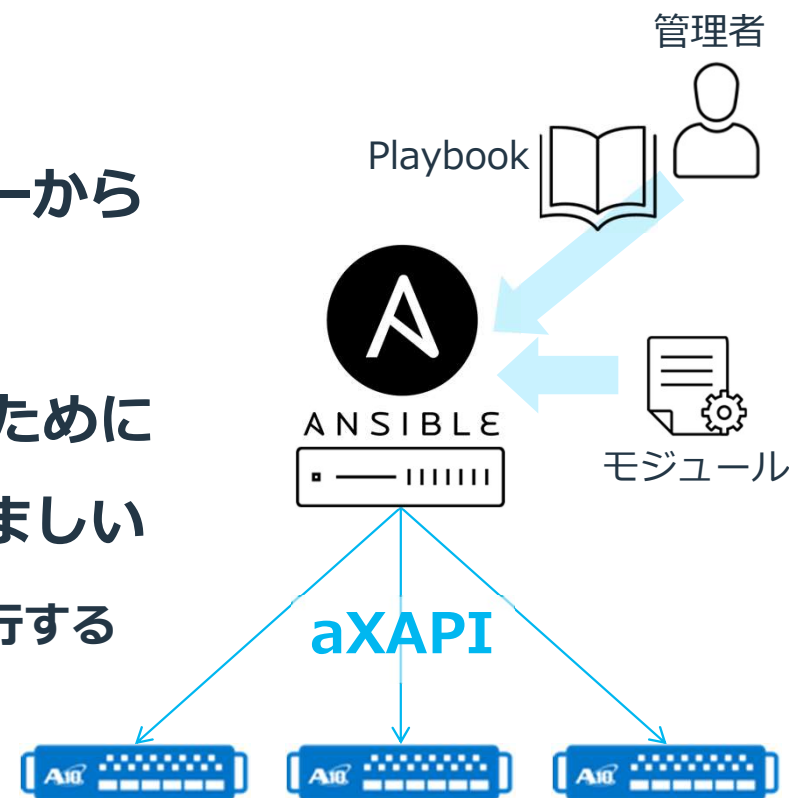
*aXAPI: Thunderの操作が可能なREST API

A10のAnsibleモジュール

- 公式のGitHubでモジュールを公開中
<https://github.com/a10networks/a10-ansible>
- 1,628のモジュール
- 447のPlaybook例
- 利用時の注意点
 - PythonのRequestsモジュールが必要
 - aXAPIv3のみ対象（ACOS4.x以降に対応）
 - 冪等性の担保/Dry Run対応がされていないモジュールもある
 - aVCSに未対応
 - 自動ではwrite memoryされない
(Write memoryのモジュールを実行する必要がある)

a10_*モジュールを利用した場合の基本動作

- A10上にPython実行環境が無いため、
実質上はAnsible Coreの動作しているサーバーから
Thunderに対しaXAPIで操作が実行される
- Playbook上では処理対象を分かりやすくするために
ThunderのホストIPを指定した方が運用上望ましい
 - connection: localを指定し、ローカルで処理を実行する



Playbookの記述例

```
---
- hosts: 10.255.211.70
  connection: local
  gather_facts: no

vars:
  a10_host: "10.255.211.70"
  a10_username: "admin"
  a10_password: "a10"
  a10_port: 443
  a10_protocol: "https"

tasks:
- name: Create a10_network_vlan instance
  a10_network_vlan:
    a10_host: "{{ a10_host }}"
    a10_port: "{{ a10_port }}"
    a10_protocol: "{{ a10_protocol }}"
    a10_username: "{{ a10_username }}"
    a10_password: "{{ a10_password }}"
    a10_partition:
      name: "shared"
      shared: True
    vlan_num: 10
    untagged_eth_list:
      - untagged_ethernet_start: 1
        untagged_ethernet_end: 1
    ve: 10
    user_tag: "test"
    state: present
```

対象ホストの指定、connection: local指定によるローカルでの実行指定、事前の情報取得はできないのでgather-facts: noに

変数の設定（インベントリ等外部ファイルに記述しても良い）

利用モジュールの指定（この例ではa10_network_vlanモジュール）

全モジュール共通の設定項目

パーティションを指定したい場合はdict型で指定（sharedの場合は省略可）

VLAN番号の設定

list型の項目設定（VLANに割り当てるethernetポート）

VE番号の設定

user-tagの設定

present/absentで設定有無を指定。設定項目を追加/Updateする場合はpresent

Playbookの実行例

```
# ansible-playbook -i hosts a10_network_vlan_create.yaml -v
```

```
PLAY [10.255.211.70] *****
```

```
TASK [Create a10_network_vlan instance] *****
```

```
changed: [10.255.211.70] => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"}, "changed": true, "message": "",  
"original_message": "", "result": {}, "vlan": {"shared-vlan": 0, "untagged-eth-list": [{"untagged-ethernet-end": 1, "untagged-ethernet-  
start": 1}], "user-tag": "test", "uuid": "530ba0f2-3254-11ea-b67a-adc5c9437e0e", "ve": 10, "vlan-num": 10}}
```

```
PLAY RECAP *****
```

```
10.255.211.70 : ok=1 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

初回の実行で構成変更されchanged=1で返る

```
# ansible-playbook -i hosts a10_network_vlan_create.yaml -v
```

```
PLAY [10.255.211.70] *****
```

```
TASK [Create a10_network_vlan instance] *****
```

```
ok: [10.255.211.70] => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"}, "changed": false, "message": "",  
"original_message": "", "result": {}, "vlan": {"shared-vlan": 0, "untagged-eth-list": [{"untagged-ethernet-end": 1, "untagged-ethernet-  
start": 1}], "user-tag": "test", "uuid": "530ba0f2-3254-11ea-b67a-adc5c9437e0e", "ve": 10, "vlan-num": 10}}
```

```
PLAY RECAP *****
```

```
10.255.211.70 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

2回実行した場合も冪等性が保たれている (ok=1, changed=0で返る)

Python、Ansibleと A10用Ansibleモジュールの インストール

CentOS 7.6.1810でPython3.6を利用する場合の例
(本例では全作業をrootで実施)

*A10のモジュールはPython2.7でも動作可能

gitのインストールおよびモジュールのClone

```
# yum install git
```

} gitのインストール

```
...  
# git clone https://github.com/a10networks/a10-ansible a10-ansible  
Cloning into 'a10-ansible'...
```

```
remote: Enumerating objects: 174, done.
```

```
remote: Counting objects: 100% (174/174), done.
```

```
remote: Compressing objects: 100% (103/103), done.
```

```
remote: Total 9320 (delta 148), reused 95 (delta 71), pack-reused 9146
```

```
Receiving objects: 100% (9320/9320), 2.06 MiB | 746.00 KiB/s, done.
```

```
Resolving deltas: 100% (8117/8117), done.
```

```
# cd a10-ansible/
```

```
# git pull
```

```
Already up-to-date.
```

```
# cd
```

} 将来的にGitHub上のモジュールが反映されたらgit pullで取り込むことができる

} Ansibleモジュールの
Clone

Python3のインストールとモジュールのインストール

```
# yum install -y https://centos7.iuscommunity.org/ius-release.rpm
...
# yum install python36u python36u-libs python36u-devel python36u-pip
...
# pip3.6 install -e a10-ansible/
Obtaining file:///root/a10-ansible
Installing collected packages: a10-ansible
  Running setup.py develop for a10-ansible
Successfully installed a10-ansible
# pip3.6 install requests
...
# ln -sf /usr/bin/python3.6 /usr/bin/python /usr/bin/pytonがpython3.6を指すようにシンボリックリンクを修正
# python --version
Python 3.6.8
# vi /etc/profile
(以下の2行を最終行に追記し保存)
ANSIBLE_LIBRARY=/root/a10-ansible/a10_ansible/library
export ANSIBLE_LIBRARY
# source /etc/profile
```

} リポジトリのインストール

} Python3.6のインストール

} A10用Ansibleモジュールのインストール

} PythonのRequestsモジュールのインストール

} Pythonのバージョン確認

} モジュールパスの設定と反映

Ansibleのインストール

```
# pip3.6 install ansible } Ansibleのインストール
```

```
...  
# pip3.6 show ansible
```

```
Name: ansible
```

```
Version: 2.8.2
```

```
Summary: Radically simple IT automation
```

```
Home-page: https://ansible.com/
```

```
Author: Ansible, Inc.
```

```
Author-email: info@ansible.com
```

```
License: GPLv3+
```

```
Location: /usr/lib/python3.6/site-packages
```

```
Requires: jinja2, PyYAML, cryptography
```

```
Required-by:
```

```
# ansible -version
```

```
ansible 2.8.2
```

```
config file = /etc/ansible/ansible.cfg
```

```
...
```

```
# ansible-doc -t module a10_slb_server
```

```
...
```

インストール状況の確認

バージョンの確認

モジュールパスが正しく設定されているか確認
(ドキュメントが表示されればOK)

Playbookの作成とテスト

```
# mkdir playbook } Playbook用ディレクトリの作成と移動
# cd playbook
# vi hosts
  (以下のような操作対象のインベントリを作成し保存)
[vThunder]
10.255.211.110
10.255.211.115
# vi a10_network_vlan_create.yaml } Playbookの作成
  (次ページ以降のPlaybookを記述し保存)
# ansible-playbook -i hosts a10_network_vlan_create.yaml } Playbookの実行
...
  (-vや-vvvを付けて実行すると
  詳細なログを表示できる)
```

a10_network_vlan_create.yamlの例

```
---
- hosts: 10.255.211.110
  connection: local
  gather_facts: no

  vars:
    a10_host: "10.255.211.110"
    a10_username: "admin"
    a10_password: "a10"
    a10_port: "443"
    a10_protocol: "https"
  tasks:
    - name: Create VLAN
      a10_network_vlan:
        a10_host: "{{ a10_host }}"
        a10_port: "{{ a10_port }}"
        a10_protocol: "{{ a10_protocol }}"
        a10_username: "{{ a10_username }}"
        a10_password: "{{ a10_password }}"
        vlan_num: 10
        untagged_eth_list:
          - untagged_ethernet_start: 1
            untagged_ethernet_end: 1
            ve: 10
            user_tag: "test"
            state: present

    - name: Write memory
      a10_write_memory:
        a10_host: "{{ a10_host }}"
        a10_port: "{{ a10_port }}"
        a10_protocol: "{{ a10_protocol }}"
        a10_username: "{{ a10_username }}"
        a10_password: "{{ a10_password }}"
        state: present
        partition: all
```

a10_network_vlan_create.yamlの実行例

```
# ansible-playbook -i hosts a10_network_vlan_create.yaml
```

```
PLAY [10.255.211.110]
```

```
*****
```

```
TASK [Create VLAN]
```

```
*****
```

```
changed: [10.255.211.110]
```

```
TASK [Write memory]
```

```
*****
```

```
changed: [10.255.211.110]
```

```
PLAY RECAP
```

```
*****
```

```
10.255.211.110 : ok=2 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

a10_network_vlan_delete.yamlの例

```
---
- hosts: 10.255.211.110
  connection: local
  gather_facts: no

  vars:
    a10_host: "10.255.211.110"
    a10_username: "admin"
    a10_password: "a10"
    a10_port: "443"
    a10_protocol: "https"
  tasks:
    - name: Create VLAN
      a10_network_vlan:
        a10_host: "{{ a10_host }}"
        a10_port: "{{ a10_port }}"
        a10_protocol: "{{ a10_protocol }}"
        a10_username: "{{ a10_username }}"
        a10_password: "{{ a10_password }}"
        vlan_num: 10
        untagged_eth_list:
          - untagged_ethernet_start: 1
            untagged_ethernet_end: 1
            ve: 10
            user_tag: "test"
            state: absent ← 設定を削除する場合はabsentに

    - name: Write memory
      a10_write_memory:
        a10_host: "{{ a10_host }}"
        a10_port: "{{ a10_port }}"
        a10_protocol: "{{ a10_protocol }}"
        a10_username: "{{ a10_username }}"
        a10_password: "{{ a10_password }}"
        state: present
        partition: all
```

補足 : Playbookが動作しない場合の確認ポイント（１）

- インベントリは正しく設定されているか
- スペースの代わりにTabを使っていないか
- Playbook内の必須パラメーターは設定されているか
- 同じ設定を再設定しようとしていないか
（冪等性がないモジュールの場合）
- Playbook内のstate（presentまたはabsent）が正しく設定されているか
- パラメーターの記述（特にリスト型パラメーターの記述）が正しいか

補足：Playbookが動作しない場合の確認ポイント（2）

- PythonのRequestモジュールがインストールされているか
- Pythonのバージョンやそれに対応したAnsibleモジュールがインストールされているか、各種パスが通っているか
- A10提供モジュールに未修正の誤った参照が存在していないか

```
f_dict["port-num"] = module.params["port-num"]
```

のような記述が含まれる場合、以下のように修正すれば多くの場合動作する

```
f_dict["port-num"] = module.params["port_num"]
```

（Ansibleの制約でPlaybookの最上位のパラメータは「-」が利用できない）

- Pythonのバージョンに対応していない記述がモジュールに含まれていないか
- 上記でも解決しない場合は、AnsibleとPythonのバージョンと併せてPlaybookそのものと-vや-vvvを付与したPlaybookの実行結果をA10に送付

Thank You



Reliable Security Always™