

# A10のAnsibleモジュールと スタートアップガイド

---

2020年6月

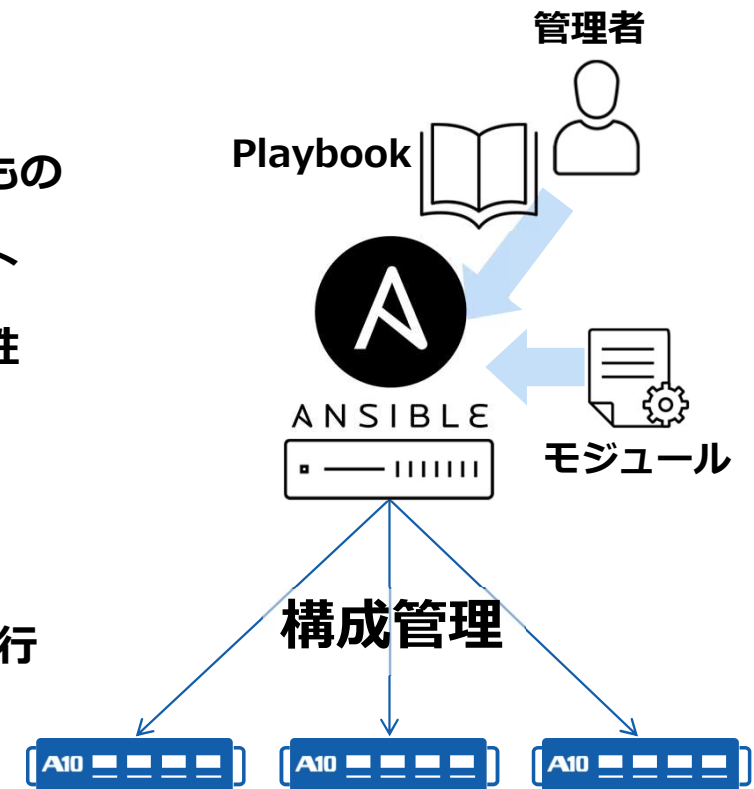
A10ネットワークス株式会社



Always Secure. Always Available.

# 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提供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



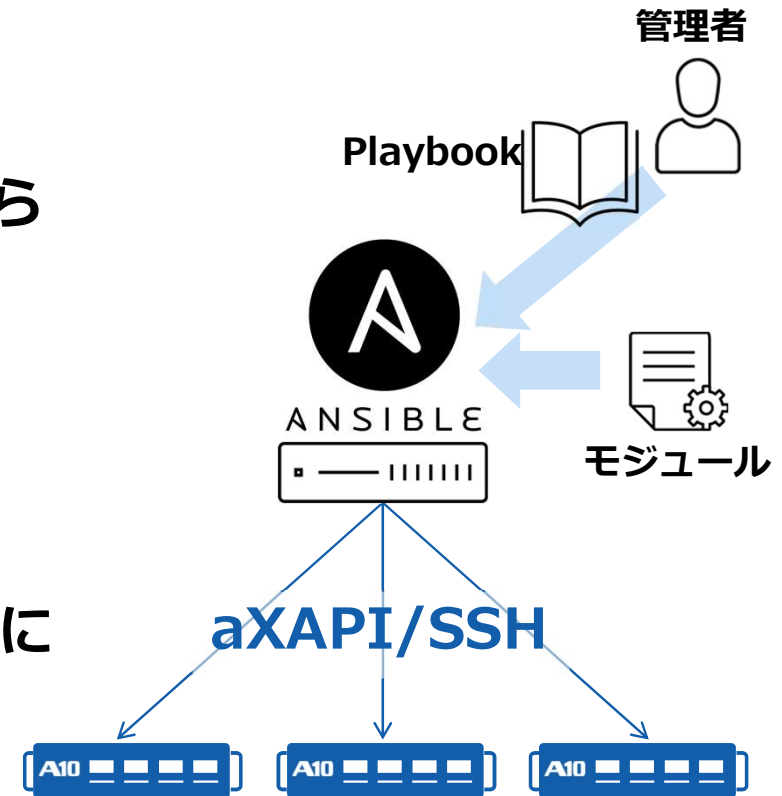
\*aXAPI: Thunderの操作が可能なREST API

# A10のAnsibleモジュール

- 公式のGitHubでaXAPIベースのモジュールを公開中  
<https://github.com/a10networks/a10-acos-axapi>
  - 1,600超のモジュール
  - 多数のPlaybook例
- CLIベースのモジュールも以下で公開されている  
[https://galaxy.ansible.com/a10/acos\\_cli](https://galaxy.ansible.com/a10/acos_cli)
  - 他のネットワーク系モジュールと同様のモジュール
  - acos\_command, acos\_config, acos\_factsの3種類
- 参考トレーニングコンテンツ  
[https://github.com/kishizuka4989/ansible\\_training\\_a10\\_thunder\\_adc/blob/master/README.ja.md](https://github.com/kishizuka4989/ansible_training_a10_thunder_adc/blob/master/README.ja.md)

# A10提供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。  
CLIベースのモジュールの場合は、Thunderに  
SSHで接続してコマンドが投入される
- 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
  collections:
    - a10.acos_axapi

vars:
  ansible_host: "10.255.211.70"
  ansible_username: "admin"
  ansible_password: "a10"
  ansible_port: 443

tasks:
- name: Create VLAN
  a10_network_vlan:
    ansible_host: "{{ ansible_host }}"
    ansible_port: "{{ ansible_port }}"
    ansible_username: "{{ ansible_username }}"
    ansible_password: "{{ ansible_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  
Ansible collectionで利用するモジュールセットを指定

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 VLAN] *****
changed: [10.255.211.70]

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、ok=1, changed=1で返る

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

PLAY [10.255.211.70] *****

TASK [Create VLAN] *****
ok: [10.255.211.70]

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

2回同じモジュールを実行した場合は構成変更はされず、  
期待した構成になっていることが確認される = 冪等性が保たれている  
(ok=1, changed=0で返る)



# Python, Ansibleと A10用モジュールのインストール

- CentOS 7.8でAnsible 2.9.9とPython 2.7\*を利用する場合の例  
(本例では全作業をrootで実施)

**A10**

Always Secure. Always Available.

\*2020年5月時点でAnsible 2.9とPython 3.xを利用すると  
Ansible collectionがうまく動作しないため留意が必要  
<https://github.com/ansible/ansible/issues/68361>

# 各種ツールとAnsibleのインストール

```
# yum install git } gitのインストール
...
# yum install epel-release
...
# yum install python-pip } pipをインストールして最新化
# pip install --upgrade pip
...
# pip install ansible
...
# pip show ansible } pipでAnsibleをインストールしてインストール状況を確認
Name: ansible
Version: 2.9.9
...
# pip install requests } Requestsモジュールのインストール (aXAPIを用いるモジュールで必須)
...
# pip install paramiko
...
```



# A10用モジュールのCloneとインストール

```
# git clone https://github.com/a10networks/a10-acos-axapi.git } aXAPI用モジュールの
... } Clone
# cd a10-acos-axapi/
# git pull
Already up-to-date. } 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-galaxy collection build aXAPI用モジュールのcollectionのビルド
# ansible-galaxy collection install a10-acos_axapi*.tar.gz -p ~/.ansible/collections
... aXAPI用モジュールのcollectionのインストール
# cd
# ansible-galaxy collection install a10.acos_cli } CLI用モジュールの
... } インストール
# export
ANSIBLE_ACTION_PLUGINS=/root/.ansible/collections/ansible_collections/a10/acos_axapi/plugins/action
... aXAPI用モジュールの環境変数のexport
# export
ANSIBLE_CLICONF_PLUGINS=/root/.ansible/collections/ansible_collections/a10/acos_cli/plugins/cliconf
... CLI用モジュールの環境変数のexport
# export
ANSIBLE_TERMINAL_PLUGINS=/root/.ansible/collections/ansible_collections/a10/acos_cli/plugins/terminal
... CLI用モジュールの環境変数のexport
```

# 環境変数の設定

```
# mkdir /etc/ansible           Ansibleの環境設定用ディレクトリを作成（すでにあれば作成不要）
# vi /etc/ansible/ansible.cfg   Ansibleの環境設定用ファイルを作成（すでにあれば作成不要）
https://github.com/ansible/ansible/blob/devel/examples/ansible.cfg
にある内容を記述した後、以下を追記（すでにファイルがあれば以下の文字列を追記）
collections_paths = /root/.ansible/collections/ansible_collections
cliconf_plugins   = /root/.ansible/collections/ansible_collections/a10/acos_cli/plugins/cliconf
terminal_plugging = /root/.ansible/collections/ansible_collections/a10/acos_cli/plugins/terminal
action_plugins    = /root/.ansible/collections/ansible_collections/a10/acos_axapi/plugins/action

# cd
# vi ~/.bashrc   ログイン時の環境変数設定用ディレクトリを作成（すでにあれば作成不要）
以下を行末に追記
export
ANSIBLE_CLICONF_PLUGINS=/root/.ansible/collections/ansible_collections/a10/acos_cli/plugins/cliconf
export
ANSIBLE_TERMINAL_PLUGINS=/root/.ansible/collections/ansible_collections/a10/acos_cli/plugins/terminal
export
ANSIBLE_ACTION_PLUGINS=/root/.ansible/collections/ansible_collections/a10/acos_axapi/plugins/action
# ansible-doc a10.acos_cli.acos_command
...
# ansible-doc a10.acos_axapi.a10_slb_server
...
} 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できればOK)
```

# Playbookの作成とテスト

```
# mkdir playbook
# 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

[vThunder:vars]
ansible_username=admin
ansible_password=a10
ansible_port=443
# vi a10_network_vlan_create.yaml
  (次ページ以降のPlaybookを記述し保存)
# ansible-playbook -i hosts a10_network_vlan_create.yaml
...
```

} Playbook用ディレクトリの作成と移動

} インベントリファイルの作成

} Playbookの作成

} 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

```
---
- name: Create VLAN and Write Memory
  connection: local
  hosts: 10.255.211.110
  gather_facts: no
  collections:
    - a10.acos_axapi

  vars:
    ansible_host: "10.255.211.110"
  tasks:
    - name: Create VLAN
      a10_network_vlan:
        ansible_host: "{{ ansible_host }}"
        ansible_username: "{{ ansible_username }}"
        ansible_password: "{{ ansible_password }}"
        ansible_port: "{{ ansible_port }}"
        vlan_num: 10
        untagged_eth_list:
          - untagged_ethernet_start: 1
            untagged_ethernet_end: 1

    - name: Write Memory
      a10_write_memory:
        ansible_host: "{{ ansible_host }}"
        ansible_username: "{{ ansible_username }}"
        ansible_password: "{{ ansible_password }}"
        ansible_port: "{{ ansible_port }}"
        state: present
        partition: all

    - name: Write Memory
      ve: 10
      user_tag: "test"
      state: present
```

# 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

```
---
- name: Delete VLAN and Write Memory
  connection: local
  hosts: 10.255.211.110
  gather_facts: no
  collections:
    - a10.acos_axapi

  vars:
    ansible_host: "10.255.211.110"
  tasks:
    - name: Delete VLAN
      a10_network_vlan:
        ansible_host: "{{ ansible_host }}"
        ansible_username: "{{ ansible_username }}"
        ansible_password: "{{ ansible_password }}"
        ansible_port: "{{ ansible_port }}"
        vlan_num: 10
        untagged_eth_list:
          - untagged_ethernet_start: 1
            untagged_ethernet_end: 1

    - name: Write Memory
      a10_write_memory:
        ansible_host: "{{ ansible_host }}"
        ansible_username: "{{ ansible_username }}"
        ansible_password: "{{ ansible_password }}"
        ansible_port: "{{ ansible_port }}"
        state: present
        partition: all

    - name: Delete VLAN and Write Memory
      connection: local
      hosts: 10.255.211.110
      gather_facts: no
      collections:
        - a10.acos_axapi

      vars:
        ansible_host: "10.255.211.110"
      tasks:
        - name: Delete VLAN
          a10_network_vlan:
            ansible_host: "{{ ansible_host }}"
            ansible_username: "{{ ansible_username }}"
            ansible_password: "{{ ansible_password }}"
            ansible_port: "{{ ansible_port }}"
            vlan_num: 10
            untagged_eth_list:
              - untagged_ethernet_start: 1
                untagged_ethernet_end: 1

        - name: Write Memory
          a10_write_memory:
            ansible_host: "{{ ansible_host }}"
            ansible_username: "{{ ansible_username }}"
            ansible_password: "{{ ansible_password }}"
            ansible_port: "{{ ansible_port }}"
            state: absent
            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
- 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モジュールとParamiko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

**A10**

Always Secure. Always Available.