

企業・DC・SMB 市場向け Apresia 用
Ansible 5(Ansible-core 2.12)対応 Ver.1.03.01

APRESIA Systems 株式会社

制定・改訂履歴表

No.	年 月 日	内 容
-	2023 年 10 月 31 日	<ul style="list-style-type: none"> ・ TD61-8172 エンタープライズ Apresia シリーズ向け Ansible5(ansible-core 2.11)対応より作成 AEOS7 機である Apresia5400/3400 シリーズに関する記述と facts モジュールに関する記述を追加しました。 ・ 表 1.1 動作確認実施装置 に装置 FW バージョン及び AEOS7 機の情報を追加 ・ 表 2.1.4 Apresia5400/3400 シリーズ向けに提供するファイル構成 を追加 ・ 表 2.3.1 モジュール名称 に AEOS7 機及び、facts モジュールに関する情報を追加 ・ 表 2.3.4 ansible-doc apresia.aeosnp.aeosnp_facts 実行結果を追加 ・ 表 2.3.9 aeosnp_facts モジュールのオプション を追加 ・ 表 2.3.10 aeosnp_facts モジュールの戻り値 を追加 ・ 表 2.5.1 対象装置に対応するネットワーク OS 名称 を追加 ・ 1. はじめに 注意事項を追加 ・ 2.3.1.1 モジュールの概要 の注意事項を修正 ・ 2.3.1.3 記述サンプル に exclude_commands の例を追加 ・ 2.3.3 aeosnp_facts (aeosnp_facts.py) の項を追加 ・ 2.5 Playbook の実行について に注意事項を追加 ・ 2.5.3 ネットワーク OS の説明を修正 ・ 2.4.4.1 ApresiaNP シリーズ、ApresiaLightGM220 シリーズ向けの場合 に Fail 発生状況の記載を追加

目次

制定・改訂来歴表	1
目次	2
1. はじめに	4
1.1 ライセンスや商標について	4
2. 説明	5
2.1 提供するファイル構成について	5
2.2 インストール方法	11
2.2.1 実行環境構築 Ansible や Python のインストール	11
2.2.2 モジュールファイルのインストールと test 環境作成	13
2.3 提供するモジュールに関する情報	15
2.3.1 aeosnp_config (aeosnp_config.py)	20
2.3.1.1 モジュールの概要	20
2.3.1.2 オプション	21
2.3.1.3 記述サンプル	22
2.3.1.4 戻り値	25
2.3.2 aeosnp_command (aeosnp_command.py)	26
2.3.2.1 モジュールの概要	26
2.3.2.2 オプション	26
2.3.2.3 記述サンプル	26
2.3.2.4 戻り値	28
2.3.3 aeosnp_facts (aeosnp_facts.py)	29
2.3.3.1 モジュールの概要	29
2.3.3.2 オプション	29
2.3.3.3 記述サンプル	30
2.3.3.4 戻り値	31
2.5 Playbook の実行について	36
2.5.1 実行前の準備	36
2.5.2 コネクションプラグイン	36
2.5.3 ネットワーク OS	37
2.5.4 Playbook の TASK が Fail する条件について	37
2.5.4.1 ApresiaNP シリーズ、ApresiaLightGM200 シリーズ向けの場合	37
2.5.4.2 Apresia15000/13200、Apresia5400/3400 シリーズ向けの場合	39
2.5.5 装置の状態確認 Playbook について	41
2.5.5.1 Playbook の内容	41
2.5.5.2 実行結果 成功時	41
2.5.5.3 実行結果 失敗時 (2 例)	42
2.5.6 構成情報保存 Playbook について	43
2.5.6.1 Playbook の内容	43

2.5.6.2 実行結果 成功時.....	43
2.5.6.3 実行結果 失敗時.....	43
2.5.7 Playbook 実行時のトラブルシューティング	44
2.6 Apresia 各シリーズにおける SSH 接続について.....	45
2.6.1 ApresiaNP シリーズの場合	45
2.6.2 ApresiaLightGM200 シリーズの場合	46
2.6.3 Apresia15000/13200 シリーズ,Apresia5400/3400 シリーズの場合	46


1. はじめに


本資料は、企業・データセンター向け及び、スモールビジネス市場向け Apresia の Ansible 対応に関する説明書です。

動作確認実施した装置及び、Ansible と Python のバージョンを、以下の表に示します。

表 1.1 動作確認実施装置

シリーズ名	装置名称	装置 FW バージョン 注	Playbook を実行した環境における Ansible 及び Python バージョン
ApresiaNP シリーズ	ApresiaNP2100-24T4X-PoE	1.10.02	Ansible 5.10.0(Ansible-core 2.12.7) Python 3.8.10
	ApresiaNP5000-48T4X	1.08.02	
ApresiaLightGM200 シリーズ	ApresiaLightGM220GT-SS	2.00.01	
Apresia15000/13200 シリーズ	Apresia13200-28GT	8.40.01	
Apresia5400/3400 シリーズ	Apresia5428GT	7.45.01	

 この注意シンボルは、そこに記述されている事項が人身の安全と直接関係しない注意書きに関するものであることを示し、注目させる為に用います。

 装置 FW バージョン番号は、このバージョンを用いて評価実施したものとなります。Apresia15000/13200 シリーズにおいては、AEOS8.39.01 未満だとコマンドモジュールでの実行においてエラーが発生します。AEOS8.39.01 以上を対象としてください。

1.1 ライセンスや商標について

本 Ansible 対応で提供されるファイルのライセンスは GPL3: GNU General Public License Version 3 です。ライセンス表記については、プラグインやモジュールファイル冒頭のライセンス表記をご覧ください。

APRESIA は、APRESIA Systems 株式会社の登録商標です。

その他の社名、ブランド名および商品名は、各所有者の商標もしくは登録商標です。

2. 説明

2.1 提供するファイル構成について

以下の表 2.1.1 に ApresiaNP シリーズ向けに提供するファイル構成について示します。ファイル名の末尾が tar.gz となるファイルを開いた場合のフォルダ構成、ファイル名の一覧となります。ApresiaLightGM200 シリーズや、Apresia15000/13200 シリーズ、Apresia5400/3400 シリーズ向けのファイル構成をそれぞれ、表 2.1.2, 表 2.1.3, 表 2.1.4 に示します。

表 2.1.1 ApresiaNP シリーズ向けに提供するファイル構成

apresia-aeosnp-1.3.1.tar.gz ファイルに含まれるフォルダおよびファイル名	説明
<pre> . -- FILES.json -- MANIFEST.json -- meta `-- runtime.yml `-- plugins -- action -- __init__.py `-- aeosnp.py -- cliconf -- __init__.py `-- aeosnp.py -- module_utils -- __init__.py `-- network -- __init__.py `-- aeosnp -- __init__.py -- aeosnp.py -- argspec -- __init__.py `-- facts -- __init__.py `-- facts.py -- facts -- __init__.py -- facts.py -- hostname -- __init__.py `-- hostname.py -- interfaces -- __init__.py `-- interfaces.py -- l2_interfaces -- __init__.py `-- l2_interfaces.py -- legacy -- __init__.py `-- base.py </pre>	

<pre> -- logging_global -- __init__.py `-- logging_global.py -- snmp_server -- __init__.py `-- snmp_server.py -- static_routes -- __init__.py `-- static_routes.py `-- vlans -- __init__.py `-- vlans.py -- rm_templates -- hostname.py -- interfaces.py -- l2_interfaces.py -- logging_global.py -- snmp_server.py -- static_routes.py `-- vlans.py -- modules -- __init__.py -- aeosnp_command.py -- aeosnp_config.py `-- aeosnp_facts.py `-- terminal -- __init__.py `-- aeosnp.py </pre> <p>21 directories, 45 files.</p>	
---	--

表 2.1.2 ApresiaLightGM200 シリーズ向けに提供するファイル構成

apresia-apl-1.3.1.tar.gz ファイルに含まれるフォルダおよびファイル名	説明
<pre> . -- FILES.json -- MANIFEST.json -- meta `-- runtime.yml `-- plugins -- action -- __init__.py `-- apl.py -- cliconf -- __init__.py `-- apl.py -- module_utils -- __init__.py `-- network -- __init__.py `-- apl </pre>	

```

|-- __init__.py
|-- apl.py
|-- argspec
|   |-- __init__.py
|   `-- facts
|       |-- __init__.py
|       `-- facts.py
|-- facts
|   |-- __init__.py
|   |-- facts.py
|   |-- hostname
|   |   |-- __init__.py
|   |   `-- hostname.py
|   |-- interfaces
|   |   |-- __init__.py
|   |   `-- interfaces.py
|   |-- l2_interfaces
|   |   |-- __init__.py
|   |   `-- l2_interfaces.py
|   |-- legacy
|   |   |-- __init__.py
|   |   `-- base.py
|   |-- logging_global
|   |   |-- __init__.py
|   |   `-- logging_global.py
|   |-- snmp_server
|   |   |-- __init__.py
|   |   `-- snmp_server.py
|   |-- static_routes
|   |   |-- __init__.py
|   |   `-- static_routes.py
|   `-- vlans
|       |-- __init__.py
|       `-- vlans.py
`-- rm_templates
    |-- hostname.py
    |-- interfaces.py
    |-- l2_interfaces.py
    |-- logging_global.py
    |-- snmp_server.py
    |-- static_routes.py
    `-- vlans.py

|-- modules
|   |-- __init__.py
|   |-- apl_command.py
|   |-- apl_config.py
|   `-- apl_facts.py
`-- terminal
    |-- __init__.py
    `-- apl.py

```

21 directories, 45 files

表 2.1.3 Apresia15000/13200 シリーズ向けに提供するファイル構成

apresia-aeos8-1.3.1.tar.gz ファイルに含まれるフォルダおよびファイル名	説明
<pre> . -- FILES.json -- MANIFEST.json -- meta `-- runtime.yml `-- plugins -- action -- __init__.py `-- aeos8.py -- cliconf -- __init__.py `-- aeos8.py -- module_utils -- __init__.py `-- network -- __init__.py `-- aeos8 -- __init__.py -- aeos8.py -- argspec `-- facts `-- facts.py -- facts -- facts.py -- hostname -- __init__.py `-- hostname.py -- interfaces -- __init__.py `-- interfaces.py -- l2_interfaces -- __init__.py `-- l2_interfaces.py -- legacy -- __init__.py `-- base.py -- logging_global -- __init__.py `-- logging_global.py -- snmp_server -- __init__.py `-- snmp_server.py -- static_routes -- __init__.py `-- static_routes.py `-- vlans -- __init__.py `-- vlans.py `-- rm_templates -- hostname.py </pre>	

<pre> -- interfaces.py -- l2_interfaces.py -- logging_global.py -- snmp_server.py -- static_routes.py `-- vlans.py -- modules -- __init__.py -- aeos8_command.py -- aeos8_config.py `-- aeos8_facts.py `-- terminal -- __init__.py `-- aeos8.py </pre> <p>22 directories, 41 files</p>	
--	--

表 2.1.4 Apresia5400/3400 シリーズ向けに提供するファイル構成

apresia-aeos7-1.3.1.tar.gz ファイルに含まれるフォルダおよびファイル名	説明
<pre> . -- FILES.json -- MANIFEST.json -- meta `-- runtime.yml `-- plugins -- action -- __init__.py `-- aeos7.py -- cliconf -- __init__.py `-- aeos7.py -- module_utils -- __init__.py `-- network -- __init__.py `-- aeos7 -- __init__.py -- aeos7.py -- argspec `-- facts `-- facts.py -- facts -- facts.py -- hostname -- __init__.py `-- hostname.py -- interfaces -- __init__.py `-- interfaces.py -- l2_interfaces -- __init__.py `-- l2_interfaces.py </pre>	

```

|-- legacy
|   |-- __init__.py
|   |-- base.py
|-- logging_global
|   |-- __init__.py
|   |-- logging_global.py
|-- snmp_server
|   |-- __init__.py
|   |-- snmp_server.py
|-- static_routes
|   |-- __init__.py
|   |-- static_routes.py
|-- vlans
|   |-- __init__.py
|   |-- vlans.py
|-- rm_templates
|   |-- hostname.py
|   |-- interfaces.py
|   |-- l2_interfaces.py
|   |-- logging_global.py
|   |-- snmp_server.py
|   |-- static_routes.py
|   |-- vlans.py
|-- modules
|   |-- __init__.py
|   |-- aeos7_command.py
|   |-- aeos7_config.py
|   |-- aeos7_facts.py
|-- terminal
|   |-- __init__.py
|   |-- aeos7.py

```

21 directories, 42 files

2.2 インストール方法

2.2.1 実行環境構築 Ansible や Python のインストール

ここでは、Linux のディストリビューションのひとつである Ubuntu 20.04 LTS が動作している環境において、Ansible と Python をインストールして実行環境を構築する例を示します。

まず、構築結果としての実行環境一覧を示し、フォルダ構成を示した後、インストール手順を説明します。

[実行環境一覧] (Ansible や Python のバージョンは、構築時および評価時のものとなります)

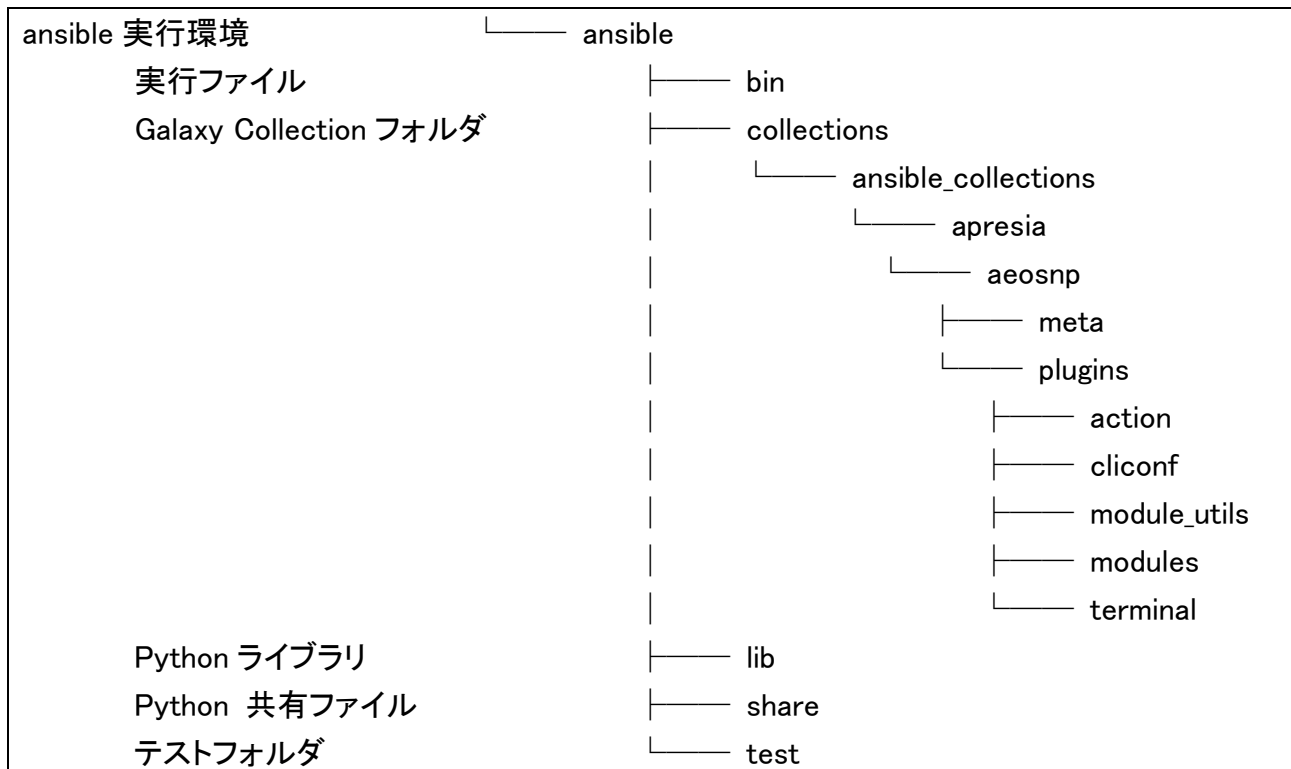
・Ansible 5(Ansible-core 2.12) + python 3.8

Ansible バージョン	2.12.7
Python バージョン	3.8.10
virtualenv 環境名	ansible(任意、以降パス中の ansible は virtualenv 環境名)
Ansible 実行バイナリパス	~/ansible/bin
python 実行バイナリパス	~/ansible/bin (python ->/usr/bin/python3)
ライブラリパス	~/ansible/lib/python3.8/site-packages
実行環境	source ~/ansible/bin/activate コマンドを実行し有効化

[フォルダ構成]

環境構築後のユーザーのホームディレクトリ構成を以下に示します。

表 2.2.1 ansible 実行環境のディレクトリ構成



(1) Python3 のインストール

```
$ sudo apt install python3
```

(2) virtualenv のインストール

```
$ sudo apt install virtualenv
```

(3) Ansible インストール先のフォルダを作成

```
$ mkdir ansible
```

(4) Python3 のバージョン確認

```
$ python3 -V
```

```
Python 3.8.xx
```

(5) virtualenv の実行と activate

(4)の結果が、Python 3.8.xx だった場合、/usr/bin/python3.8 が実行ファイル名なので、virtualenv 実行時に、python3.8 を指定し、フォルダ名は(3)で作成した ansible を指定する。

```
$ virtualenv --python=python3.8 ansible
```

```
$ cd ansible
```

```
$ source bin/activate
```

(6) Ansible のインストールと Ansible バージョン確認

(Ansible-core バージョンが 2.12 以降であることを御確認ください。)

```
$ pip install ansible==5.1
```

```
$ ansible --version
```

```
ansible [core 2.12.xx]
```

```
:
```

(7) virtualenv 終了

```
$ deactivate
```

```
$ cd ../
```

2.2.2 モジュールファイルのインストールと test 環境作成

(1) ansible-galaxy コマンドを用いたインストール

ここでは、ApreziaNP シリーズ向けの `apresia-aeosnp-1.3.1.tar.gz` (以降アーカイブファイルと呼びます)を用いて、`ansible-galaxy` コマンドにてインストールする方法を説明します。他の装置向けでも、アーカイブファイル名が異なるだけでインストール方法は同じなので説明は省略します。

まず、3.2.1 実行環境構築 Ansible や Python のインストールで作成された、`~/ansible` フォルダにカレントフォルダを変更し、アーカイブファイルを置きます。

その後、`ansible-galaxy` コマンドを実行します。

```
$ cd ~/ansible
$ cp /xxx/xxx/apresia-aeosnp-1.3.1.tar.gz ./
$ ansible-galaxy collection install apresia-aeosnp-1.3.1.tar.gz -p ./collections
```

(2) test 環境作成

test フォルダを作成し、そこにおいたファイル「`ansible.cfg`」にコレクションパスとして、Galaxy Collection が存在するフォルダへの相対パスを設定します。

```
$ mkdir test
$ cd test
```

test フォルダに置いた `ansible.cfg` の例

```
[defaults]
collections_paths = ../collections
```

これにより `collections_path` に指定したディレクトリ直下の `ansible_collections` フォルダ内に存在しているモジュールを使用可能になります。

なお `aeosnp` モジュールは下記のフォルダにあります。

```
~/ansible/collections/ansible_collections/apresia/aeosnp/
```

(3) ホスト設定のインベントリファイルについて

test フォルダに置いたインベントリファイル(`aeosnp_test_hosts`)に関して、以下の赤字の部分を変更に応じて、変更してください。

インベントリファイルの例

```
all:
```

```
vars:
```

```
ansible_connection: network_cli
```

```
ansible_network_os: apresia.aeosnp.aeosnp
```

```
ansible_python_interpreter: /usr/bin/python3
```

python 実行ファイルのパス

```
hosts:
```

```
10.249.183.96:
```

```
ansible_user: adpro
```

```
ansible_ssh_pass: adpro
```

対象装置の IP アドレス

特権ユーザー名と、SSH での
パスワード

2.3 提供するモジュールに関する情報

本章では、ApresiaNP シリーズ向けの `aeosnp_config`、`aeosnp_command`、`aeosnp_facts` モジュールについて説明します。他シリーズ向け記述は一部を除いて省略しています。他シリーズ装置で該当モジュールを使う場合には、モジュール名称を読み替える必要があります。表 2.3.1 をご覧ください。各モジュールは、`ansible-doc` コマンドを用いて、Playbook サンプルを含むドキュメントが表示可能です。`aeosnp_config`、`aeosnp_command`、`aeosnp_facts` モジュールの `ansible-doc` コマンド実行結果を、表 2.3.2、表 2.3.3、表 2.3.4 に示します。

また、モジュール全般に関係する情報は、2.4 Playbook の実行について に記載したので、合わせてご覧ください。

表 2.3.1 モジュール名称

装置名称	config モジュール	command モジュール
ApresiaNP シリーズ	<code>apresia.aeosnp.aeosnp_config</code>	<code>apresia.aeosnp.aeosnp_command</code>
ApresiaLightGM200 シリーズ	<code>apresia.apl.apl_config</code>	<code>apresia.apl.apl_command</code>
Apresia15000/13200 シリーズ	<code>apresia.aeos8.aeos8_config</code>	<code>apresia.aeos8.aeos8_command</code>
Apresia5400/3400 シリーズ	<code>apresia.aeos7.aeos7_config</code>	<code>apresia.aeos7.aeos7_command</code>
装置名称	facts モジュール	
ApresiaNP シリーズ	<code>apresia.aeosnp.aeosnp_facts</code>	
ApresiaLightGM200 シリーズ	<code>apresia.apl.apl_facts</code>	
Apresia15000/13200 シリーズ	<code>apresia.aeos8.aeos8_facts</code>	
Apresia5400/3400 シリーズ	<code>apresia.aeos7.aeos7_facts</code>	



本モジュールのパラメータにおいて、「:」が含まれる場合、「'」を用いて囲んでください。

記載例

```
- name: example show config with colon
  apresia.aeosnp.aeosnp_command:
    commands:
      - 'show config differences flash: primary.cfg flash: secondary.cfg'
- name: example logout clock command with colon
  apresia.aeosnp.aeosnp_config:
    lines:
      - 'logout clock 11:11 dot1x'
    parents:
      - access-defender
- name: example Set IPv6 address
  apresia.aeosnp.aeosnp_config:
    lines:
      - 'ipv6 address fe80::1:1 link-local'
    parents:
      - interface vlan 1
```

表 2.3.2 ansible-doc apresia.aeosnp.aeosnp_config 実行結果

```
> APRESIA.AEOSNP.AEOSNP_CONFIG
(/home/ansible/work/ansible/collections/ansible_collections/apresia/aeosnp/plugins/modules

    Apresia AEOSNP configurations use a simple block indent file syntax for
segmenting
    configuration into sections. This module provides an implementation for
working with
    AEOS NP configuration sections in a deterministic way.

ADDED IN: version 1.3.1 of apresia.aeosnp

* note: This module has a corresponding action plugin.

OPTIONS (= is mandatory):

- lines
    The ordered set of commands that should be configured in the section.
The commands must
    be the exact same commands as found in the device running-config to
ensure idempotency
    and correct diff. Be sure to note the configuration command syntax as
some commands are
    automatically modified by the device config parser.
    (Aliases: commands) [Default: (null)]
    elements: str
    type: list

- parents
    The ordered set of parents that uniquely identify the section or
hierarchy the commands
    should be checked against. If the parents argument is omitted, the
commands are checked
    against the set of top level or global commands.
    [Default: (null)]
    elements: str
... (以降は省略)
```

表 2.3.3 ansible-doc apresia.aeosnp.aeosnp_command 実行結果

```
> APRESIA.AEOSNP.AEOSNP_COMMAND
(/home/ansible/work/ansible/collections/ansible_collections/apresia/aeosnp/plugins/module

    Sends arbitrary commands to an aeosnp node and returns the results read
from the device.

    This module does not support running commands in configuration mode.
Please use
    [aeosnp_config] to configure AEOS NP devices.

ADDED IN: version 1.3.1 of apresia.aeosnp

* note: This module has a corresponding action plugin.

OPTIONS (= is mandatory):

= commands
    List of commands to send to the remote aeosnp device over the configured
provider. The
    resulting output from the command is returned. If a command sent to the
device requires
    answering a prompt, it is possible to pass a dict containing `command`,
`answer` and
    `prompt`. Common answers are 'y' or "¥r" (carriage return, must be
double quotes). See
    examples.

    elements: raw
    type: list
... (以降は省略)
```

表 2.3.4 ansible-doc apresia.aeosnp.aeosnp_facts 実行結果

```
> APRESIA.AEOSNP.AEOSNP_FACTS
(/home/ansible/work/ansible/collections/ansible_collections/apresia/aeosnp/plugins/module
Collects a base set of device facts from a remote device that is running
aeosnp. This
module prepends all of the base network fact keys with
`ansible_net_<fact>'. The facts
module will always collect a base set of facts from the device and can
enable or disable
collection of additional facts.

OPTIONS (= is mandatory):

- available_network_resources
    When 'True' a list of network resources for which resource modules are
available will be
provided.
[Default: False]
type: bool

- gather_network_resources
    When supplied, this argument will restrict the facts collected to a
given subset.
Possible values for this argument include all and the resources like
interfaces, vlans
etc. Can specify a list of values to include a larger subset. Values can
also be used
with an initial `!' to specify that a specific subset should not be
collected. Valid
subsets are 'hostname', 'interfaces', 'logging_global', 'static_routes',
'all',
'l2_interfaces', 'snmp_server', 'vlans'.
elements: str
type: list

- gather_subset
... (以降は省略)
```

2.3.1 aeosnp_config (aeosnp_config.py)

2.3.1.1 モジュールの概要

ApresiaNP シリーズの装置の構成(config)を設定します。

! 本モジュールは実行時に必ず内部で show running-config コマンドを実行します。このコマンドの出力結果と本モジュールで実行するように指定されたコマンド文字列を比較し、指定されたコマンドが show running-config の結果に含まれていない場合のみ装置で実際にコマンドが設定されます。そのため、コマンドの省略形を用いた場合や引数を省略した場合、show running-config と不一致となり、一度設定したコマンドでも再度設定が実施されます。

! 本モジュールの src パラメータを使用してファイルからコンフィグを読み込む場合、同一のコマンドモードの指定が複数回存在するときに、2 個目以降のコマンドモードは設定済と判断され除外された結果、そのコマンド配下の設定が意図しないモードで実行されることがあります。このような場合、設定ファイルおよびタクスを分けて実行してください。以下の例では、interface port1/0/1 が 2 個ありますが、各ファイルで同じコマンドモードが複数存在しないようにファイルを分けてください。

```
interface port 1/0/1
  description p1

interface port 1/0/2
  description p2

# ここでファイルを分割してください

interface port 1/0/1
  switchport mode trunk
```

なお、コマンドモード配下に続く行では、show running-config で表示される通りのインデントを入れる必要があります。

正しい記載例

```
interface port 1/0/1
  description test
```

誤った記載例

モード遷移後のインデントが多く show running-config の表示と異なる

```
interface port 1/0/1
    description test
```



ApresiaNP シリーズや ApresiaLightGM200 シリーズにおいて装置の構成情報の prompt 設定に関する注意事項があります。

- (1) prompt に空白文字を含まない文字列を事前に設定してください。装置デフォルトの場合、コマンド出力の応答において、“#”や“>”が含まれる行が出力されません。なお(2)もご覧ください。

設定例 prompt に、“NP2100”を設定

```
prompt NP2100
```

- (2) prompt <STRING>設定を実施した場合、コマンド出力の応答に“<STRING>#”や、“<STRING>>”が含まれた行は出力されません。

例えば、prompt NP2100 が設定されていて、port の description に、“NP2100#”、あるいは“NP2100>”が設定されている場合、show running-config の出力において、description 設定行が出力されません。

2.3.1.2 オプション

本モジュールのオプションを表 2.3.5 に示します。

表 2.3.5 aeosnp_config モジュールのオプション

名称	必須/任意の別	デフォルト値	説明
lines	任意	null	<ul style="list-style-type: none"> ・セクションで設定すべきコマンドの順序セットです。 ・コマンドは、デバイスの running-config にあるものと全く同じコマンドでなければなりません。 ・いくつかのコマンドは、デバイスの設定パーサーによって自動的に変更されるため、設定コマンドの構文に注意する必要があります。
parents	任意	null	<ul style="list-style-type: none"> ・コマンドをチェックするセクションまたは階層を一意に特定する親となる順序付きセットです。 ・parents 引数が省略された場合、コマンドはトップレベルまたはグローバルコマンドのセットとしてチェックされます。
src	任意	null	<ul style="list-style-type: none"> ・ロードする設定または設定テンプレートを含むファイルへのソースパスを指定します。

			<ul style="list-style-type: none"> ・ソースファイルのパスは、Ansible コントロールホスト上のフルパスか、playbook または role のルートディレクトリからの相対パスのいずれかになります。 ・この引数と lines と parents とは互いに排他的です。 ・src で指定したファイル内の設定行は、running-config 表示と同一である必要があります。
exclude_commands	任意	null	<ul style="list-style-type: none"> ・除外したいコマンド文字列を指定します。 ・ダブルクォーテーションで囲み、正規表現が利用可能です。

2.3.1.3 記述サンプル

以下に ApresiaNP シリーズ向けの記述サンプルを示します。最後の exclude_commands の例のみ他機種の記述サンプルがあります。

装置のプロンプトにインベントリーでのホスト名を設定

```
- name: configure top level configuration
  apresia.aeosnp.aeosnp_config:
    lines: prompt {{ inventory_hostname }}
```

ポート 1/0/1 に、デスクリプションと IP アドレスを設定(モード遷移する例)

```
- name: configure interface settings
  apresia.aeosnp.aeosnp_config:
    lines:
      - description testinterface
      - ip address 172.31.1.1 255.255.255.0
    parents: interface port 1/0/1
```

lldp tx-delay を設定

```
- name: configure lldp tx-delay
  apresia.aeosnp.aeosnp_config:
    lines:
      - lldp tx-delay 1
```

コマンドモード CFM MA 設定モードにて、MA の MEP ID リスト設定を追加

```
- name: configure MEP ID
  apresia.aeosnp.aeosnp_config:
```

```
lines:
  - mepid-list add 1
parents:
  - cfm domain op-domain level 2
  - cfm ma name op1 vlan 1
```

interface range 指定にて、1/0/20 と 1/0/22 ポートを trunk に設定

```
- name: 'interface port config'
  apresia.aeosnp.aeosnp_config:
    lines:
      - switchport mode trunk
    parents:
      - 'interface range port 1/0/20,1/0/22'
```

sample.cfg を読み込んで設定する例

```
- name: configure read src
  apresia.aeosnp.aeosnp_config:
    src: ./sample.cfg
```

np_include_ssh.cfg を読み込んで設定するが、指定したコマンド設定は除外する例(ApresiaNP シリーズ用)

```
- name: aosnp_config exclude ssh configuration playbook
  hosts: all
  tasks:
    - name: configure read src
      apresia.aeosnp.aeosnp_config:
        src: np_include_ssh.cfg
        exclude_commands:
          - "%ninterface mgmt 0%n( .+%n)+"
          - "%nline ssh%n( .+%n)+"
          - "%nip ssh server%n"
          - "%nip ssh service-port.+%n"
          - "%n(ssh user .+ authentication-method.+%n)+"
          - "%n(username .+%n)+"
```

apl_include_ssh.cfg を読み込んで設定するが、指定したコマンド設定は除外する例(ApresiaLightGM200 シリーズ用)

```
- name: apl_config exclude ssh configuration playbook
  hosts: all
```



```

tasks:
- name: configure read src
  apresia.apl.apl_config:
    src: apl_include_ssh.cfg
    exclude_commands:
      - "%n ip address .+"
      - "%nip route default .+"
      - "%nline ssh%n( .+%n)+"
      - "%nip ssh server%n"
      - "%nip ssh service-port. +%n"
      - "%nssh user .+ authentication-method. +%n"
      - "%n(username .+%n)+"

```

aeos_include_ssh.cfg を読み込んで設定するが、指定したコマンド設定は除外する例
(Apresia15000/13200 シリーズ用)

```

- name: aeos8_config exclude ssh configuration playbook
hosts: all
gather_facts: false
tasks:
- name: configure read src
  apresia.aeos8.aeos8_config:
    src: aeos_include_ssh.cfg
    exclude_commands:
      - "%ninterface manage%n( .+%n)+"
      - "%nip route .+%n"
      - "%nipv6 route .+%n"
      - "%nsshd server%n"
      - "%nsshd ipv6 server%n"
      - "%n(username .+%n)+"

```

aeos_include_ssh.cfg を読み込んで設定するが、指定したコマンド設定は除外する例
(Apresia5400/3400 シリーズ用)

```

- name: aeos7_config exclude ssh configuration playbook
hosts: all
gather_facts: false
tasks:

- name: configure read src
  apresia.aeos7.aeos7_config:

```

```

src: aeos_include_ssh.cfg
exclude_commands:
- "%ninterface manage%n( .+%n)+"
- "%nip route .+%n"
- "%nsshd server%n"
- "%n(username .+%n)+"

```

2.3.1.4 戻り値

本モジュールの戻り値を表 2.3.6 に示します。

表 2.3.6 aeosnp_config モジュールの戻り値

名称	定義	常時/任意の別	型	サンプル
commands	設定したコマンドのセット	常時	list	prompt foo, router ospf 1, router-id 192.0.2.1
updates	commands と同じ	常時	list	prompt foo, router ospf 1, router-id 192.0.2.1

2.3.2 aeosnp_command (aeosnp_command.py)

2.3.2.1 モジュールの概要

AEOS NP デバイスの設定に用います。

aeosnp ノードに任意のコマンドを送信し、デバイスから読み取った結果を返します。

このモジュールは、コンフィギュレーションモードでのコマンドの実行をサポートしていません。

2.3.2.2 オプション

表 2.3.7 aeosnp_command モジュールのオプション

名称	必須/任意の別	デフォルト値	説明
commands	必須		<ul style="list-style-type: none">・設定されたプロバイダを介して、リモートの AEOS NP 装置に送信するコマンドのリストです。コマンドの結果出力が返されます。・装置に送るコマンドがプロンプトへの応答が必要な場合、'command' と 'answer' と 'prompt' を含む dict を渡すことが可能です。一般的な回答は 'y' です。ダブルクォーテーションで囲み、"y" や "¥r" (Enter) も使用可能です。



prompt/answer の記述において、正規表現で用いる特殊記号「(」 「)」 「[」 「]」などを指定したい場合、「¥」を付けてエスケープする必要があります。

```
- name: clear logging
  apresia.aeosnp.aeosnp_command:
    commands:
      - command: clear logging
        prompt:
          - Clear logging¥? ¥(y/n¥) ¥[n¥]
        answer:
          - y
```

2.3.2.3 記述サンプル

以下に ApresiaNP シリーズ向けの記述サンプルを示します。

show version を実行します。

```
- name: show version
  apresia.aeosnp.aeosnp_command:
    commands:
      - show version
```

show version と show interfaces を実行します。

```
- name: run multiple commands on remote nodes
  apresia.aeosnp.aeosnp_command:
    commands:
      - show version
      - show interfaces
```

show running-config を実行します。

```
- name: show running-config
  apresia.aeosnp.aeosnp_command:
    commands:
      - show running-config
```

write memory コマンドを実行します。

```
- name: write memory
  apresia.aeosnp.aeosnp_command:
    commands:
      - command: write memory
        prompt:
          - 'Destination filename startup-config?'
        answer:
          - "y"
```

copy running-config tftp を実行します。

2 回以上の応答が必要な場合には、aeosnp_command モジュールではなく、cli_command モジュールを使用します。

```
- name: copy running-config
  cli_command:
    command: 'copy running-config tftp:'
    check_all: True
    prompt:
      - Address of remote host ¥[¥]¥?
      - Destination filename ¥[¥]¥?
    answer:
      - 10.249.23.225
      - running_config.txt
```

2.3.2.4 戻り値

本モジュールの戻り値を表 2.3.8 に示します。

表 2.3.8 aeosnp_command モジュールの戻り値

名称	定義	常時/任意の別	型	サンプル
failed_conditions	Failしたときの状態を list で返します	Failしたとき	list	..., ...
stdout	コマンドからの応答のセット	常時 action plugin などからの低レベルのエラーを含みます。	list	..., ...
stdout_lines	stdout を list で分割したもの	常時 action plugin などからの低レベルのエラーを含みます。	list	-- '...' - '...' -- '...' -- '...'
failed_conditions	Failしたときの状態を list で返します	Failしたとき	list	..., ...

2.3.3 aeosnp_facts (aeosnp_facts.py)

2.3.3.1 モジュールの概要

AEOS NP デバイスからファクトと呼ぶデバイス情報の基本セットを収集します。このモジュールは、すべてのベースネットワークファクトキーの先頭に `ansible_net_<fact>` を付加します。ファクトモジュールではデバイスからの基本セットの収集の他に、追加のファクト情報の収集を有効または無効に設定できます。

2.3.3.2 オプション

表 2.3.9 aeosnp_facts モジュールのオプション

名称	必須/任意の別	デフォルト値と(型)	説明
<code>available_network_resources</code>	任意	False (bool)	・True の場合、どのリソースのネットワークリソースの list として、利用可能なモジュールが提供されます。
<code>gather_network_resources</code>	任意	Null (str list)	この引数を指定すると、指定されたサブセットに収集されるファクトが制限されます。この引数の可能な値には、すべてと、インターフェース、VLAN などのリソースが含まれます。 この引数を指定すると、指定されたサブセットに収集されるファクトが制限されます。この引数の可能な値には、all と、interface、VLAN などのリソースが含まれます。
<code>gather_subset</code>	任意	!config (str list)	この引数を指定すると、収集されるファクトが指定されたサブセットに制限されます。この引数の可能な値には、「all」、「min」、「hardware」、「config」、および「interfaces」が含まれます。より大きなサブセットを含める値の list を指定できます。最初に「!」を付けた値を使用すると、そのサブセットを除くすべてのファクトを収集します。このモジュールは <code>network_cli</code> に接続して動作します。

2.3.3.3 記述サンプル

以下に記述サンプルを示します。

すべてのレガシーファクトを収集します。

```
- name: Gather all legacy facts
  apresia.aeosnp.aeosnp_facts:
    gather_subset: all
```

構成とデフォルトのファクトのみを収集します。

```
- name: Gather config and default facts
  apresia.aeosnp.aeosnp_facts:
    gather_subset:
      - config
```

ハードウェアのファクトを収集しない例

```
- name: Gather all legacy facts
  apresia.aeosnp.aeosnp_facts:
    gather_subset:
      - '!hardware'
```

レガシーおよびリソースのファクトを収集します。

```
- name: Gather legacy and resources facts
  apresia.aeosnp.aeosnp_facts:
    gather_subset: all
    gather_network_resources: all
```

インターフェースリソースファクトを収集し、レガシーファクトは収集しない例

```
- name: Gather all legacy facts
  apresia.aeosnp.aeosnp_facts:
    gather_subset:
      - '!all'
      - '!min'
    gather_network_resources:
      - interfaces
```

インターフェースリソースと最小限のレガシーファクトを収集する例

```
- name: Gather interface and min legacy facts
  apresia.aeosnp.aeosnp_facts:
    gather_subset: min
```

```
gather_network_resources: interfaces
```

L2 インターフェースリソースファクトと最小限のレガシーファクトを収集する例

```
- name: Gather all legacy facts
  apresia.aeosnp.aeosnp_facts:
    gather_subset: min
    gather_network_resources: l2_interfaces
```

注: “L2_intefaces”ではなく、“l2_interfaces”であり、小文字の L で始まります。

2.3.3.4 戻り値

aeosnp_facts、apl_facts、aeos8_facts、aeos7_facts モジュールの戻り値を表 2.3.10 に示します。各装置ごとの記載は、実装の文頭[NP][APL][AEOS|AEOS7|AEOS8]にて示します。[AEOS]の場合には、AEOS7,8 で共通の実装となります。また記述が無い場合には実装がありません。

表 2.3.10 aeosnp_facts モジュールの戻り値

名称	定義	実装	型	取得可能条件
ansible_net_all_ipv4_addresses	デバイスに設定されているすべてのIPv4アドレス	[NP][APL]show ip interface の IP address 値 [AEOS]show ip interface brief の IP-Address の値	list	gather_subset: interfaces が設定され、デバイスにアドレスが存在する場合
ansible_net_all_ipv6_addresses	デバイスに設定されているすべてのIPv6アドレス	[NP][APL]show ipv6 interface の Link-local address と Global unicast address の値 [AEOS8]show ipv6 interface brief の IP-Address の値	list	gather_subset: interfaces が設定され、デバイスにアドレスが存在する場合

ansible_net_config	デバイスの現在のアクティブな構成	[NP][APL][AEOS]show running-config の結果	str	gather_subset: config 設定した場合
ansible_net_filesystems	ファイルシステムの一覧を示します	[NP][APL]dir の"Directory of /c:"の値	list	gather_subset: hardware を設定した場合
ansible_net_filesystems_info	空き容量や合計容量など各ファイルシステムに関する情報を含むすべてのファイルシステムのハッシュ	spacetotal_kb の値	[NP][APL]dir コマンドの Directory の bytes total の値	dict gather_subset: hardware を設定した場合
		spacefree_kb の値	[NP][APL]dir コマンドの Directory の bytes free の値	

ansible_net_interfaces	システム上で実行されているすべてのインターフェースのハッシュ	lineprotocol の値	[NP][APL]show interfaces の link status 値 [AEOS]show interface status detail の Link status の 値	dict	gather_subset: interfaces 設 定時
		mediatype	[NP][APL]show interfaces の Interface type 値 [AEOS]show interface status detail の Media の値		
		description	[NP][APL]show interfaces の Interface description 値 [AEOS]show interfaces status detail の Description 値		
		macaddress	[NP][APL]show interfaces の MAC Address 値		
		mtu	[NP][APL]show interfaces の		

			Maximum transmit unit の値		
		duplex	[NP][APL]show interfaces の各ポートごと 8 行目の値。リンクダウンしている場合は Null [AEOS]show interface status detail の Fix speed/duplex の値		
ansible_net_memfree_mb	リモートデバイス上の利用可能な空きメモリ (Mb)	[NP][APL]show unit の DRAM Free の値 [AEOS]show vmstat の fre の値	int	gather_subset: hardware 設定時	
ansible_net_memtotal_mb	リモートデバイス上のメモリの合計 (Mb)	[NP][APL]show unit の DRAM Total の値 [AEOS]show vmstat の avm + fre の値	int	gather_subset: hardware 設定時	
ansible_net_model	デバイスから返さ	[NP][APL]show version の Module Name の値 [AEOS]show system の Hardware Model の値	str	default 及び gather_subset: min 設定時	

	れた モデ ル名			
ansible_net_serialnum	リ モート デバ イス のシ リア ル番 号	[NP][APL]show unit コマンドの Serial-Number の値 [AEOS]show system の Serial Number の値	str	default 及び gather_subset: min 設定時
ansible_net_version	リ モート デバ イス で実 行さ れて いる オペ レー ティ ング ステ ムの バー ジョン	[NP][APL]show version の Versions Runtime の値 [AEOS]show version の System Revision の値	str	default 及び gather_subset: min 設定時

2.5 Playbook の実行について

- !** Playbook を実行した際、TASK が fatal となり「command timeout triggered, timeout value is 10 secs.」などと表示される場合、Ansible 提供の network_cli における「persistent_command_timeout」設定値を変更してください。
ansible.cfg で設定する方法を以下に示します。この例では 300 秒を指定していません。デフォルト値は 30 秒です (Ansible 2.11.01 の場合)。設定値の値域は環境に依存し、例えば Ubuntu 18.04.1 LTS では、1 から $2^{31}-1 = 2147483647$ [秒] となります。

```
[persistent_connection]
command_timeout = 300
```

- !** Playbook を実行した際、「operation requires privilege escalation」と表示され権限昇格が必要というエラーになる場合、インベントリ変数 ansible_become を yes に設定すると特権モードに移行でき解消できることがあります。インベントリファイルにて設定する場合は以下となります。

```
ansible_become: yes
```

2.5.1 実行前の準備

Playbook を用いてモジュールを実行する端末から Apresia の各装置に対して SSH でログイン可能なことを確認してください。

Apresia の各装置が `~/.ssh/known_host` に登録されていない場合、その装置に接続できません。

なお、Apresia 各装置への SSH 接続設定については、2.5 Apresia 各シリーズにおける SSH 接続について をご覧ください。

2.5.2 コネクションプラグイン

コネクションプラグインには network_cli を使用してください。

具体的には下記どちらかの変数に文字列「network_cli」を設定してください。

- ・インベントリファイルで指定する場合

ansible_connection 変数

- ・プレイブックで指定する場合

connection 変数

2.5.3 ネットワーク OS

インベントリファイルもしくはプレイブック上で変数 `ansible_network_os` には、表 2.5.1 に示す対象装置に対応したネットワーク OS 名称の文字列を設定してください。

表 2.5.1 対象装置に対応するネットワーク OS 名称

対象装置	ネットワーク OS 名称
ApresiaNP シリーズ	apresia.aeosnp.aeosnp
ApresiaLightGM200 シリーズ	apresia.apl.apl
Apresia15000/13200 シリーズ	apresia.aeos8.aeos8
Apresia5400/3400 シリーズ	apresia.aeos7.aeos7

2.5.4 Playbook の TASK が Fail する条件について

2.5.4.1 ApresiaNP シリーズ、ApresiaLightGM200 シリーズ向けの場合

■ 定義箇所 `aeosnp.py` ファイル、`apl.py` ファイル

`./collections/ansible_collections/apresia/aeosnp/plugins/terminal`

`./collections/ansible_collections/apresia/apl/plugins/terminal`

■ Fail 条件定義内容

この正規表現にマッチする文字列がターミナルに出力されるとプレイブックの実行結果が `fail` になります。

```
terminal_stderr_re = [  
    re.compile(br"% ?Error"),  
    # re.compile(br"^% ¥w+", re.M),  
    re.compile(br"% ?Bad secret"),  
    re.compile(br"[¥r¥n%] Bad passwords"),  
    re.compile(br"invalid input", re.I),  
    re.compile(br"(?:incomplete|ambiguous) command", re.I),  
    re.compile(br"connection timed out", re.I),  
    re.compile(br"^[¥r¥n]+ not found"),  
    re.compile(br"'[^']*' +returned error code: ?¥d+"),  
    re.compile(br"Bad mask", re.I),  
    re.compile(br"% ?(¥S+) ?overlaps with ?(¥S+)"), re.I),  
    re.compile(br"[%¥S] ?Error: ?[¥s]+"),  
    re.compile(br"[%¥S] ?ERROR: ?[¥s]+"),  
    re.compile(br"¥n ?ERROR:[ ¥S]"),  
    re.compile(br"[%¥S] ?Informational: ?[¥s]+", re.I),  
    re.compile(br"Command authorization failed"),  
    re.compile(br"Command Rejected: ?[¥s]+", re.I),  
]
```

■ Fail 発生状況

発見されている不具合として、コマンドモジュールを用いて、show tech-support を取得した際、結果に” ERROR”で始まる行があるため、Playbook 実行がそこでエラーを出して停止します。これを回避するには、上記定義において、`re.compile(br"^\n ?ERROR:[\tS]*")`、行をコメントアウトすれば回避できます。副作用としては、コマンド実行して、” ERROR:”で始まるエラーメッセージが出力された場合、エラーだと判断できなくなります。

```
#ssl csr  
  
ERROR: No valid certificate request.
```

2.5.4.2 Apresia15000/13200、Apresia5400/3400 シリーズ向けの場合

■ 定義箇所 aeos8.py, aeos7.py ファイル

./collections/ansible_collections/apresia/aeos8/plugins/terminal

./collections/ansible_collections/apresia/aeos7/plugins/terminal

■ Fail 条件定義内容

この正規表現にマッチする文字列がターミナルに出力されるとプレイブックの実行結果が fail になります。

```
terminal_stderr_re = [  
    re.compile(rb"% ?Error"),  
    # re.compile(rb"^% ¥w+", re.M),  
    re.compile(rb"% ?Bad secret"),  
    re.compile(rb"[¥r¥n%] Bad passwords"),  
    re.compile(rb"invalid input", re.I),  
    re.compile(rb"(?:incomplete|ambiguous) command", re.I),  
    re.compile(rb"connection timed out", re.I),  
    re.compile(rb"[^¥r¥n]+ not found"),  
    re.compile(rb"'[^']' +returned error code: ?¥d+"),  
    re.compile(rb"Bad mask", re.I),  
    re.compile(rb"% ?(¥S+) ?overlaps with ?(¥S+)", re.I),  
    re.compile(rb"%¥S ?Error: ?[¥s]+"),  
    re.compile(rb"%¥S ?ERROR: ?[¥s]+"),  
    re.compile(rb"¥n ?ERROR:[ ¥S]"),  
    re.compile(rb"%¥S ?Informational: ?[¥s]+", re.I),  
    re.compile(rb"Command authorization failed"),  
    re.compile(rb"Command Rejected: ?[¥s]+", re.I),  
    re.compile(rb"Verify Error:", re.I),  
    re.compile(rb"VTY configuration is locked by other VTY"),  
    re.compile(rb"¥n% (?!" + non_error_patterns + rb").*"),  
]
```


■ Fail 未発生、発生状況

発見されている不具合として、AEOS7 のコンフィグモジュールやコマンドモジュールを用いて、特定のコマンドを実行あるいは設定した際、結果のエラーメッセージの”%”の後ろに半角スペースが無い場や、”%”で始まらない場合、Playbook 実行が Fail になりません。

```
#show ip ospf 0 database
%OSPF: No router process 0
# show memory-card files
Fail to mount memory-card
```

また、AEOS7 のコンフィグモジュールを用いて、特定の設定を実施した際、ワーニングメッセージが表示されることがあります。この場合、構成情報には設定できますが、Playbook 実行結果はエラーとなるので、ご注意ください。

```
(config)# router ospf 100
(config-router)# neighbor 1.2.3.4
% Warning: There is no corresponding ip address for this neighbor
(config-router)# end
# show running-config | include neighbor 1.2.3.4
neighbor 1.2.3.4
#
```

2.5.5 装置の状態確認 Playbook について

装置の状態確認のために、ApresiaNP シリーズや、ApresiaLightGM200 シリーズの場合は、show environment コマンドが正常応答することで確認しています。以下、ApresiaNP シリーズ向けの例を示します。

2.5.5.1 Playbook の内容

aeosnp_command_show-environment.yml

```
- name: NP 向け Ansible 対応 aeosnp_command 動作確認
  hosts: all
  gather_facts: false
  tasks:
  - name: show environment
    apresia.aeosnp.aeosnp_command:
      commands:
      - show environment
```

2.5.5.2 実行結果 成功時

TASK の応答が ok となり、PLAY RECAP の結果も ok=1 となります。

```
(ansible_user@PC:~/Ansible/ansible/aeosnp/test$ ansible-
playbook -i aeosnp_test_hosts aeosnp_command_show-environment.yml

PLAY [NP 向け Ansible 対応 aeosnp_command 動作確認]
*****
*****

TASK [show environment]
*****
*****

ok: [10.249.183.96]

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

```
10.249.183.96          : ok=1    changed=0    unreachable=0
failed=0
```

2.5.5.3 実行結果 失敗時 (2例)

TASK の応答が fatal となり、Authentication failed が表示されています。

SSH 認証失敗のケースであり、装置との SSH 接続のための設定を見直してください。

```
TASK [show environment]
*****
*****
fatal: [10.249.183.96]: FAILED! => {"changed": false, "msg":
"Failed to authenticate: Authentication failed."}

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

TASK の応答が fatal となり、Unable to connect to port が表示されています。

装置との経路に異常が発生しています。ケーブルなど装置との接続状態を確認してください。

```
TASK [show environment]
*****
*****
fatal: [localhost]: FAILED! => {"changed": false, "msg": "[Errno
None] Unable to connect to port 22 on 127.0.0.1"}

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

2.5.6 構成情報保存 Playbook について

write memory コマンドを実行して、running-config を保存します。

コマンドに対するユーザー応答に y を返しています。以下、ApresiaNP シリーズ向けの例を示します。

2.5.6.1 Playbook の内容

aeosnp_command_write-memory.yml

```
- name: NP 向け Ansible 対応 aeosnp_command 動作確認
  hosts: all
  gather_facts: false
  tasks:

- name: write memory
  apresia.aeosnp.aeosnp_command:
    commands:
      - command: write memory
    prompt:
      - 'Destination filename startup-config?'
    answer:
      - "y"
```

2.5.6.2 実行結果 成功時

成功した場合、TASK [write memory]の応答は ok となります。

```
TASK [write memory]
*****
*****
ok: [10.249.183.96]
PLAY RECAP
*****
*****
*****
10.249.183.96          : ok=1    changed=0    unreachable=0
failed=0
```

2.5.6.3 実行結果 失敗時

失敗した場合、TASK [write memory]の応答は fatalとなります。ApresiaNP シリーズでは、SD カードが挿入されていた場合、SD カードにも構成情報を保存するという動作を実施しますが、SD カードのライトプロテクトスイッチを設定すると、ERROR: File not found が返るため、失敗します。

```

TASK [write memory]
*****
*****
fatal: [10.249.183.96]: FAILED! => {"changed": false, "msg":
"write memory
Destination filename startup-config?
[y/n]: y
Saving all configurations to NV-RAM.....
0 %100 %100 %Done.
Saving all configurations to SD-
Card..... 0 %
ERROR: File not found.
# "}

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

```

2.5.7 Playbook 実行時のトラブルシューティング

(1) Playbook のシンタックスチェック

Playbook 実行時の引数に、`--syntax-check` を付けることで、Playbook の実行の代わりに、シンタックスチェックが実行されます。あらかじめシンタックスに問題が無いことを確認できます。

(2) Playbook のデバッグ実行

Playbook 実行時の引数に`-v`、`-vv`、`-vvv`、`-vvvv` を付けると、`v` の数に応じて出力結果の表示に詳細情報が追加されます。

2.6 Apresia 各シリーズにおける SSH 接続について

Playbook 実行時に paramiko のエラーが発生している場合、Ansible 実行環境の問題である可能性があるため、以下をお試しください。

- ・paramiko ライブラリのインストール

```
$ pip install paramiko
```

- ・ansible.cfg にて、SSH ホストキー確認設定の無効化

```
[defaults]
host_key_checking=False
```

2.6.1 ApresiaNP シリーズの場合



ApresiaNP シリーズでは、SSH 接続時に送るコマンドやコンフィグの文字列長の最大値は 2048 文字となります。2048 文字を超えた場合、セッションが切断され、該当行の処理は未実行となります。

(1) SSH 接続を有効にする場合の手順例を、以下に示します。

- ・ ユーザアカウントおよびパスワードの設定 と SSH の認証方式を local に設定

```
sw1# configure terminal
sw1(config)# username adpro privilege 15 password adpro
sw1(config)# line ssh
sw1(config-line)# login local
```

- ・ SSH サーバーの有効化と、SSH アカウントの設定

```
sw1(config)# ip ssh server
sw1(config)# ssh user adpro authentication-method password
sw1(config)# end
```

- ・ RSA 鍵対の生成

```
sw1# crypto key generate rsa modulus 1024
```

- ・ 構成情報の保存

```
sw1# write memory
```

(2) ApresiaNP シリーズ ユーザーズガイド 9.3.1 章 SSH 接続をパスワード認証方式で行う場合 もご覧ください。

2.6.2 ApresiaLightGM200 シリーズの場合

(1) SSH 接続を有効にする場合の手順例を、以下に示します。

- ・ ユーザアカウントおよびパスワードの設定 と SSH の認証方式を local に設定

```
sw1# configure terminal
sw1(config)# username adpro privilege 15 password adpro
sw1(config)# line ssh
sw1(config-line)# login local
```
- ・ SSH サーバーの有効化と、SSH アカウントの設定

```
sw1(config)# ip ssh server
sw1(config)# ssh user adpro authentication-method password
sw1(config)# end
```
- ・ RSA 鍵対の生成

```
sw1# crypto key generate rsa modulus 1024
```
- ・ 構成情報の保存

```
sw1# write memory
```

2.6.3 Apresia15000/13200 シリーズ, Apresia5400/3400 シリーズの場合

(1) SSH 接続を有効にする場合の手順例を、以下に示します。

- ・ ユーザアカウントおよびパスワードの設定 と SSH の認証方式を local に設定

```
sw1# configure terminal
sw1(config)# username adpro adpro
sw1(config)# password adpro
New password:
Retype new password:
```
- ・ RSA 鍵対の生成 (Apresia5400/3400 シリーズの場合のみ、未実施ならば生成が必要。Apresia15000/13200 シリーズの場合、自動生成されるので不要。)

```
sw1(config)# end
sw1# sshd keygen rsa
sw1# sshd keygen rsa1
sw1# configure terminal
```
- ・ SSH サーバーの有効化と、SSH アカウントの設定

```
sw1(config)# sshd server
```
- ・ 構成情報の保存

```
sw1# write memory
```

企業・DC・SMB 向け Apresia 用
Ansible 5(Ansible-core 2.12)対応 Ver.1.03.01

Copyright(c) 2023 APRESIA Systems, Ltd.
2023 年 10 月 初版

APRESIA Systems 株式会社
東京都中央区築地二丁目 3 番 4 号
築地第一長岡ビル

<https://www.apresiasystems.co.jp/>