

ApresiaNP シリーズ向け Ansible 4(ansible-core 2.11)対応

APRESIA Systems 株式会社

制定・改訂来歴表

No.	年 月 日	内 容
-	2022 年 03 月 31 日	新規作成
A	2022 年 09 月 30 日	<ul style="list-style-type: none"> ・2.5 Playbook の実行について 節の注意事項を修正 ・2.5.7 Playbook 実行時のトラブルシューティング の項を追加 ・2.6.1 SSH 接続 の項を修正

目次

制定・改訂履歴表	1
目次	2
1. はじめに	3
1.1 ライセンスや商標について	3
2. 説明	4
2.1 提供するファイル構成について	4
2.2 インストール方法	5
2.3 実行環境構築 Ansible や Python のインストール	5
2.3.1 モジュールファイルのインストール	7
2.4 提供するモジュールに関する情報	8
2.4.1 aeosnp_config (モジュールファイル名 aeosnp_config.py)	8
2.4.1.1 モジュールの概要	8
2.4.1.2 オプション	10
2.4.1.3 記述サンプル	10
2.4.1.4 戻り値	12
2.4.2 aeosnp_command (aeosnp_command.py)	13
2.4.2.1 モジュールの概要	13
2.4.2.2 オプション	13
2.4.2.3 記述サンプル	13
2.4.2.4 戻り値	15
2.5 Playbook の実行について	16
2.5.1 実行前の準備	16
2.5.2 コネクションプラグイン	16
2.5.3 ネットワーク OS	16
2.5.4 Playbook の TASK が Fail する条件について	16
2.5.5 装置の状態確認 Playbook について	17
2.5.5.1 Playbook の内容	17
2.5.5.2 実行結果 成功時	18
2.5.5.3 実行結果 失敗時 (2 例)	18
2.5.6 構成情報保存 Playbook について	19
2.5.6.1 Playbook の内容	19
2.5.6.2 実行結果 成功時	19
2.5.6.3 実行結果 失敗時	20
2.5.7 Playbook 実行時のトラブルシューティング	20
2.6 ApreisaNP シリーズにおける SSH 接続について	21
2.6.1 SSH 接続	21

1. はじめに

ApresiaNP シリーズ向けの 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

装置名称	Playbook を実行した環境における Ansible 及び Python バージョン
ApresiaNP2100 シリーズ	Ansible 4(ansible-core 2.11) Python 3.8.12



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。

1.1 ライセンスや商標について

ApresiaNP シリーズ向け 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 3.1.1 に提供するファイル構成について示します

表 3.1.1 提供するファイル構成

フォルダおよびファイル名	説明
<pre> apresia ├── aeosnp │ └── test │ ├── aeosnp_command_show-environment.yml │ ├── aeosnp_command_write-memory.yml │ ├── aeosnp_show-version.yml │ ├── aeosnp_test_hosts │ └── ansible.cfg </pre>	<p>test フォルダには、評価用のサンプル Playbook と設定ファイルが格納されています。</p>
<pre> ├── collections │ └── ansible_collections │ └── apresia │ └── aeosnp │ ├── meta │ │ └── runtime.yml │ └── plugins │ ├── action │ │ ├── aeosnp.py │ │ └── _init_.py │ ├── cliconf │ │ ├── aeosnp.py │ │ └── _init_.py │ ├── modules │ │ ├── aeosnp_command.py │ │ ├── aeosnp_config.py │ │ └── _init_.py │ ├── module_utils │ │ ├── _init_.py │ │ └── network │ │ ├── _init_.py │ │ └── aeosnp │ │ ├── aeosnp.py │ │ └── _init_.py │ └── terminal │ ├── aeosnp.py │ └── _init_.py </pre>	<p>プラグインとモジュール群です。</p>

2.2 インストール方法

2.3 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)

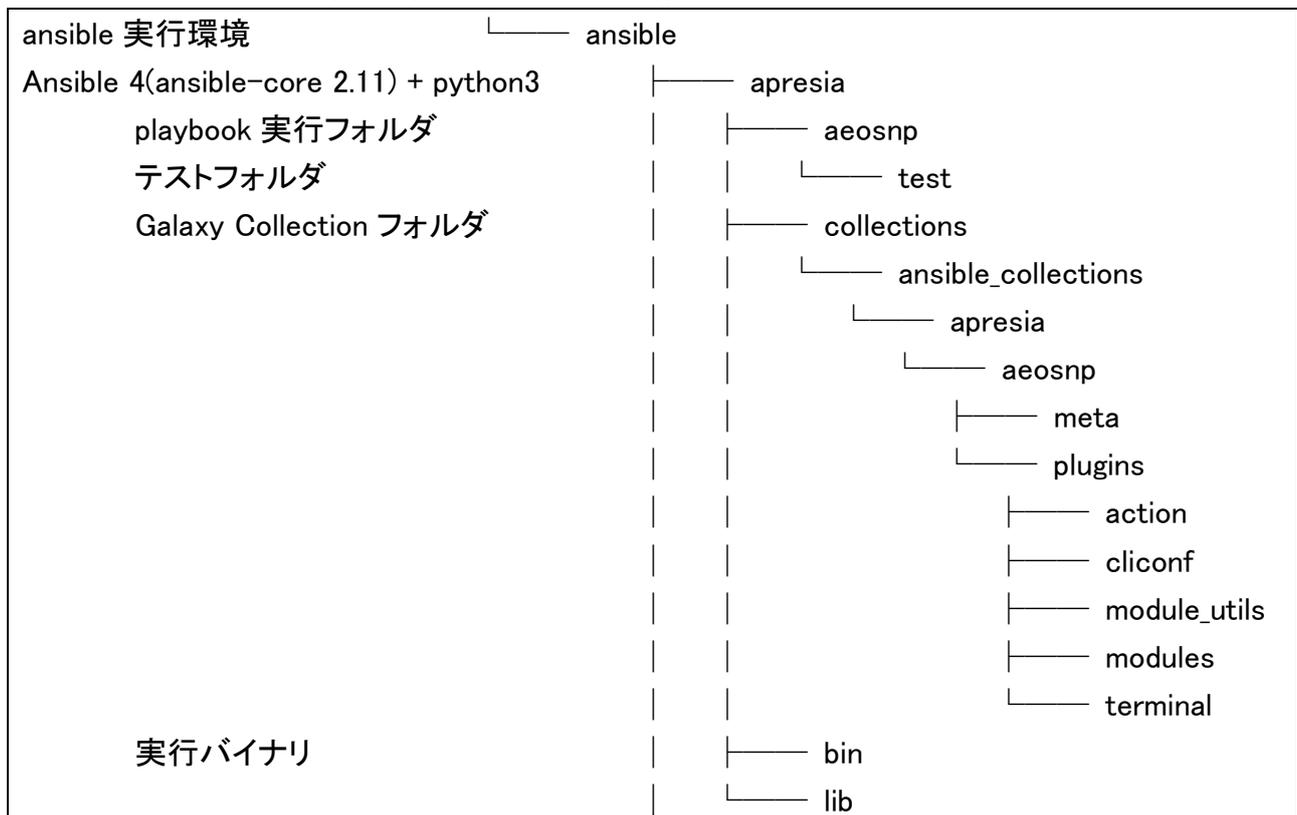
• Ansible 4(ansible-core 2.11) + python 3.8.12

Ansible バージョン	2.11.1
Python バージョン	3.8.12
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

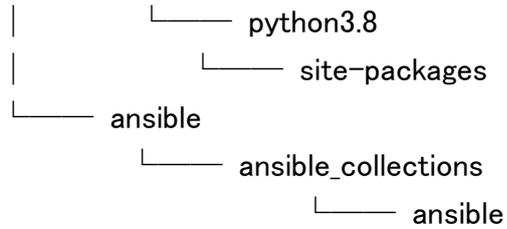
[フォルダ構成]

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 3.1.2 ansible 実行環境のディレクトリ構成



ライブラリ



(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 2.11 以降であることを御確認ください。)

```
$ pip install "ansible>=4,<5"
```

```
$ ansible --version
```

```
ansible [core 2.11.xx]
```

```
:
```

(7) virtualenv 終了

```
$ deactivate
```

```
$ cd ../
```

2.3.1 モジュールファイルのインストール

入手した `apresia-aeosnp-ansible4_R10101.tar.gz` (ファイル名は異なることがあります) を任意のフォルダに展開します。ここでは、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` フォルダにアーカイブファイルの `apresia` フォルダを展開したとします。

(1) 「`ansible.cfg`」に Galaxy Collection の配置場所として

2.1 提供するファイル構成について `aeosnp` モジュール構成の下記フォルダを指定してください。

`apresia/collections/`

test フォルダに置いた `ansible.cfg` の例

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

Galaxy Collection への相対パスを設定しています。

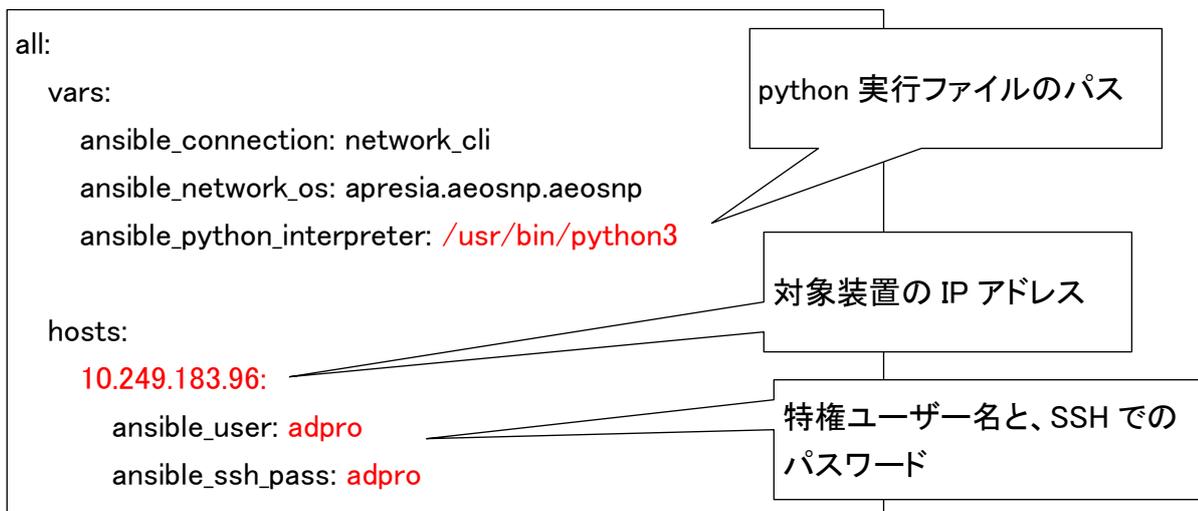
これにより `collections_path` に指定したディレクトリ直下の `ansible_collections` フォルダ内に存在しているモジュールを使用可能になります。

なお `aeosnp` モジュールは下記のフォルダにあります。

`apresia/collections/ansible_collections/apresia/aeosnp/`

(2) 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環境に応じて、変更してください。



2.4 提供するモジュールに関する情報

本章では `aeosnp_config` と `aeos_command` モジュールについて説明します。モジュール全般に関係する情報は、3.4 Playbook の実行について に記載したので、合わせてごらんください。

! 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
```

2.4.1 aeosnp_config (モジュールファイル名 aeosnp_config.py)

2.4.1.1 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 NP 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
```



装置の構成情報において、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。

2.4.1.2 オプション

本モジュールのオプションを表 2.3.1 に示します。

表 2.3.1 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。 ・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。

2.4.1.3 記述サンプル

以下に記述サンプルを示します。

装置のプロンプトにインベントリ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
```

2.4.1.4 戻り値

本モジュールの戻り値を表 2.3.1.2 に示します。

表 2.3.1.2 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	リスト	prompt foo, router ospf 1, router-id 192.0.2.1
updates	commands と同じ	常時	リスト	prompt foo, router ospf 1, router-id 192.0.2.1

2.4.2 aeosnp_command (aeosnp_command.py)

2.4.2.1 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。

2.4.2.2 オプション

表 2.3.2.1 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
```

2.4.2.3 記述サンプル

以下に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
```

2.4.2.4 戻り値

本モジュールの戻り値を表 2.3.2.2 に示します。

表 2.3.2.2 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したときの状態をリストで返します	Failしたとき	リスト	..., ...
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。	リスト	..., ...
stdout_lines	stdout をリストで分割したもの	常時 action plugin などからの低レベルのエラーを含みます。	リスト	-- '...' - '...' -- '...' -- '...'
failed_conditions	Failしたときの状態をリストで返します	Failしたとき	リスト	..., ...

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 秒を指定しています。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、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
```

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 ApresiaNP 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。

ApresiaNP 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、ApresiaNP 装置の SSH 接続設定については、2.6 ApresiaNP 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

に下記の文字列を設定してください。

apresia.aeosnp.aeosnp

2.5.4 Playbook の TASK が Fail する条件について

■ 定義箇所 aeosnp.py ファイル

/apresia/collections/ansible_collections/apresia/aeosnp/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),  
]
```

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、show environment コマンドが正常応答することで確認しています。

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) 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。

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]
```


2.6 ApresiaNP シリーズにおける SSH 接続について



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。

2.6.1 SSH 接続

(1) SSH 接続を有効とするには、以下の手順を実施する必要があります。

- ・ 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

(例)

```
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

(例)

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

- ・ RSA 鍵対の生成

(例)

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

- ・ 構成情報の保存

(例)

```
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。

ApresiaNP シリーズ向け Ansible 4(ansible-core 2.11)
対応

Copyright(c) 2022 APRESIA Systems, Ltd.
2022 年 09 月 A 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