AlaxalA

The Guaranteed Network

いちばん近くで、もっと先へ。

AX modules for Ansible
活用ガイド

第 3 版

資料 No. NTS-18-R-001

アラクサラネットワークス株式会社

はじめに

本資料は、アラクサラの AX modules for Ansible を活用したネットワークシステムの運用自動化に役立てるものとして AX modules for Ansibleの使用方法やplaybook記述例について記載しています。またplaybookのサンプルを提供します。

本書の対象読者

本資料は、Ansible を用いた構成管理についての基礎知識と、CLI によるアラクサラ機器の操作に関する知識を有する方を前提に記載しています。

関連資料

- AX modules for Ansible 運用ガイド
  (http://www.alaxala.com/jp/techinfo/archive/manual/ANSIBLE/PDF/1_2/OP-GUIDE-SOFT-AM-2303_R2.pdf)

- Ansible に関連するドキュメント
  (http://docs.ansible.com/ansible/2.7/)

本資料使用上の注意事項

- 本資料に記載の内容は、弊社が特定の環境において基本動作を確認したものであり、機能・性能・信頼性についてあらゆる環境条件すべてにおいて保証するものではありません。またマニュアルの補助資料としてご利用いただけますようお願いいたします。

- 本資料の内容は AX modules for Ansible Version 1.2 を対象に記載しています。また改良のため予告なく変更する場合があります。

- playbook をコピーして活用する際は、別途提供している サンプル playbook のファイルからコピーしてご使用ください。

輸出時の注意

本ソフトウェアを輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規制など外国の輸出関連法規をご確認の上、必要な手続きをおとりください。

なお、不明な場合は、弊社担当営業にお問い合わせ下さい。

商標一覧

- アラクサラの名称およびロゴマークは、アラクサラネットワークス株式会社の商標および登録商標です。
- Ansible は，Red Hat, Inc.およびその子会社の商標または登録商標です。
- Red Hat Enterprise Linux は，Red Hat, Inc.の商標または登録商標です。
- CentOS は，Red Hat, Inc.の商標または登録商標です。
- Python は，Python Software Foundation の商標または登録商標です。
- Ethernet は、富士ゼロックス株式会社の登録商標です。
- イーサネットは、富士ゼロックス株式会社の登録商標です。
- そのほかの記載の会社名、製品名は、それぞれの会社の商標または登録商標です。

改訂履歴

版数	Rev.	日付	変更内容	変更箇所
初版	-	2018.04	初版発行	-
第 2 版	-	2019.01	･AX modules for Ansible 運用ガイド、および、Ansible に関連するドキュメントのリンクを更新 ･本資料が対象とする AX modules for Ansible のバージョンを 1.2 に更新	はじめに
			「表 2.1-1 管理ホストの動作要件」を更新	2.1.2(1)
			「表 2.1-2 ネットワーク装置の動作要件」を更新	2.1.2(2)
			本項を追加	2.1.5
			YAML の構文についてのサイトのリンクを更新	3.2(2)
			Ansible 2.7.2、および、AX modules for Ansible Version 1.2 に対応した記述内容に変更	playbook 全般
第 3 版	-	2019.04	「3.3 基本的なモジュール，ディレクティブ，プラグインの記述サンプル」に 3.3.7～3.3.10 を追加	3.3.7～3.3.10 追加
			「4. 各ユースケースでのサンプル playbook」に以下を追加 ･4.4 ネットワーク装置の障害情報採取 ･4.5 ネットワーク装置のソフトウェアアップデート ･4.6 NTP サーバ追加，移設にともなう ネットワーク装置の設定	4.4 追加 4.5 追加 4.6 追加
			サンプル playbook の提供について	付録 2 追加

目次

1. Ansible とは

1.1 ネットワーク運用の課題と自動化

ネットワークシステムは社会の重要な通信インフラとなり、そのインフラ上で多様なサービスが構築され、24 時間いつでもビジネスが展開されています。このような ICT 利用環境では、ネットワーク装置だけではなく、サーバやストレージなども複雑に連携するようになっています。また、ネットワーク装置には、ネットワークへのアクセス制御や帯域制御、VPN の動的設定等、高度な機能が開発・実装されるなど、昨今のネットワークシステムは複雑さを増しています。このような環境のなか、機器のオペレータやユーザが高度な機能と操作を意識せずに、より簡単に利用できるような環境を構築していくことが、早急に求められています。

1.2 Ansible の概要

システム管理者は、多くの IT 機器の運用管理（アプリケーションの追加、パッチ適用、構成変更など）を行っています。また、新しいシステム環境を構築する際には、複数の IT 機器にいくつもの設定コマンドを実行するなどといった同一オペレーションを実施することが多々あります。

これらのオペレーションをシステム管理者が手動対応するには、時間・労力的に困難かつ人為的ミスが懸念されています。そこで利用されてきたのが、構成管理ツールです。構成管理ツールとは、IT 機器に対して自動でアプリケーションのインストールや設定を一括しておこなうツールです。

構成管理ツールはいくかありますが、現在 注目を集めているのが Ansible です。これまで Ansible は、サーバの運用管理のツールとして利用されてきましたが、様々なネットワークベンダのデバイスがサポートされてきました。

Ansible 適用によりサーバやストレージだけでなく、ネットワークも同一基盤で運用管理できるようになります。

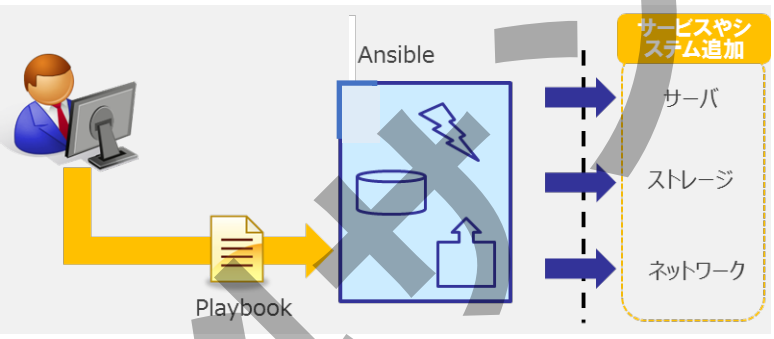


図 1.2-1 Ansible の概要

Ansible の主な特徴やメリットは以下のとおりです。

◆ **特徴**

- **エージェントレスな構成管理ツール**
  アラクサラ機器を含め Ansible の管理対象となる管理対象装置に、エージェントのインストールは不要です。
- **シンプルな記述**
  従来の構成管理ツールで手順を記述する場合に求められたプログラミングの知識は不要です。
- **Ansible モジュールの拡張性が高い**
  Ansible には管理対象の機器やシステムまたはサービスごとに様々なモジュールが用意されています。

◆ **メリット**

- オペレーション手順がコード化するため、人為的ミスを削減できます。
- 安全かつ効率的にシステム構成を管理・維持が可能で、作業工数を軽減できます。
- 運用後における手順の変更が、作業手順書に反映されていないことによる作業漏れを撲滅できます。
- 冪等性(何度実行しても最終形が一致)を考慮した設計が可能です。

1.3 Ansible の利用イメージ

運用の自動化で構成管理ツールの Ansible を利用した場合のイメージを下図に示します。管理部門やサービス利用者は、定型の繰り返し作業を運用部門の作業者に依頼するのではなく、自身が構成管理ツールの Ansible を実行することで対象装置に対する設定を反映します。つまり、定型の繰り返し作業を自動化することで、作業時間短縮や人的ミスの削減、属人性の排除が可能になります。

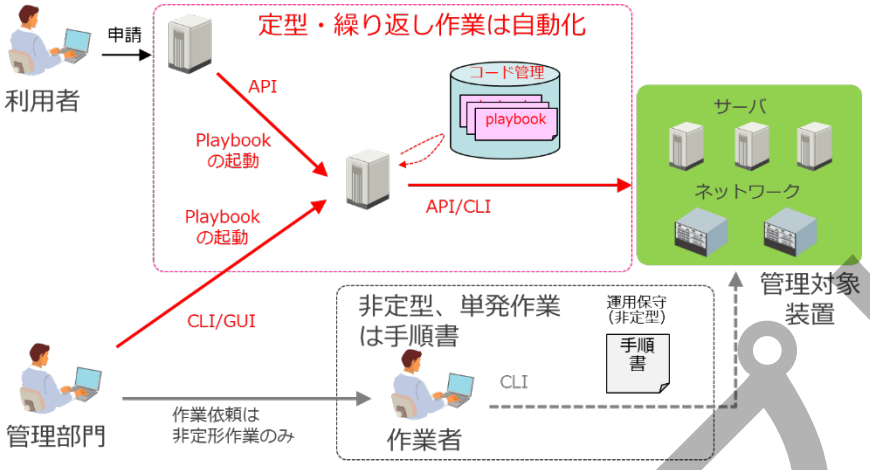


図 1.3-1 Ansible の利用イメージ

1.4 Ansible の構造

Ansible の構造を下図に示します。Ansible は大きく分類すると設定ファイル、プラグイン、モジュールの 3 つで構成されています。設定ファイルは管理対象装置について記述した inventory、管理対象装置に設定する手順を記述した playbook があります。プラグインについては、playbook に記載された制御を実現するところで、出力内容確認、条件分岐、タイマ、ファイル作成等、種類がたくさんあります。モジュールについては各ベンダが提供するもので管理対象装置を制御するプログラムになります。

Ansible では、どのような手順、条件で設定するかという設定シナリオで playbook を作成・実行することでサーバやネットワーク装置の複数台に対して一括設定を可能にしています。また一度作成した playbook は他でも再利用することができます。

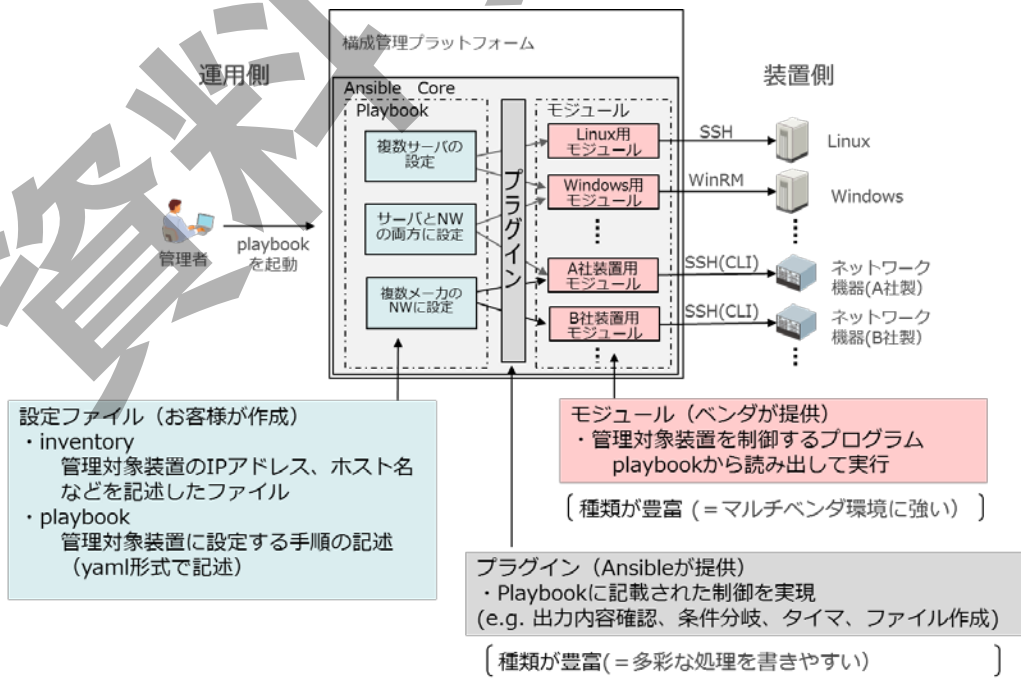


図 1.4-1 Ansible の構造

2. アラクサラの AX モジュール

2.1 AX モジュールの概要

AX modules for Ansible(以下 AX モジュール)は、構成管理ツールである Ansible を利用して、アラクサラのネットワーク装置を目的の状態にすることを可能にする構成管理機能を提供します。

2.1.1 動作概要

AX モジュールは、Ansible がインストールされている管理ホスト上で Python のプログラムとして動作し、管理ホストから SSH により ネットワーク装置へログインして、装置の構成管理をおこないます。AX モジュールの動作概要を以下に示します。

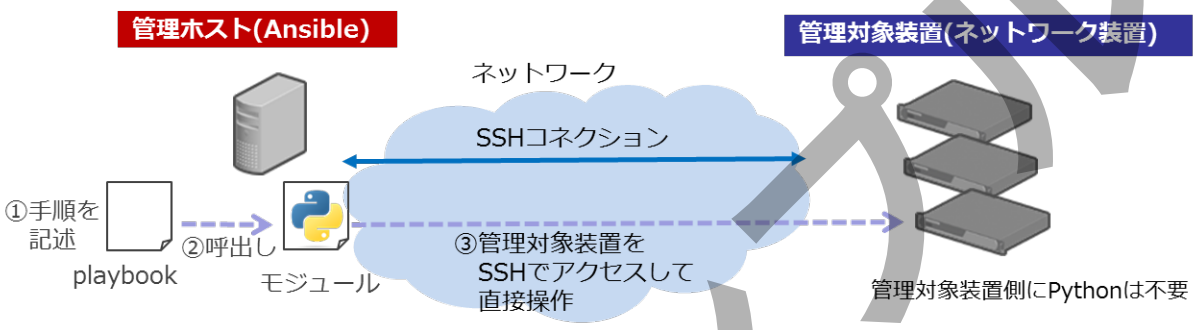


図 2.1-1 動作概要

2.1.2 動作要件

AX モジュールを動作させるにあたり、管理ホストおよびネットワーク装置について必要な用件について説明します。

(1) 管理ホスト

AX モジュール(Version 1.2)を動作させる管理ホストに関する動作要件について以下に示します。

表 2.1-1 管理ホストの動作要件

項目	構成
オペレーティングシステム	Red Hat Enterprise Linux 7.2 以降（64bit） または CentOS 7.2 以降（64bit）
Python	Python 2.7 以降 または Python 3.5 以降
Ansible	Ansible 2.7.2

(2) ネットワーク装置

サポートするアラクサラのネットワーク装置に関する動作要件について以下に示します。AX モジュールのバージョンに合わせて、ネットワーク装置のソフトウェアバージョンを準備ください。またネットワーク装置には、SSH の設定が必須となります。

表 2.1-2 ネットワーク装置の動作要件

AX モジュールのバージョン	ネットワーク装置
Ver. 1.0 以降	AX4630S Ver. 11.15.A 以降
	AX3660S Ver. 12.1.B 以降
Ver. 1.1 以降	AX6000S Ver. 11.9.S 以降
	AX2500S Ver. 4.10 以降
	AX2200S Ver. 2.5.B 以降
	AX2100S Ver. 2.6 以降
	AX260A Ver. 4.10 以降
Ver. 1.2 以降	AX8600R Ver. 12.8.D 以降
	AX8600S Ver. 12.8.D 以降
	AX8300S Ver. 12.8.D 以降

2.1.3 インストール方法

AX モジュールのインストールは、「AX modules for Ansible 運用ガイドの 2.3 インストール」を参照して、AX モジュールをインストールするために必要なパッケージのインストールと Ansible のインストールをおこなってください。またインストール後に AX モジュールを動作させるために必要な Ansible の設定ファイルの編集をおこなってください。(ログ機能の有効化、コマンドタイムアウト時間の設定)

2.1.4 AX モジュールの機能

AX モジュールに含まれるモジュールの種類と機能を以下に示します。
『ax_command』は、ネットワーク装置上で運用コマンドを実行し、その実行結果を取得します。『ax_config』は、コンフィグレーションコマンドモードに遷移し、ネットワーク装置にコンフィグレーションを設定します。『ax_facts』は、装置情報，ハードウェア情報，コンフィグレーション情報，インタフェース情報のそれぞれについて取得コマンドをネットワーク装置上で実行し、ネットワーク装置から装置情報を収集します。

表 2.1-3 AX モジュールの一覧

モジュール名	機能
ax_command	運用コマンドの実行
ax_config	コンフィグレーションの設定
ax_facts	装置情報の収集

AX モジュールのパラメータなど詳細な内容については、「AX modules for Ansible 運用ガイドの 3. モジュール」 を参照ください。

2.1.5　AX モジュールの Ansible 対応バージョンについて

AX モジュールが対応する Ansible のバージョンについて以下に示します。

表 2.1-4　AX モジュールが対応する Ansible のバージョン一覧

AX モジュールのバージョン	対応する Ansible のバージョン
AX modules for Ansible Version 1.0	Ansible 2.4.0.0 のみ
AX modules for Ansible Version 1.1	Ansible 2.4.0.0 のみ
AX modules for Ansible Version 1.2	Ansible 2.7.2 のみ

表 2.1-4 に記載の組み合わせ以外では AX モジュールの動作を保証いたしません。

2.2 AX モジュールの使用方法

AX モジュールの使用方法を playbook 例を用いて説明します。

2.2.1 ax_command

<playbook 例: 抜粋>

```
tasks:
  - name: wait for activation
    ax_command:
      commands:              # 運用コマンドを指定します
        - show system
      wait_for:              # 特定の文字列が含まれるまで、次のアクションを停止します
        - "result[0] contains 'Main board : active'"
                             # コマンド出力結果が result 変数に自動的に保存されます

  - name: execute commands
    ax_command:
      commands:              # 運用コマンドをリスト形式で指定します
        - show system
        - show version
        - show running-config
    register: system_info0     # タスクの register ディレクティブで、別変数に保存します
```

- 「commands」パラメータに実行したい運用コマンドをリスト形式で指定します。コンフィグレーションコマンドは指定できません。

2.2.2 ax_config

<playbook 例: 抜粋>

```
tasks:
  - name: configure vlan
    ax_config:
      lines:                          # コンフィグレーションコマンドを指定します
        - vlan 100

  - name: configure interface
    ax_config:
      lines:                  # コンフィグレーションコマンドをリスト形式で指定します
        - switchport mode access
        - switchport access vlan 100
      parents: interface gigabitethernet 1/0/1    # コンフィグレーションの階層を指定します
      save_when: modified         # 設定後のコンフィグレーション保存可否を指定します
```

- 「lines」パラメータに実行したいコンフィグレーションコマンドをリスト形式で指定します。冪等性の考えにより、既にネットワーク装置に設定済みのコマンドは実行されません。
- 「parents」パラメータでコンフィグレーションコマンドを実行したい階層を指定します。
- 「save_when」パラメータでコンフィグレーションの保存要否を指定します。省略時は、never になります。
  - always = 常時保存
  - modified = 変更発生時のみ保存
  - never = 常時非保存

2.2.3 ax_facts

<playbook 例: 抜粋>

```
tasks:
  - name: gather fatcs
    ax_facts:
      gather_subset:    # 収集する装置情報種別を指定します
        - "!all"
    register: result

  - name: debug
    debug:              # 収集した装置情報種別を表示します
      var: result
```

・ 収集したい装置情報種別を「gather_subset」パラメータでリスト形式に指定します。「gather_subset」パラメータの設定範囲は以下の表のとおりです。

表 2.2-1 gather_subset パラメータ

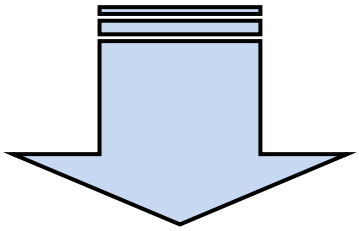
指定項目	筐体情報	インタフェース情報	コンフィグ情報	メモリ情報
all	○	○	○	○
!all	○	×	×	×
config	○	×	○	×
!config	○	○	×	○
interfaces	○	○	×	×
!interfaces	○	×	○	○
hardware	○	×	×	○
!hardware	○	○	○	×

凡例 ○: 採取指示 ×: 未採取指示

・ ax_facts で採取できる装置情報は以下のとおりです。

- 筐体情報 【常時採取】
  - 筐体シリアル、ホスト名、ソフトウェアバージョン
- インタフェース情報【interfaces】
  - 回線速度 : gigabitethernet, tengigabitethernet など
  - 運用状態 : 運用コマンド「show port」の 'Status' 列
  - 回線種別 : 運用コマンド「show port」の 'Speed' 列
  - 二重化状態 : 運用コマンド「show port」の 'Duplex' 列
  - MTU : 運用コマンド「show port」の 'FrLen' 列
- コンフィグ情報【config】
  - 「running-config」
- メモリ情報【hardware】
  - 運用コマンド「show system」のメモリ情報

気になる続きは…



・アラクサラ　インテグレータ会員

または

・ビジネスパートナー様会員

にご登録いただければ、全てをご覧いただけます！

アラクサラ インテグレータ会員またはビジネスパートナー様会員へ登録することで、アラクサラ製品のご利用にあたり役立つ各種資料(システム構築ガイドなど)を全て閲覧することができます。ぜひこの機会にご登録下さい。

アラクサラネットワークス株式会社

〒212-0058
川崎市幸区鹿島田一丁目１番２号 新川崎三井ビル西棟
http://www.alaxala.com/