



# IT Automation

## Ansible Driver 【実習編】

※本書では「Exastro IT Automation」を「ITA」として記載します。

# 目次

## ■はじめに

### ① Ansible-Legacy

1. [作業環境とシナリオ](#)
2. [必要なファイルの作成](#)
3. [Movementの設定](#)
4. [Conductorの作成](#)
5. [オペレーションの登録](#)
6. [機器一覧への登録](#)
7. [パラメータシート作成](#)
8. [データの登録](#)
9. [代入値自動登録設定](#)
10. [代入値・対象ホストの確認](#)
11. [作業の実行](#)

### ② Ansible-LegacyRole

1. [作業環境とシナリオ](#)
2. [ロールパッケージの準備](#)
3. [Movementの設定](#)
4. [オペレーションの登録](#)
5. [機器一覧への登録](#)
6. [パラメータシート作成](#)
7. [データの登録](#)
8. [代入値自動登録設定](#)
9. [代入値と対象ホストの確認](#)
10. [作業の実行](#)

### ③ Ansible-Pioneer

1. [作業環境とシナリオ](#)
2. [対話ファイルの作成](#)
3. [OS種別の作成](#)
4. [Movementの設定](#)
5. [オペレーションの登録](#)
6. [機器一覧への登録](#)
7. [パラメータシート作成](#)
8. [データの登録](#)
9. [代入値自動登録設定](#)
10. [代入値・対象ホストの確認](#)
11. [作業の実行](#)

A) [付録](#)

# はじめに 本書の使い方

## 本書の使い方

### ● 3つのシナリオを体感する

「Ansible-Legacy」「Ansible-LegacyRole」「Ansible-Pioneer」  
3モードを実際に利用し、それぞれの強みと利用法を体感いただけます。

各シナリオは独立しており、必要な章を選んで学習できます。



第1章 Ansible-Legacy編  
playbook(YAMLファイル)を登録、利用する



第2章 Ansible-LegacyRole編  
ロールパッケージを登録、利用する。



第3章 Ansible-Pioneer編  
ITAが提供するAnsible独自モジュールを用い、  
対話ファイルを登録、利用する。

# 第1章 Ansible-Legacy編

# 1.1 作業環境とシナリオ

## 作業環境

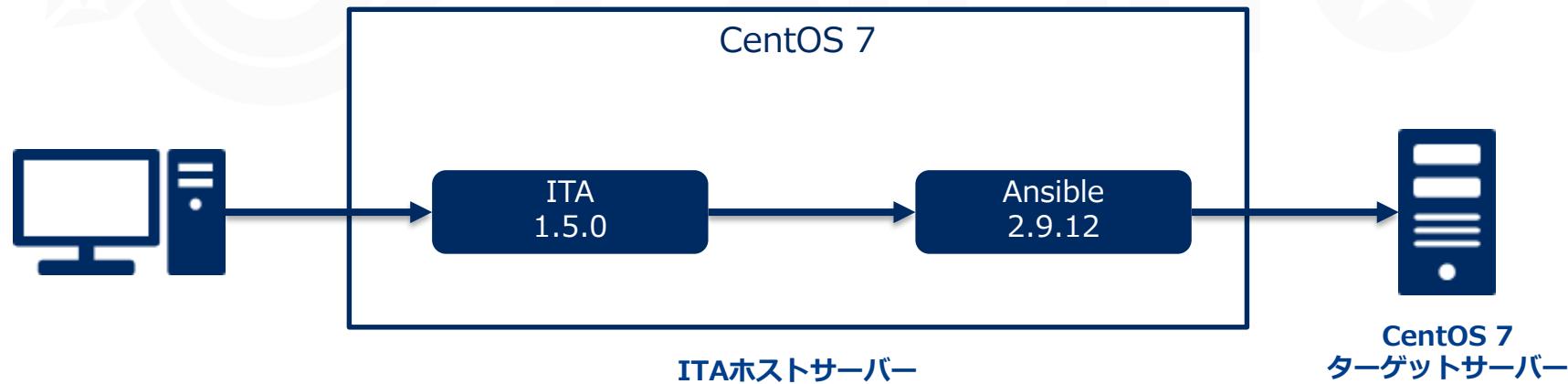
本章で使用する作業環境は以下の通りです。(第2章Ansible-LegacyRole編に共通)  
ITAホストサーバーとは別に、ターゲットとなるサーバーをご用意ください。

### ITAホストサーバー

- CentOS 7 (※1)
- ITA 1.5.0
- Ansible 2.9.12

### ターゲット

- CentOS 7 (※2)



※1 今回はホストサーバーとしてCentOS7を利用致しますが、ITAはRHEL7系およびRHEL8系のOSで導入いただけます。

※2 Ansibleの動作対象となれるOSであれば、問題なく利用いただけます。

# 1.1 作業環境とシナリオ

## シナリオ

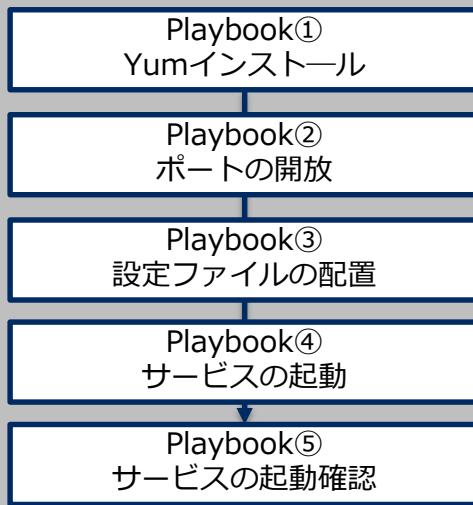
大別して3ステップから成る作業を実行し、**playbookの再利用性**を体感いただけます。

- ① Movementを組み合わせてConductorを作成する。
- ② メニューを作成し、パラメータを登録する。
- ③ 作成したConductorを実行する。

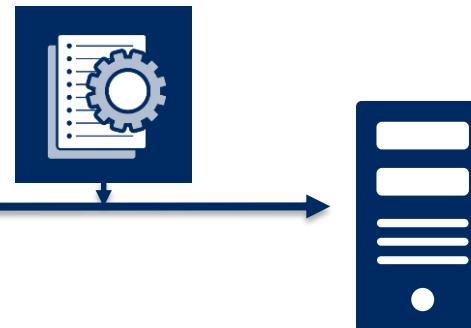
今回は「Apache・Tomcat」両サービスのインストールと起動を行います。

## シナリオのイメージ

### ① Conductorを1つ作成する



### ② パラメータシートを作成 (playbookへの代入値や作業対象を管理)



### ③ 両サービスの追加を確認する ・Apache ・Tomcat

ターゲットサーバ

# 1.2 必要なファイルの作成 (1/3)

## playbookの作成

本シナリオで使用する5つのplaybookを作成しましょう。

【注意】文字コードは“UTF-8”、改行コードは“LF”、拡張子は“yml”で作成してください。

```
- name: install package with yum
  yum:
    name: "{{ item }}"
    state: present
  with_items: "{{ VAR_package_name }}"
```

**ファイル名：1-yum\_install.yml**

指定したパッケージをインストールします。  
変数には複数具体値変数が代入されます。

```
- name: open ports
  firewalld:
    port: "{{ VAR_port_number }}"
    state: enabled
    permanent: yes
    immediate: true
```

**ファイル名：2-open\_port.yml**

指定したポート宛の通信を許可します。

# 1.2 必要なファイルの作成 (2/3)

## playbookの作成

```
- name: copy httpd.conf
copy:
  src: "{{ CPF_httpd_conf }}"
  dest: /etc/httpd/conf/httpd.conf
  owner: root
  group: root
  mode: 0644
  backup: yes
when: 'VAR_service_name == "httpd"'
```

```
- name: start service
service:
  name: "{{ VAR_service_name }}"
  state: started
  enabled: yes
```

```
- name: check if service is running and enabled
command: 'systemctl status {{ VAR_service_name }}'
register: command_result
failed_when:
  - "enabled" not in command_result.stdout
  - "running" not in command_result.stdout'
```

### ファイル名: 3-deploy\_config.yml

設定ファイルを配置します。  
※今回はhttpd.confの配置のみ行います。

### ファイル名: 4-start\_service.yml

指定したサービスを起動します。

### ファイル名: 5-check\_service.yml

サービスが起動したことを確認します。

# 1.2 必要なファイルの作成 (3/3)

## httpd.confの作成

本シナリオでは「ファイル管理機能」を利用した設定ファイルの配置を行います。

以下のテキストファイルを作成して下さい。

ファイル名: httpd\_config.txt

```
# ----- ITA Legacy Practice -----
ServerRoot "/etc/httpd"
Listen 80
Include conf.modules.d/*.conf
User apache
Group apache
ServerAdmin root@localhost
ServerName www.example.com:80

<Directory />
    AllowOverride none
    Require all denied
</Directory>
    DocumentRoot "/var/www/html"
<Directory "/var/www">
    AllowOverride None
    Require all granted
</Directory>
<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```



```
<Files ".ht*">
    Require all denied
</Files>
IncludeOptional conf.d/*.conf
ErrorLog "logs/error_log"
LogLevel warn
<IfModule log_config_module>
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common
<IfModule logio_module>
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" \"%I %O\" combinedio
</IfModule>
CustomLog logs/access_log combined
</IfModule>
<IfModule alias_module>
    ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
</IfModule>
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Require all granted
</Directory>
<IfModule mime_module>
    TypesConfig /etc/mime.types
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz
    AddType text/html .shtml
    AddOutputFilter INCLUDES .shtml
</IfModule>
    AddDefaultCharset UTF-8
<IfModule mime_magic_module>
    MIMEMagicFile conf/magic
</IfModule>
    EnableSendfile on
IncludeOptional conf.d/*.conf
```



# 1.3 Movementの設定 (1/4)

## Movementを作成する

先のplaybookを関連付けるMovementを登録しましょう。

### メニュー : Ansible-Legacy > Movement一覧

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

登録

MovementID	Movement名*	遅延タイマー	ホスト指定形式*	Ansible利用情報	
自動入力	<input type="text"/>	<input type="text"/>	<input type="button" value="▼"/>	WinRM接続	ヘッダーセクション
				<input type="text"/>	

※\*は必須項目です。

[戻る](#) [登録](#)

Movement名	ホスト指定形式
Install_Packages	IP
Open_Ports	IP
Start_Service	IP

# 1.3 Movementの設定 (2/4)

## playbookを登録する

作成したplaybookをITAに登録しましょう。

メニュー : **Ansible-Legacy > プレイブック素材集**

- ① 登録 > 登録開始 を押下する。
- ② [参照] からプレイブックを選択し、「事前アップロード」を行う。
- ③ 各項目へ下表のように入力し、「登録」を押下する。

登録

素材ID	プレイブック素材名*	プレイブック素材*	備考
自動入力	<input type="text"/>	<input type="button" value="参照..."/>	
<input type="button" value="事前アップロード"/>			
アップロード状況:			

※\*は必須項目です。

プレイブック素材名	プレイブック素材
yum_install	1-yum_install.yml
open_ports	2-open_ports.yml
deploy_config	3-deploy_config.yml
start_service	4-start_service.yml
check_service_state	5-check_service.yml

# 1.3 Movementの設定 (3/4)

## 素材ファイルを登録する

用意したhttpd設定ファイルを、ITAに登録しましょう。

### メニュー : Ansible共通 > ファイル管理

- ① 登録 > 登録開始 を押下する。
- ② [参照] から素材ファイルを選択し、「事前アップロード」を行う。
- ③ 各項目へ下表のように入力し、「登録」を押下する。

登録

素材ID	ファイル埋込変数名*	ファイル素材*	備考	最終更新日時	最終更新者
自動入力		<input type="button" value="参照..."/> 事前アップロード		自動入力	自動入力
アップロード状況:					

※\*は必須項目です。

ファイル埋込変数名	ファイル素材
CPF_httpd_conf	httpd_config.txt

# 1.3 Movementの設定 (4/4)

## Movementにplaybookを登録する

作成したMovementとプレイブック素材を関連付けましょう。

作業内容を分割し、分かりやすい名前をつけることで再利用が容易になります。

### メニュー: Ansible-Legacy > Movement詳細

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

登録

紐付項目番	Movement*	プレイブック素材*	インクルード順序	備考
自動入力	<input type="button" value="▼"/>	<input type="button" value="▼"/>	<input type="text"/>	

※\*は必須項目です。

戻る      登録

Movement	プレイブック素材	インクルード順序
Install Packages	yum_install	1
Open Ports	open_ports	1
Start Service	deploy_config	1
Start Service	start_service	2
Start Service	check_service_state	3

セットで実行するplaybookとして、「Start Service」には3つをまとめて関連付けています。

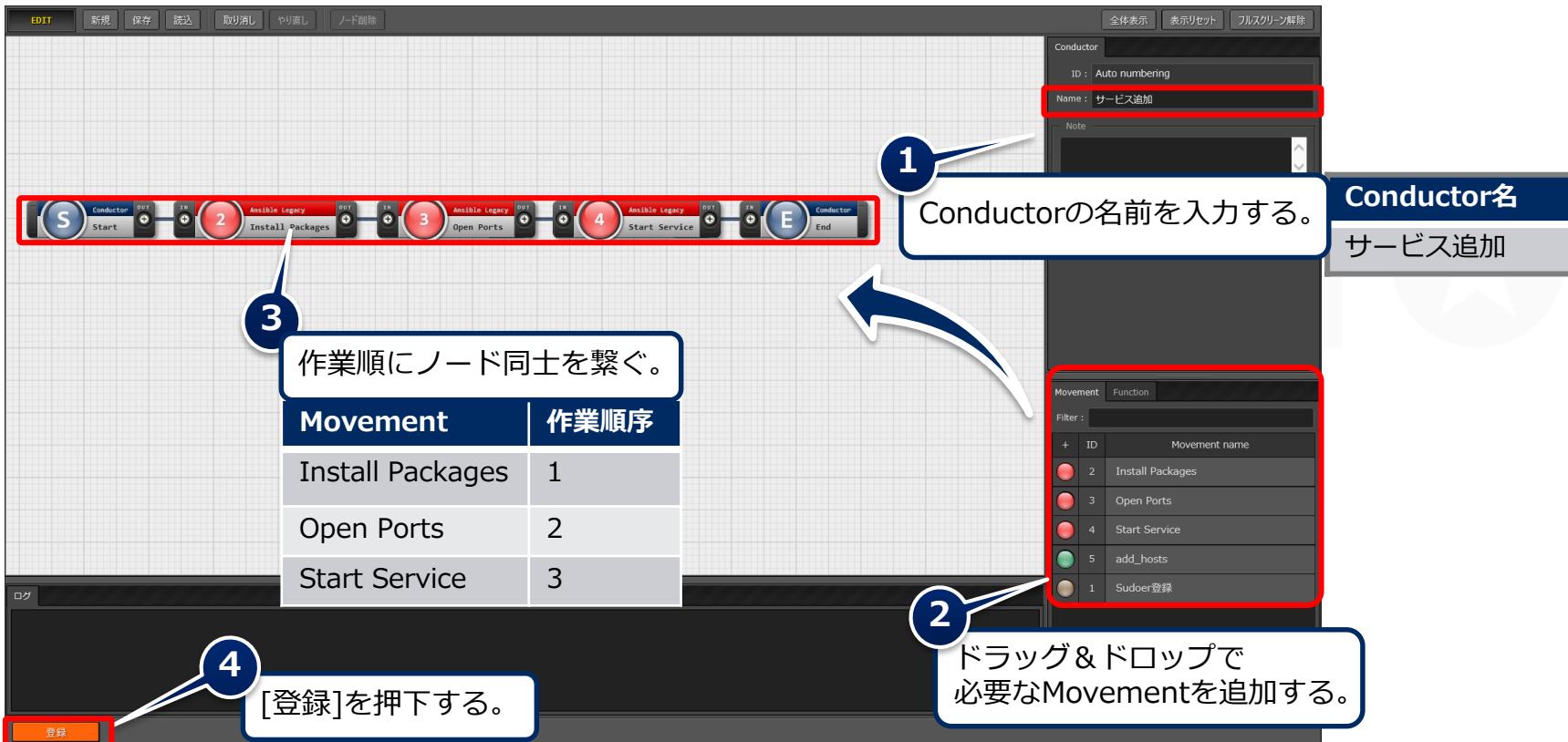
Point

# 1.4 Conductorの作成

## Conductorを作成する

定義したMovementをまとめたConductorを作成しましょう。

メニュー: Conductor > Conductorクラス編集



# 1.5 オペレーションの登録

## ■ オペレーションを新規登録する

オペレーションを作成し、Movementとホストを関連付けましょう。

※オペレーションとは、作業全体を示すITAシステム内で使用する**作業名称**です。

メニュー：基本コンソール > 投入オペレーション一覧

- ① 登録 > 登録開始 を押下する。
- ② 各項目へ下表のように入力し、[登録]を押下する。

登録

No.	オペレーションID	オペレーション名*	実施予定日時*	備考
自動入力	自動入力	<input type="text"/>	<input type="text"/>	<input type="text"/>

\*は必須項目です。

[戻る](#) [登録](#)

オペレーション名	実施予定日時
Install Apache	(任意でご入力下さい)
Install Tomcat	(任意でご入力下さい)

※ 「実施予定日時」は管理用の項目です。自動的に処理が実行されるわけではありません。

## 1.6 機器一覧への登録

### 機器一覧にホストを登録する

作業の実行対象となるホストをITAに登録しましょう。

メニュー：基本コンソール > 機器一覧

- ① 登録 > 登録開始 を押下する。
  - ② 各項目で下表のように選択または入力し、[登録]を押下する。

管理システム項目番号	HW機器種別	ホスト名*	IPアドレス*	EtherWakeOnLan		ログインユーザID	ログインパスワード	
				MACアドレス	ネットワークデバイス名		管理	ログインパスワード
自動入力	<input type="button" value="▼"/>	<input type="text"/>	<input type="button" value="▼"/>	<input type="text"/>				

※\*は必須項目です。

項目	入力内容
HW機器種別	SV
ホスト名	(任意の値)
IPアドレス	(任意の値)
ログインユーザID	(任意の値)
管理	●
ログインパスワード	(任意の値)
認証方式	ssh

# 1.7 パラメータシートの作成 (1/2)

## メニューを作成する

パラメーターシートを作成し、  
ターゲットホストに適用するパラメータを管理しましょう。

メニュー：メニュー作成 > メニュー定義/作成

メニュー作成情報

基本情報：

項目	自動入力
メニュー名*	Legacy実践
作成対象	パラメータシート(ホスト/オペレーション)
表示順序*	1

ホストブループリント： 利用する

縦メニュー利用 [?](#)： 利用する

最終更新日時：自動入力

最終更新者：自動入力

対象メニューグループ

入力用*	入力用
代入値自動登録用*	代入値自動登録用
参照用*	参照用

対象メニュー

アクセス許可ロール

ロール：

アクセス許可ロール選択

1

下表のように入力する。(次項へ)

項目名	入力内容
メニュー名	Legacy実践
作成対象	パラメータシート (ホスト/オペレーション含む)
表示順序	1

# 1.7 パラメータシートの作成 (2/2)

## パラメータシートの項目名を定義する

前項に続き、シートの項目を定義していきましょう。

メニュー： メニュー作成 > メニュー定義/作成

The screenshot shows the 'Parameter Sheet Item Definition' dialog. The top part is a grid for defining items, and the bottom part is a preview of the parameter sheet.

**Top Grid (Item Definition):**

項目名	入力方式	最大バイト数
package_name	文字列(單一行)	32
package_name_sub	文字列(單一行)	32
port_number	文字列(單一行)	32
service_name	文字列(單一行)	32

**Bottom Preview:**

一覧(プレビュー)

No	ホスト名	オペレーション名	基準日時	実施予定日	最終実行日
1	192.168.0.1	オペレーション	2020/01/01 00:00	2020/01/01 00:00	
2	192.168.0.1	オペレーション	2020/01/01 00:00	2020/01/01 00:00	
3	192.168.0.1	オペレーション	2020/01/01 00:00	2020/01/01 00:00	

**Buttons:**

- 1: '項目' button in the top-left of the dialog.
- 2: A blue circle with '2' pointing to the top-right corner of the dialog.
- 3: A red box and blue circle with '3' pointing to the '作成' (Create) button at the bottom of the preview window.

項目名	入力方式	最大バイト数
package_name	文字列(單一行)	32
package_name_sub	文字列(單一行)	32
port_number	文字列(單一行)	32
service_name	文字列(單一行)	32

# 1.8 データの登録

## パラメータシートにデータを登録する

前項の操作でパラメータシートが作成されました。

作成したメニューに移動し、データを入力していきましょう。

### メニュー：入力用 > Legacy実践(作成したメニュー)

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

No	ホスト名*	オペレーション*	パラメータ			
自動入力			package_name	package_name[2]	port_number	service_name

ホスト名	オペレーション	package_name	package_name_sub	port_number	service_name
(対象のホスト)	Apache Install	httpd	(空欄)	80/tcp	httpd
(対象のホスト)	Tomcat Install	tomcat	tomcat-webapps	8080/tcp	tomcat

# 1.9 代入値自動登録設定

## 代入値自動登録設定を行う

パラメータシートの入力が終わったところで、各項目と変数を関連付けていきます。

### メニュー: Ansible-Legacy > 代入値自動登録設定

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

登録

項目番号	パラメータシート		登録方式*	IaC変数			
	メニュー/グループ: メニュー	項目		Key変数		Value変数	
	変数名	代入順序		変数名	代入順序		
自動入力	<input type="text"/>	メニューを選択して下さい	<input type="text"/>	Movementを選択して下さい	<input type="text"/>	Movementを選択して下さい	<input type="text"/>

\* \*は必須項目です。

メニュー	項目	登録方式	Movement	Value変数 変数名	代入順序
Legacy実践	package_name	Value型	Install Packages	VAR_package_name	1
Legacy実践	package_name_sub	Value型	Install Packages	VAR_package_name	2
Legacy実践	port_number	Value型	Open Ports	VAR_port_number	(空欄)
Legacy実践	service_name	Value型	Start Service	VAR_service_name	(空欄)

# 1.10 代入値・対象ホストの確認

## 代入値と作業対象ホストを確認する

代入値自動登録により指定された値と対象ホストを確認しましょう。

メニュー: **Ansible-Legacy > 作業対象ホスト & Ansible-Legacy > 代入値管理**

- ① [フィルタ]を押下する
- ② 「legacy代入値自動登録設定プロシージャ」によって正しい値が指定されていることを確認する。

### 作業対象ホスト

一覧/更新

更新	廃止	項目番号	オペレーション	Movement	ホスト	備考	最終更新日時	最終更新者
更新	廃止	1	2:Apache Install	2:Install Packages	1:ita-test		2020/08/28 19:08:3	legacy代入値自動登録設定プロシージャ
更新	廃止	2	2:Apache Install	3:Open Ports	1:ita-test		2020/08/28 19:07:4	legacy代入値自動登録設定プロシージャ
更新	廃止	3	2:Apache Install	4:Start Service	1:ita-test		2020/08/28 19:08:4	legacy代入値自動登録設定プロシージャ
更新	廃止	4	3:Tomcat Install	2:Install Packages	1:ita-test		2020/08/28 19:08:4	legacy代入値自動登録設定プロシージャ
更新	廃止	5	3:Tomcat Install	3:Open Ports	1:ita-test		2020/08/28 19:08:5	legacy代入値自動登録設定プロシージャ
更新	廃止	6	3:Tomcat Install	4:Start Service	1:ita-test		2020/08/28 19:08:5	legacy代入値自動登録設定プロシージャ

フィルタ結果件数: 6

### 代入値管理

一覧/更新

更新	廃止	項目番号	オペレーション	Movement	ホスト	変数名	具体値	代入順序	備考	最終更新日時	最終更新者
更新	廃止	4	2:Apache Install	2:Install Packages	1:ita-test	1:VAR_package_name	httpd	1		2020/08/27 10:10:3	legacy代入値自動登録設定プロシージャ
更新	廃止	2	2:Apache Install	3:Open Ports	1:ita-test	2:VAR_port_number	80/tcp			2020/08/28 19:03:3	legacy代入値自動登録設定プロシージャ
更新	廃止	3	2:Apache Install	4:Start Service	1:ita-test	3:VAR_service_name	httpd			2020/08/28 19:03:5	legacy代入値自動登録設定プロシージャ
更新	廃止	5	3:Tomcat Install	2:Install Packages	1:ita-test	1:VAR_package_name	tomcat	1		2020/08/27 10:10:3	legacy代入値自動登録設定プロシージャ
更新	廃止	6	3:Tomcat Install	2:Install Packages	1:ita-test	1:VAR_package_name	tomcat-webapps	2		2020/08/27 10:10:5	legacy代入値自動登録設定プロシージャ
更新	廃止	7	3:Tomcat Install	3:Open Ports	1:ita-test	2:VAR_port_number	8080/tcp			2020/08/27 10:12:6	legacy代入値自動登録設定プロシージャ
更新	廃止	8	3:Tomcat Install	4:Start Service	1:ita-test	3:VAR_service_name	tomcat			2020/08/27 10:12:3	legacy代入値自動登録設定プロシージャ

フィルタ結果件数: 7

# 1.11 作業の実行 (1/3)

## Conductorを実行する

前項までの操作で、Conductorの作成と代入値の登録が終了しました。  
最後にConductorを実行し、結果を対象ホストで確認してください。

### メニュー： Conductor > Conductor作業実行

The screenshot shows the Conductor interface with the following steps:

- 1** In the top-level "Conductor[一覧]" list, a callout points to the first item "1 サービス追加" with the text "実行するConductorを選択する。".
- 2** In the "オペレーション[一覧]" list, a callout points to the second item "2 Apache Install" with the text "オペレーションを選択する。".
- 3** At the bottom of the screen, a callout points to the orange "[実行]" button with the text "画面下部より、[実行]を押下する。".

**Tips**

実行後、自動で【Conductor作業確認】へ画面遷移します。

# 1.11 作業の実行 (2/3)

## Conductorの実行結果を確認する

作業確認画面では、全体およびノードごとの実行結果を確認できます。  
投入したMovementを選択すると、**詳細結果へのリンク**を表示できます。

メニュー： Conductor > Conductor作業確認

The screenshot shows the 'Conductor Task Confirmation' interface. On the left, a workflow diagram is displayed with nodes: 'Conductor Start' (S), 'Ansible Legacy Install\_Packages' (DONE), 'Ansible Legacy Open\_Ports' (DONE), 'Ansible Legacy Start\_Service' (DONE), and 'Conductor End'. A callout bubble labeled '1' points to the 'Ansible Legacy Install\_Packages' node with the text '確認したいノードを選択する。' (Select the node you want to check). On the right, a detailed task view is shown for the selected node. It includes fields for Node type (movement), Node instance ID (83), Node name (node-3), Status (正常終了), Start time (2020/12/04 12:32:38), End time (2020/12/04 12:33:09), and Operation status (作業状態確認). A red box highlights this section. A callout bubble labeled '2' points to the 'Operation status' field with the text 'リンクや作業の情報が表示される。' (Links and job information are displayed). Below this, there are sections for 'Specified individually operation' (Operation ID: 2, Operation name: apa) and 'Operation' (Operation ID: 2, Operation name: apa).

## 1.11 作業の実行 (3/3)

Movement毎の詳細結果を確認する

リンクをクリックすると画面が遷移し、**実行ステータス**や**ログ**が表示されます。  
投入データや出力データを確認することもできます。

```
進行状況(実行ログ)

フィルタ : [ ] 該当のみ表示

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : httpd-2.4.6-93.el7.centos.x86_64           1/1
  Verifying  : httpd-2.4.6-93.el7.centos.x86_64           1/1

Installed:
  httpd.x86_64 0:2.4.6-93.el7.centos

Complete!
"
}
}
META: ran handlers
META: ran handlers

PLAY RECAP ****
ita-test                  : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

## Tips

投入データや結果データをまとめたzipファイルをダウンロードできます。

## Tips

以下のURLにて、Apache及びTomcatのインストールを確認できます。

Apache- http://(ホストのIPアドレス):80  
Tomcat- http://(ホストのIPアドレス):8080

# 第2章 Ansible-LegacyRole編

# 2.1 作業環境とシナリオ

## 作業環境

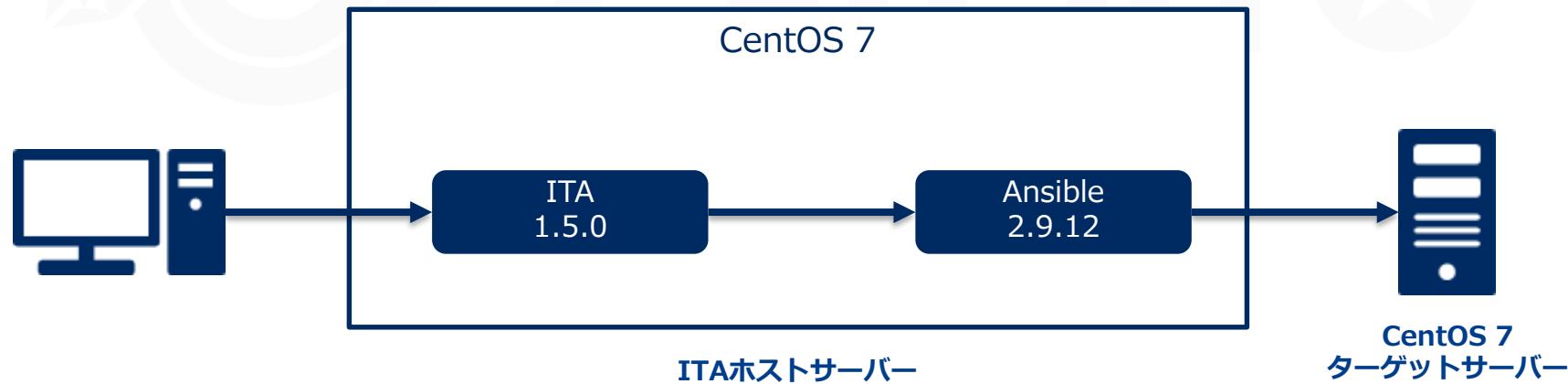
本章で使用する作業環境は以下の通りです。(第1章Ansible-Legacy編と共に)  
ITAホストサーバーとは別に、ターゲットとなるサーバーをご用意ください。

### ITAホストサーバー

- CentOS 7 (※1)
- ITA 1.6.0
- Ansible 2.9.12

### ターゲット

- CentOS 7 (※2)



※1 今回はホストサーバーとしてCentOS7を利用致しますが、ITAはRHEL7系およびRHEL8系のOSで導入いただけます。

※2 Ansibleの動作対象となれるOSであれば、問題なく利用いただけます。

## 2.1 作業環境とシナリオ

### シナリオ

LegacyRoleの最たる特長は、**ロールパッケージの登録・利用が可能な点**です。

本章ではAnsible GalaxyからダウンロードしたRoleをITAに登録・実行する作業を行います。

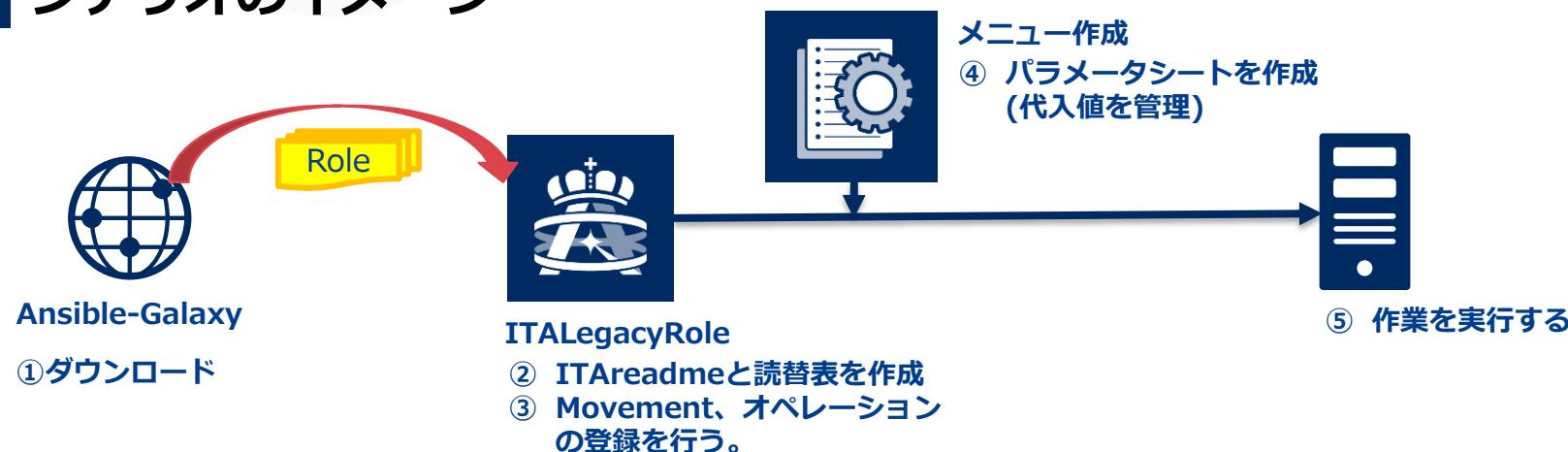
まず、以下のURLからRoleを取得してください。



<https://galaxy.ansible.com/weareinteractive/sudo>

/etc/sudoers.d配下にファイルを追加するロールパッケージです。

### シナリオのイメージ



## 2.2 ロールパッケージの準備(1/4)

### パッケージ準備のまえに

今回使用するRoleのdefaults/main.ymlを見てみましょう(下図参照)。実行前に変更するべき箇所が2点あります。

このような場合、ITAreadmeと読替表を作成することで、パッケージ中のファイルを変更することなく**変数定義に必要な変更を加える**ことができます。

```
---  
# sudo_defaults:  
#(中略) ~~~~  
# package name (version)  
sudo_package: sudo  
# list of username or %groupname  
sudo_users: []  
# list of username or %groupname and their defaults  
sudo_defaults: []  
# default sudoers file  
sudo_sudoers_file: ansible  
# path of the sudoers.d directory  
sudo_sudoers_d_path: /etc/sudoers.d  
# delete other files in `sudo_sudoers_d_path`  
purge_other_sudoers_files: no  
  - defaults: env_reset  
# - name: user1  
#  defaults: requiretty  
# sudo_users:  
#  - name: '%group1'  
#  - name: 'bar'  
#  nopasswd: yes  
~~~
```

正しいマッピング例はコメントアウトされ、空の配列だけが定義されています。

1

➤ **ITAreadme**を用いて構造を変更し、  
**読替表**を用いてITAでの編集を可能にしましょう。

2

実行時に生成するファイル名は  
利用者が変更したい箇所です。

➤ **読替表**を用いてITAでの編集を可能にしましょう。

## 2.2 ロールパッケージの準備(2/4)

### ITAreadmeの記述

ita\_readmeは、変数の定義を追加・変更するための設定ファイルです。  
※ITAreadmeについて、詳細は[マニュアル](#)をご参照ください。

ファイル名: ita\_readme\_ansible-sudo-master.yml

```
sudo_users:  
  - name:
```

### イメージ図

正しいマッピングに変更する。

```
defaults/main.yml  
sudo_package: sudo  
sudo_users: []
```

1

実際に利用される変数

```
sudo_package: sudo  
sudo_users:  
  - name: example_name
```

正しいマッピングを記入する。

```
ITAreadme  
sudo_users:  
  - name:
```

2



代入値管理

変数名	メンバー変数名	具体値
LCA_sudo_users	[0].name	example_name

読み替表については次項で説明します。

## 2.2 ロールパッケージの準備(3/4)

### 読替表の記述

読替表は、defaults変数定義ファイルまたはITAreadmeに定義されている「VAR\_...」以外の変数に対して、ITAで具体値を設定できるようにするファイルです。  
※読替表について、詳細は[マニュアル](#)をご参照ください。

ファイル名: ita\_translation-table\_ansible-sudo-master.txt

LCA\_sudo\_users: sudo\_users

LCA\_sudo\_sudoers\_file: sudo\_sudoers\_file

### イメージ図

`defaults/main.yml`  
sudo\_users: []  
~~~~~ (中略) ~~~~  
`sudo_sudoers_file: ansible`

`ITAreadme`  
`sudo_users:`  
- name:

実際に利用される変数

sudo\_users:  
- name: example\_name  
sudo\_sudoers\_file: example\_sudoers

### 読替表

LCA\_sudo\_users: `sudo_users`  
LCA\_sudo\_sudoers\_file: `sudo_sudoers_file`



#### 代入値管理

| 変数名                   | メンバー変数名<br>(ITAreadmeで<br>追加) | 具体値             |
|-----------------------|-------------------------------|-----------------|
| LCA_sudo_users        | [0].name                      | example_name    |
| LCA_sudo_sudoers_file |                               | example_sudoers |

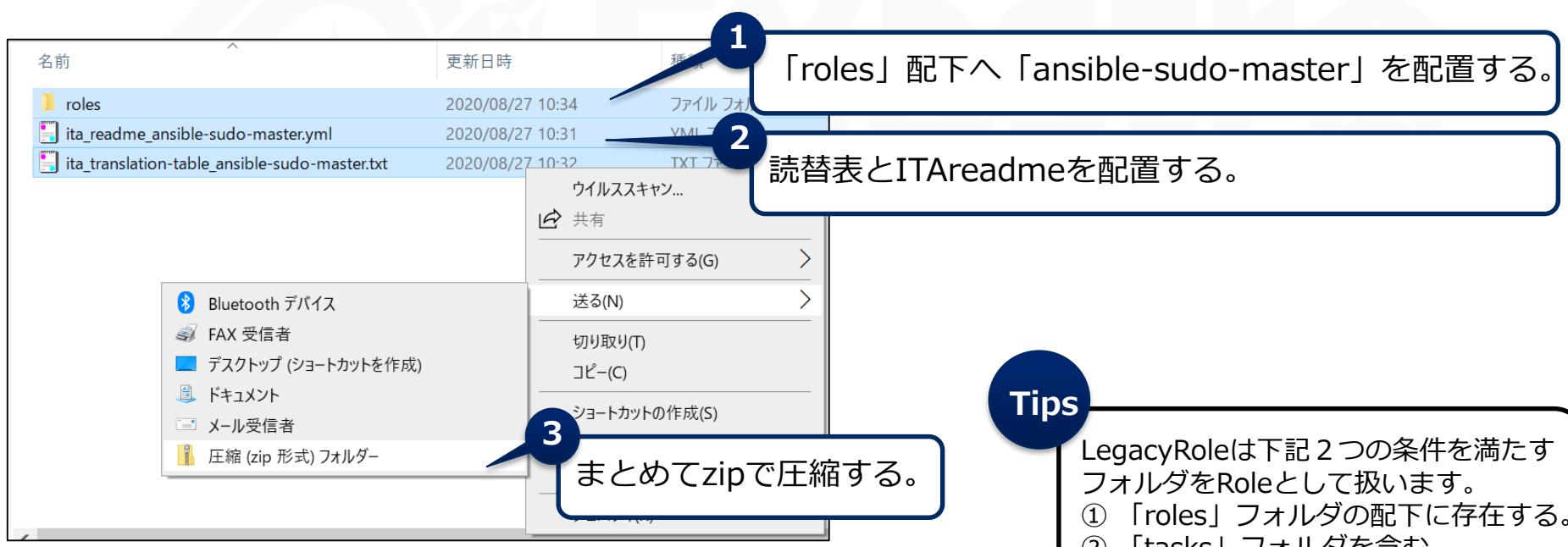
## 2.2 ロールパッケージの準備(4/4)

### 必要なファイルをzipにまとめる

これまでの作成物をzipファイルにまとめ、ITAに登録しましょう。

下記の構造通りのzipファイルを作成して下さい。

- ① 「roles」 フォルダを作成し、ダウンロードしたRoleを入れる。
- ② 「roles」 フォルダに並べて読替表とITAreadmeを配置する
- ③ 「roles」 フォルダ、読替表、 ITAreadmeをまとめてzipで圧縮する。



## 2.3 Movementの設定 (1/3)

### Movementを作成する

Roleを関連付けるMovementを1つ作成しましょう。

#### メニュー : Ansible-LegacyRole > Movement一覧

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

登録

| MovementID | Movement名*           | 遅延タイマー               | ホスト指定形式*                         | Ansible利用情報          |           |
|------------|----------------------|----------------------|----------------------------------|----------------------|-----------|
| 自動入力       | <input type="text"/> | <input type="text"/> | <input type="button" value="▼"/> | WinRM接続              | ヘッダーセクション |
|            |                      |                      |                                  | <input type="text"/> |           |

※\*は必須項目です。

| 項目名       | 入力内容     |
|-----------|----------|
| Movement名 | Sudoer登録 |
| ホスト指定形式   | IP       |

## 2.3 Movementの設定 (2/3)

### ロールパッケージの登録

作成したロールパッケージファイルを登録しましょう。

メニュー: **Ansible-LegacyRole > ロールパッケージ管理**

- ① 登録 > 登録開始 を押下する。
- ② [参照] から作成した**zip**ファイルを選択し、「事前アップロード」を行う。
- ③ 各項目へ下表のように入力し、[登録]を押下する

登録

| 項目番号      | ロールパッケージ名*           | ロールパッケージファイル(ZIP形式)*                                                         |                      |
|-----------|----------------------|------------------------------------------------------------------------------|----------------------|
| 自動入力      | <input type="text"/> | <input type="button" value="事前アップロード"/> <input type="button" value="参照..."/> | <input type="text"/> |
| アップロード状況: |                      |                                                                              |                      |

ロールパッケージ名 | sudo-master

## 2.3 Movementの設定 (3/3)

### Movementにロール名を登録する

Movementに個別のロール名を登録しましょう。

#### メニュー: Ansible-LegacyRole > Movement詳細

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

登録

| 紐付項目番号 | Movement*            | ロールパッケージ名            | ロール名                 | インクルード順序             |
|--------|----------------------|----------------------|----------------------|----------------------|
| 自動入力   | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |

\*<sup>\*</sup>は必須項目です。

[戻る](#) [登録](#)

| Movement名 | ロールパッケージ名   | ロール名                | インクルード順序 |
|-----------|-------------|---------------------|----------|
| Sudoer登録  | sudo-master | ansible-sudo-master | 1        |

※同一Movement内に複数のロールパッケージを登録しないでください。実行時に想定外エラーとなります。

## 2.4 オペレーションの設定

### オペレーションの新規登録

オペレーションを作成し、Movementとホストを関連付けましょう。

メニュー：基本コンソール > 投入オペレーション一覧

- ① 登録 > 登録開始 を押下する。
- ② 各項目へ下表のように入力し、[登録]を押下する。

登録

| No.  | オペレーションID | オペレーション名* | 実施予定日時* | 備考 |
|------|-----------|-----------|---------|----|
| 自動入力 | 自動入力      |           |         |    |
|      |           |           |         |    |
|      |           |           |         |    |

\*は必須項目です。

[戻る](#) [登録](#)

| 項目名      | 入力内容                |
|----------|---------------------|
| オペレーション名 | LegacyRole_Practice |
| 実施予定日時   | (任意の値を入力下さい。)       |

※ 「実施予定日時」は管理用の項目です。自動的に処理が実行されるわけではありません。

## 2.5 機器一覧への登録

### 機器一覧へホストを登録する

作業の実行対象となるホストをITAに登録しましょう。

#### メニュー：基本コンソール > 機器一覧

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

登録

|            |        |       |         |                |             |           |              |
|------------|--------|-------|---------|----------------|-------------|-----------|--------------|
| 管理システム項目番号 | HW機器種別 | ホスト名* | IPアドレス* | EtherWakeOnLan |             | ログインパスワード |              |
| 自動入力       |        |       |         | MACアドレス        | ネットワークデバイス名 | ログインユーザID | 管理 ログインパスワード |
|            |        |       |         |                |             |           |              |

※\*(は必須項目です。)

戻る 登録

| 項目        | 入力内容   |
|-----------|--------|
| HW機器種別    | SV     |
| ホスト名      | (任意の値) |
| IPアドレス    | (任意の値) |
| ログインユーザID | (任意の値) |
| 管理        | ●      |
| ログインパスワード | (任意の値) |
| 認証方式      | ssh    |

# 2.6 パラメータシート作成(1/2)

## メニューを作成する

パラメータシートを作成し、  
ターゲットホストに適用するパラメータを管理しましょう。

メニュー：メニュー作成 > メニュー定義/作成

メニュー作成情報

基本情報：

|        |                       |
|--------|-----------------------|
| 項目     | 自動入力                  |
| メニュー名* | LegacyRole実践          |
| 作成対象   | パラメータシート(ホスト/オペレーション) |
| 表示順序*  | 2                     |

ホストグループ利用： ■ 利用する

縦メニュー利用 (?:) ■ 利用する

最終更新日時： 自動入力

最終更新者： 自動入力

対象メニューグループ

|           |          |
|-----------|----------|
| 入力用*      | 入力用      |
| 代入値自動登録用* | 代入値自動登録用 |
| 参照用*      | 参照用      |

対象メニュー

アクセス許可ロール

|      |
|------|
| ロール： |
|------|

アクセス許可ロール選択

1

下表のように入力する。(次項へ)

| 項目名   | 入力内容                        |
|-------|-----------------------------|
| メニュー名 | LegacyRole実践                |
| 作成対象  | パラメータシート<br>(ホスト/オペレーション含む) |
| 表示順序  | 2                           |

## 2.6 パラメータシート作成(2/2)

### パラメータシートの項目名を定義する

前項に続き、シートの項目を定義していきましょう。

メニュー： メニュー作成 > メニュー定義/作成

The screenshot shows the 'Parameter Sheet Definition' screen. On the left, there are two columns of fields for defining parameters:

| 項目名             | 入力方式     | 最大バイト数 |
|-----------------|----------|--------|
| sudoer_name     | 文字列(單一行) | 32     |
| sudoer_filename | 文字列(單一行) | 32     |

On the right, a callout box points to a table titled '下表のように項目の内容を定義する。' (Define item content as shown in the table below):

| 項目名             | 入力方式     | 最大バイト数 |
|-----------------|----------|--------|
| sudoer_name     | 文字列(單一行) | 32     |
| sudoer_filename | 文字列(單一行) | 32     |

At the bottom, a preview window titled '一覧(プレビュー)' (List Preview) shows a table with three rows of data:

| No. | ホスト名        | オペレーション名 | 基準日時             | 実施予定期            | 最終実行日 |
|-----|-------------|----------|------------------|------------------|-------|
| 1   | 192.168.0.1 | オペレーション  | 2020/01/01 00:00 | 2020/01/01 00:00 |       |
| 2   | 192.168.0.1 | オペ       |                  |                  |       |
| 3   | 192.168.0.1 | オペ       |                  |                  |       |

Three numbered callouts provide instructions:

- 1 押下して新しい項目を追加する。 (Press to add a new item.)
- 2 下表のように項目の内容を定義する。 (Define item content as shown in the table below.)
- 3 押下してメニュー作成を実行する。 (Press to execute menu creation.)

## 2.7 データの登録

### パラメータシートにデータを登録する

前項までの操作で、パラメータシートが作成できました。  
ターゲットホストの設定に使用するデータを登録しましょう。

#### メニュー: 入力用> LegacyRole実践(作成したメニュー)

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

| No   | ホスト名* | オペレーション | パラメータ                                                                 |
|------|-------|---------|-----------------------------------------------------------------------|
| 自動入力 |       |         | <input type="text"/> sudoer_name <input type="text"/> sudoer_filename |
|      |       |         |                                                                       |

#### 入力例

| ホスト名     | オペレーション             | sudoer_name  | sudoer_filename |
|----------|---------------------|--------------|-----------------|
| (対象のホスト) | LegacyRole_Practice | example_name | example_sudoers |

# 2.8 代入値自動登録設定

## 代入値自動登録設定を行う

パラメータシートの入力が終わったところで、各項目と変数を関連付けていきます。

### メニュー: Ansible-LegacyRole > 代入値自動登録設定

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

登録

| 項目番号 | パラメータシート     |                  | 登録方式* | Iac変数            |       |                  |         |      |
|------|--------------|------------------|-------|------------------|-------|------------------|---------|------|
|      | メニュー         | 項目               |       | Movement         | Key変数 |                  | Value変数 |      |
|      |              |                  |       |                  | 変数名   | 代入順序             | 変数名     | 代入順序 |
| 自動入力 | メニューを選択して下さい | Movementを選択して下さい |       | Movementを選択して下さい |       | Movementを選択して下さい |         |      |

\*は必須項目です。

| メニュー グループ :<br>メニュー | 項目              | 登録 方式  | Movement | Value変数<br>変数名        | Value変数<br>メンバー変数名 |
|---------------------|-----------------|--------|----------|-----------------------|--------------------|
| LegacyRole実践用       | sudoer_name     | Value型 | Sudoer登録 | LCA_sudo_users        | [0].name           |
| LegacyRole実践用       | sudoer_filename | Value型 | Sudoer登録 | LCA_sudo_sudoers_file | (空欄)               |

# 2.9 代入値・対象ホストの確認

## 代入値と作業対象ホストを確認する

代入値自動登録により指定された値と対象ホストを確認しましょう。

メニュー: **Ansible-LegacyRole > 作業対象ホスト & Ansible-LegacyRole > 代入値管理**

- ① [フィルタ]を押下する
- ② 「legacyRole代入値自動登録設定プロシージャ」によって正しい値が指定されていることを確認する。

### 作業対象ホスト

| 一覧/更新 |    |    |                       |            |            |    |                     |                           |
|-------|----|----|-----------------------|------------|------------|----|---------------------|---------------------------|
| 更新    | 廃止 | 項目 | オペレーション               | Movement   | ホスト        | 備考 | 最終更新日時              | 最終更新者                     |
| 更新    | 廃止 | 1  | 1:LegacyRole_Practice | 1:Sudoer登録 | 1:ita-test |    | 2020/08/28 16:04:58 | legacyRole代入値自動登録設定プロシージャ |

フィルタ結果件数: 1

### 代入値管理

| 一覧/更新 |    |    |                       |            |            |                         |            |                 |      |    | 最終更新者               |                           |
|-------|----|----|-----------------------|------------|------------|-------------------------|------------|-----------------|------|----|---------------------|---------------------------|
| 更新    | 廃止 | 項目 | オペレーション               | Movement   | ホスト        | 変数名                     | メンバー変数名    | 具体値             | 代入順序 | 備考 | 最終更新日時              | 最終更新者                     |
| 更新    | 廃止 | 1  | 1:LegacyRole_Practice | 1:Sudoer登録 | 1:ita-test | 2:LCA_sudo_users        | 1:[0].name | example_name    |      |    | 2020/08/28 15:53:53 | legacyRole代入値自動登録設定プロシージャ |
| 更新    | 廃止 | 2  | 1:LegacyRole_Practice | 1:Sudoer登録 | 1:ita-test | 1:LCA_sudo_sudoers_file |            | example_sudoers |      |    | 2020/08/28 15:53:53 | legacyRole代入値自動登録設定プロシージャ |

フィルタ結果件数: 2

# 2.10 作業の実行 (1/2)

## Movementを直接実行する

本シナリオで作成したMovementは一つです。

Conductorの作成を省き、**直接実行**機能を使用しましょう。

### メニュー: Ansible-LegacyRole > 作業実行

The screenshot shows the 'Movement' execution interface. The left sidebar has sections like 'Menu', 'Movement一覧', 'Movement詳細', 'オペレーション確認', and '作業実行'. The main area has tabs for '説明', 'スケジューリング', 'Movement[フィルタ]', and 'Movement[一覧]'. Step 1 highlights the 'Movement[一覧]' tab. Step 2 highlights the 'オペレーション[一覧]' tab. Step 3 highlights the '実行' button.

1 実行するMovementを選択する。

2 オペレーションを選択する。

3 「実行」を押下する。

| Ansible利用情報                      |            |           |                     |        |         |         |           |            |        |    |                     |    |
|----------------------------------|------------|-----------|---------------------|--------|---------|---------|-----------|------------|--------|----|---------------------|----|
| 選択                               | MovementID | Movement名 | オーケストレータ            | 遅延タイマー | ホスト指定形式 | WinRM接続 | ヘッダーセクション | オプションパラメータ | 変数カウント | 備考 | 最終更新日時              | 操作 |
| <input checked="" type="radio"/> | 1          | Sudoer登録  | Ansible Legacy Role |        | IP      |         |           |            | 2      |    | 2020/08/21 20:02:35 |    |

| オペレーション[一覧]                      |     |           |                     |                  |                  |    |                     |         |
|----------------------------------|-----|-----------|---------------------|------------------|------------------|----|---------------------|---------|
| 選択                               | No. | オペレーションID | オペレーション名            | 実施予定日時           | 最終実行日時           | 備考 | 最終更新日時              | 最終更新者   |
| <input checked="" type="radio"/> | 1   | 1         | LegacyRole_Practice | 2020/08/21 17:20 | 2020/08/21 17:18 |    | 2020/08/21 20:03:36 | システム管理者 |

Tips

実行後、自動で【作業確認】へ画面遷移します。

# 2.10 作業の実行 (2/2)

## 実行結果を確認する

作業を実行すると画面が遷移し、**実行ステータスやログ**が表示されます。

### メニュー： Ansible-LegacyRole > 作業確認

The screenshot shows the 'Job Confirmation' screen. On the left is a vertical navigation menu with the following items:

- ☰ Menu
- メインメニュー
- Movement一覧
- ロールパッケージ管理
- Movement詳細
- 多段変数局大継返数管理
- 代入値自動登録設定
- 作業対象ホスト
- 代入値管理
- 作業実行
- 作業状態確認
- 作業管理

The main content area has two tabs: '説明' (Description) and '対象作業' (Target Job). The '対象作業' tab is selected, showing a table with job details:

| 項目                                                       | 値                         |                     |
|----------------------------------------------------------|---------------------------|---------------------|
| 作業No.                                                    | 2                         |                     |
| 実行種別                                                     | 通常                        |                     |
| ステータス                                                    | 完了                        |                     |
| 実行エンジン                                                   | Ansible Engine            |                     |
| 呼出元Symphony                                              |                           |                     |
| 呼出元Conductor                                             |                           |                     |
| 実行ユーザ                                                    | システム管理者                   |                     |
| Movement                                                 | ID                        | 1                   |
|                                                          | 名称                        | Sudoer登録            |
|                                                          | 遅延タイム(分)                  |                     |
| Ansible利用情報                                              | ホスト指定形式                   | IP                  |
|                                                          | WinRM接続                   |                     |
| オペレーション                                                  | No.                       | 1                   |
|                                                          | 名称                        | LegacyRole_Practice |
|                                                          | ID                        | 1                   |
| 作業対象ホスト                                                  |                           |                     |
| 代入値                                                      |                           |                     |
| <input type="button" value="確認"/>                        |                           |                     |
| <input type="button" value="確認"/>                        |                           |                     |
| <input type="button" value="InputData_0000000002.zip"/>  | InputData_0000000002.zip  |                     |
| <input type="button" value="ResultData_0000000002.zip"/> | ResultData_0000000002.zip |                     |
| <input type="button" value="予約日時"/>                      |                           |                     |
| <input type="button" value="開始日時"/>                      | 2020/08/22 03:27:40       |                     |
| <input type="button" value="終了日時"/>                      | 2020/08/22 03:28:16       |                     |

At the bottom right of the table, there is a button labeled '管理者に連絡' (Contact Manager).

The screenshot shows the 'Execution Status (Execution Log)' screen. It displays the execution log for the job. The log output is as follows:

```
task path: /exastro/data_relay_storage/ansible_driver/legacy/r1/0000000002/in/roles/sudoers/tasks/config.yml:12
skipping: [testserver] => {
    "changed": false,
    "skip_reason": "Conditional result was False"
}

TASK [Remove unmanaged /etc/sudoers.d files] *****
task path: /exastro/data_relay_storage/ansible_driver/legacy/r1/0000000002/in/roles/sudoers/tasks/config.yml:20
skipping: [testserver] => {
    "changed": false,
    "skip_reason": "Conditional result was False"
}
META: ran handlers
META: ran handlers

PLAY RECAP *****
testserver : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Tips

投入データや結果データをまとめたzipファイルをダウンロードできます。

Tips

結果を対象ホストで確認する場合、/etc/sudoers.d配下をご参照ください。

# 第3章 Ansible-Pioneer編



# 3.1 作業環境とシナリオ

## 作業環境

本書で使用する作業環境は以下の通りです。

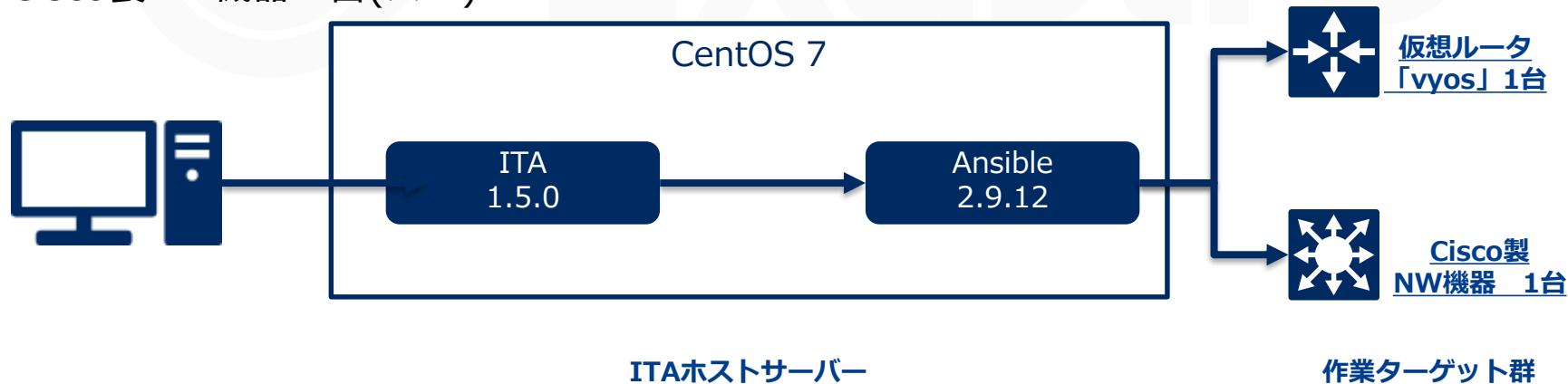
NW機器の操作を実施しますので、下記の通り環境をご用意ください。

### ITAホストサーバー

- CentOS 7 (※1)
- ITA 1.6.0
- Ansible 2.9.12

### ターゲット

- 仮想ルータ「vyos」 1台
- Cisco製NW機器 1台(※ 2 )



※1 今回はホストサーバーとしてCentOS7を利用致しますが、ITAはRHEL7系およびRHEL8系のOSで導入いただけます。

※2 本資料では各項目への入力例としてレイヤ3スイッチを採用しています。ルーターやレイヤ2スイッチを利用する場合、適宜読み替えてください。

# 3.1 作業環境とシナリオ

## シナリオ

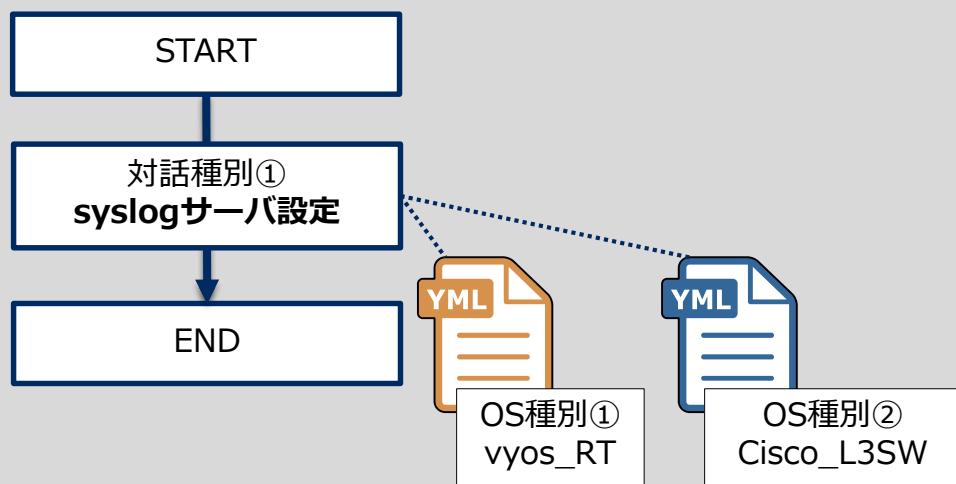
Ansible-Pioneerを用いて**ベンダの異なるNW機器**に対してログサーバの指定を行います。  
以下3つのPioneerモードの特長を体験いただけるシナリオとなっています。

- ① telnetもしくはsshの疎通さえあれば、対話ファイルの実行が可能
- ② 対話種別とOS種別を活かした、OSの差異を意識しない作業実行
- ③ 独自モジュールによる作業の繰り返しや条件分岐

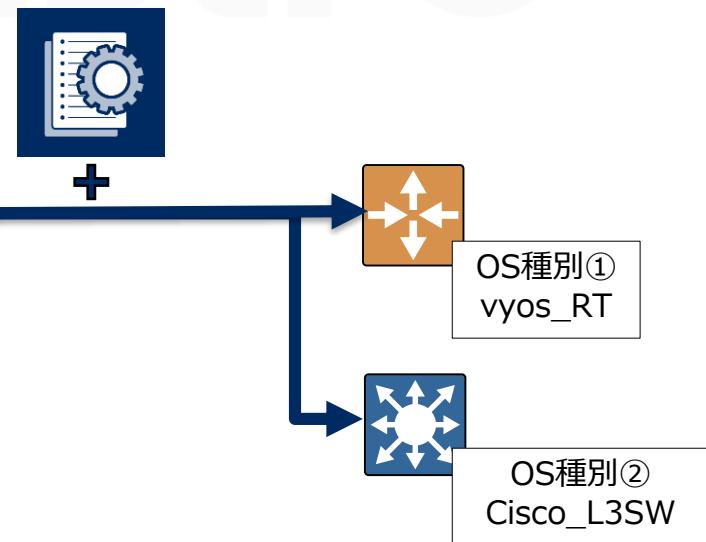
## シナリオイメージ

### Movementの作成

#### Movement①



パラメータシートを利用した  
代入値や作業対象ホストの設定



## 3.2 対話ファイルの作成(1/2)

### 対話ファイルの作成

本シナリオで使用するファイルを作成しましょう。

【注意】

文字コードは“UTF-8”を使用してください。

vyosへのログサーバ登録コマンドです。  
“with\_items”を用いて繰り返し処理を行っています。

設定の確認を行っています。  
設定情報を出力し、所定の文字列が無い場合に  
結果を“failed”とします。  
“with\_items”による繰り返しごとに条件判定  
が行われます。

```
conf:  
  timeout: 10  
  
exec_list:  
  - expect: 'password:'  
    exec: '{{ __loginpassword__ }}'  
  
  - expect: '{{ __loginuser__ }}@{{ __loginhostname__ }}'  
    exec: 'set terminal length 0'  
  
  - expect: '{{ __loginuser__ }}@{{ __loginhostname__ }}'  
    exec: 'configure'  
  
  - command: 'set system syslog host {{ item.0 }} facility all level {{ VAR_log_severity }}'  
    prompt: 'vyos@{{ __loginhostname__ }}'  
    with_items:  
      - '{{ VAR_syslog_server_ip }}'  
    when:  
      - VAR_log_severity is define  
  
  - expect: 'vyos@{{ __loginhostname__ }}'  
    exec: 'commit'  
  
  - expect: 'vyos@{{ __loginhostname__ }}'  
    exec: 'save'  
  
  - expect: 'vyos@{{ __loginhostname__ }}'  
    exec: 'exit'  
  
  - command: 'show configuration'  
    prompt: 'vyos@{{ __loginhostname__ }}'  
    with_items:  
      - '{{ VAR_syslog_server_ip }}'  
    failed_when:  
      - stdout match(host *{{ item.0 }})
```

ファイル名: vyos\_set\_syslog\_server.yml

## 3.2 対話ファイルの作成(2/2)

### 対話ファイルを用意する

同様に、右記のファイルも作成しましょう。

ログについての設定を表示し、  
"register"で標準出力の内容を格納しています。

Cisco IOSへのログサーバ登録コマンドです。  
"with\_items"を用いて繰り返し処理を行っています。  
"exec\_when"では、上で格納した内容に応じて処理の実行判定を行っています。

```
conf:  
  timeout: 10  
  ファイル名: ios_set_syslog_server.yml  
  
exec_list:  
  - expect: 'Username:'  
    exec: '{{ __loginuser__ }}'  
  
  - expect: 'Password:'  
    exec: '{{ __loginpassword__ }}'  
  
  - expect: '{{ __loginhostname__ }}'  
    exec: 'enable'  
  
  - expect: 'Password:'  
    exec: '{{ __loginpassword__ }}'  
  
  - expect: '{{ __loginhostname__ }}'  
    exec: 'terminal length 0'  
  
  - command: 'show logging'  
    prompt: '{{ __loginhostname__ }}'  
    register: result_stdout  
  
  - expect: '{{ __loginhostname__ }}'  
    exec: 'configure terminal'  
  
  - command: 'logging host {{ item.0 }}'  
    prompt: '{{ __loginhostname__ }}'  
    with_items:  
      - '{{ VAR_syslog_server_ip }}'  
    exec_when:  
      - result_stdout no match(Logging to {{ item.0 }})  
  
  - command: 'logging facility {{ VAR_log_facility }}'  
    prompt: '{{ __loginhostname__ }}'  
    when:  
      - VAR_log_facility is define  
  
  - command: 'logging trap {{ VAR_log_severity }}'  
    prompt: '{{ __loginhostname__ }}'  
    when:  
      - VAR_log_severity is define
```

### 3.3 OS種別の作成

#### 「OS種別」を作成する

PioneerはターゲットホストのOSに応じて、実際に投下されるコードを選択することができます。

まずは「OS種別」をITAに登録しましょう。

メニュー: **Ansible-Pioneer > OS種別マスター**

- ① 登録 > 登録開始 を押下する。
- ② 各項目へ下表のように入力し、[登録]を押下する

登録

| OS種別ID | OS種別名* | 機器種別  |
|--------|--------|-------|
| 自動入力   | SV     | NW ST |

※\*は必須項目です。

| OS種別名      | 機器種別 / NW |
|------------|-----------|
| vyos_RT    | ●         |
| Cisco_L3SW | ●         |

## 3.4 Movementの設定 (1/4)

### Movementを作成する

MovementはITAにおける作業の最小単位です。

Movementを作成し、**対話種別**と紐づけていきましょう。

#### メニュー : Ansible-Pioneer > Movement一覧

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

登録

| MovementID | Movement名*           | 遅延タイマー               | ホスト指定形式*                         | Ansible利用情報                      | WinRM接続 | ヘッダーセクション |
|------------|----------------------|----------------------|----------------------------------|----------------------------------|---------|-----------|
| 自動入力       | <input type="text"/> | <input type="text"/> | <input type="button" value="▼"/> | <input type="button" value="▼"/> |         |           |
|            |                      |                      |                                  |                                  |         |           |

※\*は必須項目です。

| 項目        | 入力内容    |
|-----------|---------|
| Movement名 | ログサーバ設定 |
| ホスト指定形式   | IP      |

## 3.4 Movementの設定 (2/4)

### 対話種別を作成する

次に「対話種別」を作成しておきましょう。

メニュー: Ansible-Pioneer > 対話種別リスト

- ① 登録 > 登録開始 を押下する。
- ② 各項目へ下表のように入力し、押下する。

登録

| 項目番  | 対話種別名*               | 備考 | 最終 |
|------|----------------------|----|----|
| 自動入力 | <input type="text"/> |    | 自動 |
|      |                      |    |    |

\*<sup>\*</sup>は必須項目です。

| 項目名   | 入力内容        |
|-------|-------------|
| 対話種別名 | syslogサーバ指定 |

## 3.4 Movementの設定 (3/4)

### 対話ファイルを登録する

準備を終えたところで対話ファイルを登録しましょう。  
これまでに作成した対話種別やOS種別と紐づけます。

#### メニュー: Ansible-Pioneer > 対話ファイル素材集

- ① 登録 > 登録開始 を押下する。
- ② 「参照」から対話ファイルを選び、「事前アップロード」を押下する。
- ③ 他項目を下表のように選択し、「登録」を押下する。

登録

| 素材ID | 対話種別* | OS種別* | 対話ファイル素材*                                                  |
|------|-------|-------|------------------------------------------------------------|
| 自動入力 |       |       | <input type="button" value="参照..."/> 事前アップロード<br>アップロード状況: |

\*は必須項目です。

| 対話種別        | OS種別       | 対話ファイル素材                   |
|-------------|------------|----------------------------|
| syslogサーバ指定 | vyos_RT    | vyos_set_syslog_server.yml |
| syslogサーバ指定 | Cisco_L3SW | ios_set_syslog_server.yml  |

## 3.4 Movementの設定 (4/4)

### Movementに对话種別を登録する

Movementと对话種別を関連付けましょう。

#### メニュー: Ansible-Pioneer > Movement詳細

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

登録

| 紐付項番 | Movement*            | 对话種別*                | インクルード順序*            |
|------|----------------------|----------------------|----------------------|
| 自動入力 | <input type="text"/> | <input type="text"/> | <input type="text"/> |

※\*は必須項目です。

[戻る](#) [登録](#)

| 項目       | 入力内容        |
|----------|-------------|
| Movement | ログサーバ設定     |
| 对话種別     | syslogサーバ指定 |
| インクルード順序 | 1           |

# 3.5 オペレーションの登録

## オペレーションを新規登録する

オペレーションを作成し、Movementとホストを関連付けましょう。

メニュー：基本コンソール > 投入オペレーション一覧

- ① 登録 > 登録開始 を押下する。
- ② 各項目へ下表のように入力し、[登録]を押下する。

登録

| No.  | オペレーションID | オペレーション名* | 実施予定日時* | 備考 |
|------|-----------|-----------|---------|----|
| 自動入力 | 自動入力      |           |         |    |
|      |           |           |         |    |
|      |           |           |         |    |

※\*は必須項目です。

[戻る](#) [登録](#)

| 項目名      | 入力内容             |
|----------|------------------|
| オペレーション名 | Pioneer_Practice |
| 実施予定日時   | (任意の値をご入力下さい)    |

※ 「実施予定日時」は管理用の項目です。自動的に処理が実行されるわけではありません。

# 3.6 機器一覧への登録

## 機器一覧にNW機器を登録する

機器一覧から今回の作業対象を登録しましょう。

LegacyモードやLegacy-Roleモードと異なり、「Pioneer利用情報」の記入が必要となる点に留意してください。

メニュー：基本コンソール > 機器一覧

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下図の通り選択し、[登録]を押下する。

登録

| 管理システム項目番号 | HW機器種別 | ホスト名* | IPアドレス* | EtherWakeOnLan |             | ログインパスワード |    |
|------------|--------|-------|---------|----------------|-------------|-----------|----|
|            |        |       |         | MACアドレス        | ネットワークデバイス名 | ログインユーザID | 管理 |
| 自動入力       |        |       |         |                |             |           |    |

※\*は必須項目です。

戻る      登録

| 項目                | vyos仮想ルータ | Cisco機器    |
|-------------------|-----------|------------|
| HW機器種別            | NW        |            |
| ホスト名              | (任意の値)    |            |
| IPアドレス            | (任意の値)    |            |
| ログインユーザID         | (任意の値)    |            |
| 管理                | ●         |            |
| ログインパスワード         | (任意の値)    |            |
| Pioneer利用情報/プロトコル | ssh       | telnet     |
| Pioneer利用情報/OS種別  | vyos_RT   | Cisco_L3SW |

### 3.7 パラメータシート作成 (1/2)

#### メニューを作成する

パラメータシートを作成し、  
ターゲットホストに適用するパラメータを作成・管理しましょう。

メニュー: メニュー作成 > メニュー定義/作成

メニュー作成情報

基本情報 :

|        |                         |
|--------|-------------------------|
| 項目番号   | 自動入力                    |
| メニュー名* | Pioneer実践               |
| 作成対象   | パラメータシート(ホスト/オペレーション含む) |
| 表示順序*  | 3                       |

ホストグループ利用 :  利用する

縦メニュー利用 ? :  利用する

最終更新日時 : 自動入力

最終更新者 : 自動入力

対象メニューグループ

|           |          |
|-----------|----------|
| 入力用*      | 入力用      |
| 代入値自動登録用* | 代入値自動登録用 |
| 参照用*      | 参照用      |

対象メニューグループ

アクセス許可ロール

ロール :

アクセス許可ロール選択

下表のように入力する。(次項へ)

| 項目    | 入力内容                    |
|-------|-------------------------|
| メニュー名 | Pioneer実践               |
| 作成対象  | パラメータシート(ホスト/オペレーション含む) |
| 表示順序  | 3                       |

### 3.7 パラメータシート作成 (2/2)

#### パラメータシートの項目名を定義する

前項に続き、シートの項目を定義していきましょう。

メニュー： メニュー作成 > メニュー定義/作成

1 押下して新しい項目を追加する。

2 下表のように項目の内容を定義する。

3 押下してメニュー作成を実行する。

| 項目名                  | 入力方式     | 最大バイト数 |
|----------------------|----------|--------|
| syslog_server_ip     | 文字列(單一行) | 32     |
| sub_syslog_server_ip | 文字列(單一行) | 32     |
| log_facility         | 文字列(單一行) | 32     |
| log_severity         | 文字列(單一行) | 32     |

# 3.8 データの登録

## パラメータシートにデータを登録する

メニューを作成できたところで、  
ターゲットホストの設定に使用するデータを登録しましょう。

メニュー: 入力用 > Pioneer実践(作成したメニュー)

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

The screenshot shows a registration form for a target host. The 'Host Name' field is highlighted with a red box. Below it, the 'Operation' dropdown and the 'Parameter' section (syslog\_server\_ip, sub\_syslog\_server\_ip, log\_facility, log\_severity) are also highlighted. The 'Parameter' section has a red border around its four input fields.

| ホスト名         | オペレーション          | syslog_server_ip | sub_syslog_server_ip | log_facility | log_severity |
|--------------|------------------|------------------|----------------------|--------------|--------------|
| (Cisco機器を選択) | Pioneer_practice | 任意のIPアドレス        | 任意のIPアドレス            | local7       | info         |
| (vyosルータを選択) | Pioneer_practice | 任意のIPアドレス        | 任意のIPアドレス            | local7       | info         |

# 3.9 代入値自動登録設定

## 代入値自動登録設定を行う

パラメータシートの入力が終わったところで、各項目と変数を関連付けていきます。

### メニュー: Ansible-Pioneer > 代入値自動登録設定

- ① 登録 > 登録開始 を押下する。



登録

パラメータシート

| 項目番号 | メニュー名  | 項目           | 登録方式*    | Movement         |      | Iac変数            |      |
|------|--------|--------------|----------|------------------|------|------------------|------|
|      |        |              |          | 変数名              | 代入順序 | 変数名              | 代入順序 |
| 自動入力 | メニュー選択 | メニューを選択して下さい | Movement | Movementを選択して下さい | 1    | Movementを選択して下さい | 2    |

※\*は必須項目です。

- ② 下表のように関連付け設定を入力し、「登録」を押下する。

| メニュー      | 項目                   | 登録方式   | Movement | Value変数<br>変数名       | 代入順序 |
|-----------|----------------------|--------|----------|----------------------|------|
| Pioneer実践 | syslog_server_ip     | Value型 | ログサーバ設定  | VAR_syslog_server_ip | 1    |
| Pioneer実践 | sub_syslog_server_ip | Value型 | ログサーバ設定  | VAR_syslog_server_ip | 2    |
| Pioneer実践 | log_facility         | Value型 | ログサーバ設定  | VAR_log_facility     | 空欄   |
| Pioneer実践 | log_severity         | Value型 | ログサーバ設定  | VAR_log_severity     | 空欄   |

# 3.10 代入値・対象ホストの確認

## 代入値と作業対象ホストを確認する

代入値自動登録により指定された値と対象ホストを確認しましょう。

メニュー: **Ansible-Pioneer > 作業対象ホスト & Ansible-Pioneer > 代入値管理**

- ① [フィルタ]を押下する
- ② 「pioneer代入値自動登録設定プロシージャ」によって正しい値が指定されていることを確認する。

作業対象ホスト

| 更新 | 廃止 | 項目 | オペレーション                       | Movement         | ホスト | アクセス権<br>アクセス許可ロール | 備考 | 最終更新日時              | 最終更新者                  |
|----|----|----|-------------------------------|------------------|-----|--------------------|----|---------------------|------------------------|
| 更新 | 廃止 | 1  | 3:Pioneer_practice 4:ログサーバー設定 | 2:dev-nw-01_ios  |     |                    |    | 2020/12/02 11:26:18 | pioneer代入値自動登録設定プロシージャ |
| 更新 | 廃止 | 2  | 3:Pioneer_practice 4:ログサーバー設定 | 3:dev-nw-02-vyos |     |                    |    | 2020/12/02 11:21:18 | pioneer代入値自動登録設定プロシージャ |

代入値管理

| 更新 | 廃止 | 項目 | オペレーション                       | Movement         | ホスト                    | 変数名 | Sensitive設定 | 具体値           | 代入順序 | アクション | 最終更新日時              | 最終更新者                  |
|----|----|----|-------------------------------|------------------|------------------------|-----|-------------|---------------|------|-------|---------------------|------------------------|
| 更新 | 廃止 | 10 | 3:Pioneer_practice 4:ログサーバー設定 | 2:dev-nw-01_ios  | 8:VAR_syslog_server_ip | OFF |             | 10.15.128.100 | 1    |       | 2020/12/02 11:21:18 | pioneer代入値自動登録設定プロシージャ |
| 更新 | 廃止 | 12 | 3:Pioneer_practice 4:ログサーバー設定 | 2:dev-nw-01_ios  | 8:VAR_syslog_server_ip | OFF |             | 10.15.128.101 | 2    |       | 2020/12/02 11:22:14 | pioneer代入値自動登録設定プロシージャ |
| 更新 | 廃止 | 14 | 3:Pioneer_practice 4:ログサーバー設定 | 2:dev-nw-01_ios  | 7:VAR_log_facility     | OFF |             | local7        | 1    |       | 2020/12/02 11:25:33 | pioneer代入値自動登録設定プロシージャ |
| 更新 | 廃止 | 16 | 3:Pioneer_practice 4:ログサーバー設定 | 2:dev-nw-01_ios  | 5:VAR_log_severity     | OFF |             | info          | 1    |       | 2020/12/02 11:25:55 | pioneer代入値自動登録設定プロシージャ |
| 更新 | 廃止 | 11 | 3:Pioneer_practice 4:ログサーバー設定 | 3:dev-nw-02-vyos | 8:VAR_syslog_server_ip | OFF |             | 10.15.128.100 | 1    |       | 2020/12/02 11:21:18 | pioneer代入値自動登録設定プロシージャ |
| 更新 | 廃止 | 13 | 3:Pioneer_practice 4:ログサーバー設定 | 3:dev-nw-02-vyos | 8:VAR_syslog_server_ip | OFF |             | 10.15.128.101 | 2    |       | 2020/12/02 11:22:14 | pioneer代入値自動登録設定プロシージャ |
| 更新 | 廃止 | 15 | 3:Pioneer_practice 4:ログサーバー設定 | 3:dev-nw-02-vyos | 7:VAR_log_facility     | OFF |             | local7        | 1    |       | 2020/12/02 11:25:33 | pioneer代入値自動登録設定プロシージャ |
| 更新 | 廃止 | 17 | 3:Pioneer_practice 4:ログサーバー設定 | 3:dev-nw-02-vyos | 5:VAR_log_severity     | OFF |             | info          | 1    |       | 2020/12/02 11:25:55 | pioneer代入値自動登録設定プロシージャ |

# 3.11 作業の実行 (1/2)

## Movementを直接実行する

本シナリオで作成したMovementは一つです。Conductorを作成する必要はありません。  
「作業実行」メニューから個別実行しましょう。

メニュー：Ansible-Pioneer > 作業実行

The screenshot shows the Ansible-Pioneer application interface. On the left is a vertical menu bar with various options like 'Menu', 'Movement[一覧]', 'Operation[一覧]', etc. The main area has two stacked tables.

**Top Table (Movement Selection):**

| 選択                               | MovementID | Movement名 | オーケストレーター       | 遅延タイマー | Ansible利用情報 | 変数カウント | 最終更新日時              | 最終更新者   |
|----------------------------------|------------|-----------|-----------------|--------|-------------|--------|---------------------|---------|
| <input checked="" type="radio"/> | 4          | ログサーバー設定  | Ansible Pioneer |        | ホスト指定形式 IP  |        | 2020/11/25 15:17:40 | システム管理者 |

フィルタ結果件数: 1

**Bottom Table (Operation Selection):**

| 選択                               | No. | オペレーションID | オペレーション名           | 実施予定期間           | 最終実行日時           | アクセス権    | 最終更新日時              | 最終更新者             |
|----------------------------------|-----|-----------|--------------------|------------------|------------------|----------|---------------------|-------------------|
| <input checked="" type="radio"/> | 3   |           | 3_Pioneer_practice | 2020/11/26 16:00 | 2020/12/02 13:52 | アクセス許可 - | 2020/12/02 13:52:36 | pioneer 作業実行プロセッサ |

フィルタ結果件数: 1

**Action Buttons:**

- MovementID: 4
- Movement名: ログサーバー設定
- ドライラン (Orange button)
- 実行 (Red button)

Tips

実行後、自動で【作業状態確認】へ  
画面遷移します。

# 3.11 作業の実行 (2/2)

## 実行結果を確認する

作業を実行すると画面が遷移し、**実行ステータスやログ**が表示されます。

### メニュー： Ansible-Pioneer > 作業状態確認

実行ステータス

説明

対象作業

| 項目                       | 値                                                                                                                                                                  |          |                          |                          |  |  |  |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|--------------------------|--------------------------|--|--|--|
| 作業No.                    | 103                                                                                                                                                                |          |                          |                          |  |  |  |
| 実行種別                     | 通常                                                                                                                                                                 |          |                          |                          |  |  |  |
| ステータス                    | 完了                                                                                                                                                                 |          |                          |                          |  |  |  |
| 実行エンジン                   | Ansible Engine                                                                                                                                                     |          |                          |                          |  |  |  |
| 呼出元Symphony              |                                                                                                                                                                    |          |                          |                          |  |  |  |
| 呼出元Conductor             |                                                                                                                                                                    |          |                          |                          |  |  |  |
| 実行ユーザ                    | システム管理者                                                                                                                                                            |          |                          |                          |  |  |  |
| Movement                 | <table border="1"><thead><tr><th>ID</th><th>名称</th><th>ログサーバー設定</th></tr></thead><tbody><tr><td></td><td></td><td></td></tr></tbody></table>                       | ID       | 名称                       | ログサーバー設定                 |  |  |  |
| ID                       | 名称                                                                                                                                                                 | ログサーバー設定 |                          |                          |  |  |  |
|                          |                                                                                                                                                                    |          |                          |                          |  |  |  |
|                          | 遅延タイム(分)                                                                                                                                                           |          |                          |                          |  |  |  |
| Ansible利用情報              | ホスト指定形式 IP<br>並列実行数                                                                                                                                                |          |                          |                          |  |  |  |
| No.                      | 3                                                                                                                                                                  |          |                          |                          |  |  |  |
| オペレーション                  | 名称 Pioneer_practice                                                                                                                                                |          |                          |                          |  |  |  |
| ID                       | 3                                                                                                                                                                  |          |                          |                          |  |  |  |
| 作業対象ホスト                  | <table border="1"><thead><tr><th>確認</th></tr></thead></table>                                                                                                      | 確認       |                          |                          |  |  |  |
| 確認                       |                                                                                                                                                                    |          |                          |                          |  |  |  |
| 代入値                      | <table border="1"><thead><tr><th>確認</th></tr></thead></table>                                                                                                      | 確認       |                          |                          |  |  |  |
| 確認                       |                                                                                                                                                                    |          |                          |                          |  |  |  |
| 入力データ                    | 投入データ <a href="#">InputData_0000000103.zip</a>                                                                                                                     |          |                          |                          |  |  |  |
| 出力データ                    | 結果データ <a href="#">ResultData_0000000103.zip</a>                                                                                                                    |          |                          |                          |  |  |  |
| 作業状況                     | <table border="1"><thead><tr><td>予約日時</td></tr></thead><tbody><tr><td>開始日時 2020/12/02 13:52:37</td></tr><tr><td>終了日時 2020/12/02 13:52:46</td></tr></tbody></table> | 予約日時     | 開始日時 2020/12/02 13:52:37 | 終了日時 2020/12/02 13:52:46 |  |  |  |
| 予約日時                     |                                                                                                                                                                    |          |                          |                          |  |  |  |
| 開始日時 2020/12/02 13:52:37 |                                                                                                                                                                    |          |                          |                          |  |  |  |
| 終了日時 2020/12/02 13:52:46 |                                                                                                                                                                    |          |                          |                          |  |  |  |

ログ

進行状況(実行ログ)

ファイル:   該当のみ表示

```
"exec_file": "/exastro/data_relay_storage/ansible_driver/pioneer/ns/0000000103/in/dialog_files/vyos-test/0000000001-对话-v.  
extra_args": "__undefinedsymbol__",  
"grep_shell_dir": "./library",  
"host_vars_file": "/exastro/data_relay_storage/ansible_driver/pioneer/ns/0000000103/tmp/original_host_vars/vyos-test",  
"inventory_hostname": "____",  
"log_file_dir": "/exastro/data_relay_storage/ansible_driver/pioneer/ns/0000000103/out",  
"protocol": "ssh",  
"ssh_key_file": "__undefinedsymbol__",  
"username": "vyos"  
},  
"msg": "____:nomal exit"  
}  
META: ran handlers  
META: ran handlers  
PLAY RECAP *****  
____ : ok=1 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

Tips

投入データや結果データをまとめたzipファイルをダウンロードできます。

Tips

コマンドで実行結果を確認する場合、以下のコマンドでログの設定を確認できます。

IOS - "# show logging"  
vyos - "\$ show configuration"

Exastro

62

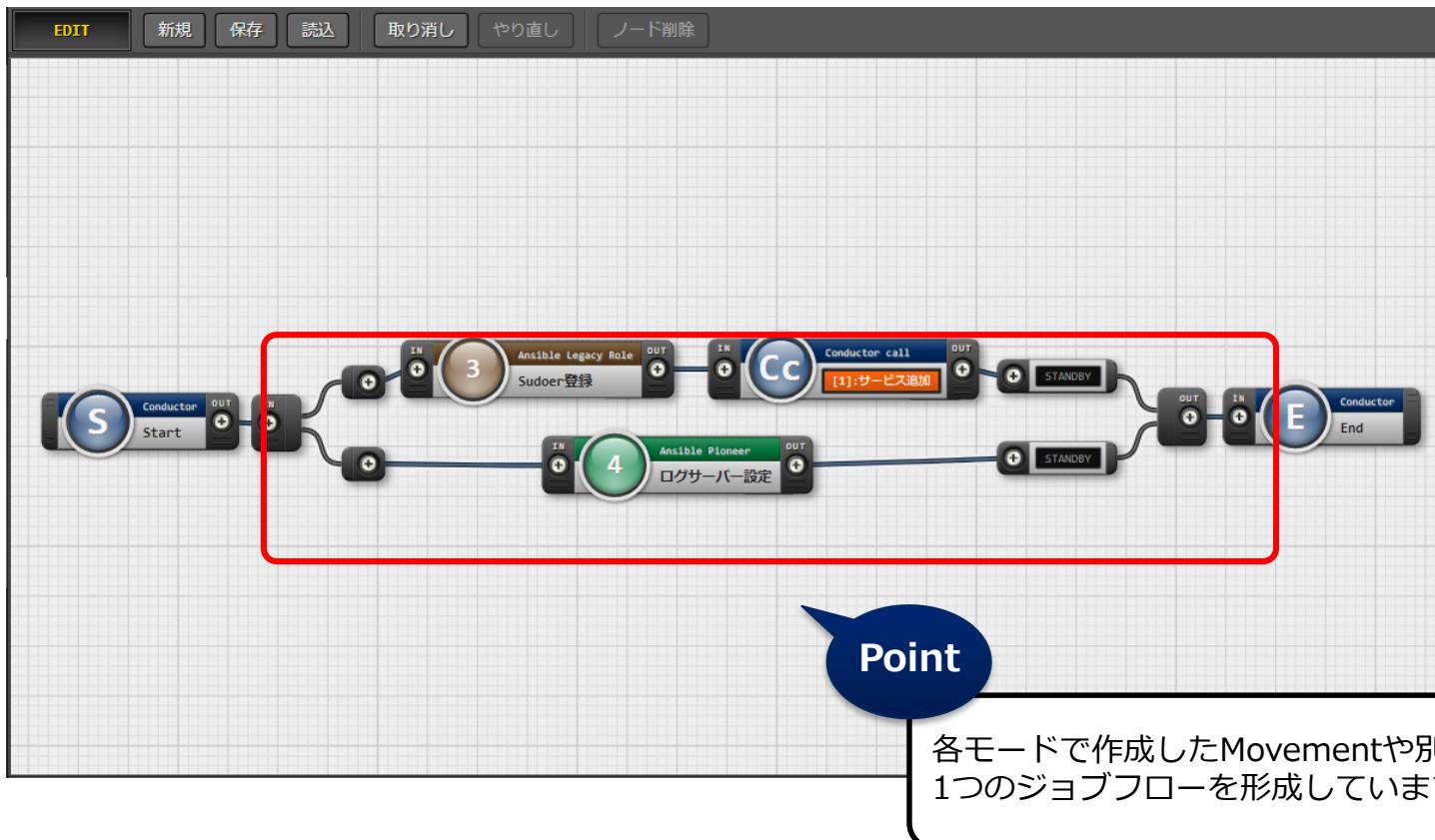
# A 付録



# 付録① Conductorで3モードを束ねて実行する

## 3モードを束ねて実行する

本編ではモードごとの作業を個別に実行しましたが、Conductorを利用することで  
複数のモードの作業を実行するジョブフローを作成することもできます。





**Exastro**