



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

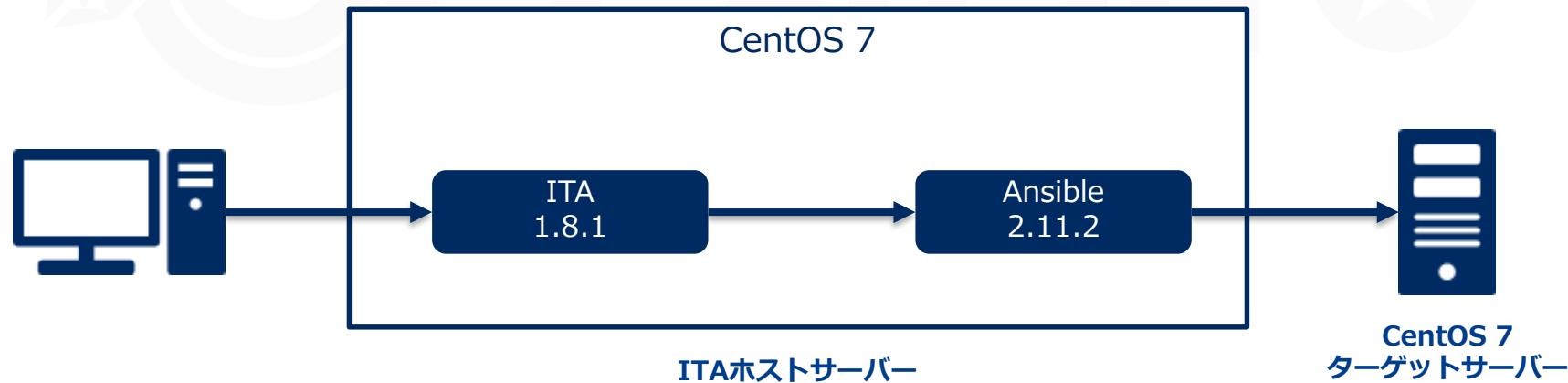
本章で使用する作業環境は以下の通りです。(第2章Ansible-LegacyRole編に共通)
ITAホストサーバーとは別に、ターゲットとなるサーバーをご用意ください。

ITAホストサーバー

- CentOS 7 (※1)
- ITA 1.8.1
- Ansible 2.11.2

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

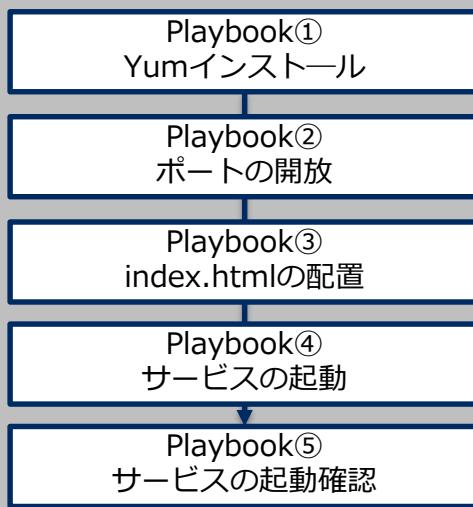
大別して3ステップから成る作業を実行し、**playbookの再利用性**を体感いただけます。

- 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。

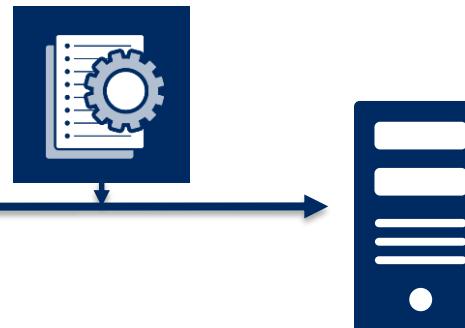
今回は「Apache・Tomcat」両サービスのインストールと起動を行います。

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: install firewalld
  yum: pkg=firewalld state=present

- name: start firewalld
  service: name=firewalld state=started
  enabled=yes

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

ファイル名：2-open_port.yml

firewalldをインストール、起動し、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 index.html
  copy:
    src: "{{ CPF_index_html }}"
    dest: /var/www/html/index.html
    owner: root
    group: root
    mode: 0644
    backup: yes
  when: 'VAR_service_name == "httpd"'
```

ファイル名: 3-copy_index.yml

ファイルをコピーし配置します。

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

ファイル名: 4-start_service.yml

指定したサービスを起動します。

```
- 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: 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)

Index.html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。
以下のファイルを作成してください。

ファイル名: index.html

このhtmlファイルをドキュメントルート直下に配置します。

```
<html>
<head>
<title>私のページ</title>
</head>
<body>
```

Ansible-Driver実習

```
<br>
```

```
</body>
</html>
```

実行作業完了後、正しくindex.htmlファイルが配置されるとこのような画面が見えます。



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	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。

2

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

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 > Playbook素材集

- ① 登録 > 登録開始 を押下する。
- ② [ファイルを選択] 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	Playbook素材名*	Playbook素材*	アクセス権	備考
自動入力		<input type="button" value="ファイルを選択"/> 選択されていません <input type="button" value="事前アップロード"/> アップロード状況:	設定	

* *は必須項目です。



3

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
copy_index	3-copy_index.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 [ファイルを選択] 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

△閉じる

素材ID	ファイル埋込変数名	ファイル素材	アクセス権		備考	最終更新日時	最終更新者
			設定	アクセス許可ロール			
自動入力		<p>ファイルを選択 選択されていません</p> <p>事前アップロード</p> <p>アップロード状況:</p>	設定			自動入力	自動入力

※ *は必須項目です。

戻る 登録

ファイル埋込変数名	ファイル素材
CPF_index_html	index.html

3

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-Playbook紐付

- 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*	Playbook素材*	インクルード順序*	アクセス権	備考
自動入力	<input type="button" value="▼"/>	<input type="button" value="▼"/>	<input type="button" value=""/>	<input type="button" value="設定"/> アクセス許可ロール	

*は必須項目です。

戻る

関連付け表

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	copy_index	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

The screenshot shows the Exastro Conductor editor interface. On the left, a workflow diagram is displayed with nodes S (Start), 15 (Install_Packages), 16 (Open_Ports), 17 (Start_Service), and E (End). A red box highlights the sequence of nodes 15, 16, and 17. On the right, a sidebar titled "Conductor名" contains the text "サービス追加". The main panel shows the "Conductor" configuration with fields for Name (サービス追加) and ID (Auto numbering). A red callout labeled "1" points to the Name field with the instruction "Conductorの名前を入力する。". Below the configuration, a table lists the movements in the order they appear in the workflow:

Movement	作業順序
Install Packages	1
Open Ports	2
Start Service	3

A red callout labeled "2" points to the sidebar with the instruction "ドラッグ&ドロップで必要なMovementを追加する。". A blue arrow points from the sidebar back to the movement table. A red callout labeled "3" points to the table with the instruction "作業順にノード同士を繋ぐ。". A red callout labeled "4" points to the bottom-left corner with the instruction "[登録]を押下する。".

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 > 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*	アクセス権	備考
自動入力	自動入力			設定	

*必須項目です。

2

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。

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	パスワード認証

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

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を選択

一意制約(複数項目):

パターン: 一意制約(複数項目)を選択

アクセス許可ロール:

ロール: アクセス許可ロールを選択

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/オペレーションあり)
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

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

The screenshot shows the configuration interface for creating a parameter sheet. It consists of three main parts:

- Left Panel (1):** A grid of four input fields for defining parameters. Each field has a dropdown menu icon at the top right. The first field is labeled "package_name" with type "文字列(單一行)" and max length "32". The second is "package_name_sub" with type "文字列(單一行)" and max length "32". The third is "port_number" with type "文字列(單一行)" and max length "32". The fourth is "service_name" with type "文字列(單一行)" and max length "32". Each field also has "説明" (Description) and "備考" (Remarks) sections below it.
- Middle Panel (2):** A table showing the defined parameters. The columns are "項目名" (Item Name), "入力方式" (Input Method), and "最大バイト数" (Max Byte Length). The data is as follows:

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 Panel:** A preview window titled "一覧(プレビュー)" (List Preview) showing a table of operations. The columns are "No.", "ホスト名" (Host Name), "オペレーション名" (Operation Name), "基準日時" (Standard Date and Time), "実施予定日" (Planned Implementation Date), and "最終実行日" (Final Execution Date). The data shows three entries for host "192.168.0.1" with operation name "オペレーション" and standard date/time "2020/01/01 00:00".

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。

2

登録

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)	Install Apache	httpd	(空欄)	80/tcp	httpd
(対象のホスト)	Install Tomcat	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

パラメータシート(From)		登録方式	To(変数(To))		
項目番号	メニュー名	項目	Movement	Key変数 変数名	Value変数 変数名
自動入力	メニュー名:メニュー	メニューを選択して下さい	Movement	Movementを選択して下さい	Movementを選択して下さい
※赤字は必須項目です。					

2 戻る 登録

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												
履歴	複製	更新	廃止	項目番号	オペレーション	Movement	ホスト	代入値管理	アクセス権	備考	最終更新日時	最終更新者
履歴	複製	更新	廃止	29	11:Install Apache	15:Install_Packages	2:webB	代入値管理	アクセス許可ロール		2021/08/24 16:29:52	legacy代入値自動登録設定プロシージャ
履歴	複製	更新	廃止	30	11:Install Apache	16:Open_Ports	2:webB	代入値管理	アクセス許可ロール		2021/08/24 16:29:52	legacy代入値自動登録設定プロシージャ
履歴	複製	更新	廃止	31	11:Install Apache	17:Start_Service	2:webB	代入値管理	アクセス許可ロール		2021/08/24 16:29:52	legacy代入値自動登録設定プロシージャ
履歴	複製	更新	廃止	32	12:Install Tomcat	15:Install_Packages	2:webB	代入値管理	アクセス許可ロール		2021/08/24 16:30:00	legacy代入値自動登録設定プロシージャ
履歴	複製	更新	廃止	33	12:Install Tomcat	16:Open_Ports	2:webB	代入値管理	アクセス許可ロール		2021/08/24 16:30:00	legacy代入値自動登録設定プロシージャ
履歴	複製	更新	廃止	34	12:Install Tomcat	17:Start_Service	2:webB	代入値管理	アクセス許可ロール		2021/08/24 16:30:00	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													
履歴	複製	更新	廃止	項目番号	オペレーション	Movement	ホスト	変数名	Sensitive設定	具体値	代入順序	最終更新日時	最終更新者
履歴	複製	更新	廃止	51	11:Install Apache	15:Install_Packages	2:webB	7:VAR_package_name	OFF	httpd	1	2021/08/24 16:29:52	legacy代入値自動登録設定プロシージャ
履歴	複製	更新	廃止	52	11:Install Apache	16:Open_Ports	2:webB	8:VAR_port_number	OFF	80/tcp		2021/08/24 16:29:52	legacy代入値自動登録設定プロシージャ
履歴	複製	更新	廃止	53	11:Install Apache	17:Start_Service	2:webB	9:VAR_service_name	OFF	httpd		2021/08/24 16:29:52	legacy代入値自動登録設定プロシージャ
履歴	複製	更新	廃止	54	12:Install Tomcat	15:Install_Packages	2:webB	7:VAR_package_name	OFF	tomcat	1	2021/08/24 16:30:00	legacy代入値自動登録設定プロシージャ
履歴	複製	更新	廃止	55	12:Install Tomcat	16:Open_Ports	2:webB	7:VAR_package_name	OFF	tomcat-webapps	2	2021/08/24 16:30:00	legacy代入値自動登録設定プロシージャ
履歴	複製	更新	廃止	56	12:Install Tomcat	17:Start_Service	2:webB	8:VAR_port_number	OFF	8080/tcp		2021/08/24 16:30:00	legacy代入値自動登録設定プロシージャ
履歴	複製	更新	廃止	57	12:Install Tomcat	17:Start_Service	2:webB	9:VAR_service_name	OFF	tomcat		2021/08/24 16:30:00	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 Exastro interface with the following steps highlighted:

1. In the 'Conductor' menu, 'Conductor[一覧]' is selected. A red box and arrow point to it with the text '実行するConductorを選択する。' (Select the Conductor to be executed.).
2. In the 'オペレーション' section, 'オペレーション[一覧]' is selected. A red box and arrow point to it with the text 'オペレーションを選択する。' (Select the operation.).
3. At the bottom of the screen, the '実行' (Execute) button is highlighted with a red box and arrow, with the text '画面下部より、[実行]を押下する。' (Press the [Execute] button at the bottom of the screen.).

Below the main interface, a small window titled 'ログ' (Log) shows the status of the execution.

No.	オペレーションID	オペレーション名	実行予定期間	最終実行日時	アクセス権	備考	最終更新日時	最終更新者
1	1	1 Install Apache	2021/01/26 11:12 - 2021/01/26 11:34	2021/01/26 11:34:07	Legacy作業実行プロシージャ			
2	2	2 Install Tomcat	2021/01/26 11:12 - 2021/01/26 11:37	2021/01/26 11:37:24	Legacy作業実行プロシージャ			
3	3	3 LegacyRole_Practice	2021/01/26 16:28 - 2021/01/26 16:39	2021/01/26 16:39:27	LegacyRole作業実行プロシージャ			
4	4	4 Pioneer_Practice	2021/01/27 10:52 - 2021/01/27 11:13	2021/01/27 11:13:31	pioneer作業実行プロシージャ			

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 'CHECKING' tab of the Conductor Task Confirmation interface. On the left, a horizontal workflow diagram consists of several nodes connected by arrows. The nodes are labeled: 'Conductor Start' (green), 'Install_Packages' (green with 'DONE' status), 'Open_Ports' (green with 'DONE' status), 'Start_Service' (green with 'DONE' status), and 'Conductor End' (green). Each node has an 'Available Legacy' button next to it. A red callout box with the number '1' points to the '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). A red callout box with the number '2' points to a detailed information panel on the right, which is expanded for the 'Install_Packages' node. This panel displays various details: Node type: movement, Node Instance ID: 138, Node name: node-3, Status: 正常終了 (Normal completion), Start time: 2021/08/30 18:53:23, End time: 2021/08/30 18:53:39, Operation status: 作業状態確認 (Operation status: Operation status confirmed), and Operation name: (empty). A red callout box at the bottom right contains the text 'Tips' and '「DONE」をクリックすることで 詳細結果へ移動できます。' (By clicking 'DONE', you can move to the detailed result page).

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。

The screenshot shows the Exastro UI interface. On the left is a dark sidebar menu with various options like 'Menu', 'Movement一覧', 'Playbook素材集', etc. The main area has tabs for '説明' (Description) and '対象作業' (Target Job). Below these are two large tables: one for the job itself and one for the movement. A red callout bubble labeled 'Tips' points to the '確認' (Check) button in the movement table, which is highlighted in orange. Another red callout bubble labeled 'Tips' points to a red box around two zip files: 'InputData_0000000089.zip' and 'ResultData_0000000089.zip'. To the right, there's a separate window titled '進行状況(実行ログ)' (Execution Status (Execution Log)) showing Ansible log output.

項目	値
作業No.	89
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	サービス追加
実行ユーザ	システム管理者
ID	16
名称	Open_Ports
Movement	遅延タイム(分)
Ansible利用情報	ホスト指定形式 WinRM接続
No.	11
オペレーション	名称 Install Apache
ID	11
作業対象ホスト	確認
代入値	確認
入力データ	投入データ InputData_0000000089.zip
出力データ	結果データ ResultData_0000000089.zip
作業状況	予約日時 開始日時 終了日時
	2021/09/03 15:45:31 2021/09/03 15:45:37

進行状況(実行ログ)

```
permanent": true,
"port": "80/tcp",
"port_forward": null,
"rich_rule": null,
"service": null,
"source": null,
"state": "enabled",
"target": null,
"timeout": 0,
"zone": null
},
"msg": "Permanent and Non-Permanent(immediate) operation"
}
META: ran handlers
META: ran handlers

PLAY RECAP ****
webB : ok=3    changed=0    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](http://(ホストのIPアドレス):80)
Tomcat- [http://\(ホストのIPアドレス\):8080](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(1/3)

作業環境

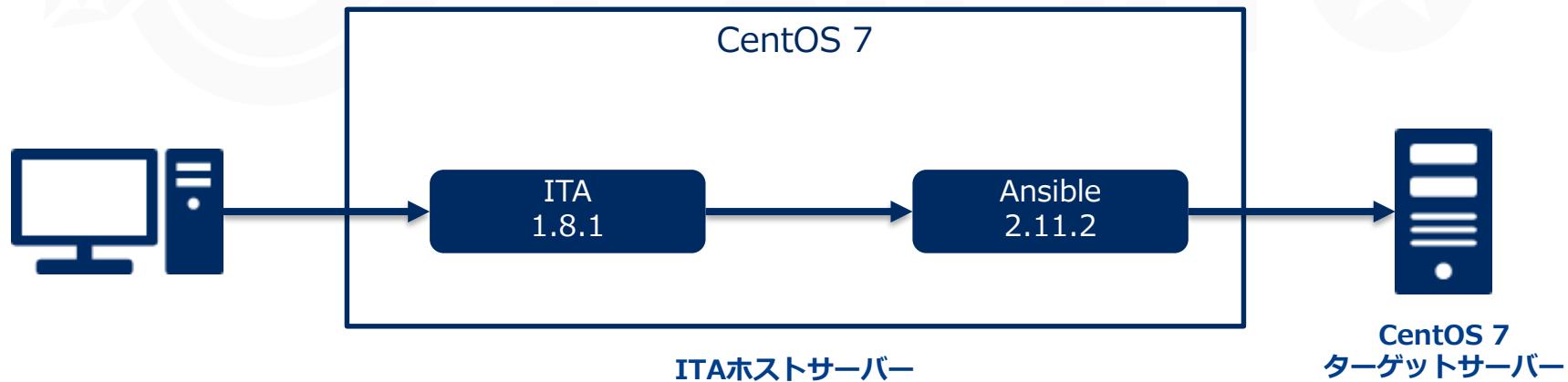
本章で使用する作業環境は以下の通りです。(第1章Ansible-Legacy編と共に)
ITAホストサーバーとは別に、ターゲットとなるサーバーをご用意ください。

ITAホストサーバー

- CentOS 7 (※1)
- ITA 1.8.1
- Ansible 2.11.2

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(2/3)

シナリオのイメージ

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。



2.1 作業環境とシナリオ(3/3)

ロールダウンロード

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。

ダウンロード方法

- ① Githubリポジトリを押下する
- ② Codeを押下する
- ③ DownLoadZIPを押下する

The screenshot shows a GitHub repository page for 'sudo' by 'weareinteractive'. At the top right, there is a red box around the 'Code' button. In the middle section, there is another red box around the 'Code' button in a dropdown menu. At the bottom left, there is a red box around the 'Download ZIP' button.

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。

1

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

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。

2

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



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 |

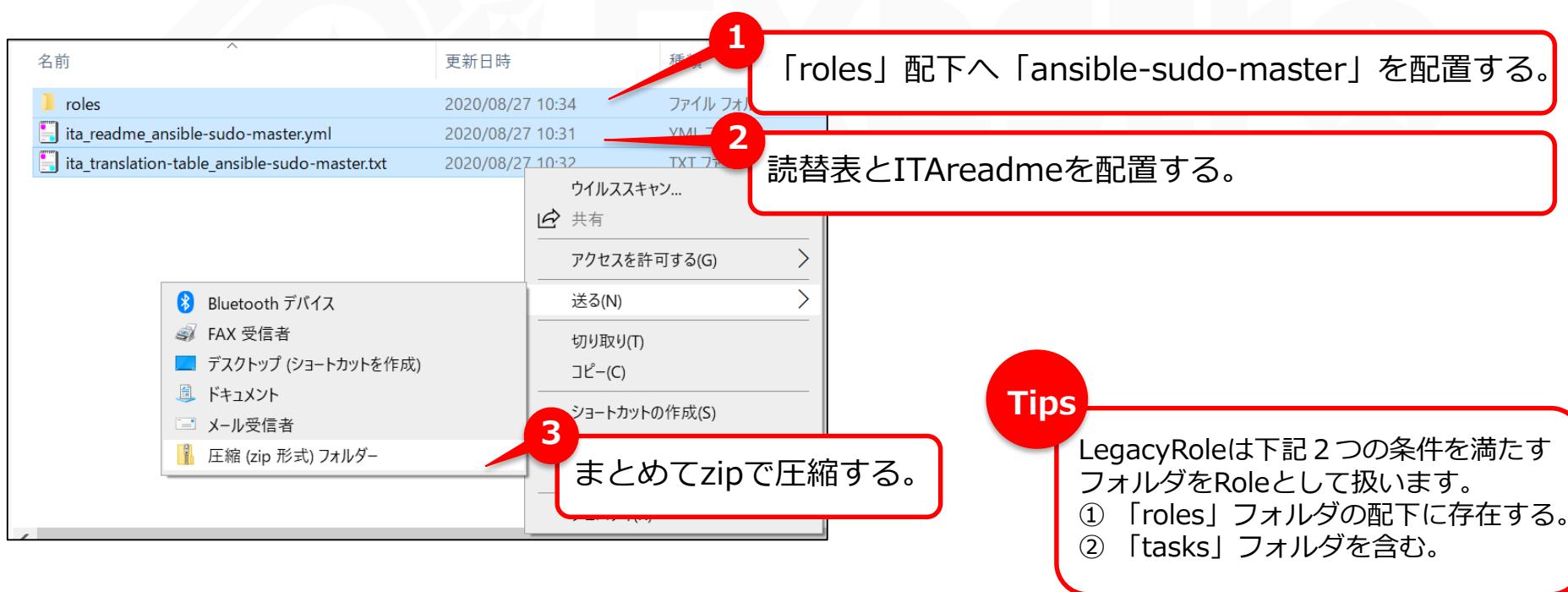
## 2.2 ロールパッケージの準備(4/4)

### 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 | 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。

2

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

戻る 登録

## 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 [ファイルを選択] 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)*          | アクセス権 |           |
|------|------------|-------------------------------|-------|-----------|
| 自動入力 |            | ファイルを選択 選択されていません<br>事前アップロード | 設定    | アクセス許可ロール |
|      |            | アップロード状況:                     |       |           |

※\*は必須項目です。

戻る      登録

3

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-ロール紐付

- 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"/> |
|        |                      |                      | ロールパッケージを選択して下さい     |                      |

\* \*は必須項目です。

戻る 登録

2

| 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 > 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。

The screenshot shows the 'Operation Registration' screen. A red box highlights the first two columns of the table: 'Operation ID' and 'Operation Name'. A red arrow points from a red circle containing the number '2' to a callout box below. The callout box contains two rows of input fields:

| 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。)       |

\* \*は必須項目です。

戻る      登録

※ 「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。

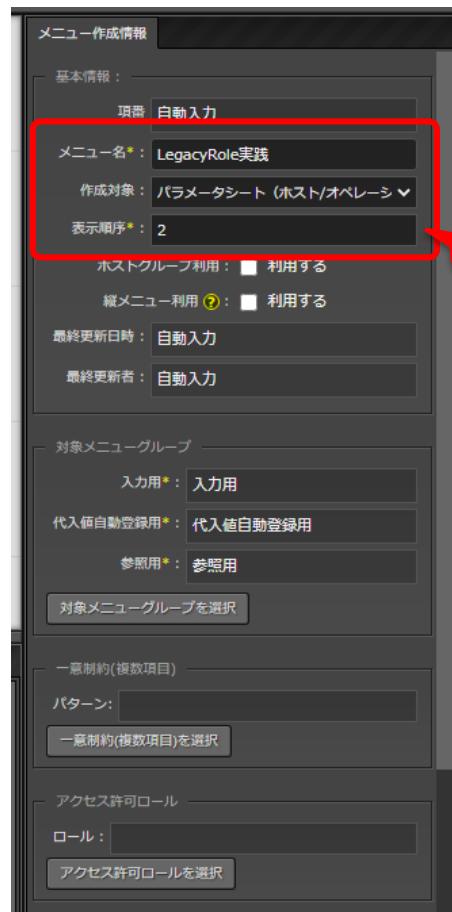
| 項目        | 入力内容    |
|-----------|---------|
| HW機器種別    | SV      |
| ホスト名      | (任意の値)  |
| IPアドレス    | (任意の値)  |
| ログインユーザID | (任意の値)  |
| 管理        | ●       |
| ログインパスワード | (任意の値)  |
| 認証方式      | パスワード認証 |

## 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/オペレーションあり)

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

対象メニューグループを選択

一意制約(複数項目)

パターン:

一意制約(複数項目)を選択

アクセス許可ロール

ロール:

アクセス許可ロールを選択

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/オペレーションあり) |
| 表示順序  | 2                           |

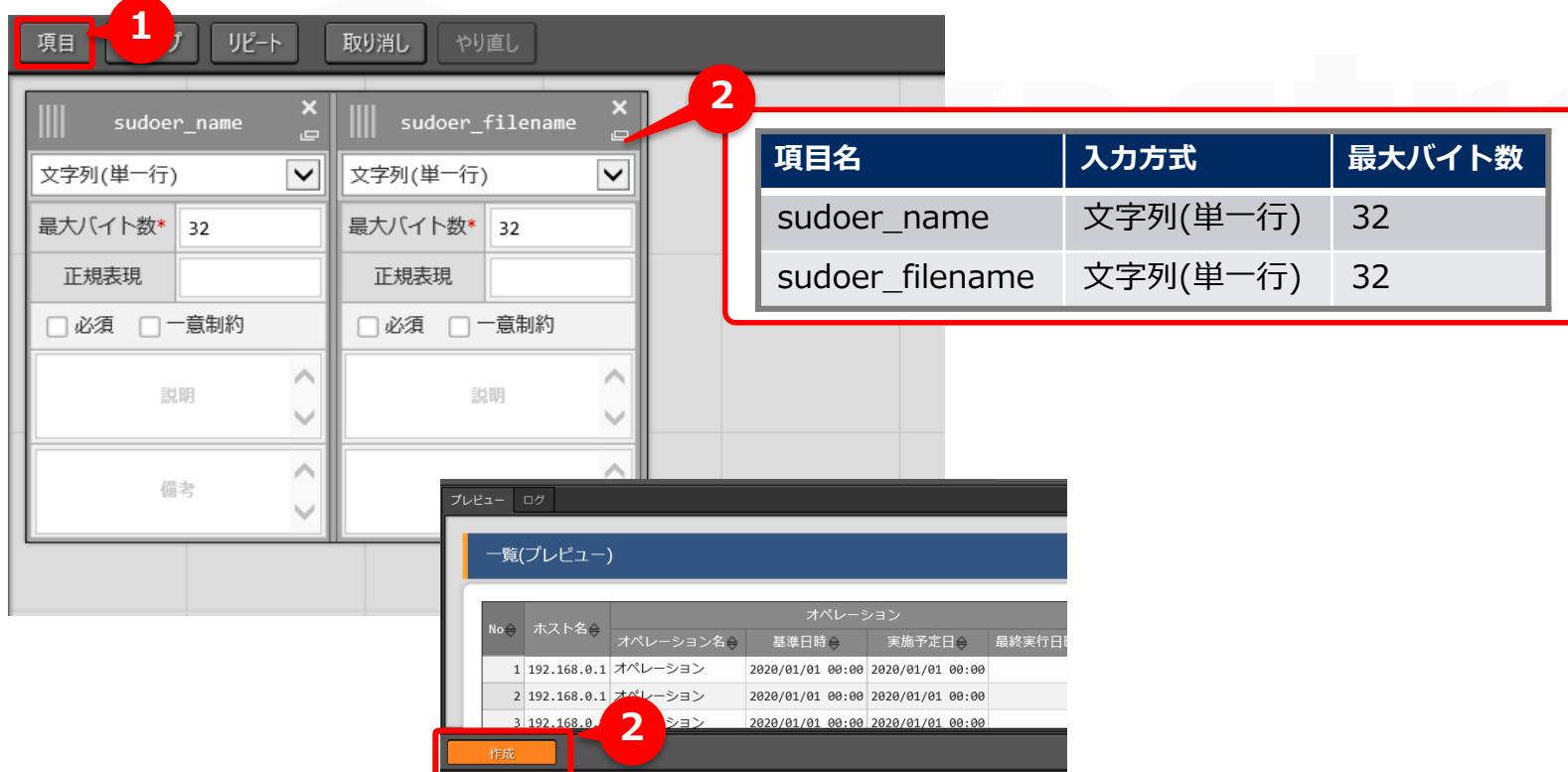
## 2.6 パラメータシート作成(2/2)

### 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

- ① 項目を押下して新しい項目を追加する。
- ② 各項目で下表のように入力し、[作成]を押下する。



## 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。

登録

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

2

| 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

| パラメータシート(From)                    |       | 登録方式                              | IaC変数(To) |          |                               |
|-----------------------------------|-------|-----------------------------------|-----------|----------|-------------------------------|
| 項目番号                              | メニュー名 | 項目                                | 登録方式      | 変数名      | メンバーバンドル名                     |
| 自動入力                              | メニュー名 | メニューを選択して下さい                      | Movement  | Movement | Movementを選択して下さい。変数名を選択して下さい。 |
| ※*は必須項目です。                        |       |                                   |           |          |                               |
| <input type="button" value="戻る"/> |       | <input type="button" value="登録"/> |           |          |                               |

2

| メニュー名         | 項目名             | 登録方式   | Movement | Value変数<br>変数名        | Value変数<br>メンバーバンドル名 |
|---------------|-----------------|--------|----------|-----------------------|----------------------|
| 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 |    |    |    |    |                        |             |        |       |           |    | △閉じる                |                           |
|-------|----|----|----|----|------------------------|-------------|--------|-------|-----------|----|---------------------|---------------------------|
| 履歴    | 複製 | 更新 | 廃止 | 項目 | オペレーション                | Movement    | ホスト    | 代入値管理 | アクセス権     | 備考 | 最終更新日時              | 最終更新者                     |
|       |    |    |    |    |                        |             |        |       | アクセス許可ロール |    |                     |                           |
| 履歴    | 複製 | 更新 | 廃止 | 1  | 13:LegacyRole_Practice | 18:Sudoer登録 | 2:webB | 代入値管理 |           |    | 2021/08/25 14:56:14 | 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 |    |    |    |    |                        |             |        |                         |            |             |         |                     | △閉じる                      |
|-------|----|----|----|----|------------------------|-------------|--------|-------------------------|------------|-------------|---------|---------------------|---------------------------|
| 履歴    | 複製 | 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    | Sensitive設定 | 具       | 最終更新日時              | 最終更新者                     |
|       |    |    |    |    |                        |             |        |                         |            |             |         |                     | アクセス権                     |
| 履歴    | 複製 | 更新 | 廃止 | 1  | 13:LegacyRole_Practice | 18:Sudoer登録 | 2:webB | 2:LCA_sudo_users        | 1:[0].name | OFF         | example | 2021/08/25 14:56:14 | legacyRole代入値自動登録設定プロシージャ |
| 履歴    | 複製 | 更新 | 廃止 | 2  | 13:LegacyRole_Practice | 18:Sudoer登録 | 2:webB | 1:LCA_sudo_sudoers_file |            | OFF         | example | 2021/08/25 14:56:55 | 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 'Ansible-LegacyRole' interface under the 'Movement' section. It displays two main tables:

- Movement[一覧]**: Shows a single entry: '4 Sudoer登録' (Movement ID: 4, Name: Ansible Legacy Role). A red circle labeled '1' points to this row with the text: '実行するMovementを選択する。' (Select the Movement to execute.).
- オペレーション[一覧]**: Shows four operations: 1 Install Apache, 2 Install Tomcat, 3 LegacyRole\_Practice (selected), and 4 Pioneer\_Practice. A red circle labeled '2' points to this table with the text: 'オペレーションを選択する。' (Select the operation.).

At the bottom, there are two buttons: 'ドライラン' (Dry Run) and '実行' (Execute). A red circle labeled '3' points to the '実行' button with the text: '[実行]を押下する。' (Press the [Execute] button.).

**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 > 作業状態確認

The screenshot shows the 'Job Status Confirmation' screen. On the left is a vertical menu with options like 'Menu', 'Movement', 'Ansible Role Management', 'Job Status Confirmation' (which is selected), and 'Job Management'. The main area has two tabs: 'Explanation' (説明) and 'Target Job' (対象作業). The 'Target Job' tab displays a table of job parameters:

| 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       | 確認             |                                           |
| 代入値           | 確認             |                                           |
| 入力データ         | 投入データ          | <a href="#">InputData_0000000002.zip</a>  |
| 出力データ         | 結果データ          | <a href="#">ResultData_0000000002.zip</a> |
| 予約日時          |                |                                           |
| 作業状況          | 開始日時           | 2020/08/22 03:27:40                       |
|               | 終了日時           | 2020/08/22 03:28:16                       |

The screenshot shows the 'Execution Log' screen. It includes a search bar and a checkbox for filtering by target. The log output shows Ansible tasks being run on a 'testserver' host. The log ends with a 'PLAY RECAP' summary:

```
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。

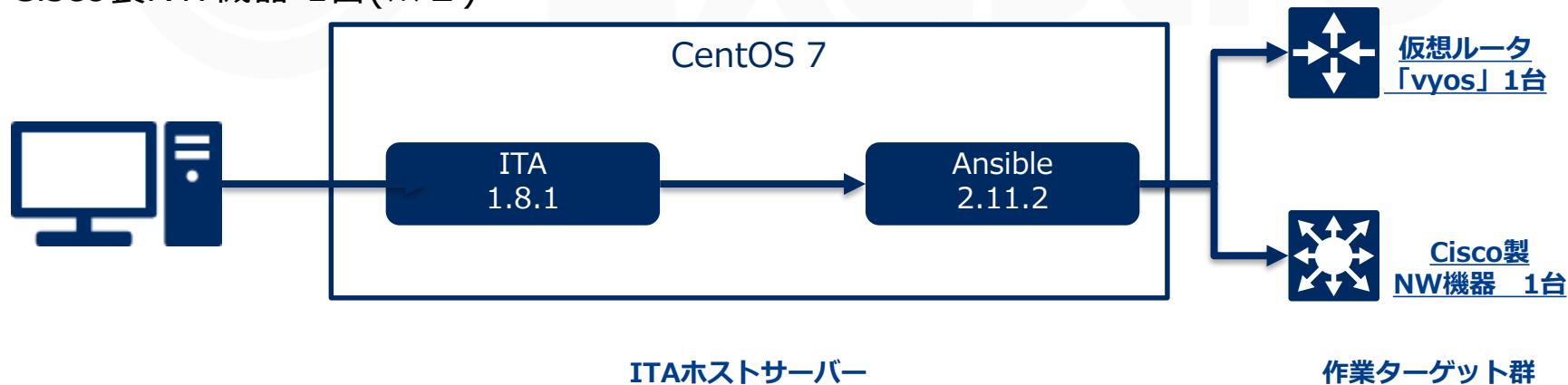
NW機器の操作を実施しますので、下記の通り環境をご用意ください。

### ITAホストサーバー

- CentOS 7 (※1)
- ITA 1.8.1
- Ansible 2.11.2

### 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

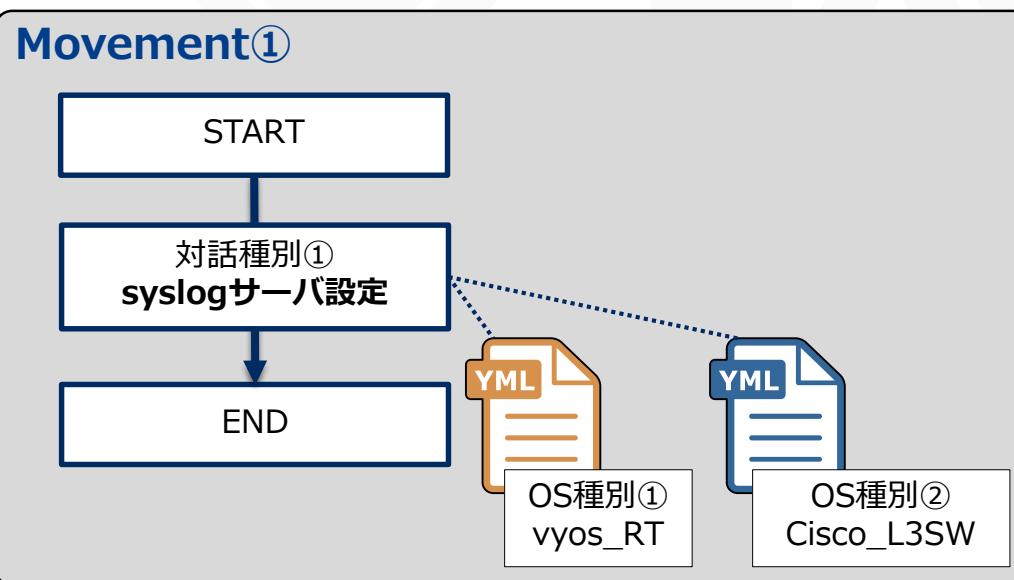
## シナリオ

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。

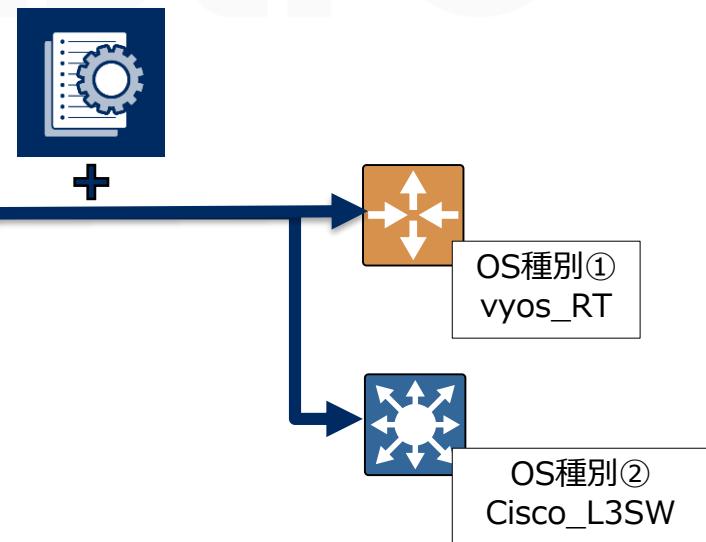
- ① telnetもしくはsshの疎通さえあれば、対話ファイルの実行が可能
- ② 対話種別とOS種別を活かした、OSの差異を意識しない作業実行
- ③ 独自モジュールによる作業の繰り返しや条件分岐

## シナリオイメージ

### Movementの作成



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

- 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。

2

| 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 | Ansible利用情報                      |       | アクセス権                             |           |
|------------|------------|--------|----------------------------------|-------|-----------------------------------|-----------|
|            |            |        | ホスト指定形式                          | 並列実行数 | 設定                                | アクセス許可ロール |
| 自動入力       |            |        | <input type="button" value="▼"/> |       | <input type="button" value="設定"/> |           |

※\*は必須項目です。

戻る  2

| 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*               | アクセス権     | 備考                   |
|------|----------------------|-----------|----------------------|
| 自動入力 | <input type="text"/> | 設定        | <input type="text"/> |
|      | <input type="text"/> | アクセス許可ロール | <input type="text"/> |

\* \*は必須項目です。

2 戻る 登録

| 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 「ファイルを選択」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 | ▼     | ▼     | ファイルを選択 選択されていません<br>事前アップロード<br>アップロード状況: |

※\*は必須項目です。

戻る 登録

2

| 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に対話種別を登録する

Movementと対話種別を関連付けましょう。

メニュー: Ansible-Pioneer > Movement-対話種別紐付

- 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*            | 対話種別*                | 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。

2

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

戻る      登録

# 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 > 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。

The screenshot shows the 'Operation Registration' screen. A red box highlights the 'Operation Name' and 'Implementation Date' input fields. A red arrow points from this box to a detailed input dialog box labeled '2'. The dialog box contains two rows: '項目名' (Item Name) with 'オペレーション名' (Operation Name) and 'Pioneer\_Practice', and '実施予定日時' (Implementation Date) with '(任意の値をご入力下さい)' (Please enter any value). Both the main screen and the dialog box have their respective '登録' (Register) buttons highlighted.

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

※\*は必須項目です。

2

| 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 (1/2)

## 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。 (次項に続く)

| 項目                | Vyos仮想ルータ |
|-------------------|-----------|
| 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)    |
| 認証方式              | パスワード認証   |
| Pioneer利用情報/プロトコル | ssh       |
| Pioneer利用情報/OS種別  | vyos_RT   |

### 3.6 機器一覧への登録 (2/2)

#### 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。

#### 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。

The screenshot shows the '登録' (Registration) screen for a device. The form includes fields for HW機器種別 (HW device type), ホスト名 (Host name), IPアドレス (IP address), EtherWakeOnLan (EtherWakeOnLan), ログインユーザID (Login user ID), 管理 (Management), and ログインパスワード (Login password). A red box highlights the HW機器種別 and ログインユーザID fields. A red arrow points from the HW機器種別 field to a detailed configuration table below.

| 項目                | 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 | telnet     |
| Pioneer利用情報/OS種別  | 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)
- 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を選択

一意制約(複数項目) :

- パターン:
- 一意制約(複数項目)を選択

アクセス許可ロール :

- ロール:
- アクセス許可ロールを選択

1

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/オペレーションあり) |
| 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 項目名

| 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 作成

押下してメニュー作成を実行する。

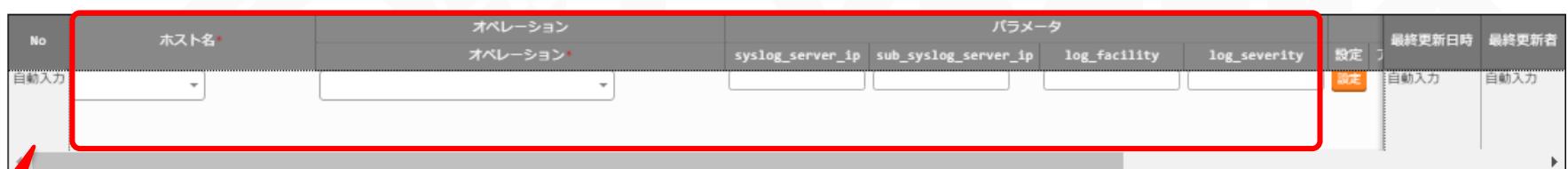
# 3.8 データの登録

## 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。



2

| 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。
- ② 下表のように関連付け設定を入力し、「登録」を押下する。

登録

| パラメータシート(From) |      | 登録方式         | Movement | IaC変数(To)               |                          |
|----------------|------|--------------|----------|-------------------------|--------------------------|
| 項目             | メニュー | 項目           | 登録方式     | Key変数                   | Value変数                  |
| 自動入力           | メニュー | メニューを選択して下さい | Movement | 変数名<br>Movementを選択して下さい | 代入順序<br>Movementを選択して下さい |

※\*は必須項目です。

2

| メニュー      | 項目                   | 登録方式   | Movement | Value変数<br>変数名       | Value変数<br>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 |    |    |    |      |                     |            |                  |       |       |    |                     |                        |
|---------|----|----|----|------|---------------------|------------|------------------|-------|-------|----|---------------------|------------------------|
| 一覧/更新   |    |    |    |      |                     |            |                  |       |       |    |                     |                        |
| 履歴      | 複製 | 更新 | 廃止 | 項目番号 | オペレーション名            | Movement名  | ホスト名             | 代入値管理 | アクセス権 | 備考 | 最終更新日時              | 最終更新者                  |
| 履歴      | 複製 | 更新 | 廃止 | 1    | 14:Pioneer_Practice | 19:ログサーバ設定 | 6:ky-labo-nwk-02 | 代入値管理 |       |    | 2021/08/30 09:31:19 | pioneer代入値自動登録設定プロシージャ |
| 履歴      | 複製 | 更新 | 廃止 | 2    | 14:Pioneer_Practice | 19:ログサーバ設定 | 7:vyos           | 代入値管理 |       |    | 2021/08/30 09:31:19 | pioneer代入値自動登録設定プロシージャ |

| 代入値管理 |    |    |    |      |                     |            |                  |                        |             |               |    |                     |                        |
|-------|----|----|----|------|---------------------|------------|------------------|------------------------|-------------|---------------|----|---------------------|------------------------|
| 一覧/更新 |    |    |    |      |                     |            |                  |                        |             |               |    | △閉じる                |                        |
| 履歴    | 複製 | 更新 | 廃止 | 項目番号 | オペレーション名            | Movement名  | ホスト名             | 変数名                    | Sensitive設定 | 具体値           | 代入 | 最終更新日時              | 最終更新者                  |
| 履歴    | 複製 | 更新 | 廃止 | 1    | 14:Pioneer_Practice | 19:ログサーバ設定 | 6:ky-labo-nwk-02 | 2:VAR_syslog_server_ip | OFF         | 127.0.0.1     |    | 2021/08/30 10:05:38 | pioneer代入値自動登録設定プロシージャ |
| 履歴    | 複製 | 更新 | 廃止 | 2    | 14:Pioneer_Practice | 19:ログサーバ設定 | 7:vyos           | 2:VAR_syslog_server_ip | OFF         | 127.0.0.1     |    | 2021/08/30 10:06:00 | pioneer代入値自動登録設定プロシージャ |
| 履歴    | 複製 | 更新 | 廃止 | 3    | 14:Pioneer_Practice | 19:ログサーバ設定 | 6:ky-labo-nwk-02 | 2:VAR_syslog_server_ip | OFF         | 10.197.19.172 |    | 2021/08/30 09:31:52 | pioneer代入値自動登録設定プロシージャ |
| 履歴    | 複製 | 更新 | 廃止 | 4    | 14:Pioneer_Practice | 19:ログサーバ設定 | 7:vyos           | 2:VAR_syslog_server_ip | OFF         | 10.197.17.224 |    | 2021/08/30 09:31:52 | pioneer代入値自動登録設定プロシージャ |
| 履歴    | 複製 | 更新 | 廃止 | 7    | 14:Pioneer_Practice | 19:ログサーバ設定 | 6:ky-labo-nwk-02 | 1:VAR_log_severity     | OFF         | info          |    | 2021/08/30 09:33:21 | pioneer代入値自動登録設定プロシージャ |
| 履歴    | 複製 | 更新 | 廃止 | 8    | 14:Pioneer_Practice | 19:ログサーバ設定 | 7:vyos           | 1:VAR_log_severity     | OFF         | info          |    | 2021/08/30 09:33:21 | pioneer代入値自動登録設定プロシージャ |
| 履歴    | 複製 | 更新 | 廃止 | 9    | 14:Pioneer_Practice | 19:ログサーバ設定 | 6:ky-labo-nwk-02 | 3:VAR_log_facility     | OFF         | local7        |    | 2021/08/30 09:33:32 | pioneer代入値自動登録設定プロシージャ |
| 履歴    | 複製 | 更新 | 廃止 | 10   | 14:Pioneer_Practice | 19:ログサーバ設定 | 7:vyos           | 3:VAR_log_facility     | OFF         | local7        |    | 2021/08/30 09:33:32 | pioneer代入値自動登録設定プロシージャ |

フィルタ結果件数: 8

# 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, a vertical menu lists various options like Main Menu, Movement, and Job Status Confirmation. The 'Job Execution' option is currently selected.

**Step 1:** A red callout points to the 'Movement [List]' section of the 'Movement' filter dialog. It contains a table with one item:

| 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 |

**Step 2:** A red callout points to the 'Operation [List]' section of the 'Operation' filter dialog. It also contains a table with one item:

| 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 | アクセス許可ロード | 2020/12/02 13:52:36 | pioneer 作業実行プロジェクト |

**Step 3:** A red callout points to the 'Execute' button in the bottom right corner of the main execution dialog. The dialog displays the selected Movement and Operation details.

**Buttons:** ドライラン (Dry Run) and 実行 (Execute).

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

Menu

- メインメニュー
- Movement一覧
- 対話種別リスト
- OS種別マスタ
- 対話ファイル素材集
- Movement-対話種別紐付
- 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

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                                 |         |          |   |                     |                     |   |       |    |
|              |                                                                                                                                                                                                       |                                          |         |          |   |                     |                     |   |       |    |
|              | <table border="1"><thead><tr><th>遅延タイム(分)</th><th>ホスト指定形式</th><th>IP</th></tr></thead><tbody><tr><td></td><td>Ansible利用情報</td><td></td></tr><tr><td></td><td>並列実行数</td><td></td></tr></tbody></table> | 遅延タイム(分)                                 | ホスト指定形式 | IP       |   | Ansible利用情報         |                     |   | 並列実行数 |    |
| 遅延タイム(分)     | ホスト指定形式                                                                                                                                                                                               | IP                                       |         |          |   |                     |                     |   |       |    |
|              | Ansible利用情報                                                                                                                                                                                           |                                          |         |          |   |                     |                     |   |       |    |
|              | 並列実行数                                                                                                                                                                                                 |                                          |         |          |   |                     |                     |   |       |    |
|              | <table border="1"><thead><tr><th>No.</th><th>名称</th><th>操作</th></tr></thead><tbody><tr><td>3</td><td>Pioneer_practice</td><td>確認</td></tr><tr><td>3</td><td></td><td>確認</td></tr></tbody></table>     | No.                                      | 名称      | 操作       | 3 | Pioneer_practice    | 確認                  | 3 |       | 確認 |
| No.          | 名称                                                                                                                                                                                                    | 操作                                       |         |          |   |                     |                     |   |       |    |
| 3            | Pioneer_practice                                                                                                                                                                                      | 確認                                       |         |          |   |                     |                     |   |       |    |
| 3            |                                                                                                                                                                                                       | 確認                                       |         |          |   |                     |                     |   |       |    |
| 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        | 入力データ                                                                                                                                                                                                 | <a href="#">InputData_000000103.zip</a>  |         |          |   |                     |                     |   |       |    |
| 出力データ        | 結果データ                                                                                                                                                                                                 | <a href="#">ResultData_000000103.zip</a> |         |          |   |                     |                     |   |       |    |
| 作業状況         | <table border="1"><thead><tr><th>予約日時</th><th>開始日時</th><th>終了日時</th></tr></thead><tbody><tr><td></td><td>2020/12/02 13:52:37</td><td>2020/12/02 13:52:46</td></tr></tbody></table>                    | 予約日時                                     | 開始日時    | 終了日時     |   | 2020/12/02 13:52:37 | 2020/12/02 13:52:46 |   |       |    |
| 予約日時         | 開始日時                                                                                                                                                                                                  | 終了日時                                     |         |          |   |                     |                     |   |       |    |
|              | 2020/12/02 13:52:37                                                                                                                                                                                   | 2020/12/02 13:52:46                      |         |          |   |                     |                     |   |       |    |

ログ

進行状況(実行ログ)

ファイルタグ:   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"

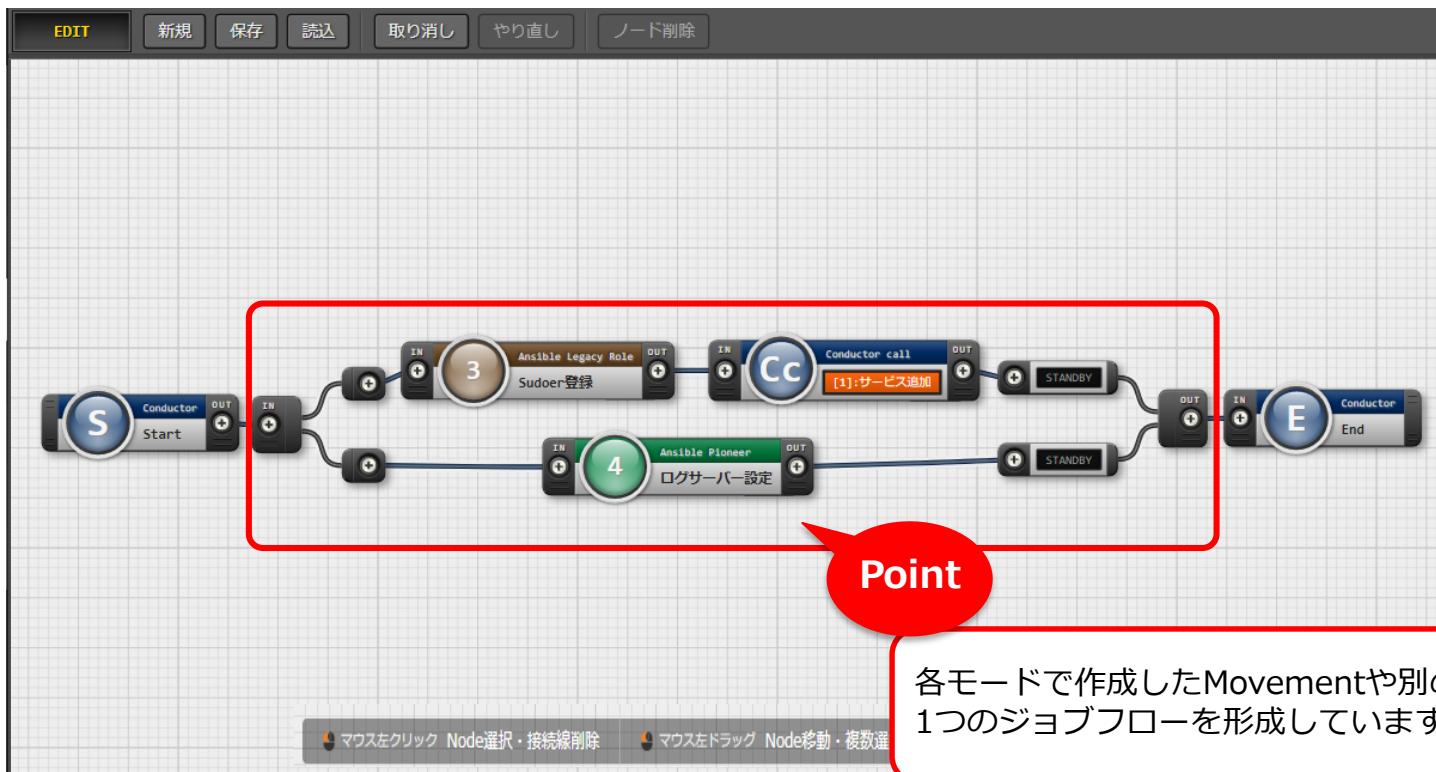
# A 付録



# 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**