



IT Automation

Ansible Driver 【実習編】

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

第1.0版
Exastro developer

目次

- ・はじめに
- ・下準備

① Ansible-Legacy

1. シナリオ
2. 事前準備
3. Movementの設定
4. Conductorの作成
5. オペレーションの登録
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・対象ホストの確認
10. 作業の実行

② Ansible-LegacyRole

1. シナリオ
2. 事前準備
3. Movementの設定
4. オペレーションの登録
5. パラメータシート作成
6. データの登録
7. 代入値自動登録設定
8. 代入値と対象ホストの確認
9. 作業の実行

③ Ansible-Pioneer

1. シナリオ
2. 事前準備
3. Movementの設定
4. オペレーションの登録
5. パラメータシート作成
6. データの登録
7. 代入値自動登録設定
8. 代入値・対象ホストの確認
9. 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。

下準備(次項)を終えた後、実習を開始して下さい。



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。

はじめに 作業環境

作業環境

本書で使用する作業環境は以下の通りです。

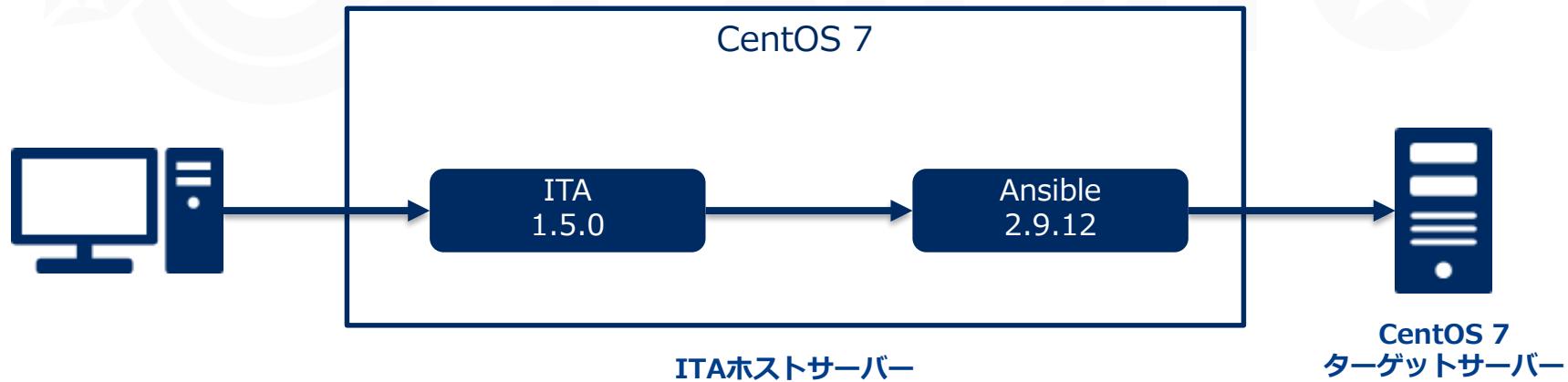
ITAホストサーバーとは別に、クライアントとなるサーバーをご用意ください。

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。

下準備 (1/2) 作業対象ホストの登録

作業対象ホストの登録

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	(任意の値をご設定下さい)
IPアドレス	対象機器のIPアドレス
ログインユーザID	(任意の値をご設定下さい)
管理	●
ログインパスワード	(任意の値をご設定下さい)
認証方式	ssh

下準備 (2/2) メニューグループの作成

メニューグループの作成

各章で作成するメニュー（パラメータシート）が所属するメニューグループを作成しましょう。

参照用に所属させるメニューグループも合わせて作成してください。

メニュー：管理コンソール > メニューグループ管理

- ① 登録 > 登録開始 を押下する。
- ② 各項目へ下記のように入力する。

登録

メニューグループID	メニューグループ名称*	表示順序	パネル用画像
自動入力			<input type="button" value="参照..."/> 事前アップロード アップロード状況:

※*は必須項目です。

メニューグループ名称	表示順序
AnsibleDriver実践用	110
AnsibleDriver実践用（参照用）	115

第1章 Ansible-Legacy編

1.1 シナリオ

シナリオ

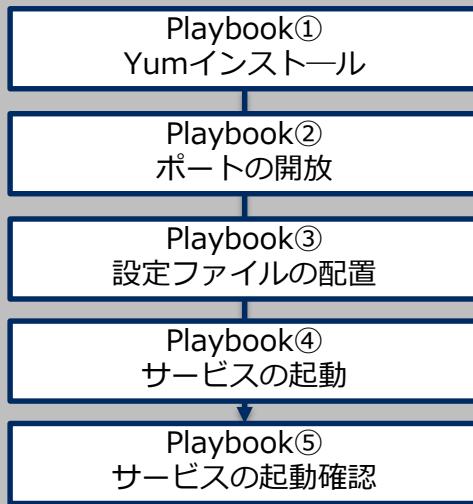
大別して3ステップから成る作業を実行し、**playbookの再利用性**を体感いただけます。

- ① Movementを組み合わせてConductorを作成する。
- ② メニューを作成し、代入値を登録する。
- ③ 作成したConductorを実行する。

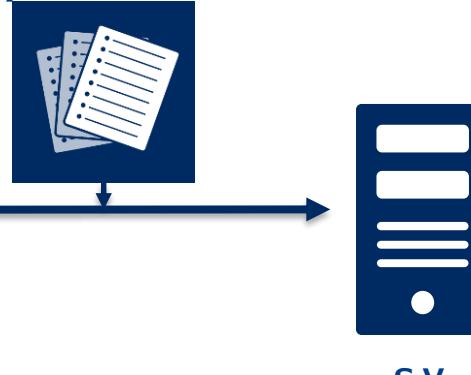
今回は「Apache・Tomcat」両サービスのインストールと起動を行います。

成果物イメージ

① Conductorを1つ作成する



② パラメータシートを作成 (playbookへ代入する値を登録)



③ 両サービスの追加を確認する ・Apache ・Tomcat



1.2 事前準備 playbookの用意 (1/2)

playbookの作成

本シナリオで使用するplaybookは以下の5つです。

【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 事前準備 playbookの用意 (2/2)

playbookの作成

```
- name: deploy 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 事前準備 httpd.confの用意

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	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。

[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="text"/>	<input type="button" value="参照..."/> 事前アップロード	<input type="text"/>	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。

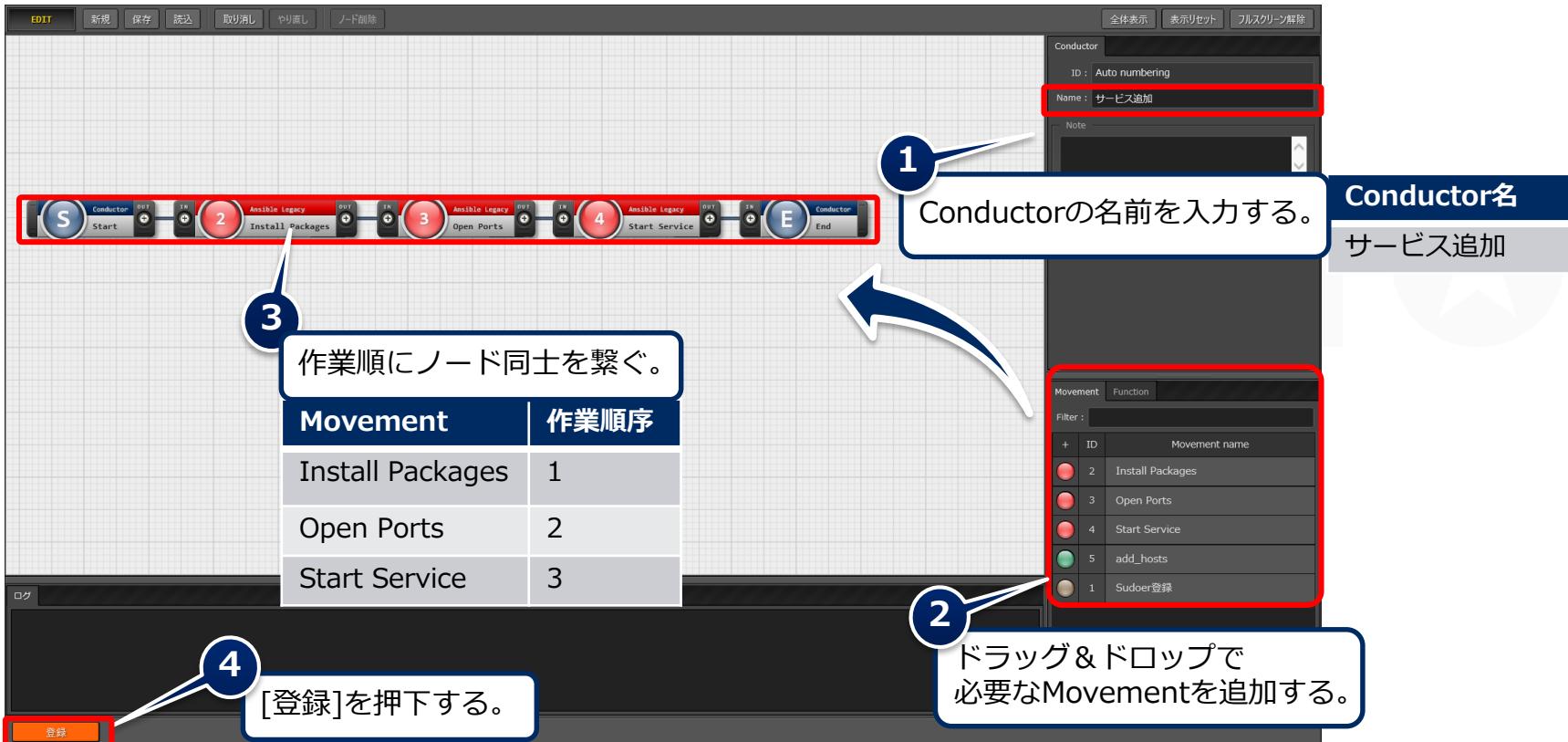
Point

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。

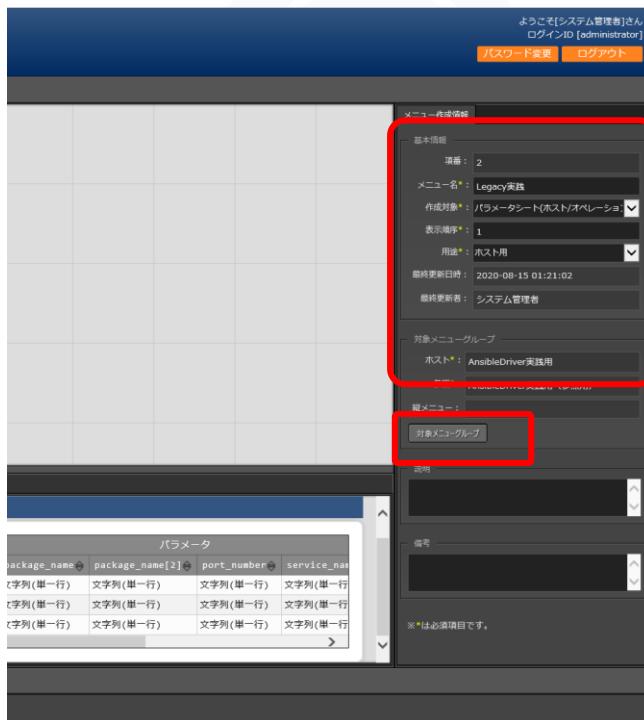
1.6 パラメータシートの作成 (1/3)

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

- ① 各項目へ下表のように入力する。
- ② [対象メニューグループ]を押下する。(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
用途	ホスト用

1.6 パラメータシートの作成 (2/3)

メニューを作成する

[メニューグループ選択]では、
作成するメニューが所属するメニューグループを選択します。

- ① 下図の様に選択する。
- ② 画面下部より[決定]を押下する。



1.6 パラメータシートの作成 (3/3)

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]を押し下し、新しい項目を追加する。
- ② 各項目について、下表のように入力する。
- ③ 画面下部の[作成]を押下する。

The screenshot shows the 'Items' tab selected in the top-left corner of the configuration interface (1). A large blue arrow points from the 'Items' tab to a table on the right, which lists the defined parameters. Below the table is a preview window showing a list of operations (3).

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

Below the table is a preview window titled '一覧(プレビュー)' (List Preview) showing a table of operations:

オペレーション				
No	ホスト名	オペレーション名	基準日時	実施予定日
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	ーション	2020/01/01 00:00	2020/01/01 00:00

The bottom of the preview window has a red box around the '作成' (Create) button (3).

1.7 パラメータの登録

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: AnsibleDriver 実践用 > 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

1.8 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-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。

The screenshot shows the 'Ansible-Legacy' configuration interface. At the top, there's a blue header bar with the text 'Ansible-Legacy' and 'Ansible Configuration'. Below it is a white main area with a dark blue navigation bar on the left containing icons for 'Ansible', 'Ansible-Legacy', 'Ansible Tower', 'Ansible Galaxy', 'Ansible Engine', and 'Ansible Playbook'. The main content area has a title '登録' (Registration). It contains several input fields and dropdown menus. A red box highlights the 'Parameter Sheet' section, which includes fields for 'メニュー名' (Menu Name), '項目' (Item), '登録方式' (Registration Method), 'Movement' (Movement), 'IaC変数' (IaC Variable), and 'Value変数' (Value Variable). Below this is a table with columns for 'メニュー' (Menu), '項目' (Item), '登録方式' (Registration Method), 'Movement' (Movement), 'Value変数' (Value Variable), and '代入順序' (Assignment Order). The table rows correspond to the items listed in the 'Parameter Sheet'.

※*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	
Legacy実践	service_name	Value型	Start Service	VAR_service_name	

1.9 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-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	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 19:08:36	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 19:07:40	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 19:08:48	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 19:08:48	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 19:08:59	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 19:08:59	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	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 10:10:34	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 19:03:33	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 19:03:56	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 10:10:34	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 10:10:56	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 10:12:04	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 10:12:37	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

1.10 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 menu, "Conductor[一覧]" is selected. A callout points to the "Conductor[一覧]" item with the text "実行するConductorを選択する。".
2. In the "Conductor[一覧]" screen, the "オペレーション[一覧]" tab is selected. A callout points to the "オペレーション[一覧]" tab with the text "オペレーションを選択する。".
3. At the bottom of the screen, the "実行" button is highlighted with a red box and a callout pointing to it 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。

1.10 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' screen. On the left, a horizontal workflow diagram is displayed with several nodes: 'Conductor Start' (S), 'Ansible Legacy: Install Packages' (green 'DONE' status), 'Ansible Legacy: Open Ports' (green 'DONE' status), 'Ansible Legacy: Start Services' (red 'RUNNING' status), and 'Conductor End'. A callout bubble labeled '1' points to the 'Ansible Legacy: Start Services' 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 view of the selected node is shown in a modal window. The window title is 'Node' and it displays the following information:

- Node type : movement
- Node instance ID : 13
- Node name : node-3
- Status : 正常終了 (Normal Completion)
- Start time : 2020/08/22 02:45:23
- End time : 2020/08/22 02:45:58
- Operation status : http://192.168.11.23/default/menu01_browse.php?no=2100020112&execution_no=7
- Specified individually operation
 - Operation ID :
 - Operation name :
- Note (Note field is empty)

A callout bubble labeled '2' points to the 'Operation status' section with the text 'リンクや作業の情報が表示される。' (Links and job information are displayed). At the bottom center, a blue circle contains the word 'Tips'. A callout bubble from this circle points to the URL 'http://192.168.11.23/default/menu01_browse.php?no=2100020112&execution_no=7' with the text '以下のURLにて、Apache及びTomcatのインストールを確認できます。' (You can confirm the installation of Apache and Tomcat via the following URL). Below this, another callout bubble points to the URLs 'Apache- http://(ホストのIPアドレス):80' and 'Tomcat- http://(ホストのIPアドレス):8080'.

1.10 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。

三 Menu		
説明		
対象作業		
項目		値
作業No.		19
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
Movement	ID	2
	名称	<button>Install Packages</button>
	運延タイム(分)	
	Ansible利用情報	ホスト指定形式 WinRM接続
オペレーション	No.	2
	名称	Apache Install
	ID	2
作業対象ホスト		<button>確認</button>
代入値		<button>確認</button>
入力データ	投入データ	InputData_0000000019.zip
出力データ	結果データ	ResultData_0000000019.zip
予約日時		
作業状況	開始日時	2020/08/28 19:43:18
	終了日時	2020/08/28 19:43:34

```
進行状況(実行ログ)

フィルタ: [ ]  該当行のみ表示

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。

第2章 Ansible-LegacyRole編

2.1 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。

そこで今回は、

Ansible GalaxyからダウンロードしたRoleをITAに登録し、実行するまでを体験いただきます。

以下のURLからRoleを取得してください。



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

本ロールは/etc/sudoers.d配下にファイルを追加することができます。

シナリオイメージ



2.2 事前準備 (1/4)

事前準備のまえに

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:  
# - defaults: env_reset  
# - name: user1  
# _ defaults: requiretty  
# sudo_users:  
# - name: '%group1'  
# - name: 'bar'  
# nopasswd: yes  
~~~~~ (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
```

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 事前準備 (2/4) ITAreadme

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

読替表については次項で説明します。

2.2 事前準備 (3/4) 讀替表

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:

**読替表**  
LCA\_sudo\_users: `sudo_users`  
LCA\_sudo\_sudoers\_file: `sudo_sudoers_file`

### 実際に利用される変数

`sudo_users:`  
- name: example\_name  
`sudo_sudoers_file: example_sudoers`

#### 代入値管理

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

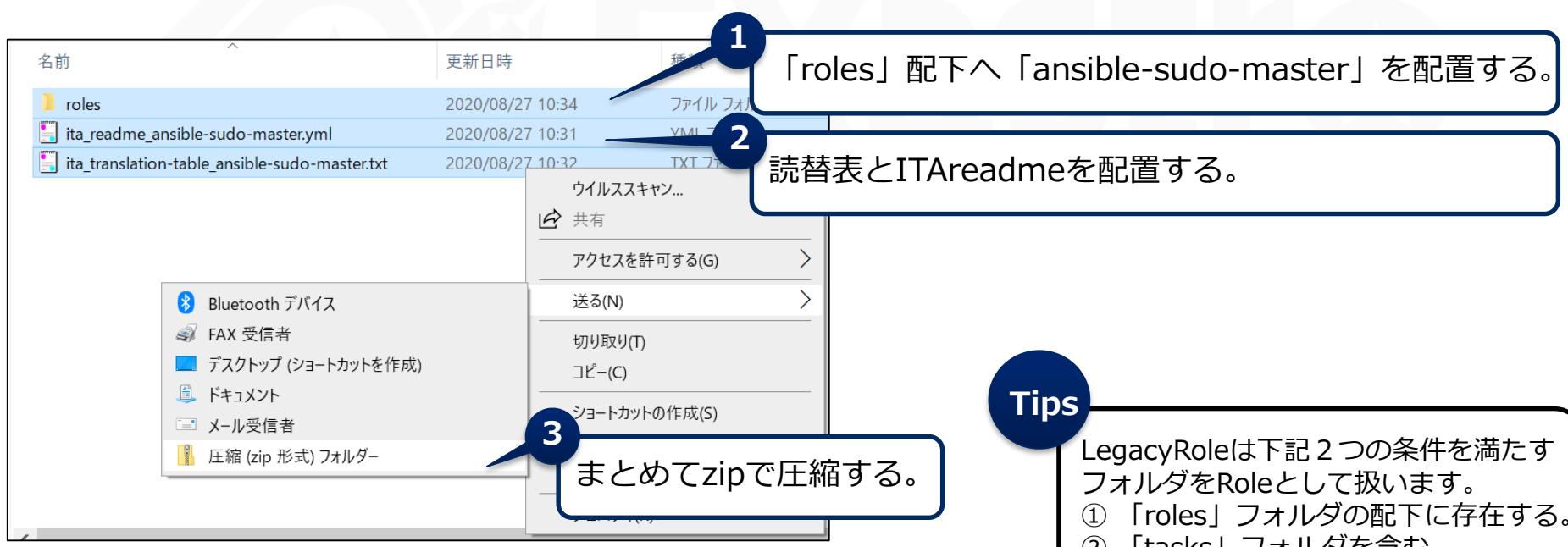
## 2.2 事前準備 (4/4) zipファイルを作成する

### 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 パラメータシート作成(1/3)

## 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

- ① 各項目へ下表のように入力する。
- ② [対象メニューグループ]を押下する。(次項へ)

メニュー作成情報

項目名: 2  
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: 1  
用途\*: ホスト用  
最終更新日時: 2020-08-15 01:21:02  
最終更新者: システム管理者

対象メニューグループ  
ホスト\*: AnsibleDriver実践用

対象メニュー

説明

権限

※\*は必須項目です。

| 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                           |
| 用途    | ホスト用                        |

## 2.5 パラメータシート作成(2/3)

### メニューを作成する

[メニュー グループ選択]では、  
作成するメニューが所属するメニュー グループを選択します。

- ① 下図の様に選択する。
- ② 画面下部より[決定]を押下する。



## 2.5 パラメータシート作成(3/3)

### 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]を押し下し、新しい項目を追加する。
- ② 各項目へ下表のように入力する。
- ③ 画面下部の[作成]を押下する。



| 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     |

## 2.6 パラメータの登録

### 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: AnsibleDriver 実践用 > 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                |                      |
|------|----------------------|----------------------|----------------------|----------------------|
|      |                      | 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"/> |
|      |                      |                      |                      |                      |

### 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 |

## 2.7 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            |              |                  |                |
|------|--------------|------------------|----------|------------------|--------------|------------------|----------------|
|      | メニュー名        | 項目               |          | Movement         | key変数<br>変数名 | 代入順序             | Value変数<br>変数名 |
| 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 | Movementを選択して下さい | Movement     | Movementを選択して下さい | Movement       |

\*は必須項目です。

| メニュー名         | 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 |                    |

# 2.8 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 |    |    |                       |            |            |                         |            |                 |      |    |                     |                           |
|-------|----|----|-----------------------|------------|------------|-------------------------|------------|-----------------|------|----|---------------------|---------------------------|
| 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

# 2.9 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. It consists of three main sections:

- Step 1:** A list of Movements. One entry is highlighted: "1 Sudoer登録 Ansible Legacy Role". This step is labeled "実行するMovementを選択する。" (Select the Movement to execute).
- Step 2:** A list of Operations. One entry is highlighted: "1 LegacyRole\_Practice 2020/08/21 17:20 2020/08/21 17:18". This step is labeled "オペレーションを選択する。" (Select the Operation).
- Step 3:** A confirmation screen for executing Movement ID 1. The "実行" (Execute) button is highlighted with a red box. This step is labeled "「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。

# 2.9 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 menu with options like 'Menu', 'Movement', 'Ansible Configuration', 'Job Status Confirmation', and 'Job Management'. The main area has tabs for 'Explanation' and 'Target Job'. The 'Target Job' tab is active, displaying 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      | <table border="1"><tr><td>ID</td><td>1</td></tr><tr><td>名称</td><td>Sudoer登録</td></tr><tr><td>遅延タイム(分)</td><td></td></tr><tr><td>Ansible利用情報</td><td>ホスト指定形式 IP</td></tr><tr><td>WinRM接続</td><td></td></tr></table> | 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 |  |
| 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       | <table border="1"><tr><td>No.</td><td>1</td></tr><tr><td>名称</td><td>LegacyRole_Practice</td></tr><tr><td>ID</td><td>1</td></tr></table>                                                                            | No.  | 1                   | 名称   | LegacyRole_Practice | ID       | 1 |             |            |         |  |
| No.           | 1                                                                                                                                                                                                                  |      |                     |      |                     |          |   |             |            |         |  |
| 名称            | LegacyRole_Practice                                                                                                                                                                                                |      |                     |      |                     |          |   |             |            |         |  |
| ID            | 1                                                                                                                                                                                                                  |      |                     |      |                     |          |   |             |            |         |  |
| 作業対象ホスト       | <table border="1"><tr><td>確認</td></tr></table>                                                                                                                                                                     | 確認   |                     |      |                     |          |   |             |            |         |  |
| 確認            |                                                                                                                                                                                                                    |      |                     |      |                     |          |   |             |            |         |  |
| 代入値           | <table border="1"><tr><td>確認</td></tr></table>                                                                                                                                                                     | 確認   |                     |      |                     |          |   |             |            |         |  |
| 確認            |                                                                                                                                                                                                                    |      |                     |      |                     |          |   |             |            |         |  |
| 入力データ         | 投⼊データ                                                                                                                                                                                                              |      |                     |      |                     |          |   |             |            |         |  |
| 出力データ         | 結果データ                                                                                                                                                                                                              |      |                     |      |                     |          |   |             |            |         |  |
| 予約日時          |                                                                                                                                                                                                                    |      |                     |      |                     |          |   |             |            |         |  |
| 作業状況          | <table border="1"><tr><td>開始日時</td><td>2020/08/22 03:27:40</td></tr><tr><td>終了日時</td><td>2020/08/22 03:28:16</td></tr></table>                                                                                     | 開始日時 | 2020/08/22 03:27:40 | 終了日時 | 2020/08/22 03:28:16 |          |   |             |            |         |  |
| 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' window. It contains a search bar and a scrollable log area. 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 シナリオ

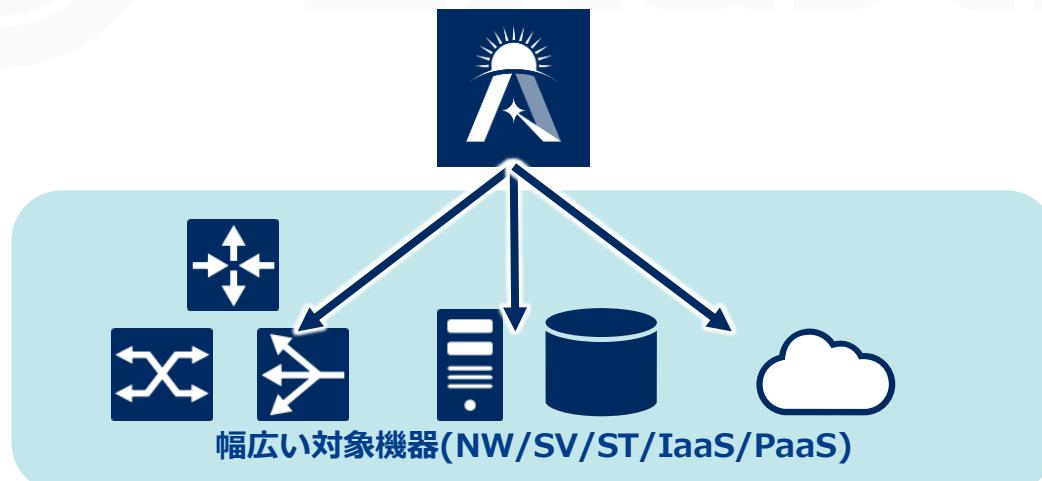
## シナリオ

Pioneer独自モジュールを用い、  
対象ホストの/tmp/hostsを対象に、ホスト追加作業を実施します。

## 備考

今回は便宜上、CentOS7を対象に作業を行いますが、  
Pioneerは**対象機器の幅広さ**を長所にもつモードです。

サーバ、ストレージ、ネットワークを問わず、Telnet・SSHでログイン可能なあらゆる機器  
に対応しており、実務ではより柔軟な対応力を発揮します。



## 3.2 事前準備

### 対話ファイルを用意する

本シナリオは、対話ファイルを1つ作成・実行するまでを目標としています。

今回使用するファイルの内容は以下の通りです。

※書式やモジュールについての詳細は[マニュアル](#)を参照して下さい。

【注意】文字コードは“UTF-8”、ファイル形式は“yml”を使用してください

ファイル名: add\_hosts.yml

```
conf:  
  timeout: 20  
  
exec_list:  
  - expect: 'password'  
    exec: '{{ __loginpassword__ }}'  
  
  - command: 'cat /tmp/hosts'  
    prompt: root@{{ __loginhostname__ }}  
    register: result_stdout  
    when:  
      - VAR_hosts_ip is define  
      - VAR_hosts_name is define  
  
  - command: 'echo {{ item.0 }} {{ item.1 }} >> /tmp/hosts'  
    prompt: 'root@{{ __loginhostname__ }}'  
    when:  
      - VAR_hosts_ip is define  
      - VAR_hosts_name is define  
with_items:  
  - '{{ VAR_hosts_ip }}'  
  - '{{ VAR_hosts_name }}'  
exec_when:  
  - result_stdout no match('{{ item.0 }} *{{ item.1 }}')
```

1

“register”を用い、  
‘cat /tmp/hosts’ の標準出力を登録しています。

2

“with\_items”を用いた繰り返し処理を定義しています。

3

“exec\_when”を用いた分岐処理を定義しています。  
判定は繰り返しごとに行われます。

この場合、登録された出力中に  
指定文字列が存在しないときに処理を行います。

### 3.3 Movementの設定 (1/6)

#### 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。

**戻る** **登録**

| 項目        | 入力内容      |
|-----------|-----------|
| Movement名 | add_hosts |
| ホスト指定形式   | IP        |

### 3.3 Movementの設定 (2/6)

#### 「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。

| 項目名   | 入力内容    |
|-------|---------|
| OS種別名 | CentOS7 |
| SV    | ●       |

### 3.3 Movementの設定 (3/6)

#### 作成した「OS種別」を機器一覧の情報に追加する

機器一覧に登録したホストについて、OS種別の情報を追記しましょう。

メニュー：基本コンソール > 機器一覧

- ① [filtrタ]を押下する。
- ② 作業対象ホストの列から、[更新]を押下する。
- ③ 各項目で下図の通り選択し、[更新]を押下する。

The screenshot shows the 'Machine List' screen with a red box highlighting the 'Protocol' and 'OS Type' dropdowns in the 'Ansible Utilization Information' section for the selected host.

| 項目    | 入力内容    |
|-------|---------|
| プロトコル | ssh     |
| OS種別  | CentOS7 |

### 3.3 Movementの設定 (4/6)

#### 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 |

※\*は必須項目です。

| 項目名   | 入力内容    |
|-------|---------|
| 対話種別名 | hosts追加 |

### 3.3 Movementの設定 (5/6)

#### 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。

#### 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"/> | <input type="button"/> | <input type="button"/> 参照...<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。

| 項目       | 入力内容          |
|----------|---------------|
| 対話種別     | hosts追加       |
| OS種別     | CentOS7       |
| 対話ファイル素材 | add_hosts.yml |

### 3.3 Movementの設定 (6/6)

#### 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

| 紐付項目番 | 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 | add_hosts |
| 对话種別     | hosts追加   |
| インクルード順序 | 1         |

# 3.4 オペレーションの設定

## 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。

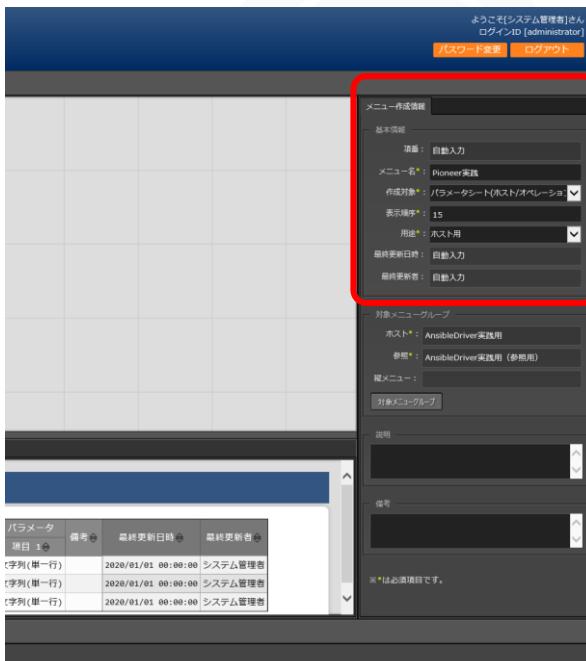
# 3.5 パラメータシート作成 (1/3)

## 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

- ① 各項目へ下表のように入力する。
- ② [対象メニューグループ]を押下する。(次項へ)



メニュー作成画面

項目: 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: 15  
用途: ホスト用  
最終更新日時: 自動入力  
最終更新者: 自動入力

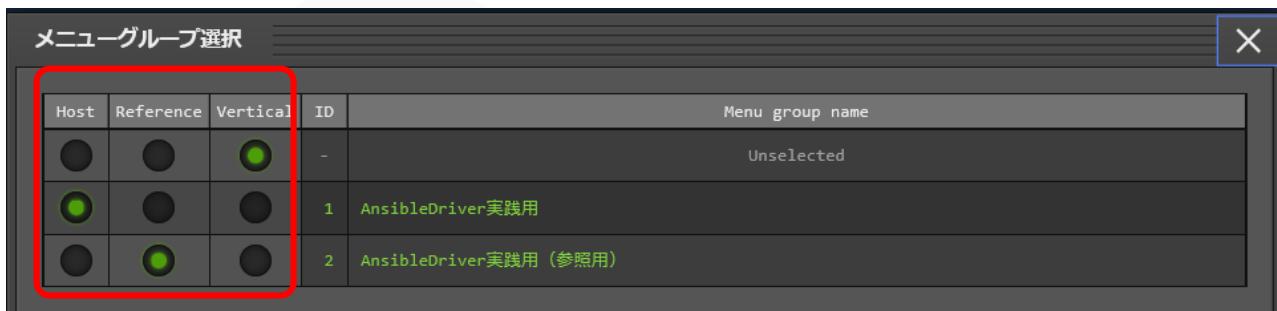
| 項目    | 入力内容                    |
|-------|-------------------------|
| メニュー名 | Pioneer実践               |
| 作成対象  | パラメータシート(ホスト/オペレーション含む) |
| 表示順序  | 3                       |
| 用途    | ホスト用                    |

# 3.5 パラメータシート作成 (2/3)

## メニューを作成する

[メニューグループ選択]では、  
作成するメニューが所属するメニューグループを選択します。

- ① 下図の様に選択する。
- ② 画面下部より、[決定]を押下する。



### 3.5 パラメータシート作成 (3/3)

#### 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]を押し下し、新しい項目を追加する。
- ② 各項目について、下表のように入力する。
- ③ 画面下部の[作成]を押下する。

1 リピート 取り消し やり直し

2

| 項目名               | 入力方式     | 最大バイト数 |
|-------------------|----------|--------|
| hosts_ip          | 文字列(單一行) | 32     |
| hosts_ip_second   | 文字列(單一行) | 32     |
| hosts_name        | 文字列(單一行) | 32     |
| hosts_name_second | 文字列(單一行) | 32     |

3 作成

| 項目名               | 入力方式     | 最大バイト数 |
|-------------------|----------|--------|
| hosts_ip          | 文字列(單一行) | 32     |
| hosts_ip_second   | 文字列(單一行) | 32     |
| hosts_name        | 文字列(單一行) | 32     |
| hosts_name_second | 文字列(單一行) | 32     |

# 3.6 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: **AnsibleDriver 実践用 > 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。

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 | オペレーション | hosts_ip | hosts_ip(2) | hosts_name | hosts_name(2) |
| 自動入力 |       |         |         |          |             |            |               |
|      |       |         |         |          |             |            |               |
|      |       |         |         |          |             |            |               |

| ホスト名     | オペレーション          | hosts_ip      | hosts_ip_second | hosts_name   | hosts_name_second |
|----------|------------------|---------------|-----------------|--------------|-------------------|
| (対象のホスト) | Pioneer_practice | 192.168.10.10 | 192.168.20.5    | www.test.com | www.pioneer.com   |

# 3.7 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 |            |              | Iac変数 |          |                         |                          |
|----------|------------|--------------|-------|----------|-------------------------|--------------------------|
| 項目番      | メニュー名      | 項目           | 登録方式* | Movement | key変数                   | Value変数                  |
| 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 | 登録方式  | 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 |
|          |            |              |       |          | 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 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 | hosts_ip          | Value型 | add_hosts | VAR_hosts_ip   | 1    |
| Pioneer実践 | hosts_ip_second   | Value型 | add_hosts | VAR_hosts_ip   | 2    |
| Pioneer実践 | hosts_name        | Value型 | add_hosts | VAR_hosts_name | 1    |
| Pioneer実践 | hosts_name_second | Value型 | add_hosts | VAR_hosts_name | 2    |

# 3.8 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

| 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  | 4:Pioneer_Practice | 5:add_hosts | 1:ita-test |    | 2020/08/31 13:30:11 | pioneer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

| 更新 | 廃止 | 項目 | 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 | 1  | 4:Pioneer_Practice | 5:add_hosts | 1:ita-test | 1:VAR_hosts_ip   | 192.168.10.10   | 1    |    | 2020/08/31 13:30:22 | pioneer代入値自動登録設定プロシージャ |
| 更新 | 廃止 | 2  | 4:Pioneer_Practice | 5:add_hosts | 1:ita-test | 2:VAR_hosts_name | www.test.com    | 1    |    | 2020/08/31 13:30:22 | pioneer代入値自動登録設定プロシージャ |
| 更新 | 廃止 | 3  | 4:Pioneer_Practice | 5:add_hosts | 1:ita-test | 1:VAR_hosts_ip   | 192.168.20.5    | 2    |    | 2020/08/27 11:14:33 | pioneer代入値自動登録設定プロシージャ |
| 更新 | 廃止 | 4  | 4:Pioneer_Practice | 5:add_hosts | 1:ita-test | 2:VAR_hosts_name | www.pioneer.com | 2    |    | 2020/08/27 11:15:18 | pioneer代入値自動登録設定プロシージャ |

フィルタ結果件数: 4

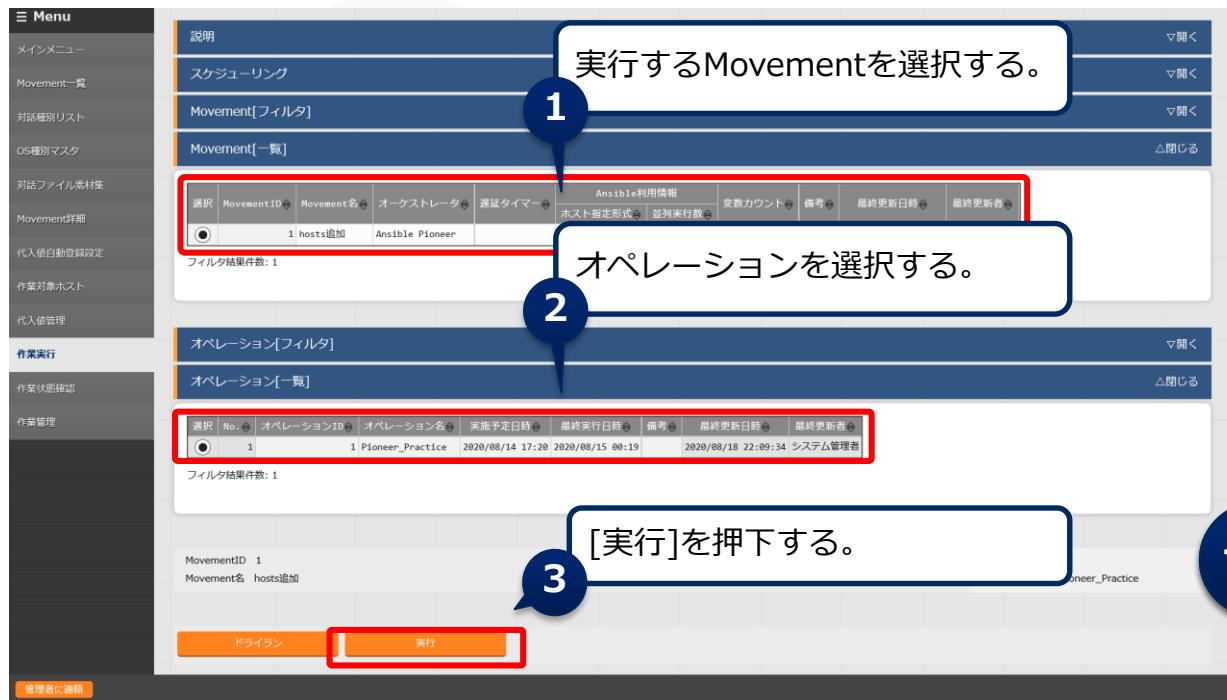
# 3.9 作業の実行 (1/2)

## 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 'Movement' selection screen. A callout bubble labeled '1' points to the 'Movement[一覧]' tab, with the text '実行するMovementを選択する。'. Another callout bubble labeled '2' points to the 'オペレーション[一覧]' tab, with the text 'オペレーションを選択する。'. A third callout bubble labeled '3' points to the '実行' button at the bottom of the screen, with the text '[実行]を押下する。'.

説明  
スケジューリング  
Movement[フィルタ]  
Movement[一覧]

選択 MovementID: 1 Movement名: hosts追加 オーケストレータ: Ansible Pioneer 遅延タイマー: Ansible利用情報  
ホスト割定形式: 並列実行数: 变数カウント: 備考: 最終更新日時: 最終更新者:

1 hosts追加 Ansible Pioneer

1 フィルタ結果件数: 1

オペレーション[フィルタ]  
オペレーション[一覧]

選択 No. 1 オペレーションID: 1 オペレーション名: Pioneer\_Practice 実施予定日時: 2020/08/14 17:20 最終実行日時: 2020/08/15 00:19 備考: 最終更新日時: 2020/08/18 22:09:34 最終更新者: システム管理者

1 フィルタ結果件数: 1

MovementID: 1  
Movement名: hosts追加

ドライラン 実行

### Tips

実行後、自動で【作業確認】へ  
画面遷移します。

# 3.9 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 > 作業実行

The screenshot shows the 'Job Execution' screen with a red box highlighting the 'Job Details' section. This section contains various parameters such as Job ID, Type, Status, Engine, and User. It also includes a 'Movement' section with 'add\_hosts' and 'Ansible' configuration, and an 'Operation' section with 'Pioneer\_Practice'. At the bottom, there are two orange '确认' (Confirm) buttons.

The screenshot shows the 'Execution Log' and 'Error Log' sections. The 'Execution Log' contains the following Ansible command output:

```
"module_args": { "exec_file": "/exastro/data_relay/storage/ansible_driver/pioneer/ns/0000000007/bin/dialog_files/ita-test/0000000001-Pioneer", "exec_args": "-c", "grep_shell": "grep", "grep_symbol": "\_+", "host_vars_file": "/tmp/original_host_vars/ita-test", "ignore_errors": "no", "log_file_dir": "/exastro/data_relay/storage/ansible_driver/pioneer/ns/0000000007/tmp", "proto": "ssh", "script": "test.sh", "skip": "no", "use_sshkey": "no", "username": "root" }, "msg": "192.168.11.11: normal exit" } META: ran handlers META: ran handlers
```

The 'Error Log' section is currently empty.

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、/tmp/hostsをご参照ください。

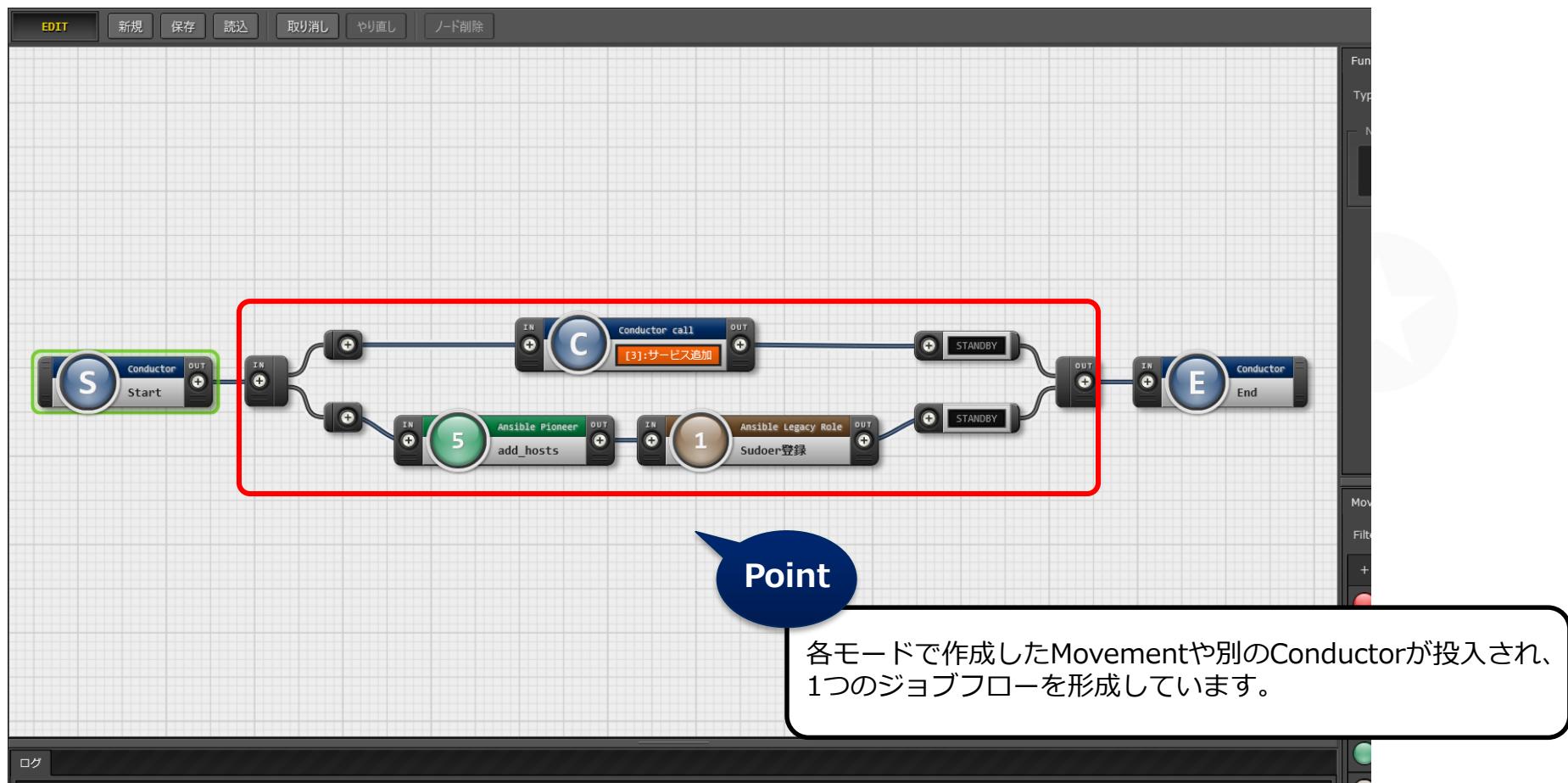
# A 付録



# 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1つのジョブフローとして作業を実行することもできます。





**Exastro**