



# IT Automation

## クイックスタート

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

# 目次

1. はじめに
  - 1.1 [はじめに](#)
  - 1.2 [本書のシナリオと作業範囲の位置づけ](#)
  - 1.3 [各種用語の説明](#)
  
2. 画面説明
  - 2.1 [Webコンソール画面\(ログイン\)](#)
  - 2.2 [画面説明\(メインメニュー\)](#)
  - 2.3 [画面説明\(各メニュー\)](#)
  
3. 実行前準備
  - 3.1 [Playbookをアップロードしてジョブ\(Movement\)に紐付け](#)
  - 3.2 [ジョブ\(Movement\)をジョブフロー\(Conductor\)に組み込み](#)
  - 3.3 [CMDBにパラメータシートを設定](#)
  - 3.4 [パラメータシートの項目とPlaybookの変数の紐付け](#)
  
4. 実行操作(1回目)
  - 4.1 [機器一覧にターゲットとなるLinuxマシンを登録](#)
  - 4.2 [作業名\(Operation\)の登録](#)
  - 4.3 [パラメータシートにデータを登録](#)
  - 4.4 [ジョブフロー\(Conductor\)の実行](#)
  - 4.5 [実行結果の確認](#)

# 目次

## 5. 実行操作(2回目)

- 5.1 [作業名\(Operation\)の登録](#)
- 5.2 [パラメータシートにデータを登録](#)
- 5.3 [ジョブフロー\(Conductor\)の実行](#)
- 5.4 [実行結果の確認](#)

## 6. CMDBパラメータの履歴確認

- 6.1 [作業実行と履歴管理](#)
- 6.2 [CMDBパラメータの履歴を確認する](#)

## A 付録

- 参考① [【Ansible-Legacy】単体実行](#)
- 参考② [【Ansible-Legacy】実行確認](#)

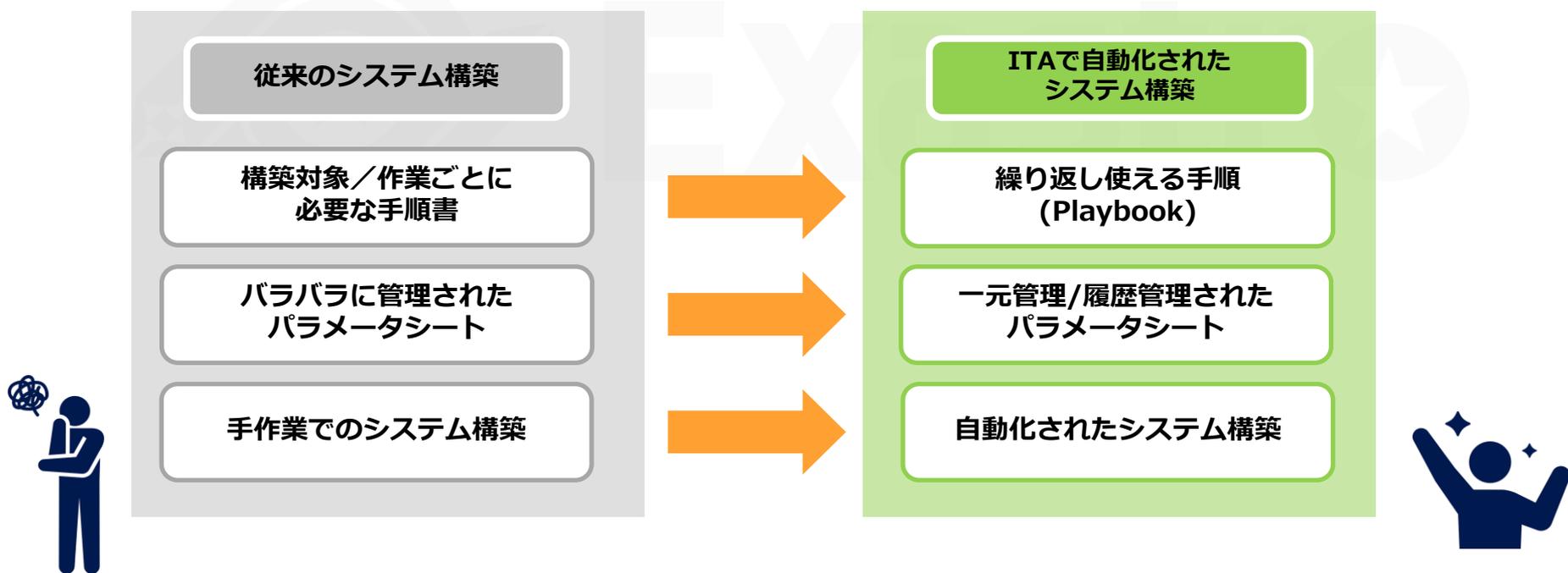
# 1. はじめに



# 1.1 はじめに (1/2)

本書は、はじめて IT Automation (以下ITAと記載) に触れるユーザが、ITAのインタフェースをスムーズに体感できるクイックスタートの手順書としてご活用できます。

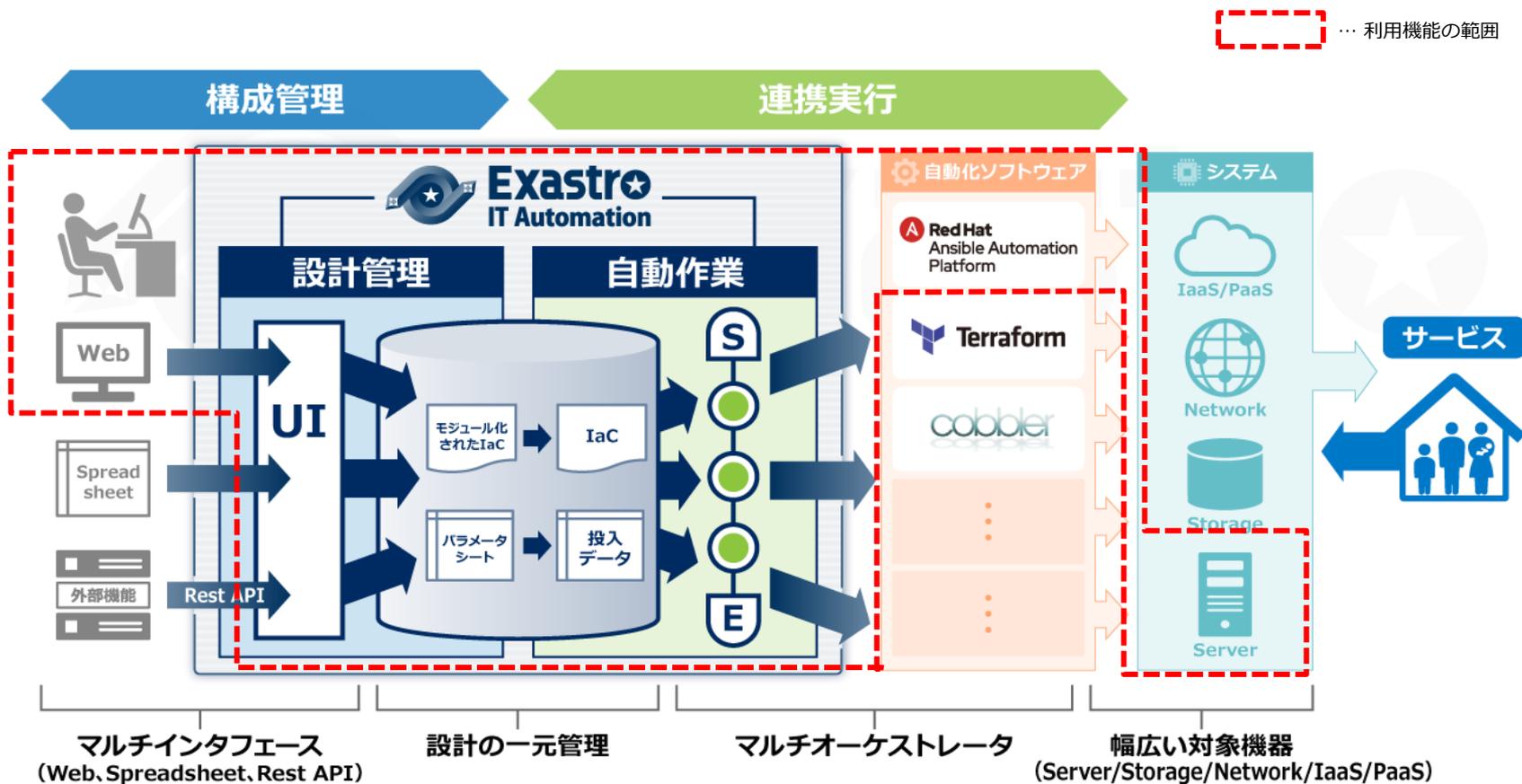
システム構築においてよくあるLinuxサーバのパッケージのインストール作業を通して、構築対象サーバごとの作業とパッケージ管理を自動化・一元管理化を行い、従来のシステム構築とは異なるITAを使用した効率的なシステム構築を体感できます。



# 1.1 はじめに (2/2)

## 本クイックスタートで体感できる主なITA機能の範囲

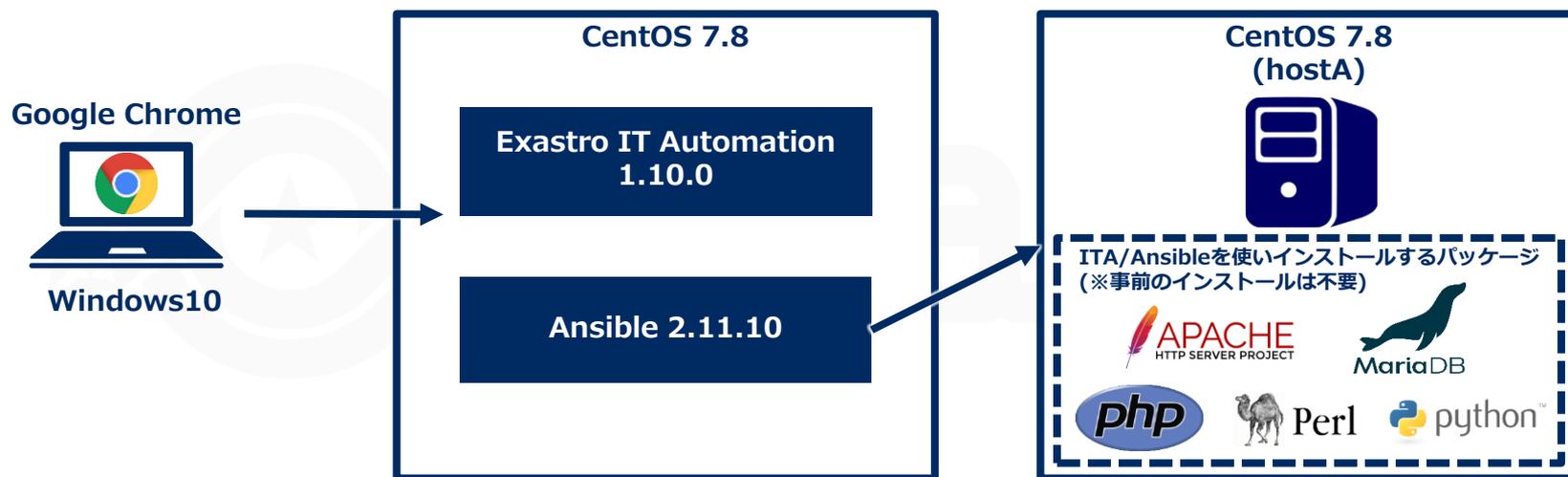
- ・ 自動化ソフトウェア(Ansible)との連携
- ・ パラメータ管理(メニュー作成・登録・履歴管理等)
- ・ 変数紐付け(代入値自動登録)



## 1.2 本書のシナリオと作業範囲の位置づけ (1/3)

本シナリオではAnsibleドライバを使用し、Linuxサーバ構築で実施するyumパッケージのインストール作業を構築対象サーバごとにパラメータ管理し、構築作業の自動化を行う内容となっています。

### 作業環境



### 使用するシステム

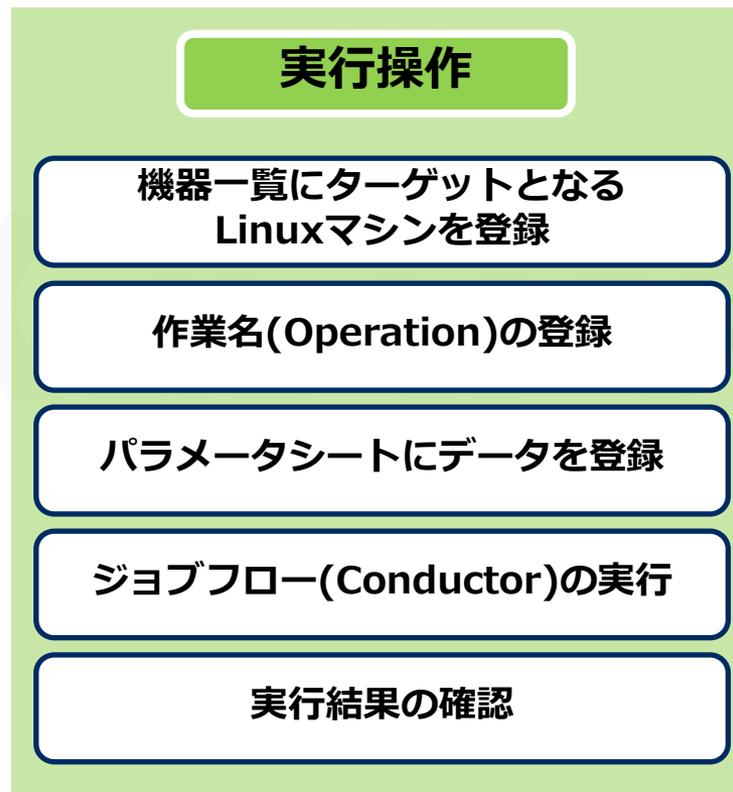
- Exastro IT Automation 1.10.0
- CentOS Linux 7.8(ITAサーバ用)
- CentOS Linux 7.8(ターゲットマシン用)
- Windows 10(クライアント)
- Google Chrome (Win10側)



## 1.2 本書のシナリオと作業範囲の位置づけ (3/3)

インストール後からAnsible-Legacyを実行するまでのシナリオ

- シナリオと、開発者(実行前準備)／作業員(実行操作)の作業範囲については以下の通り。



☆  
繰  
り  
返  
し  
実  
行  
☆

## 1.3 各種用語の説明

### 本シナリオに登場する主な用語

用語	説明
Playbook	定型業務をタスクで記述し、Ansibleに実行させるためのファイルです。YAML形式で使用します。
Ansible-Legacy	ITA から Ansible を利用する機能です。Legacy コンソールでは、構築コードとして単体のYAMLファイルを使う場合に使用します。
オペレーション名 (Operation)	ITA での作業実行単位。作業予定、実行履歴などを管理することができます。
Conductor	ITA での一連の作業の単位。オペレーション名と関連付けて実行します。Node と呼ぶ各種パーツを組み合わせて、ジョブフローを作成し、複数の機器に対して、一連の構築・設定などの作業を行います。
Movement	各機器に対する構築ツールを使った構築、設定などの作業の単位です。

Exastro ITAについてのより詳しい説明は[ドキュメントサイト](#)をご活用ください。

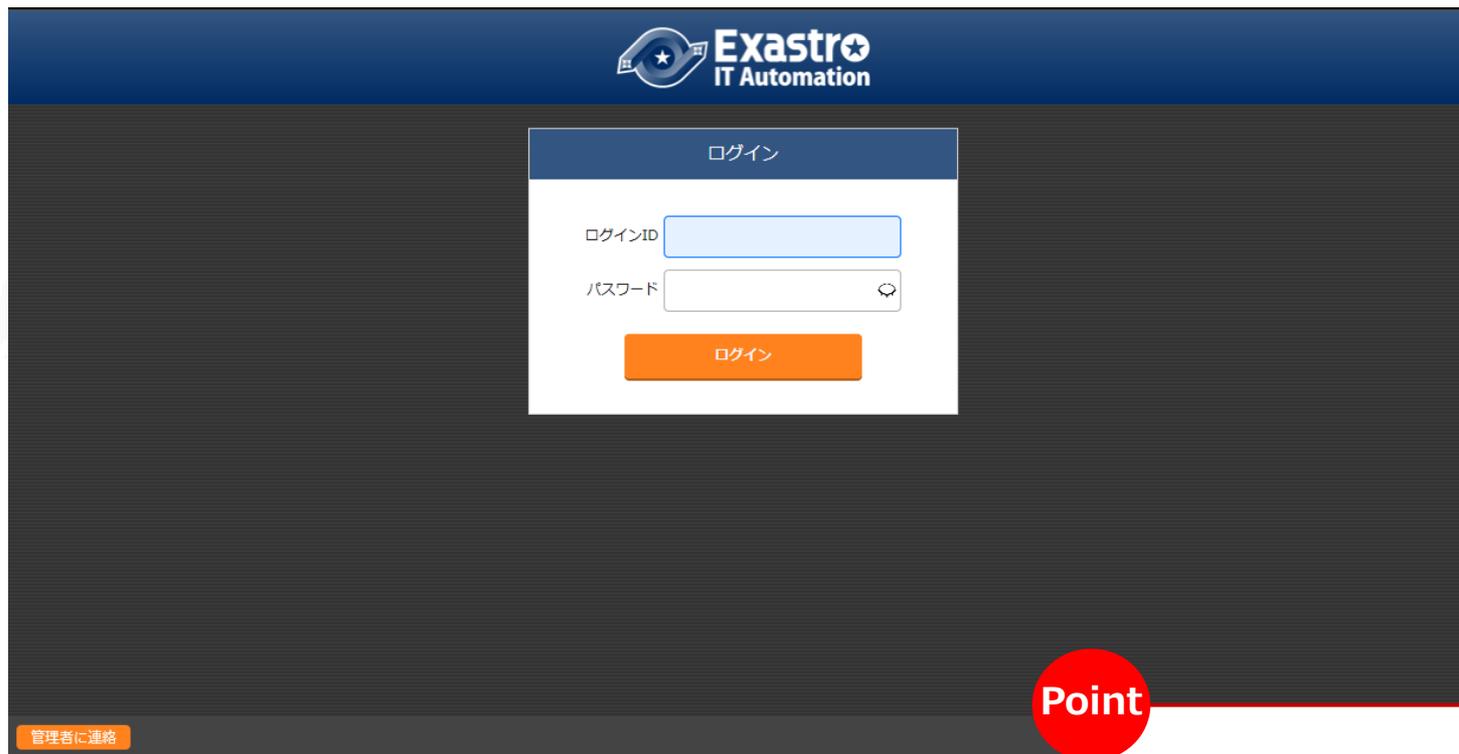
## 2. 画面説明



## 2.1 Webコンソール画面（ログイン）

### Webコンソールログイン

- ITAのインストールが完了しURLへアクセスすると、ログイン画面が表示されます。  
※インストール手順については” [IT Automation オンラインインストール](#)”をご参照ください。



ログイン

ログインID

パスワード

ログイン

管理者に連絡

#### Point

初回ログイン時は、ログイン直後にパスワード変更を求められます。

## 2.2 画面説明(メインメニュー)

### 画面説明 (メインメニュー)

- 基本的な名称は以下の通りです。

ようこそ[システム管理者]さん  
ログインID [administrator]

ロール パスワード変更 ログアウト

Menu DASHBOARD

メインメニュー

システム設定

メニューグループ管理

メニュー管理

ロール管理

ユーザ管理

ロール・メニュー紐付管理

ロール・ユーザ紐付管理

シーケンス管理

SSO基本情報管理

SSO属性情報管理

バージョン確認

管理者に連絡

メニューグループ

管理コンソール 基本コンソール エクスポート/… Symphony Conductor メニュー作成

比較 ホストグル… Ansible共通 Ansible-Legacy Ansible-Pioneer Ansible-Lega…

Terraform CI/CD for IaC

Movement

作業状況

作業結果

作業履歴

メニュー

メニューグループ

## 2.3 画面説明(各メニュー) (1/2)

### 画面説明 (各メニュー)

- 基本的な名称は以下の通りです。

The screenshot shows the Exastro Ansible-Legacy web interface. On the left is a navigation menu with items like 'Movement一覧', 'Playbook素材集', and '作業管理'. The main content area is titled '説明' and contains a '表示フィルタ' (Display Filter) section with search boxes and buttons for 'フィルタ' and 'フィルタクリア'. Below this is a table of movement records. A red box highlights the main content area, and a callout box explains the sub-menu items.

**サブメニューの概略**

- 説明** : 表示中メニューの説明
- 表示フィルタ** : 登録情報の検索機能
- 一覧/更新** : 登録情報の表示

履歴	複製	更新	廃止	MovementID	Movement名	オーケストレータ	遅延タイマー	Ansible利用情報				最終更新日時	最終更新者	
履歴	複製	更新	廃止	2	パッケージインストール	Ansible Legacy		ホスト指定形式	WinRM接続	virtuallenv	ヘッダーセクション	オプシ	2021/12/16 11:51:37	システム管理者

## 2.3 画面説明(各メニュー) (2/2)

### 画面説明 (各メニュー)

- 基本的な名称は以下の通りです。

ようこそ[システム管理者]さん  
ログインID [administrator]

メニュー

説明

表示フィルタ

一覧/更新

登録

登録開始

全件ダウンロードとファイルアップロード編集

全件ダウンロード

全件ダウンロード(Excel)

新規登録用ダウンロード(Excel)

ファイルを選択 選択されていません  
アップロード状況:  
ファイルアップロード

変更履歴全件ダウンロード

変更履歴全件ダウンロード(Excel)

変更履歴

MovementID

表示 リセット

登録者に連絡

サブメニュー

#### ■ サブメニューの概略

**登録** : Webからのレコード登録

**全件ダウンロードとファイルアップロード編集**

: Excelのダウンロード/アップロードからの編集  
Excelでの変更履歴全件をダウンロード

**変更履歴** : 登録レコードの変更履歴

### 3. 実行前準備

# 3.1 Playbookをアップロードしてジョブ(Movement)に紐付け (1/5)

## Playbookの準備

- 最初に今回利用するPlaybookの作成をします。

お好みのエディタを使用してymlを作成し自身のローカルフォルダに保存してください。

yum\_package\_install.yml

```
- name: install the latest version of packages
  yum:
    name: "{{ item }}"
    state: latest
  with_items:
    - "{{ VAR_packages }}"
```

Playbookをアップロードして  
ジョブ(Movement)に紐付け

ジョブ(Movement)を  
ジョブフロー(Conductor)に組み込み

CMDBにパラメータシートを設定

パラメータシートの項目と  
Playbookの変数の紐付け

機器一覧にターゲットとなる  
Linuxマシンを登録

Point

文字コードは“UTF-8 BOMなし”、改行コードは“LF”、  
拡張子は“yml”形式。また、インデントにご注意ください。

```
yum_package_install.yml X
1 | name: install the latest version of packages
2 | yum:
3 |   name: "{{ item }}"
4 |   state: latest
5 | with_items:
6 |   - "{{ VAR_packages }}"
7
```

# 3.1 Playbookをアップロードしてジョブ(Movement)に紐付け (2/5)

## 「Movement一覧」へ新規Movementを登録

- 次にMovementの登録を行っていきます。

メインメニューより、「Ansible-Legacy」メニューグループ >> 「Movement一覧」メニューをクリックします。

The screenshot shows the Exastro Ansible-Legacy dashboard. The left sidebar contains a 'Menu' section with 'Movement一覧' highlighted by a red box and a red circle with the number '2'. Below it, the 'Ansible-Legacy' menu item is also highlighted with a red box and a red circle with the number '1'. The main dashboard area displays various metrics and charts, including a 'Movement' status chart showing 8 total movements at 100% completion, and a '作業履歴' (Job History) bar chart.

**Playbookをアップロードして  
ジョブ(Movement)に紐付け**

ジョブ(Movement)を  
ジョブフロー(Conductor)に組み込み

CMDBにパラメータシートを設定

パラメータシートの項目と  
Playbookの変数の紐付け

機器一覧にターゲットとなる  
Linuxマシンを登録

Movement	SUM
Ansible Legacy	
Ansible Pioneer	
Ansible Legacy Role	
Terraform	

Status	CON	SYM	SUM
実行中			
未実行(予約)			
未実行			

作業履歴	Count
21	7
25	2

# 3.1 Playbookをアップロードしてジョブ(Movement)に紐付け (3/5)

## 「Movement一覧」へ新規Movementを登録

- 「登録開始」ボタンをクリック。

各項目へ下表のように入力し、登録をクリックしてください。



Playbookをアップロードして  
ジョブ(Movement)に紐付け

ジョブ(Movement)を  
ジョブフロー(Conductor)に組み込み

CMDBにパラメータシートを設定

パラメータシートの項目と  
Playbookの変数の紐付け

機器一覧にターゲットとなる  
Linuxマシンを登録

Movement名	ホスト指定形式	オプションパラメータ
パッケージインストール	IP	-vvv

4



Point

詳細なログ確認を行うために  
オプションパラメータを-vvvにして  
ログレベル上げます。

詳細は[マニュアル](#)を参照してください。

# 3.1 Playbookをアップロードしてジョブ(Movement)に紐付け (4/5)

## 「Playbook素材集」へ新規Playbookを登録

- 次に作成したPlaybookの登録を行います。

「Ansible-Legacy」メニューグループ >> 「Playbook素材集」メニューをクリック。

登録開始をクリックし、各項目へ下表のように入力し登録をクリックしてください。

The screenshot shows the Exastro Ansible-Legacy interface. The sidebar menu on the left has 'Playbook素材集' highlighted with a red box and a red circle labeled '1'. The main content area shows a table with columns for '素材ID', 'Playbook素材名', 'Playbook素材', 'アクセス権', '備考', '最終更新日時', and '最終更新者'. Below the table, there is a form for adding a new Playbook. The form has two main input fields: 'Playbook素材名' (containing 'yum\_package\_install') and 'Playbook素材' (containing 'yum\_package\_install.yml'). A red box labeled '2' surrounds these fields. Below the form, there is a '事前アップロード' (Pre-upload) button and a '登録' (Register) button. A red box labeled '3' surrounds the '登録' button. A red callout box labeled 'Point' points to the '事前アップロード' button, containing the text: 'Playbookをアップロードする場合、ファイル指定後は必ず「事前アップロード」ボタンをクリックしてください。' (When uploading a Playbook, after specifying the file, be sure to click the 'Pre-upload' button.)

Playbookをアップロードして  
ジョブ(Movement)に紐付け

ジョブ(Movement)を  
ジョブフロー(Conductor)に組み込み

CMDBにパラメータシートを設定

パラメータシートの項目と  
Playbookの変数の紐付け

機器一覧にターゲットとなる  
Linuxマシンを登録

# 3.1 Playbookをアップロードしてジョブ(Movement)に紐付け (5/5)

## 「Movement-Playbook紐付」への登録

- 次に登録したPlaybookをMovementに紐付けます。

「Ansible-Legacy」メニューグループ >> 「Movement-Playbook紐付」メニューをクリック。各項目へ下表のように入力、選択し登録をクリックしてください。

Playbookをアップロードして  
ジョブ(Movement)に紐付け

ジョブ(Movement)を  
ジョブフロー(Conductor)に組み込み

CMDBにパラメータシートを設定

パラメータシートの項目と  
Playbookの変数の紐付け

機器一覧にターゲットとなる  
Linuxマシンを登録



Movement	Playbook素材	インクルード順序
パッケージインストール	yum_package_install	1

2



Point

Movement 1つに対し複数のPlaybookを登録する場合、インクルード順序を指定 1 : 1 の場合は1を入力してください。

## 3.2 ジョブ(Movement)をジョブフロー(Conductor)に組み込み

### 「Conductor」を作成する

- 次にMovementをConductorに組み込んでいきます。

「Conductor」メニューグループ >> 「Conductorクラス編集」メニューをクリック。

下記の順番の通り入力、移動、連結させ登録をクリックしてください。

1 Conductorクラス編集

2 Name: パッケージインストール

3 Movement: パッケージインストール

4 「OUT」から「IN」へドラッグで連結

5 登録

ドラッグ&ドロップ

作業説明等の文字入力可

Playbookをアップロードして  
ジョブ(Movement)に紐付け

ジョブ(Movement)を  
ジョブフロー(Conductor)に組み込み

CMDBにパラメータシートを設定

パラメータシートの項目と  
Playbookの変数の紐付け

機器一覧にターゲットとなる  
Linuxマシンを登録

## 3.3 CMDBにパラメータシートを設定(1/3)

### パラメータシートを作成する

- 次にパラメータシートの作成を行います。

「メニュー作成」メニューグループ >> 「メニュー定義・作成」メニューをクリック。  
各項目へ下表のように入力、選択して下さい。(次ページへ続く)

**2** グループをクリック

**3** グループ名  
インストールパッケージ

**1**

メニュー名	作成対象	表示順序
インストールパッケージ一覧	パラメータシート (ホスト/オペレーションあり)	1

Playbookをアップロードして  
ジョブ (Movement) に紐付け

ジョブ (Movement) を  
ジョブフロー (Conductor) に組み込み

CMDBにパラメータシートを設定

パラメータシートの項目と  
Playbookの変数の紐付け

機器一覧にターゲットとなる  
Linuxマシンを登録

# 3.3 CMDBにパラメータシートを設定(2/3)

## パラメータシートを作成する

- 項目を追加し、各項目へ下表のように入力、選択して下さい。(次ページへ続く)

4 項目が5個になるようにクリック



- Playbookをアップロードしてジョブ(Movement)に紐付け
- ジョブ(Movement)をジョブフロー(Conductor)に組み込み
- CMDBにパラメータシートを設定**
- パラメータシートの項目とPlaybookの変数の紐付け
- 機器一覧にターゲットとなるLinuxマシンを登録

項目名	入力方式	選択項目
httpd	プルダウン選択	メニュー作成:選択1:*-(空白)
MariaDB-server	プルダウン選択	メニュー作成:選択1:*-(空白)
php	プルダウン選択	メニュー作成:選択1:*-(空白)
perl	プルダウン選択	メニュー作成:選択1:*-(空白)
python	プルダウン選択	メニュー作成:選択1:*-(空白)

Point

今回はCentOS7.8を対象としています。  
CentOS7系以外は『mariadb-server』と小文字で入力してください。

# 3.3 CMDBにパラメータシートを設定(3/3)

## パラメータシートを作成する

- 項目の移動が完了できたら作成をクリックして下さい。

6 カラムグループの領域に項目をドラッグ&ドロップ

7 すべての項目を移動すると以下ようになります

8 作成

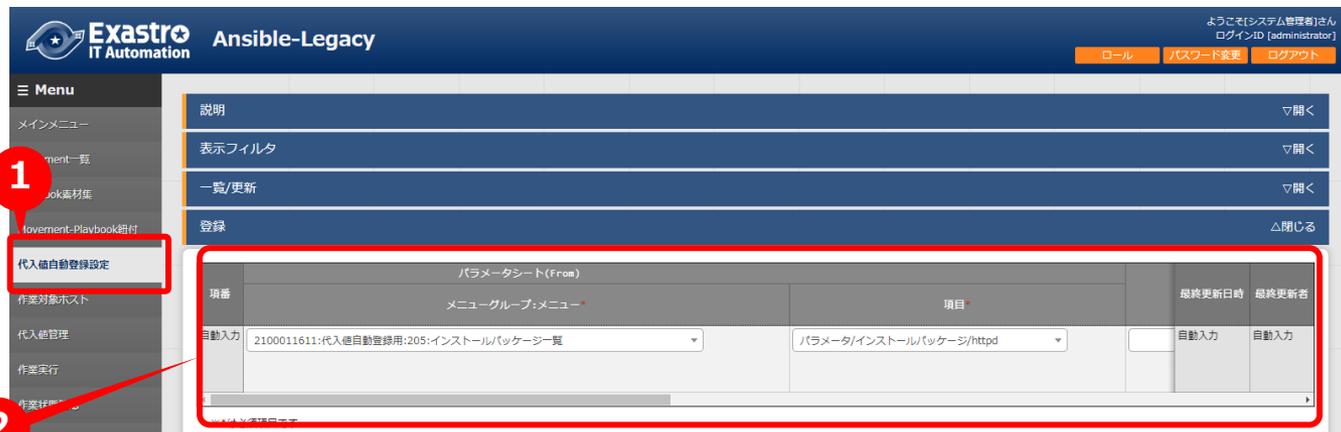
- Playbookをアップロードしてジョブ(Movement)に紐付け
- ジョブ(Movement)をジョブフロー(Conductor)に組み込み
- CMDBにパラメータシートを設定**
- パラメータシートの項目とPlaybookの変数の紐付け
- 機器一覧にターゲットとなるLinuxマシンを登録

# 3.4 パラメータシートの項目とPlaybookの変数の紐付け(1/3)

## 「代入値自動登録設定」作成

- 最後に代入値自動登録を行います。

「Ansible-Legacy」メニューグループ >> 「代入値自動登録設定」メニューをクリック。  
各項目へ下表のように入力、選択して下さい。(次ページへ続く)



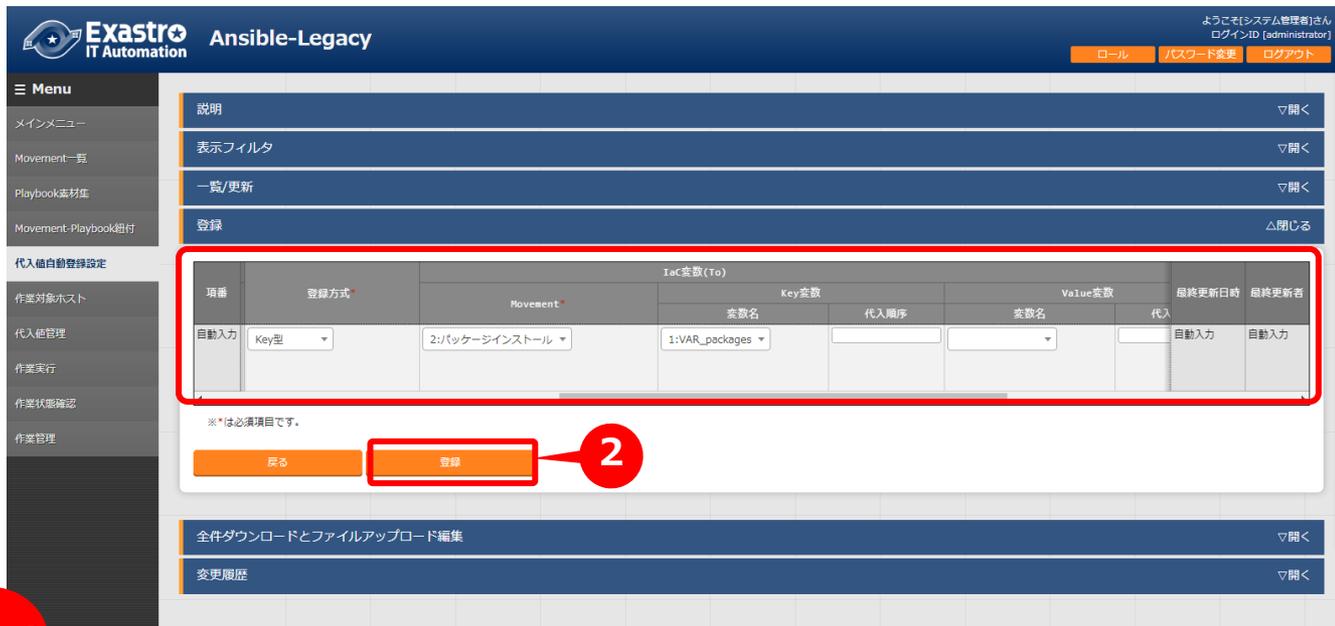
- Playbookをアップロードしてジョブ(Movement)に紐付け
- ジョブ(Movement)をジョブフロー(Conductor)に組み込み
- CMDBにパラメータシートを設定
- パラメータシートの項目とPlaybookの変数の紐付け**
- 機器一覧にターゲットとなるLinuxマシンを登録

メニューグループ:メニュー	項目	登録方式	Movement	Key変数変数名	代入順序
2100011611:代入値自動登録用:3:インストールパッケージ一覧	パラメータ/インストールパッケージ/httpd	Key型	1:パッケージインストール	1:VAR_packages	1
2100011611:代入値自動登録用:3:インストールパッケージ一覧	パラメータ/インストールパッケージ/MariaDB-server	Key型	1:パッケージインストール	1:VAR_packages	2
2100011611:代入値自動登録用:3:インストールパッケージ一覧	パラメータ/インストールパッケージ/php	Key型	1:パッケージインストール	1:VAR_packages	3
2100011611:代入値自動登録用:3:インストールパッケージ一覧	パラメータ/インストールパッケージ/perl	Key型	1:パッケージインストール	1:VAR_packages	4
2100011611:代入値自動登録用:3:インストールパッケージ一覧	パラメータ/インストールパッケージ/python	Key型	1:パッケージインストール	1:VAR_packages	5

# 3.4 パラメータシートの項目とPlaybookの変数の紐付け(2/3)

## 「代入値自動登録設定」作成

- 各項目へ入力、選択が完了したら登録をクリックして下さい。



Playbookをアップロードして  
ジョブ(Movement)に紐付け

ジョブ(Movement)を  
ジョブフロー(Conductor)に組み込み

CMDBにパラメータシートを設定

パラメータシートの項目と  
Playbookの変数の紐付け

機器一覧にターゲットとなる  
Linuxマシンを登録

### Point

変数紐づけの登録方式は以下の3タイプがあります。

登録方式	今回使用	説明
Value型		基本的なタイプであり、表の中の値を変数に紐づけるものです。
Key型	●	表の項目(列名)を変数に紐づけるものです。項目の設定値が空白の場合は紐づけ対象外になります。
Key-Value型		項目の名称(Key)と設定値(Value)の両方を変数に紐づけることができます。

今回のシナリオでは、表の項目(列名)をPlaybookに具体値として代入したいので、登録方式は「Key型」を選択します。詳細はこちら[システム構築・運用の効率化ガイドブック](#)を参照してください。

# 3.4 パラメータシートの項目とPlaybookの変数の紐付け(3/3)

## 「代入値自動登録設定」作成

- 表示フィルタで5件のデータが登録できているかの確認を行って下さい。  
ここまでで実行準備は終了になります。

Exastro IT Automation Ansible-Legacy

ようこそ[システム管理者]さん  
ログインID [administrator]

ロール パスワード変更 ログアウト

Menu

メインメニュー

Movement一覧

Playbook素材集

Movement-Playbook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

表示フィルタ

パラメータシート (From)

停止	項番	メニューグループ		メニュー		最終更新日時	最終更新者
		ID	名称	ID	名称		
廃止含まず	~	▼ブルダウン検索	▼ブルダウン検索	▼ブルダウン検索	▼ブルダウン検索	▼ブルダウン検索	▼ブルダウン検索

フィルタ フィルタクリア

5件のデータが登録できているかを確認

パラメータシート (from)

履歴	複製	更新	廃止	項番	メニューグループ	メニュー	項目	登録方	最終更新日時	最終更新者
履歴	複製	更新	廃止	9	1100011611 代入値自動登録用	205 インストールパッケージ	パラメータ/インストール/パッケージ/httd	Key型	2021/12/16 12:08:11	システム管理者
履歴	複製	更新	廃止	10	1100011611 代入値自動登録用	205 インストールパッケージ	パラメータ/インストール/パッケージ/MariaDB-server	Key型	2021/12/16 12:56:58	システム管理者
履歴	複製	更新	廃止	11	1100011611 代入値自動登録用	205 インストールパッケージ	パラメータ/インストール/パッケージ/php	Key型	2021/12/16 12:57:19	システム管理者
履歴	複製	更新	廃止	12	1100011611 代入値自動登録用	205 インストールパッケージ	パラメータ/インストール/パッケージ/perl	Key型	2021/12/16 12:57:39	システム管理者
履歴	複製	更新	廃止	13	1100011611 代入値自動登録用	205 インストールパッケージ	パラメータ/インストール/パッケージ/python	Key型	2021/12/16 12:57:54	システム管理者

フィルタ結果件数: 5

Excel出力

登録

Playbookをアップロードして  
ジョブ(Movement)に紐付け

ジョブ(Movement)を  
ジョブフロー(Conductor)に組み込み

CMDBにパラメータシートを設定

パラメータシートの項目と  
Playbookの変数の紐付け

機器一覧にターゲットとなる  
Linuxマシンを登録

## 4. 実行操作(1回目)

# 4.1 機器一覧にターゲットとなるLinuxマシンを登録(1/3)

## 「機器一覧」へ新規ターゲットホストの登録

- 最初に機器一覧へ今回パッケージをインストールするターゲットホストを登録します。「基本コンソール」メニューグループ >> 「機器一覧」メニューをクリック。各項目へ下表のように入力して下さい。(次ページへ続く)

The screenshot shows the Exastro IT Automation console interface. The 'Menu' on the left has '機器一覧' (Machine List) highlighted with a red box and a red circle containing the number '1'. The main content area shows the '登録' (Registration) form for a host. A red box highlights the form fields (2). The form includes a table with the following columns: 管理システム番号 (Management System Number), HW機器種別 (HW Device Type), ホスト名 (Host Name), IPアドレス (IP Address), EtherWakeOnLan (MAC Address, Network Device Name), ログインユーザID (Login User ID), 最終更新日時 (Last Updated Time), and 最終更新者 (Last Updated By). The 'HW機器種別' field is set to 'SV'. The 'ホスト名' field contains 'hostA'. The 'IPアドレス' field contains '192.XXX.XXX.X'. Below the table, there are '戻る' (Back) and '登録' (Register) buttons.

管理システム番号	HW機器種別	ホスト名*	IPアドレス*	MACアドレス	ネットワークデバイス名	ログインユーザID	最終更新日時	最終更新者
自動入力	SV	hostA	192.XXX.XXX.X				自動入力	自動入力

\*は必須項目です。

戻る 登録

- Playbookをアップロードしてジョブ(Movement)に紐付け
- ジョブ(Movement)をジョブフロー(Conductor)に組み込み
- CMDBにパラメータシートを設定
- パラメータシートの項目とPlaybookの変数の紐付け

**機器一覧にターゲットとなるLinuxマシンを登録**

HW機器種別	ホスト名	IPアドレス
SV	(任意のホスト名)	(任意のIPアドレス)

# 4.1 機器一覧にターゲットとなるLinuxマシンを登録(2/3)

## 「機器一覧」へ新規ターゲットホストの登録

- スクロールバーを右にスライドし各項目へ下表のように入力して下さい。(次ページへ続く)

Exastro IT Automation 基本コンソール

ようこそ[システム管理者]さん  
ログインID [administrator]

ロール パスワード変更 ログアウト

Menu  
メインメニュー  
機器一覧  
オペレーション一覧  
Movement一覧  
ER回表示

説明 ▾開く  
表示フィルタ ▾開く  
一覧/更新 ▾開く  
登録 △閉じる

管理システム項目	ログインユーザID	管理	ログインパスワード	ssh鍵認証情報	パスフレーズ	最終更新日時	最終更新者
自動入力	root	● ▾	●●●●●●	ファイルを選択   選択されていません	●●●●●●	自動入力	自動入力

※\*は必須項目です。

戻る 登録

管理者に連絡

- Playbookをアップロードしてジョブ(Movement)に紐付け
- ジョブ(Movement)をジョブフロー(Conductor)に組み込み
- CMDBにパラメータシートを設定
- パラメータシートの項目とPlaybookの変数の紐付け
- 機器一覧にターゲットとなるLinuxマシンを登録

ログインユーザID	ログインパスワード管理	ログインパスワード
(任意のログインユーザID)	●	(任意のパスワード)

# 4.1 機器一覧にターゲットとなるLinuxマシンを登録(3/3)

## 「機器一覧」へ新規ターゲットホストの登録

- 最後の項目へ下表のように選択し登録をクリックして下さい。

Exastro IT Automation 基本コンソール

ようこそ[システム管理者]さん  
ログインID [administrator]  
パスワード変更 ログアウト

Menu  
メインメニュー  
機器一覧  
オペレーション一覧  
Movement一覧  
ER図表示

説明  
表示フィルタ  
一覧/更新  
登録

3 Legacy/Role利用情報 認証方式  
パスワード認証

Ansible利用情報  
Legacy/Role利用情報

管理システム項番	認証方式	ポート番号	WinRM接続情報	プロトコル	最終更新日時	最終更新者
自動入力	パスワード認証		ファイルを選択   選択されていません		自動入力	自動入力

事前アップロード  
アップロード状況:

戻る 登録 4

全件ダウンロードとファイルアップロード編集  
変更履歴

管理者に連絡

Playbookをアップロードして  
ジョブ(Movement)に紐付け

ジョブ(Movement)を  
ジョブフロー(Conductor)に組み込み

CMDBにパラメータシートを設定

パラメータシートの項目と  
Playbookの変数の紐付け

機器一覧にターゲットとなる  
Linuxマシンを登録

### Point

Ansible-Legacyを実行するための  
必須入力項目は以下の6項目です。

[ホスト名][IPアドレス][ログインユーザID]  
[ログインパスワード管理][ログインパスワード]  
[認証方式※]

※本書は「パスワード認証」で記載します。

## 4.2 作業名(Operation)の登録

### 「オペレーション一覧」へ新規オペレーション名を登録

- 次にオペレーション名を登録していきます。

「基本コンソール」メニューグループ >> 「オペレーション一覧」メニューをクリック。  
各項目へ入力、選択が完了したら登録をクリックして下さい。

Exastro IT Automation 基本コンソール

ようこそ[システム管理者]さん  
ログインID [administrator]

ロール パスワード変更 ログアウト

Menu  
メインメニュー  
機器一覧  
**オペレーション一覧**  
Movement一覧  
ER回表示

説明  
表示フィルタ  
No. オペレーションID オペレーション名 実施 最終更新日時 最終更新者

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 設定 アグ レール 備考 最終更新日時 最終更新者

登録

戻る 登録

全件ダウンロードとファイルアップロード編集  
変更履歴

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

オペレーション名	実施予定日時
オペレーション1	(任意の実行予定日時)

## 4.3 パラメータシートにデータを登録(1/2)

### 「インストールパッケージ一覧」へ新規データを登録

- 次に実行前準備で用意したインストールパッケージ一覧(パラメータシート)にデータを入力していきます。

「入力用」メニューグループ >> 「インストールパッケージ一覧」メニューをクリック。  
各項目へ入力、選択が完了したら登録をクリックして下さい。

No	ホスト名	オペレーション	パラメータ	最終更新日時	最終更新者
自動入力	hostA	2021/12/16 12:35_21:オペレーション1	httpd MariaDB-server php perl	自動入力	自動入力

ホスト名	オペレーション	httpd	MariaDB-server	php	perl	python
(機器登録で登録したホスト名)	(選択した実行予定日時)_1: オペレーション1	*		*	*	*

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

## 4.3 パラメータシートにデータを登録(2/2)

### 「インストールパッケージ一覧」への登録

- 実行前準備の代入値自動登録設定の時と同様、表示フィルタを開き「フィルタ」ボタンをクリックして登録したデータを確認してください。

Exastro IT Automation 入力用

ようこそ[システム管理者]さん  
ログインID [administrator]

ロール パスワード変更 ログアウト

Menu

説明

表示フィルタ

廃止	No	ホスト名	ID	オペレーション名	基底	最終更新日時	最終更新者
廃止含まず	~	プルダウン検索	~	プルダウン検索		~	プルダウン検索

フィルタ フィルタクリア

一覧/更新

履歴	複製	更新	廃止	No	ホスト名	ID	オペレーション名	基準日時	実施予定日時	最終実行日時	インストールパッケージ	最終更新日時	最終更新者
履歴	複製	更新	廃止	1	hostA	21	オペレーション 1	2021/12/17 14:32	2021/12/16 12:35	2021/12/17 14:32	httpd MariaDB-server php perl python	2021/12/16 13:08:52	システム管理者

Excel出力

登録

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

# 4.4 ジョブフロー(Conductor)の実行(1/3)

## Conductorの実行

- いよいよ実行を行っていきます。  
「Conductor」メニューグループ>>「Conductor作業実行」メニューをクリック。  
実行する「Conductor」と「オペレーション」を選択し実行をクリックして下さい。

ようこそ[システム管理者]さん  
ログインID [administrator]

メニュー

- メインメニュー
- Conductorインターフェース情報
- Conductor通知先定義
- Conductorクラス一覧
- Conductorクラス編集
- Conductor作業実行**
- Conductor作業確認
- Conductor作業一覧
- Conductor定期作業実行

説明

スケジュールリング

Conductor[フィルタ]

Conductor[一覧]

選択	ConductorクラスID	Conductor名称	説明	アクセス権	備考	最終更新日時	最終更新者
<input checked="" type="radio"/>	1	パッケージインストール		アクセス許可ロール		2021/12/17 11:34:47	システム管理者

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.	オペレーションID	オペレーション名	実施予定日時	最終実行日時	アクセス権	備考	最終更新日時
<input checked="" type="radio"/>	21	21	オペレーション1	2021/12/16 12:35	2021/12/16 13:09	アクセス許可ロール		2021/12/16 13:09

EXECUTE

管理者に連絡

全体表示

表示

ジョブフロー(Conductor)の実行

作業名(Operation)の登録

パラメータシートにデータを登録

実行結果の確認

## 4.4 ジョブフロー(Conductor)の実行(2/3)

### 実行結果確認

- 実行すると「Conductor作業確認」メニュー画面に切替わり、実行ステータスやログが表示されます。

Exastro IT Automation Conductor

ようこそ(システム管理者)さん  
ログインID [administrator]

ロール パスワード変更 ログアウト

全体表示 表示リセット フルスクリーン

Conductor名称  
Conductor instance ID : 48  
Conductor name : バックアップインストール  
Status : 正常終了  
Start time : 2021/12/17 11:47:15  
End time : 2021/12/17 11:49:04  
Execution user : システム管理者  
Reservation date :  
Emergency stop :  
Note

Operation  
Operation ID : 21  
Operation name : オペレーション1

マウスをクリック Node選択・作業状態確認 マウスホイール 画面の拡大・縮小 マウスをドラッグ 画面の移動

緊急停止

管理者に連絡

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

Point

実行ステータスやログをリアルタイムで確認可能です。

# 4.4 ジョブフロー(Conductor)の実行(3/3)

## 実行結果確認

- ジョブ(Movement)を選択し、Doneのアイコンまたは右側のOperation statusをクリックすると詳細が表示されます。

The screenshot shows the Exastro Conductor interface. A job flow is displayed with three nodes: 'Start', 'Ansible Legacy', and 'End'. The 'Ansible Legacy' node is highlighted with a red box and a red arrow pointing to it from a '1 選択' (Select) callout. The 'Ansible Legacy' node has a 'DONE' status. A red arrow points from the 'DONE' icon to a detailed view of the operation. The detailed view shows the following information:

項目	値
作業No.	113
実行種別	通常
ステータス	完了
実行エンジン	Ansible Engine
Ansible engine virtualenv /パス	
呼出元Symphony	
呼出元Conductor	パッケージインストール
実行ユーザ	システム管理者
Movement	
ID	2
名称	パッケージインストール
遅延タイム(秒)	
Ansible利用情報	ホスト指定形式: IP
URL接続	
オペレーション	
No.	21
名称	オペレーション1
ID	21
作業対象ホスト	
代入値	抽出
代入値	抽出
入力データ	投入データ
InoutData_0000000113_zio	
出力データ	結果データ
ResultData_0000000113_zio	
予約日時	
作業状況	
開始日時	2021/12/17 11:47:18
終了日時	2021/12/17 11:48:57

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

# 4.5 実行結果の確認(1/3)

## 実行ログの確認

- 詳細画面の進行状況(実行ログ)でAnsibleの実行ログを確認していきます。

No.	名称	ID	操作
3	オペレーション1	3	実行

入力データ	投入データ
出力データ <td>結果データ</td>	結果データ

作業状況	開始日時	終了日時
	2021/08/19 18:26:21	2021/08/19 18:27:28

```
Updating : python-2.7.5-90.el7.x86_64 2/4
Cleanup : python-2.7.5-88.el7.x86_64 3/4
Cleanup : python-libs-2.7.5-88.el7.x86_64 4/4
Verifying : python-2.7.5-90.el7.x86_64 1/4
Verifying : python-libs-2.7.5-88.el7.x86_64 2/4
Verifying : python-libs-2.7.5-88.el7.x86_64 3/4
Verifying : python-2.7.5-88.el7.x86_64 4/4

Updated:
python.x86_64 0:2.7.5-90.el7

Dependency Updated:
python-libs.x86_64 0:2.7.5-90.el7

Complete!
}}

PLAY RECAP *****
ita-test-target01 : ok=1 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

## 4.5 実行結果の確認(2/3)

### 実行ログの確認

- httpd,php,perl,pythonをインストールされているか実行ログから確認して下さい。

進行状況(実行ログ)の一部の例

```
~~~~~省略~~~~~
Installed:
  httpd.x86_64 0:2.4.6-97.el7.centos
Dependency Installed:
  httpd-tools.x86_64 0:2.4.6-97.el7.centos mailcap.noarch 0:2.1.41-2.el7
Complete!
"]}"
~~~~~省略~~~~~
Installed:
  php.x86_64 0:5.4.16-48.el7
Dependency Installed:
  libzip.x86_64 0:0.10.1-8.el7 php-cli.x86_64 0:5.4.16-48.el7 php-common.x86_64 0:5.4.16-48.el7
Complete!
"]}"
~~~~~省略~~~~~
Updated:
  perl.x86_64 4:5.16.3-299.el7_9
Dependency Updated:
  perl-libs.x86_64 4:5.16.3-299.el7_9
Complete!
"]}"
~~~~~省略~~~~~
Updated:
  python.x86_64 0:2.7.5-90.el7
Dependency Updated:
  python-libs.x86_64 0:2.7.5-90.el7
Complete!
"]}"
```

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

**実行結果の確認**

## 4.5 実行結果の確認(3/3)

### ターゲットマシンの確認

- ターゲットマシンでもパッケージがインストールできていることを確認して下さい。

hostA

```
$ yum list installed httpd
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: ftp-srv2.kddilabs.jp
* extras: ftp-srv2.kddilabs.jp
* updates: ftp-srv2.kddilabs.jp
Installed Packages
httpd.x86_64                2.4.6-97.el7.centos        @updates
```

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

## 5. 実行操作(2回目)



# 5.1 作業名(Operation)の登録

## 「オペレーション一覧」へ新規オペレーション名を登録

- ここからは1回目のオペレーション名登録以降の作業と同様になります。  
「基本コンソール」メニューグループ >> 「オペレーション一覧」メニューをクリック。  
各項目へ入力、選択が完了したら登録をクリックして下さい。

ようこそ[システム管理者]さん  
ログインID [administrator]

ロール パスワード変更 ログアウト

Menu  
メインメニュー  
機器一覧  
**オペレーション一覧**  
Movement一覧  
ER図表示

説明 ▾開く  
表示フィルタ △閉じる

No.	オペレーションID	オペレーション名	最終更新日時	最終更新者
全レコード				

フィルタ フィルタクリア  
 オートフィルタ

一覧/更新 ▾開く  
登録 △閉じる

No.	オペレーションID	オペレーション名	実施予定日時	実行権	許可ロール	最終更新日時	最終更新者
自動入力	自動入力	オペレーション2	2021/12/17 12:10	設定			

※\*は必須項目です。

戻る 登録

管理者に連絡

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

オペレーション名  
オペレーション2

実施予定日時  
(任意の日時)

## 5.2 パラメータシートにデータを登録

### 「インストールパッケージ一覧」新規データを登録

- 「入力用」メニューグループ >> 「インストールパッケージ一覧」メニューをクリック。各項目へ入力、選択が完了したら登録をクリックして下さい。  
1回目とインストールするパッケージが異なっているので注意して下さい。

No.	ホスト名	オペレーション	パラメータ				最終更新日時	最終更新者	
オペレーション			インストールパッケージ						
自動入力	hostA	2021/12/16 13:25_22:オペレーション2	httpd	MariaDB-server	php	perl	python	自動入力	自動入力

ホスト名	オペレーション	httpd	MariaDB-server	php	perl	python
(機器登録で登録したホスト名)	(選択した実行予定日時)_2: オペレーション2	*	*	*	*	*

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

# 5.3 ジョブフロー(Conductor)の実行(1/3)

## Conductorの実行

- 2回目の実行も行っていきます。

「Conductor」メニューグループ>>「Conductor作業実行」メニューをクリック。  
実行する「Conductor」と「オペレーション」を選択し実行をクリックして下さい。

The screenshot shows the Exastro Conductor web interface. The left sidebar has a menu with 'Conductor作業実行' highlighted by a red box and a red circle with the number '1'. The main content area has a 'Conductor[一覧]' button highlighted with a red circle and the number '2'. Below it is a table of Conductor classes:

選択	ConductorクラスID	Conductor名称	説明	アクセス権	備考	最終更新日時	最終更新者
<input checked="" type="radio"/>	1	パッケージインストール		アクセス許可ロール		2021/12/17 11:34:47	システム管理

Below the table, it says 'フィルタ結果件数: 1'. Further down, there is an 'オペレーション[一覧]' button highlighted with a red circle and the number '3'. Below it is a table of operations:

選択	No.	オペレーションID	オペレーション名	実施予定日時	最終実行日時	アクセス権	最終更新日時
<input type="radio"/>	21	21	オペレーション1	2021/12/16 12:35	2021/12/17 14:32	アクセス許可ロール	2021/12/17 14:3
<input checked="" type="radio"/>	22	22	オペレーション2	2021/12/16 13:25	2021/12/17 14:44		2021/12/17 14:4

Below the table, it says 'フィルタ結果件数: 2'. At the bottom of the interface, there is a 'Conductor実行' button highlighted with a red box and a red circle with the number '4'.

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

The screenshot shows a job flow diagram with three steps: 'Start', 'Conductor', and 'End'. The 'Conductor' step is highlighted with a red circle and the number '2'. The diagram is displayed in a window titled 'Conductor実行'.

## 5.3 ジョブフロー(Conductor)の実行(2/3)

### 作業結果確認

- 実行すると「Conductor作業確認」メニュー画面に切替わり、実行ステータスやログが表示されます。

The screenshot displays the Exastro Conductor web interface. The main area shows a job flow diagram on a grid background. The flow consists of three nodes: a 'Start' node (blue circle with 'S'), an 'Ansible Legacy' node (red circle with 'DONE'), and an 'End' node (blue circle with 'E'). The 'Ansible Legacy' node is labeled 'パッケージインストール' (Package Installation). The status bar at the top indicates 'CHECKING'. On the right, a sidebar shows execution details: Conductor Instance ID: 46, Conductor name: パッケージインストール, Status: 正常終了 (Normal End), Start time: 2021/12/16 14:38:02, End time: 2021/12/16 14:39:47, Execution user: システム管理者 (System Administrator). The bottom right shows Operation ID: 22 and Operation name: オペレーション2 (Operation 2). A 'ログ' (Log) section is visible at the bottom left.

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

Point

実行ステータスやログをリアルタイムで確認可能です。

# 5.3 ジョブフロー(Conductor)の実行(3/3)

## 作業結果確認

- ジョブ(Movement)を選択し、Doneのアイコンまたは右側のOperation statusをクリックすると詳細が表示されます。

The screenshot shows the Exastro Conductor interface. A job flow is displayed with three nodes: 'Start', 'Ansible Legacy (DONE)', and 'End'. The 'Ansible Legacy' node is highlighted with a red box and a callout '1 選択'. A red arrow points from this node to a detailed view of the operation. The detailed view shows the following information:

項目	値
作業者No.	111
実行種別	通常
ステータス	完了
実行エンジン	Ansible Engine
Ansible engine virtualenvパス	
呼出元Symphony	
呼出元Conductor	パッケージインストール
実行ユーザ	システム管理者
ID	2
名称	パッケージインストール
Movement	
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.	22
名称	オペレーション2
ID	22
作業対象ホスト	<a href="#">確認</a>
代入値	<a href="#">確認</a>
入力データ	投入データ <a href="#">InputData 9988880111.jic</a>

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

# 5.4 実行結果の確認(1/2)

## 作業結果確認

- 詳細画面の進行状況(実行ログ)でAnsibleの実行ログを確認します。

入力データ	投入データ	<a href="#">InputData_0000000028.iid</a>
出力データ	結果データ	<a href="#">ResultData_0000000028.iid</a>
作業状況	予約日時	
	開始日時	2021/08/20 10:44:57
	終了日時	2021/08/20 10:46:01

### 進行状況(実行ログ)

フィルタ:   該当行のみ表示

```
update_only: false,
update_only: false,
use_backend: "auto",
validate_certs: true
}
},
"item": "python",
"msg": "",
"rc": 0,
"results": [
  "All packages providing python are up to date",
  ""
]
]
]
META: ran handlers
META: ran handlers
PLAY RECAP *****
hosta : ok=1  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

### 進行状況(エラーログ)

フィルタ:   該当行のみ表示

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

実行結果の確認

## 5.4 実行結果の確認(2/2)

### 実行ログの確認

- 新たにMariaDBのインストールと他のパッケージとの依存関係の解決、他の4つのパッケージ(httpd,php,perl,python)のバージョンアップが行われていることを確認して下さい。

進行状況(実行ログ)の一部の例

```
~~~~~省略~~~~~
"results": ["All packages providing httpd are up to date",
""}]
~~~~~省略~~~~~
Installed:
  MariaDB-server.x86_64 0:10.8.4-1.el7.centos
Dependency Installed:
  perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.el7
  perl-Compress-Raw-Zlib.x86_64 1:2.061-4.el7
  perl-DBD-MySQL.x86_64 0:4.023-6.el7
  perl-DBI.x86_64 0:1.627-4.el7
  perl-IO-Compress.noarch 0:2.061-2.el7
  perl-Net-Daemon.noarch 0:0.48-5.el7
  perl-PIRPC.noarch 0:0.2020-14.el7
Complete!
"}]
~~~~~省略~~~~~
"results": ["All packages providing php are up to date",
""}]
~~~~~省略~~~~~
"results": ["All packages providing perl are up to date",
""}]
~~~~~省略~~~~~
"results": ["All packages providing python are up to date",
""}]
```

作業名(Operation)の登録

パラメータシートにデータを登録

ジョブフロー(Conductor)の実行

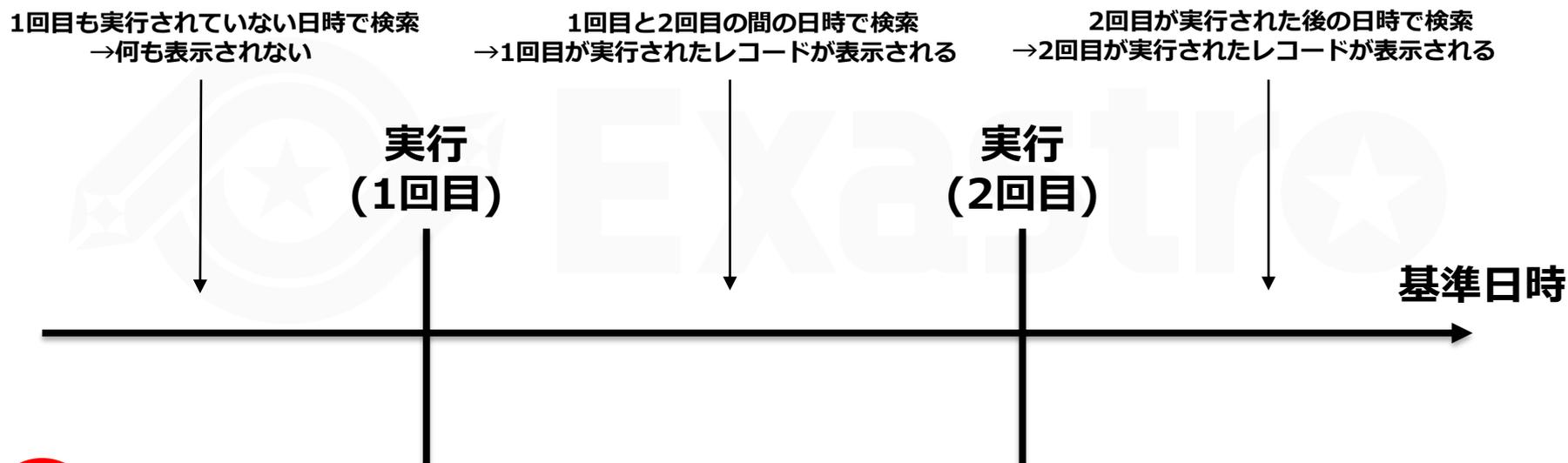
実行結果の確認

## 6. CMDBパラメータの履歴確認

## 6.1 作業実行と履歴管理

### 履歴管理と本シナリオのポイント

- ITAはCMDBに「誰が・いつ・何をしたのか？」を履歴管理し、その時の時点でシステムのパラメータはどうなっているのかを抽出できる機能があります。パラメータの履歴管理をすることにより、設計者や運用者がストレスなくシステム更改を行うことができます。



#### Point

今回のクイックスタートではパラメータの履歴管理を体験していただくために、実行を2回行いました。

## 6.2 CMDBパラメータの履歴を確認する(1/3)

### 履歴確認

- 実際にパラメータが管理できているかどうか確認をしていきます。  
「参照用」メニューグループ >> 「インストールパッケージ一覧」メニューをクリック。  
まずは基準日付を入力せずにフィルタをかけます。(次ページへ続く)

ようこそ[システム管理者]さん  
ログインID [administrator]

ロール パスワード変更 ログアウト

Menu

メインメニュー

インストールパッケージ一覧

説明

フィルタ

ホスト名 オペレーション  
基準日付

▼フルダウ検索

フィルタ フィルタクリア

オートフィルタ

一覧

履歴 No	ホスト名	オペレーション				パラメータ				アクセス		最終更新日時	最終更新者	
		ID	オペレーション名	基準日時	実施予定日時	最終実行日時	インストールパッケージ				アクセス許可			
2	hostA	22	オペレーション2	2021/12/17 14:44	2021/12/16 13:25	2021/12/17 14:44	httpd	MariaDB-server	php	perl	python		2021/12/17 14:44:28	システム管理

フィルタ結果件数: 1

Excel出力

全件ダウンロード

変更履歴

管理者に連絡

1 何も入力せずにフィルタをクリック

2

3

4 現時点での最新のデータが表示されることを確認する

## 6.2 CMDBパラメータの履歴を確認する(2/3)

### 履歴確認

- 次に2回目の実行を行った基準日時より前の日付を入力してフィルタをかけます。  
(次ページへ続く)

Exastro IT Automation 参照用

ようこそ[システム管理者]さん  
ログインID [administrator]  
ロール パスワード変更 ログアウト

Menu  
メインメニュー  
インストールパッケージ一覧

説明

表示フィルタ

1 2回目の実行を行った日時より前の日時を入力し  
フィルタをクリック

2

フィルタ フィルタクリア

オートフィルタ

一覧

履歴	No	ホスト名	オペレーション			パラメータ				アクセス		最終更新日時	最終更新者
			ID	オペレーション名	基準日時	実施予定日時	最終実行日時	インストールパッケージ					
履歴	1	hostA	21	オペレーション1	2021/12/17 14:32	2021/12/16 12:35	2021/12/17 14:32	*	*	*	アクセス許可	2021/12/16 13:08:52	システム管理

フィルタ結果件数: 1

3 1回目の実行結果が表示されることを確認する

Excel出力

全件ダウンロード

変更履歴

管理者に連絡

## 6.2 CMDBパラメータの履歴を確認する(3/3)

### 履歴確認

- 最後に1回目の実行を行った基準日時より前の日付を入力してフィルタをかけます。

The screenshot shows the Exastro IT Automation interface. The header includes the Exastro logo, the text '参照用', and user information: 'ようこそ[システム管理者]さん', 'ログインID [administrator]', and buttons for 'ロール', 'パスワード変更', and 'ログアウト'. The left sidebar has a 'Menu' section with 'メインメニュー' and 'インストールパッケージ一覧'. The main content area has a '説明' section with a '表示フィルタ' button. Below this is a table with columns 'ホスト名' and 'オペレーション'. The 'オペレーション' column has a sub-column '経過日時' with the value '2021/12/15 14:35'. A red box labeled '1' highlights this date. Below the table are buttons for 'フィルタ' and 'フィルタクリア', with a red box labeled '2' around the 'フィルタ' button. Below the buttons is a checkbox for 'オートフィルタ'. Below the table is a '一覧' section with a 'レコードはありません。' message, highlighted by a red box labeled '3'. At the bottom are buttons for '全件ダウンロード' and '変更履歴'. A red box labeled '3' highlights the 'レコードはありません。' message with the text '何も表示されないことを確認する'.

1 1回目の実行を行った日時より前の日時を入力し  
フィルタをクリック

2 フィルタ

3 レコードはありません。  
何も表示されないことを確認する

# A 付録

# 参考① 【Ansible-Legacy】 単体実行

## 作業実行

- Ansible-Legacyは「作業実行」メニューがあり Movementごとに個別実行や、ドライランが可能です。

The screenshot shows the Exastro Ansible-Legacy web interface. The left sidebar contains a 'Menu' with '作業実行' (Job Execution) highlighted. The main content area displays a table of Movements and a table of Operations. A red box highlights the first movement in the 'Movement' table, which is selected. A second red box highlights the first operation in the 'Operation' table, which is selected. A third red box highlights the 'Dry Run' button at the bottom of the interface.

選択	MovementID	Movement名	オケストレータ	遅延タイマー	ホスト指定形式	WinRM接続	virtuaIenv	ヘッダーセクション	最終更新日時	最終更新者
<input checked="" type="radio"/>	2	パッケージインストール Ansible Legacy			IP				2021/12/17 11:34:12	システム管理者

選択	No.	オペレーションID	オペレーション名	実施予定日時	最終実行日時	アクセス権	備考	最終更新日時	最終更新者
<input checked="" type="radio"/>	21	21	オペレーション1	2021/12/16 12:35	2021/12/17 14:32			2021/12/17 14:32:50	legacy作業実行プロセス
<input type="radio"/>	22	22	オペレーション2	2021/12/16 13:25	2021/12/17 14:44			2021/12/17 14:44:54	legacy作業実行プロセス

1 作成済みのMovementを選択

2 Movementと紐づいたオペレーションを選択

3 ドライラン

: プレイブックの接続確認/構文チェック

実行

: プレイブックを実行





**Exastro**