



# 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。

Exastro IT Automation Version 1.8  
Exastro developer

# 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](#)

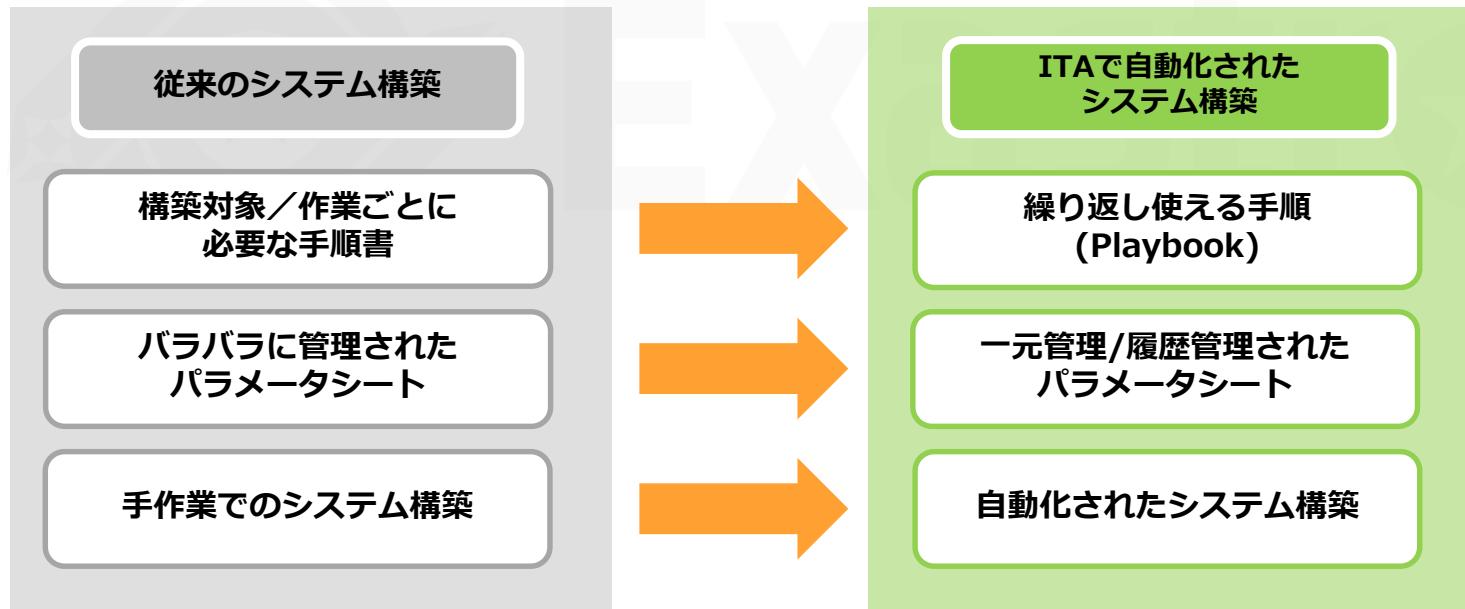
# 1. はじめに



# 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のインター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。

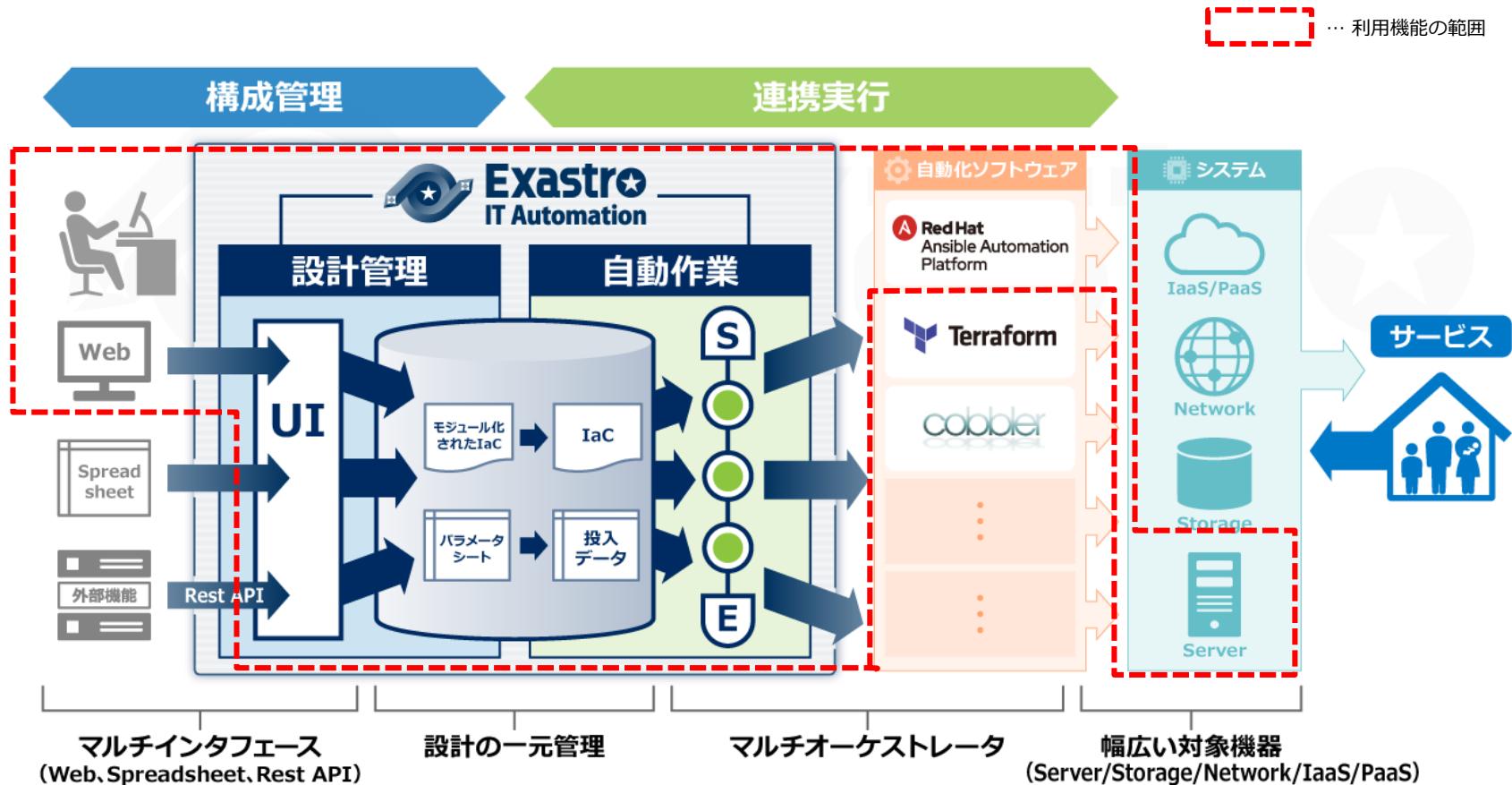
システム構築においてよくあるLinuxサーバのパッケージのインストール作業を通して、構築対象サーバごとの作業とパッケージ管理を自動化・一元管理化を行い、従来のシステム構築とは異なるITAを使用した効率的なシステムを体感できます。



# 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

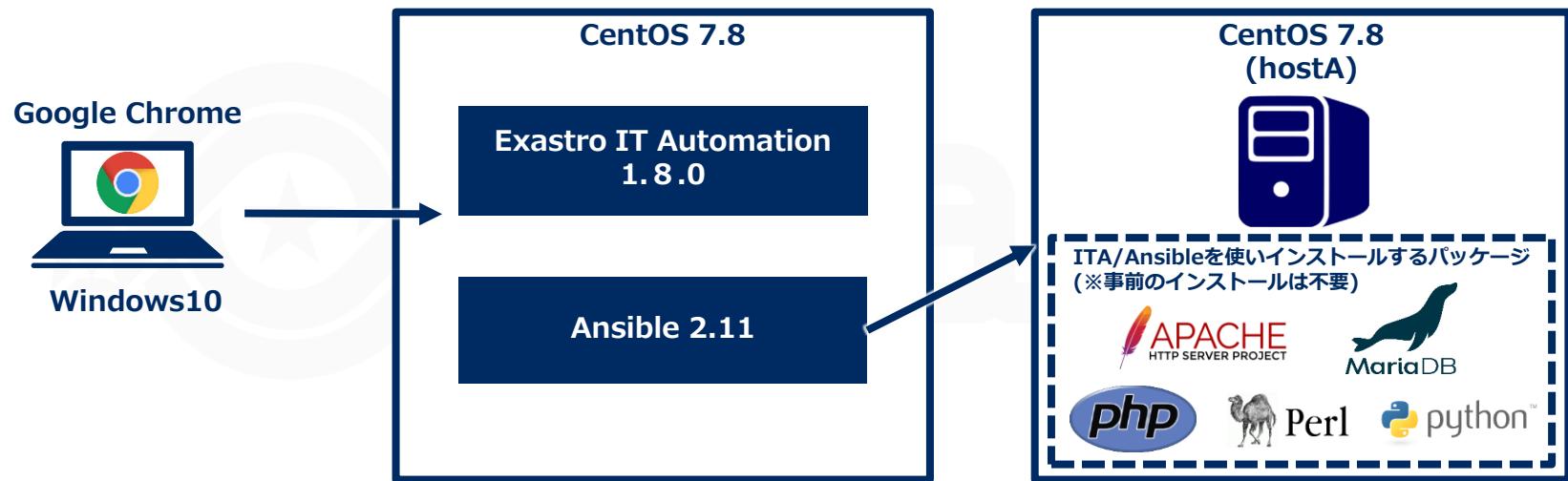
- ・自動化ソフトウェア(Ansible)との連携
- ・パラメータ管理(メニュー作成・登録・履歴管理等)
- ・変数紐付け(代入値自動登録)



# 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

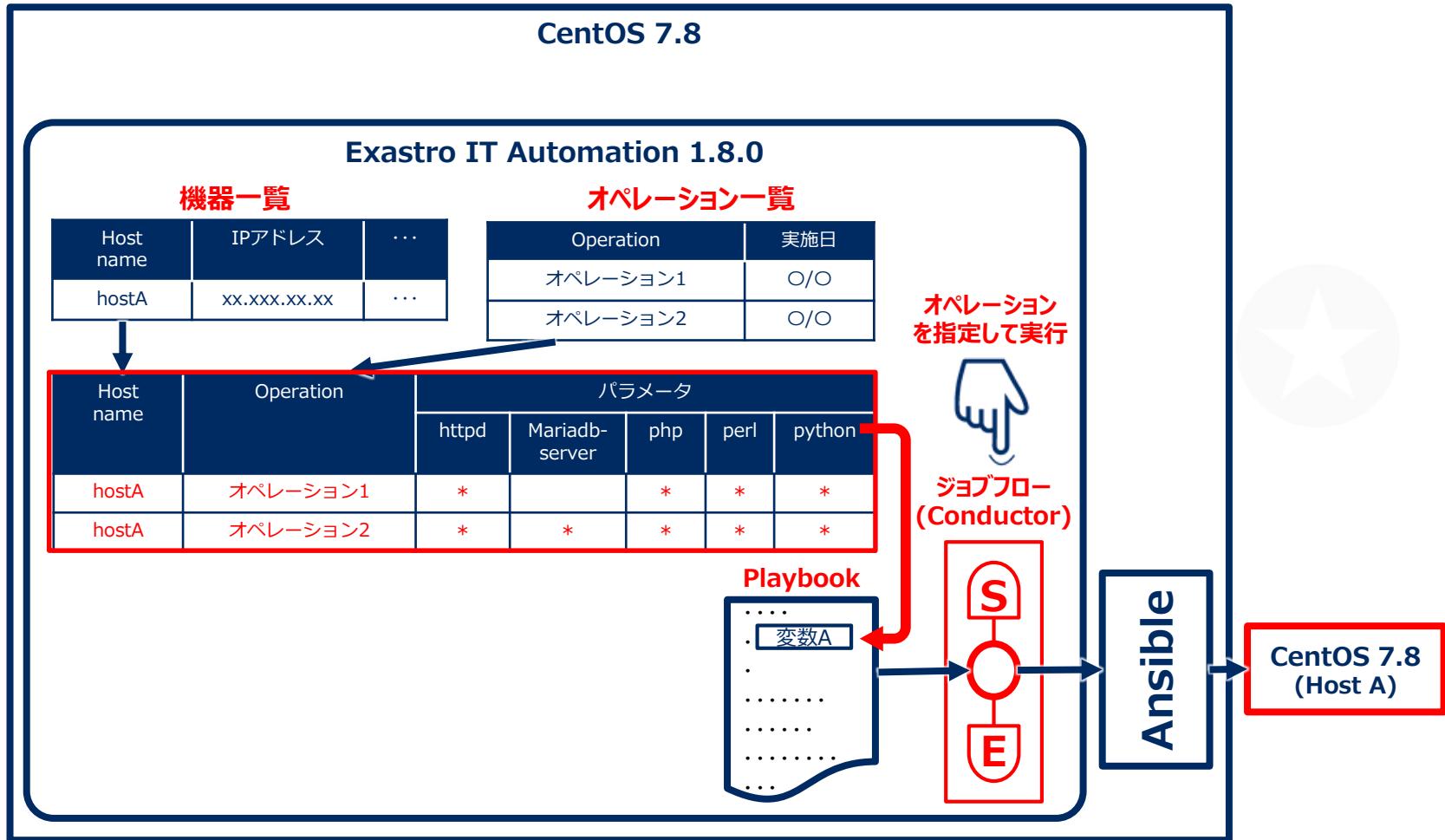


## 使用するシステム

- Exastro IT Automation 1.8.0
- 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 (2/3)

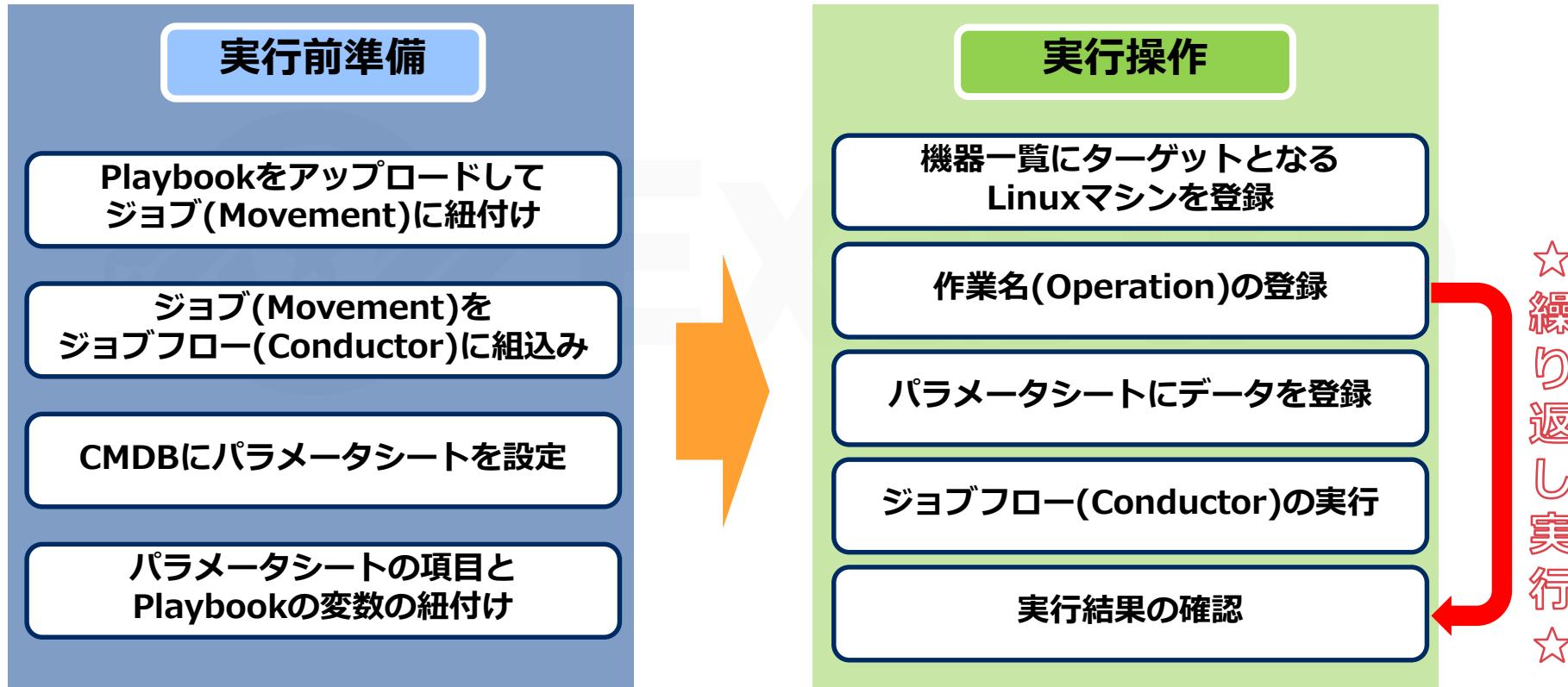
## シナリオの実行イメージ



# 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)／作業者(実行操作)の作業範囲については以下の通り。



# 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<br>(Oparation) | 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 と呼ぶ各種パートを組み合わせて、ワークフローを作成し、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。   |

用語の説明については[ファーストステップガイド](#)をご参照ください。

また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。



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。

メニュー

メニュー グループ

Point

各機能の詳細は  
[マニュアル](#)を参照してください。

Exastro IT Automation 管理コンソール

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

編集

Menu

DASHBOARD

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・メニュー組付管理

ロール・ユーザ組付管理

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

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 入力用

代入値自動登… 参照用 比較 ホストグルー… Ansible共通 Ansible-Legacy

Ansible-Pioneer Ansible-Lega… サーバ基本設定 サーバ基本設… Terraform メニュー作成

Movement

Status

Result

Movement SUM

Status CON SYM SUM

Result CON SYM SUM

Point

## 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。

**■ 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

**Point**

各機能の詳細は  
[マニュアル](#)を参照してください。

## 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。

The screenshot shows the Ansible-Legacy application interface. On the left is a vertical navigation menu with the following items:

- Menu
- メインメニュー
- Movement一覧
- Playbook素材集
- Movement-Playbook紐付
- 代入登録自動登録設定
- 作業対象ホスト
- 代入登録管理
- 作業実行
- 作業状態確認
- 作業管理

The main content area displays several sub-menus:

- 説明**: A sub-menu with three options: マニュアル (Manual), 表示フィルタ (Display Filter), and 一覧/更新 (List/Update).
- 登録**: A sub-menu with three options: 登録開始 (Registration Start), 全件ダウンロードとファイルアップロード編集 (Full Download and File Upload Edit), and 全件ダウソード (Full Download).
- 変更履歴**: A sub-menu with two options: 変更履歴全件ダウンロード (Change History Full Download) and 実行履歴全件ダウンロード(Excel) (Execution History Full Download (Excel)).

At the top right of the screen, there is user information: ようこそ[シナプス管理者]さん, ログインID [administrator], パスワード変更 (Password Change), and ログアウト (Logout).

A red box highlights the "説明" sub-menu, which is labeled "サブメニュー" (Sub-menu) with a red arrow pointing to it.

A red callout box contains the following text:

**■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からのIN/OUT処理
- 変更履歴** : 登録レコードの変更履歴

A red circle labeled "Point" is positioned over the "変更履歴" sub-menu.

A red callout box at the bottom right contains the text:

各機能の詳細は  
[マニュアル](#)を参照してください。

### 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”、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 interface. On the left, a sidebar lists various menu items under 'Ansible-Legacy'. The 'Movement' item is highlighted with a red box and has a red circle with the number '2' above it. In the main content area, there's a grid of icons representing different Ansible modules. One icon, labeled 'Ansible Legacy', is highlighted with a red box and has a red circle with the number '1' above it. To the right of the dashboard, a callout box contains five steps for uploading a Playbook:

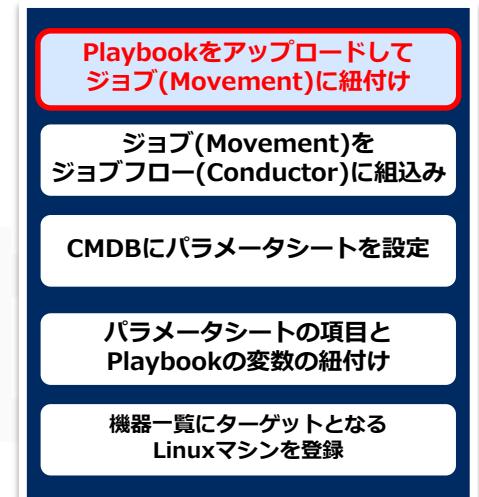
- Playbookをアップロードしてジョブ(Movement)に紐付け
- ジョブ(Movement)をジョブフロー(Conductor)に組込み
- CMDBにパラメータシートを設定
- パラメータシートの項目とPlaybookの変数の紐付け
- 機器一覧にターゲットとなるLinuxマシンを登録

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

The screenshot shows the Ansible-Legacy interface with the title 'Movement一覧'. On the left is a sidebar with various menu items. The main area has a search bar and a table with columns: 廃止, MovementID, Movement名, オーケストレータ, 遅延タイマー, 最終更新日時, and 最終更新者. Below the table are two buttons: 'フィルタ' and 'フィルタクリア'. At the bottom of the registration form, there is a red box around the '登録開始' button.



The screenshot shows the '登録' (Registration) screen. It includes a table with columns: MovementID, Movement名, 遅延タイマー, and ホスト指定形式\*. The 'Movement名' field contains 'パッケージインストール' and the 'ホスト指定形式\*' dropdown is set to 'IP'. At the bottom of the form, there are two buttons: '戻る' and '登録'. A red box highlights the '登録' button. To the right, a callout box shows the input values: Movement名: パッケージインストール and ホスト指定形式: IP.

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

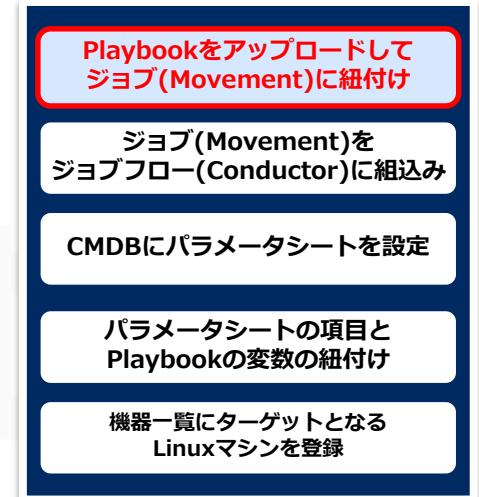
## 「Playbook素材集」へ新規Playbookを登録

- 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 Ansible-Legacy interface with the 'Playbook素材集' menu item highlighted. A red box labeled '1' points to the 'Playbook素材名' field in the registration dialog, which contains 'yum\_package\_install'. Another red box labeled '2' points to the 'Playbook素材' field, which contains 'yum\_package\_install.yml'. A red box labeled '3' points to the '登録' (Register) button at the bottom of the dialog. A red callout bubble labeled 'Point' points to the '事前アップロード' (Pre-upload) button in the file selection dialog, with 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, please click the "Pre-upload" button after specifying the file).



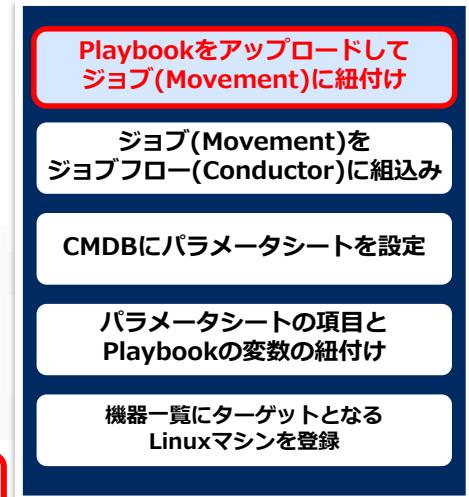
# 3.1 Playbookをアップロードしてジョブ(Movement)に紐付け (5/5)

## 「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。

| 組付項目番号 | Movement    | Playbook素材          | インクルード順序 |
|--------|-------------|---------------------|----------|
| 自動入力   | パッケージインストール | yum_package_install | 1        |



| 組付項目番号 | Movement    | Playbook素材          | インクルード順序 |
|--------|-------------|---------------------|----------|
| 自動入力   | パッケージインストール | yum_package_install | 1        |

※\*は必須項目です。

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

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。



### 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  
インストールパッケージ一覧

| 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    |



### 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)

項目が5個になるようにクリック

| 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:*(-ブランク) |
| mariadb-server | プルダウン選択 | メニュー作成:選択1:*(-ブランク) |
| php            | プルダウン選択 | メニュー作成:選択1:*(-ブランク) |
| perl           | プルダウン選択 | メニュー作成:選択1:*(-ブランク) |
| python         | プルダウン選択 | メニュー作成:選択1:*(-ブランク) |

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

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(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。



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)

The screenshot shows the Ansible-Legacy interface with the 'Parameter Sheet' configuration screen. A red box highlights the table where values are being mapped from the 'Movement' column to the 'Variable Name' column.

| 項目番号 | パラメータシート(From) | 項目                      | 最終更新日時 | 最終更新者 |
|------|----------------|-------------------------|--------|-------|
| 1    | メニュー:メニュー      | パラメータ/インストールパッケージ/httpd | 自動入力   | 自動入力  |

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:メニュー                           | 項目                               | 登録方式 | Movement      | Key変数<br>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。

Exastro IT Automation Ansible-Legacy

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  
Movement一覧  
Playbook素材集  
Movement-Playbook紐付  
代入値自動登録設定  
作業対象リスト  
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  
一覧/更新  
登録

IaC変数(To)

| 項目番号 | 登録方式* | Movement*     | Key変数          |      | 最終更新日時 | 最終更新者 |
|------|-------|---------------|----------------|------|--------|-------|
|      |       |               | 変数名            | 代入順序 |        |       |
| 自動入力 | Key型  | 1:パッケージインストール | 1:VAR_packages | 1    | 自動入力   | 自動入力  |

\* \*は必須項目です。

戻る **登録** 2

△閉じる

▽開く

▽開く

▽開く

△閉じる

▽開く

全件ダウンロードとファイルアップロード編集

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。

3  
4  
5 5件のデータが登録できているかを確認  
一 更新

| ID | 名前            | ID | 名前                         | 登録方式           | Movement | 変数名 | Value変数 | NULL連携 | 最終更新日時              | 最終更新者            |
|----|---------------|----|----------------------------|----------------|----------|-----|---------|--------|---------------------|------------------|
| 1  | インストールパッケージ-1 | 1  | パッケージインストール                | 1:VAR_packages | 1        |     |         |        | 2021/07/15 15:33:22 | データポータビリティプロシージャ |
| 2  | インストールパッケージ-2 | 2  | パッケージインストール                | 1:VAR_packages | 2        |     |         |        | 2021/07/15 15:33:54 | データポータビリティプロシージャ |
| 3  | インストールパッケージ-3 | 3  | パッケージインストール 1:VAR_packages | 3              |          |     |         |        | 2021/07/15 15:34:47 | データポータビリティプロシージャ |
| 4  | インストールパッケージ-4 | 4  | パッケージインストール 1:VAR_packages | 4              |          |     |         |        | 2021/07/15 15:35:18 | データポータビリティプロシージャ |
| 5  | インストールパッケージ-5 | 5  | パッケージインストール 1:VAR_packages | 5              |          |     |         |        | 2021/07/15 15:36:06 | データポータビリティプロシージャ |

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)

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

フィルタ

登録

△閉じる

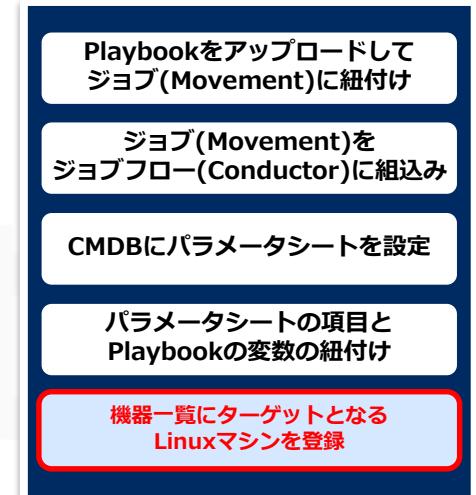
管理システム項目番号 HW機器種別 ホスト名\* IPアドレス\* EtherWakeOnLan MACアドレス ネットワークデバイス名 ログ 最終更新日時 最終更新者

自動入力 SV hostA 192.168.10.1 自動入力 自動入力

※\*は必須項目です。

| 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) |

管理者に連絡



# 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]  
[パスワード変更] [ログアウト]

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

管理システム項目番号 ログインユーザID ログインパスワード ssh鍵認証情報

| 管理システム項目番号 | 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   | 最終更新日時 | 最終更新者 |
|------------|-----------|----|-----------|--|--------|-------|
| 自動入力       | root      | ●  | *****     | ssh秘密鍵ファイル<br>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: | 自動入力   | 自動入力  |

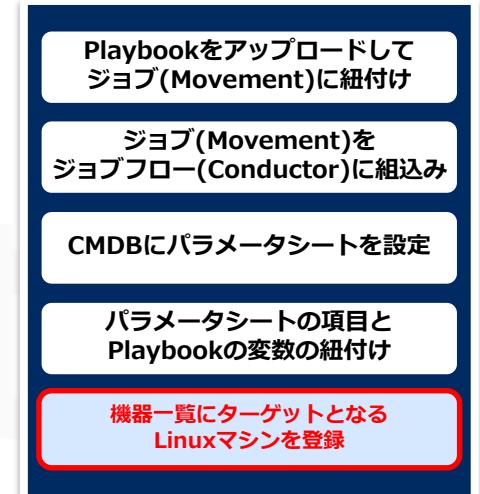
\*<sup>\*</sup>は必須項目です。

2

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

変更履歴 ▽開く

管理者に連絡



# 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]

3

Legacy/Role利用情報  
認証方式  
パスワード認証

△閉じる

Ansible利用情報  
Legacy/Role利用情報  
WinRM接続情報  
最終更新日時 最終更新者

管理システム項目 認証方式 ポート番号 サーバー証明書 フ

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  
事前アップロード  
アップロード状況:

※\*は必須項目です。

戻る 登録

4

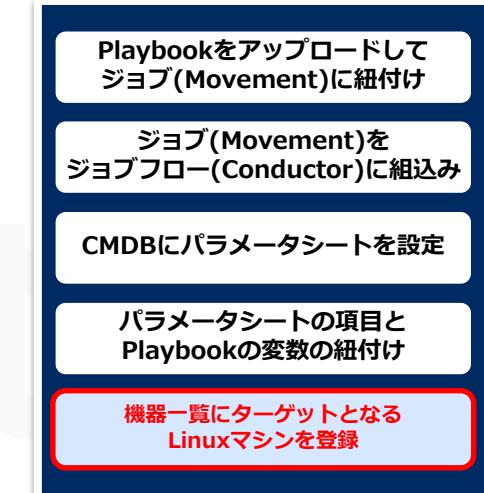
全件ダウンロードとファイルアップロード編集  
変更履歴

管理者に連絡

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。

The screenshot shows the Exastro basic console interface. On the left, there's a sidebar with a menu. The 'Operations List' item is highlighted with a red box and a red number '1'. The main area is titled 'Operation List' and contains a search form with dropdown menus for filtering by status, ID, name, date, and user. Below the search is a 'Filter' button and a 'Clear Filter' button. A note at the bottom says '\* \* is a required field'. At the bottom right of the main form is a 'Register' button, which is also highlighted with a red box and a red number '3'. To the right of the main form, a callout box highlights the 'Operation Name' and 'Implementation Date' fields, both of which have 'Operation 1' and '(Any implementation date)' entered respectively. Red numbers '2' and '3' correspond to the numbered steps in the adjacent text block.

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(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。

Exastro IT Automation 入力用

説明  
表記  
一覧/更新  
登録

| No | ホスト名  | オペレーション                     | パラメータ                                | アクセス権 | 最終更新日時    | 最終更新者 |
|----|-------|-----------------------------|--------------------------------------|-------|-----------|-------|
| 1  | hostA | 2021/07/16 10:45_3:オペレーション1 | httpd mariadb-server php perl python | 設定    | アクセス許可ロール | 自動入力  |

※\*は必須項目です。

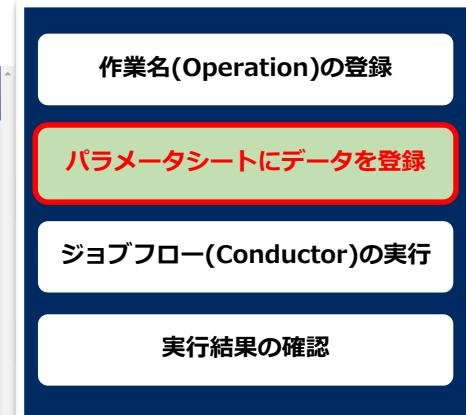
戻る **登録** 3

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

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 \* \* \*

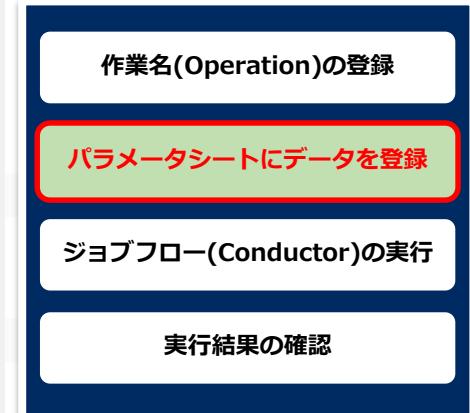


## 4.3 パラメータシートにデータを登録(2/2)

### 「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。

The screenshot shows the Exastro IT Automation interface. The top navigation bar includes 'Exastro IT Automation 入力用' and 'ログインID [administrator] ログアウト'. The main toolbar has tabs like '説明', 'ホストフィルタ', 'オペレーション', 'パラメータ', 'アクセス権', and '変更履歴'. A red box labeled '1' highlights the 'ホストフィルタ' tab. A red box labeled '2' highlights the 'Filter' button in the toolbar. A red box labeled '3' highlights the 'オートフィルタ' checkbox in the 'Filter' section. The main content area displays a table titled '一覧/更新' with columns for '操作', '複数', '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', 'パラメータ', 'アクセス権', '最後更新日時', and '最終更新者'. One row is selected, showing details: ID 1, オペレーション名 3\_オペレーション1, 基準日時 2021/08/19 18:26, 実施予定日時 2021/07/16 10:45, 最終実行日時 2021/08/19 18:26, パラメータ インストールパッケージ http://mariadb-server.php perl python, アクセス権 アクセス許可ロール, 最終更新日時 2021/08/19 18:25:48, 最終更新者 管理者. The bottom of the screen features a 'Excel出力' button and a '登録' section with buttons for '全件ダウンロードとファイルアップロード編集' and '変更履歴'.



# 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。

The screenshot shows the Exastro IT Automation Conductor interface with the following steps highlighted:

- 1**: Click on the "Conductor作業実行" button in the left sidebar.
- 2**: Select the "Conductor" from the list of available Conductors.
- 3**: Select the "オペレーション" (Operation) from the list of available Operations.
- 4**: Click the "EXECUTE" button at the bottom of the screen.

On the right side, there is a vertical stack of five boxes representing the workflow:

- 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** (This box is highlighted in green, indicating the current step)
- 実行結果の確認

The "Conductor実行" window shows a simple workflow diagram with three nodes: Start, Package Install, and End.

## 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。

The screenshot shows the Exastro IT Automation Conductor interface. On the left, there's a navigation menu with items like 'Menu', 'Checking' (which is selected), 'Main Menu', 'Conductor Interface Information', 'Conductor Cluster Management', 'Conductor Class Collection', 'Conductor Job Execution', 'Conductor Job Confirmation', 'Conductor Job Log', and 'Conductor Periodic Job Execution'. The main area displays a job flow diagram with nodes: 'Start' (Conductor), 'Ansible legacy' (DONE), and 'End' (Conductor). The 'Ansible legacy' node has the label 'パッケージインストール'. Below the diagram is a log window with three icons: 'Mouse click on Node selection / Operation status confirmation', 'Mouse wheel / Zoom in / Out', and 'Mouse right drag / Move the screen'. To the right of the diagram is a detailed status panel. At the top of this panel, it says 'ようこそ[システム管理者]さん ログインID [administrator]' with 'ログアウト' (Logout) and 'パスワード変更' (Change Password) buttons. Below this are buttons for '全表示' (All Display), '表示リセット' (Reset Display), and 'フルスクリーン' (Full Screen). The status panel contains fields for 'Conductor name' (Conductor instance ID: 17, Name: パッケージインストール, Status: 正常終了, Start time: 2021/08/19 18:26:15, End time: 2021/08/19 18:27:36, Execution user: システム管理者, Reservation date:, Emergency stop:), 'Note' (empty), and 'Operation' (Operation ID: 3, Operation name: オペレーション1). A red circle labeled 'Point' points to the 'Operation' section. To the right of the main interface are four callout boxes: '作業名(Operation)の登録' (Job name (Operation) registration), 'パラメータシートにデータを登録' (Register data to parameter sheet), 'ジョブフロー(Conductor)の実行' (Execute job flow (Conductor)), and '実行結果の確認' (Check execution results).

Point

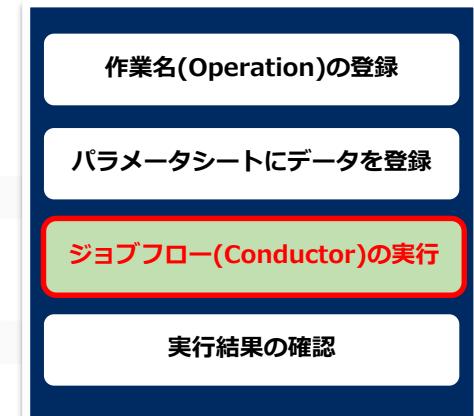
実行ステータスやログをリアルタイムで確認可能です。

## 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. On the left, there's a workflow diagram with nodes labeled 'Start', 'Ansible Legacy', and 'End'. The 'Ansible Legacy' node has a green 'DONE' button highlighted with a red box and a red circle containing the number '1'. A red arrow points from this button to a detailed view window on the right. This window displays information for a 'node-3' movement, including its status as '正常終了' (Normal Completion), start time (2021/08/19 18:26:15), end time (2021/08/19 18:27:32), and operation status (正常終了確認). Below this, a sub-menu for 'Ansible-Legacy' is open, showing various configuration options like 'Movement' and 'Playbook'. At the bottom of the main window, there's a log section.



# 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。

The screenshot shows the Exastro UI interface. On the left, there's a vertical sidebar with a 'Menu' section containing various options like 'メインメニュー', 'Movement一覧', 'Playbook収集庫', etc. Below that is a '作業実行' section with '作業状態確認'. The main area has a table for 'オペレーション' (Operation) details:

| No.     | 名前    | 3                         |
|---------|-------|---------------------------|
| オペレーション | 名前    | オペレーション1                  |
| ID      |       | 3                         |
| 作業対象ホスト |       | <button>確認</button>       |
| 代入値     |       | <button>確認</button>       |
| 入力データ   | 入力データ | InputData_0000000027.zip  |
| 出力データ   | 結果データ | ResultData_0000000027.zip |
| 予約日時    |       |                           |
| 作業状況    | 開始日時  | 2021/08/19 18:26:21       |
|         | 終了日時  | 2021/08/19 18:27:28       |

A large red box highlights a modal window titled '進行状況(実行ログ)' (Execution Status (Execution Log)). This window contains the Ansible execution log output:

```
Upgrading : python-2.7.5-90.el7.x86_64          2/4
Cleanup   : python-2.7.5-90.el7.x86_64          3/4
Cleanup   : python-11bs-2.7.5-68.el7.x86_64      4/4
Verifying  : python-2.7.5-90.el7.x86_64          1/4
Verifying  : python-11bs-2.7.5-68.el7.x86_64      2/4
Verifying  : python-11bs-2.7.5-68.el7.x86_64      3/4
Verifying  : python-2.7.5-90.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
```

Below the log, there are two input fields for filtering log entries. To the right of the main window, there's a vertical panel with four sections:

- 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 (highlighted with a red box)

# 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

```
~~~~~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。

The screenshot shows the Exastro IT Automation basic console interface. The left sidebar has a red box around the 'オペレーション一覧' (Operations List) item, which is highlighted with a red circle labeled '1'. The main content area shows a search filter dialog with several dropdown menus and search buttons. Below it is a table for listing operations, with a specific row highlighted by a red box and red circle labeled '2'. A callout box labeled '3' points to the '登録' (Register) button at the bottom of the table. To the right, there are three green callout boxes: '作業名(Operation)の登録' (Operation registration), 'パラメータシートにデータを登録' (Register data to parameter sheet), and 'ジョブフロー(Conductor)の実行' (Execute job flow).

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

三 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

廃止含まず

~

▼ ブラウザ検索

~

▼ ブラウザ検索

~

▼ ブラウザ検索

Table setting

filtration

filtration clear

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\*

アクセス権

設定

アクセス許可

備考

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

2021/07/08 18:00

登録

\*<sup>\*</sup>は必須項目です。

オペレーション名

実施予定日時

オペレーション2

(任意の日時)

全件ダウンロードとファイルアップロード編集

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

## 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。

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]  
パスワード変更 ログアウト

説明  
表示  
一覧  
登録

△開く  
▽開く  
▽開く  
△閉じる

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

1  
2  
3

| No   | ホスト名* | オペレーション*                    | パラメータ                                | インストールパッケージ | 最終更新日時 | 最終更新者 |
|------|-------|-----------------------------|--------------------------------------|-------------|--------|-------|
| 自動入力 | hostA | 2021/07/27 12:00_4_オペレーション2 | httpd mariadb-server php perl python | 自動入力 自動入力   |        |       |

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

| 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:<br>オペレーション2 | *     | *              | *   | *    | *      |

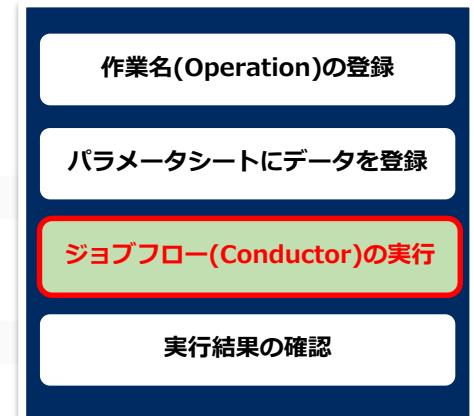
# 5.3 ジョブフロー(Conductor)の実行(1/3)

## 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。

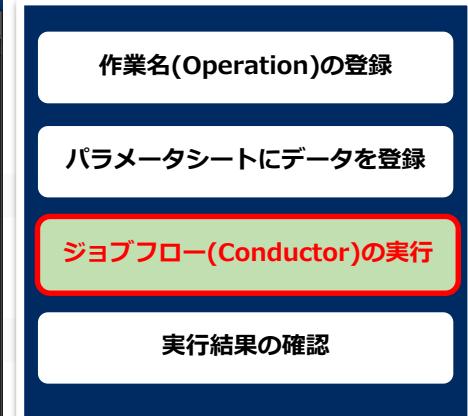
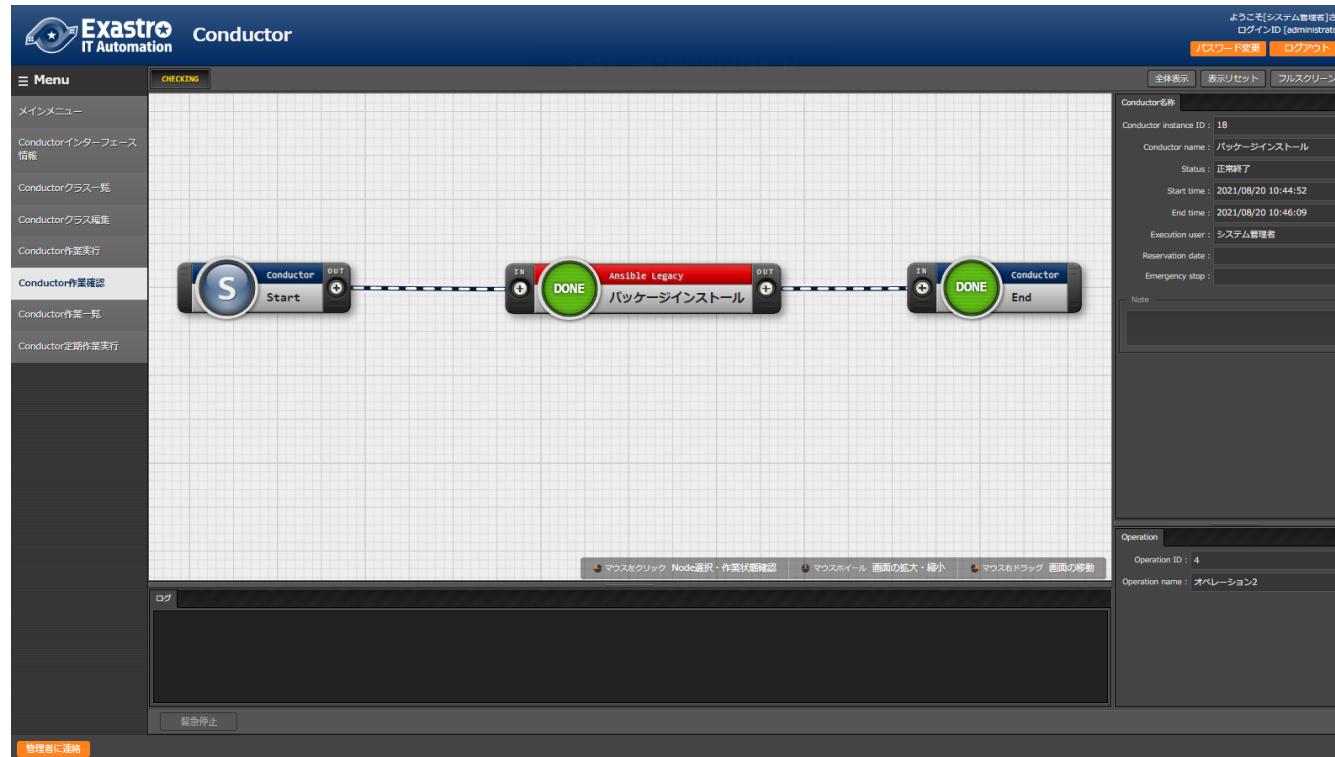
The screenshot shows the Exastro IT Automation Conductor interface. On the left, a sidebar menu lists various Conductor-related options. A red box labeled 1 highlights the 'Conductor作業実行' (Conductor Job Execution) option. The main content area has two tabs: 'Conductor[フィルタ]' (Conductor [Filter]) and 'Conductor[一覧]' (Conductor [List]). A red box labeled 2 surrounds the 'Conductor[一覧]' tab, which displays a table of Conductor entries. The table columns include '選択' (Select), 'ConductorクラスID' (Conductor Class ID), 'Conductor名称' (Conductor Name), '説明' (Description), 'アクセス権' (Access Rights), '備考' (Remarks), '最終更新日時' (Last Updated Date), and '最終更新者' (Last Updated By). The first entry in the table is highlighted with a red box. Below the table, a message indicates 'フィルタ結果件数: 1'. The bottom section contains tabs for 'オペレーション[フィルタ]' (Operation [Filter]) and 'オペレーション[一覧]' (Operation [List]), with a red box labeled 3 surrounding the 'オペレーション[一覧]' tab. This tab also displays a table of operations, with the second entry highlighted by a red box. At the bottom right of the interface, a red box labeled 4 surrounds the '実行' (Execute) button.



# 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。



Point

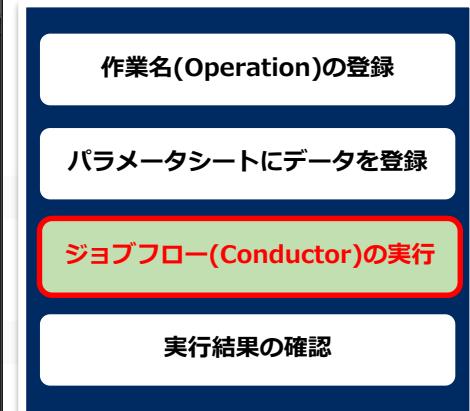
実行ステータスやログを  
リアルタイムで  
確認可能です。

# 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. On the left, there's a navigation menu with items like 'Menu', 'Conductor', 'Ansible Legacy', 'Ansible-Legacy', 'Movement', 'Job Status', 'Job Management', and 'Logs'. The main area displays a workflow diagram with nodes: 'Start' (Conductor), 'Ansible Legacy' (DUT), and 'End' (Conductor). The 'Ansible Legacy' node has a green 'DONE' button. A red box highlights this button with the number '1' and the text '選択' (Select). A red arrow points from this box to a detailed job execution log window titled 'Exastro IT Automation Ansible-Legacy'. This window shows a table with various parameters and their values, such as '作業名' (Job Name), '実行機別' (Execution Type), 'ステータス' (Status), and '実行エンジン' (Execution Engine). The status column for the 'Ansible Engine' row is '完了' (Completed). Another red arrow points from the 'Ansible Legacy' node in the main diagram to this log window.



# 5.4 実行結果の確認(1/2)

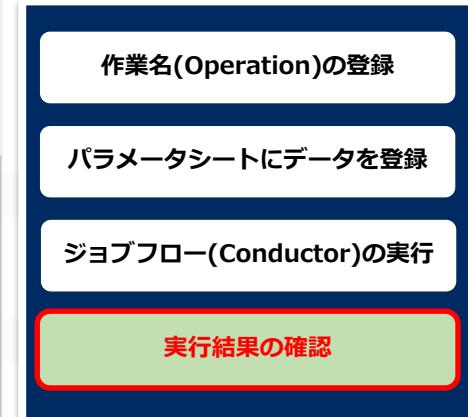
## 作業結果確認

- 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。

The screenshot shows the Ansible execution log details. On the left, there's a sidebar with various menu items. In the center, a table displays input and output data, and a log viewer window is open. The log viewer has two tabs: '進行状況(実行ログ)' and '進行状況(エラーログ)'. The '進行状況(実行ログ)' tab is selected and contains the following Ansible log output:

```
        "update_cache": false,
        "update_only": false,
        "use_backed": "auto",
        "validate_certs": true
    },
    "item": "python",
    "msg": "",
    "rc": 0,
    "results": [
        "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
```

The '進行状況(実行ログ)' tab is highlighted with a red box.



# 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

```
~~~~~省略~~~~~  
"results": ["All packages providing httpd are up to date",  
  ""}]  
~~~~~省略~~~~~  
Installed:  
  mariadb-server.x86_64 1:5.5.68-1.el7  
Dependency Installed:  
  mariadb.x86_64 1:5.5.68-1.el7  
  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  
Dependency Updated:  
  mariadb-libs.x86_64 1:5.5.68-1.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

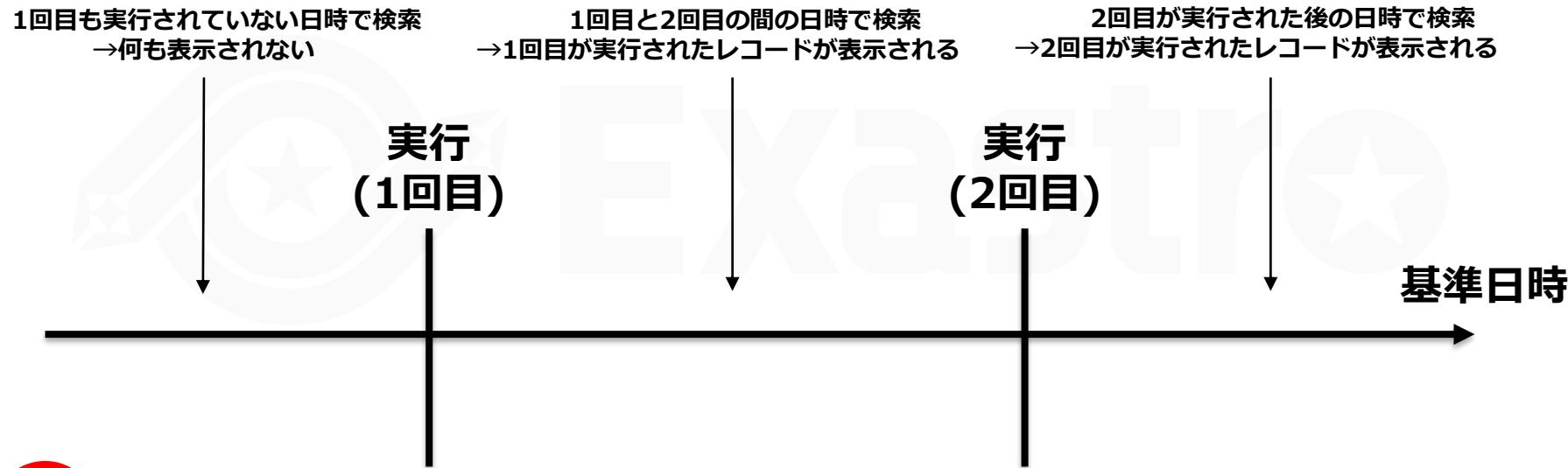
実行結果の確認

## 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)

The screenshot shows the Exastro IT Automation interface with the following steps highlighted:

1. The '参照用' (Reference) menu group is selected.
2. The 'インストールパッケージ一覧' (Installed Package List) menu item is selected. A red box and arrow point to the 'フィルタ' (Filter) button.
3. The 'オートフィルタ' (Auto Filter) checkbox is checked.
4. The '一覧' (List) view displays a table of installed packages. A red box highlights the first row, which includes columns for History, ID, Host Name, Operation Name, Baseline Date, Implementation Date, Final Execution Date, Parameters, and Access. The 'History' column shows the value '履歴' (History). A red box and arrow point to the 'Excel出力' (Excel Output) button.

| 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 |   |        |                             |
| 履歴 | 2  | hostA | 4       | オペレーション2 | 2021/08/20 10:44 | 2021/07/27 12:00 | 2021/08/20 10:44 | *           | *      | * | *      | 2021/08/20 11:38:13 システム管理者 |

# 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)

The screenshot shows the Exastro IT Automation interface with the following steps highlighted:

- 1** A red circle highlights the "基準日時" (Baseline Date) input field in the "表示フィルタ" (Display Filter) section. A callout box contains the text: "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** A red circle highlights the "フィルタ" (Filter) button in the "表示フィルタ" section.
- 3** A red circle highlights the first row in the "履歴" (History) table in the "一覧" (List) section. A callout box contains the text: "1回目の実行結果が表示されることを確認する".

**表示フィルタ** (Display Filter) section details:

| ホスト名  | オペレーション   | 基準日時             |
|-------|-----------|------------------|
| hostA | オペレーション 1 | 2021/08/20 10:30 |

**一覧** (List) section details:

| 履歴 | 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             | パラメータ                                | アクセス | 最終更新日時 | 最終更新者               |         |
|----|----|-------|----|-----------|------------------|------------------|--------------------|--------------------------------------|------|--------|---------------------|---------|
| 履歴 | 1  | hostA | 3  | オペレーション 1 | 2021/08/19 18:26 | 2021/08/16 10:45 | 2021/08/19 18:26 * | httpd mariadb-server php perl python | *    | *      | 2021/08/19 18:25:40 | システム管理者 |

## 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 with the following steps highlighted:

- A red circle labeled "1" points to the "基準日時" (Baseline Date) input field in the "表示フィルタ" (Display Filter) panel. A callout box contains the text: "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".
- A red circle labeled "2" points to the "フィルタ" (Filter) button in the "表示フィルタ" panel.
- A red circle labeled "3" points to the "変更履歴" (Change History) section in the main content area, which displays the message "レコードはありません" (No records found). A callout box contains the text: "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 Ansible-Legacy interface under the 'Execution' menu. It displays three main steps:

- 1** Movement selection: A red box highlights the 'Movement' section where 'MovementID 1' (名前: パッケージインストール, 状態: Ansible Legacy) is selected.
- 2** Operation selection: A red box highlights the 'Operation' section where 'OperationID 1' (名前: オペレーション1, 状態: 未実行) is selected.
- 3** Dry Run: A red box highlights the 'Dry Run' button in the bottom navigation bar.

**作業実行**

**Movement[一覧]**

| 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   |
|------------|-------------|----------------|--------|-------------|---------------------|---------|
| 1          | パッケージインストール | Ansible Legacy |        | IP          | 2021/04/19 14:43:47 | システム管理者 |

**オペレーション[一覧]**

| No. | オペレーションID | オペレーション名   | 実施予定期間                              | 実施実行日時              | アクセス権            | 権限 | 最終更新日時 | 最終更新者 |
|-----|-----------|------------|-------------------------------------|---------------------|------------------|----|--------|-------|
| 1   | 1         | 1 オペレーション1 | 2021/05/02 12:00 - 2021/05/12 14:49 | 2021/05/12 14:49:48 | legacy作業実行プロシージャ |    |        |       |
| 3   | 3         | 3 オペレーション2 | 2021/07/08 18:00 - 2021/05/13 20:13 | 2021/05/13 20:13:48 | legacy作業実行プロシージャ |    |        |       |

**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

# 参考② 【Ansible-Legacy】 実行確認

## 作業結果確認

- 実行(またはドライラン)すると画面が切替わり、実行ステータスやログが表示されます。

The screenshot shows the Exastro IT Automation interface with the Ansible-Legacy module selected. The main window displays the execution details of a task, including movement information, operation details, and job status.

**Point**: 実行ステータスや、投入データが確認可能です。

**Point**: 実行ログやエラーログをリアルタイムで確認可能です。

**Point**: 投入データや結果データをダウンロードすることも可能です。

**Table: Movement Details**

| 項目           | 値                                  |
|--------------|------------------------------------|
| 作業No.        | 1                                  |
| 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 | 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           | 1                                  |
| 名称           | パッケージインストール                        |
| 遅延タイム(分)     |                                    |
| Ansible利用情報  |                                    |
| ホスト指定形式      | IP                                 |
| No.          | 1                                  |
| オペレーション      |                                    |
| 名称           | オペレーション1                           |
| ID           | 1                                  |
| 作業対象ホスト      |                                    |
| 確認           |                                    |
| 代入値          |                                    |
| 入力データ        | 投入データ<br>InputData_0000000001.zip  |
| 出力データ        | 結果データ<br>ResultData_0000000001.zip |
| 予約日時         |                                    |
| 作業状況         |                                    |
| 開始日時         | 2021/04/22 14:04:45                |
| 終了日時         | 2021/04/22 14:05:04                |

**Table: Job Status Log**

| ステータス     | 説明                         | 回数  |
|-----------|----------------------------|-----|
| Setting   | python-2.7.5-98-e17-0b6_34 | 2/4 |
| Status    | python-2.7.5-98-e17-0b6_34 | 1/4 |
| Verifying | python-2.7.5-98-e17-0b6_34 | 4/4 |
| Verifying | python-2.7.5-98-e17-0b6_34 | 1/4 |
| Verifying | python-2.7.5-98-e17-0b6_34 | 3/4 |
| Verifying | python-2.7.5-98-e17-0b6_34 | 4/4 |

**Table: Job Status Error Log**

| エラー内容                      |
|----------------------------|
| python-2.7.5-98-e17-0b6_34 |



**Exastro**