



# IT Automation Quickstart

※ “IT Automation” will be written as “ITA” in this document

Exastro IT Automation Version 1.9  
Exastro developer

# Table of contents

1. Introduction
  - 1.1 [Introduction](#)
  - 1.2 [Scenario overview](#)
  - 1.3 [Terminology](#)
2. Screen Description
  - 2.1 [Web console screen description \(Login\)](#)
  - 2.2 [Screen Description \(Main Menu\)](#)
  - 2.3 [Screen Description \(Menu\)](#)
3. Preparation
  - 3.1 [Uploading Playbook and Linking it to a job \(Movement\)](#)
  - 3.2 [Implementing job \(Movement\) into Jobflow \(Conductor\)](#)
  - 3.3 [Configure CMDB Data sheet and Parameter sheet](#)
  - 3.4 [Register Data to Datasheet](#)
  - 3.5 [Link Parameter sheet item to Playbook variable](#)
4. Execution(First time)
  - 4.1 [Register target \(Linux machine\) to Device list](#)
  - 4.2 [Register Operation name \(Operation\)](#)
  - 4.3 [Register data to Parameter sheet](#)
  - 4.4 [Execute Jobflow \(Conductor\)](#)
  - 4.5 [Check Execution results](#)

# Table of Contents

- 5 . Execution (Second time)
    - 5.1 [Register Operation name \(Operation\)](#)
    - 5.2 [Register data to Parameter sheet](#)
    - 5.3 [Execute Jobflow \(Conductor\)](#)
    - 5.4 [Check Execution results](#)
  - 6 . Checking the CMDB Parameter history
    - 6.1 [Executing operations and History Management](#)
    - 6.2 [Checking the CMDB Parameter history](#)
- A Appendix
- Reference① [【Ansible-Legacy】 Single Execution](#)
- Reference② [【Ansible-Legacy】 Execution check](#)



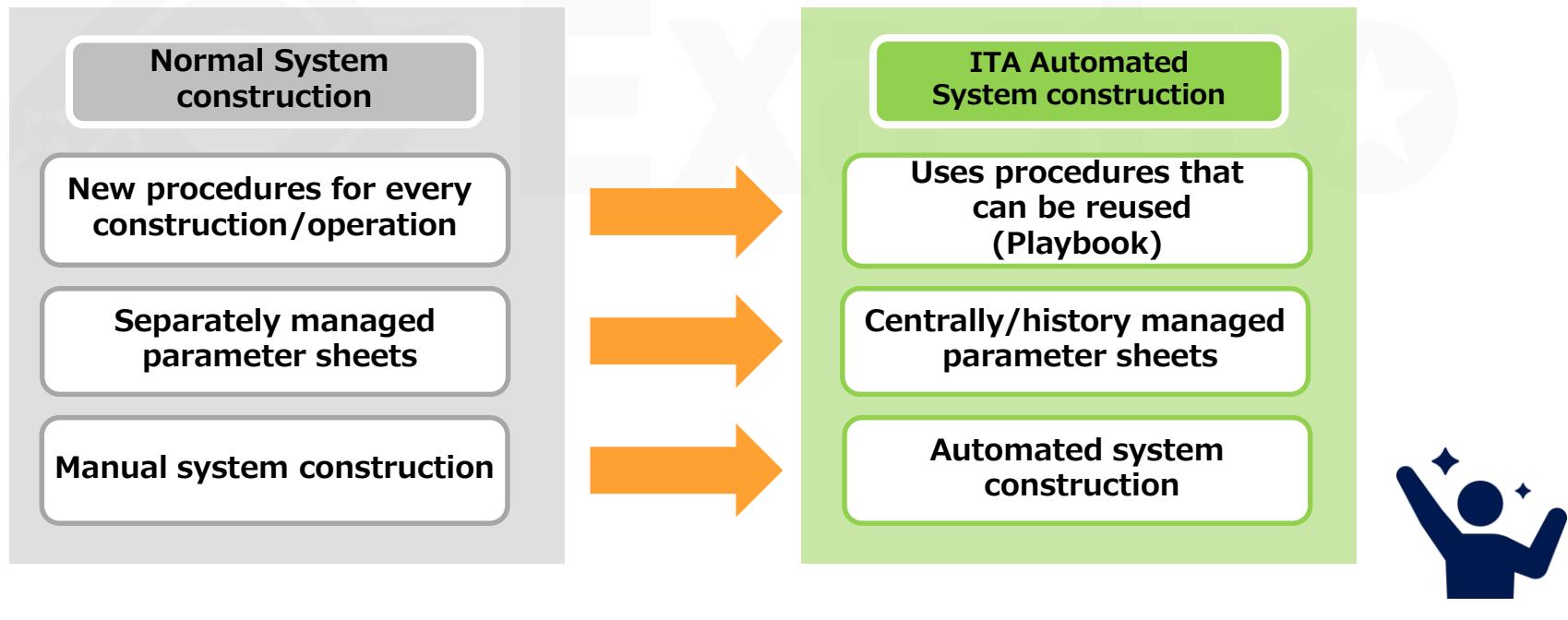
# 1. Introduction



# 1.1 Introduction (1/2)

This document serves as a quick start guide for users who are using IT Automation (written as ITA) for the first time.

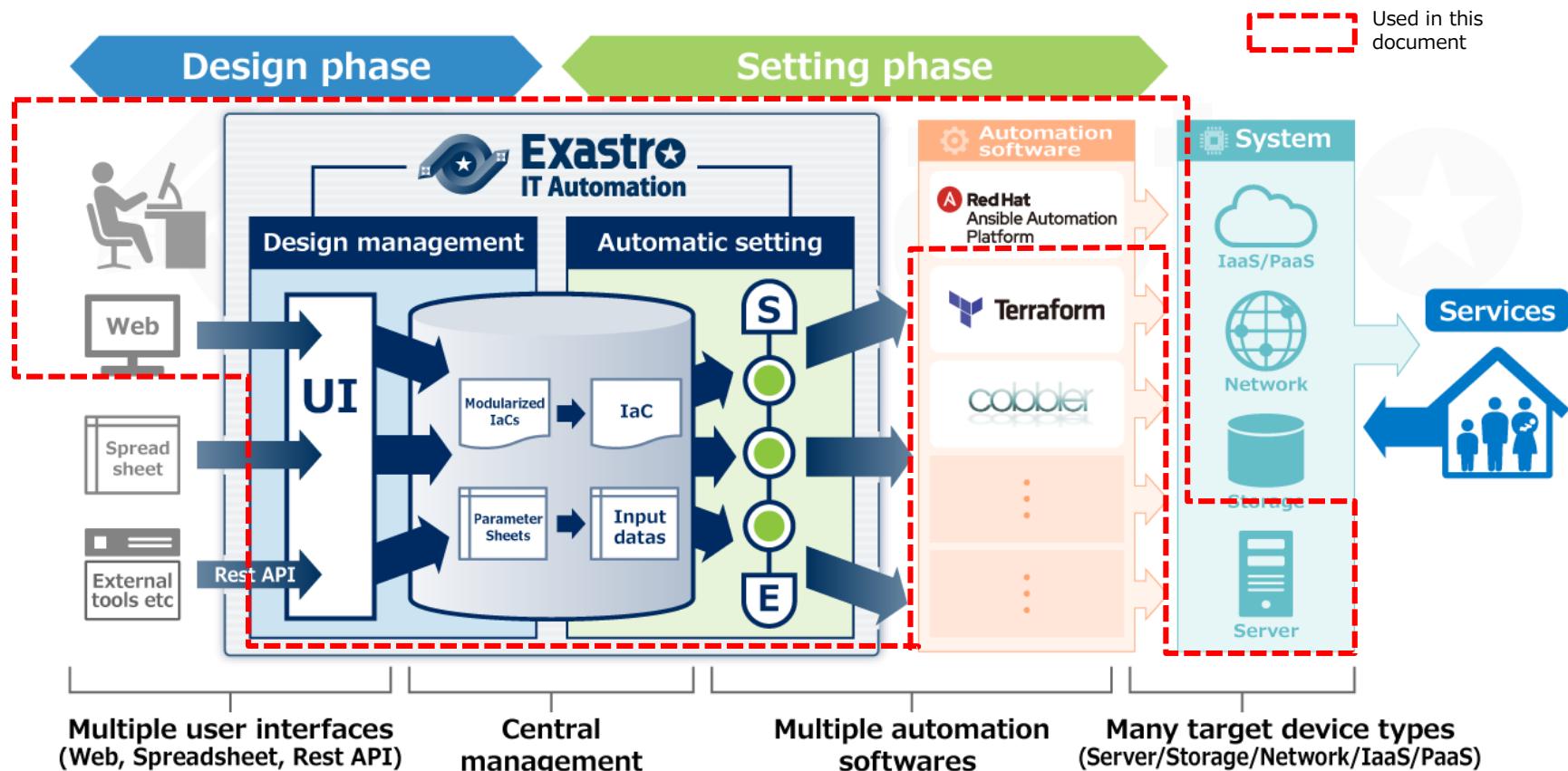
By installing Linux server packages, we can automate and centralize and automate package management for each server. That way, we can use ITA and have a more efficient system than we could achieve from a conventional system construction.



# 1.1 Introduction (2/2)

## Main ITA functions used in this document.

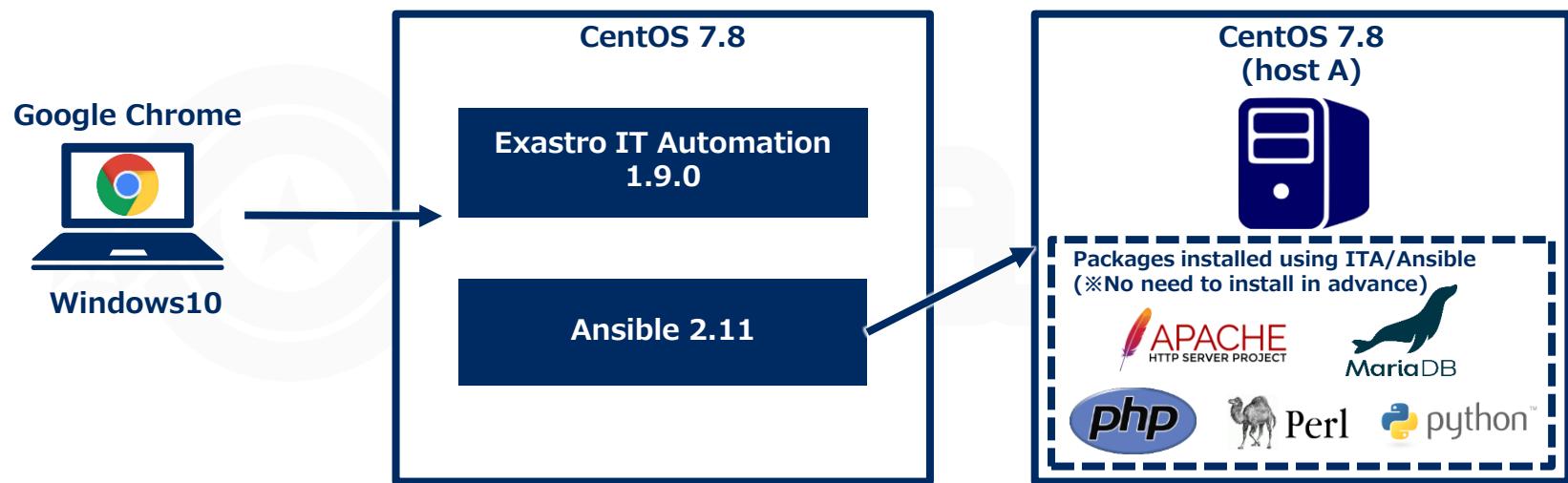
- Linking with Automation software (Ansible).
- Parameter management (Creation, Registration and history management of Menus)
- Linking Variables (Automatic registration of substitute values)



## 1.2 Scenario overview(1/3)

In this scenario, we will use Ansible Driver to manage the parameters for each server and automate the Yum Package installations, which is often used when constructing Linux servers.

### Environment

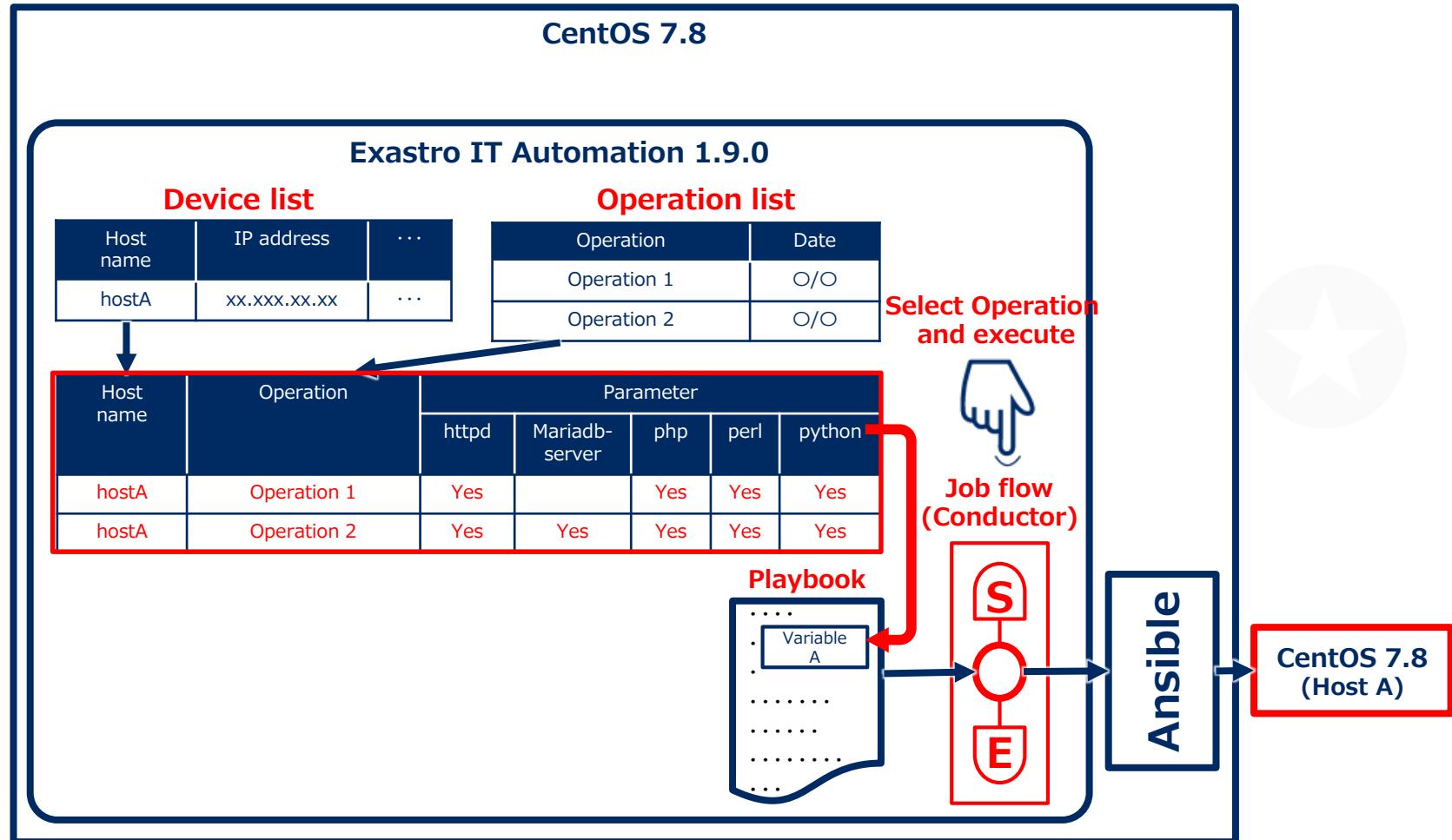


### Systems used

- Exastro IT Automation 1.9
- CentOS Linux 7.8(for ITA Server)
- CentOS Linux 7.8(for Target machine)
- Windows 10(Client)
- Google Chrome (Win10 side)

# 1.2 Scenario overview (2/3)

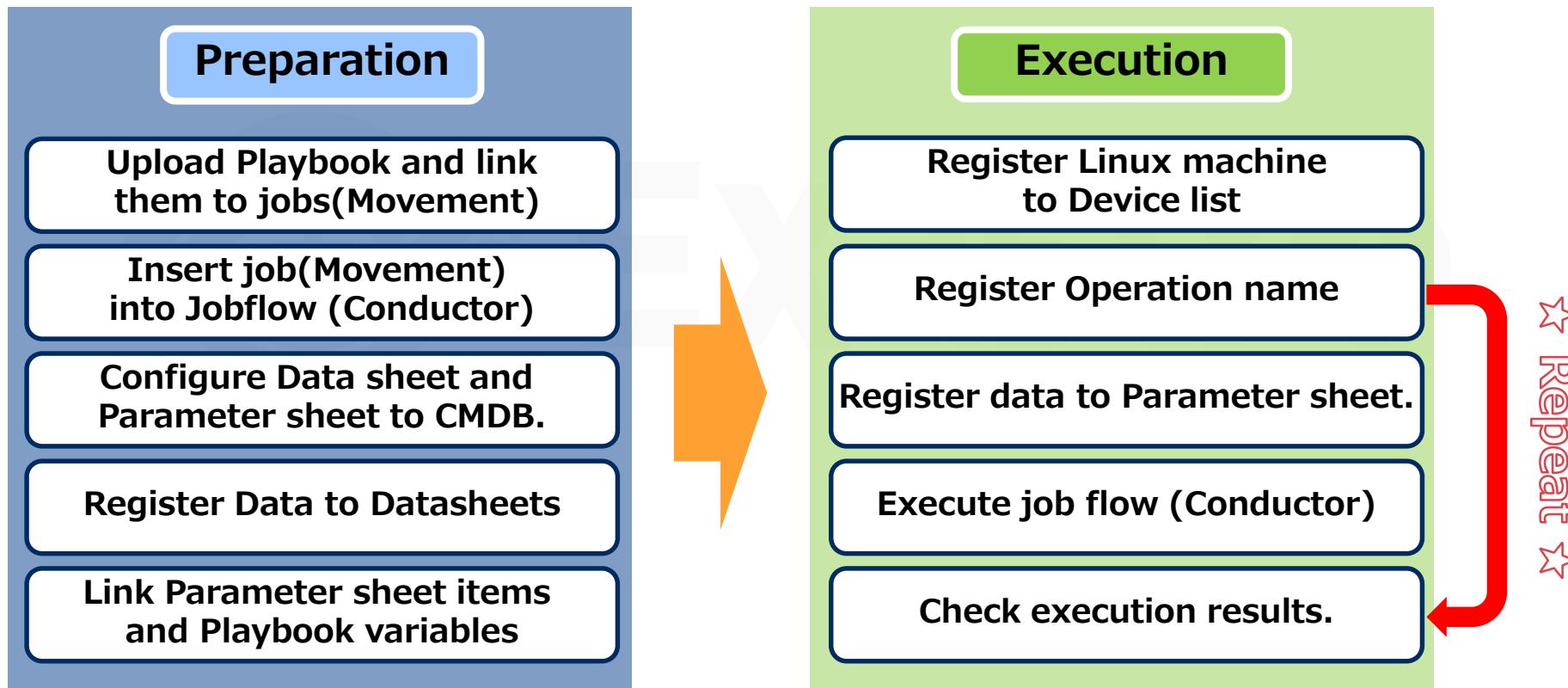
## Scenario execution image



## 1.2 Scenario overview(3/3)

### Post-installation Ansible Legacy execution scenario.

- The figure below illustrates the scenario as well as the Developer(Preparation)/Operator(Execution) operations.



# 1.3 Terminology

The following table explains the different terminology used in this document

Word	Description
Playbook	A file that describes routine tasks that can be executed with Ansible. All Playbook are written in YAML format.
Ansible-Legacy	A function that allows users to use Ansible from ITA. In the Legacy console, this is used when YAML files are used for the building code.
Operation name(Operation)	Operation unit in ITA. Users can set their execution dates in advance, manage the execution history and more.
Conductor	A sequence of work units. It can be executed after an operation name has been linked to it. Combine several parts called Nodes to create a job flow. It can then be used to execute configuration/construction operations on multiple machines.
Movement	Configuration/Construction units used with each of the devices construction tools.

For more information regarding Exastro ITA, please refer to the Document page on the community website.

## 2. Screen Description

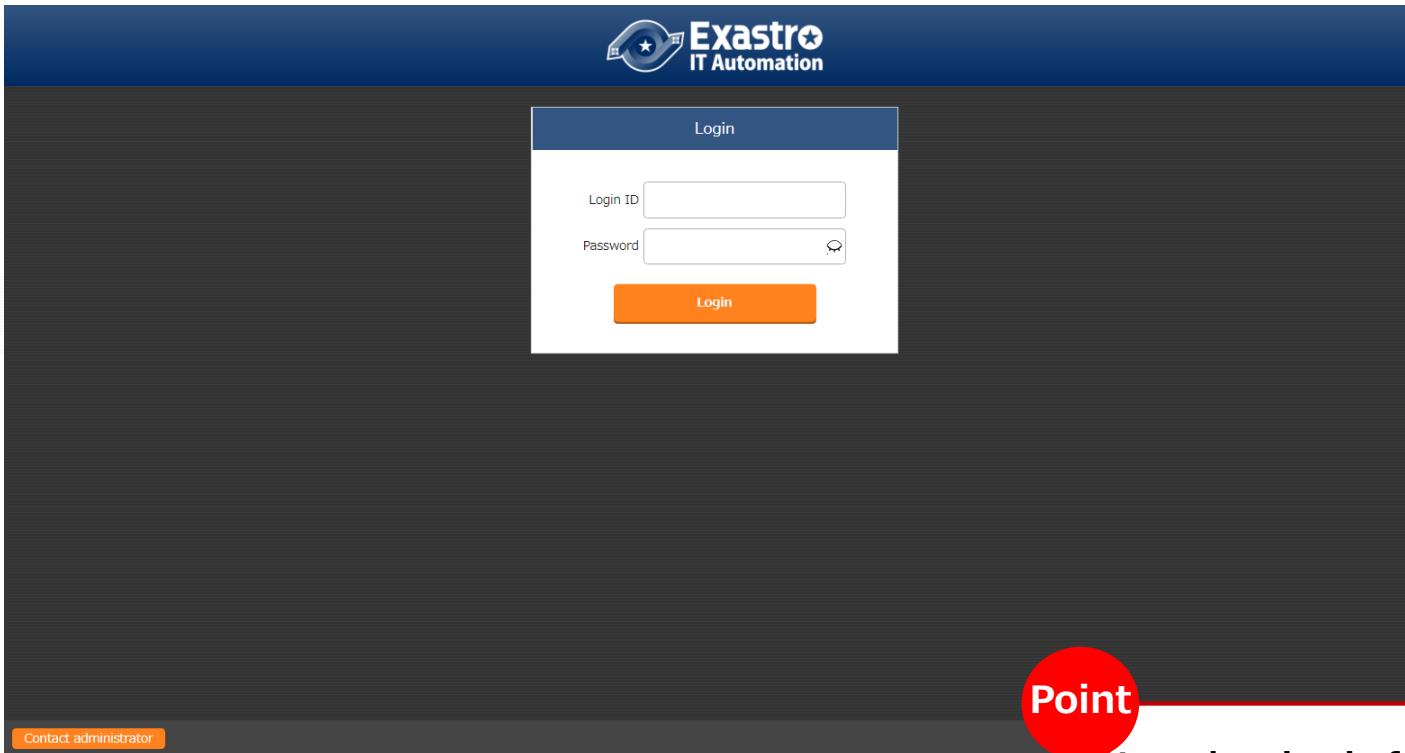


## 2.1 Web console screen description (Login)

### Web Console login screen

- Accessing ITA via URL after it has been installed will direct the user to the login screen

※For information regarding how to install ITA, please refer to the Online Install manual



Point

Users logging in for the first time  
will be asked to change  
their password

## 2.2 Screen Description (Main Menu)

### Screen description (Main Menu)

- The main functions of the Main Menu screen is as following

The screenshot shows the Exastro Management Console interface. On the left is a vertical sidebar labeled "Main menu" containing various system management options like "System settings" and "User list". Above the sidebar is the "Exastro IT Automation Management Console" logo. The top right corner shows user information ("User name [System Administrator]", "Login ID [administrator]"), and buttons for "Change password" and "Logout". The main content area is titled "DASHBOARD" and contains several sections: "Menu group" (with icons for Management, Basic Console, Export/Import, Symphony, Conductor, Create Menu, File control m..., File control ch..., Input, Substitution v..., Reference, Contrast, HostGroup ma..., Ansible-Legacy, Ansible-Pioneer, Ansible-Legacy..., Ansible Comm..., Cobbler, Terraform), "Movement" (a donut chart showing 13 Total, 76.9% in orange, with breakdown by Ansible Legacy, Pioneer, Legacy Role, and Terraform), "Work status" (a donut chart showing 0 Total, with breakdown by Status, CON, SYM, SUM), and "Work result" (a donut chart showing 1 Total, 100% in red, with breakdown by Result, CON, SYM, SUM). Below these is a "Work history" section showing a single entry. A red annotation points from the bottom left to the "Terraform" icon in the "Menu group" section. Another red annotation points from the bottom center to the "Point" (a red circle) on the bottom right. A large red callout box on the bottom right contains the text: "For more detailed information regarding the different functions, please refer to the manual."

Point

For more detailed information regarding the different functions, please refer to the manual.

Menu bar

Menu groups

## 2.3 Screen Description (Menu) (1/2)

### Screen Description (Menus)

- The name of the basic functions are as following.

The screenshot shows the Ansible-Legacy application interface. On the left is a vertical navigation menu with items like 'Main menu', 'Movement list', 'Playbook files', etc. The main area displays a submenu for 'Movement list' with sections for 'Description', 'Display filter', and 'List/Update'. The 'List/Update' section contains a table of movement records. A red box highlights the 'List/Update' section, and a red callout points to it with the text 'Point'.

**Submenu outline**

**Explanation**: Contains a brief description regarding the menu.

**Display Filter**: Lets the user search for registered information

**List/Update**: Displays registered information

**For more detailed information regarding the different functions, please refer to the manual.**

History	Duplicate	Update	Discard	Movement ID	Movement Name	Orchestrator	Access per role	Role to allow access	Last update date/time	Last updated by
History	Duplicate	Update	Discard	1	Gatherfacts	Ansible Legacy			2021/08/31 18:19:41	System Administrator
History	Duplicate	Update	Discard	2	getSSL	Ansible Legacy			2021/09/01 15:46:15	System Administrator
History	Duplicate	Update	Discard	5	move1	Ansible Legacy	Role A,Role B		2021/12/08 15:22:20	System Administrator
History	Duplicate	Update	Discard	6	move2	Ansible Legacy	Role A		2021/12/08 15:22:45	System Administrator
History	Duplicate	Update	Discard	7	move3	Ansible Legacy	Role B		2021/12/08 15:23:03	System Administrator
History	Duplicate	Update	Discard	8	move4	Ansible Legacy	Role C		2021/12/08 15:24:36	System Administrator
History	Duplicate	Update	Discard	9	movement1	Ansible Legacy	Role A		2021/12/08 15:29:35	Test1
History	Duplicate	Update	Discard	10	movement2	Ansible Legacy	Role B			
History	Duplicate	Update	Discard	11	Package install	Ansible Legacy				

## 2.3 Screen Description (Menu) (2/2)

### Screen description (Menu)

- The name of the basic functions are as following

The screenshot shows the Ansible-Legacy application interface. At the top, there is a header with the Exastro IT Automation logo, the title "Ansible-Legacy", and user information: "User name [System Administrator]", "Login ID [administrator]", "Change password", and "Logout". On the left, a vertical sidebar menu lists several options: "Main menu", "Movement list", "Playbook files", "Movement playbook link", "Substitution value auto-registration setting", "Target host", "Substitution value list", "Execution", "Check operation status", and "Execution list". The main content area displays a list of functions: "Description", "Display filter", "List/Update", "Register", "Download all and edit file uploads", "Change history", and "Trace history". The "Register" function is highlighted with a red box and a red arrow pointing to it from the text "Submenu". A callout box labeled "Submenu" points to the "Register" item. Another callout box labeled "Point" points to the "Change history" item. A large callout box at the bottom right contains the text: "For more detailed information regarding the different functions, please refer to the manual."

**Submenu**

**■ Submenu outline**

**Register** : Allows the user to register records from the browser

**Download all and edit file uploads** : IN/OUT processing with Excel

**Change history** : Change history of registered records

**Point**

For more detailed information regarding the different functions, please refer to the manual.

### 3. Preparation



# 3.1 Uploading Playbook and linking it to a job(Movement) (1/5)

## Playbook preparation

- First, we need to create the Playbook files that we are going to use.

Use your desired editor program to create the following YML file and save it to your local hard drive.

yum\_package\_install.yml

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

Uploading Playbook  
and linking it to a job(Movement)

Implementing job (Movement)  
into Jobflow (Conductor)

Configure CMDB Data sheet  
and Parameter sheet

Register Data to Datasheet

Link Parameter sheet item  
to Playbook variable.

Register target (Linux machine)  
to Device list.

Point

Make sure that the character code is "UTF-8" and the newline code is "LF".  
The file should be saved as an yml file.  
Please be check that the indents are correct.

```
** 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 Uploading Playbook and linking it to a job(Movement) (2/5)

## Register Movement to the Movement list.

- In the next step, we will register a Movement.

From the main menu , go to the Ansible-Legacy menu and then to the Movement list menu.

The screenshot shows the Exastro IT Automation interface. On the left, the 'Ansible-LegacyRole' section of the dashboard is visible. In the 'Main menu' sidebar, the 'Movement list' item is highlighted with a red box and the number '2'. In the main content area, the 'Ansible-Playbook' icon is highlighted with a red box and the number '1'. To the right, there is a vertical stack of six boxes, each containing a step or tip:

- Uploading Playbook and linking it to a job(Movement)**
- Implementing job (Movement) into Jobflow (Conductor)**
- Configure CMDB Data sheet and Parameter sheet**
- Register Data to Datasheet**
- Link Parameter sheet item to Playbook variable.**
- Register target (Linux machine) to Device list.**

# 3.1 Uploading Playbook and linking it to a job(Movement) (3/5)

## Register Movement to the Movement list.

- Click the “Start Registration” button.

Follow the table listed below and fill out the different items before pressing the “Register” button.

The screenshot shows the Exastro IT Automation interface for the Ansible-LegacyRole module. On the left, there's a sidebar with various menu items like 'Main menu', 'Movement list', 'Role package list', etc. The main area has a title 'Ansible-LegacyRole'. Below it, there are sections for 'Description' (with an 'Open' button), 'Display filter' (with a 'Close' button), and 'Movement list'. The 'Movement list' section contains a table with columns: Discard, Movement ID, Movement Name, Orchestrator, Delay timer, Last update date/time, and Last updated by. There are search fields for each column and buttons for 'Filter' and 'Clear filter'. Below the table, there are buttons for 'List/Update' (with an 'Open' button) and 'Register' (with a 'Close' button). A red circle with the number '3' points to the 'Start Registration' button in the 'Register' sub-section. Other buttons in this section include 'Download all and edit file uploads' (with an 'Open' button) and 'Trace history' (with an 'Open' button).

This block contains a vertical stack of six boxes, each with a title and a brief description. The titles correspond to the steps in the process:

- Uploading Playbook and linking it to a job(Movement)**
- Implementing job (Movement) into Jobflow (Conductor)**
- Configure CMDB Data sheet and Parameter sheet**
- Register Data to Datasheet**
- Link Parameter sheet item to Playbook variable.**
- Register target (Linux machine) to Device list.**

The screenshot shows the 'Register' sub-section of the 'Movement list' interface. It has a form with fields for 'Movement ID' (Auto-input), 'Movement Name\*' (set to 'PackageInstall'), 'Delay timer' (empty), 'Host specific' (dropdown set to 'IP'), and 'Format' (dropdown set to 'WinRM connection'). Below this is a section titled 'Dedicated information for ansible' with fields for 'Header' (empty), 'Last update date/time' (Auto-input), and 'Last updated by' (Auto-input). At the bottom of the form are buttons for 'Back' and 'Register'. A red circle with the number '4' points to the 'Movement Name' field. Another red circle with the number '5' points to the 'Register' button. A callout box at the bottom right shows the values: Movement name: PackageInstall, Host format: IP.

# 3.1 Uploading Playbook and linking it to a job(Movement) (4/5)

## Register Playbook to the Playbook file menu.

- Next, we will register the Playbook we created earlier to the Playbook files menu.

From the main menu , go to the Ansible-Legacy menu and then to the Playbook files menu. Fill out the items marked with red using the information from table listed below and press the "Register" button.

The screenshot shows the Ansible-Legacy interface with the following steps highlighted:

- Step 1: The "Playbook files" menu item is highlighted with a red box and a red number 1.
- Step 2: The "Playbook file name" and "Playbook file" fields are highlighted with a red box and a red number 2. The "Playbook file name" field contains "yum\_package\_install" and the "Playbook file" field contains "yum\_package\_install.yml".
- Step 3: The "Register" button at the bottom of the registration form is highlighted with a red box and a red number 3.

A callout bubble with the text "Point" and a red border points to the "Upload in advance" button in the registration form. Another callout bubble with the text "If you are uploading a Playbook, make sure to hit the \"Upload in advance\" button before pressing the \"Register\" button." is also present.

**Uploading Playbook and linking it to a job(Movement)**

**Implementing job (Movement) into Jobflow (Conductor)**

**Configure CMDB Data sheet and Parameter sheet**

**Register Data to Datasheet**

**Link Parameter sheet item to Playbook variable.**

**Register target (Linux machine) to Device list.**

# 3.1 Uploading Playbook and linking it to a job(Movement) (5/5)

## Register “Movement-Playbook link”

- Next, we will link the playbook to the earlier registered Movement

From the main menu, go to the Ansible-Legacy menu and then to the “Movement-Playbook link” menu. Fill out the items marked with red using the information from table listed below and press the “Register” button.

The screenshot shows the Ansible-Legacy interface with the following steps highlighted:

- Movement playbook link**: A red box highlights the "Movement playbook link" item in the left sidebar menu. A red circle with the number 1 points to this item.
- Execution list**: A red box highlights the "Execution list" section on the left side of the main content area. A red circle with the number 2 points to this section.
- Register**: A red box highlights the "Register" button at the bottom of the "Movement playbook link" form. A red circle with the number 3 points to this button.

**Table Data:**

Movement	Playbook file	Include order
Package Install	yum_package_install	1

**Form Fields (Movement playbook link screen):**

Associated item No.	Movement	Playbook files	Include order*	Access setting	Last update date/time	Last updated by
Auto-input	17:PackageInstall	yum_package_install	1	Setting	2023-09-11 14:53:27	Auto-input

**Callout Boxes:**

- Uploading Playbook and linking it to a job(Movement)**: A callout box pointing to the "Movement playbook link" menu item.
- Implementing job (Movement) into Jobflow (Conductor)**: A callout box pointing to the "Movement" column in the table.
- Configure CMDB Data sheet and Parameter sheet**: A callout box pointing to the "Playbook file" column in the table.
- Register Data to Datasheet**: A callout box pointing to the "Include order" column in the table.
- Link Parameter sheet item to Playbook variable.**: A callout box pointing to the "Access setting" field in the form.
- Point**: A callout box pointing to the "Include order" field in the form.
- If you want to registered a single movement to multiple Playbooks. For 1:1, please input 1.**: A callout box pointing to the "Include order" field in the form.

## 3.2 Implementing job (Movement)into Jobflow (Conductor)

### Create “Conductor”

- In the next step, we will implement the Movement into a conductor.  
From the Conductor menu group, access the Conductor Class edit screen.  
Follow the numbered steps below and press the “Register” button.

Uploading Playbook and linking it to a job(Movement)

Implementing job (Movement) into Jobflow (Conductor)

Configure CMDB Data sheet and Parameter sheet

Register Data to Datasheet

Link Parameter sheet item to Playbook variable.

Register target (Linux machine) to Device list.

Input field for Remarks and such.

Drag and Drop

Drag a line between "OUT" and "IN"

Registration

### 3.3 Configure CMDB Data sheet and Parameter sheet (1/3)

#### Create Parameter sheet

- In the next step, we will create a parameter sheet.

In the "Create menu" menu group, go to the "Define/Create Menu" menu.

Follow the steps below and fill out the items with the values written in the tables.

Click "Group"

2

3

1

Group name

Install Package

Menu creation information

Basic information :  
Id : Auto-input  
Menu name\* : Install Package list  
Creation target : Parameter Sheet(Host/Operati  
Display order\* : 1  
Create as hostgroup menu :  Yes  
Create as vertical menu  Yes  
Last modified : Auto-input  
Last updated by : Auto-input

Target menu group

Input : Input  
Substitution value\* : Substitution value  
Reference\* : Reference  
Target menu group  
Permission role  
Role :

List(Preview)

No Host name Operation name Reference date

1 Operation 2020/01/01 00:00

2 Operation 2020/01/01 00:00

3 Operation 2020/01/01 00:00

Create

Contact administrator

Uploading Playbook and linking it to a job(Movement)

Implementing job (Movement) into Jobflow (Conductor)

Configure CMDB Data sheet and Parameter sheet

Register Data to Datasheet

Link Parameter sheet item to Playbook variable.

Register target (Linux machine) to Device list.

### 3.3 Configure CMDB Data sheet and Parameter sheet (2/3)

#### Create Parameter sheet

- Add items and fill the items with the values written in the table below.

4 Add items so there are 5 in total

5

Item name	Input method	Selection item
httpd	Pulldown Selection	Create Menu: Selection 1:*- (blank)
mariadb-server	Pulldown Selection	Create Menu: Selection 1:*- (blank)
php	Pulldown Selection	Create Menu: Selection 1:*- (blank)
perl	Pulldown Selection	Create Menu: Selection 1:*- (blank)
python	Pulldown Selection	Create Menu: Selection 1:*- (blank)

User name [System Administrator]  
Logout

Uploading Playbook  
and linking it to a job(Movement)

Implementing job (Movement)  
into Jobflow (Conductor)

Configure CMDB Data sheet  
and Parameter sheet

Register Data to Datasheet

Link Parameter sheet item  
to Playbook variable.

Register target (Linux machine)  
to Device list.

Explanation

Remark

# 3.3 Configure CMDB Data sheet and Parameter sheet (3/3)

## Create Parameter sheet

- After following the steps below, click the “Create” button.

The screenshot shows the 'Create Menu' interface in Exastro IT Automation. On the left, there's a sidebar with options like 'Main menu', 'Create - Define menu', 'Menu definition information', 'Menu creation history', 'Menu conversion information', and 'Reference Item Info'. The main area is titled 'Create Menu' and shows a grid of items for different menu groups: 'Install Package', 'httpd', 'mariadb-server', 'php', 'perl', and 'python'. Each item has a 'Selection item' dropdown, an 'Input' field, a 'Reference item' dropdown, and checkboxes for 'Required reference item' and 'Unique constraint'. Below each item are sections for 'Explanation' and 'Remark'. A red callout with the number 6 points to the top row of the grid, with the text 'Drag and drop the items into the column group'. A red arrow points from the bottom of the grid up towards the header. A red callout with the number 7 points to the right side of the screen, with the text 'When finished, the items should look like this'. This callout contains a smaller screenshot of the same interface where all items have been moved into their respective columns. A red callout with the number 8 points to the bottom-left corner of the main interface, with the text 'Click the "Create" button'.

Uploading Playbook and linking it to a job(Movement)

Implementing job (Movement) into Jobflow (Conductor)

Configure CMDB Data sheet and Parameter sheet

Register Data to Datasheet

Link Parameter sheet item to Playbook variable.

Register target (Linux machine) to Device list.

## 3.4 Link Parameter sheet item to Playbook variable (1/3)

### Create "Substitution value auto-registration settings".

- Lastly, we will automatically register substitute values.

In the "Ansible-Legacy" Menu group, go to the "Substitution value auto-registration setting" menu.

Follow the steps below and fill the items with the values written in the table below.

The screenshot shows the Exastro IT Automation interface with the 'Ansible-Legacy' menu group selected. A red box highlights the 'Substitution value auto-registration setting' menu item in the left sidebar. The 'Register' sub-menu is currently active, indicated by a red border around its tab. The main content area displays a table titled 'Parameter sheet (From)' with one row of data. The table has columns for Item No., Menu group:Menu\*, Item\*, Last update date/time, and Last updated by. The data row shows 'Auto-input' for Item No., '2100011611:Substitution value:58:Install Package list' for Menu group:Menu\*, 'Parameter/Install Package/httpd' for Item\*, and 'Auto-input' for both Last update date/time and Last updated by.

- Uploading Playbook and linking it to a job(Movement)
- Implementing job (Movement) into Jobflow (Conductor)
- Configure CMDB Data sheet and Parameter sheet
- Register Data to Datasheet
- Link Parameter sheet item to Playbook variable.
- Register target (Linux machine) to Device list.

## 3.4 Link Parameter sheet item to Playbook variable (2/3)

### Create "Substitution value auto-registration settings".

- Follow the table below and press the "Register" button.

The screenshot shows the Exastro IT Automation web interface. In the top left, there's a logo for 'Exastro IT Automation'. The top right shows the user information: 'User name [System Administrator]', 'Login ID [administrator]', 'Change password', and 'Logout'. On the left, there's a sidebar with various menu items like 'Main menu', 'Movement list', 'Playbook files', etc. The main area is titled 'Ansible-Legacy'. It contains a table with columns: 'Item No.', 'Registration method\*', 'Movement\*', 'IaC variable(To)', 'Key varia', 'Last update date/time', and 'Last updated by'. The table has one row with values: 'Auto-input', 'Key type', '17:PackageInstall', '6:VAR\_packages', 'Auto-input', and 'Auto-input'. Below the table, a note says '※\* is a required item.' At the bottom of the table area, there are two buttons: 'Back' and 'Register'. A red circle with the number '2' is drawn around the 'Register' button. At the very bottom of the page, there's a link 'Download all and edit file uploads'.

This block contains a vertical stack of six callout boxes. From top to bottom:

- Uploading Playbook and linking it to a job(Movement)
- Implementing job (Movement) into Jobflow (Conductor)
- Configure CMDB Data sheet and Parameter sheet
- Register Data to Datasheet
- Link Parameter sheet item to Playbook variable.** (This box is highlighted with a red border.)
- Register target (Linux machine) to Device list.

Point

The following table describes the 3 different variable link registration methods.

Registration method	Use	Description
Value type		Basic registration type. Links the value written in the table to the variable.
Key type	●	Links the table item (column name) to the variable. If the item's setting value is blank, it will not be linked.
Key-Value type		Links both the item name (Key) and the setting value (Value) to the variable.

In this scenario, we want to assign the table items (column name) to the Playbook as a specific value, so we will choose the "Key" registration method.

For more information, please see the [Exastro System Operation and Construction Efficiency Guide](#)

### 3.4 Link Parameter sheet item to Playbook variable (3/3)

#### Create "Substitution value auto-registration settings".

Use the Display filter to check that you have registered 5 items.

This will end the preparation operations.

Exastro IT Automation Ansible-Legacy

User name [System Administrator] Login ID [administrator] Change password Logout

☰ Menu

Main menu Movement list Playbook files Movement playbook Substitution value auto-registration setting Target host Substitution value Execution Check operation status Execution list

Description

Display filter

Parameter sheet(From)

Discard Item No. Menu group ID Name ID Last update date/time Last updated by

Exclude discarded records ▾ Search from pulldown Filter ▾ Search from pulldown

Filter Clear filter Auto-filter

Check that 5 items are registered

Parameter sheet(From)

History	Duplicate	Update	Discard	Item No. ▾	Menu group	ID ▾	Name ▾	Menu	ID ▾	Name ▾	Last update date/time ▾	Last updated by ▾
History	Duplicate	Update	Discard	2	2100011611 Substitution value	19	Install Package list	Parameter/Install Package/http	2021/12/09 13:27:33	System Administrato		
History	Duplicate	Update	Discard	3	2100011611 Substitution value	19	Install Package list	Parameter/Install Package/mari	2021/12/09 13:27:54	System Administrato		
History	Duplicate	Update	Discard	4	2100011611 Substitution value	19	Install Package list	Parameter/Install Package/php	2021/12/09 13:28:11	System Administrato		
History	Duplicate	Update	Discard	5	2100011611 Substitution value	19	Install Package list	Parameter/Install Package/perl	2021/12/09 13:28:25	System Administrato		
History	Duplicate	Update	Discard	6	2100011611 Substitution value	19	Install Package list	Parameter/Install Package/pyth	2021/12/09 13:28:40	System Administrato		

Filter result count: 5

Output Excel

Uploading Playbook and linking it to a job(Movement)

Implementing job (Movement) into Jobflow (Conductor)

Configure CMDB Data sheet and Parameter sheet

Register Data to Datasheet

Link Parameter sheet item to Playbook variable.

Register target (Linux machine) to Device list.

## 4. Execution (First time)

# 4.1 Register target (Linux machine) to Device list (1/3)

## Register target host to "Device list".

- First, we will have to register the target host to which we will install packages to. From the "Basic Console" Menu group, go to the "Device list" menu. Fill in the information written in the table below.

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  
フィルタ ▽開く  
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

※\*は必須項目です。

HW Device type Host name IP Address

HW Device type	Host name	IP Address
SV	(Free host name)	(Free IP Address)

管理者に連絡

Uploading Playbook and linking it to a job(Movement)

Implementing job (Movement) into Jobflow (Conductor)

Configure CMDB Data sheet and Parameter sheet

Register Data to Datasheet

Link Parameter sheet item to Playbook variable.

Register target (Linux machine) to Device list.

# 4.1 Register target (Linux machine) to Device list (2/3)

Register target host to "Device list".

- Use the scrollbar to scroll to the left and fill in the items listed below.

The screenshot shows the Exastro Basic Console interface. On the left, there's a sidebar with a 'Menu' section containing 'Main menu', 'Device list' (which is selected), 'Operation list', 'Movement list', 'ER Diagram', 'ER Diagram Menu List', and 'ER Diagram Item List'. Below the sidebar is a red circle with the number '2' pointing to the 'Device list' item in the menu. The main content area is titled 'Basic Console' and shows a 'User name [System Administrator] Login ID [administrator]' header with 'Change password' and 'Logout' buttons. The 'Device list' section has a 'Description' dropdown, a 'Display filter' dropdown, a 'List/Update' button, and a 'Register' button. The 'Register' button is highlighted with a red box. Below these are several input fields: 'Managed system item number' (Auto-input), 'HW device type' (SV dropdown), 'Host name\*' (input field), 'IP address\*' (input field), 'EtherWakeOn' (checkbox), 'MAC address' (input field), 'Netw' (input field), 'Last update date/time' (input field), and 'Last updated by' (input field). At the bottom of the registration form is a large red box containing three input fields: 'Login user ID' (with placeholder '(Login user ID)'), 'Login password management' (with a black dot placeholder), and 'Login Password' (with placeholder '(Login password)').

Uploading Playbook  
and linking it to a job(Movement)

Implementing job (Movement)  
into Jobflow (Conductor)

Configure CMDB Data sheet  
and Parameter sheet

Register Data to Datasheet

Link Parameter sheet item  
to Playbook variable.

Register target (Linux machine)  
to Device list.

# 4.1 Register target (Linux machine) to Device list (3/3)

## Register target host to "Device list".

- Use the table below to fill in the last item and press the "Register" button.

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

Dedicated information for Legacy/Role Authentication method  
Password authentication

△閉じる

管理システム項目番	Ansible利用情報		Legacy/Role利用情報		最終更新日時	最終更新者
	認証方式	WinRM接続情報	ポート番号	サーバー証明書		
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		自動入力	自動入力

※\*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

Uploading Playbook and linking it to a job(Movement)

Implementing job (Movement) into Jobflow (Conductor)

Configure CMDB Data sheet and Parameter sheet

Register Data to Datasheet

Link Parameter sheet item to Playbook variable.

**Register target (Linux machine) to Device list.**

Point

In order to run Ansible-Legacy, the following 6 items must be filled.  
**"Host name", "IP Address", "Login User ID", "Login Password Management", "Login password", "Authentication method※"**

※In this document, "Authentication method" is written as "Password authentication".

## 4.2 Register Operation name (Operation)

### Register "Operation name" to "Operation list"

In this step, we will register an Operation name. From the "Basic Console" menu group, go to the "Operation list" menu.

Input the following information and press the "Register" button.

The screenshot shows the Exastro Basic Console interface. On the left, a sidebar lists various menu items: Main menu, Device list, **Operation list**, Movement list, ER Diagram, ER Diagram Menu List, and ER Diagram Item List. The 'Operation list' item is highlighted with a red box and a red circle containing the number 1. The main content area has two tabs: 'Description' (selected) and 'Display filter'. Below these are search filters for 'Exclude discarded records', 'No.', 'Operation ID', 'Operation', 'Last update date/time', and 'Last updated by'. There are 'Filter' and 'Clear filter' buttons. A 'List/Update' tab is also visible. On the right side of the screen, there is a vertical stack of four rounded rectangular boxes with a dark blue background and white text. The top box is red and says 'Register Operation name (Operation)'. The second box is light blue and says 'Register data to Parameter sheet'. The third box is light blue and says 'Execute Jobflow (Conductor)'. The bottom box is light blue and says 'Check Execution results'. In the bottom right corner of the main content area, there is a callout box with a red border containing two columns: 'Operation name' (containing 'Operation 1') and 'Scheduled date and time' (containing '(Free date/time)').

User name [System Administrator]  
Login ID [administrator]  
Change password Logout

Description ▽Open △Close

Display filter ▽Open △Close

Exastro IT Automation Basic Console

Main menu Device list **Operation list** 1 Movement list ER Diagram ER Diagram Menu List ER Diagram Item List

Filter Clear filter Auto-filter

No. Operation ID Operation name\* Scheduled date for execution\* Access permission Last update date/time Last updated by

Auto-input Auto-input Operation 1 2021/07/02 14:08

\* is a required item.

Back Register

Register Operation name (Operation)

Register data to Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

Operation name: Operation 1  
Scheduled date and time: (Free date/time)

## 4.3 Register data to Parameter sheet (1/2)

### Register data to Install Package list.

In the next step, we are going to input data to the Install package list (Parameter sheet) that we prepared earlier.

Go to the "Input" menu and then the "Install package list" menu.

Input the following information and press the "Register" button.

The screenshot shows the Exastro IT Automation Input interface. On the left, there's a sidebar with a menu. The 'Install Package list' option is highlighted with a red box and a red number 1. In the main area, there's a table for registering data. A red box highlights this table. A red number 2 points to the 'Host name' field, which contains '(Previously registered host)'. A red number 3 points to the 'Register' button at the bottom of the table. To the right of the table, there's a vertical stack of four boxes:

- Register Operation name (Operation)
- Register data to Parameter sheet (highlighted in red)
- Execute Jobflow (Conductor)
- Check Execution results

**Table Data:**

No	Host name*	Operation	Parameter	Last update date/time	Last updated by
1	Auto-Input aaa-test-target02	2021/07/02 14:06:21:Operation 1	httpd mariadb-server php perl python	Yes Yes Yes Yes Yes	Auto-Input Auto-Input

\* Is a required item.

**Host name**  
(Previously registered host)

**Operation**  
(previously specified date )\_1:Operation 1

**httpd**  
True

**mariadb-server**

**php**  
\*

**perl**  
\*

**python**  
\*

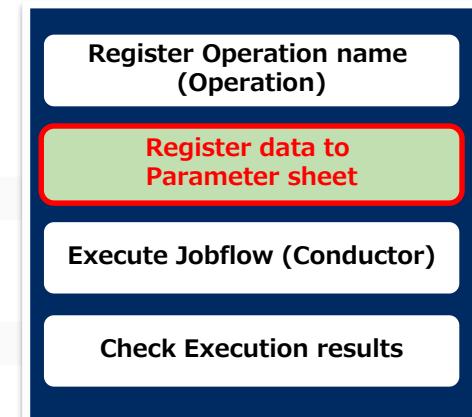
## 4.3 Register data to Parameter sheet (2/2)

### Register data to Install Package list.

- Similarly to when we configured substitution value settings, open the display filter and press the "filter" button to check if the registration was done correctly.

The screenshot shows the Exastro Input interface with the 'Install Package list' selected in the left sidebar. A red box highlights the 'Display filter' button in the top right corner of the main content area. Three numbered callouts point to specific elements: 1 points to the 'Display filter' button; 2 points to the 'Filter' button within the filter panel; and 3 points to the result table below, which contains one row of data.

History	Duplicate	Update	Discard	No	Host name	ID	Operation name	Reference date	Scheduled date for execution	Last update date/time	Last updated by
History	Duplicate	Update	Discard	1	targethost	7	Operation 1	2021/12/30 13:32	2021/12/30 13:32	2021/12/09 13:33:44	System Administrator



# 4.4 Execute Jobflow (Conductor) (1/3)

## Run Conductor

- We will now start the Conductor.

From the "Conductor" Menu group, go to the "Conductor Execution" Menu.

Next, select "Conductor" and "Operation" and press "Execute".

The screenshot shows the Exastro IT Automation Conductor interface. On the left, a sidebar lists menu items: Main menu, Conductor interface information, Conductor class list, Conductor class edit, **Conductor execution** (highlighted with a red box and number 1), Conductor confirmation, Conductor list, Conductor Regularly execution, Contact administrator, and a system status bar.

The main content area has several tabs: Description, Scheduling, Conductor [filter], and **Conductor [List]** (highlighted with a red box and number 2). The **Conductor [List]** tab displays a table with one row selected, showing details: Select, Conductor class ID, Conductor name, Explanation, Access permission, Remarks, Last update date/time, and Last updated by. The table header includes columns for Select, Conductor class ID, Conductor name, Explanation, Access permission, Remarks, Last update date/time, and Last updated by.

Below the table, a message says "Filter result count: 1".

Further down, there are tabs for Operation [Filter] and **Operation [List]** (highlighted with a red box and number 3). The **Operation [List]** tab displays a table with one row selected, showing details: Select, No., Operation ID, Operation name, Scheduled date for execution, Last execution date, and Role. The table header includes columns for Select, No., Operation ID, Operation name, Scheduled date for execution, Last execution date, and Role.

Below the table, a message says "Filter result count: 1".

To the right of the interface, a large green box contains four steps:

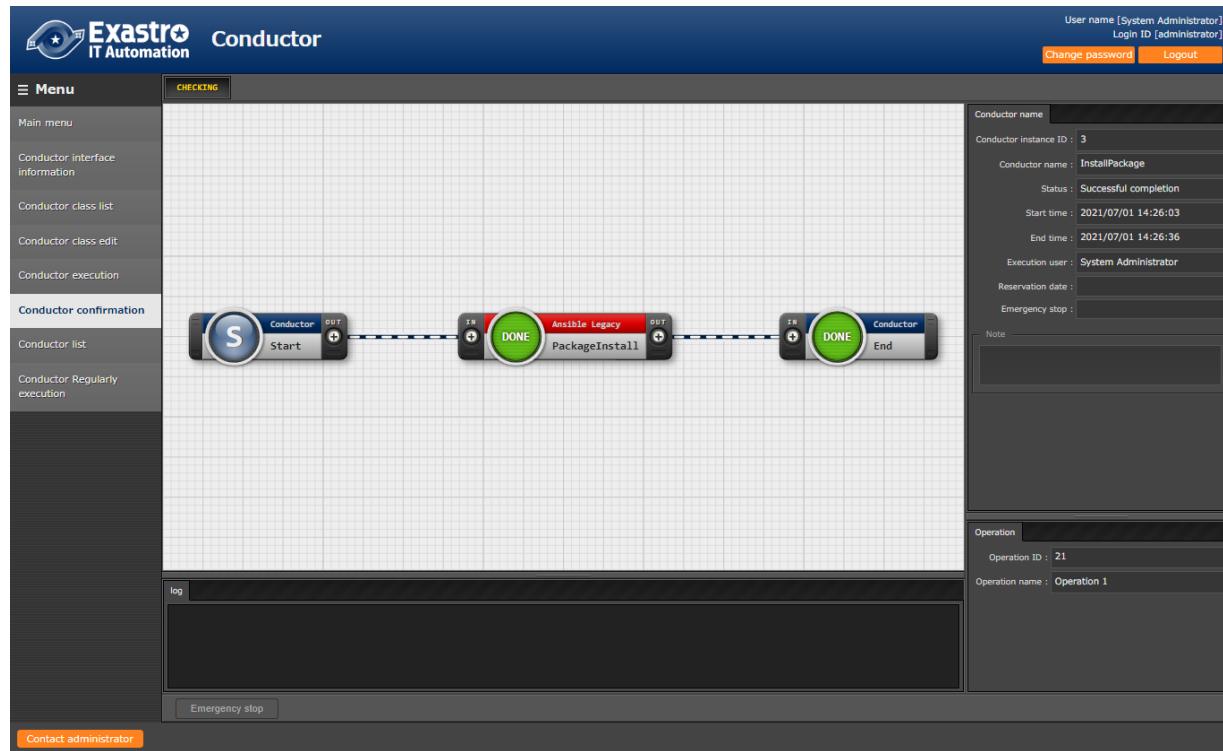
- Register Operation name (Operation)
- Register data to Parameter sheet
- Execute Jobflow (Conductor)** (highlighted with a red box and number 4)
- Check Execution results

At the bottom right, a small window shows the jobflow diagram: S (Conductor start) --> 17 (PackageInstall) --> E (Conductor End).

# 4.4 Execute Jobflow (Conductor) (2/3)

## Execution results

- Executing the Conductor will move the user to the "Conductor confirmation" screen where execution status and execution logs are displayed.



Register Operation name  
(Operation)

Register data to  
Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

Point

The execution status  
and execution log  
can be checked  
in real-time.

# 4.4 Execute Jobflow (Conductor) (3/3)

## Execution results

- Select a job (Movement) and press either the "Done" icon or the Operation status on the right to see more details.

The screenshot shows the Exastro IT Automation Conductor interface. On the left, there's a navigation menu with items like Main menu, Conductor interface information, Conductor class list, Conductor class edit, Conductor execution, Conductor confirmation, Conductor list, Conductor Regularly execution, and Contact administrator. A red callout bubble with the number '1' and the text 'Select' points to a green 'DONE' button on a movement node in a job flow diagram. To the right of the flow diagram is a detailed view of a movement node. The node details include:

Node	node
Node type	movement
Node instance ID	11
Node name	node-3
Status	Normal end
Start time	2021/07/01 14:26:03
End time	2021/07/01 14:26:33
Operation status	<a href="#">Check execution status</a>

A red arrow points from the 'Check execution status' link to the 'Execute Jobflow (Conductor)' section on the right. This section contains a table with execution details:

Description	Item
Target Operation	
Execution No.	3
Execution type	Normal
Status	Completed
execution engine	Ansible Engine
Caller symphony	
Caller conductor	InstallPackage
Execution user	System Administrator
Movement	
ID	17
Name	PackageInstall
Delay timer (minutes)	
Dedicated information for ansible	Host specific format
No.	IP
Operation	
ID	21
Name	Operation 1
Host management	
Substitution value	
Input data	Populated data
Output data	Result data Scheduled date/time
Operation status	Start date/time: 2021/07/01 14:26:08 End date/time: 2021/07/01 14:26:30

Other sections on the right include 'Register Operation name (Operation)', 'Register data to Parameter sheet', and 'Check Execution results'.

# 4.5 Check Execution results (1/3)

## Execution results

- In the detailed results screen, we can use the progress status (Execution log) to check the Ansible execution log.

The screenshot shows the Exastro UI interface. On the left is a vertical sidebar with a 'Menu' section containing links: Main menu, Movement list, Playback files, Movement playbook link, Substitution value auto-registration setting, Target host, Substitution value list, Execution, Check operation status, and Execution list. Below the sidebar is a 'Contact administrator' button and a three-dot menu icon.

The main area displays several windows:

- A top-level window titled 'Input data' with tabs for 'Input data' (selected), 'Output data', 'Scheduled date/time', 'Operation status', 'Input data URL' (ResultData @0000000003.zip), 'Result data', 'Start date/time' (2021/07/01 14:26:08), and 'End date/time' (2021/07/01 14:26:30).
- A large central window titled 'Progress status(Execution log)' with a red border. It contains a 'Filter:' input field and a checkbox for 'Display only corresponding lines'. The log output shows:

```
Verifying : httpd-2.4.6-97.el7.centos.x86_64          1/3
Verifying : mailcap-2.1.41-2.el7.noarch                2/3
Verifying : httpd-tools-2.4.6-97.el7.centos.x86_64      3/3

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!
"
]
META: ran handlers
META: ran handlers

PLAY RECAP ****
ita-test-target02 : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```
- A bottom-level window titled 'Progress status(Error log)' with a 'Filter:' input field and a checkbox for 'Display only corresponding lines'. It is currently empty.
- A context menu on the right side of the central window, enclosed in a blue rounded rectangle, with the following options:
  - Register Operation name (Operation)
  - Register data to Parameter sheet
  - Execute Jobflow (Conductor)
  - Check Execution results** (highlighted with a red rounded rectangle)

# 4.5 Check Execution results (2/3)

## Execution results

- Use the Execution log to see if httpd, php, perl and python are installed.

Progress log(Execution log)

```
~~~~~  
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!  
    "}  
~~~~~  
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!  
    "}  
~~~~~  
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!  
    "}  
~~~~~  
Updated:  
    python.x86_64 0:2.7.5-90.el7  
Dependency Updated:  
    python-libs.x86_64 0:2.7.5-90.el7  
Complete!  
    "}
```

Register Operation name  
(Operation)

Register data to  
Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

## 4.5 Check Execution results (3/3)

### Check the Target machine.

- Check that the packages are installed on the Target machine.

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
```

Register Operation name  
(Operation)

Register data to  
Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

## 5. Execution (Second time)

# 5.1 Register Operation name (Operation)

## Register Operation name to the "Operation list".

- This step will be the same as the first time we registered an operation name.  
From the "Basic Console" menu group, go to the "Operation list" menu.  
Input the information below and press the "Register" button.

The screenshot shows the Exastro Basic Console interface. On the left, a sidebar menu includes 'Operation list' (highlighted with a red box and circled with a red number 1). The main area has three tabs: 'Description' (selected), 'Display filter', and 'List/Update'. The 'List/Update' tab displays a message: '\*List is displayed here.' Below it is a 'Register' tab. The 'Register' tab contains a form with fields: 'Operation name\*' (highlighted with a red box and circled with a red number 2), 'Scheduled date for execution' (containing '2021/07/03 14:37'), and a 'Register' button (highlighted with a red box and circled with a red number 3). A callout box highlights the 'Operation name' field with the value 'Operation 2' and the 'Reservation date/time' field with the value '(Free date/time)'. The top right corner of the screen shows a navigation bar with 'User name [System Administrator]', 'Login ID [administrator]', 'Change password', and 'Logout'.

Register Operation name (Operation)

Register data to Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

User name [System Administrator]  
Login ID [administrator]  
Change password Logout

Exastro IT Automation Basic Console

Description

Display filter

No. Operation ID Operation name Scheduled date for execution Action Setting Last update date/time Last updated by

Auto-input Auto-input Operation 2 2021/07/03 14:37

Back Register

Contact administrator

Operation name Reservation date/time

Operation 2 (Free date/time)

# 5.2 Register data to Parameter sheet

## Register data to "Install Package list"

- From the "Input" menu group, go to the "Install package list" menu

Input the information below and press the "Register" button. Please note that the packages we are installing are different from the first time.

The screenshot shows the Exastro IT Automation interface. On the left, there's a sidebar with a 'Main menu' section containing 'Install Package list' (highlighted with a red box and circled with a red number 1). Below it are 'Flag Definition' and 'Register'. The main area has a title 'Description' with three dropdown menus: 'Host name' (set to 'ita-test-target02'), 'Operation' (set to '2021/07/03 14:37:22:Operation 2'), and 'Parameter' (checkboxes for httpd, mariadb-server, php, perl, python, all set to 'Yes'). At the bottom of this panel are 'Back' and 'Register' buttons (the latter is highlighted with a red box and circled with a red number 3). Below this panel is a link 'Download all and edit file uploads' and a link 'Trace history'. To the right of the main area is a vertical sidebar with four buttons: 'Register Operation name (Operation)' (highlighted with a red box), 'Register data to Parameter sheet' (highlighted with a red box and circled with a red number 2), 'Execute Jobflow (Conductor)', and 'Check Execution results'. A large red box highlights the entire registration panel, and another red box highlights the 'Register' button.

Host name	Operation	httpd	mariadb-server	php	perl	python
(Previously registered host)	(Previously specified date )_1:Operation 1	Yes	Yes	Yes	Yes	Yes

# 5.3 Execute Jobflow (Conductor) (1/3)

## Run Conductor

- We will now run the Conductor a second time.

From the “Conductor” Menu group, go to the “Conductor execution” menu.

Select the Conductor and Operation you want to run and press “Execute”.

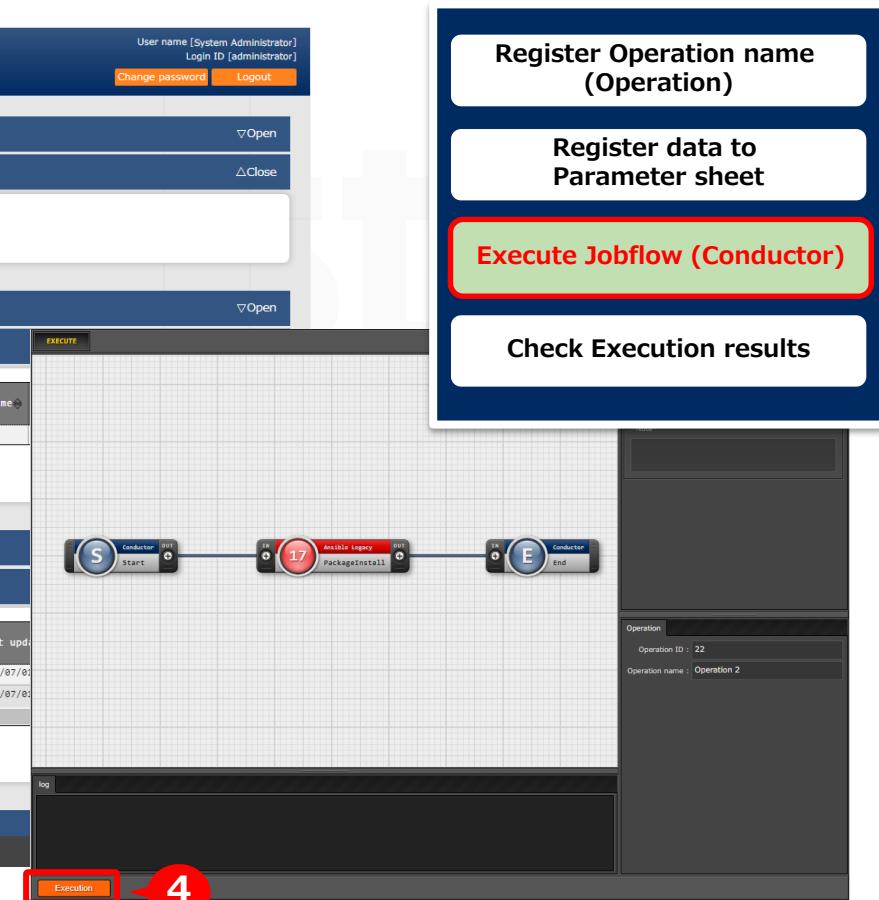
1. Click on the 'Conductor execution' menu item.

2. Select the 'InstallPackage' conductor from the 'Conductor [List]' table.

3. Select 'Operation 2' from the 'Operation [List]' table.

Conductor class ID	Conductor name	Explanation	Access permission	Remarks	Last update date/time
1	InstallPackage				2021/07/01 11:34:52

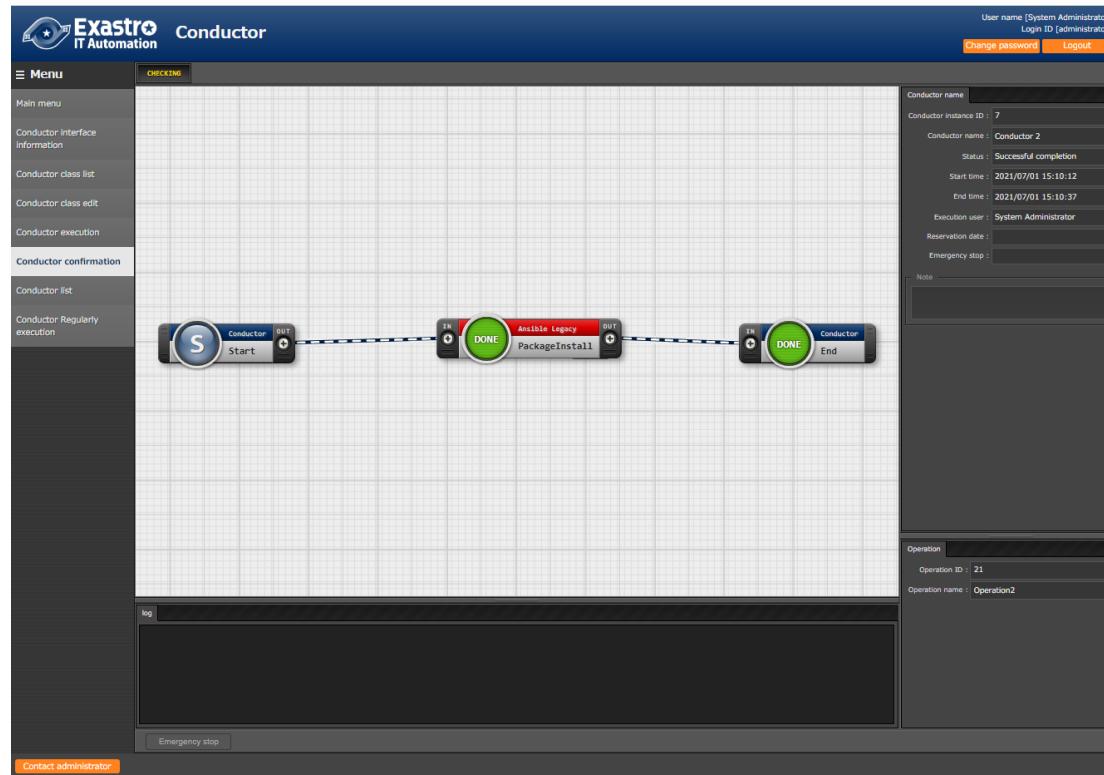
No.	Operation ID	Operation name	Scheduled date for execution	Last execution date	Access	Role to allow access	Last update date/time
1	21	Operation 1	2021/07/02 14:06	2021/07/01 14:26			2021/07/01 14:26
2	22	Operation 2	2021/07/03 14:37				2021/07/03 14:37



# 5.3 Execute Jobflow (Conductor) (2/3)

## Execution results

- Executing the Conductor will move the user to the "Conductor confirmation" screen where execution status and execution logs are displayed.



Register Operation name  
(Operation)

Register data to  
Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

Point

The Execution status and  
the Execution log can  
be checked in real-time.

# 5.3 Execute Jobflow (Conductor) (3/3)

## Execution results

- Select a job (Movement) and press either the "Done" icon or the Operation status on the right to see more details.

The screenshot shows the Exastro Conductor interface. On the left, there's a navigation menu with items like Main menu, Conductor Interface information, Conductor class list, Conductor class edit, Conductor execution, Conductor confirmation, Conductor list, Conductor Regularly execution, and Contact administrator. The main area displays a job flow diagram with nodes: 'S' (Start), 'Ansible Legacy', and 'PackageInstall'. A red box highlights the 'Ansible Legacy' node, which has a green 'DONE' button. A red callout bubble with the number '1' and the text 'Select' points to this button. To the right of the flow diagram, a detailed view of a node is shown in a modal window. The node details include: Node type: movement, Node instance ID: 26, Node name: node-3, Status: Normal end, Start time: 2021/07/01 15:11:54, and Operation status: Check execution status. Below this, a table shows the 'Get Operation' details:

Item	Value
Execution No.	9
Execution type	Normal
Status	Completed
Execution engine	Ansible Engine
Caller symphony	
Caller conductor	InstallPackage
Execution user	System Administrator
Movement	
ID	17
Name	PackageInstall
Delay timer (minutes)	
Dedicated information for ansible	
Host specific format	IP
Operation	
No.	21
Name	Operation2
ID	21
Host management	
Substitution value	
Input data	Populated data
Output data	Result data
Scheduled date/time	2021/07/01 15:11:57
Operation status	Start date/time End date/time
	2021/07/01 15:12:03

On the right side of the interface, there are four boxes with rounded corners:

- Register Operation name (Operation)
- Register data to Parameter sheet
- Execute Jobflow (Conductor) (highlighted in green)
- Check Execution results

# 5.4 Check Execution results (1/2)

## Execution results

- In the detailed results screen, we can use the progress status (Execution log) to check the Ansible execution log.

The screenshot shows the Exastro interface with a sidebar menu and several windows displaying execution details and logs.

**Left Sidebar (Menu):**

- Main menu
- Movement list
- Playbook files
- Movement playbook link
- Substitution value auto-registration setting
- Target host
- Substitution value list
- Execution
- Check operation status
- Execution list

**Central Window (Execution Log):**

Movement	Delay timer (minutes)	Host specific format	IP
	Dedicated information for ansible	WinRM connection	
Operation	No.		21
	Name		Operation2
	ID		21
Host management		confirmation	
Substitution value		confirmation	
Input data	Populated data		InputData_0000000009.zip
Output data	Result data		ResultData_0000000009.zip
Operation status	Scheduled date/time		
	Start date/time		2021/07/01 15:11:57
	End date/time		2021/07/01 15:12:03

**Bottom Window (Error Log):**

```
Progress status(Execution log)
Filter : □ Display only corresponding lines
php", "perl", "python"
],
"msg": "", "rc": 0,
"results": [
"All packages providing httpd are up to date",
"All packages providing php are up to date",
"All packages providing perl are up to date",
"All packages providing python are up to date",
""
]
}
META: ran handlers
META: ran handlers

PLAY RECAP ****
ita-test-target02 : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

**Right Panel (Buttons):**

- Register Operation name (Operation)
- Register data to Parameter sheet
- Execute Jobflow (Conductor)
- Check Execution results

**Bottom Left Buttons:**

- Contact administrator
- ☰

# 5.4 Check Execution results (2/2)

## Execution results

- Check that the new installed Maria DB's dependency with other packages are correct and that the other 4 packages (httpd,php,perl,python) has been updated.

Progress log (Execution log)

```
~~~~~  
"results": ["All packages providing httpd are up to date",  
  ""]}  
~~~~~
```

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!

"]}  
~~~~~

```
"results": ["All packages providing php are up to date",  
  ""]}  
~~~~~
```

```
"results": ["All packages providing perl are up to date",  
  ""]}  
~~~~~
```

```
"results": ["All packages providing python are up to date",  
  ""]}
```

Register Operation name  
(Operation)

Register data to  
Parameter sheet

Execute Jobflow (Conductor)

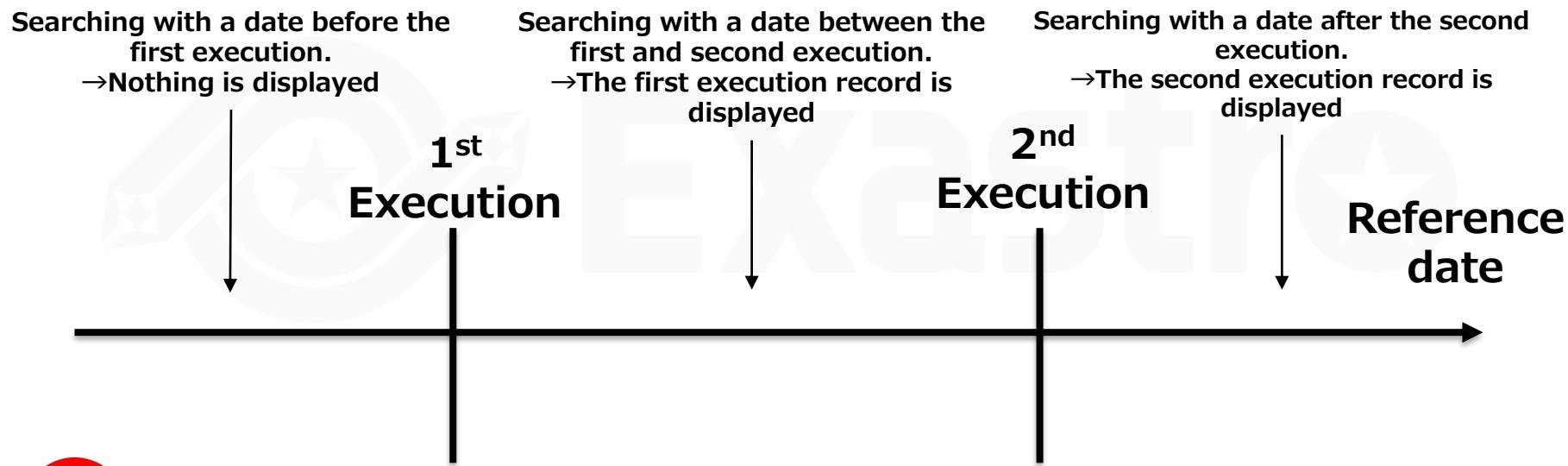
Check Execution results

## 6. Checking the CMDB Parameter history

# 6.1 Executing operations and History Management

## Scenario and History Management

- ITA Manages parameter history and keeps track on who last used it, when it happened and why in the CMDB.
- ITA also comes with functions that are able to extract the parameters of the system at said time. By historically managing parameters, designers and operators both can perform system maintenance without any worries or problems



Point

In order for the user to experience history management of Parameters, this scenario contained 2 executions.

# 6.2 Checking the CMDB Parameter history (1/3)

## History Check

- Check if the parameters are actually managed.

From the "Reference" menu group, go to the "Install package list" menu.

First, press the "Filter" button without inputting any filters.

The screenshot shows the Exastro IT Automation interface. In the top left, there's a navigation bar with 'Main menu' and a red box around the 'Install Package list' option under the 'Reference' menu group. A red circle labeled '1' points to this button. Below it is a 'Filter' dialog window. Inside the dialog, there are two input fields: 'Host name' and 'Operation', both of which have red boxes around them. A red circle labeled '2' points to the 'Operation' field. Below these fields are 'Filter' and 'Clear filter' buttons, with a red box around the 'Filter' button and a red circle labeled '3' pointing to it. To the right of the dialog is a large red box containing the text: 'Press the “Filter” button without inputting anything'. At the bottom of the dialog is a checkbox for 'Auto-filter'. The main content area shows a table titled 'List' with columns: History, No., Host name, ID, Operation name, Reference date, Scheduled date for execution, Last update date/time, and Last updated by. A single row is visible: History 1, Host name: ita-test-target02, ID: 21, Operation name: Operation2, Reference date: 2021/07/01 15:11, Scheduled date for execution: 2021/07/02 14:06, Last update date/time: 2021/07/01 14:16:46, Last updated by: System Administrator. A red box surrounds this table, and a red circle labeled '4' points to the 'Last updated by' column. Below the table are 'Output Excel' and 'Download all' buttons. At the bottom of the page are 'Contact administrator' and 'Trace history' links. A red box on the right contains the text: 'Check that the newest data is displayed correctly.'

## 6.2 Checking the CMDB Parameter history (2/3)

### History Check

- Now, we will input a reference date that took place earlier than the second execution and filter.

The screenshot shows the Exastro IT Automation interface with the 'Reference' module selected. The top navigation bar includes 'User name [System Administrator]', 'Login ID [administrator]', 'Change password', and 'Logout'. On the left, there's a 'Main menu' and 'Install Package list' option. The main content area has a 'Display filter' section and a 'List' section.

**1** In the 'Display filter' section, the 'Operation Reference date' field is highlighted with a red box and contains the value '2021/07/01 15:37'. A callout box with the text 'Input a date earlier than the second execution date.' points to this field.

**2** The 'Filter' button in the 'Display filter' section is highlighted with a red box.

**3** In the 'List' section, a single row of data is shown in a red box, indicating the result of the filter. The data includes:

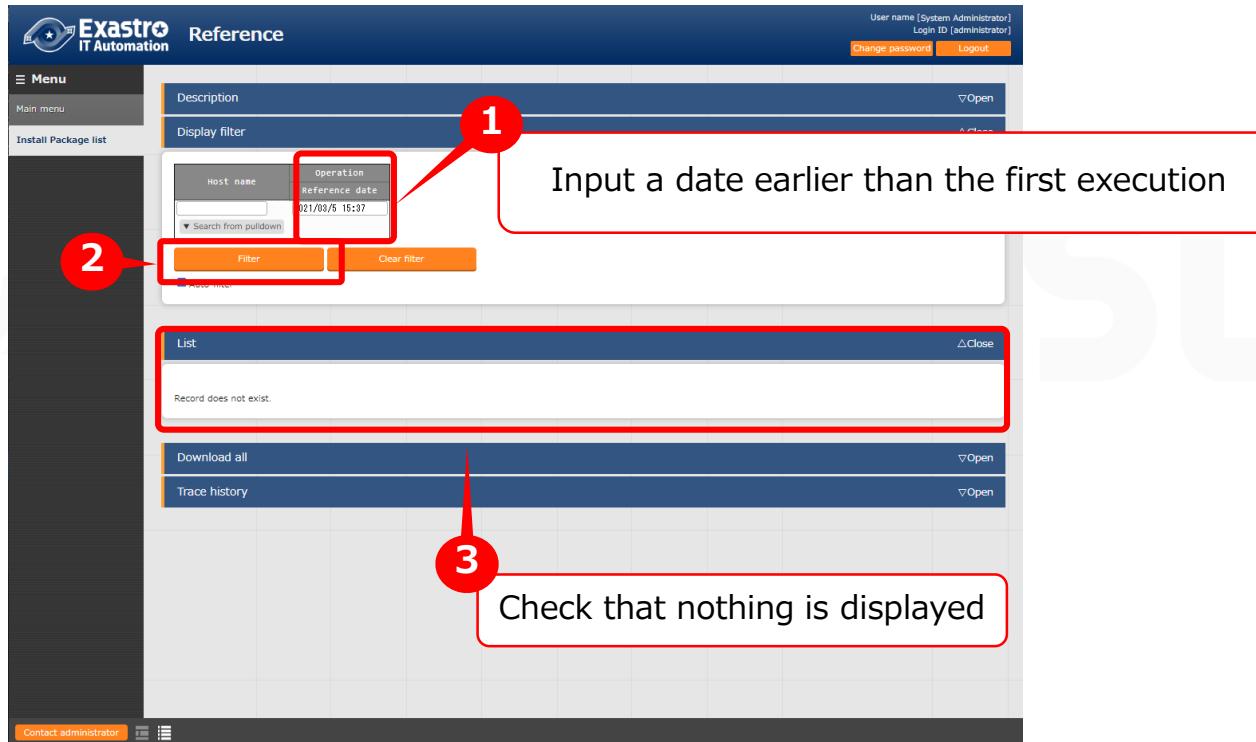
| History | No. | Host name         | ID | Operation name | Reference date   | Scheduled date for execution | Last                | Last update date/time | Last updated by     |
|---------|-----|-------------------|----|----------------|------------------|------------------------------|---------------------|-----------------------|---------------------|
| History | 1   | ita-test-target02 | 21 | Operation 1    | 2021/04/32 14:04 | 2021/07/02 14:06             | 2021/07/01 14:16:46 | 2021/07/01 14:16:46   | System Administrato |

A callout box with the text 'Check that only the first execution is displayed.' points to the list table.

# 6.2 Checking the CMDB Parameter history (3/3)

## History Check

- Lastly, input a date earlier than the first execution.



# A Appendix



# Reference ① [Ansible-Legacy] Single Execution

## Execution menu

- Ansible-Legacy has a "Execution" menu where users can execute individual movements and dry run them.

The screenshot shows the Ansible-Legacy interface with the following steps highlighted:

- Select a created Movement**: Step 1, selecting the first row in the Movement [List] table.
- Select an operation linked to the Movement**: Step 2, selecting the first row in the Operation [List] table.
- Dry run**: Step 3, performing a dry run to check connectivity and syntax.  
- Subsequent steps: **Execute**: Step 4, executing the playbook.

**Movement [List]** (Step 1):

| Movement ID | Movement Name  | Orchestrator   | Delay timer | Host specific format | WinRM connection | Header section | Last update date/time | Last updated by      |
|-------------|----------------|----------------|-------------|----------------------|------------------|----------------|-----------------------|----------------------|
| 17          | PackageInstall | Ansible Legacy | IP          |                      |                  |                | 2021/07/01 14:53:13   | System Administrator |

**Operation [List]** (Step 2):

| No. | Operation ID | Operation name | Scheduled date for execution | Last execution date | Access permission | Role to allow access | Last update date/time | Last updated by |
|-----|--------------|----------------|------------------------------|---------------------|-------------------|----------------------|-----------------------|-----------------|
| 1   | 21           | 21 Operation 1 | 2021/07/02 14:06             | 2021/07/01 15:11    |                   |                      | 2021/07/01 15:48:58   | System Admin    |
| 2   | 22           | 22 Operation 2 | 2021/07/03 14:37             | 2021/07/01 15:08    |                   |                      | 2021/07/01 15:08:32   | Legacy ex       |

**Dry run**: Checks the playbook's connectivity and syntax

**Execute**: Executes playbook.

# Reference ② [Ansible-Legacy] Execution check

## Execution result

- Pressing either the Execute or the Dry run button will move the user to a screen where execution status and logs are displayed.

The screenshot shows the Exastro IT Automation interface with the following components:

- Main Menu:** On the left, listing options like Main menu, Movement list, Playbook files, etc.
- Current Screen:** "Ansible-Legacy" - "Check operation status". It displays a table of execution parameters and a log table.
- Execution Log:** A table showing the progress of the execution, with rows for perl, python, and others.
- Error Log:** An empty table for displaying error logs.
- Download Buttons:** Buttons for "Input data" (populated data) and "Result data" (ResultData\_000000010.zip).
- Red Annotations:** Three red callout bubbles with the word "Point" pointing to specific areas:
  - Top Right:** Points to the execution log table with the text: "Here you can see the input data and the execution status."
  - Middle Right:** Points to the error log table with the text: "Here you can see both the execution log and the error log in real time."
  - Bottom Right:** Points to the download buttons with the text: "Here you can download both the input data and the result data."



**Exastro**