



IT Automation

Quickstart

※"IT Automation" will be written as "ITA" in this document

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](#)

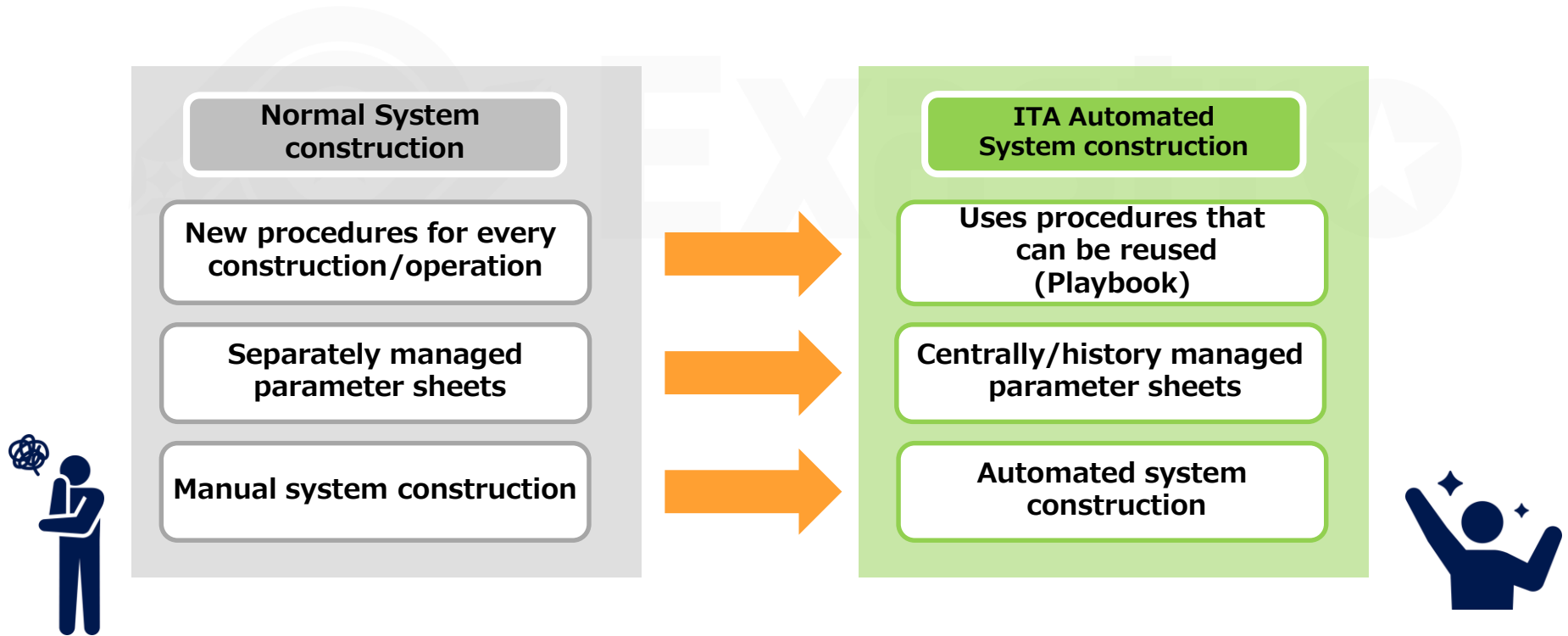
1. Introduction



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.

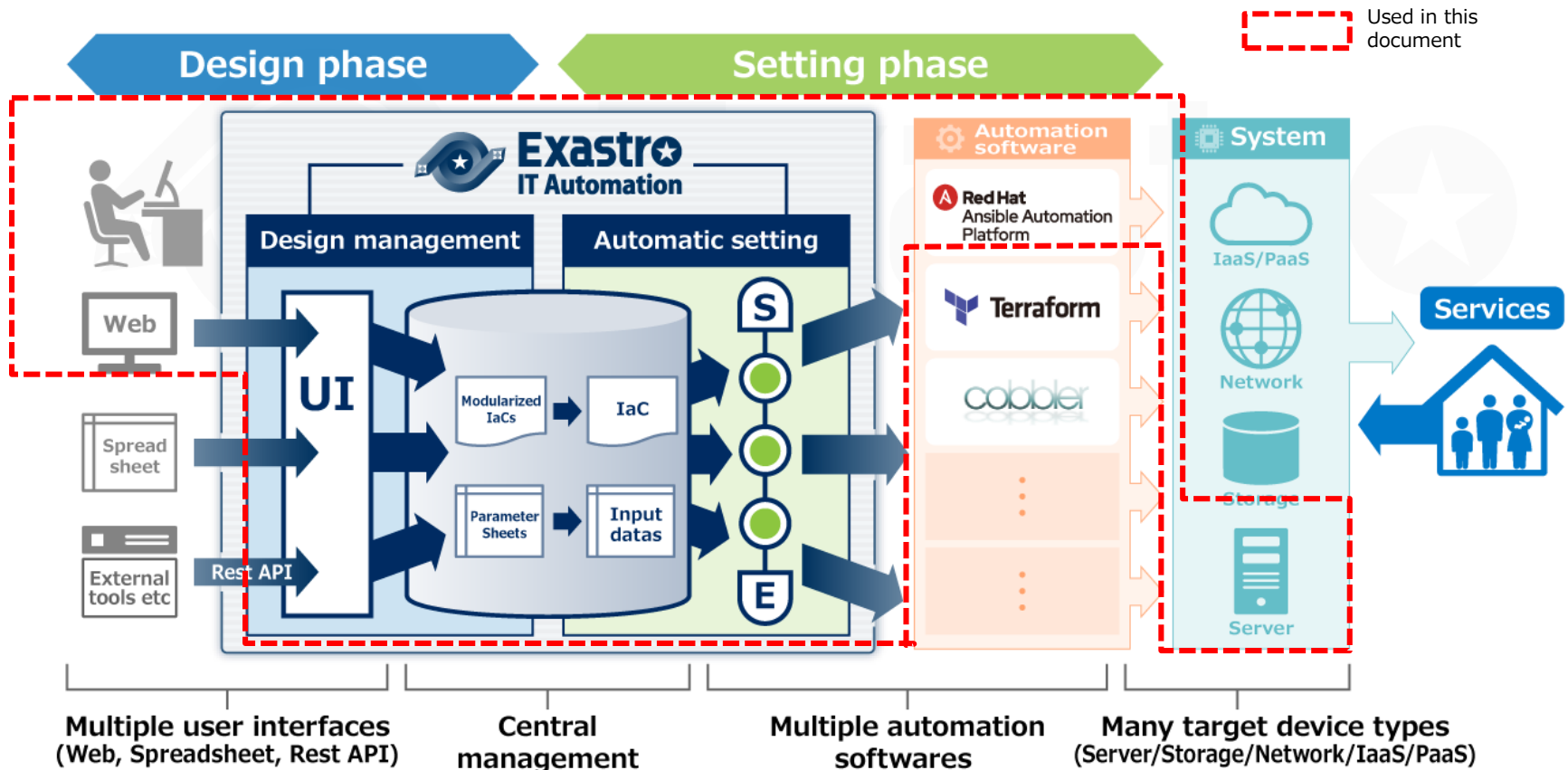
By installing Linux server packages, we can automate and centralize and automate package management for each server. That way, we can use ITA and have a more efficient system than we could achieve from a conventional system construction.



1.1 Introduction (2/2)

Main ITA functions used in this document.

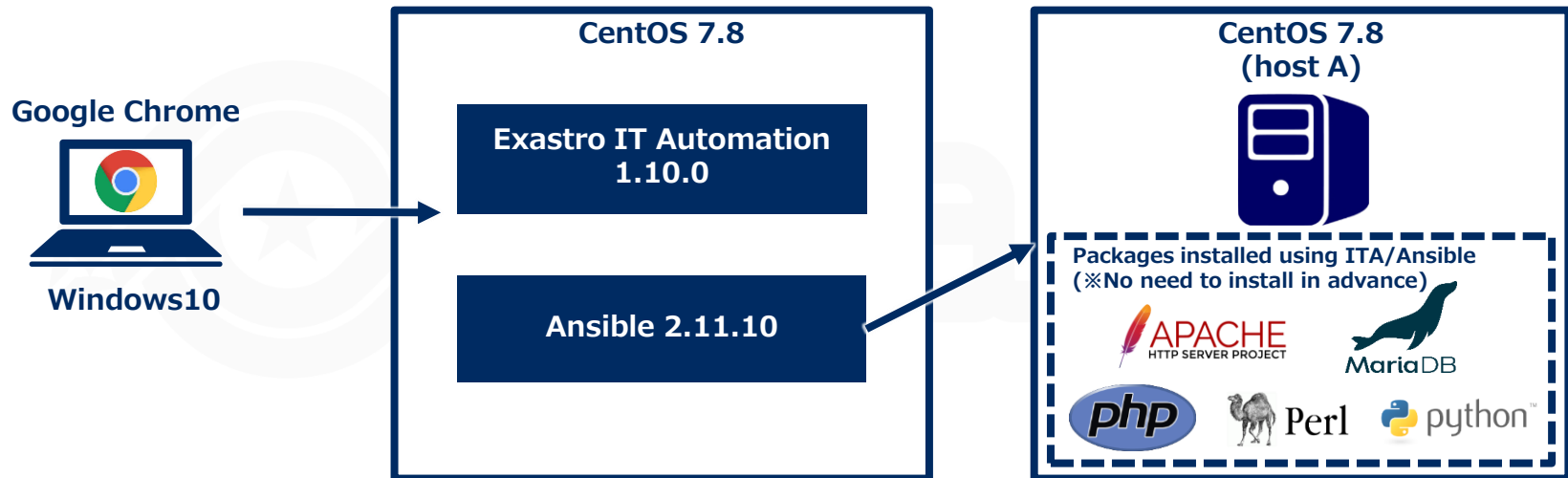
- Linking with Automation software (Ansible).
- Parameter management (Creation, Registration and history management of Menus)
- Linking Variables (Automatic registration of substitute values)



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

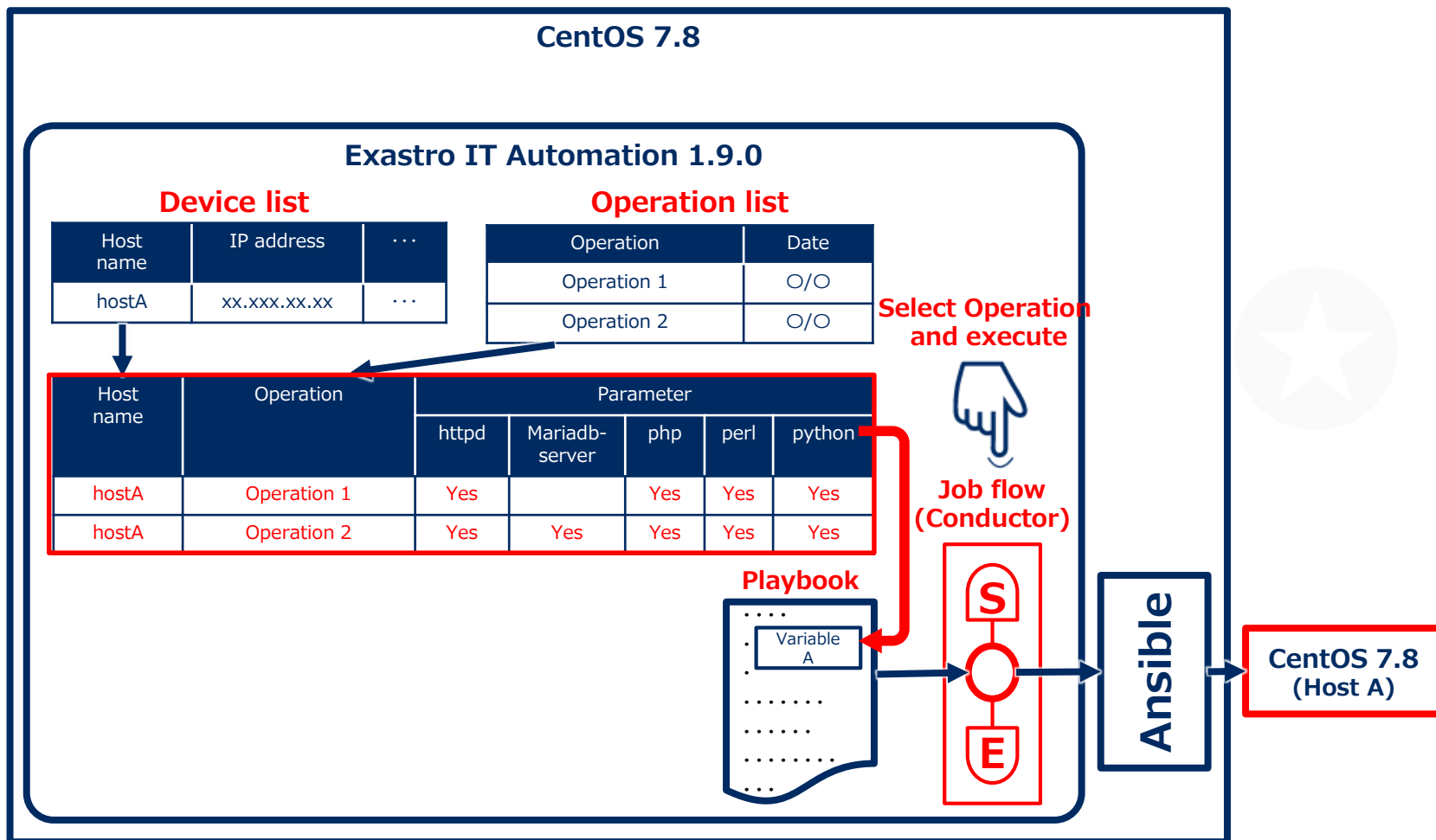


Systems used

- Exastro IT Automation 1.10.0
- 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)

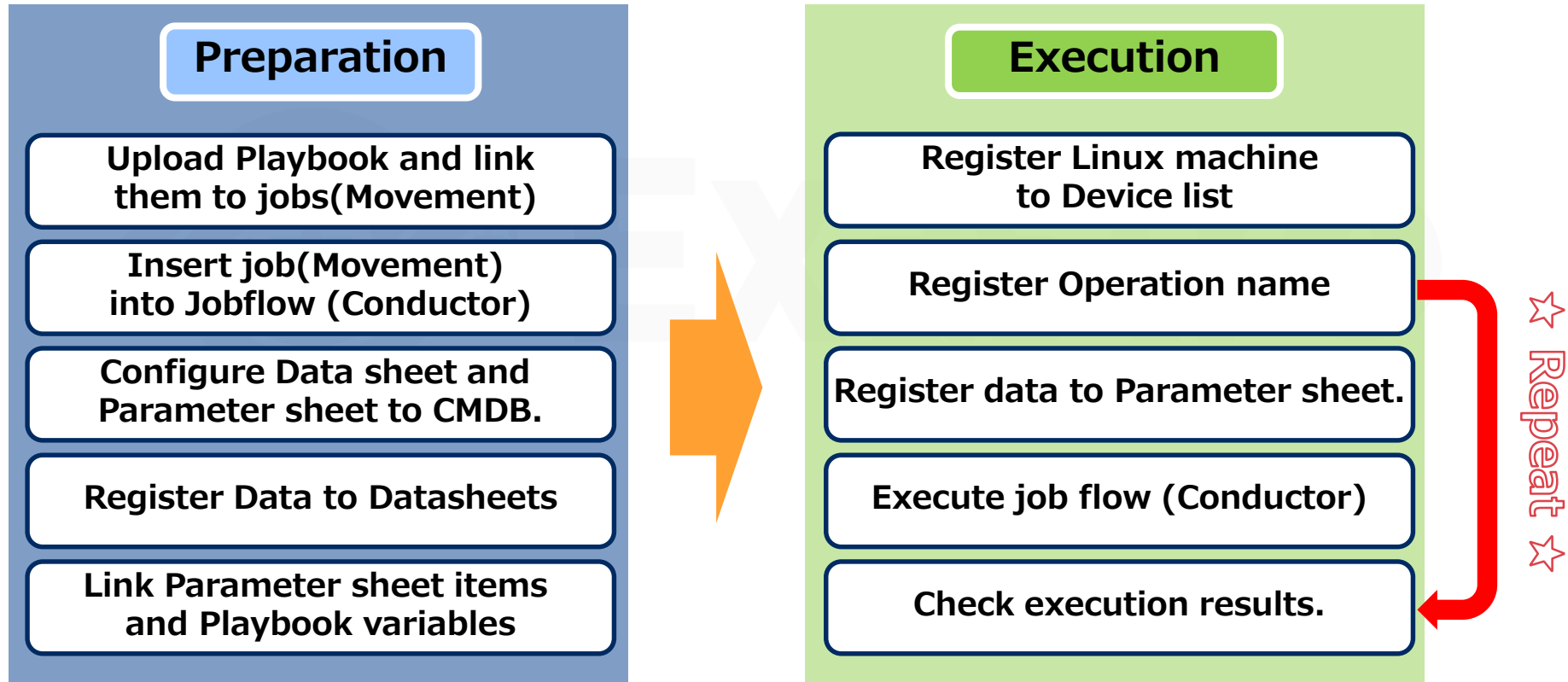
Scenario execution image



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.

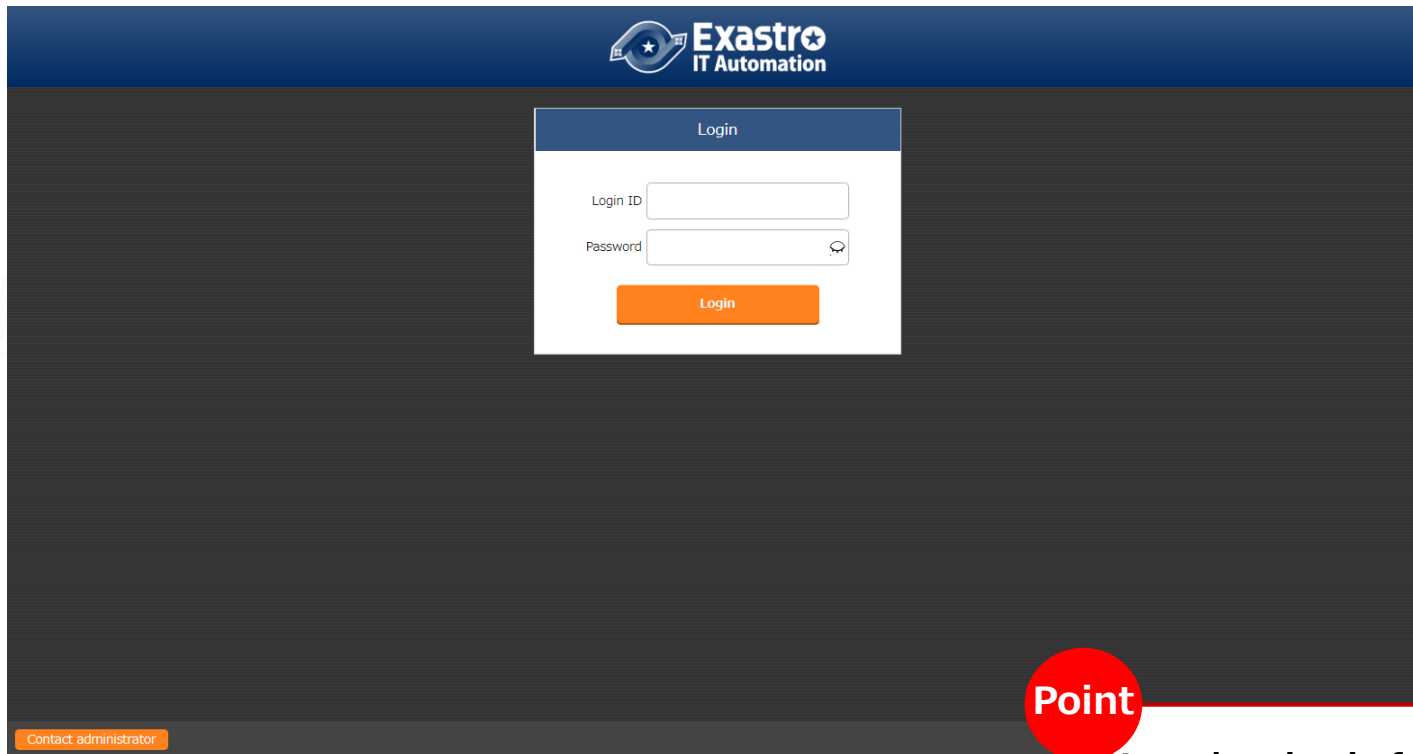
2. Screen Description



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 ask 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

Menu bar

Menu groups

Point

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

2.3 Screen Description (Menu) (1/2)

Screen Description (Menus)

- The name of the basic functions are as following.

The screenshot shows the Exastro Ansible-Legacy interface. The top navigation bar includes the Exastro logo, 'Ansible-Legacy', and user information (System Administrator). A left sidebar contains a 'Menu' section with items like 'Main menu', 'Movement list', 'Playbook files', etc. The main content area is divided into 'Description' and 'List/Update' sections. The 'Description' section has a 'Display filter' area with search fields and buttons. The 'List/Update' section contains a table of movement records.

Submenu

Submenu outline

- Explanation** : Contains a brief description regarding the menu.
- Display Filter** : Lets the user search for registered information
- List/Update** : Displays registered information

History	Duplicate	Update	Discard	Movement ID	Movement Name	Orchestrator	Access pe	Role to allow access	Created	Created 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		2021/12/08 15:37:25	Test2
History	Duplicate	Update	Discard	11	Package install	Ansible Legacy			2021/12/09 10:22:00	System Administrator

2.3 Screen Description (Menu) (2/2)

Screen description (Menu)

- The name of the basic functions are as following

The screenshot shows the Exastro Ansible-Legacy web interface. The top navigation bar includes the Exastro logo, the text 'Ansible-Legacy', and user information: 'User name [System Administrator]' and 'Login ID [administrator]'. There are 'Change password' and 'Logout' buttons. A left sidebar contains a 'Menu' section with various options like 'Main menu', 'Movement list', 'Playbook files', etc. The main content area displays a 'Register' sub-menu with several buttons: 'Start Registration', 'Download all and edit file uploads', 'Download all (Excel)', 'Download for new registration (Excel)', and 'Upload file'. Below these is a 'Trace history' section with a 'Movement ID' input field and 'Display' and 'Reset' buttons. A red box highlights the 'Register' sub-menu and its associated buttons. A red arrow points from the 'Submenu' label to the 'Register' sub-menu header.

■ 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

Submenu

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 YAML 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 without BOM" and the newline code is "LF". The file should be saved as an yml file. Please be check that the indents are correct.

```
yml 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 displays the Exastro Ansible-LegacyRole dashboard. The main menu on the left has 'Movement list' highlighted with a red box and a red circle containing the number '2'. The dashboard area shows several icons for management tools, with 'Ansible-Pioneer' highlighted by a red box and a red circle containing the number '1'. The dashboard includes three main panels: 'Movement' with a donut chart showing 13 total movements, 'Work status' with a gauge showing 0 total, and 'Work result' with a gauge showing 1 total. Below these are tables for 'Movement' and 'Work history'.

Movement	SUM
Ansible Legacy	10
Ansible Pioneer	1
Ansible Legacy Role	1
Terraform	1

Status	CON	SYM	SUM
Executing			
Unexecuted (schedule)			
Unexecuted			

Result	CON
Normal end	
Abnormal end	
Unexpected error	
Emergency stop	
Schedule cancellation	

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 'Ansible-LegacyRole' interface. The 'Register' button is highlighted with a red box and a callout '3'. The interface includes a menu on the left, a search bar at the top, and a table with columns: Discard, Movement ID, Movement Name, Orchestrator, Delay timer, Last update date/time, and Last updated by. Below the table are 'Filter' and 'Clear filter' buttons, and an 'Auto-filter' checkbox.

- 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' form. A table titled 'Dedicated information for ansible' is visible. The 'Register' button is highlighted with a red box and a callout '5'. The table has columns: Movement ID, Movement Name, Delay timer, Host specific, format, MIN/RN connection, Header sec, Last update date/time, and Last updated by.

Movement name	Host format
PackageInstall	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.

Uploading Playbook and linking it to a job(Movement)

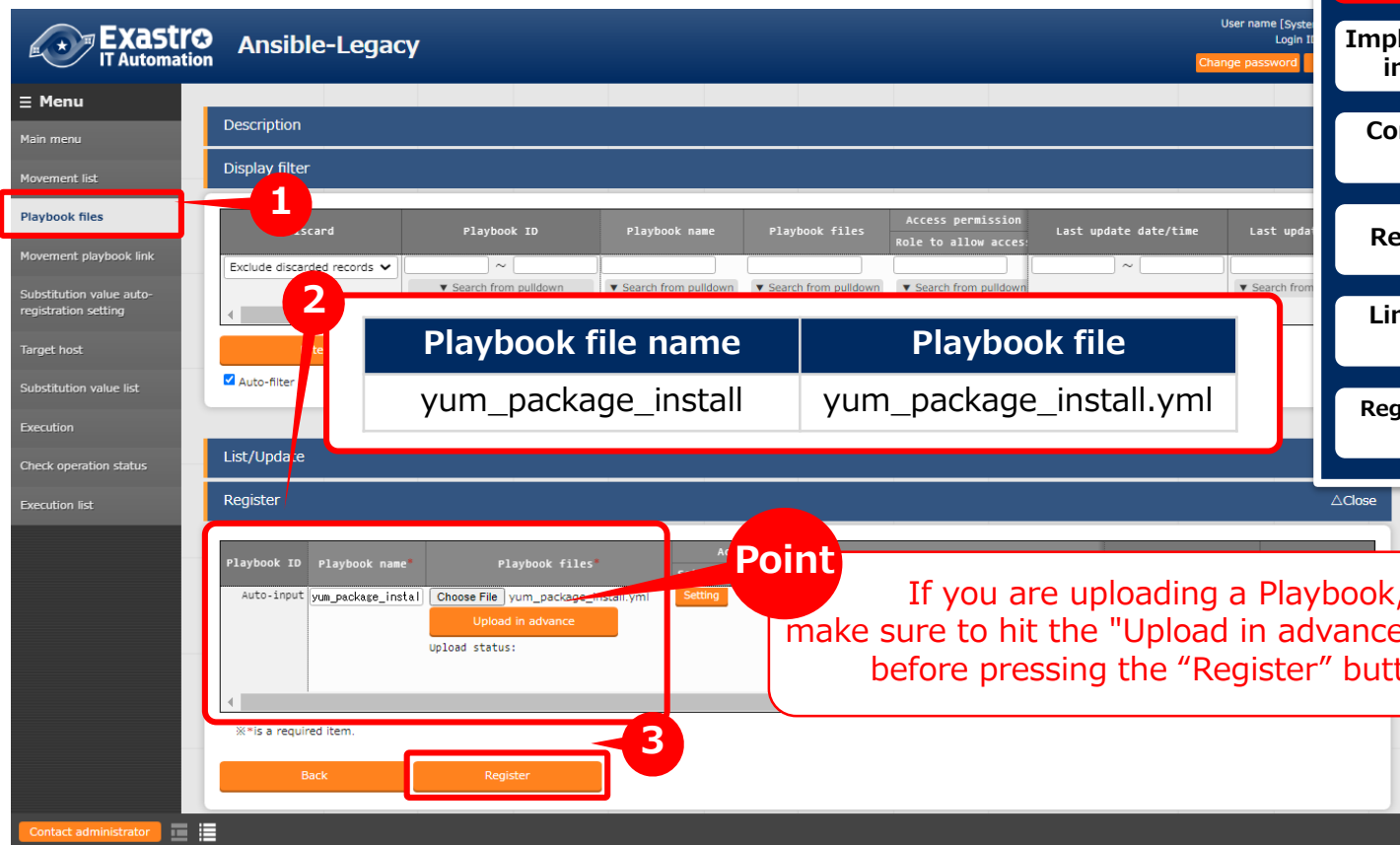
Implementing job (Movement) into Jobflow (Conductor)

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

If you are uploading a Playbook, make sure to hit the "Upload in advance" button before pressing the "Register" button.

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.

Uploading Playbook and linking it to a job(Movement)

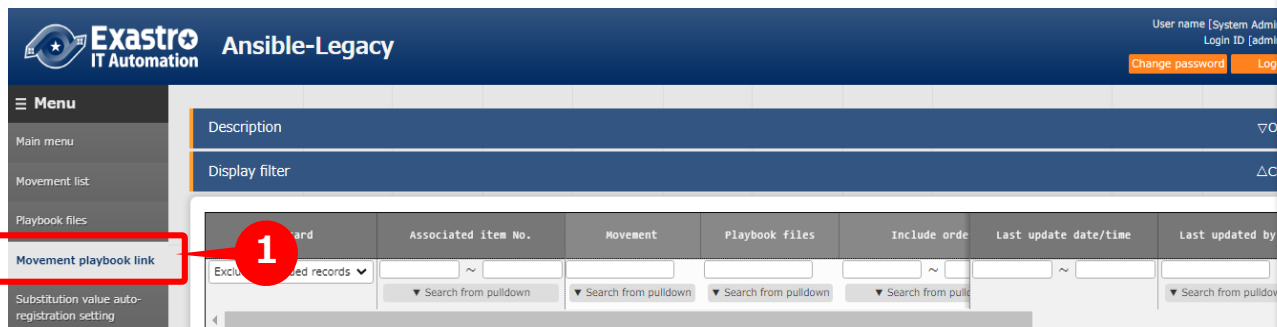
Implementing job (Movement) into Jobflow (Conductor)

Configure CMDB Data sheet and Parameter sheet

Register Data to Datasheet

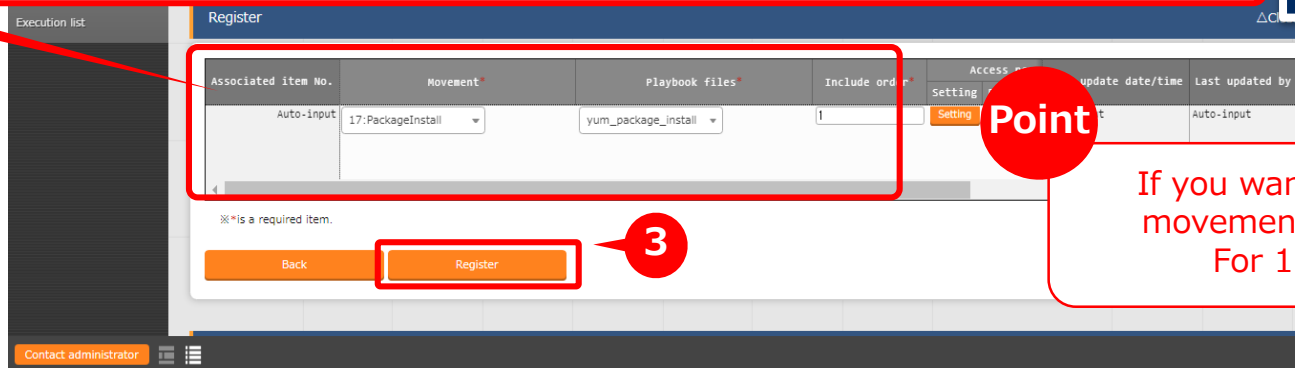
Link Parameter sheet item to Playbook variable.

Register target (Linux machine) to Device list.



Movement	Playbook file	Include order
Package Install	yum_package_install	1

2



Point

If you want to registered a single movement to multiple Playbooks. For 1:1, please input 1.

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.

The screenshot shows the Exastro Conductor interface. The main workspace displays a job flow diagram with nodes: Start, Conductor, Ansible Legacy, PackageInstall, and End. Annotations are as follows:

- 1**: Points to the "Conductor class edit" option in the left-hand menu.
- 2**: Points to the "Name" field in the right-hand configuration panel, which contains the text "InstallPackage".
- 3**: Points to the "Movement" section in the right-hand panel, which lists a movement with ID 17 and name "PackageInstall".
- 4**: Points to a red line being dragged between the "OUT" and "IN" ports of the "Conductor" node.
- 5**: Points to the "Registration" button at the bottom left of the interface.

Additional annotations include:

- A red box labeled "Name" containing "InstallPackage".
- A red arrow labeled "Input field for Remarks and such." pointing to a text area in the right-hand panel.
- A red arrow labeled "Drag and Drop" pointing to the red line between the "OUT" and "IN" ports.

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 (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.

1 Click "Group"

2 Click "Group"

3 Group name
Install Package

Menu name	Creation target	Display Order
Install Package list	Parameter sheet (Host/ Operation)	1

- 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

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)

5

- 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 (3/3)

Create Parameter sheet

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

6 Drag and drop the items into the column group

7 When finished, the items should look like this

8 Create

- 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.

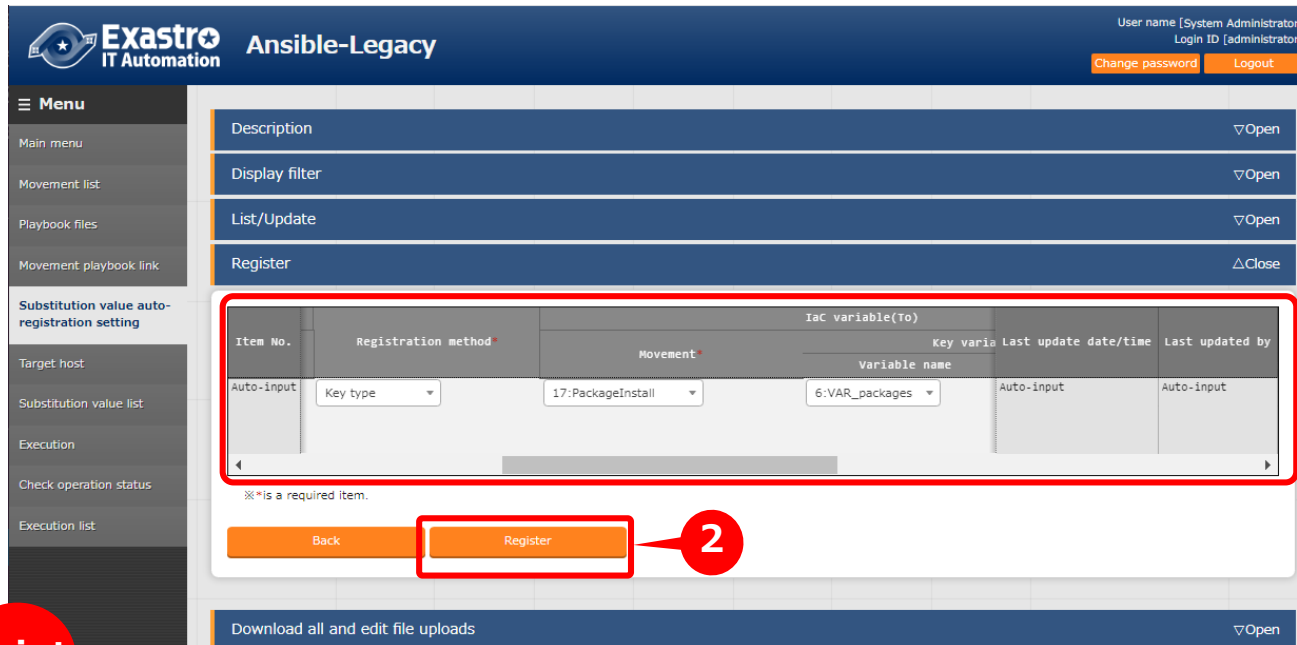
Menu group:Menu	Item	Registration method	Movement	Key Variable Variable name	Substitution order
2100011611:Substitution value:3:Install Package list	Parameter/Install Package/httpd	Key type	1:PackageInstall	1:VAR_packages	1
2100011611:Substitution value:3:Install Package list	Parameter/Install Package/mariadb-server	Key type	1:PackageInstall	1:VAR_packages	2
2100011611:Substitution value:3:Install Package list	Parameter/Install Package/php	Key type	1:PackageInstall	1:VAR_packages	3
2100011611:Substitution value:3:Install Package list	Parameter/Install Package/perl	Key type	1:PackageInstall	1:VAR_packages	4
2100011611:Substitution value:3:Install Package list	Parameter/Install Package/python	Key type	1:PackageInstall	1:VAR_packages	5

- 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.



- 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

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.

The screenshot shows the Exastro Ansible-Legacy interface. The left sidebar contains a menu with options like 'Main menu', 'Movement list', 'Playbook files', 'Movement playbook', 'Substitution value auto-registration setting', 'Target host', 'Substitution value', 'Execution', 'Check operation status', and 'Execution list'. The main content area is titled 'Substitution value auto-registration setting' and includes a 'Description' field, a 'Display filter' section, and a 'List' section. The 'Display filter' section contains a table with columns: 'Discard', 'Item No.', 'Menu group', 'Menu', 'Item', 'Last update date/time', and 'Last updated by'. Below this table are search filters and a 'Filter' button. The 'List' section shows a table with columns: 'History', 'Duplicate', 'Update', 'Discard', 'Item No.', 'Menu group', 'Menu', 'Item', 'Last update date/time', and 'Last updated by'. The table contains 5 rows of data. A red circle with the number '3' highlights the 'Filter' button. A red circle with the number '4' highlights the 'Filter' button. A red circle with the number '5' highlights the 'List' section. A red box highlights the 'List' section. A red box highlights the 'Filter' button. A red box highlights the 'List' section. A red box highlights the 'Filter' button. A red box highlights the 'List' section.

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.

The screenshot shows the Exastro IT Automation Basic Console interface. The left sidebar menu has '機器一覧' (Device List) highlighted with a red box and a red circle containing the number '1'. The main content area shows a registration form for a device. The form has several fields: '管理システム項番' (Management System Item No.) with '自動入力' (Automatic Input) and 'SV' dropdown; 'HW機器種別' (HW Device Type) with 'SV' dropdown; 'ホスト名*' (Host Name*) with 'hostA'; 'IPアドレス*' (IP Address*) with '192.168.10.1'; 'MACアドレス' (MAC Address); 'ネットワークデバイス名' (Network Device Name); 'ログ' (Log); '最終更新日時' (Last Update Date) with '自動入力'; and '最終更新者' (Last Updated By) with '自動入力'. A red box highlights the form fields, and a red circle with the number '2' points to a summary table below the form.

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

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 'Basic Console' interface for Exastro IT Automation. The 'Register' form is open, and a red box highlights the input fields. A red circle with the number '2' points to the 'Managed system item number' field. Below the form, a table shows the login details for the user.

Managed system item number	HW device type	Host name*	IP address*	MAC address	Netw	Last update date/time	Last updated by
Auto-Input	SV					Auto-Input	Auto-Input

Login user ID	Login password management	Login Password
(Login user ID)	●	(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.

The screenshot shows the 'Ansible利用情報' form in the Exastro IT Automation console. The form is divided into several sections: 'Ansible利用情報', 'Legacy/Role利用情報', and 'WinRM接続情報'. The 'Legacy/Role利用情報' section is highlighted with a red box and contains the following fields:

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

Below the form, there are two buttons: '戻る' and '登録'. The '登録' button is highlighted with a red box and a red circle with the number '4'. A red circle with the number '3' points to the 'パスワード認証' dropdown menu.

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. The 'Operation list' menu item is highlighted in the left sidebar with a red box and a red circle containing the number '1'. The main content area shows the 'Register' form with the following fields filled out:

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	Role to allow access	Auto-input	Auto-input

The 'Register' button is highlighted with a red box and a red circle containing the number '3'. A red circle containing the number '2' points to the 'Scheduled date for execution' field. A red box highlights the 'Operation name' and 'Scheduled date and time' fields, with a callout box containing the text:

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

Register Operation name (Operation)

Register data to Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

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.

Register Operation name
(Operation)

Register data to
Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

The screenshot shows the Exastro IT Automation 'Input' interface. The 'Install Package list' menu item is highlighted with a red box and a '1' in a red circle. The 'Register' form is highlighted with a red box and a '2' in a red circle. The 'Register' button is highlighted with a red box and a '3' in a red circle. The form contains a table with columns for Host name, Operation, and various parameters (httpd, mariadb-server, php, perl, python).

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

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 IT Automation 'Input' interface. The left sidebar contains a 'Menu' with options like 'Main menu', 'Global IP(Data sheet)', 'Install Package list', 'SSL certificate', 'Private IP(Data sheet)', 'OS information', 'Compare test 1', and 'SSL certificate name'. The main area displays the 'Install Package list' with a search bar and a table of operations. The table has columns for 'History', 'Duplicate', 'Update', 'Discard', 'No.', 'Host name', 'ID', 'Operation name', 'Reference date', 'Scheduled date for execution', 'Last update date/time', and 'Last updated by'. A red box highlights the 'Display filter' button, the 'Filter' button, and the first row of the table. The first row contains: History (green), Duplicate (purple), Update (orange), Discard (red), No. (1), Host name (targethost), ID (7), Operation name (Operation 1), Reference date (2021/12/30 13:32), Scheduled date for execution (2021/12/30 13:32), Last update date/time (2021/12/09 13:33:44), and Last updated by (System Administrator). Below the table, it says 'Filter result count: 1' and there is an 'Output Excel' button.

Register Operation name
(Operation)

Register data to
Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

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".

1 Conductor execution

Select	Conductor class ID	Conductor name	Explanation	Access permission	Remarks	Last update date/time	Last updated by
<input type="radio"/>	2	InstallPackage		Role to allow access		2021/07/01 11:34:52	System Administrator

Filter result count: 1

2

Select	No.	Operation ID	Operation name	Scheduled date for execution	Last execution date	Role
<input type="radio"/>	21	21	Operation 1	2021/07/02 14:06		

Filter result count: 1

3

4 Execution

Register Operation name (Operation)

Register data to Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

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.

The screenshot displays the Exastro Conductor interface. The top navigation bar includes the Exastro logo, the word "Conductor", and user information: "User name [System Administrator]" and "Login ID [administrator]". There are buttons for "Change password" and "Logout". A left sidebar menu lists various options, with "Conductor confirmation" selected. The main area shows a workflow diagram with three nodes: "Start" (blue circle with 'S'), "Assemble Legacy PackageInstall" (red circle with 'DONE'), and "End" (blue circle with 'DONE'). The status "CHECKING" is visible at the top left of the main area. On the right, a panel shows details for the conductor instance: "Conductor name: InstallPackage", "Conductor instance ID: 3", "Status: Successful completion", "Start time: 2021/07/01 14:26:03", "End time: 2021/07/01 14:26:36", "Execution user: System Administrator", "Reservation data:", and "Emergency stop:". Below this, there is a "Note" field. At the bottom, there is a "log" section and an "Emergency stop" button. A "Contact administrator" button is located at the bottom left.

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 displays the Exastro Conductor interface. The main workspace shows a job flow with a 'Start' node and a 'PackageInstall' node. The 'PackageInstall' node is highlighted with a red box and a '1 Select' callout. A red arrow points from this node to a detailed view of the 'Ansible-Legacy' operation. This view includes a table of execution items and a list of operation details.

Item	Execution No.	Execution type	Status	execution engine	Caller symphony	Caller conductor	Execution user
PackageInstall	3	Normal	Completed	Ansible Engine		InstallPackage	System Administrator

Operation		
No.	21	
Name	Operation 1	
ID	21	
Host management		confirmation
Substitution value		confirmation
Input data	Populated data	InputData_000000002.zip
Output data	Result data	ResultData_000000003.zip
Operation status		
Start date/time		2021/07/01 14:26:08
End date/time		2021/07/01 14:26:30

On the right side of the interface, a vertical list of actions is shown:

- Register Operation name (Operation)
- Register data to Parameter sheet
- Execute Jobflow (Conductor)** (highlighted in red)
- 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 displays the Ansible execution results interface. On the left is a navigation menu with options like 'Menu', '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 is divided into several sections:

- Summary Table:** A table with columns for 'Input data', 'Output data', 'Operation status', 'Populated data', 'Result data', 'Scheduled date/time', 'Start date/time', and 'End date/time'. It contains links for 'AllData @000000002.rjk' and 'ResultData @000000003.rjk', and dates '2021/07/01 14:26:08' and '2021/07/01 14:26:30'.
- Progress status(Execution log):** A section with a filter input and a checkbox 'Display only corresponding lines'. It contains a scrollable log with the following content:

```
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
```
- Progress status(Error log):** A section with a filter input and a checkbox 'Display only corresponding lines'. It is currently empty.

At the bottom left, there is a 'Contact administrator' button and a hamburger menu icon.

Register Operation name
(Operation)

Register data to
Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

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. The 'Operation list' menu item is highlighted with a red box and a red circle containing the number 1. The 'Register' form is highlighted with a red box and a red circle containing the number 2. The 'Register' button is highlighted with a red box and a red circle containing the number 3. The form contains the following fields:

No.	Operation ID	Operation name	Scheduled date for execution	Setting	Ac	Session	Last update date/time	Last updated by
Auto-Input	Auto-Input	Operation 2	2021/07/03 14:37	Setting				

**Register Operation name
(Operation)**

**Register data to
Parameter sheet**

Execute Jobflow (Conductor)

Check Execution results

Operation name

Operation 2

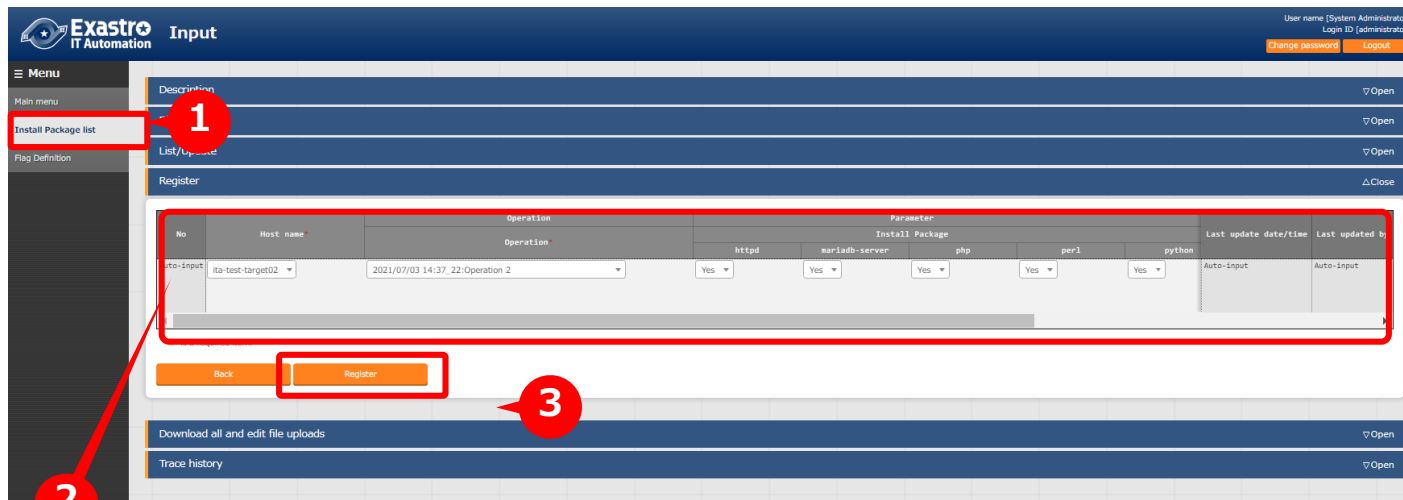
**Reservation
date/time**

(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.



Register Operation name (Operation)

Register data to Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

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 Conductor execution

2 Conductor [List]

Select	Conductor class ID	Conductor name	Explanation	Access permission	Remarks	Last update date/time
<input checked="" type="radio"/>	2	InstallPackage		Role to allow access		2021/07/01 11:34:52

Filter result count: 1

3 Operation [List]

Select	No.	Operation ID	Operation name	Scheduled date for execution	Last execution date	Access	Last update
<input type="radio"/>	21	21	Operation 1	2021/07/02 14:06	2021/07/01 14:26		2021/07/01
<input checked="" type="radio"/>	22	22	Operation 2	2021/07/03 14:37			2021/07/01

Filter result count: 2

4 Execution

Register Operation name (Operation)

Register data to Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

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.

The screenshot displays the Exastro Conductor interface. The top navigation bar includes the Exastro logo and the word "Conductor". A user menu in the top right corner shows "User name [System Administrator]" and "Login ID [administrator]", with "Change password" and "Logout" buttons. A left sidebar menu lists various options, with "Conductor confirmation" selected. The main workspace features a grid with a jobflow diagram consisting of three nodes: "Start" (blue circle with 'S'), "Ansible Legacy PackageInstall" (red circle with 'DONE'), and "End" (blue circle with 'DONE'). A right-hand panel provides details for the selected conductor instance, including its name, ID, status ("Successful completion"), start and end times, execution user, and emergency stop options. Below this, a "log" section is visible, and an "Emergency stop" button is located at the bottom left of the main workspace.

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. A job execution flow is visible, with a 'DONE' icon highlighted. A red box and a callout '1 Select' point to this icon. A red arrow points from the 'DONE' icon to the 'Check operation status' button in the right-hand panel. Another red arrow points from the 'Check operation status' button to the 'Check operation status' menu item in the left-hand sidebar. The right-hand panel displays a table of execution details.

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

Register Operation name
(Operation)

Register data to
Parameter sheet

Execute Jobflow (Conductor)

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.

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

Progress status(Execution log) △Close

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
```

Progress status(Error log) △Close

Register Operation name
(Operation)

Register data to
Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

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 (htpd,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)

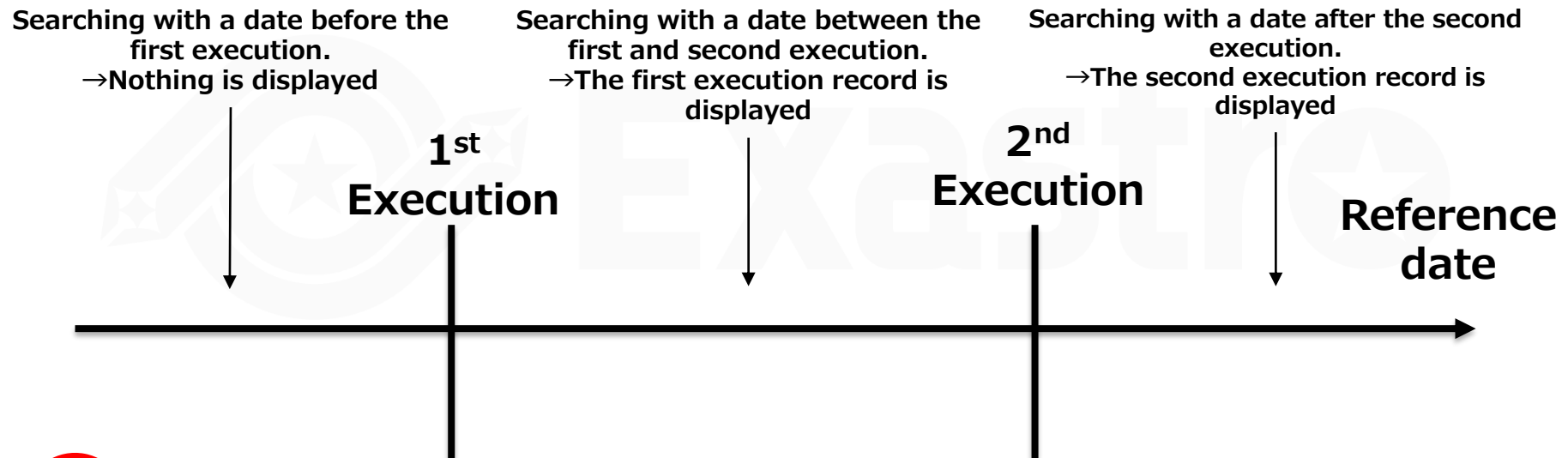
Check Execution results

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 Reference interface. The top navigation bar includes the Exastro logo, the word 'Reference', and user information: 'User name [System Administrator]', 'Login ID [administrator]', 'Change password', and 'Logout'. A left sidebar menu is open, showing 'Main menu' and 'Install Package list' (highlighted with a red box and number 1). The main content area has a 'Description' section with a search bar for 'Host name' and 'operation Reference date'. A 'Filter' button (highlighted with a red box and number 3) is visible. A callout box with number 2 points to the search area, containing the text 'Press the "Filter" button without inputting anything'. Below the filter section is a 'List' section containing a table of history data (highlighted with a red box and number 4). The table has columns: History, No, Host name, ID, Operation name, Reference date, Scheduled date for execution, Last update date/time, and Last updated by. The first row of data is: History, 1, ita-test-target02, 21, Operation2, 2021/07/01 15:11, 2021/07/02 14:06, 2021/07/01 14:16:46, System Administrator. Below the table is an 'Output Excel' button. At the bottom of the interface, there is a 'Download all' button and a 'Trace history' button.

History	No	Host name	ID	Operation name	Reference date	Scheduled date for execution	Last update date/time	Last updated by
History	1	ita-test-target02	21	Operation2	2021/07/01 15:11	2021/07/02 14:06	2021/07/01 14:16:46	System Administrator

Press the "Filter" button without inputting anything

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 'Reference' page in the Exastro IT Automation system. The interface includes a menu on the left, a top navigation bar with user information, and a main content area. The 'Display filter' section is active, showing a filter form with fields for 'Host name', 'Operation', and 'Reference date'. The 'Reference date' field is set to '2021/07/01 15:37'. A red box highlights the 'Filter' button, and another red box highlights the 'Reference date' field. A red callout box with the number '1' points to the 'Reference date' field, containing the text 'Input a date earlier than the second execution date.' Below the filter form, the 'List' section displays a table of history records. A red box highlights the table, and a red callout box with the number '3' points to the table, containing the text 'Check that only the first execution is displayed.' The table has the following data:

History	No	Host name	ID	Operation name	Reference date	Scheduled date for execution	Last update date/time	Last updated by
History	1	lita-test-target02	21	Operation 1	2021/04/22 14:04	2021/07/02 14:06	2021/07/01 14:16:46	System Administrator

6.2 Checking the CMDB Parameter history (3/3)

History Check

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

The screenshot shows the Exastro IT Automation Reference interface. The top navigation bar includes the Exastro logo, the word 'Reference', and user information: 'User name [System Administrator]' and 'Login ID [administrator]'. There are 'Change password' and 'Logout' buttons. A left sidebar contains a 'Menu' section with 'Main menu' and 'Install Package list'. The main content area has a 'Display filter' section with a table containing 'Host name', 'Operation', and 'Reference date'. The 'Reference date' field is set to '2021/08/06 16:37'. Below the table are 'Filter' and 'Clear filter' buttons. A 'List' section below shows 'Record does not exist.' There are also 'Download all' and 'Trace history' buttons. Three red callouts are present: '1' points to the 'Reference date' field, '2' points to the 'Filter' button, and '3' points to the 'List' section.

1 Input a date earlier than the first execution

2

3 Check that nothing is displayed

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 web interface. The top navigation bar includes the Exastro IT Automation logo, the text "Ansible-Legacy", and user information: "User name [System Administrator]", "Login ID [administrator]", "Change password", and "Logout". A left sidebar menu is visible with options like "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 "Movement [List]" table with the following data:

Select	Movement ID	Movement Name	Orchestrator	Delay timer	Dedicated information for ansible			Last update date/time	Last updated by	
					Host specific format	MinRM connection	Header section	Opt		
<input checked="" type="radio"/>	17	PackageInstall	Ansible Legacy		IP				2021/07/01 14:53:13	System Administrator

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

The "Operation [List]" table below it has the following data:

Select	No.	Operation ID	Operation name	Scheduled date for execution	Last execution date	Access permission	Last update date/time	Last updated by
						Role to allow access		
<input checked="" type="radio"/>	21	21	Operation 1	2021/07/02 14:06	2021/07/01 15:11		2021/07/01 15:48:58	System Administrator
<input type="radio"/>	22	22	Operation 2	2021/07/03 14:37	2021/07/01 15:08		2021/07/01 15:08:32	Legacy ex...

Below the table, it says "Filter result count: 2".

At the bottom, the "Movement ID 17" and "Movement Name PackageInstall" are shown. Two buttons are visible: "Dry run" and "Execute".

1 Select a created Movement

2 Select an operation linked to the Movement

3 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 Ansible-Legacy interface. The main content area displays a table with execution details:

Item	Value
Execution No.	10
Execution type	Normal
Status	Completed
execution engine	Ansible Engine
Caller symphony	
Caller conductor	
Execution user	System Administrator
ID	17
Name	Packageinstall
Delay timer (minutes)	
Dedicated information for ansible	Host specific format WINRM connection
No.	21
Name	Operation 1
ID	21
Host management	confirmation
Substitution value	confirmation
Input data	Populated data InputData_0000000010.zip
Output data	Result data ResultData_0000000010.zip
Scheduled date/time	
Operation status	Start date/time 2021/07/01 15:43:58 End date/time 2021/07/01 15:44:04

Below the table, there are two tabs: "Progress status(Execution log)" and "Progress status(Error log)". The "Progress status(Execution log)" tab is active, showing a filter input and a checkbox for "Display only corresponding lines". The log content is partially visible at the bottom of the screenshot.

Point

Here you can see the input data and the execution status.

Point

Here you can see both the execution log and the error log in real time.

Point

Here you can download both the input data and the result data.



Exastro