IT Automation

Quickstart

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

Exastro IT Automation Version 1.10

Exastro developer

Table of contents

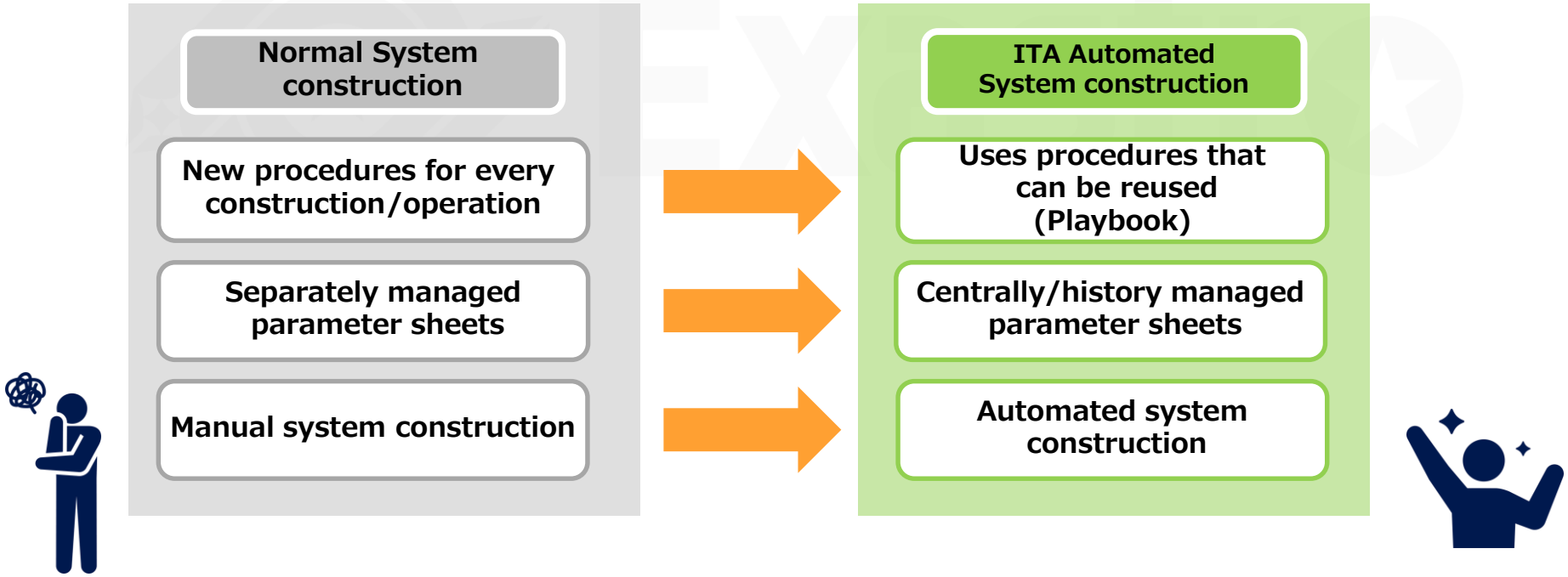
Table of Contents

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.

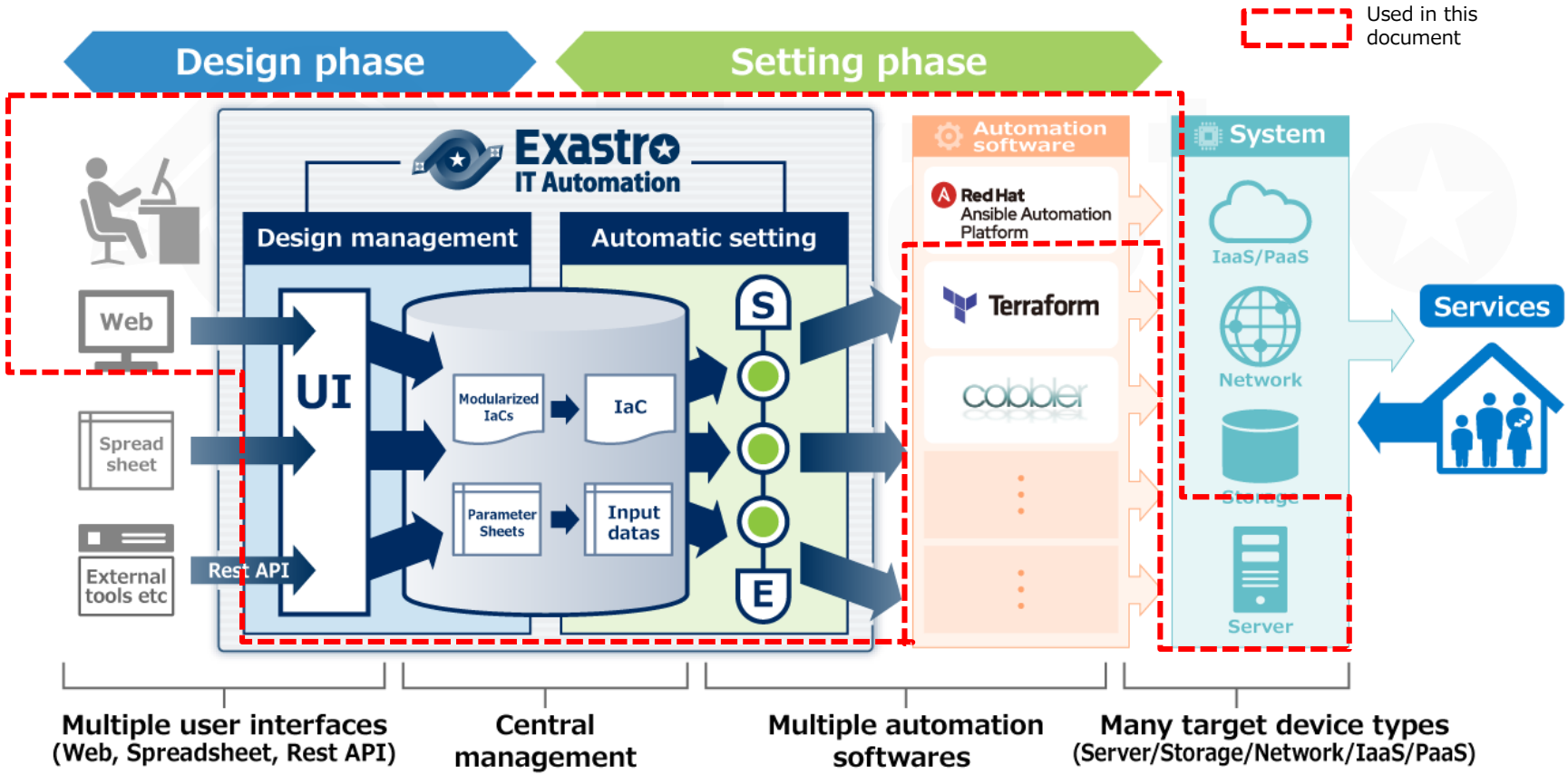
By installing Linux server packages, we can automate and centralize and automate package management for each server. That way, we can use ITA and have a more efficient system than we could achieve from a conventional system construction.



1.1 Introduction (2/2)

Main ITA functions used in this document.

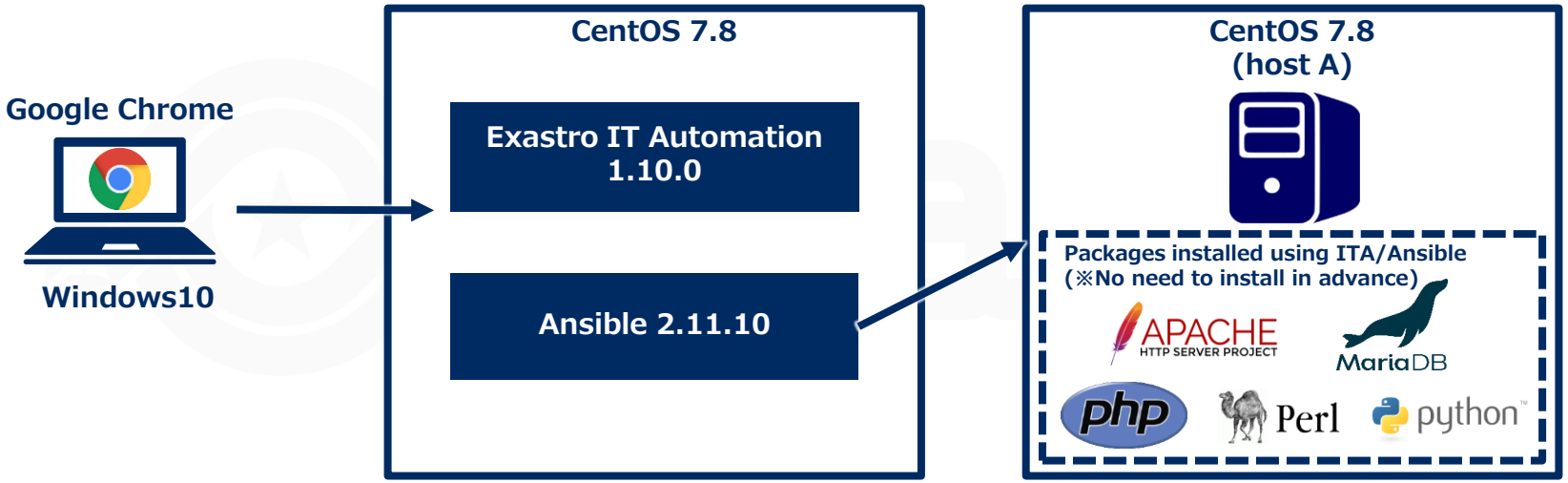
- Linking with Automation software (Ansible).
- Parameter management (Creation, Registration and history management of Menus)
- Linking Variables (Automatic registration of substitute values)



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

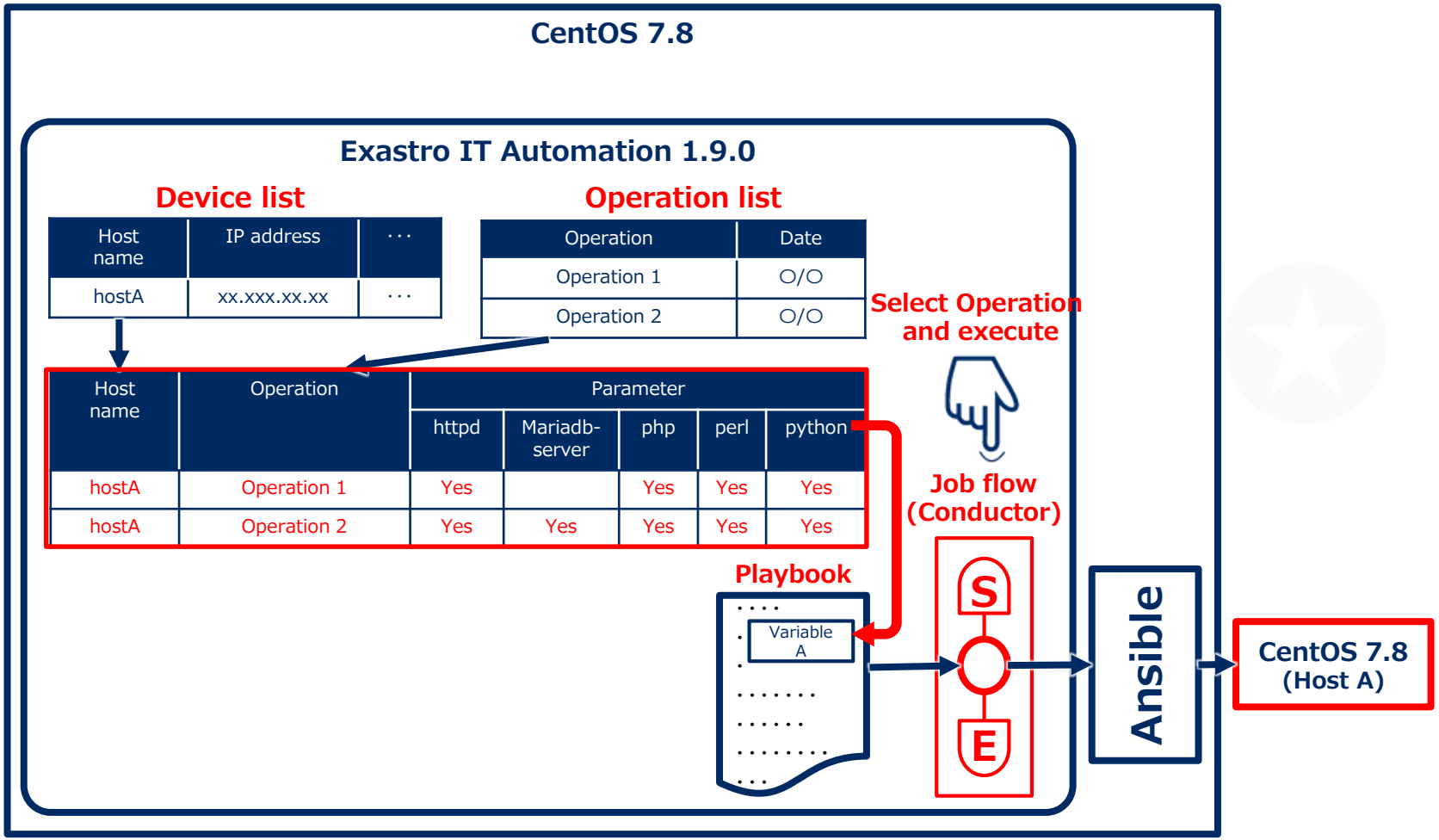


Systems used

- 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)

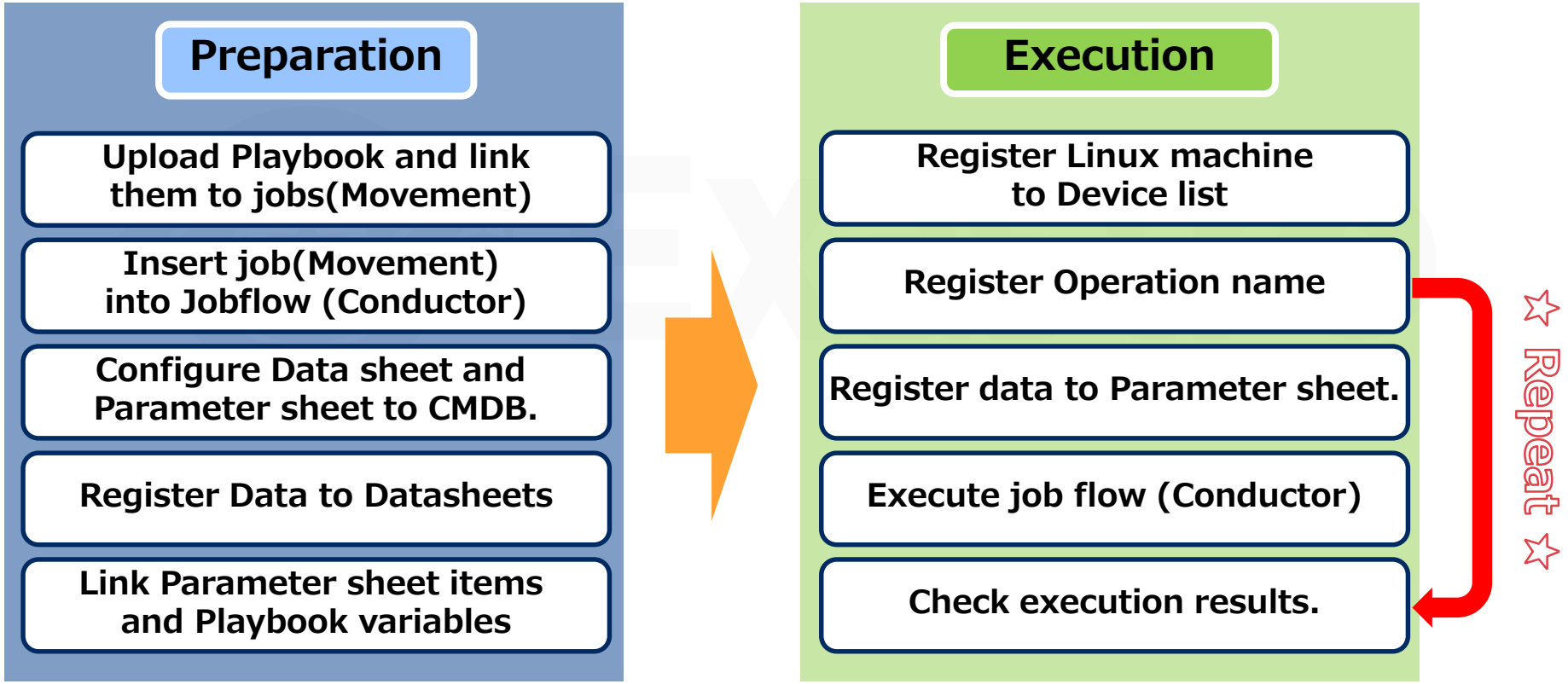
Scenario execution image



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.

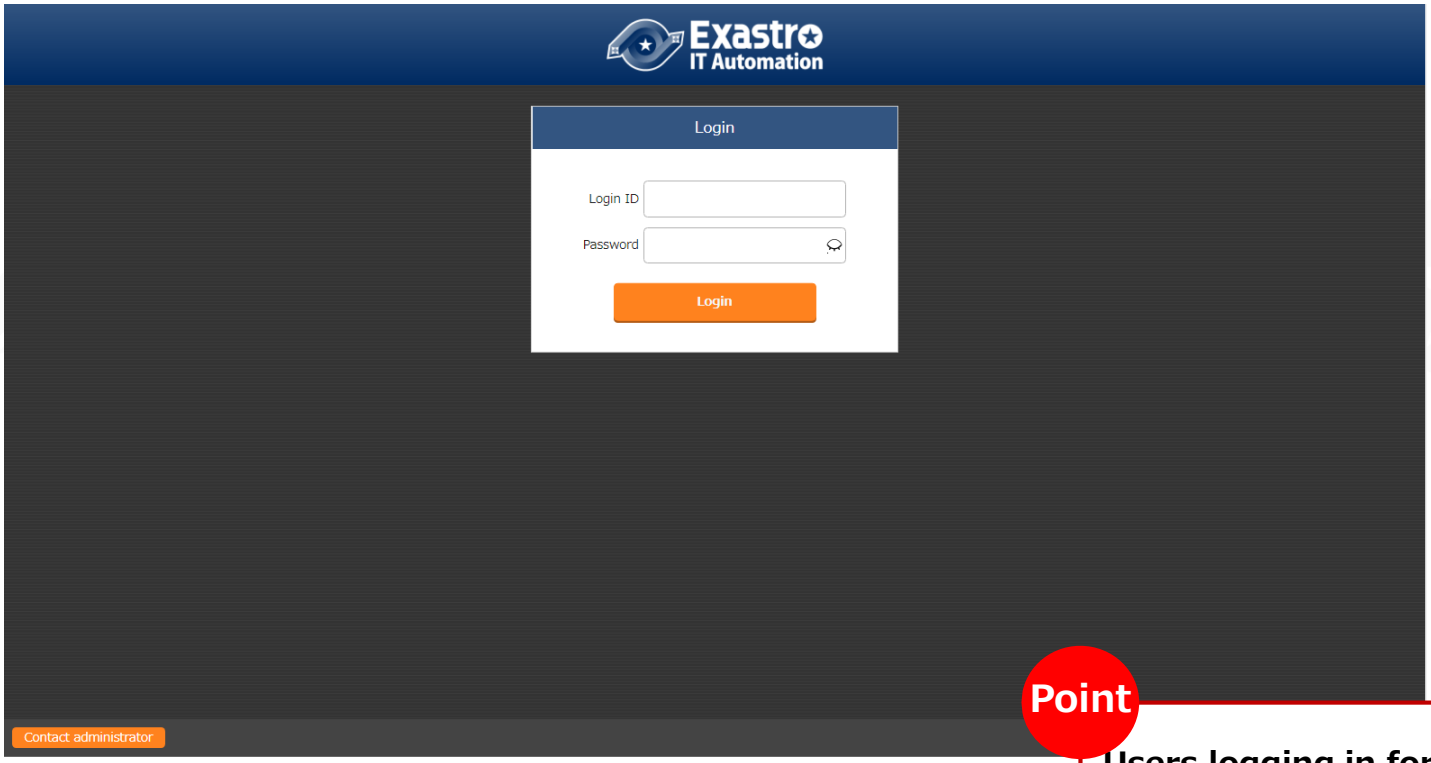
2. Screen Description

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

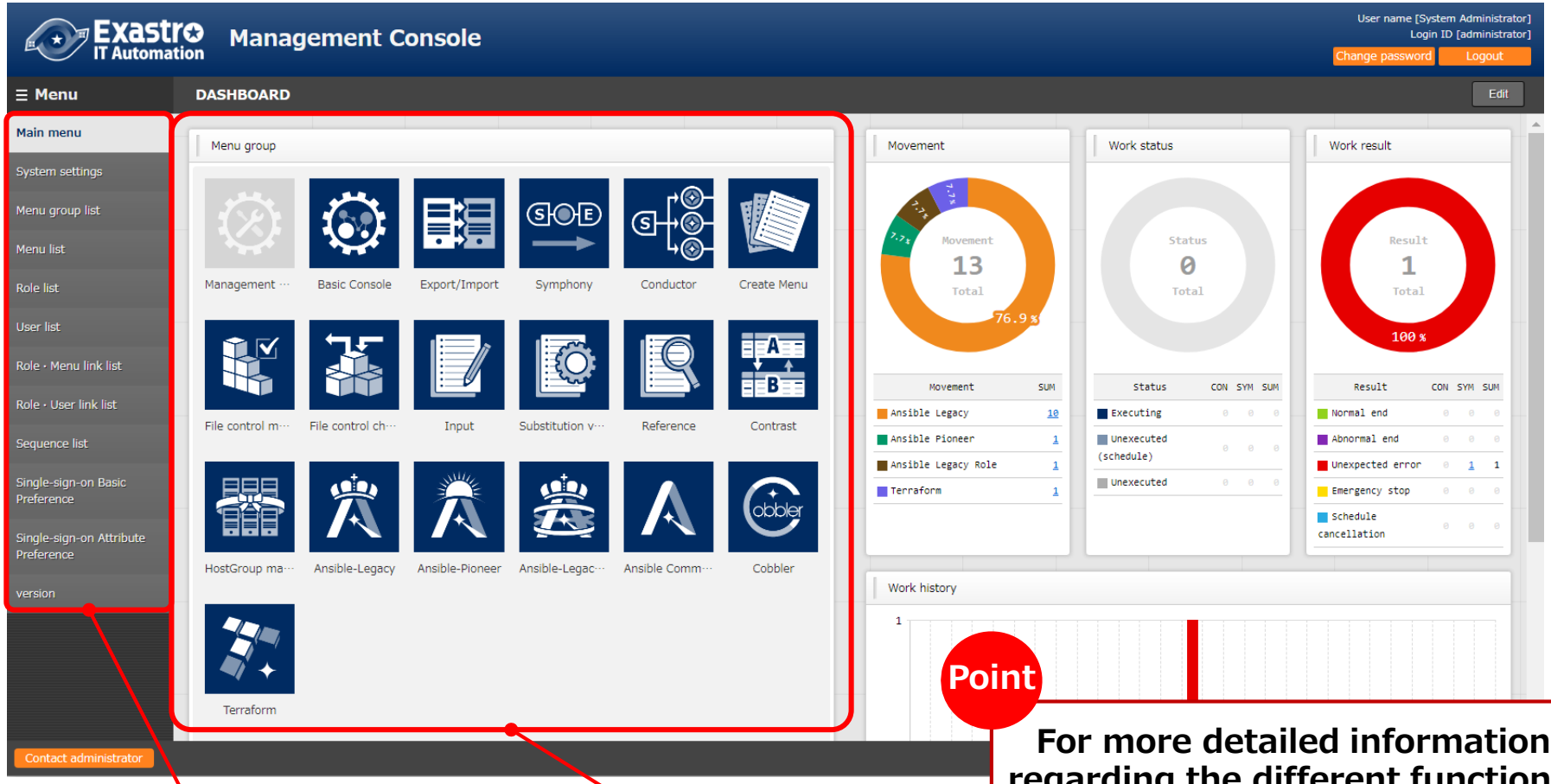


Point

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

Screen description (Main Menu)

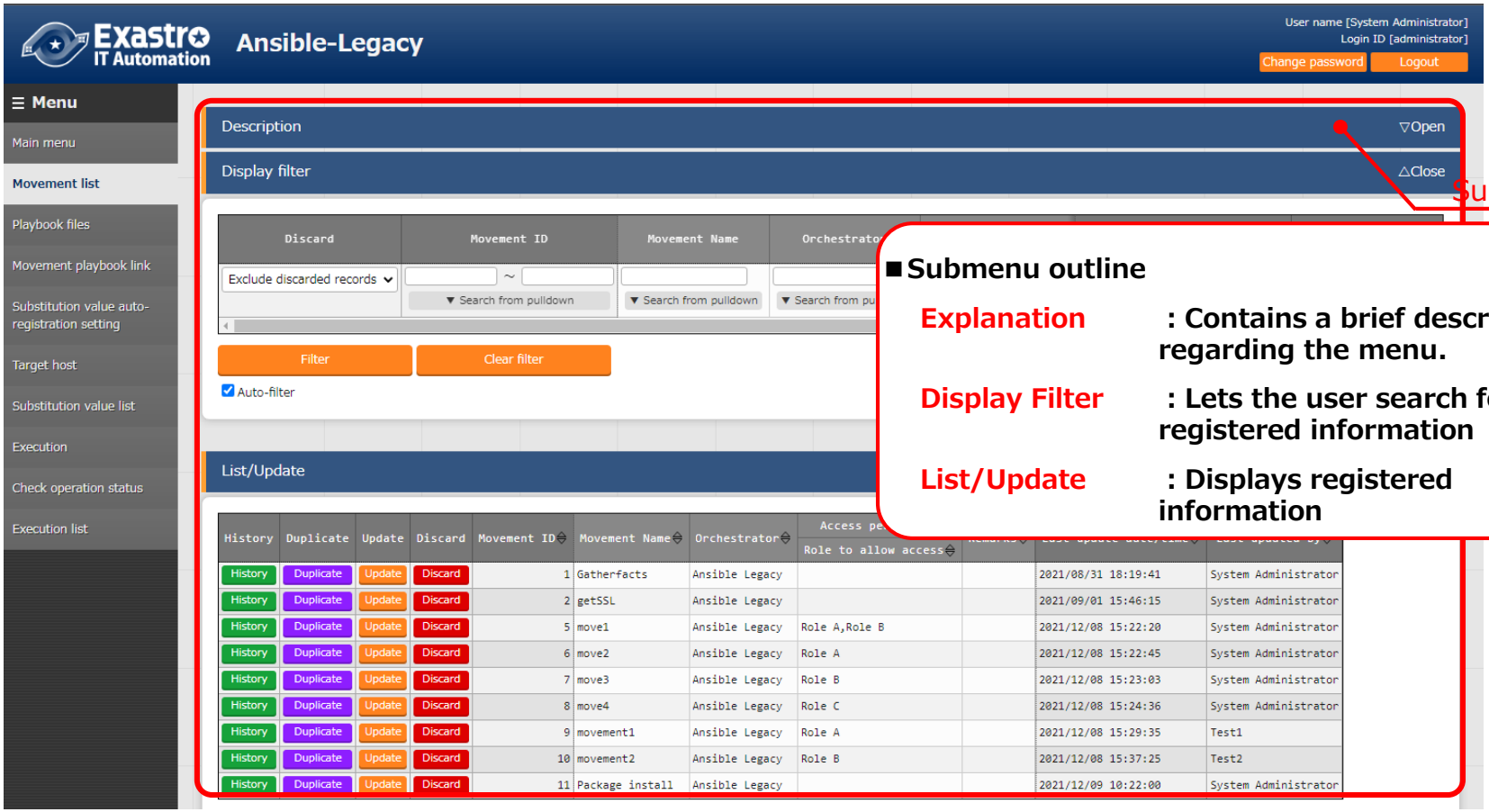
- The main functions of the Main Menu screen is as following



2.3 Screen Description (Menu) (1/2)

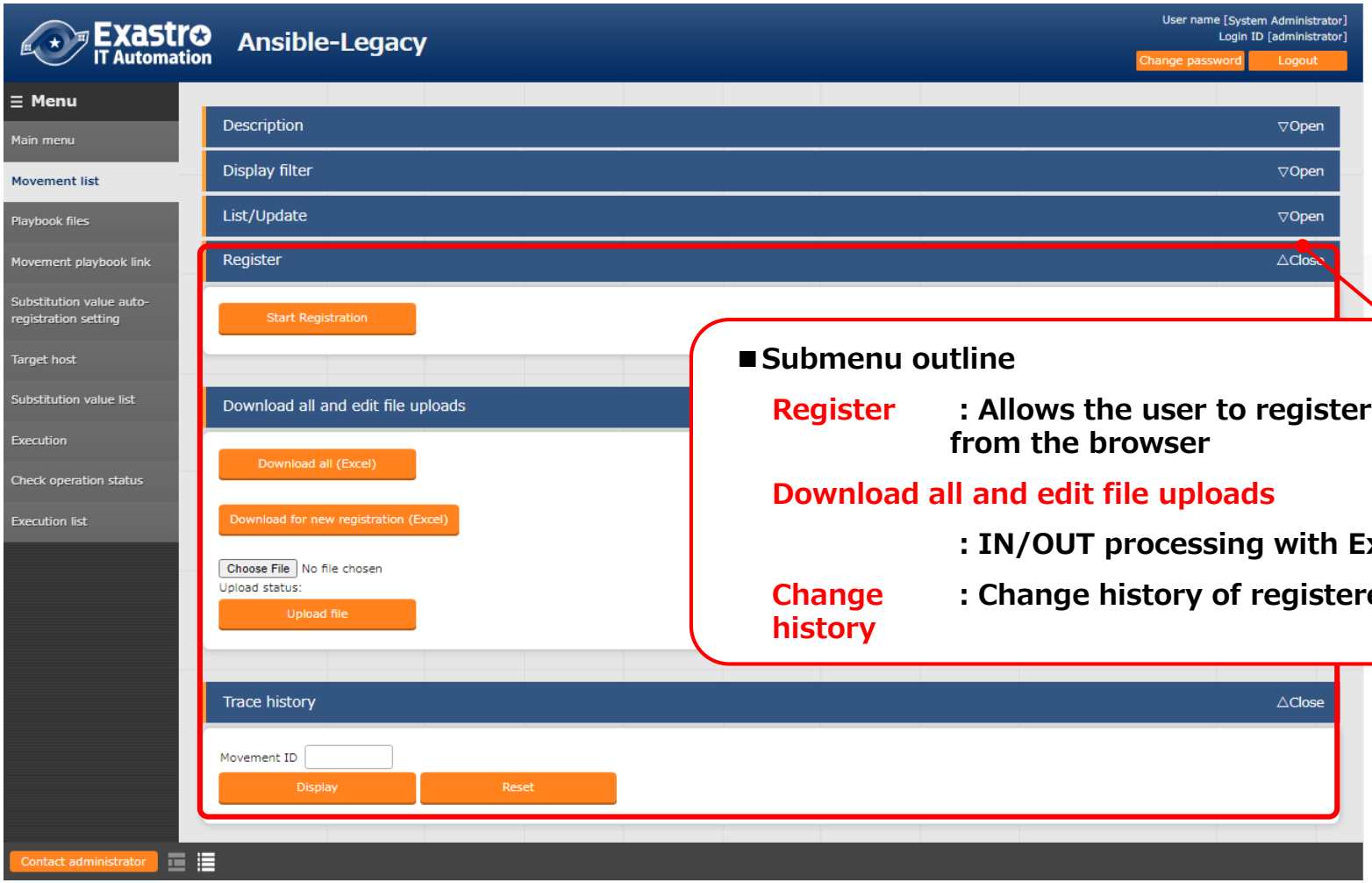
Screen Description (Menus)

- The name of the basic functions are as following.



Screen description (Menu)

- The name of the basic functions are as following



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.

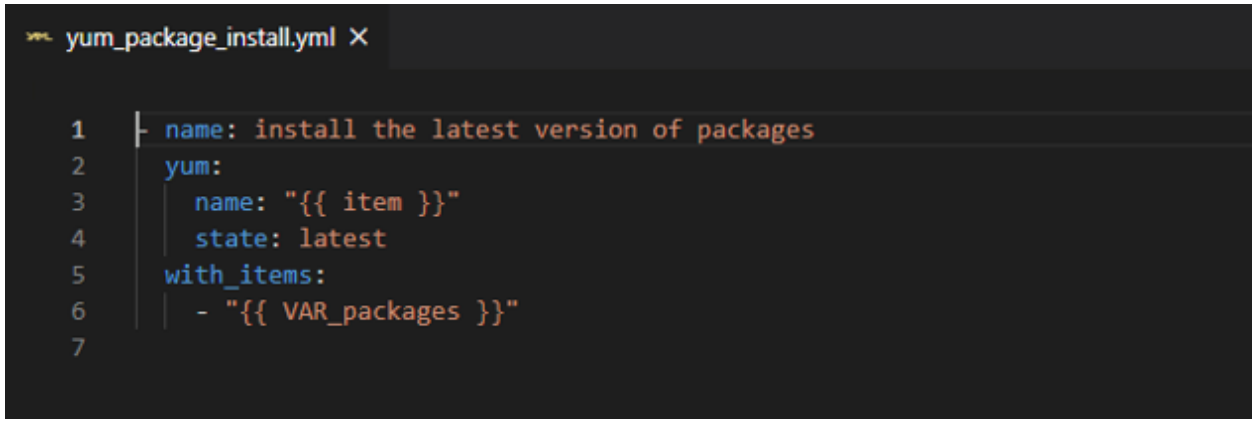
Use your desired editor program to create the following YML file and save it to your local hard drive.

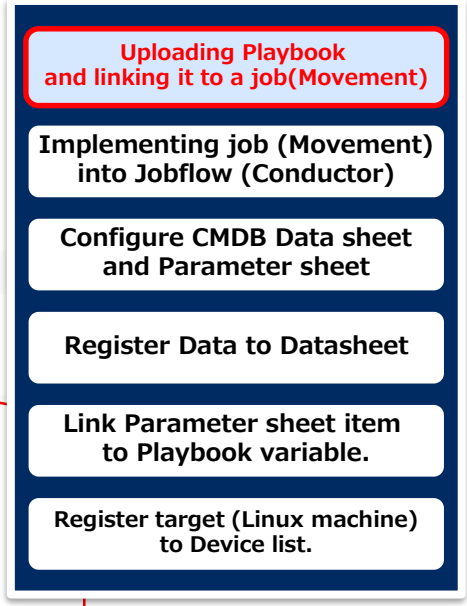
yum_package_install.yml

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

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.



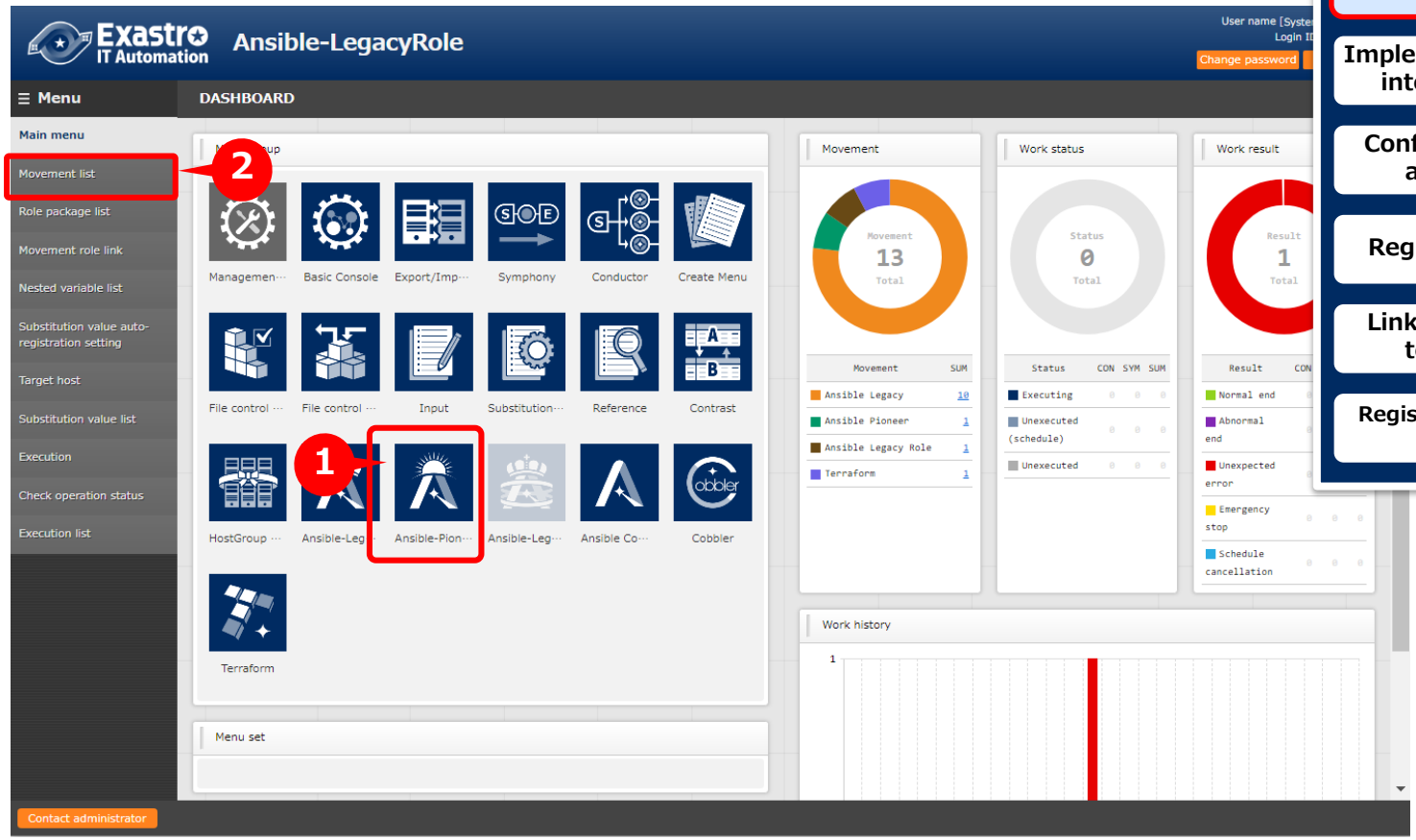


3.1 Uploading Playbook and linking it to a job(Movement) (2/5)

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.



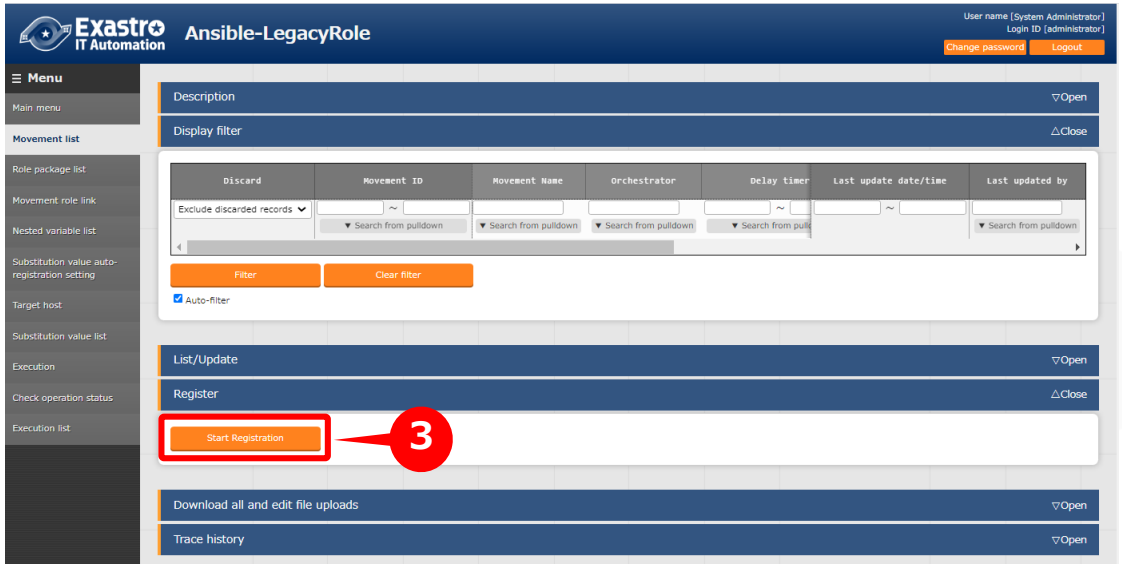
- Uploading Playbook and linking it to a job(Movement)
- Implementing job (Movement) into Jobflow (Conductor)
- Configure CMDB Data sheet and Parameter sheet
- Register Data to Datasheet
- Link Parameter sheet item to Playbook variable.
- Register target (Linux machine) to Device list.

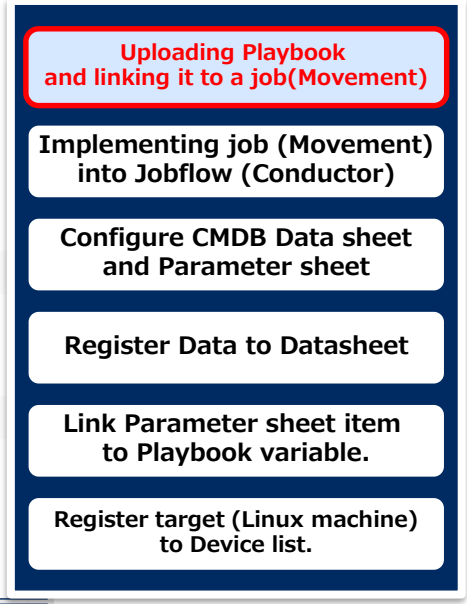
3.1 Uploading Playbook and linking it to a job(Movement) (3/5)

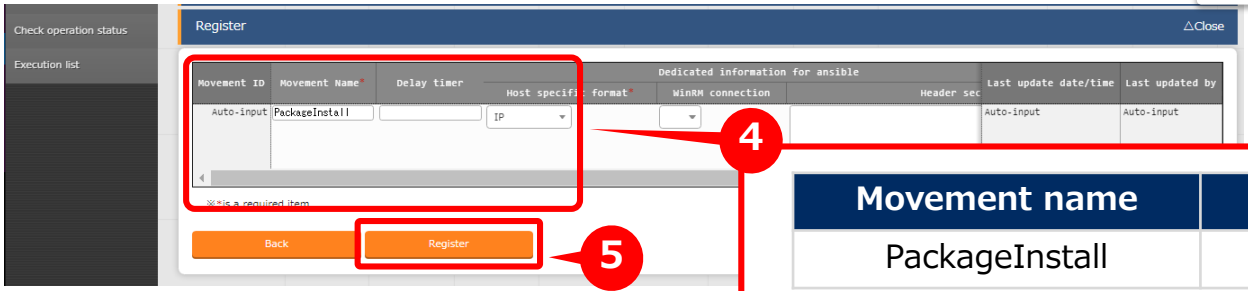
Register Movement to the Movement list.

- Click the “Start Registration” button.

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







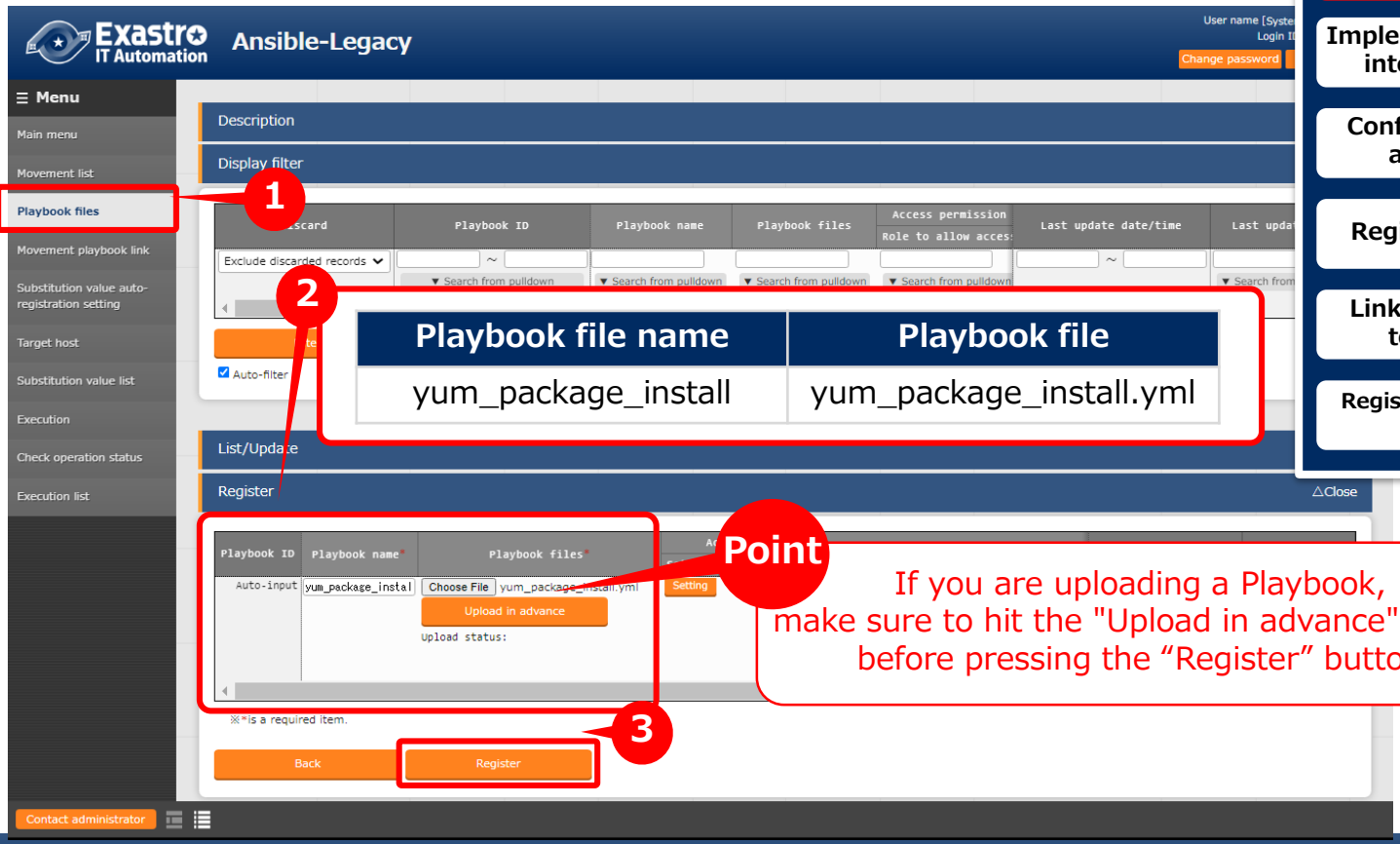
Movement name	Host format
PackageInstall	IP

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.



Playbook file name	Playbook file
yum_package_install	yum_package_install.yml

Point

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

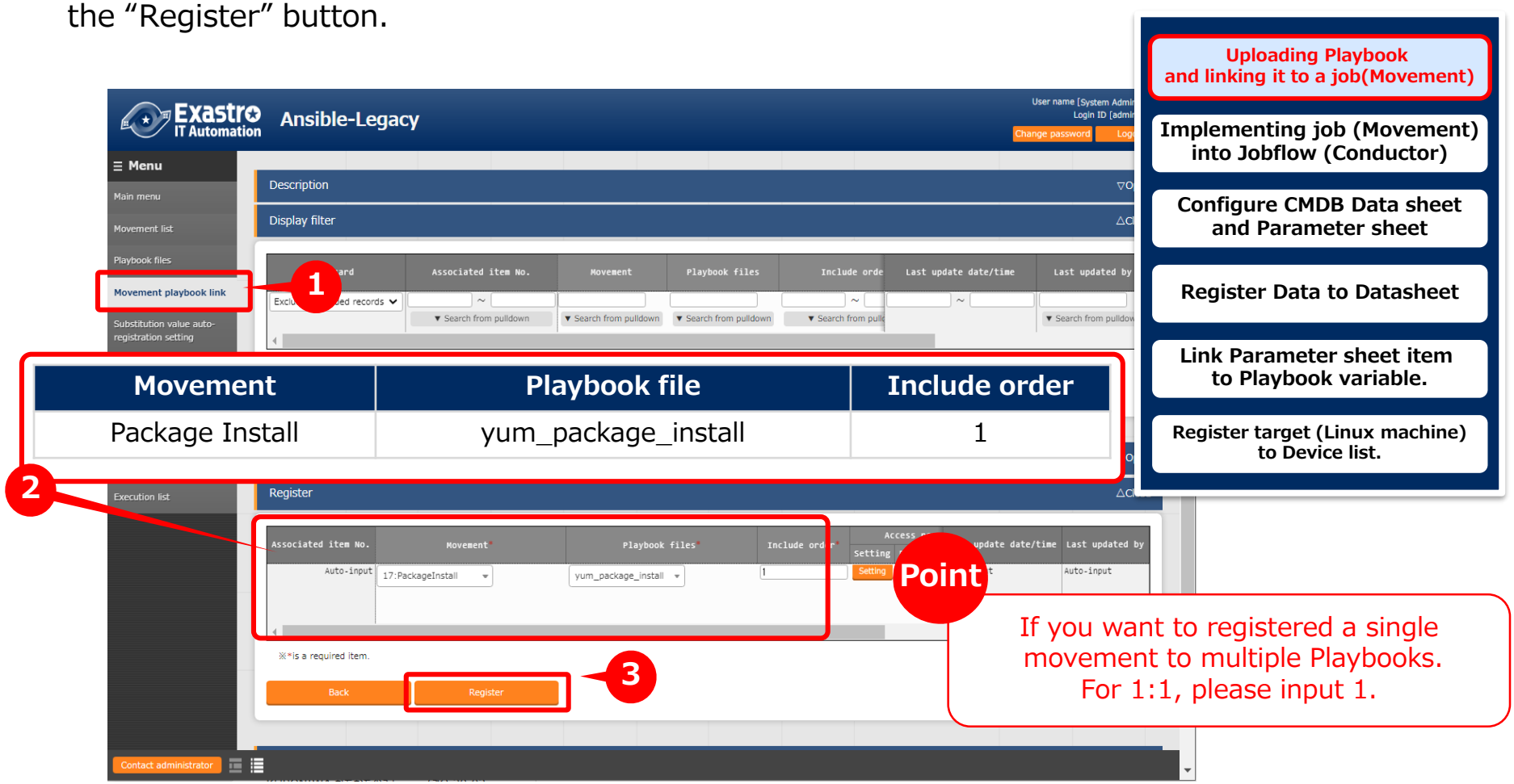
- **Uploading Playbook and linking it to a job(Movement)**
- Implementing job (Movement) into Jobflow (Conductor)
- Configure CMDB Data sheet and Parameter sheet
- Register Data to Datasheet
- Link Parameter sheet item to Playbook variable.
- Register target (Linux machine) to Device list.

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.



Movement	Playbook file	Include order
Package Install	yum_package_install	1

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

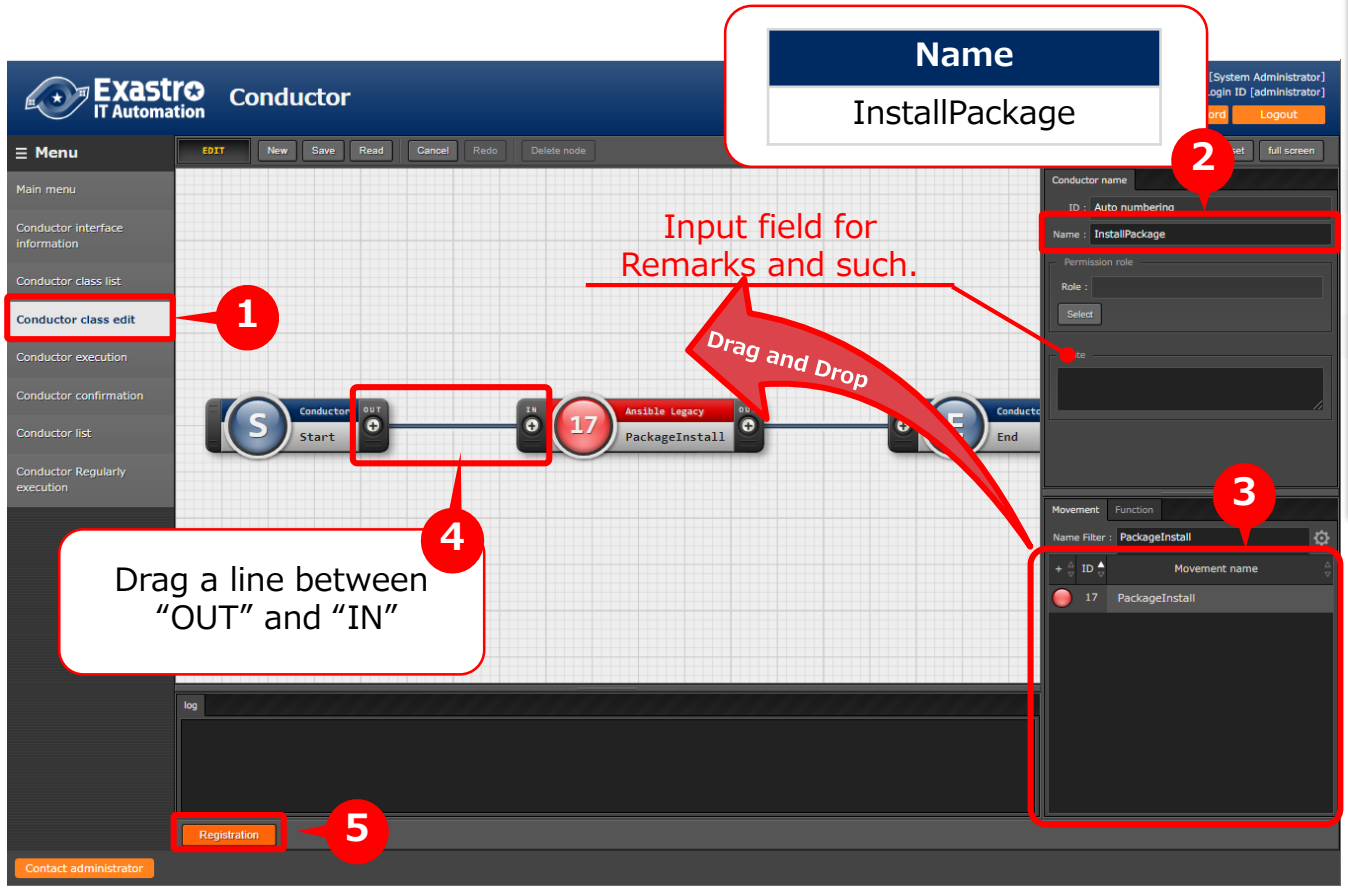
3.2 Implementing job (Movement)into Jobflow (Conductor)

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.

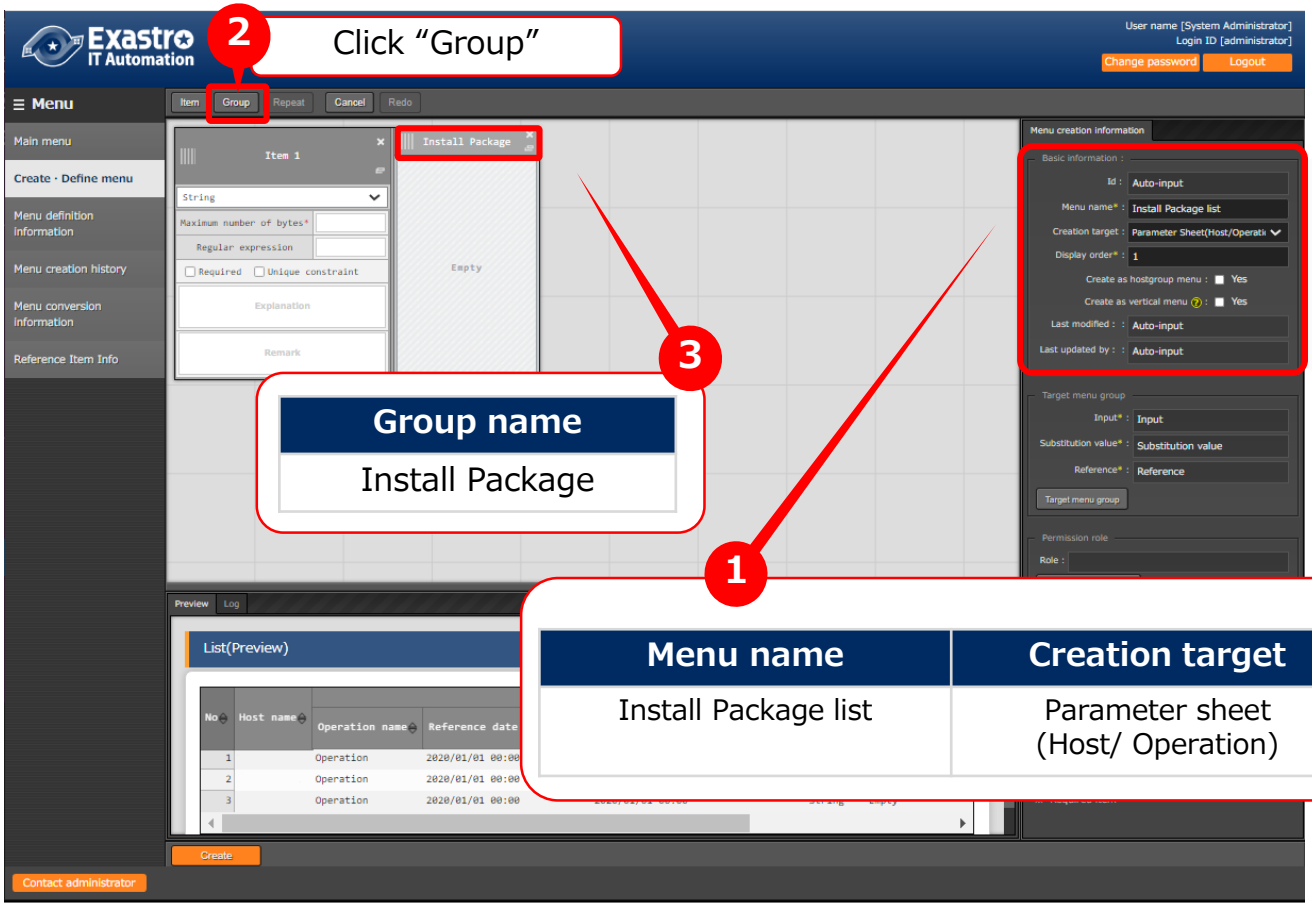
3.3 Configure CMDB Data sheet and Parameter sheet (1/3)

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.



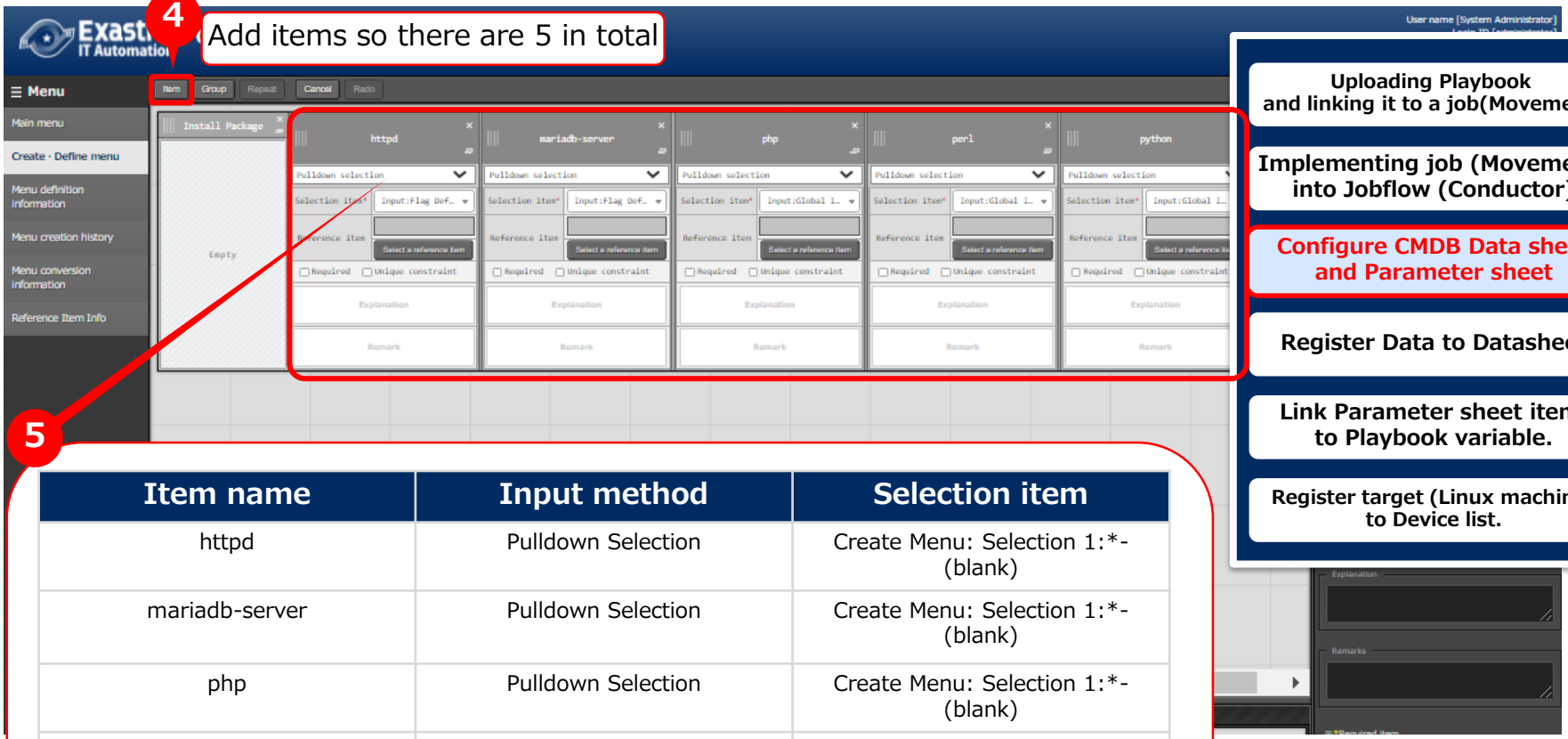
Menu name	Creation target	Display Order
Install Package list	Parameter sheet (Host/ Operation)	1

- 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

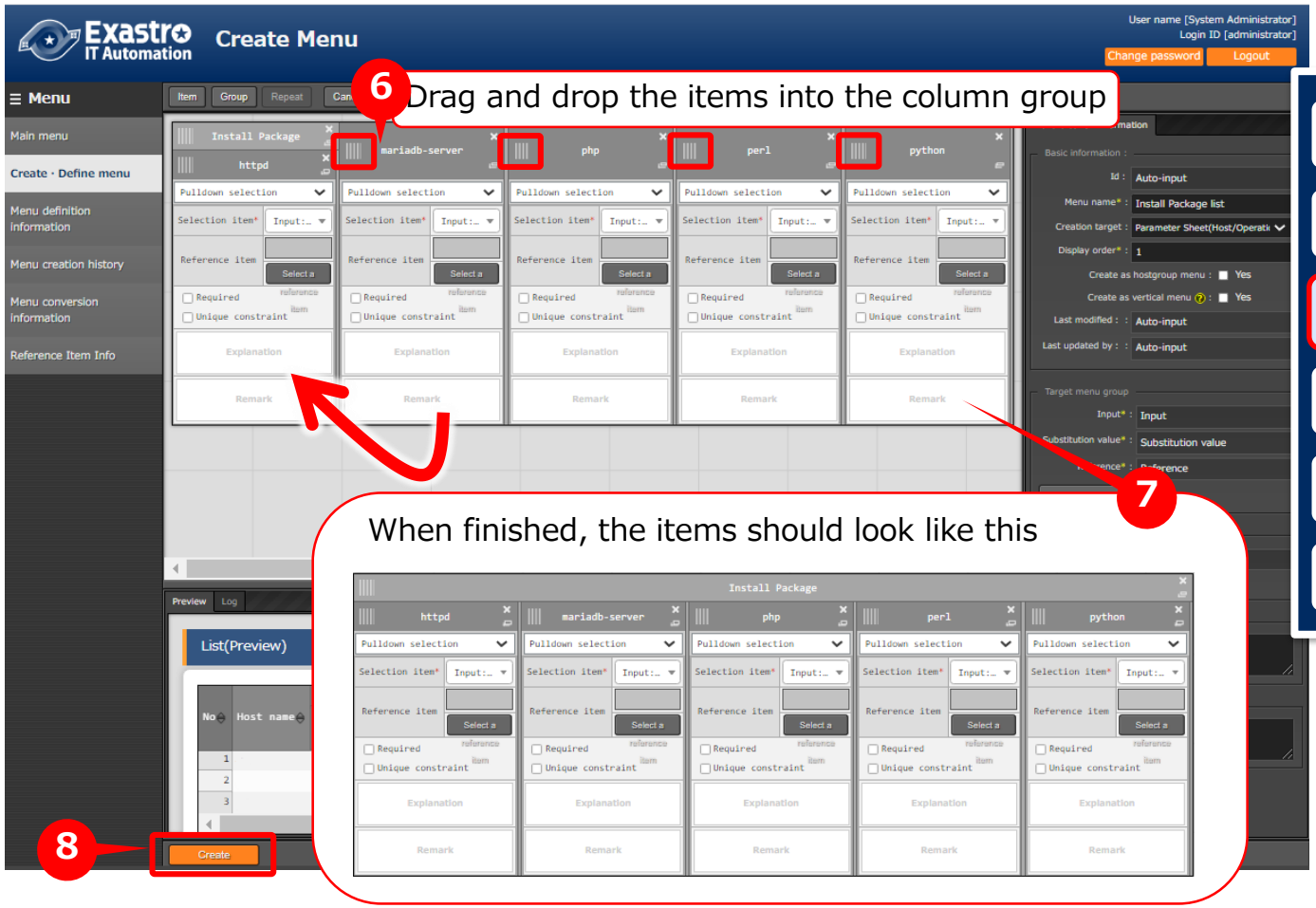
- Add items and fill the items with the values written in the table below.



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)

Create Parameter sheet

- After following the steps below, 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.

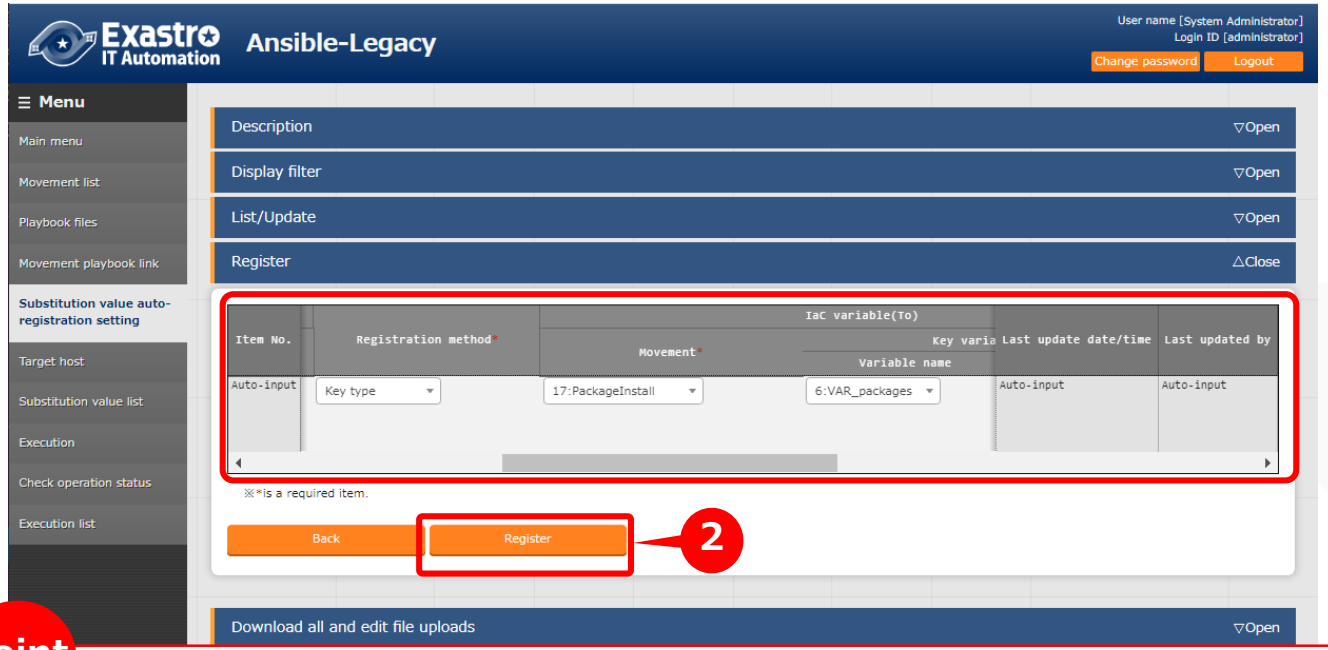
Menu group:Menu	Item	Registration method	Movement	Key Variable Variable name	Substitution order
2100011611:Susbtituti on value:3:Install Package list	Parameter/Install Package/httpd	Key type	1:PackageInstall	1:VAR_packages	1
2100011611:Susbtituti on value:3:Install Package list	Parameter/Install Package/mariadb-server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

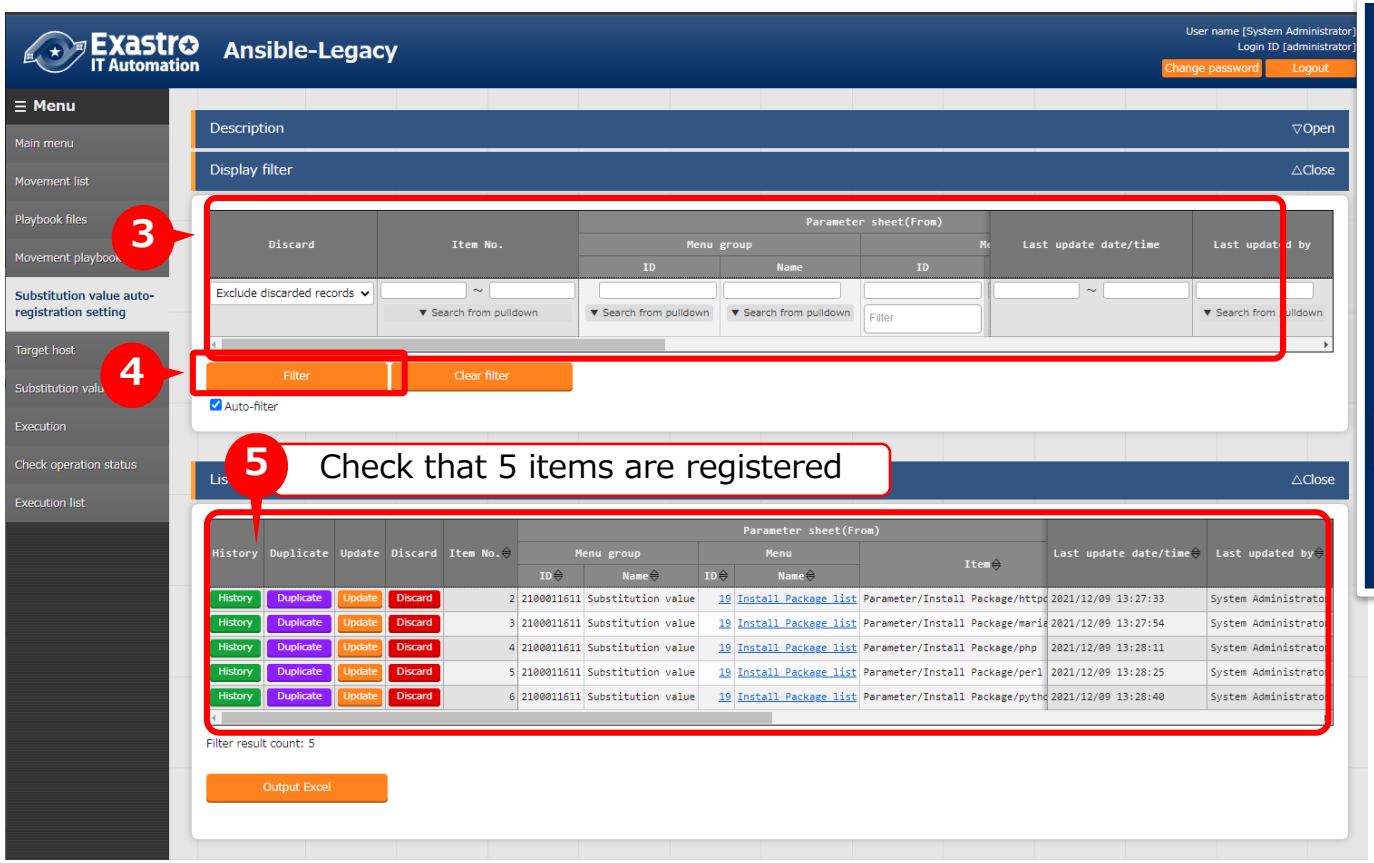
In this scenario, we want to assign the table items (column name) to the Playbook as a specific value, so we will choose the "Key" registration method.
For more information, please see the Exastro_System_Operation_and_Construction_Efficiency_Guide

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.



- 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.

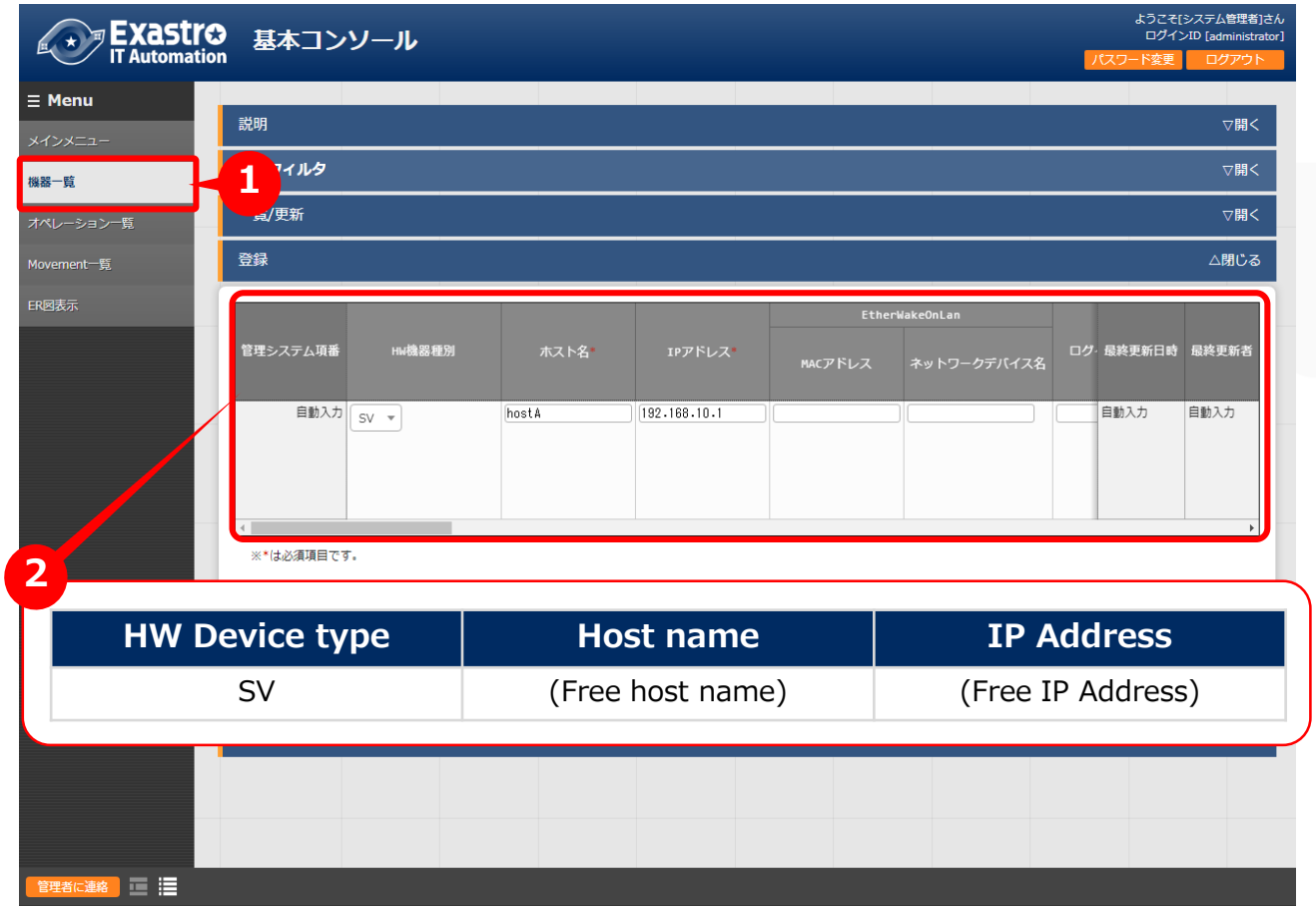
4. Execution (First time)

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.



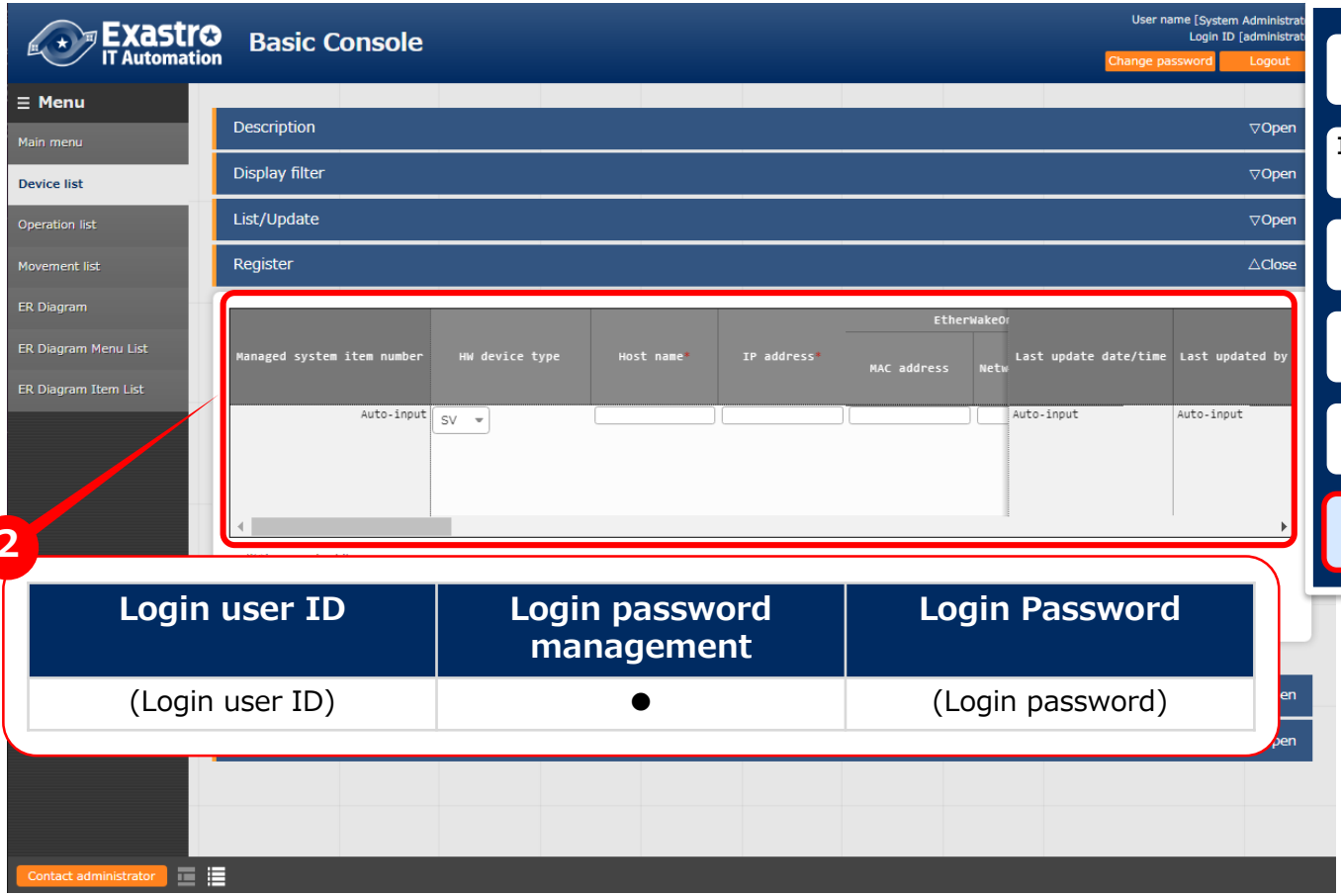
HW Device type	Host name	IP Address
SV	(Free host name)	(Free IP Address)

- 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.



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

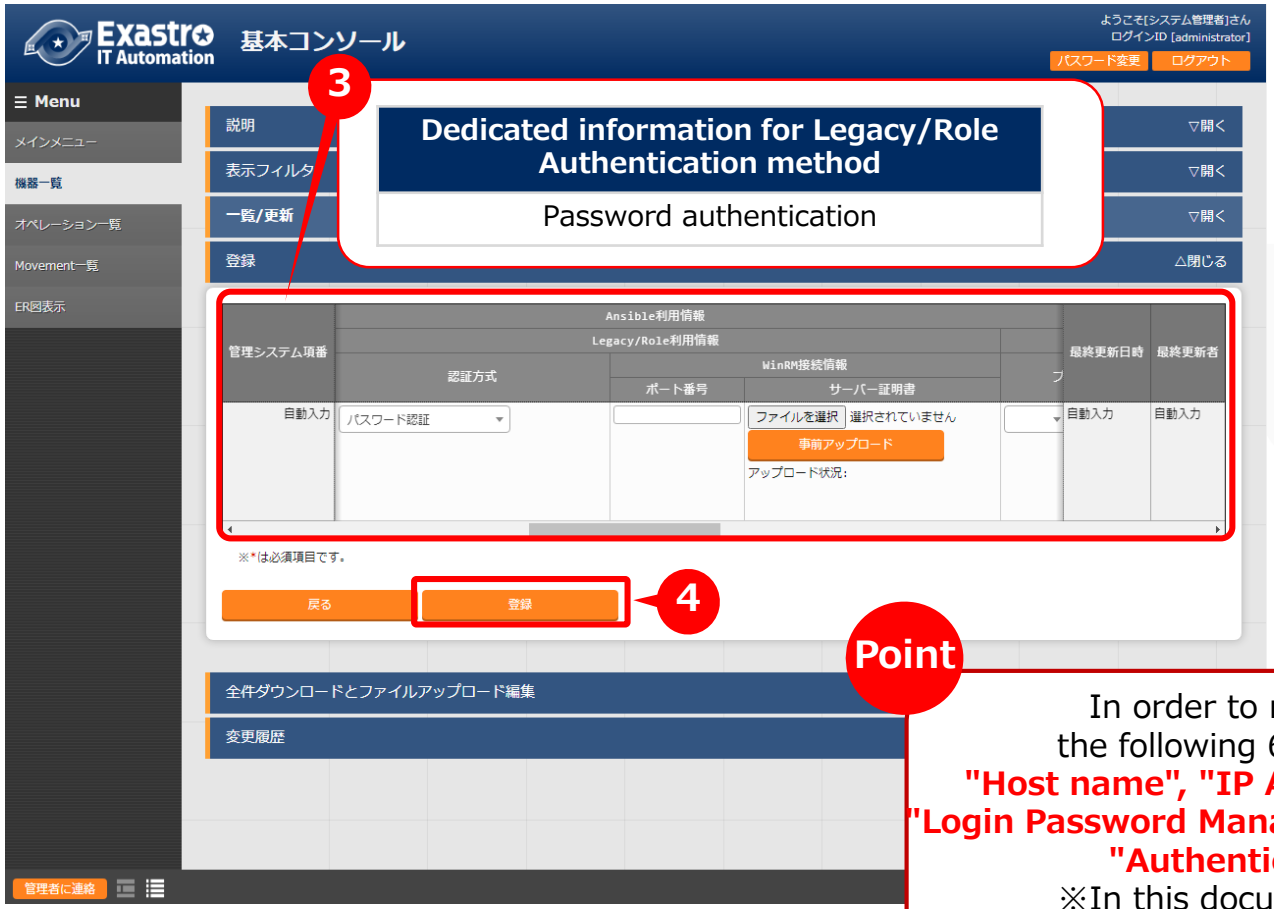
Link Parameter sheet item to Playbook variable.

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.



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

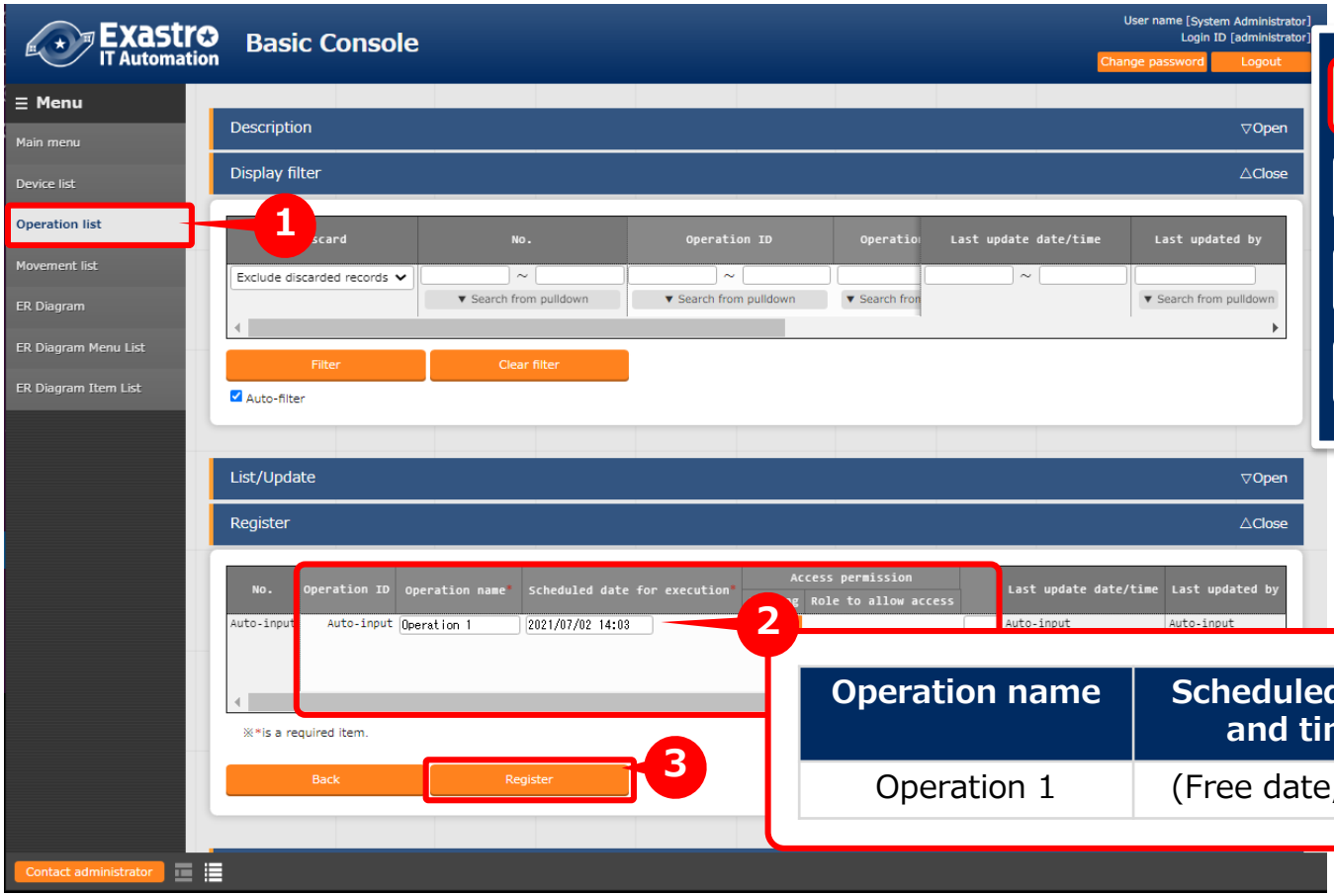
In order to run Ansible-Legacy,
the following 6 items must be filled.
"Host name", "IP Address", "Login User ID", "Login Password Management", "Login password", "Authentication method※"
※In this document, "Authentication method" is written as "Password authentication".

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.



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

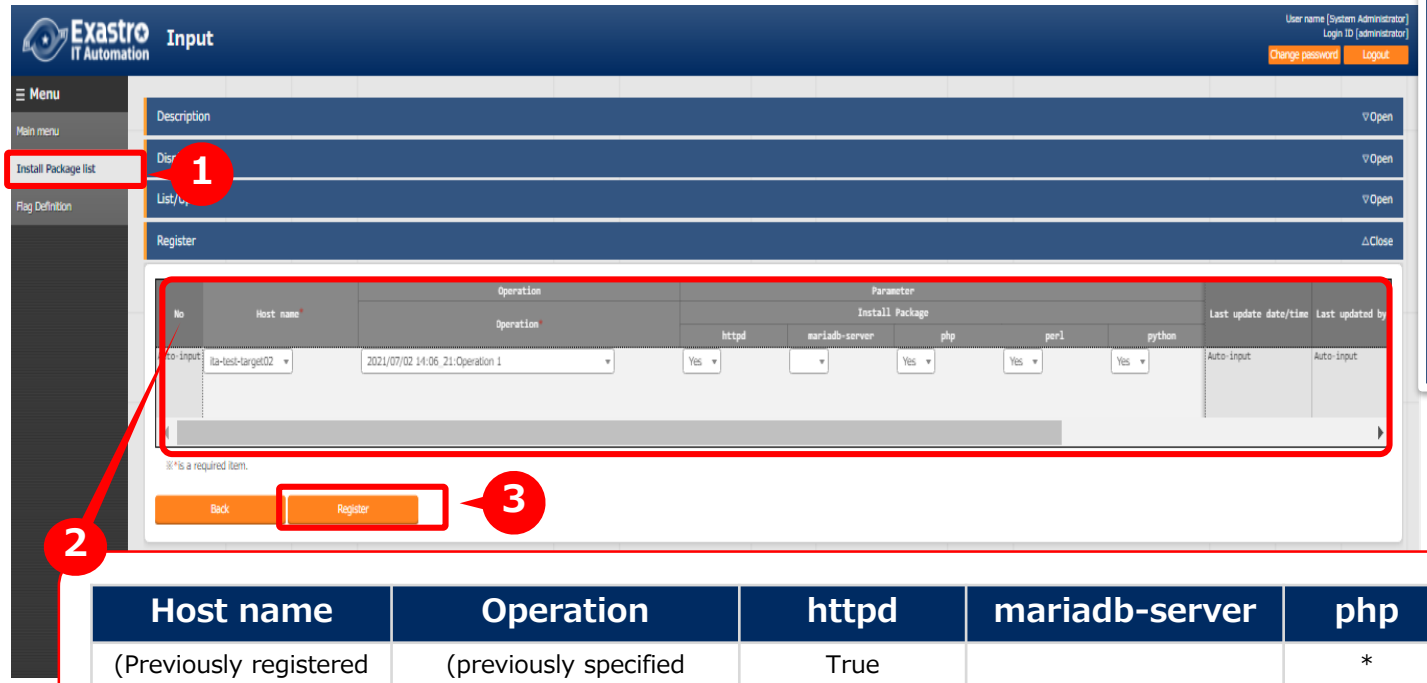
4.3 Register data to Parameter sheet (1/2)

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.



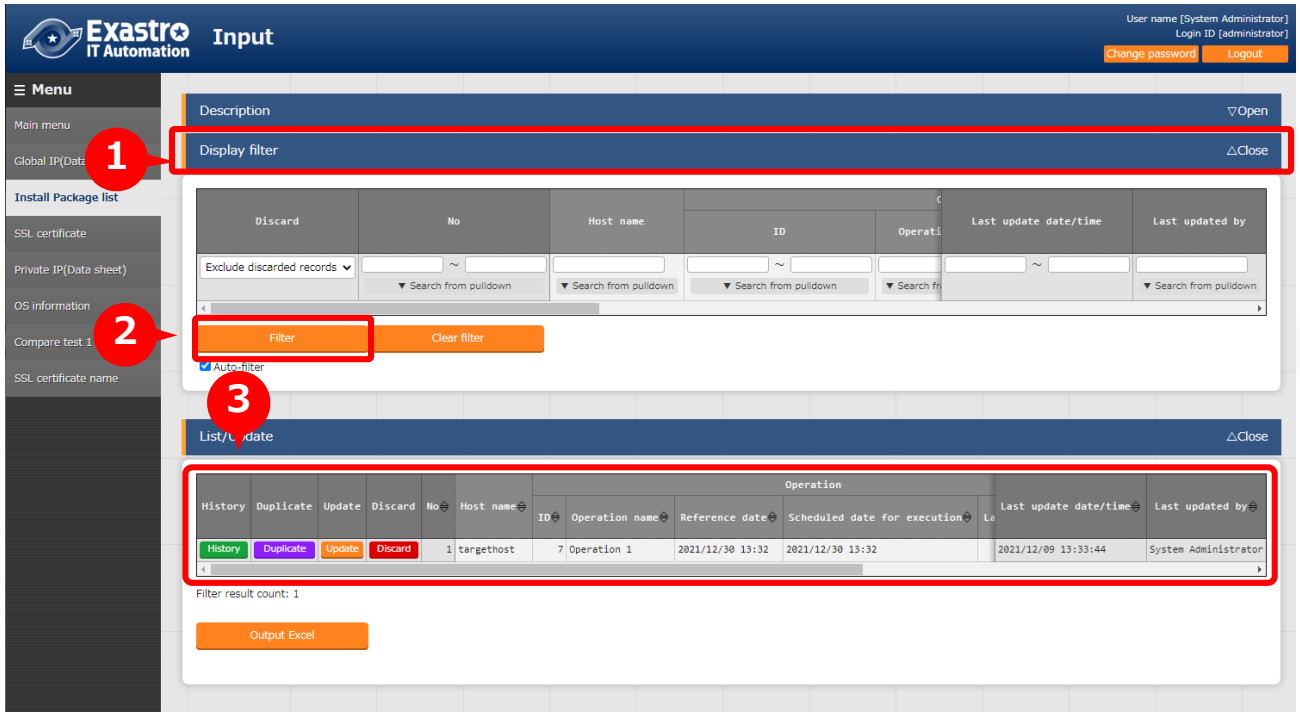
- Register Operation name (Operation)
- **Register data to Parameter sheet**
- Execute Jobflow (Conductor)
- Check Execution results

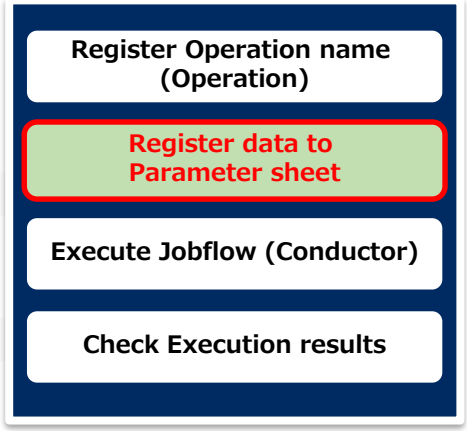
Host name	Operation	httpd	mariadb-server	php	perl	python
(Previously registered host)	(previously specified date)_1:Operation 1	True		*	*	*

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.





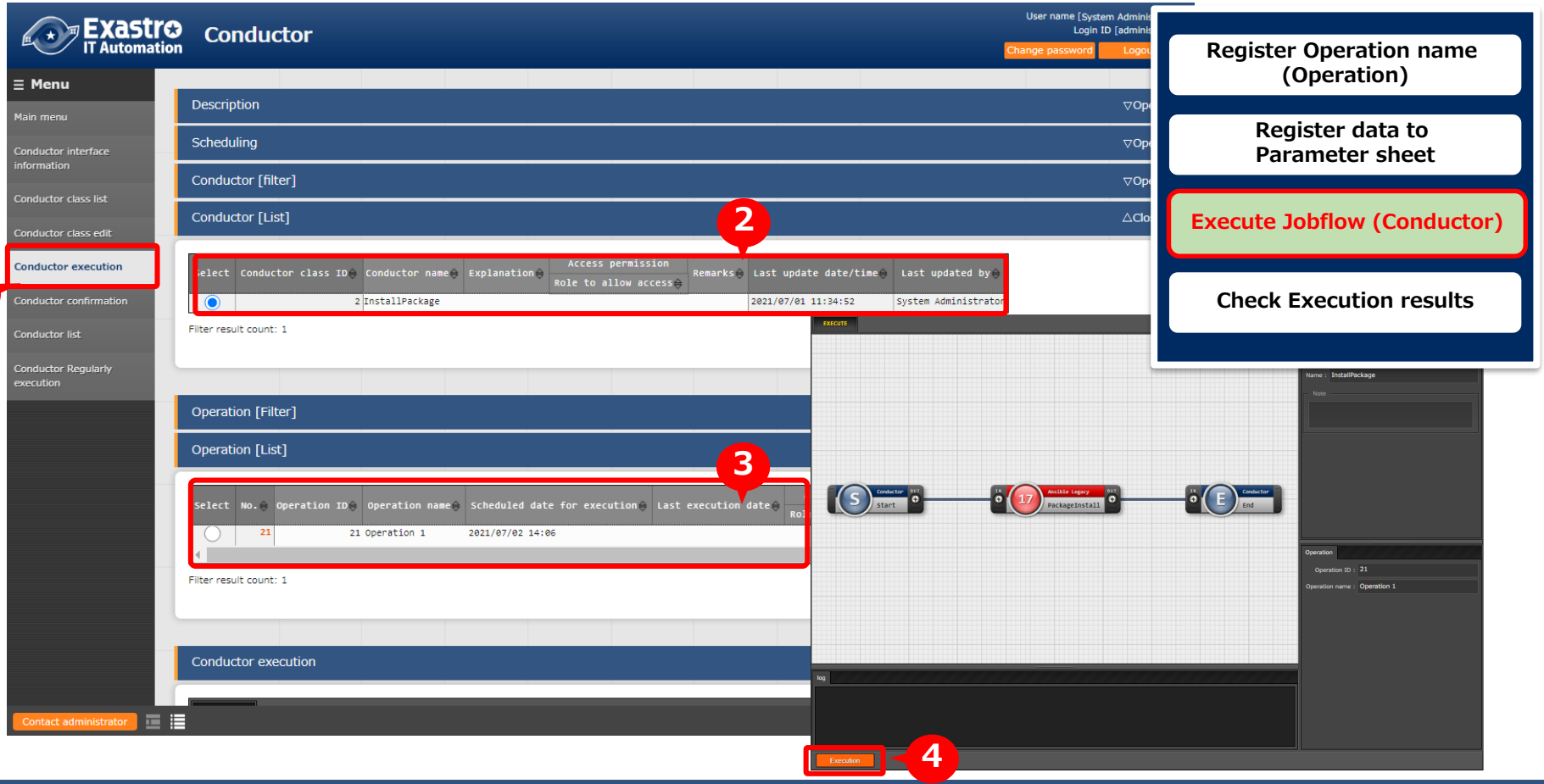
4.4 Execute Jobflow (Conductor) (1/3)

Run Conductor

- We will now start the Conductor.

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

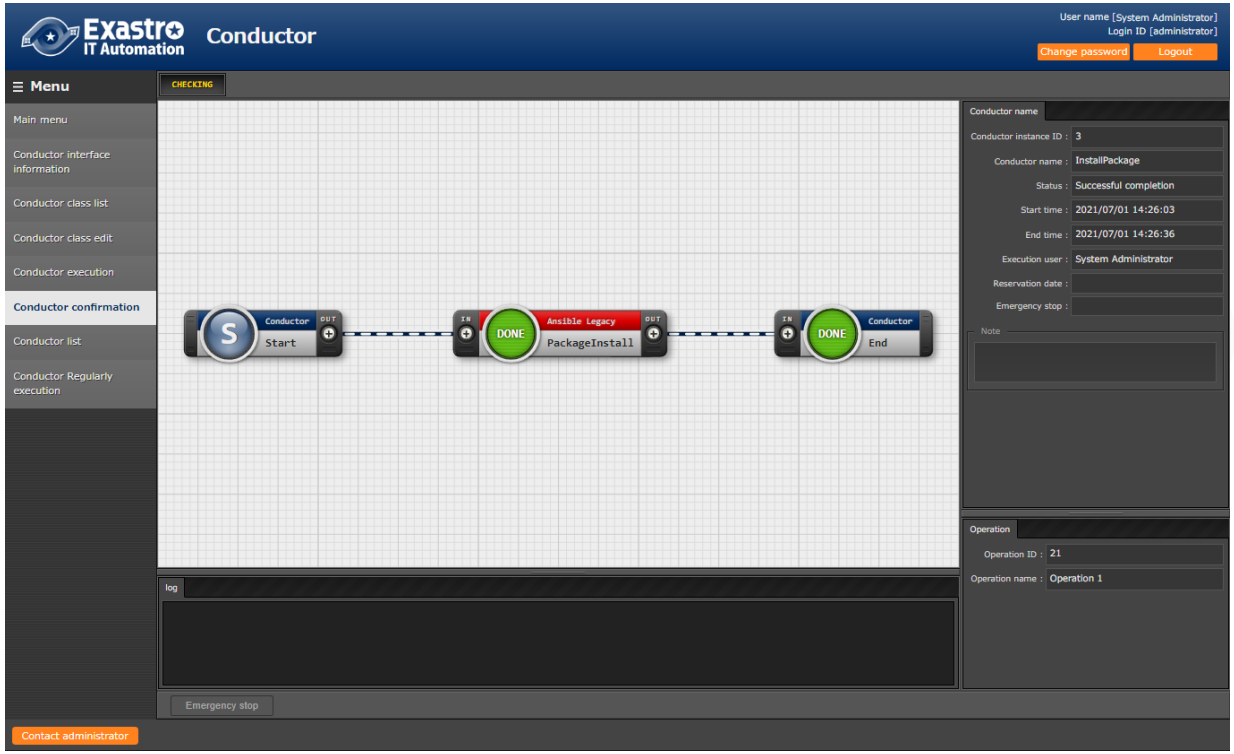
Next, select "Conductor" and "Operation" and press "Execute.

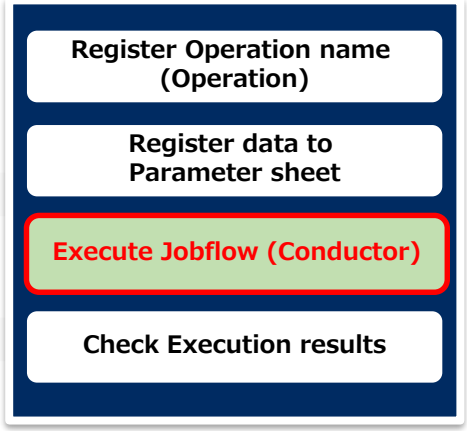


4.4 Execute Jobflow (Conductor) (2/3)

Execution results

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





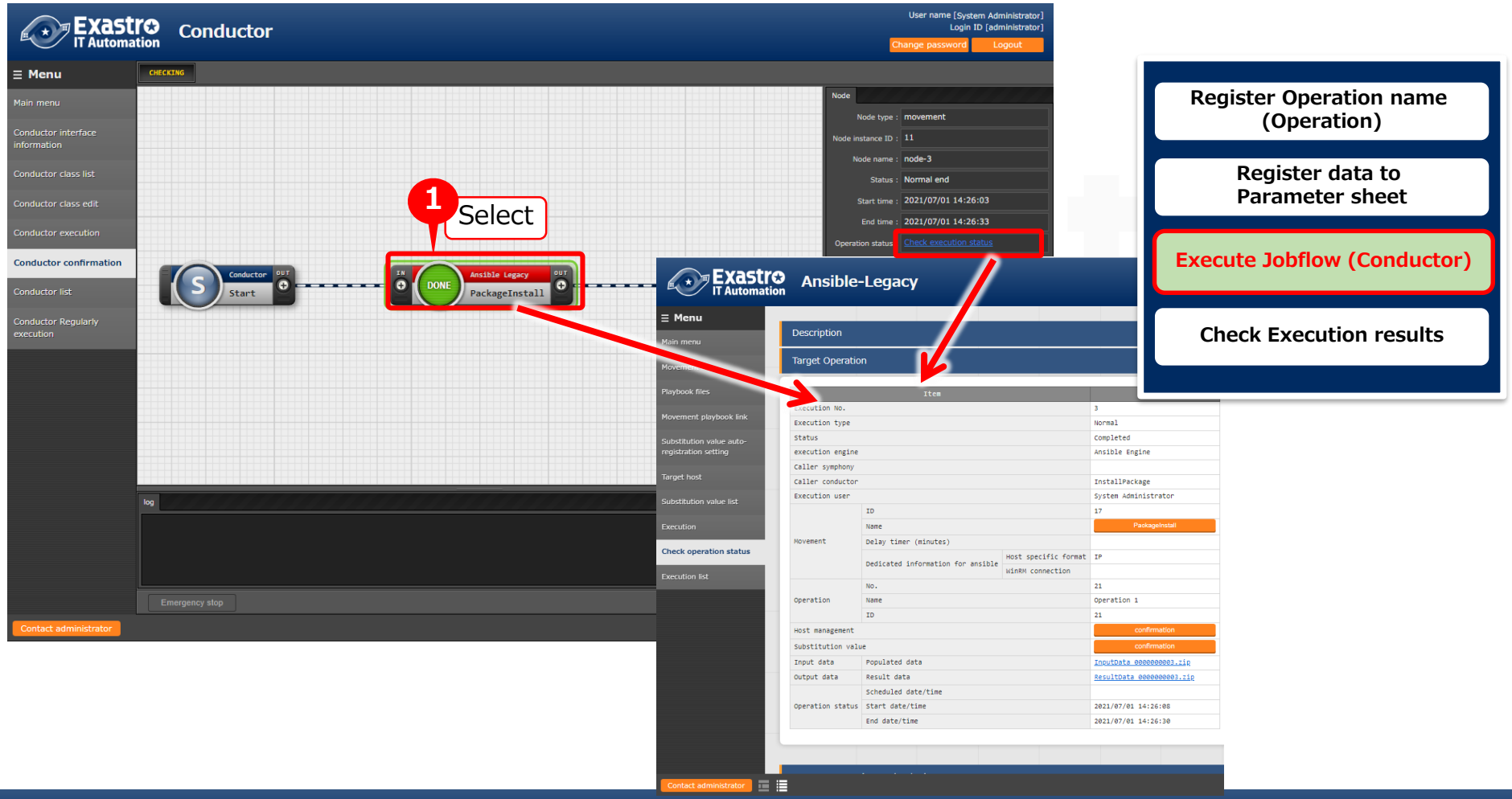
Point

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

4.4 Execute Jobflow (Conductor) (3/3)

Execution results

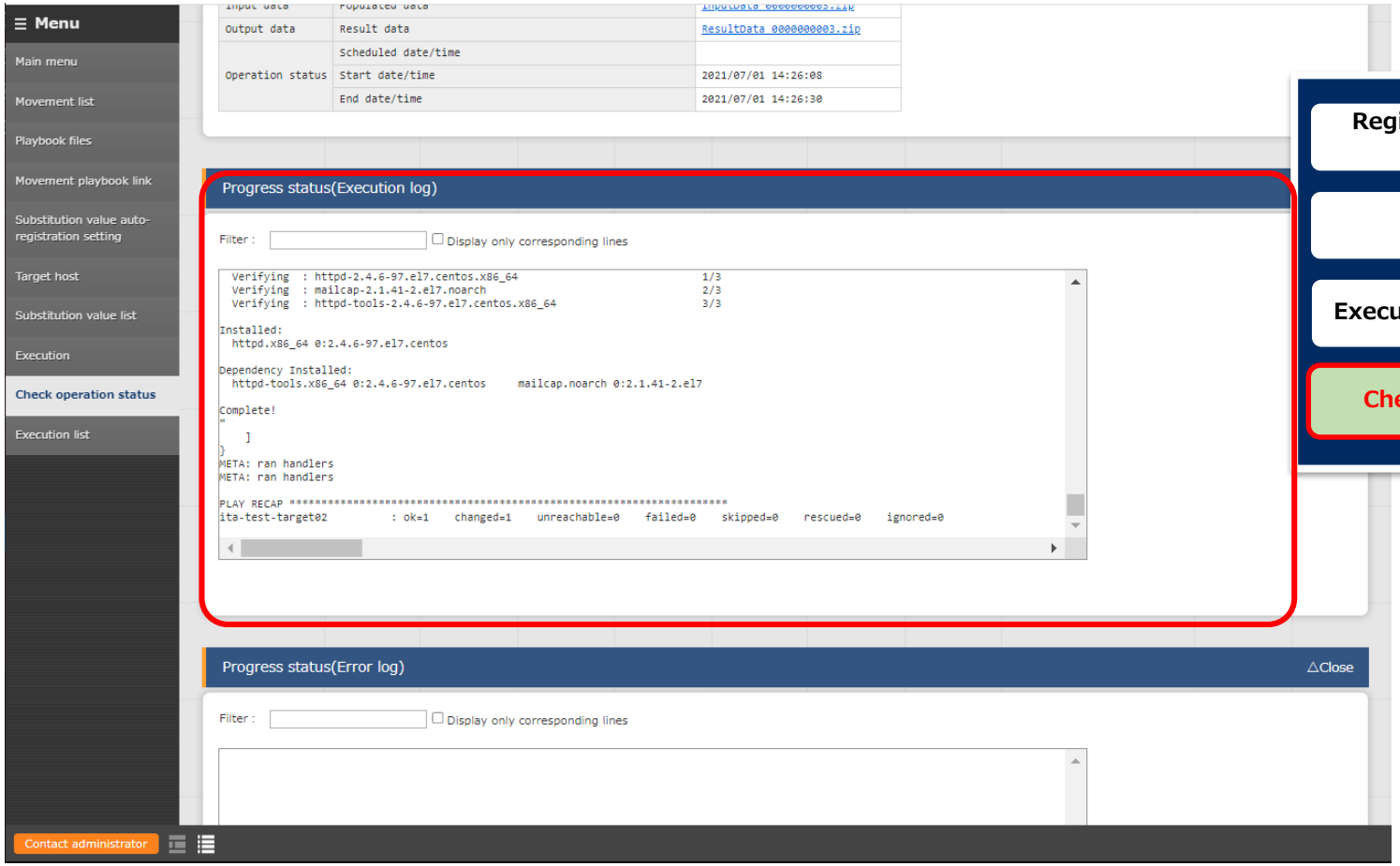
- Select a job (Movement) and press either the "Done" icon or the Operation status on the right to see more details.



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.



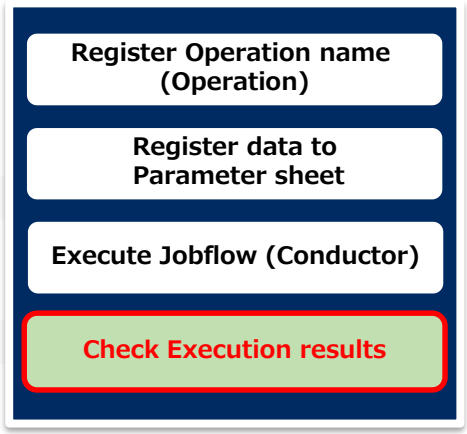
4.5 Check Execution results (2/3)

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!
"]}
```



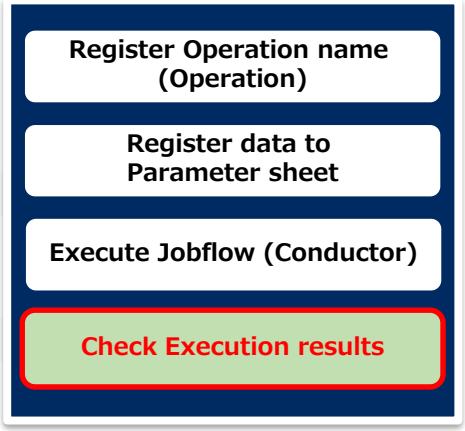
4.5 Check Execution results (3/3)

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
```

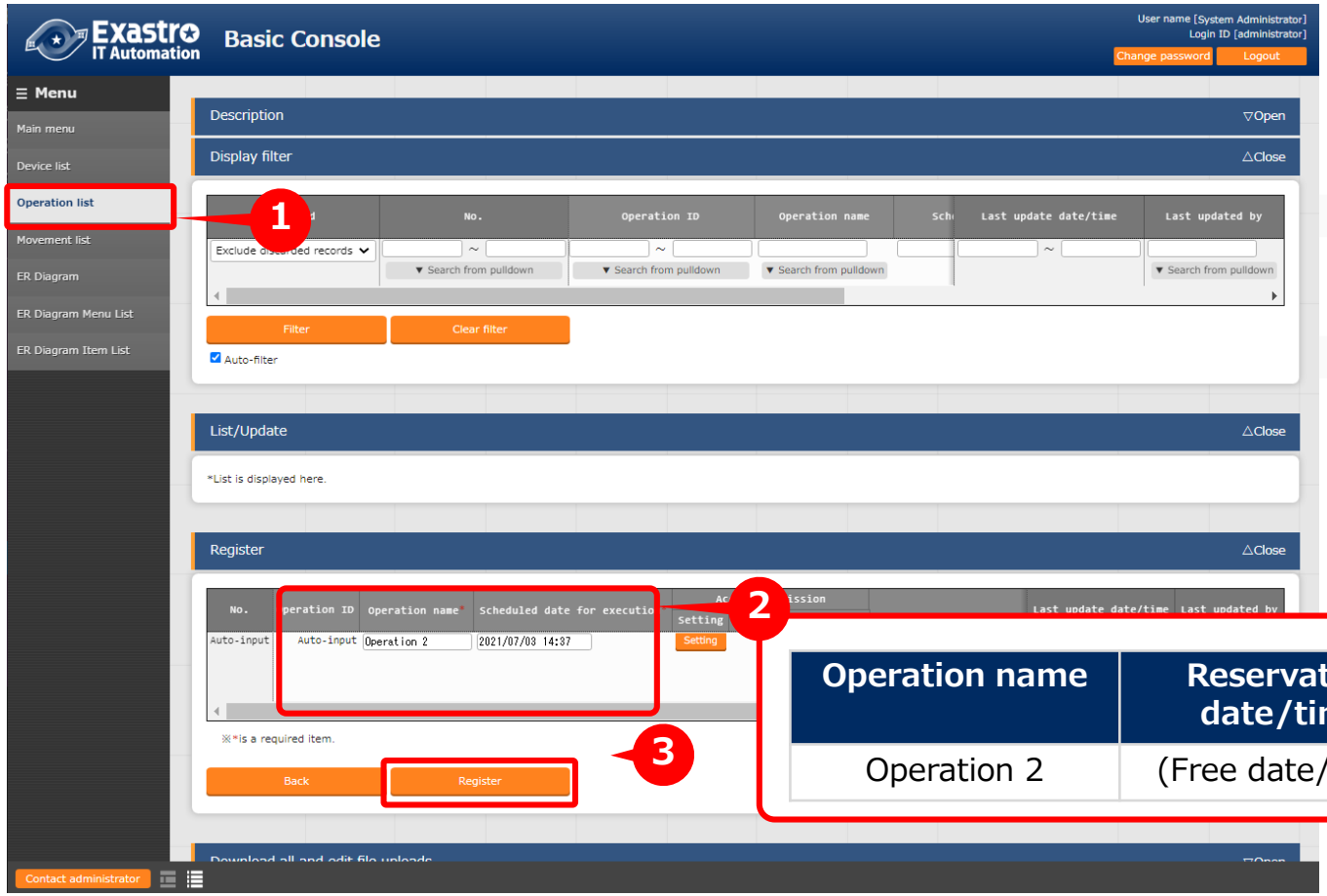


5. Execution (Second time)

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.



Operation name	Reservation date/time
Operation 2	(Free date/time)

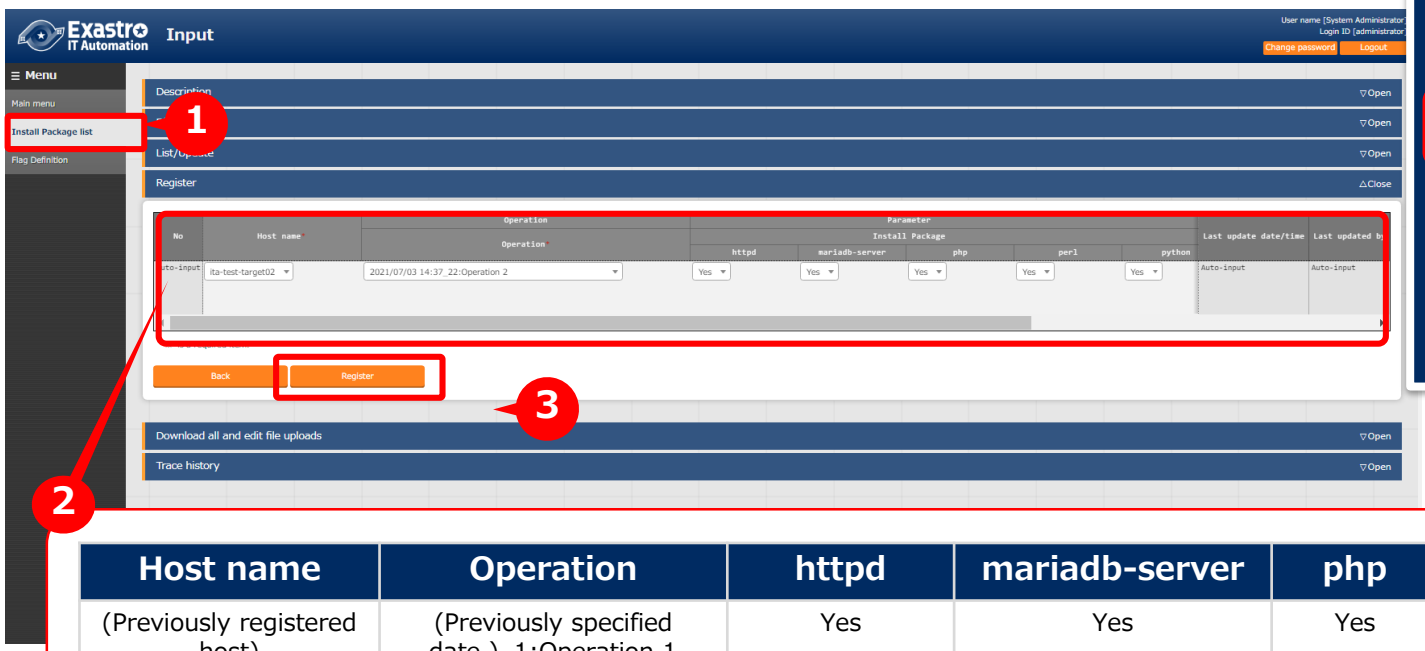
- **Register Operation name (Operation)**
- Register data to Parameter sheet
- Execute Jobflow (Conductor)
- Check Execution results

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.

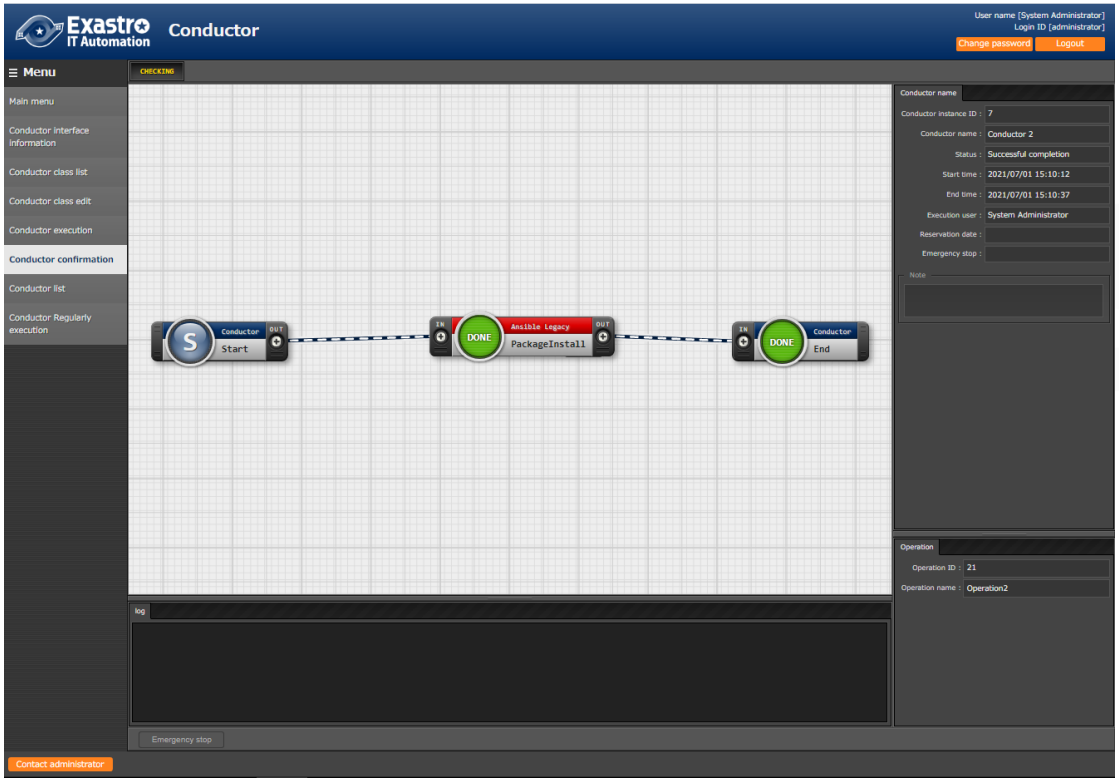
From the “Conductor” Menu group, go to the “Conductor execution” menu.

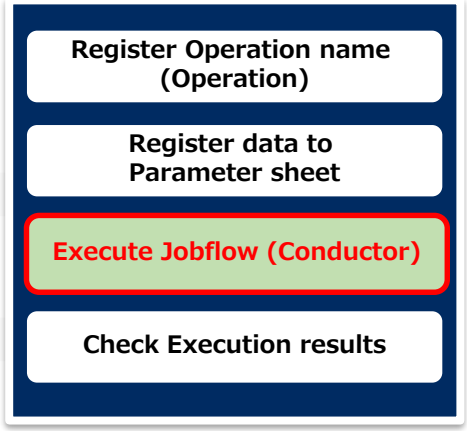
Select the Conductor and Operation you want to run and press “Execute”.

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.





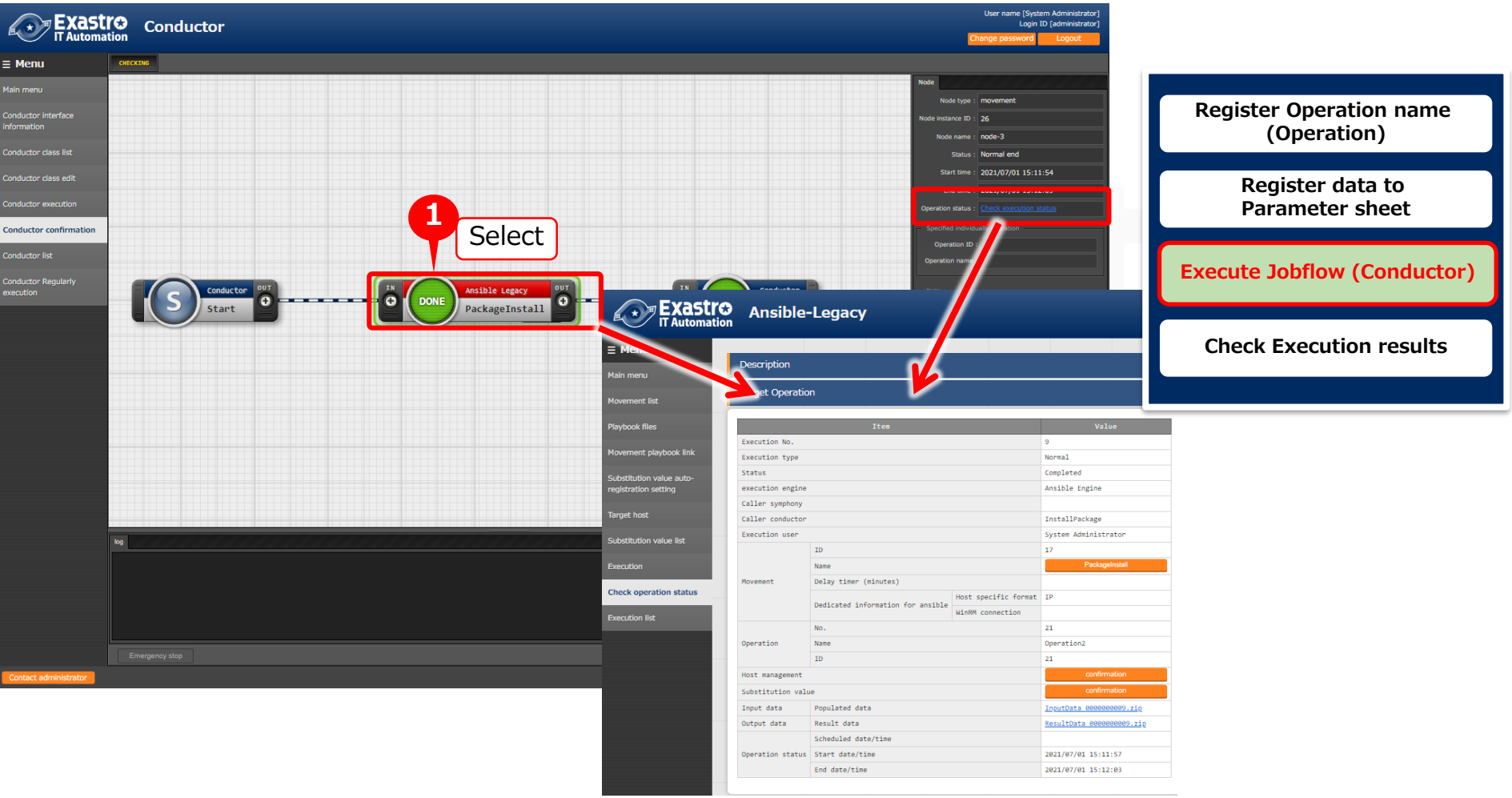
Point

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

5.3 Execute Jobflow (Conductor) (3/3)

Execution results

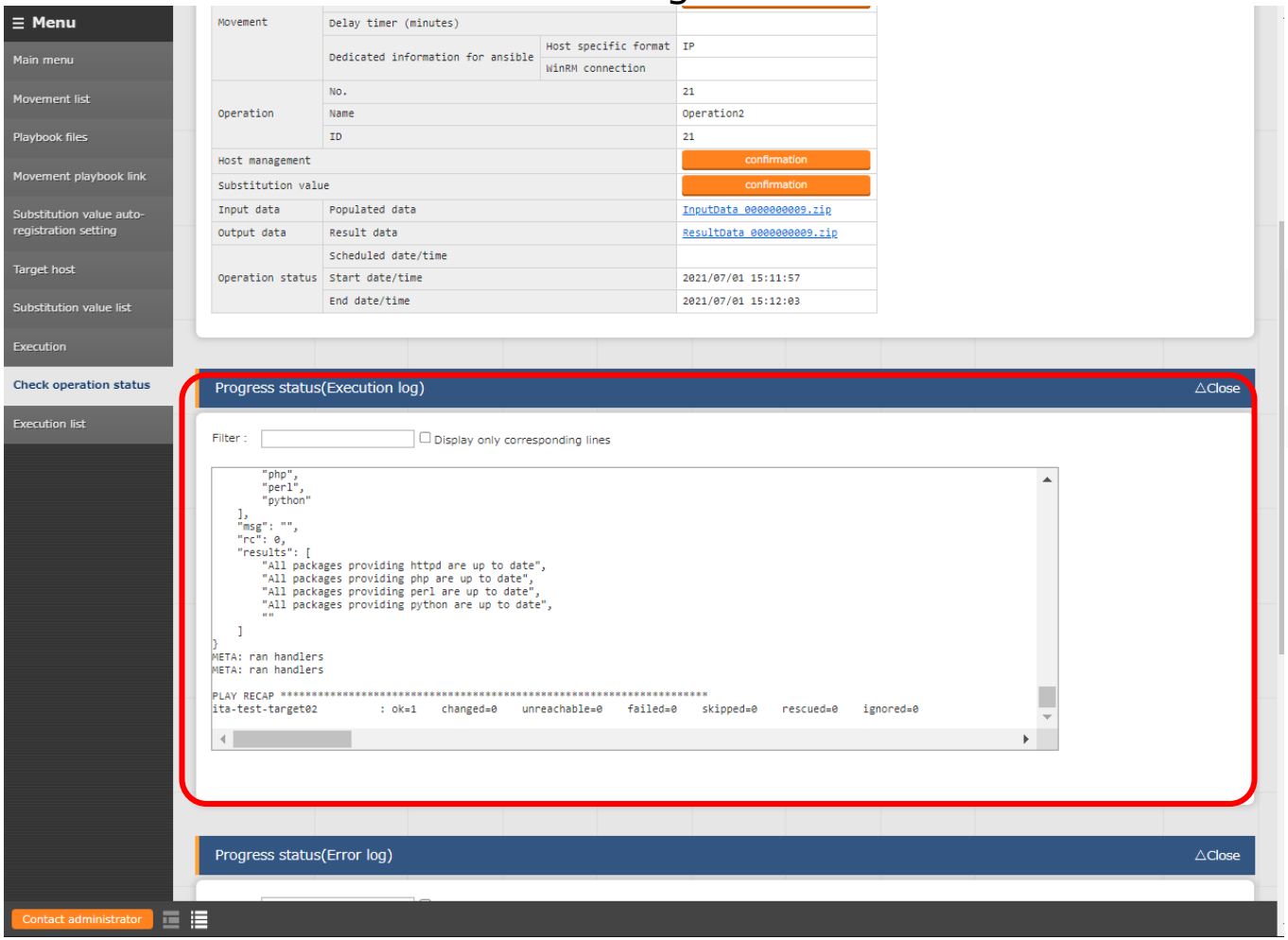
- Select a job (Movement) and press either the "Done" icon or the Operation status on the right to see more details.

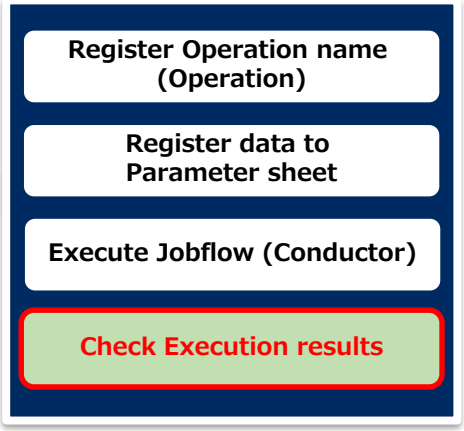


5.4 Check Execution results (1/2)

Execution results

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





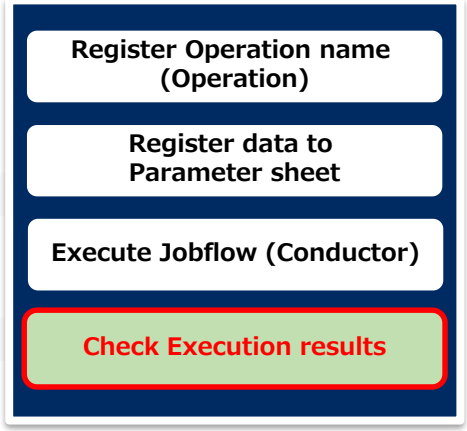
5.4 Check Execution results (2/2)

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-PlRPC.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",
""]}
```

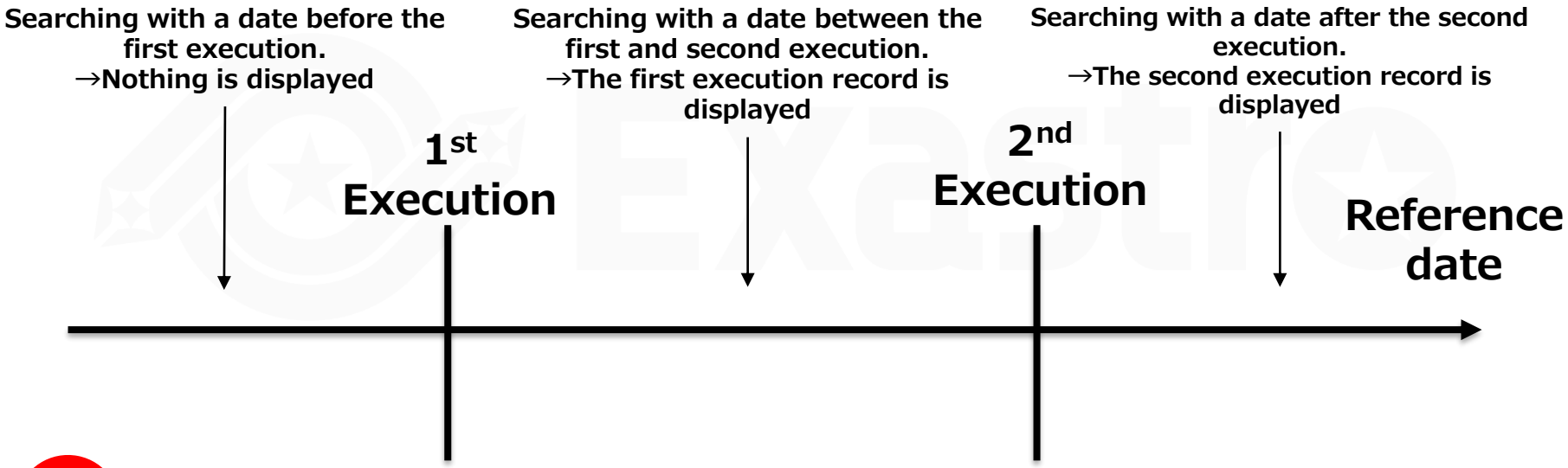


6. Checking the CMDB Parameter history

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.

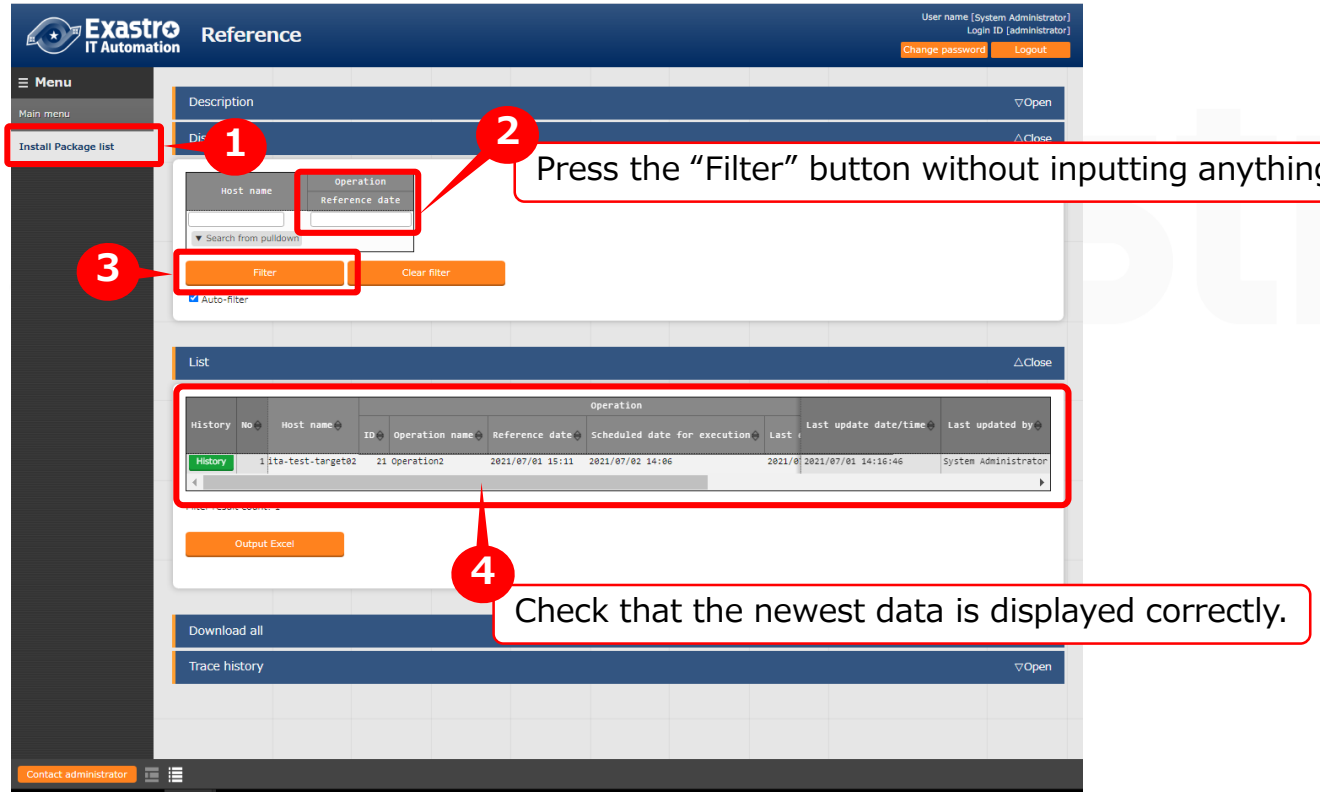
6.2 Checking the CMDB Parameter history (1/3)

History Check

- Check if the parameters are actually managed.

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

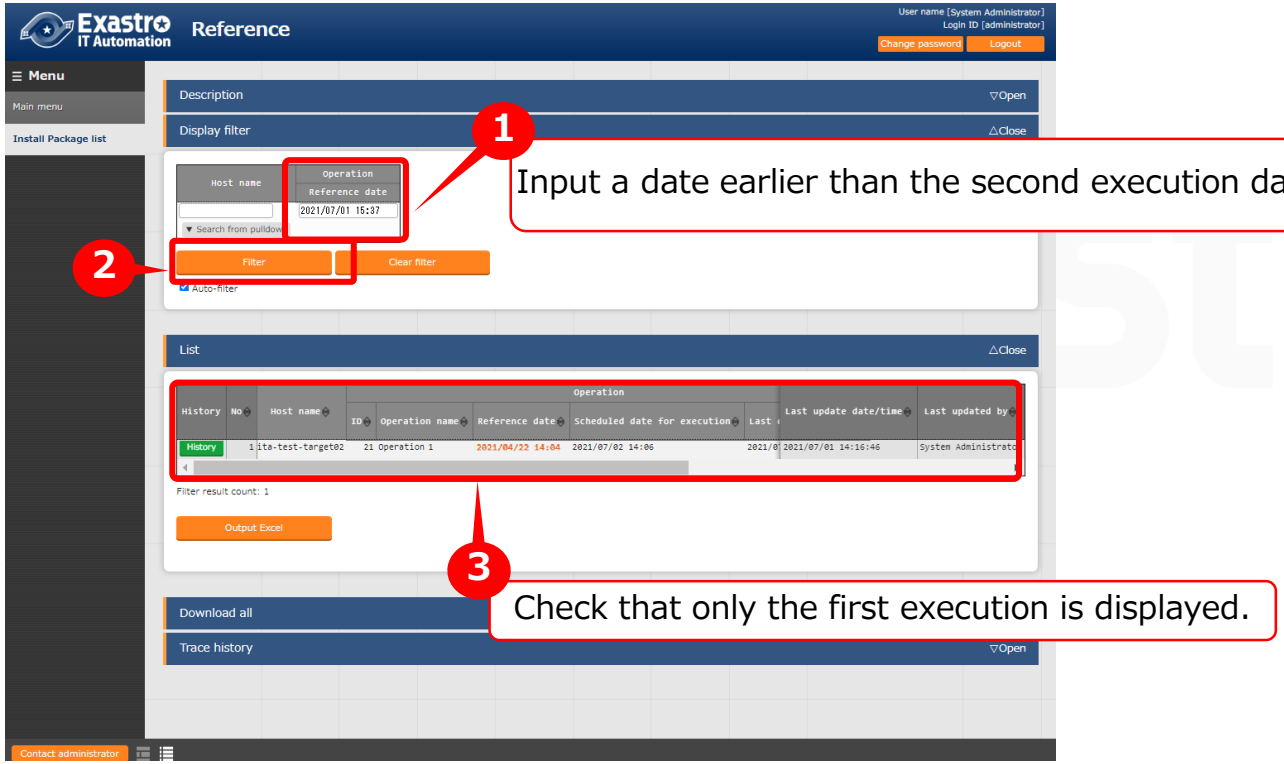
First, press the "Filter" button without inputting any filters.



6.2 Checking the CMDB Parameter history (2/3)

History Check

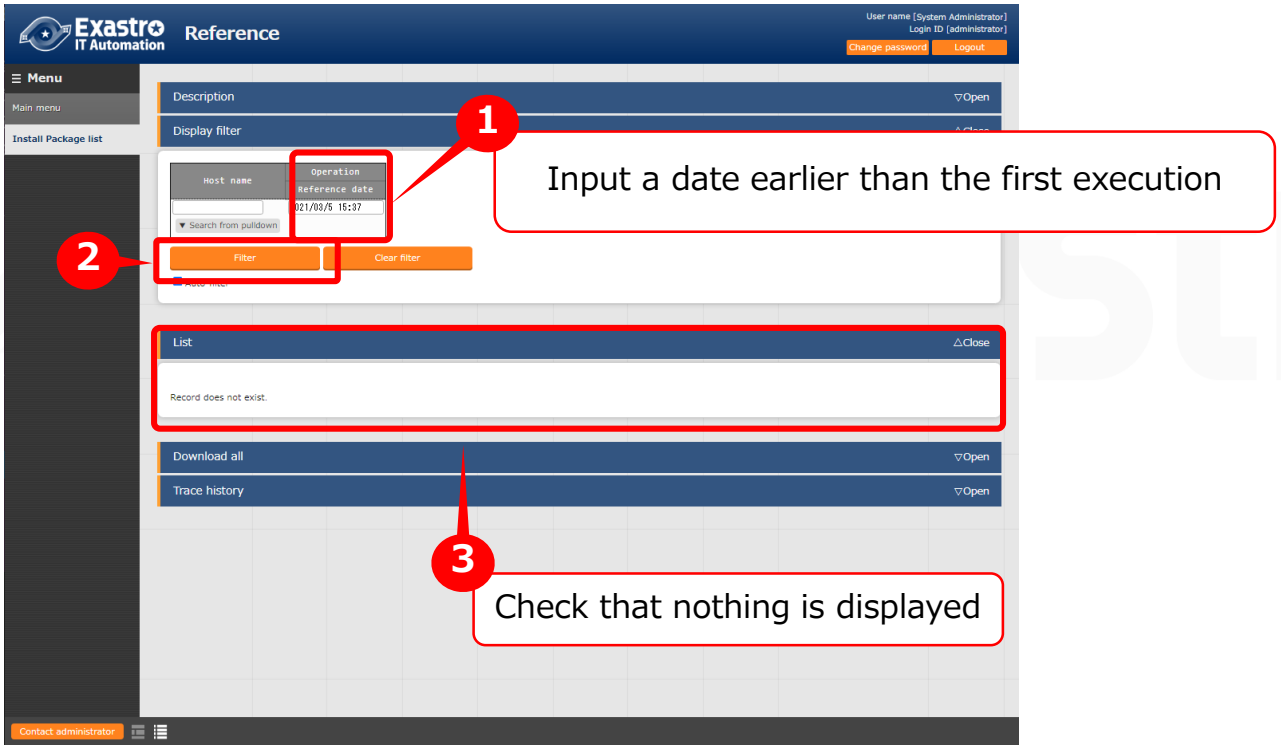
- Now, we will input a reference date that took place earlier than the second execution and filter.



6.2 Checking the CMDB Parameter history (3/3)

History Check

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

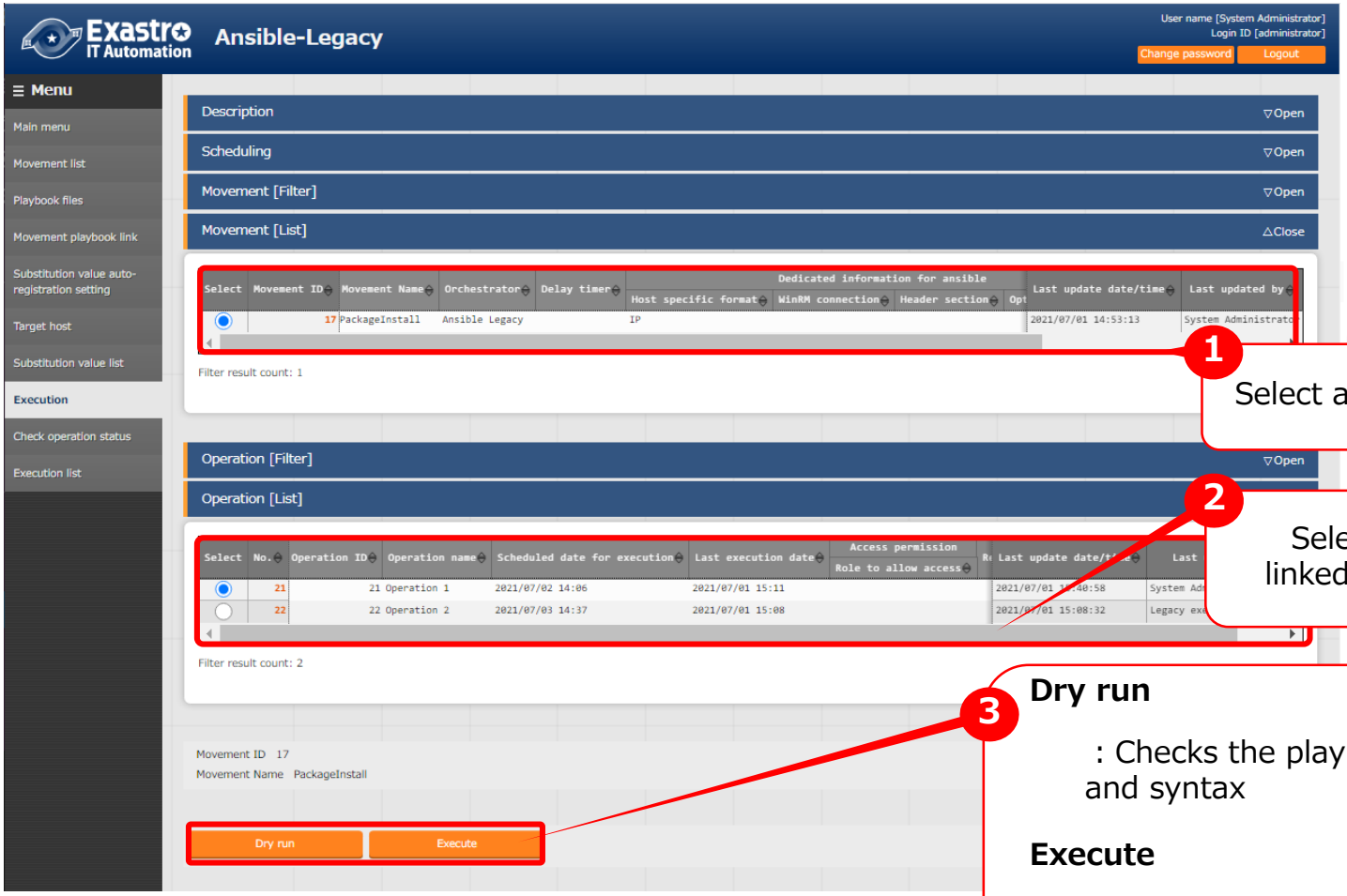


A Appendix

Reference ① 【Ansible-Legacy】Single Execution

Execution menu

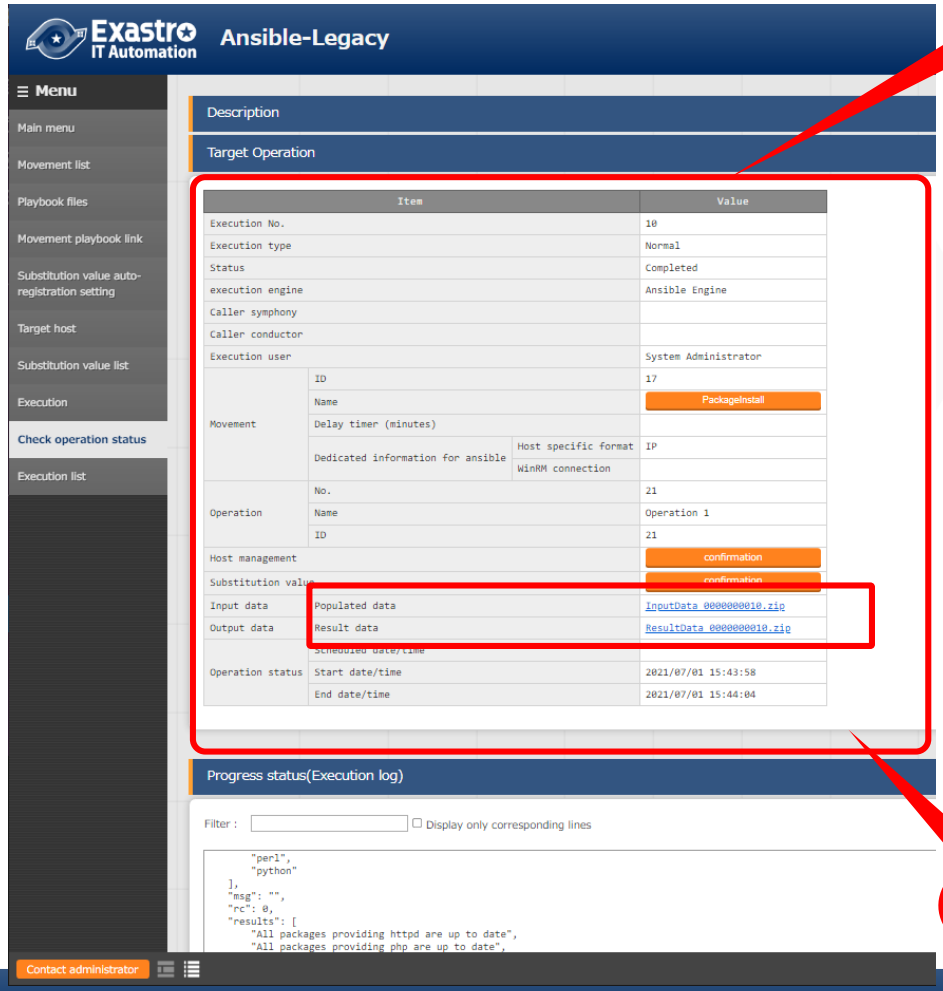
- Ansible-Legacy has a "Execution" menu where users can execute individual movements and dry run them.

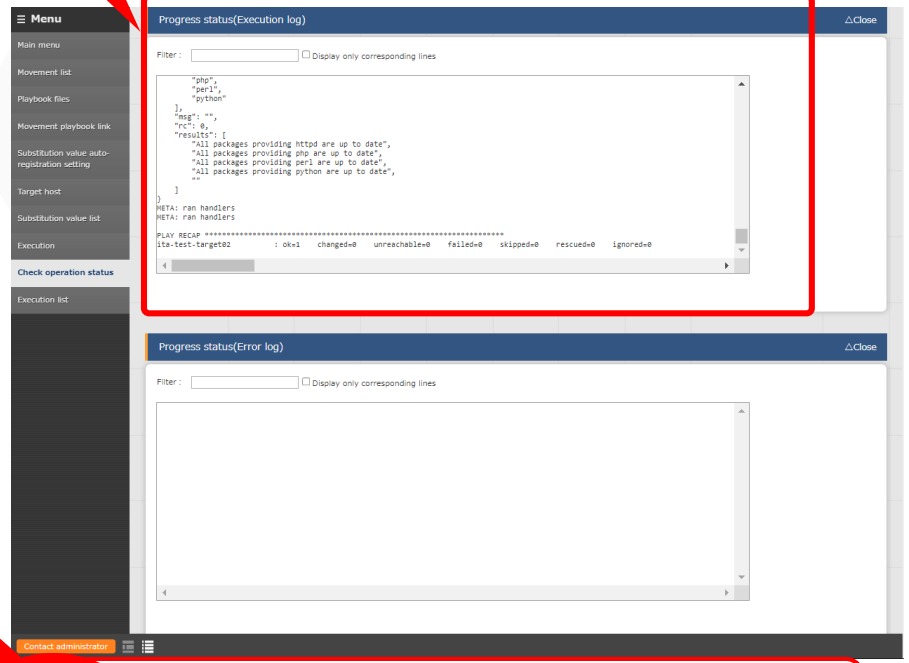


Reference ② 【Ansible-Legacy】Execution check

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.





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