Exastro

IT Automation
Quickstart

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

Exastro IT Automation Version 1.7.2
Exastro developer

Table of contents

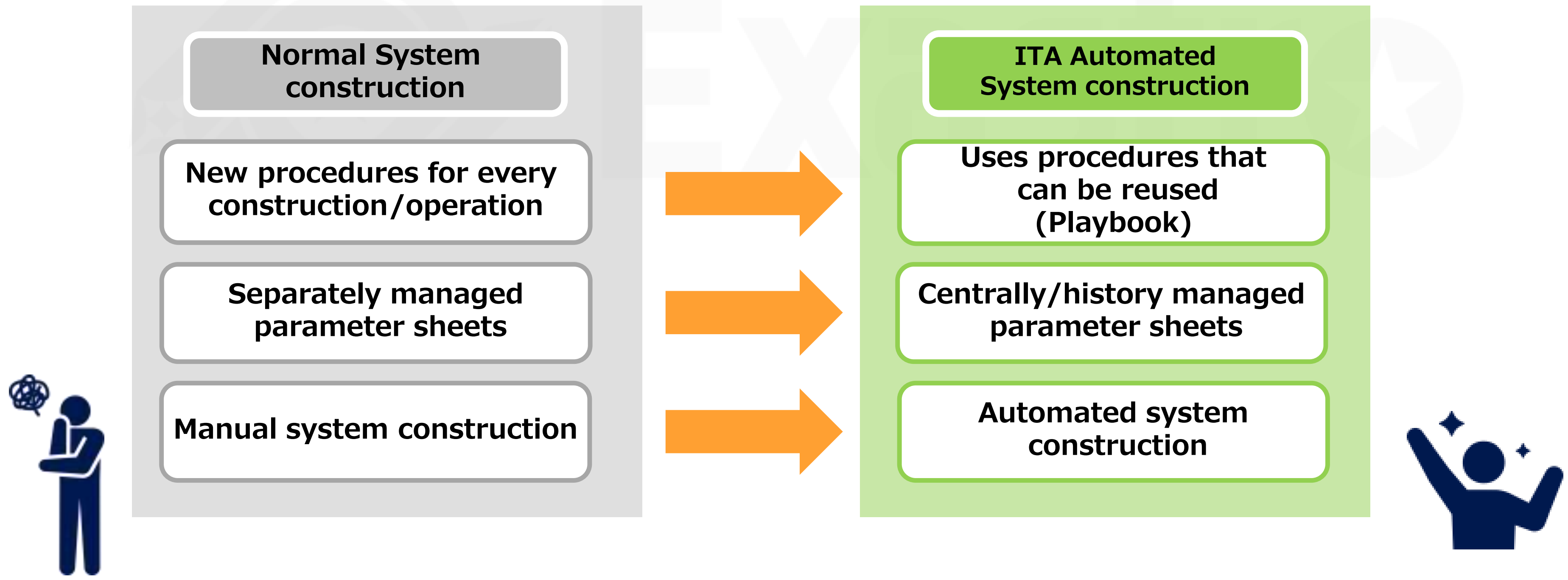
Table of Contents

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.


Normal System construction
New procedures for every construction/operation
Separately managed parameter sheets
Manual system construction
ITA Automated System construction
Uses procedures that can be reused (Playbook)
Centrally/history managed parameter sheets
Automated system construction

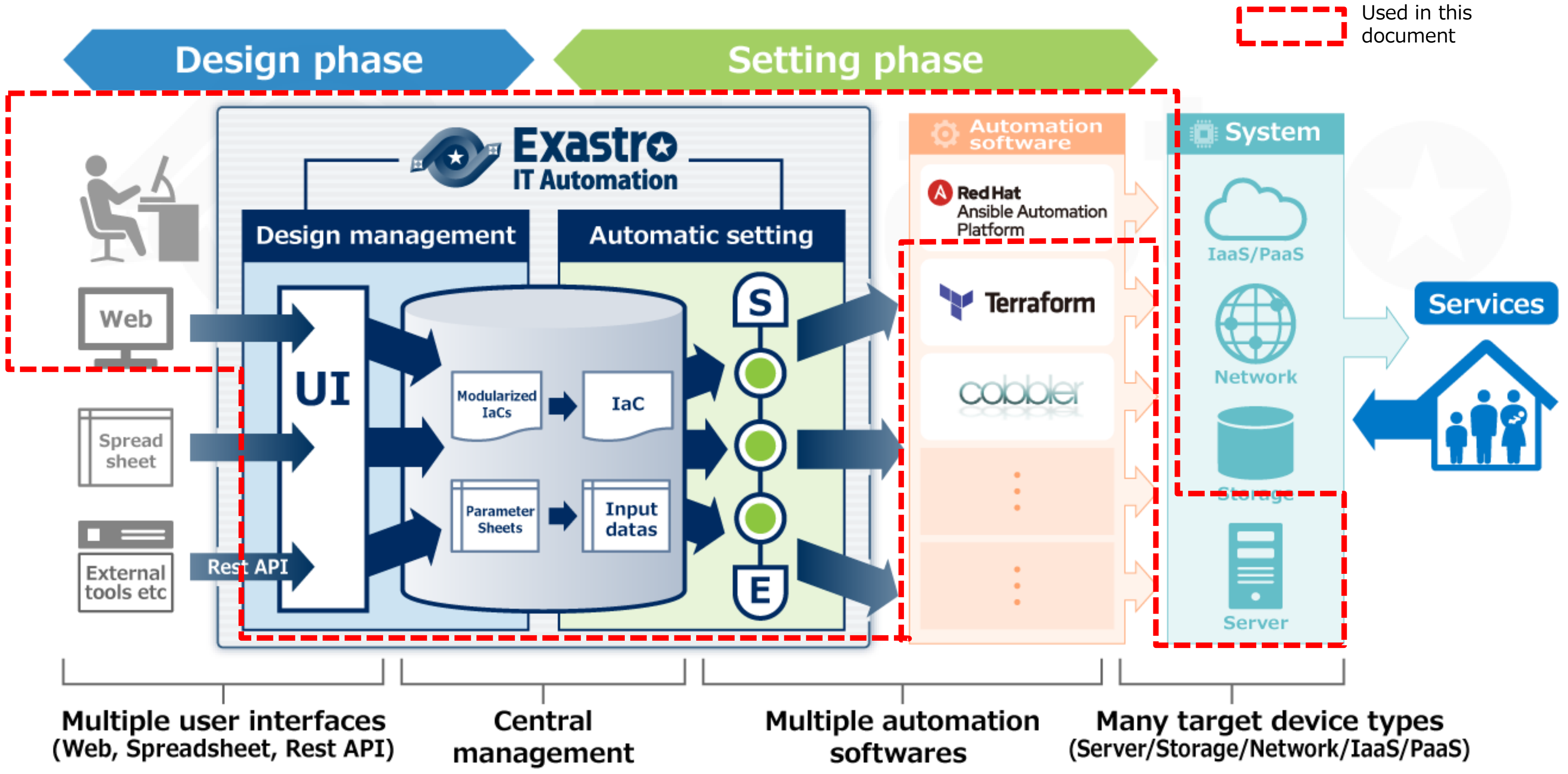
1.1 Introduction (2/2)

Main ITA functions used in this document.

・ Linking with Automation software (Ansible).

・ Parameter management (Creation, Registration and history management of Menus)

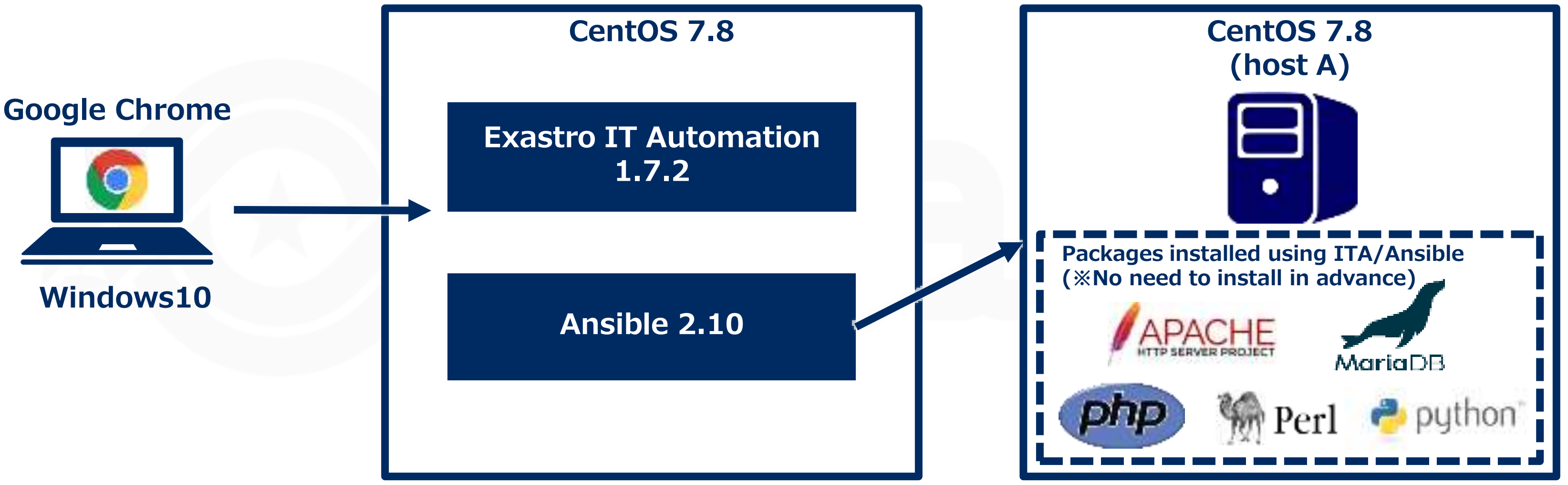
・ Linking Variables (Automatic registration of substitute values)



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



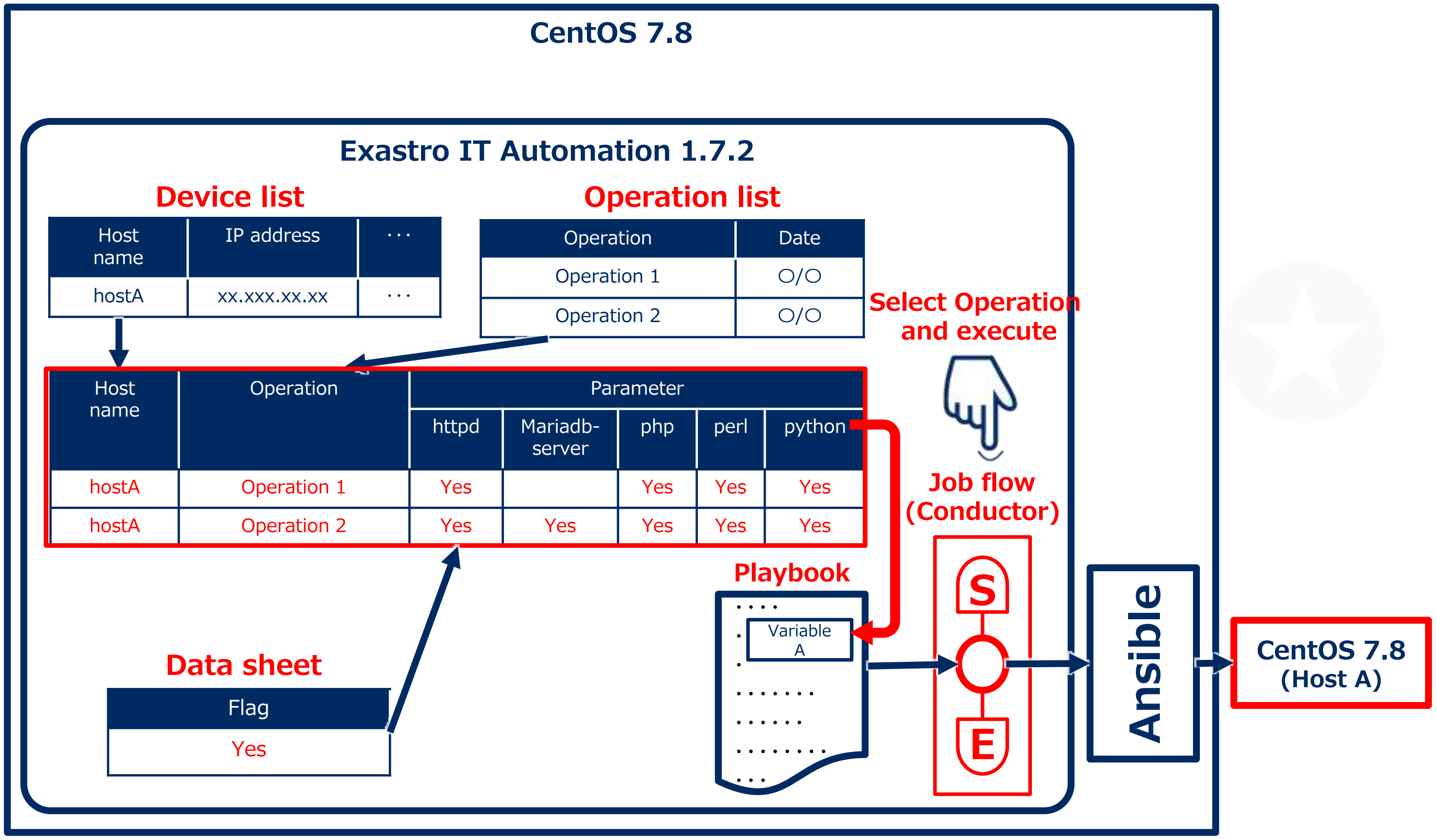
Google Chrome
Windows10
CentOS 7.8
Exastro IT Automation 1.7.2
Ansible 2.10
CentOS 7.8
(host A)
Packages installed using ITA/Ansible
(※No need to install in advance)
APACHE HTTP SERVER PROJECT
MariaDB
php
Perl
python

Systems used

- Exastro IT Automation 1.7.2
- 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



Device list

Host name	IP address	・・・
hostA	xx.xxx.xx.xx	・・・

Operation list

Operation	Date
Operation 1	○/○
Operation 2	○/○

Host name	Operation	Parameter				
		httpd	Mariadb-server	php	perl	python
hostA	Operation 1	Yes		Yes	Yes	Yes
hostA	Operation 2	Yes	Yes	Yes	Yes	Yes

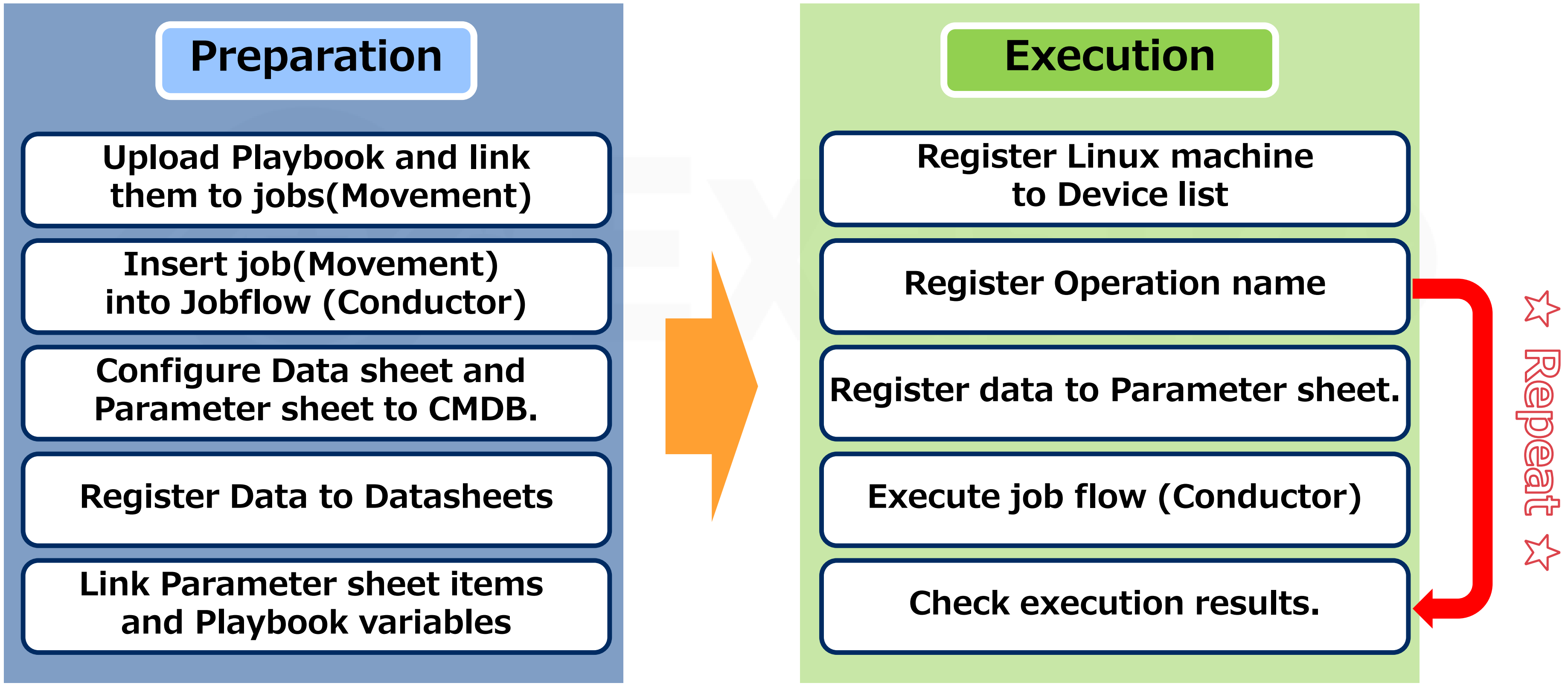
Data sheet

Flag
Yes

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.


Preparation
Upload Playbook and link them to jobs(Movement)
Insert job(Movement) into Jobflow (Conductor)
Configure Data sheet and Parameter sheet to CMDB.
Register Data to Datasheets
Link Parameter sheet items and Playbook variables
Execution
Register Linux machine to Device list
Register Operation name
Register data to Parameter sheet.
Execute job flow (Conductor)
Check execution results.
☆ Repeat ☆

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 workflow. 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 the terminology, please refer to the first step guide.
If you want more information regarding Exastro ITA, please refer to the Document page on the community website.

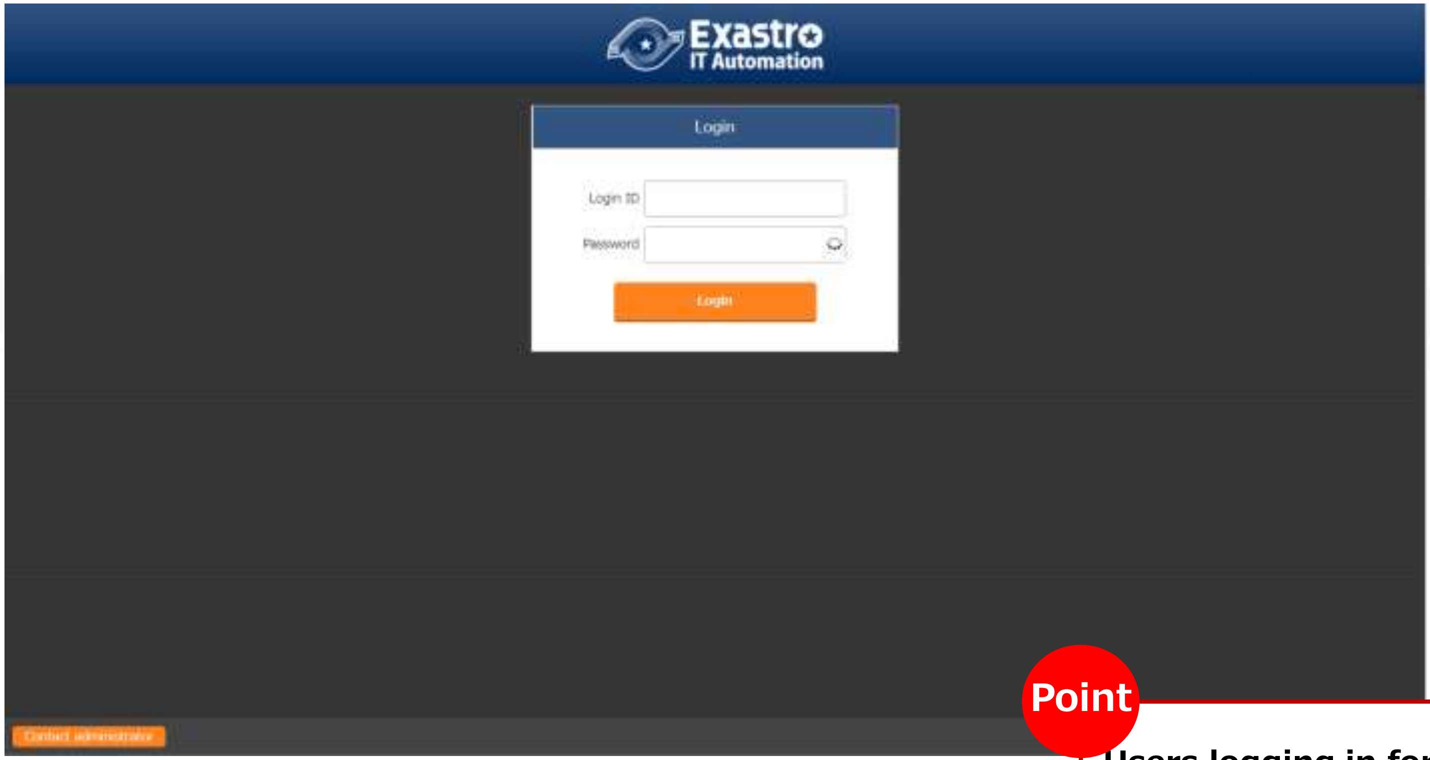
2.　Screen Description

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



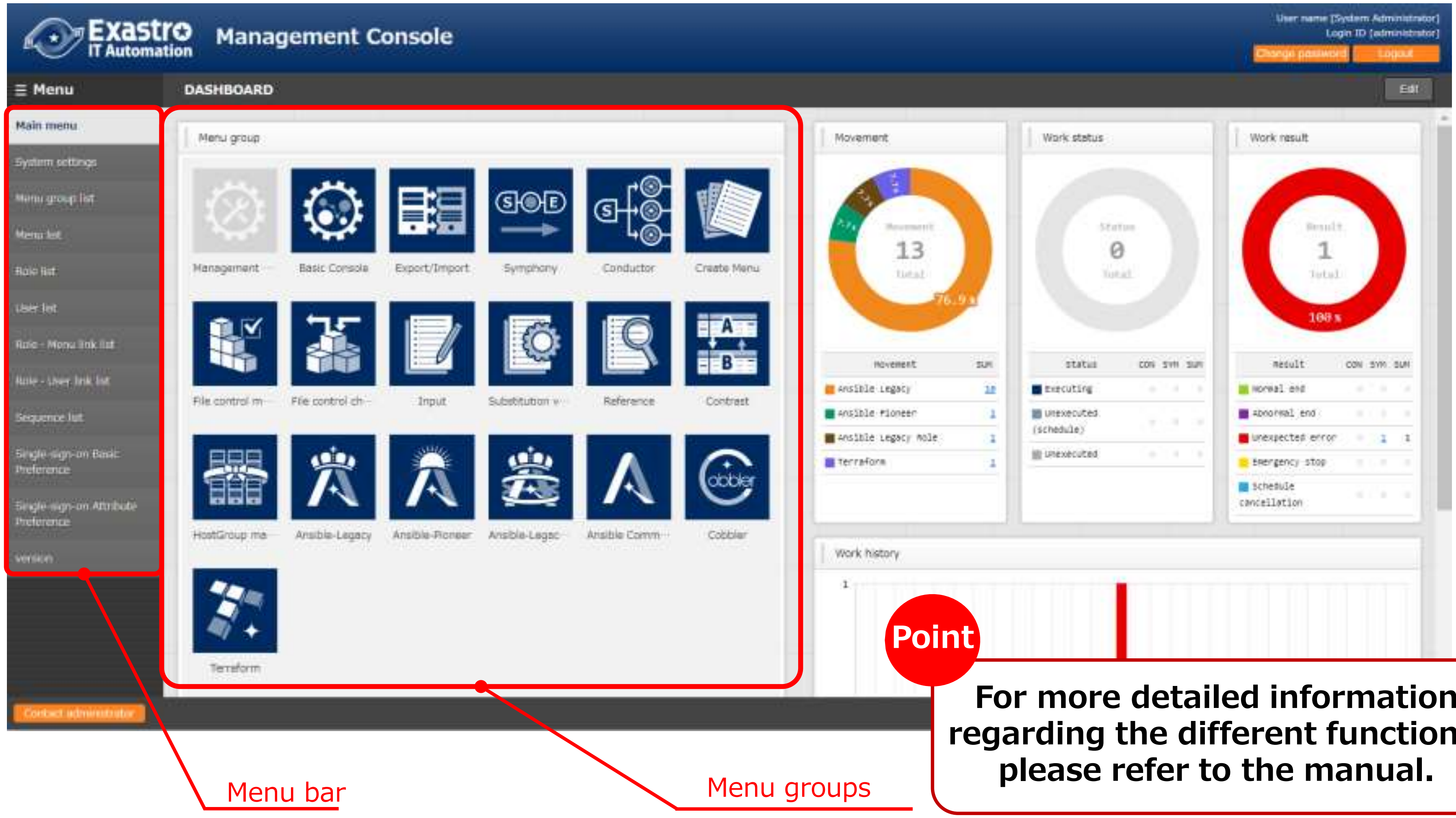
Point

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



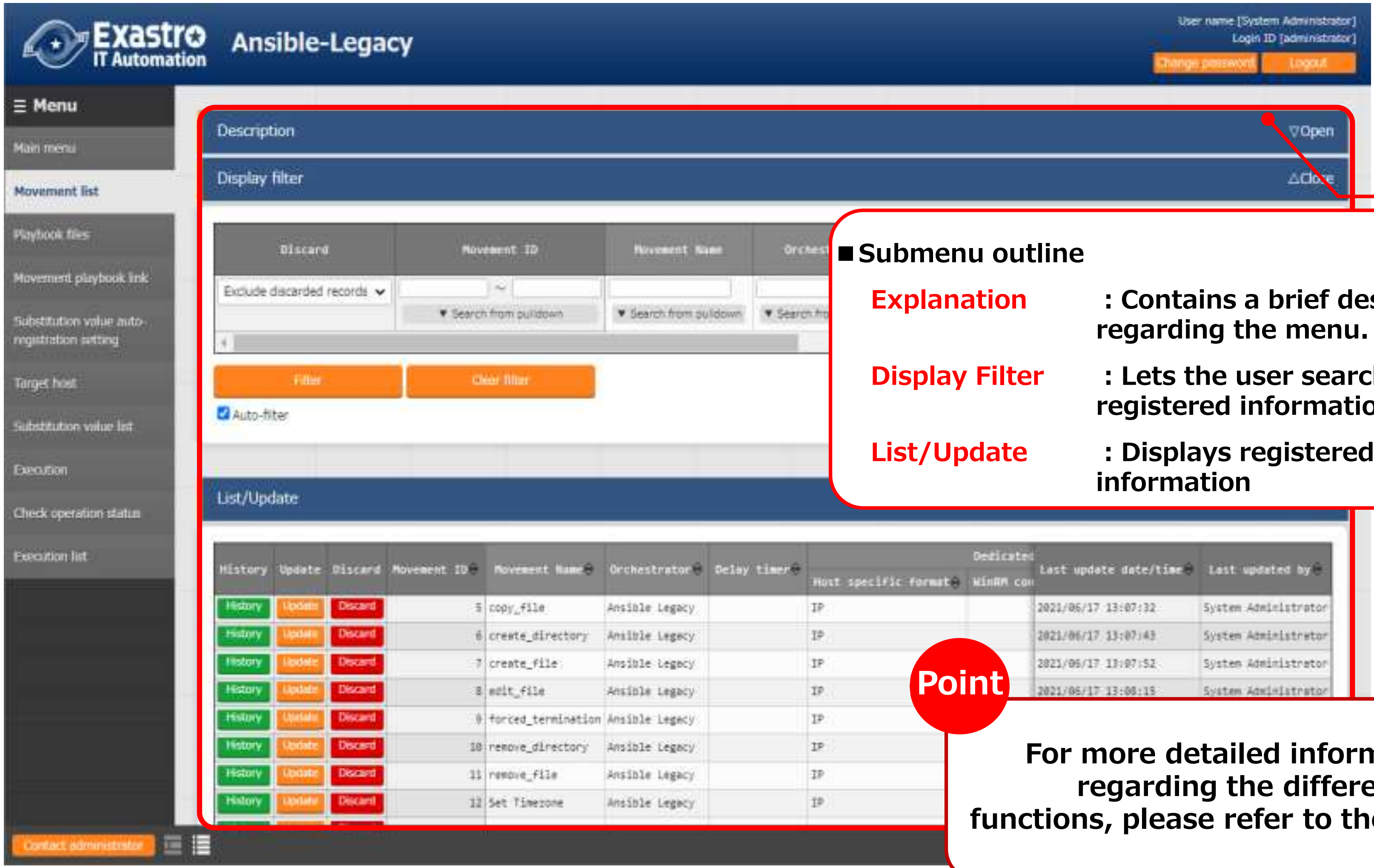
Point

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.



History	Update	Discard	Movement ID	Movement Name	Orchestrator	Delay timer	Dedicated		Last update date/time	Last updated by
							Host specific format	WinRM con		
History	Update	Discard	5	copy_file	Ansible Legacy		IP		2021/06/17 13:07:32	System Administrator
History	Update	Discard	6	create_directory	Ansible Legacy		IP		2021/06/17 13:07:43	System Administrator
History	Update	Discard	7	create_file	Ansible Legacy		IP		2021/06/17 13:07:52	System Administrator
History	Update	Discard	8	edit_file	Ansible Legacy		IP		2021/06/17 13:08:15	System Administrator
History	Update	Discard	9	forced_termination	Ansible Legacy		IP			
History	Update	Discard	10	remove_directory	Ansible Legacy		IP			
History	Update	Discard	11	remove_file	Ansible Legacy		IP			
History	Update	Discard	12	Set Timezone	Ansible Legacy		IP			

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**

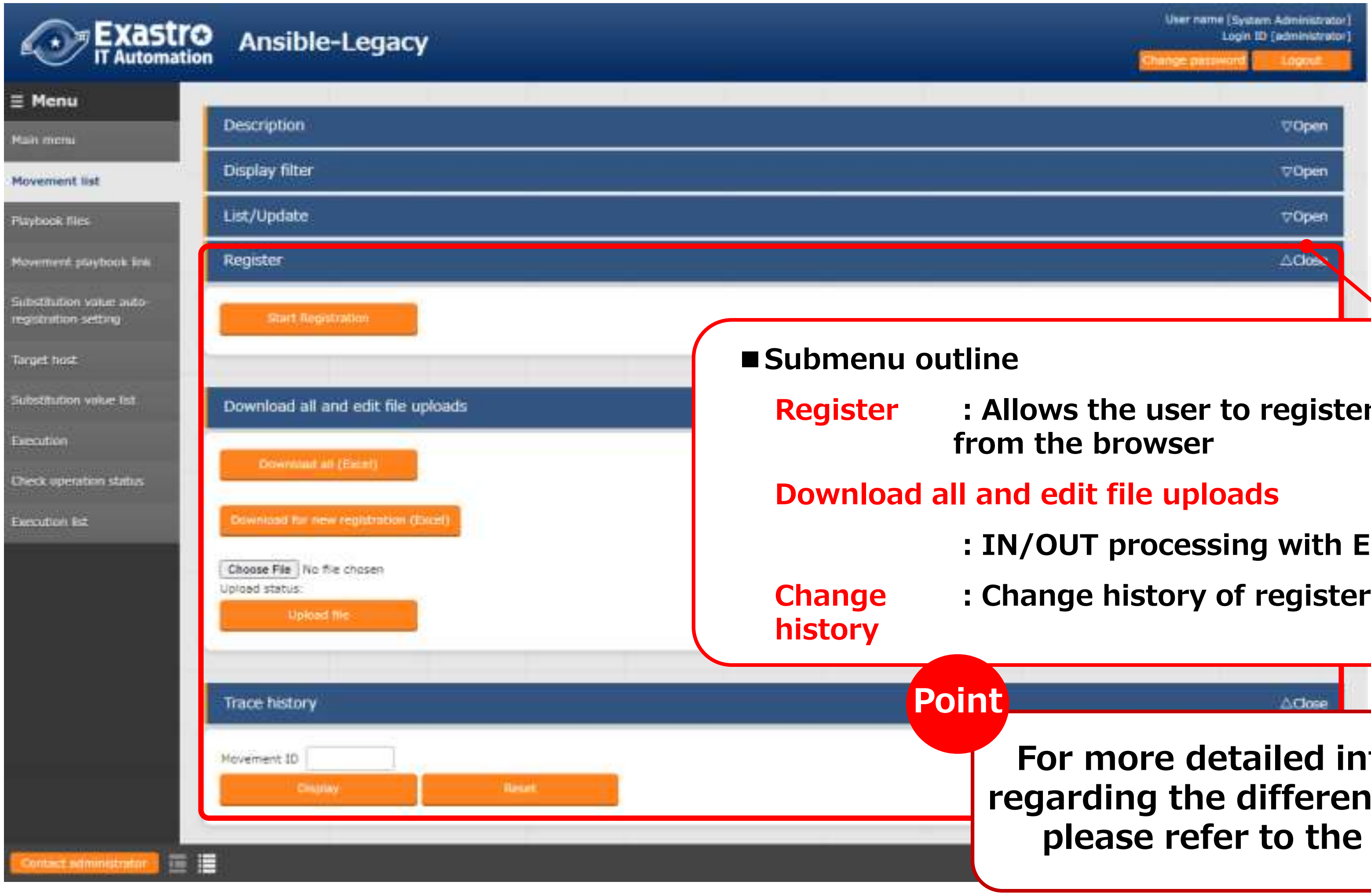
Point

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

2.3 Screen Description (Menu) (2/2)

Screen description (Menu)

● The name of the basic functions are as following



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.

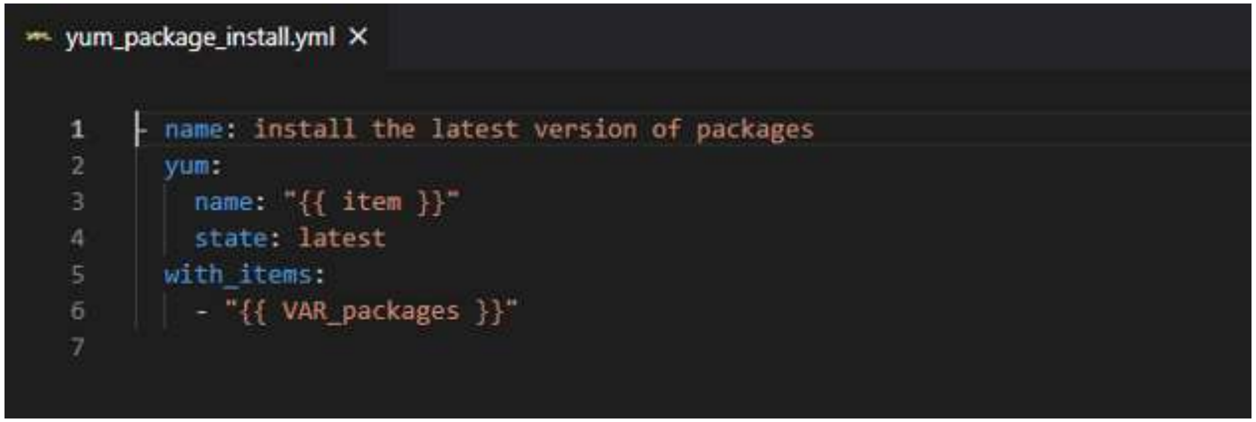
Use your desired editor program to create the following YML file and save it to your local hard drive.

yum_package_install.yml

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

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.

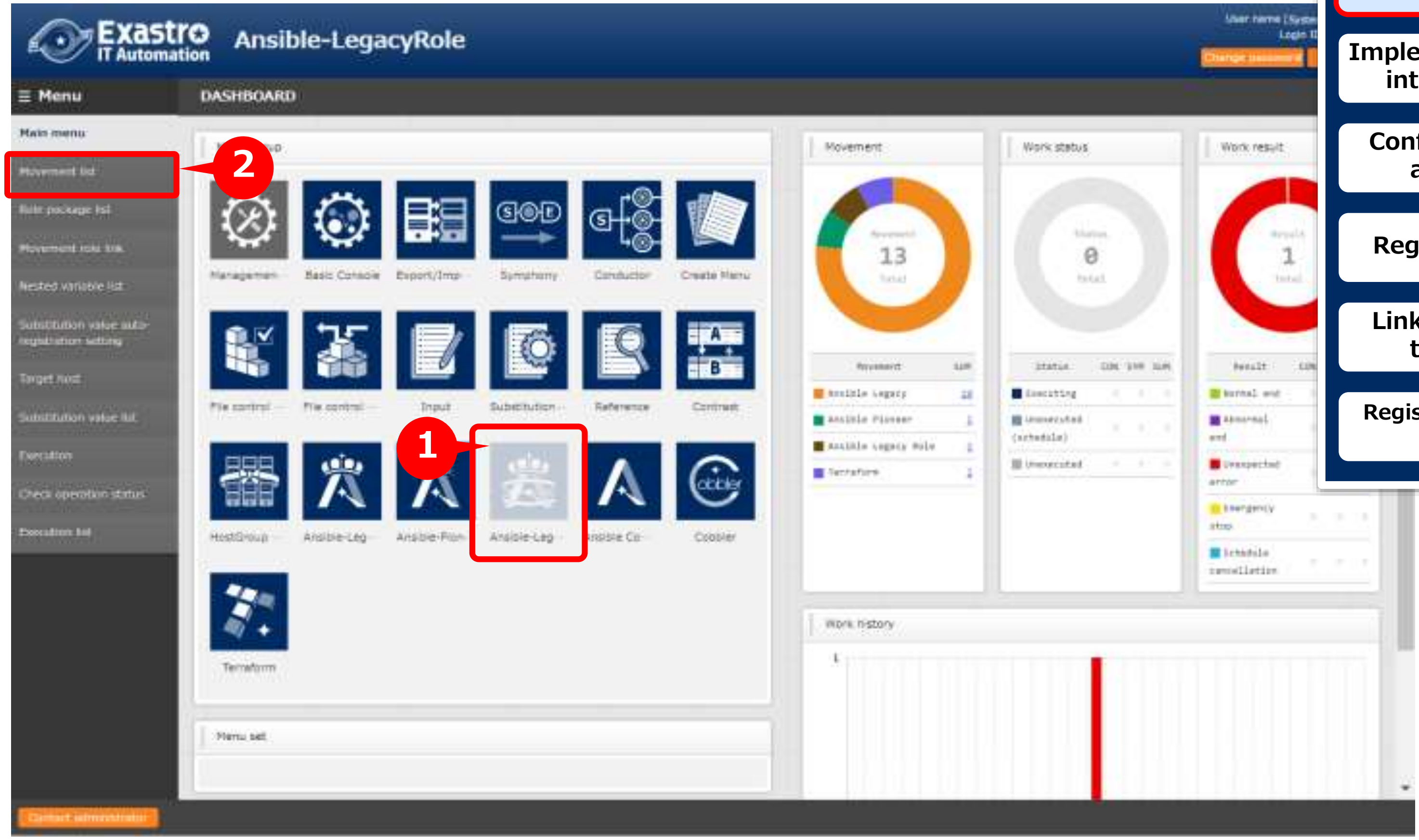


```
- name: install the latest version of packages
  yum:
    name: "{{ item }}"
    state: latest
  with_items:
    - "{{ VAR_packages }}"

```

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-LegacyRole 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.

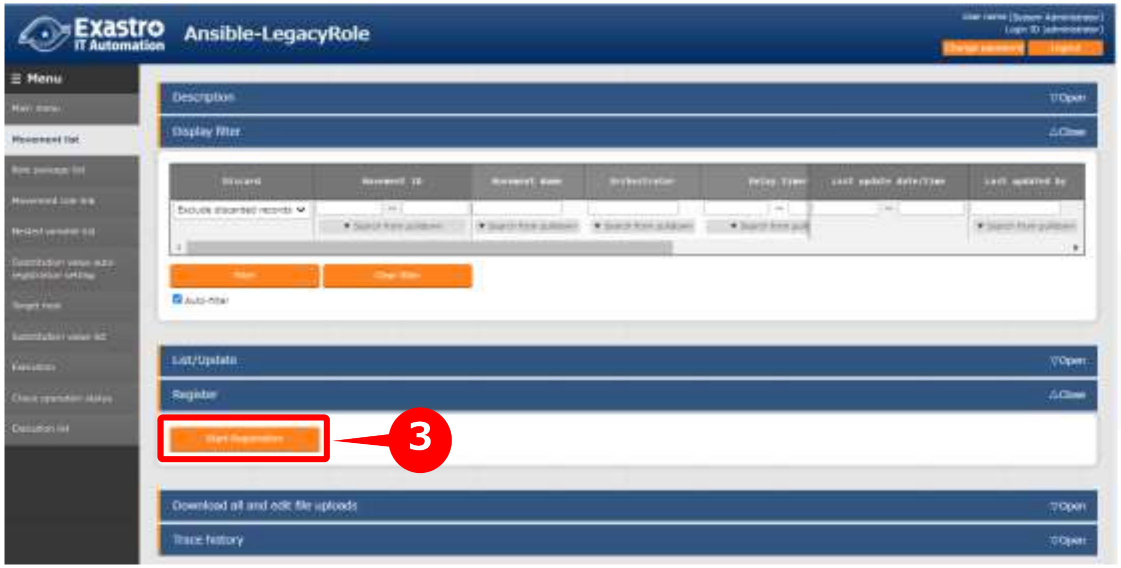
Register target (Linux machine) to Device list.

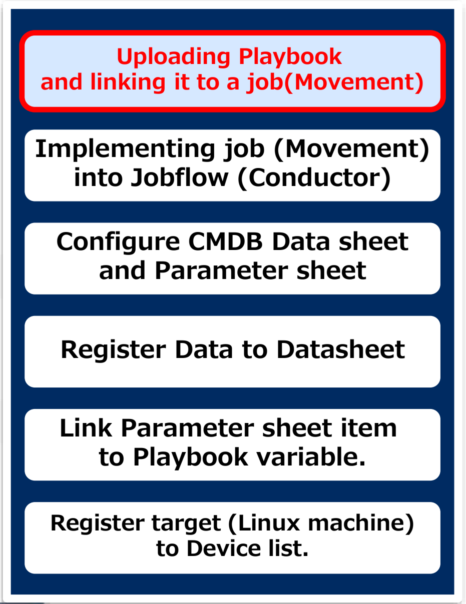
3.1 Uploading Playbook and linking it to a job(Movement) (3/5)

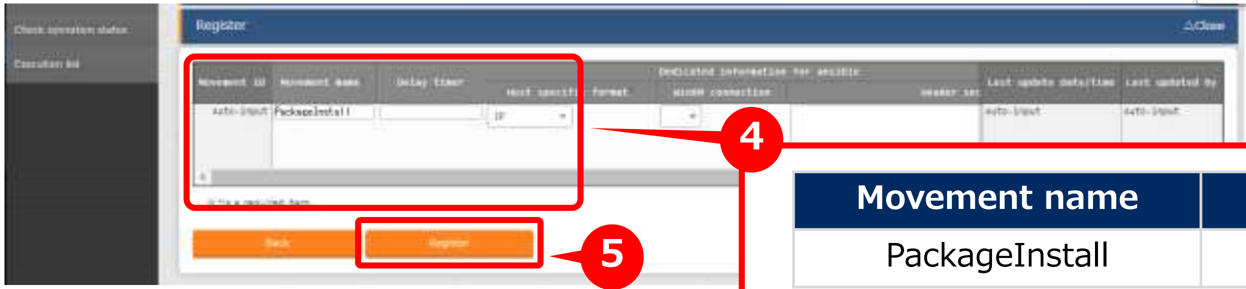
Register Movement to the Movement list.

● Click the “Start Registration” button.

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




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 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)

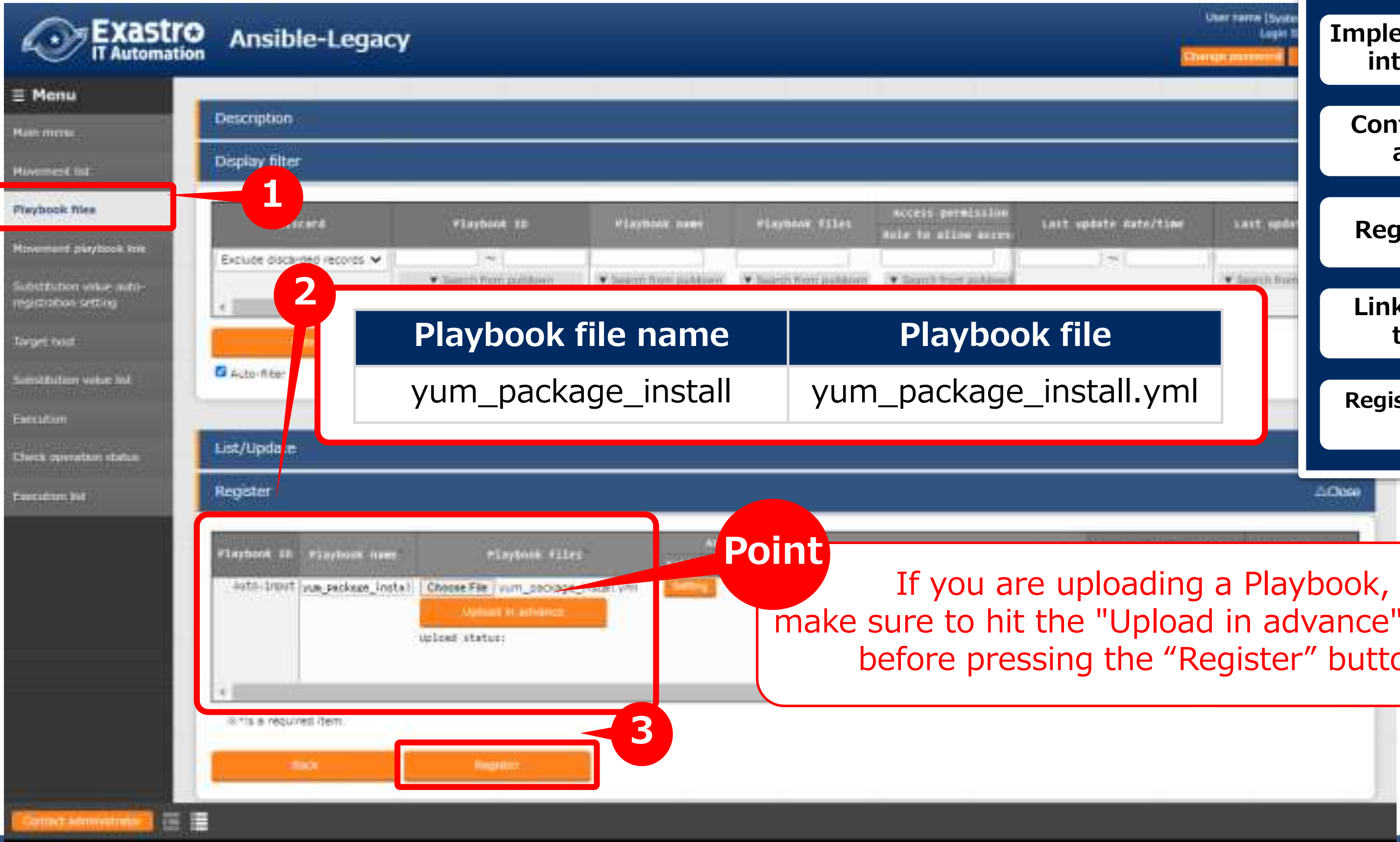
Implementing job (Movement) into Jobflow (Conductor)

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.



Playbook file name	Playbook file
yum_package_install	yum_package_install.yml

Point

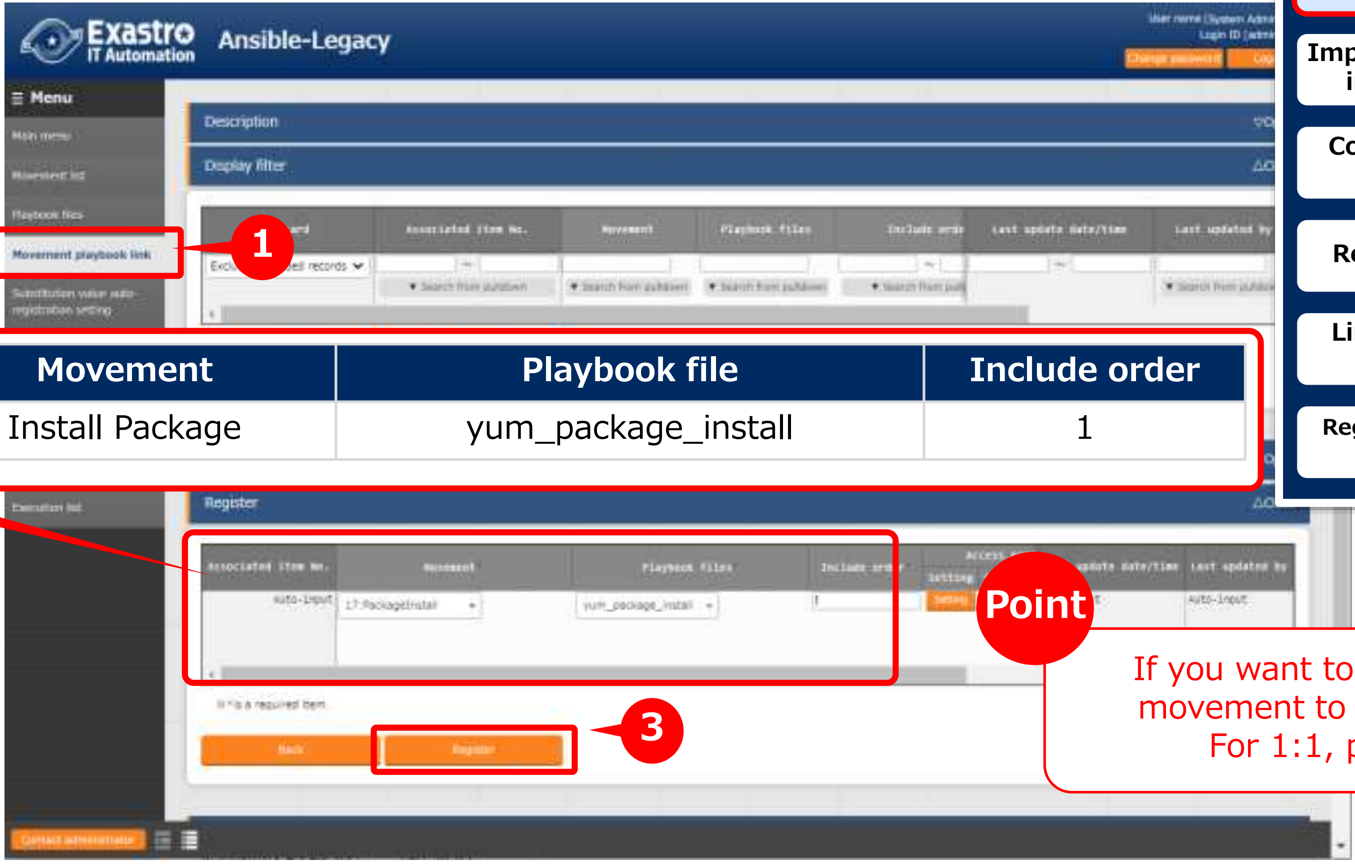
If you are uploading a Playbook, make sure to hit the "Upload in advance" button before pressing the “Register” button.

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
Install Package	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.

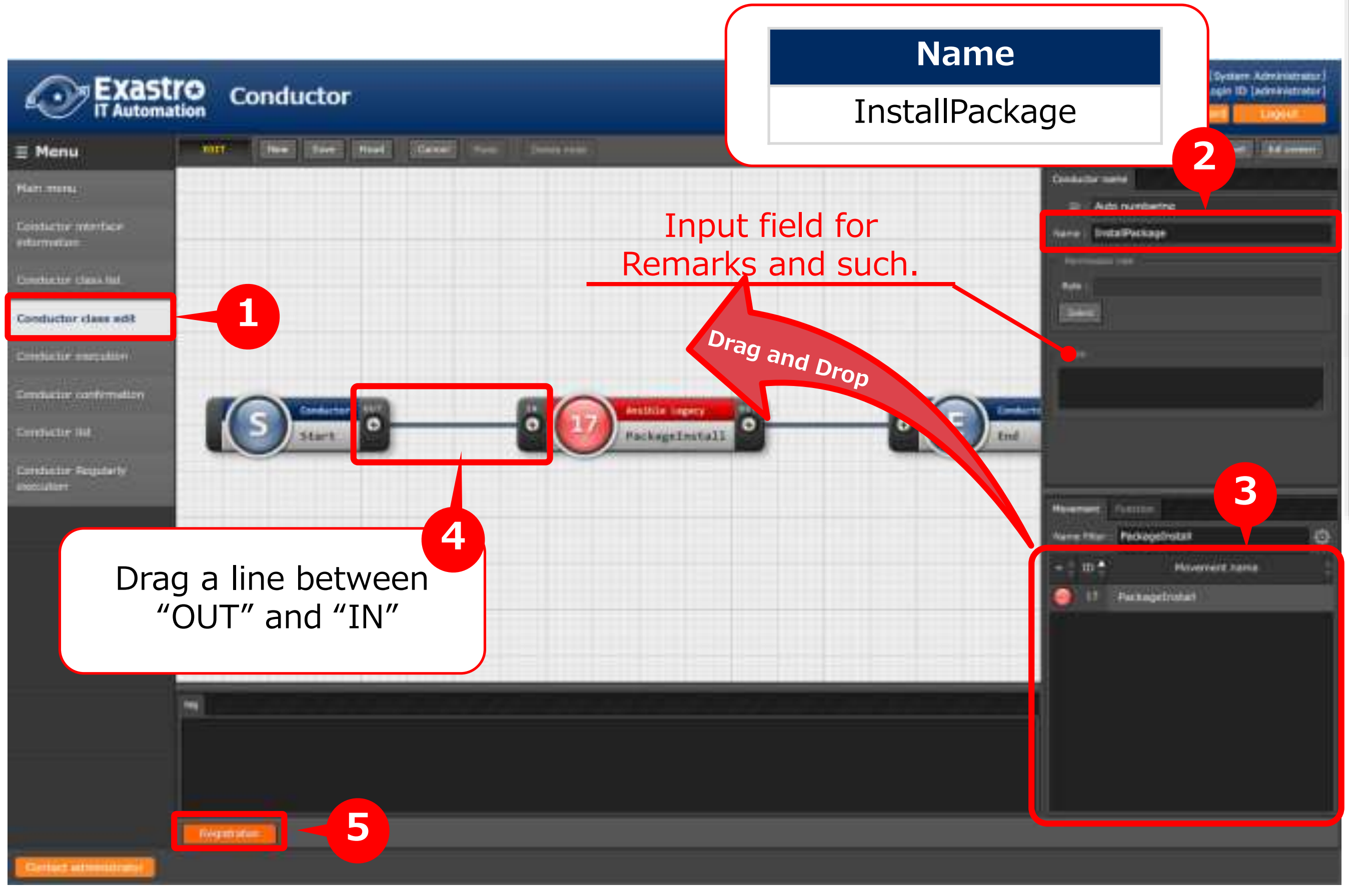
3.2 Implementing job (Movement)into Jobflow (Conductor)

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.



Name
InstallPackage

Drag a line between “OUT” and “IN”

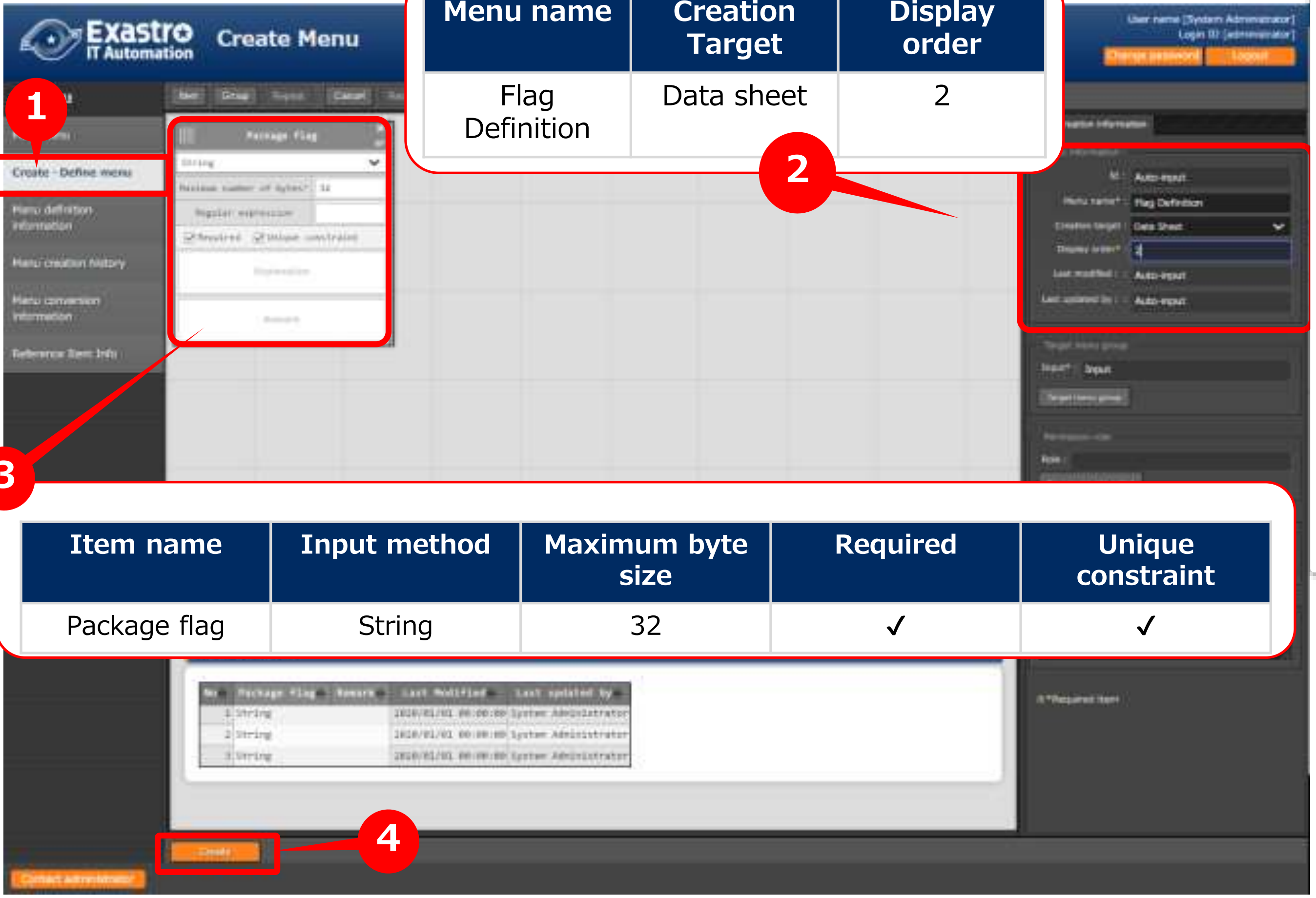
3.3 Configure CMDB Data sheet and Parameter sheet (1/4)

Create Data sheet

- Next, we will create a Data sheet

From the “Menu create” menu group, access the “Create・Define menu” menu.

Fill out the items marked with red using the information from table listed below and press the “”Create”” button.



Menu name	Creation Target	Display order
Flag Definition	Data sheet	2

Item name	Input method	Maximum byte size	Required	Unique constraint
Package flag	String	32	✓	✓

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.

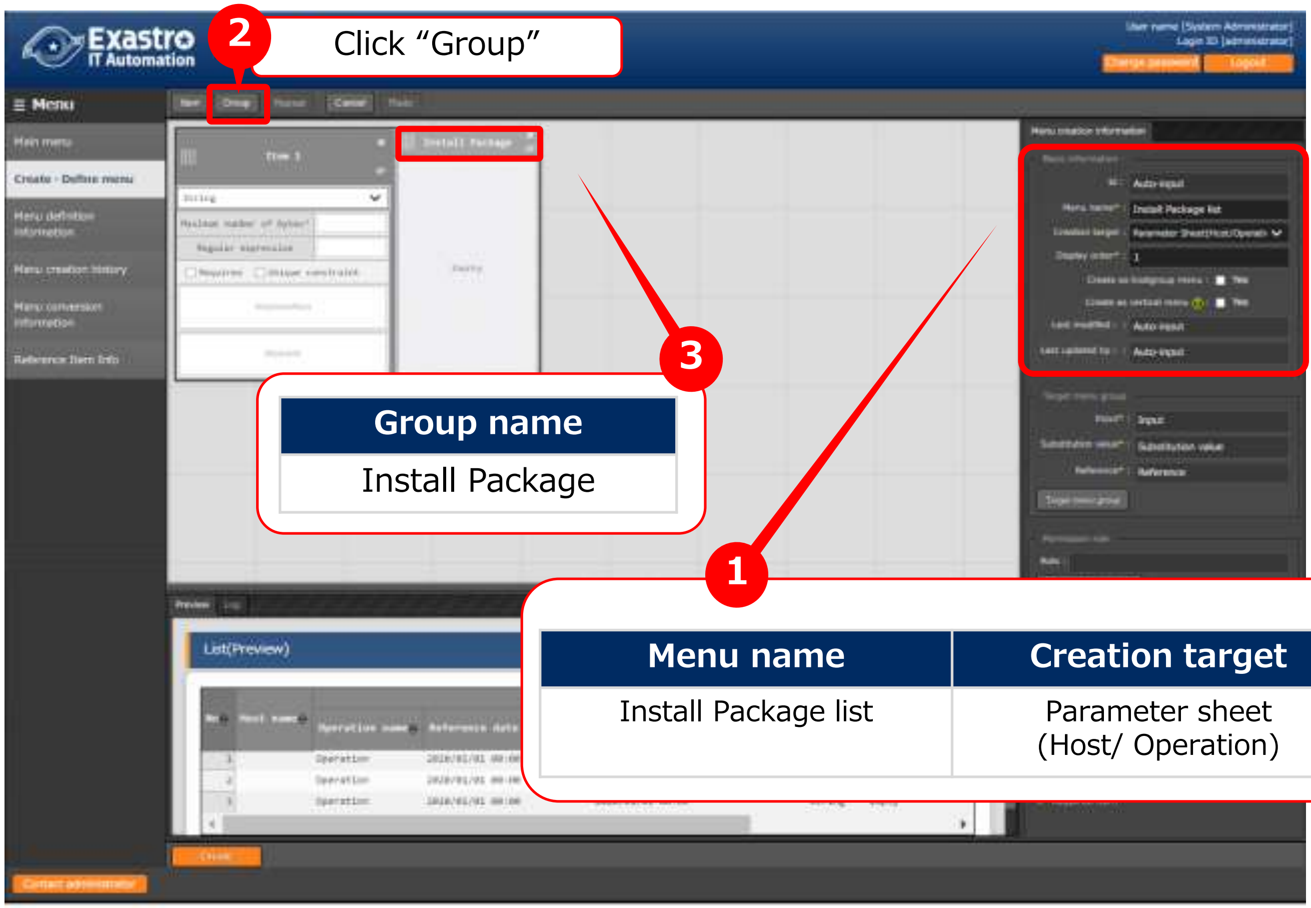
Register target (Linux machine) to Device list.

3.3 Configure CMDB Data sheet and Parameter sheet (2/4)

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.



Group name
Install Package

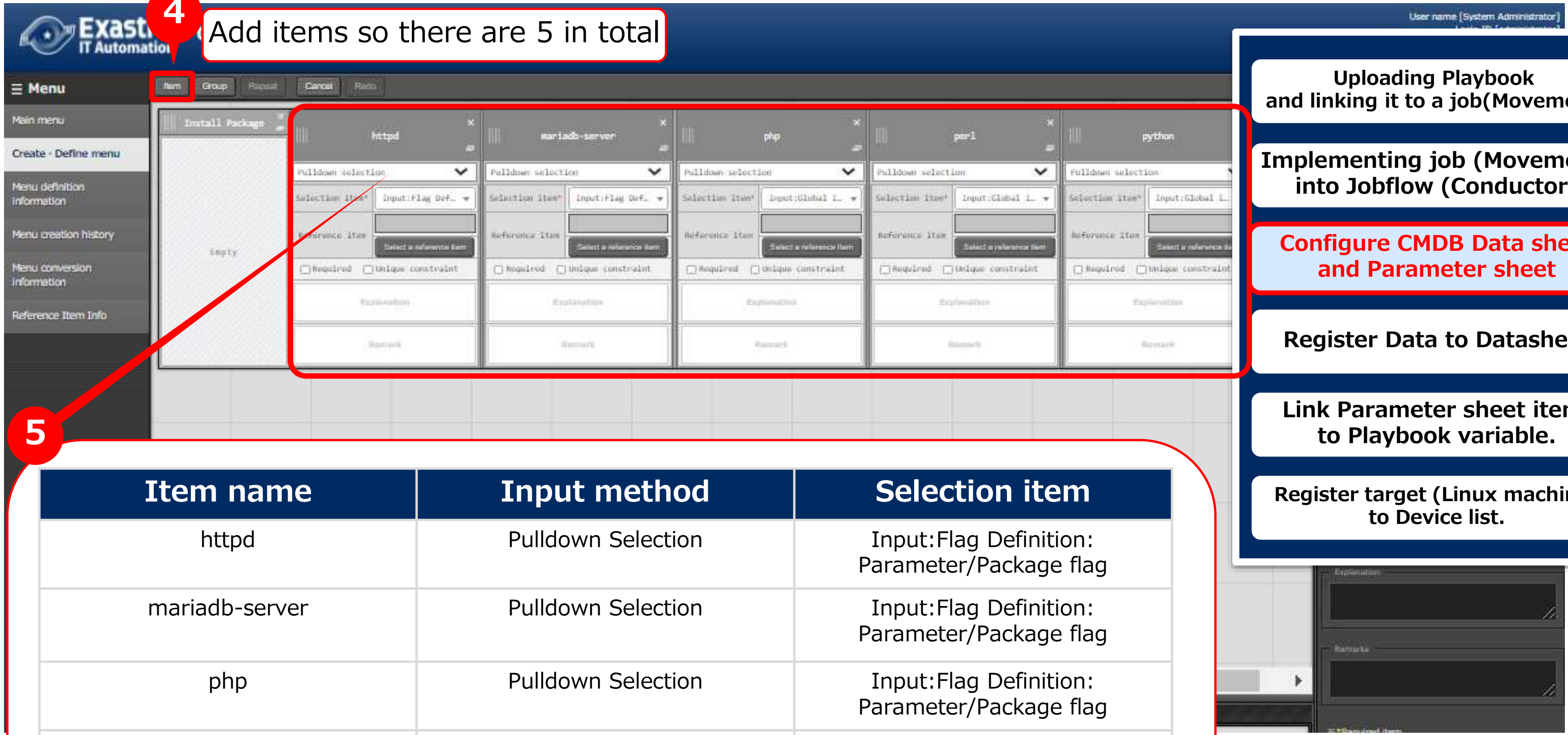
Menu name	Creation target	Display Order
Install Package list	Parameter sheet (Host/ Operation)	1

- 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 (3/4)

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

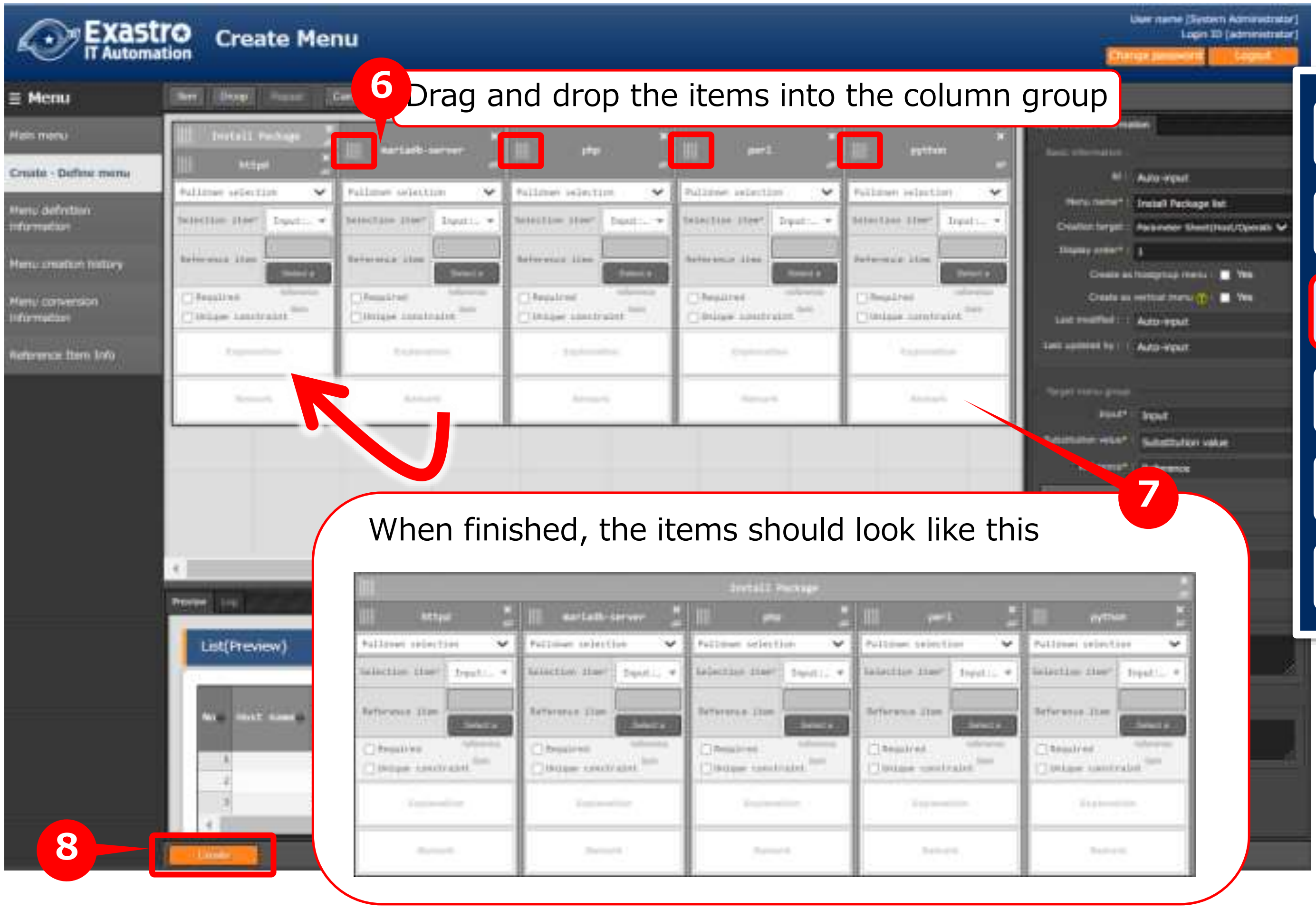
Item name	Input method	Selection item
httpd	Pulldown Selection	Input:Flag Definition: Parameter/Package flag
mariadb-server	Pulldown Selection	Input:Flag Definition: Parameter/Package flag
php	Pulldown Selection	Input:Flag Definition: Parameter/Package flag
perl	Pulldown Selection	Input:Flag Definition: Parameter/Package flag
python	Pulldown Selection	Input:Flag Definition: Parameter/Package flag

- 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 (4/4)

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

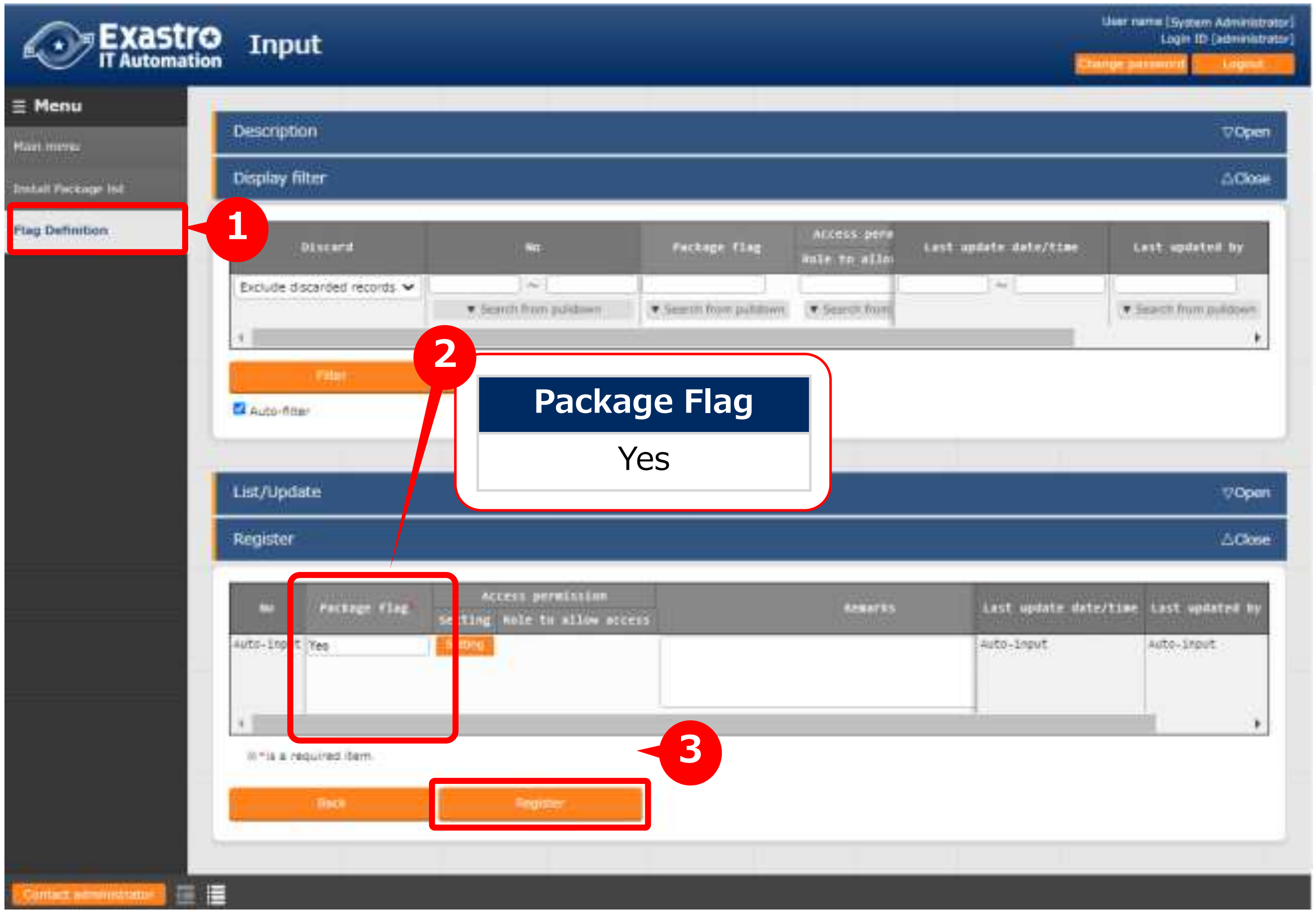
Link Parameter sheet item to Playbook variable.

Register target (Linux machine) to Device list.

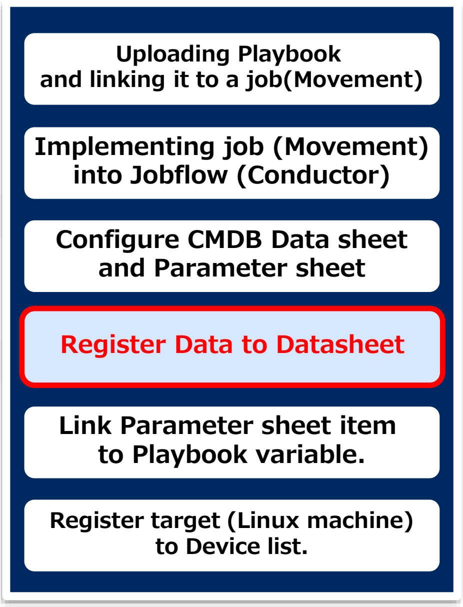
3.4 Register Data to Datasheet

Register data to “Flag definition”

We are now going to register data in the Flag Definition (Datasheet) that we created earlier.
Go to the "Input" menu group and then to the "Flag Definition" Menu.
Then in the Register submenu, fill the items with the values below and press "Register".



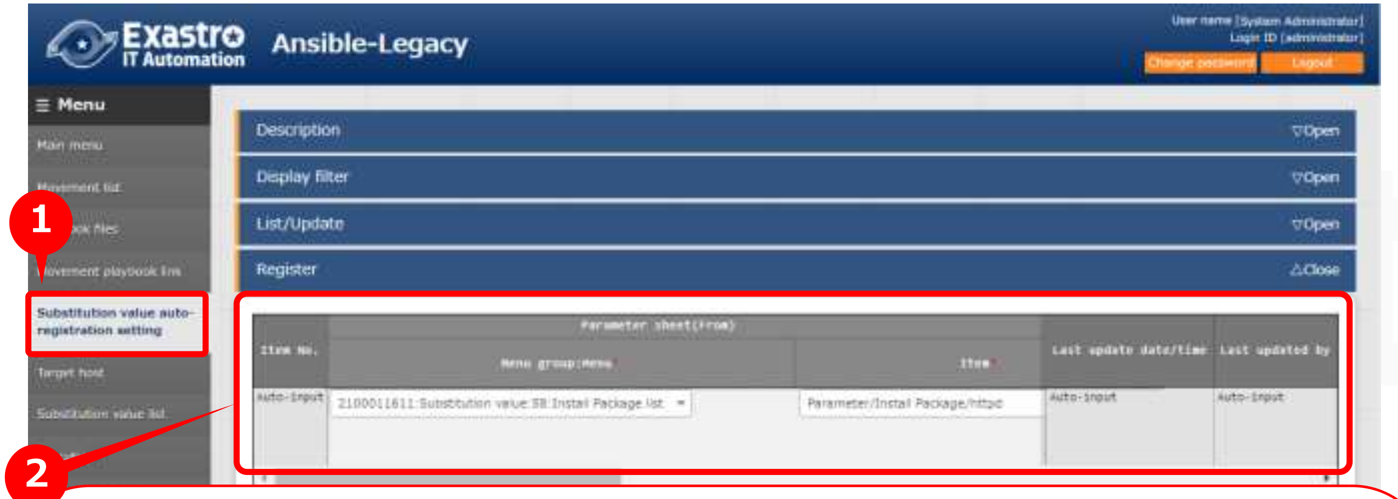
Package Flag
Yes


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.

3.5 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.



Item No.	Parameter sheet(From)		Last update date/time	Last updated by
	Menu group:Menu	Item		
Auto-input	2100011611:Substitution value:38:Install Package list	Parameter/Install Package/httpd	Auto-input	Auto-input

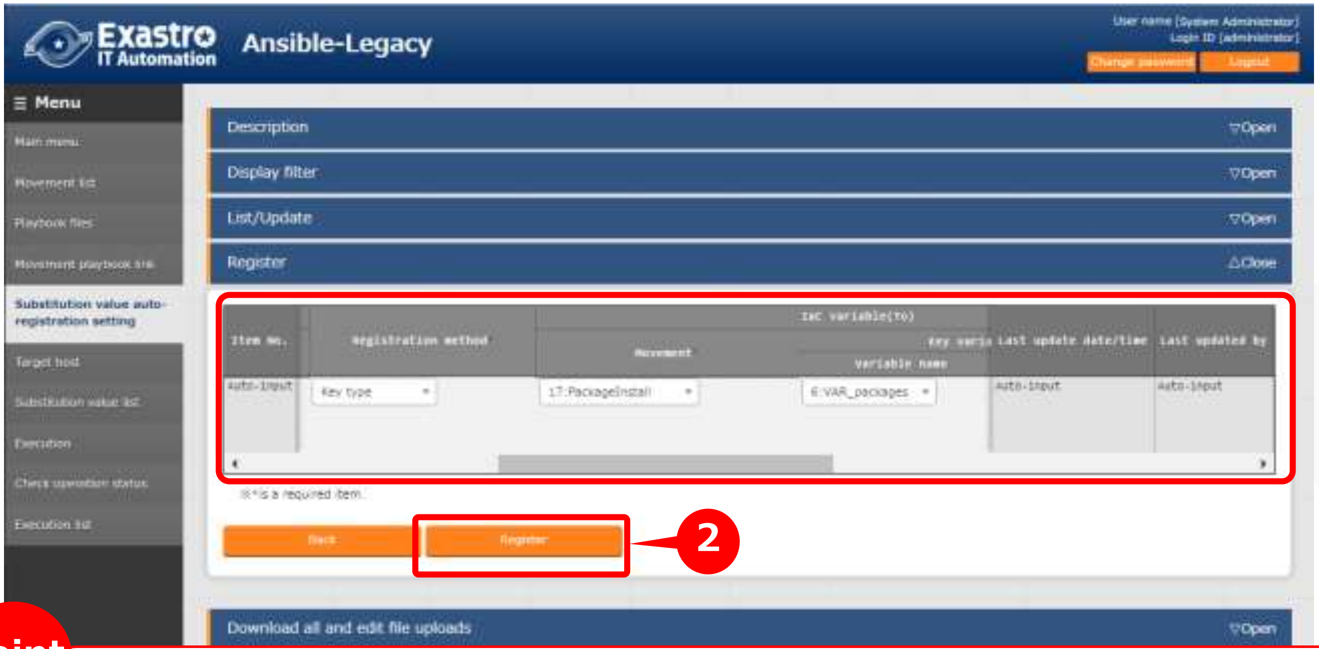
Menu group:Menu	Item	Registration method	Movement	Key Variable Variable name	Substitution order
2100011611:Susbstitution value:3:Install Package list	Parameter/Install Package/httpd	Key type	1:PackageInstall	1:VAR_packages	1
2100011611:Susbstitution value:3:Install Package list	Parameter/Install Package/mariadb-server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

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

Create "Substitution value auto-registration settings".

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



Item No.	Registration method	IaC variable(To) Movement	IaC variable(To) Variable name	Last update date/time	Last updated by
Auto-input	Key type	17:PackageInstall	6:VAR_packages	Auto-input	Auto-input

- 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.

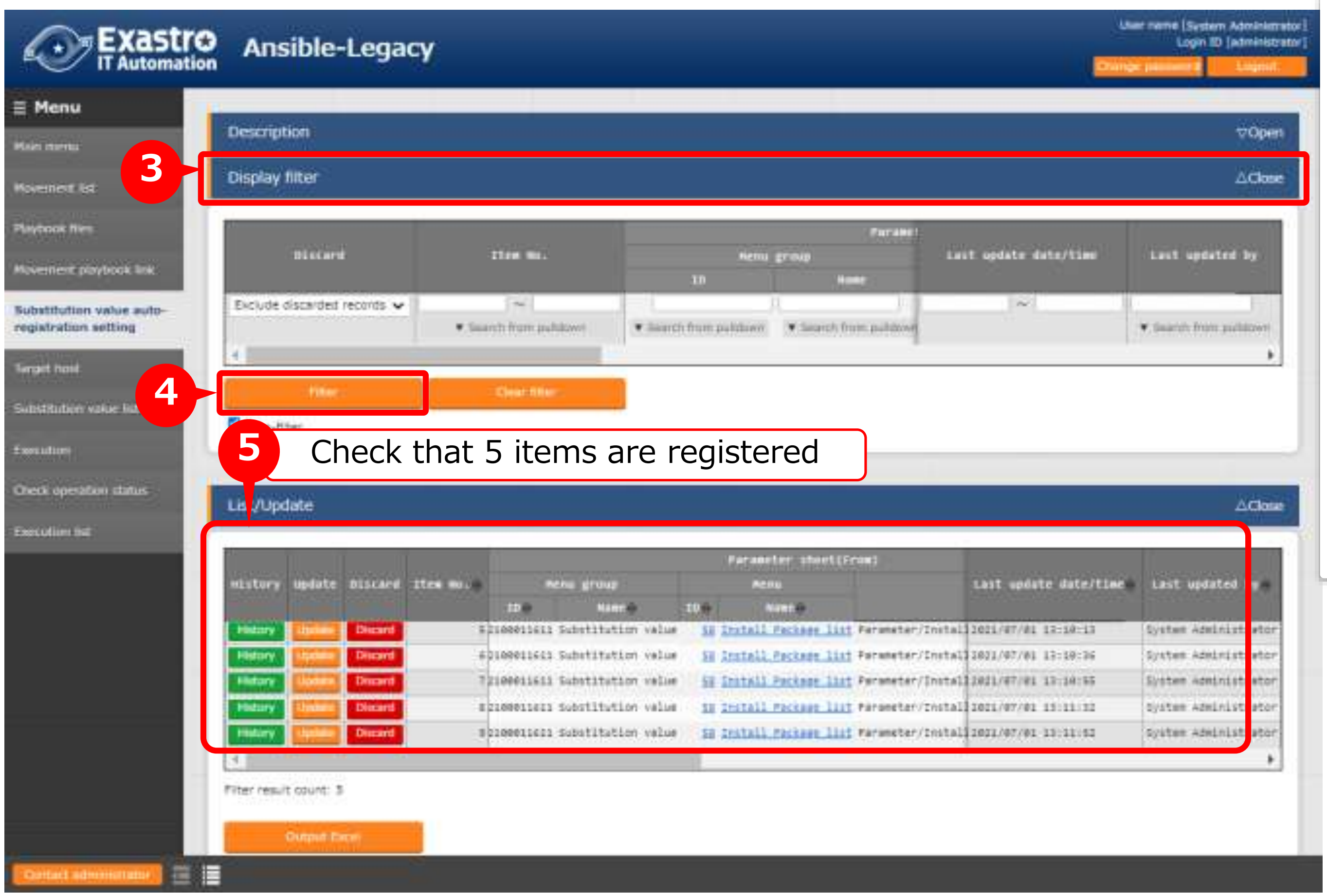
In this scenario, we want to assign the table items (column name) to the Playbook as a specific value, so we will choose the "Key" registration method.
For more information, please see the Exastro System Operation and Construction Efficiency Guide

3.5 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.



History	Update	Discard	Item no.	Parameter sheet(From)					Last update date/time	Last updated by
				Menu group		Menu				
				ID	Name	ID	Name			
History	Update	Discard	5	2100011611	Substitution value	58	Install Package list	Parameter/Install	2021/07/01 13:10:13	System Administrator
History	Update	Discard	6	2100011611	Substitution value	58	Install Package list	Parameter/Install	2021/07/01 13:10:36	System Administrator
History	Update	Discard	7	2100011611	Substitution value	58	Install Package list	Parameter/Install	2021/07/01 13:10:55	System Administrator
History	Update	Discard	8	2100011611	Substitution value	58	Install Package list	Parameter/Install	2021/07/01 13:11:32	System Administrator
History	Update	Discard	9	2100011611	Substitution value	58	Install Package list	Parameter/Install	2021/07/01 13:11:52	System 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.

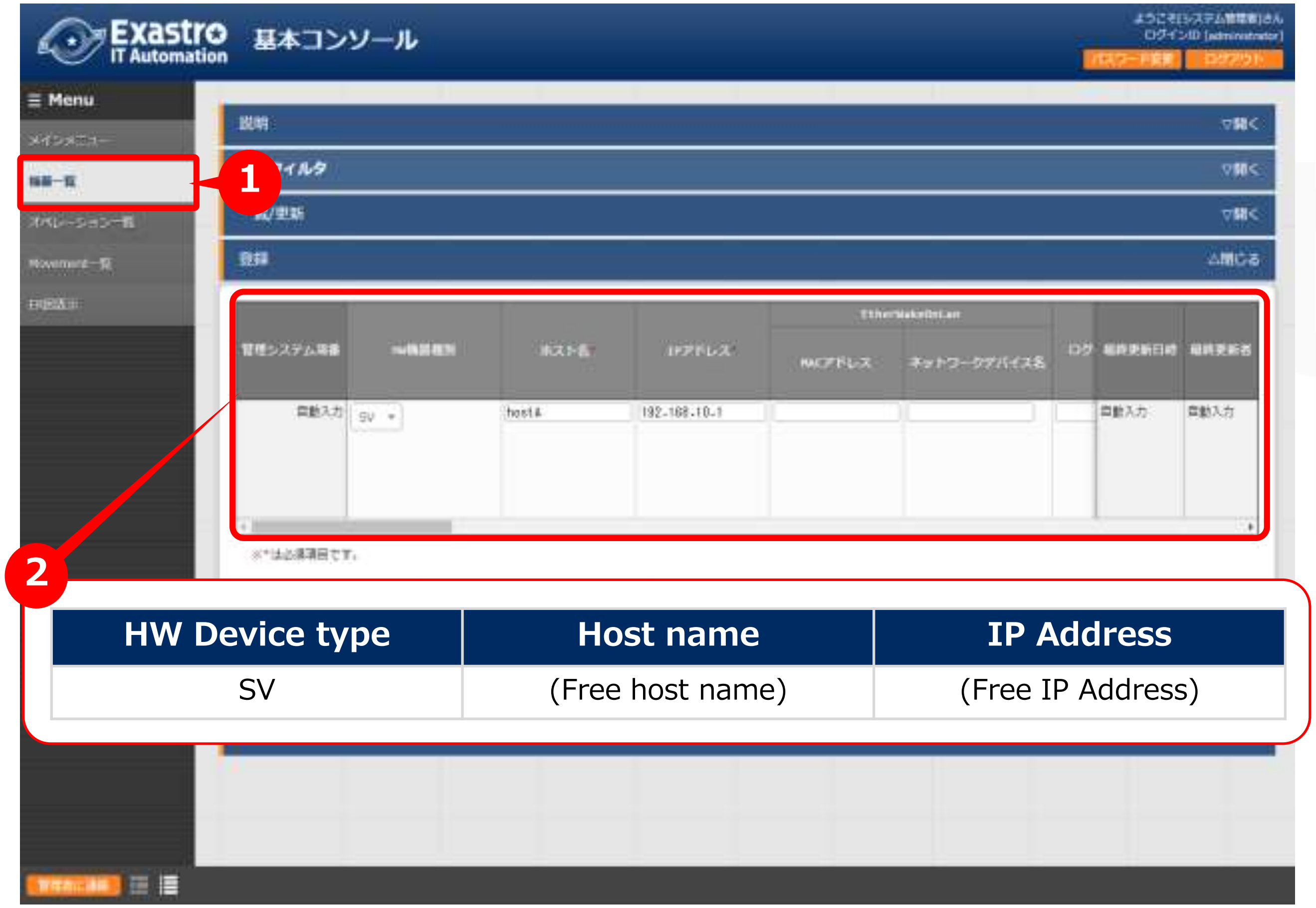
4. Execution (First time)

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

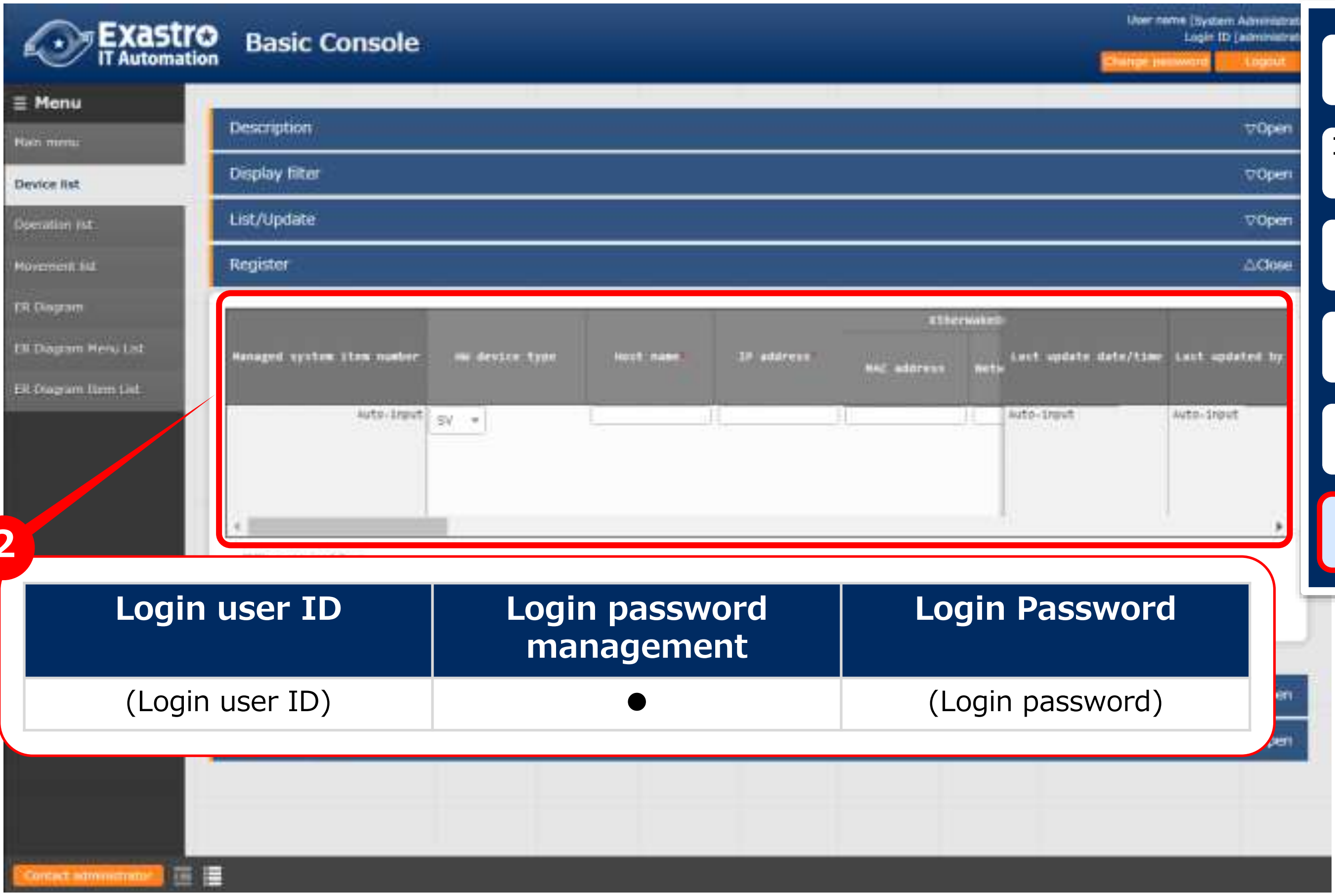
Link Parameter sheet item to Playbook variable.

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.



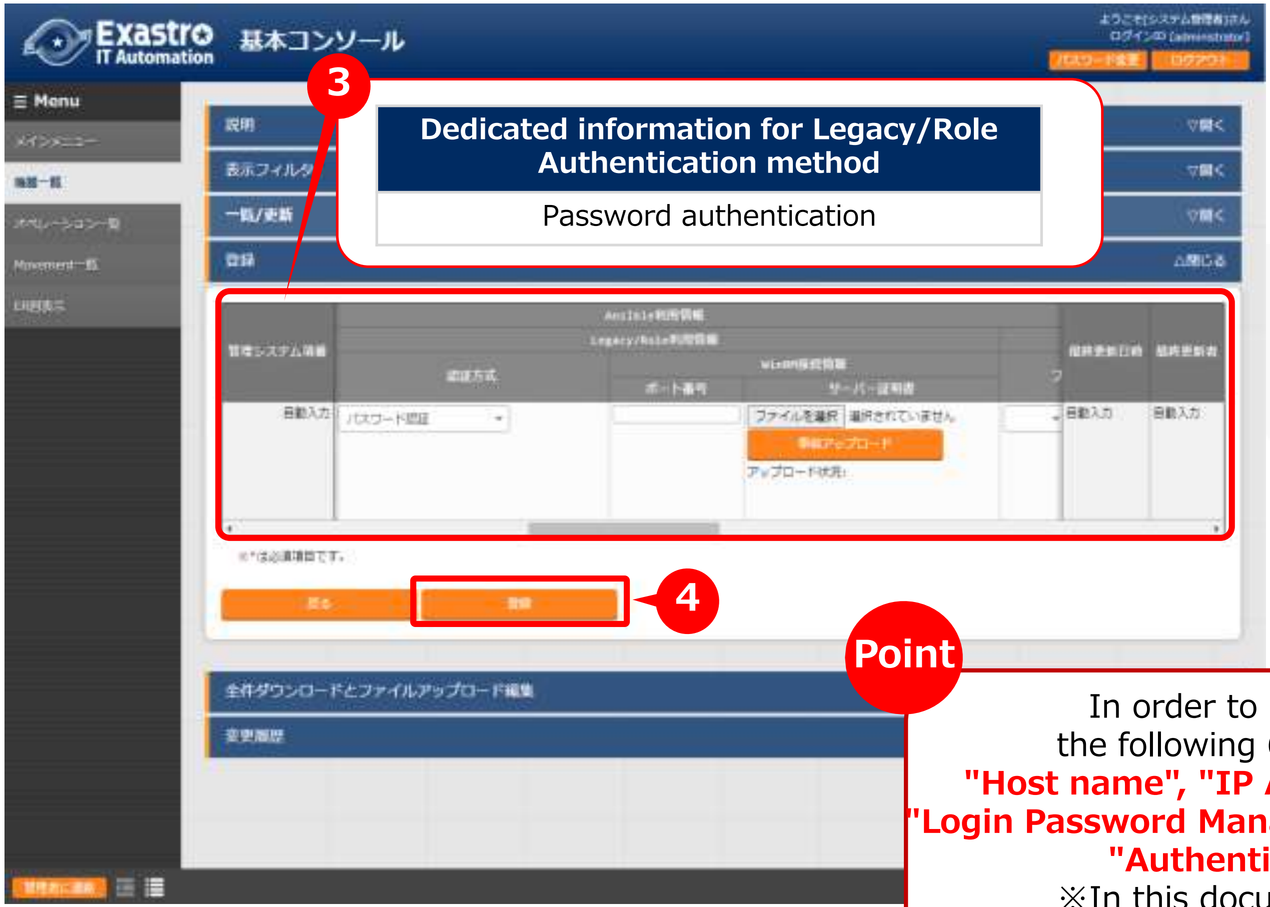
Login user ID	Login password management	Login Password
(Login user ID)	●	(Login password)

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.



Dedicated information for Legacy/Role Authentication method
Password authentication

Point

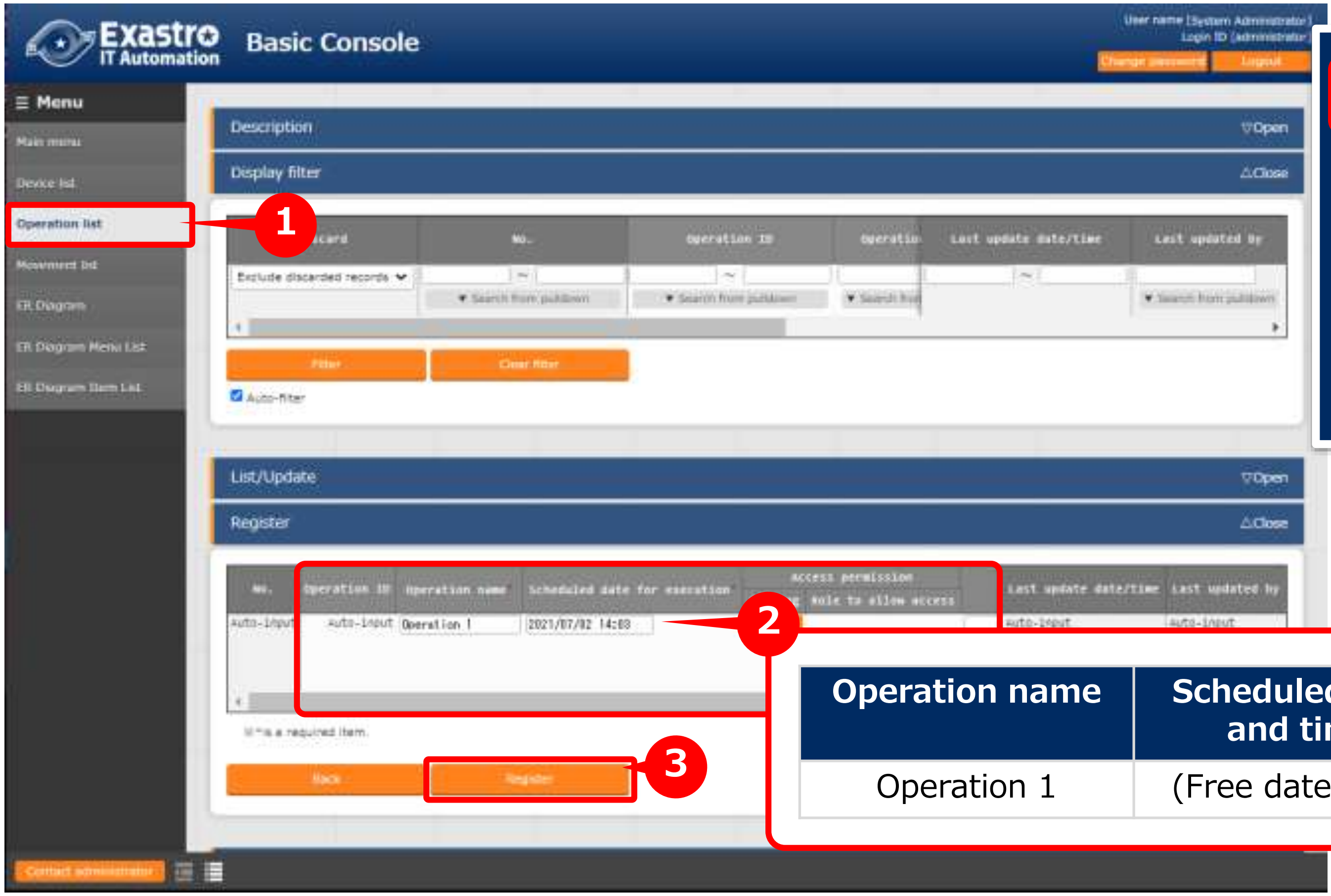
In order to run Ansible-Legacy, the following 6 items must be filled.
"Host name", "IP Address", "Login User ID", "Login Password Management", "Login password", "Authentication method※"
※In this document, "Authentication method" is written as "Password authentication".

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

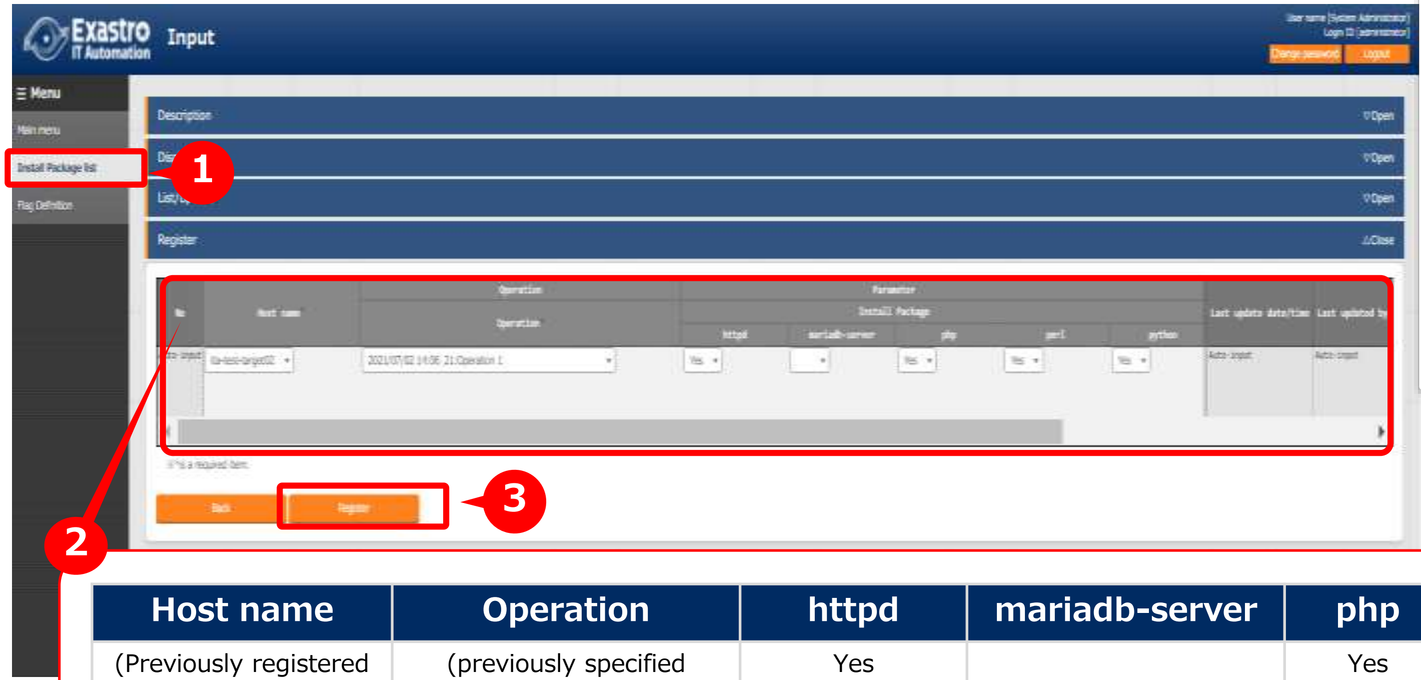
4.3 Register data to Parameter sheet (1/2)

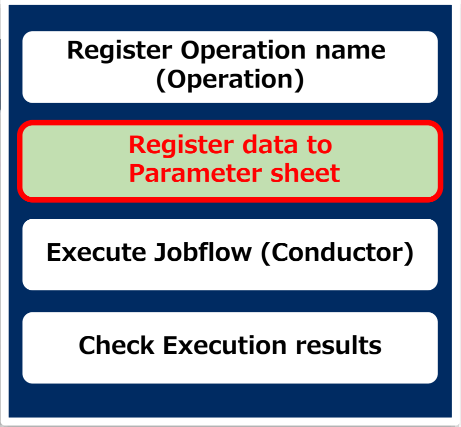
Register data to Install Package list.

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.



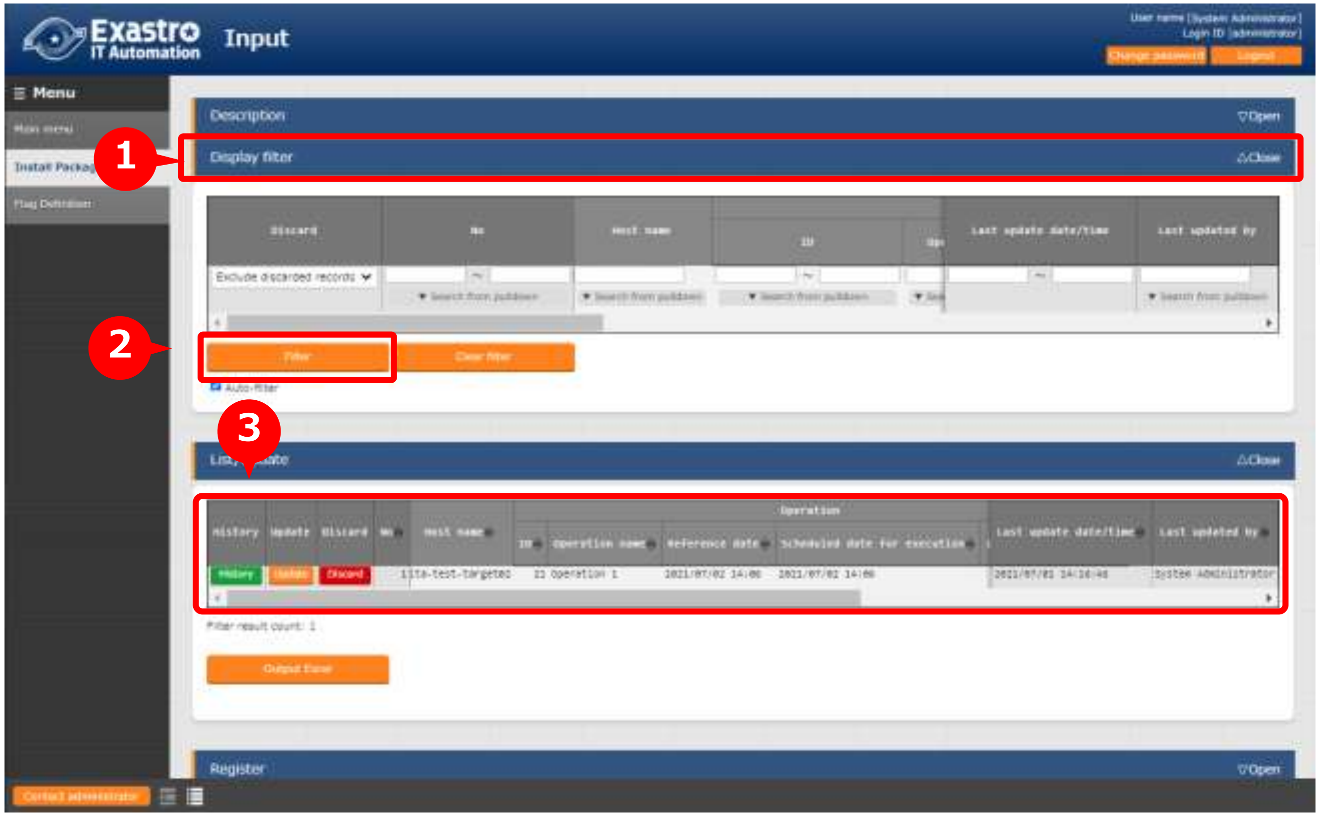

Register Operation name (Operation)
Register data to Parameter sheet
Execute Jobflow (Conductor)
Check Execution results

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

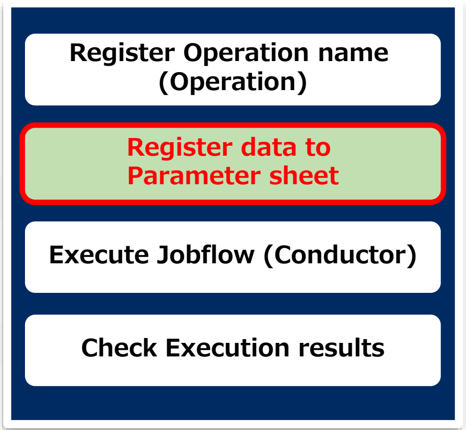
4.3 Register data to Parameter sheet (2/2)

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.



History	Update	Discard	No	Host name	Operation				Last update date/time	Last updated by
					ID	Operation name	Reference date	Scheduled date for execution		
History	Update	Discard	1	ita-test-target01	11	operation 1	2021/07/02 14:00	2021/07/02 14:00	2021/07/01 14:16:48	System Administrator

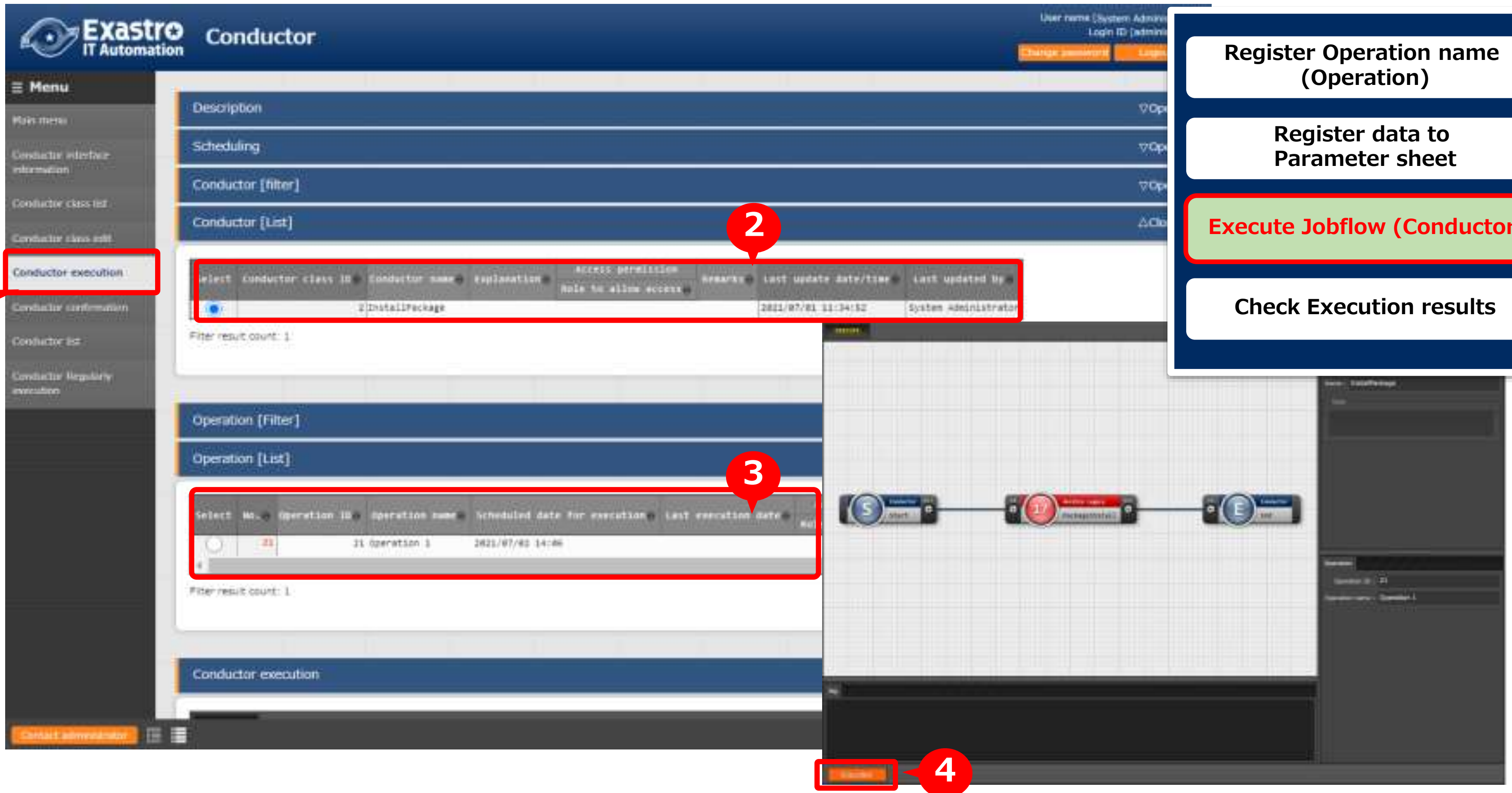


Register Operation name (Operation)
Register data to Parameter sheet
Execute Jobflow (Conductor)
Check Execution results

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.



Select	Conductor class ID	Conductor name	Explanation	Access permission Role to allow access	Remarks	Last update date/time	Last updated by
●	2	InstallPackage				2021/07/01 11:34:52	System Administrator

Filter result count: 1

Select	No.	Operation ID	Operation name	Scheduled date for execution	Last execution date
○	21	21	Operation 1	2021/07/02 14:46	

Filter result count: 1

Register Operation name (Operation)

Register data to Parameter sheet

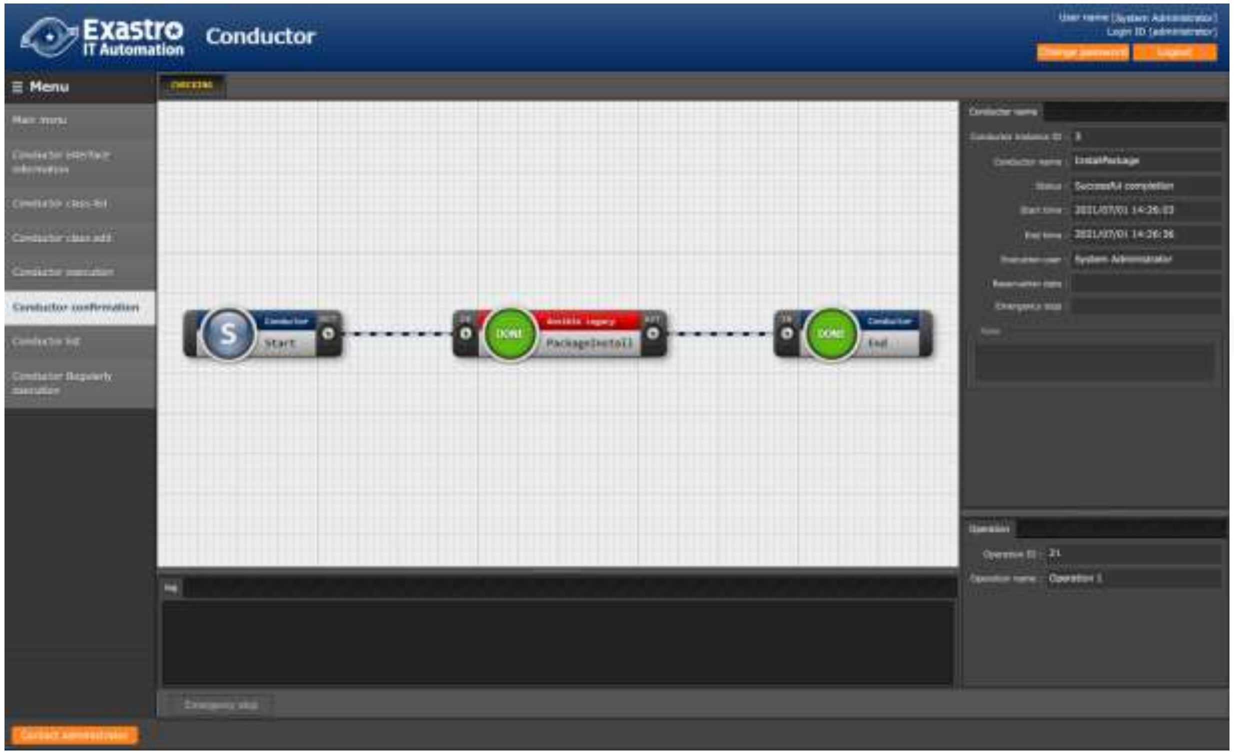
Execute Jobflow (Conductor)

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.



Conductor name

Conductor instance ID : 3
Conductor name : InstallPackage
Status : Successful completion
Start time : 2021/07/01 14:26:03
End time : 2021/07/01 14:26:36
Execution user : System Administrator
Reservation date :
Emergency stop :

Operation

Operation ID : 21
Operation name : Operation 1

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

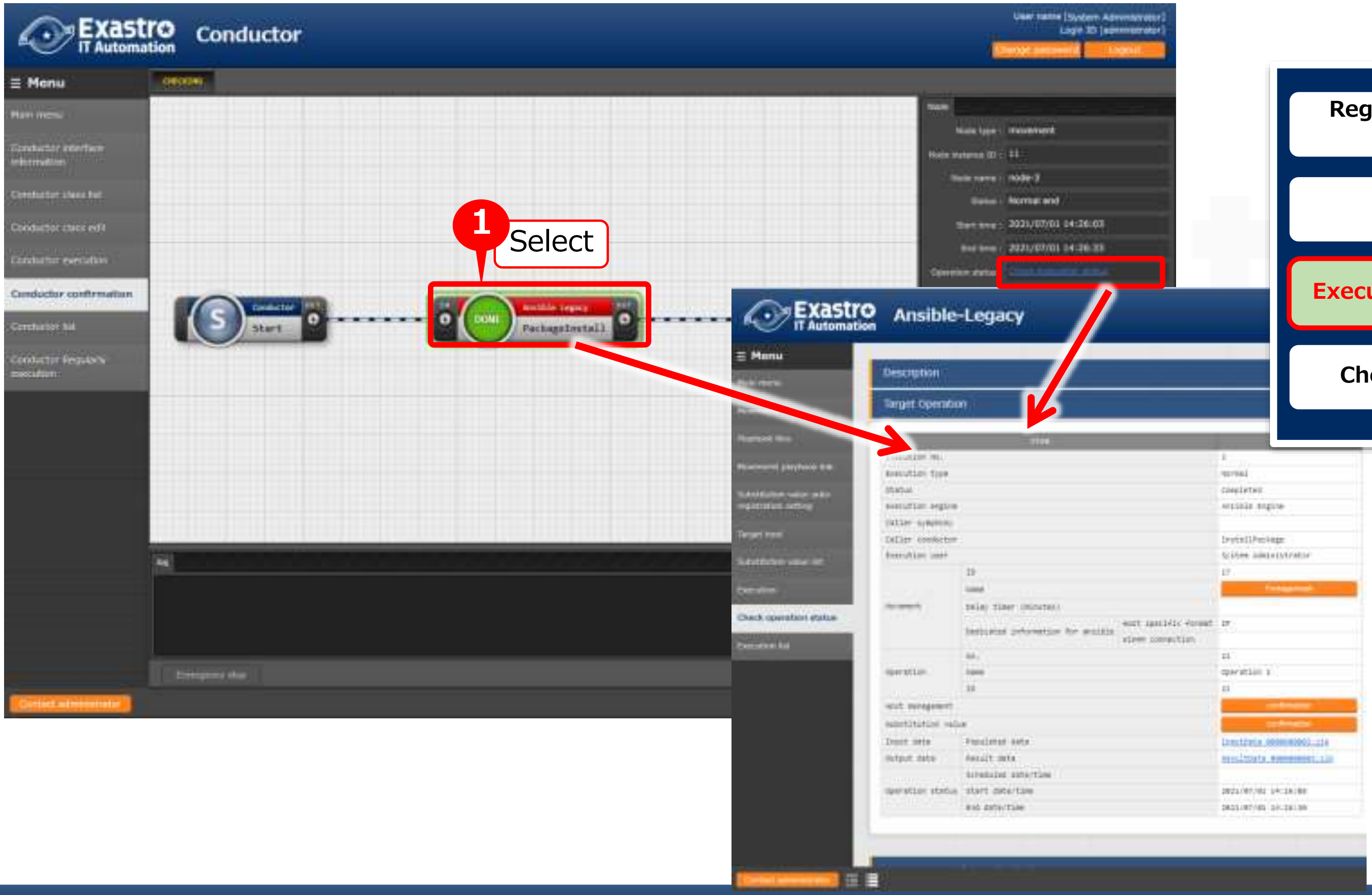
Point

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.



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 : Check operation status

Item			
Execution No.			1
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
		Winrm connection	
Operation	No.		21
	Name		operation 1
	ID		21
Host management			confirmation
Substitution value			confirmation
Input data	Populated data		InputData_0000000001.zip
Output data	Result data		ResultData_0000000001.zip
Operation status	Scheduled date/time		
	Start date/time		2021/07/01 14:26:03
	End date/time		2021/07/01 14:26:33

Register Operation name (Operation)

Register data to Parameter sheet

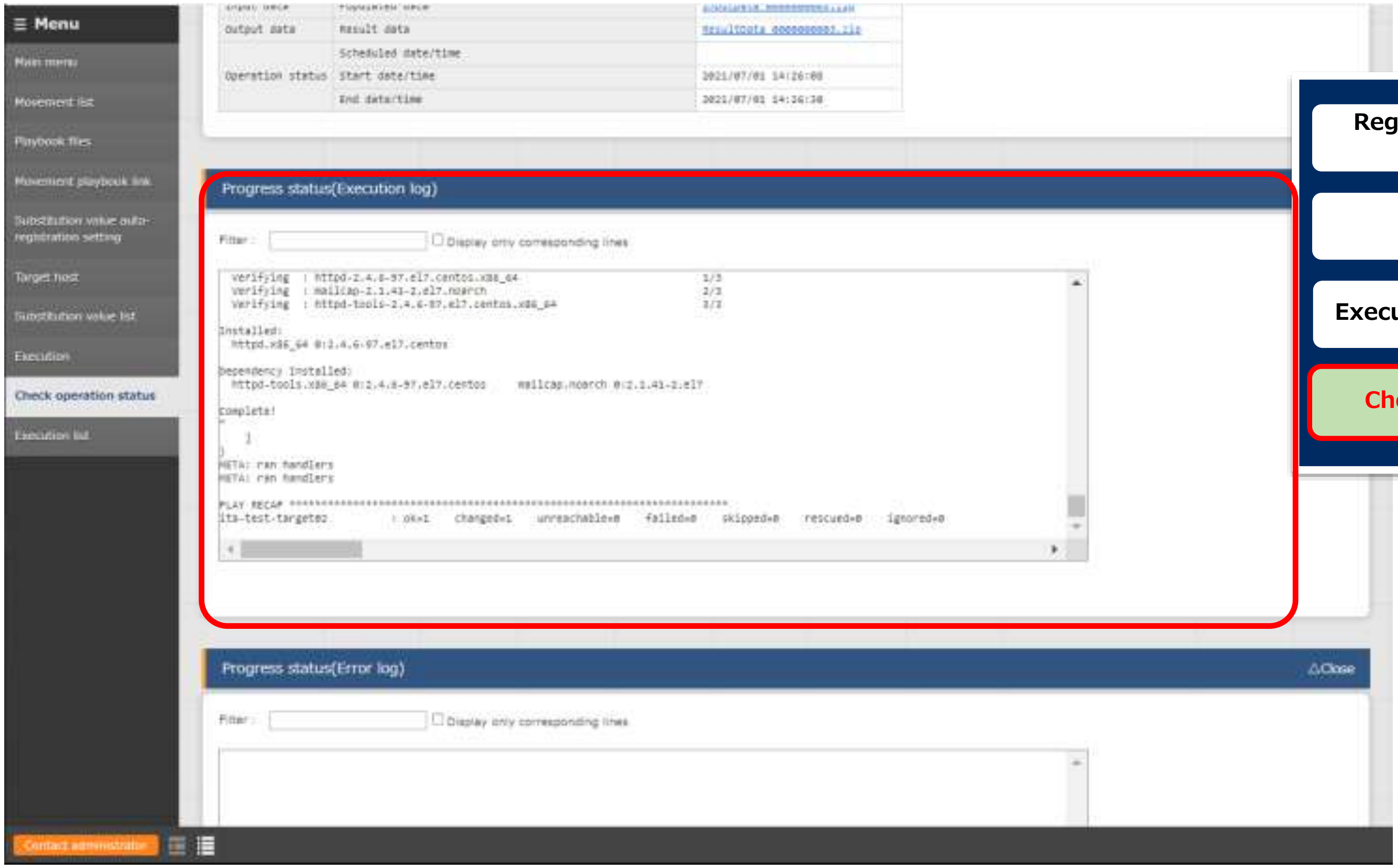
Execute Jobflow (Conductor)

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.

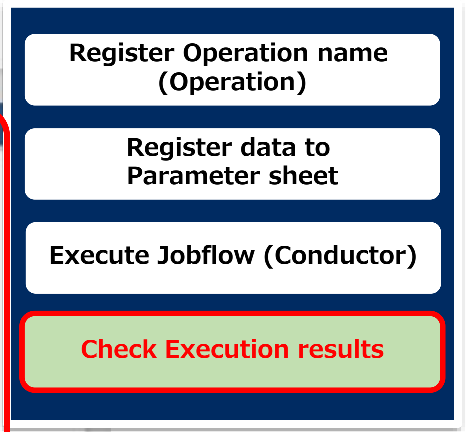


Output data	Result data	ResultData_0000000003.zip
Operation status	Scheduled date/time	
	Start date/time	2021/07/01 14:26:08
	End date/time	2021/07/01 14:26:38

Filter : ☐ Display only corresponding lines

```
  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
```

Filter : ☐ Display only corresponding lines



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

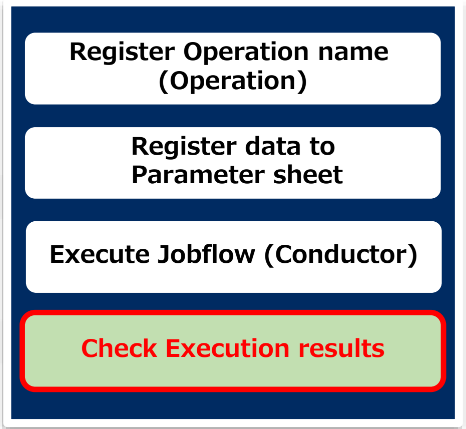
4.5 Check Execution results (2/3)

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        php.x86_64 0:5.4.16-48.el7

Dependency Installed:
  httpd-tools.x86_64 0:2.4.6-97.el7.centos    libzip.x86_64 0:0.10.1-8.el7
  mailcap.noarch 0:2.1.41-2.el7               php-cli.x86_64 0:5.4.16-48.el7
  php-common.x86_64 0:5.4.16-48.el7

Updated:
  perl.x86_64 4:5.16.3-299.el7_9          python.x86_64 0:2.7.5-90.el7

Dependency Updated:
  perl-libs.x86_64 4:5.16.3-299.el7_9     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

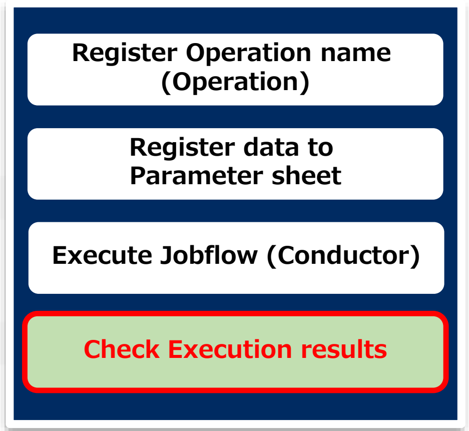
4.5 Check Execution results (3/3)

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
```

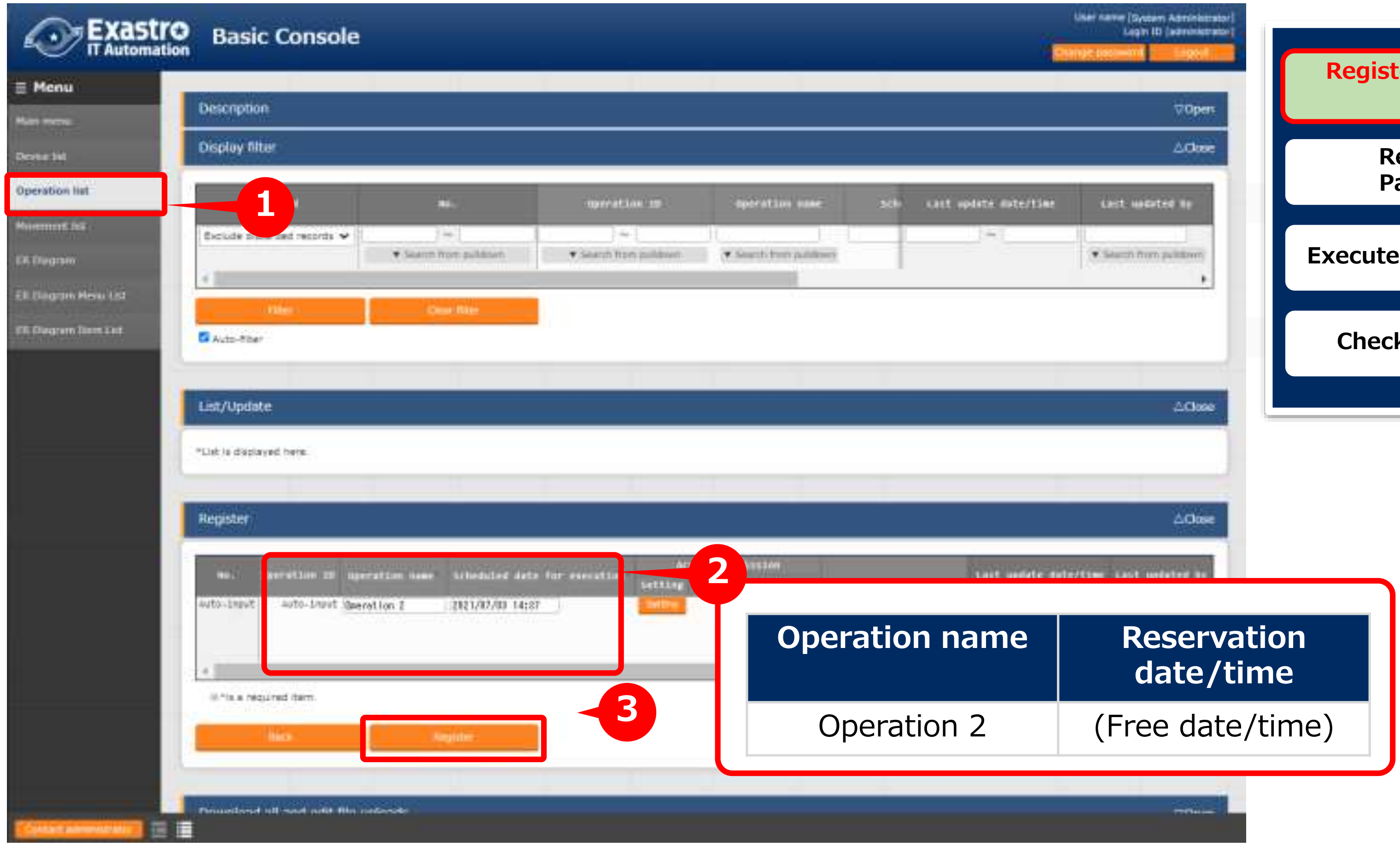

Register Operation name (Operation)
Register data to Parameter sheet
Execute Jobflow (Conductor)
Check Execution results

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.



No.	Operation ID	Operation name	Scheduled date for execution
Auto-input	Auto-input	Operation 2	2021/07/03 14:37

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

Register Operation name (Operation)

Register data to Parameter sheet

Execute Jobflow (Conductor)

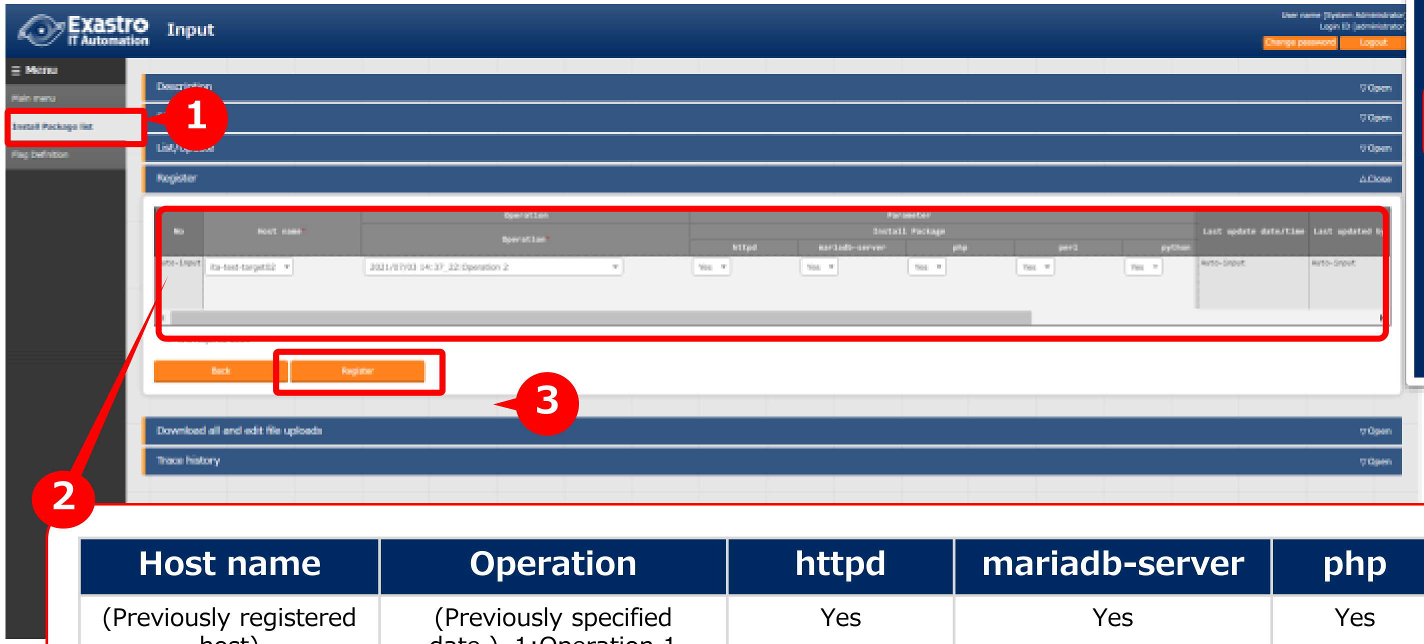
Check Execution results

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

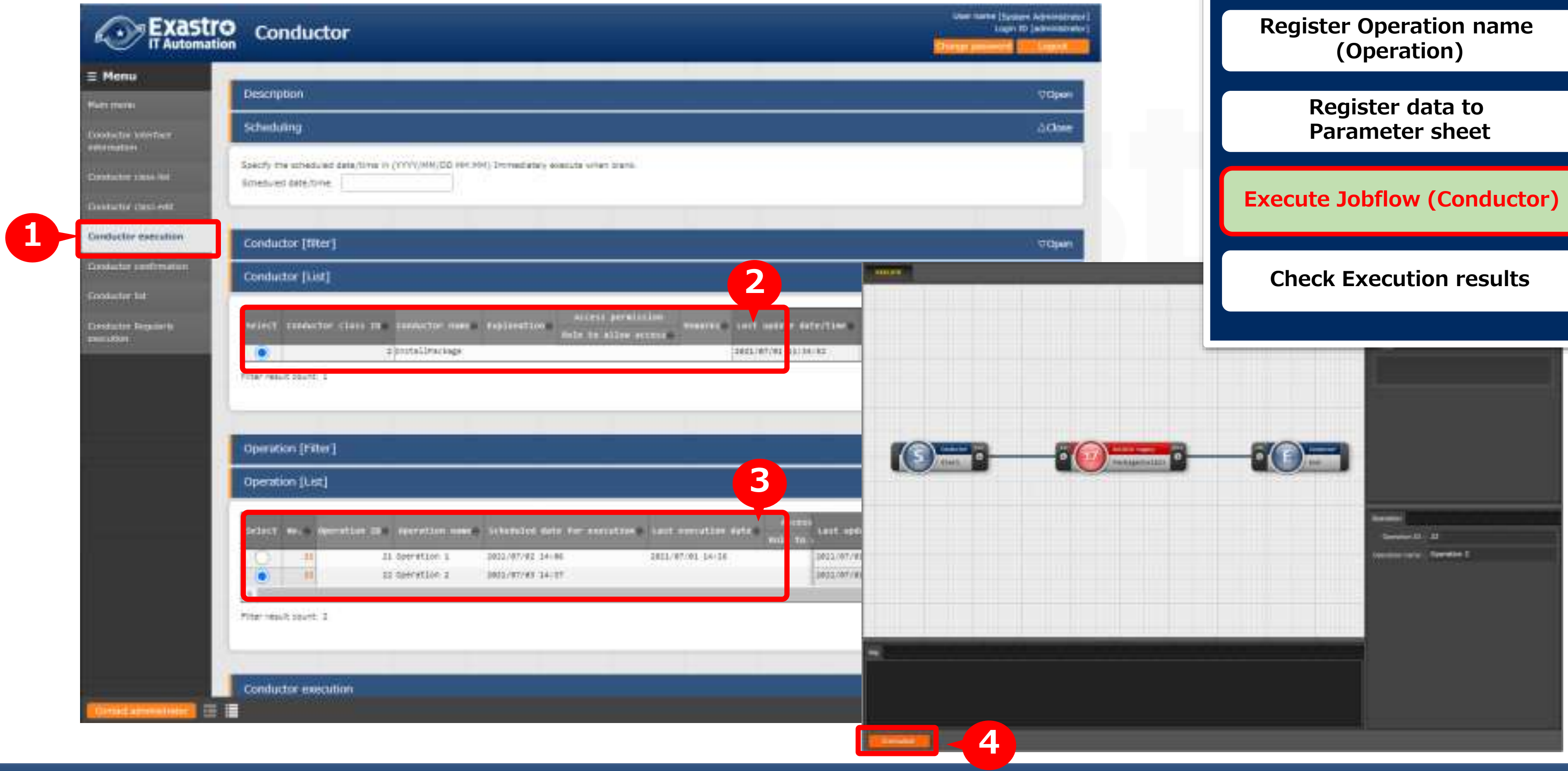
5.3 Execute Jobflow (Conductor) (1/3)

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”.



Register Operation name (Operation)

Register data to Parameter sheet

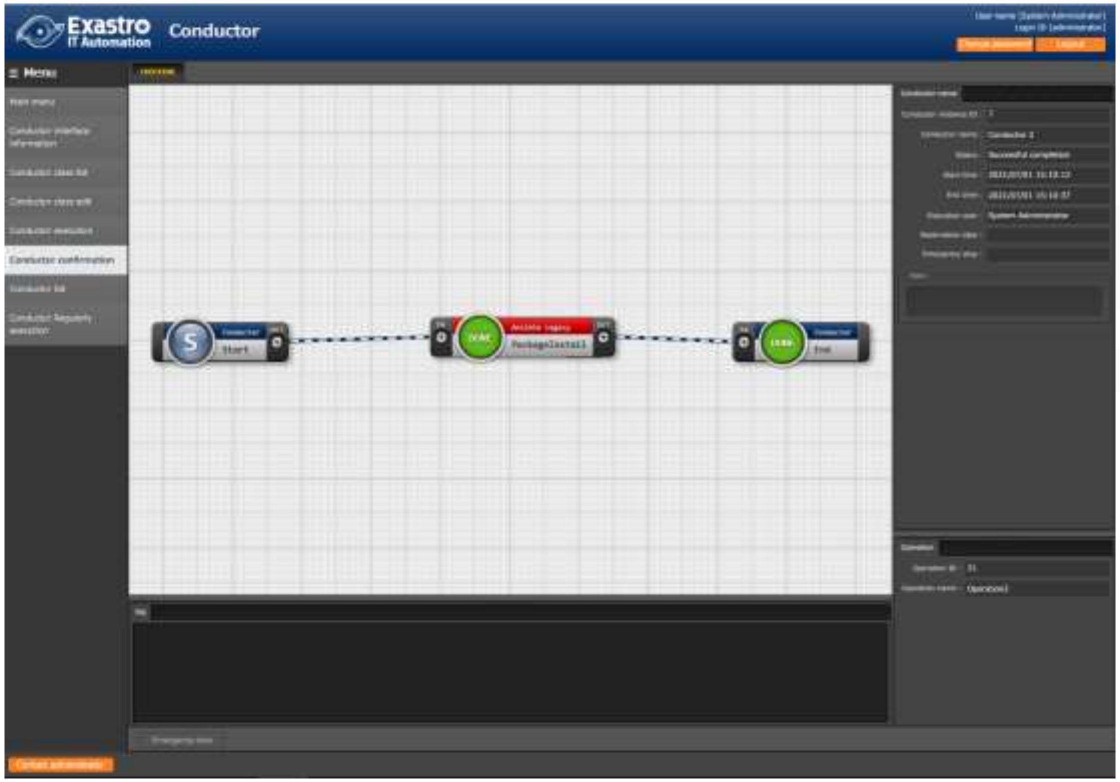
Execute Jobflow (Conductor)

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.



Register Operation name (Operation)

Register data to Parameter sheet

Execute Jobflow (Conductor)

Check Execution results

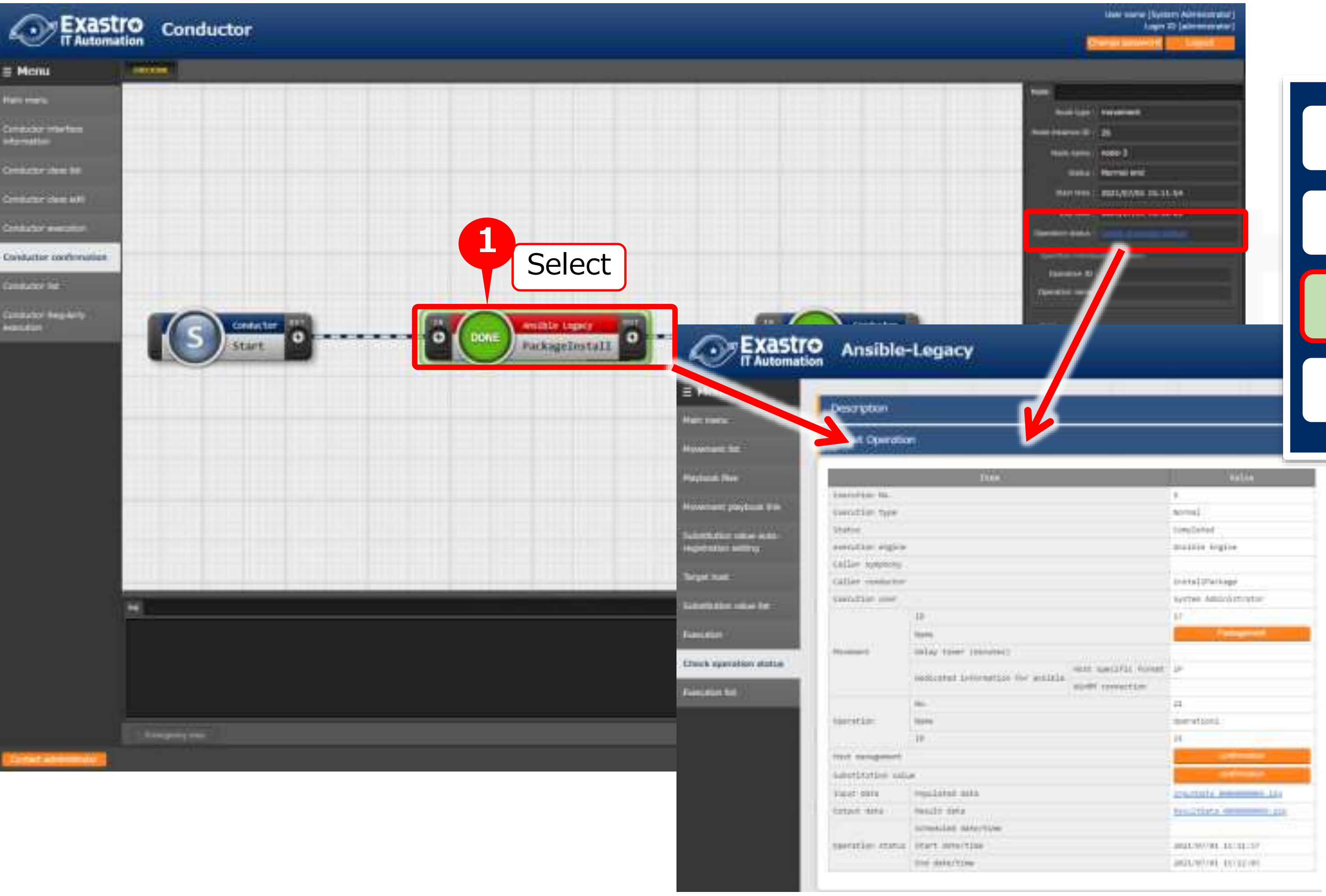
Point

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.



Register Operation name (Operation)

Register data to Parameter sheet

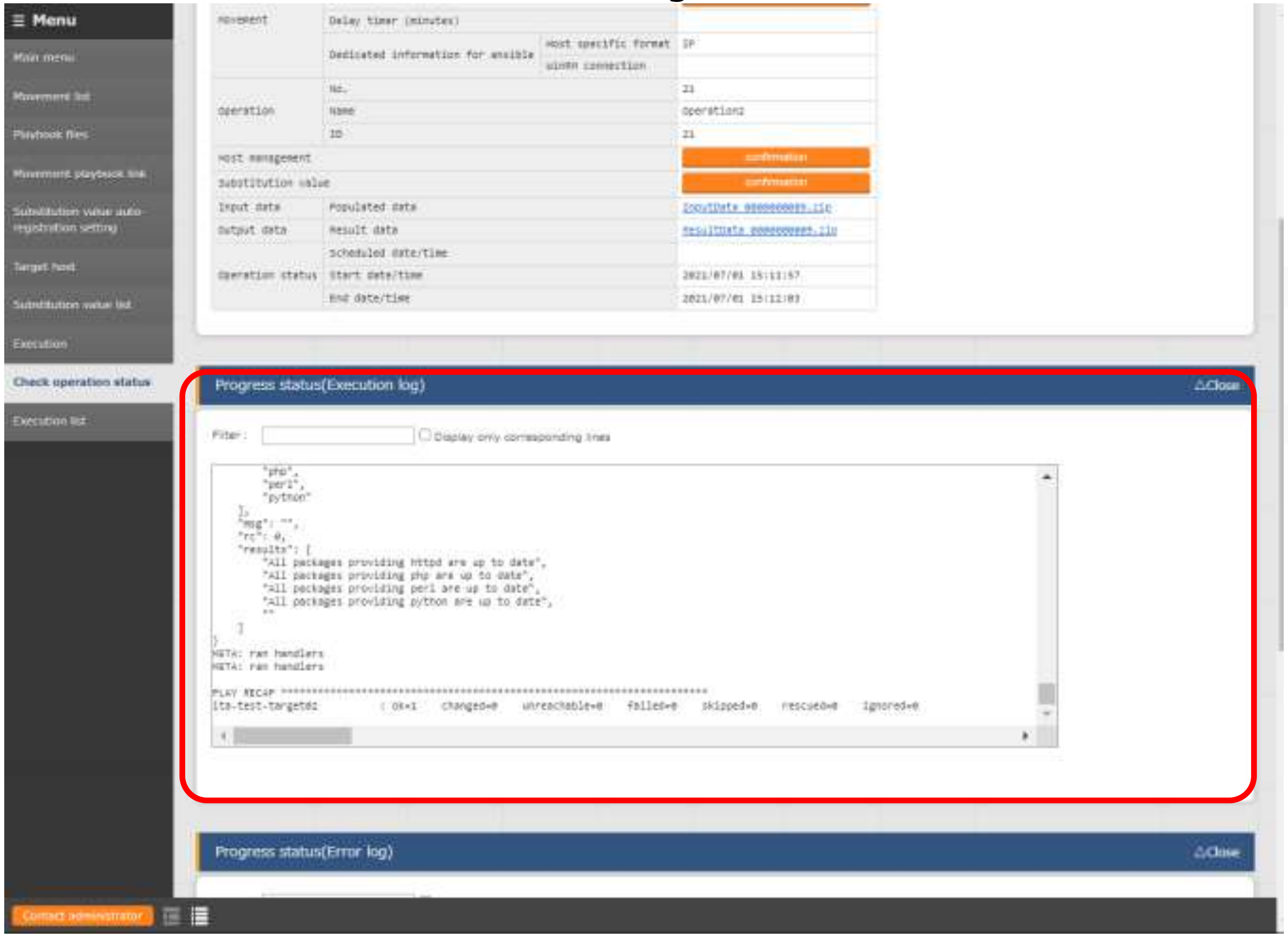
Execute Jobflow (Conductor)

Check Execution results

5.4 Check Execution results (1/2)

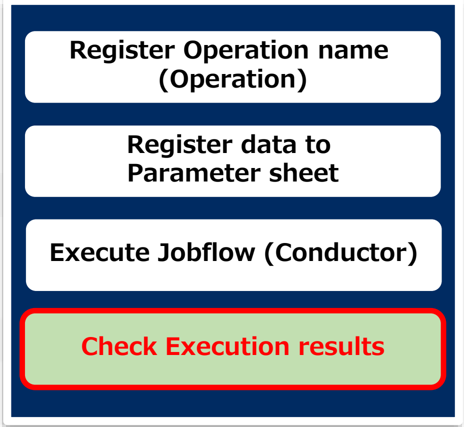
Execution results

- 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_0000000089.zip
Output data	Result data		ResultData_0000000089.zip
Operation status	Scheduled date/time		
	Start date/time		2021/07/01 15:11:57
	End date/time		2021/07/01 15:12:03

```
            "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
```



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

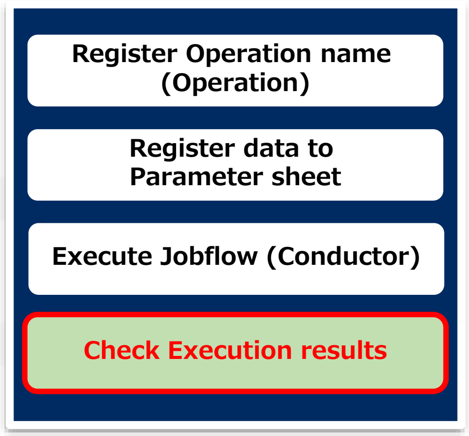
5.4 Check Execution results (2/2)

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)

```
Package httpd-2.4.6-97.el7.centos.x86_64 already installed and latest version¥
Package php-5.4.16-48.el7.x86_64 already installed and latest version¥
Package 4:perl-5.16.3-299.el7_9.x86_64 already installed and latest version¥
Package python-2.7.5-90.el7.x86_64 already installed and latest version¥

～～～～～～～～～～～～～～～～～～～～～Abbreviation～～～～～～～～～～～～～～～～～～～～～
～～～

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!
"
    ]
}
```

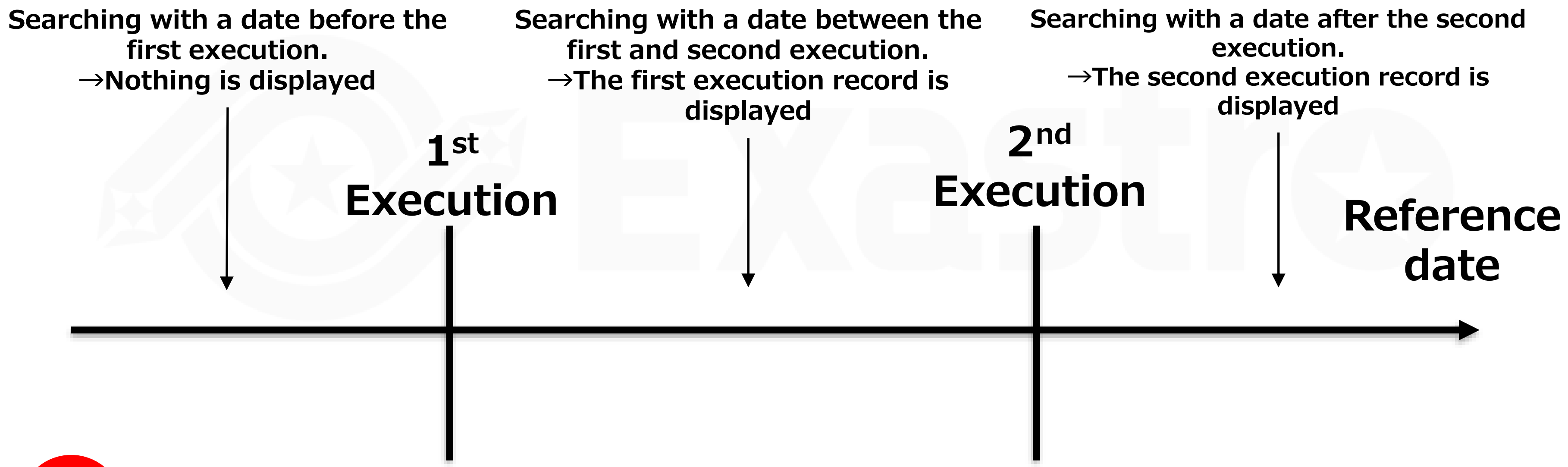

Register Operation name (Operation)
Register data to Parameter sheet
Execute Jobflow (Conductor)
Check Execution results

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


Searching with a date before the first execution.
→Nothing is displayed
Searching with a date between the first and second execution.
→The first execution record is displayed
Searching with a date after the second execution.
→The second execution record is displayed
1st Execution
2nd Execution
Reference date

Point

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

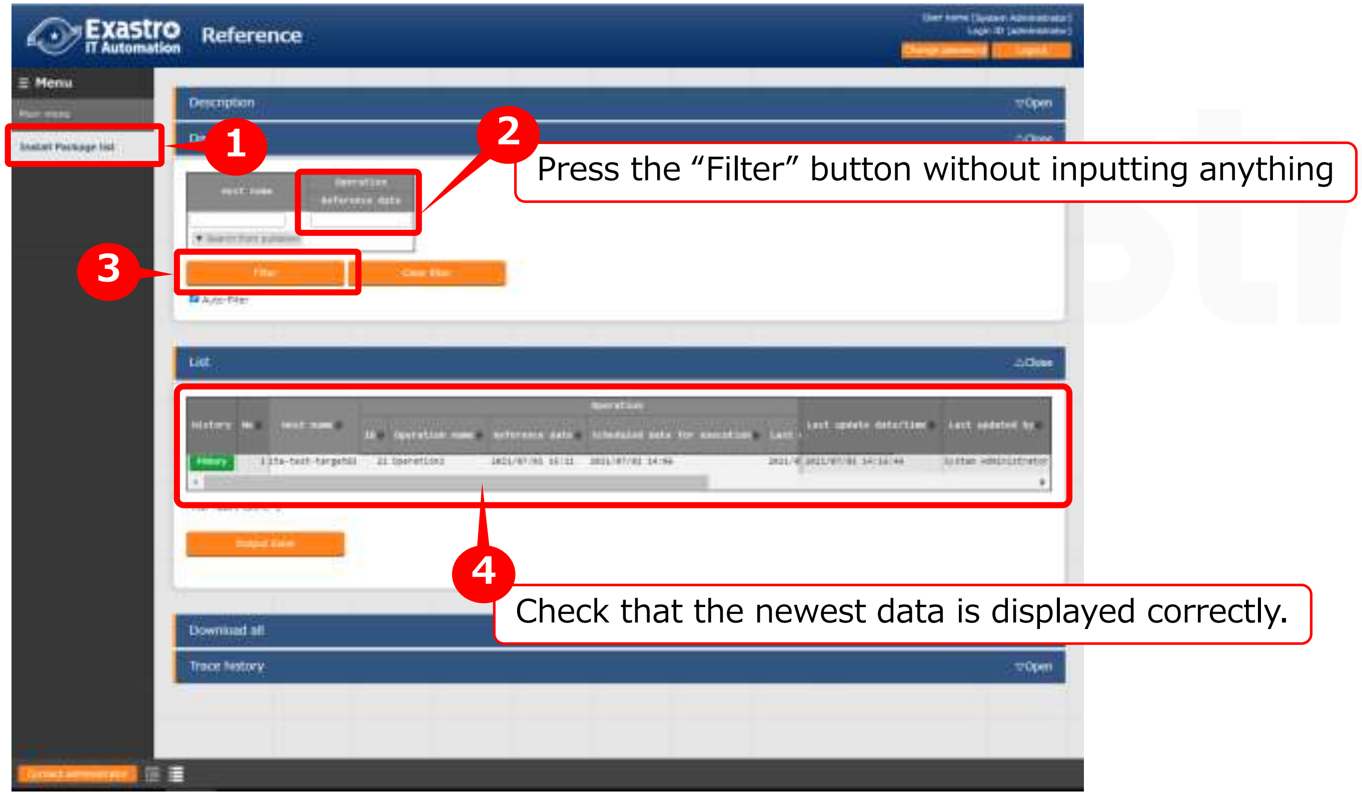
6.2 Checking the CMDB Parameter history (1/3)

History Check

● Check if the parameters are actually managed.

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

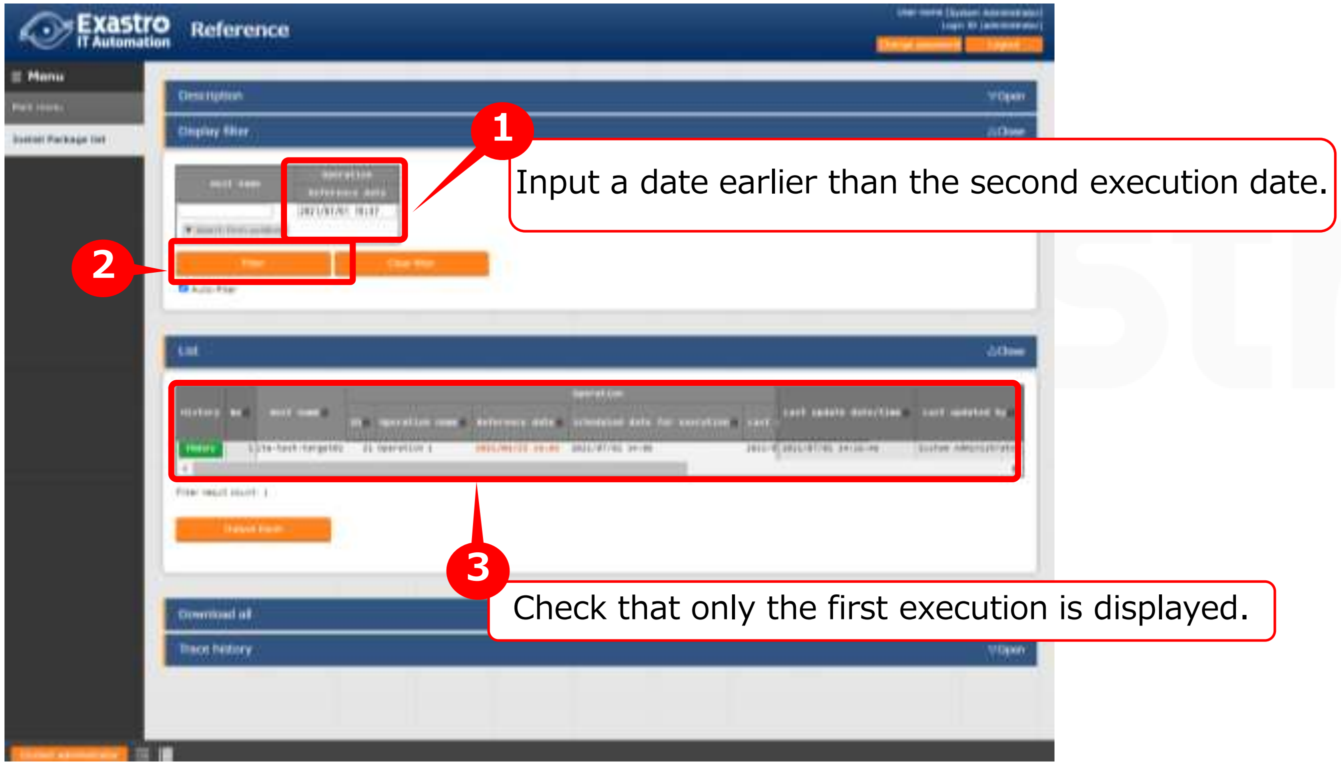
First, press the "Filter" button without inputting any filters.



6.2 Checking the CMDB Parameter history (2/3)

History Check

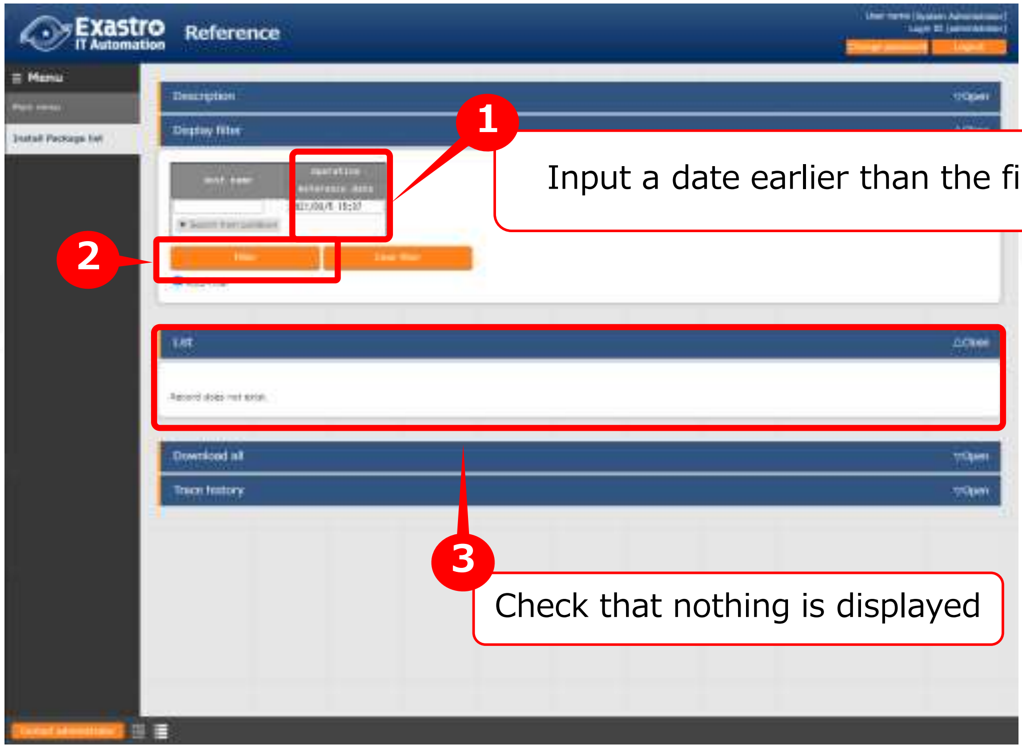
- Now, we will input a reference date that took place earlier than the second execution and filter.



Exastro IT Automation
Reference
Menu
Description
Display filter
List
Download all
Trace history
1
Input a date earlier than the second execution date.
2
3
Check that only the first execution is displayed.

History Check

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



1 Input a date earlier than the first execution

2

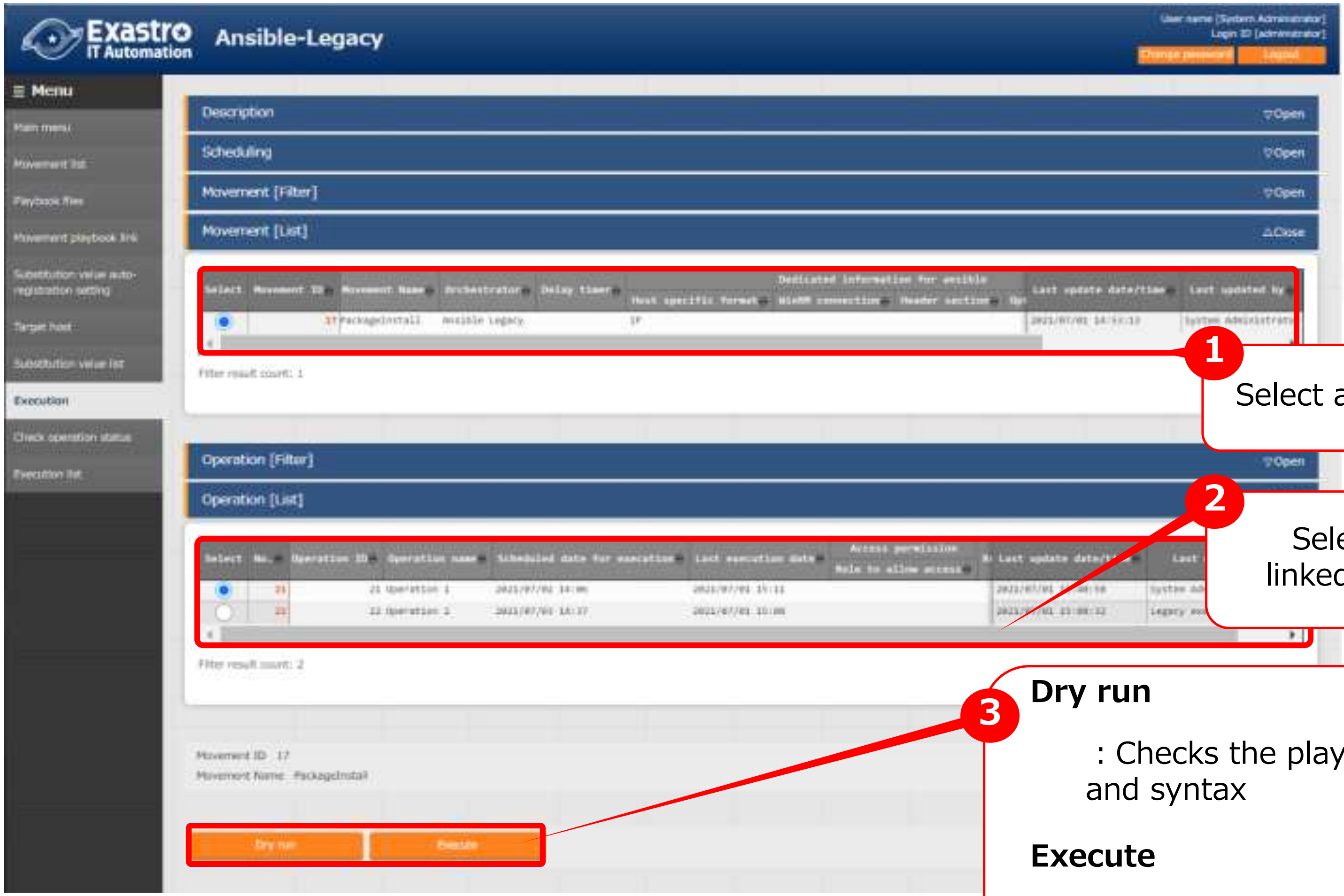
3 Check that nothing is displayed

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.



Movement [List]

Select	Movement ID	Movement Name	Orchestrator	Delay timer	Dedicated information for ansible				Last update date/time	Last updated by
					Host specific format	WinRM connection	Header section	Op		
●	17	PackageInstall	Ansible Legacy		IP				2021/07/01 14:53:13	System Administrato

Filter result count: 1

Operation [List]

Select	No.	Operation ID	Operation name	Scheduled date for execution	Last execution date	Access permission	Last update date/ti	Last
						Role to allow access		
●	21	21	Operation 1	2021/07/02 14:00	2021/07/01 15:11		2021/07/01 [illegible]	System Ad
○	22	22	Operation 2	2021/07/01 14:37	2021/07/01 15:08		2021/07/01 15:08:32	Legacy exe

Filter result count: 2

Movement ID 17
Movement Name PackageInstall

1 Select a created Movement

2 Select an operation linked to the Movement

3 **Dry run**

: Checks the playbook's connectivity and syntax

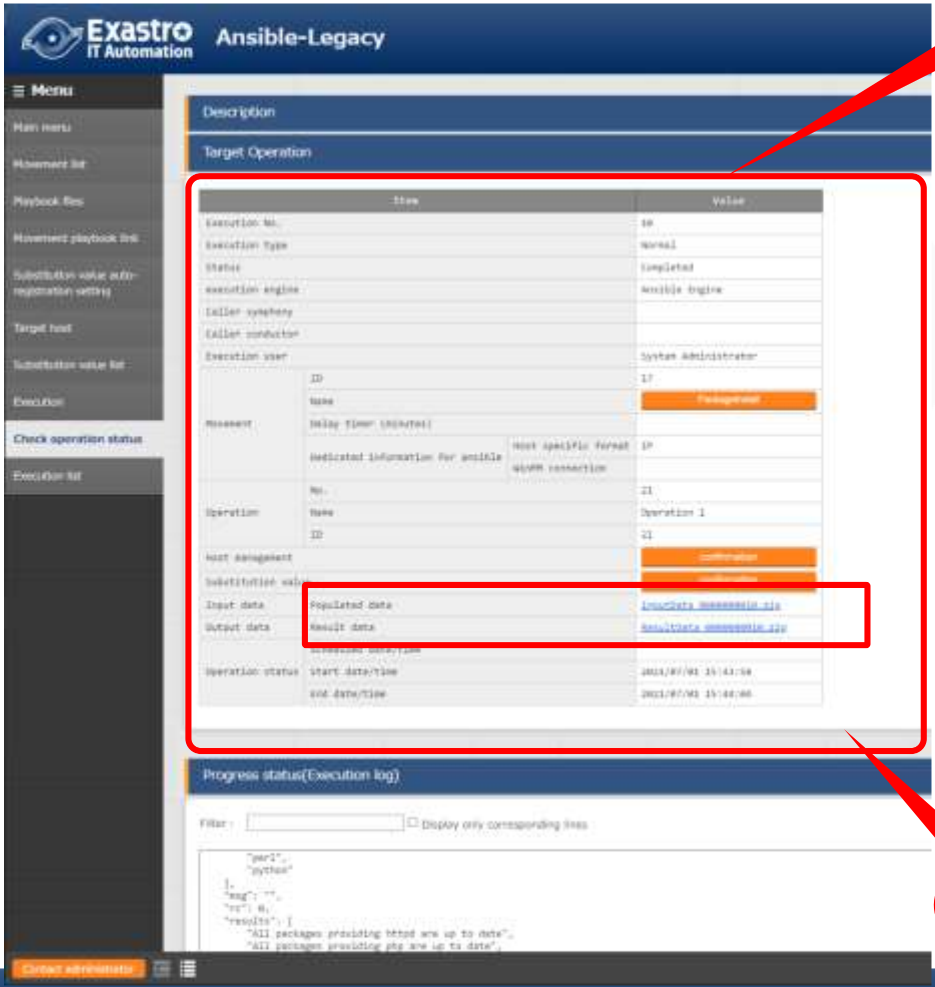
Execute

: 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.

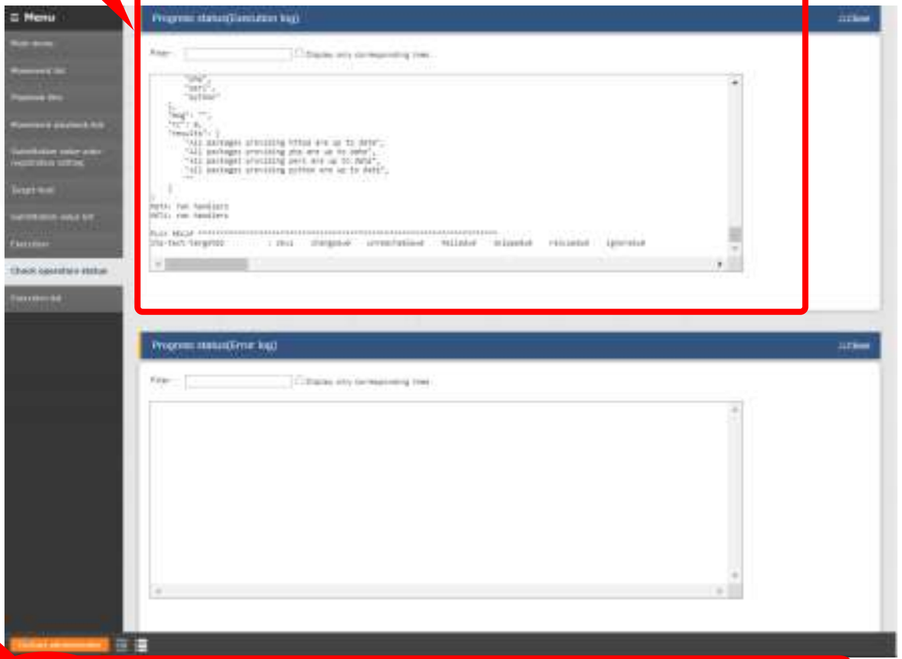


Item			Value
Execution No.			10
Execution Type			Normal
Status			Completed
execution engine			Ansible Engine
Caller symphony			
Caller conductor			
Execution user			System Administrator
Movement	ID		17
	Name		[illegible]
	Delay timer (minutes)		
	Dedicated information for ansible	Host specific format	IP
		WinRM connection	
Operation	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
Operation status	Scheduled date/time		
	Start date/time		2021/07/01 15:43:58
	End date/time		2021/07/01 15:44:00

```
        "perl",
        "python"
    ],
    "msg": "",
    "rc": 0,
    "results": [
        "All packages providing httpd are up to date",
        "All packages providing php are up to date",
```

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.


Menu
Check operation status
Execution list
Progress status(Execution log)
Progress status(Error log)
Filter :
Display only corresponding lines

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

Exastro