IT Automation

Ansible Driver【Practice】

※In this document, “Exastro IT Automation” is described as “ITA".

Exastro IT Automation ver 1.10

Exastro developer

Table of contents

How to use this document

- **This document will cover 3 scenarios.**
  [Ansible-Legacy] [Ansible-LegacyRole] [Ansible-Pioneer]
  Users can learn how to use the different modes as well as their strengths by getting hands on experience.

  As each scenario is independent, users can choose and learn whatever chapter they want.

Chapter 1 Ansible-Legacy
Register and use Playbook (YAML file).

Chapter 2 Ansible-LegacyRole
Register and use Role package.

Chapter 3 Ansible-Pioneer
Register and use Dialog files by using Ansible's original module provided by ITA.

Chapter 1 Ansible-Legacy

Work environment

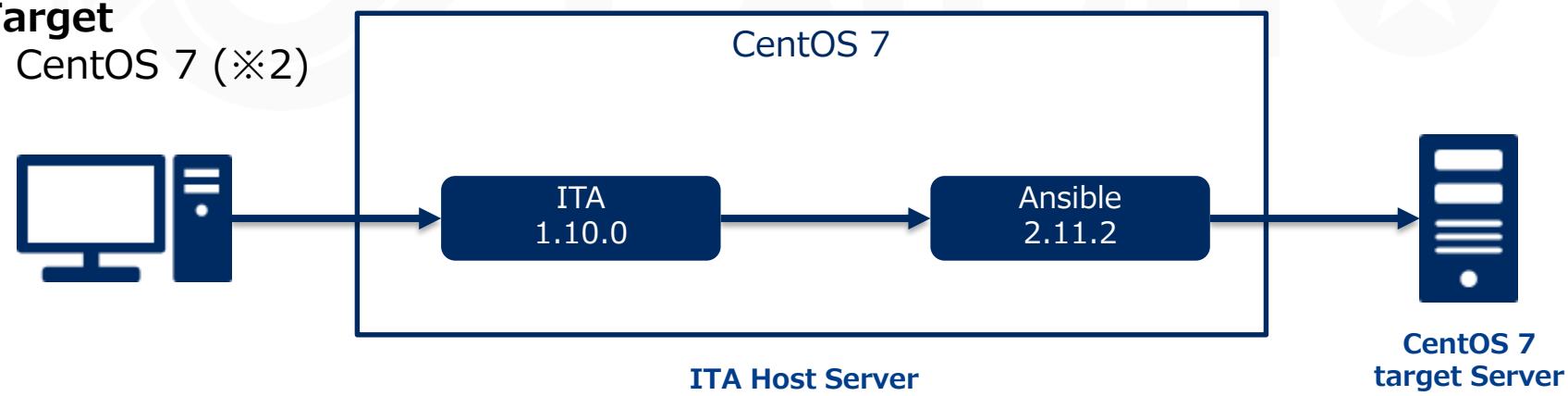
The work environment used in this document is as follows.(The same as Chapter 2 Ansible-Legacy Role)
Please prepare a server in addition to the ITA Host server. The additional server will be target for operation.

ITA Host Server

- CentOS 7 (※1)
- ITA 1.10.0
- Ansible 2.11.2

Target

- CentOS 7 (※2)



※1 In this scenario, CentOS7 will be used for the host server, but ITA can be installed on any RHEL7 and RHEL8 type OS.
※2 Any OS compatible with Ansible can be used.

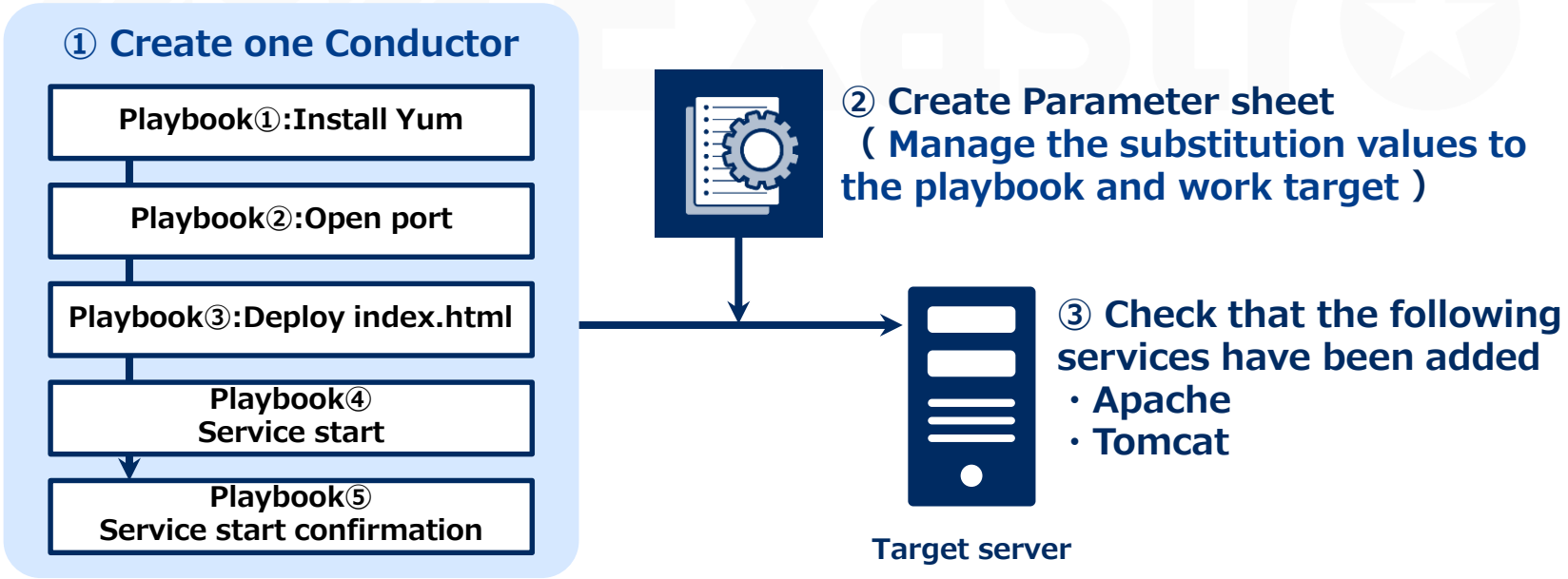
1.1 Environment and Scenario

Scenario

You can experience the reusability of Playbooks by going through a task that consists of three major steps.
① Combine Movements and create Conductor.
② Create menu and register parameters.
③ Execute the created Conductor.

Diagram

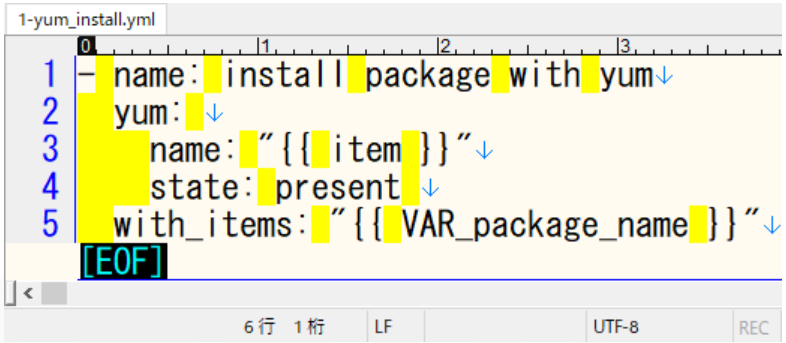
In this scenario, we will install and start both "Apache" and "Tomcat".



Playbook creation

In this section, we will create the 5 Playbooks that will be used in this scenario.

- **Conditions**
  - ✓“UTF-8” Character code
  - ✓“LF” Line code
  - ✓”yml” File type
  - ✓Enter half space after colons
  - ✓Unify indents



(E.g.) 1-yum_install.yml

```
- name: install package with yum
  yum:
    name: "{{ item }}"
    state: present
  with_items: "{{ VAR_package_name }}"
```

File name : 1-yum_install.yml

Installs the specified packages.
Multiple specific value variables are substituted in the Variables.

Playbook creation

```
- name: install firewalld
  yum:
    pkg: firewalld
    state: present

- name: start firewalld
  service:
    name: firewalld
    state: started
    enabled: yes

- name: open ports
  firewalld:
    port: "{{ VAR_port_number }}“
    state: enabled
    permanent: yes
    immediate: true
```

File name : 2-open_port.yml

Installs Firewalld, starts it and opens the specified port.

1.2 Creating files(3/4)

Playbook creation

```
- name: copy index.html
  copy:
    src: “{{ CPF_index_html }}”
    dest: /var/www/html/index.html
    owner: root
    group: root
    mode: 0644
    backup: yes
  when: ‘VAR_service_name == “httpd”’
```

File name: 3-copy_index.yml

Copies and deploys file

```
- name: start service
  service:
    name: "{{ VAR_service_name }}"
    state: started
    enabled: yes
```

File name: 4-start_service.yml

Starts the specified service.

```
- name: check if service is running and enabled
  command: 'systemctl status {{ VAR_service_name }}'
  register: command_result
  failed_when:
    - '"enabled" not in command_result.stdout'
    - '"running" not in command_result.stdout'
```

File name: 5-check_service.yml

Checks if the service has started.

Create Index.html

In this scenario, the configuration file is deployed using the "File Management Function“.

Create the following file and put it under the document route.

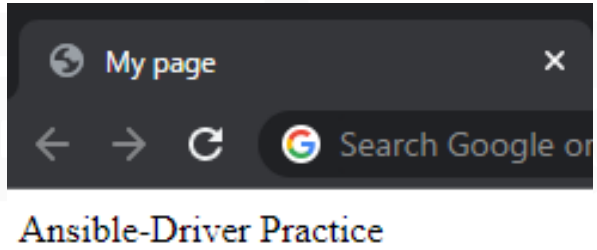
File name: index.html

```
<html>
<head>
<title>My Page</title>
</head>
<body>

Ansible-Driver Practice

<br>

</body>
</html>
```



When finished, you will see a screen that shows that the index.html is placed correctly.

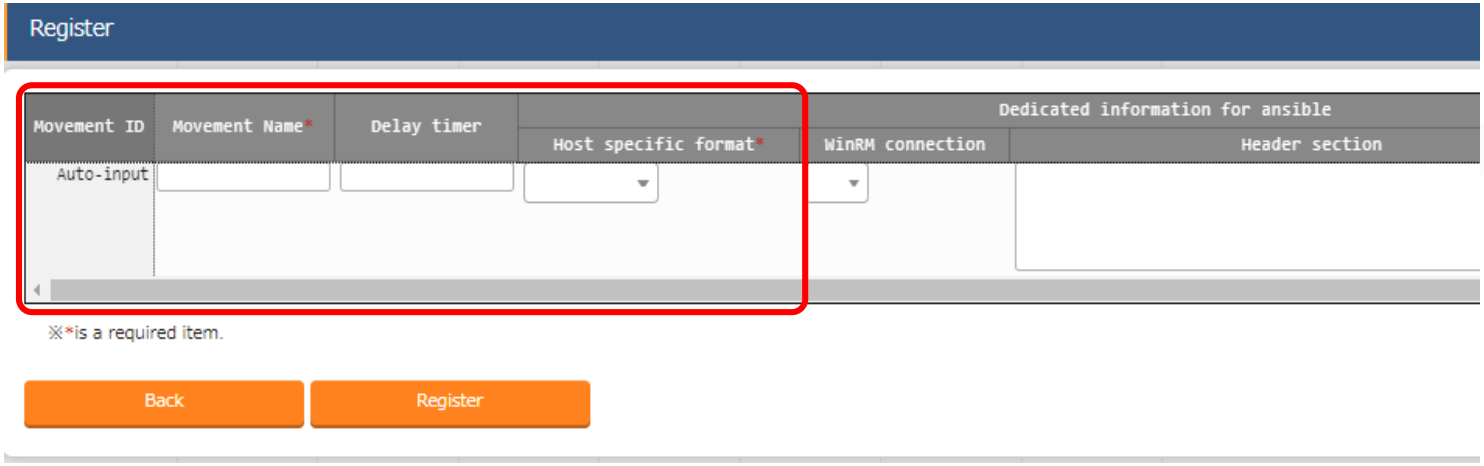
Create Movement

Register the Movements that is going to be associated with the playbooks.

Menu : **Ansible-Legacy > Movement list**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



Movement name	Host specific format
Install_Packages	IP
Open_Ports	IP
Start_Service	IP

Register playbook

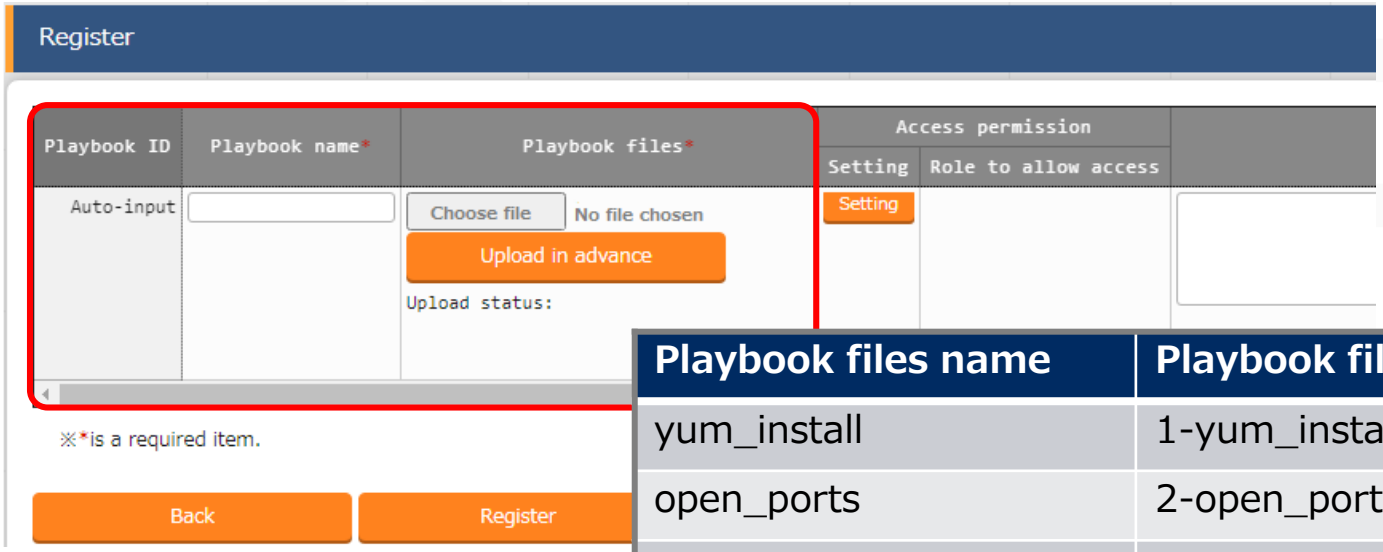
Register the playbooks we created.

Menu **: Ansible-Legacy > Playbook files**

① Click Register > Start Registration.

② Select a playbook from "Browse" and click "Upload in advance".

③ Input the following information for each item and click "Register".



Playbook files name	Playbook files
yum_install	1-yum_install.yml
open_ports	2-open_ports.yml
deploy_config	3-deploy_config.yml
start_service	4-start_service.yml
check_service_state	5-check_service.yml

Register file

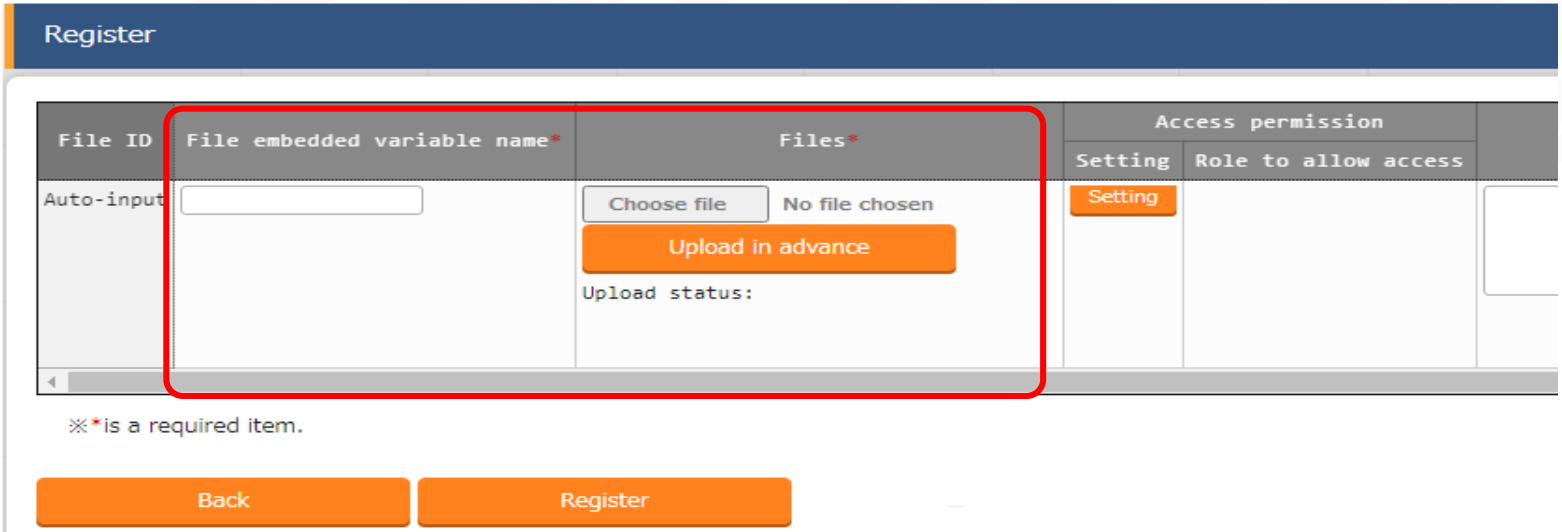
Register the httpd configuration file.

Menu **: Ansible Common > File list**

① Click Register > Start Registration.

② Select the file from "Browse" and click "Upload in advance".

③ Input the following information for each item and click "Register".



File embedded variable name	File
CPF_httpd_html	index.html

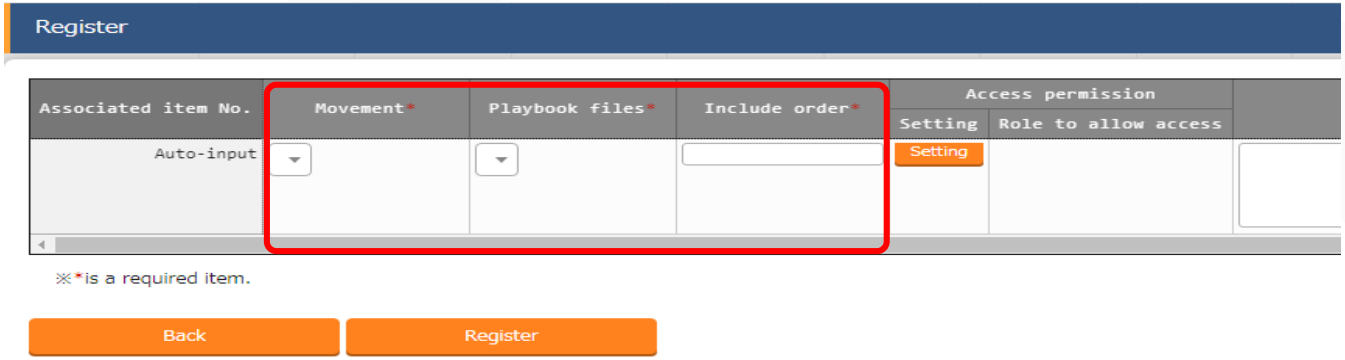
1.3 Movement configuration(4/4)

Register the playbook to Movement

Link the created Movement and Playbook files.
By dividing the contents of the operation and giving them names makes it easier to reuse them.
Menu: **Ansible-Legacy > Movement playbook link**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



Associate Table

Movement	Playbook files	Include order
Install Packages	yum_install	1
Open Ports	open_ports	1
Start Service	deploy_config	1
Start Service	start_service	2
Start Service	check_service_state	3

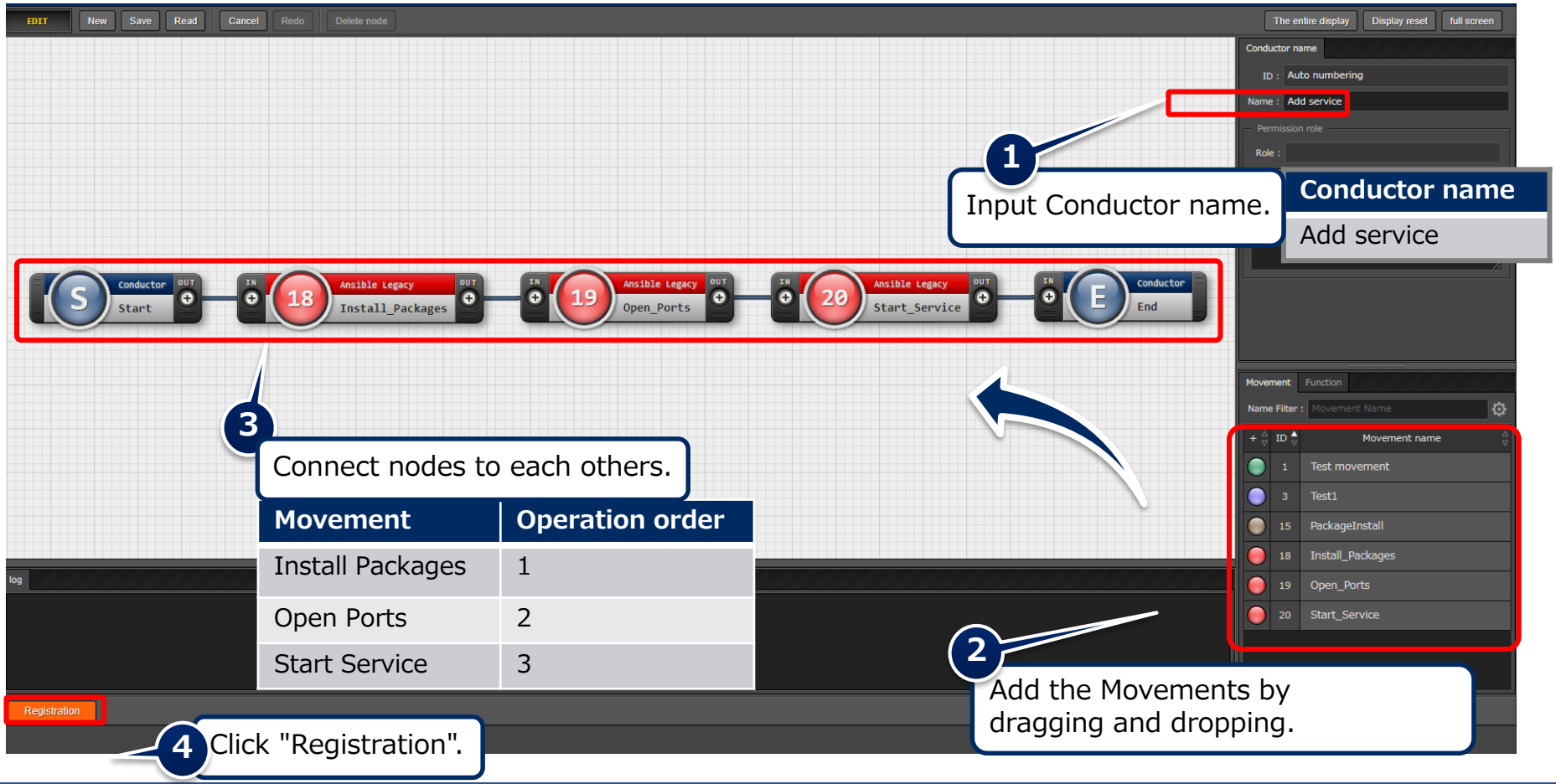
Point

"Start Service" links the 3 Playbook files and executes them together.

Create Conductor

Create a Conductor that collects the defined Movements.

Menu: **Conductor > Conductor class edit**



Movement	Operation order
Install Packages	1
Open Ports	2
Start Service	3

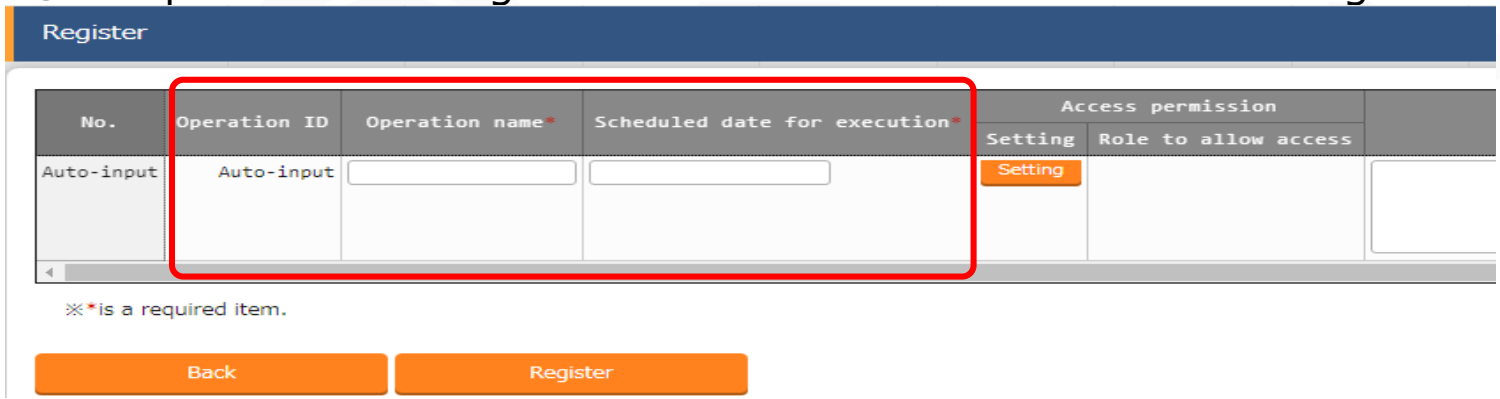
Register new operation

Create operation. Link Movement and Host.
※ “Operation” is the name of the work operation used in the ITA System and represents the entire operation.

Menu **: Basic Console > Operation list**

① Click Register > Start Registration.

② Input the following information for each item and click "Register".



Operation name	Scheduled date for execution
Install Apache	(Enter arbitrary value)
Install Tomcat	(Enter arbitrary value)

※ "Scheduled date for execution" is just an item for management. It will not be executed automatically.

1.6 Register to device list

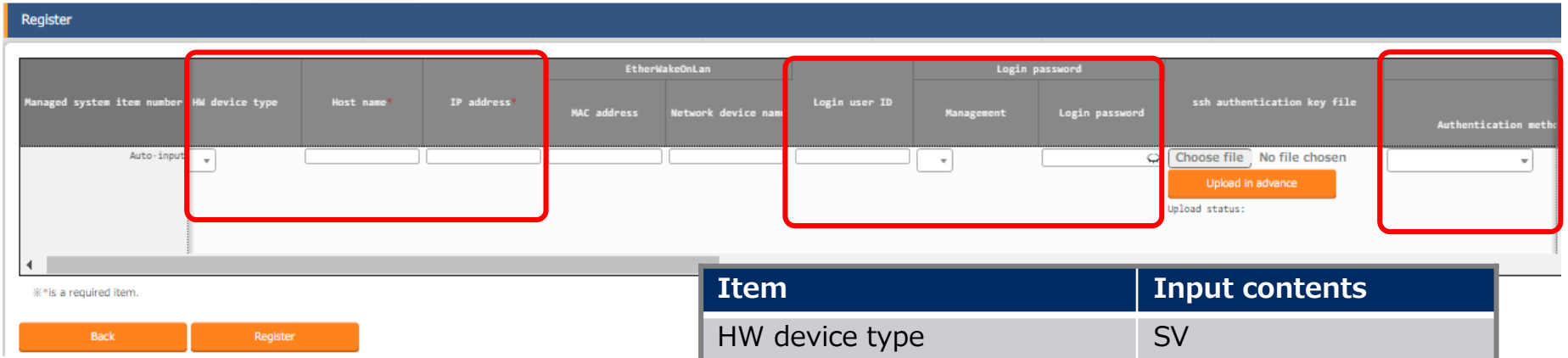
Register a host to the device list

Register host that will execute the operation in ITA.

Menu **: Basic Console > Device list**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



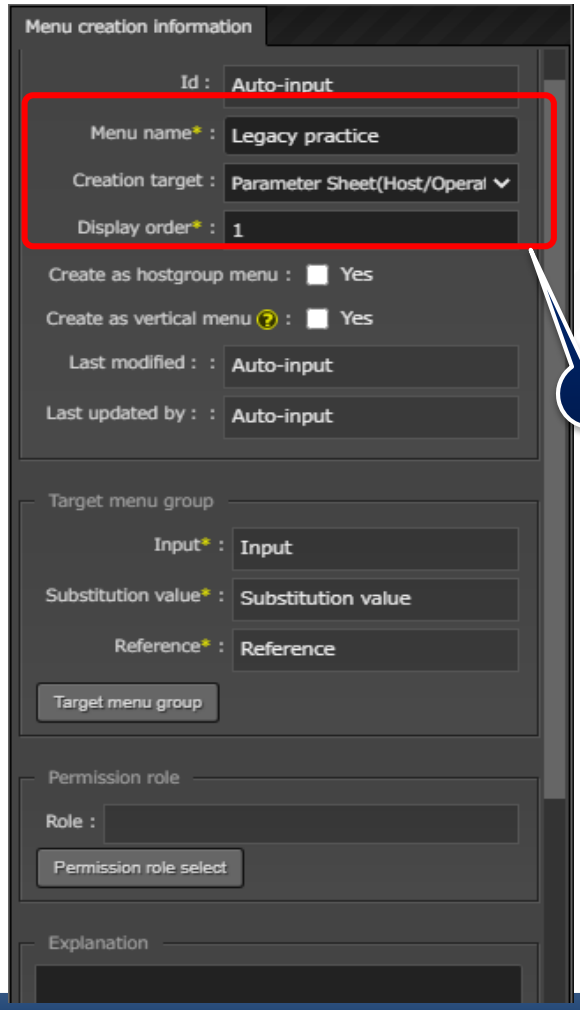
Item	Input contents
HW device type	SV
Host name	(Arbitrary value)
IP address	(Arbitrary value)
Login user ID	(Arbitrary value)
Management	●
Login password	(Arbitrary value)
Authentication method	Password Authentication

1.7 Parameter sheet creation (1/2)

Create menu

Create a parameter sheet and manage the parameters that apply to the target host.

Menu **: Create menu > Create/Define menu**



1 Input the following information.(Next item)

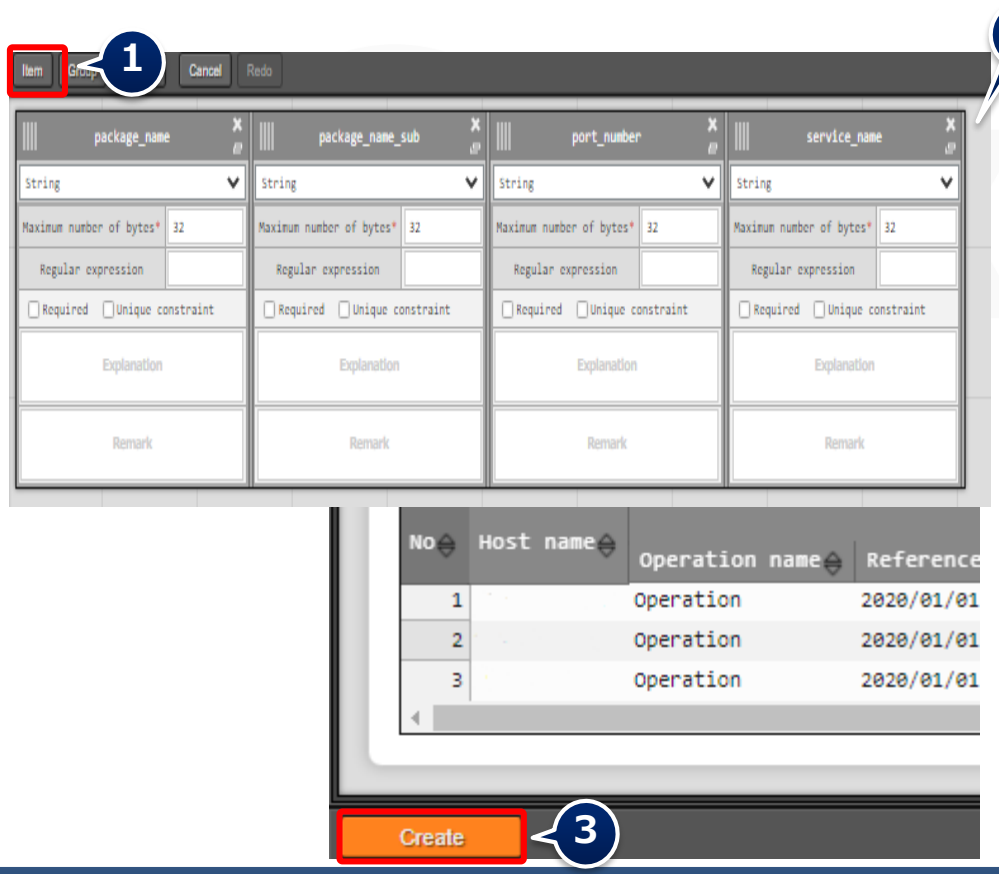
Item name	Input contents
Menu name	Legacy practice
Target	Parameter sheet (Host/Operation)
Display order	1

1.7 Parameter sheet creation (2/2)

Define the item name of the parameters sheet

Continuing from the previous section, define the items on the sheet.

Menu **: Create menu > Create/Define menu**



Item name	Input method	Maximum number of bytes
package_name	String	32
package_name_sub	String	32
port_number	String	32
service_name	String	32

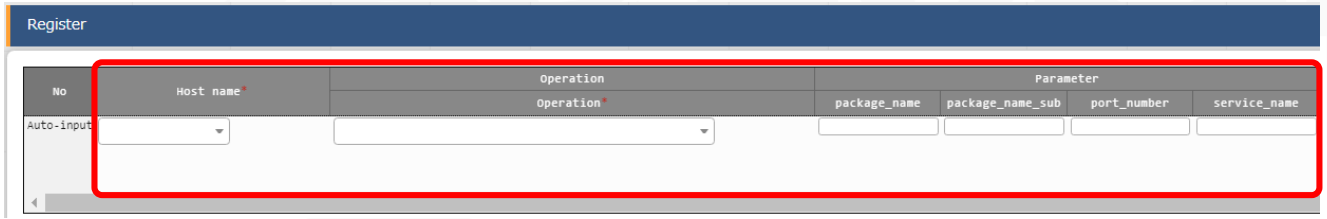
Register data to the parameter sheet

The parameter sheet was created by the operation in the previous section.
Move to the created menu and input the data.

Menu: **Input > Legacy practice(Created menu)**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



Host name	Operation	package_name	package_name_sub	port_number	service_name
(Target host)	Install Apache	httpd	(blank)	80/tcp	httpd
(Target host)	Install Tomcat	tomcat	tomcat-webapps	8080/tcp	tomcat

1.9 Substitution value automatic registration setting

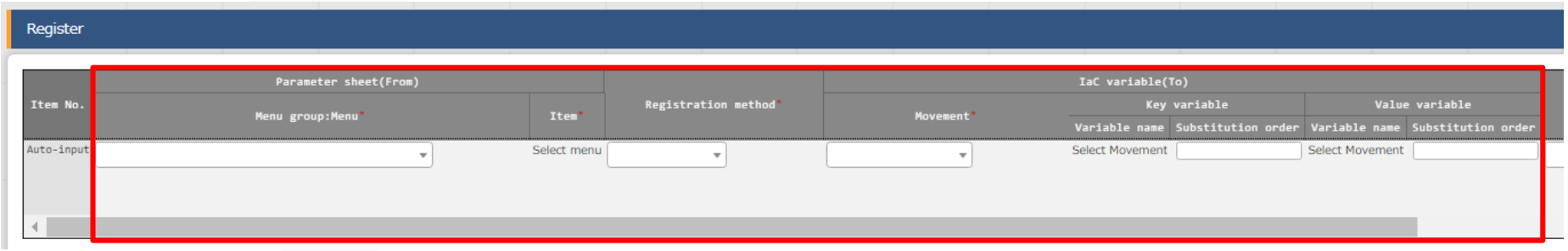
Set Substitute Value Automatic Registration settings

Connect the variables to each item after entering the data in the parameter sheet.

Menu: **Ansible-Legacy > Substitution value automatic registration setting**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



Menu	Item	Registration method	Movement	Value variable variable name	Substitution order
Legacy practice	package_name	Value type	Install Packages	VAR_package_name	1
Legacy practice	package_name_sub	Value type	Install Packages	VAR_package_name	2
Legacy practice	port_number	Value type	Open Ports	VAR_port_number	(Blank)
Legacy practice	service_name	Value type	Start Service	VAR_service_name	(Blank)

1.10 Check Substitution value and Target host

Check Substitution value and Target host

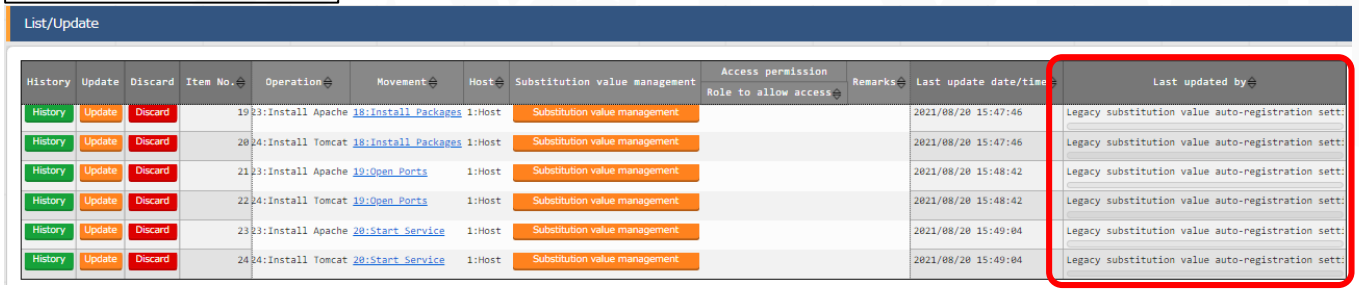
Check target host and values specified by substitution value automatic registration.

Menu: **Ansible-Legacy > Target host & Ansible-Legacy > Substitution value list**

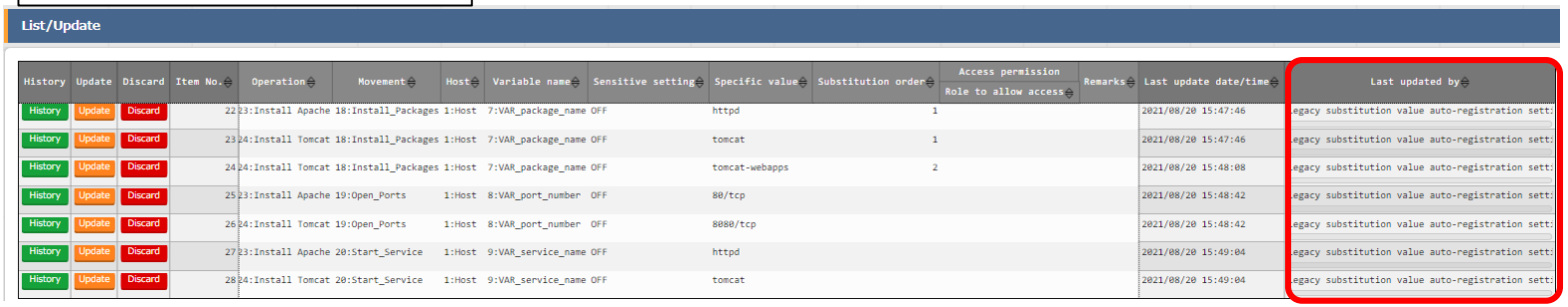
① Click "Filter".

② Check that the correct value is specified by "legacy substitution value automatic registration setting procedure".

Target host



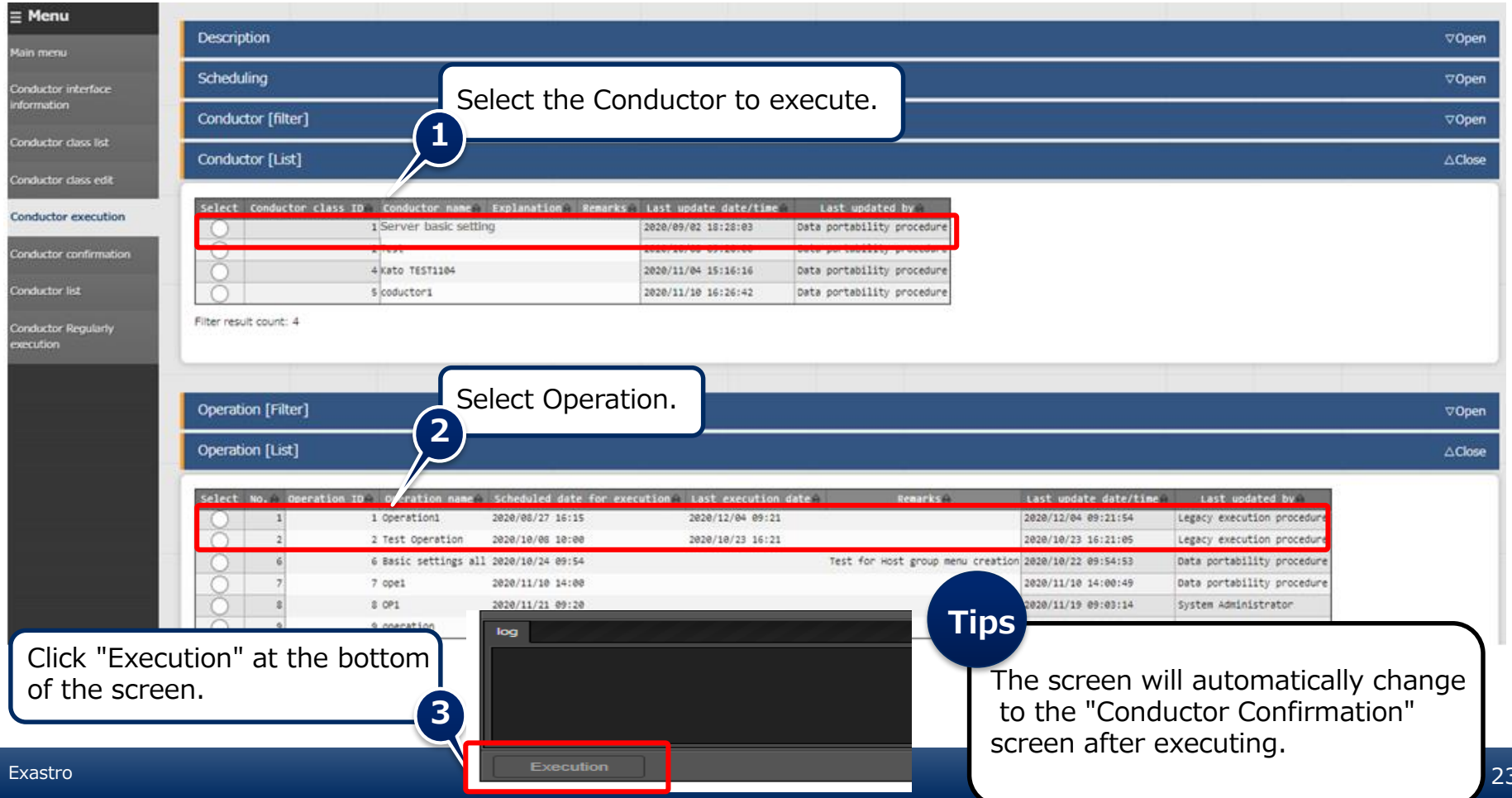
Substitution value list



Execute Conductor

If you finished the operations in the previous section, the Conductor should be created and the substitute values should be registered.
Finally, execute Conductor and check the result on the target host.

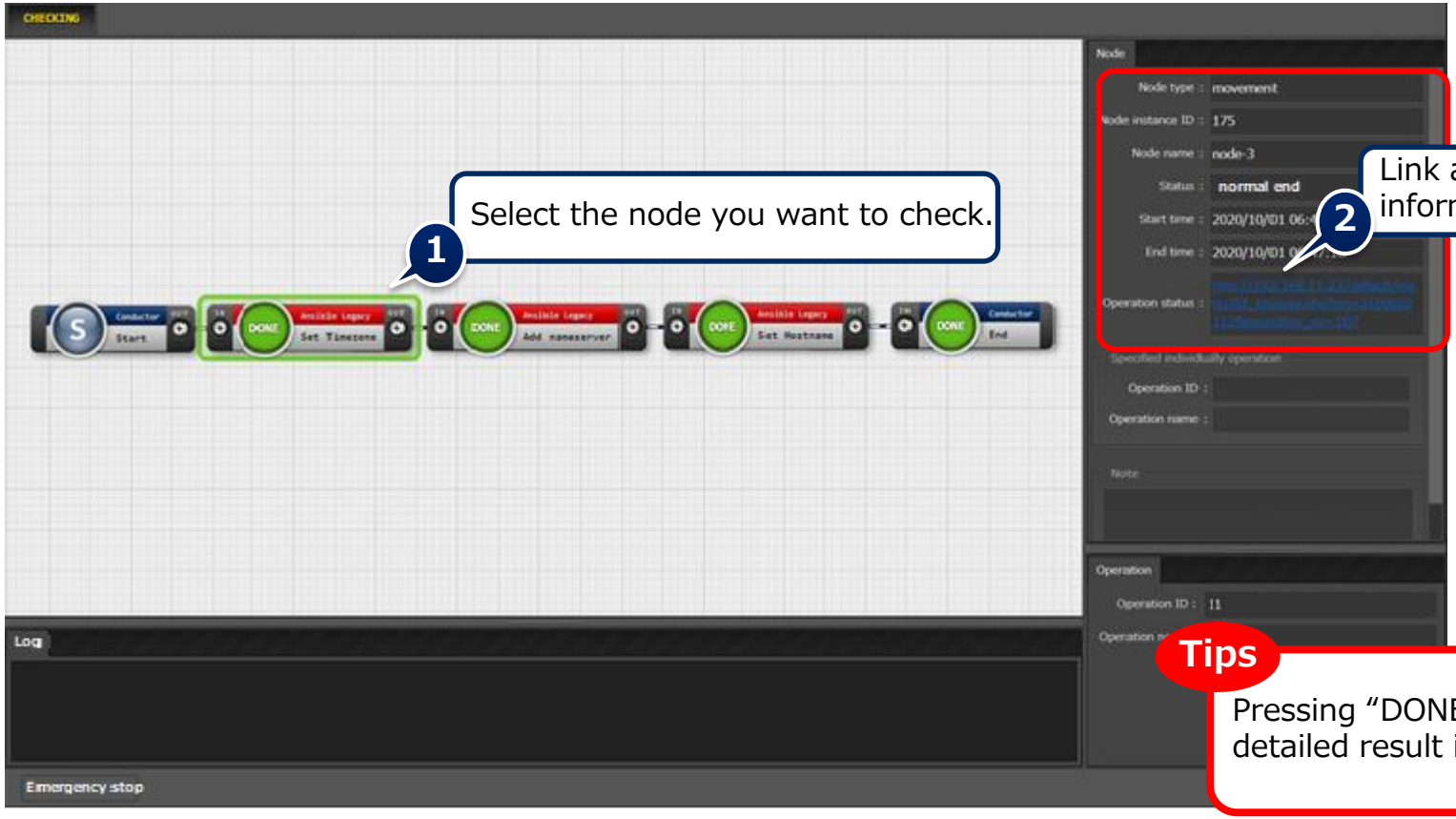
Menu : **Conductor > Conductor Execution**



Check the Conductor Execution results

In the work confirmation screen, you can check the results of the whole execution or execution per node.
Selecting an inputted Movement will show a link to a more detailed result screen.

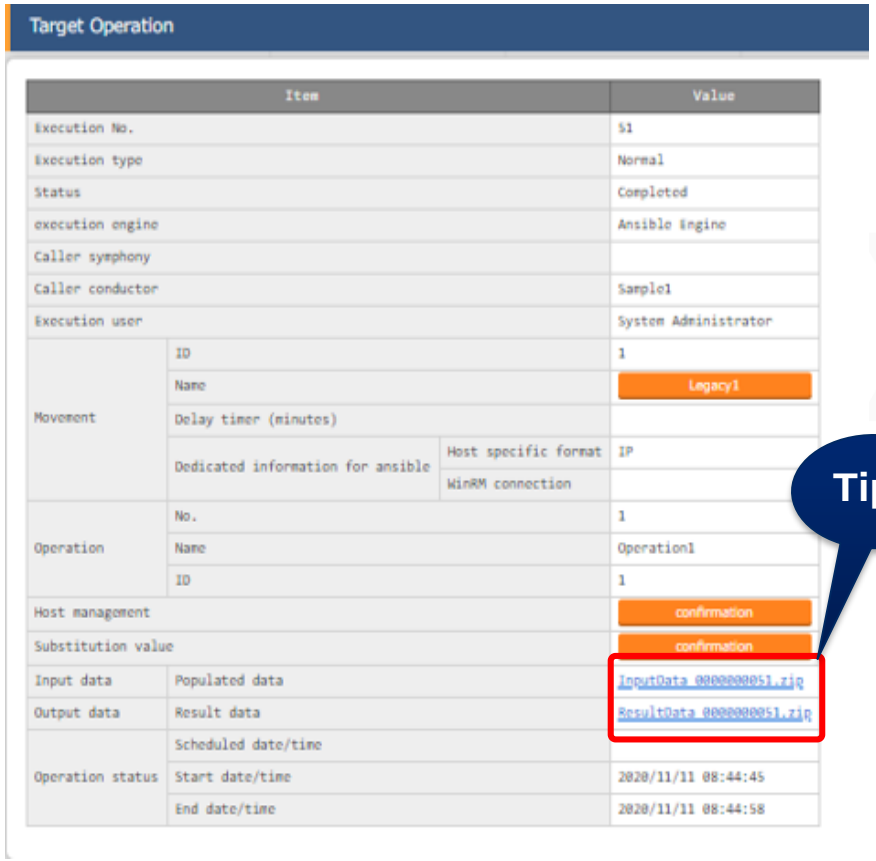
Menu : **Conductor > Conductor Confirmation**

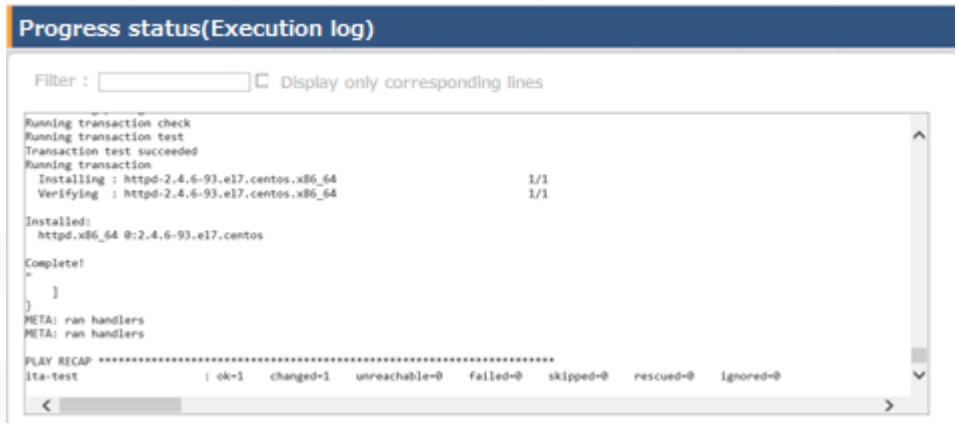


1.11 Execution (3/3)

Check the detailed results for each Movement

Clicking the link will transfer the user to a screen where Execution status and logs are displayed..
Users can also check the input data and output data.





Tips

Users can download a zip file that collects input data and result data.

Tips

Users can check the installation of Apache and Tomcat in the following URLs.

Apache- http://(Host IP address):80
Tomcat- http://(Host IP address):8080

Chapter 2 Ansible-LegacyRole

2.1 Operation Environment and Scenario(1/3)

Work environment

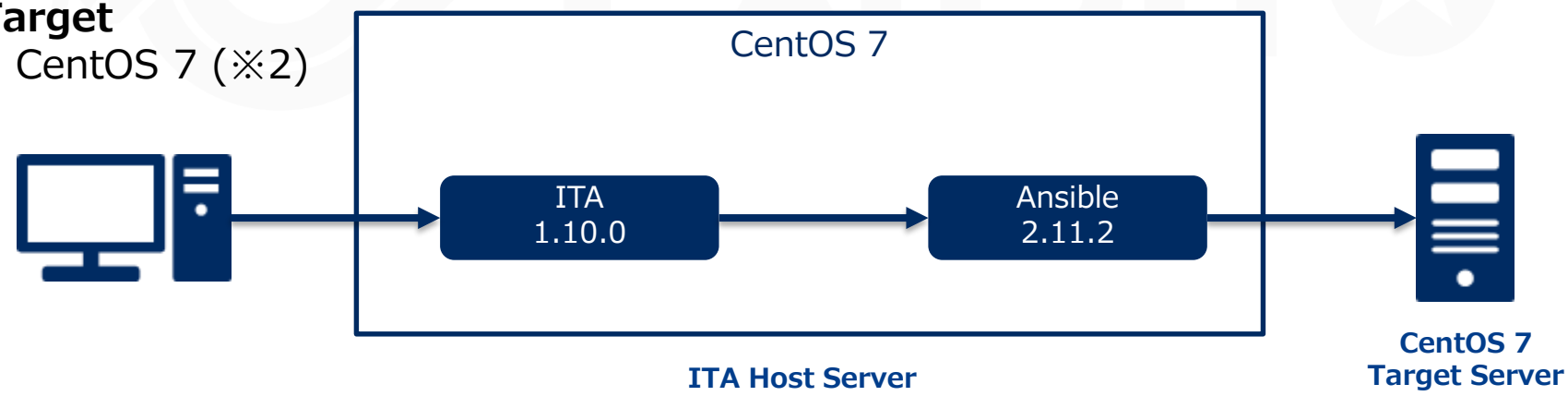
The work environment used in this document is as follows.(The same as Chapter 1 Ansible-Legacy Role)
Please prepare a server in addition to the ITA Host server. The additional server will be target for operation.

ITA Host Servers

- CentOS 7 (※1)
- ITA 1.10.0
- Ansible 2.11.2

Target

- CentOS 7 (※2)

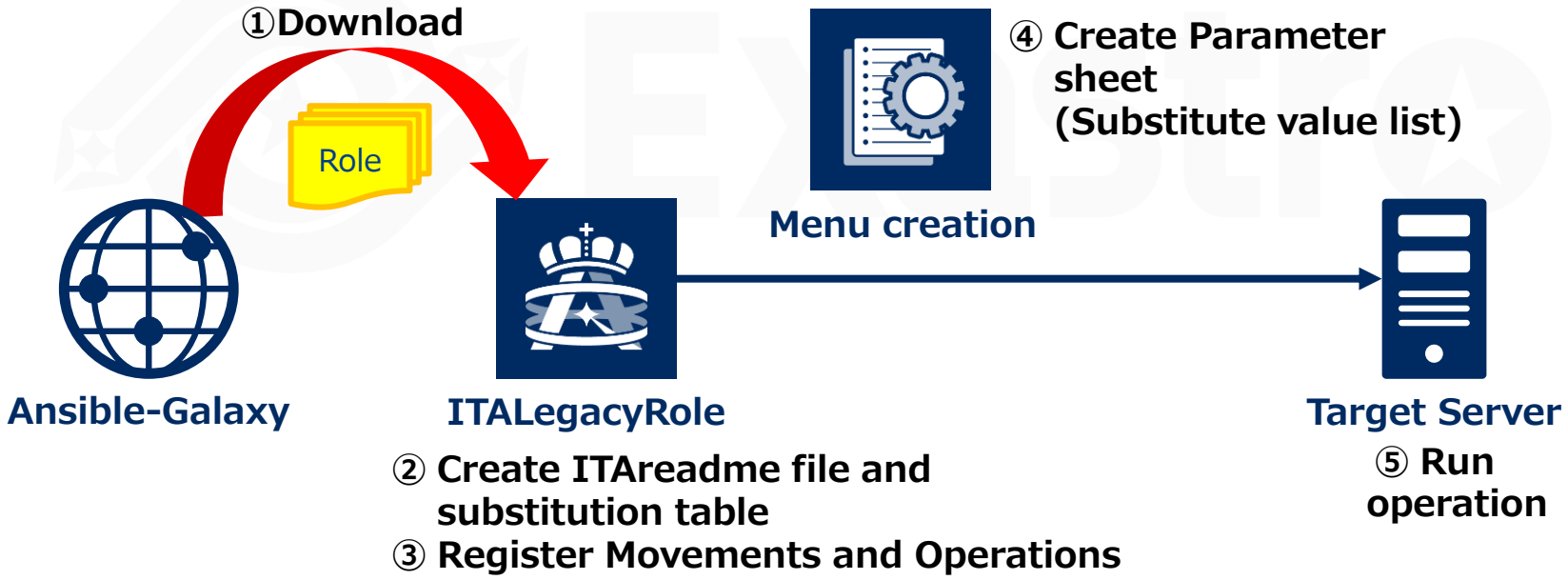


※1 In this scenario, CentOS7 will be used for the host server, but ITA can be installed on any RHEL7 and RHEL8 type OS.
※2 Any OS compatible with Ansible can be used.

Scenario diagram

The most important feature of LegacyRole is that allows users to register and use role packages.

In this document, we will register and execute the Role downloaded from Ansible Galaxy to ITA.



2.1 Environment and Scenario(3/3)

Download Role

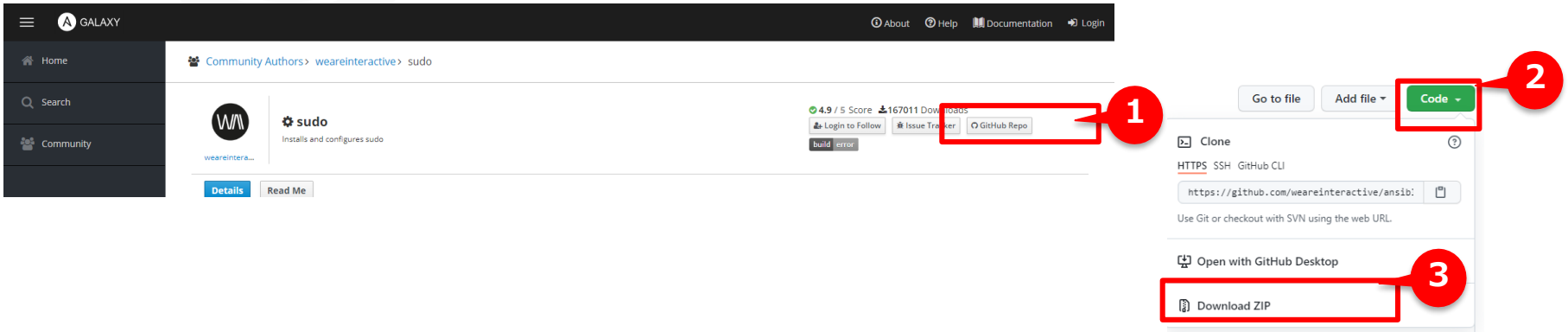
Access the URL below and download the role.

https://galaxy.ansible.com/weareinteractive/sudo

This role package adds files under /etc/sudoers.d

Download steps

① Press “GitHub Repo”
② Press “Code”
③ Press “Download ZIP”



Before preparing the package

Let's look at the role's defaults/main.yml that we are going to use.(Refer to the figure below)
There are two points that need to be changed before execution.

```
---
# sudo_defaults:
#(Abbr) ~~~~
# package name (version)
sudo_package: sudo
# list of username or %groupname
sudo_users: []
# list of username or %groupname and their defaults
sudo_defaults: []
# default sudoers file
sudo_sudoers_file: ansible
# path of the sudoers.d directory
sudo_sudoers_d_path: /etc/sudoers.d
# delete other files in `sudo_sudoers_d_path`
purge_other_sudoers_files: no
  - defaults: env_reset
#   - name: user1
#     defaults: requiretty
# sudo_users:
#   - name: '%group1'
#   - name: 'bar'
#     nopasswd: yes
~~~~
```

1 The correct mapping example is commented out and only an empty array is defined.

- **Change the structure using ITAreadme. Use the substitution table to enable editing in ITA.**

2 The part that the user wants to change is the name of the file when it is generated during execution time.

- **Use the substitution table to enable editing in ITA.**

In cases like these, by creating ITAreadme and substitution tables, users can make necessary changes to the variable definitions without changing the file inside the package.

ITAreadme descriptions

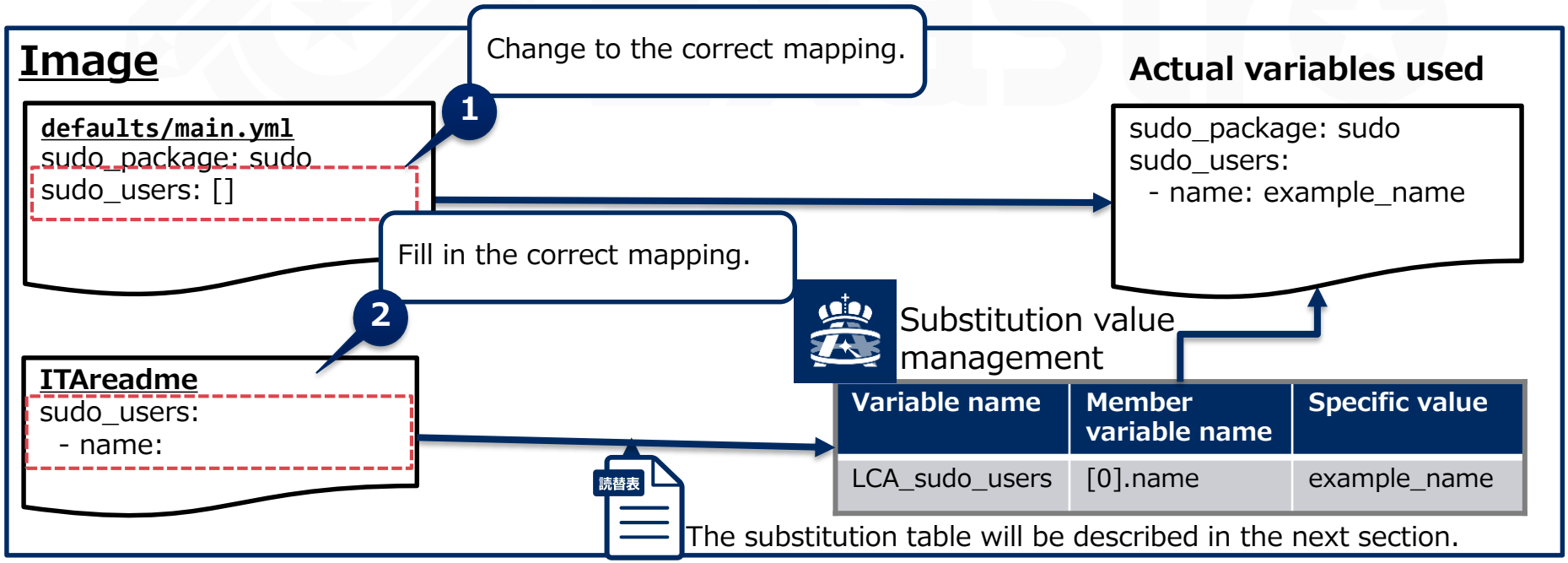
ita_readme is a configuration file for adding and changing variable definitions.
※ For details about ITAreadme, please refer to this Manual.

Tips

If you are copying text straight from this document, make sure that the symbols are not shown.

File name: ita_readme_ansible-sudo-master.yml

```
sudo_users:
  - name:
```



2.2 Role package preparation(3/5)

Substitution table description

The substitution table is defined in the defaults variable definition file or ITAreadme. It is a file that allows for specific values of variables other than “VAR_...” to be set in ITA.

※For details about the substitution table, please refer to this Manual.

File name: ita_translation-table_ansible-sudo-master.txt

```
LCA_sudo_users: sudo_users
LCA_sudo_sudoers_file: sudo_sudoers_file
```

Image

defaults/main.yml
sudo_users: []
~~~~ (omission) ~~~~
sudo_sudoers_file: ansible

ITAreadme
sudo_users:
  - name:

Actual variables used

sudo_users:
  - name: example_name
sudo_sudoers_file: example_sudoers

Substitution value management

substitution table
LCA_sudo_users: sudo_users
LCA_sudo_sudoers_file: sudo_sudoers_file

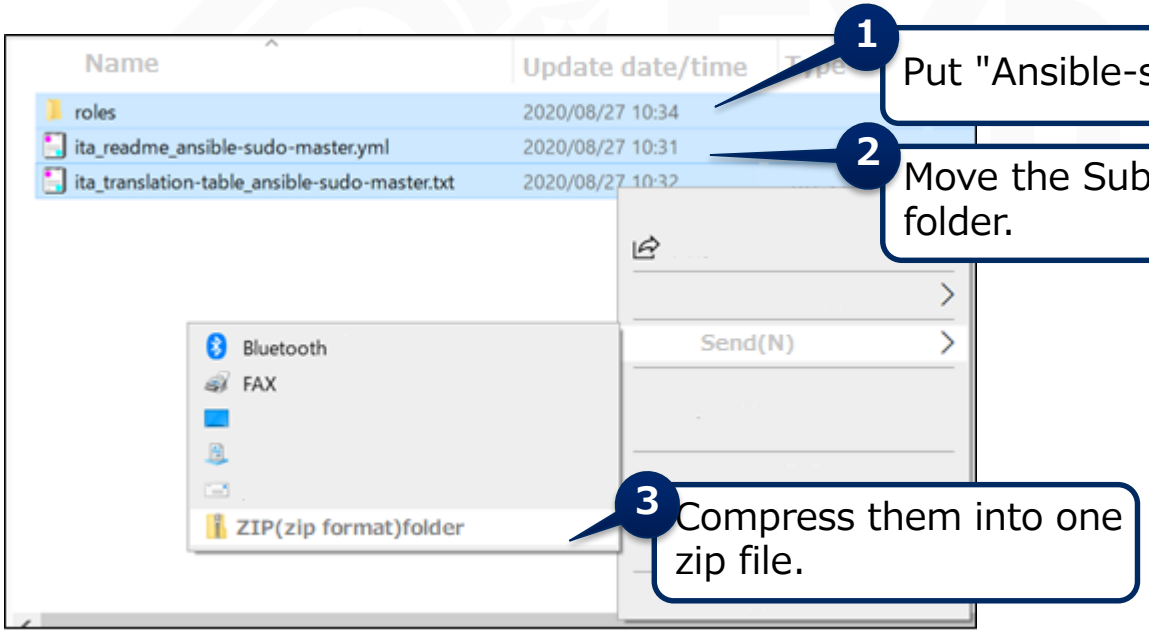
Variable name	Member variable name (Added ITAreadme)	Specific value
LCA_sudo_users	[0].name	example_name
LCA_sudo_sudoers_file		example_sudoers
~~~~

2.2 Role package preparation(4/5)

Collect the necessary files into a zip file

Let's collect all the items we've created into a zip file and register them to ITA.
Please create a zip file as described below.

① Create a "roles" folder and put the downloaded Role in it.
② Arrange the substitution table and ITAreadme side by side in the "roles" folder.
③ Compress the "roles" folder, Substitute table and the ITAreadme into one zip file.



2.2 Role package preparation(5/5)

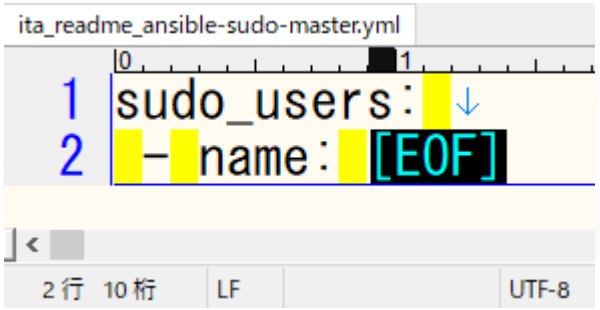
Check the contents of the Role package

Check that the hierarchy of the contents in the role packages are as following.

```
roles.zip
 ├roles
 │  ├ansible-sudo-master
 │  ├defaults
 │  │  └main.yml
 │  ├tasks
 │  │  └main.yml
 │  │  ~~~~~
 │  │  ~~~~~
 │  └README.md
 ├ita_readme_ansible-sudo-master.yml
 └ita_translation-table_ansible-sudo-master.txt
```

Tips

Make sure that all characters are present, including special characters
(E.g.) Underscore, Colon, Space, Hyphen



(E.g.) ita_readme_ansible-sudo-master.yml

Tips

Folders that meet the following two requirements are handled as “Role”s by Legacy Role

① The folder is placed under the “Roles” folder
② The folder contains a “tasks” folder.

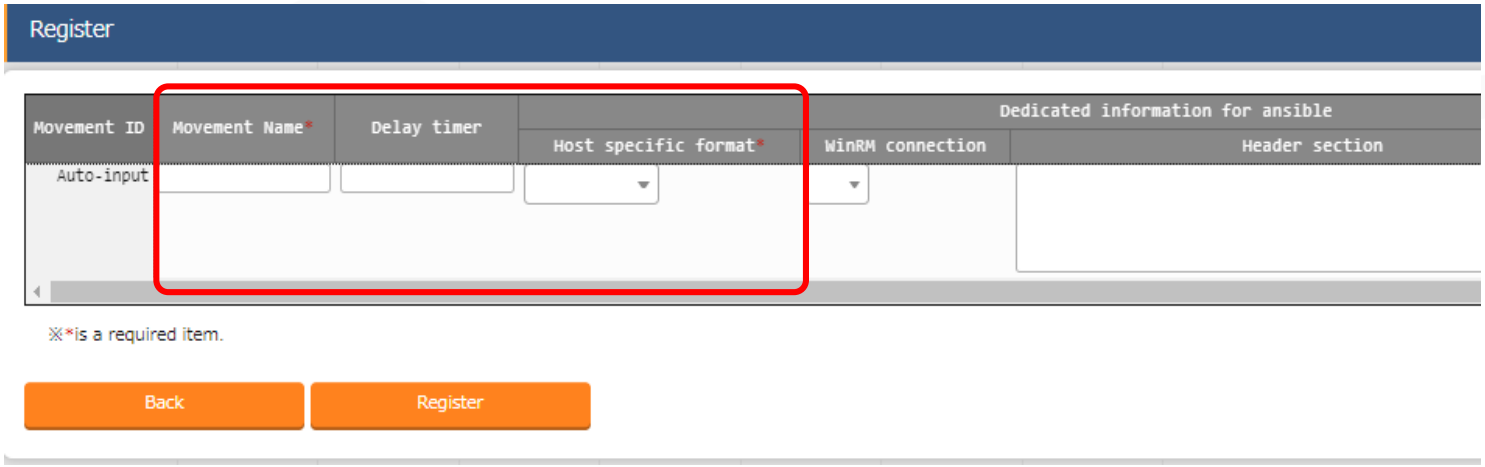
Create Movement

Create a movement that is going to be linked to the Role.

Menu : **Ansible-LegacyRole > Movement list**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



Item name	Input contents
Movement name	RegisterSudoer
Host specific format	IP

Role package registration

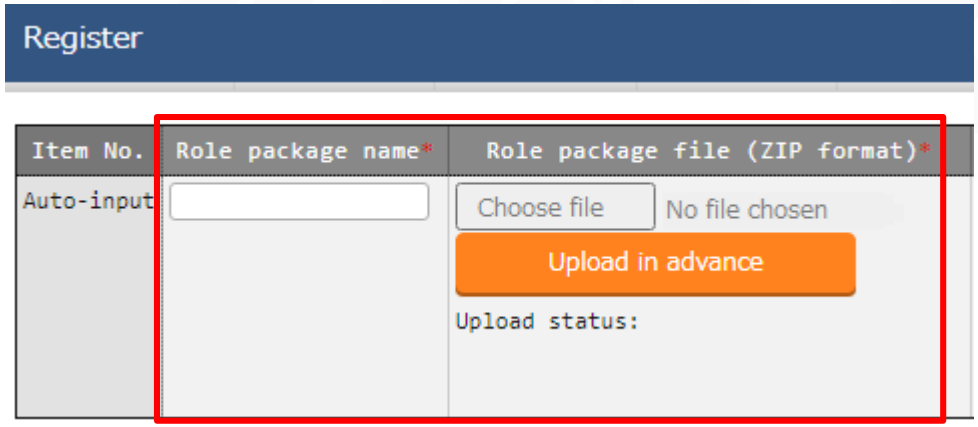
Register the role package file you created.

Menu: **Ansible-LegacyRole > Role package list**

① Click Register > Start Registration.

② Select the created **zip file** from "Browse" and click "Upload in advance".

③ Input the following information for each item and click "Register".



Role package name	sudo-master

2.3 Movement configuration(3/3)

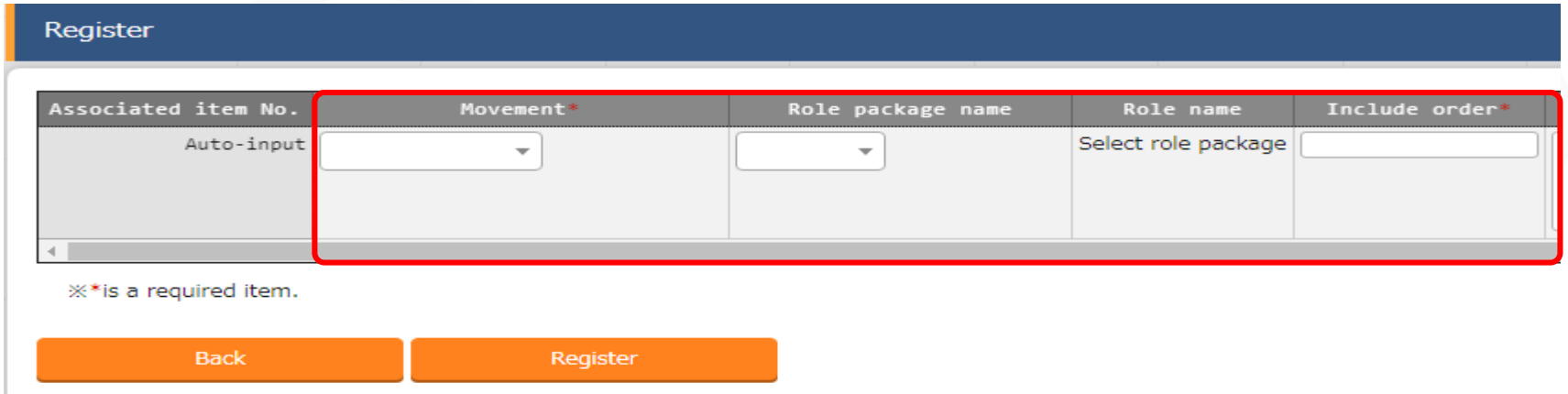
Register role names to Movement

Register individual role names to Movement.

Menu: **Ansible-LegacyRole > Movement details**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



Movement name	Role package name	Role name	Include order
Sudoer registration	sudo-master	ansible-sudo-master	1

※ Do not register multiple role packages in the same Movement. Unexpected errors will occur when executed.

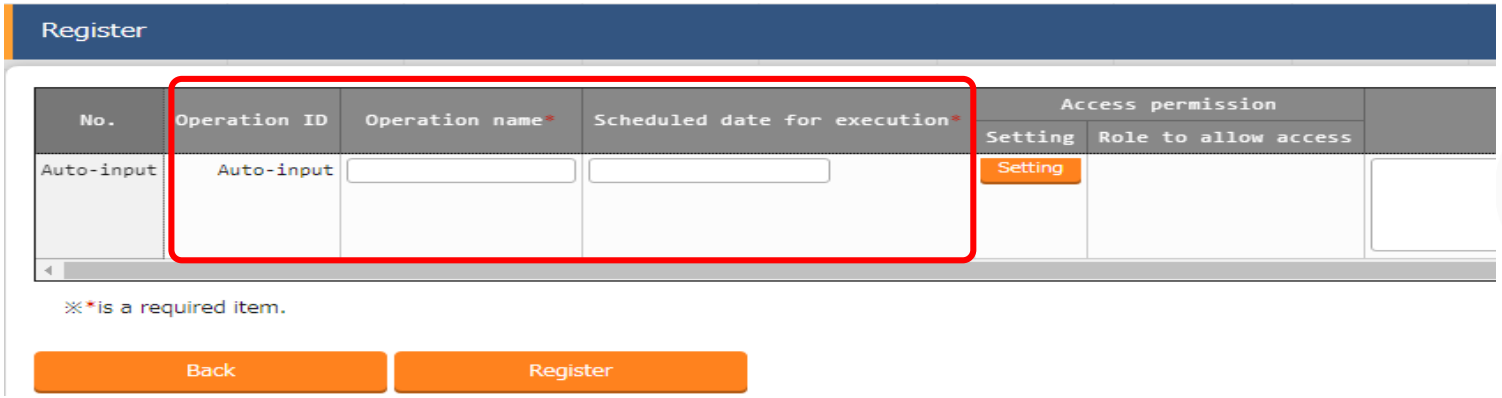
Register new operation

Create operation. Link Movement and Host.

Menu **: Basic Console > Input operation list**

① Click Register > Start Registration.

② Input the following information for each item and click "Register".



Item name	Input contents
Operation name	LegacyRole_Practice
Scheduled date for execution	(Enter arbitrary value)

※ "Scheduled date for execution" is just an item for management. It will not be executed automatically.

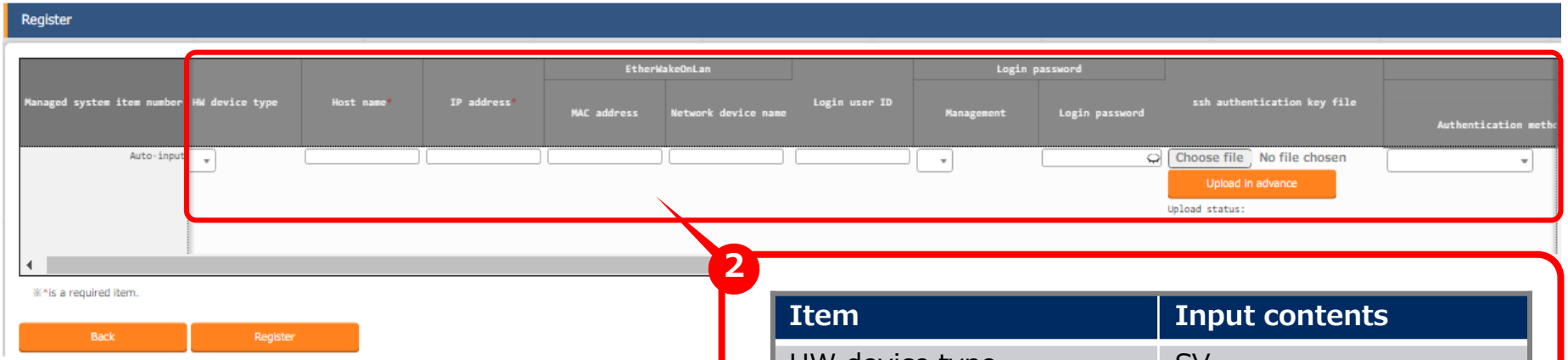
Register a host to the device list

Register host to execute the operation in ITA.

Menu : **Basic Console > Device list**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



2

Item	Input contents
HW device type	SV
Host name	(Arbitrary value)
IP address	(Arbitrary value)
Login user ID	(Arbitrary value)
Management	●
Login password	(Arbitrary value)
Authentication method	Password Authentication

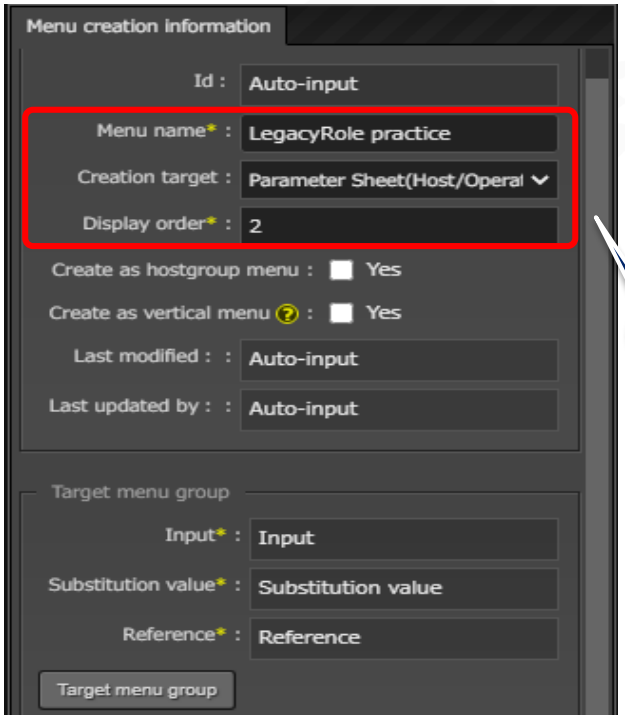
2.6 Parameter sheet creation(1/2)

Create menu

Create a parameter sheet and manage the parameters that apply to the target host.

Menu **: Create menu > Create/Define menu**

① Input/Select the following and see the next page



1

Item name	Input contents
Menu name	LegacyRole practice
Creation target	Parameter sheet (Host/Operation)
Display order	2

2.6 Parameter sheet creation(2/2)

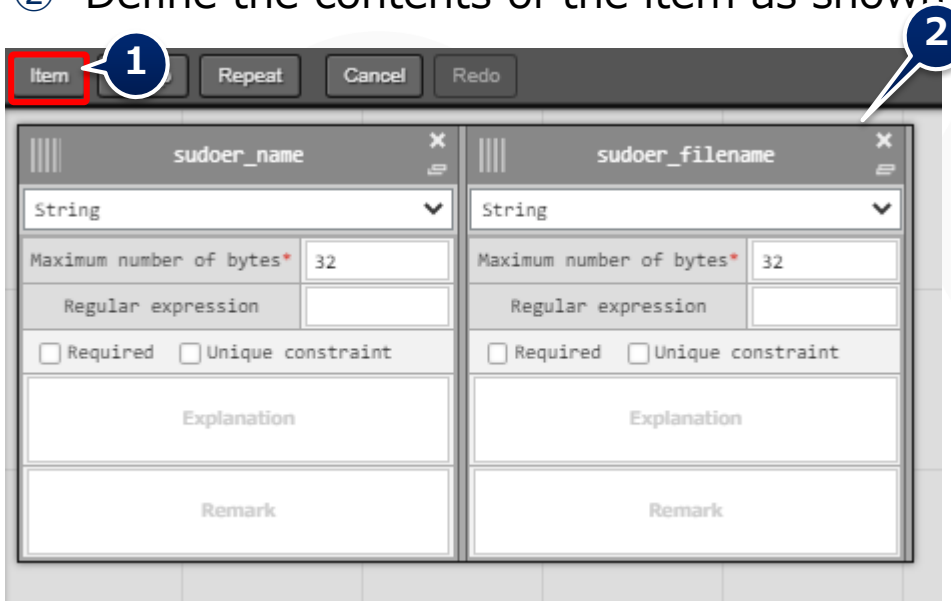
Define the item name of the parameters sheet

Continuing from the previous section, define the items on the sheet.

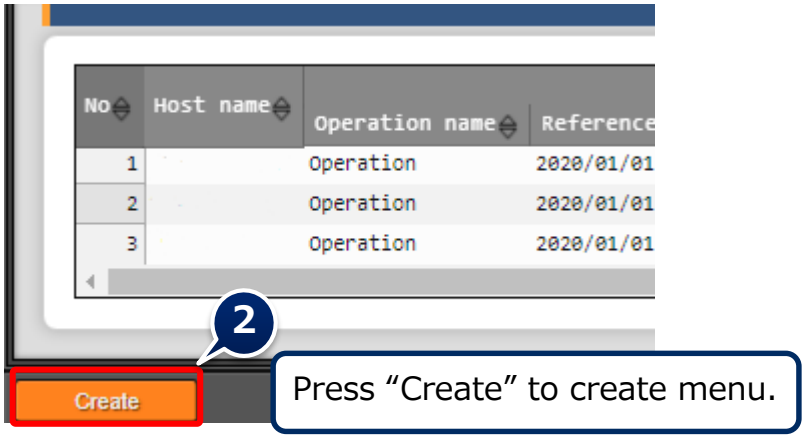
Menu **: Create menu > Create/Define menu**

① Press “Item” to add a new item.

② Define the contents of the item as shown in the table and press “Create”



Item name	Input method	Maximum number of bytes
sudoer_name	String	32
sudoer_filename	String	32



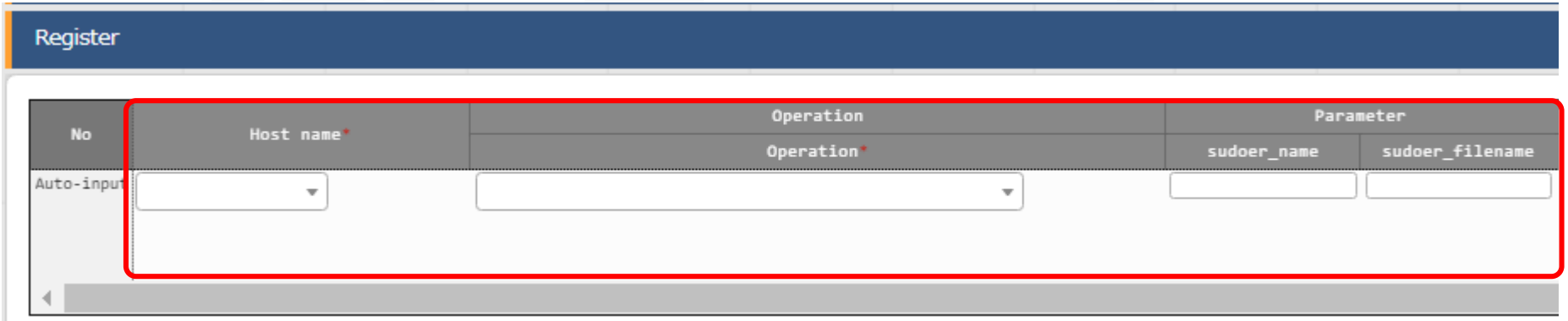
Register data to the parameter sheet

The parameter sheet was created by the operation in the previous section.
Move to the created menu and input the data.

Menu: **Input > LegacyRole practice(Created menu)**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



Input Example

Host name	Operation	sudoer_name	sudoer_filename
(Target host)	LegacyRole_Practice	example_name	example_sudoers

2.8 Substitution value automatic registration setting

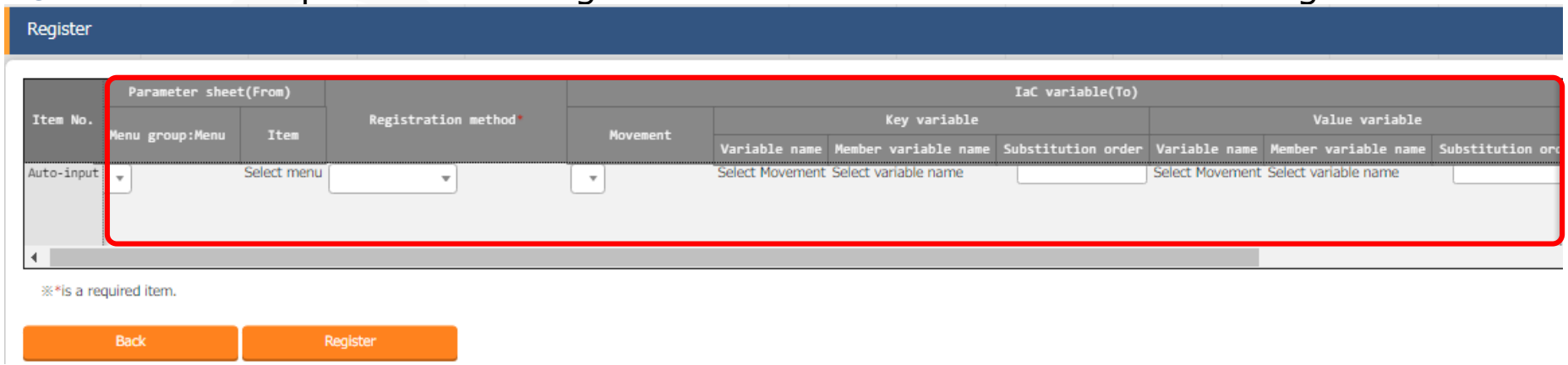
Set Substitute Value Automatic Registration settings

Connect the variables to each item after entering the data in the parameter sheet.

Menu: **Ansible-LegacyRole > Substitution value auto-registration setting**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



Menu group : Menu	Item	Registration method	Movement	Value variable Variable name	Value variable member variable name
LegacyRole for practice	sudoer_name	Value type	RegisterSudoer	LCA_sudo_users	[0].name
LegacyRole for practice	sudoer_filename	Value type	RegisterSudoer	LCA_sudo_sudoers_file	(blank)

Check Substitution value and Target host

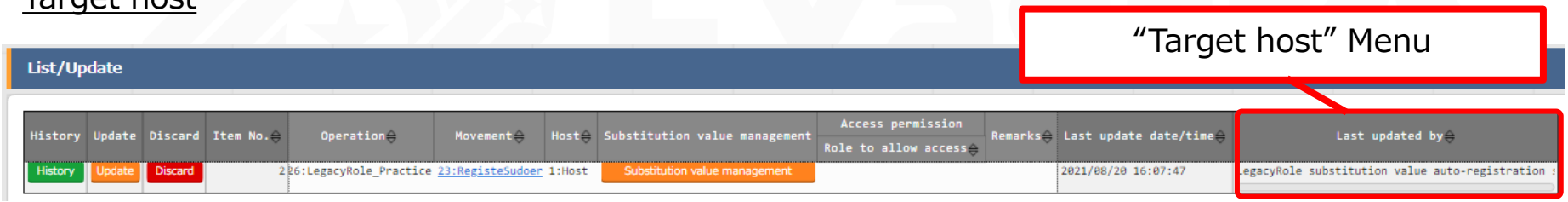
Check the value specified by the substituted value automatic registration and the target host.

Menu: **Ansible-LegacyRole > Target host &**
Ansible-LegacyRole > Substitution value list

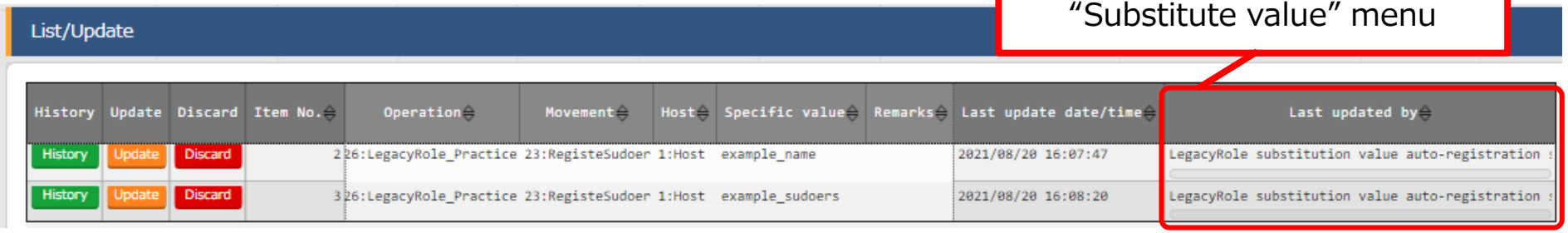
① Click "Filter".

② Check that the correct value is specified by “LegacyRole substitution value automatic registration setting procedure".

Target host



Substitution value list

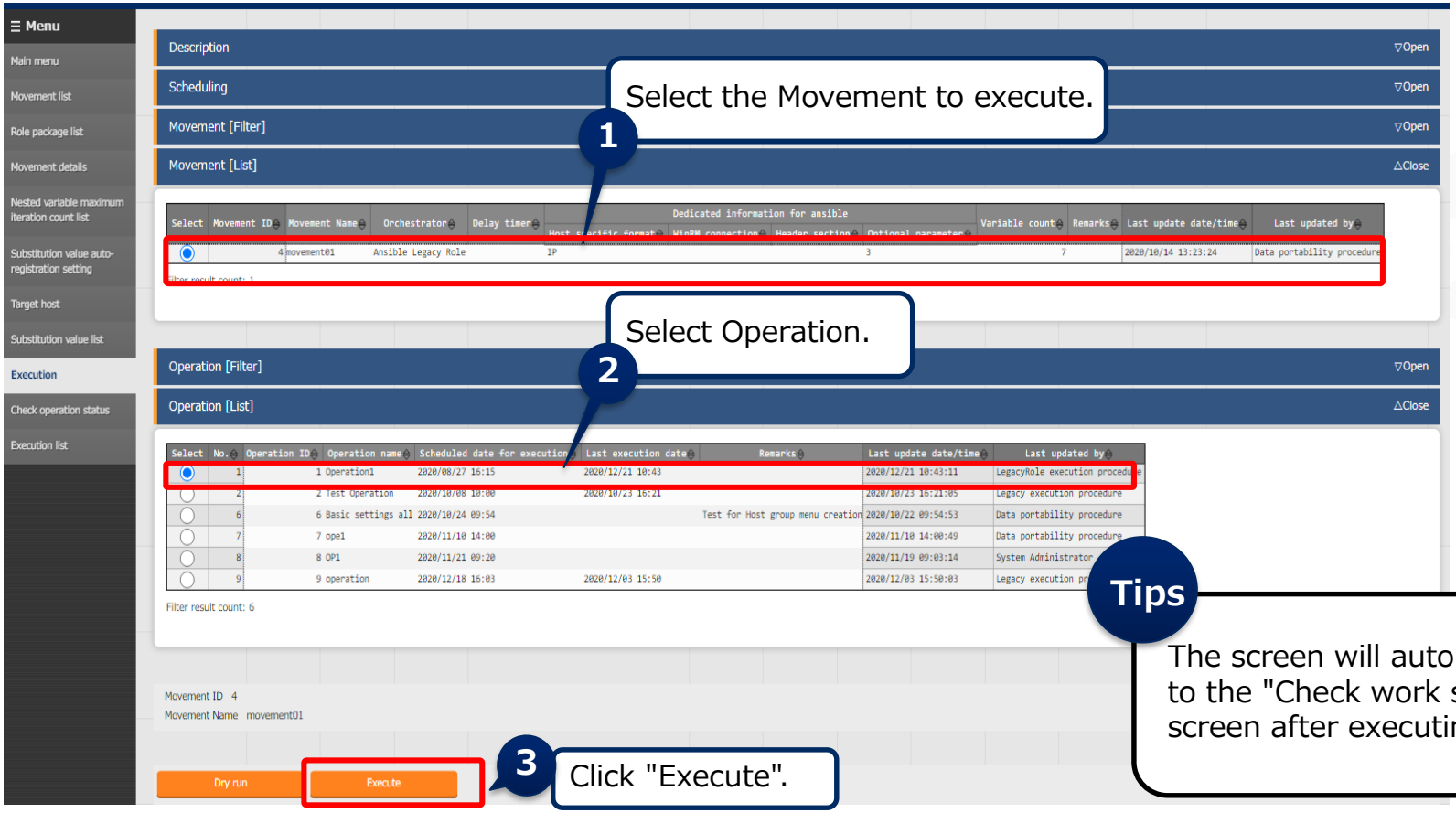


2.10 Execution (1/2)

Execute Movement directly

The Movement created in this scenario is one.
Let's skip the Conductor creation and use the direct execution feature.

Menu: **Ansible-LegacyRole > Execution**



Tips

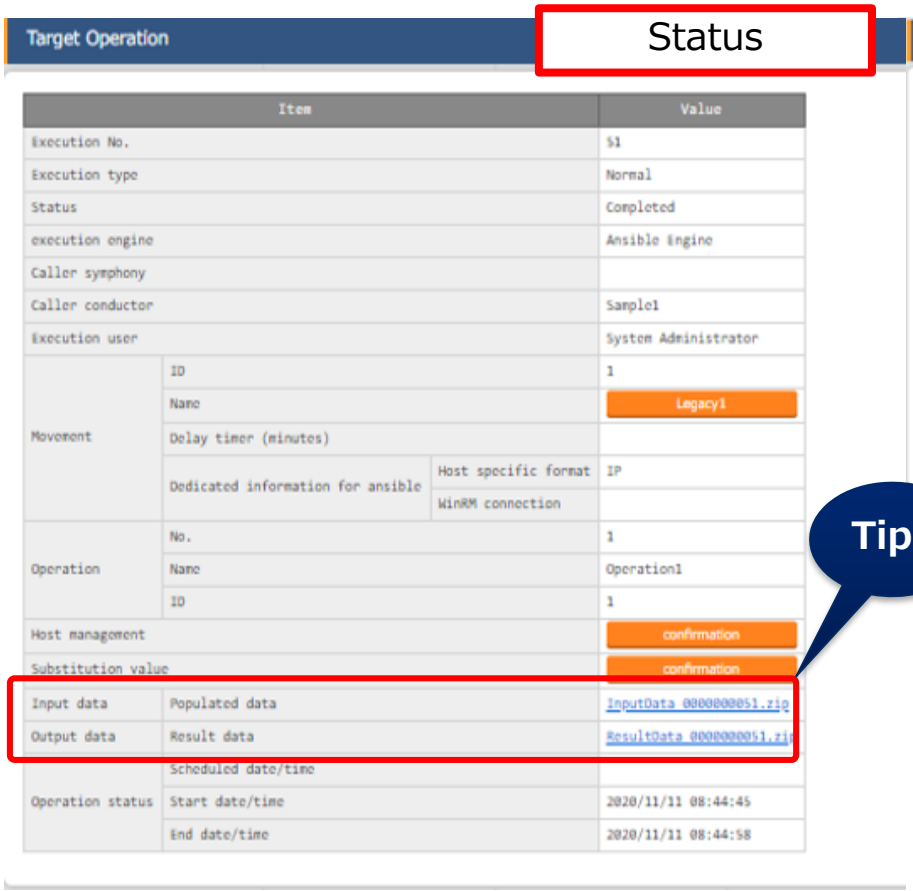
The screen will automatically change to the "Check work status" screen after executing.

2.10 Execution (2/2)

Check execution results

Clicking the link will transfer the user to a screen where Execution status and logs are displayed.

Menu : **Ansible-LegacyRole > Check operation status**



Tips

Users can download a zip file that contains input and result data..

Tips

To check the results on the target host, please refer to /etc/sudoers.d

Chapter 3 Ansible-Pioneer

Work environment

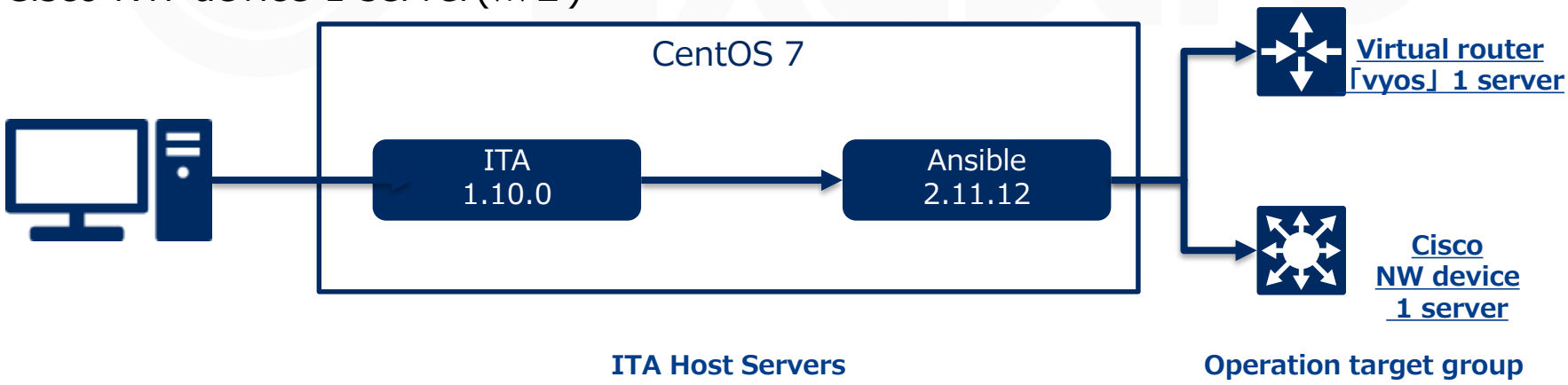
The work environment used in this document is as follows.
We will operate on a NW Device, so please prepare the following environment.

ITA Host Servers

- CentOS 7 (※1)
- ITA 1.10.0
- Ansible 2.11.12

Target

- Virtual router「vyos」 1 server
- Cisco NW device 1 server(※2)



※1 In this scenario, CentOS7 will be used for the host server, but ITA can be installed on any RHEL7 and RHEL8 type OS.
※2 This document uses Layer 3 switches for input examples for each item. If you are using a router or Layer 2 switch, please make sure to replace it accordingly.

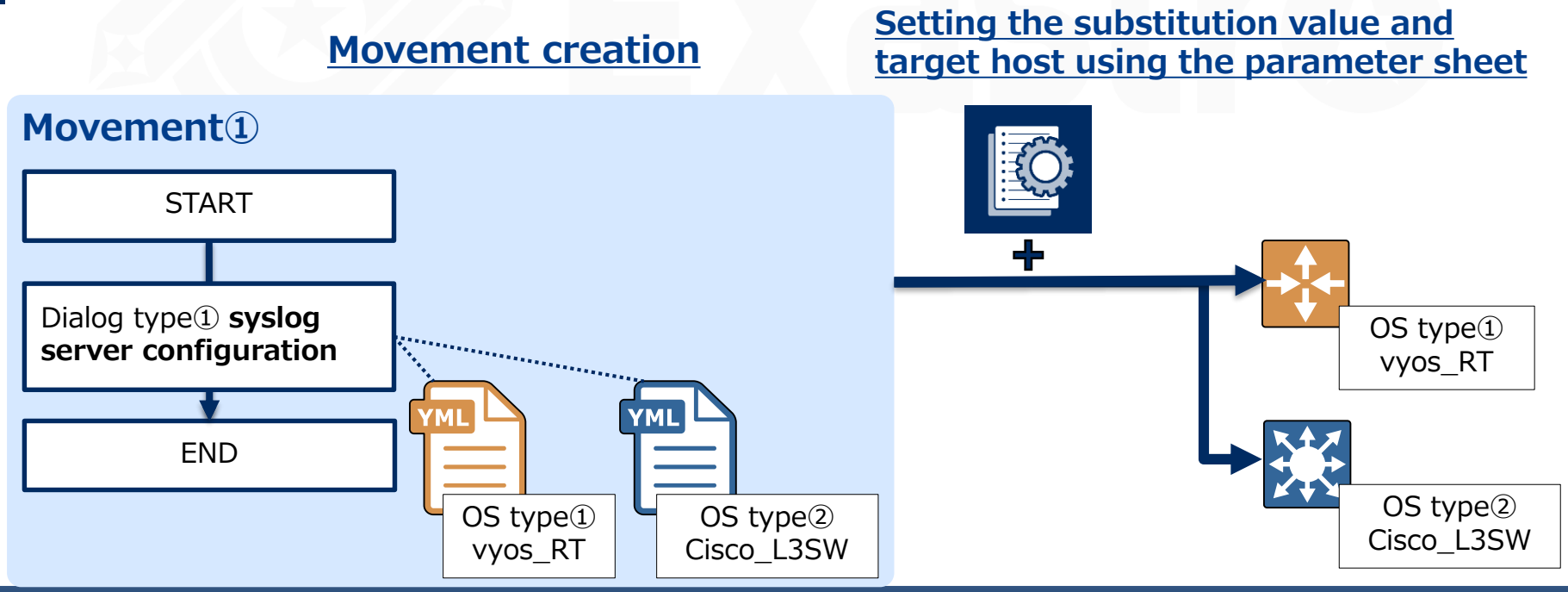
3.1 Work environment and Scenario

Scenario

Ansible-Pioneer is used to specify log servers for NW device with different vendors.
This scenario lets you experience the following 3 features of the Pioneer mode.

① Being able to execute dialog files as well as **Telnet or ssh** communication is provided.
② **Execution without having to worry about OS,** Differences by utilizing Dialog types and OS types.
③ **Repetition of tasks and condutional branching** using proprietary modules.

Scenario Diagram



Dialog file creation

Create the files that we are going to use in this scenario.

- **Conditions**
  - ✓“UTF-8” Character code
  - ✓“LF” Line code
  - ✓”yml” File type
  - ✓Enter half space after colons
  - ✓Unify indents

Log server registration command to vyos.
This will use "with_items" to do iterative processing.

This will check settings.
Outputs settings information and returns "failed“ if the specified string is missing.
Conditional judgement will be made for each iteration by "with_items".

File name: vyos_set_syslog_server.yml

```
conf:
  timeout: 10

exec_list:
  - expect: 'password:'
    exec: '{{ __loginpassword__ }}'

  - expect: '{{ __loginuser__ }}@{{ __loginhostname__ }}'
    exec: 'set terminal length 0'

  - expect: '{{ __loginuser__ }}@{{ __loginhostname__ }}'
    exec: 'configure'

  - command: 'set system syslog host {{ item.0 }} facility all level {{ VAR_log_severity }}'
    prompt: 'vyos@{{ __loginhostname__ }}'
    with_items:
      - '{{ VAR_syslog_server_ip }}'
    when:
      - VAR_log_severity is define

  - expect: 'vyos@{{ __loginhostname__ }}'
    exec: 'commit'

  - expect: 'vyos@{{ __loginhostname__ }}'
    exec: 'save'

  - expect: 'vyos@{{ __loginhostname__ }}'
    exec: 'exit'

  - command: 'show configuration'
    prompt: 'vyos@{{ __loginhostname__ }}'
    with_items:
      - '{{ VAR_syslog_server_ip }}'
    failed_when:
      - stdout match(host *{{ item.0 }})
```

3.2 Dialog file creation(2/3)

Dialog file creation

Create the file on the right.

File name: ios_set_syslog_server.yml

Displays configurations for the log and stores the contents of the standard output in "register".

Log server registration command to CiscoIOS.
Will use"with_items" to **do iterative processing**.
In "exec_when", the decision to execute the process is made according to the contents stored above.

```
conf:
  timeout: 10

exec_list:
  - expect: 'Username:'
    exec: '{{ __loginuser__ }}'

  - expect: 'Password:'
    exec: '{{ __loginpassword__ }}'

  - expect: '{{ __loginhostname__ }}'
    exec: 'enable'

  - expect: 'Password:'
    exec: '{{ __loginpassword__ }}'

  - expect: '{{ __loginhostname__ }}'
    exec: 'terminal length 0'

  - command: 'show logging'
    prompt: '{{ __loginhostname__ }}'
    register: result_stdout

  - expect: '{{ __loginhostname__ }}'
    exec: 'configure terminal'

  - command: 'logging host {{ item.0 }}'
    prompt: '{{ __loginhostname__ }}'
    with_items:
      - '{{ VAR_syslog_server_ip }}'
    exec_when:
     - result_stdout no match(Logging to {{ item.0 }})

  - command: 'logging facility {{ VAR_log_facility }}'
    prompt: '{{ __loginhostname__ }}'
    when:
      - VAR_log_facility is define

  - command: 'logging trap {{ VAR_log_severity }}'
    prompt: '{{ __loginhostname__ }}'
    when:
      - VAR_log_severity is define
```

3.2 Dialog file creation (3/3)

Check the Dialog files

Make sure that all the indents are correct

For more information regarding description rules, see the manual.

vyos_set_syslog_server.yml

```
conf: |
  timeout: 10
  |
exec_list: |
  - expect: 'password:'
    exec: '{{ __loginpassword__ }}'
  - expect: '{{ __loginuser__ }}@{{ __loginhostname__ }}'
    exec: 'set terminal length 0'
  - expect: '{{ __loginuser__ }}@{{ __loginhostname__ }}'
    exec: 'configure'
  - command: 'set system syslog host {{ item.0 }} facility all level {{ VAR_log_severity }}'
    prompt: 'vyos@{{ __loginhostname__ }}'
    with_items:
      - '{{ VAR_syslog_server_ip }}'
    when:
      - VAR_log_severity is define
  - expect: 'vyos@{{ __loginhostname__ }}'
    exec: 'commit'
  - expect: 'vyos@{{ __loginhostname__ }}'
    exec: 'save'
  - expect: 'vyos@{{ __loginhostname__ }}'
    exec: 'exit'
  - command: 'show configuration'
    prompt: 'vyos@{{ __loginhostname__ }}'
    with_items:
      - '{{ VAR_syslog_server_ip }}'
    failed_when:
      - stdout match(host *{{ item.0 }})
[EOF]
```

29 行　1 桁　LF　UTF-8　REC　挿入

(E.g.) vyos_set_syslog_server.yml

ios_set_syslog_server.yml

```
conf: |
  timeout: 10
  |
exec_list: |
  - expect: 'Username:'
    exec: '{{ __loginuser__ }}'
  - expect: 'Password:'
    exec: '{{ __loginpassword__ }}'
  - expect: '{{ __loginhostname__ }}'
    exec: 'enable'
  - expect: 'Password:'
    exec: '{{ __loginpassword__ }}'
  - expect: '{{ __loginhostname__ }}'
    exec: 'terminal length 0'
  - command: 'show logging'
    prompt: '{{ __loginhostname__ }}'
    register: result_stdout
  - expect: '{{ __loginhostname__ }}'
    exec: 'configure terminal'
  - command: 'logging host {{ item.0 }}'
    prompt: '{{ __loginhostname__ }}'
    with_items:
      - '{{ VAR_syslog_server_ip }}'
    exec_when:
      - result_stdout no match(Logging to {{ item.0 }})
  - command: 'logging facility {{ VAR_log_facility }}'
    prompt: '{{ __loginhostname__ }}'
    when:
      - VAR_log_facility is define
  - command: 'logging trap {{ VAR_log_severity }}'
    prompt: '{{ __loginhostname__ }}'
    when:
      - VAR_log_severity is define [EOF]
```

33 行　35 桁　LF　UTF-8　REC

(E.g.) ios_set_syslog_server.yml

3.3 OS type creation

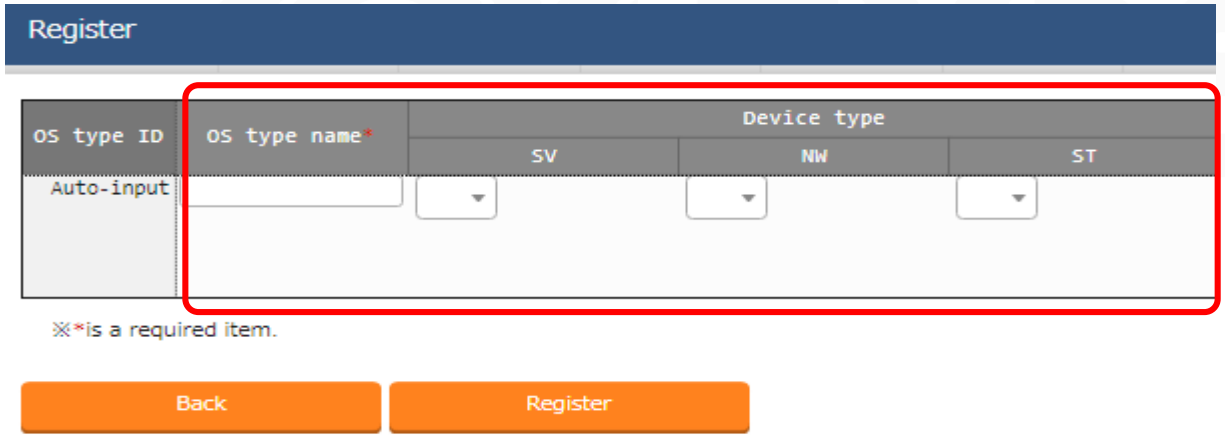
Create “OS type”

Pioneer can choose the code to be dropped depending on the OS of the target host.
First, register "OS type" to ITA.

Menu: **Ansible-Pioneer > OS type master**

① Click Register > Start Registration.

② Input the following information for each item and click "Register".



OS type name	Device type/NW
vyos_RT	●
Cisco_L3SW	●

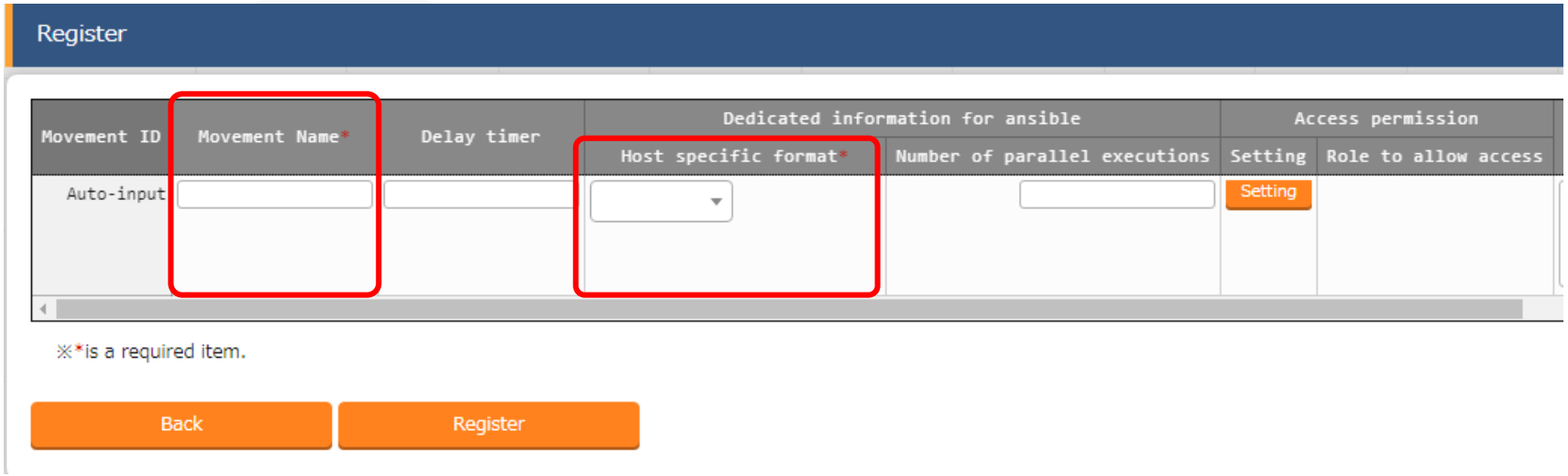
Create Movement

Movement is the smallest unit of operation in ITA.
Create a Movement and link it to the dialog type.

Menu : **Ansible-Pioneer > Movement list**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



Item	Input contents
Movement name	Log server configuration
Host specific format	IP

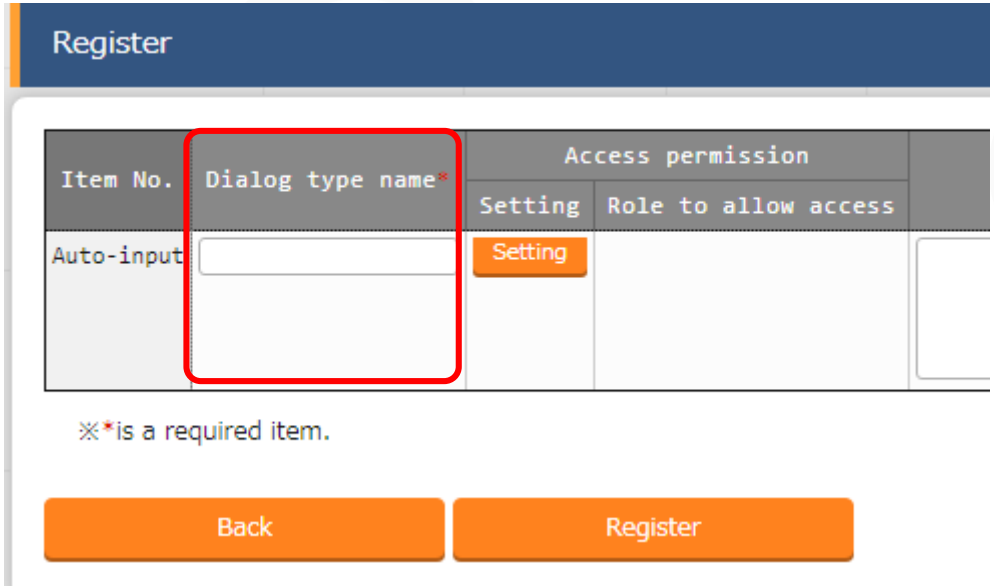
Create dialog type

Next, create "Dialog type".

Menu: **Ansible-Pioneer > Dialog type list**

① Click Register > Start Registration.

② Input the following information for each item and click "Register".



Item name	Input contents
Dialog type name	syslog server specification

Register Dialog files

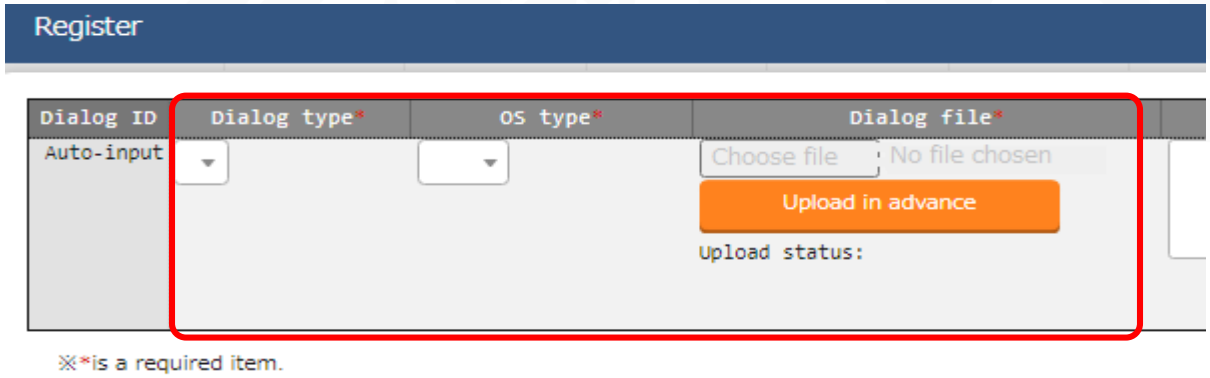
When you're done preparing, register the Dialog file.
We will link all the Dialog types and the OS Types we've created up until this point.

Menu: **Ansible-Pioneer > Dialog files**

① Click Register > Start Registration.

② Choose a Dialog file from "Browse" and click "Upload in advance".

③ Select the following for the other items and click "Register".



Dialog type	OS type	Dialog files
syslog server specification	vyos_RT	vyos_set_syslog_server.yml
syslog server specification	Cisco_L3SW	ios_set_syslog_server.yml

3.4 Movement configuration(4/4)

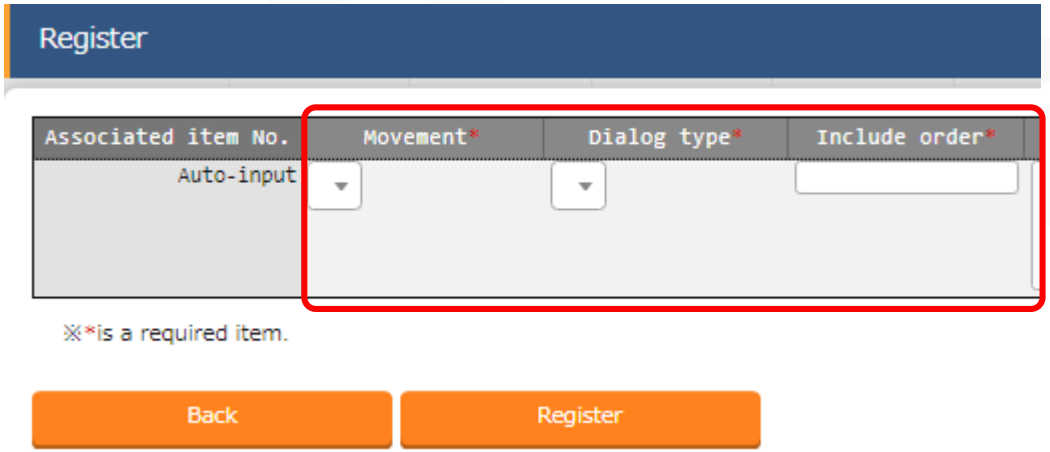
Register the Dialog type to Movement

Link movement and Dialog type.

Menu: **Ansible-Pioneer > Movement dialogue type link**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



Item	Input contents
Movement	Log server configuration
Dialog type	syslog server specification
Include order	1

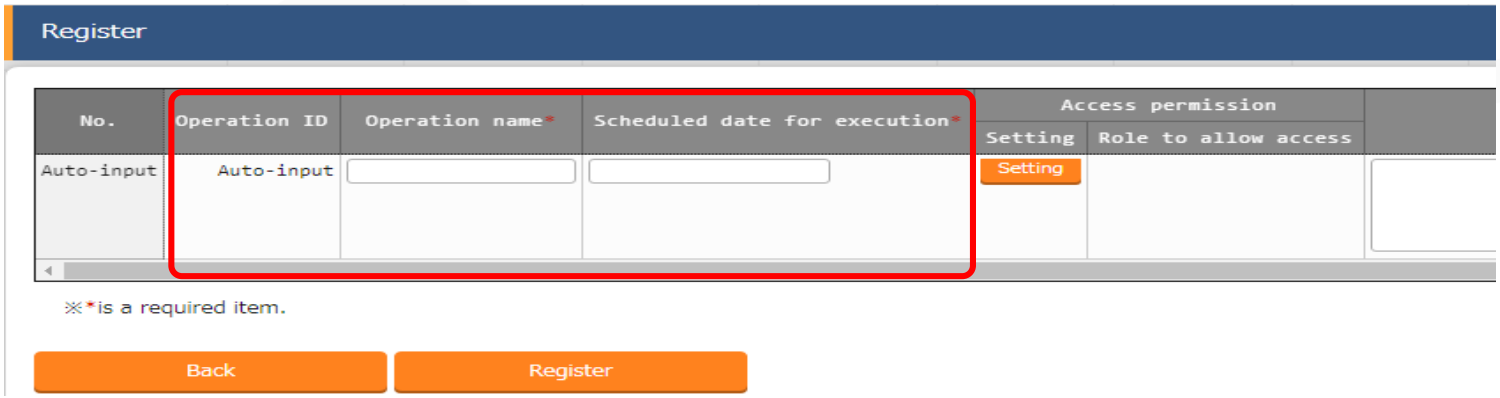
Register new operation

Create operation. Link Movement and Host.

Menu **: Basic Console > Input operation list**

① Click Register > Start Registration.

② Input the following information for each item and click "Register".



Item name	Input contents
Operation name	Pioneer_Practice
Scheduled date for execution	(Enter arbitrary value)

※ "Scheduled date for execution" is just an item for management. It will not be executed automatically.

3.6 Register to device list (1/2)

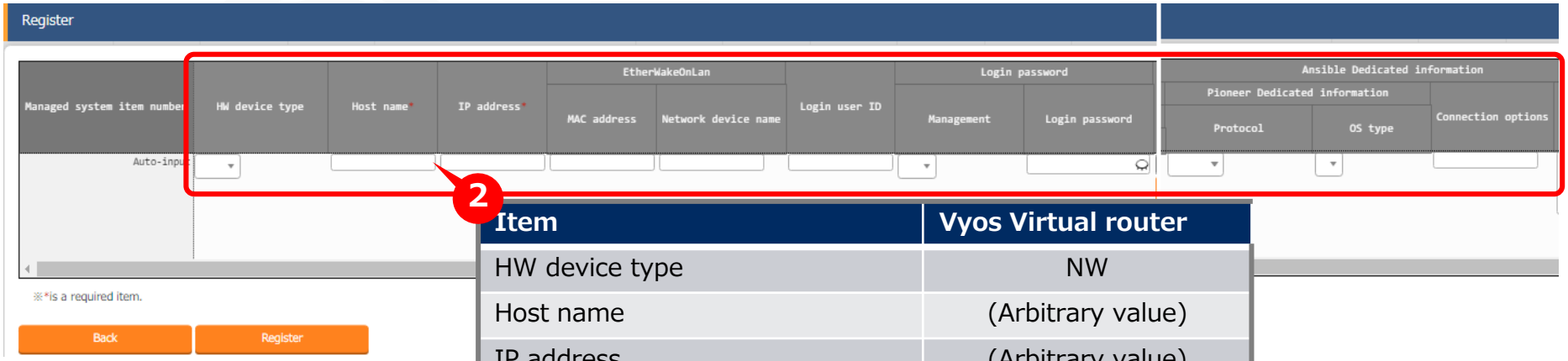
Register NW device to Device list

Register this scenario's operation information from the device list.
Please note that, different from Legacy mode and Legacy-Role mode, you will need to fill in "Pioneer Usage Information".

Menu**: Basic Console > Device list**

① Click Register > Start Registration.

② Select the following information for each item and click "Register".



Item	Vyos Virtual router
HW device type	NW
Host name	(Arbitrary value)
IP address	(Arbitrary value)
Login user ID	(Arbitrary value)
Management	●
Login password	(Arbitrary value)
Pioneer uses information/Protocol	ssh
Pioneer uses information/OS type	vyos_RT

3.6 Register to device list (2/2)

Register NW device to Device list

Do the same for the Cisco device.

Menu**: Basic Console > Device list**

① Click Register > Start Registration.

② Select the following information for each item and click "Register".

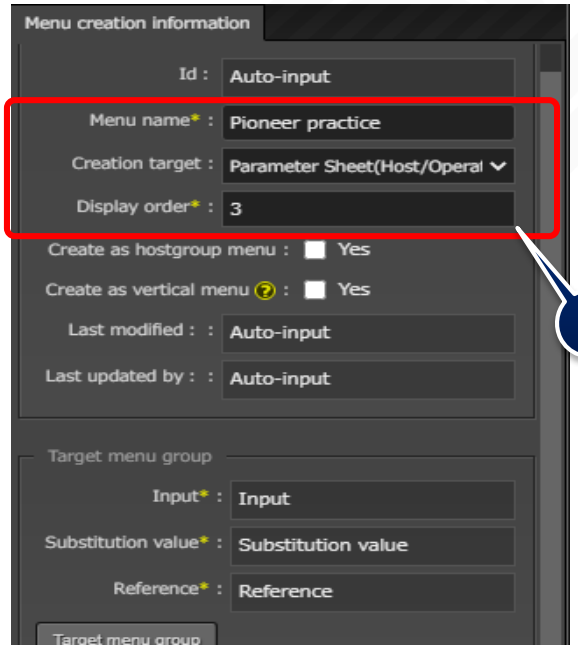
Item	Cisco device
HW device type	NW
Host name	(Free value)
IP Address	(Free value)
Login User ID	(Free value)
Management	●
Login password	(Free value)
Pioneer user information/Protocol	telnet
Pioneer user information/OS type	Cisco_L3SW

Create menu

Create parameter sheet.
create and manage parameters to apply to the target host.

Menu **: Create menu > Create/Define menu**

① Input the following information and see the next page.



1 Input the following information.(Next item)

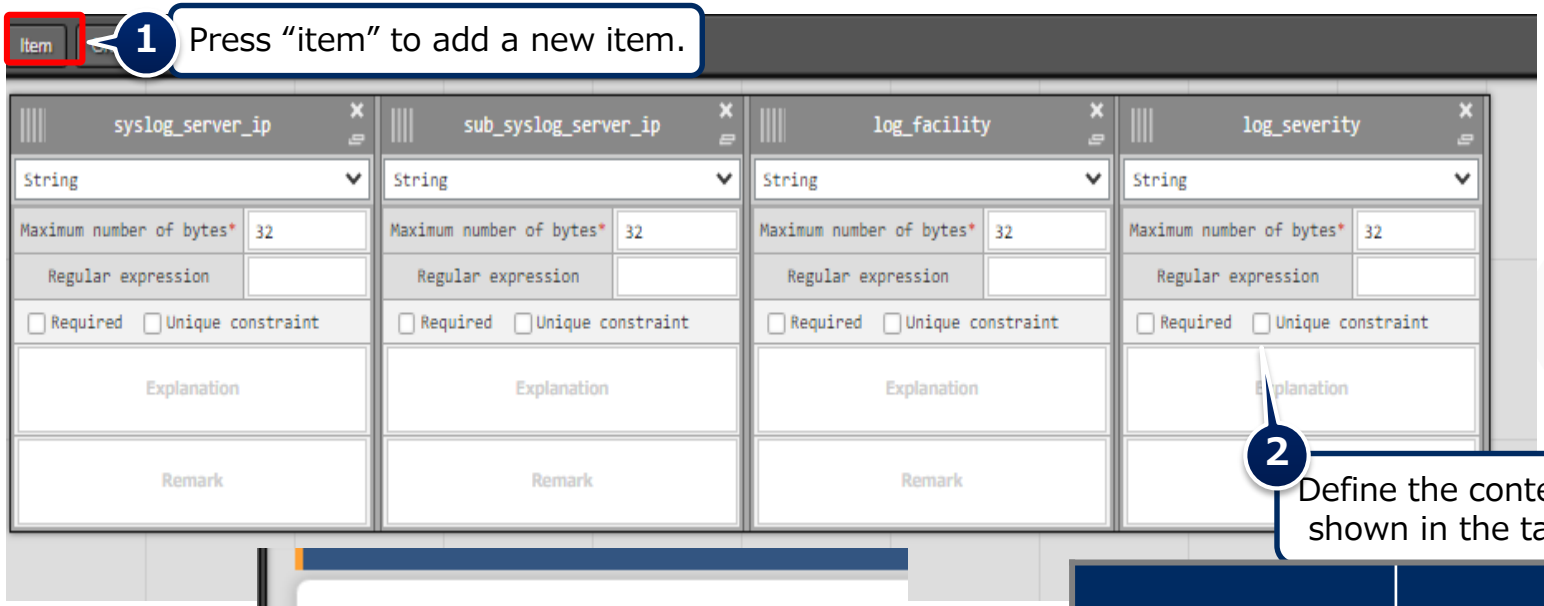
Item	Input contents
Menu name	Pioneer practice
Creation target	Parameter sheet(Host/Operation)
Display order	3

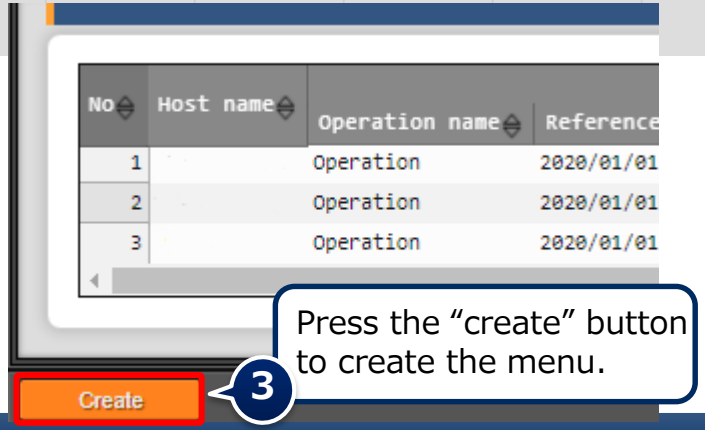
3.7 Parameter sheet creation (2/2)

Define the item name of the parameters sheet

Continuing from the previous section, define the items on the sheet.

Menu **: Create menu > Create/Define menu**





Item name	Input method	Maximum number of bytes
syslog_server_ip	String	32
sub_syslog_server_ip	String	32
log_facility	String	32
log_severity	String	32

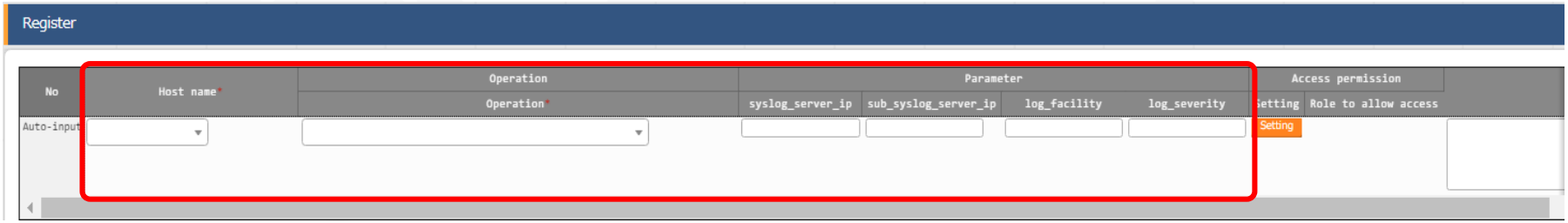
Register data to the parameter sheet

Now that we have created the Menu, let's register the data we're going to use to configure the target host.

Menu**: Input > Pioneer practice (Created Menu)**

① Click Register > Start Registration.

② Select or input the following information for each item and click "Register".



Host name	Operation	syslog_server_ip	sub_syslog_server_ip	log_facility	log_severity
(Cisco device selection)	Pioneer_practice	Arbitrary IP address	Arbitrary IP address	local7	info
(vyos router selection)	Pioneer_practice	Arbitrary IP address	Arbitrary IP address	local7	info

3.9 Substitution value automatic registration setting

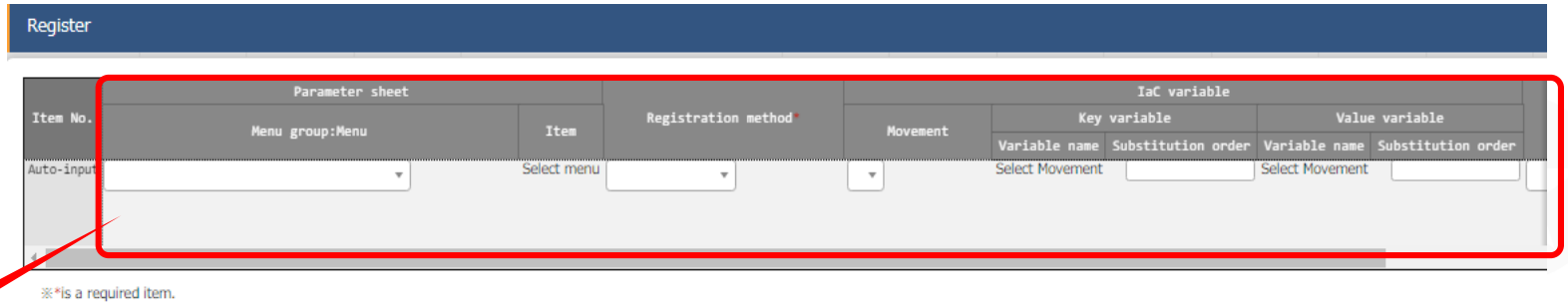
Set Substitute Value Automatic Registration settings

Connect the variables to each item after entering the data in the parameter sheet.

Menu: **Ansible-Pioneer > Substitution value auto-registration setting**

① Click Register > Start Registration.

② Input link settings as shown in the table below and click "Register".



2

Menu	Item	Registration method	Movement	Value variable Variable name	Substitution order
Pioneer practice	syslog_server_ip	Value type	Log server configuration	VAR_syslog_server_ip	1
Pioneer practice	sub_syslog_server_ip	Value type	Log server configuration	VAR_syslog_server_ip	2
Pioneer practice	log_facility	Value type	Log server configuration	VAR_log_facility	Blank
Pioneer practice	log_severity	Value type	Log server configuration	VAR_log_severity	Blank

3.10 Check Substitution value and Target host

Check Substitution value and Target host

Check target host and values specified by substitution value automatic registration.

Menu: **Ansible-Pioneer > Target host&**
Ansible-Pioneer > Substitution value setting

① Click "Filter".

② Check that the correct value is specified by “Pioneer substitution value automatic registration setting procedure".

“Target host” menu

History	Update	Discard	Item No.	Operation	Movement	Host	Last update date/time	Last updated by
History	Update	Discard	1	27:Pioneer_Practice	24:Log server configuration	10:NWHost	2021/08/20 16:50:02	Pioneer substitution value auto-registration set
History	Update	Discard	2	27:Pioneer_Practice	24:Log server configuration	11:NWHost2	2021/08/20 16:50:02	Pioneer substitution value auto-registration set

“Substitute value” menu

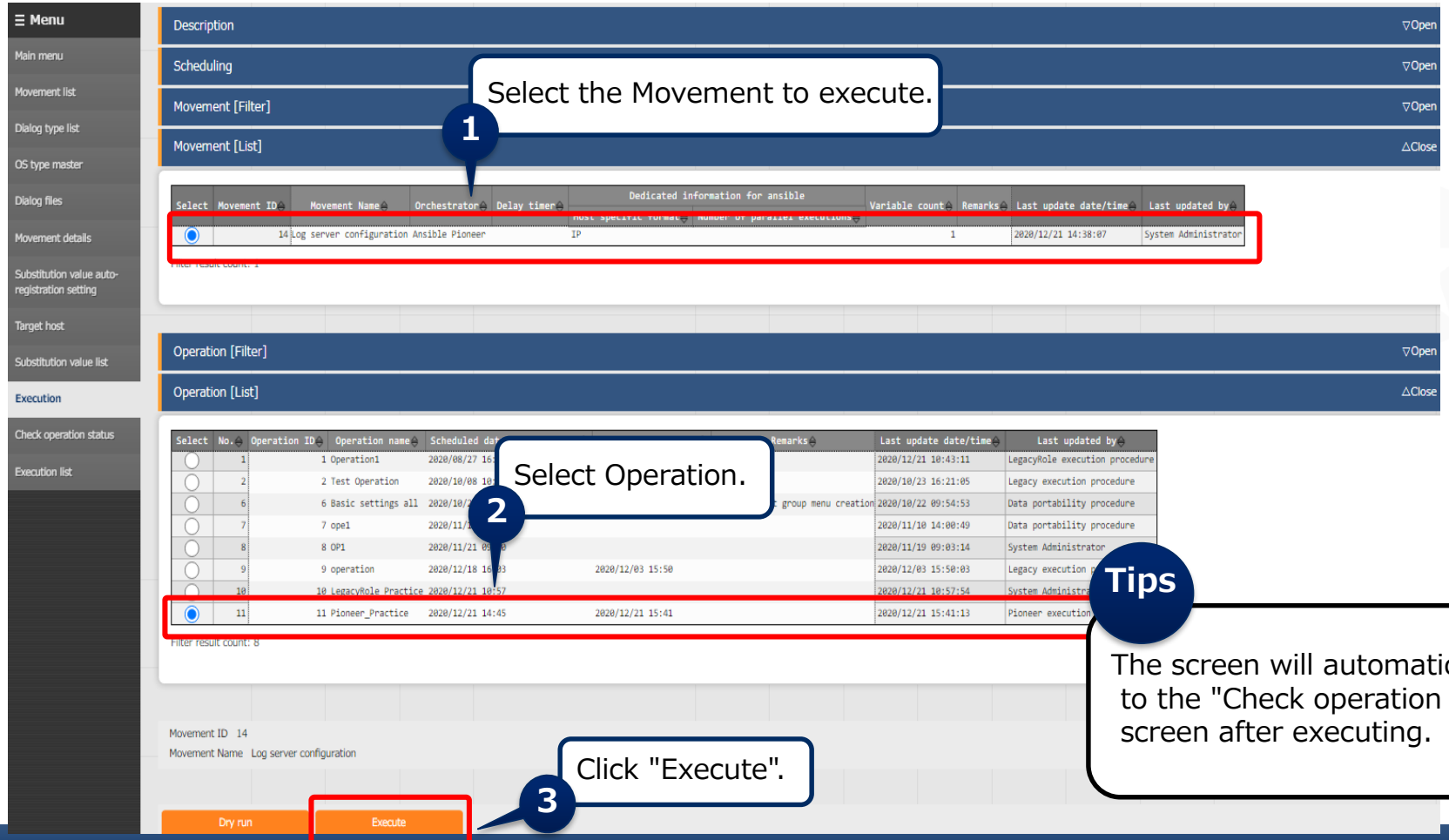
History	Update	Discard	Item No.	Operation	Movement	Host	Variable name	Last update date/time	
History	Update	Discard	1	27:Pioneer_Practice	24:Log server configuration	10:NWHost	2:VAR_syslog_server_ip	2021/08/20 16:50:02	Pioneer substitution value auto-registration set
History	Update	Discard	2	27:Pioneer_Practice	24:Log server configuration	11:NWHost2	2:VAR_syslog_server_ip	2021/08/20 16:50:02	Pioneer substitution value auto-registration set
History	Update	Discard	3	27:Pioneer_Practice	24:Log server configuration	10:NWHost	2:VAR_syslog_server_ip	2021/08/20 16:50:24	Pioneer substitution value auto-registration set
History	Update	Discard	4	27:Pioneer_Practice	24:Log server configuration	11:NWHost2	2:VAR_syslog_server_ip	2021/08/20 16:50:24	Pioneer substitution value auto-registration set
History	Update	Discard	5	27:Pioneer_Practice	24:Log server configuration	10:NWHost	1:VAR_log_severity	2021/08/20 16:50:58	Pioneer substitution value auto-registration set
History	Update	Discard	6	27:Pioneer_Practice	24:Log server configuration	11:NWHost2	1:VAR_log_severity	2021/08/20 16:50:58	Pioneer substitution value auto-registration set
History	Update	Discard	7	27:Pioneer_Practice	24:Log server configuration	10:NWHost	1:VAR_log_severity	2021/08/20 16:52:15	Pioneer substitution value auto-registration set
History	Update	Discard	8	27:Pioneer_Practice	24:Log server configuration	11:NWHost2	1:VAR_log_severity	2021/08/20 16:52:15	Pioneer substitution value auto-registration set

3.11 Execution (1/2)

Execute Movement directly

The number of Movements created in this scenario is 1. There is no need to create any Conductors. Execute them individually from the "Execution" menu.

Menu : **Ansible-Pioneer > Execution**



Tips

The screen will automatically change to the "Check operation status" screen after executing.

3.11 Execution (2/2)

Check execution results

Executing the operation will transfer the user to a screen where Execution status and logs are displayed.

Menu : **Ansible-Pioneer > Check operation status**

Execution status

Target Operation

Item			Value
Execution No.			51
Execution type			Normal
Status			Completed
execution engine			Ansible Engine
Caller symphony			
Caller conductor			Sample1
Execution user			System Administrator
Movement	ID		1
	Name		Legacy1
	Delay timer (minutes)		
	Dedicated information for ansible	Host specific format	IP
		WinRM connection	
Operation	No.		1
	Name		Operation1
	ID		1
Host management			confirmation
Substitution value			confirmation
Input data	Populated data		InputData_0000000051.zip
Output data	Result data		ResultData_0000000051.zip
Operation status	Scheduled date/time		
	Start date/time		2020/11/11 08:44:45
	End date/time		2020/11/11 08:44:58

Log

Progress status (Execution status)

Tips

Users can download a zip file that contains input and result data.

Tips

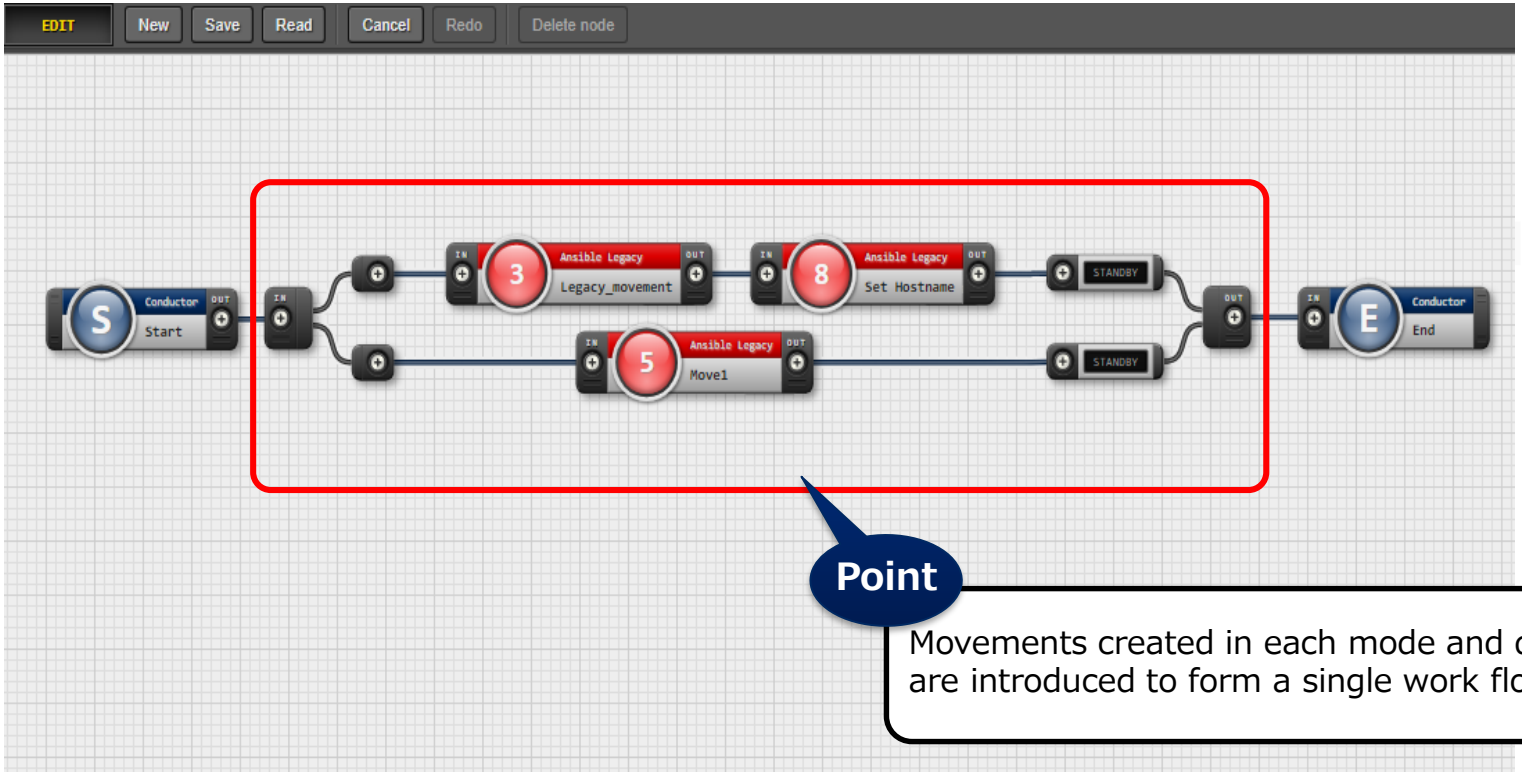
If you want to check the results of the execution by using commands, you can use the commands below to check the log settings.
IOS - “# show logging”
vyos - ”$ show configuration”

A) Appendix

Appendix 1) Bundling and running 3 modes in Conductor

Bundle up the 3 modes and execute them

In this document, we executed the work for each mode individually, but users can also create a work flow that executes work in multiple modes by using Conductor.




Exastro