



# IT Automation

## Ansible Driver [Practice]

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

Exastro IT Automation ver 1.9.0  
Exastro developer

# Table of contents

## ■ Introduction

### ① Ansible-Legacy

1. [Work environment and Scenario](#)
2. [Required file creation](#)
3. [Moment configuration](#)
4. [Conductor creation](#)
5. [Operation registration](#)
6. [Register to device list](#)
7. [Parameter sheet creation](#)
8. [Data registration](#)
9. [Substitution value automatic registration configuration](#)
10. [Check Substitution value and Target host](#)
11. [Execution](#)

### ② Ansible-LegacyRole

1. [Work environment and Scenario](#)
  2. [Role package preparation](#)
  3. [Moment configuration](#)
  4. [Operation registration](#)
  5. [Register to device list](#)
  6. [Parameter sheet creation](#)
  7. [Data registration](#)
  8. [Substitution value automatic registration setting](#)
  9. [Check Substitution value and Target host](#)
  10. [Execution](#)
- A) [Appendix](#)

### ③ Ansible-Pioneer

1. [Work environment and Scenario](#)
2. [Dialog file creation](#)
3. [OS type creation](#)
4. [Moment configuration](#)
5. [Operation registration](#)
6. [Register to device list](#)
7. [Parameter sheet creation](#)
8. [Data registration](#)
9. [Substitution value automatic registration configuration](#)
10. [Check Substitution value and Target host](#)
11. [Execution](#)

# Introduction - How to use this document

## 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



# 1.1 Work environment and Scenario

## Work environment

The work environment used in this document is as follows.(The same as Chapter 2 Ansible-Legacy Role)

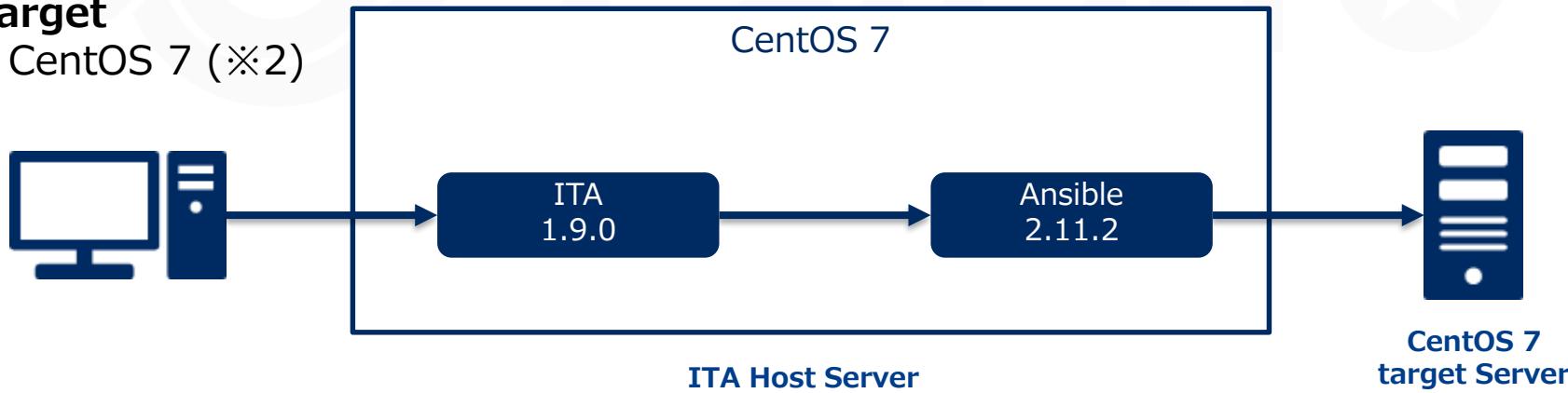
Please prepare a server in addition to the ITA Host server. The additional server will be target for operation.

### ITA Host Server

- CentOS 7 (※1)
- ITA 1.9.0
- 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 Work environment and Scenario

## Scenario

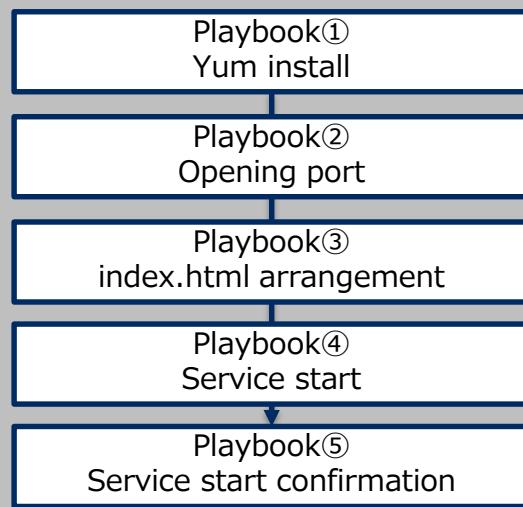
You can experience the **reusability of Playbooks** by going through a task that consists of three major steps.

- ① Combine Movements and create Conductor.
- ② Create menu and register parameters.
- ③ Execute the created Conductor.

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

## Scenario Image

### ① Create one Conductor



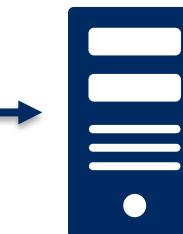
### ② Create Parameter sheet

(Manage the substitution values to the Playbook and work target)



### ③ Check the addition of both services.

- Apache
- Tomcat



Target Server

# 1.2 Creation of required files(1/3)

## Playbook creation

This scenario will create the following five Playbooks.

[Attention] Create the yml file with Character code " UTF-8" and New line code "LF".

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

### File name : 1-yum\_install.yml

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

```
- 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

Install and run firewalld.  
Allows communication to the specified port.

# 1.2 Creation of required files(2/3)

## 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"'
```

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

```
- 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: 3-copy\_index.yml

Copies and arranges files

### File name: 4-start\_service.yml

Starts the specified service.

### File name: 5-check\_service.yml

Checks if the service has started.

# 1.2 Creation of required files (3/3)

## index.html creation

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

### File name: index.html

Place this html file directory under document root.

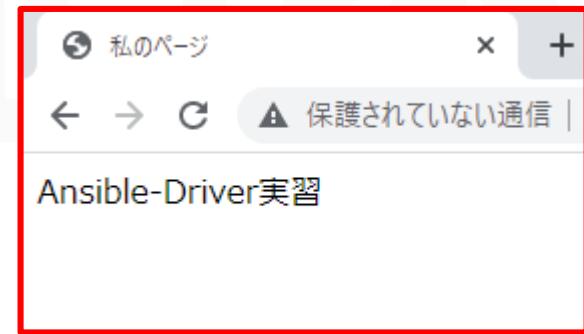
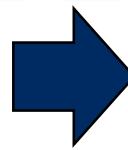
```
<html>
<head>
<title>my page</title>
</head>
<body>

Ansible-Driver practice

<br>

</body>
</html>
```

When correct index.html file is placed after operation is completed, following screen should appear.



# 1.3 Movement configuration(1/4)

## 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".

The screenshot shows the 'Register' screen for creating a new movement. The 'Movement Name\*' field is highlighted with a red box and a red callout pointing to a table below. The table lists three movements: 'Install\_Packages', 'Open\_Ports', and 'Start\_Service', each associated with 'IP' as the host specific format.

Movement name	Host specific format
Install_Packages	IP
Open_Ports	IP
Start_Service	IP

# 1.3 Movement configuration(2/4)

## 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".

The screenshot shows the 'Register' interface in Ansible-Legacy. On the left, there's a form for registering a new playbook:

Playbook ID	Playbook name*	Playbook files*	Access permission
		Setting Role to allow access	
Auto-input	<input type="text"/>	<input type="button" value="Choose file"/> No file chosen <input type="button" value="Upload in advance"/> Upload status: <span style="color: #00A0A0;">Uploading</span>	<input type="button" value="Setting"/>

A red box highlights the 'Playbook files\*' section, and a red arrow labeled '3' points to the 'Upload in advance' button. Below the form is a note: \* is a required item.

At the bottom are 'Back' and 'Register' buttons.

On the right, a table lists registered playbooks:

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

# 1.3 Movement configuration(3/4)

## 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".

The screenshot shows the 'Register' interface for file upload. A red box highlights the 'File embedded variable name\*' field, and a red circle with the number '3' points to it. Below the interface, a summary table provides the registered file details:

File embedded variable name	File
CPF_httpd_conf	httpd_config.txt

# 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".

Register

Associated item No.	Movement*	Playbook files*	Include order*	Access permission	
	Setting	Role to allow access			
Auto-input	<input type="button" value="▼"/>	<input type="button" value="▼"/>	<input type="text"/>	<input type="button" value="Setting"/>	

\* is a required item.

2 Back Register

Associate Table

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

Point

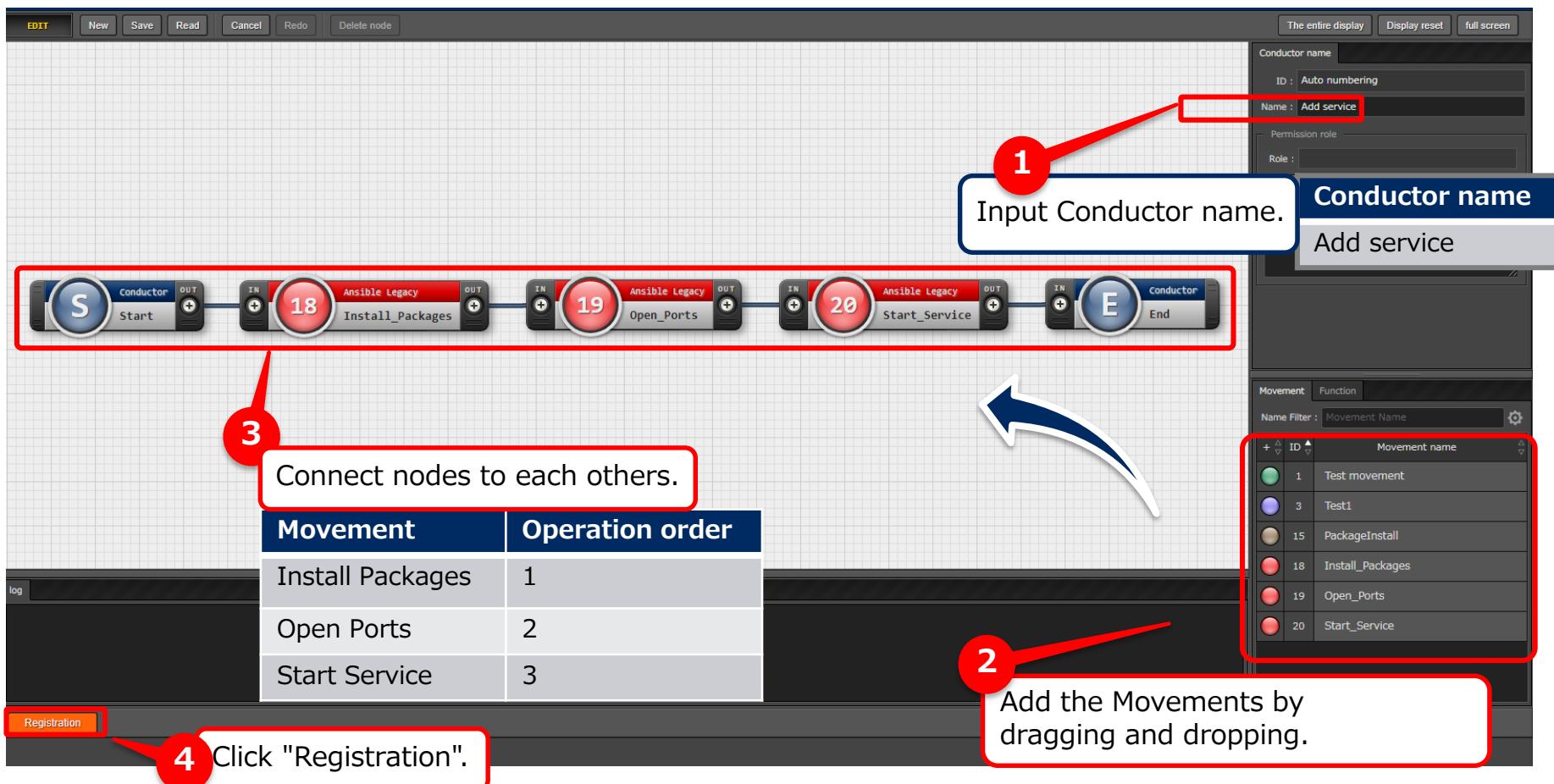
"Start Service" links the 3 Playbook files and executes them together.

# 1.4 Conductor creation

## Create Conductor

Create a Conductor that collects the defined Movements.

Menu: Conductor > Conductor class edit



# 1.5 Operation registration

## 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".

The screenshot shows the 'Register' screen for creating a new operation. At the top, there's a table with columns: No., Operation ID, Operation name\*, Scheduled date for execution\*, and Access permission. A red box highlights the first three columns. Below the table, a note says '\* is a required item.' A red arrow points from the number '2' to the 'Operation name' field in a second table at the bottom. This second table has two rows, each with 'Operation name' and 'Scheduled date for execution' fields. Both fields in these rows contain placeholder text '(Enter arbitrary value)'. The 'Operation name' field is also highlighted with a red box.

No.	Operation ID	Operation name*	Scheduled date for execution*	Access permission	
				Setting	Role to allow access
Auto-input	Auto-input	<input type="text"/>	<input type="text"/>	<input type="button" value="Setting"/>	<input type="text"/>

\* is a required item.

2

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".

The screenshot shows the 'Register' page in the Basic Console. It has two main sections: 'Managed system item number' and 'EtherWakeOnLan'. The 'Managed system item number' section contains fields for 'HW device type', 'Host name\*', 'IP address\*', 'MAC address', and 'Network device name'. The 'EtherWakeOnLan' section contains fields for 'Login user ID', 'Management', 'Login password', 'ssh authentication key file', and 'Authentication method'. A red box highlights the 'HW device type', 'Host name\*', and 'IP address\*' fields. Another red box highlights the 'Management', 'Login password', 'ssh authentication key file', and 'Authentication method' fields. A red callout points to a table below, which lists the items and their input contents. The table has columns for 'Item' and 'Input contents'.

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

Menu creation information

<b>Id :</b> Auto-input
<b>Menu name*</b> : Legacy practice
<b>Creation target :</b> Parameter Sheet(Host/Operal ▾)
<b>Display order*</b> : 1
<b>Create as hostgroup menu :</b> <input type="checkbox"/> Yes
<b>Create as vertical menu</b> ⓘ <input type="checkbox"/> Yes
<b>Last modified :</b> Auto-input
<b>Last updated by :</b> Auto-input
<b>Target menu group</b>
<b>Input*</b> : Input
<b>Substitution value*</b> : Substitution value
<b>Reference*</b> : Reference
<b>Target menu group</b>
<b>Permission role</b>
<b>Role :</b> <input type="text"/>
<b>Permission role select</b>
<b>Explanation</b>

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

- ① Click item.
- ② Select or input the following information for each item and click "create".

The screenshot shows the 'Create menu > Create/Define menu' window. On the left, there is a table with four columns: package\_name, package\_name\_sub, port\_number, and service\_name. Each column has dropdown menus for type ('String'), maximum bytes ('32'), regular expression, and checkboxes for 'Required' and 'Unique constraint'. Below the table are sections for 'Explanation' and 'Remark'. A red circle labeled '1' points to the 'Item' button in the top-left corner of the window. A red circle labeled '2' points to a table on the right where four items are being defined:

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

Below this table is a smaller window showing a list of operations with columns: No, Host name, Operation name, and Reference. The list contains three entries, each with 'Operation' in the 'Operation name' column and '2020/01/01' in the 'Reference' column. A red circle labeled '2' points to the 'Create' button at the bottom of this window.

# 1.8 Data registration

## 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".

The screenshot shows a software interface titled 'Register'. At the top, there's a header bar with the title. Below it is a table with columns: 'No', 'Host name\*', 'Operation', and 'Parameter'. Under 'Host name\*', there's a dropdown menu with 'Auto-input' selected. Under 'Operation', there's a dropdown menu. The 'Parameter' column contains four input fields: 'package\_name', 'package\_name\_sub', 'port\_number', and 'service\_name'. A large red box highlights the 'Host name\*' and 'Operation\*' fields.

2

The screenshot shows a table titled 'Legacy practice' with a red border. The table has six columns: 'Host name', 'Operation', 'package\_name', 'package\_name\_sub', 'port\_number', and 'service\_name'. There are two rows of data. The first row has 'Host name' as '(Target host)', 'Operation' as 'Install Apache', 'package\_name' as 'httpd', 'package\_name\_sub' as '(blank)', 'port\_number' as '80/tcp', and 'service\_name' as 'httpd'. The second row has 'Host name' as '(Target host)', 'Operation' as 'Install Tomcat', 'package\_name' as 'tomcat', 'package\_name\_sub' as 'tomcat-webapps', 'port\_number' as '8080/tcp', and 'service\_name' as 'tomcat'. A red box highlights the entire table.

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".

Item No.	Parameter sheet(From)		Registration method*	IaC variable(To)	
	Menu group:Menu*	Item*		Movement*	Key variable
Variable name	Substitution order	Variable name	Substitution order		
Auto-input	Select menu	Select Movement	Select Movement	Select Movement	Select Movement

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**

History	Duplicate	Update	Discard	Item No.	Operation	Movement	Host	Substitution value management	Access permission	Remarks	Last update date/time	Last updated by
History	Duplicate	Update	Discard	1	3:Install Apache	1:Install Packages	1:targethost	Substitution value management			2022/01/26 15:44:37	System Administrator
History	Duplicate	Update	Discard	2	4:Install Tomcat	1:Install Packages	1:targethost	Substitution value management			2022/01/26 16:06:16	Legacy substitution value auto-registration sett:
History	Duplicate	Update	Discard	3	3:Install Apache	2:Open Ports	1:targethost	Substitution value management			2022/01/26 16:06:54	Legacy substitution value auto-registration sett:
History	Duplicate	Update	Discard	4	4:Install Tomcat	2:Open Ports	1:targethost	Substitution value management			2022/01/26 16:06:54	Legacy substitution value auto-registration sett:
History	Duplicate	Update	Discard	5	3:Install Apache	3:Start_Service	1:targethost	Substitution value management			2022/01/26 16:08:09	Legacy substitution value auto-registration sett:
History	Duplicate	Update	Discard	6	4:Install Tomcat	3:Start_Service	1:targethost	Substitution value management			2022/01/26 16:08:09	Legacy substitution value auto-registration sett:

**Substitution value list**

History	Duplicate	Update	Discard	Item No.	Operation	Movement	Host	Variable name	Specific value			Last updated by
									Sensitive setting	String	File	
History	Duplicate	Update	Discard	1	3:Install Apache	1:Install_Packages	1:targethost	1:VAR_package_name	OFF			2022/01/26 15:51:07
History	Duplicate	Update	Discard	2	3:Install Apache	1:Install_Packages	1:targethost	1:VAR_package_name	OFF	httpd		2022/01/26 16:06:16
History	Duplicate	Update	Discard	3	4:Install Tomcat	1:Install_Packages	1:targethost	1:VAR_package_name	OFF	tomcat		2022/01/26 16:06:16
History	Duplicate	Update	Discard	4	4:Install Tomcat	1:Install_Packages	1:targethost	1:VAR_package_name	OFF	tomcat-webapps		2022/01/26 16:06:33
History	Duplicate	Update	Discard	5	3:Install Apache	2:Open_Ports	1:targethost	2:VAR_port_number	OFF	80/tcp		2022/01/26 16:06:54
History	Duplicate	Update	Discard	6	4:Install Tomcat	2:Open_Ports	1:targethost	2:VAR_port_number	OFF	8080/tcp		2022/01/26 16:06:54
History	Duplicate	Update	Discard	7	3:Install Apache	3:Start_Service	1:targethost	3:VAR_service_name	OFF	httpd		2022/01/26 16:08:09
History	Duplicate	Update	Discard	8	4:Install Tomcat	3:Start_Service	1:targethost	3:VAR_service_name	OFF	tomcat		2022/01/26 16:08:09

# 1.11 Execution (1/3)

## 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

The screenshot shows the 'Conductor Execution' menu selected in the left sidebar. Two execution lists are displayed:

- Conductor [List]**: Shows four conductors with details like name, creation date, and last update by. The first conductor is highlighted with a red box and a callout labeled '1'.

Select	Conductor class ID	Conductor name	Explanation	Remarks	Last update date/time	Last updated by
<input type="checkbox"/>	1	Server basic setting			2020/09/02 10:28:03	Data portability procedure
<input type="checkbox"/>	2	KATO TEST1104			2020/11/04 15:16:16	Data portability procedure
<input type="checkbox"/>	3	Conductor1			2020/11/10 16:26:42	Data portability procedure
<input type="checkbox"/>	4					

Filter result count: 4
- Operation [List]**: Shows several operations with details like name, scheduled date, last execution date, and last update by. The second operation is highlighted with a red box and a callout labeled '2'.

Select	No.	Operation ID	Operation name	Scheduled date for execution	Last execution date	Remarks	Last update date/time	Last updated by
<input type="checkbox"/>	1	1	1 Operation1	2020/08/27 16:15	2020/12/04 09:21		2020/12/04 09:21:54	Legacy execution procedure
<input type="checkbox"/>	2	2	Test Operation	2020/10/08 10:00	2020/10/23 16:21		2020/10/23 16:21:05	Legacy execution procedure
<input type="checkbox"/>	6	6	Basic settings all	2020/10/24 09:54		Test for Host group menu creation	2020/10/24 09:54:53	Data portability procedure
<input type="checkbox"/>	7	7	open	2020/11/10 14:00			2020/11/10 14:00:49	Data portability procedure
<input type="checkbox"/>	8	8	OPI	2020/11/21 09:20			2020/11/19 09:03:14	System Administrator
<input type="checkbox"/>	9	9	operation				2020/12/03 15:50:03	Legacy execution procedure

**Click "Execution" at the bottom of the screen.**

**2** Select Operation.

**3** Execution

**Tips**

The screen will automatically change to the "Conductor Confirmation" screen after executing.

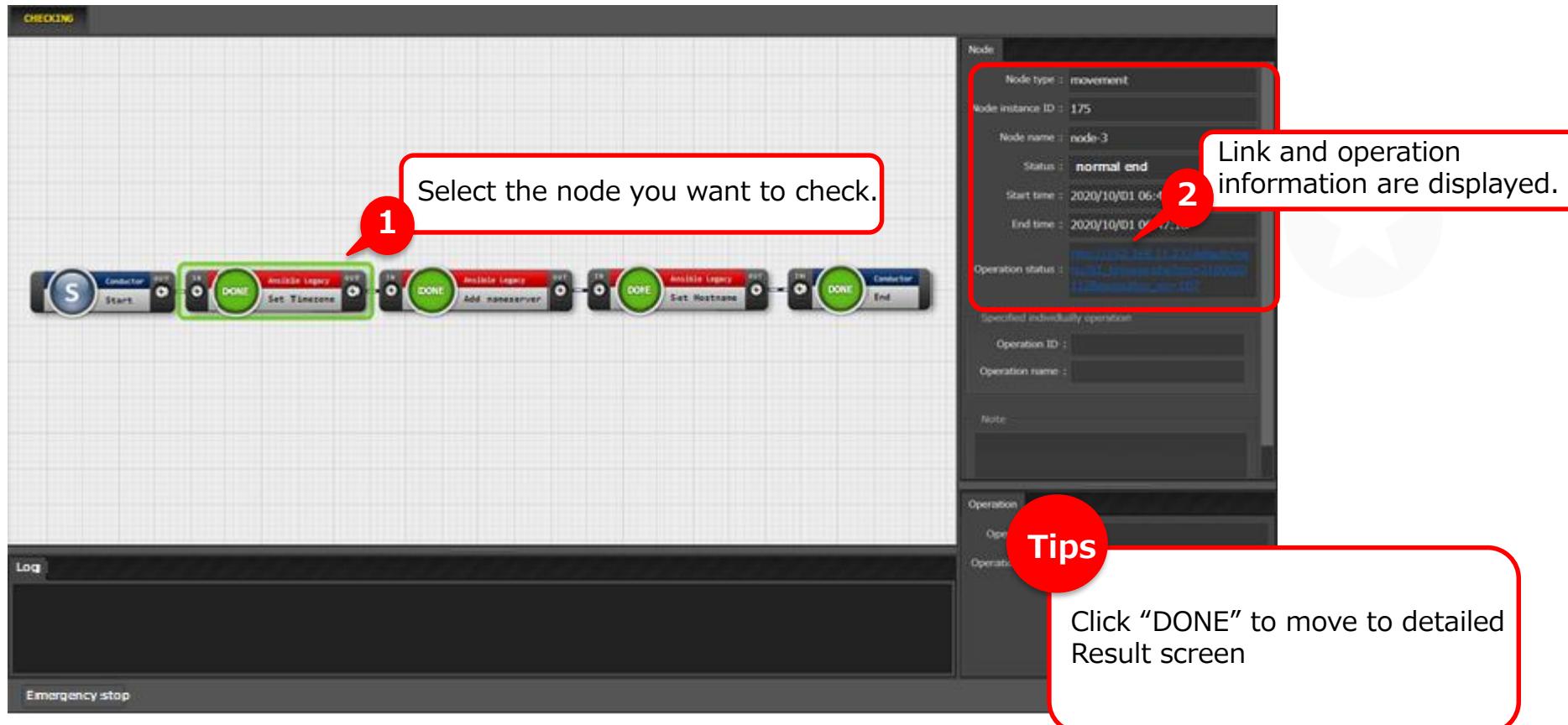
# 1.11 Execution (2/3)

## 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.

The screenshot shows two main sections: 'Target Operation' and 'Progress status(Execution log)'.

**Target Operation:** This section displays various configuration parameters for the execution:

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

**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](http://(Host IP address):80)  
Tomcat- [http://\(Host IP address\):8080](http://(Host IP address):8080)

# Chapter 2 Ansible-LegacyRole



# 2.1 Work 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)

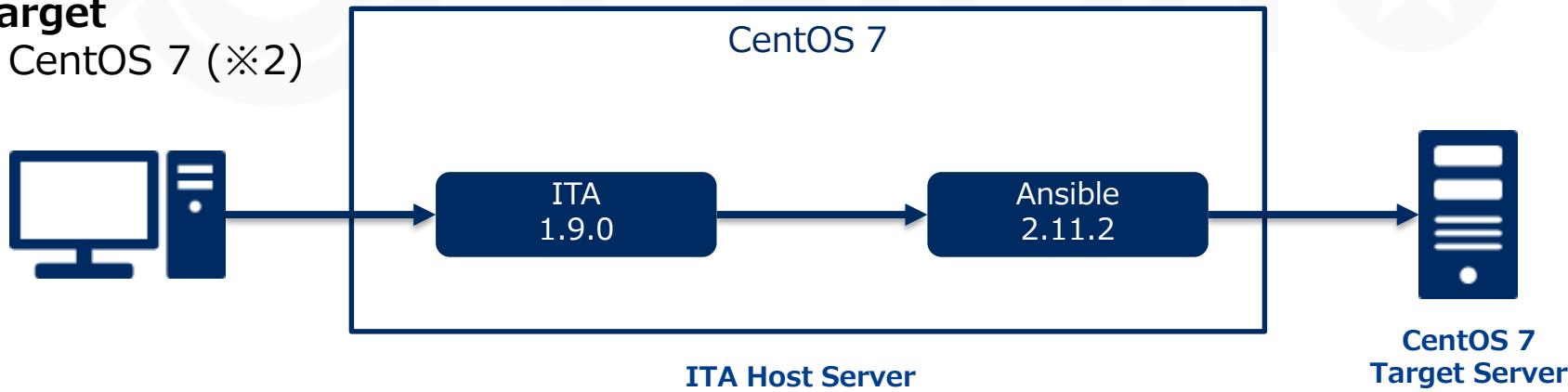
Please prepare a server in addition to the ITA Host server. The additional server will be target for operation.

### ITA Host Servers

- CentOS 7 (※1)
- ITA 1.9.0
- 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.

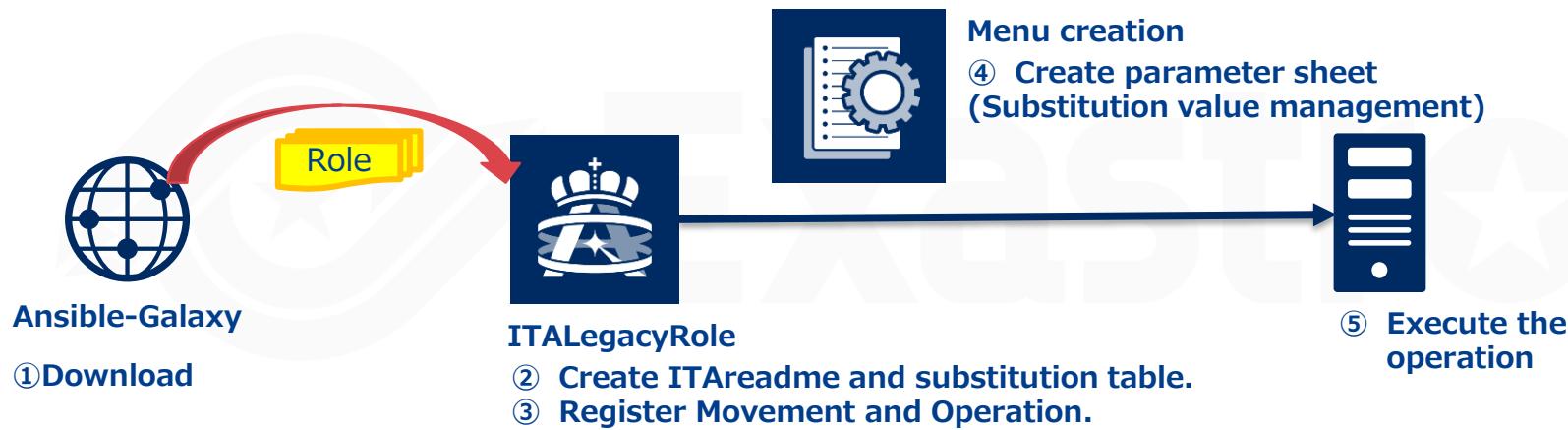
※2 Any OS compatible with Ansible can be used.

## 2.1 Work environment and Scenario (2/3)

### Scenario image

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 Work environment and Scenario (3/3)

## Role download

First, access the URL below and download the role.

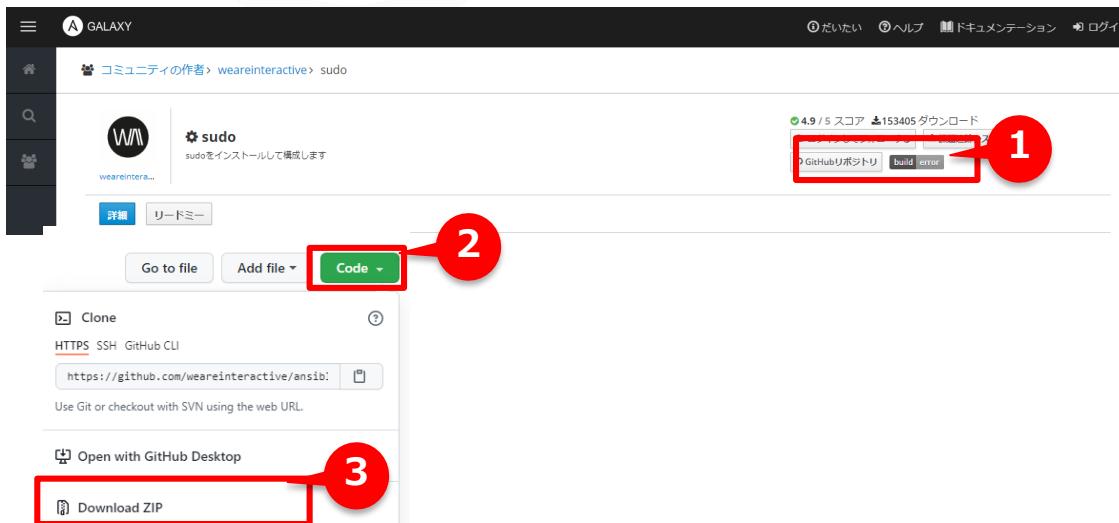


<https://galaxy.ansible.com/weareinteractive/sudo>

Role package that adds files under /etc/sudoers.d

## Download method

- ① Click Github repository
- ② Click Code
- ③ Click DownLoadZIP



## 2.2 Role package preparation(1/4)

### 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.

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.

```
---  
# sudo_defaults:  
#(omission) ~~~~  
# 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  
~~~
```

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

1

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

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

2

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

## 2.2 Role package preparation(2/4)

### ITAreadme descriptions

ita\_readme is a configuration file for adding and changing variable definitions.  
※ For details about ITAreadme, please refer to this [Manual](#).

File name: ita\_readme\_ansible-sudo-master.yml

```
sudo_users:  
  - name:
```

#### Image

Change to the correct mapping.

1

```
defaults/main.yml  
sudo_package: sudo  
sudo_users: []
```

Fill in the correct mapping.

2

```
ITAreadme  
sudo_users:  
  - name:
```

Actual variables used

```
sudo_package: sudo  
sudo_users:  
  - name: example_name
```



Substitution value  
management

Variable name	Member variable name	Specific value
LCA_sudo_users	[0].name	example_name



The substitution table will be described in the next section.

## 2.2 Role package preparation(3/4)

### 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
```

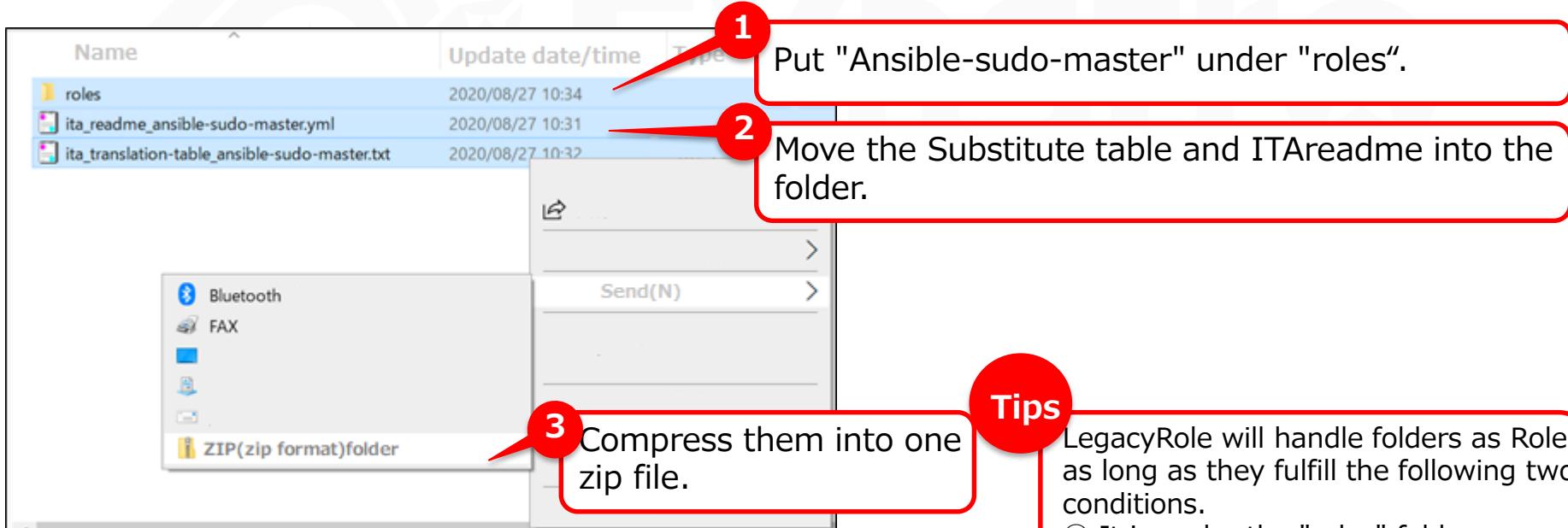
Variable name	Member variable name (Added ITAreadme)	Specific value
LCA_sudo_users	[0].name	example_name
LCA_sudo_sudoers_file		example_sudoers

## 2.2 Role package preparation(4/4)

### 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.



## 2.3 Movement configuration(1/3)

### 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".

The screenshot shows the 'Movement list' registration interface. A red box highlights the 'Movement Name\*' field, which is required. A red arrow labeled '2' points from this field to a secondary input dialog box below. This dialog contains two rows: 'Item name' (Movement name) with 'Input contents' (RegisterSudoer), and 'Item name' (Host specific format) with 'Input contents' (IP).

Register

Movement ID	Movement Name*	Delay timer	Host specific format*	Dedicated information for ansible	
Auto-input				WinRM connection	Header section

※\*is a required item.

Back Register

2

Item name	Input contents
Movement name	RegisterSudoer
Host specific format	IP

## 2.3 Movement configuration(2/3)

### 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".

The screenshot shows a user interface for registering a role package. At the top, there's a blue header bar with the word 'Register'. Below it is a table with two columns. The first column has a 'Role package name\*' label and a text input field containing 'Auto-input'. The second column has a 'Role package file (ZIP format)\*' label, a 'Choose file' button, and a message 'No file chosen'. Below these are two buttons: 'Upload in advance' (orange) and 'Upload status:' (blue). To the right of the table, there's a separate dark blue box containing the text 'Role package name' followed by 'sudo-master'. A red box highlights the 'Role package name' input field and the 'Role package file' section. A red circle with the number '3' points to the 'Role package name' input field.

Item No.	Role package name*	Role package file (ZIP format)*
Auto-input	<input type="text"/>	<input type="button" value="Choose file"/> No file chosen <input type="button" value="Upload in advance"/> Upload status:

**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".

Register

Associated item No.	Movement*	Role package name	Role name	Include order*
Auto-input	<input type="text"/>	<input type="text"/>	Select role package	<input type="text"/>

\* is a required item.

2

<a href="#">Back</a>	<a href="#">Register</a>		
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

## 2.4 Operation configuration

### 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".

The screenshot shows the 'Input operation list' registration screen. At the top, there is a table with columns: No., Operation ID, Operation name\*, Scheduled date for execution\*, Access permission, and Role to allow access. The first row contains 'Auto-input' in the Operation ID column. A red box highlights this row. Below the table, a note says '\* is a required item.' At the bottom, there are two orange buttons: 'Back' and 'Register'. A red circle with the number '2' points to the 'Register' button. Below the buttons, there is a table with two columns: Item name and Input contents. It contains two rows: one for 'Operation name' with value 'LegacyRole\_Practice' and one for 'Scheduled date for execution' with value '(Enter arbitrary value)'. A red box highlights this table.

No.	Operation ID	Operation name*	Scheduled date for execution*	Access permission	Role to allow access
Auto-input	Auto-input			Setting	

\* is a required item.

Back 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.

## 2.5 Register to device list

### 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".

Register

Managed system item number	HW device type	Host name*	IP address*	EtherWakeOnLan		Login user ID	Management	Login password	ssh authentication key file	Authentication method
Auto-input				MAC address	Network device name				Choose file No file chosen	
									<input type="button" value="Upload in advance"/>	
Upload status:										

\* is a required item.

Back 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

## 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

Menu creation information

Id :	Auto-input
Menu name*	LegacyRole practice
Creation target :	Parameter Sheet(Host/Operal ▾
Display order*	2
Create as hostgroup menu : <input type="checkbox"/> Yes	
Create as vertical menu  <input type="checkbox"/> Yes	
Last modified :	Auto-input
Last updated by :	Auto-input
Target menu group	
Input*	Input
Substitution value*	Substitution value
Reference*	Reference
Target menu group	
Permission role	
Role :	
Permission role select	
Explanation	

1

Input the following information.(Next item)

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 input the following information for each item and click "Create".

The screenshot shows the software interface for creating parameter items. At the top, there is a toolbar with buttons for 'Item' (highlighted with a red circle and number 1), 'Repeat', 'Cancel', and 'Redo'. Below the toolbar, two parameter item definitions are shown side-by-side:

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

Each item has fields for 'Type' (String), 'Maximum number of bytes' (32), 'Regular expression', and checkboxes for 'Required' and 'Unique constraint'. Below these definitions is a table of operations:

No	Host name	Operation name	Reference
1		Operation	2020/01/01
2		Operation	2020/01/01
3		Operation	2020/01/01

At the bottom of the interface, there is a large orange 'Create' button highlighted with a red circle and number 2.

## 2.7 Data registration

### 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".

The screenshot shows a software interface titled 'Register'. At the top, there is a table with columns: No, Host name\*, Operation, and Parameter (with sub-columns sudoer\_name and sudoer\_filename). Below this is a list of entries, with the first entry highlighted by a red box and a red number '2' indicating the second step. The entry details are: Host name (Auto-input), Operation (Operation\*), sudoer\_name (sudoer\_name), and sudoer\_filename (sudoer\_filename). A second, smaller red box highlights the 'Host name' and 'Operation' fields in the list view, likely indicating where to click to select values from dropdown menus.

No	Host name*	Operation	Parameter
		Operation*	sudoer_name    sudoer_filename
Auto-input			

2

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".

Register

Item No.	Parameter sheet(From)		Registration method*	IaC variable(To)			
	Menu group:Menu	Item		Key variable		Value variable	
		Variable name	Member variable name	Substitution order	Variable name	Member variable name	Substitution order
Auto-input	Select menu	Select Movement	Select variable name		Select Movement	Select variable name	
* * a required item.							
2	Back	Register					
Menu group : Menu	Item	Registration method	Movement	Value variable Variable name	Value variable member variable name		
LegacyRole for practice	sudoer_name	Value type	RegisterSud oer	LCA_sudo_users	[0].name		
LegacyRole for practice	sudoer_filename	Value type	RegisterSud oer	LCA_sudo_sudoers_file	(blank)		

## 2.9 Check Substitution value and Target host

### 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

List/Update										
History	Update	Discard	Item No.	Operation	Movement	Host	Substitution value management	Access permission	Remarks	Last update date/time
							Role to allow access			
History	Update	Discard	226:LegacyRole_Practice 23:RegisteSudoer 1:Host	Substitution value management					2021/08/20 16:07:47	LegacyRole substitution value auto-registration :

#### Substitution value list

List/Update										
History	Update	Discard	Item No.	Operation	Movement	Host	Specific value	Remarks	Last update date/time	Last updated by
							example_name			
History	Update	Discard	226:LegacyRole_Practice 23:RegisteSudoer 1:Host				example_name		2021/08/20 16:07:47	LegacyRole substitution value auto-registration :
History	Update	Discard	326:LegacyRole_Practice 23:RegisteSudoer 1:Host				example_sudoers		2021/08/20 16:08:20	LegacyRole substitution value auto-registration :

# 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

The screenshot shows the Ansible-LegacyRole Execution menu. The left sidebar lists various options like Main menu, Movement list, and Execution. The main area has two tabs: 'Movement [List]' (selected) and 'Operation [List]'. A red box labeled '1' points to the 'Movement [List]' tab with the instruction 'Select the Movement to execute.' A red box labeled '2' points to the 'Operation [List]' tab with the instruction 'Select Operation.' A red box labeled '3' points to the 'Execute' button at the bottom with the instruction 'Click "Execute".' A red callout labeled 'Tips' contains the note: 'The screen will automatically change to the "Check work status" screen after executing.'

1 Select the Movement to execute.

2 Select Operation.

3 Click "Execute".

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**

**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	<table border="1"><tr><td>ID</td><td>1</td></tr><tr><td>Name</td><td>Legacy1</td></tr><tr><td>Delay timer (minutes)</td><td></td></tr><tr><td>Dedicated information for ansible</td><td><table border="1"><tr><td>Host specific format</td><td>IP</td></tr><tr><td>WinRM connection</td><td></td></tr></table></td></tr></table>	ID	1	Name	Legacy1	Delay timer (minutes)		Dedicated information for ansible	<table border="1"><tr><td>Host specific format</td><td>IP</td></tr><tr><td>WinRM connection</td><td></td></tr></table>	Host specific format	IP	WinRM connection	
ID	1												
Name	Legacy1												
Delay timer (minutes)													
Dedicated information for ansible	<table border="1"><tr><td>Host specific format</td><td>IP</td></tr><tr><td>WinRM connection</td><td></td></tr></table>	Host specific format	IP	WinRM connection									
Host specific format	IP												
WinRM connection													
Operation	<table border="1"><tr><td>No.</td><td>1</td></tr><tr><td>Name</td><td>Operation1</td></tr><tr><td>ID</td><td>1</td></tr></table>	No.	1	Name	Operation1	ID	1						
No.	1												
Name	Operation1												
ID	1												
Host management	<table border="1"><tr><td>confirmation</td></tr></table>	confirmation											
confirmation													
Substitution value	<table border="1"><tr><td>confirmation</td></tr></table>	confirmation											
confirmation													
Input data	Populated data	<a href="#">InputData_000000051.zip</a>											
Output data	Result data	<a href="#">ResultData_000000051.zip</a>											
Operation status	Scheduled date/time												
	Start date/time	2020/11/11 08:44:45											
	End date/time	2020/11/11 08:44:58											

**Progress status(Execution log)**

Filter:	<input type="checkbox"/> Display only corresponding lines
task path: /exastro/data_relay_storage/ansible_driver/legacy/r1/0000000002/in/roles/sudoers/tasks/config.yml:12	
skipping: [testserver] => {	
"changed": false,	
"skip_reason": "Conditional result was False"	
}	
TASK [Remove unmanaged /etc/sudoers.d files] ****	
task path: /exastro/data_relay_storage/ansible_driver/legacy/r1/0000000002/in/roles/sudoers/tasks/config.yml:20	
skipping: [testserver] => {	
"changed": false,	
"skip_reason": "Conditional result was False"	
}	
META: ran handlers	
META: ran handlers	
PLAY RECAP	
testserver	: ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

**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

Exastro

44

# Chapter 3 Ansible-Pioneer



# 3.1 Work environment and Scenario

## Work environment

The work environment used in this document is as follows.

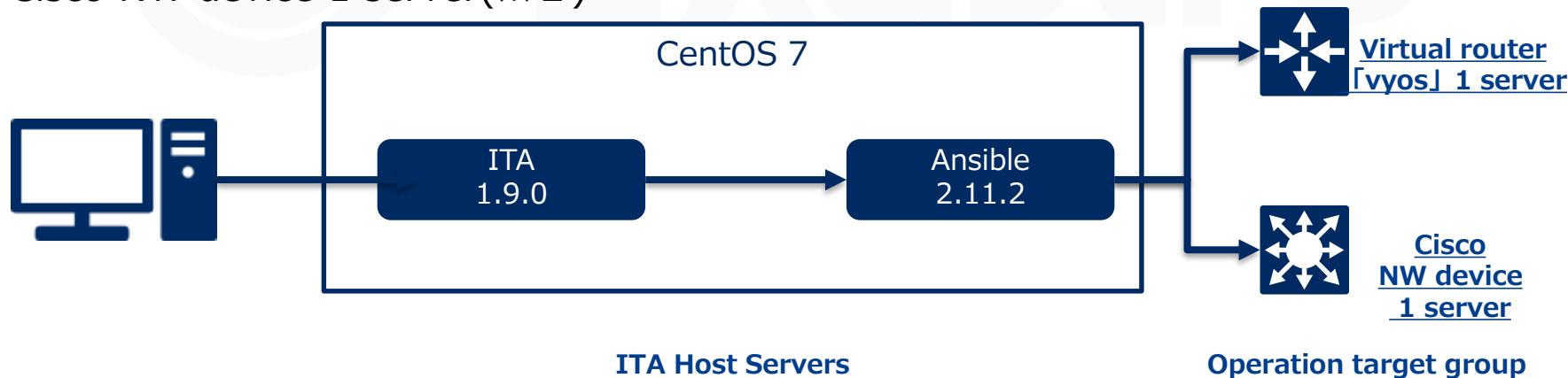
We will operate on a **NW Device**, so please prepare the following environment.

### ITA Host Servers

- CentOS 7 (※1)
- ITA 1.9.0
- Ansible 2.11.2

### 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

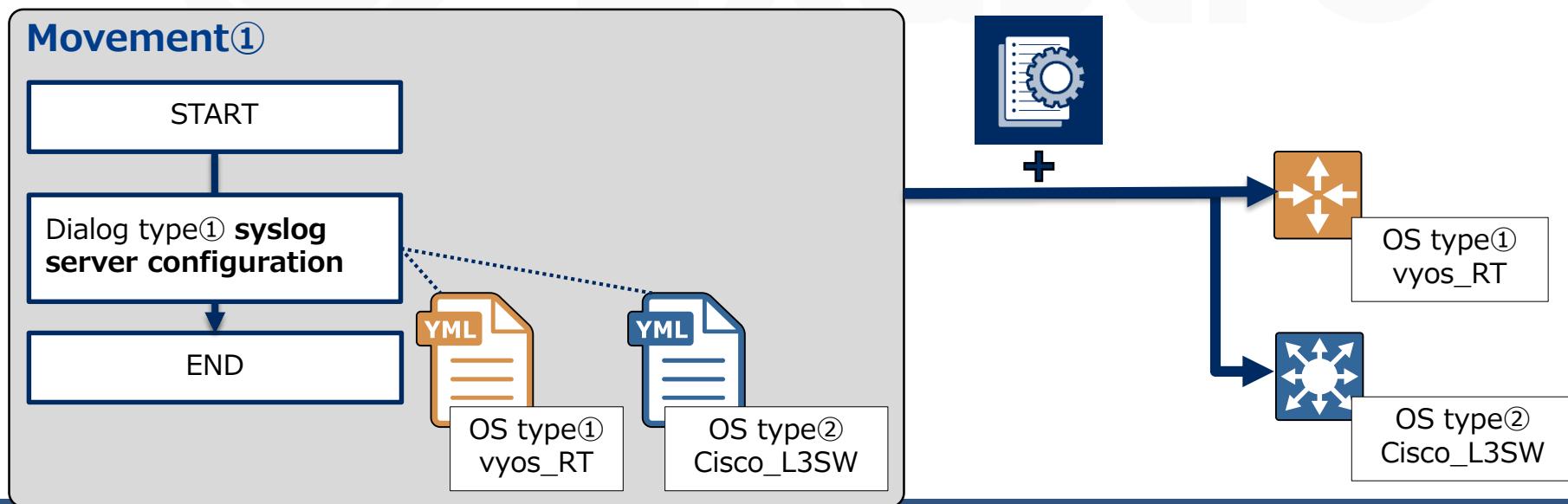
## Scenario

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 conditional branching using proprietary modules.

## Scenario images

### Movement creation



## 3.2 Dialog file creation(1/2)

### Dialog file creation

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

[Attention]

Please use "UTF-8" for the character code.

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".

```
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 }})  
  
File name: vyos_set_syslog_server.yml
```

## 3.2 Dialog file creation(2/2)

### Prepare the dialog file

Similarly, create file on the right.

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  
  
  File name: ios_set_syslog_server.yml  
  
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
```

### 3.3 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".

Register

os type ID	os type name*	Device type
Auto-input		SV NW ST

\*is a required item.

Back      Register

2

OS type name	Device type/NW
vyos_RT	●
Cisco_L3SW	●

## 3.4 Movement configuration(1/4)

### 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".

Register

Movement ID	Movement Name*	Delay timer	Dedicated information for ansible		Access permission	
			Host specific format*	Number of parallel executions	Setting	Role to allow access
Auto-input	<input type="text"/>		<input type="text"/>	<input type="text"/>	Setting	

\* is a required item.

2

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

Back      Register

### 3.4 Movement configuration(2/4)

#### 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".

Register

Item No.	Dialog type name*	Access permission	
		Setting	Role to allow access
Auto-input	<input type="text"/>	Setting	

\* is a required item.

2 Back Register

Item name	Input contents
Dialog type name	syslog server specification

## 3.4 Movement configuration(3/4)

### 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".

The screenshot shows a user interface for registering a Dialog file. At the top, there's a blue header bar with the word 'Register'. Below it is a form with several fields:

- Dialog ID:** Auto-input (with a dropdown arrow)
- Dialog type\***: A dropdown menu with an arrow icon.
- OS type\***: A dropdown menu with an arrow icon.
- Dialog file\***: A file input field with a placeholder 'Choose file' and a status message 'No file chosen'. Below it is an orange button labeled 'Upload in advance'.

A red box highlights the 'Dialog type\*', 'OS type\*', and 'Dialog file\*' fields, and a red arrow points from the number '2' to the bottom table, which also has a red border around its header row.

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".

The screenshot shows a registration interface for linking a movement to a dialog type. The top section has a header 'Register' and a table with three required fields: 'Associated item No.' (set to 'Auto-input'), 'Movement\*', and 'Dialog type\*'. The 'Include order\*' field is also present. A note at the bottom says '※\*is a required item.' Below the table are 'Back' and 'Register' buttons. To the right, a red callout labeled '2' points to a table of input contents:

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

# 3.5 Operation registration

## 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".

No.	Operation ID	Operation name*	Scheduled date for execution*	Access permission	
				Setting	Role to allow access
Auto-input	Auto-input			Setting	

\* is a required item.

Back      Register

2

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 the device list

Register this work target 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".

Register

Managed system item number	HW device type	Host name*	IP address*	EtherWakeOnLan		Login user ID	Login password		Ansible Dedicated information			Pioneer Dedicated information		
				MAC address	Network device name		Management	Login password	Authentication method	WinRM connection information	Port no	Server certificate	Protocol	OS type
Auto-input	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>								

\* is a required item.

2

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

Back Register

## 3.6 Register to device list (2/2)

### Register NW device to the device list

Register this work target from the device list.

Menu: **Basic Console > Device list**

- ① Click Register > Start Registration.
- ② Select the following information for each item and click "Register".

Register

Managed system item number	HW device type	Host name*	IP address*	EtherWakeOnLan		Login password		Ansible Dedicated information			Pioneer Dedicated information	
				MAC address	Network device name	Login user ID	Management	Login password	Authentication method	WinRM connection information	Protocol	OS type
Auto-input	▼								▼ Search from pulldown	▼ Search from pulldown	▼ Search from pulldown	▼ Search from pulldown

※\* is a required item.

Back Register

2

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

### 3.7 Parameter sheet creation (1/2)

#### 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.(Next item)

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

The screenshot shows the 'Create menu' configuration interface. On the left, there is a form with various fields. The 'Menu name' field is set to 'Pioneer practice', 'Creation target' is 'Parameter Sheet(Host/Operal)', and 'Display order' is '3'. These three fields are highlighted with a red box. A red circle with the number '1' is positioned next to the input table on the right. The input table lists the 'Item' and 'Input contents' for each configuration item.

## 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

The screenshot shows the 'Create/Define menu' window with four parameter items listed:

- syslog\_server\_ip: String, Maximum number of bytes: 32, Regular expression: [0-9.]{1,3}
- sub\_syslog\_server\_ip: String, Maximum number of bytes: 32, Regular expression: [0-9.]{1,3}
- log\_facility: String, Maximum number of bytes: 32, Regular expression: [0-9.]{1,3}
- log\_severity: String, Maximum number of bytes: 32, Regular expression: [0-9.]{1,3}

A red box highlights the 'Item' button in the toolbar, with a callout '1 Press "item" to add a new item.' A red circle with '2' points to a table on the right containing the following data:

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

A red box highlights the 'Create' button at the bottom left, with a callout '3 Press the "create" button to create the menu.'

## 3.8 Data registration

### 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".

2

No	Host name*	Operation	Parameter				Access permission
		Operation*	syslog_server_ip	sub_syslog_server_ip	log_facility	log_severity	Setting Role to allow access
Auto-input							Setting

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

Register

Parameter sheet

Item No.	Menu group: Menu	Item	Registration method*	Movement	Iac variable	Value variable
					Key variable	Value variable
					Variable name	Substitution order
Auto-input	Select menu	Select menu	Select	Select	Select Movement	Select Movement

\* is a required item.

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".

History	Update	Discard	Item No. ▾	Operation ▾	Movement ▾	Host ▾	Last update date/time ▾	Last updated by ▾
History	Update	Discard	127:Pioneer_Practice 24:Log server configuration	10:NWHost	2021/08/20 16:50:02	Pioneer substitution value auto-registration set		
History	Update	Discard	227:Pioneer_Practice 24:Log server configuration	11:NWHost2	2021/08/20 16:50:02	Pioneer substitution value auto-registration set		

Target host

History	Update	Discard	Item No. ▾	Operation ▾	Movement ▾	Host ▾	Variable name ▾	Last update date/time ▾	Last updated by ▾
History	Update	Discard	127: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	227: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	327: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	427: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	527: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	627: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	727: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	827:Pioneer_Practice 24:Log server configuration	11:NWHost2	1:VAR_log_severity	2021/08/20 16:52:15	Pioneer substitution value auto-registration set		

Substitution  
value list

# 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

The screenshot shows the Ansible-Pioneer software interface with the following details:

- Left Sidebar (Menu):** Includes options like Main menu, Movement list, Dialog type list, OS type master, Dialog files, Movement details, Substitution value auto-registration setting, Target host, Substitution value list, and **Execution**.
- Movement Execution Steps:**
  - Select the Movement to execute.** (Red callout 1) - A red circle highlights the "Movement [List]" section of the main pane, which contains a table with one row: Movement ID 14, Movement Name Log server configuration, Ansible Pioneer, IP, Variable count 1, Last update date/time 2020/12/21 14:38:07, and Last updated by System Administrator.
  - Select Operation.** (Red callout 2) - A red circle highlights the "Operation [List]" section of the main pane, which contains a table with multiple rows. Row 11 is highlighted with a blue selection circle.
  - Click "Execute".** (Red callout 3) - A red circle highlights the "Execute" button at the bottom of the screen.
- Tips:** A red callout with the text "The screen will automatically change to the "Check operation status" screen after executing." is located on the right side.

# 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	<table border="1"><tr><td>ID</td><td>1</td></tr><tr><td>Name</td><td>Legacy1</td></tr><tr><td>Delay timer (minutes)</td><td></td></tr><tr><td>Dedicated information for ansible</td><td><table border="1"><tr><td>Host specific format</td><td>IP</td></tr><tr><td>WinRM connection</td><td></td></tr></table></td></tr></table>	ID	1	Name	Legacy1	Delay timer (minutes)		Dedicated information for ansible	<table border="1"><tr><td>Host specific format</td><td>IP</td></tr><tr><td>WinRM connection</td><td></td></tr></table>	Host specific format	IP	WinRM connection	
ID	1												
Name	Legacy1												
Delay timer (minutes)													
Dedicated information for ansible	<table border="1"><tr><td>Host specific format</td><td>IP</td></tr><tr><td>WinRM connection</td><td></td></tr></table>	Host specific format	IP	WinRM connection									
Host specific format	IP												
WinRM connection													
Operation	<table border="1"><tr><td>No.</td><td>1</td></tr><tr><td>Name</td><td>Operation1</td></tr><tr><td>ID</td><td>1</td></tr></table>	No.	1	Name	Operation1	ID	1						
No.	1												
Name	Operation1												
ID	1												
Host management													
Substitution value													
Input data	Populated data												
	<a href="#">InputData_0000000051.zip</a>												
Output data	Result data												
	<a href="#">ResultData_0000000051.zip</a>												
Operation status	<table border="1"><tr><td>Scheduled date/time</td><td></td></tr><tr><td>Start date/time</td><td>2020/11/11 08:44:45</td></tr><tr><td>End date/time</td><td>2020/11/11 08:44:58</td></tr></table>	Scheduled date/time		Start date/time	2020/11/11 08:44:45	End date/time	2020/11/11 08:44:58						
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)

```
Filter :  □, Display only corresponding lines

"exec_file": "/exastro/data_relay_storage/ansible_driver/pioneer/ns/00000000103/in/dialog_files/vyos-test/00000000001-7112.v",
"extra_args": " _undefinedsymbol_",
"exec_shell_dir": "/library",
"host_vars_file": "/exastro/data_relay_storage/ansible_driver/pioneer/ns/00000000103/tmp/original_host_vars/vyos-test",
"inventory_hostname": " _undefinedsymbol_",
"log_file_dir": "/exastro/data_relay_storage/ansible_driver/pioneer/ns/00000000103/out",
"protocol": "ssh",
"ssh_key_file": " _undefinedsymbol_",
"username": "vyos"
}
},
"msg": "normal exit"
}
META: ran handlers
META: ran handlers
PLAY RECAP ****
: ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
: ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
4
```

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"

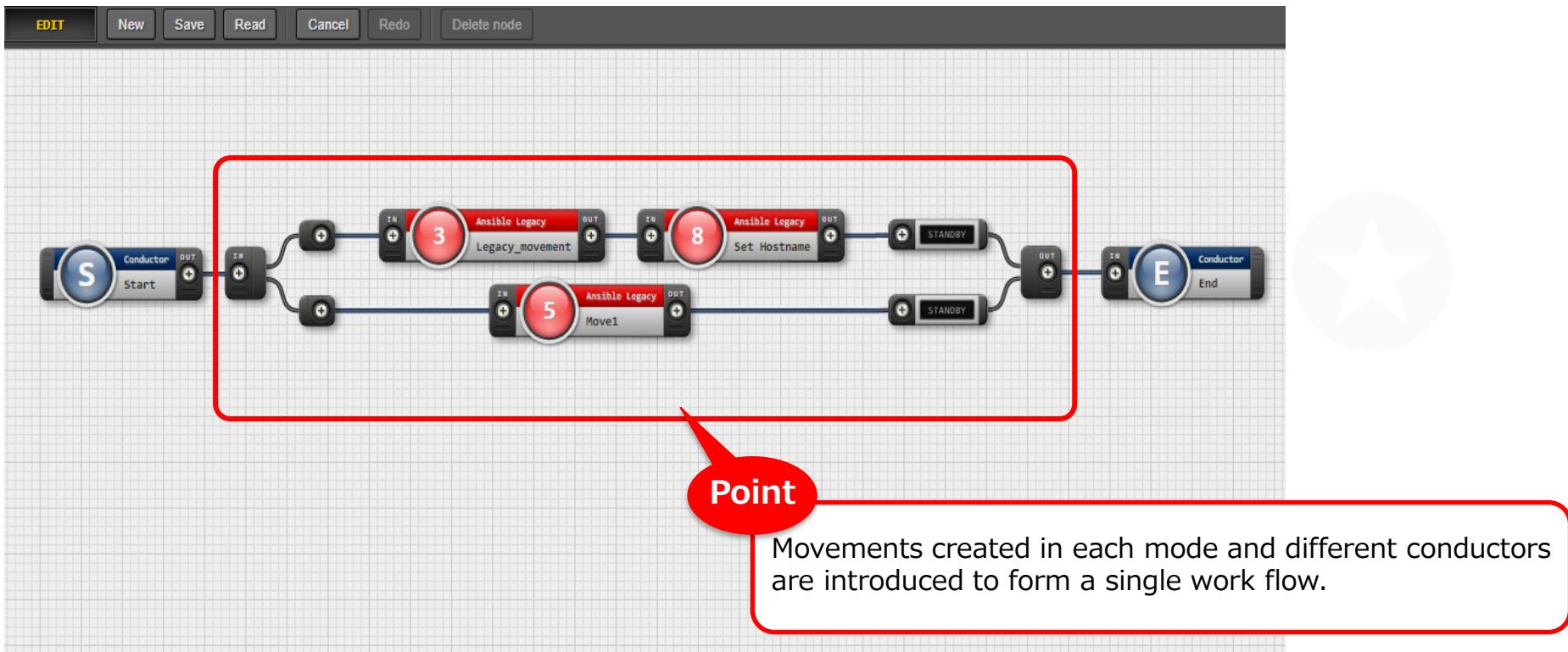
# A) Appendix



# 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**