



# 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

## ■ 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](#)

### 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](#)

### A) Appendix

# 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)

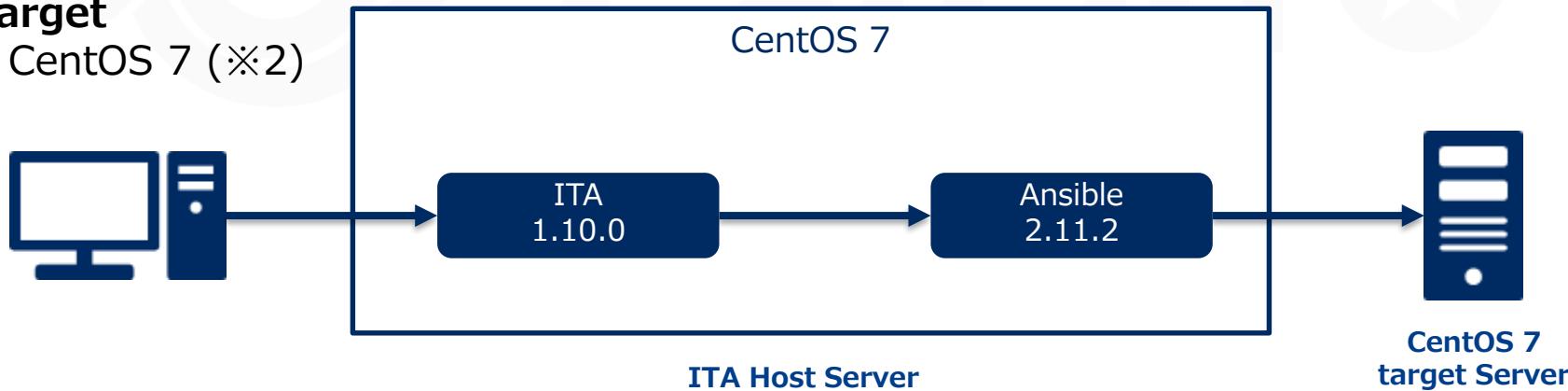
Please prepare a server in addition to the ITA Host server. The additional server will be target for operation.

### ITA Host Server

- CentOS 7 (※1)
- ITA 1.10.0
- 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

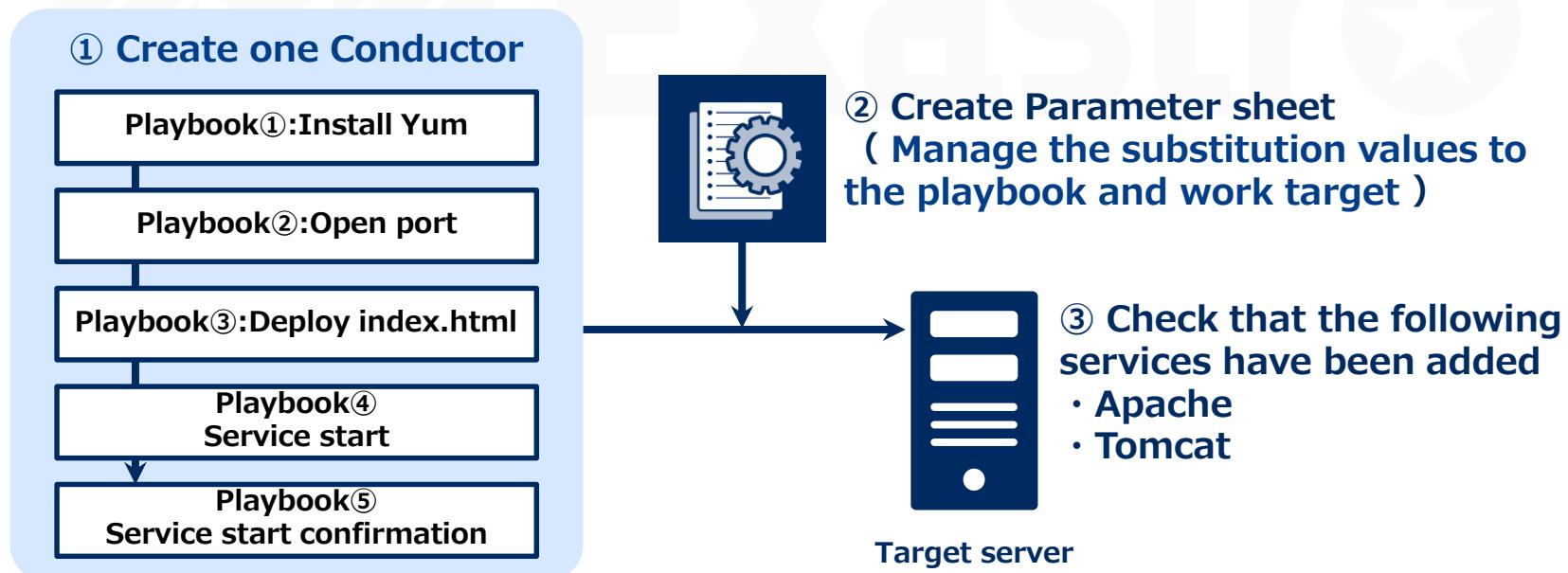
## Scenario

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".



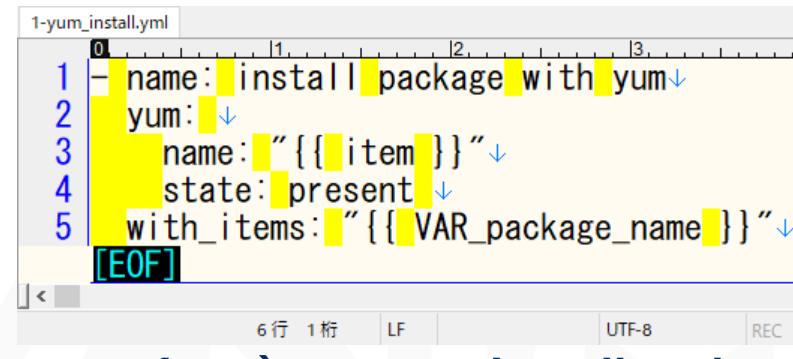
# 1.2 Creating files (1/4)

## 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



```
1-yum_install.yml
1 - name: install package with yum
2   yum:
3     name: "{{ item }}"
4     state: present
5   with_items: "{{ VAR_package_name }}"
[EOF]
```

6 行 1 桁 LF UTF-8 REC

(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.

# 1.2 Creating files(2/4)

## 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"'
```

```
- 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 deploys file

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

Starts the specified service.

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

Checks if the service has started.

## 1.2 Creating files(4/4)

### 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.

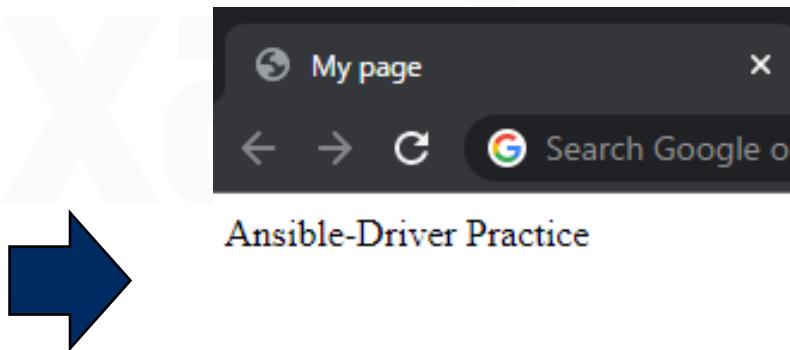
**File name: index.html**

```
<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.

# 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".

Register

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

\* is a required item.

[Back](#) [Register](#)

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".

Register

Playbook ID	Playbook name*	Playbook files*	Access permission	
			Setting	Role to allow access
Auto-input	<input type="text"/>	<div><button>Choose file</button> No file chosen <button>Upload in advance</button> Upload status:</div>	Setting	

\* is a required item.

[Back](#) [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

# 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".

Register

File ID	File embedded variable name*	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:	Setting
※* is a required item.			
<input type="button" value="Back"/>		<input type="button" value="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".

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.

Back      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

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

Point

# 1.4 Conductor creation

## Create Conductor

Create a Conductor that collects the defined Movements.

Menu: Conductor > Conductor class edit

The screenshot shows the 'Conductor name' section with the 'Name' field set to 'Add service'. A callout '1' points to this field with the text 'Input Conductor name.' Below the conductor name, there is a diagram of a workflow with nodes S, 18, 19, 20, and E. Nodes 18, 19, and 20 are highlighted with a red box and labeled 'Ansible Legacy'. A callout '3' points to this diagram with the text 'Connect nodes to each others.' To the right of the diagram is a table titled 'Movement' with rows for 'Test movement', 'Test1', 'PackageInstall', 'Install\_Packages', 'Open\_Ports', and 'Start\_Service'. A callout '2' points to this table with the text 'Add the Movements by dragging and dropping.' At the bottom left, a red box highlights the 'Registration' button, with a callout '4' pointing to it with the text 'Click "Registration".'

1 Input Conductor name.  
2 Add the Movements by dragging and dropping.  
3 Connect nodes to each others.  
4 Click "Registration".

**Conductor name**  
Add service

**Movement**

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

**Registration**

# 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".

Register

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"/>	<button>Setting</button>	<input type="text"/>

\* is a required item.

[Back](#) [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".

The screenshot shows the 'Register' page in the Basic Console. The page has a header 'Register' and a main form area. The form includes fields for Managed system item number, HW device type, Host name\*, IP address\*, EtherWakeOnLan, Login user ID, Management, Login password, ssh authentication key file, and Authentication method. The first three rows (Managed system item number, HW device type, Host name\*) are highlighted with a red box. The Login user ID, Management, and Login password rows are also highlighted with a red box. The ssh authentication key file and Authentication method rows are highlighted with a red box.

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

Id : Auto-input	
Menu name* : Legacy practice	
Creation target : Parameter Sheet(Host/Operal ▾)	
Display order* : 1	
Create as hostgroup menu : <input type="checkbox"/> Yes	
Create as vertical menu <span style="color: yellow;">?</span> : <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	

Input the following information.(Next item)

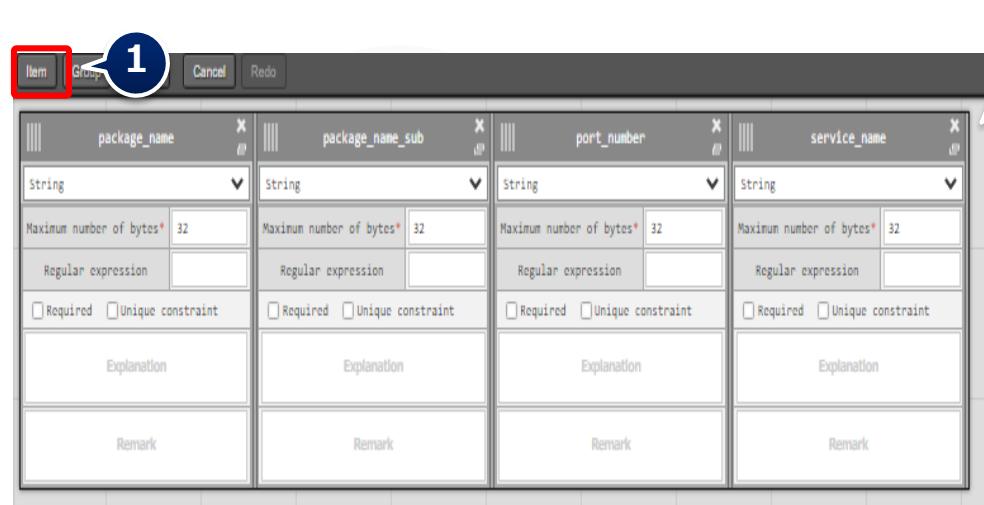
Item name	Input contents
Menu name	Legacy practice
Target	Parameter sheet (Host/Operation)
Display order	1

# 1.7 Parameter sheet creation (2/2)

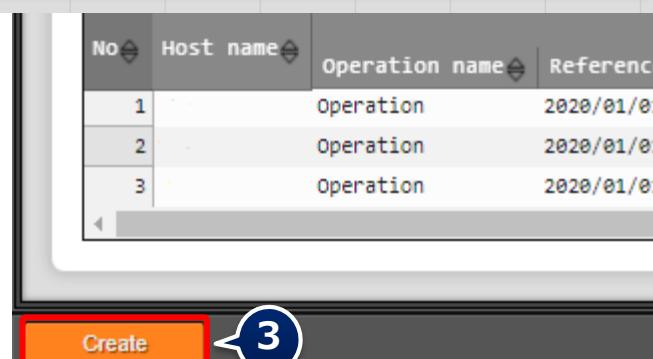
## Define the item name of the parameters sheet

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



# 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 blue header bar. Below it, a table with several columns. The first two columns are 'No.' and 'Host name\*', both marked with a red asterisk indicating they are required fields. These two columns are highlighted with a large red rectangular box. The other columns in the row are 'Operation\*', 'Parameter', 'package\_name', 'package\_name\_sub', 'port\_number', and 'service\_name'. The 'Parameter' column contains sub-fields: 'package\_name', 'package\_name\_sub', 'port\_number', and 'service\_name'. The 'Host name\*' field has a dropdown arrow icon. The 'Operation\*' field also has a dropdown arrow icon. The 'Parameter' section has four separate input fields.

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".

The screenshot shows the 'Register' screen with the following fields highlighted by a red box:

- Parameter sheet(From):** Contains 'Item No.', 'Menu group:Menu\*', and 'Item\*' dropdowns.
- IaC variable(To):** Contains 'Registration method\*', 'Movement\*', and two tables:
  - Key variable:** Columns: Variable name, Substitution order.
  - Value variable:** Columns: Variable name, Substitution order.

Below these sections are dropdowns for 'Select menu', 'Select Movement', and '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

List/Update										
History	Update	Discard	Item No. ↴	Operation ↴	Movement ↴	Host ↴	Substitution value management	Access permission	Remarks ↴	Last update date/time ↴
							Role to allow access ↴	Last updated by ↴		
History	Update	Discard	19:23:Install Apache 18:Install_Packages	1:Host	Substitution value management			2021/08/20 15:47:46	Legacy substitution value auto-registration sett:	
History	Update	Discard	20:24:Install Tomcat 18:Install_Packages	1:Host	Substitution value management			2021/08/20 15:47:46	Legacy substitution value auto-registration sett:	
History	Update	Discard	21:23:Install Apache 19:Open_Ports	1:Host	Substitution value management			2021/08/20 15:48:42	Legacy substitution value auto-registration sett:	
History	Update	Discard	22:24:Install Tomcat 19:Open_Ports	1:Host	Substitution value management			2021/08/20 15:48:42	Legacy substitution value auto-registration sett:	
History	Update	Discard	23:23:Install Apache 20:Start_Service	1:Host	Substitution value management			2021/08/20 15:49:04	Legacy substitution value auto-registration sett:	
History	Update	Discard	24:24:Install Tomcat 20:Start_Service	1:Host	Substitution value management			2021/08/20 15:49:04	Legacy substitution value auto-registration sett:	

Substitution value list

List/Update										
History	Update	Discard	Item No. ↴	Operation ↴	Movement ↴	Host ↴	Variable name ↴	Sensitive setting ↴	Specific value ↴	Substitution order ↴
History	Update	Discard	22:23:Install Apache 18:Install_Packages	1:Host	7:VAR_package_name OFF	httpd			1	2021/08/20 15:47:46
History	Update	Discard	23:24:Install Tomcat 18:Install_Packages	1:Host	7:VAR_package_name OFF	tomcat			1	2021/08/20 15:47:46
History	Update	Discard	24:24:Install Tomcat 18:Install_Packages	1:Host	7:VAR_package_name OFF	tomcat-webapps			2	2021/08/20 15:48:08
History	Update	Discard	25:23:Install Apache 19:Open_Ports	1:Host	8:VAR_port_number OFF	80/tcp				2021/08/20 15:48:42
History	Update	Discard	26:24:Install Tomcat 19:Open_Ports	1:Host	8:VAR_port_number OFF	8080/tcp				2021/08/20 15:48:42
History	Update	Discard	27:23:Install Apache 20:Start_Service	1:Host	9:VAR_service_name OFF	httpd				2021/08/20 15:49:04
History	Update	Discard	28:24:Install Tomcat 20:Start_Service	1:Host	9:VAR_service_name OFF	tomcat				2021/08/20 15:49:04

# 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' interface with a sidebar menu on the left. The main area has two tabs: 'Conductor [filter]' and 'Conductor [List]'. A callout '1' points to the 'Conductor [filter]' tab with the text 'Select the Conductor to execute.' A red box highlights the first row of the table in the 'Conductor [filter]' tab, which contains entries for 'Server basic setting' and 'Kato TEST1104'. The 'Conductor [List]' tab is shown below it with a filter result count of 4.

The second tab, 'Operation [Filter]', is selected. A callout '2' points to it with the text 'Select Operation.'. A red box highlights the first row of the table in the 'Operation [Filter]' tab, which contains entries for '1 Operation1' and '2 Test Operation'. The 'Operation [List]' tab is shown below it.

A callout '3' points to the bottom of the screen with the text 'Click "Execution" at the bottom of the screen.' A red box highlights the 'Execution' button in the footer bar of the interface.

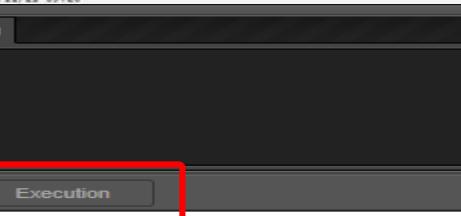
**Tips**: The screen will automatically change to the "Conductor Confirmation" screen after executing.

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

Select	No.	Operation ID	Operation name	Scheduled date for execution	Last execution date	Remarks	Last update date/time	Last updated by
<input type="radio"/>	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="radio"/>	2	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="radio"/>	6	6	6 Basic settings all	2020/10/24 09:54		Test for Host group menu creation	2020/10/22 09:54:53	Data portability procedure
<input type="radio"/>	7	7	open	2020/11/10 14:00			2020/11/10 14:00:49	Data portability procedure
<input type="radio"/>	8	8	OPI	2020/11/21 09:20			2020/11/19 09:03:14	System Administrator
<input type="radio"/>	9	9	operation					

Click "Execution" at the bottom of the screen.



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

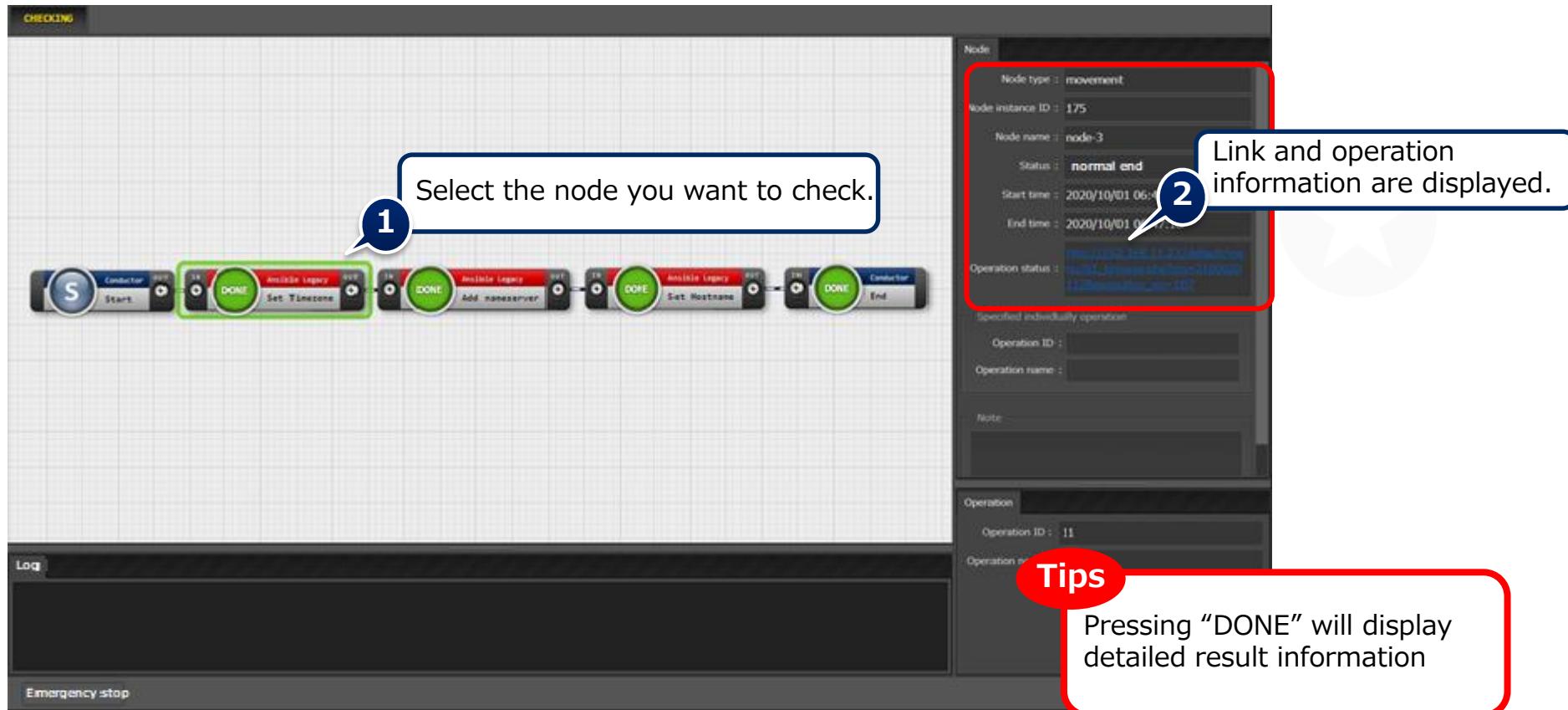
# 1.11 Execution (2/3)

## 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 contains various configuration parameters:

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></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												

**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 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)

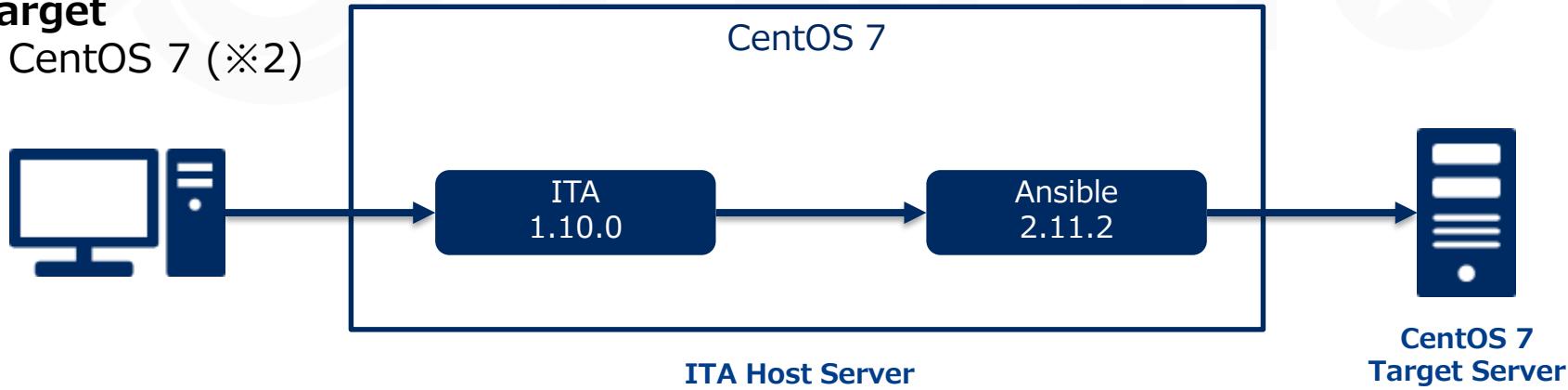
Please prepare a server in addition to the ITA Host server. The additional server will be target for operation.

### 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.

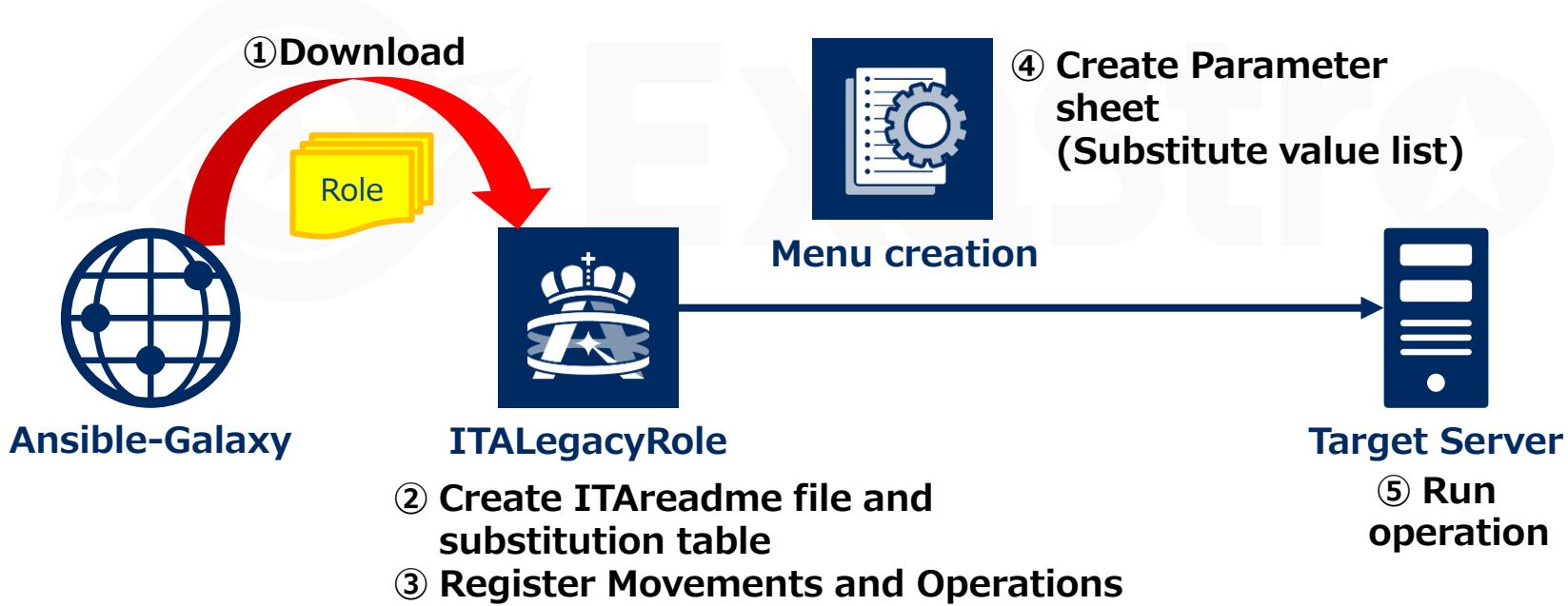
※2 Any OS compatible with Ansible can be used.

## 2.1 Environment and Scenario(2/3)

### 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”

The screenshot shows the Galaxy Ansible Role page for the 'sudo' role by 'weareinteractive'. The page includes a sidebar with links for Home, Search, and Community. The main content area displays the role details: name 'sudo', description 'Installs and configures sudo', version '0.1.0', and maintainer 'weareinteractive'. It shows a rating of 4.9 / 5 Score, 167011 Downloads, and a GitHub Repo link. A red box labeled '1' highlights the 'GitHub Repo' link. Another red box labeled '2' highlights the 'Code' button in the top right corner. A third red box labeled '3' highlights the 'Download ZIP' button at the bottom right.

## 2.2 Role package preparation(1/5)

### 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

2

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.

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.

## 2.2 Role package preparation(2/5)

### 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:
```

#### Tips

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

#### 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(3/5)

### 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:
```

```
substitution table  
LCA_sudo_users: sudo_users  
LCA_sudo_sudoers_file: sudo_sudoers_file
```

#### Actual variables used

```
sudo_users:  
- name: example_name  
sudo_sudoers_file: example_sudoers
```



Substitution value  
management

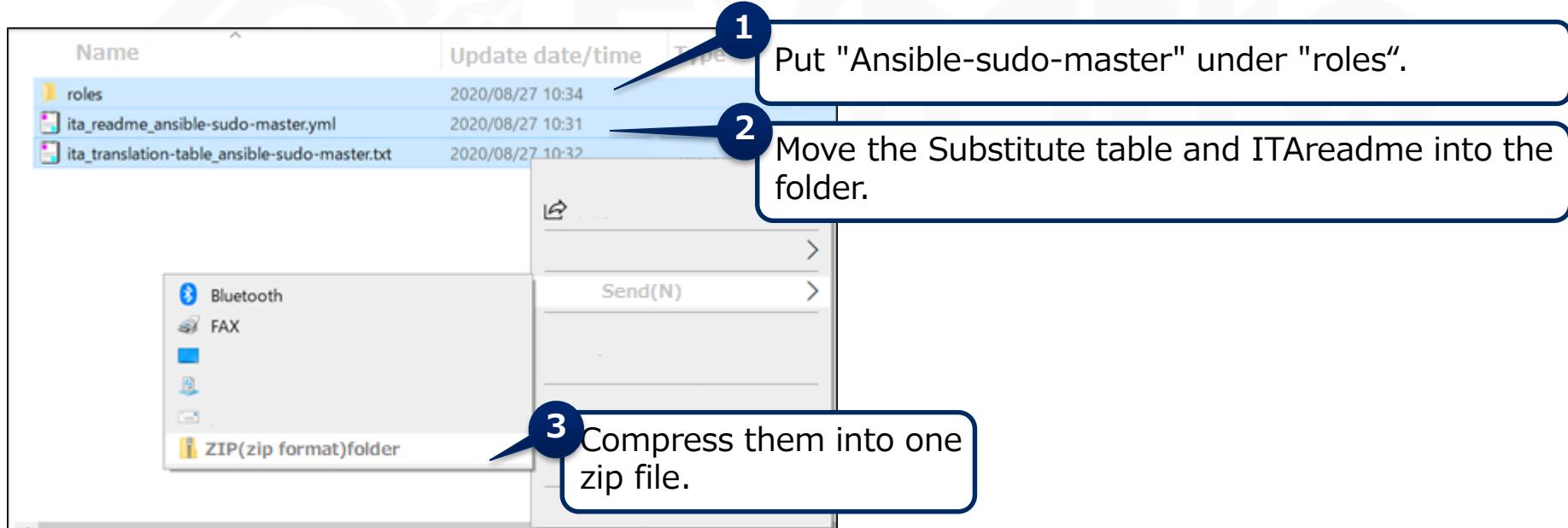
Variable name	Member variable name (Added ITAreadme)	Specific value
LCA_sudo_users	[0].name	example_name
LCA_sudo_sudoers_file		example_sudoers

## 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

```
ita_readme_ansible-sudo-master.yml  
1 sudo_users:  
2 - name: [EOF]
```

(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.

## 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".

Register

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

\* is a required item.

[Back](#) [Register](#)

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".

Register

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

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.

Back

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.

## 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".

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

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".

The screenshot shows the 'Register' page in the Basic Console. The registration form is highlighted with a red box. A red arrow points from the 'Management' field in the form to a table below, which lists the input contents for each registered item.

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

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 <span style="color: yellow;">?</span> : <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	

1

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

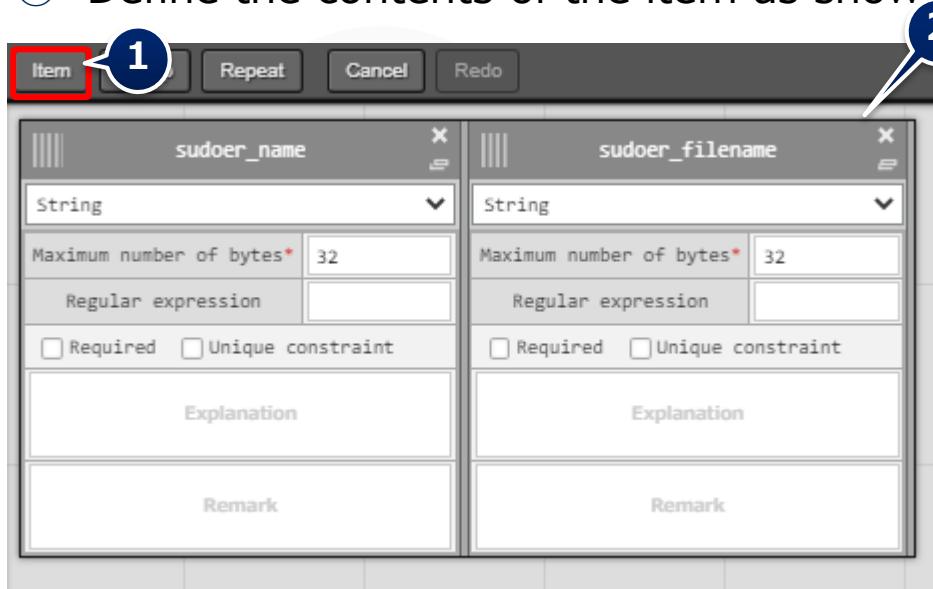
## 2.6 Parameter sheet creation(2/2)

### 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

The screenshot shows a parameter sheet with a table:

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

A callout bubble points to the 'Create' button at the bottom left, with the text "Press ‘Create’ to create menu."

## 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".

No	Host name*	Operation	Parameter	
		Operation*	sudoer_name	sudoer_filename
Auto-input	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

### 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".

Register

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

\* is a required item.

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

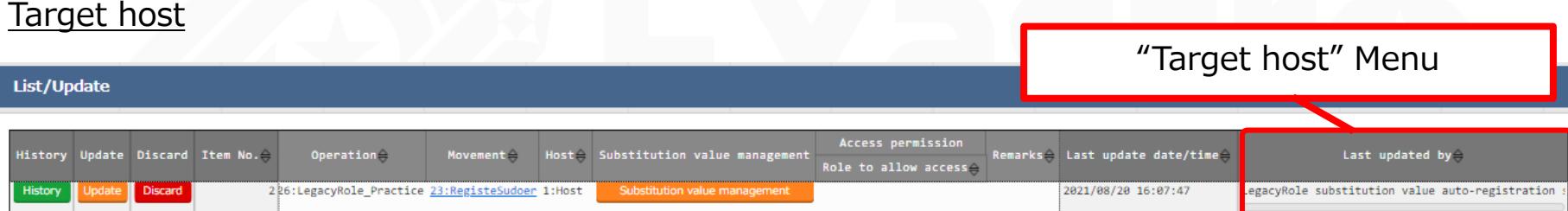
### Check Substitution value and Target host

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 ▾	Role to allow access ▾		
History	Update	Discard		2 26: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		2 26:LegacyRole_Practice 23:RegisteSudoer 1:Host			example_name		2021/08/20 16:07:47	LegacyRole substitution value auto-registration :
History	Update	Discard		3 26: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 three tabs: Movement [Filter] (selected), Movement [List], and Operation [List].

- Select the Movement to execute.** (Step 1)  
The Movement [Filter] tab shows a table with one row selected. A callout points to the first column of the selected row.

Select	Movement ID	Movement Name	Orchestrator	Delay timer	Host specific format	WinRM connection	Header section	Optional parameters	Variable count	Remarks	Last update date/time	Last updated by
<input checked="" type="radio"/>	4movement01	Ansible Legacy Role	IP						3	7	2020/10/14 13:23:24	Data portability procedure
- Select Operation.** (Step 2)  
The Operation [List] tab shows a table with multiple rows. A callout points to the first column of the first row.

Select	No.	Operation ID	Operation name	Scheduled date for execution	Last execution date	Remarks	Last update date/time	Last updated by
<input checked="" type="radio"/>	1	1	1 Operation1	2020/08/27 16:15	2020/12/21 10:43		2020/12/21 10:43:11	LegacyRole execution procedure
<input type="radio"/>	2	2	2 Test Operation	2020/10/08 18:00	2020/10/23 16:21		2020/10/23 16:21:05	Legacy execution procedure
<input type="radio"/>	6	6	6 Basic settings all	2020/10/24 09:54		Test for Host group menu creation	2020/10/22 09:54:53	Data portability procedure
<input type="radio"/>	7	7	7 opel	2020/11/10 14:00			2020/11/10 14:00:49	Data portability procedure
<input type="radio"/>	8	8	8 OP1	2020/11/21 09:20			2020/11/19 09:03:14	System Administrator
<input type="radio"/>	9	9	9 operation	2020/12/18 16:03	2020/12/03 15:58		2020/12/03 15:58:03	Legacy execution procedure
- Click "Execute".** (Step 3)  
At the bottom, there are two buttons: "Dry run" and "Execute". The "Execute" button is highlighted with a red box.

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**

The screenshot shows the Ansible-LegacyRole interface with the following sections:

- Target Operation**: A table showing execution details like execution number, type, status, engine, and user.
- Status**: A table showing movement details like ID, name, and delay timer.
- Progress status(Execution log)**: A log window displaying the execution log with tasks, skipping, and play recap.
- Log**: A button to view the log.
- Input data**: A table showing populated input data as `InputData_000000051.zip`.
- Output data**: A table showing result data as `ResultData_000000051.zip`.
- Operation status**: A table showing start and end dates/times.

**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



# 3.1 Work environment and Scenario

## Work environment

The work environment used in this document is as follows.

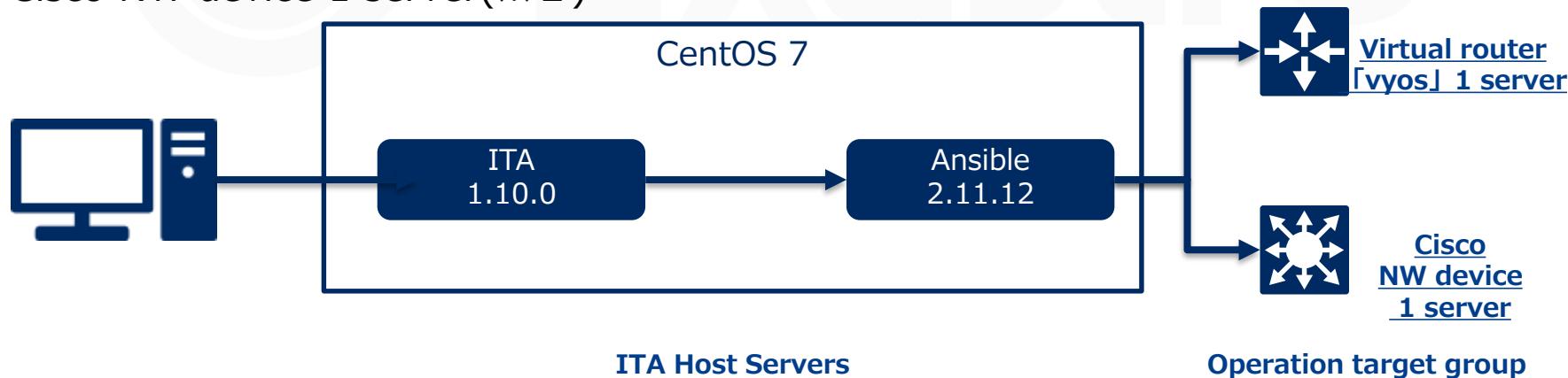
We will operate on a **NW Device**, so please prepare the following environment.

### ITA Host Servers

- CentOS 7 (※1)
- ITA 1.10.0
- 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

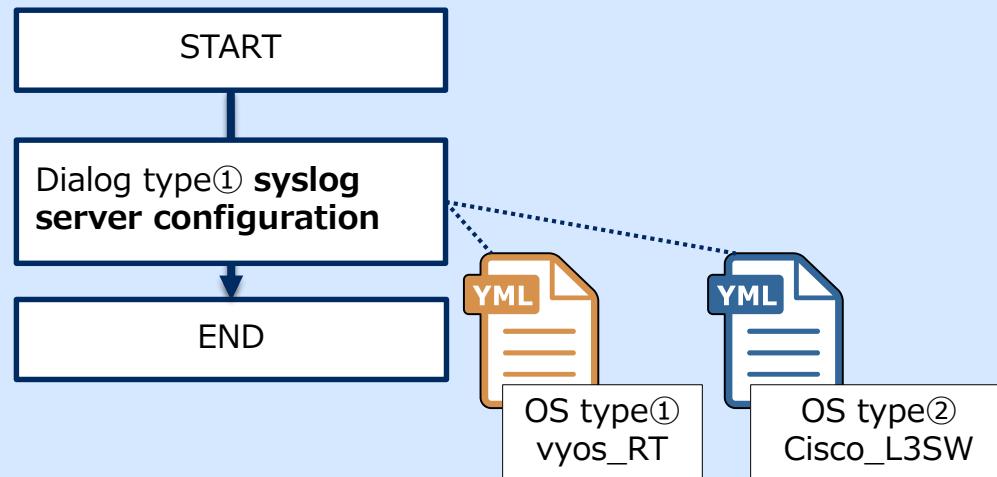
Ansible-Pioneer is used to **specify log servers for NW device with different vendors**. This scenario lets you experience the following 3 features of the Pioneer mode.

- 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 conditional branching using proprietary modules.

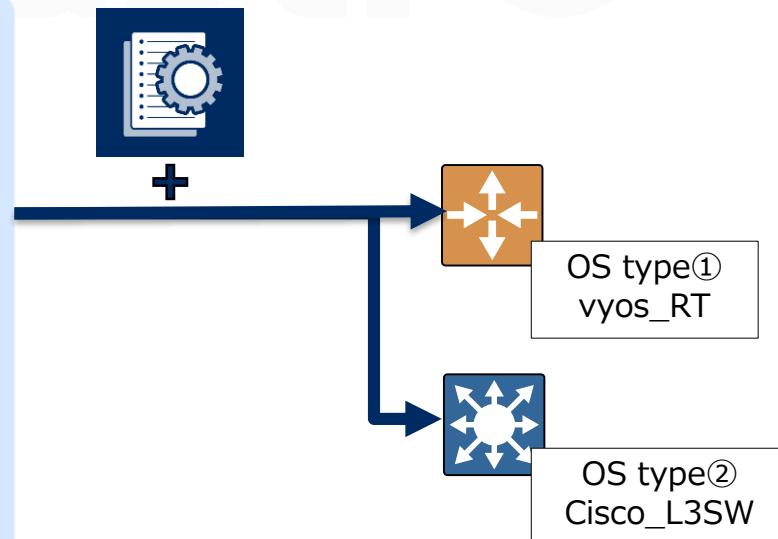
## Scenario Diagram

### Movement creation

#### Movement①



### Setting the substitution value and target host using the parameter sheet



# 3.2 Dialog file creation(1/3)

## 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
1 conf: +
2   timeout: 10 +
3   +
4   exec_list: +
5     - expect: 'password:' +
6       exec: '${{__loginpassword__}}' +
7     - expect: '${{__loginuser__}}@${{__loginhostname__}}' +
8       exec: 'set terminal length 0' +
9     - expect: '${{__loginuser__}}@${{__loginhostname__}}' +
10    - exec: 'configure' +
11    - command: 'set system syslog host ${{item.0}} facility all level ${{VAR_log_severity}}' +
12      prompt: 'vyos@${{__loginhostname__}}' +
13      with_items: +
14        - '${{VAR_syslog_server_ip}}' +
15      when: +
16        - VAR_log_severity is define +
17      expect: 'vyos@${{__loginhostname__}}' +
18      exec: 'commit' +
19      - expect: 'vyos@${{__loginhostname__}}' +
20      exec: 'save' +
21      - expect: 'vyos@${{__loginhostname__}}' +
22      exec: 'exit' +
23      - command: 'show configuration' +
24      prompt: 'vyos@${{__loginhostname__}}' +
25      with_items: +
26        - '${{VAR_syslog_server_ip}}' +
27      failed_when: +
28        - stdout match(host ${{item.0}}) +
29
30 EOF
```

(E.g.) vyos\_set\_syslog\_server.yml

```
ios_set_syslog_server.yml
1 conf: +
2   timeout: 10 +
3   +
4   exec_list: +
5     - expect: 'Username:' +
6       exec: '${{__loginuser__}}' +
7     - expect: 'Password:' +
8       exec: '${{__loginpassword__}}' +
9     - expect: '${{__loginhostname__}}' +
10    - exec: 'enable' +
11    - expect: 'Password:' +
12      exec: '${{__loginpassword__}}' +
13    - expect: '${{__loginhostname__}}' +
14      exec: 'terminal length 0' +
15    - command: 'show logging' +
16      prompt: '${{__loginhostname__}}' +
17      register: result_stdout +
18    - expect: '${{__loginhostname__}}' +
19      exec: 'configure terminal' +
20      command: 'logging host ${{item.0}}' +
21      prompt: '${{__loginhostname__}}' +
22      with_items: +
23        - '${{VAR_syslog_server_ip}}' +
24      exec_when: +
25        - result_stdout no match(Logging to ${{item.0}}) +
26      command: 'Logging facility ${{VAR_log_facility}}' +
27      prompt: '${{__loginhostname__}}' +
28      when: +
29        - VAR_log_facility is define +
30      command: 'logging trap ${{VAR_log_severity}}' +
31      prompt: '${{__loginhostname__}}' +
32      when: +
33        - VAR_log_severity is define [EOF]
```

(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".

Register

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

\*is a required item.

[Back](#) [Register](#)

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"/>	<input type="button" value="Setting"/>	

\* is a required item.

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

## 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.

**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".

Register

Dialog ID	Dialog type*	OS type*	Dialog file*
Auto-input	<input type="button"/>	<input type="button"/>	<input type="button"/> Choose file No file chosen <input type="button"/> Upload in advance Upload status:

\* is a required item.

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".

Register

Associated item No.	Movement*	Dialog type*	Include order*
Auto-input	<input type="button" value="▼"/>	<input type="button" value="▼"/>	<input type="text"/>

\* is a required item.

[Back](#) [Register](#)

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".

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

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".

Register

Managed system item number	HW device type	Host name*	IP address*	EtherWakeOnLan	Login user ID	Management	Ansible Dedicated information
				MAC address	Network device name		Pioneer Dedicated information
							Protocol
Auto-input							OS type
							Connection options

\* is a required item.

2

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

Back Register

## 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

### 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 and see the next page.

Menu creation information

**Id :** Auto-input

**Menu name\*** : Pioneer practice

**Creation target :** Parameter Sheet(Host/Operal ▾)

**Display order\*** : 3

**Create as hostgroup menu :**  Yes

**Create as vertical menu** ⓘ  Yes

**Last modified :** Auto-input

**Last updated by :** Auto-input

**Target menu group**

**Input\*** : Input

**Substitution value\*** : Substitution value

**Reference\*** : Reference

1

Input the following information.(Next item)

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

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

- syslog\_server\_ip**: String, Maximum number of bytes: 32, Regular expression: (empty), Required: unchecked, Unique constraint: unchecked.
- sub\_syslog\_server\_ip**: String, Maximum number of bytes: 32, Regular expression: (empty), Required: unchecked, Unique constraint: unchecked.
- log\_facility**: String, Maximum number of bytes: 32, Regular expression: (empty), Required: unchecked, Unique constraint: unchecked.
- log\_severity**: String, Maximum number of bytes: 32, Regular expression: (empty), Required: unchecked, Unique constraint: unchecked.

A callout bubble with the number 1 points to the 'Item' button in the toolbar. A callout bubble with the number 2 points to the second item in the list. A callout bubble with the number 3 points to the 'Create' button at the bottom of the detailed view for the first item, which shows a table with three rows of data: No., Host name, Operation name, and Reference.

Below the table, a callout bubble with the number 3 points to the 'Create' button.

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

## 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".

Register

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	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".

Register

Parameter sheet		Registration method*	IaC variable			
Item No.	Menu group:Menu		Item	Movement	Key variable	Value variable
Variable name	Substitution order	Variable name	Substitution order			
Auto-input		Select menu	Select movement	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".

"Target host" menu

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

"Substitute value" menu

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

# 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

1 Select the Movement to execute.

2 Select Operation.

3 Click "Execute".

**Tips**

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

Menu

Main menu

Movement list

Dialog type list

OS type master

Dialog files

Movement details

Substitution value auto-registration setting

Target host

Substitution value list

Execution

Check operation status

Execution list

Description

Scheduling

Movement [Filter]

Movement [List]

Select Movement ID: 14 Log server configuration Ansible Pioneer

Movement Name: Log server configuration

Orchestrator: IP

Delay timer: 0

Dedicated information for ansible

host specific format: Number of parallel executions: 1

Variable count: 1

Remarks: Last update date/time: 2020/12/21 14:38:07

Last updated by: System Administrator

Operation [Filter]

Operation [List]

Select No. Operation ID: 1 Operation1

No. Operation ID: 2 Test Operation

No. Operation ID: 6 Basic settings all

No. Operation ID: 7 opel

No. Operation ID: 8 OPI

No. Operation ID: 9 operation

No. Operation ID: 10 LegacyHole Practice

No. Operation ID: 11 Pioneer\_Practice

Operation name: 1 Operation1

Operation name: 2 Test Operation

Operation name: 6 Basic settings all

Operation name: 7 opel

Operation name: 8 OPI

Operation name: 9 operation

Operation name: 10 LegacyHole Practice

Operation name: 11 Pioneer\_Practice

Scheduled date: 2020/08/27 16:00:00

Scheduled date: 2020/10/08 16:00:00

Scheduled date: 2020/10/08 16:00:00

Scheduled date: 2020/11/12 16:00:00

Scheduled date: 2020/11/21 00:00:00

Scheduled date: 2020/12/18 16:00:00

Scheduled date: 2020/12/21 10:57:57

Scheduled date: 2020/12/21 14:45:00

Remarks: group menu creation

Remarks: LegacyHole execution procedure

Remarks: Legacy execution procedure

Remarks: Data portability procedure

Remarks: Data portability procedure

Remarks: System Administrator

Remarks: Legacy execution procedure

Remarks: System Administrator

Remarks: Pioneer execution

Last update date/time: 2020/12/21 10:43:11

Last update date/time: 2020/10/23 16:21:05

Last update date/time: 2020/10/22 09:54:53

Last update date/time: 2020/11/10 14:00:49

Last update date/time: 2020/11/19 09:03:14

Last update date/time: 2020/12/03 15:50:03

Last update date/time: 2020/12/21 10:57:57

Last update date/time: 2020/12/21 15:41:13

Last updated by: LegacyHole execution procedure

Last updated by: Legacy execution procedure

Last updated by: Data portability procedure

Last updated by: Data portability procedure

Last updated by: System Administrator

Last updated by: Legacy execution procedure

Last updated by: System Administrator

Last updated by: Pioneer execution

Movement ID: 14

Movement Name: Log server configuration

Dry run

Execute

# 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/0000000001-7112.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
:
```

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"

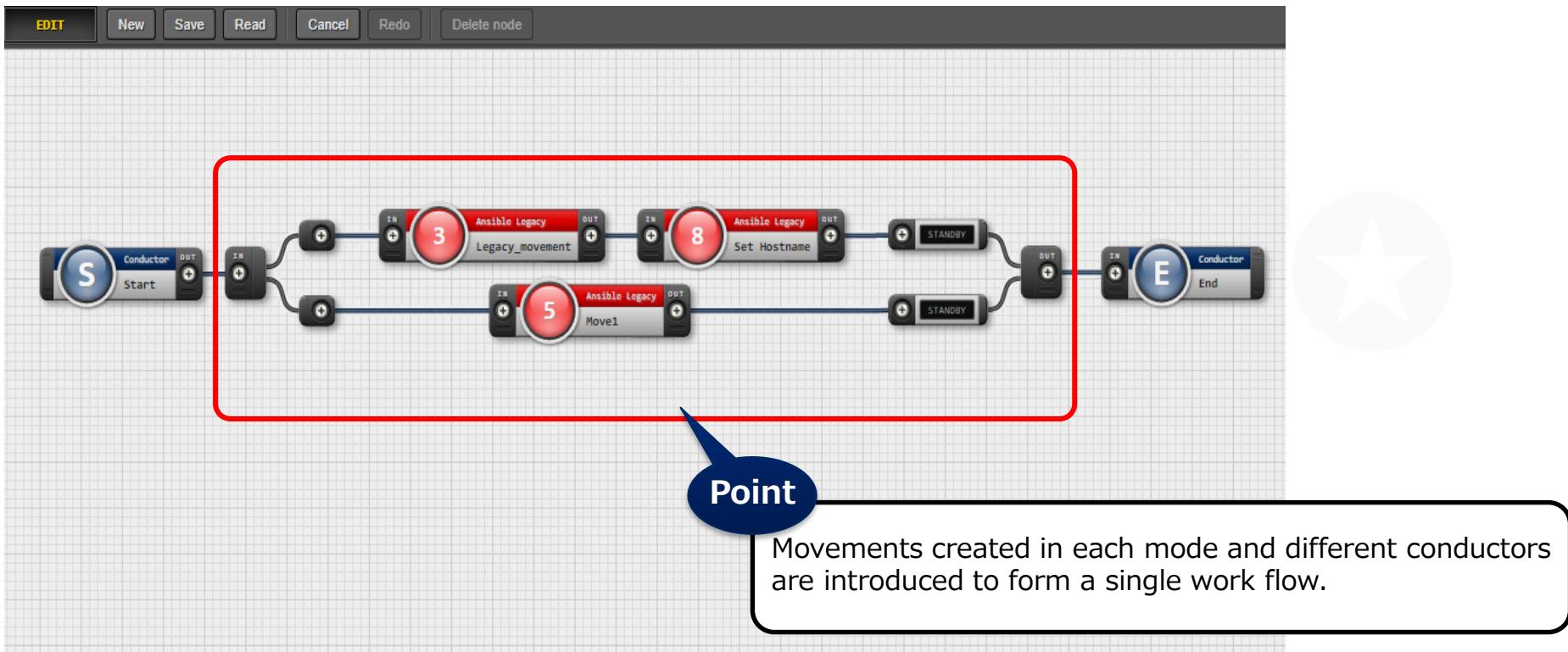
# A) Appendix



# 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**