IT Automation
CI/CD for IaC
【Practice】

※In this Document “Exastro IT Automation” will be written as “ITA”.

Exastro IT Automation Version 1.8

Exastro developer

Table of contents

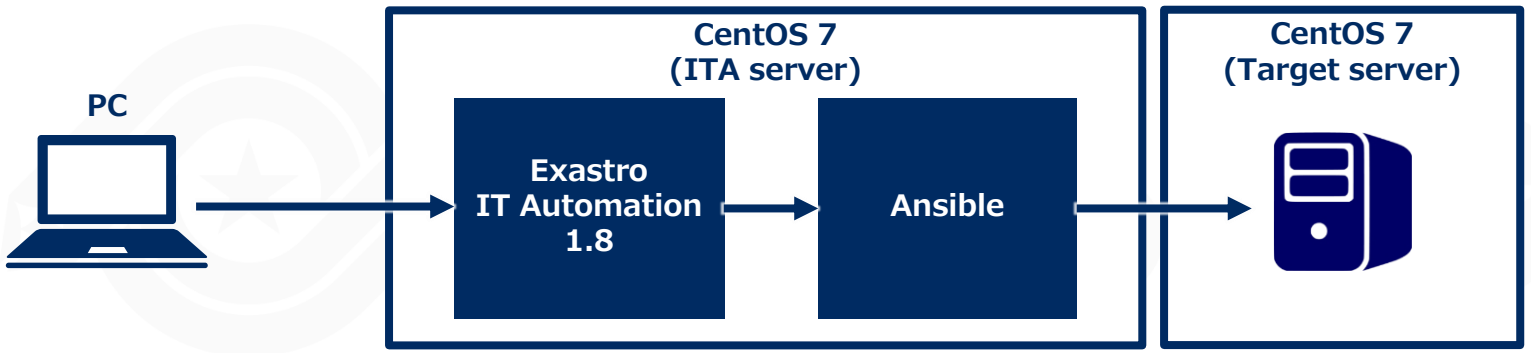
1. Introduction

Main Menu

- This document aims to teach the reader about the CI/CD for IaC function by guiding them through a simple scenario.
- This document requires the reader to have finished the scenario in the “Quickstart” guide before they can follow this scenario.

Operation environment

- The environment used in this document is as follows.
- The user must have a GitHub account in addition to the ITA server and the Target server.



- Exastro IT Automation 1.8
- CentOS 7(for ITA server)
- CentOS 7(for Target machine)

Scenario

- The figure below illustrates the steps and contents of this document's scenario.

①Register remote repository

②Register registration account

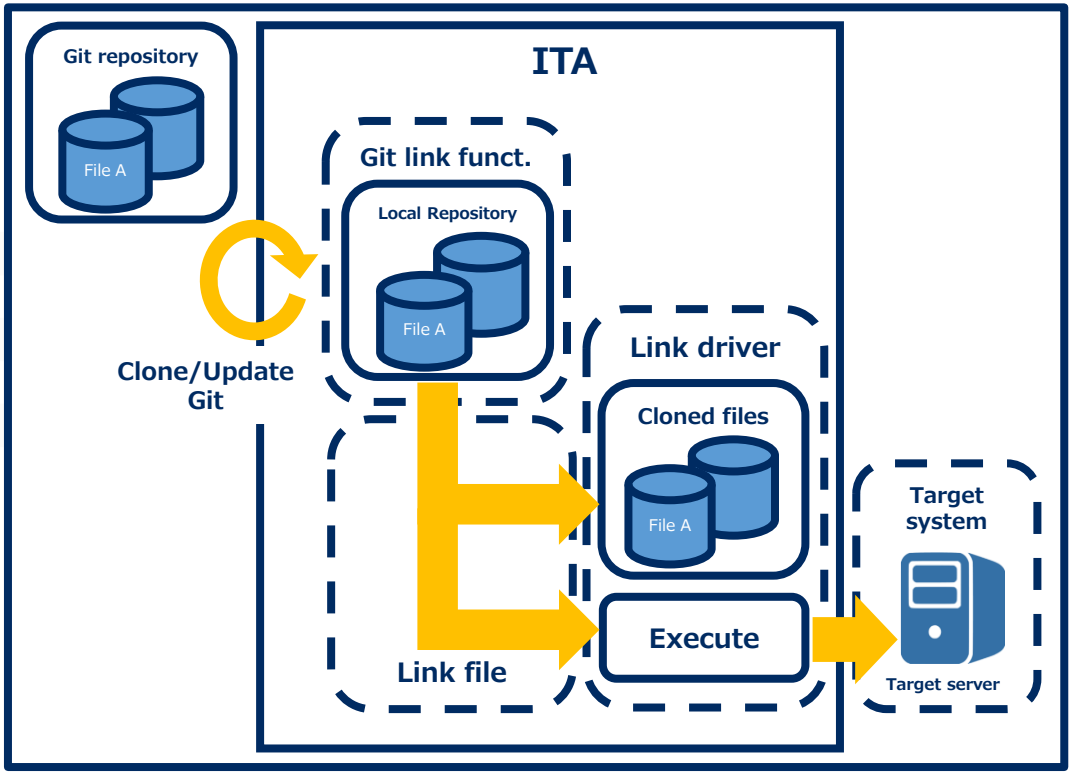
③Register file link

④Dry run(No.1)

⑤Edit Playbook file

⑥Dry run (No.2)

⑦Execute to target server



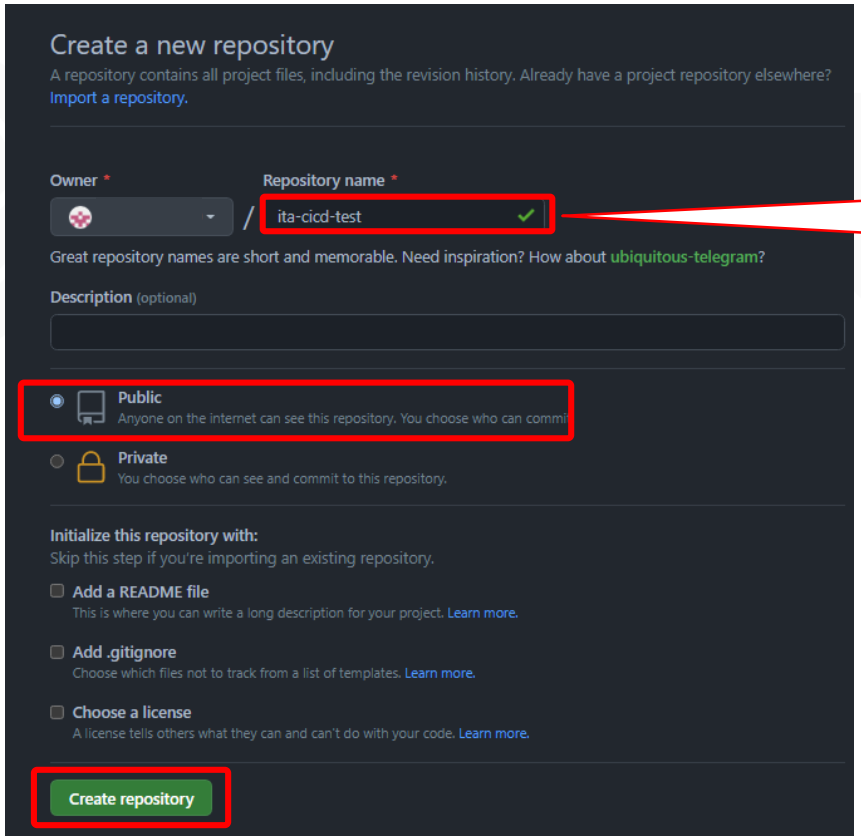
1.4 Preparation(1/3)

Prepare Git repository

- In this scenario, we will use GitHub.

  Go to Repository and select “New” to create a new repository.

  Input your desired repository name and select Public. Create the repository by pressing the “Create repository” button at the bottom of the screen.



Repository name
(Free name)

Select Public

1.4 Preparation(2/3)

Prepare Playbook file

- In this scenario, we will use the following Playbook

yum_package_install_check.yml

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

- name: Check yum list
  shell:yum list installed | grep "{{ item }}"
  register: result
  with_items:
    - "{{ VAR_packages }}"
```

Point

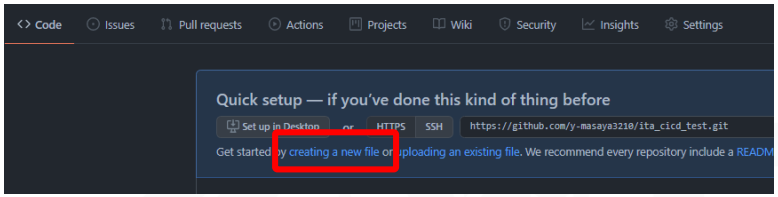
※The file has an invalid indent here on purpose.
We will fix it later as a part of the scenario.

1.4 Preparation(3/3)

Upload Playbook

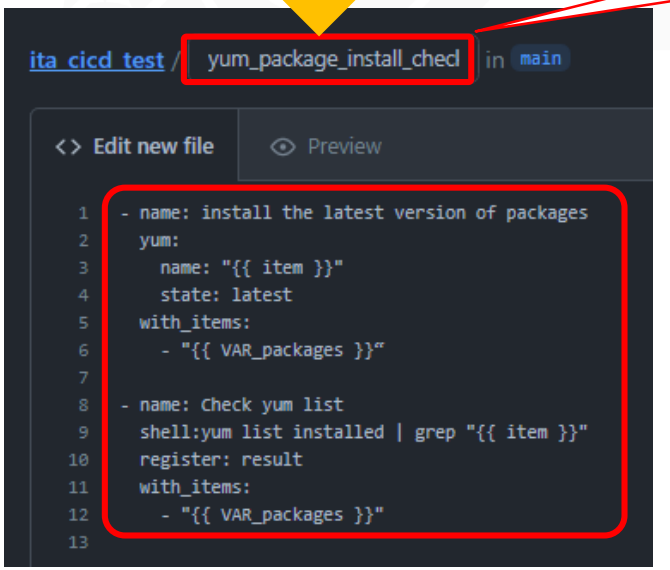
- Upload the Playbook to GitHub.

From the “Code” tab, click “Create a new file” and paste the contents of the playbook from the previous slide. Scroll down to the bottom of the page and press “Commit new file”.

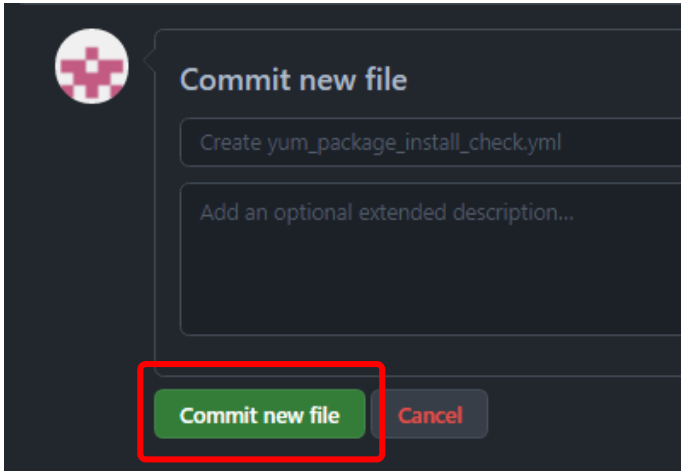


Repository name
yum_package_install_check.yml

See this page for the contents.



Scroll down

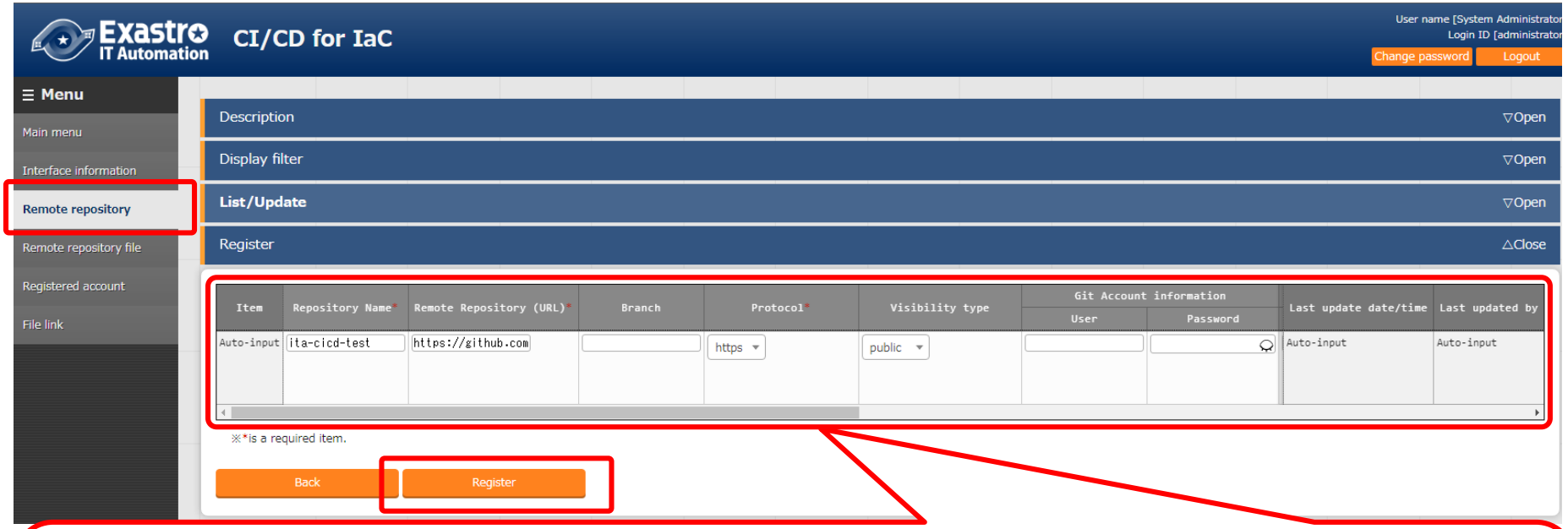


2. Scenario

2.1 Register Remote repository

Register the Git repository information.

- In this step, we will register the information of the GitHub account we prepared earlier. In ITA, go to the “CI/CD for IaC” menu and click “Remote repository”. Follow the table below and register a new item.

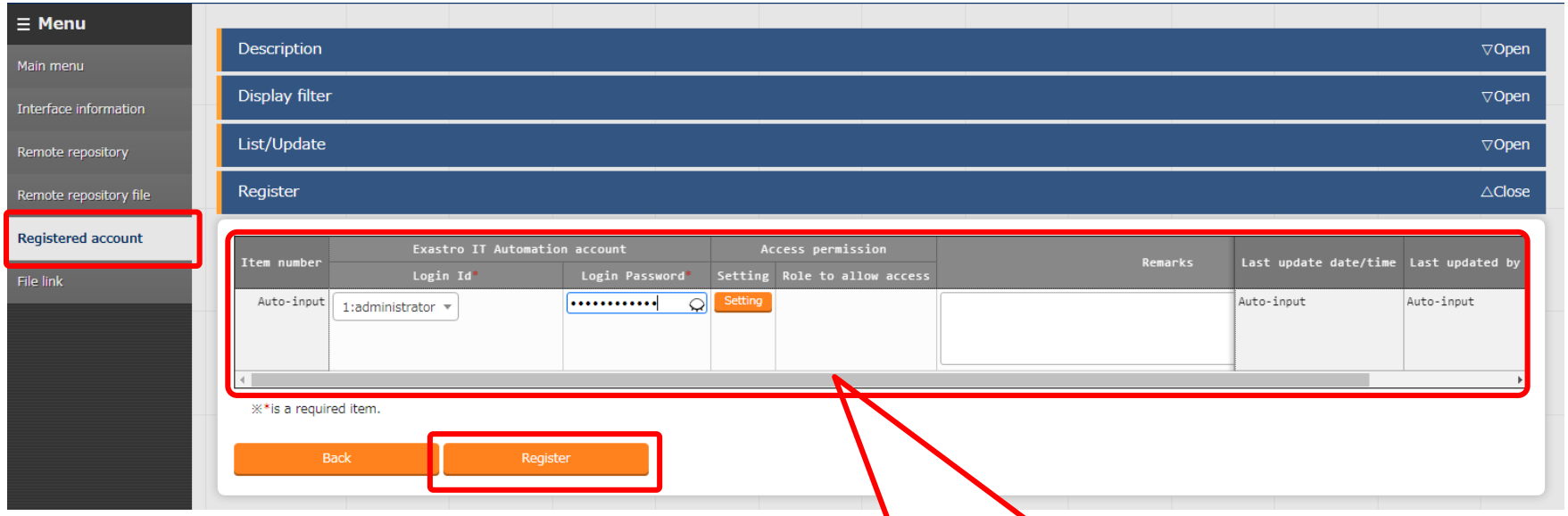


Remote Repository name	Remote Repository (URL)	Protocol	Visibility type	Remote repository sync info(Sync cycle)
(Name of the repository name)	(URL of the repository name)	https	public	Valid

2.2 Register Registered account

Register the account information needed to access the cloned files.

- Register a new item with the Exastro ITA account information.
  In this scenario, we will use the “administrator” user
  Press “Registered account” and follow the table below to create a new item.

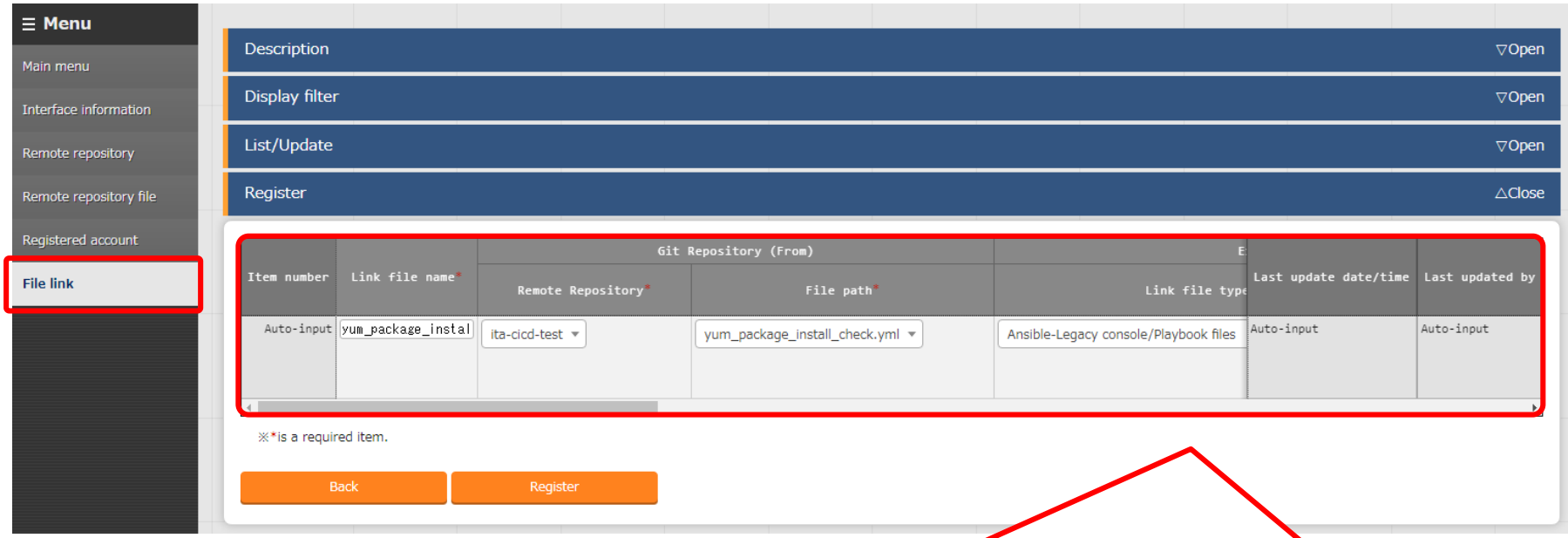


Login ID	Login PW
administrator	(Password)

2.3 Register file link(1/2)

Register a file link for the source files and the cloned files.

- In this section, we will link the source files and the cloned files and register an Operation and a Movement that will check the validity of the cloned files.
  Go to the “file link” menu and create a new item using the table below.

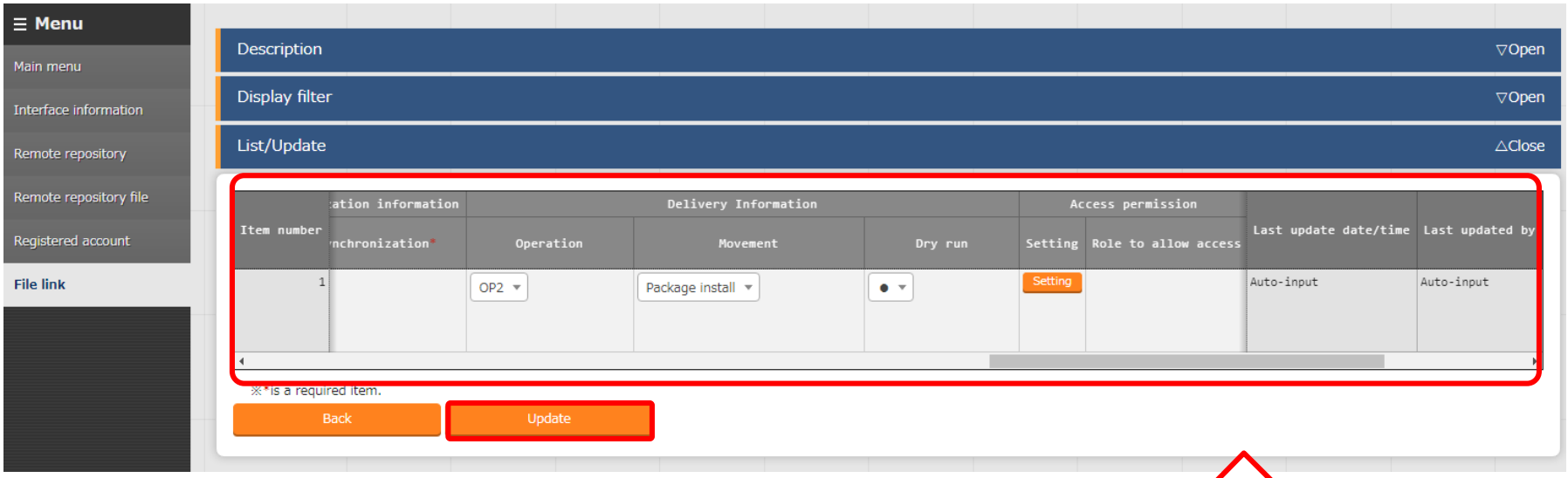


Link destination file name	Remote repository	File path	Link destination file type	Execution login ID	Automatic synchronization
yum_package_install	ita_cicd_test	yum_package_install_check.yml	Ansible-Legacy console /Playbook files	1:administrator	Valid

2.3 Register file link(2/2)

Register Operation and Movement and select “Dry run”.

- After having filled out the items from the previous slide, scroll to the right and fill out the following 3 items and press “Register”.



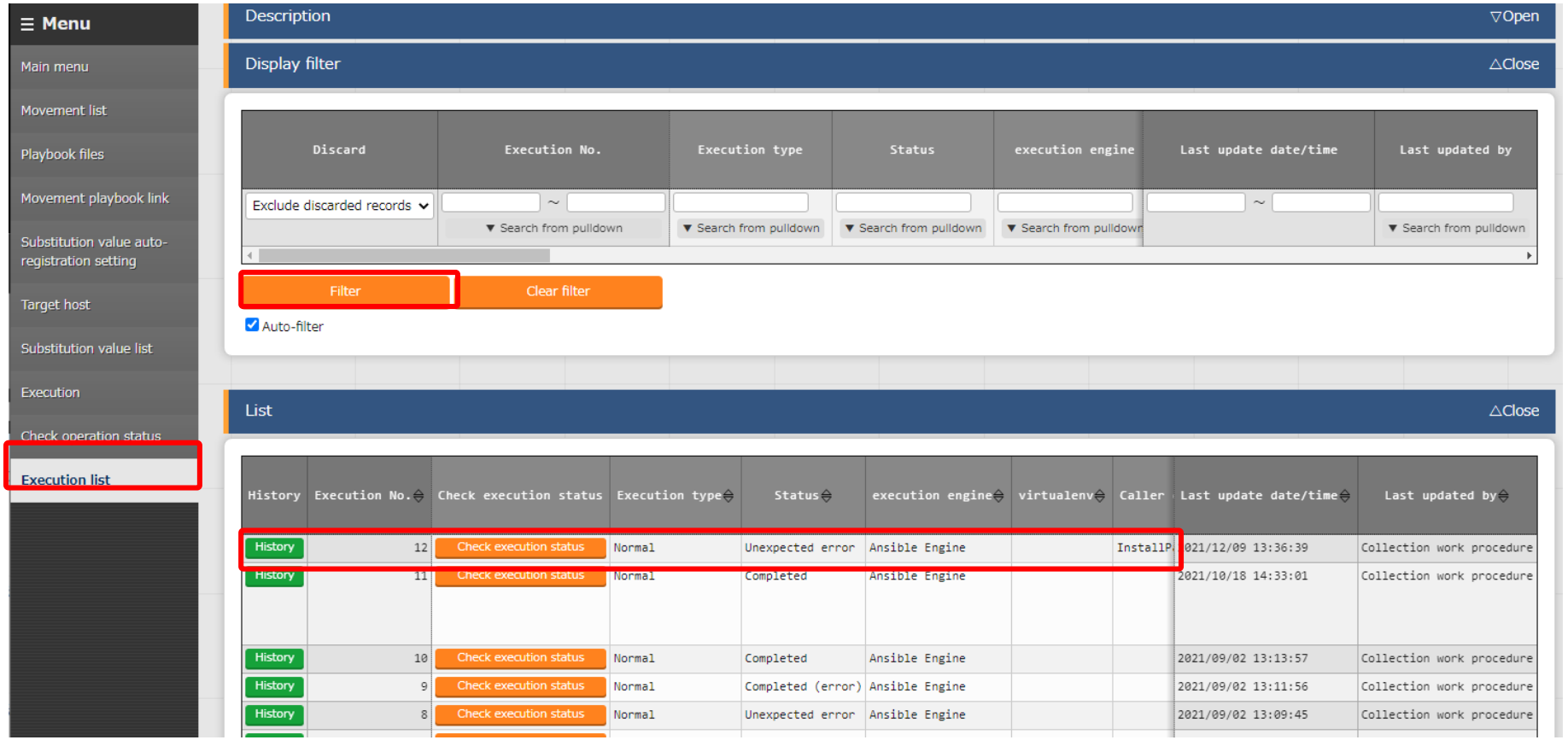
Operation	Movement	Dry run
Operation 2	Package install	●

2.4 Dry run (No.1)(1/2)

Check that the operation has dry run.

- Linking files will automatically start a dry run.

Go to the Ansible Legacy menu and click “Execution list”. From there, click “Filter” to see all executed operations. Find the operation we dry ran earlier and press the “Operation status check” to see the error contents.



2.4 Dry run (No.1)(2/2)

Check that the operation has dry run.

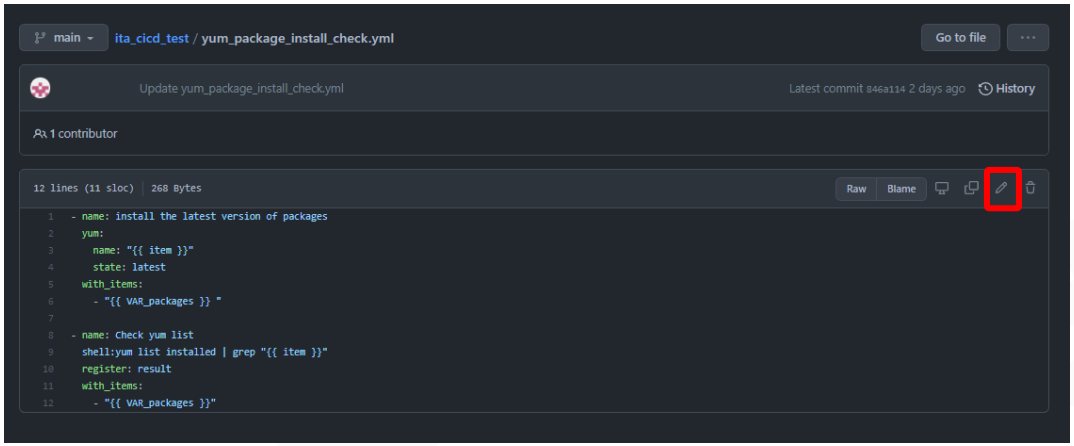
- Scroll down to see the Progress log. Users can use this to see the contents of the error. As mentioned earlier, the file contained an invalid indent which caused the error.

2.5 Edit Playbook

Access GitHub and edit the Playbook file.

- In this section, we will fix the error from the previous slide.

Access the GitHub file and press the edit icon. Follow the steps below and press “Commit changes”.

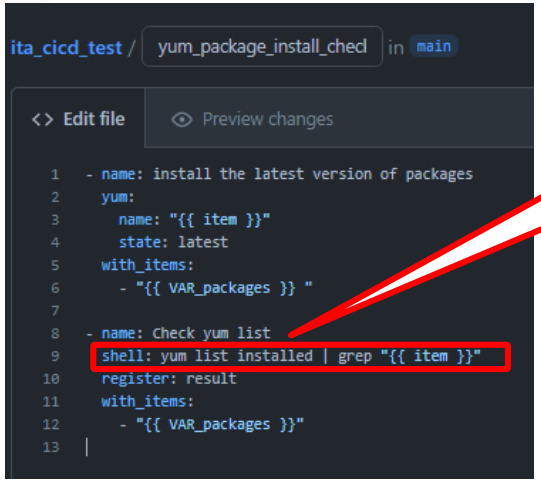


Input a space between “:” and “y”

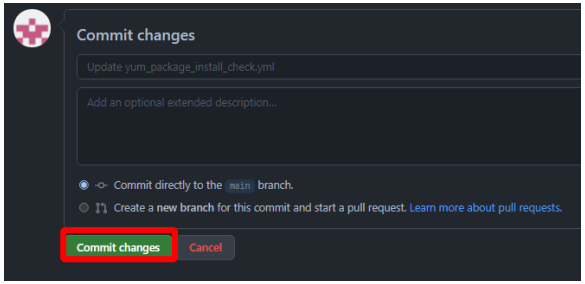
```
shell:yum list installed | grep "{{ item }}"
```

↓

```
shell: yum list installed | grep "{{ item }}"
```



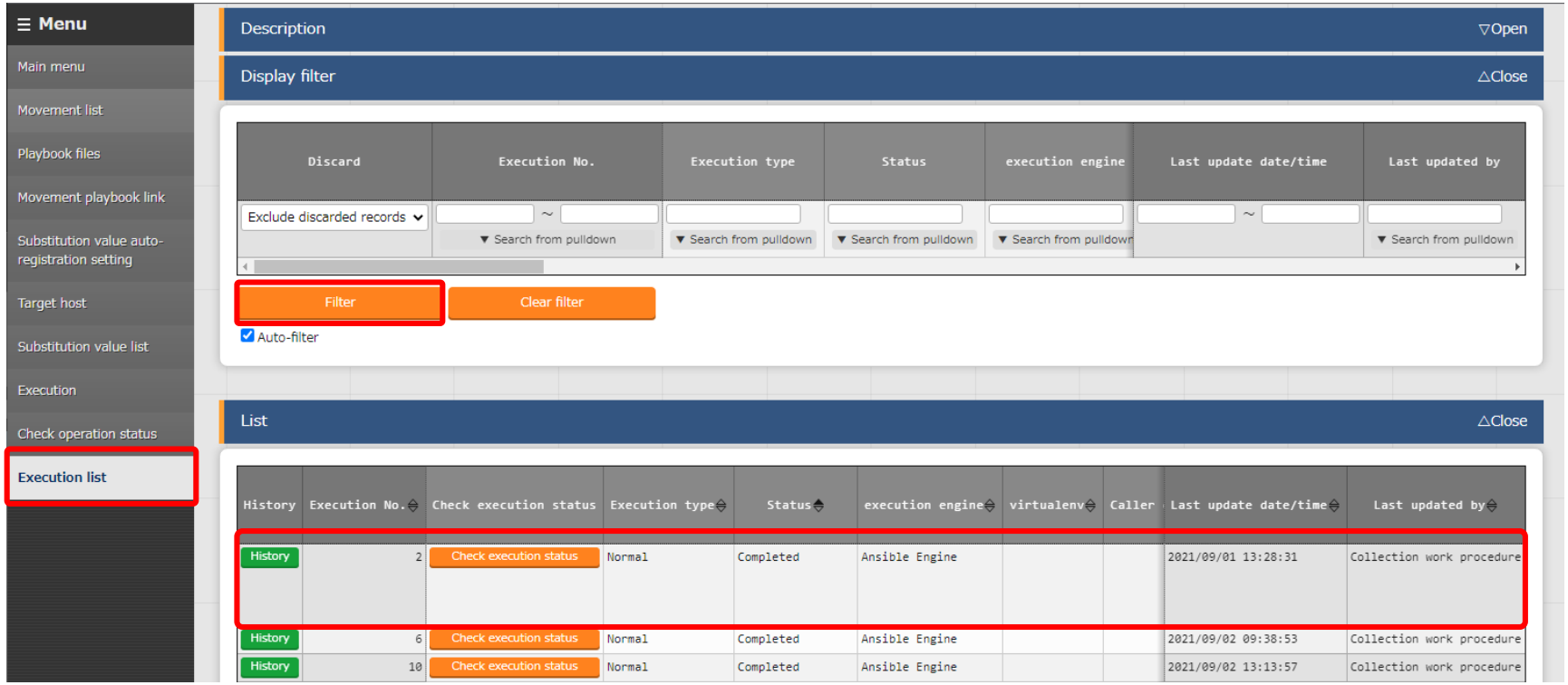
Scroll down



2.6 Dry run (No.2)

Check that the operation has dry run.

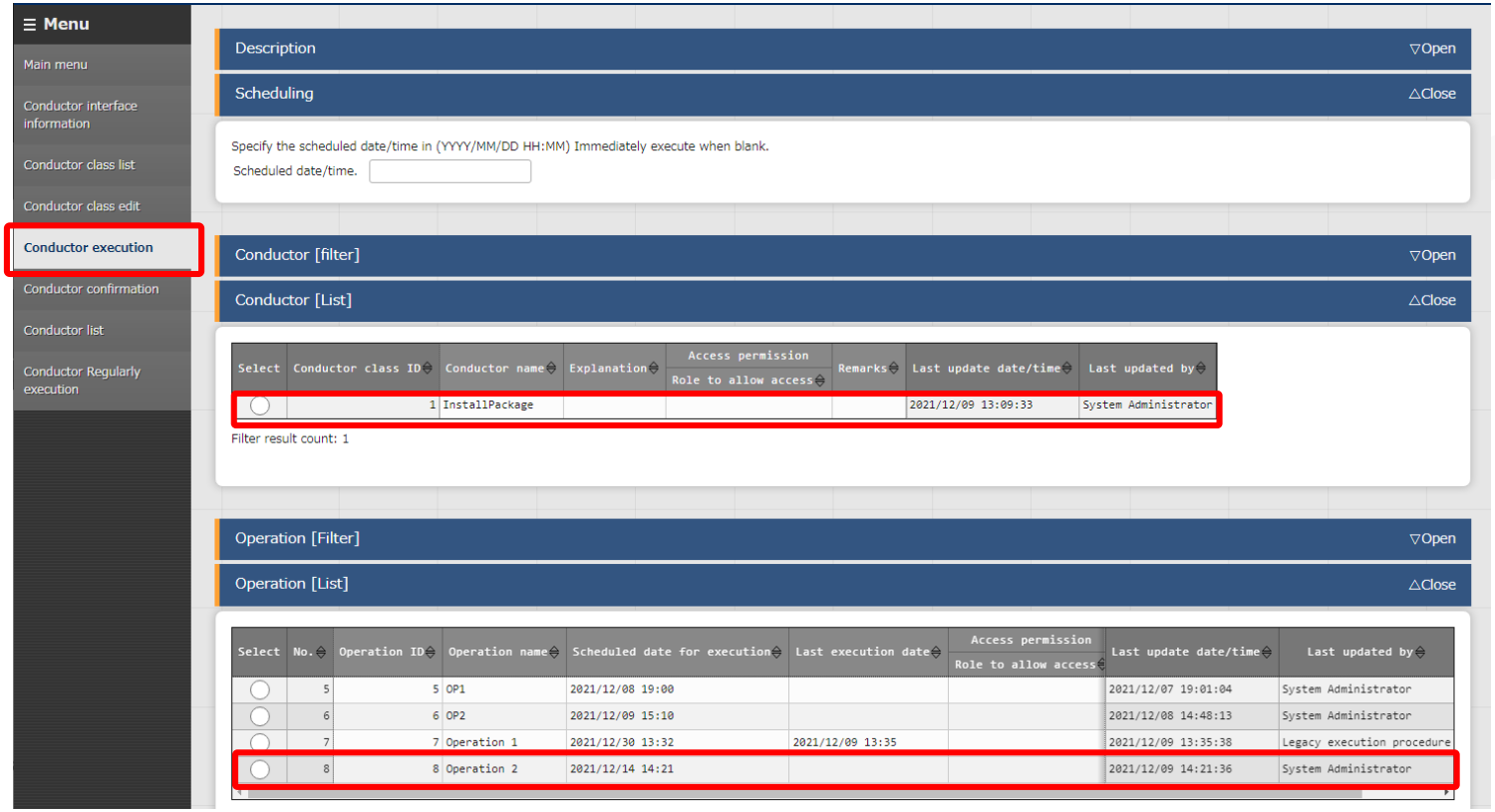
- Updating the GitHub source file will automatically update and dry run the ITA Playbook clone file.
  Do the same as Dry run(No.1) and go to “Ansible Legacy”-> “Execution list” and find the operation. The last dry run ended in an error, but this run should end successfully.



2.7 Execute to Target server(1/2)

Run the operation and apply to the Target server.

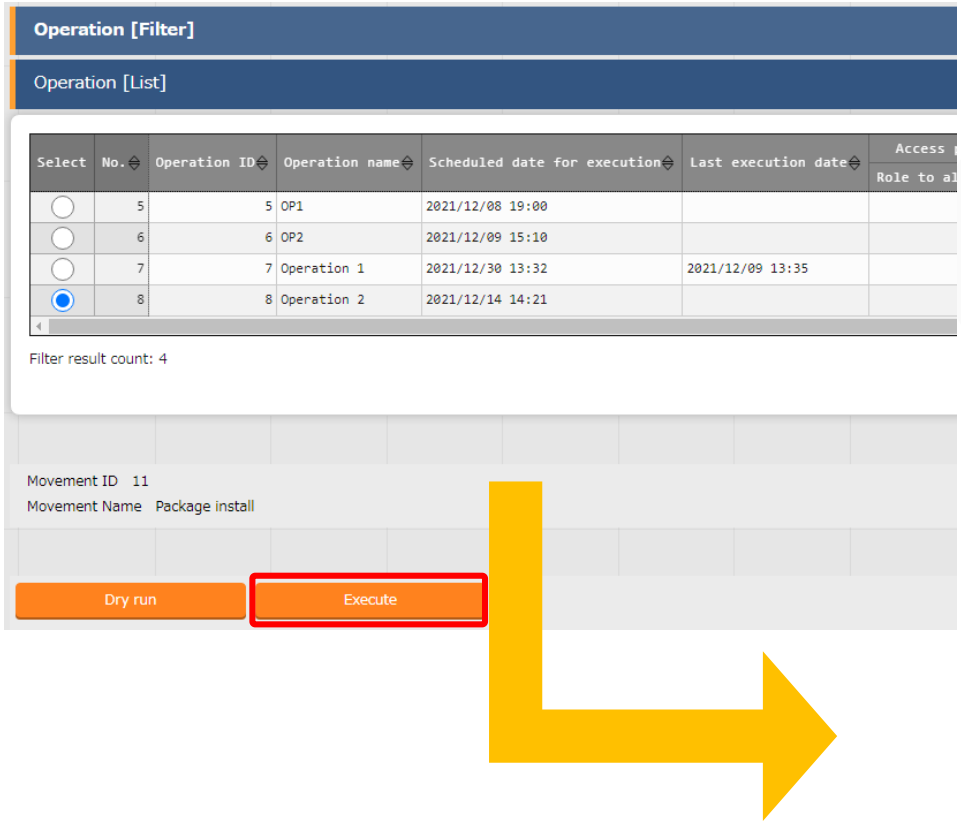
- Now that we have used the Dry run to see that there are no problems with the Playbook, we can apply it to the target server. Go to Ansible Legacy > Execution. Here we can select what Movement and Operation that we want to run.



2.7 Execute to Target server(2/2)

Run the operation and apply to the Target server.

- After selecting the operation and Movement, press “Execute”.
  Executing any operation will move the user to a screen where they can see the status of the running operation. If the operation status says “Completed”, the operation has ended successfully.



Item			Value
Execution No.			11
Execution type			Normal
Status			Completed
execution engine			Ansible Engine
Caller symphony			
Caller conductor			
Execution user			System Administrator
Movement	ID		11
	Name		Package install
	Delay timer (minutes)		
	Dedicated information for ansible	Host specific format	IP
		WinRM connection	
Operation	No.		8
	Name		Operation 2
	ID		8
Host management			confirmation
Substitution value			confirmation
Input data	Populated data		InputData_0000000011.zip
Output data	Result data		ResultData_0000000011.zip
Operation status	Scheduled date/time		
	Start date/time		2021/10/18 14:32:47
	End date/time		2021/10/18 14:32:53


Exastro