



# **IT Automation Terraform Driver 【 Practice 】**

In this Document, "IT Automation" will be written as "ITA".

# Table of contents

## [1.Introduction](#)

### [1.1 Introduction](#)

### [1.2 Environment](#)

## [2. Terraform Driver Practice](#)

### [2.1 Scenario](#)

### [2.2 Preparation](#)

## [3. Preparation](#)

### [3.1 Interface Information Registration](#)

### [3.2 Registering and linking Organization](#)

### [3.3 Registering and linking Workspace](#)

### [3.4 Operation pattern \(Movement\) Registration](#)

### [3.5 Module file Registration](#)

### [3.6 Specify Module file to Movement](#)

## [4. Execution](#)

### [4.1 Operation registration](#)

### [4.2 Setting variable values](#)

### [4.3 Execution](#)

### [4.4 Checking Operation status](#)

### [4.5 Change values and re-run](#)

# 1. Introduction

# 1. Introduction

## Main Menu

- This document aims to teach the readers about the **Terraform** Menu group while giving them hands-on experience.

The screenshot displays the Exastro IT Automation dashboard. The top navigation bar includes the Exastro logo, user information (System Administrator), and links for password change and logout. The main content area is divided into a left sidebar for menu navigation and a central workspace for widgets.

**Main menu:** A grid of icons representing different menu groups. The **Terraform** icon, located in the bottom row, is highlighted with a red rectangle. Other visible icons include Management Console, Basic Console, Export/Import, Symphony, Conductor, Create Menu, Input, Substitution value, Reference, Contrast, HostGroup management, Ansible Common, Ansible-Legacy, Ansible-Pioneer, Ansible-LegacyRole, Cobbler, and File control management.

**Widgets:** The right side of the dashboard features several data visualization widgets:

- Movement:** A donut chart showing 14 total movements, with 78.6% (orange) and 21.4% (green) segments. Below the chart is a table:

Movement	SUM
Ansible Legacy	11
Ansible Pioneer	1
Ansible Legacy Role	1
Terraform	1

- Work status:** A circular progress indicator showing 0 total status. Below the chart is a table:

Status	CON	SYM	SUM
Executing	0	0	0
Unexecuted (schedule)	0	0	0
Unexecuted	0	0	0

- Work result:** A donut chart showing 8 total results, with 62.5% (red) and 37.5% (green) segments. Below the chart is a table:

Result	CON	SYM	SUM
Normal end	2	0	2
Abnormal end	0	0	0
Unexpected error	4	1	5
Emergency stop	0	0	0
Schedule cancellation	0	0	0

- Work history:** A table listing individual work items.

The bottom left corner of the dashboard includes a "Contact administrator" link.

## 1.2 Environment

### Environment

- The working environment used in this manual is as follows.
- In addition to an ITA Server, please prepare an account for AWS, Azure and Terraform Cloud.

#### ITA host server

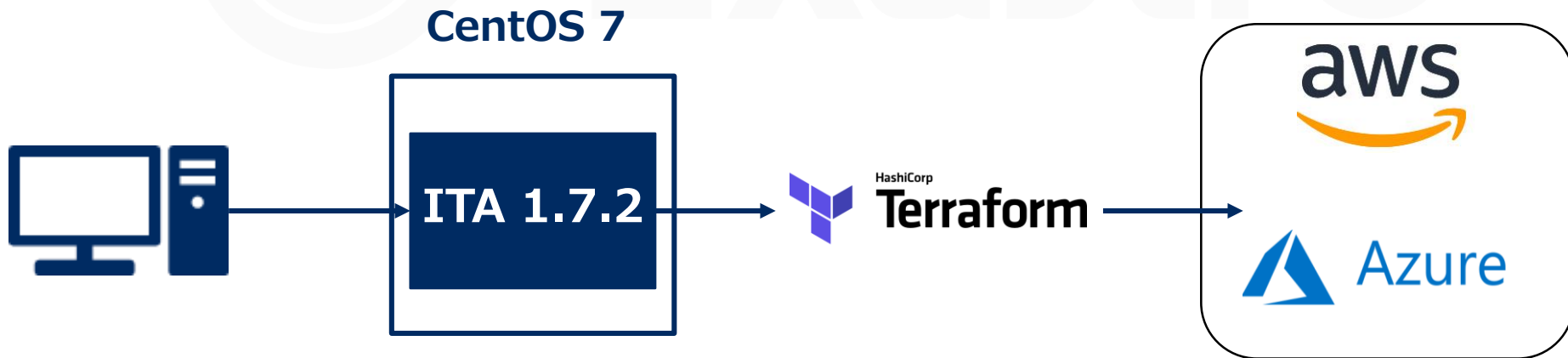
- CentOS7 (\*)
- ITA ver 1.7.2

#### Terraform

- Terraform Cloud

#### Target AWS

- Microsoft Azure



\*In this scenario, the host server will be running on CentOS7, but ITA can be installed on RHEL7 and RHEL8 type OS as well.

## 2. Terraform Driver Practice

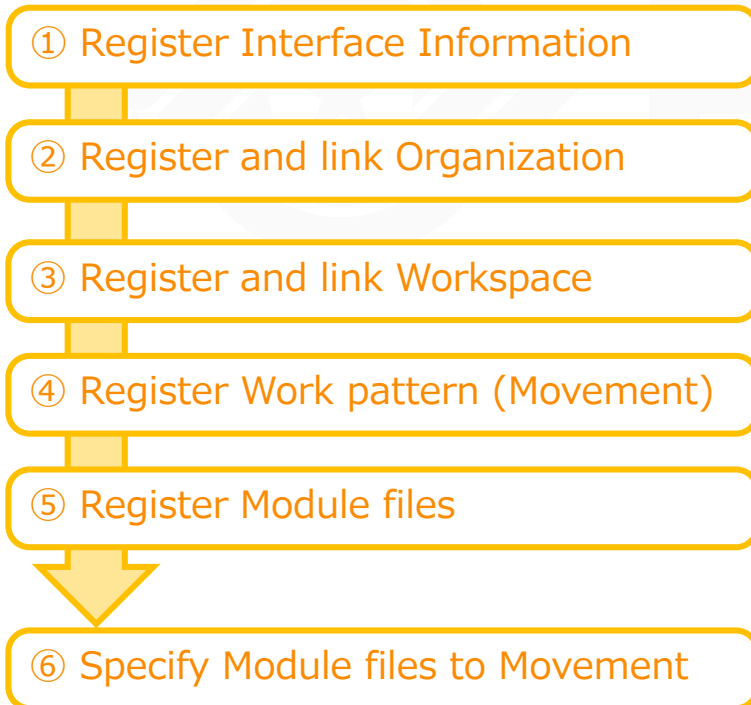
## 2.1 Scenario

### About the scenario

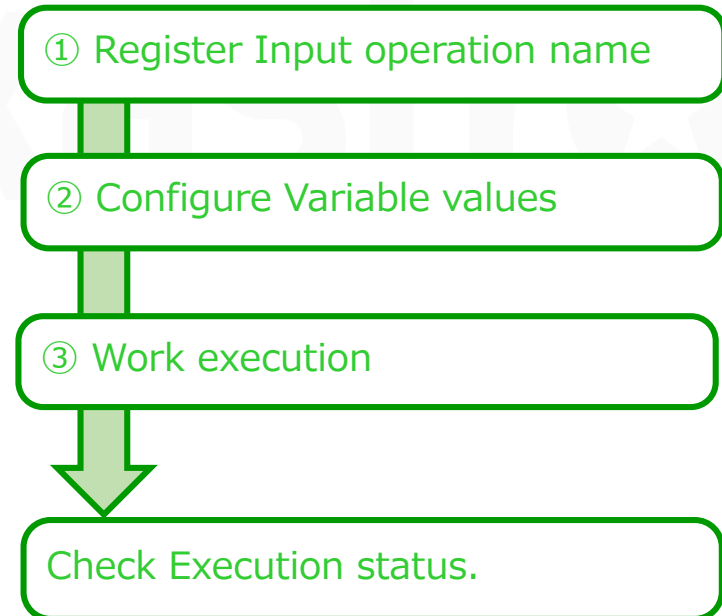
This scenario uses ITA's Terraform Driver to create 3 VM's on Public cloud (AWS,Azure)

Once you have followed the "Preparation" part of this document and have linked/registered all the necessary parts, you can repeat the "Execution" part of the scenario and reconfigure/re-register target machines. **(Automation)**

#### [Preparation]



#### [Execution]



## 2.2 Preparation (1/6)

### Create Modules

- Here, we will create the 4 modules that will be used in this scenario

【 Attention 】 Make sure that the character code is "UTF-8", the Line feed code is LF and the file extension is ".tf".

```
variable "access_key" {}
variable "secret_key" {}
variable "region" {}
variable "ami" {}
variable "key_name" {}
variable "security_group" {}
variable "tags_name" {}
variable "hello_tf_instance_count" {
  default = 2
}
variable "hello_tf_instance_type" {
  default = "t2.micro"
}
```

#### File name:

**aws\_create\_instance\_variables.tf**

This file defines variables for creating AWS Instances.

A concrete value variable will be assigned to the variable



## 2.2 Preparation (2/6)

### Create IaC

```
provider "aws" {  
  access_key = var.access_key  
  secret_key = var.secret_key  
  region = var.region  
}  
  
resource "aws_instance" "hello-tf-instance" {  
  ami          = var.ami  
  key_name     = var.key_name  
  security_groups = [var.security_group]  
  tags = {  
    Name = "${var.tags_name}-${count.index+1}"  
  }  
  count = var.hello_tf_instance_count  
  instance_type = var.hello_tf_instance_type  
}
```

**File name:**  
**aws\_create\_instance.tf**

This file defines variables for creating AWS Instances.

Create Security groups and key pairs in AWS in advance.

## 2.2 Preparation (3/6)

### Create IaC

```
variable "subscription_id" {}
variable "tenant_id" {}
variable "client_id" {}
variable "client_secret" {}
variable "resource_group_name" {}
variable "security_group" {}
variable "location" {}
variable "Vnet_name" {}
variable "Vnet_address_space" {}
variable "subnet_name" {}
variable "address_prefixes" {}
variable "public_ip_name" {}
variable "allocation_method" {}
variable "domain_name_label" {}
variable "network_interface_name" {}
variable "NIC_name" {}
variable "VM_name" {}
variable "VM_size" {}
variable "publisher" {}
```

```
variable "offer" {}
variable "sku" {}
variable "source_image_version" {}
variable "admin_username" {}
variable "ssh_public_key" {}
variable "os_disk_name" {}
variable "caching" {}
variable "storage_account_type" {}
variable "VM_count" {}
```

### File

#### **name:azure\_create\_instance\_variables.tf**

This file defines variables for creating Azure instances.

A concrete value variable will be assigned to the variable

## 2.2 Preparation (4/6)

### Create IaC

File name:


**azure\_create\_instance.tf (1/3)**

Resources for creating Azure instance  
Definition file.


This file creates resource groups, as well  
as their network security group and virtual  
networks.

It will also create the a virtual machine,  
disk and network interface for each VM.

```
provider "azurerm" {  
  features {}  
  subscription_id = var.subscription_id  
  client_id       = var.client_id  
  client_secret   = var.client_secret  
  tenant_id      = var.tenant_id  
}  
  
resource "azurerm_resource_group" "hoge" {  
  name = var.resource_group_name  
  location = var.location  
}
```




```
resource "azurerm_network_security_group" "hoge" {  
  name = var.security_group  
  location = azurerm_resource_group.hoge.location  
  resource_group_name = azurerm_resource_group.hoge.name  
  
  security_rule {  
    name           = "SSH"  
    priority       = 1001  
    direction      = "Inbound"  
    access         = "Allow"  
    protocol       = "Tcp"  
    source_port_range = "*"   
    destination_port_range = "22"  
    source_address_prefix = "*"   
    destination_address_prefix = "*"   
  }  
  
  security_rule {  
    name           = "HTTP"  
    priority       = 1002  
    direction      = "Inbound"  
    access         = "Allow"  
    protocol       = "Tcp"  
    source_port_range = "*"   
    destination_port_range = "80"  
    source_address_prefix = "*"   
    destination_address_prefix = "*"   
  }  
}
```



## 2.2 Preparation (5/6)

### Create IaC

**File name:**  
**azure\_create\_instance.tf (2/3)**




```
resource "azurerm_virtual_network" "hoge hoge" {
  name = var.Vnet_name
  address_space = [var.Vnet_address_space]
  location = azurerm_resource_group.hoge hoge.location
  resource_group_name = azurerm_resource_group.hoge hoge.name
}

resource "azurerm_subnet" "hoge hoge" {
  name = var.subnet_name
  resource_group_name = azurerm_resource_group.hoge hoge.name
  virtual_network_name = azurerm_virtual_network.hoge hoge.name
  address_prefixes = [var.address_prefixes]
}

resource "azurerm_public_ip" "hoge hoge" {
  count = var.VM_count
  name = "${var.public_ip_name}-${count.index}"
  location = azurerm_resource_group.hoge hoge.location
  resource_group_name = azurerm_resource_group.hoge hoge.name
  allocation_method = var.allocation_method
  domain_name_label = "${var.domain_name_label}-${count.index}"
}
```

```
resource "azurerm_network_interface" "hoge hoge" {
  count = var.VM_count
  name = "${var.network_interface_name}-${count.index}"
  location = azurerm_resource_group.hoge hoge.location
  resource_group_name = azurerm_resource_group.hoge hoge.name

  ip_configuration {
    name = var.NIC_name
    subnet_id = azurerm_subnet.hoge hoge.id
    private_ip_address_allocation = var.allocation_method
    public_ip_address_id = azurerm_public_ip.hoge hoge[count.index].id
  }
}
```



## 2.2 Preparation (6/6)

### Create IaC

**File name:**  
**azure\_create\_instance.tf (3/3)**



```
resource "azurerm_network_interface_security_group_association" "hoge hoge" {
  count = var.VM_count
  network_interface_id = azurerm_network_interface.hoge hoge[count.index].id
  network_security_group_id = azurerm_network_security_group.hoge hoge.id
}

resource "azurerm_linux_virtual_machine" "hoge hoge" {
  count = var.VM_count
  name = "${var.VM_name}-${count.index}"
  resource_group_name = azurerm_resource_group.hoge hoge.name
  location = azurerm_resource_group.hoge hoge.location
  size = var.VM_size
  admin_username = var.admin_username
  network_interface_ids = [azurerm_network_interface.hoge hoge[count.index].id]

  admin_ssh_key {
    username = var.admin_username
    public_key = var.ssh_public_key
  }

  os_disk {
    name = "${var.os_disk_name}-${count.index}"
    caching = var.caching
    storage_account_type = var.storage_account_type
  }

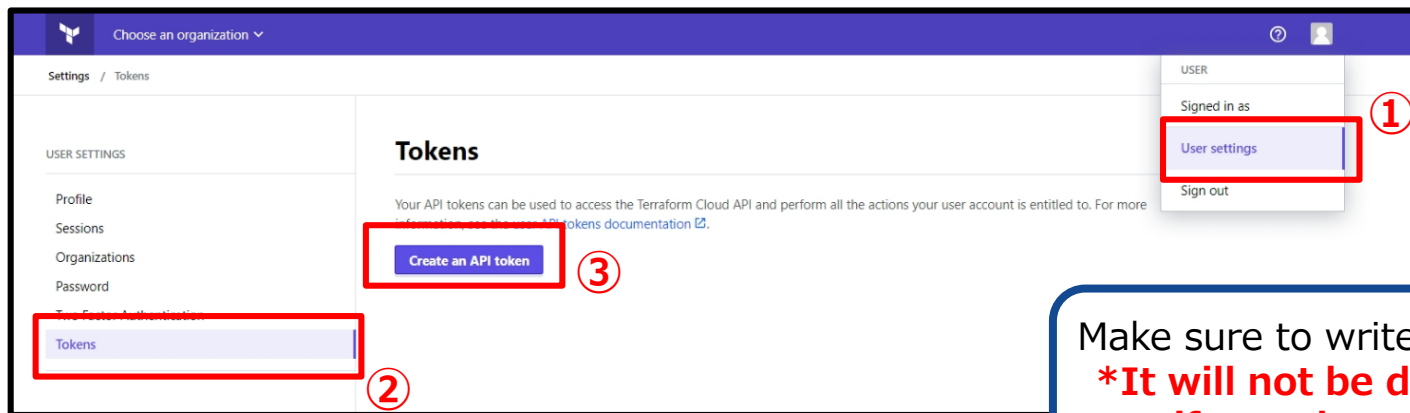
  source_image_reference {
    publisher = var.publisher
    offer = var.offer
    sku = var.sku
    version = var.source_image_version
  }
}
```

### 3. Preparation

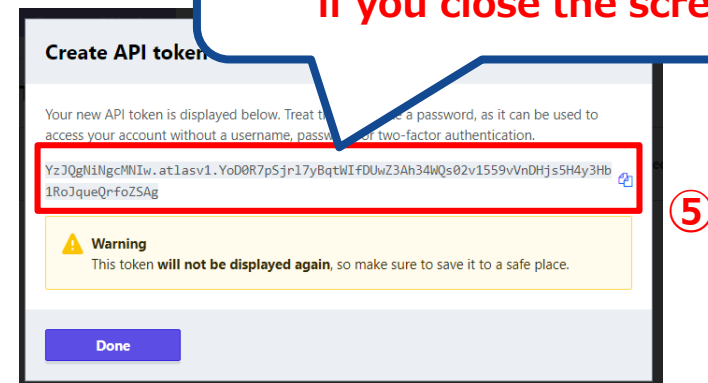
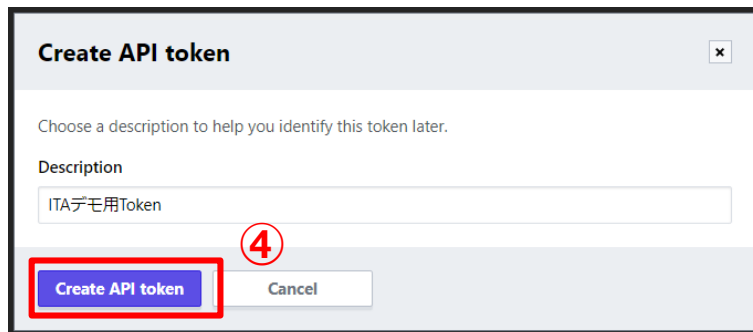
## 3.1 Register Interface Information(1/2)

### Create User Token

- In order to link Terraform Driver with Terraform, we will need to create a User Token from Terraform
- Log in to Terraform from your browser and go to [User Setting]→[Tokens]→[Create an API token]



Make sure to write down the token  
**\*It will not be displayed again if you close the screen.**



## 3.1 Registration of interface information(2/2)

### Interface information

- Input the Terraform Hostname and the created UserToken

\*Since only one Terraform can be linked to ITA, you need to update the item that is already there.

#### Menu: Terraform> Interface Information

- ① Open the list and press the item's update button.
- ② Enter the following information and press "Register".

The screenshot shows the ITA interface. On the left, the 'Main menu' has 'Interface information' highlighted. The main area shows a 'List' view of interface information. A table with the following columns is displayed: No., Hostname, User Token, Proxy (Address, Port), Status monitoring cycle (milliseconds)\*, and Number of rows to display progress status. The first row shows '1' in the 'No.' column, 'app.terraform.io' in the 'Hostname' column, and '\*\*\*\*\*' in the 'User Token' column. Below the table, there are 'Back' and 'Update' buttons. A red box highlights the 'Update' button in the bottom left corner of the screenshot.

Hostname

app.terraform.io

User Token

(Input User Token)



## 3.2 Register and Link Organization(1/2)

### Register Organization

In this step, we will create an Organization.

Menu: **Terraform > Organizations list**

- ① Click Register> Start Registration.
- ② Enter the following information and press "Register".

Organization ID	Organization Name*	Email address*	Access permission
Auto-input	ITAllearn_org		Setting Role to allow access

※\*is a required item.

Back Register

Organization Name	Email address
ITAllearn_org	(Input Mail Address)

## 3.2 Register and Link Organization(2/2)

### Link Organization

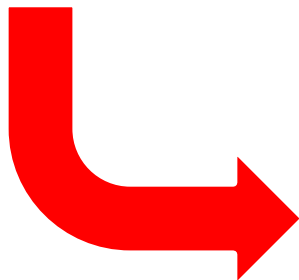
- After creating the Organization item from Organization Management  
You can check if the Organization has been added to the target Terraform by clicking the [Linkage status check].
- If it says "Not registered", you can press the register button to create an Organization in Terraform.

List/Update

History	Update	Discard	Organization ID	Organization Name	Email address	Terraform association				
						Status check	Association status	Register	update	Delete
History	Update	Discard	3	LearnTest1	ita-exastro@sample.com	Association status check	No registration	Register	update	Delete

Filter result count: 1

Output Excel



#### USER SETTINGS

Profile  
**Organizations**  
Password  
Two Factor Authentication  
Tokens

#### Organizations

You are a member of the following organizations:

LearnTest1 OWNER ...

## 3.3 Register and Link Workspace(1/2)

### Register Workspace

In this section, we will create a Workspace.

Menu : **Terraform > Workspaces list**

- ① Click Register> Start Registration.
- ② Enter the following information and press "Register".

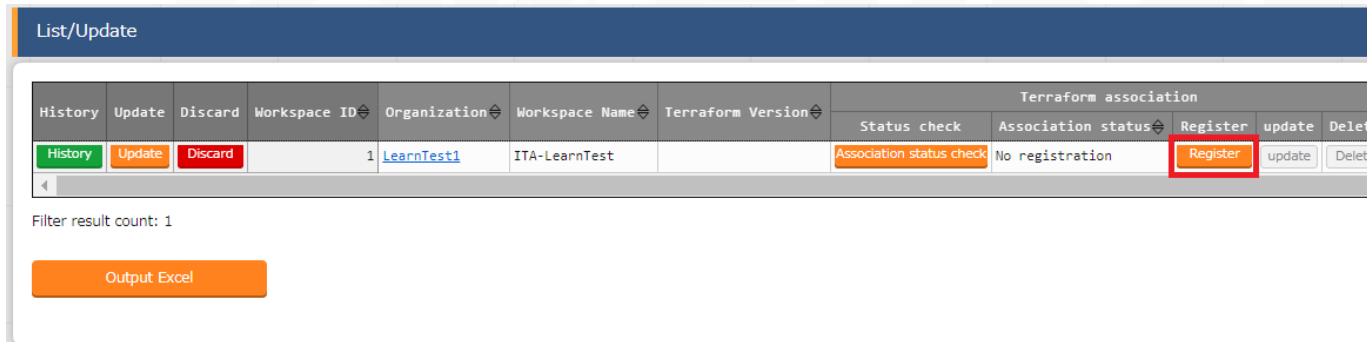
Organization	Workspace Name
ITAllearn_org	ITA-demo-AWS
ITAllearn_org	ITA-demo-Azure

## 3.3 Register and Link Workspace(2/2)

### Link Workspace

- After creating a Workspace item in Workspaces list, You can check if the Workspace has been added to the target Terraform by clicking the “Association status check” button
- If it says “Not registered”, you can press the “Register” button to create a Workspace on the target Terraform

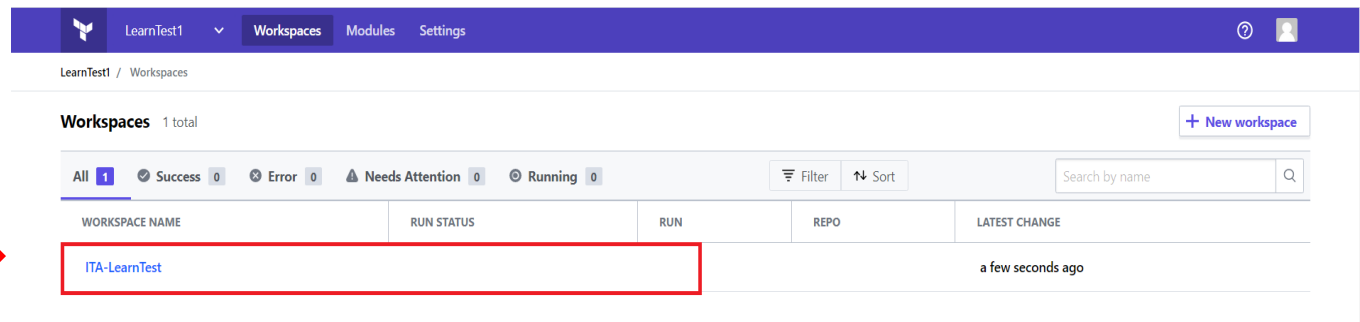
**\* As a Workspace is created inside an Organization, make sure to create an Organization first.**



List/Update										
History	Update	Discard	Workspace ID	Organization	Workspace Name	Terraform Version	Terraform association			
							Status check	Association status	Register	update
History	Update	Discard	1	LearnTest1	ITA-LearnTest		Association status check	No registration	Register	update

Filter result count: 1

Output Excel



LearnTest1 / Workspaces

Workspaces 1 total

+ New workspace

All 1 Success 0 Error 0 Needs Attention 0 Running 0

Filter Sort Search by name

WORKSPACE NAME	RUN STATUS	RUN	REPO	LATEST CHANGE
ITA-LearnTest				a few seconds ago

## 3.4 Register Operation pattern(Movement)

### Create Movement

In this section, we will register a Movement that we can link to the playbook we created earlier.

Menu: **Terraform> Movement list**

- ① Click Register> Start Registration.
- ② Select or enter the following and press "Register".

Movement ID	Movement Name*	Delay timer	Terraform integration
Auto-input	<input type="text"/>	<input type="text"/>	<input type="text" value="Organization:Workspace*"/>

※\* is a required item.

Back Register

Movement name	Terraform User information Organization: Workspace
CreateVM(AWS)	ITA-demo-AWS
CreateVM(Azure)	ITA-demo-Azure

## 3.5 Register Module files

### Register Module

In this section, we will register our Modules to ITA.

Menu: **Terraform> Module Files**

- ① Click Register> Start Registration.
- ② Press "Browse" and select your playbook and press "Upload in advance".
- ③ Follow the table below and press "Register"

Module file ID	Module file name*	Module file*
Auto-input	<input type="text"/>	<div>Choose File No file chosen</div> <div>Upload in advance</div> <div>Upload status:</div>

Module file name	Module file
aws_create_instance_variables	aws_create_instance_variables.tf
aws_create_instance_body	aws_create_instance.tf
azure_create_instance_variables	azure_create_instance_variables.tf
azure_create_instance_body	azure_create_instance.tf

## 3.6 Specify Module file to Movement

### Link Module to Movement

In this section, we will link our Movement and Module file

Menu: Terraform> **Movement-Module Link**

- ① Click Register> Start Registration.
- ② Select or enter the following and press "Register".

Register

Associated item number	Movement*	Module file*
Auto-input	<input type="text"/>	<input type="text"/>

※\* is a required item.

Back Register

Movement	Module file
Create VM (AWS)	aws_create_instance_variables
Create VM (AWS)	aws_create_instance_body
Create VM (Azure)	azure_create_instance_variables
Create VM (Azure)	azure_create_instance_body

## 4. Execution



# 4.1 Operation registration

## Register new Operation

In this section, we will create an Operation

Menu: **Basic Console > Operation List**

- ① Click Register> Start Registration.
- ② Enter the following information and press "Register".

Register

No.	Operation ID	Operation name*	Scheduled date for execution*
Auto-input	Auto-input	<input type="text"/>	<input type="text"/>

※\*is a required item.

Back

Register

Operation name	Scheduled implementation date and time
Terraform_demo	(Free field)

\* (Scheduled implementation date and time) is an item for management. The operation will not be automatically executed when the scheduled date passes .

## 4.2 Variable value setting (1/4)

### Configure values to Variables.

In this section, we will configure specific values to the Module variables

Menu: **Terraform> Substitution value list**

- ① Click Register> Start Registration.
- ② Select or enter the following and press "Register".

Register

Item number	Operation*	Movement*	Variable name*	HCL setting*	Sensitive setting*
Auto-input	<input type="text"/>	<input type="text"/>	Please select a Movement	OFF <input type="text"/>	OFF <input type="text"/>

※ \* is a required item.

Back

Register

## 4.2 Variable value setting (2/4)

### Configure values to Variables(1/3)

Please refer to the table below and register substitute values.

Operation	Movement	Variable name	Specific value
Terraform_demo	CreateVM(AWS)	security_group	ita-demo-sg *
Terraform_demo	CreateVM(AWS)	key_name	ita-demo-key *
Terraform_demo	CreateVM(AWS)	access_key	(AWS access key )
Terraform_demo	CreateVM(AWS)	secret_key	(AWS secret key )
Terraform_demo	CreateVM(AWS)	Region	( Any region )
Terraform_demo	CreateVM(AWS)	tags_name	ita-demo-instance
Terraform_demo	CreateVM(AWS)	hello_tf_instance_type	t2.micro
Terraform_demo	CreateVM(AWS)	hello_tf_instance_count	3
Terraform_demo	CreateVM(AWS)	AMI	( Any AMI)

\* Security groups and key pairs must be created in advance.

## 4.2 Variable value setting (3/4)

### Configure values to Variables(2/3)

Please refer to the table below and register substitute values.

Operation	Movement	Variable name	Specific value
Terraform_demo	CreateVM(Azure)	subscription_id	(Azure Authentication information )
Terraform_demo	CreateVM(Azure)	tenant_id	
Terraform_demo	CreateVM(Azure)	client_id	
Terraform_demo	CreateVM(Azure)	client_secret	
Terraform_demo	CreateVM(Azure)	resource_group_name	ita-demo-rg
Terraform_demo	CreateVM(Azure)	Location	ITA-demo-Azure
Terraform_demo	CreateVM(Azure)	security_group	ita-demo-security-group
Terraform_demo	CreateVM(Azure)	Vnet_name	ita-demo-vnet
Terraform_demo	CreateVM(Azure)	Vnet_address_space	10.0.0.0/16
Terraform_demo	CreateVM(Azure)	subnet_name	ita-demo-subnet
Terraform_demo	CreateVM(Azure)	address_prefixes	10.0.2.0/24
Terraform_demo	CreateVM(Azure)	public_ip_name	ita-demo-public-ip
Terraform_demo	CreateVM(Azure)	allocation_method	Dynamic
Terraform_demo	CreateVM(Azure)	domain_name_label	ita-demo-domain
Terraform_demo	CreateVM(Azure)	network_interface_name	ita-demo-nwif

## 4.2 Setting variable values(4/4)

### Configure values to Variables(3/3)

Please refer to the table below and register substitute values.

Operation	Movement	Variable name	Specific value
Terraform_demo	Create VM (Azure)	NIC_name	ita-demo-NIC
Terraform_demo	Create VM (Azure)	VM_name	ITA-demo-Azure
Terraform_demo	Create VM (Azure)	publisher	OpenLogic
Terraform_demo	Create VM (Azure)	offer offer	centOS
Terraform_demo	Create VM (Azure)	sku	8_2
Terraform_demo	Create VM (Azure)	source_image_version	Latest
Terraform_demo	Create VM (Azure)	os_disk_name	ita-demo-os-disk
Terraform_demo	Create VM (Azure)	storage_account_type	Standard_LRS
Terraform_demo	Create VM (Azure)	caching	ReadWrite
Terraform_demo	Create VM (Azure)	admin_username	ita-demo
Terraform_demo	Create VM (Azure)	ssh_public_key	(SSHpublic key)
Terraform_demo	Create VM (Azure)	VM_size	Standard_B1ls
Terraform_demo	Create VM (Azure)	VM_count	3

## 4.3 Execution

### Execute Movement

We have now finished creating the Movements and registering the Substitute values we are going to use.

Lastly, we will execute the Movement and check the result in the target host.

Menu: **Terraform> Execution**

The screenshot shows the Terraform Execution interface. The left sidebar contains a 'Menu' with options like 'Main menu', 'Interface information', 'Organizations list', 'Workspaces list', 'Movement list', 'Module files', 'Policies list', 'Policy Sets list', 'PolicySet-Policy link list', 'PolicySet-Workspace link list', 'Movement module link', 'Substitution value auto-registration setting', and 'Substitution value list'. The main area is divided into sections: 'Description', 'Scheduling', 'Movement [Filter]', 'Movement [List]', 'Operation [Filter]', 'Operation [List]', and 'Execution'. The 'Movement [List]' table has two rows: '21 createVM(AWS)' and '22 createVM(Azure)'. The 'Operation [List]' table has one row: '25 Terraform\_demo'. The 'Execution' section shows 'Movement ID 21' and 'Movement Name CreateVM(AWS)' with 'Plan check' and 'Execute' buttons. Numbered callouts indicate: 1. Select the Movement you want to Execute. 2. Select an operation. 3. Click [Execute]. A 'Tips' box states: 'After execution, the user will automatically be moved to the "check operation status" screen. "Plan Check" runs the operation and stops after the Plan/PolicyCheck'.

**1** Select the Movement you want to Execute.

**2** Select an operation.

**3** Click [Execute].

**Tips** After execution, the user will automatically be moved to the "check operation status" screen. "Plan Check" runs the operation and stops after the Plan/PolicyCheck

## 4.4 Checking Operation status

### Check the detailed results of the Movement

After executing, the user will be moved to a screen where they can see the Execution status and Execution logs.

It is also possible to see the input and output data.

The screenshot displays the 'Check operation status' interface. On the left is a sidebar menu with options like 'Main menu', 'Interface information', 'Organizations list', 'Workspaces list', 'Movement list', 'Module files', 'Policies list', 'Policy Sets list', 'PolicySet-Policy link list', 'PolicySet-Workspace link list', 'Movement module link', 'Substitution value auto-registration setting', 'Substitution value list', and 'Execution'. The main area shows a table with columns 'Item' and 'Value'. The table contains details about the execution, including 'Execution No.', 'Execution type', 'Status', 'Caller Symphony', 'Caller Conductor', 'Executing user', 'Movement', 'Operation', 'Variable', 'Input data', 'Output data', 'Execution status', 'Start date/time', and 'End date/time'. A red box highlights the 'Input data' and 'Output data' rows, which show 'Populated data' and 'Result data' respectively. A 'Confirm' button is visible next to the 'Output data' row. To the right, a 'Progress status(Plan log)' window is open, showing a JSON plan for creating a VM. The plan includes details like 'device\_index', 'network\_interface\_id', 'root\_block\_device', 'delete\_on\_termination', 'device\_name', 'encrypted', 'sops', 'key\_id', 'tags', 'throughput', 'volume\_id', 'volume\_size', and 'volume\_type'. The plan is for adding 3 VMs.

Item	Value
Execution No.	2
Execution type	Normal
Status	Completed (error)
Caller Symphony	
Caller Conductor	
Executing user	System Administrator
Movement	
ID	21
Name	CreateVM(AWS)
Delay timer (minutes)	
Terraform Dedicated information	
Organization:Workspace	LearnTest1:ITA-demo-AWS
RUN-ID	run-FTrUjddQAbnKscj8e
Operation	
No.	25
Name	Terraform_demo
ID	25
Variable	
Input data	Populated data
Output data	Result data
Scheduled date/time	
Start date/time	2020/08/20 15:14:26
End date/time	2020/08/20 15:14:40

Progress status(Plan log)

Filter:  ☐ Display only corresponding lines

```

+ device_index = (known after apply)
+ network_interface_id = (known after apply)
}
+ root_block_device {
+   delete_on_termination = (known after apply)
+   device_name = (known after apply)
+   encrypted = (known after apply)
+   sops = (known after apply)
+   key_id = (known after apply)
+   tags = (known after apply)
+   throughput = (known after apply)
+   volume_id = (known after apply)
+   volume_size = (known after apply)
+   volume_type = (known after apply)
}
}
Plan: 3 to add, 0 to change, 0 to destroy.

```

#### Tips

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

#### Tips

For the results, please access the Azure account and check that all of the 3 VM has been created.

## 4.5 Change the value and execute again(1/2)

### Change the number of instances and execute again.

Finally, change the number of instances to deploy and repeat.

From **Terraform> Substitution Value Management**, refer to the table below and change the specific values and do the same as we did in chapter 4.3

#### Before

Operation	Movement	Variable name	Specific value
Terraform_demo	CreateVM(AWS)	hello_tf_instance_count	3
Terraform_demo	CreateVM(Azure)	VM_count	3



Change the Deploy instances:  
AWS: From 3 to 5  
Azure: From 3 to 1

#### After

Operation	Movement	Variable name	Specific value
Terraform_demo	CreateVM(AWS)	hello_tf_instance_count	5
Terraform_demo	CreateVM(Azure)	VM_count	1

**Tips**



## 4.5 Change the value and execute again(2/2)

Check that the instances has been reduced.

Access AWS ・ Azure from your browser  
and check that the VM instances has been reduced.

### AWS

<input type="checkbox"/>	Name	インスタンス ID	イン
<input type="checkbox"/>	ita-demo-instance-1	i-01ce909628624f4fc	✓
<input type="checkbox"/>	ita-demo-instance-3	i-0a9937555aca7be96	✓
<input type="checkbox"/>	ita-demo-instance-2	i-0d34ba73c42144d54	✓



<input type="checkbox"/>	Name	インスタンス ID	イン
<input type="checkbox"/>	ita-demo-instance-1	i-01ce909628624f4fc	✓
<input type="checkbox"/>	ita-demo-instance-3	i-0a9937555aca7be96	✓
<input type="checkbox"/>	ita-demo-instance-2	i-0d34ba73c42144d54	✓
<input type="checkbox"/>	ita-demo-instance-5	i-004f22d98cf7f6303	✓
<input type="checkbox"/>	ita-demo-instance-4	i-0f42e3212f538c8d2	✓

### Azure

<input type="checkbox"/>	名前 ↑↓
<input type="checkbox"/>	ita-demo-nwif-0
<input type="checkbox"/>	ita-demo-nwif-1
<input type="checkbox"/>	ita-demo-nwif-2
<input type="checkbox"/>	ita-demo-os-disk-0
<input type="checkbox"/>	ita-demo-os-disk-1
<input type="checkbox"/>	ita-demo-os-disk-2
<input type="checkbox"/>	ita-demo-public-ip-0
<input type="checkbox"/>	ita-demo-public-ip-1
<input type="checkbox"/>	ita-demo-public-ip-2
<input type="checkbox"/>	ita-demo-security-group
<input type="checkbox"/>	ita-demo-vnet
<input type="checkbox"/>	ita-demo-web-azure-0
<input type="checkbox"/>	ita-demo-web-azure-1
<input type="checkbox"/>	ita-demo-web-azure-2



<input type="checkbox"/>	名前 ↑↓
<input type="checkbox"/>	ita-demo-nwif-0
<input type="checkbox"/>	ita-demo-os-disk-0
<input type="checkbox"/>	ita-demo-public-ip-0
<input type="checkbox"/>	ita-demo-security-group
<input type="checkbox"/>	ita-demo-vnet
<input type="checkbox"/>	ita-demo-web-azure-0



**Exastro**