IT Automation
Terraform Driver
【 Practice 】

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

Exastro IT Automation ver 1.10

Exastro developer

Table of contents

1.Introduction
- 1.1 Introduction
- 1.2 Environment

2. Terraform Driver Practice
- 2.1 Scenario
- 2.2 Preparation

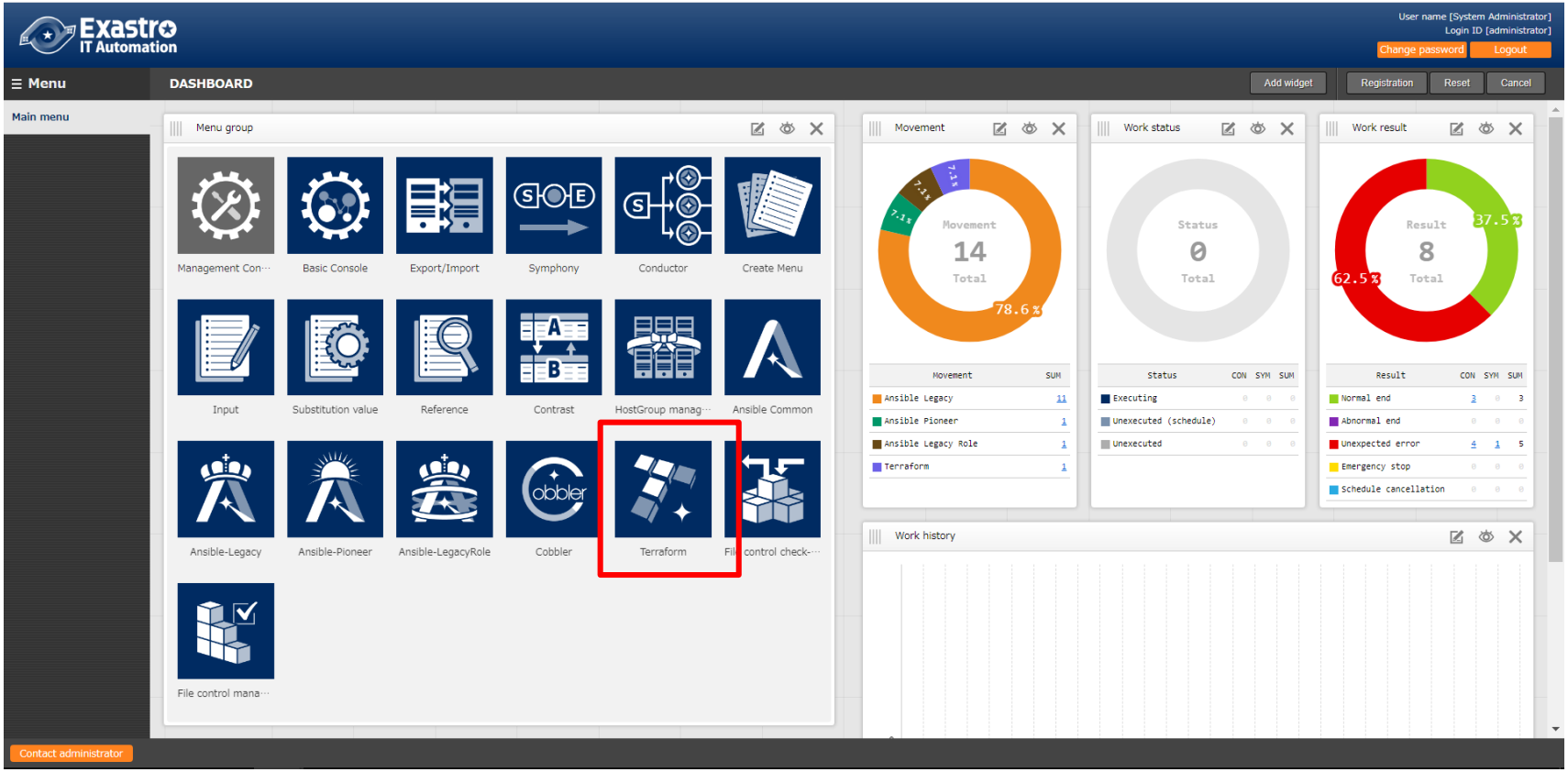
3. Preparation
- 3.1 Interface Information Registration
- 3.2 Registering and linking Organization
- 3.3 Registering and linking Workspace
- 3.4 Operation pattern (Movement) Registration
- 3.5 Module file Registration
- 3.6 Policy file Registration
- 3.7 Policy Set Registration
- 3.8 Linking Policy Set and Policy file
- 3.9 Linking Policy Set and Workspace
- 3.10 Specifying Module file in Movement

4. Execution
- 4.1 Operation Registration
- 4.2 Setting variable values
- 4.3 Confirm Plan
- 4.4 Confirm PolicyCheck log
- 4.5 Change and check VM size
- 4.6 Re-confirm PolicyCheck log
- 4.7 Execution
- 4.8 Checking Operation status
- 4.9 Change and re-run values

1. Introduction

Main Menu

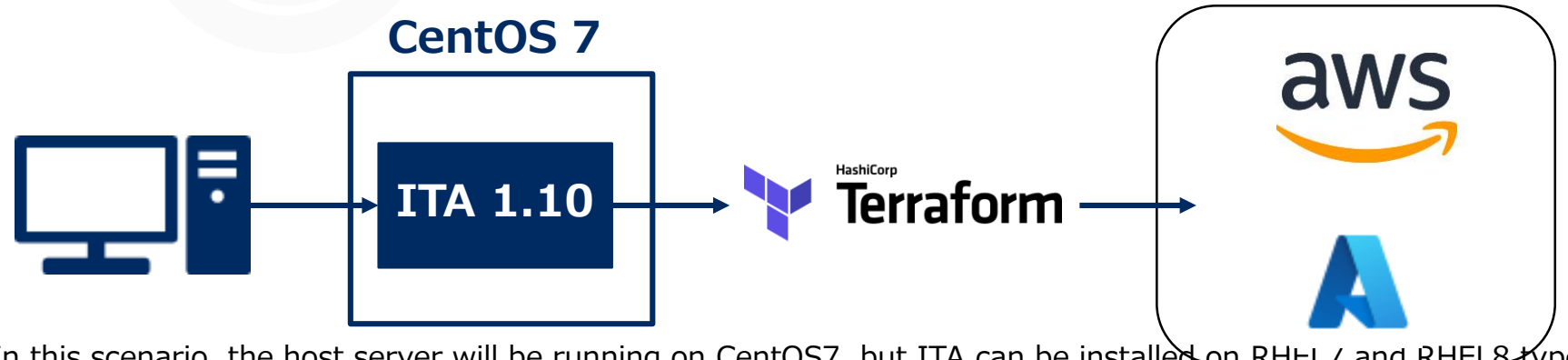
- This document aims to teach the readers about the **Terraform** Menu group while giving them hands-on experience.



Environment

- The environment used in this manual is as follows.
- In addition to an ITA Server, please prepare an AWS and Azure account and a Terraform environment (Terraform account if you are using Terraform Cloud).

ITA host server	Terraform	Target
• CentOS7(※) • ITA ver 1.10	• Terraform Enterprise or • Terraform Cloud	• AWS • Microsoft Azure



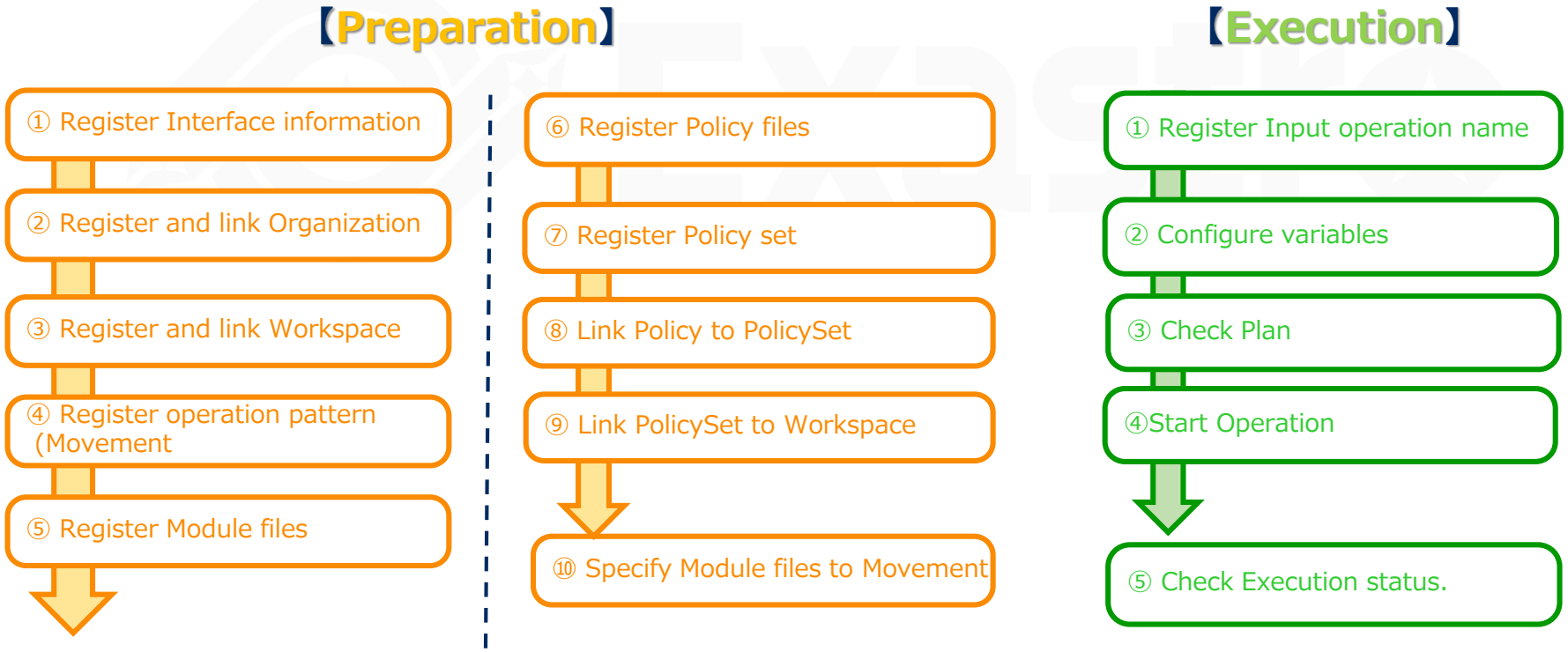
*In this scenario, the host server will be running on CentOS7, but ITA can be installed on RHEL7 and RHEL8 type OS as well.

2. Terraform Driver Practice

About the scenario

This scenario uses ITA's Terraform Driver to check the plan to create the VM's on Public cloud (AWS,Azure)
After that, it will use the **defined policies** to create the VM on the different cloud environments.

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)**



Create Module①

- 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 "aws_info"{
  type = object({
    access_key = string
    secret_key = string
    region = string
  })
}
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

Create Module②

```
provider "aws" {
  access_key = var. aws_info.access_key
  secret_key = var. aws_info.secret_key
  region = var. aws_info.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.

Create Module③

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

```
variable "azure_info"{
  type = object({
    subscription_id = string
    tenant_id = string
    client_id = string
    client_secret = string
  }
)}
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" {}
```

Create Module④

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. azure_info.subscription_id
  client_id       = var. azure_info.client_id
  client_secret   = var. azure_info.client_secret
  tenant_id       = var. azure_info.tenant_id
}

resource "azurerm_resource_group" "hogehoge" {
  name = var.resource_group_name
  location = var.location
}
```

```
resource "azurerm_network_security_group" "hogehoge" {
   name =var.security_group
   location = azurerm_resource_group.hogehoge.location
   resource_group_name = azurerm_resource_group.hogehoge.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 = "*"
   }
}
```

Create Module④

File name:
azure_create_instance.tf (2/3)

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

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

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


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

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

Create Module④

File name:
azure_create_instance.tf (3/3)

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

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

Create Policy

File name : limit-proposed-monthly-cost.sentinel

This policy limits the monthly cost.

The Terraform will not apply if the monthly cost exceeds 50$.

It will also output an estimate of the monthly cost.

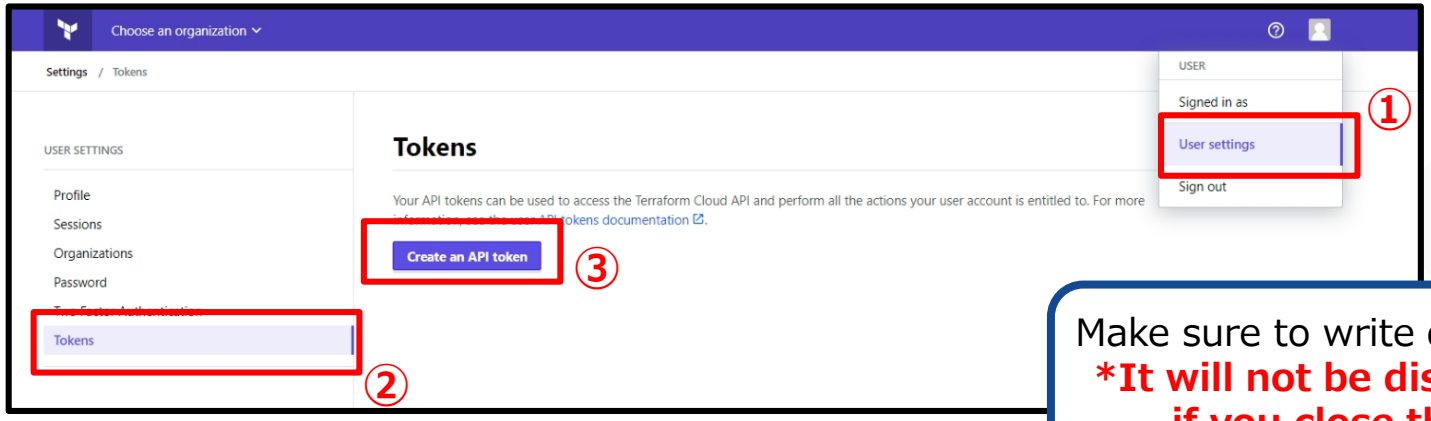
This can be used for both AWS and Azure.

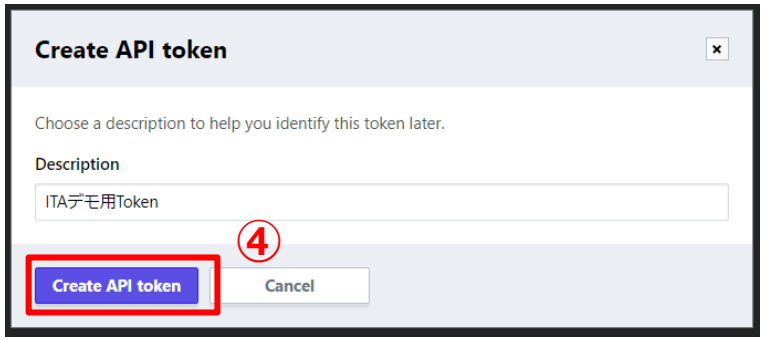
```
import "tfrun"
import "decimal"

limit = decimal.new(50)

cost_limit_by_workspace = func() {
  if tfrun.cost_estimate else null is null {
    print("no cost estimates available")
    return false
    }

  workspace_name = tfrun.workspace.name

  proposed_cost = decimal.new(tfrun.cost_estimate.proposed_monthly_cost)

  if proposed_cost.less_than(limit) {
    print("Proposed monthly cost", proposed_cost.string,
      "of workspace", workspace_name,
      "is under the limit: $", limit)
    return true
  }

  if proposed_cost.greater_than(limit) {
    print("Proposed monthly cost", proposed_cost.string,
       "of workspace", workspace_name,
       "is over the limit: $", limit)
    return false
    }
}

cost_validated = cost_limit_by_workspace()

main = rule {
  cost_validated
}
```

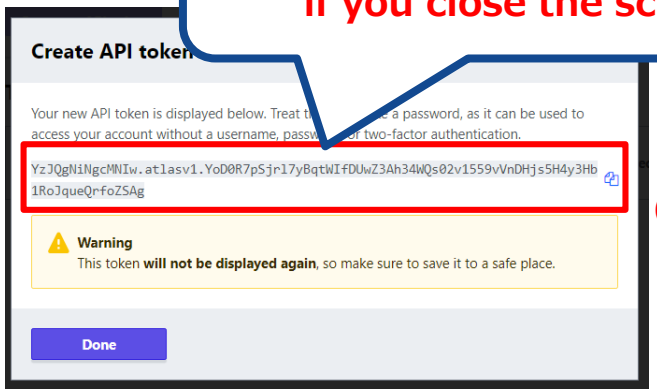
3. Preparation

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

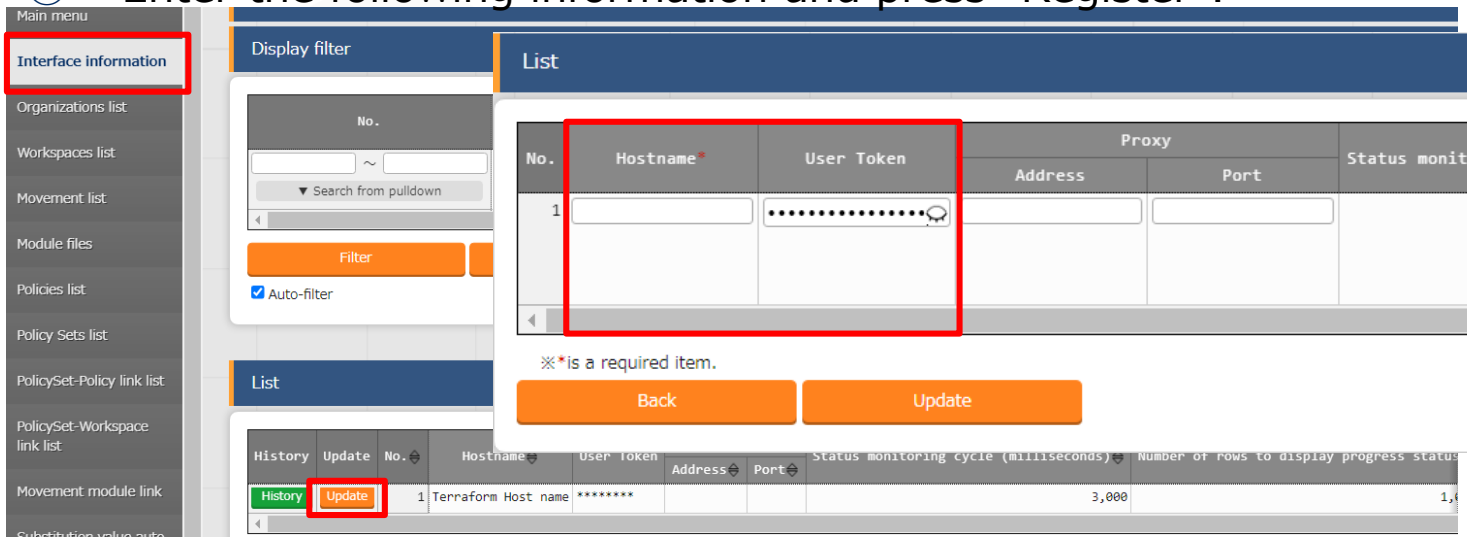
- Input the Terraform Hostname and the created UserToken

***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".



Hostname	User Token
app.terraform.io	(Input User Token)

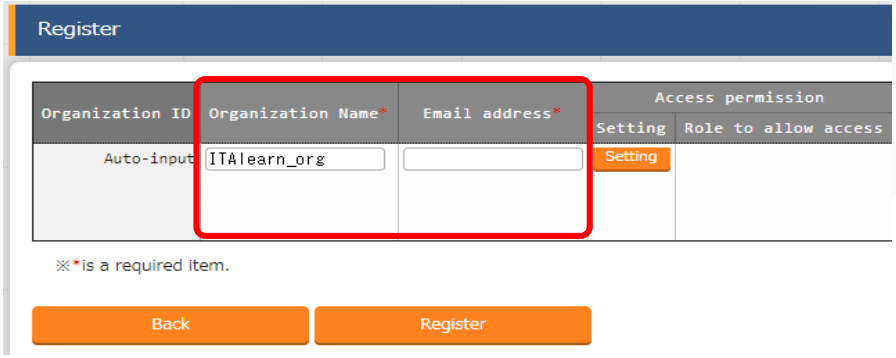
Register Organization

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 Name	Email address
ITAlearn_org	(Input Mail Address)

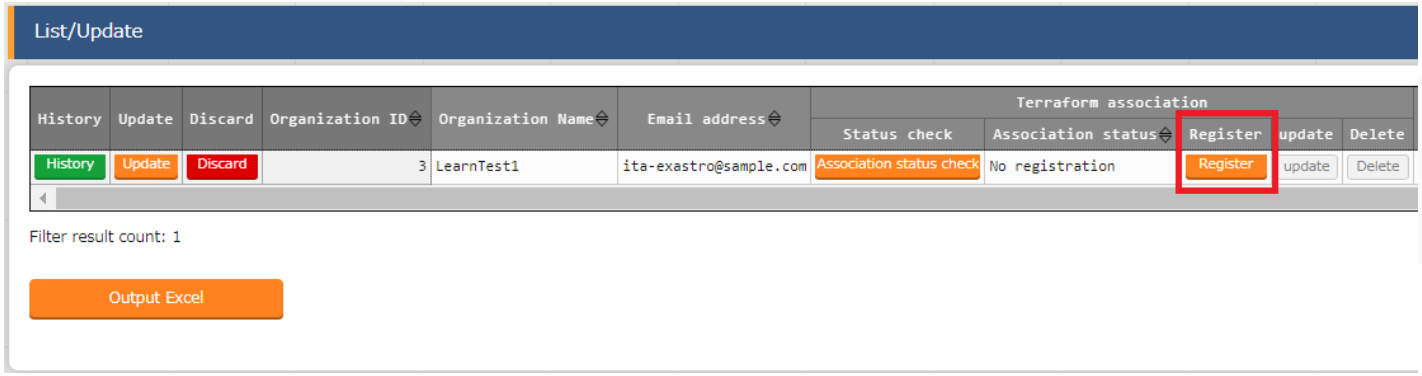
3.2 Register and Link Organization(2/2)

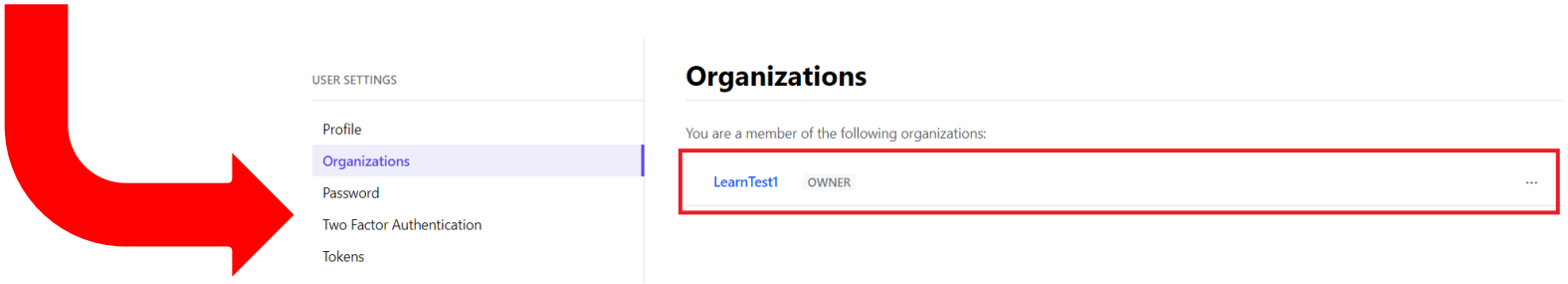
Link Organization

- 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.





3.3 Register and Link Workspace(1/2)

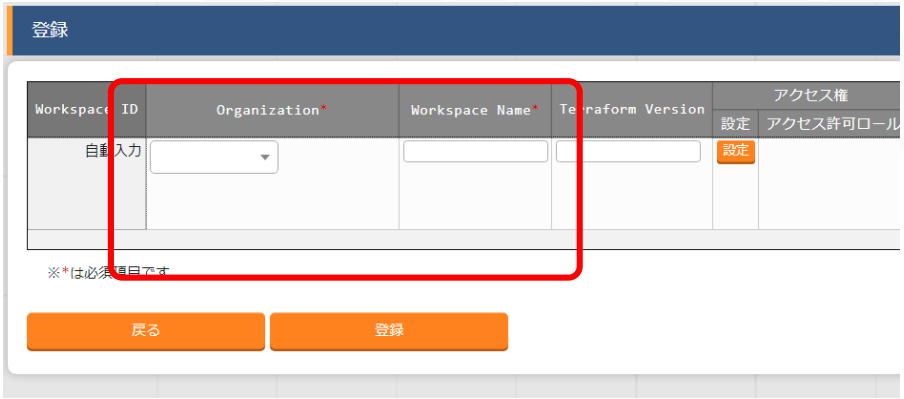
Register Workspace

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".



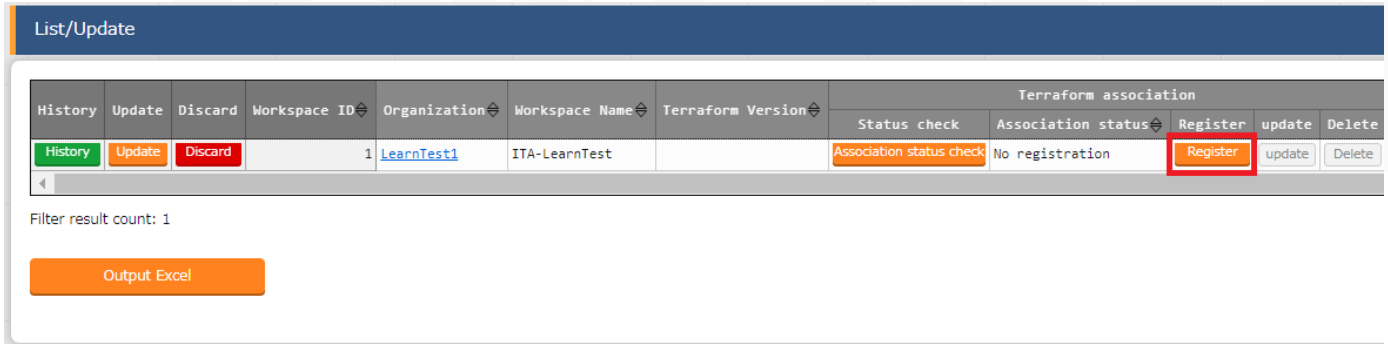
Organization	Workspace Name
ITAlearn_org	ITA-demo-AWS
ITAlearn_org	ITA-demo-Azure

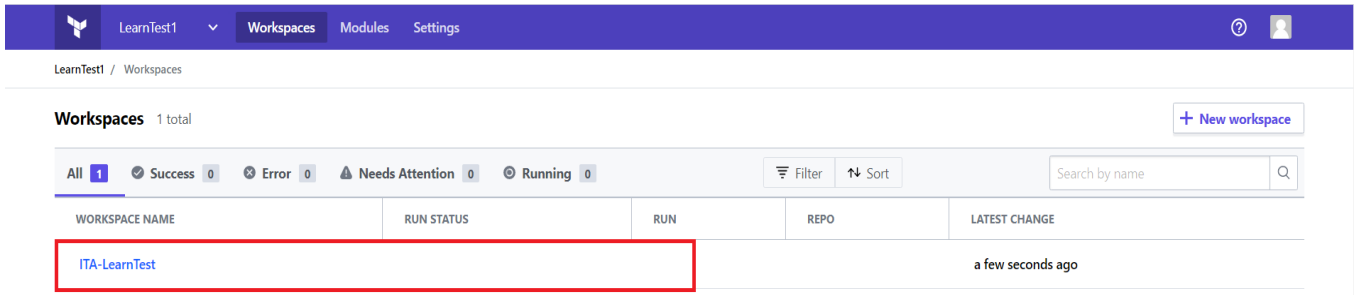
3.3 Register and Link Workspace(2/2)

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.**





3.4 Register Operation pattern(Movement)

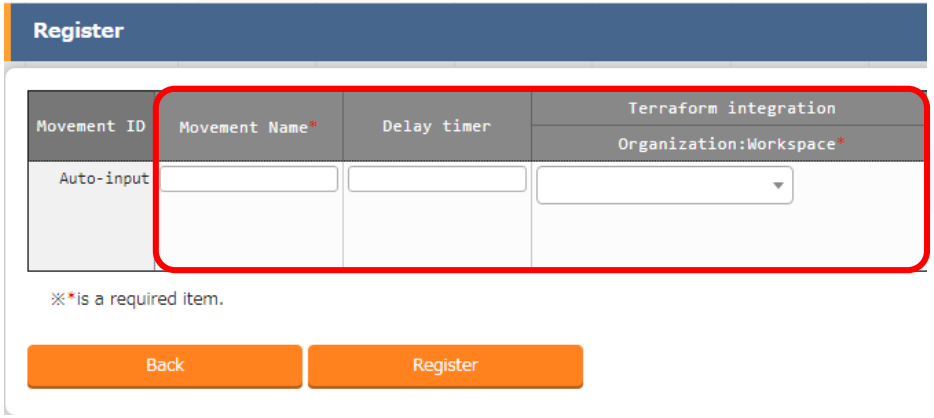
Create Movement

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

Register Module

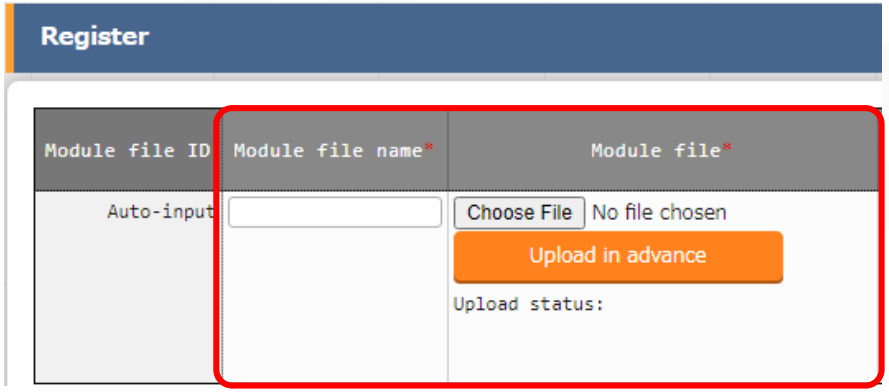
In this section, we will register our Modules to ITA.

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 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

Register Policy file

In this section, we will register the policy file we created.

Menu : **Terraform > Policies list**

① Click Register> Start Registration.

② Select the policy you want to upload and press “Upload in advance”.

③ Follow the table below and press “Register”

Policy name	Policy file
limit-proposed-monthly-cost	limit-proposed-monthly-cost.sentinel

Register Policy Set

In this section, we will register a Policy set.

Menu : **Terraform > Policy Sets list**

① Click Register> Start Registration.

② Follow the table below and press “Register”

PolicySet name
PolicySet_demo

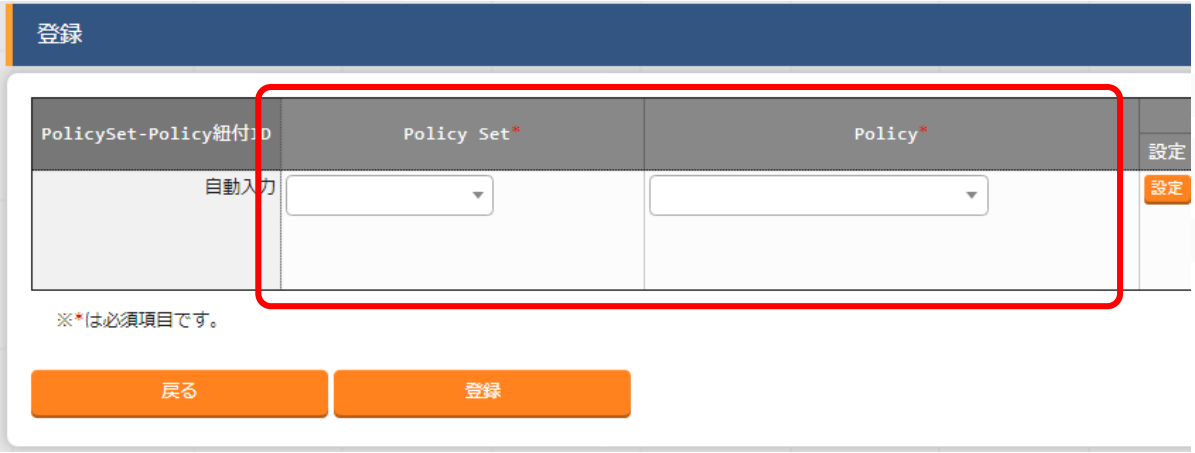
Link Policy Set and Policy

In this section, we will link the previously created Policy Set and Policy file.

Menu : **Terraform > PolicySet-Policy link list**

① Click Register> Start Registration.

② Follow the table below and press “Register”



Policy Set	Policy
1:PolicySet_demo	1:limit-proposed-monthly-cost

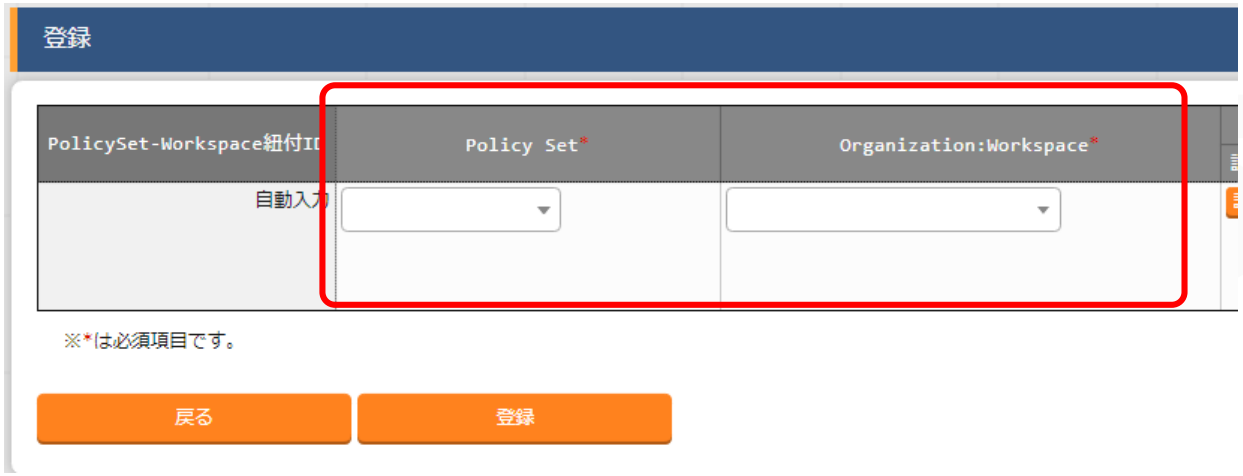
Link Policy Set and Workspace

In this section, we will link the Policy Set and the Workspace.

Menu : **Terraform > PolicySet-Workspace link list**

① Click Register> Start Registration.

② Follow the table below and press “Register”



Policy Set	Organization:Workspace
1:PolicySet_demo	ITAlearn_org:ITA-demo-AWS
1:PolicySet_demo	ITAlearn_org:ITA-demo-Azure

3.10 Specify Module file to Movement

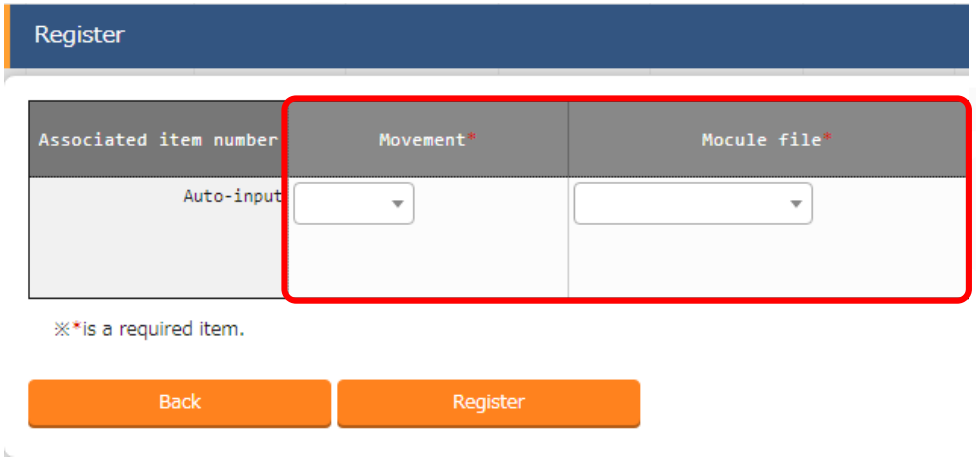
Link Module to Movement

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".



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

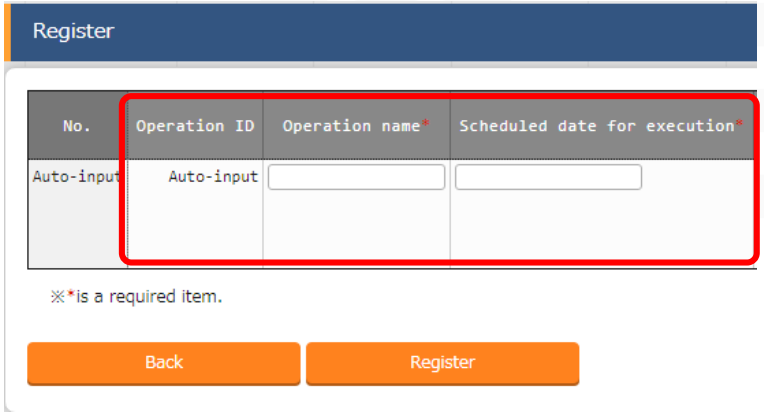
Register new Operation

In this section, we will create an Operation

Menu: **Basic Console > Input Operation List**

① Click Register> Start Registration.

② Enter the following information and press "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)

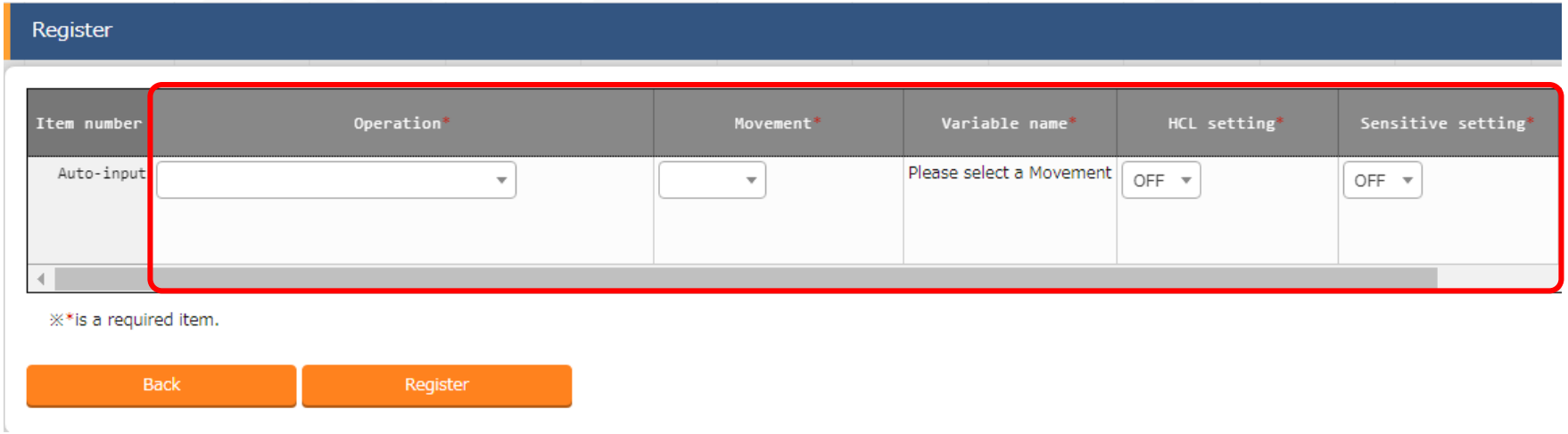
Configure values to Variables.

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".



※The upcoming slides will explain Specific setting values.

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.large
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	(Input desired Azure location)
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	(Global domain name)

Configure values to Variables(3/3)

Please refer to the table below and register substitute values.

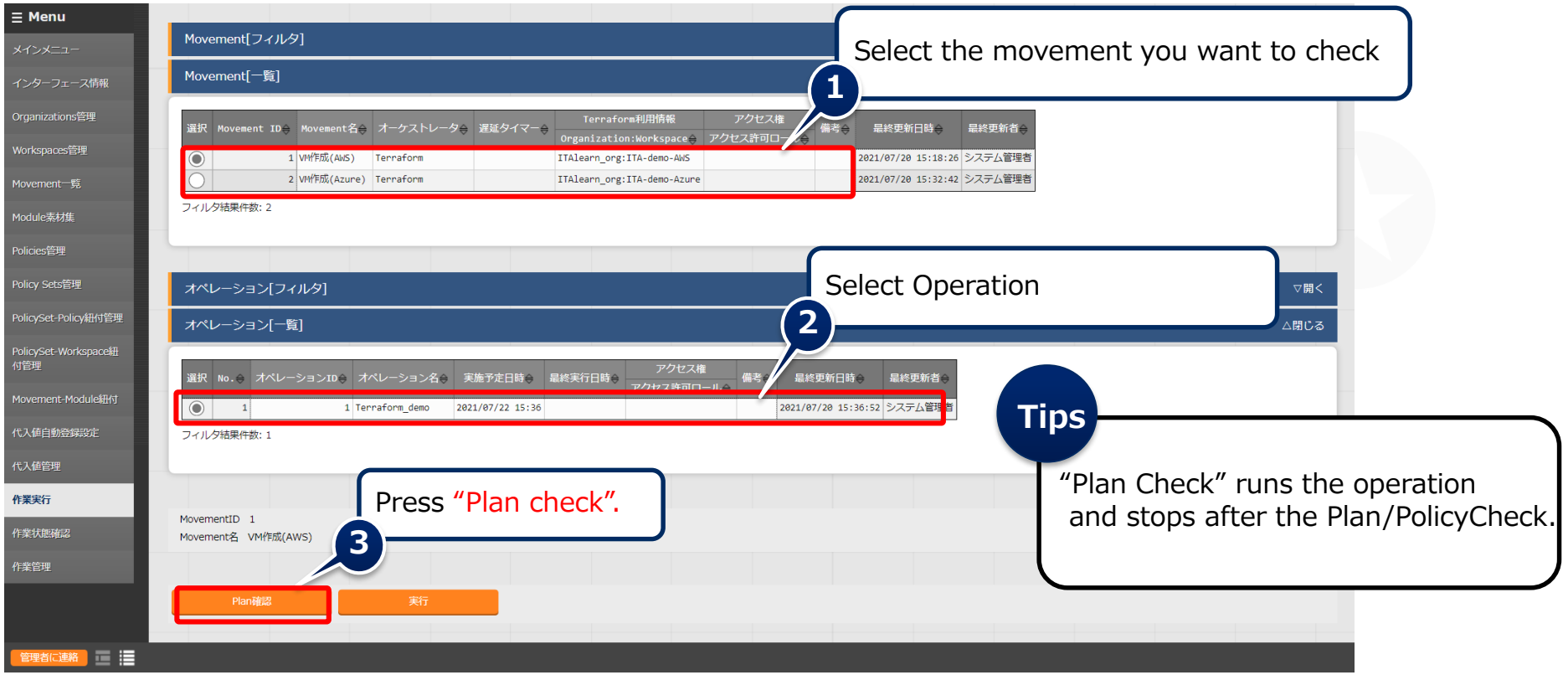
Operation	Movement	Variable name	Specific value
Terraform_demo	CreateVM(Azure)	network_interface_name	ita-demo-nwif
Terraform_demo	CreateVM(Azure)	NIC_name	ita-demo-NIC
Terraform_demo	CreateVM(Azure)	VM_name	ita-demo-web-azure
Terraform_demo	CreateVM(Azure)	publisher	OpenLogic
Terraform_demo	CreateVM(Azure)	offer	CentOS
Terraform_demo	CreateVM(Azure)	sku	8_2
Terraform_demo	CreateVM(Azure)	source_image_version	latest
Terraform_demo	CreateVM(Azure)	os_disk_name	ita-demo-os-disk
Terraform_demo	CreateVM(Azure)	storage_account_type	Standard_LRS
Terraform_demo	CreateVM(Azure)	caching	ReadWrite
Terraform_demo	CreateVM(Azure)	admin_username	ita-demo
Terraform_demo	CreateVM(Azure)	ssh_public_key	(Public SSH key)※
Terraform_demo	CreateVM(Azure)	VM_size	Standard_B2MS
Terraform_demo	CreateVM(Azure)	VM_count	3

※Use the SSH key that you prepared. The Specific value is the text of the key, ”ssh-rsa xxxxxxxx～”

Check Plan

In the previous section, we have created the Movement and registered the substitute values.
In the next section, we are going to check that the module follows the policy.

Menu : **Terraform > Execution**



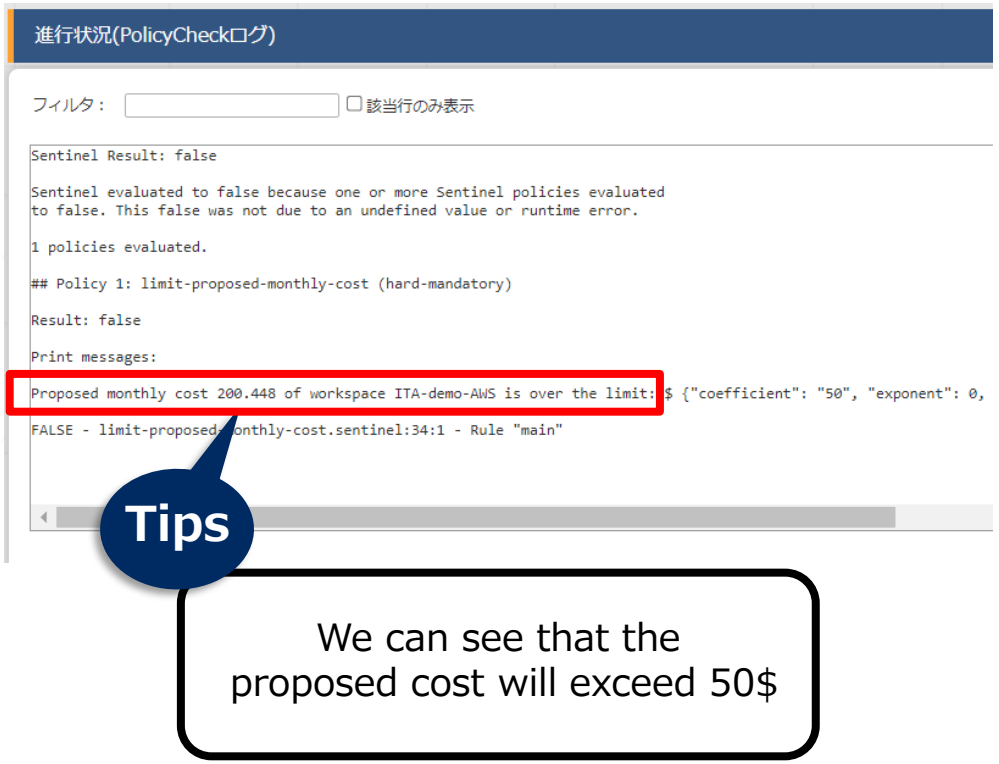
Check PolicyCheck log

Checking the PolicyCheck log will move the user to the screen below and tell that an error has occurred. Scroll down to see the PolicyCheck log.

説明

対象作業

項目			値
作業No.			61
実行種別			Plan確認
ステータス			完了(異常)
呼出元Symphony			
呼出元Conductor			
実行ユーザ			システム管理者
Movement	ID		1
	名称		VM作成(AWS)
	遅延タイマ(分)		
	Terraform利用情報	Organization:Workspace	ITAlearn_org:ITA-demo-AWS
		RUN-ID	run-GJFbJfkfFREKUwDb
オペレーション	No.		1
	名称		Terraform_demo
	ID		1
代入値			確認
入力データ	投入データ		InputData_0000000061.zip
出力データ	結果データ		ResultData_0000000061.zip
作業状況	予約日時		
	開始日時		2021/09/06 07:51:09
	終了日時		2021/09/06 07:51:41



進行状況(PolicyCheckログ)

フィルタ： 該当行のみ表示

```
Sentinel Result: false

Sentinel evaluated to false because one or more Sentinel policies evaluated
to false. This false was not due to an undefined value or runtime error.

1 policies evaluated.

## Policy 1: limit-proposed-monthly-cost (hard-mandatory)

Result: false

Print messages:

Proposed monthly cost 200.448 of workspace ITA-demo-AWS is over the limit: $ {"coefficient": "50", "exponent": 0,

FALSE - limit-proposed-monthly-cost.sentinel:34:1 - Rule "main"
```

Tips

We can see that the proposed cost will exceed 50$

4.5 Change the VM size

Change the size of the VM and re-run the operation.

Lastly, we will change the size of the VM and run the operation again.

Go to Terraform > Substitute list and use the table below to change the values.
Then check the plan like we did in Chapter 4.3

Before

Operation	Movement	Variable name	Specific value
Terraform_demo	CreateVM(AWS)	hello_tf_instance_type	t2.large
Terraform_demo	CreateVM(Azure)	VM_size	Standard_B2MS

Instance size
AWS:t2.large → t2.micro
Azure:Standard_B2MS → Standard_B1LS

Tips

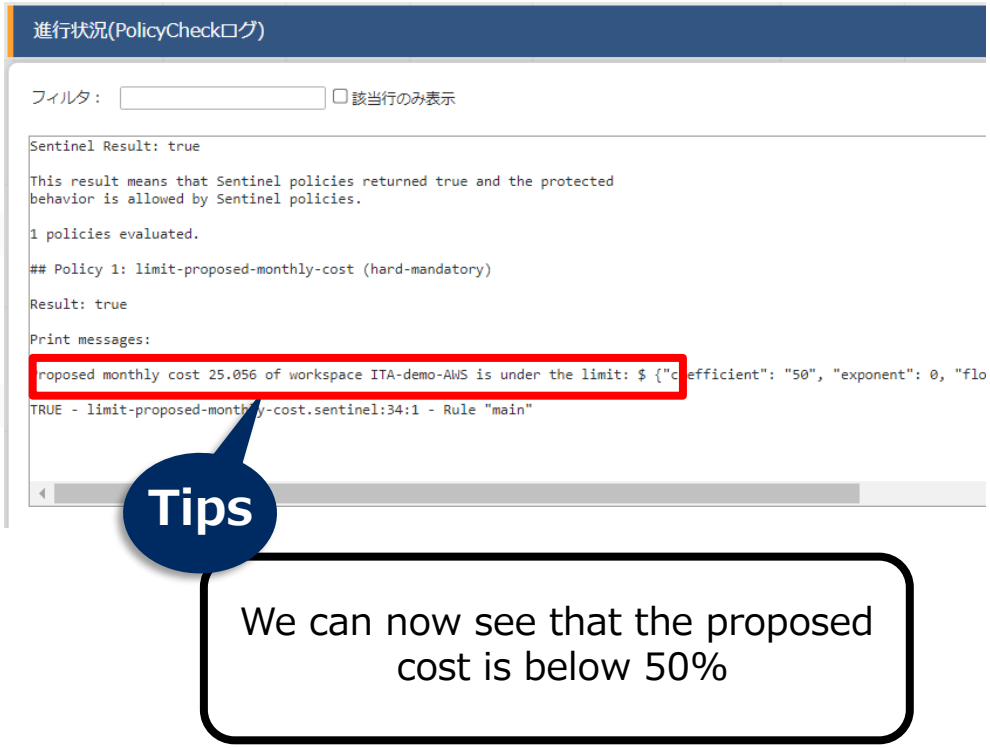
After

Operation	Movement	Variable name	Specific value
Terraform_demo	CreateVM (AWS)	hello_tf_instance_type	t2.micro
Terraform_demo	CreateVM(Azure)	VM_size	Standard_B1LS

Confirm PolicyCheck log

Now if check the PolicyCheck log, we can see that it has finished successfully.

After checking the log, we can go to the next step and run the Movement.

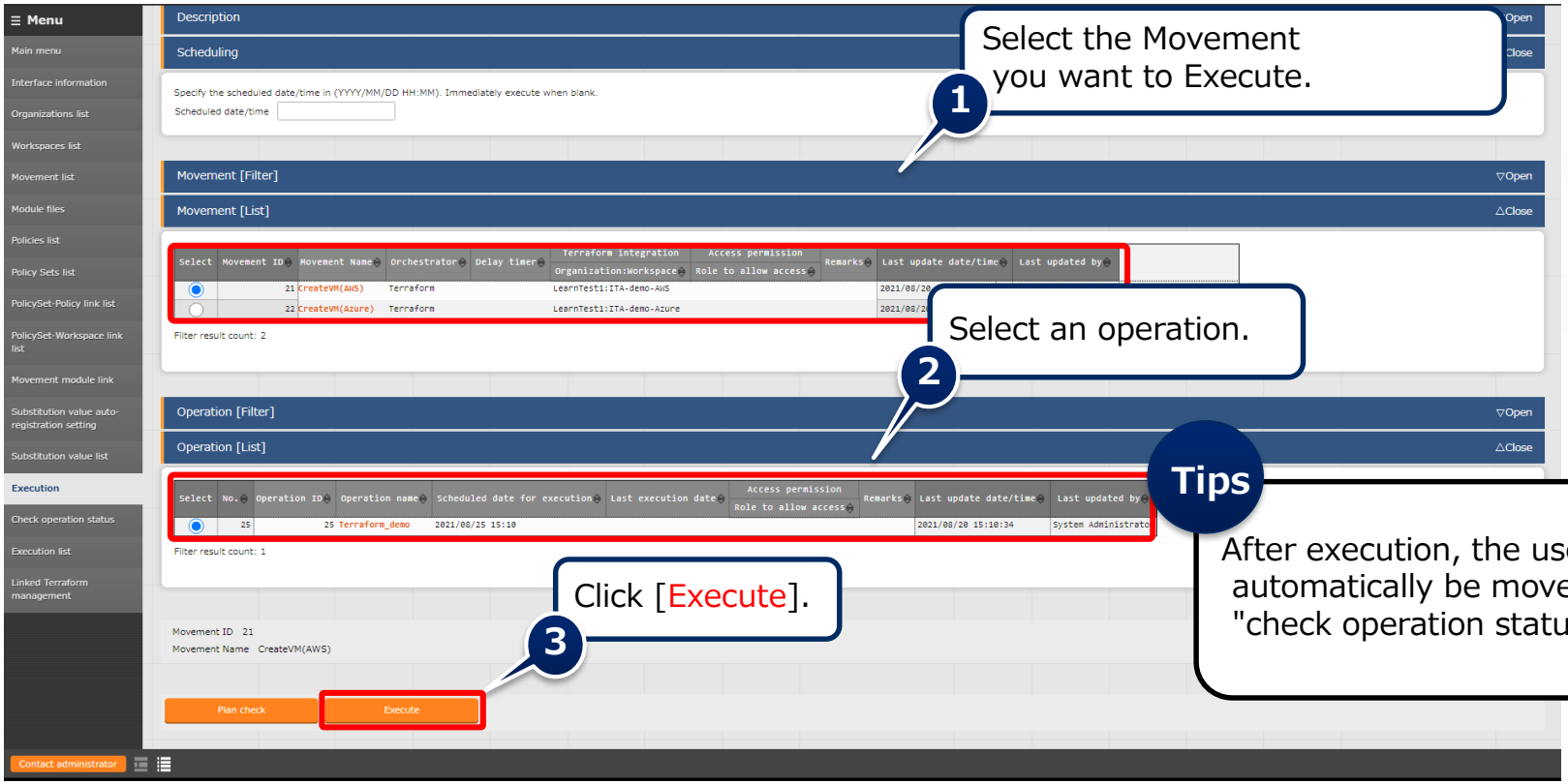


Execute Movement

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.

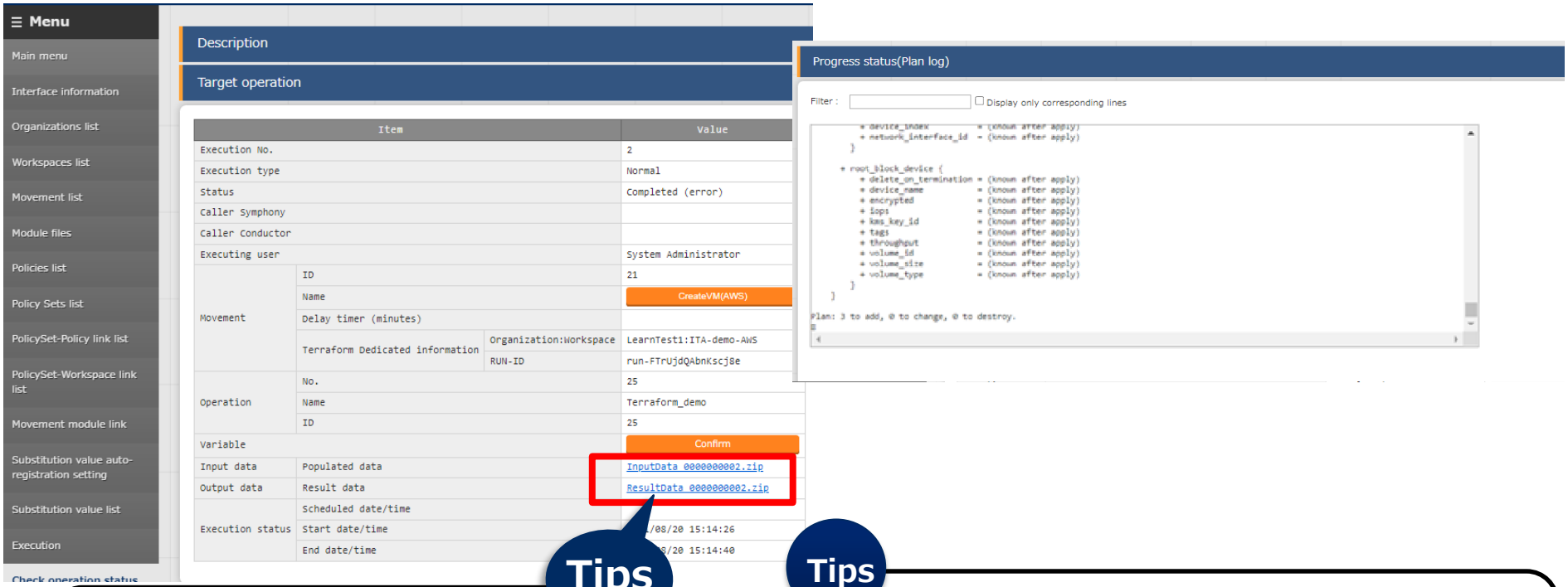
Menu**: Terraform> Execution**



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.



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.

4.9 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: Increased from 3 to 5
Azure: Decreased from 3 to 1

Tips

After

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

4.9 Change the value and execute again(2/2)

Check that the instances has been reduced.

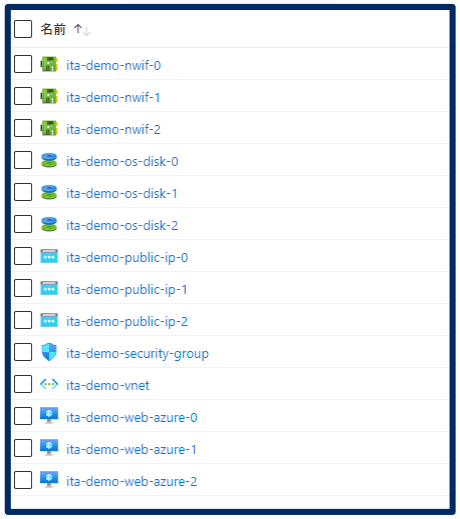
Access AWS ・Azure from your browser
and check that the VM instances has been reduced.

AWS

Name	インスタンス ID	イン
ita-demo-instance-1	i-01ce909628624f4fc	
ita-demo-instance-3	i-0a9937555aca7be96	
ita-demo-instance-2	i-0d34ba73c42144d54	

Name	インスタンス ID	イン
ita-demo-instance-1	i-01ce909628624f4fc	
ita-demo-instance-3	i-0a9937555aca7be96	
ita-demo-instance-2	i-0d34ba73c42144d54	
ita-demo-instance-5	i-004f22d98cf7f6303	
ita-demo-instance-4	i-0f42e3212f538c8d2	

Azure



Exastro