



IT Automation Terraform Driver 【実習編】

※本書では「Exastro IT Automation」を「ITA」として記載します。

Exastro IT Automation ver 1.7.2
Exastro developer

目次

1. はじめに

1.1 はじめに

1.2 作業環境

2. 実習 Terraform Driver

2.1 シナリオ

2.2 事前準備

3. 仕込み編

3.1 インターフェース情報の登録

3.2 Organizationの登録と連携

3.3 Workspaceの登録と連携

3.4 作業パターン(Movement)の登録

3.5 Module素材の登録

3.6 MovementにModule素材を指定

4. 実行編

4.1 オペレーションの登録

4.2 変数値の設定

4.3 作業実行

4.4 実行状態確認

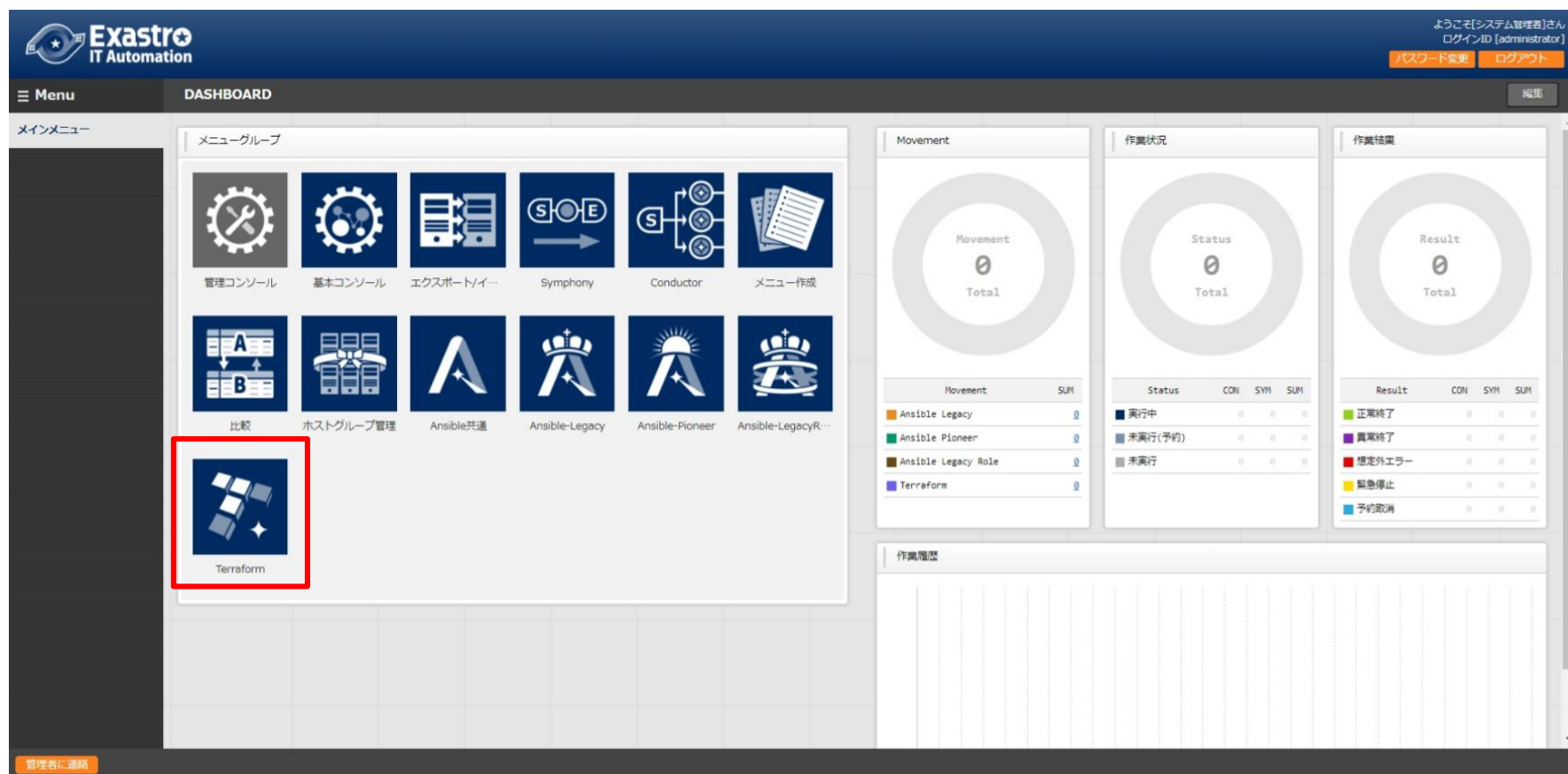
4.5 数値を変更して再度実行

1. はじめに

1.1 はじめに

メインメニュー

- 本書では、メニューグループの「**Terraform**」について、実践形式で学習いただけます。



1.2 作業環境

作業環境

- 本書で使用する作業環境は以下の通りです。
- ITAサーバとは他に、AWS、Azure、Terraform Cloudアカウントをご用意ください。

ITAホストサーバ

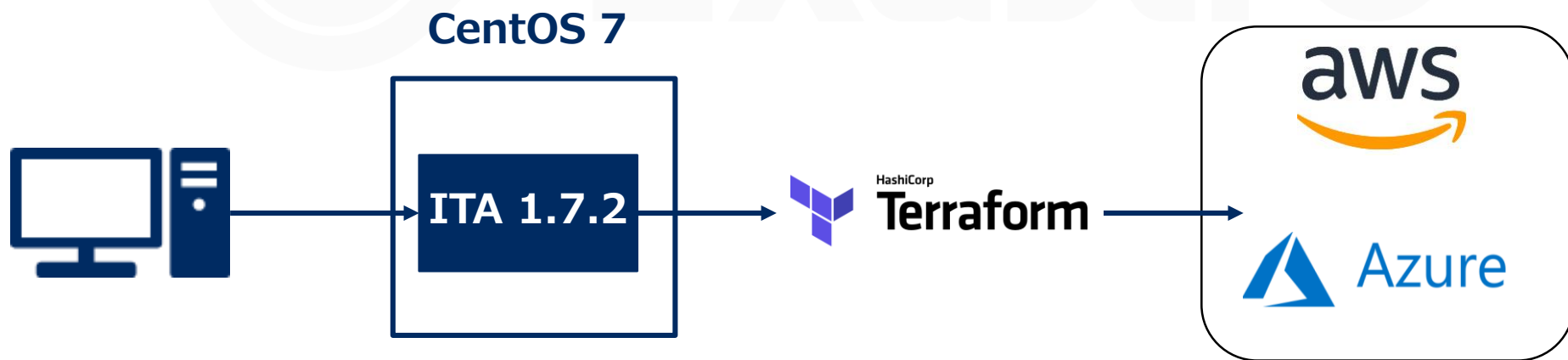
- CentOS7(※)
- ITA ver 1.7.2

Terraform

- Terraform Cloud

ターゲット

- AWS
- Microsoft Azure



※今回はホストサーバーとしてCentOS7を利用致しますが、ITAはRHEL7系およびRHEL8系のOSで導入いただけます。

2. 実習 Terraform Driver

2.1 シナリオ

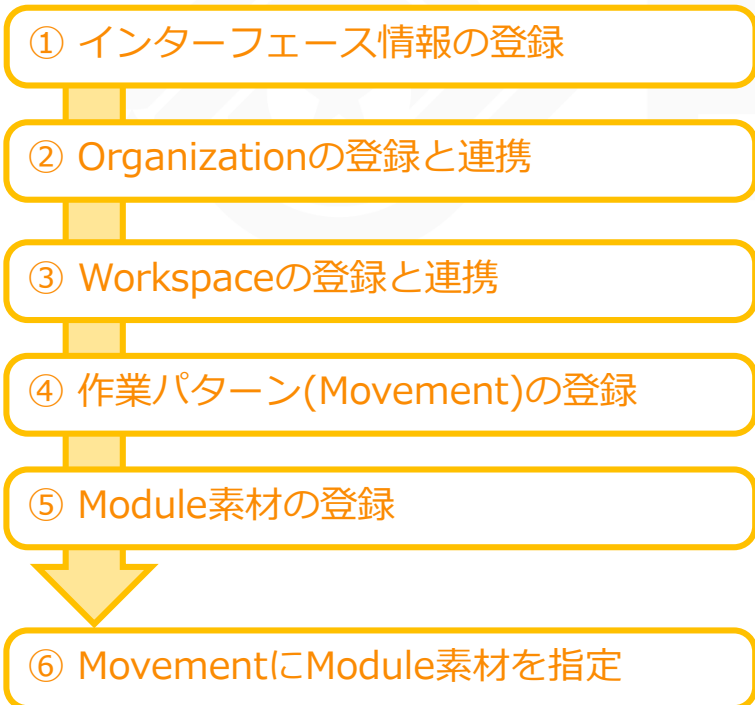
シナリオについて

本シナリオは、ITAのTerraform Driverを利用して、

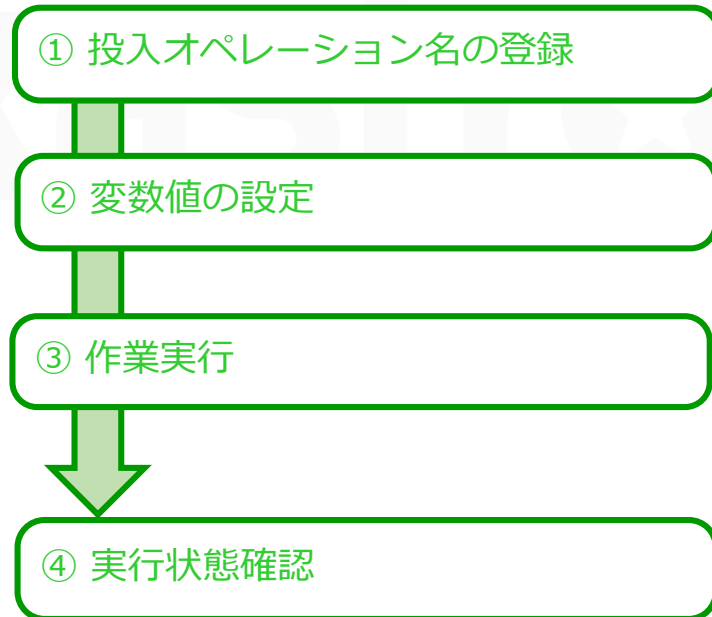
パブリッククラウド上(AWS、Azure)にVMを各3台作成する内容となっております。

下記の【仕込み編】までを一度登録・連携すると、以降の操作は【実行編】を繰り返し行うことで、対象の再設定・再登録を行うことができます。(自動化)

【仕込み編】



【実行編】



2.2 事前準備(1/6)

Moduleの作成

- 本シナリオで使用する4つのModuleを作成します。

【注意】 文字コードは"UTF-8"、改行コードは"LF"、拡張子は"tf"で作成してください。

```
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"
}
```

ファイル名 :

aws_create_instance_variables.tf

AWSインスタンス作成用の変数定義ファイルです。
変数には具体値変数が代入されます

2.2 事前準備(2/6)

IaCの作成

```
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  
}
```

ファイル名 :
aws_create_instance.tf

AWSインスタンス作成用の
リソース定義ファイルです。
セキュリティグループ、キーペアは
事前にAWSで作成・用意します。

2.2 事前準備(3/6)

IaCの作成

```
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" {}
```

ファイル名 :
azure_create_instance_variables.tf

Azureインスタンス作成用の変数定義ファイルです。
変数には具体値変数が代入されます

2.2 事前準備(4/6)

IaCの作成


ファイル名 : **azure_create_instance.tf(1/3)**

Azureインスタンス作成用のリソース
定義ファイルです。


リソースグループと、そのネットワークセキュ
リティグループ、仮想ネットワークの作成。

また、作成するVM台数分の仮想マシン本体、
ディスク、ネットワークインターフェースの
作成を行います。

```
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 事前準備(5/6)

IaCの作成

ファイル名 :
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 事前準備(6/6)

IaCの作成

ファイル名 :
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
  }

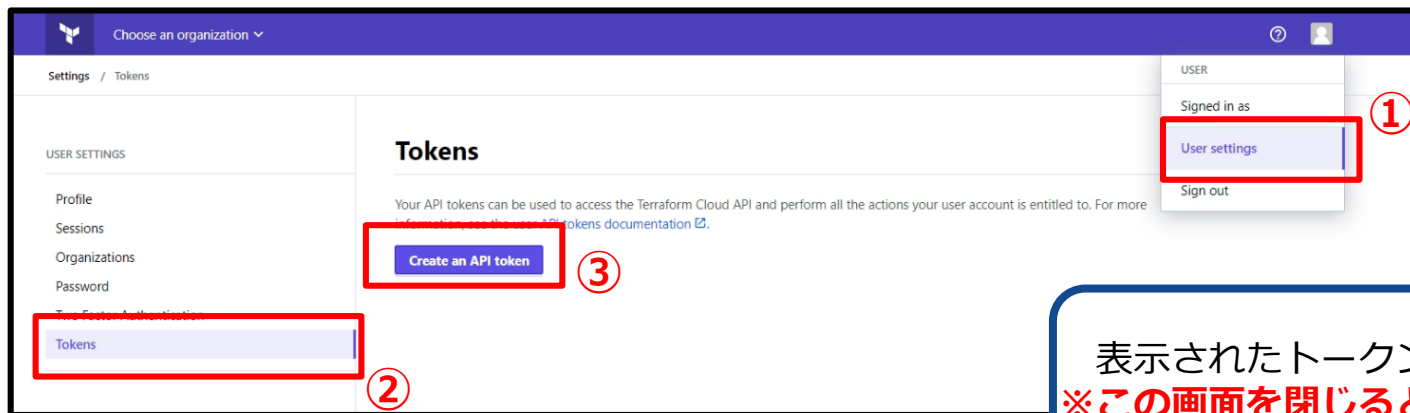
  source_image_reference {
    publisher = var.publisher
    offer     = var.offer
    sku       = var.sku
    version   = var.source_image_version
  }
}
```

3. 仕込み編

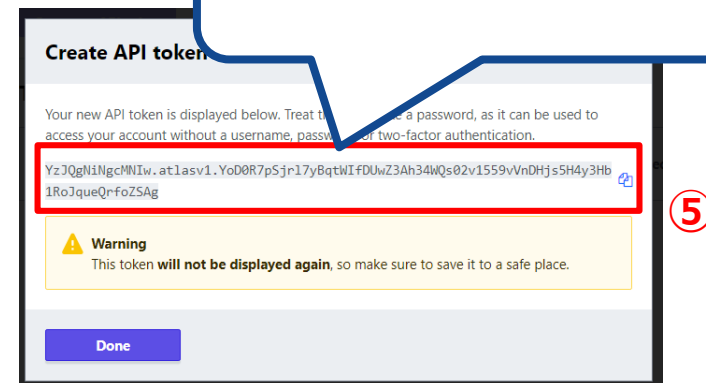
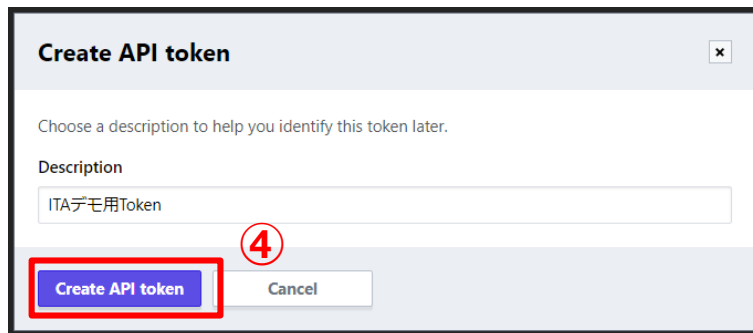
3.1 インターフェース情報の登録(1/2)

User Tokenの発行

- Terraform DriverからTerraformに連携するために、Terraformからユーザートークンを発行する必要があります。
- ブラウザよりTerraformにログインし、[User Setting]→[Tokens]→[Creat an API token]の順に押下することで発行することができます。



表示されたトークンをメモしておく
※この画面を閉じると再表示できません



3.1 インターフェース情報の登録(2/2)

■ インターフェース情報

- 連携するTerraformのHostnameと、発行したUserTokenを入力します

※ITAに連携できるTerraformは1つのみなので、インストール時に最初からある項目を「更新」して値を入力する必要があります。

メニュー：Terraform>インターフェース情報

- ① 一覧から登録済み項目の [更新] を押下する。
- ② 各項目へ下表のように入力し、[登録]を押下する。

メインメニュー

インターフェース情報

Organizations管理

Workspaces管理

Movement一覧

Module素材集

Policies管理

Policy Sets管理

PolicySet-Policy組付管理

フィルタ

オートフィルタ

一覧

No.	Hostname	User Token	Proxy		状態監視周期(単位ミリ秒)	進行状態表示行数	NULL連
			Address	Port			
1				3000	1000	無効

※は必須項目です。

戻る 更新

Hostname

app.terraform.io

User Token

(任意でご入力下さい)

3.2 Organizationの登録と連携(1/2)

Organizationを登録する

Organizationを作成しましょう。

メニュー : Terraform > Organizations管理

- ① 登録 > 登録開始 を押下する。
- ② 各項目へ下表のように入力し、[登録]を押下する。

Organization ID	Organization Name*	Email address*	アクセス権	備考
自動入力			認定 アクセス許可ロール	

※*は必須項目です。

[戻る](#) [登録](#)

Organization Name	Email address
ITAllearn_org	(任意でご入力下さい)

3.2 Organizationsの登録と連携(2/2)

Organizationを連携する

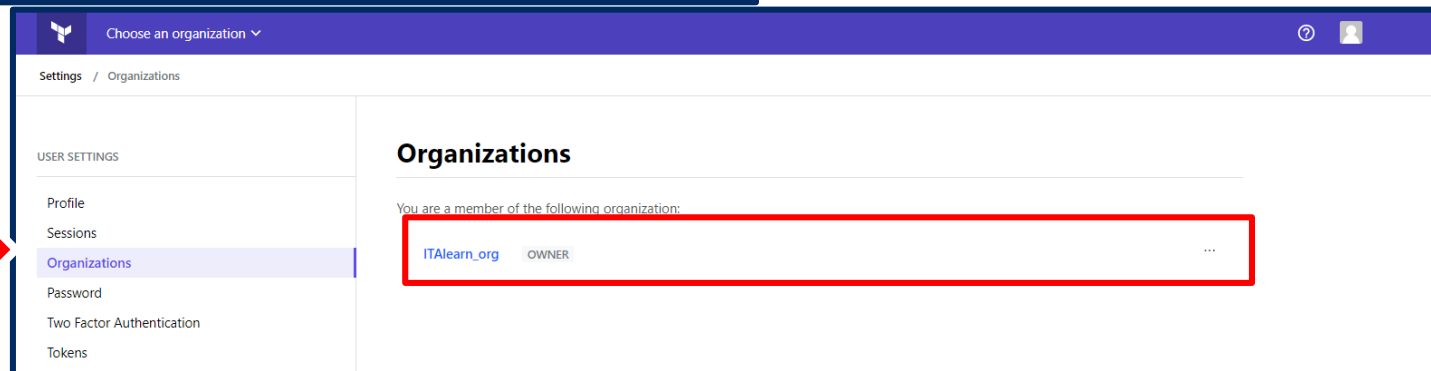
- Organization管理からOrganizationの項目を作成した後、
[連携状態チェック]で対象のTerraformに追加したOrganizationがあるかどうかをチェックすることができます。
- 「登録なし」であれば[登録]を押下することで対象のTerraformにOrganizationを作成できます。

一覧/更新

履歴	更新	廃止	Organization ID	Organization Name	Email address	Terraform連携				Wor	
						状態チェック	連携状態	登録	更新	削除	Wor
履歴	更新	廃止	1	ITAllearn_org	ita-exastro@sample.com	連携状態チェック	登録なし	登録	更新	削除	V

フィルタ結果件数: 1

Excel出力



3.3 Workspaceの登録と連携(1/2)

Workspaceを登録する

Workspaceを作成しましょう。

メニュー：Terraform > Workspaces管理

- ① 登録 > 登録開始 を押下する。
- ② 各項目へ下表のように入力し、[登録]を押下する。

Workspace ID	Organization*	Workspace Name*	Terraform Version	アクセス権
自動入力				設定

※*は必須項目です

戻る 登録

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

3.3 Workspaceの登録と連携(2/2)

Workspaceを連携する

- Workspaces管理からWorkspaceの項目を作成した後、
[連携状態チェック]で対象のTerraformに追加したWorkspaceがあるかどうかをチェックすることができます。
- 「登録なし」であれば[登録]を押下することで対象のTerraformにWorkspaceを作成できます。

※WorkspaceはOrganization上に作成されるため、必ず先にOrganizationを対象のTerraformに作成しておく必要があります

履歴	更新	廃止	Workspace ID	Organization	Workspace Name	Terraform Version	Terraform連携					Movement一覧
							状態チェック	連携状態	登録	更新	削除	
履歴	更新	廃止	1	ITAllearn_org	ITA-demo-AWS		連携状態チェック	登録済み	登録	更新	削除	Movement一覧
履歴	更新	廃止	2	ITAllearn_org	ITA-demo-Azure		連携状態チェック	登録なし	登録	更新	削除	Movement一覧

フィルタ結果件数: 2

Excel出力

ITAllearn_org / Workspaces

Workspaces 3 total

All 3 Success 0 Error 0 Needs Attention 0 Running 0

WORKSPACE NAME	RUN STATUS	RUN	REPO	LATEST CHANGE
ITA-demo-AWS				a few seconds ago
ITA-demo-Azure				9 minutes ago

3.4 作業パターン(Movement)の登録

Movementを作成する

先のplaybookを関連付けるMovementを登録しましょう。

メニュー：Terraform > Movement一覧

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

Movement ID	Movement名*	遅延タイマー	Terraform利用情報 Organization:Workspace*	アクセス権
自動入力	<input type="text"/>	<input type="text"/>	<input type="text"/>	設定 アクセス許可ロー

※*は必須項目です。

Movement名	Terraform利用情報 Organization:Workspace
VM作成(AWS)	ITA-demo-AWS
VM作成(Azure)	ITA-demo-Azure

3.5 Module素材の登録

Moduleを登録する

作成したModuleをITAに登録しましょう。

メニュー：Terraform > Module素材集

- ① 登録 > 登録開始 を押下する。
- ② [参照] からプレイブックを選択し、「事前アップロード」を行う。
- ③ 各項目へ下表のように入力し、「登録」を押下する。

Module素材名	Module素材
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 MovementにModule素材を指定

MovementにModuleを紐付ける

作成したMovementとModule素材を関連付けましょう。

メニュー: Terraform > Movement-Module紐付

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

Movement	Module素材
VM作成(AWS)	aws_create_instance_variables
VM作成(AWS)	aws_create_instance_body
VM作成(Azure)	azure_create_instance_valiables
VM作成(Azure)	azure_create_instance_body

4. 実行編

4.1 オペレーションの登録

オペレーションを新規登録する

オペレーションを作成しましょう。

メニュー：基本コンソール > 投入オペレーション一覧

- ① 登録 > 登録開始 を押下する。
- ② 各項目へ下表のように入力し、[登録]を押下する。

登録

No.	オペレーションID	オペレーション名*	実施予定日時*	備考
自動入力	自動入力	<input type="text"/>	<input type="text"/>	

※*は必須項目です。

戻る

登録

オペレーション名	実施予定日時
Terraform_demo	(任意でご入力下さい)

※ 「実施予定日時」は管理用の項目です。自動的に処理が実行されるわけではありません。

4.2 変数値の設定(1/4)

変数に数値を設定する

Moduleの変数に具体的な数値を設定しましょう。

メニュー: **Terraform > 代入値管理**

- ① 登録 > 登録開始 を押下する。
- ② 各項目で下表のように選択または入力し、[登録]を押下する。

登録

項番	オペレーション*	Movement*	変数名*	HCL設定*	Sensitive設定*	
自動入力	<input type="text"/>	<input type="text"/>	Movementを選択して下さい	OFF <input type="text"/>	OFF <input type="text"/>	<input type="text"/>

※*は必須項目です。

戻る

登録

4.2 変数値の設定(2/4)

変数に数値を設定する(1/3)

代入値の登録は以下の表を参考に行ってください。

オペレーション	Movement	変数名	具体値
Terraform_demo	VM作成(AWS)	security_group	ita-demo-sg※
Terraform_demo	VM作成(AWS)	key_name	ita-demo-key※
Terraform_demo	VM作成(AWS)	access_key	(AWSアクセスキー)
Terraform_demo	VM作成(AWS)	secret_key	(AWSシークレットキー)
Terraform_demo	VM作成(AWS)	region	(任意のリージョン)
Terraform_demo	VM作成(AWS)	tags_name	ita-demo-instance
Terraform_demo	VM作成(AWS)	hello_tf_instance_type	t2.micro
Terraform_demo	VM作成(AWS)	hello_tf_instance_count	3
Terraform_demo	VM作成(AWS)	ami	(任意のAMI)

※セキュリティグループ、キーペアは事前に作成しておく必要があります。

4.2 変数値の設定(3/4)

変数に数値を設定する(2/3)

代入値の登録は以下の表を参考に行ってください。

オペレーション	Movement	変数名	具体値
Terraform_demo	VM作成(Azure)	subscription_id	(Azure認証情報)
Terraform_demo	VM作成(Azure)	tenant_id	
Terraform_demo	VM作成(Azure)	client_id	
Terraform_demo	VM作成(Azure)	client_secret	
Terraform_demo	VM作成(Azure)	resource_group_name	ita-demo-rg
Terraform_demo	VM作成(Azure)	location	ita-demo-web-azure
Terraform_demo	VM作成(Azure)	security_group	ita-demo-security-group
Terraform_demo	VM作成(Azure)	Vnet_name	ita-demo-vnet
Terraform_demo	VM作成(Azure)	Vnet_address_space	10.0.0.0/16
Terraform_demo	VM作成(Azure)	subnet_name	ita-demo-subnet
Terraform_demo	VM作成(Azure)	address_prefixes	10.0.2.0/24
Terraform_demo	VM作成(Azure)	public_ip_name	ita-demo-public-ip
Terraform_demo	VM作成(Azure)	allocation_method	Dynamic
Terraform_demo	VM作成(Azure)	domain_name_label	ita-demo-domain
Terraform_demo	VM作成(Azure)	network_interface_name	ita-demo-nwif

4.2 変数値の設定(4/4)

変数に数値を設定する(3/3)

代入値の登録は以下の表を参考に行ってください。

オペレーション	Movement	変数名	具体値
Terraform_demo	VM作成(Azure)	NIC_name	ita-demo-NIC
Terraform_demo	VM作成(Azure)	VM_name	ita-demo-web-azure
Terraform_demo	VM作成(Azure)	publisher	OpenLogic
Terraform_demo	VM作成(Azure)	offer	CentOS
Terraform_demo	VM作成(Azure)	sku	8_2
Terraform_demo	VM作成(Azure)	source_image_version	latest
Terraform_demo	VM作成(Azure)	os_disk_name	ita-demo-os-disk
Terraform_demo	VM作成(Azure)	storage_account_type	Standard_LRS
Terraform_demo	VM作成(Azure)	caching	ReadWrite
Terraform_demo	VM作成(Azure)	admin_username	ita-demo
Terraform_demo	VM作成(Azure)	ssh_public_key	(任意のSSH公開鍵)
Terraform_demo	VM作成(Azure)	VM_size	Standard_B1ls
Terraform_demo	VM作成(Azure)	VM_count	3

4.3 作業実行

Movementを実行する

前項までの操作で、実行するMovementの作成と代入値の登録が終了しました。
最後にMovementを実行し、結果を対象ホストで確認してください。

メニュー： Terraform > 作業実行

The screenshot shows the Terraform console interface. On the left is a sidebar menu with options like 'メインメニュー', 'インターフェース情報', 'Organizations管理', 'Workspaces管理', 'Movement一覧', 'Module素材集', 'Policies管理', 'Policy Sets管理', 'PolicySet-Policy紐付管理', 'PolicySet-Workspace紐付管理', 'Movement-Module紐付', '代入値自動登録設定', '代入値管理', '作業実行', '作業状態確認', and '作業管理'. The main area is divided into two sections: 'Movement[フィルタ]' and 'オペレーション[フィルタ]'. In the 'Movement[フィルタ]' section, a table lists two movements, both named 'Terraform'. The first movement is selected with a radio button. A callout bubble with the number '1' points to this selection, with the text '実行するMovementを選択する。'. Below the table, it says 'フィルタ結果件数: 2'. In the 'オペレーション[フィルタ]' section, a table lists one operation named 'Terraform_demo'. It is also selected with a radio button. A callout bubble with the number '2' points to this selection, with the text 'オペレーションを選択する。'. Below the table, it says 'フィルタ結果件数: 1'. At the bottom, there are two buttons: 'Plan確認' and '実行'. A callout bubble with the number '3' points to the '実行' button, with the text '[実行]を押下する。'. A 'Tips' box on the right contains the text: '実行後、自動で【作業状態確認】へ画面遷移します。' and '[Plan確認]はPlan/PolicyCheckまでを行い、処理を停止します。'.

Menu

メインメニュー
インターフェース情報
Organizations管理
Workspaces管理
Movement一覧
Module素材集
Policies管理
Policy Sets管理
PolicySet-Policy紐付管理
PolicySet-Workspace紐付管理
Movement-Module紐付
代入値自動登録設定
代入値管理
作業実行
作業状態確認
作業管理

Movement[フィルタ]

Movement[一覧]

選択	Movement ID	Movement名	オーケストレータ	遅延タイマー	Terraform利用情報 Organization:Workspace	アクセス権 アクセス許可ロー	備考	最終更新日時	最終更新者
<input checked="" type="radio"/>	1	VM作成(AWS)	Terraform		ITAllearn_org:ITA-demo-AWS			2021/07/20 15:18:26	システム管理者
<input type="radio"/>	2	VM作成(Azure)	Terraform		ITAllearn_org:ITA-demo-Azure			2021/07/20 15:32:42	システム管理者

フィルタ結果件数: 2

オペレーション[フィルタ]

オペレーション[一覧]

選択	No.	オペレーションID	オペレーション名	実施予定日時	最終実行日時	アクセス権 アクセス許可ロー	備考	最終更新日時	最終更新者
<input checked="" type="radio"/>	1	1	Terraform_demo	2021/07/22 15:36				2021/07/20 15:36:52	システム管理者

フィルタ結果件数: 1

MovementID 1
Movement名 VM作成(AWS)

Plan確認 実行

管理者に連絡

Tips

実行後、自動で【作業状態確認】へ画面遷移します。

[Plan確認]はPlan/PolicyCheckまでを行い、処理を停止します。

4.4 実行状態確認

Movementの詳細結果を確認する

実行後の画面遷移先で、実行ステータスやログを確認することができます。
投入データや出力データを確認することも可能です。

Menu

メインメニュー

インターフェース情報

Organizations管理

Workspaces管理

Movement一覧

Module素材集

Policies管理

Policy Sets管理

PolicySet-Policy紐付管理

PolicySet-Workspace紐付管理

Movement-Module紐付

代入値自動登録設定

代入値管理

作業実行

作業状態確認

作業管理

説明

対象作業

項目	値
作業No.	2
実行種別	通常
ステータス	完了
呼出元Symphony	
呼出元Conductor	
実行ユーザ	システム管理者
ID	1
名称	VM作成(AWS)
Movement	
遅延タイム(分)	
Terraform利用情報	Organization:Workspace ITAlearn_org:ITA-demo-AWS
RUN-ID	run-Dc9RkRGdQekkCrHc
No.	1
オペレーション	
名称	Terraform_demo
ID	1
代入値	確認
入力データ	投入データ
出力データ	結果データ
予約日時	
作業状況	
開始日時	2021/07/21 14:00
終了日時	2021/07/21 14:00

Tips

投入データや結果データをまとめたZipファイルをダウンロードできます

進行状況(Planログ)

フィルタ: ☐ 該当行のみ表示

```
+ 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)
+   iops                   = (known after apply)
+   kms_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

結果はAWS、Azureアカウントにアクセスし各3台のVMが新たに作成されているかを確認してください。

4.5 数値を変更して再度実行(1/2)

■ インスタンス数を変更して再度実行する。

最後に、デプロイするインスタンス数を変更して同様に実行します。

Terraform > 代入値管理から、下表を参考に具体値を変更し、

4.3同様に作業実行しましょう。

変更前

オペレーション	Movement	変数名	具体値
Terraform_demo	VM作成(AWS)	hello_tf_instance_count	3
Terraform_demo	VM作成(Azure)	VM_count	3



変更後

オペレーション	Movement	変数名	具体値
Terraform_demo	VM作成(AWS)	hello_tf_instance_count	5
Terraform_demo	VM作成(Azure)	VM_count	1

デプロイするインスタンス数を
AWS: 3台→5台に増設
Azure: 3台→1台に減設

Tips

4.5 数値を変更して再度実行(2/2)

■ インスタンスの増減を確認

AWS・Azureにブラウザから接続し、

VMインスタンスの数が増減した通りに増減しているか確認しましょう。

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