

# 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 1.10 Exastro developer

#### <u>1.はじめに</u> <u>1.1 はじめに</u> <u>1.2 作業環境</u>

目次

2. 実習 Terraform Driver
 <u>2.1 シナリオ</u>
 <u>2.2 事前準備</u>

<u>3. 仕込み編</u> <u>3.1 インターフェース情報の登録</u> <u>3.2 Organizationの登録と連携</u> <u>3.3 Workspaceの登録と連携</u> <u>3.4 作業パターン(Movement)の登録</u> <u>3.5 Module素材の登録</u> <u>3.5 Module素材の登録</u> <u>3.6 Policy素材の登録</u> <u>3.7 Policy Setの登録</u> <u>3.8 Policy Setの登録</u> <u>3.8 Policy SetとPolicyの紐付け</u> <u>3.9 Policy SetとWorkspaceの紐付け</u> <u>3.10 MovementにModule素材を指定</u>

#### 4. 実行編

- 4.1 オペレーションの登録
- <u>4.2 変数値の設定</u>
- 4.3 Planを確認
- 4.4 PolicyCheckログを確認
- 4.5 VMのサイズを変更して再度確認
- 4.6 再度PolicyCheckログを確認
- 4.7 作業実行
- 4.8 実行状態確認
- 4.9 数値を変更して再度実行

## 1. はじめに



## 1.1 はじめに

## ▲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。

	tro Terraform			ようこそ[システム管理者]さん ログインID [administrator] パスワード変更 ログアウト
⊒ Menu	DASHBOARD			編集
メインメニュー	メニューグループ	Movement	作業状況	作業結果
インターフェース情報				
Organizations管理		Movement	Status	Result
Workspaces管理		0	0	0
Movement一覧	管理コンソール 基本コンソール エクスポート/… Symphony Conductor メニュー作成	Total	Total	Total
Module素材集				
Policies管理		Movement SUM	Status CON SYM SUM	Result CON SYM SUM
Policy Sets管理	比較 ホストグルー… Ansible共通 Ansible-Legacy Ansible-Pioneer Ansible-Lega…	Ansible Legacy 0	■実行中 0 0 0	■正常終了 0 0 0
PolicySet-Policy紐付管理		Ansible Pioneer 0	■ 未実行(予約) 0 0 0 0	
PolicySet-Workspace細		Terraform Q	■ 木美行 0 0 0 	■ 想定外エラー 0 0 0 
付管理				■ 予約取消 0 0 0
Movement-Module紐付	Tarafam			
変数ネスト管理	Terraron III	作業履歴		
代入値自動登録設定				
代入値管理	•			-
管理者に連絡				

## 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ホストサーバとは他に、Terraform環境(Terraform Cloudの場合はアカウント)および AWS、Azureのアカウントをご用意ください。

	ITAホストサーバ	Terraform	ターゲット
•	CentOS7(※) ITA ver 1.10	<ul> <li>Terraform Enterprise or</li> <li>Terraform Cloud</li> </ul>	<ul><li>AWS</li><li>Microsoft Azure</li></ul>



※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



## 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を作成するPlanを確認します。

その後、**定義されたポリシー**に沿った設定を行い各クラウドにVMを作成します。

【仕込み編】の内容を一度設定すると、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)



#### Moduleの作成①

●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。

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。

```
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"
}
```

## Moduleの作成②

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。
```

```
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
}
```

### Moduleの作成③

#### ファイル名:azure\_create\_instance\_valiables.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。

```
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 "storage_account_type" {}
variable "VM_count" {}
```

## Moduleの作成④ (1/3)

#### ファイル名: azure\_create\_instance.tf

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. 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 {
                         = "SSH"
     name
                        = 1001
     priority
     direction
                        = "Inbound"
                        = "Allow"
     access
                        = "Tcp"
     protocol
                             = "*"
     source port range
    destination port range
                              = "22"
                              = "*"
    source address prefix
     destination address prefix = "*"
  }
  security rule {
                         = "HTTP"
     name
                        = 1002
     priority
     direction
                        = "Inbound"
     access
                        = "Allow"
                         = "Tcp"
     protocol
                             = "*"
     source_port_range
     destination port range
                              = "80"
                              = "*"
    source address prefix
     destination address prefix = "*"
  }
```

## 2.2 事前準備(5/7)

## Moduleの作成④ (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" {
                  = var.subnet name
  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" {
                = var.VM count
 count
                 = "${var.public ip name}-${count.index}"
 name
                = azurerm resource group.hogehoge.location
 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" {
                 = var.VM count
  count
  name
                 = "${var.network interface name}-${count.index}"
                = azurerm_resource_group.hogehoge.location
  location
  resource group name = azurerm resource group.hogehoge.name
  ip_configuration {
                          = var.NIC name
     name
                          = azurerm_subnet.hogehoge.id
     subnet id
    private ip address allocation = var.allocation method
     public_ip_address_id
                             = azurerm public ip.hogehoge[count.index].id
```

## Moduleの作成④ (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
                 = "${var.VM_name}-${count.index}"
 name
 resource_group_name = azurerm_resource_group.hogehoge.name
                = azurerm resource group.hogehoge.location
 location
 size
               = var.VM size
                     = var.admin username
 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 {
                  = "${var.os_disk_name}-${count.index}"
  name
                 = var.caching
  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
```

}

## 2.2 事前準備(7/7)

## Policyの作成

ファイル名: limit-proposed-monthlycost.sentinel

月額のコストを制限するポリシー です。

月額コストが\$50を上回る場合は Applyを行いません。

また、その月額コストの総見積り が出力されます。

AWS,Azureどちらのクラウドにおいても適用可能です。

```
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. 仕込み編



#### 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。

۲	Choose an organization $\checkmark$			0 🔼	
Settings	/ Tokens			USER	
USER SETT	TINGS	Tokens		Signed in as User settings	
Profile Session Organiz Passwo	ns zations ord	Your API tokens can be used to access the Terraform Cloud API and perform all tokens documentation 2, Create an API token	the actions your user account is entitled to. For more	Sign out	
Tokens		2	表示 ※ <b>この</b>	されたトークン? 画面を閉じると	をメモしておく <b>再表示できません</b>
6	Create API token	×	Create API token Your new API token is displayed below. Treat t access your account without a username, pass	this token like a password, as it can be used sword, or two-factor authentication.	
	Choose a description to help you ider	tify this token later.	YzJQgNiNgcMNIw.atlasv1.YoD0R7pSjr17yB 1RoJqueQrfoZSAg	8qtWIfDUwZ3Ah34WQs02v1559vVnDHjs5	SHb @
	ITAデモ用Token		A Warning This token will not be displayed agai	<b>in</b> , so make sure to save it to a safe place. (	5
	Create API token Canc	el	Done		

## 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つのみのため、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。

#### 「Terraform」メニューグループ>「インターフェース情報」メニュー

① 「一覧」サブメニューの登録済み項目に表示されている 「更新」ボタン を押下する。

② 各項目へ下表のように入力し「登録」ボタンを押下する。

メインメニュー	▼ ブルダウン検索	▼ プルダウ	>検索 ▼プル	ダウン検索 ▼ プルダウ	シ検索	▼ プ.	~	▼ プルダウン検索		
インターフェース情報										
Organizations管理	フィルタ	一覧								
Workspaces管理										
Movement一覧	一覧	No.	Hostname*	User Token	Address	Proxy	Port	状態監視周期(単位ミリ秒)*	進行状態表示行数*	NULL連
Module素材集	~	1		]				3000	1000	無効 👻
Policies管理	履歴 更新 No.⊖ Hos									
Policy Sets管理	REAL LEAST 1 Terraform									
PolicySet-Policy紐付管理	フィルタ結果件数:1	※*は必	滇項目です。							
			戻る	更新	ŕ					

Hostname	User Token
(Terraformのドメイン名)	(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。

#### 「Terraform」メニューグループ > 「Organizations管理」メニュー

- ① 「登録」サブメニューの 「登録開始」ボタン 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 設定     備考       自動入力	Organization ID     Organization Name*     Email address*     アクセス権       自動入力           (回動入力             ※*は必須項目です。			
自動入力 設定	自動入力 () ※*は必須項目です。	ganization ID Organization Name* Email address*	アクセス権 設定 アクセス許可ロール	備考
		自動入力	設定	
	※*は必須項目です。			

Organization Name	Email address
ITAlearn_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」メニューでOrganizationの項目を作成した後、「連携状態チェック」ボタンを押下することで対象の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」ボタン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					•					
	居歴 複製 可	同新 廢止	Organization TD	Organization Name	Email address≙		Terraform	誘			Work
	ARE BEEC .		or gamillaction 100	or gonizzación nume o		状態チェック	連携状態令	登録	更新	削除	NOT K.
ſ	履歴 複製 🧕	瓦新 廃止	1	ITAlearn_org	test@example.com	連携状態チェック	登録なし	登録	更新	削除	Work
	フィルタ結果件数	数: 1									

	Choose an organization 🗸		0	
	Settings / Organizations			
	USER SETTINGS	Organizations		
	Profile	You are a member of the following organization:		
	Sessions			
	Organizations	ITAlearn_org OWNER		
	Password			
,	Two Factor Authentication			
	Tokens			

## 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。

#### 「Terraform」メニューグループ > 「Workspaces管理」メニュー

- ① 「登録」サブメニューの 「登録開始」ボタン を押下する。
- ② 各項目へ下表のように入力し「登録」ボタンを押下する。

登録				
Workspace ID 自動入力	Organization*	Workspace Name*	Terraform Version	アクセス権 設定 アクセス許可ロール 設定
※*は必須項目です。 戻る		登録		

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

#### Workspaceを連携する

- 「Workspaces管理」メニューでWorkspaceの項目を作成した後「連携状態チェック」 ボタンを押下することで、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」ボタンを押下することで、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

一覧/更新					
屆歴 複製 更新 廃止 Workspace ID 最歴 復製 更新 廃止 扇歴 複製 更新 廃止	Organization       Workspace Name       Terraform Vers         ITAlearn_org       ITA-demo-AWS         ITAlearn_org       ITA-demo-Azure	ion	t N		
フィルタ結果件数: 2	ITAlearn_org V Workspaces Registry Settings	HashiCorp Cloud Platform I2	Upgrade Now 🕜 📔		
Workspaces 3 total					
	WORKSPACE NAME RUN STAT	US 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。

#### 「Terraform」メニューグループ >「Movement一覧」メニュー

- ① 「登録」サブメニューの「登録開始」ボタンを押下する。
- ② 各項目で下表のように選択または入力し「登録」ボタンを押下する。

A* 遅延タイマー Organization:Workspace* 設定 アクセス許可ロー, していたいのではないのでありたいでは、「ないのではない」では、「ないのでは、「ないのではない」では、「ないのでは、「ないのでは、「ないのではない」では、「ないのではない」では、「ないのではない」では、「ないのではない」では、「ないのでは、「ないのではない」では、「ないのでは、「ないのでは、「ないのでは、「ないのでは、「ないの」では、「ないのでは、「ないのでは、「ないのでは、「ないのでは、「ないのでは、「ないのでは、「ないのでは、「ないのでは、「ないのでは、「ないのでは、「ないのでは、「ないのでは、「ないのでは、「ないの」では、「ないのでは、「ないの」」では、「ないの」では、「ないのでは、「ないの」」では、「ないの」では、「ないのでは、「ないの」」、「ないのでは、「ないのでは、「ないのでは、「ないのでは、「ないの」」では、「ないのでは、「ないのでは、「ないの」」では、「ないのでは、「ないの」」では、「ないのでは、「ないの」」では、「ないの」」、「ないの」」、「ないの」」

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

#### Moduleを登録する

作成したModuleをITAに登録しましょう。

#### 「Terraform」メニューグループ > 「Module素材集」メニュー

- ① 「登録」サブメニューの「登録開始」ボタンを押下する。
- ② 「ファイルの選択」ボタンを押下しModuleを選択、 「事前アップロード」ボタンを押下する。
- ③ 各項目へ下表のように入力し、「登録」ボタンを押下する。

登録		
Module素材ID Module素材名* 自動入力	Module素材         アクセス権           設定         アクセス許可ロール           ファイルの選択         ファイルが選…れていません           事前アップロード         認定           アップロード状況:         レーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーー	
※*は必須項目です。	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_valiables	azure_create_instance_valiables.tf
	azure_create_instance_body	azure_create_instance.tf

## 3.6 Policy素材の登録

#### Policyを登録する

作成したPolicyをITAに登録しましょう。

#### 「Terraform」メニューグループ > 「Policies管理」メニュー

- ① 「登録」サブメニューの「登録開始」ボタンを押下する。
- 「ファイルの選択」ボタンを押下しPolicyを選択、「事前アップロード」ボタンを押下 する。
- ③ 各項目へ下表のように入力し、「登録」ボタンを押下する。



Policy名	Policy素材
limit-proposed-monthly-cost	limit-proposed-monthly-cost.sentinel

## 3.7 Policy Setの登録

#### Policy Setを登録する

Policy SetをITAに登録しましょう。

#### 「Terraform」メニューグループ > 「Policy Sets管理」メニュー

- ① 「登録」サブメニューの「登録開始」ボタンを押下する。
- ② 各項目へ下表のように入力し、「登録」ボタンを押下する。

登録			
PolicySet ID 自動入力	PolicySet名*	設定 設定	アクセス権 アクセス許可ロール
:*(は必須項目で	ರಕ.		
戻	3		登録
	PolicySet名		
olicySet (	demo		

## 3.8 Policy SetとPolicyの紐付け

### Policy SetとPolicyを紐付ける

作成したPolicy SetとPolicyを紐付けましょう。

#### 「Terraform」メニューグループ > 「PolicySet-Policy紐付管理」メニュー

- ① 「登録」サブメニューの「登録開始」ボタンを押下する。
- ② 各項目へ下表のように入力し、「登録」ボタンを押下する。

登録			
PolicySet-Policy紐付ID	Policy Set*	Policy*	設定
自動入力	T	· · · · · · · · · · · · · · · · · · ·	設定
※*は必須項目です。			
戻る	登録		

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

## 3.9 Policy SetとWorkspaceの紐付け

#### Policy SetとWorkspaceを紐付ける

作成したPolicy SetとWorkspaceを紐付けましょう。

#### 「Terraform」メニューグループ > 「PolicySet-Workspace紐付管理」メニュー

- ① 「登録」サブメニューの「登録開始」ボタンを押下する。
- ② 各項目へ下表のように入力し、「登録」ボタンを押下する。

登録		
PolicySet-Workspace紐付ID 自動入力	Policy Set*	Organization:Workspace*
※*は必須項目です。		
戻る	登録	

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

## 3.10 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。

#### 「Terraform」メニューグループ > 「Movement-Module紐付」メニュー

- ① 「登録」サブメニューの「登録開始」ボタンを押下する。
- ② 各項目へ下表のように入力し、「登録」ボタンを押下する。

登録				
紐付項番	Movement*		Module素材*	設定
自動入力	•		•	設定
※*は必須項目	です。			
戻	3	登錄	Movement	Module素材
		V	M作成(AWS)	aws_create_instance_variables
		V	M作成(AWS)	aws_create_instance_body
		V	M作成(Azure)	azure_create_instance_valiables





## 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。

#### 「基本コンソール」メニューグループ > 「オペレーション一覧」メニュー

- ① 「登録」サブメニューの「登録開始」ボタンを押下する。
- ② 各項目へ下表のように入力し、「登録」ボタンを押下する。



※「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。

#### 「Terraform」メニューグループ > 「代入値管理」メニュー

- ① 「登録」サブメニューの「登録開始」ボタンを押下する。
- ② 各項目へ下表のように入力し、「登録」ボタンを押下する。

登録								△閉じる
項番	オペレーション	Movement*	変数名*	HCL設定*	メンバー変数	代入順序	最終更新日時	最終更新者
自動入力	•	<b></b>	Movementを選択して下さい	OFF V	変数名を選択して下さい	変数名を選択して下さい	自動入力	自動入力
۹ **(ئىل	必須項目です。							
	<b>戻</b> る <sup>登</sup>	<b>绿</b>						

※具体的な設定値については以降のスライドで説明します。

## 以下の具体値を設定する(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	メンバー変数	具体値
Terraform_demo	VM作成(AWS)	security_group		ita-demo-sg <mark></mark> %
Terraform_demo	VM作成(AWS)	key_name		ita-demo-key <mark>※</mark>
Terraform_demo	VM作成(AWS)	aws_info	access_key	(AWSアクセスキー)
Terraform_demo	VM作成(AWS)	aws_info	secret_key	(AWSシークレットキー)
Terraform_demo	VM作成(AWS)	aws_info	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.large
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。

※「access\_key, secret\_key, region」については「aws\_create\_instance\_variables.tf」にてobject 型で記述しているため「変数名:aws\_info」を選択の上「メンバー変数」をそれぞれ選択します。

## 以下の具体値を設定する(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	メンバー変数	具体値
Terraform_demo	VM作成(Azure)	azure_info	subscription_id	
Terraform_demo	VM作成(Azure)	azure_info	tenant_id	(∧っぃゃっ言刃言エ/書志巳)
Terraform_demo	VM作成(Azure)	azure_info	client_id	
Terraform_demo	VM作成(Azure)	azure_info	client_secret	
Terraform_demo	VM作成(Azure)	resource_group_name		ita-demo-rg
Terraform_demo	VM作成(Azure)	location		japaneast
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		10.0.0/16
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		(任意のグローバルなドメイン名)

※「subscription\_id, tenant\_id, client\_id, client\_secret」については「azure\_create\_instance\_variables.tf」にてobject型で記述し ているため「変数名:azure\_info」を選択の上「メンバー変数」をそれぞれ選択します。

## 以下の具体値を設定する(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)	network_interface_name	ita-demo-nwif
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_B2MS
Terraform_demo	VM作成(Azure)	VM_count	3

※事前に用意したSSH公開鍵を利用してください。入力する具体値はSSH公開鍵のテキスト「ssh-rsa xxxxxxxx~」となります。

#### Plan確認を実施する

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。 次にPlan確認を実施し、Moduleが定義されたポリシーに沿っているか確認します。

#### 「Terraform」メニューグループ > 「作業実行」メニュー



## 4.4 PolicyCheckログを確認

#### PolicyCheckログを確認する

押下後の画面遷移先で、ステータスが「完了(異常)」となっていることが確認できます。 画面下に移動し、PolicyCheckログを確認しましょう。

#### 対象作業

	項目		値
作業No.			40
実行種別			Plan確認
ステータス			完了(異常)
呼出元Symphony			
呼出元Conducto	r		
実行ユーザ			システム管理者
	ID		1
	名称		VM作成(AWS)
Movement	遅延タイマ(分)		
		Organization:Workspace	ky-exastro-demo:ITA-demo-AWS
	lerratorm木小/H1月¥Q	RUN-ID	run-qM3wTwemyWLMp735
	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		InputData 0000000040.zip
出力データ	結果データ		ResultData 000000040.zip
	予約日時		
作業状況	開始日時		2022/05/23 14:15:04
	終了日時		2022/05/23 14:15:30

進行状況(	PolicyCheckログ)
フィルタ:	□該当行のみ表示
this was du sub-rules. violation(s or non-zero	e to the discovery of violations by the main rule and other Please see the details of the policies executed below to find the ), which is usually indicated by a rule with a false boolean value, collection data.
1 policies	evaluated.
## Policy 1	: limit-proposed-monthly-cost (hard-mandatory)
Result: fal	se
Print messa	ges:
Proposed mo	nthly cost 207.1296 of workspace ITA-demo-AWS is over the limit: \$ {"coefficient
./limit-pro Value: false	posed-monthly-cost sentinel:33:1 - Rule "main"
4	Tips
	コストが\$50を大幅に
	上回っていることが確認できる

#### VMのサイズを変更して再度実行する。

最後に、デプロイするVMのサイズを変更して同様に実行します。

「Terraform」メニューグループ > 「代入値管理」メニューから、下表を参考に具体値を 変更し、4.3同様にPlan確認を実施しましょう。

#### 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_type	t2.large	
Terraform_demo	VM作成(Azure)	VM_size	Standard_B2MS	
変更後		デプロイするインスタ AWS : t2.large Azure: Standard_B	マンスサイズ → t2.micro 2MS → Standard_	_B1LS Tips
オペレーション	Movement	変数名	具体値	
Terraform_demo	VM作成(AWS)	hello_tf_instance_type	t2.micro	
Terraform_demo	VM作成(Azure)	VM_size	Standard_B1LS	

### 再びPolicyCheckログを確認する

押下後の画面遷移先で、ステータスが「完了」となっていることが確認できます。 画面下に移動しPolicyCheckログを確認後、Movementを実際に実行しましょう。

対象作業				進行状況(PolicyCheckログ)	
	項目		値		
作業No.	作業No.		41		
実行種別			Plan確認	Sentinel Result: true	
ステータス			完了	This papelt mapps that all Sentinal policies passed and the protested	
呼出元Symphony	,			behavior is allowed.	
呼出元Conducto	r			1 policies evaluated.	
実行ユーザ			システム管理者		
	ID		1	<pre>## Policy 1: limit-proposed-monthly-cost (hard-mandatory)</pre>	
	名称		VM作成(AWS)	Result: true	
Movement	遅延タイマ(分)			Print messages:	
	Organization:Work	Organization:Workspace	ky-exastro-demo:ITA-demo-AWS	Proposed monthly cost 25 8012 of workspace ITA-damo_AWS is under the limit \$ {"coefficie	
	RUN-ID		run-DocemGy9tAARgsFu	Proposed monthly cost 23.6512 of workspace TrA-demo-Aws is under the limit 3 { coefficient :	
	No.		1	./limit-proposed-monthly-tost.sentinel:33:1 - Rule "main"	
オペレーション	名称		Terraform_demo	true	
	ID		1		
代入値	代入値		確認		
入力データ 投入データ		InputData 0000000041.zip	Tins		
出力データ 結果データ		ResultData 000000041.zip			
予約日時			夏史俊、コストか\$50を		
作業状況	開始日時		2022/05/23 14:20:13	下回っていることが確認できた。	
	終了日時		2022/05/23 14:20:42		

#### Movementを実行する

実行するModuleが定義したポリシーを適用していることが確認できました。 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。

#### 「Terraform」メニューグループ > 「作業実行」メニュー

enu	Movement[フィルタ]
-ב=אי	Movement 「町」 実行するMovementを選択する。
?ーフェース情報	
anizations管理	選択 Movement J <sup>1+</sup> , Movement名☆ オーケストレータ令 遅延タイマー令 遅延タイマー令 Organization:Workspace令 アクセス許可ロール会 備考令 最終更新日時令 最終更新者令
spaces管理	①         1         VM作成(AWS)         Terraform         ky-exastro-demo:ITA-demo-AWS         2022/05/16         10:54:52         システム管理者
ient一覧	2     VW作成(Azure)     Terraform     ky-exastro-demo-Azure     2022/05/16 10:55:06     システム管理者
le素材集	フィルタ結果件数: 2
es管理	
Sets管理	オペレーション[フィルタ]
Set-Policy紐付管理	オペレーション[一覧] 2 オペレーションを選択する。
icySet-Workspace紐	7/1/7 接
理	選択 No.
ment-Module紐付	①         1         Terraform_demo         2022/05/16         14:04         2022/05/23         14:19         2022/05/23         14:19:46         Terraform/作業実行プロシージャ
マト管理	フィルタ結果件数: 1
自動登録設定	
値管理	3 「実行」ボタンを押下する。 Tipe
ŧī	MovementD 1 MovementS VM/rtd(AWS)
《態確認	
聲理	

#### 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。

対象作業				進行状況(Planログ)
	項目		値	
作業No.			42	フィルタ: 回該当行のみ表示
実行種別			通常	
ステータス			完了	DTerraform v1.2.0
呼出元Symphony				on linux_amd64
呼出元Conducto	r			Initializing Terraform configuration
実行ユーザ			システム管理者	{"@level":"info","@message":"Terraform 1.2.0","@module":"terraform.ui","@timestamp":
	ID		1	راتوبوبوبوبا: "Info", "@message": "aws_instance.neiio-tf-instance[0]: Refreshing state {"@level":"info"."@message":"aws instance.hello-tf-instance[1]: Refreshing state
	名称		VM作成(AWS)	<pre>{"@level":"info","@message":"aws_instance.hello-tf-instance[2]: Refreshing state</pre>
Movement	遅延タイマ(分)			["@level":"info", "@message":"aws_instance.hello-tf-instance[0]: Refresh complete [id {"@level":"info", "@message":"aws_instance.hello-tf-instance[2]: Refresh complete [id
		Organization:Workspace	ky-exastro-demo:ITA-demo-AWS	{"@level":"info","@message":"aws_instance.hello-tf-instance[1]: Refresh complete [id
	lerratorm不引用1月程	RUN-ID	run-3fk7SA9fhSTr9eXZ	<pre>{"@level":"info","@message":"aws_instance.hello-tf-instance[3]: Plan to create","@mo {"@lavel":"info"""@message":"aws_instance hello-tf-instance[4]: Plan to create","@me </pre>
	No.		1	<pre>{"@level::"info", "@message": "Plan: 2 to add, 0 to change, 0 to destroy.", "@module":"</pre>
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		InputData 000000042.zip	
出力データ	結果データ		ResultData 000000042.zip	
	予約日時			
作業状況	開始日時		2022/05/23 14:31:51	Tine
	終了日時		2022/05/23 14:33:17	lips
投入 Zipフ	データヤ アイルを	■結果デーク ダウンロー	タをまとめた -ドできます。	ips 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。

#### 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」メニューグループ > 「代入値管理」メニューから、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		デプロイするイン AWS : 3台→ Azure : 3台 →	vスタンス数を 5台 に <mark>増設</mark> 1台 に <mark>減設</mark>	Tir
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	

### 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変更した通りに増減している か確認しましょう。

#### AWS

Name	▲ インスタンス ID
ita-demo-instance-1	i-09d4ef69fbb6dbefc
ita-demo-instance-2	i-00279e828d1584f73
ita-demo-instance-3	i-08f638e6fb9fcd2a2

#### Azure

□ 名前 ↑↓
🗌 🌆 ita-demo-nwif-0
🔲 🌆 ita-demo-nwif-1
🔲 🌇 ita-demo-nwif-2
🗌 🛢 ita-demo-os-disk-0
🗌 🛢 ita-demo-os-disk-1
🗌 🛢 ita-demo-os-disk-2
🔲 📼 ita-demo-public-ip-0
🔲 📼 ita-demo-public-ip-1
🔲 🚟 ita-demo-public-ip-2
🗌 🎈 ita-demo-security-group
☐ < i>ita-demo-vnet
🗌 🖳 ita-demo-web-azure-0
🗌 👤 ita-demo-web-azure-1
🔲 🖳 ita-demo-web-azure-2

Name	▲ インスタンス ID
ita-demo-instance-1	i-09d4ef69fbb6dbefc
ita-demo-instance-2	i-00279e828d1584f73
ita-demo-instance-3	i-08f638e6fb9fcd2a2
ita-demo-instance-4	i-0892491d496eb3cf5
ita-demo-instance-5	i-0adb40124875a7b06



