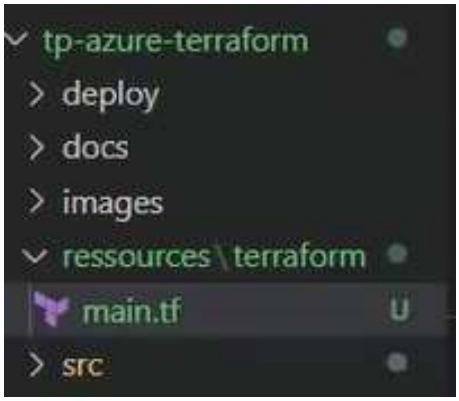
Correction – TP Global Azure

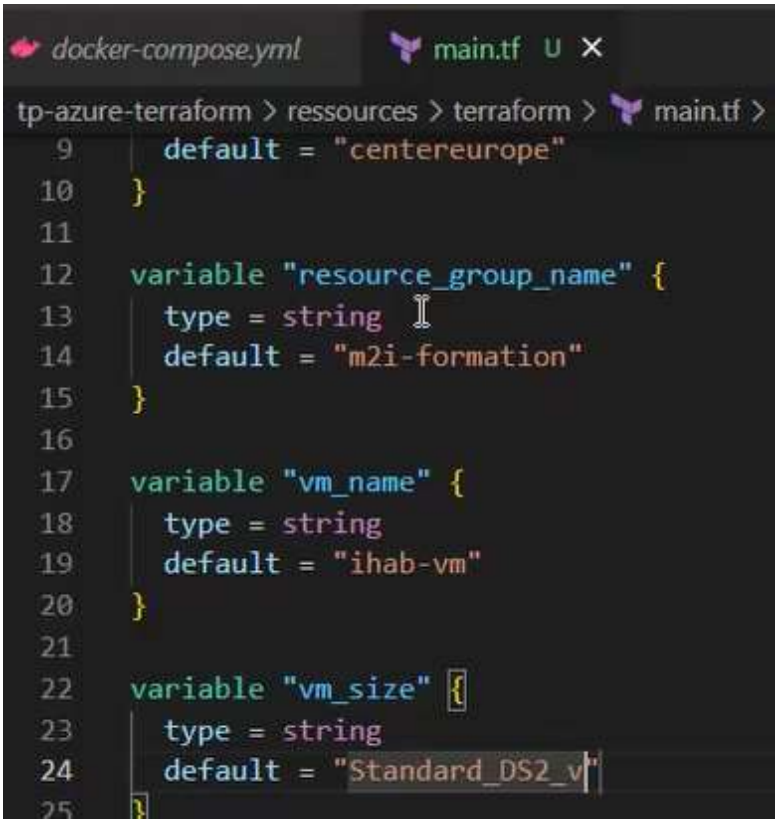
lundi 27 mars 2023 09:35

Formateur : Ihab ABADI

Création dossier nécessaire :



Définition des variables :



Création ressources Terraform :

Network/subnet :

```
### Création des ressources

resource "azurerm_virtual_network" "ihab_vnet" {
   name = "ihab-vnet"
   resource_group_name = var.resource_group_name
   location = var.location
   address_space = ["10.0.0.0/16"]
}

resource "azurerm_subnet" "ihab_vsubnet" {
```

```
    name = "ihab-vsubnet"
    virtual_network_name = azurerm_virtual_network.ihab_vnet
    resource_group_name = var.resource_group_name
}
```

Network interface :

```
resource "azurerm_network_interface" "ihab_nic" {
    name = "ihab-nic"
    location = var.location
    resource_group_name = var.resource_group_name
    ip_configuration {
        name = "ip-ihab"
        subnet_id = azurerm_subnet.ihab_vsubnet.id
        private_ip_address_allocation = "dynamic"
    }
}
```

Private key :

```
resource "tls_private_key" "ihab_key" {
  algorithm = "RSA"
  rsa_bits = "4096"
}
```

VM :

```
resource "azurerm_linux_virtual_machine" "ihab-vm" {
  count = 2
  name = "${var.vm_name}-${count}"
  resource_group_name = var.resource_group_name
  location = var.location
  size = var.vm_size
  admin_username = "azureuser"
  admin_password = var.admin_password
  network_interface_ids = [azurerm_network_interface.ihab_nic.id]
  admin_ssh_key {
    username = "azureuser"
    public_key = tls_private_key.ihab_key.public_key_openssh
  }
  os_disk {
    storage_account_type = "Standard_LRS"
```

ACR (container registry) :

```
#création d'un acr
resource "azurerm_container_group" "ihab-acr" {
  name = var.acr_name
  location = var.location
  resource_group_name = var.resource_group_name
  sku = "Premium"
  admin_enabled = true
}
```

abc true
abc terraform

Cluster AKS :

```
# création du cluster aks

resource "azurerm_kubernetes_cluster" "ihab-aks" {
  name = var.aks_name
  location = var.location
  resource_group_name = var.resource_group_name
  dns_prefix = "my-aks"
  default_node_pool {
    name = "default"
    node_count = var.aks_nodes_number
    vm_size = var.vm_size
    vnet_subnet_id = azurerm_subnet.ihab_vsubnet.id
  }
}
```

Bloc **service_principal** pour la connexion AKS

```
service_principal {
  client_id = ""
  client_secret = ""
}
```

Bloc output

```
output "acr_server" {
  value = azurerm_container_registry.ihab-acr.login_server
}

output "acr_admin_acr" {
  value = azurerm_container_registry.ihab-acr.admin_username
}

output "acr_admin_password_acr" {
  value = azurerm_container_registry.ihab-acr.admin_password
}
```

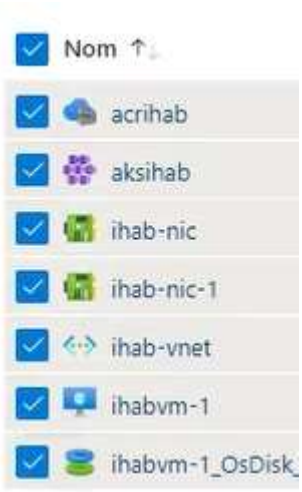
Bloc ip public :

```
resource "azurerm_public_ip" "ip_public" {
  count = 2
  name = "ihab-t-ip-public-${count.index}"
  location = var.location
  resource_group_name = var.resource_group_name
  allocation_method = "Dynamic"
}
```

Run du fichier main Terraform

```
tls_private_key.ihab_key: Creating...
tls_private_key.ihab_key: Creation complete after
]
azurerm_virtual_network.ihab_vnet: Creating...
```

```
azurerm_container_registry.ihab-acr: Creating...
azurerm_virtual_network.ihab_vnet: Creation compl
37-b1a6-578c7faa36fd/resourceGroups/m2i-formation
b-vnet]
azurerm_subnet.ihab_vsubnet: Creating...
azurerm_container_registry.ihab-acr: Still creati
azurerm_subnet.ihab_vsubnet: Creation complete af
6-578c7faa36fd/resourceGroups/m2i-formation/provi
/subnets/ihab-vsubnet]
azurerm_kubernetes_cluster.ihab-aks: Creating...
```



Code utilisé :

```
provider "azurerm" {
  features {

  }
  skip_provider_registration = true
}


#### Déclaration des variables
variable "location" {
  type = string
  default = "westeurope"
}


variable "resource_group_name" {
  type = string
  default = "m2i-formation"
}


variable "vm_name" {
  type = string
  default = "ihabvm"
}


variable "vm_size" {
  type = string
  default = "Standard_DS2_v2"
}


variable "admin_password" {
  type = string
  default = "Azure123456."
}


variable "aks_name" {
```

```
  type = string
  default = "aksihab"
}



variable "acr_name" {
  type = string
  default = "acrihab"
}


variable "aks_nodes_number" {
    type = number
    default = 2
}


variable "aks_client_id" {
  type = string
  default = "ac865e4d-92cd-43e9-a3ac-fb3e5dc483ef"
}


variable "aks_secret_id" {
  type = string
  default = "Jv38Q~IKp-otYhUem.DVcL.UlsymJpW6CnWJccXO"
}
### Création des ressources


resource "azurerm_virtual_network" "ihab_vnet" {
   name = "ihab-vnet"
   resource_group_name = var.resource_group_name
   location = var.location
   address_space = ["10.0.0.0/16"]
}


resource "azurerm_subnet" "ihab_vsubnet" {
    name = "ihab-vsubnet"
    virtual_network_name = azurerm_virtual_network.ihab_vnet.name
    resource_group_name = var.resource_group_name
    address_prefixes = ["10.0.1.0/24"]
}


resource "azurerm_public_ip" "ip_public" {
  count = 2
  name = "ihab-t-ip-public-${count.index}"
  location = var.location
  resource_group_name = var.resource_group_name
  allocation_method = "Dynamic"
}


resource "azurerm_network_interface" "ihab_nic" {
    count = 2
    name = "ihab-nic-${count.index}"
    location = var.location
    resource_group_name = var.resource_group_name

    ip_configuration {
        name = "ip-ihab"
        subnet_id = azurerm_subnet.ihab_vsubnet.id
        private_ip_address_allocation = "Dynamic"
        public_ip_address_id = azurerm_public_ip.ip_public[count.index].id
    }
}


resource "tls_private_key" "ihab_key" {
  algorithm = "RSA"
  rsa_bits = "4096"
}



# création de vm
```

```
resource "azurerm_linux_virtual_machine" "ihab-vm" {
  count = 2
  name = "${var.vm_name}-${count.index}"
  resource_group_name = var.resource_group_name
  location = var.location
  size = var.vm_size
  admin_username = "azureuser"
  admin_password = var.admin_password
  network_interface_ids = [azurerm_network_interface.ihab_nic[count.index].id]
  admin_ssh_key {
    username = "azureuser"
    public_key = tls_private_key.ihab_key.public_key_openssh
  }
  os_disk {
    storage_account_type = "Standard_LRS"
    caching = "ReadWrite"
  }
  source_image_reference {
    publisher = "Canonical"
    offer = "UbuntuServer"
    sku = "18.04-LTS"
    version = "latest"
  }


  connection {
    type     = "ssh"
    user     = "azureuser"
    password = var.admin_password
    host     = self.public_ip
  }


  provisioner "remote-exec" {
    inline = [
      "sudo apt update",
      "sudo apt upgrade -y",
      "sudo apt install python3-pip -y",
      "sudo python3 -m pip install --user ansible",
      "sudo python3 -m pip install --upgrade --user ansible",
      "ansible-galaxy collection install community.docker",
      "ansible-galaxy collection install azure.azcollection"
    ]
  }
}



#création d'un acr
resource "azurerm_container_registry" "ihab-acr" {
  name = var.acr_name
  location = var.location
  resource_group_name = var.resource_group_name
  sku = "Premium"
  admin_enabled = true
}



# création du cluster aks


resource "azurerm_kubernetes_cluster" "ihab-aks" {
  name = var.aks_name
  location = var.location
  resource_group_name = var.resource_group_name
  dns_prefix = "my-aks"
  default_node_pool {
    name = "default"
    node_count = var.aks_nodes_number
    vm_size = var.vm_size
    #vnet_subnet_id = azurerm_subnet.ihab_vsubnet.id
  }


  service_principal {
    client_id = var.aks_client_id
    client_secret = var.aks_secret_id
  }
}
```

```
#output
output "private_key" {
  value = tls_private_key.ihab_key.private_key_pem
  sensitive = true
}


output "acr_server" {
  value = azurerm_container_registry.ihab-acr.login_server
}


output "acr_admin_acr" {
  value = azurerm_container_registry.ihab-acr.admin_username
}


output "acr_admin_password_acr" {
  value = azurerm_container_registry.ihab-acr.admin_password
  sensitive = true
}
```

Création IP publique :

Pour afficher la clé qu'on a généré dans notre **main.tf** : **terraform output private_key > nom_fichier**

```
PS C:\Users\Administrateur\Desktop\Azure\tp-azure-terraform\ressources\terraform> terraform outpu
t private_key > ihab.pem
```

Connexion VM : SSH

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
```

```
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@ihabvm-0:~$
```

Installation Ansible :

```
azureuser@ihabvm-0:~$ sudo apt update && upgrade -y && sudo apt install ansible -y
Hit:1 http://azure.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://azure.archive.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
Building dependency tree
```

Configuration host + playbook :

```
hosts U    build-playbook.yml U ×
tp-azure-terraform > ressources > playbook > build-playbook.yml
1  ---
2  - hosts: ihab
3    become: true
4    tasks:
5    - name: install git
6      apt:
7        name: git
8        update_cache: yes
9    - name: copy script install docker
10     copy:
11       src: ./docker.sh
12       dest: ./docker.sh
13   - name: run install docker
14     shell:
15       cmd: chmod +x docker.sh && ./docker.sh
16   - name: clone project
17     git:
```

```
azureuser@ihabvm-0:~$ touch hosts
azureuser@ihabvm-0:~$ touch ansible.cfg
azureuser@ihabvm-0:~$ touch playbook.yml
azureuser@ihabvm-0:~$ nano ansible.cfg
azureuser@ihabvm-0:~$ nano hosts
azureuser@ihabvm-0:~$ nano ihab.pem
azureuser@ihabvm-0:~$ nano playbook.yml
azureuser@ihabvm-0:~$ chmod 400 ihab.pem
azureuser@ihabvm-0:~$
```

Test du playbook

Installation Python/Git/Docker/Ansible :

```
azureuser@ihabvm-0:~$ python3 -m pip install --user ansibl
Collecting ansible
  Downloading https://files.pythonhosted.org/packages/fd/f
eeb97fdf4068642b22edcf3/ansible-4.10.0.tar.gz (36.8MB)
    100% |                                | 36.8MB 32kB/s
```

```
 executed
azureuser@ihabvm-0:~$ sudo python3 -m pip show ansible
WARNING: The directory '/home/azureuser/.cache/pip' or its par
t writable by the current user. The cache has been disabled. (
hat directory. If executing pip with sudo, you should use sudo
```

```
Name: ansible
Version: 4.10.0
Summary: Radically simple IT automation
Home-page: https://ansible.com/
Author: Ansible, Inc.
Author-email: info@ansible.com
License: GPLv3+
Location: /home/azureuser/.local/lib/python3.6/site-packages
Requires: ansible-core
Required-by:
```

Extrait du build playbook :

```
- name: Build an image and push
  community.docker.docker_image:
    build:
      path: /pokemon-app/{{item}}
    name: m2informationihab.azurecr.io/{{item}}
    source: build
    push: true
  with_items:
    - dollop
    - fastapp
```

```
- name: copy java docker
  copy:
    src: /tp-azure-terraform/images/java17/Dockerfile
    dest: /tp-azure-terraform/src/{{item}}
  with_items:
    - ui
    - cart
    - orders
- name: Build an image and push
```

Le bloc **provisioner "remote-exec"** permet d'exécuter des commandes sur la machine en admin à sa création :

```
provisioner "remote-exec" {
  inline = [
    "sudo apt update",
    "sudo apt upgrade -y",
    "sudo apt install python3-pip -y",
    "sudo python3 -m pip install --user ansible",
    "sudo python3 -m pip install --upgrade --user ansible",
    "ansible-galaxy collection install community.docker",
    "ansible-galaxy collection install azure.azcollection"
  ]
}
```

Code du playbook :

```
---
- hosts: ihab
  become: true
  tasks:
  - name: install git
    apt:
      name: git
      update_cache: yes
  - name: copy script install docker
    shell:
      cmd: wget get.docker.com -o ./docker.sh
  - name: copy script install docker
```

```
  copy:
    src: ./docker.sh
    dest: ./docker.sh
- name: clone project
  git:
    repo: https://github.com/utopios/tp-azure-terraform
    dest: /tp-azure-terraform

- name: connect to azurecr
  community.docker.docker_login:
    registry_url: acrihab.azurecr.io
    username: acrihab
    password: eMsoLSwJe5+5kclHnK11IAbsqAp+ShGoyuGLBKpJBh+ACRAbauCq

- name: copy java docker
  copy:
    src: /tp-azure-terraform/images/java17/Dockerfile
    dest: /tp-azure-terraform/src/{{item}}
  with_items:
    - ui
    - cart
    - orders

- name: copy nodejs docker
  copy:
    src: /tp-azure-terraform/images/nodejs/Dockerfile
    dest: /tp-azure-terraform/src/{{item}}
  with_items:
    - checkout

- name: Build an image and push
  community.docker.docker_image:
    build:
      path: /tp-azure-terraform/src/{{item}}
    name: acrihab.azurecr.io/{{item}}
    source: build
    push: true
  with_items:
    - ui
    - cart
    - orders
    - checkout
    - assets
    - catalog

- name: Get Aks Credentials
  azure_rm_aks_info:
    resource_group: "m2i-formation"
    client_id: "ac865e4d-92cd-43e9-a3ac-fb3e5dc483ef"
    secret: "Jv38Q~IKp-otYhUem.DVcL.UlsymJpW6CnWJccXO"
    name: "aks-ihab"
  register: aks_credentials

- name: create k8s ressources
  k8s:
    src: /tp-azure-terraform/k8s/{{item}}
    kubeconfig: {{aks_credentials.kubeconfig}}
  with_items:
    - resources.yml
```

Mise à jour Dockerfiles :

Création Manifeste :

Extrait des manifestes

```
  app: ui
pec:
 containers:
   - image: charlesracr.azurecr.io/ui
     name: ui
     env:
       - name: JAVA_OPTS=-XX:MaxRAMPercentage
         value: 75.0 -Djava.security.egd=file:/dev/urandom
       - name: SERVER_TOMCAT_ACCESSLOG_ENABLED
         value: true
       - name: ENDPOINTS_CATALOG
         value: http://local.default.svc.catalog:8080
       - name: ENDPOINTS_CARTS
         value: http://local.default.svc.carts:8080
       - name: ENDPOINTS_ORDERS
         value: http://local.default.svc.orders:8080
       - name: ENDPOINTS_CHECKOUT
```

```
metadata:
  labels:
    app: mongodb
spec:
  containers:
    - image: mongo
      name: mongodb
      imagePullPolicy: Always
      ports:
        - containerPort: 27017
          name: mongodb
```

Code utilisé :

```
---
- hosts: ihab
  become: true
  tasks:
  - name: install git
    apt:
      name: git
      update_cache: yes
  - name: copy script install docker
    shell:
      cmd: wget get.docker.com -o ./docker.sh
  - name: copy script install docker
    copy:
      src: ./docker.sh
      dest: ./docker.sh
  - name: clone project
    git:
      repo: https://github.com/utopios/tp-azure-terraform
      dest: /tp-azure-terraform
```

```
- name: connect to azurecr
  community.docker.docker_login:
    registry_url: acrihab.azurecr.io
    username: acrihab
    password: eMsoLSwJe5+5kclHnK11IAbsqAp+ShGoyuGLBKpJBh+ACRAbauCq

- name: copy java docker
  copy:
    src: /tp-azure-terraform/images/java17/Dockerfile
    dest: /tp-azure-terraform/src/{{item}}
  with_items:
    - ui
    - cart
    - orders

- name: copy nodejs docker
  copy:
    src: /tp-azure-terraform/images/nodejs/Dockerfile
    dest: /tp-azure-terraform/src/{{item}}
  with_items:
    - checkout

- name: Build an image and push
  community.docker.docker_image:
    build:
      path: /tp-azure-terraform/src/{{item}}
    name: acrihab.azurecr.io/{{item}}
    source: build
    push: true
  with_items:
    - ui
    - cart
    - orders
    - checkout
    - assets
    - catalog

- name: Get Aks Credentials
  azure_rm_aks_info:
    resource_group: "m2i-formation"
    client_id: "ac865e4d-92cd-43e9-a3ac-fb3e5dc483ef"
    secret: "Jv38Q~IKp-otYhUem.DVcL.UlsymJpW6CnWJccXO"
    name: "aks-ihab"
  register: aks_credentials

- name: create k8s ressources
  k8s:
    src: /tp-azure-terraform/k8s/{{item}}
    kubeconfig: {{aks_credentials.kubeconfig}}
  with_items:
    - resources.yml
```

Authentification basé sur mot de passe :

L'authentification basée sur un mot de passe

Avec l'authentification par mot de passe, un mot de passe aléatoire est créé pour vous. Si vous ne spécifiez pas de valeur pour le paramètre `--name`, un nom contenant un horodatage est créé pour vous. Vous devez spécifier un paramètre `--scopes`, car il n'a pas de valeur par défaut. Si vous préférez, vous pouvez définir

l'attribution de rôle ultérieurement en utilisant az role assignment create.

Azure CLI Copier Open Cloudshell

```
# Create a service principal with required parameter
az ad sp create-for-rbac --scopes /subscriptions/mySubscriptionID

# Create a service principal for a resource group using a preferred name a
az ad sp create-for-rbac --name myServicePrincipalName \
                         --role reader \
                         --scopes /subscriptions/mySubscriptionID/resource
```

Inscription d'application : Permet de se connecter uniquement à l'application -> réduit la surface d'attaque

Accueil > utopios | Applications d'entreprise > Applications d'entreprise | Toutes les applications > Parcourir la galerie Azure AD >

Inscrire une application ...

* Nom

Nom d'affichage côté utilisateur pour cette application (il peut être modifié ultérieurement).

demo-web

Types de comptes pris en charge

Qui peut utiliser cette application ou accéder à cette API ?

- (●) Comptes dans cet annuaire d'organisation uniquement (utopios uniquement - Locataire unique)
- ( ) Comptes dans un annuaire d'organisation (tout annuaire Azure AD - Multilocataire)
- ( ) Comptes dans un annuaire d'organisation (tout annuaire Azure AD - Multilocataire) et comptes Microsoft personnels (par exemple, Skype, Xbox)
- ( ) Comptes Microsoft personnels uniquement

Parcourir la galerie Azure AD ...

+ Créer votre propre application Des commentaires ?

La galerie d'applications Azure AD est un catalogue de milliers d'applications qui facilitent le déploiement et la configuration de l'authentification unique (SSO) et de l'approvisionnement automatisé des utilisateurs. Lors du déploi partir de la galerie d'applications, vous tirez parti des modèles prédéfinies pour connecter vos utilisateurs de manière plus sécurisée à leurs applications. Parcourez ou créez votre propre application ici. Si vous voulez publier une a développée dans la galerie Azure AD pour que d'autres organisations puissent la découvrir et l'utiliser, vous pouvez créer une demande à l'aide du processus décrit dans cet article.

Rechercher dans l'application Authentification unique : Tout Gestion du compte utilisateur : All Catégories : Tout

Plateformes cloud

Amazon Web Services (AWS)	Google Cloud Platform	Oracle	SAP
aws	Google Cloud		SAP

Ajouter une attribution ...

utopios

⚠ Les groupes ne sont pas disponibles pour l'attribut
pouvez affecter des utilisateurs individuels à l'appli

Utilisateurs

2 utilisateurs sélectionnés


2 utilisateurs sélectionnés.
Sélectionner un rôle
Default Access