



Terraform – MS Azure

Initiation à l'automatisation
4ArcTIC

2024-2025

Introduction :

Dans ce TP, nous allons apprendre à utiliser Terraform pour automatiser la création et la gestion de ressources sur Microsoft Azure.

Prérequis :

- Un compte **Azure for Students** actif.
- Une machine sous **Windows** avec une connexion Internet stable

Objectif :

L'objectif de cet atelier est de :

- ✓ Installer Terraform sur une machine Windows
- ✓ Installer et configurer la CLI Azure sur une machines Windows.
- ✓ Automatiser la création d'un ressource group
- ✓ Automatiser le déploiement d'une instance.

Partie 1 : Installation des outils

I- Installation de Terraform

Pour installer Terraform sur votre machine Windows vous devez :

1. Télécharger l'outil qui correspond à votre architecture matérielle à partir de [ce lien](#).
2. Extraire l'exécutable et l'enregistrer dans un emplacement donné (par exemple c:\Apps\terraform.exe)
3. Rajouter l'emplacement de l'exécutable terraform.exe à votre variable d'environnement path comme suit :
 - Sur votre panneau de configuration, allez vers system → system settings → Environment Variables.
 - Dans la section variables systèmes, cherchez la variable Path.
 - Cliquez sur le bouton modifier
 - Rajouter le chemin d'accès vers le fichier terraform.exe (par exemple c:\Apps\terraform.exe)

Vous pouvez suivre [cette vidéo](#).

4. Ouvrez **PowerShell** et vérifiez l'installation avec :

```
terraform -version
```

```
Administrateur - Windows PowerShell
PS C:\Windows\system32> terraform -version
Terraform v1.10.5
on windows_amd64
PS C:\Windows\system32>
```

Remarque : Si Terraform n'est pas reconnu, redémarrez votre machine.

II- Installation et configuration de Azure CLI

1. Téléchargez et installez **Azure CLI** depuis [ce lien](#).

The screenshot shows the 'Install or update' page for Azure CLI. On the left is a navigation menu with options like 'Overview', 'Install, update and run', 'Sign in', etc. The 'Install - Windows' option is selected. The main content area has a heading 'Install or update' followed by a paragraph explaining that MSI and ZIP are used for installation. Below this is an 'Important' note in a blue box stating that after installation, the user must close and reopen any active terminal window. Further down, there's a section for 'Microsoft Installer (MSI)' and 'Latest version', which includes two buttons: 'Latest MSI of the Azure CLI (32-bit)' and 'Latest MSI of the Azure CLI (64-bit)'.

2. Vérifiez l'installation avec :

```
az version
```

```
Administrateur : Windows PowerShell (x86)
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows
PS C:\Windows\system32> az version
{
  "azure-cli": "2.69.0",
  "azure-cli-core": "2.69.0",
  "azure-cli-telemetry": "1.1.0",
  "extensions": {}
}
PS C:\Windows\system32>
```

3. Connectez-vous à Azure :

az login

- Une page web s'ouvrira, connectez-vous avec votre **compte étudiant Azure**.
- Après connexion, les informations relatives à votre compte vont être affichées comme suit :

```
Sélection Administrateur : Windows PowerShell (x86)
PS C:\Windows\system32> az login
Select the account you want to log in with. For more information on login with Azure CLI, see https://go.microsoft.com/fwlink/?linkid=2271136

Retrieving tenants and subscriptions for the selection...

[Tenant and subscription selection]

No      Subscription name      Subscription ID      Tenant
-----
[1] *   Azure for Students  [REDACTED]          ESPRIT

The default is marked with an *; the default tenant is 'ESPRIT' and subscription is 'Azure for Students' [REDACTED].

Select a subscription and tenant (Type a number or Enter for no changes): 1

Tenant: ESPRIT
Subscription: Azure for Students [REDACTED]

[Announcements]
With the new Azure CLI login experience, you can select the subscription you want to use more easily. Learn more about it and its configuration at https://go.microsoft.com/fwlink/?linkid=2271236

If you encounter any problem, please open an issue at https://aka.ms/azclibug

[Warning] The login output has been updated. Please be aware that it no longer displays the full list of available subscriptions by default.

PS C:\Windows\system32>
```

4. Vérifiez que l'abonnement actif est bien Azure for Students :

```
PS C:\Windows\system32> az account show
{
  "environmentName": "AzureCloud",
  "homeTenantId": [REDACTED],
  "id": [REDACTED],
  "isDefault": true,
  "managedByTenants": [],
  "name": "Azure for Students",
  "state": "Enabled",
  "tenantDefaultDomain": "Esprit.tn",
  "tenantDisplayName": "ESPRIT",
  "tenantId": "604f1a96-cbe8-43f8-abbf-f8eaf5d85730",
  "user": {
    "name": [REDACTED],
    "type": "user"
  }
}
```

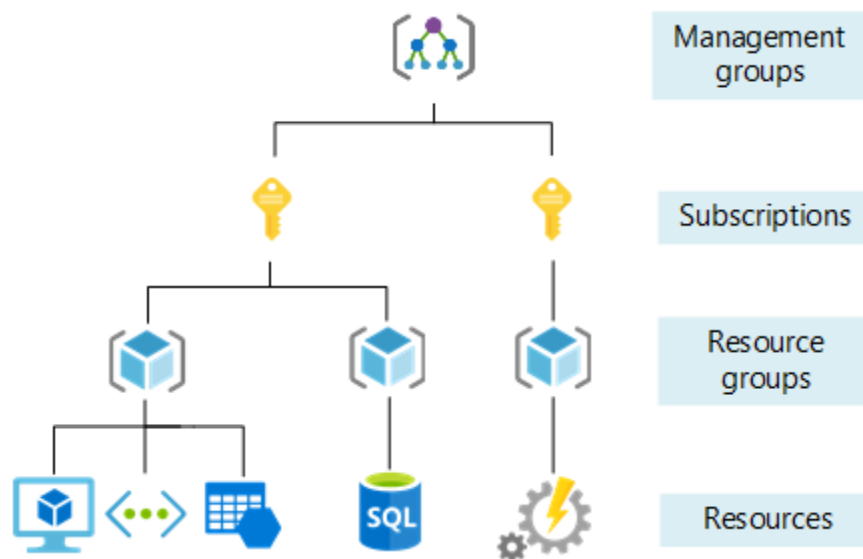
Si vous disposez de plusieurs souscriptions et vous voulez utiliser la souscription Azure for students, vous devez utiliser la commande suivante :

az account set --subscription "Azure for Students"

Get details about a specified subscription.

```
PS C:\Windows\system32> az account subscription list
Preview version of extension is disabled by default for extension installation, enabled for modules without stable versions.
Please run 'az config set extension.dynamic_install_allow_preview=true or false' to config it specifically.
The command requires the extension account. Do you want to install it now? The command will continue to run after the extension is installed. (Y/n): y
Run 'az config set extension.use_dynamic_install=yes_without_prompt' to allow installing extensions without prompt.
Command group 'account subscription' is experimental and under development. Reference and support levels: https://aka.ms/CLI_refstatus
[
  {
    "authorizationSource": "RoleBased",
    "displayName": "Azure for Students",
    "id": "/subscriptions/[REDACTED]",
    "state": "Enabled",
    "subscriptionId": "[REDACTED]",
    "subscriptionPolicies": {
      "locationPlacementId": "Public_2014-09-01",
      "quotaId": "AzureForStudents_2018-01-01",
      "spendingLimit": "0n"
    }
  }
]
PS C:\Windows\system32> az account subscription show
```

Avant de continuer, il est essentiel de savoir que les ressources créées et gérées par MS Azure sont organisées d'une manière hiérarchique à la manière d'un annuaire Active Directory. Pour pouvoir créer des ressources tel que les instances, espace de stockage, réseaux, volumes... , il est nécessaire de créer d'abord les ressources conteneurs comme expliquer ci-dessous :



Pour en savoir plus, consulter la [documentation](#) Microsoft Azure.

Partie 2 : Automatisation avec Terraform

I- Automatisation de la création d'un ressource group

Le groupe de ressources Azure est un conteneur logique dans lequel les ressources Azure telles que les applications Web, les bases de données et les comptes de stockage sont déployées et gérées. Dans cette partie du Lab, nous allons utiliser Terraform pour automatiser la création d'un groupe de ressources.

1. Création d'un projet Terraform :

```
PS C:\Windows\system32> mkdir terraform-azure

Répertoire : C:\Windows\system32

Mode                LastWriteTime         Length Name
----                -
d-----         22/02/2025   10:54             terraform-azure

PS C:\Windows\system32> cd terraform-azure
PS C:\Windows\system32\terraform-azure>
```

2. Dans ce répertoire, créez un fichier main.tf et ajoutez la configuration suivante :

```
terraform {
  required_version = ">=0.12"
  required_providers {
    azurearm = {
      source = "hashicorp/azurearm"
      version = "~>4.20"
    }
  }
}

provider "azurearm" {
  features {}
  subscription_id = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}

resource "random_pet" "rg-name" {
  prefix = var.resource_group_name_prefix
}

resource "azurearm_resource_group" "rg" {
  name     = random_pet.rg-name.id
  location = var.resource_group_location
}
```

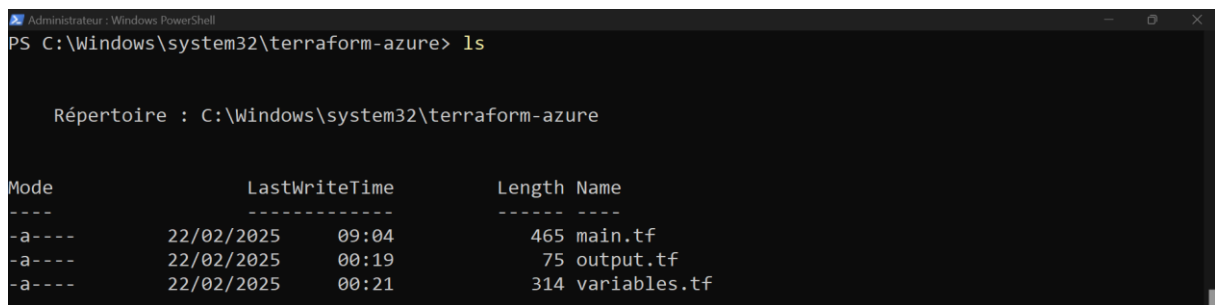
Remarque : Obtenez l'ID de votre abonnement avec la commande suivante : **az account show --query id -o tsv**, puis, remplacez xxxxxxxx... par la valeur retournée.

3. Pour cet exemple, le préfixe à utiliser pour les noms des ressources groupe ainsi que la localisation seront introduit comme variables grâce à l'utilisation du fichier variables.tf comme suit :

```
variable "resource_group_name_prefix" {
  default    = "rg"
  description = "Prefix of the resource group name that's combined with a random ID so name is unique in your Azure subscription."
}
variable "resource_group_location" {
  default    = "eastus"
  description = "Location of the resource group."
}
```

4. Enfin, pour configurer l'output à afficher à la fin de la création du resource group, nous allons utiliser le fichier output.tf comme ci-dessous :

```
output "resource_group_name" {
  value = azurerm_resource_group.rg.name
}
```

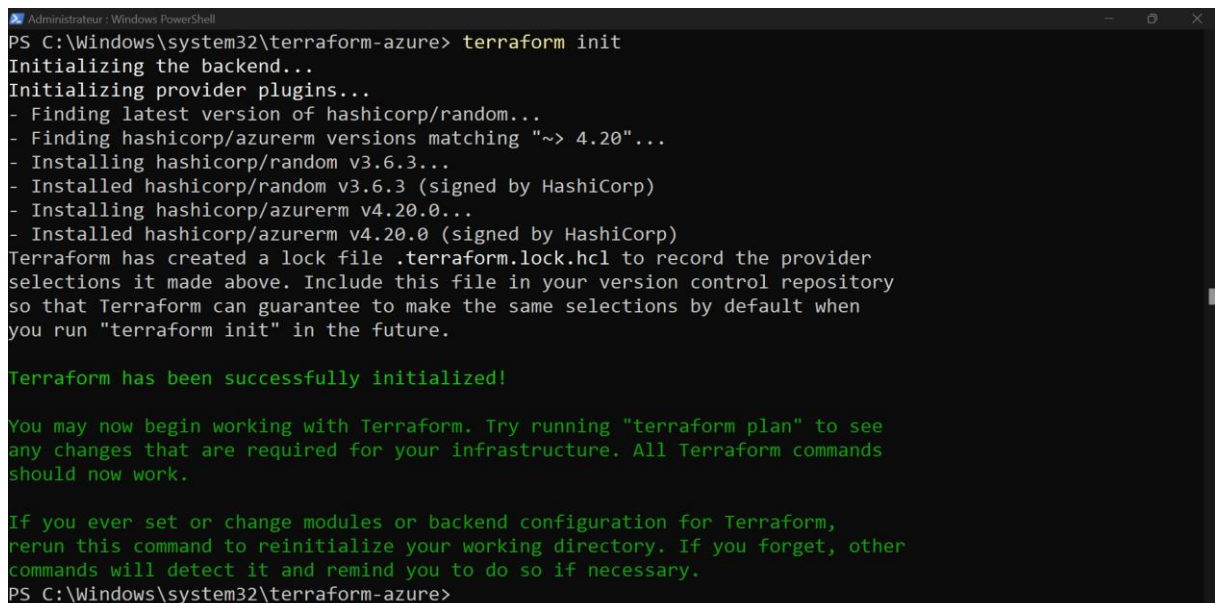


```
Administrateur : Windows PowerShell
PS C:\Windows\system32\terraform-azure> ls

Répertoire : C:\Windows\system32\terraform-azure

Mode                LastWriteTime         Length Name
----                -
-a----           22/02/2025    09:04         465 main.tf
-a----           22/02/2025    00:19          75 output.tf
-a----           22/02/2025    00:21         314 variables.tf
```

5. Nous allons par la suite initialiser Terraform avec la commande : **Terraform init**



```
Administrateur : Windows PowerShell
PS C:\Windows\system32\terraform-azure> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/random...
- Finding hashicorp/azurerm versions matching "~> 4.20"...
- Installing hashicorp/random v3.6.3...
- Installed hashicorp/random v3.6.3 (signed by HashiCorp)
- Installing hashicorp/azurerm v4.20.0...
- Installed hashicorp/azurerm v4.20.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\Windows\system32\terraform-azure>
```

```
PS C:\Windows\system32\terraform-azure> ls

Répertoire : C:\Windows\system32\terraform-azure

Mode                LastWriteTime         Length Name
----                -
d-----         22/02/2025    11:04             .terraform
-a----         22/02/2025    11:05         2233 .terraform.lock.hcl
-a----         22/02/2025    09:04         465 main.tf
-a----         22/02/2025    00:19          75 output.tf
-a----         22/02/2025    00:21         314 variables.tf
```

6. Pour formater les fichiers de configuration dans l'espace de travail, nous allons utiliser la commande : **terraform fmt**

```
Administrateur : Windows PowerShell
PS C:\Windows\system32\terraform-azure> terraform fmt
PS C:\Windows\system32\terraform-azure>
```

7. Pour faire la vérification syntaxique des fichiers de configuration, nous allons utiliser la commande : **terraform validate**

```
PS C:\Windows\system32\terraform-azure> terraform validate
Success! The configuration is valid.
PS C:\Windows\system32\terraform-azure>
```

8. Il est maintenant possible de passer à la phase de génération du plan avec la commande : **terraform plan**

```
Administrateur : Windows PowerShell
PS C:\Windows\system32\terraform-azure> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the follow
symbols:
+ create

Terraform will perform the following actions:

# azurerms_resource_group.rg will be created
+ resource "azurerms_resource_group" "rg" {
  + id          = (known after apply)
  + location    = "eastus"
  + name        = (known after apply)
}

# random_pet.rg-name will be created
+ resource "random_pet" "rg-name" {
  + id          = (known after apply)
  + length      = 2
  + prefix      = "rg"
  + separator    = "-"
}

Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ resource_group_name = (known after apply)

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run
"terraform apply" now.
PS C:\Windows\system32\terraform-azure>
```


9. Enfin pour appliquer ces modifications, la commande suivante doit être exécutée :
terraform apply

```
Administrateur : Windows PowerShell
PS C:\Windows\system32\terraform-azure> terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
+ create

Terraform will perform the following actions:

# azurerm_resource_group.rg will be created
+ resource "azurerm_resource_group" "rg" {
+   id       = (known after apply)
+   location = "eastus"
+   name     = (known after apply)
+ }

# random_pet.rg-name will be created
+ resource "random_pet" "rg-name" {
+   id       = (known after apply)
+   length   = 2
+   prefix   = "rg"
+   separator = "-"
+ }

Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ resource_group_name = (known after apply)

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

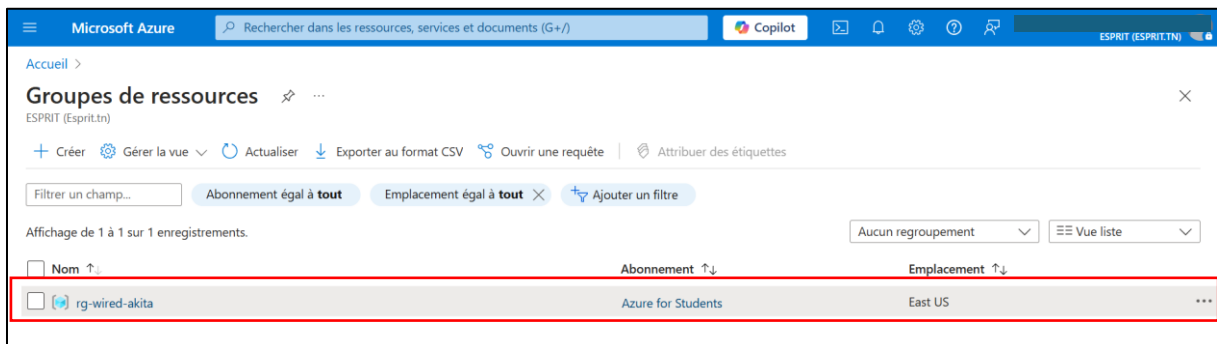
Enter a value: yes

random_pet.rg-name: Creating...
random_pet.rg-name: Creation complete after 0s [id=rg-lucky-vervet]
azurerm_resource_group.rg: Creating...
azurerm_resource_group.rg: Still creating... [10s elapsed]
azurerm_resource_group.rg: Creation complete after 11s [id=/subscriptions/ad45eb7f-73f3-4c22-82f3-0193c2173862/resourceGroups/]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:
resource_group_name = "rg-lucky-vervet"
PS C:\Windows\system32\terraform-azure>
```

Il est possible de vérifier que le ressource group a bien été créer à travers le portail Azure :



Il est aussi possible de lister les ressources créées à partir de Terraform avec la commande :
terraform state list

```
Administrateur : Windows PowerShell
PS C:\Windows\system32\terraform-azure> terraform state list
azurerm_resource_group.rg
random_pet.rg-name
PS C:\Windows\system32\terraform-azure>
```

10. Enfin, pour supprimer les ressources que nous venons de créer, nous allons utiliser la commande : **terraform destroy**

```
Administrateur : Windows PowerShell
PS C:\Windows\system32\terraform-azure> terraform destroy
random_pet.rg-name: Refreshing state... [id=rg-lucky-vervet]
azurerm_resource_group.rg: Refreshing state... [id=/subscriptions/ad45eb7f-73f3-4c22-82f3-0193c2173062/resourceGroups/rg-lucky-vervet]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
- destroy

Terraform will perform the following actions:

# azurerm_resource_group.rg will be destroyed
- resource "azurerm_resource_group" "rg" {
  id       = "/subscriptions/ad45eb7f-73f3-4c22-82f3-0193c2173062/resourceGroups/rg-lucky-vervet" -> null
  location = "eastus" -> null
  name     = "rg-lucky-vervet" -> null
  tags    = {} -> null
} # (1 unchanged attribute hidden)

# random_pet.rg-name will be destroyed
- resource "random_pet" "rg-name" {
  id       = "rg-lucky-vervet" -> null
  length   = 2 -> null
  prefix   = "rg" -> null
  separator = "-" -> null
}

Plan: 0 to add, 0 to change, 2 to destroy.

Changes to Outputs:
- resource_group_name = "rg-lucky-vervet" -> null

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

azurerm_resource_group.rg: Destroying... [id=/subscriptions/ad45eb7f-73f3-4c22-82f3-0193c2173062/resourceGroups/rg-lucky-vervet]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/ad45eb7f-73f3-4c22-82f3-0193c2173062/resourceGroups/rg-lucky-vervet, 10s elapsed]
azurerm_resource_group.rg: Destruction complete after 20s
random_pet.rg-name: Destroying... [id=rg-lucky-vervet]
random_pet.rg-name: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
PS C:\Windows\system32\terraform-azure>
```

II- Automatisation d'une Machine Virtuelle

1. Création d'un projet Terraform-VM :

```
Administrateur : Windows PowerShell
PS C:\Windows\system32> mkdir terraform-VM

Répertoire : C:\Windows\system32

Mode                LastWriteTime         Length Name
----                -
d-----          22/02/2025   11:20         terraform-VM

PS C:\Windows\system32> cd terraform-VM
PS C:\Windows\system32\terraform-VM>
```

2. Ajoutez cette configuration à main.tf pour créer une VM :

```
# Configure the Microsoft Azure Provider
terraform {
  required_providers {
    azurerm = {
      source = "hashicorp/azurerm"
      version = "~>4.20"
    }
  }
}
```

```

provider "azurerm" {
  features {}
  subscription_id = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
# Create a resource group if it doesn't exist
resource "azurerm_resource_group" "myterraformgroup" {
  name     = "myResourceGroupTerraform"
  location = "eastus"
  tags = {
    environment = "Terraform Demo"
  }
}
# Create virtual network
resource "azurerm_virtual_network" "myterraformnetwork" {
  name            = "myVnet"
  address_space   = ["10.0.0.0/16"]
  location        = "eastus"
  resource_group_name = azurerm_resource_group.myterraformgroup.name
  tags = {
    environment = "Terraform Demo"
  }
}
# Create subnet
resource "azurerm_subnet" "myterraformsubnet" {
  name                = "mySubnet"
  resource_group_name = azurerm_resource_group.myterraformgroup.name
  virtual_network_name = azurerm_virtual_network.myterraformnetwork.name
  address_prefixes    = ["10.0.1.0/24"]
}
# Create public IPs
resource "azurerm_public_ip" "myterraformpublicip" {
  name            = "myPublicIP"
  location        = "eastus"
  resource_group_name = azurerm_resource_group.myterraformgroup.name
  allocation_method = "Static" # Changé de "Dynamic" à "Static"
  sku             = "Standard" # Assurez-vous que la SKU est définie sur "Standard"
  tags = {
    environment = "Terraform Demo"
  }
}
# Create Network Security Group and rule
resource "azurerm_network_security_group" "myterraformnsg" {
  name            = "myNetworkSecurityGroup"
  location        = "eastus"
  resource_group_name = azurerm_resource_group.myterraformgroup.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 = "*"
  }
  tags = {
    environment = "Terraform Demo"
  }
}
# Create network interface
resource "azurerm_network_interface" "myterraformnic" {
  name          = "myNIC"
  location      = "eastus"
  resource_group_name = azurerm_resource_group.myterraformgroup.name
  ip_configuration {
    name                = "myNicConfiguration"
    subnet_id           = azurerm_subnet.myterraformsubnet.id
    private_ip_address_allocation = "Dynamic"
    public_ip_address_id = azurerm_public_ip.myterraformpublicip.id
  }
  tags = {
    environment = "Terraform Demo"
  }
}
# Connect the security group to the network interface
resource "azurerm_network_interface_security_group_association" "example" {
  network_interface_id = azurerm_network_interface.myterraformnic.id
  network_security_group_id = azurerm_network_security_group.myterraformnsg.id
}
# Generate random text for a unique storage account name
resource "random_id" "randomId" {
  keepers = {
    # Generate a new ID only when a new resource group is defined
    resource_group = azurerm_resource_group.myterraformgroup.name
  }
  byte_length = 8
}
# Create storage account for boot diagnostics
resource "azurerm_storage_account" "mystorageaccount" {
  name          = "diag${random_id.randomId.hex}"
  resource_group_name = azurerm_resource_group.myterraformgroup.name
  location      = "eastus"
  account_tier   = "Standard"
  account_replication_type = "LRS"
  tags = {
    environment = "Terraform Demo"
  }
}
# Create (and display) an SSH key
resource "tls_private_key" "example_ssh" {

```

```

algorithm = "RSA"
rsa_bits = 4096
}
output "tls_private_key" {
  value = tls_private_key.example_ssh.private_key_pem
  sensitive = true
}
# Create virtual machine
resource "azurerm_linux_virtual_machine" "myterraformvm" {
  name = "myVM"
  location = "eastus"
  resource_group_name = azurerm_resource_group.myterraformgroup.name
  network_interface_ids = [azurerm_network_interface.myterraformnic.id]
  size = "Standard_DS1_v2"
  os_disk {
    name = "myOsDisk"
    caching = "ReadWrite"
    storage_account_type = "Premium_LRS"
  }
  source_image_reference {
    publisher = "Canonical"
    offer = "UbuntuServer"
    sku = "18.04-LTS"
    version = "latest"
  }
  computer_name = "myvm"
  admin_username = "azureuser"
  disable_password_authentication = true
  admin_ssh_key {
    username = "azureuser"
    public_key = tls_private_key.example_ssh.public_key_openssh
  }
  boot_diagnostics {
    storage_account_uri = azurerm_storage_account.mystorageaccount.primary_blob_endpoint
  }
  tags = {
    environment = "Terraform Demo"
  }
}

```

Remarque : Obtenez l'ID de votre abonnement avec la commande suivante : **az account show --query id -o tsv**, puis, remplacez xxxxxxxxxx... par la valeur retournée.

```

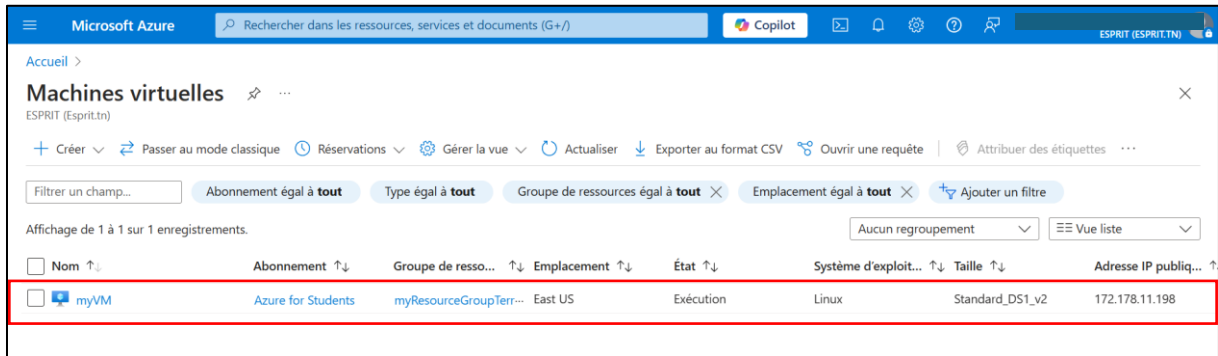
PS C:\Windows\system32\terraform-VM> ls

Répertoire : C:\Windows\system32\terraform-VM

Mode                LastWriteTime         Length Name
----                -
-a----          22/02/2025   10:09         5064 main.tf

```

3. Faire les commandes terraform **init**, **fmt**, **validate**, **plan** et **apply**.
4. Vérifiez la création de la VM à travers le portail Azure.



The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with the Microsoft Azure logo, a search bar, and various icons. Below the navigation bar, the page title is 'Machines virtuelles' (Virtual Machines) for the subscription 'ESPRIT (Esprit.tn)'. There are several action buttons like 'Créer', 'Passer au mode classique', 'Réservations', 'Gérer la vue', 'Actualiser', 'Exporter au format CSV', 'Ouvrir une requête', and 'Attribuer des étiquettes'. Below these, there are filter buttons: 'Abonnement égal à tout', 'Type égal à tout', 'Groupe de ressources égal à tout', and 'Emplacement égal à tout'. A table of virtual machines is displayed below the filters. The table has columns for 'Nom', 'Abonnement', 'Groupe de resso...', 'Emplacement', 'État', 'Système d'exploit...', 'Taille', and 'Adresse IP public...'. The first row of the table is highlighted with a red border and contains the following data: 'myVM', 'Azure for Students', 'myResourceGroupTerr...', 'East US', 'Exécution', 'Linux', 'Standard_DS1_v2', and '172.178.11.198'.

Nom	Abonnement	Groupe de resso...	Emplacement	État	Système d'exploit...	Taille	Adresse IP public...
myVM	Azure for Students	myResourceGroupTerr...	East US	Exécution	Linux	Standard_DS1_v2	172.178.11.198