



# Infrastructure as code (IAC) Simplifiez-vous la vie (et l'IT)

<https://github.com/randb13/GAB2019>

Un grand merci à Lucas Nassibou pour son aide 😊

## Table des matières

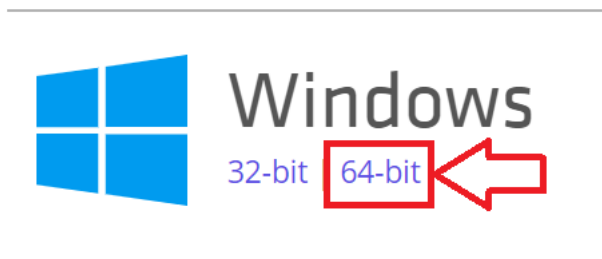
1	Installation de Terraform .....	2
2	Création de l'authentification sur Azure .....	2
3	Création du provider Terraform .....	8
4	Création du Backend Terraform .....	8
5	Bonne Pratiques Terraform .....	11
6	Premier Script IAC (Infra As Code) .....	11

# AZURE : TERRAFORM

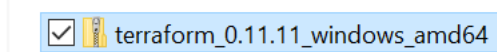
## 1 Installation de Terraform

Pour installer Terraform , allez sur le lien suivant : <https://www.terraform.io/downloads.html>

Puis cliquez sur : « Windows 64 bit »

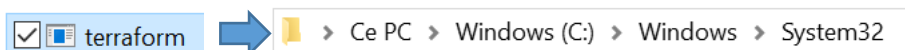


Un fichier .zip est alors télécharger :



Dézipper le fichier .

Puis dans le dossier : « terraform\_0.11.11\_windows\_amd64 », copier le fichier : « terraform.exe » vers le répertoire : « C:\Windows\System32 » :



Enfin ouvrez une commande dos ou Windows Powershell et entrez la commande suivante : terraform -v

```
PS C:\Users\Inassibou\Documents\Terraform-Formation> terraform -v
Terraform v0.11.11
```

Si le numéro de version s'affiche , terraform est bien installé .

## 2 Création de l'authentification sur Azure

Une fois Terraform installé , il faut le configurer pour qu'il puisse interagir avec les API d'Azure .

Pour cela nous allons créer un APP ID :

- **Via PowerShell :**
  - Prérequis : avoir installer azure-cli : <https://docs.microsoft.com/fr-fr/cli/azure/install-azure-cli-windows?view=azure-cli-latest>

# AZURE : TERRAFORM

Ouvrez Windows PowerShell et lancez les commandes suivantes :

Connecter vous via la commande :

```
az login
```

Si vous avez plusieurs subscription active commande pour lister l'ensemble des subscriptions avec leur IDs:

```
az account list --output table
```

Commande pour sélectionnez la subscription que vous souhaitez utiliser :

```
az account set --subscription "SubscriptionID"
```

Cette commande permet de récupérer l'id de subscription :

```
az account show --query "{subscriptionId:id, tenantId:tenantId}"
```

```
az ad sp create-for-rbac --name="terraform" --role="Contributor" --scopes="/subscriptions/VOTRE_SUBSCRIPTION_ID"
```

Cette commande va vous retourner des valeurs qu'il va falloir conserver :

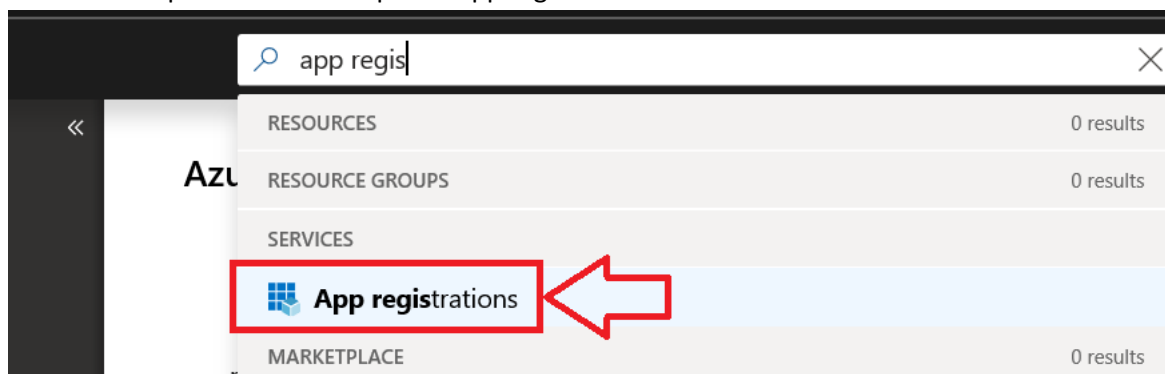
```
PS C:\Users\lnassibou> az ad sp create-for-rbac --role="Contributor" --scopes="/subscriptions/154d251f-6932-4035-b012-62c672c72154"
{
  "appId": "cc5530c8-f9b8-4d0c-b34a-1a5d2650c0b4",
  "displayName": "azure-cli-2019-03-05-08-58-40",
  "name": "http://azure-cli-2019-03-05-08-58-40",
  "password": "x!cf00f0-ucje-43a1-uta-3003f0d03f",
  "tenant": "9a5860f2-2e30-4b45-9fd9-e0ad59c80197"
}
```

Nous avons besoins des informations « appId », « password », « tenant » et l'id de subscription.

**Remarque : le mot de passe n'apparaîtra qu'une et une seule fois, il n'est pas visible par la suite.**

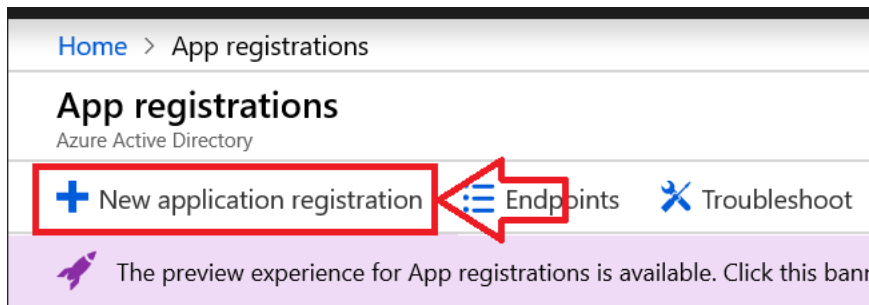
- **Via Le portail :**

Dans le champ de recherche tapez : «App registrations »



Puis cliquez sur «New Application Registration »

# AZURE : TERRAFORM






Entrez un nom puis une URL de connexion (l'url de connexion n'a pas d'importance, elle n'a pas besoin d'être valide) :

Puis cliquez sur « Create »

Puis sélectionnez ce nouvel App et cliquez sur « Settings » sur le menu de droite :

# AZURE : TERRAFORM

**terraform**  
Registered app

 Settings  Manifest  Delete

Display name  
terraform

Application type  
Web app / API

Home page  
<https://terraform>

Application ID  
509a95e5-8402-4e1b-b9c1


Object ID  
1eef928b-cba9-4049-8138

Managed application in lo  
[terraform](#)


⌵


Puis cliquez sur « keys »


**Settings**

 Filter settings


**GENERAL**


 Properties

 Reply URLs

 Owners

**API ACCESS**

 Required permissions

 Keys

Puis entrez une description , une date d'expiration , le champ value sera remplis automatiquement lorsque l'on clique sur « save »

## Passwords

DESCRIPTION	EXPIRES	VALUE	
terraform	31/12/2299	pJ/o/AOvXVhFnQUcCxo9pFUKBmQuGYI6rcsxnhXWoE=	...

**Prenez soin de noter le mot de passe car il ne sera visible qu'une et une seule fois !!**

Puis dans le champ de recherche tapez : « Subscriptions »

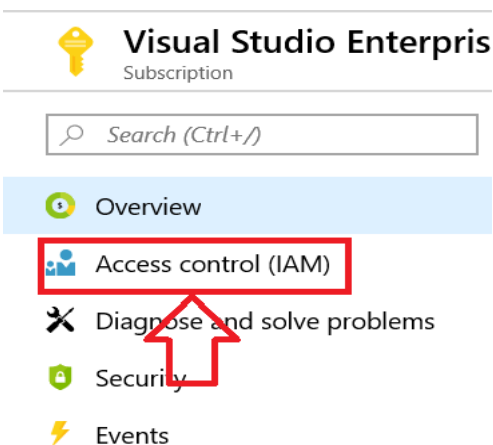
# AZURE : TERRAFORM



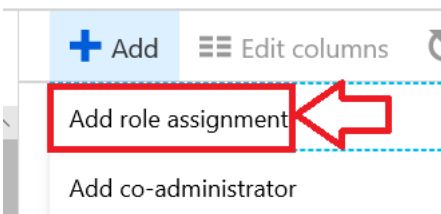
Cliquez sur la souscription de votre choix :



Puis sur le menu de gauche , cliquez sur « Access control (IAM) » :



Puis cliquez sur « Add » et sélectionnez «Add rôle assignment »



Puis sélectionnez « contributor » dans le champ rôle, puis sélectionnez « Azure AD user, group or service principal » dans le champ Assign access to et enfin dans le champ select tapez : « terraform »

# AZURE : TERRAFORM

Role ⓘ

Contributor



Assign access to ⓘ

Azure AD user, group, or service principal



Select ⓘ

terr



Selected members:



terraform

Remove

Save

Discard

Cliquez sur « save »

Et voilà le service terraform a son accès aux API Azure.

Créons notre premier script !

# AZURE : TERRAFORM

## 3 Création du provider Terraform

Une fois que nous avons créer notre service principale sur Azure, nous allons créer le fichier d'authentification sur Terraform. Ouvrez votre IDE ( Visual Studio Code par exemple) et créer un fichier : **provider.tf**  
Puis insérer les informations d'authentification comme suit :

```
provider.tf x
1 provider "azurerm" {
2   subscription_id = 
3   client_id= 
4   client_secret = 
5   tenant_id = 
6 }
7
```

## 4 Création du Backend Terraform

Les bonnes pratiques Terraform , nous conseille de déposer le fichier :terraform.tfstate sur un storage . Ce fichier contient l'inventaire et l'état de l'infrastructure , à chaque exécution de terraform ce fichier est « lock » . Pour pouvoir le déposer sur notre storage, nous allons déjà en créer un :

Sur le portail Azure sélectionnez : Storage Account

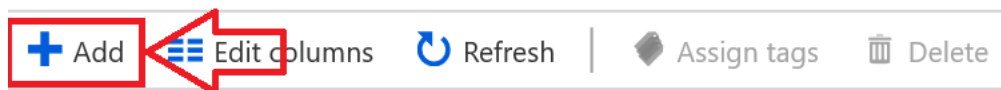


Storage  
accounts

Puis cliquez sur « Add »

**Storage accounts**

PROJETLYS



Ensuite entrez les informations de création du storage :



# AZURE : TERRAFORM

[Basics](#) [Advanced](#) [Tags](#) [Review + create](#)

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more](#)

## PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

\* Subscription

\* Resource group  [Create new](#)

## INSTANCE DETAILS

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

\* Storage account name

\* Location

Performance ☒ Standard ☐ Premium

Account kind

Replication

Access tier (default) ☐ Cool ☒ Hot

[Review + create](#)

[Previous](#)


[Next : Advanced >](#)

Puis sélectionnez le storage account nouvellement créer :

 <a href="#">procedureterraform</a>	Storage account	StorageV2
---	-----------------	-----------

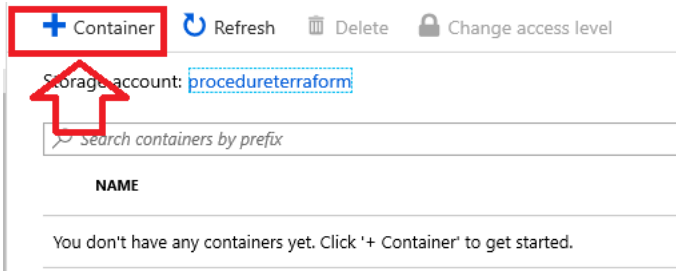
Dans les différents services , sélectionnez « Blobs »

Services

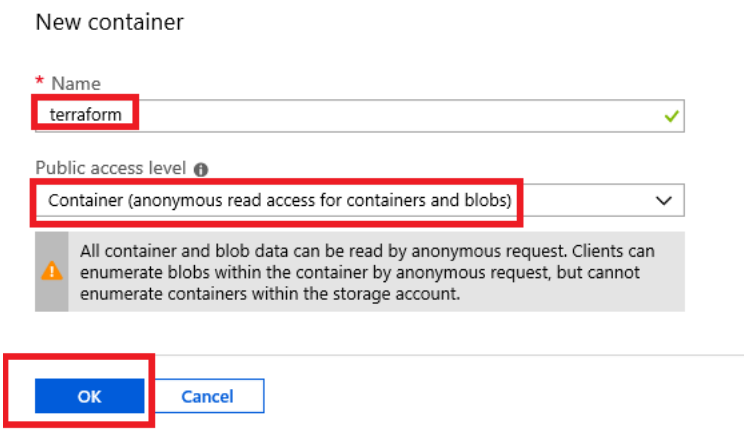
**Blobs**  
REST-based object storage for unstructured data  
[Learn more](#)

Ensuite cliquez sur « container »

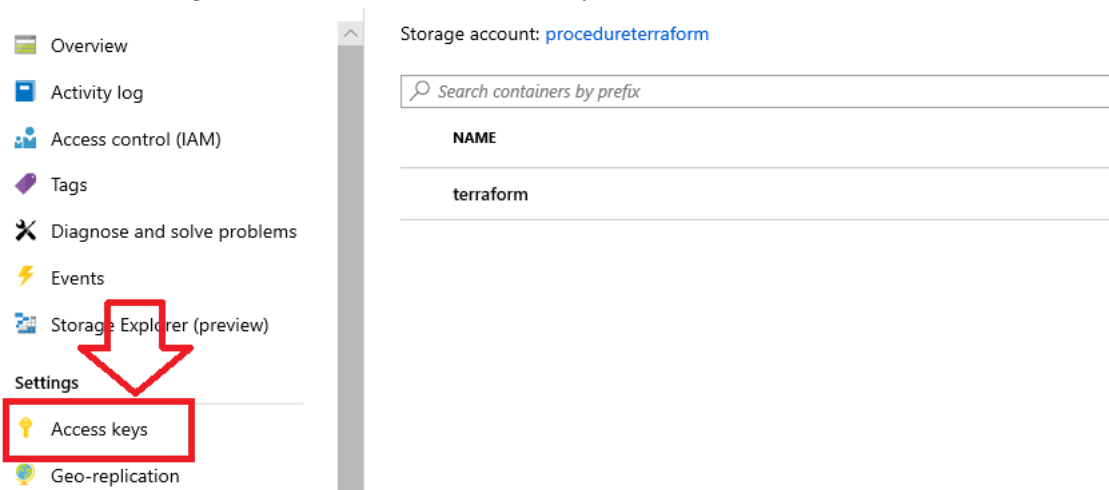
# AZURE : TERRAFORM



Puis entrez un nom pour le nouveau container et sélectionnez en Public access level : Container



Dans le menu de gauche, sélectionner « Access keys »



Puis copier une des access key disponible :

key1   
Key  
vJd1pvEF7WTBix8bMPUotvoMidVM3LYfhLiZnKnR7cofSl8qtZvGEnuqL/g15Bd/6z1wyvXPnz2/tUozQ==



# AZURE : TERRAFORM

Maintenant dans votre IDE, créer le fichier : **backend.tfvars** et insérez les informations comme suit :

```
backend.tfvars x
1 terraform {
2   0 references
3   backend "azurerm" {
4     storage_account_name = 
5     container_name       = 
6     key                   = "terraform.tfstate"
7     access_key            = 
8   }
```

La valeur **key** contient le nom de fichier qui sera déposer sur le storage.

## 5 Bonne Pratiques Terraform

- Le code doit être versionné . ( GIT , SVN ...).
- Le fichier tfstate doit être mis sur un storage en ligne .
- L'arborescence doit être structurée par des répertoires.

## 6 Premier Script IAC (Infra As Code)

Nous avons déjà deux fichiers terraform : **provider.tf** et **backend.tf** , Il faut initialiser ces deux fichiers avant de commencer notre premier script terraform, pour cela ouvrez une commande DOS , puis positionnez-vous dans le répertoire qui contient les deux fichiers et lancer la commande :

**terraform init**

Maintenant nous sommes prêts à déployer notre première infrastructure.

Vous pouvez maintenant créer votre fichier de définition (ex : **first.tf**)

Un exemple est disponible ici : <https://github.com/randb13/GAB2019>

Pour visualiser le déploiement : **terraform plan**

Pour lancer le déploiement : **terraform apply**

Pour aller plus loin , vous trouverez toutes les informations via :

<https://www.terraform.io/docs/providers/azurerm/index.html#>

Tous les modules sont recensés sur le menu de gauche ☺