

# TP2: Terraform

## CONFIG AWS

### Etape 1: configuration IAM coté AWS

Sur votre compte aws:

- Nouvelle stratégie IAM, autorisant un access a EC2: Utilisez l'éditeur JSON avec ces informations:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    }
  ]
}
```

- Ajoutez cette nouvelle stratégie sur le role deja existant : Il y a maintenant un role avec deux strategies, une pour S3 et une pour EC2

- Ajoutez un nouvel utilisateur dans IAM, lui rajouter les droits super admin et un accès via la programmation. AWS va vous donner deux information, acces\_key et private\_key, il faut bien les noter !!!

### Etape 2: Nettoyage du compte

Assurez vous de ne plus avoir de security groups de serveur EC2 ou de bucket S3 sur votre compte aws

### Etape 3: Me donner votre ID de compte

En haut a droite, ou dans la facturation a vous de voir !

# CONFIG TERRAFORM

## Etape 1: Prise en main du tool

(<https://github.com/antoiner77/tp-cloud-assr>)

- Terraform a 3 commande basique. Ouvrez un terminal dans le dossier terraform comprenant un fichier en .tf
- \* terraform init -> Cette commande permet de générer la configuration de base et de charger les plugin
- \* terraform plan -> Permet de voir les modifications en attentes
- \* terraform apply -> Permet d'appliquer les changements

Dans le fichier tp.tf, ligne 3 et 4 rajoutez les clefs obtenues a l'étape AWS lors de la creation de l'utilisateur

## Creation du security group

Objectif: Modifier le fichier tp.tf pour lui faire créer un security group:

2 règles IN: ssh sur l'ip de l'iut et http en publique

1 règle OUT: Tout ouvert

[https://www.terraform.io/docs/providers/aws/r/security\\_group.html](https://www.terraform.io/docs/providers/aws/r/security_group.html)

Pour tester les modification, il faut rentrer les 2 premieres commandes plus haut. Faites la 3eme si tout est bon !

Aspect du fichier:

```

1  provider "aws" {
2    region = "us-west-2"
3    access_key = "AKIAI44QH8DHBEXAMPLE"
4    secret_key = "wJalrXU3WhFMbZdC441sA2zT3r7e4MFUdYvDQJ1"
5  }
6
7  resource "aws_security_group" "tg-sg-a" {
8    name = "tg-a-tout-sg-a"
9    description = "tg-a-tout-sg-a"
10    ingress {
11      from_port = "80"
12      to_port = "80"
13      protocol = "tcp"
14      cidr_blocks = ["0.0.0.0/0"]
15    }
16    ingress {
17      from_port = 80
18      to_port = 80
19      protocol = "tcp"
20      cidr_blocks = ["19.228.186.4/32"]
21    }
22    egress {
23      from_port = 0
24      to_port = 0
25      protocol = "-1"
26      cidr_blocks = ["0.0.0.0/0"]
27    }
28  }

```

---

## Creation du bucket s3

Rajoutez le bloc suivant après le premier bloc de code:

```
variable "Bucket_Name" {  
  type    = "string"  
  default = « MON_BUCKET »  
}
```

Il nous permettra de définir une variable pour le nom du bucket (pensez a changer MON\_BUCKET par un mot aléatoire !)

Ensuite a la suite de votre fichier tp.tf, ajoutez le bloc permettant de créer le bucket s3:

```
resource "aws_s3_bucket" "Bucket" {  
  bucket = "${var.Bucket_Name}"  
  acl    = "private"  
}
```

Pour tester les modification, il faut rentrer la commande 2 indiquée plus haut. Faites la 3eme si tout est bon !

---

## Reste du TP en autonomie

Nous voulons maintenant que Terraform nous fabrique un serveur ec2 nous voulons ajouter a ce serveur notre security group et nous voulons aussi y ajouter notre role IAM

En plus de cela nous voulons deux Tag:

Name -> Un nom au hasard

Et enfin

Bucket qui prendra en valeur la variable du nom de bucket (Inspirez vous du bloc pour créer le bucket s3)

Aide:

Pour le security group, utilisez cette ligne:

```
security_groups = [ "${aws_security_group.Tp-secu.name}" ]
```

Pour tester les modification, il faut rentrer la commande 2 indiquée plus haut. Faites la 3eme si tout est bon !

---

## Reste du TP en autonomie 2eme partie

Nous voulons maintenant que terraform nous rajoute un load balancer ( resource « aws\_elb » sur terraform pensez a ouvrir la documentation en ligne )

Aide configuration:

```
availability_zones = ["eu-west-1a","eu-west-1b","eu-west-1c"]  
target             = "HTTP:80/index.php"  
instances          = ["${aws_instance.Tp-Instance.id}"]
```

Pour tester les modification, il faut rentrer la commande 2 indiquée plus haut. Faites la 3eme si tout est bon !

---

## Reste du TP en autonomie 3eme partie

Nous voulons 2 serveur ec2 et non un seul ! Basez vous sur ce lien pour modifier le bloc créant le serveur:

<https://blog.gruntwork.io/terraform-tips-tricks-loops-if-statements-and-gotchas-f739bbae55f9>

Aide:

La ligne du Tag Name devient:

Name = "Tp-Instance-\${count.index}"

Pensez aussi a modifier le load balancer avec cette configuration:

```
instances          = ["${aws_instance.Tp-Instance.*.id}"]
```

Pour tester les modification, il faut rentrer la commande 2 indiquée plus haut. Faites la 3eme si tout est bon !

Fin ;)