

laC avec  
Terraform / OpenTofu  
et Ansible

# Terraform / OpenTofu : Qu'est-ce que c'est ?

Terraform est :

- un outil open-source d'Infrastructure as Code (IaC)
- créé par HashiCorp en 2014
- permet de définir et de gérer une infrastructure en utilisant un langage déclaratif.
- changement de licence en août 2023 : MPL > BUSL
- OpenTofu : fork de la communauté supporté par la Linux Foundation

# Principales caractéristiques

- Syntaxe déclarative pour définir l'infrastructure
- Prise en charge de multiples fournisseurs et services cloud
- Compatible avec le contrôle de version et la collaboration
- Planification et application des changements de manière prévisible

# Pourquoi utiliser Terraform ?

Terraform permet aux équipes DevOps d'automatiser le provisionnement de l'infrastructure, d'améliorer la cohérence et de gérer efficacement des environnements multi-cloud complexes.

Il favorise les pratiques d'infrastructure as code, améliorant l'évolutivité et réduisant les erreurs humaines.

# Autres outils de déploiement

- OpenTofu : Un fork communautaire de Terraform, maintenu par la Linux Foundation
- Pulumi : Outil IaC utilisant des langages de programmation courants (Javascript, Python, Go, C#, YAML)
- AWS Cloud Formation
- Azure Resource Manager
- Google Cloud Deployment Manager

# Ansible : Qu'est-ce que c'est ?

Ansible est :

- un outil d'automatisation IT open-source
- simplifie la gestion de configuration
- le déploiement d'applications
- l'orchestration des tâches.

# Caractéristiques clés d'Ansible

- Sans agent (Agentless)
- Utilise SSH pour la communication
- Configuration basée sur YAML
- Facile à apprendre et à utiliser
- Extensible avec des modules

# Composants principaux

- Playbooks: Scripts d'automatisation
- Inventaire: Liste des hôtes gérés
- Modules: Unités de travail dans Ansible
- Rôles: Réutilisation et organisation du code



# Démonstration

## Etape 0 :

- Créer un compte AWS avec votre adresse EFREI
- Les comptes bénéficient par défaut du “[Free Tier](#)” pendant 1 an qui vous permet d’avoir accès un certain nombre de ressources gratuitement chaque mois. Le détail est sur la page.

# Démonstration

Etape 0.1 :

- Créer une clé SSH pour les instances qui sera utilisée pour s'y connecter :  
[https://eu-west-3.console.aws.amazon.com/ec2/home?region=eu-west-3#Key Pairs:](https://eu-west-3.console.aws.amazon.com/ec2/home?region=eu-west-3#KeyPairs:)

Key pairs | EC2 | eu-west-3

https://eu-west-3.console.aws.amazon.com/ec2/home?region=eu-west-3#KeyPairs:myKey

aws

Services

Search

[Option+S]

Paris

AkaEFREI

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

▼ Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

▼ Load Balancing

Load Balancers

Target Groups

Trust Stores [New](#)

Key pairs (1) Info

↻

Actions ▼

Create key pair

Find Key Pair by attribute or tag

< 1 >

| <input type="checkbox"/> | Name  | Type | Created                |
|--------------------------|-------|------|------------------------|
| <input type="checkbox"/> | myKey | rsa  | 2024/10/14 19:20 GMT+2 |

CloudShell

Feedback

Privacy

Terms

Cookie preferences

© 2024, Amazon Web Services, Inc. or its affiliates.

Create key pair | EC2 | eu-west-3

+

← → ↺

https://eu-west-3.console.aws.amazon.com/ec2/home?region=eu-west-3#CreateKeyPair

☆

📧 👤 📌 🔄 ☰

aws

Services

Search

[Option+S]

📺 🔔 ? ⚙️

Paris ▼

AkaEFREI ▼

☰

Name

myKey

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

Info

☒ RSA

☐ ED25519

Private key file format

☒ .pem

For use with OpenSSH

☐ .ppk

For use with PuTTY

Tags - optional

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Create key pair

📺 CloudShell Feedback

Privacy Terms Cookie preferences

© 2024, Amazon Web Services, Inc. or its affiliates.

# Démonstration

Après cette étape, un fichier “myKey.pem” sera téléchargé automatiquement.

Il vous servira à vous connecter via SSH aux instances que vous allez créer.

ATTENTION aux droits d'accès à ce fichier, pour l'utiliser correctement, vous devrez utiliser la commande suivante :

```
chmod 600 myKey.pem
```

Ce fichier sera utilisé par Ansible plus tard dans la démonstration

# Démonstration

Etape 0.2 :

- Créer une ACCESS KEY qui permettra à Terraform de se connecter au compte AWS :

[https://us-east-1.console.aws.amazon.com/iam/home?region=eu-west-3#/security\\_credentials](https://us-east-1.console.aws.amazon.com/iam/home?region=eu-west-3#/security_credentials)

Dashboard | IAM | Global

Security credentials | IAM | Glob: X

+

https://us-east-1.console.aws.amazon.com/iam/home?region=eu-west-3#/security\_credentials

aws

Services

Search

[Option+S]

Global

AkaEFREI

Identity and Access Management (IAM)

X

Search IAM

Dashboard

▼ Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

▼ Access reports

Access Analyzer

External access

Unused access

Analyzer settings

Credential report

No MFA devices. Assign an MFA device to improve the security of your AWS environment

Assign MFA device

Access keys (1)

Actions ▼

Create access key

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

|                       | Access key ID        | Created on   | Access key last used | Region last used |
|-----------------------|----------------------|--------------|----------------------|------------------|
| <input type="radio"/> | AKIAXYKJRHVY22DNHB6B | 22 hours ago | 15 hours ago         | eu-west-3        |

CloudFront key pairs (0)

Actions ▼

Upload

Create CloudFront key pair

You use key pairs in Amazon CloudFront to create signed URLs. You can have a maximum of two CloudFront key pairs (active or inactive) at a time.

| Creation time              | CloudFront key ID | Status |
|----------------------------|-------------------|--------|
| No CloudFront key pairs    |                   |        |
| Create CloudFront key pair |                   |        |

CloudShell

Feedback

Privacy

Terms

Cookie preferences

© 2024, Amazon Web Services, Inc. or its affiliates.

Dashboard | IAM | Global

Create access key | IAM | Global

https://us-east-1.console.aws.amazon.com/iam/home?region=eu-west-3#/security\_credentials

ServicesSearch[Option+S]

GlobalAkaEFREI

IAM > Security credentials > Create access key


Step 1

Alternatives to root user access keys

Step 2

Retrieve access key

## Alternatives to root user access keys



**Root user access keys are not recommended**

We don't recommend that you create root user access keys. Because you can't specify the root user in a permissions policy, you can't limit its permissions, which is a best practice.

Instead, use alternatives such as an IAM role or a user in IAM Identity Center, which provide temporary rather than long-term credentials. [Learn More](#)

If your use case requires an access key, create an IAM user with an access key and apply least privilege permissions for that user. [Learn More](#)

Continue to create access key?

☐ I understand creating a root access key is not a best practice, but I still want to create one.

Cancel

Create access key

CloudShellFeedback

PrivacyTermsCookie preferences

© 2024, Amazon Web Services, Inc. or its affiliates.



Dashboard | IAM | Global

Create access key | IAM | Global

https://us-east-1.console.aws.amazon.com/iam/home?region=eu-west-3#/security\_credentials

ServicesSearch[Option+S]

GlobalAkaEFREI

Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1

[Alternatives to root user access keys](#)

Step 2

**Retrieve access key**

Retrieve access key

Info

Access key

AKIAXYKJRHVYQFQZNS4B

Secret access key

\*\*\*\*\*[Show](#)

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

CloudShellFeedback

© 2024, Amazon Web Services, Inc. or its affiliates.

PrivacyTermsCookie preferences

# Démonstration

Créer un fichier credentials selon le format suivant :

```
[default]  
aws_access_key_id = <ACCESSKEY>  
aws_secret_access_key = <SECRETKEY>
```

Selon le système que vous utilisez, le fichier est à mettre dans un répertoire différent : <https://docs.aws.amazon.com/sdkref/latest/guide/file-location.html>

# Démonstration

Etape 0.3

Installer Terraform (ou OpenTofu) et Ansible

# Démonstration

## Etape 1

On utilise Terraform pour créer un container nommé `tutorial` à partir de l'image `nginx:1.27`

Les commandes Terraform utilisées :

```
terraform init
```

```
terraform plan
```

```
terraform apply --auto-approve
```

```
terraform destroy --auto-approve
```

# Démonstration

## Etape 2

On utilise Terraform pour créer 2 containers :

- le premier nommé `php-fpm` à partir de l'image `php:8.3-fpm`
- le premier nommé `nginx` à partir de l'image `nginx:1.27`

Les 2 containers communiquent à la manière de ce qu'on avait vu précédemment avec Docker : NGINX interroge PHP pour délivrer le contenu de l'exécution de la fonction `phpinfo()`

# Démonstration

## Etape 3

On crée maintenant une instance EC2 sur AWS.

La commande en haut du fichier permet de récupérer l'AMI utilisé en paramètre à la création de la ressource

key\_name contient le nom de la clé créée au préalable et qui permettra ensuite de se connecter en SSH par la suite

Le type d'instance t3.micro correspond au type d'instance gratuite dans le cadre de l'utilisation avec AWS Free Tier

# Démonstration

## Etape 4

On rajoute un output à Terraform qui permet de récupérer l'adresse IP de l'instance créée et qui permettra par la suite de s'y connecter

# Démonstration

## Etape 5

L'identifiant d'AMI est récupéré directement par une requête de type data dans le fichier Terraform au lieu de spécifier un identifiant d'ami "en dur".

L'AMI changera donc dans le temps au fur et à mesure des mises à jour.



# Démonstration

## Etape 6

Un nouveau type de ressource est ajouté : un security group nommé “allow\_ssh”

Dans celui-ci, on ouvre le port 22 (SSH) à la connexion depuis l'extérieur et nous permet donc de nous connecter à l'instance avec la commande suivante :

```
ssh -i myKey.pem ubuntu@<ADRESSE-IP>
```

# Démonstration

## Etape 7

Dans Terraform, on ajoute maintenant un autre security group nommé “allow\_http\_s” qui ouvre les ports 80 et 443, respectivement HTTP et HTTPS

On ajoute maintenant Ansible :

- Le fichier `playbook.yml` contient la liste des tâches qu’il va effectuer sur les instances
- le fichier `inventory.ini` contient la liste des instances auxquelles il va se connecter (penser à mettre à jour l’adresse IP à partir de celle qui a été récupérée dans les étapes précédentes)
- La commande : `ansible-playbook -i inventory.ini playbook.yml`

# Démonstration

## Etape 8

Nouvelle tâche Ansible :

Remplacer le fichier `index.html` par défaut par un fichier contenant le message  
"Bonjour le Monde"

# Démonstration

## Etape 9

Mise à jour de la tâche Ansible :

Le fichier `index.html` n'est plus mis à jour depuis Ansible mais par la copie d'un fichier présent localement.

# Démonstration

## Etape 10

L'inventaire d'Ansible est maintenant généré dynamiquement :

Il se connecte à l'API d'AWS pour récupérer la liste des instances et alimenter automatiquement l'inventaire :

```
ansible-playbook -i aws_ec2.yml playbook.yml
```