

Titre du projet

Configuration d'un Accès Sécurisé pour une Application Web sur AWS

Introduction

Ce projet démontre ma capacité à configurer une infrastructure Cloud sécurisée sur AWS. J'ai mis en œuvre un environnement où une application web hébergée sur EC2 utilise un rôle IAM pour accéder aux ressources S3, tout en suivant les meilleures pratiques de sécurité.

Objectifs du projet

1. Héberger une application web sur une instance EC2.
 2. Configurer un rôle IAM pour accéder aux fichiers stockés dans un compartiment S3 sans utiliser de clés statiques.
 3. Appliquer le principe du moindre privilège en limitant les autorisations.
 4. Automatiser le téléchargement de fichiers depuis S3 vers l'application web.
-

Étapes du projet

1. Création de l'Environnement

- Une instance EC2 a été lancée avec Amazon Linux 2 pour héberger une application web simple.
- Un compartiment S3 a été configuré pour stocker les fichiers nécessaires à l'application.



Capture d'écran :

Résumé de l'instance pour i-086e652634d185bf5 (web-app-instance) Informations

Se connecter

État de l'instance

Actions

Mis à jour il y a less than a minute

ID d'instance

i-086e652634d185bf5

Adresse IPv6

-

Type de nom d'hôte

Nom de l'adresse IP: ip-172-31-22-243.eu-west-3.compute.internal

Réponse à un nom DNS de ressource privée

IPv4 (A)

Adresse IP attribuée automatiquement

13.38.124.164 [IP publique]

Adresse IPv4 publique

13.38.124.164 | adresse ouverte

État de l'instance

En cours d'exécution

Nom DNS de l'IP privé (IPv4 uniquement)

ip-172-31-22-243.eu-west-3.compute.internal

Type d'instance

t2.micro

ID de VPC

vpc-047d8248b5c1553e0

Adresses IPv4 privées

172.31.22.243

DNS IPv4 public

ec2-13-38-124-164.eu-west-3.compute.amazonaws.com | adresse ouverte

Adresses IP élastiques

-

Recherche d'AWS Compute Optimizer

Inscrivez-vous à AWS Compute Optimizer pour

-instance EC2 active-

Amazon S3

Compartiments à usage général

Compartiments de répertoires

Compartiments de table

Nouveau

Access Grants

Points d'accès

Points d'accès de l'objet Lambda

Points d'accès multi-région

Opérations par lot

IAM Access Analyzer pour S3

Paramètres de blocage de l'accès public pour ce compte

Storage Lens

Tableaux de bord

Groupes Storage Lens

Paramètres AWS Organizations

web-app-storage-2025

Objets

Propriétés

Autorisations

Métriques

Gestion

Points d'accès

Objets (19)

Charger

Les objets sont les entités fondamentales stockées dans Amazon S3. Vous pouvez utiliser l'inventaire Amazon S3 pour obtenir une liste de tous les objets de votre compartiment. Pour que d'autres personnes puissent accéder à vos objets, vous devez leur accorder explicitement des autorisations. En savoir plus

Rechercher des objets en fonction du préfixe

	Nom	Type	Dernière modification	Taille	Classe de stockage
	admin.php	php	07 Dec 2024 12:21:14 AM GMT	835.0 o	Standard
	admin.php	php	07 Dec 2024 12:21:14 AM GMT	4.3 Ko	Standard
	adminconnect.html	html	07 Dec 2024 12:21:15 AM GMT	1.0 Ko	Standard

Active Windows

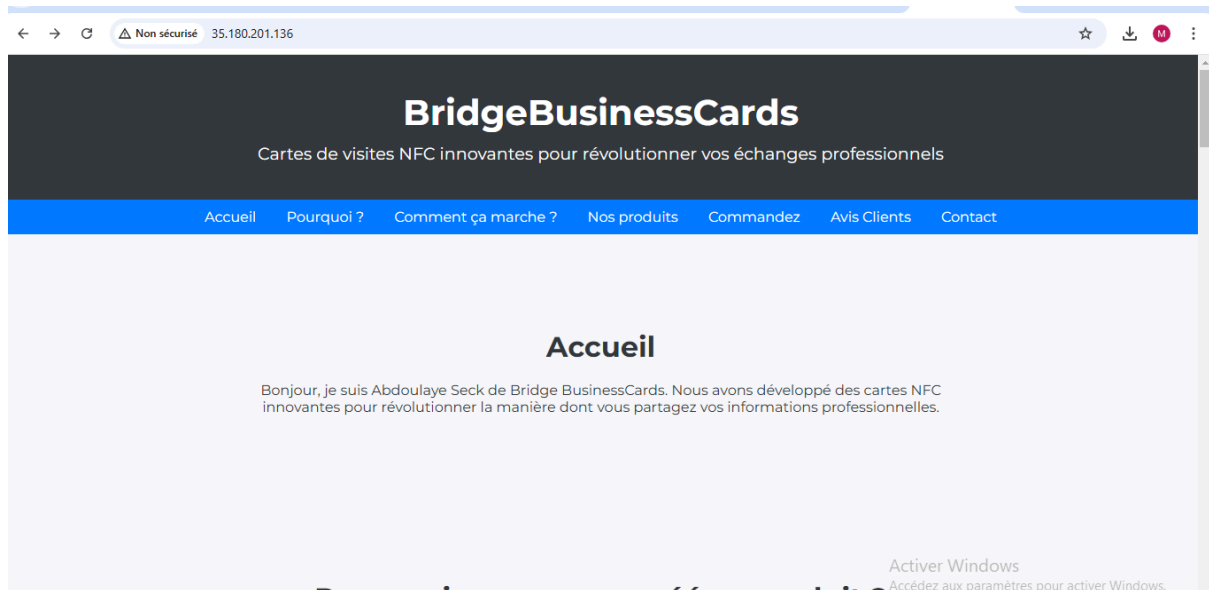
Accédez aux paramètres pour activer Windows.

- compartiment S3 créé-

2. Configuration du rôle IAM

- J'ai créé une stratégie IAM JSON permettant uniquement les actions nécessaires sur le compartiment S3 (lecture et listing).
- Un rôle IAM a été configuré et attaché à l'instance EC2.

 Capture d'écran :



-Page web affichant la page initiale -

4. Interaction avec S3 via le Rôle IAM

- Installation d'AWS CLI sur l'instance EC2.
- Tests réussis de la commande `aws s3 ls` pour vérifier l'accès sécurisé au compartiment S3.
- Automatisation d'un téléchargement de fichier depuis S3 vers EC2 via un script Bash.

 Capture d'écran :

```
ubuntu@ip-172-31-22-243:~$ aws s3 ls s3://web-app-storage-2025
PRE image/
2024-12-07 00:21:16      1528 ContactProcessor.java
2024-12-07 00:21:17      2329 OrderProcessor.java
2024-12-07 00:21:14         835 admin.php
2024-12-07 00:21:14      4433 admin.php
2024-12-07 00:21:15      1060 adminconnect.html
2024-12-07 00:21:15         65 composer.json
2024-12-07 00:21:15      4124 composer.lock
2024-12-07 00:21:16      9047 index.html
2024-12-07 00:21:16         830 insertadmin.php
2024-12-07 00:21:17         917 list_messages.php
2024-12-07 00:21:17      1225 list_orders.php
2024-12-07 00:21:17      2539 login.php
2024-12-07 00:21:18         865 process_contact_form.php
2024-12-07 00:21:18      1219 process_order.php
2024-12-07 00:21:18         564 scripts.js
2024-12-07 00:21:19      5928 styles.css
2024-12-07 00:21:19         922 styleless.css
2024-12-07 00:21:19         683 web.xml
ubuntu@ip-172-31-22-243:~$
```

-Résultat de la commande `aws s3 ls`-

```
ubuntu@ip-172-31-22-243:~$ cd /var/www/html/
ubuntu@ip-172-31-22-243:/var/www/html$ ls
9d4659572f8d4e567d444a087e4a27a2.jpg  ContactProcessor.java  Ntag.png
'OK.png.jpg'                          OrderProcessor.java    PetTag.jpg
'White Transparent background T-shirt - Fait avec PosterMyWall (2).png'
abdallah.jpg                          image                  module_table_top.png
admin.php                             index.html             process_contact_form.php
adminconnect.html                     insertadmin.php        process_order.php
'complet 02.jpg'                       list_messages.php      scripts.js
composer.json                          list_orders.php        styles.css
composer.lock                          login.php              styleless.css
module_table_bottom.png                web.xml
```

-Résultat de la commande `ls` du dossier `/var/www/html/`.

```
ubuntu@ip-172-31-22-243:~$ nano script.sh
ubuntu@ip-172-31-22-243:~$ chmod +x script.sh
ubuntu@ip-172-31-22-243:~$ cat script.sh
#!/bin/bash

# Télécharger tous les fichiers de la racine du bucket
sudo aws s3 cp s3://web-app-storage-[ton-id]/ /var/www/html/ --recursive

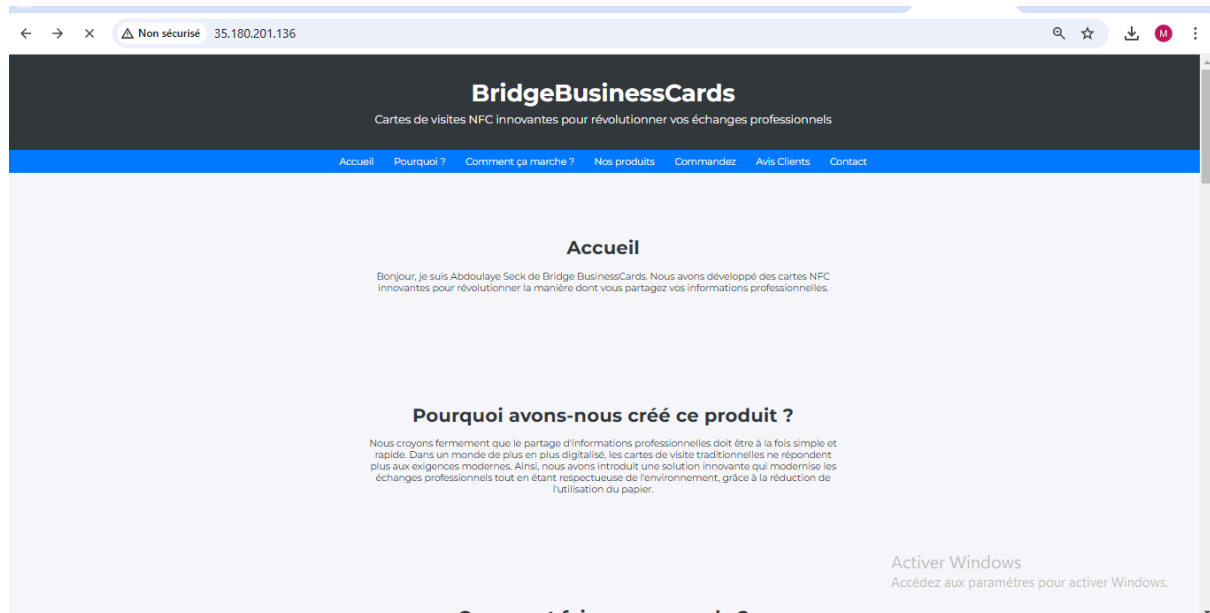
# Télécharger un dossier spécifique
sudo aws s3 cp s3://web-app-storage-[ton-id]/dossier-exemple /var/www/html/dossier-exemple --recursive
ubuntu@ip-172-31-22-243:~$
```

-Script Bash utilisé pour le téléchargement-

5. Résultat Final

- L'application web s'affiche automatiquement depuis S3.
- Le projet a rencontré en avant une interaction sécurisée entre EC2 et S3 grâce au rôle IAM configuré.

🖼️ Capture d'écran :



- Capture de la page web finale -

Compétences développées

1. **Gestion des Rôles IAM** : Création et configuration de rôles avec des permissions granulaires.
2. **Configuration et Automatisation AWS** : Interaction sécurisée entre EC2 et S3 via AWS CLI et scripts Bash.
3. **Sécurisation des Ressources Cloud** : Application du principe du moindre privilège pour réduire les risques de sécurité.
4. **Hébergement Web** : Mise en place d'un serveur Apache pour une application simple.

Conclusion

Ce projet illustre ma capacité à combiner **sécurité, automatisation et hébergement Cloud** dans un environnement AWS. Il reflète ma compréhension des meilleures pratiques IAM et de l'importance d'une gestion sécurisée des ressources AWS.