Rapport TP4

INF8102 - Sécurité dans les environnements infonuagiques

Eric-Pascal TIONFE
(2471139)

Lucas VINCENT
(2486417)

Novembre-Décembre 2025

Table des matières

1. Introduction

L'objectif de ce laboratoire est d'apprendre à mettre en œuvre une architecture en **Infrastructure as Code**. Au cours de cette séance pratique, nous avons travaillés les points suivants :

- Mise en place d'un Virtual Private Cloud (VPC) et ses périphériques à l'aide CloudFormation.
- Mise en place d'un compartiment S3 à l'aide de CloudFormation.
- Mise en place d'une instance EC2 à l'aide de CloudFormation.
- Scan des vulnérabilités à partir des fichiers de configuration à l'aide de trivy.
- Mise en place d'un Virtual Private Cloud (VPC) et ses périphériques à l'aide de *boto3*.
- Sécurisation du VPC en modifiant le *script python boto3*.
- Mise en place d'un compartiment S3 à l'aide de *boto3*.
- Sécurisation du compartiment S3 à l'aide de *boto3*.

Le projet a été publié dans un répertoire Github. Il est disponible grâce au lien suivant :

https://github.com/lucobyX6/Deploy_AWS_with_IaC

2. Déploiement à l'aide de Cloud Formation

2.1. Mise en place du VPC

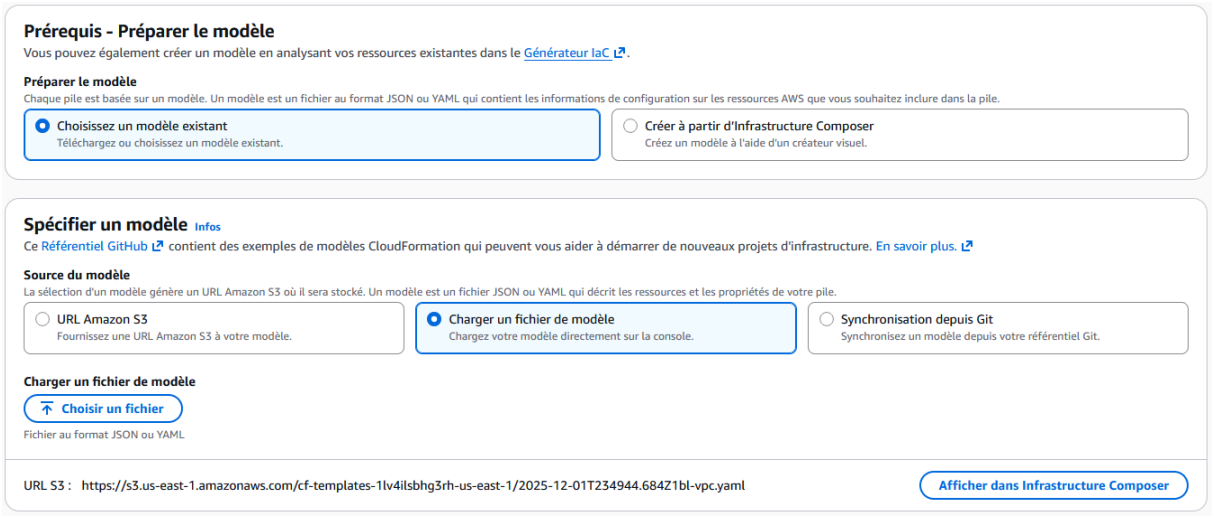


Fig. 1. – Création à partir du modèle json.

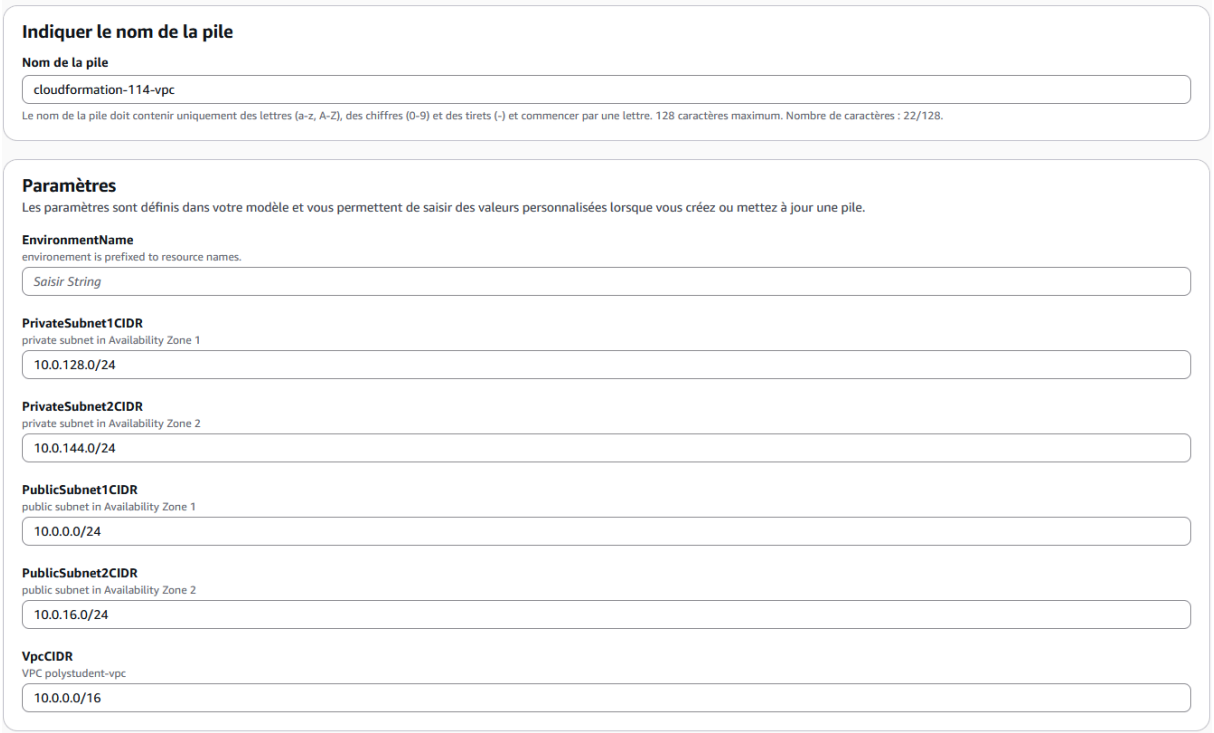
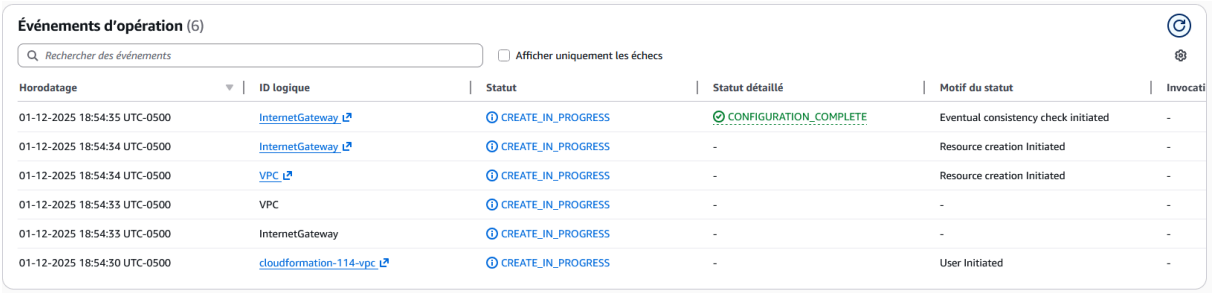


Fig. 2. – Validation des paramètres.



Événements d'opération (6)

Rechercher des événements | Afficher uniquement les échecs

Horodatage	ID logique	Statut	Statut détaillé	Motif du statut	Invocati
01-12-2025 18:54:35 UTC-0500	InternetGateway	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated	-
01-12-2025 18:54:34 UTC-0500	InternetGateway	CREATE_IN_PROGRESS	-	Resource creation Initiated	-
01-12-2025 18:54:34 UTC-0500	VPC	CREATE_IN_PROGRESS	-	Resource creation Initiated	-
01-12-2025 18:54:33 UTC-0500	VPC	CREATE_IN_PROGRESS	-	-	-
01-12-2025 18:54:33 UTC-0500	InternetGateway	CREATE_IN_PROGRESS	-	-	-
01-12-2025 18:54:30 UTC-0500	cloudformation-114-vpc	CREATE_IN_PROGRESS	-	User Initiated	-

Fig. 3. – Création du VPC.

2.2. Mise en place de l'instance EC2

Examiner et créer

Étape 1: Spécifier un modèle — Modifier

Prérequis - Préparer le modèle

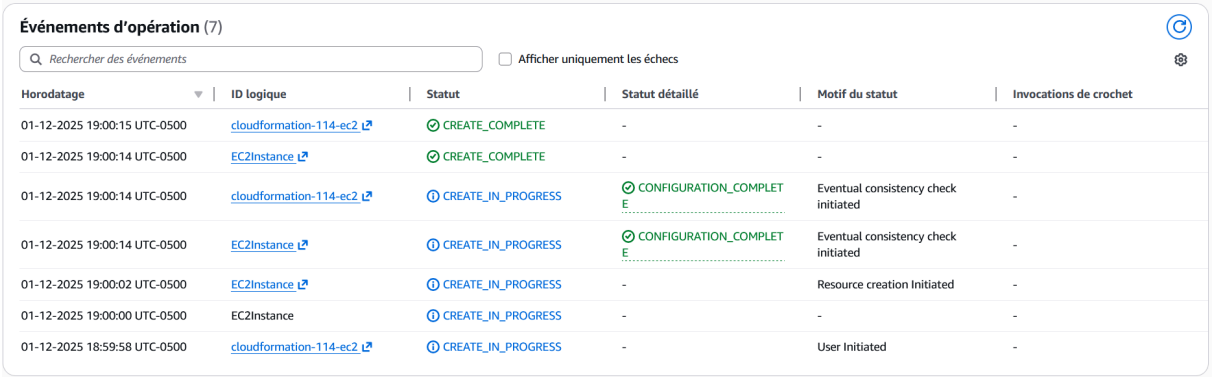
Modèle
Le modèle est prêt

Modèle

URL modèle
https://s3.us-east-2.amazonaws.com/cf-templates-1lv4ilsbhg3rh-us-east-2/2025-12-01T231236.568Z40b-ec2.json

Description de la pile
Deploy a secure EC2 instance on the public subnet of AZ1

Fig. 4. – Création à partir d'un modèle json.



Événements d'opération (7)

Rechercher des événements | Afficher uniquement les échecs

Horodatage	ID logique	Statut	Statut détaillé	Motif du statut	Invocations de crochet
01-12-2025 19:00:15 UTC-0500	cloudformation-114-ec2	CREATE_COMPLETE	-	-	-
01-12-2025 19:00:14 UTC-0500	EC2Instance	CREATE_COMPLETE	-	-	-
01-12-2025 19:00:14 UTC-0500	cloudformation-114-ec2	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated	-
01-12-2025 19:00:14 UTC-0500	EC2Instance	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated	-
01-12-2025 19:00:02 UTC-0500	EC2Instance	CREATE_IN_PROGRESS	-	Resource creation Initiated	-
01-12-2025 19:00:00 UTC-0500	EC2Instance	CREATE_IN_PROGRESS	-	-	-
01-12-2025 18:59:58 UTC-0500	cloudformation-114-ec2	CREATE_IN_PROGRESS	-	User Initiated	-

Fig. 5. – Création de l'instance EC2.

2.3. Mise en place du compartiment S3

Fig. 6. – Création à partir du modèle json.

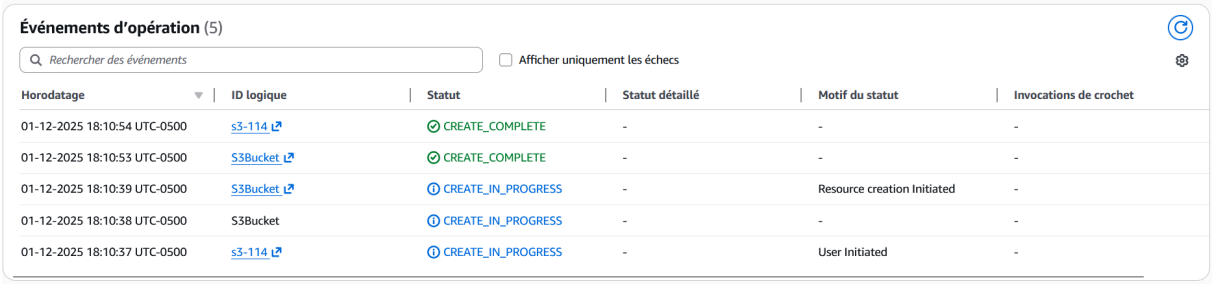


Fig. 7. – Création du compartiment S3.

2.4. Scan des vulnérabilités

Nous avons exécuté la commande suivante sur le dossier des configurations avec CloudFormation :

```
trivy fs --scanners vuln,secret,misconfig CloudFormation
```

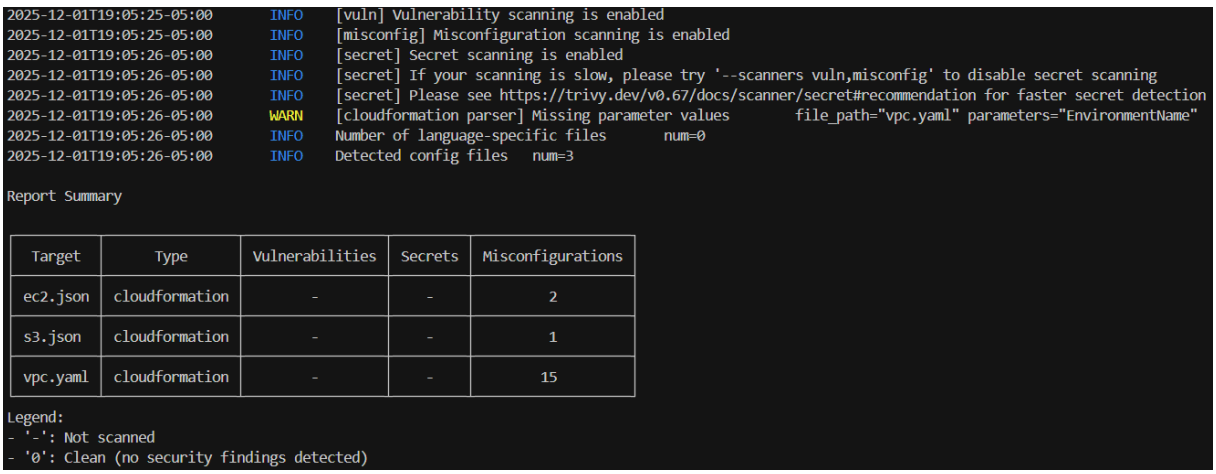


Fig. 8. – Extrait du résultat du scan de trivy.

3. Déploiement à l'aide de boto3

3.1. Question 1

Le VPC, ses deux sous-réseaux, sa passerelle Internet, ses passerelles NAT, ses quatre instances et son groupe de sécurité ont été construit à l'aide de boto3. Les images ci-dessous illustrent le bon fonctionnement du Script.

```
[INFO] Connection to aws session
[INFO] Create VPC
VPC Id : vpc-002aee732e8e99a7a
[INFO] Create subnets
Subnets Id : {'public_az1': 'subnet-0f6a4a61821b5e1c0', 'private_az1': 'subnet-0e4da2f6a79ec398d', 'public_az2': 'subnet-0c0fef3cd7db1c5a4', 'private_az2': 'sub
net-04b3e9debe3d0315d'}
[INFO] Create internet gateway
[INFO] Create nat gateways
[INFO] Create security group
[INFO] Security group : sg-0c23b88b9c5036c41
[INFO] Enable automatic public IP
[INFO] Create EC2 in private and public subnets
[INFO] End
```

Fig. 9. – L'exécution s'est déroulée sans erreurs. Le terminal décrit les étapes.

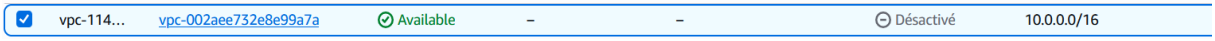


Fig. 10. – Le VPC a été créé.

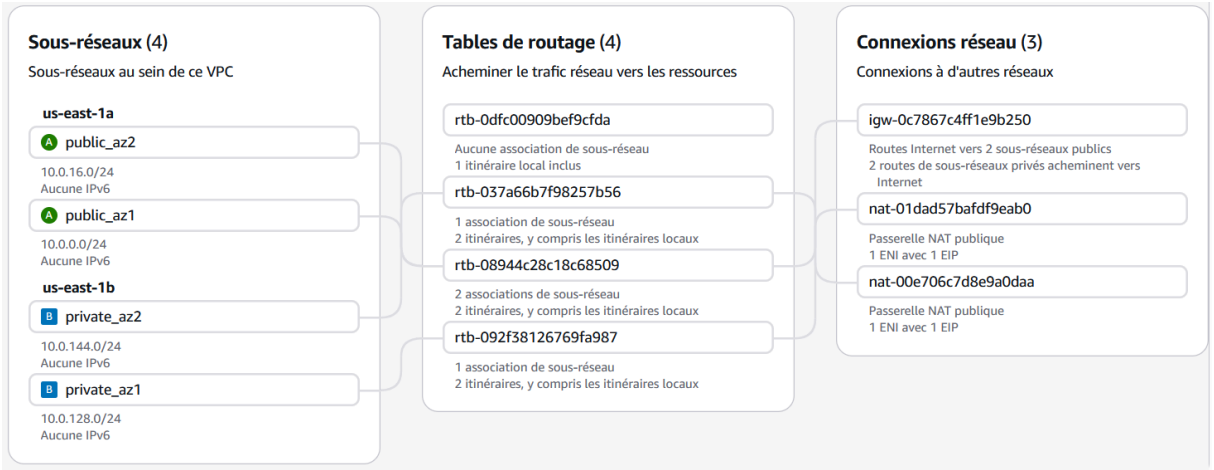


Fig. 11. – Les sous-réseaux publics sont reliés par une passerelle internet avec le réseau extérieur et les sous-réseaux privés sont reliés avec deux passerelles NAT.

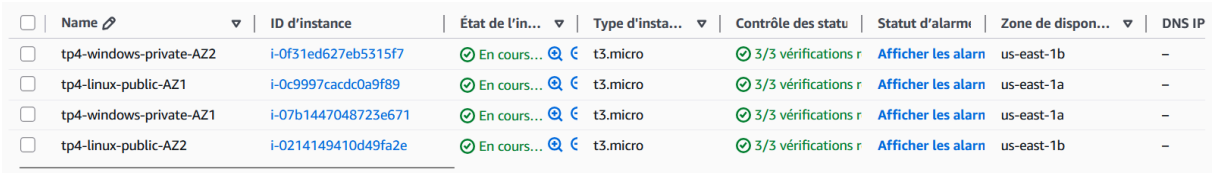


Fig. 12. – Les quatre instances EC2 ont été créé et affectée aux bonnes zones de disponibilités.

▼ Détails de sécurité

Rôle IAM
–

ID du propriétaire
107079351100

Heure de lancement
Sun Nov 30 2025 18:05:54 GMT-0500 (heure normale de l'Est nord-américain)

Groupes de sécurité
sg-0c23b88b9c5036c41 (vpc-114-security-group)

▼ Règles entrantes

Filtrer les règles

Nom	ID de règle du groupe de s...	Plage de ports	Protocole	Source	Groupes de sécurité
–	sgr-0a1828322b7fe6921	5432	TCP	0.0.0.0/0	vpc-114-security-group
–	sgr-0c194ba2f273af39b	3306	TCP	0.0.0.0/0	vpc-114-security-group
–	sgr-0291c292a8c0e62a7	80	TCP	0.0.0.0/0	vpc-114-security-group
–	sgr-0643fb5e147991892	53	TCP	0.0.0.0/0	vpc-114-security-group
–	sgr-0e052671a5ba67c5f	3389	TCP	0.0.0.0/0	vpc-114-security-group
–	sgr-00af0c36028bc05b8	22	TCP	0.0.0.0/0	vpc-114-security-group
–	sgr-0e55ca1c3015188e8	1514	TCP	0.0.0.0/0	vpc-114-security-group
–	sgr-0ebb3d4f35d8db96e	9200 - 9300	TCP	0.0.0.0/0	vpc-114-security-group
–	sgr-030c6a7febf24a79f	443	TCP	0.0.0.0/0	vpc-114-security-group
–	sgr-07df630791b46eccf	1433	TCP	0.0.0.0/0	vpc-114-security-group

Fig. 13. – Le groupe de sécurité est associé aux instances EC2.

3.2. Question 2

Le compartiment S3, avec son chiffrement statique et dynamique, son versionnement et le scan de ses vulnérabilités a été construit à l'aide de boto3. Les images ci-dessous illustrent le bon fonctionnement du Script.

```
[INFO] Connection to aws session
[INFO] Create S3
[INFO] Enable encryption
[INFO] Enable versionning
[INFO] Upload sourcefile to bucket
[INFO] Execute a scan on sourcecode
[INFO] End
```

Fig. 14. – L'exécution s'est déroulée sans erreurs. Le terminal décrit les étapes.

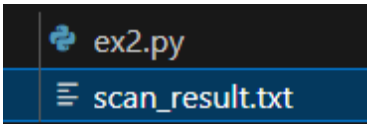


Fig. 15. – Un fichier d'analyse du fichier python avec bandit.

Nom	Région AWS	Date de création
s3-114-python	USA Est (Virginie du Nord) us-east-1	30 Nov 2025 06:18:30 PM EST

Fig. 16. – Le compartiment S3 a été créé.

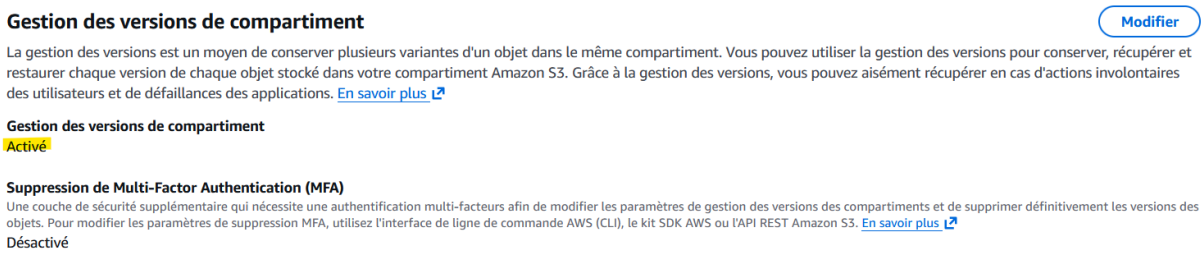


Fig. 17. – Le versionnage est actif.

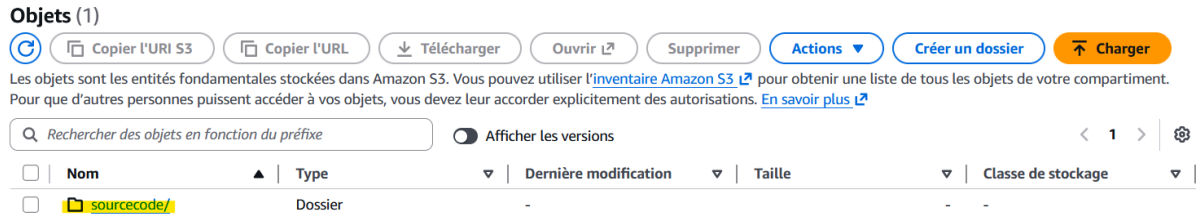


Fig. 18. – Le dossier de stockage du code source a été créé.

Fig. 19. – Le chiffrement du compartiment S3 est actif.

3.3. Question 3

3.3.1. Partie 1

```
[INFO] Connection to aws session
[INFO] Create VPC
VPC Id : vpc-0c0f1e51aa95265ba
[INFO] Create subnets
Subnets Id : {'public_az1': 'subnet-073c2bc221126de7d', 'private_az1': 'subnet-0a2067e7f6b89fd13', 'public_az2': 'subnet-09fa86fa7ba0095b2', 'private_az2': 'sub
net-023c7ad6aa877e09a'}
[INFO] Create internet gateway
[INFO] Create nat gateways
[INFO] Create security group
[INFO] Security group : sg-0d491911f334e92f7
[INFO] Enable automatic public IP
[INFO] Enable flow logs
[INFO] Create EC2 in private and public subnets
[INFO] End
```

Fig. 20. – L'exécution s'est déroulée sans erreurs. Le terminal décrit les étapes.

Fig. 21. – Le flowlog pour le VPC est actif est transmis dans le compartiment S3.

3.3.2. Partie 2

Fig. 22. – Le rôle de sécurité est associé aux instances EC2.

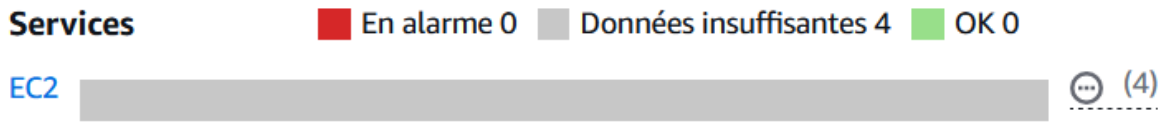


Fig. 23. – Les instances ont une alarme associée.

	Nom	État	Dernière mise à jour de l'état (UTC)	Conditions	Actions
☐	alarm-packetsIn-i-066e3cdefce000c27-python	Données insuffisantes	2025-12-01 21:27:01	NetworkPacketsIn > 1000 pour 1 points de données dans 5 minutes	Actions activées Avertissement
☐	alarm-packetsIn-i-0f74d889c6dc62a1c-python	Données insuffisantes	2025-12-01 21:27:00	NetworkPacketsIn > 1000 pour 1 points de données dans 5 minutes	Actions activées Avertissement
☐	alarm-packetsIn-i-08e254abb81d310e1-python	Données insuffisantes	2025-12-01 21:27:00	NetworkPacketsIn > 1000 pour 1 points de données dans 5 minutes	Actions activées Avertissement
☐	alarm-packetsIn-i-0080b385907f046bd-python	Données insuffisantes	2025-12-01 21:26:59	NetworkPacketsIn > 1000 pour 1 points de données dans 5 minutes	Actions activées Avertissement

Fig. 24. – Les alarmes sont actives et associées aux instances EC2.

3.3.3. Partie 3

```
[INFO] Connection to aws session
[INFO] Create S3
[INFO] Enable encryption
[INFO] Enable versionning
[INFO] Upload sourcefile to bucket
[INFO] Enable cloudtrail for S3 bucket
[INFO] Execute a scan on sourcecode
[INFO] Create replication S3
[INFO] Enable encryption
[INFO] Enable versionning
[INFO] Enable replication
[INFO] End
```

Fig. 25. – L'exécution s'est déroulée sans erreurs. Le terminal décrit les étapes..

○	cloudtrail-s3-python	USA Est (Virginie du Nord)	Non	arn:aws:cloudtrail:us-east-1:107079351100:trail/cloudtrail-s3-python	Désactivé	Non	s3-114-python	-	-	Journalisation

Fig. 26. – Cloudtrail est actif et associé au compartiment S3.

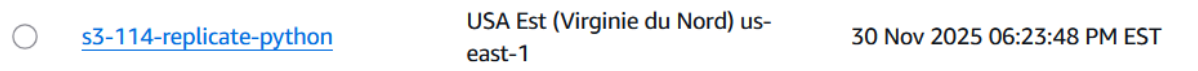


Fig. 27. – La réplication est active et un compartiment de réplication a été créé.

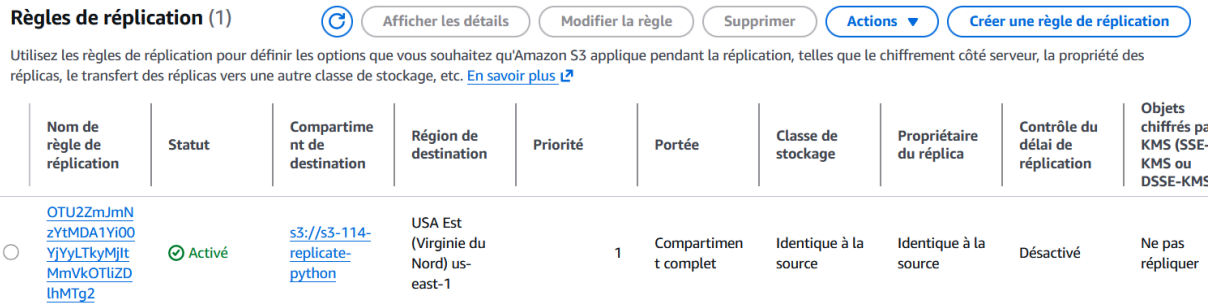


Fig. 28. – La réplication est bien active dans le compartiment *s3-114-python*.

3.4. Question 4

Le script *ex4.py* permet de scanner les scripts de génération en python et génère un fichier avec les résultats. *Trivy ne permet pas de réaliser un scan d'un fichier python, nous avons donc utilisé* ***bandit*** *en remplacement. Ce logiciel crée un rapport semblable à trivy. Malheureusement, bandit se concentre sur les vulnérabilités du code python et non de l'architecture générée.*

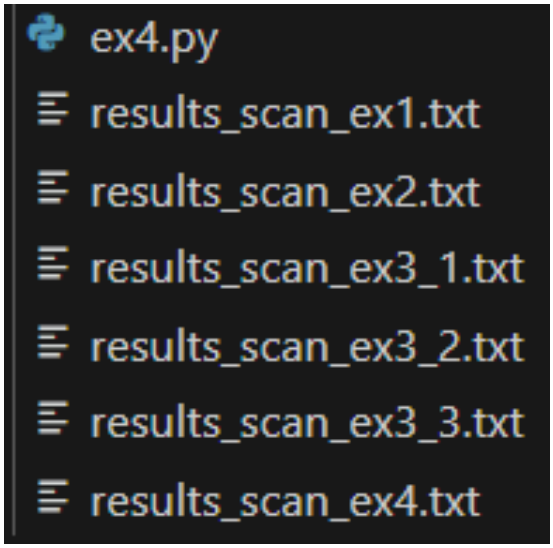
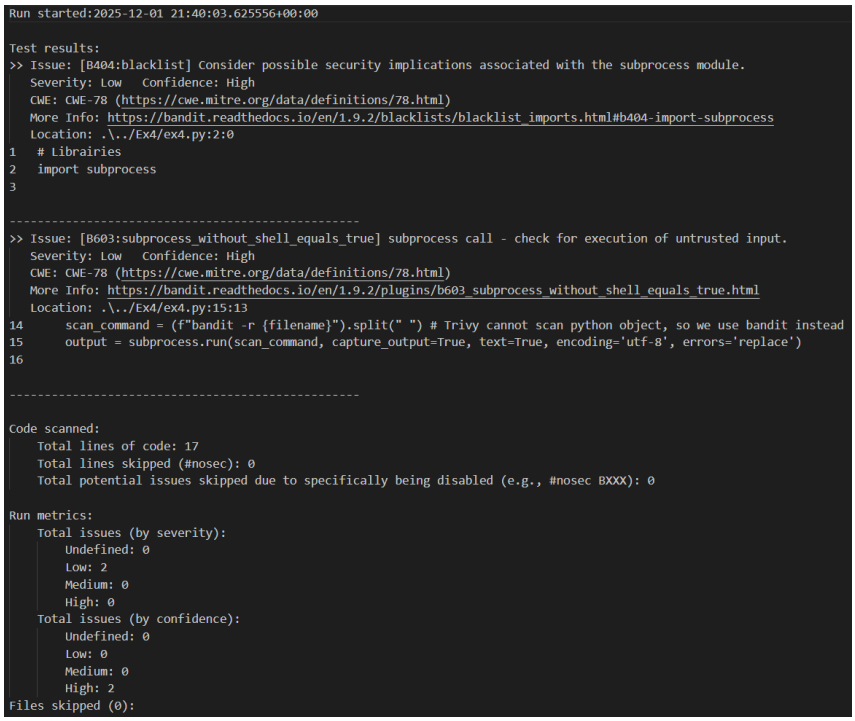


Fig. 29. – Les fichiers d'analyse de bandit ont été générés.



```
Run started:2025-12-01 21:40:03.625556+00:00

Test results:
>> Issue: [B404:blacklist] Consider possible security implications associated with the subprocess module.
   Severity: Low   Confidence: High
   CWE: CWE-78 (https://cwe.mitre.org/data/definitions/78.html)
   More Info: https://bandit.readthedocs.io/en/1.9.2/blacklists/blacklist_imports.html#b404-import-subprocess
   Location: .\../Ex4/ex4.py:2:0
1   # Librairies
2   import subprocess
3

--------------------------------------------------
>> Issue: [B603:subprocess_without_shell_equals_true] subprocess call - check for execution of untrusted input.
   Severity: Low   Confidence: High
   CWE: CWE-78 (https://cwe.mitre.org/data/definitions/78.html)
   More Info: https://bandit.readthedocs.io/en/1.9.2/plugins/b603_subprocess_without_shell_equals_true.html
   Location: .\../Ex4/ex4.py:15:13
14      scan_command = (f"bandit -r {filename}").split(" ") # Trivy cannot scan python object, so we use bandit instead
15      output = subprocess.run(scan_command, capture_output=True, text=True, encoding='utf-8', errors='replace')
16

--------------------------------------------------

Code scanned:
    Total lines of code: 17
    Total lines skipped (#nosec): 0
    Total potential issues skipped due to specifically being disabled (e.g., #nosec BXXX): 0

Run metrics:
    Total issues (by severity):
        Undefined: 0
        Low: 2
        Medium: 0
        High: 0
    Total issues (by confidence):
        Undefined: 0
        Low: 0
        Medium: 0
        High: 2
Files skipped (0):
```

Fig. 30. – Exemple d'un fichier d'analyse de bandit.

Mesures de mitigation : (voir Tableau 1 ci-dessous)

	Vulnérabilité	Gravité	Script	Stratégie de mitigation
Vulnérabilités & Mitigations	AVD-AWS-0028 → Instance does not require IMDS access to require a token.	HAUTE	ec2.json	Metadata Service (IMDS) permet à l'instance de récupérer des informations utiles à son fonctionnement. Actuellement, cette connexion se fait sans gestion des accès. Une bonne pratique serait d'activer l'**IMDSv2** pour gérer les accès par token.
	AVD-AWS-0107 Security group rule allows unrestricted ingress from any IP address.	HAUTE	vpc.yaml	Les ports sont restreint, mais un utilisateur peut se connecter depuis n'importe quelle adresse IP. Pour assurer une défense en profondeur, il faut restreinte l'accès à certaines IP de l'entreprise. De plus, il faut mettre en place une gestion du cycle de vie pour cette table d'adresses, de manière à retirer les accès lorsque l'opérateur n'en a plus besoin. L'objectif est d'appliquer le **minimum de privilèges pour une durée minimale**.
	AVD-AWS-0178 : VPC does not have VPC Flow Logs enabled	Moyenne	vpc.yaml	Pour garantir une sécurité maximale, il faut qu'un audit du système soit possible. Pour réaliser cet audit, il est préférable d'activer **VPC Flow Logs** et de le lier à un compartiment chiffré pour garantir l'**intégrité des informations**. *La question 3.2 corrige cette vulnérabilité.*
	VD-AWS-0131 : Root block device is not encrypted.	Haute	ec2.json	Le disque racine n'est pas chiffré. Un acteur malveillant peut donc récupérer les données écrites en clair dans l'instance. Pour empêcher cette action, il faudrait mettre en place un **chiffrement statique** des données du disque grâce à une paire de clef KMS ou une paire stockée sur un HSM (Hardware Security Module).
	AVD-AWS-0089 : Bucket has logging disabled	Faible	s3.json	Le compartiment n'enregistre pas les requêtes. Dans un objectif d'audit du système, il est préférable de stocker les activités. Le **logging est une fonctionnalité activable du compartiment S3 AWS**.

Tableau 1. – Table de cinq vulnérabilités et mitigations associées