

Projet 1 - Infrastructure Serverless pour la Gestion des Données des Clients

Description du projet

Ce projet vise à construire une infrastructure **serverless** pour traiter les données clients en temps réel. Le scénario simule une entreprise télécom qui collecte des fichiers de données clients, les stocke dans Amazon S3, et déclenche automatiquement une fonction AWS Lambda pour l'analyseur.

Étape 1 : Mise en place de la base

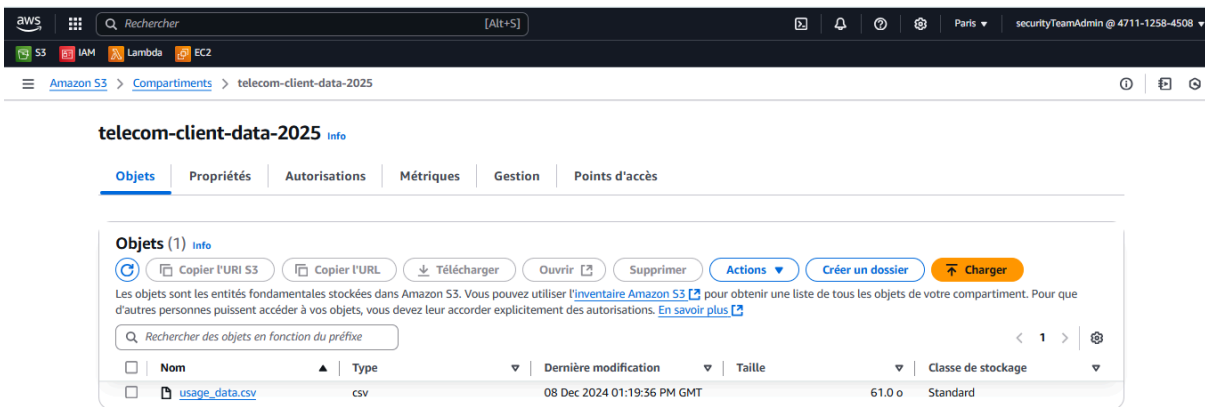
Création du Compartiment S3

- **Nom du compartiment** : `telecom-client-data-[ton-id]`.
- **Paramètres** : Paramètres par défaut, sans autorisation publique.

Téléchargement d'un fichier exemple

Un fichier CSV nommé `usage_data.csv` a été créé .

- Ce fichier a été téléchargé dans le compartiment S3.



 **Capture 1** : Capture d'écran du compartiment S3 contenant le fichier téléchargé.

Étape 2 : Création de la fonction Lambda

Configuration de la Fonction Lambda

- **Nom de la fonction** : **ProcessTelecomData**.
- **Exécution** : Python 3.9.
- **Type** : Fonction vide.

Configuration des autorisations (IAM)

- **Nom du rôle** : **LambdaS3AccessRole**.

Attachée politique IAM :

The screenshot shows the AWS IAM console for the **LambdaS3AccessRole** policy. The **Détails de la stratégie** (Policy Details) tab is active, displaying the following information:

Type	Heure de création	Heure de modification	ARN
Gérées par le client	December 08, 2024, 13:25 (UTC)	December 08, 2024, 13:25 (UTC)	arn:aws:iam::471112584508:policy/LambdaS3AccessRole

Below the details, the **Autorisations définies dans cette politique** (Permissions defined in this policy) tab is active, showing the JSON policy document:

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "s3:GetObject",
8         "s3:ListBucket"
9       ],
10      "Resource": [
11        "arn:aws:s3:::telecom-client-data-2025",
12        "arn:aws:s3:::telecom-client-data-2025/*"
13      ]
14    }
15  ]
16 }
```

Buttons for **Copier**, **Modifier**, **Récapitulatif**, and **JSON** are available. An **Activer Windows** watermark is visible at the bottom right.

Le code suivant a été implémenté pour lire et afficher les données du fichier CSV téléchargé dans S3 :

The screenshot shows the AWS Lambda console for the **ProcessTelecomData** function. The **Code** tab is active, displaying the following Python code:

```

1 import boto3
2 import csv
3
4 def lambda_handler(event, context):
5     # Nom du compartiment et fichier
6     bucket_name = "telecom-client-data-2025"
7     file_name = event['Records'][0]['s3']['object']['key']
8
9     # Connexion à S3
10    s3 = boto3.client('s3')
11    response = s3.get_object(Bucket=bucket_name, Key=file_name)
12    content = response['Body'].read().decode('utf-8').splitlines()
13
14    # Lire le fichier CSV
15    csv_reader = csv.reader(content)
16    for row in csv_reader:
17        print(f"ClientID: {row[0]}, Usage: {row[1]}, Date: {row[2]}")
18
```

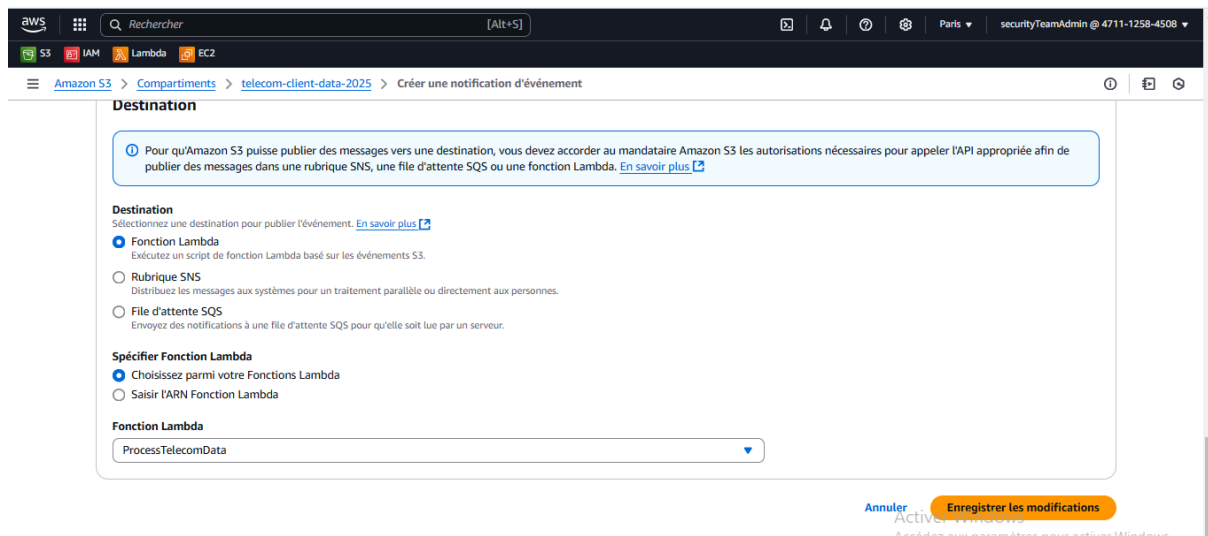
The **EXPLORER** pane on the left shows the project structure with **PROCESSTELECOMDATA** and **lambda_function.py**. The **TEST EVENTS** pane shows **Private saved events** and **S3UploadEvent**. An **Activer Windows** watermark is visible at the bottom right.


 **Capture 2** : Capture d'écran de la fonction Lambda et des autorisations configurées.

Étape 3 : Configuration d'un Déclencheur S3

Création de la Règle d'Événement S3

- **Nom de la règle** : `NewFileTrigger`.
- **Événement suivi** : `ObjectCreated:*`(tous les événements de création d'objets).
- **Destination** : La fonction Lambda `ProcessTelecomData`.



 **Capture 3** : Capture d'écran de la règle d'événement configurée dans S3.

Étape 4 : Test et résultats

Téléchargement d'un Nouveau Fichier

Un nouveau fichier CSV a été créé avec les données suivantes :

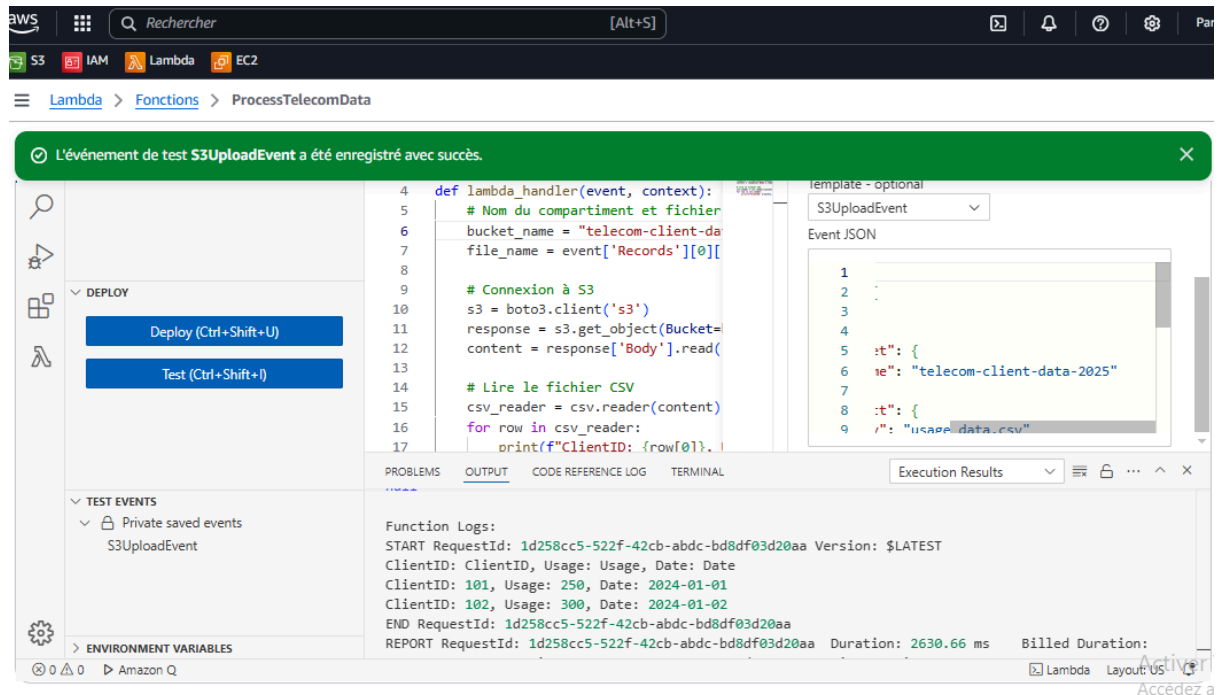
`ClientID, Usage, Date`
`103,400, 2024-01-03`


- Ce fichier a été téléchargé dans le compartiment S3.

Résultat dans CloudWatch Logs

- La fonction Lambda s'ouvre automatiquement et a lu les données du fichier.

Logs obtenus dans CloudWatch Logs :



 **Capture 4** : Capture d'écran des logs dans CloudWatch affichant les données traitées.

Résumé du projet

- **Technologies utilisées** : Amazon S3, AWS Lambda, IAM, CloudWatch Logs.
- **Résultat obtenu** :
 - Une architecture serverless qui traite automatiquement les fichiers téléchargés dans S3.
 - Une solution simple et efficace pour l'analyse de données en temps réel.
- **Points forts** :
 - Configuration robuste avec des rôles IAM sécurisés.
 - Automatisation complète grâce au déclenchement S3 et à Lambda.