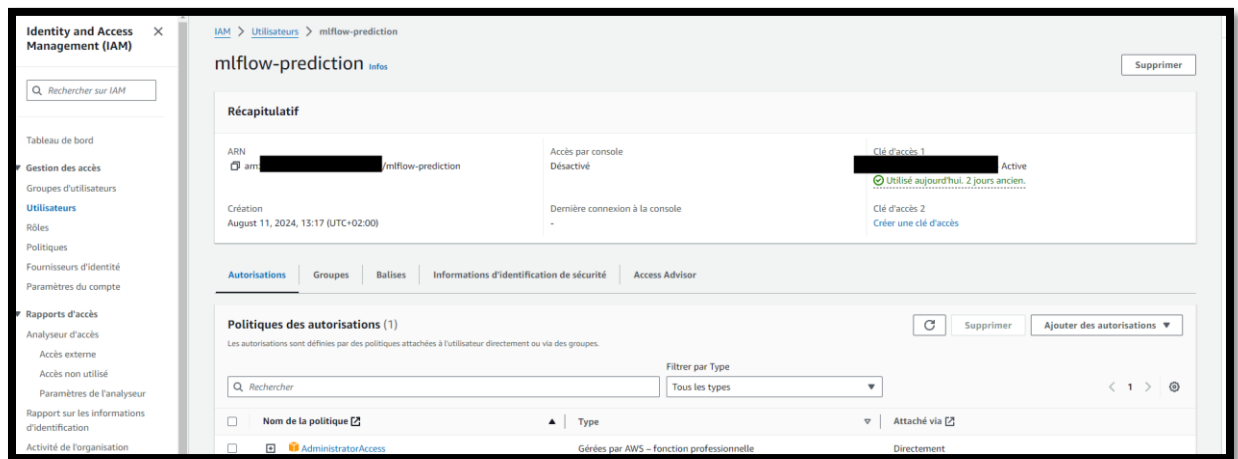


Configuration de MLflow sur AWS :

1. Connectez-vous à la console AWS.

2. Créer un utilisateur IAM avec les droits AdministratorAccess :

- Accédez à la section **IAM** (Identity and Access Management) dans la console AWS.
- Créez un nouvel utilisateur avec les permissions AdministratorAccess.
- Téléchargez les informations d'identification (clé d'accès et clé secrète) pour les utiliser dans la configuration de votre CLI AWS.



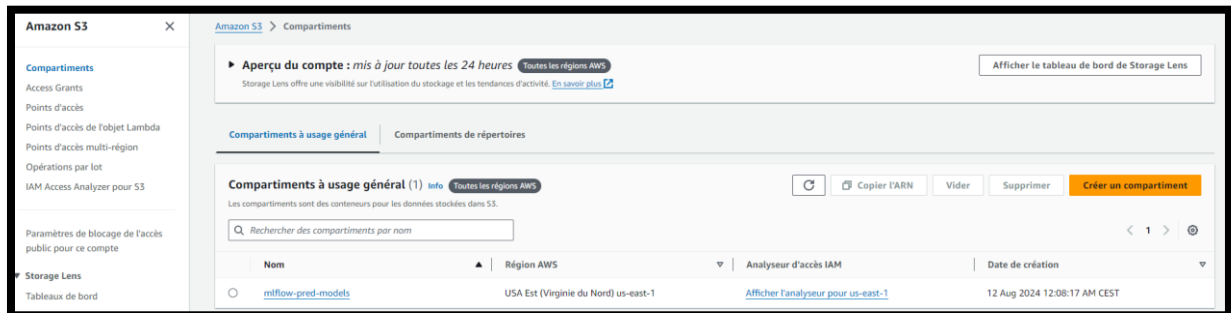
3. Exporter les informations d'identification dans votre AWS CLI :

- Ouvrez votre terminal et exécutez la commande suivante :
 - aws configure
- Lors de l'exécution de cette commande, entrez les informations suivantes :
 - AWS Access Key ID: clé d'accès IAM.
 - AWS Secret Access Key: clé secrète IAM.
 - Default region name : La région AWS (ex. us-east-1).
 - Default output format : Le format de sortie souhaité (ex. json).

```
(base) abbasbil@ABBASBILL:~/Openclassrooms/Projet_5$ aws configure
AWS Access Key ID [*****WNP6]:
AWS Secret Access Key [*****u7RC]:
Default region name [us-east-1]:
Default output format [None]:
```

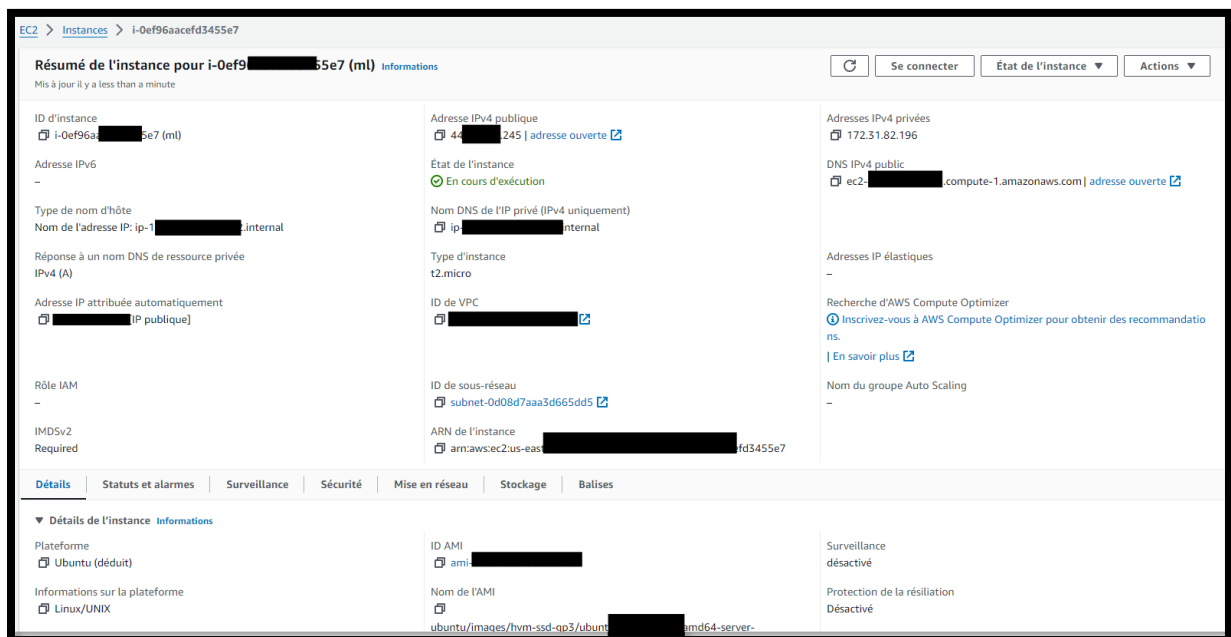
4. Créer un bucket S3 :

- Accédez à la section S3 dans la console AWS.
- Créez un nouveau bucket, par exemple nommé mlflow-pred-models.
- Notez le nom du bucket, car il sera utilisé lors de la configuration de MLflow.



5. Créer une machine EC2 (Ubuntu) :

- Accédez à la section EC2 dans la console AWS.
- Lancez une nouvelle instance EC2 en choisissant Ubuntu comme système d'exploitation.
- Configurez les paramètres de l'instance selon vos besoins (type d'instance, réseau, etc.).
- Assurez-vous d'ajouter une règle de sécurité pour autoriser le trafic entrant sur le port 5000 afin de permettre l'accès au serveur MLflow.



6. Mettre à jour le système et installer les dépendances Python :

- Se connecter à l'instance EC2 avec le bouton « se connecter »
- Exécutez les commandes suivantes pour mettre à jour le système et installer les dépendances nécessaires :
 - `sudo apt update && sudo apt full-upgrade -y`
 - `sudo apt install -y python3-pip python3-venv`

7. Créer et activer un environnement virtuel :

- Créez un nouvel environnement virtuel Python et activez-le :
 - `python3 -m venv mlflow_env`
 - `source mlflow_env/bin/activate`

8. Installer Pipenv dans l'environnement virtuel :

- Installez Pipenv pour la gestion des dépendances Python :
 - `pip install pipenv`

9. Configurer le projet MLflow :

- Créez un répertoire pour votre projet MLflow et installez les dépendances :
 - `mkdir mlflow`
 - `cd mlflow`
 - `pipenv install mlflow awscli boto3`

10. Activer l'environnement Pipenv :

- Lancez l'environnement virtuel Pipenv :
 - `pipenv shell`

11. Configurer les identifiants AWS :

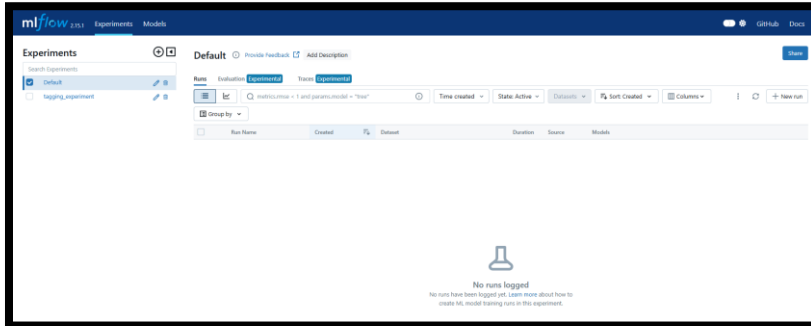
- Configurez votre environnement pour utiliser les identifiants AWS :
 - `aws configure`
- Répétez cette étape si vous devez modifier ou mettre à jour les informations d'identification AWS.

12. Lancer le serveur MLflow :

- Exécutez MLflow en tant que serveur, en spécifiant l'emplacement par défaut pour les artefacts (votre bucket S3) :
 - `mlflow server -h 0.0.0.0 --default-artifact-root s3://mlflow-pred-models`

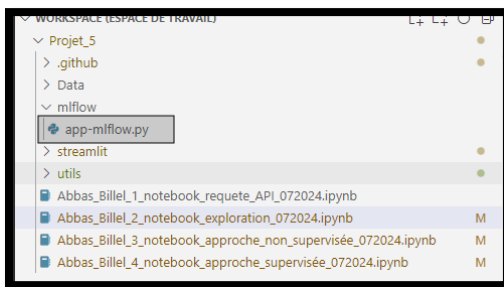
13. Accéder à l'interface MLflow :

- Ouvrez un navigateur web et accédez à l'**adresse IPv4 publique** de votre instance EC2 sur le **port 5000** pour accéder à l'interface MLflow.
 - Exemple d'URL : <http://ec2-44-204-37-245.compute-1.amazonaws.com:5000/>

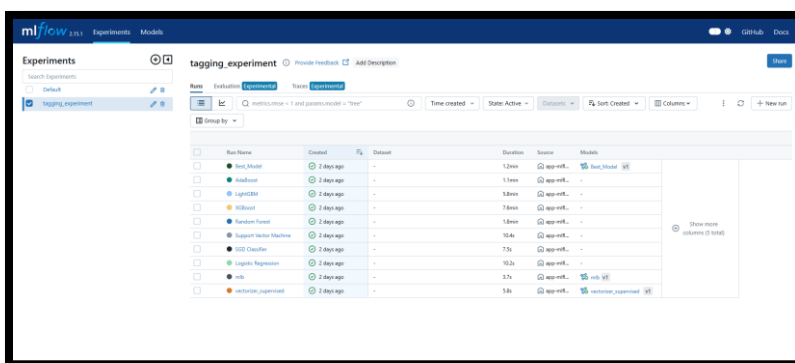


14. Lancer en locale le module python app-mlflow.py :

- Lancez votre application MLflow en utilisant la commande suivante :
 - `python app-mlflow.py`



- On obtient dans la partie experiments les modules entraînés :



- Résultats :
 - Le meilleur modèle : XGBoost, qui est stocké dans la section Models en version 1.
 - Stockage du modèle mlb (MultiLabelBinarizer()) en version 1.
 - Stockage du vectoriseur TFidf en version 1.