

Mini-Projet : Gestion d'Utilisateurs avec AWS, Python et SQL

Objectif du Projet

Créer une application qui permet :

- L'ajout, la lecture, la modification et la suppression d'utilisateurs via une API.
 - La sauvegarde des utilisateurs dans une base de données MySQL hébergée sur AWS RDS.
 - L'enregistrement des logs d'activités dans un bucket S3.
 - L'automatisation du traitement des logs avec AWS Lambda.
-

Plan des Exercices

Étape 1 : Création d'une API CRUD en Python

Objectifs :

- Développer une API REST en utilisant FastAPI ou Flask.
- Implémenter les endpoints suivants :
 - POST /users : Ajouter un utilisateur.
 - GET /users/{id} : Récupérer un utilisateur.
 - PUT /users/{id} : Modifier un utilisateur.
 - DELETE /users/{id} : Supprimer un utilisateur.
- Stocker temporairement les utilisateurs dans une base SQLite.

Tâches :

- Installer les dépendances (FastAPI, SQLite, SQLAlchemy).
 - Tester l'API en local avec Postman ou curl.
 - Ajouter des validations sur les entrées utilisateur (nom, email, âge).
-

Étape 2 : Migration vers AWS RDS (MySQL)

Objectifs :

- Créer une base MySQL sur AWS RDS.
- Connecter l'API à cette base.

Tâches :

- Créer une instance RDS MySQL sur AWS.
 - Configurer la connexion (pymysql ou SQLAlchemy).
 - Mettre à jour l'API pour enregistrer les utilisateurs dans RDS.
 - Vérifier les performances avec SELECT et EXPLAIN.
-

Étape 3 : Gestion des logs avec AWS S3

Objectifs :

- Enregistrer les logs des opérations utilisateur dans un fichier local.
- Sauvegarder automatiquement ces logs sur S3.

Tâches :

- Créer un bucket S3 sur AWS.
 - Utiliser `boto3` pour :
 - Uploader les logs.
 - Lister les fichiers du bucket.
 - Télécharger un fichier de log spécifique.
-

Étape 4 : Automatisation avec AWS Lambda**Objectifs :**

- Déclencher une fonction Lambda lorsqu'un fichier est ajouté à S3.
- Lire le fichier de log et analyser les opérations utilisateur.

Tâches :

- Développer une fonction Lambda en Python.
 - Configurer un trigger sur S3.
 - Stocker les statistiques dans une table MySQL.
-

Étape 5 : Analyse des logs avec AWS Athena & Glue**Objectifs :**

- Interroger les logs directement depuis S3.
- Automatiser la création du schéma avec AWS Glue.

Tâches :

- Configurer un Data Catalog avec Glue.
 - Exécuter des requêtes SQL avec Athena.
 - Analyser l'activité des utilisateurs.
-

Étape 6 : Déploiement sur AWS EC2**Objectifs :**

- Héberger l'API sur une instance EC2.

Tâches :

- Configurer une instance `t3.micro`.
 - Installer Python, l'API et `gunicorn`.
 - Tester l'API en ligne.
-

Finalisation du projet

- Ajouter des tests unitaires (`pytest`).
 - Documenter le projet (README.md, tutoriel d'installation).
 - Héberger le code sur GitHub avec un guide d'utilisation.
-

Technologies utilisées

- **Python** : API avec FastAPI/Flask, interactions AWS avec boto3.
 - **AWS** : S3, RDS (MySQL), Lambda, Glue, Athena, EC2.
 - **Base de données** : MySQL (RDS).
 - **Déploiement** : EC2 avec gunicorn et nginx.
-

Estimation du Temps

Étape	Temps estimé
API CRUD	4 - 6h
Migration RDS	6 - 8h
Gestion des logs S3	5 - 7h
Lambda	6 - 8h
Athena & Glue	6 - 8h
Déploiement EC2	8 - 12h
Total	40 - 55h (~1-2 semaines)

Résultat final : Un mini-projet concret démontrant tes compétences en **Python, AWS et bases de données**, idéal pour ton CV.