

Devoir AWS

Sujet : Architecture événementielle avec Dynamo, API Gateway et AWS Lambda

Participants

Arnaud NGABGNA

Mansour KEBE

Cheikh Ndiaye

Accès à la console

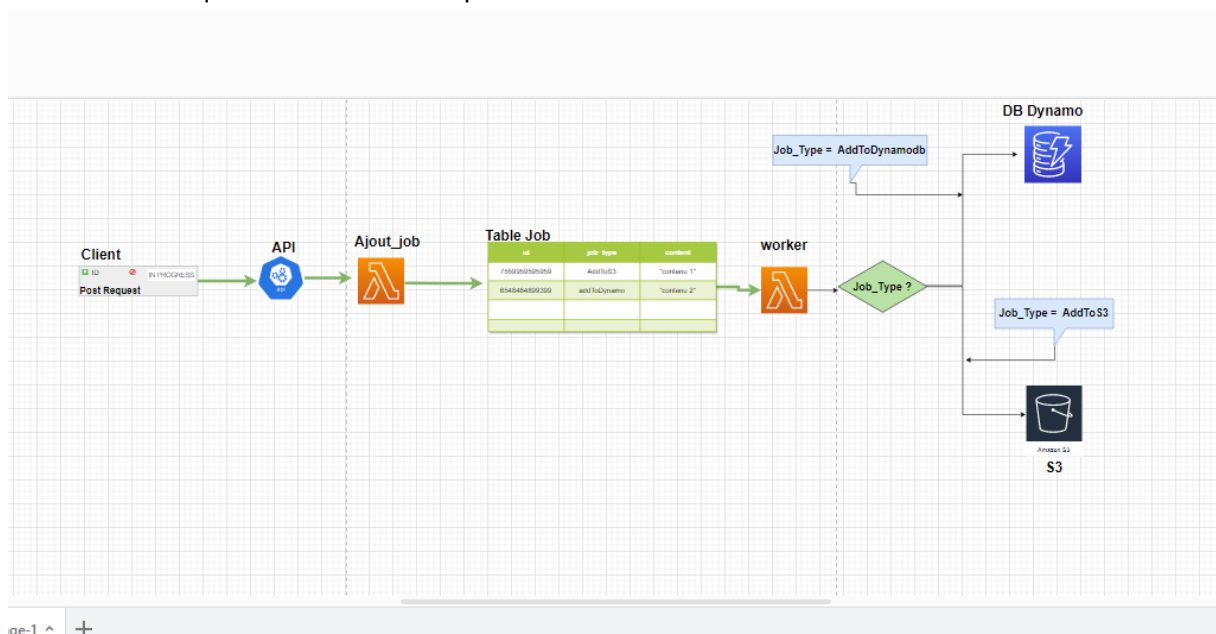
Login : ndiayecheikh0396@gmail.com

mot de passe : Etudes2016.\$

Introduction

Dans le cadre de ce TP, notre objectif est de créer une architecture événementielle en utilisant les services fournis par AWS tels que DynamoDB, API Gateway et AWS Lambda. Cette architecture nous permettra d'ajouter des jobs à une table DynamoDB via une API, et ensuite de déclencher le traitement de ces jobs en utilisant des lambdas appropriés en fonction de leur type.

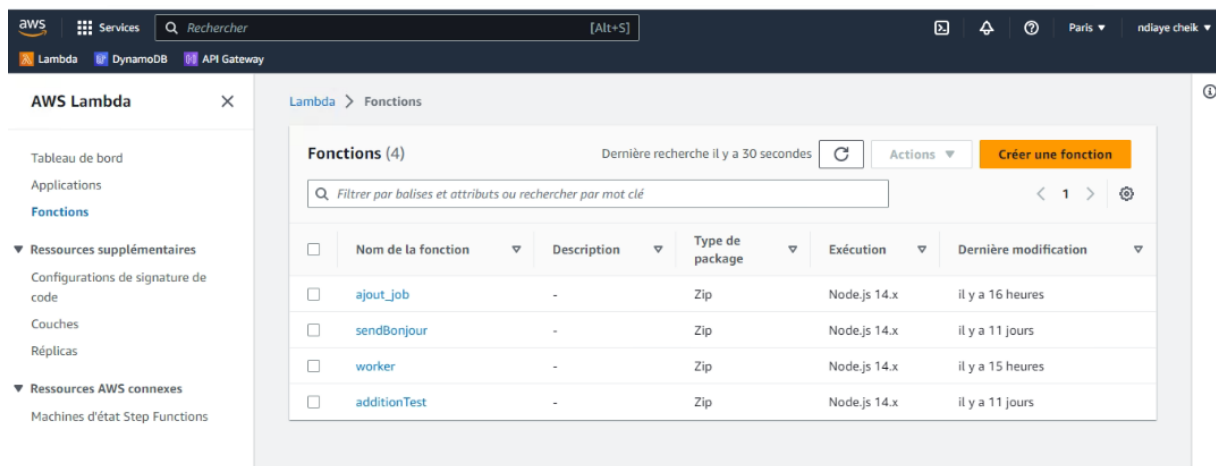
Le schéma de l'infrastructure nous offre une représentation visuelle de l'architecture événementielle que nous mettrons en place dans ce TP.



Étapes de résolution :

1. Schéma de l'infrastructure : Analysez attentivement le schéma de l'architecture événementielle proposée pour comprendre comment les différents services interagissent entre eux.
2. Création de l'API Gateway : Configurez et créez une API Gateway dans AWS. Cette API sera utilisée pour recevoir les requêtes d'ajout de jobs.
3. Création des tables DynamoDB : Concevez et configurez les tables DynamoDB nécessaires pour stocker les informations des jobs et leur état.
4. Création du fichier S3 : Créez un compartiment S3 pour stocker les fichiers associés aux jobs, le cas échéant.
5. Création des Lambdas et mise en relation avec les tables DynamoDB : Écrivez et configurez les fonctions Lambda appropriées pour traiter les jobs en fonction de leur type. Assurez-vous de les associer correctement aux tables DynamoDB correspondantes pour récupérer les données nécessaires.
6. Déploiement de l'infrastructure avec Terraform : Utilisez Terraform pour déployer et gérer l'infrastructure AWS. Écrivez les fichiers de configuration Terraform nécessaires pour créer et configurer les ressources AWS requises, notamment l'API Gateway, les tables DynamoDB, le compartiment S3 et les fonctions Lambda.
7. Test et validation : Effectuez des tests pour vérifier le bon fonctionnement de l'ensemble de l'infrastructure. Vérifiez que l'ajout de jobs via l'API déclenche le traitement approprié des Lambdas et que les données sont stockées et récupérées correctement dans les tables DynamoDB.

1. La création des lambda



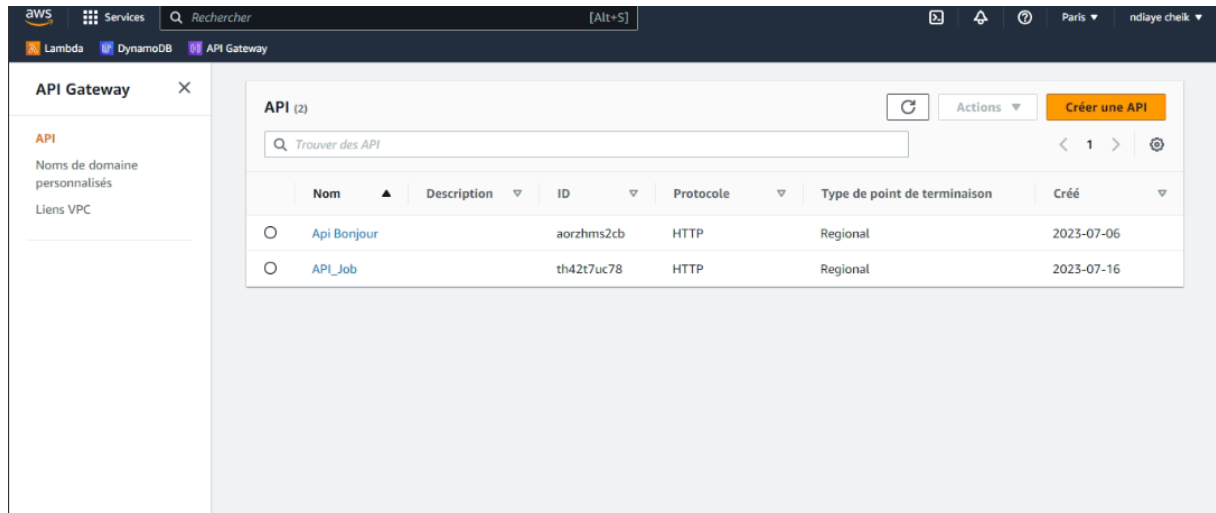
Dans le cadre de ce TP nous avons deux fonctions Lambda:

-Ajout_job: chargé d'enregistrer les données dans la table_job

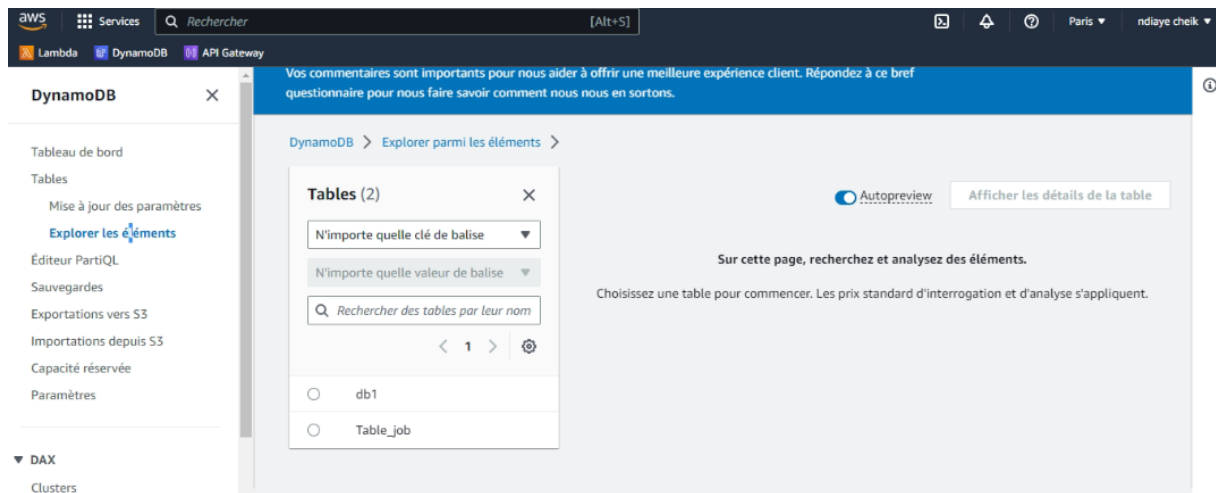
-Worker qui en fonction de ce qui est dans la colonne job_type, alimenté soit la base de données, soit le bucket S3

2. La création de l'API Gateway

Nous avons une API_Job qui fait appel à la fonction lambda ajout_job



3. La création des DynamoDB



3. La création du bucket S3

The screenshot displays the AWS Management Console for Amazon S3. The top navigation bar includes the AWS logo, a search bar, and user information. The left sidebar lists various S3 features, with 'Compartiments' (Buckets) selected. The main panel shows the 'Compartiments (1)' section, indicating one bucket is present. Below this, a table lists the bucket details:

Nom	Région AWS	Accéder	Date de création
mansouramaudcheikh	EU (Paris) eu-west-3	Compartiment et objets non publics	03 Jul 2023 12:28:03 PM CEST