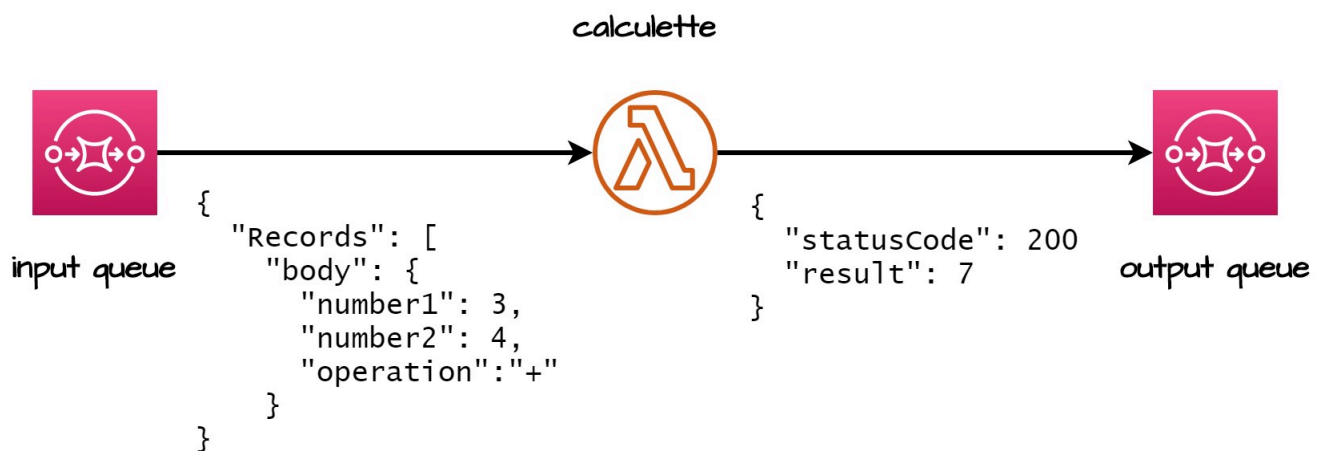


Attendu

Cet exercice est à faire par groupe de 2 max. Vous pouvez ainsi le faire seul ou à deux. Vous noterez les membres du groupe dans un commentaire en en-tête du fichier `main.py`. Vous rendrez une Moodle une archive .zip contenant le code Python de la fonction Lambda et le code de l'infrastructure. **Attention votre code doit fonctionner tel quel.** C'est-à-dire qu'il suffit de faire un `cdktf deploy` pour tout lancer.

Sujet

Dans cet exercice vous allez automatiser le déploiement du code final du TP3 via `cdktf`. Néanmoins, vous allez devoir intégrer un léger changement à votre application. En effet il n'est pas possible avec `cdktf` de définir une cible pour une fonction Lambda. À la place, vous allez poster les messages dans la queue SQS directement depuis le code de la Lambda.



Pour faire cela vous allez utiliser le module `boto3` qui permet via Python d'interagir avec les services AWS. Voici un code minimal pour poster dans une file SQS depuis Python :

```
import boto3
sqs = boto3.client('sqs') #client is required to interact with
sqs.send_message(
    QueueUrl="https://sqs.us-east-1.amazonaws.com/XXXXXXX/YYYYYY", # Doit être remplacé la
    bonne URL
    MessageBody="Hello world from boto3"
)
```

Pour résoudre cet exercice :

1. Créez via le terminal un projet `cdktf`. Créez un dossier `graded_lab` et ouvrez ce dossier dans VScode. Dans un terminal saisissez la commande `cdktf init --template="Python" --providers="aws@~>4.0" --local` et validez les différentes questions (voir ce [lien](#)). Changez l'interpréteur Python avec un `ctrl+shift+p` dans VS code et cherchez `Python: Select Interpreter` et prenez celui qui pointe vers un `virtualenv`. Il a été créé pas Terraform et contient toutes les dépendances pour l'exercice (voir ce [lien](#))
2. Commencez par créer les services nécessaires (les deux files, la Lambda) en vous inspirant du cours. Déployez-les et vérifiez que tout se crée correctement avec la commande `cdktf deploy`

3. Récupérez l'URL de la file qui vous servira d'output et ajoutez-la comme variable environnement de votre Lambda (cf cours). Vous pouvez accéder à l'URL de votre file via son attribut `url` (cf TP2 quand vous alliez chercher l'id de certains services)
4. Mettez à jour le code de votre fonction Lambda en utilisant le code ci-dessus et en récupérant l'URL de votre file via `os.getenv()`. Faites que votre code boucle sur les messages s'il y en a plusieurs.
5. Redéployez votre application et testez-la.