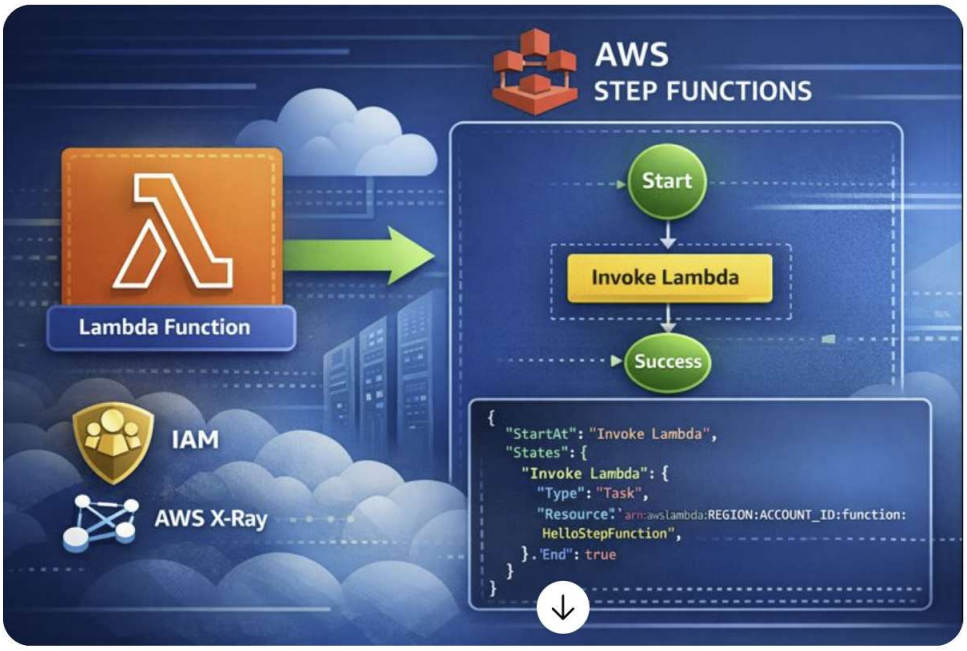
AWS Hands-on Lab | Lambda (Python) & Step Functions



Réalisation d'un lab très simple pour comprendre le fonctionnement des **AWS Step Functions** et leur intégration avec **AWS Lambda**.

Objectif :

- Créer une fonction Lambda minimale, codée en Python
- Orchestrer son exécution avec une **State Machine (Standard)**
- Comprendre le workflow visuel, le code JSON généré automatiquement et les rôles IAM associés
- Tester l'exécution et analyser le résultat

Services utilisés :

- AWS Lambda (Python 3.11)
- AWS Step Functions (Standard)
- IAM
- AWS X-Ray (permissions)

Sommaire :

1 : Créer une fonction Lambda simple 2

2 : Créer un ''state machine'' (Step function) 2

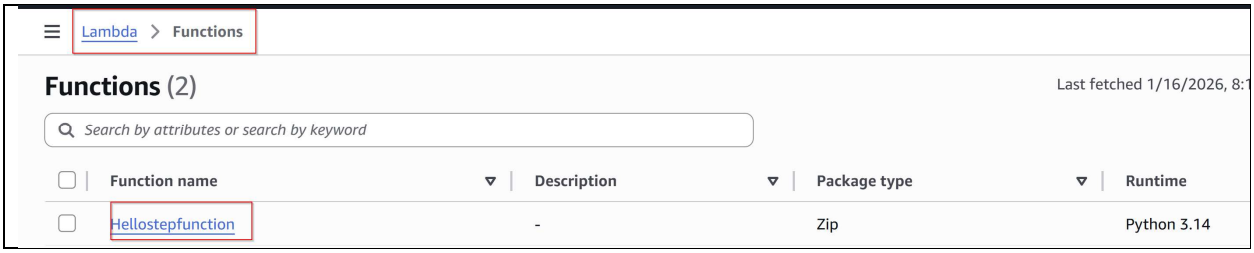
3 Créer la Step Function. 4
4 : Tester la Step Function 6

1 : Créer une fonction Lambda simple

Dan la console AWS, Aller sur AWS Lambda → Create Function → Author from scratch.
Nom : HelloStepFunction.
Runtime : Python 3.11 (ou Node.js si tu préfères).
Dans le code Lambda, mettons quelque chose de minimal :

```
def lambda_handler(event, context):
    return {
        "statusCode": 200,
        "message": "Hello from Step Function!"
    }
```

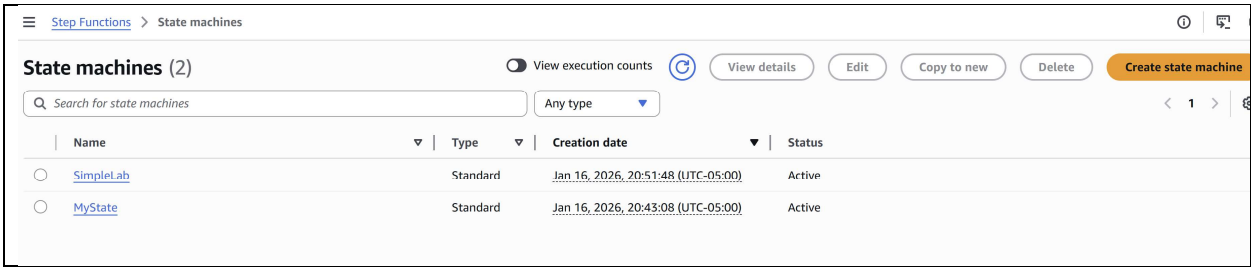
Déployer la fonction.

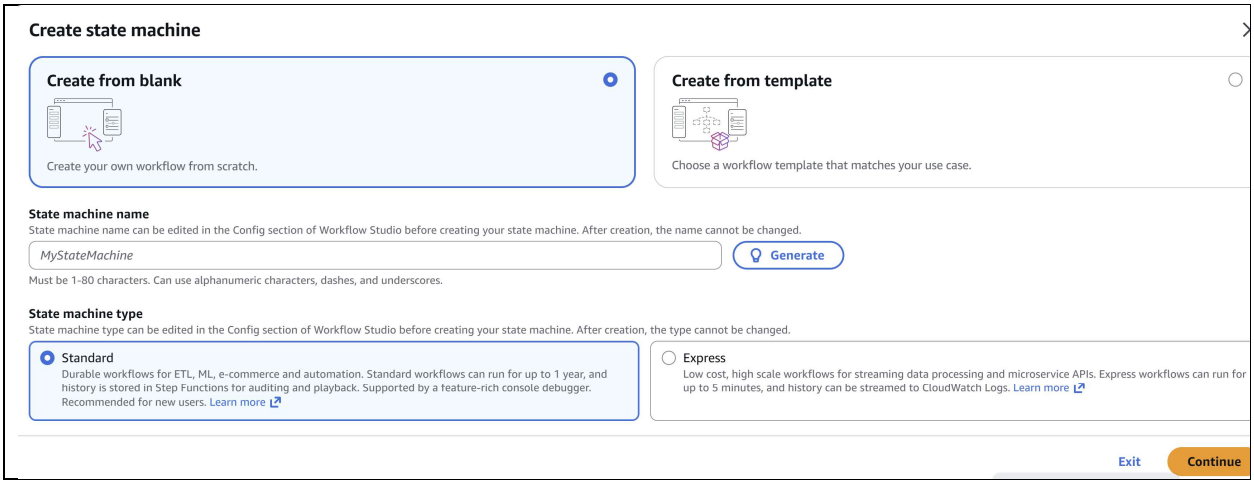


---

2 : Créer un ‘’state machine’’ (Step function)

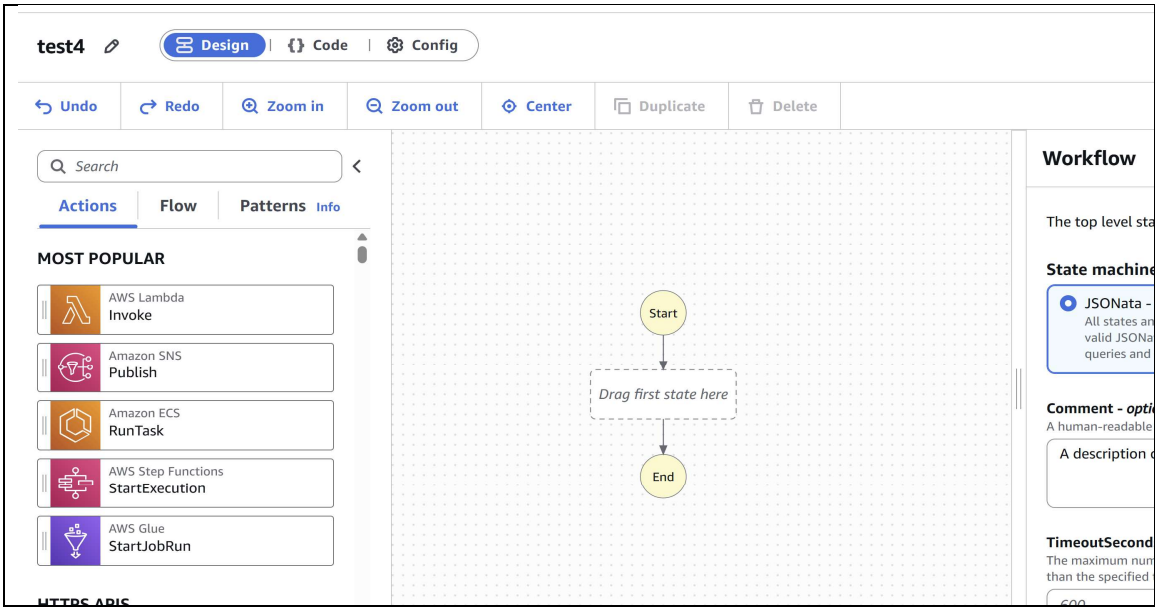
Dans la console AWS, Aller sur Step Functions → Create state machine → Standard.et ‘’create from blank’’
Nom : SimpleLab.
Choisir : Design your workflow visually.



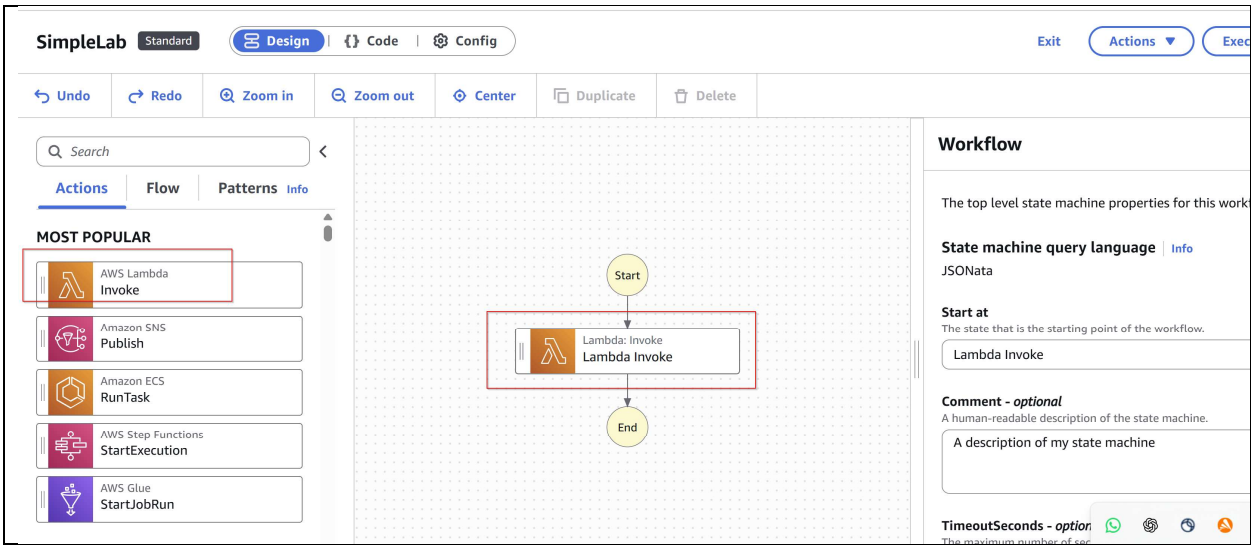


Cliquer sur ‘’continuer’’

Ajouter un état Task :
Resource : votre function Lambda HelloStepFunction.
Pour cela faire glisser la fonction du pannel de gauche, intitulée : ‘’AWS Lambda invoke’’ vers le graphique dans le champs : ‘’drag first state here’’. Le ‘’first state’’ correspond au premier ‘’état’’ qui est : ‘’Lambda Invoke’’.



Cela remplis automatiquement le panel, intitulé ‘’workflow sur la droite’’ avec les données de la fonction lambda. Voir le graphique ci-dessous.



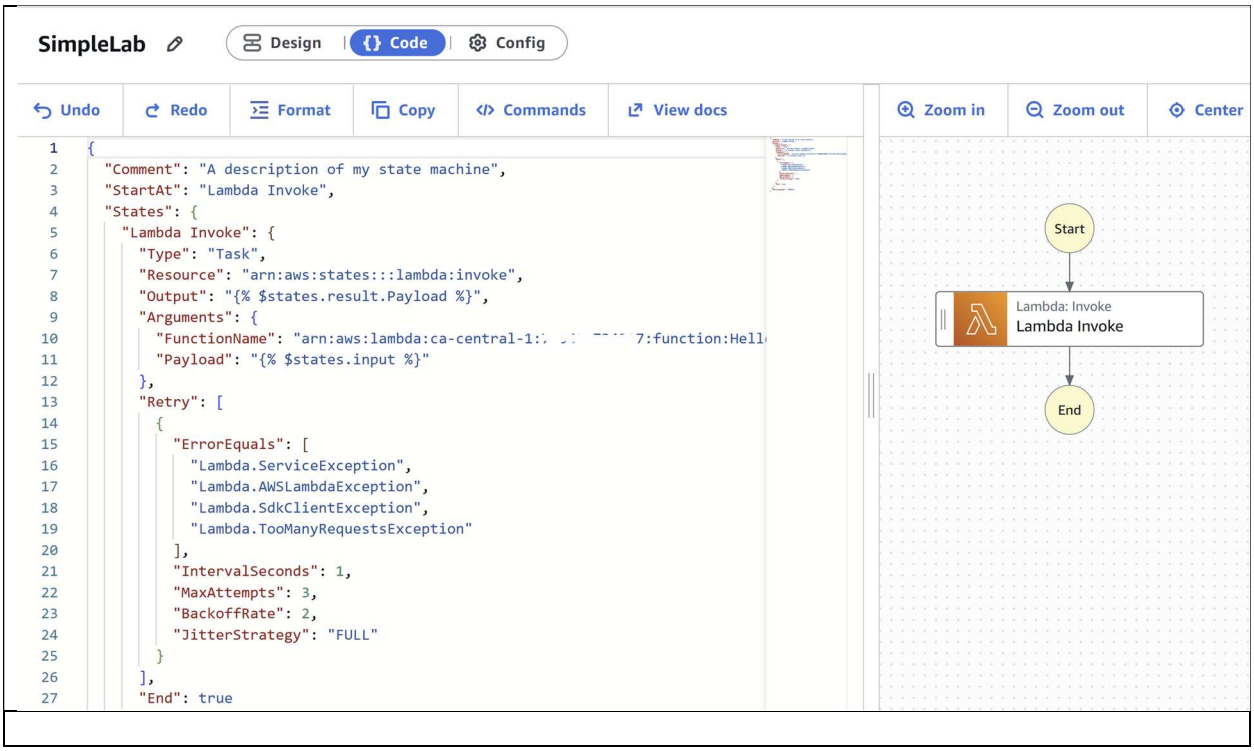
Exemple de définition JSON très simple :

```
{
 "Comment": "A simple Step Function",
 "StartAt": "HelloTask",
 "States": {
  "HelloTask": {
   "Type": "Task",
   "Resource": "arn:aws:lambda:REGION:ACCOUNT_ID:function:HelloStepFunction",
   "End": true
  }
 }
}
```

On n’utilisera pas ce code Json , mais on utilisera a la pace le code Json , décrit au paragraphe suivant.

3 Créer la Step Function.

L’outil AWS Step function workflow , génère du code Json automatiquement, que l’on peut voir dans la fenêtre ‘’code’’. Le code Json appelle (invoke) la fonction lambda crée précédemment.

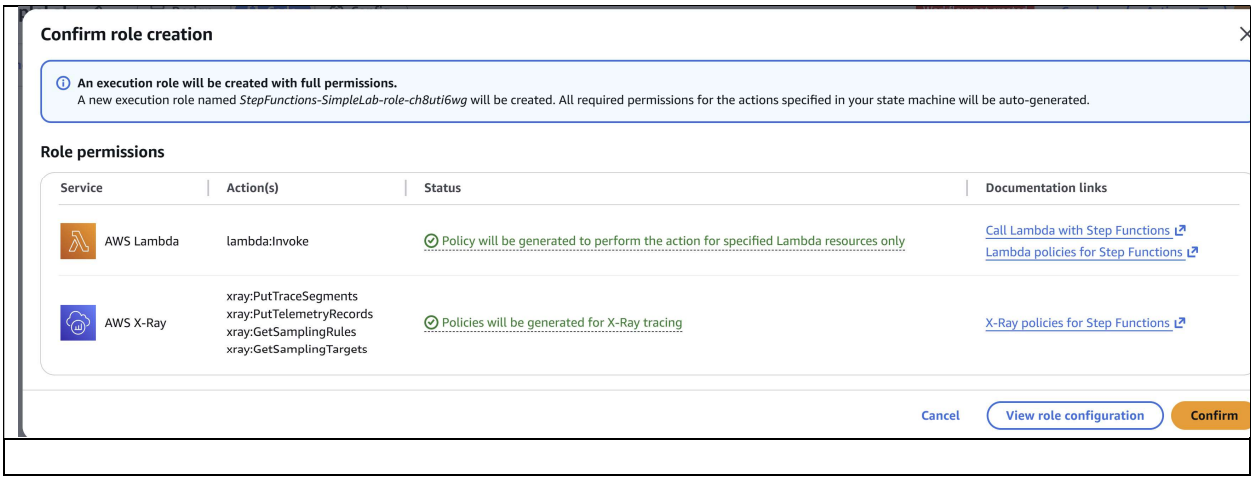


Dans l'URL :
"arn:aws:lambda:ca-central-1: **ACCOUNT_ID**:function:Hellostepfunction:$LATEST"

A la place de **ACCOUNT_ID** , on aura l'ID de votre compte AWS

Cliquer sur le bouton ‘’create”

Lorsqu'on clique sur ‘’create’’ , une autre fenêtre s'Affiche qui demande confirmation pour ajouter des rôles et permissions pour les rôles, afin que AWS Step Function puisse exécuter la fonction Lambda et aussi avoir le droit d'utiliser le service AWS X-Ray pur le debugging.



Cliquer sur ‘’confirmation ‘’

On peut aussi cliquer sur le bouton ‘’view rôle configuration’’ afin de voir le détail des rôles.

---

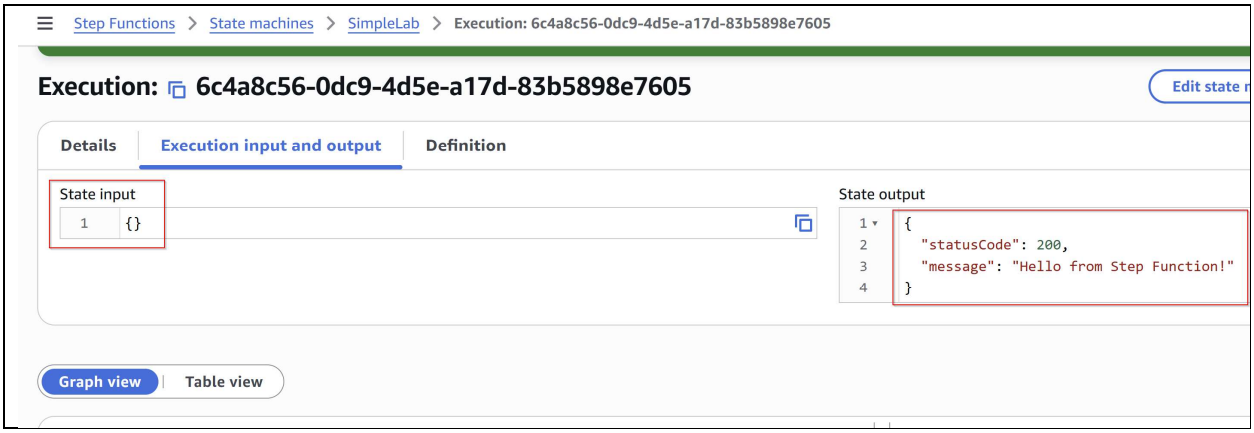
4 : Tester la Step Function

Cliquer sur Start execution.
La fenetre suivante s’Affiche :

Pour ce premier test , laisser entrée : {} (vide).

Résultat attendu :

{
"statusCode": 200,
"message": "Hello from Step Function!"



Vérifier l’état des boutons sur le graphique :

Le status ‘’vert” indique le succès de l’operation