

Passo a passo

1. Pré-requisitos

- Conta AWS com permissões para **Lambda**, **Step Functions**, **IAM** e **CloudWatch**.
- Escolha e use **a mesma Region** para Lambda e Step Functions.

2. Criar a função Lambda

- Console AWS → **Lambda** → **Create function** → **Author from scratch**.
 - **Name:** hello-stepfunctions
 - **Runtime:** ex.: Python 3.12 ou Node.js 20.x
 - **Permissions:** *Create a new role with basic Lambda permissions (AWSLambdaBasicExecutionRole)*
- Em **Code**, cole e **Deploy**.

```
# Python (lambda_handler)
def lambda_handler(event, context):
    name = event.get("name", "Mundo")
    return {
        "statusCode": 200,
        "body": f"Olá, {name}!",
        "input": event
    }

// Node.js (exports.handler)
exports.handler = async (event) => {
    const name = event?.name || "Mundo";
    return { statusCode: 200, body: `Olá, ${name}!`, input: event };
};
```

- **Test:** Create Test event { "name": "Nathana" } → Run → confirmar resposta 200.

3. Criar role IAM para Step Functions

- Console → **IAM** → **Roles** → **Create role**.
 - **Trusted entity:** AWS service → **Step Functions**.
 - Anexe policy personalizada (exemplo — substitua REGION e ACCOUNT_ID e nome da função):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "InvokeLambda",
      "Effect": "Allow",
      "Action": ["lambda:InvokeFunction"],
      "Resource": "arn:aws:lambda:REGION:ACCOUNT_ID:function:hello-stepfunctions"
    },
    {
      "Sid": "Logs",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",

```

```

    "logs:CreateLogStream",
    "logs:PutLogEvents"
  ],
  "Resource": "*"
}
]
}

```

- Nomeie a role (ex.: SFN-InvokeLambda-Role) e **Create role**.

4. Criar a State Machine (Step Functions)

- Console → **Step Functions** → **Create state machine** → **Author with code**.
- Cole a definição (substitua REGION/ACCOUNT_ID/function:hello-stepfunctions):

```

{
  "Comment": "Invoca Lambda e retorna resultado",
  "StartAt": "InvocarLambda",
  "States": {
    "InvocarLambda": {
      "Type": "Task",
      "Resource": "arn:aws:states:::lambda:invoke",
      "Parameters": {
        "FunctionName": "arn:aws:lambda:REGION:ACCOUNT_ID:function:hello-
stepfunctions",
        "Payload.$": "$"
      },
      "End": true
    }
  }
}

```

- **Role for executions:** selecione SFN-InvokeLambda-Role.
- **Name:** hello-lambda-sfn → **Create state machine**.

5. Iniciar execução

- Dentro da state machine → **Start execution**.
- **Input:**

```

{ "name": "Nathana" }

```

- Clique **Start execution**.
- Acompanhe em **Graph view** (estado ficará verde). Veja **Output** da execução.

6. Verificar logs / resultado

- **Output:** painel da execução (Execution → Output).
- **Logs da Lambda:** Lambda → **Monitor** → **View logs in CloudWatch**.
- **Logs do Step Functions** (se habilitado): CloudWatch Logs (grupo criado conforme sua configuração de logging).

7. Erros comuns rápidos

- AccessDeniedException → role da State Machine precisa de lambda:InvokeFunction para o ARN correto.
- Payload vazio/incorreto → use "Payload.\$": "\$" para repassar todo o input.
- Timeout da Lambda → aumente Timeout na configuração da função.
- Region diferente → confirme mesma Region para Lambda e State Machine.

8. Limpeza (opcional)

- Delete a **state machine**, delete a **Lambda function** e a **IAM role** criada se não forem mais necessárias.