# Serverless Data Processing (CSCI 5410)

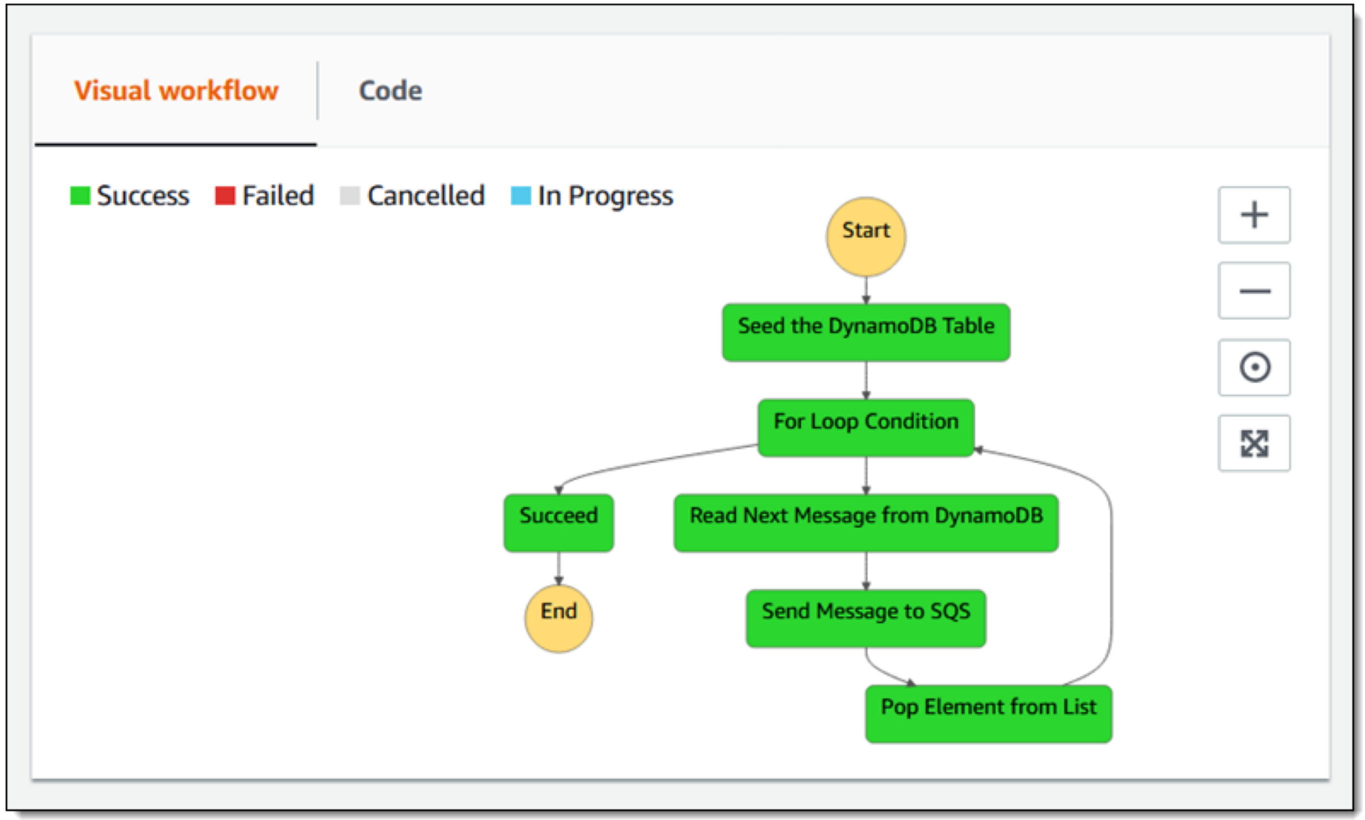
Dr. Saurabh Dey

## Outline

AWS Step Functions

# Step Functions

- AWS Step Functions is a web service that enables us to coordinate components of distributed applications and microservices using visual workflows.



https://aws.amazon.com/blogs/aws/new-compute-database-messaging-analytics-and-machine-learning-integration-for-aws-step-functions/

## Overview of Step Functions

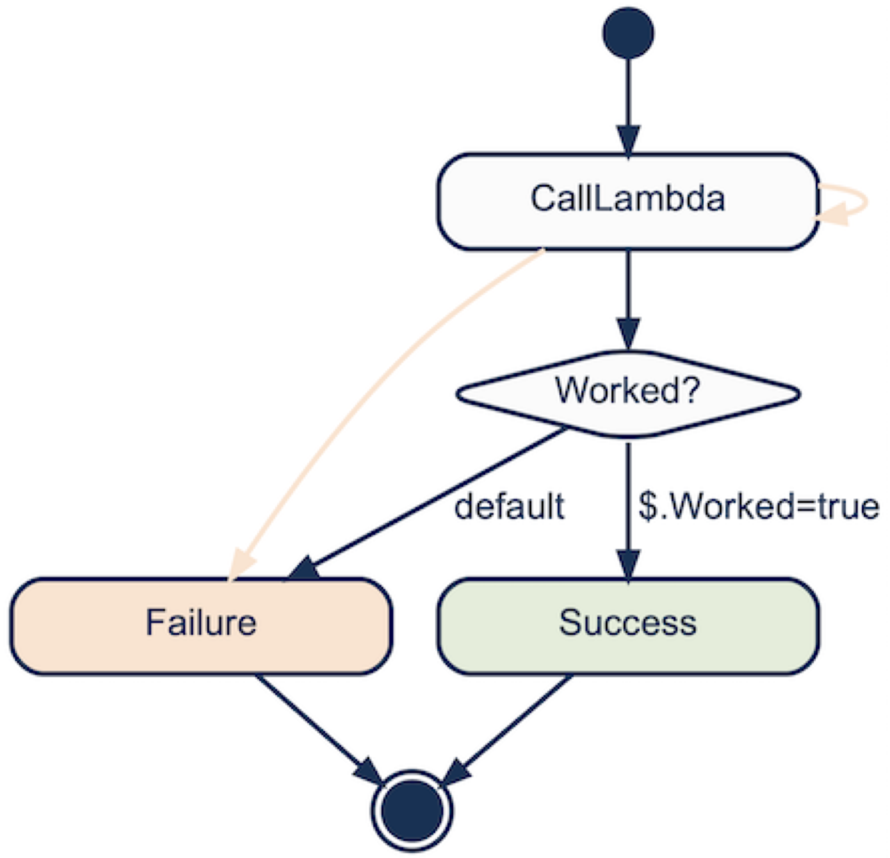
- Concepts of Tasks and State machines
- JSON-based Amazon State Language
- Monitor using graphical view

# What is a State Machine?

- A state machine contains finite number of states and some input-output.

- States are elements in the state machine.
    - Must be unique within the scope of the entire state machine.

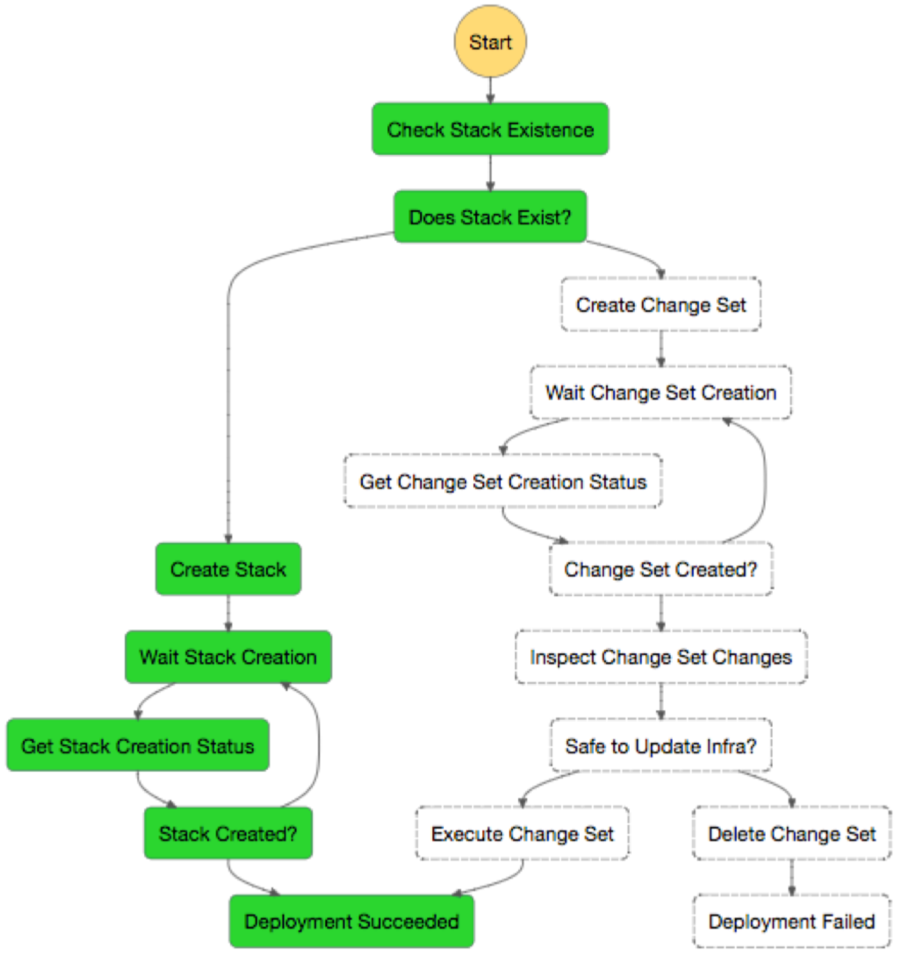
- Individual states can make decisions based on their input, perform actions, and pass output to other states.



https://blog.coinbase.com/aws-step-functions-state-machines-bifrost-and-building-deployers-5e3745fe645b

# What can states perform in a state machine?

- A state can perform some tasks
- Make a choice between branches of execution
  - E.g. Stop an execution with success or a failure
- Simply pass its input to its output
- Provide a delay for a certain amount of time
- Begin parallel branches of execution



https://aws.amazon.com/blogs/devops/using-aws-step-functions-state-machines-to-handle-workflow-driven-aws-codepipeline-actions/

# Features shared by States

- A **Type** field associated with each state indicates its type.
- An optional **Comment** field, which indicates human-readable description
- Each state (except a **Succeed** or **Fail** state) requires a **Next** field or, alternatively, can become a terminal state by specifying an **End** field.
- A **Choice** state may have more than one **Next**, but only one within each Choice Rule. A **Choice** state cannot use **End**.

**Current State**

**Next State**

```
"HelloWorld": {
"Type": "Task",
"Resource": "arn:aws:lambda:us-east-1:123456789012:function:HelloFunction",
"Next": "AfterHelloWorldState",
"Comment": "Run the HelloWorld Lambda function"
}
```

https://docs.aws.amazon.com/step-functions/latest/dg/concepts-states.html

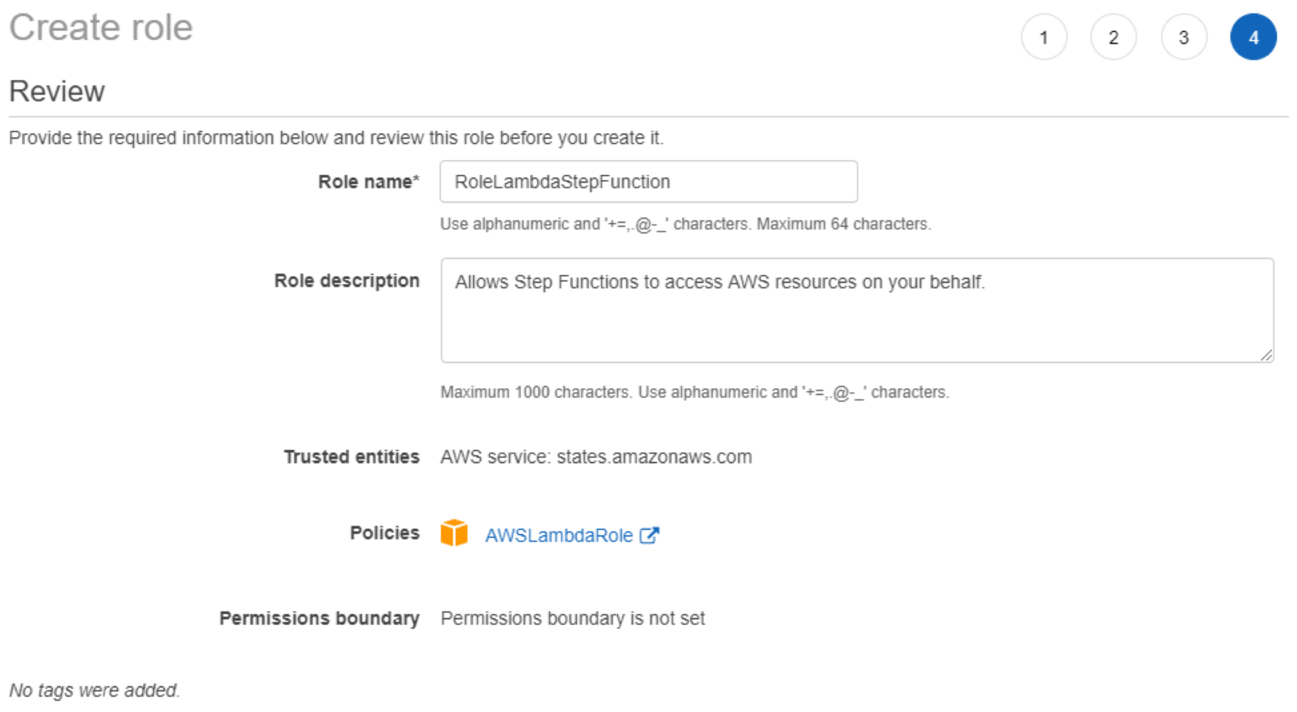
# Structure of a State Machine

```
{
        "State1" : {
        },
        "State2" : {
        },
        ...
}
```

- State machines are defined using JSON text that represents a structure.
- Some fields, such as (**StartAt**, **States**) are required
- Other fields (**Comment**, **TimeoutSeconds**, **Version**) are optional

```
{
  "Comment": "A Hello World example of the Amazon States Language using a Pass state",
  "StartAt": "HelloWorld",
  "States": {
    "HelloWorld": {
      "Type": "Pass",
      "Result": "Hello World!",
      "End": true
    }
  }
}
```

# Create a State Machine (1-3)


Create role

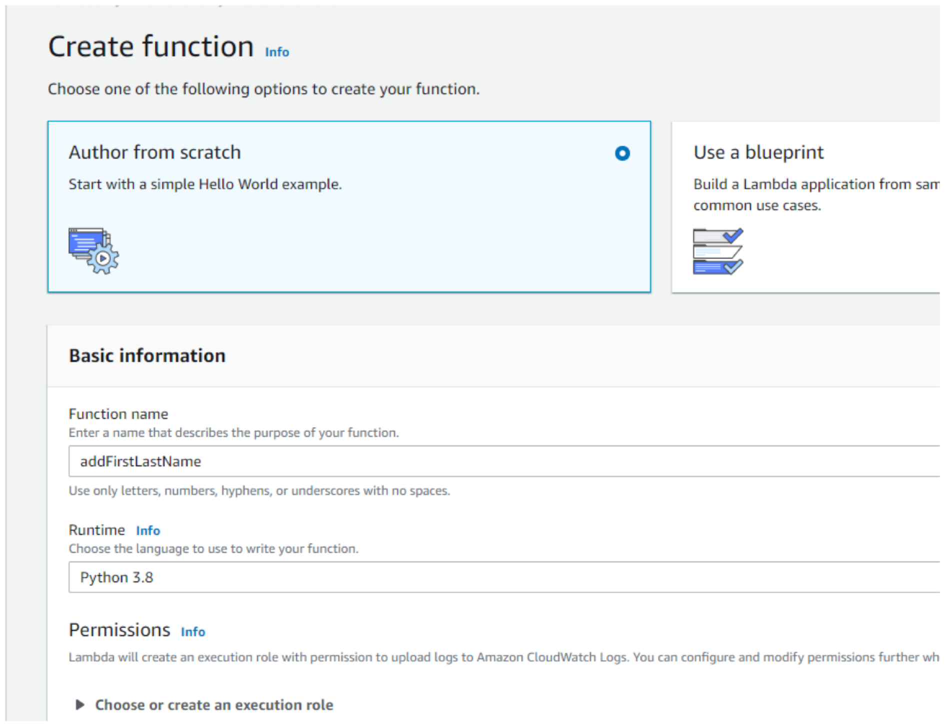
1 2 3 4

Review

Provide the required information below and review this role before you create it.

| | |
|---|---|
| Role name* | RoleLambdaStepFunction |
| | Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters. |
| Role description | Allows Step Functions to access AWS resources on your behalf. |
| | Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters. |
| Trusted entities | AWS service: states.amazonaws.com |
| Policies | AWSLambdaRole |
| Permissions boundary | Permissions boundary is not set |

*No tags were added.*


Create function Info

Choose one of the following options to create your function.

Author from scratch [✓]
Start with a simple Hello World example.

Use a blueprint
Build a Lambda application from sam
common use cases.

**Basic information**

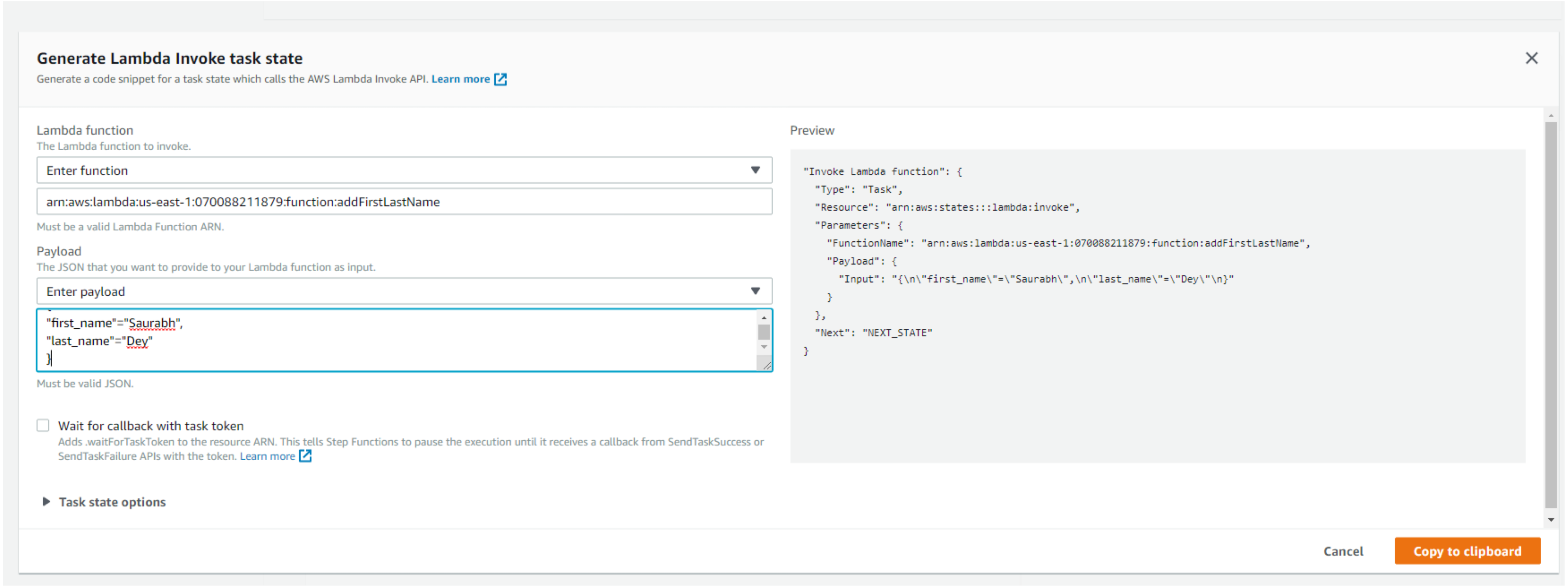
Function name
Enter a name that describes the purpose of your function.
addFirstLastName
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function.
Python 3.8

Permissions Info
Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further wh

▶ Choose or create an execution role

# Create a State Machine (2-3)



Generate a code snippet for a task state which calls the AWS Lambda Invoke API. Learn more

Lambda function
The Lambda function to invoke.

Enter function

arn:aws:lambda:us-east-1:070088211879:function:addFirstLastName

Must be a valid Lambda Function ARN.

Payload
The JSON that you want to provide to your Lambda function as input.

Enter payload

```
"first_name"="Saurabh",
"last_name"="Dey"
}
```

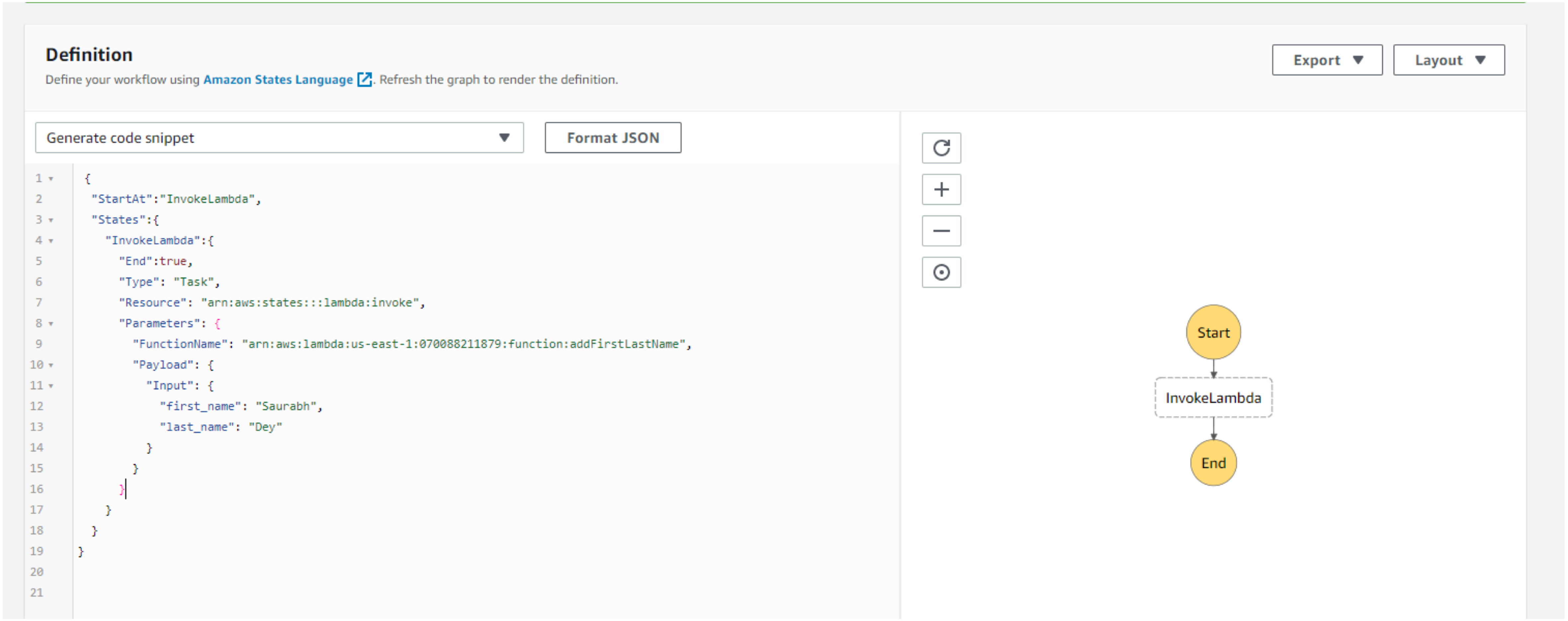
Must be valid JSON.

Wait for callback with task token
Adds .waitForTaskToken to the resource ARN. This tells Step Functions to pause the execution until it receives a callback from SendTaskSuccess or SendTaskFailure APIs with the token. Learn more

Preview

```
"Invoke Lambda function": {
  "Type": "Task",
  "Resource": "arn:aws:states:::lambda:invoke",
  "Parameters": {
    "FunctionName": "arn:aws:lambda:us-east-1:070088211879:function:addFirstLastName",
    "Payload": {
      "Input": "{\n\"first_name\"=\"Saurabh\",\n\"last_name\"=\"Dey\"\n}"
    }
  },
  "Next": "NEXT_STATE"
}
```

# Create a State Machine (3-3)



**Definition**

Define your workflow using **Amazon States Language**. Refresh the graph to render the definition.

```
{
 "StartAt":"InvokeLambda",
 "States":{
   "InvokeLambda":{
     "End":true,
     "Type": "Task",
     "Resource": "arn:aws:states:::lambda:invoke",
     "Parameters": {
       "FunctionName": "arn:aws:lambda:us-east-1:070088211879:function:addFirstLastName",
       "Payload": {
         "Input": {
           "first_name": "Saurabh",
           "last_name": "Dey"
         }
       }
     }
   }
 }
}
```

## Questions to Consider

- How does AWS step function help in debugging?

- Is it possible to automate a process using AWS step function?