

Use case

Build a Step Function for Order Workflow

Use case Description

Schedule Lambda every evening and stop running Ec2 instances to reduce billing

In this use case, we create a step function state machine to understand how the workflow actually runs

1. For order workflow, let's say there are four stages Validate order, Process Payment, Ship order, Notify customer
2. Each stage has its own functionality (lambda code)
3. So to orchestrate these steps, we use step functions

The image displays two screenshots from the AWS console. The top screenshot shows the AWS Lambda 'Functions' page with a list of four functions: `Lambda_OrderFns_NotifyCustomer`, `Lambda_OrderFns_ValidateOrder`, `Lambda_OrderFns_ProcessPayment`, and `Lambda_OrderFns_ShipOrder`. All functions are of type 'Zip' and use 'Python 3.12' runtime. The bottom screenshot shows the AWS Step Functions 'OrderProcessingWorkflow' in 'Design' mode. The workflow diagram consists of a linear sequence of five steps: 'Start' (yellow circle), 'Validate Order' (Lambda Invoke), 'Process Payment' (Lambda Invoke), 'Ship Order' (Lambda Invoke), and 'Notify Customer' (Lambda Invoke), ending with an 'End' (yellow circle). The right-hand panel shows the 'Workflow' definition, including the state machine query language (JSONPath), the start state ('Validate Order'), a human-readable comment ('Order Processing Workflow'), and an optional timeout of 600 seconds.

Functions (21)

Function name	Description	Package type	Runtime	Last modified
Lambda_OrderFns_NotifyCustomer	-	Zip	Python 3.12	3 days ago
Lambda_OrderFns_ValidateOrder	-	Zip	Python 3.12	3 days ago
Lambda_OrderFns_ProcessPayment	-	Zip	Python 3.12	3 days ago
Lambda_OrderFns_ShipOrder	-	Zip	Python 3.12	3 days ago

Step Functions

OrderProcessingWorkflow Standard Design Code Config

Workflow Definition

The top level state machine properties for this workflow. [Learn more](#)

State machine query language [Info](#)

JSONPath

Start at

The state that is the starting point of the workflow.

Validate Order

Comment - optional

A human-readable description of the state machine.

Order Processing Workflow

TimeoutSeconds - optional

The maximum number of seconds an execution of the state machine can run. If it runs longer than the specified time, the execution fails with a `States.Timeout`.

600

AWS Step Functions console showing the **OrderProcessingWorkflow** in **Code** view. The workflow is defined as follows:

```
1 {
2   "Comment": "Order Processing WorkFlow",
3   "StartAt": "Validate Order",
4   "States": {
5     "Validate Order": {
6       "Type": "Task",
7       "Resource": "arn:aws:lambda:ap-south-1:770424767712:function:Lambda_OrderFns_V",
8       "Next": "Process Payment"
9     },
10    "Process Payment": {
11      "Type": "Task",
12      "Resource": "arn:aws:lambda:ap-south-1:770424767712:function:Lambda_OrderFns_P",
13      "Next": "Ship Order"
14    },
15    "Ship Order": {
16      "Type": "Task",
17      "Resource": "arn:aws:lambda:ap-south-1:770424767712:function:Lambda_OrderFns_S",
18      "Next": "Notify Customer"
19    },
20    "Notify Customer": {
21      "Type": "Task",
22      "Resource": "arn:aws:lambda:ap-south-1:770424767712:function:Lambda_OrderFns_N",
23      "End": true
24    }
25  }
26 }
```

The visual representation shows a linear flow: **Start** → **Validate Order** → **Process Payment** → **Ship Order** → **Notify Customer** → **End**.

AWS Step Functions console showing the **OrderProcessingWorkflow** in **Graph view** for execution **1f41126a-4739-4135-90af-b2cf4498cb4b**. The workflow is shown as a sequence of tasks: **Validate Order**, **Process Payment**, **Ship Order**, and **Notify Customer**, all of which are **Succeeded**.

Notify Customer details:

- Input/Output** tab selected.
- State input:**

```
1 {
2   "status": "Order Shipped",
3   "order_id": "1234",
4   "tracking_id": "TRAC12345"
5 }
```
- State output:**

```
1 {
2   "status": "Customer Notified",
3   "order_id": "1234"
4 }
```