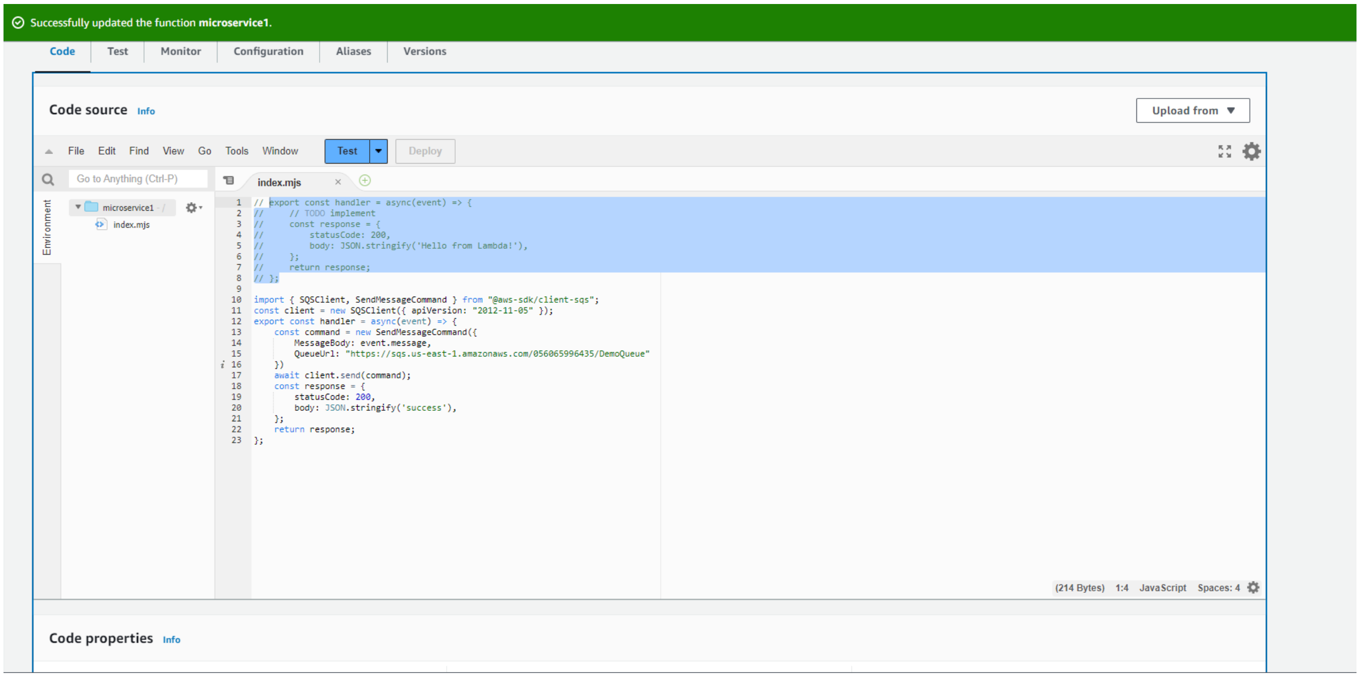


Successfully updated the function **microservice1**.

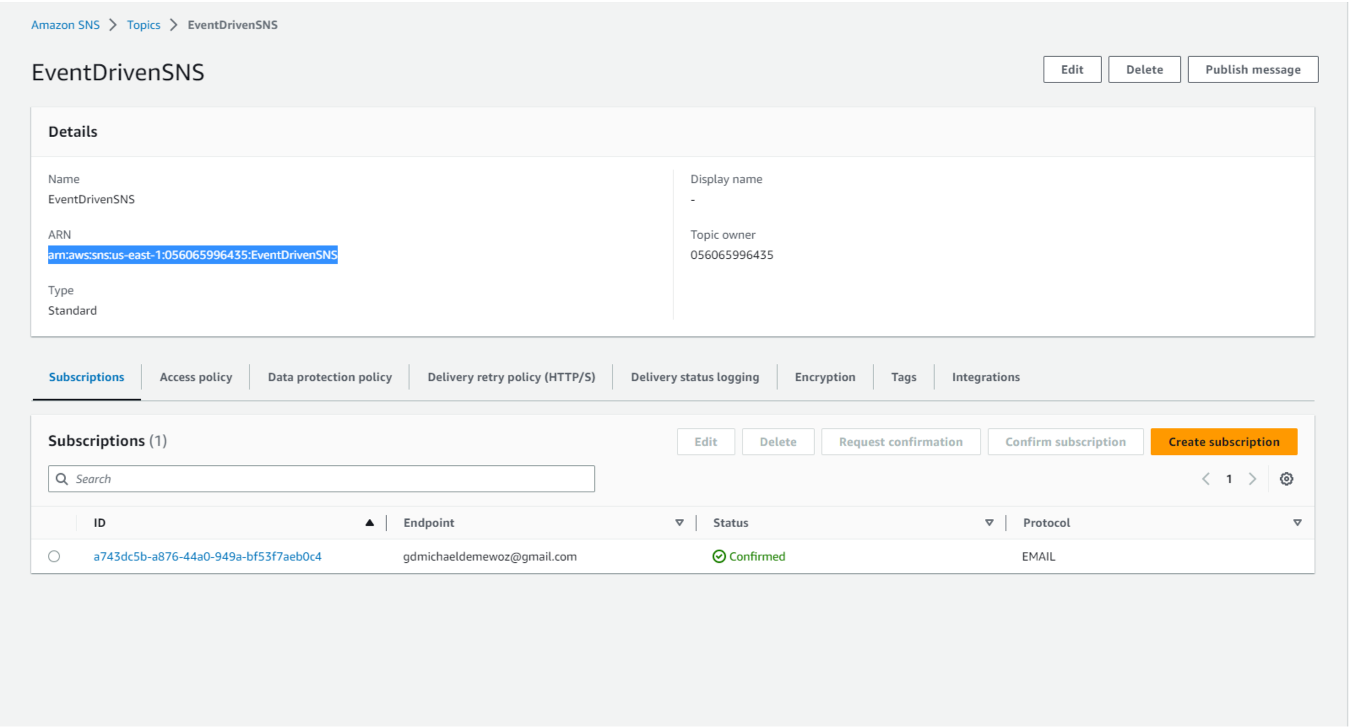
Code source Info

index.mjs

```javascript
// export const handler = async(event) => {
//     // TODO implement
//     const response = {
//         statusCode: 200,
//         body: JSON.stringify('Hello from Lambda!'),
//     };
//     return response;
// };

import { SQSClient, SendMessageCommand } from "@aws-sdk/client-sqs";
const client = new SQSClient({ apiVersion: "2012-11-05" });
export const handler = async(event) => {
    const command = new SendMessageCommand({
        MessageBody: event.message,
        QueueUrl: "https://sqs.us-east-1.amazonaws.com/056065996435/DemoQueue"
    })
    await client.send(command);
    const response = {
        statusCode: 200,
        body: JSON.stringify('success'),
    };
    return response;
};
```

Code properties Info



EventDrivenSNS

Details

Name
EventDrivenSNS

ARN
arn:aws:sns:us-east-1:056065996435:EventDrivenSNS

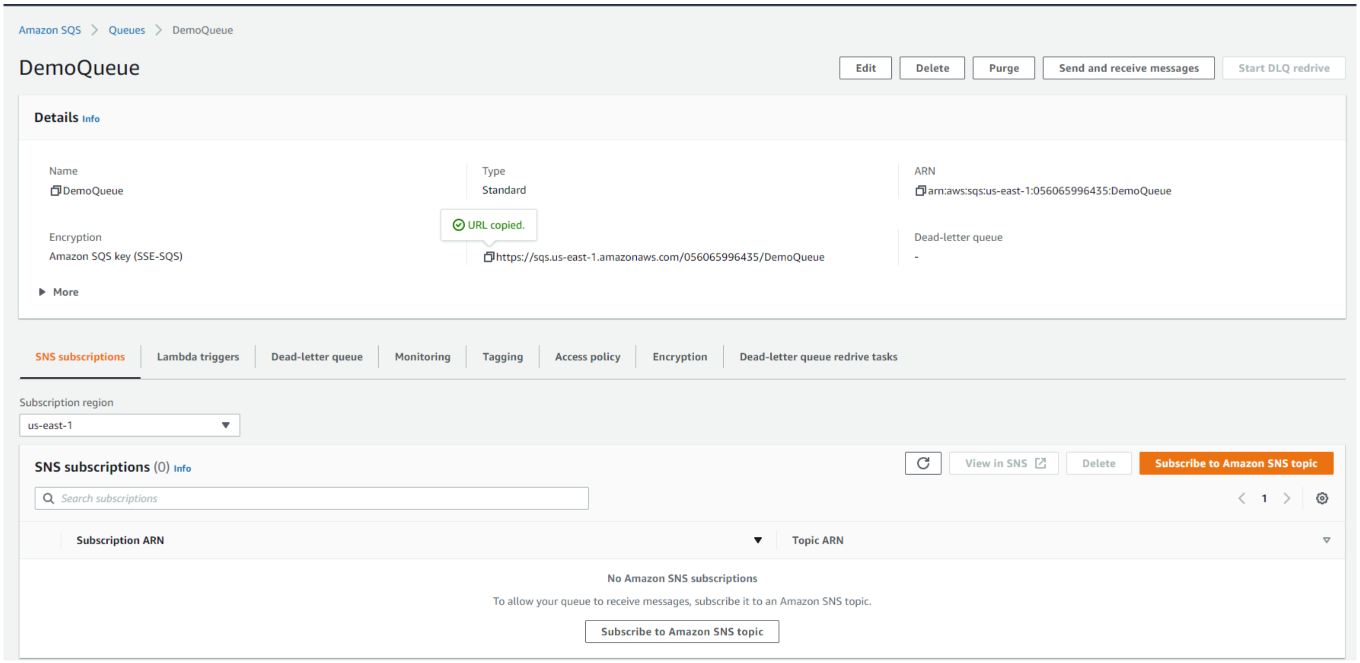
Type
Standard

Display name
-

Topic owner
056065996435

Subscriptions (1)

	ID ▲	Endpoint ▽	Status ▽	Protocol ▽
○	a743dc5b-a876-44a0-949a-bf53f7aeb0c4	gdmichaeldemewoz@gmail.com	✓ Confirmed	EMAIL



DemoQueue

Details

Name
DemoQueue

Type
Standard

ARN
arn:aws:sqs:us-east-1:056065996435:DemoQueue

Encryption
Amazon SQS key (SSE-SQS)

https://sqs.us-east-1.amazonaws.com/056065996435/DemoQueue

Dead-letter queue
-

Subscription region
us-east-1

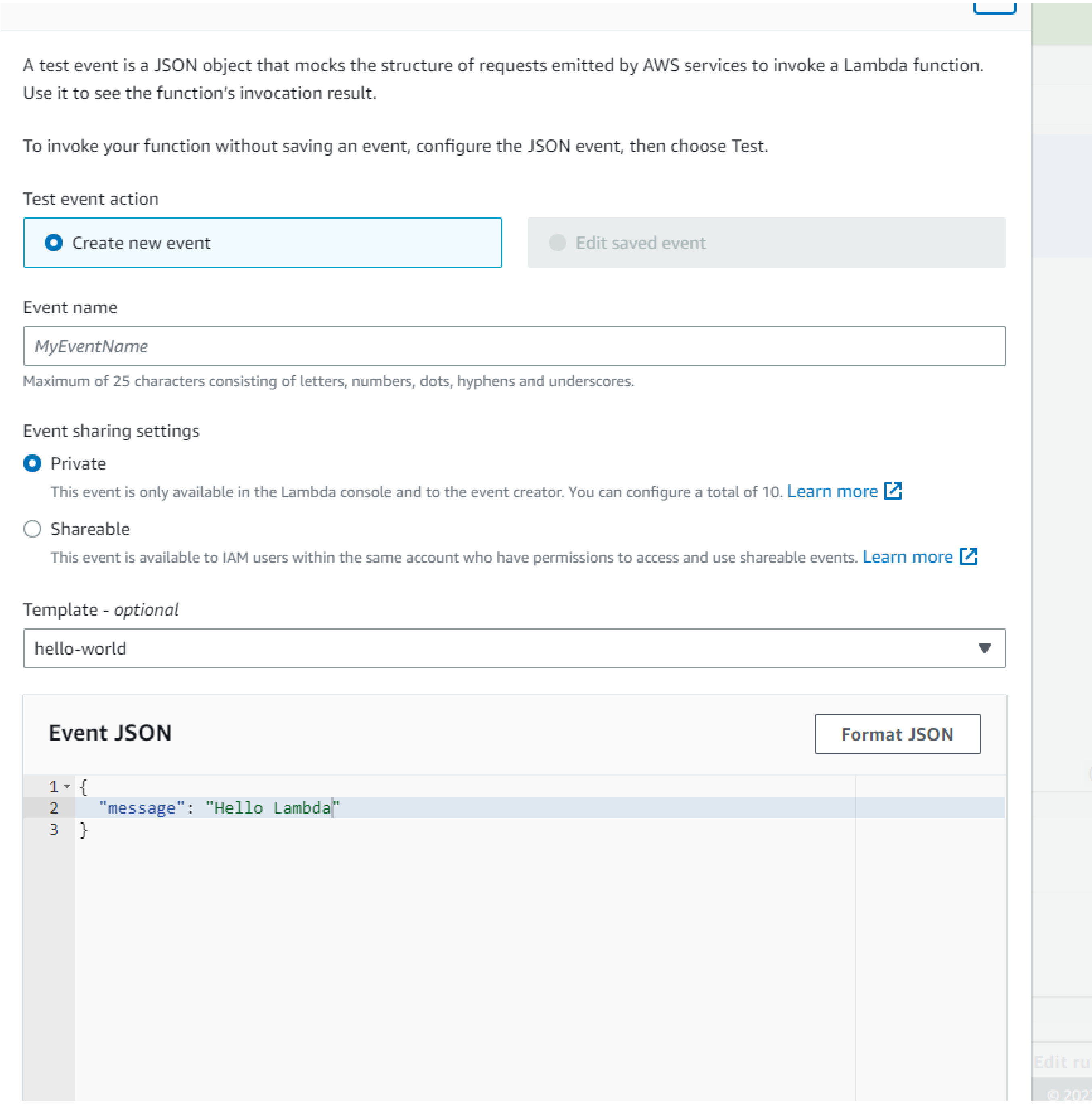
SNS subscriptions (0)

Subscription ARN	Topic ARN

No Amazon SNS subscriptions

To allow your queue to receive messages, subscribe it to an Amazon SNS topic.

Subscribe to Amazon SNS topic



A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

● Create new event ○ Edit saved event

Event name

MyEventName

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

● Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

○ Shareable

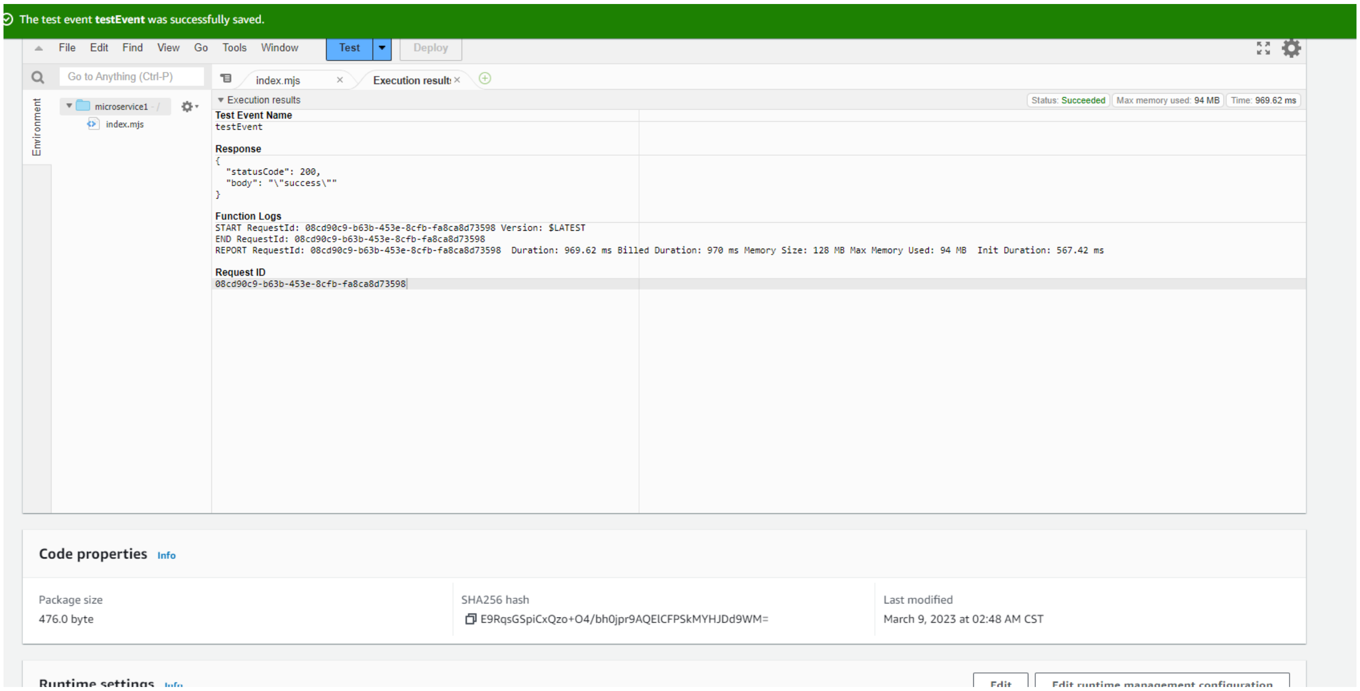
This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

Template - *optional*

hello-world ▼

Event JSON

```
{
  "message": "Hello Lambda"
}
```



The test event **testEvent** was successfully saved.

Status: Succeeded | Max memory used: 94 MB | Time: 969.62 ms

Test Event Name

```
testEvent
```

Response

```
{
  "statusCode": 200,
  "body": "\"success\""
}
```

Function Logs

```
START RequestId: 08cd90c9-b63b-453e-8cfb-fa8ca8d73598 Version: $LATEST
END RequestId: 08cd90c9-b63b-453e-8cfb-fa8ca8d73598
REPORT RequestId: 08cd90c9-b63b-453e-8cfb-fa8ca8d73598  Duration: 969.62 ms Billed Duration: 970 ms Memory Size: 128 MB Max Memory Used: 94 MB  Init Duration: 567.42 ms
```

Request ID

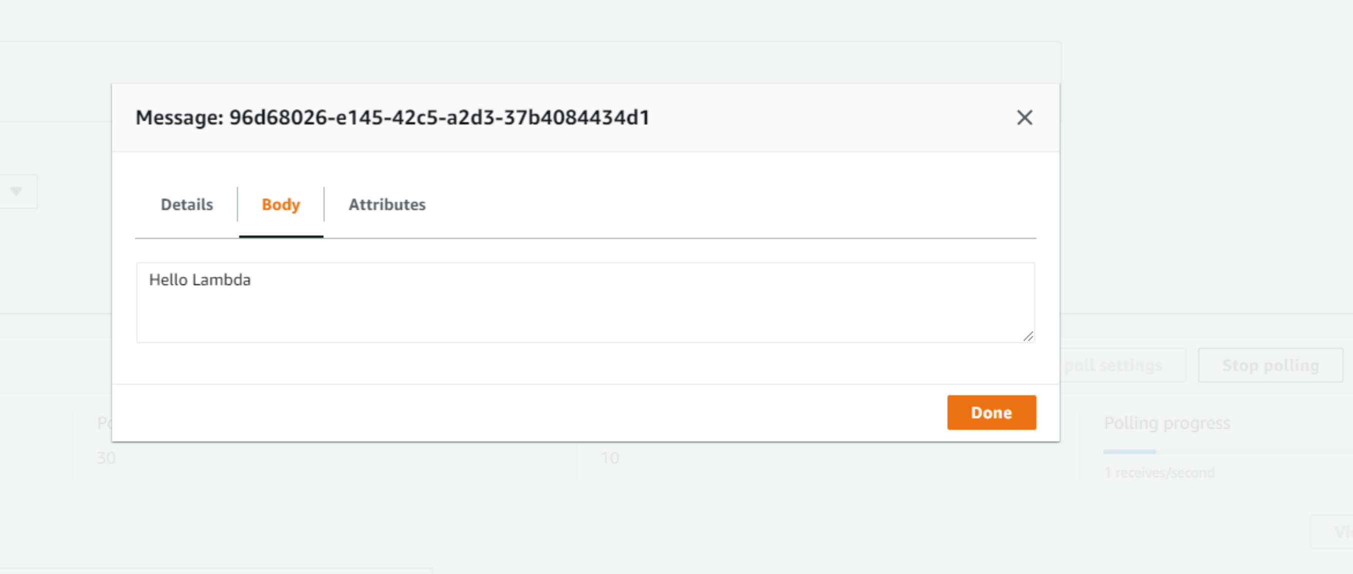
```
08cd90c9-b63b-453e-8cfb-fa8ca8d73598
```

Code properties Info

Package size: 476.0 byte

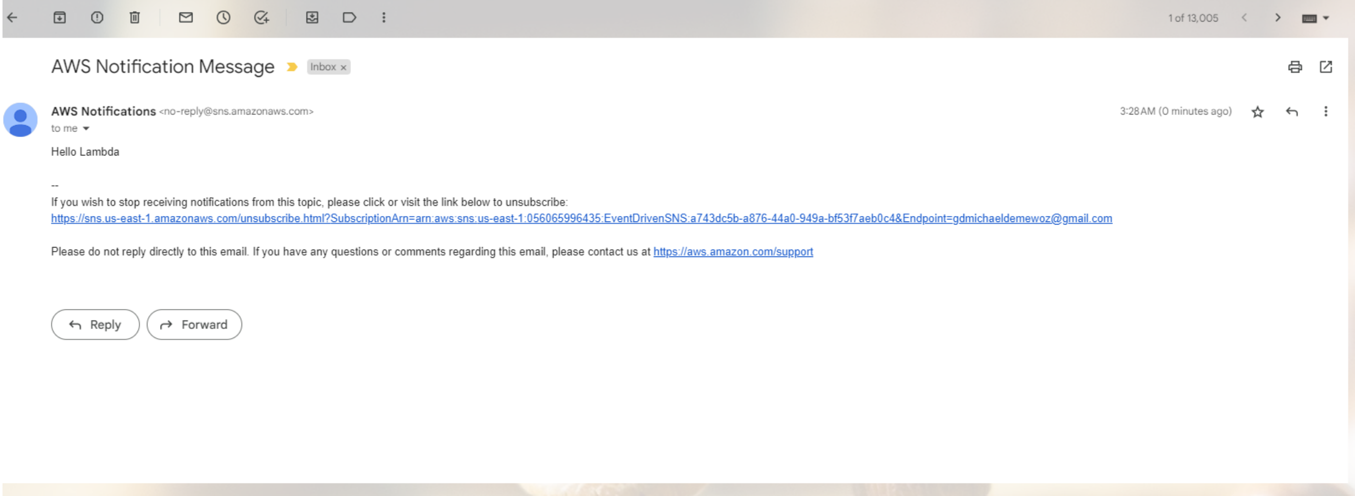
SHA256 hash: E9RqsGSpiCxQzo+O4/bh0jpr9AQElCFPSkMYHJDd9WM=

Last modified: March 9, 2023 at 02:48 AM CST



Message: 96d68026-e145-42c5-a2d3-37b4084434d1

Hello Lambda



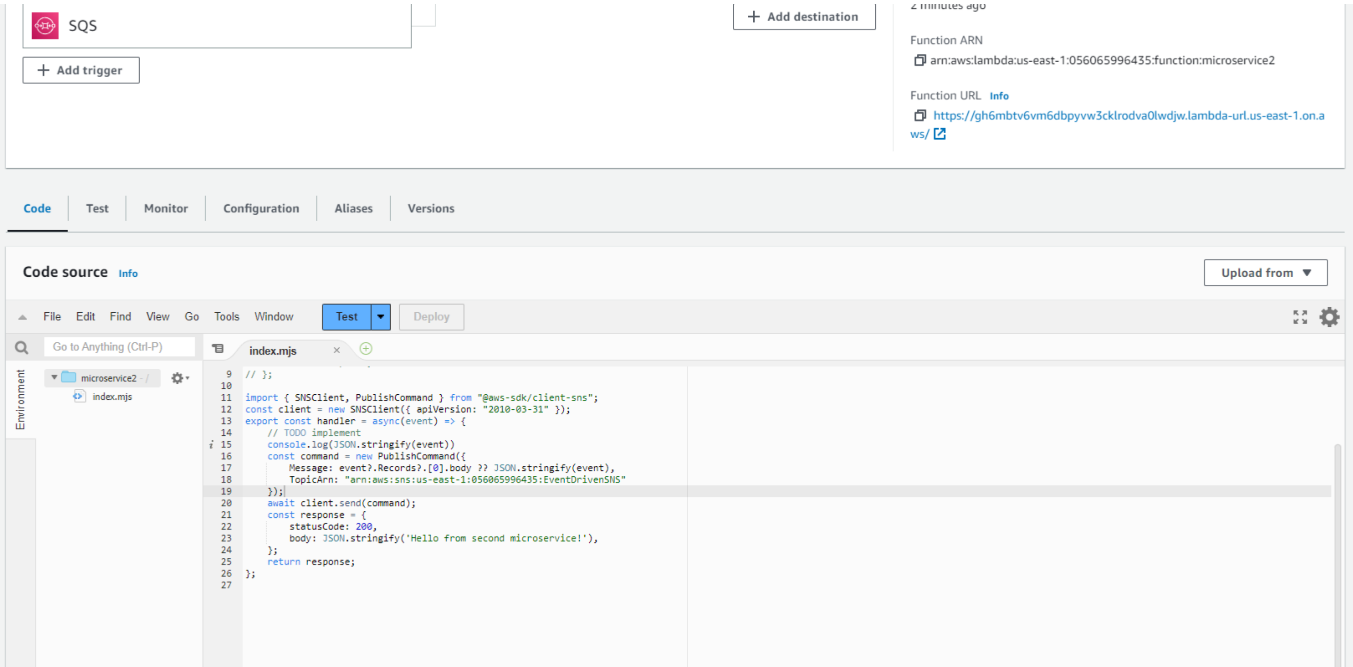
AWS Notification Message

AWS Notifications <no-reply@sns.amazonaws.com>
to me
3:28 AM (0 minutes ago)

Hello Lambda

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:056065996435:EventDrivenSNS:a743dc5b-a876-44a0-949a-bf53f7aeb0c4&Endpoint=gdmichaeldemewoz@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support



2 minutes ago

Function ARN
arn:aws:lambda:us-east-1:056065996435:function:microservice2

Function URL Info
https://gh6mbtv6vm6dbpyvw3cklrodva0lwdjw.lambda-url.us-east-1.on.aws/

Code source Info

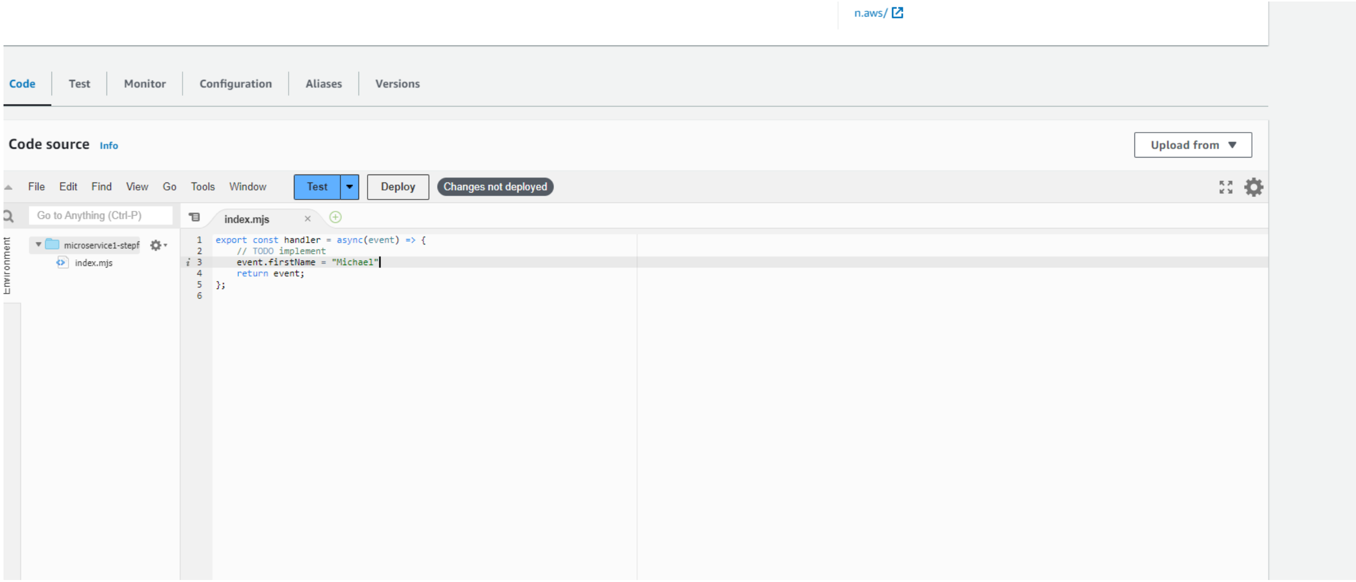
index.mjs

```
// };

import { SNSClient, PublishCommand } from "@aws-sdk/client-sns";
const client = new SNSClient({ apiVersion: "2010-03-31" });
export const handler = async(event) => {
    // TODO implement
    console.log(JSON.stringify(event))
    const command = new PublishCommand({
        Message: event?.Records?.[0].body ?? JSON.stringify(event),
        TopicArn: "arn:aws:sns:us-east-1:056065996435:EventDrivenSNS"
    });
    await client.send(command);
    const response = {
        statusCode: 200,
        body: JSON.stringify('Hello from second microservice!'),
    };
    return response;
};

```

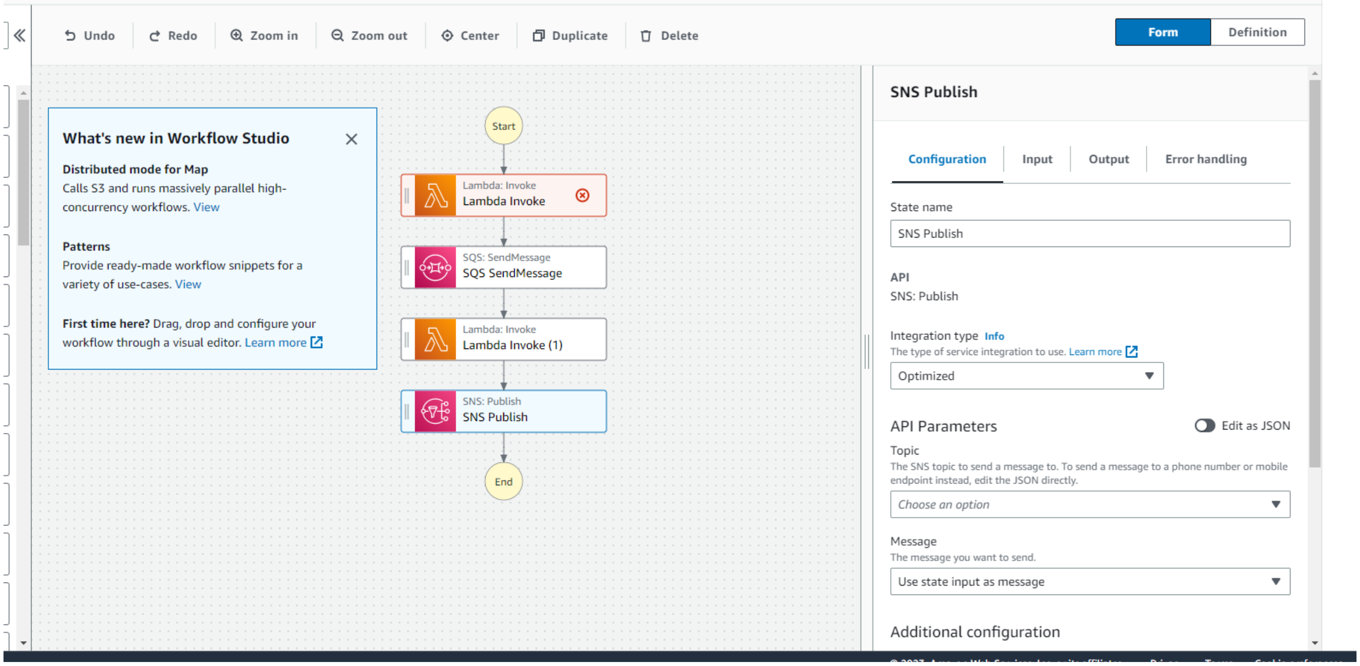
Q2



Code source Info

index.mjs

```
export const handler = async(event) => {
    // TODO implement
    event.firstName = "Michael"
    return event;
};
```



What's new in Workflow Studio

Distributed mode for Map
Calls S3 and runs massively parallel high-concurrency workflows. View

Patterns
Provide ready-made workflow snippets for a variety of use-cases. View

First time here? Drag, drop and configure your workflow through a visual editor. Learn more

SNS Publish

State name: SNS Publish

API: SNS: Publish

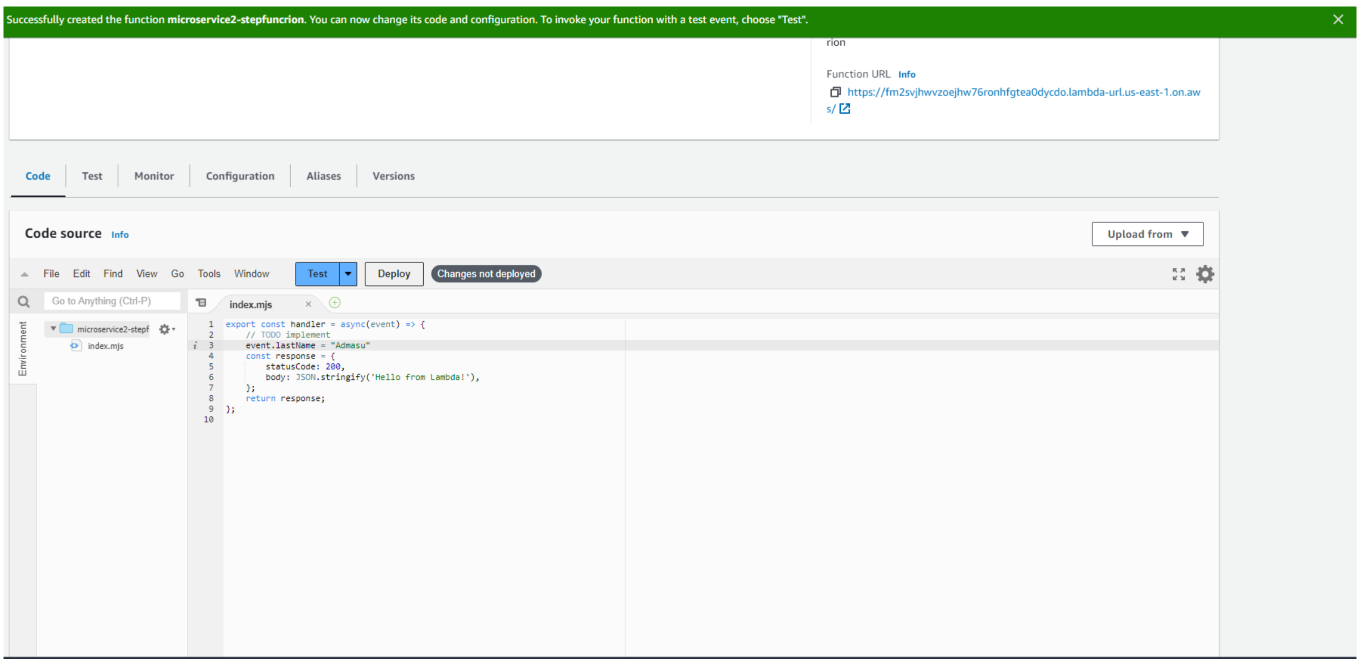
Integration type Info
The type of service integration to use. Learn more
Optimized

API Parameters — Edit as JSON

Topic
The SNS topic to send a message to. To send a message to a phone number or mobile endpoint instead, edit the JSON directly.
Choose an option

Message
The message you want to send.
Use state input as message

Additional configuration



Successfully created the function **microservice2-stepfuncrion**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

rion

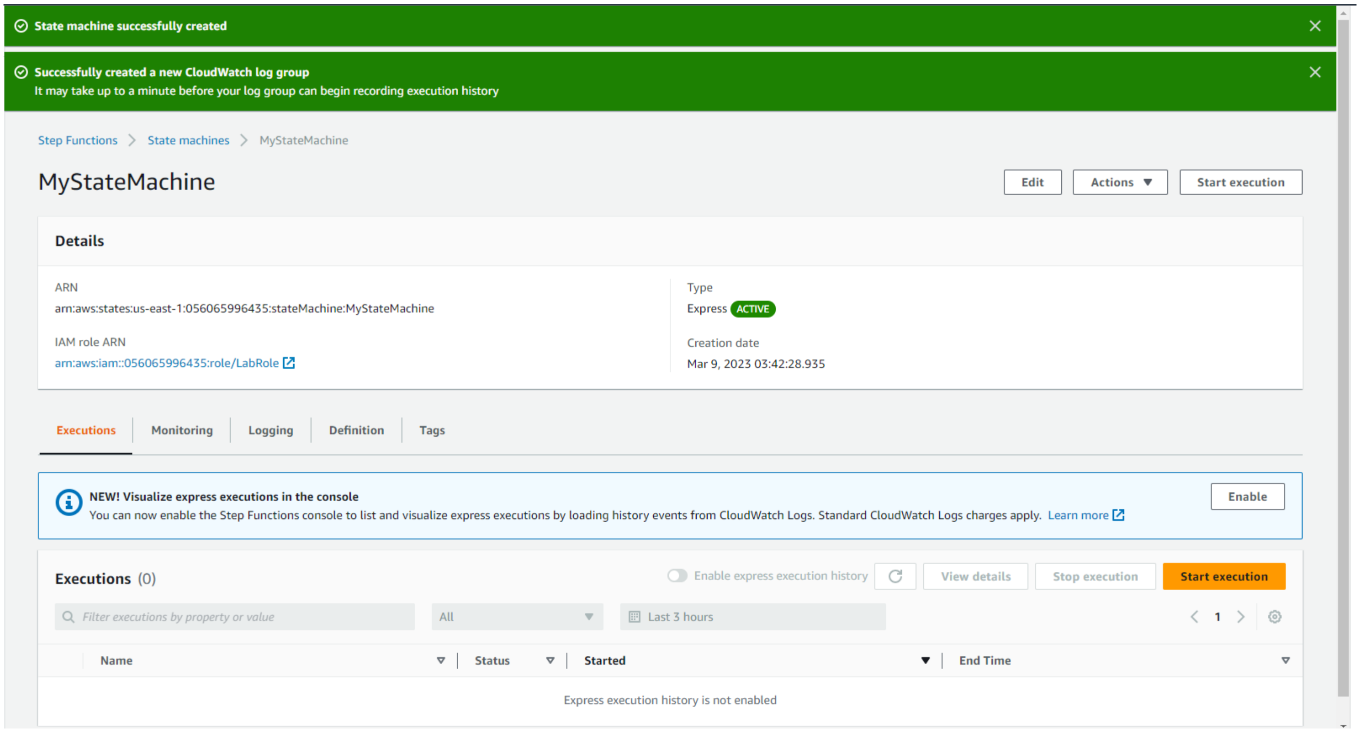
Function URL Info
https://fm2svjhwvzoejhw76ronhfgtea0dycdo.lambda-url.us-east-1.on.aws/

Code source Info

index.mjs

```
export const handler = async(event) => {
    // TODO implement
    event.lastName = "Admasu"
    const response = {
        statusCode: 200,
        body: JSON.stringify('Hello from Lambda!'),
    };
    return response;
};
```

```
{
"Comment": "A description of my state machine",
"StartAt": "Lambda Invoke",
"States": {
"Lambda Invoke": {
"Type": "Task",
"Resource": "arn:aws:states:::lambda:invoke",
"OutputPath": ".Payload", "Parameters": { "Payload.": "
$'',''FunctionName'' :''arn : aws : lambda : us - east - 1 : 056065996435 : function : microservice1 -$
LATEST"
},
"Retry": [
{
"ErrorEquals": [
"Lambda.ServiceException",
"Lambda.AWSLambdaException",
"Lambda.SdkClientException",
"Lambda.TooManyRequestsException"
],
"IntervalSeconds": 2,
"MaxAttempts": 6,
"BackoffRate": 2
}
],
"Next": "SQS SendMessage"
},
"SQS SendMessage": {
"Type": "Task",
"Resource": "arn:aws:states:::sqs:sendMessage",
"Parameters": {
"MessageBody.$'' :''$",
"QueueUrl": "https://sqs.us-east-
1.amazonaws.com/056065996435/QueueDemoStepFunction"
},
"Next": "Lambda Invoke (1)"
},
"Lambda Invoke (1)": {
"Type": "Task",
"Resource": "arn:aws:states:::lambda:invoke",
"OutputPath": ".Payload", "Parameters": { "Payload.": "
″,″FunctionName″ :″arn : aws : lambda : us − east − 1 : 056065996435 : function : microservice2 −
LATEST"
},
"Retry": [
{
"ErrorEquals": [
"Lambda.ServiceException",
"Lambda.AWSLambdaException",
"Lambda.SdkClientException",
"Lambda.TooManyRequestsException"
],
"IntervalSeconds": 2,
"MaxAttempts": 6,
"BackoffRate": 2
}
],
"Next": "SNS Publish"
},
"SNS Publish": {
"Type": "Task",
"Resource": "arn:aws:states:::sns:publish",
"Parameters": {
"Message.″ :″",
"TopicArn": "arn:aws:sns:us-east-1:056065996435:EventDrivenSNS"
},
"End": true
}
}
}
```



State machine successfully created

Successfully created a new CloudWatch log group
It may take up to a minute before your log group can begin recording execution history

MyStateMachine

Details

ARN
arn:aws:states:us-east-1:056065996435:stateMachine:MyStateMachine

IAM role ARN
arn:aws:iam::056065996435:role/LabRole

Type
Express ACTIVE

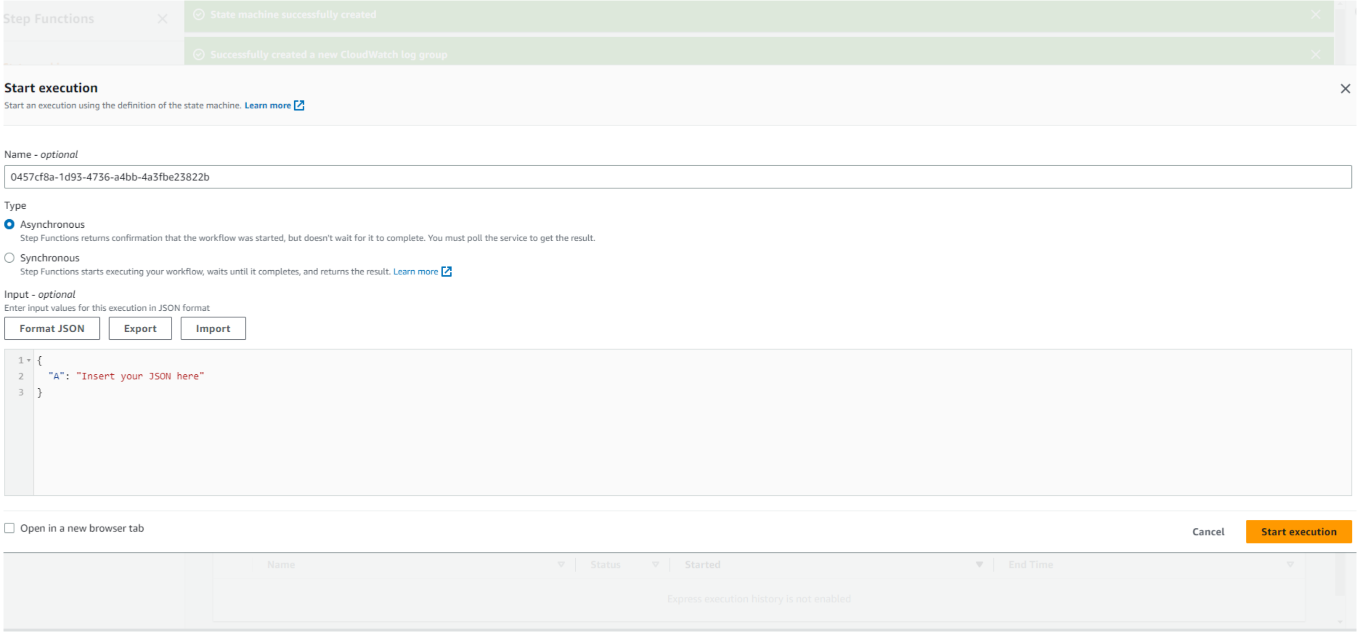
Creation date
Mar 9, 2023 03:42:28.935

NEW! Visualize express executions in the console
You can now enable the Step Functions console to list and visualize express executions by loading history events from CloudWatch Logs. Standard CloudWatch Logs charges apply. Learn more

Executions (0)

Name	Status	Started	End Time

Express execution history is not enabled



Start execution
Start an execution using the definition of the state machine. Learn more

Name - *optional*
0457cf8a-1d93-4736-a4bb-4a3fbe23822b

Type
● Asynchronous
Step Functions returns confirmation that the workflow was started, but doesn't wait for it to complete. You must poll the service to get the result.
○ Synchronous
Step Functions starts executing your workflow, waits until it completes, and returns the result. Learn more

Input - *optional*
Enter input values for this execution in JSON format

```
{
  "A": "Insert your JSON here"
}
```

□ Open in a new browser tab