
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

API309

Advanced serverless workflow patterns and best practices

Benjamin Smith

Principal Developer Advocate, Serverless
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Nice to meet you!

- Ben Smith
  - Email: benjasl@amazon.com
  - Twitter: @benjamin_l_s
- Most interested in
  - Event-driven architectures with Step Functions
  - Serverless and PHP
- Random fact

  I once taught David Hasselhoff to ski

Everything in one place



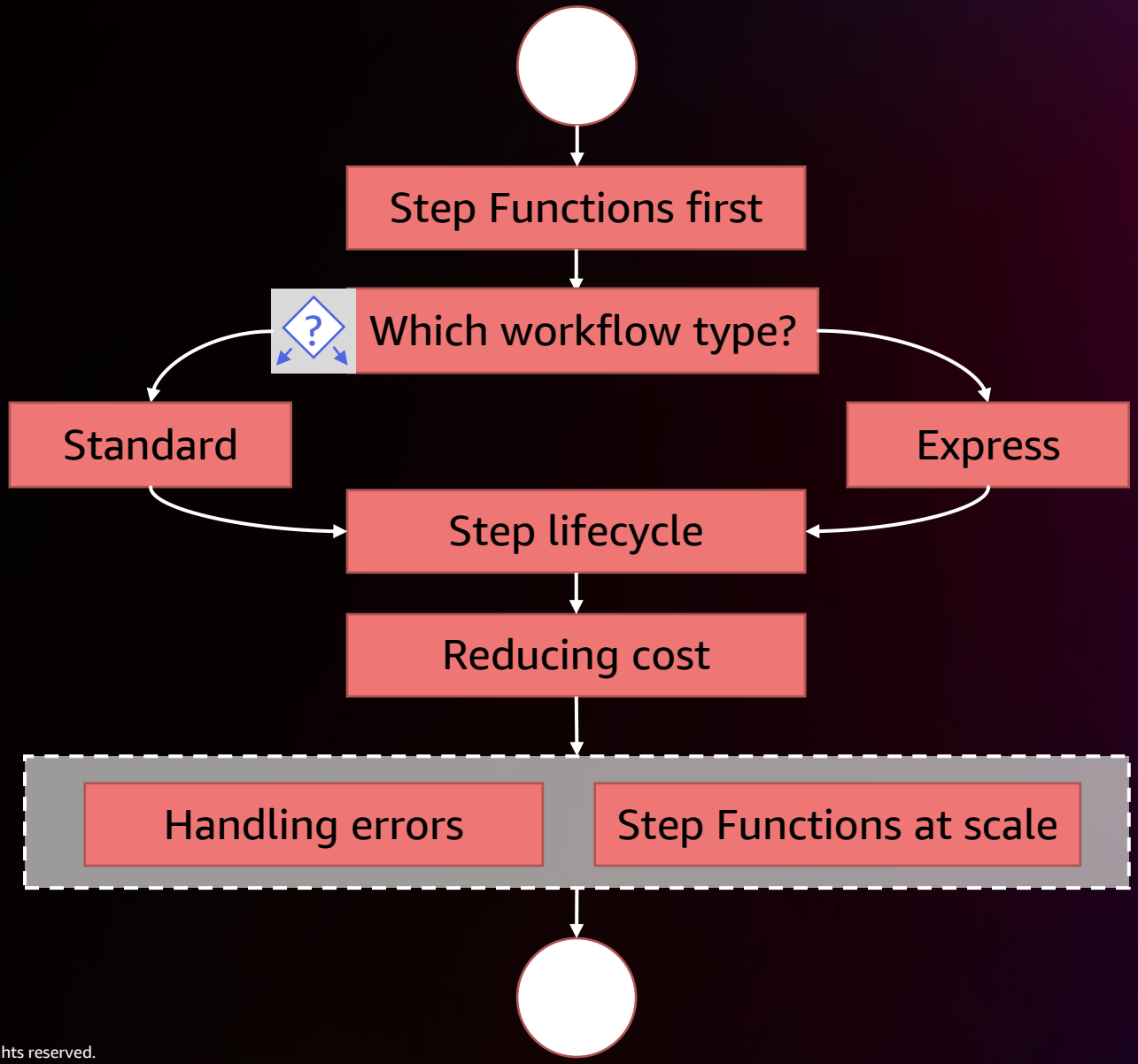
https://s12d.com/api309

Resources for API309
Advanced serverless workflow patterns and best practices

- Downloadable templates
- Blog posts
- Videos
- Workshops
- Code samples

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda (as a Step Functions Workflow)



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The end goal



Customer value



Reliability



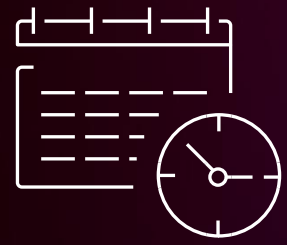
Resilience



Scalability



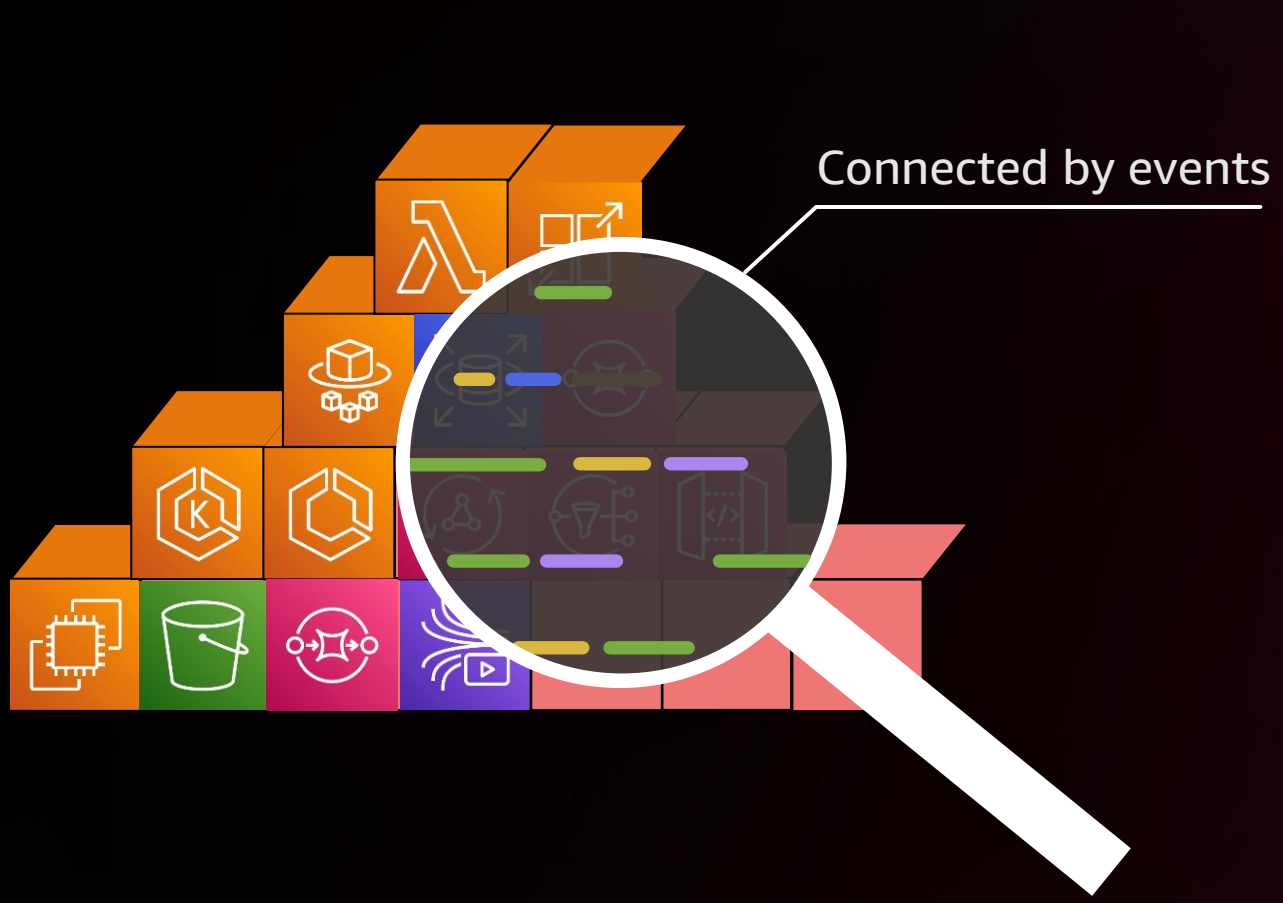
Simplicity



And do it faster

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Managed



Manual

The

Serverless

Event-Driven Pyramid in 2023

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The serverless challenge

When your application is powered by multiple connected services, how do you build, track, inspect, visualize, and orchestrate those connections?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Step Functions is different

AWS Lambda

Amazon API Gateway

Amazon DynamoDB

Amazon EC2

Amazon EventBridge

Amazon S3

Amazon ECS

Amazon SNS

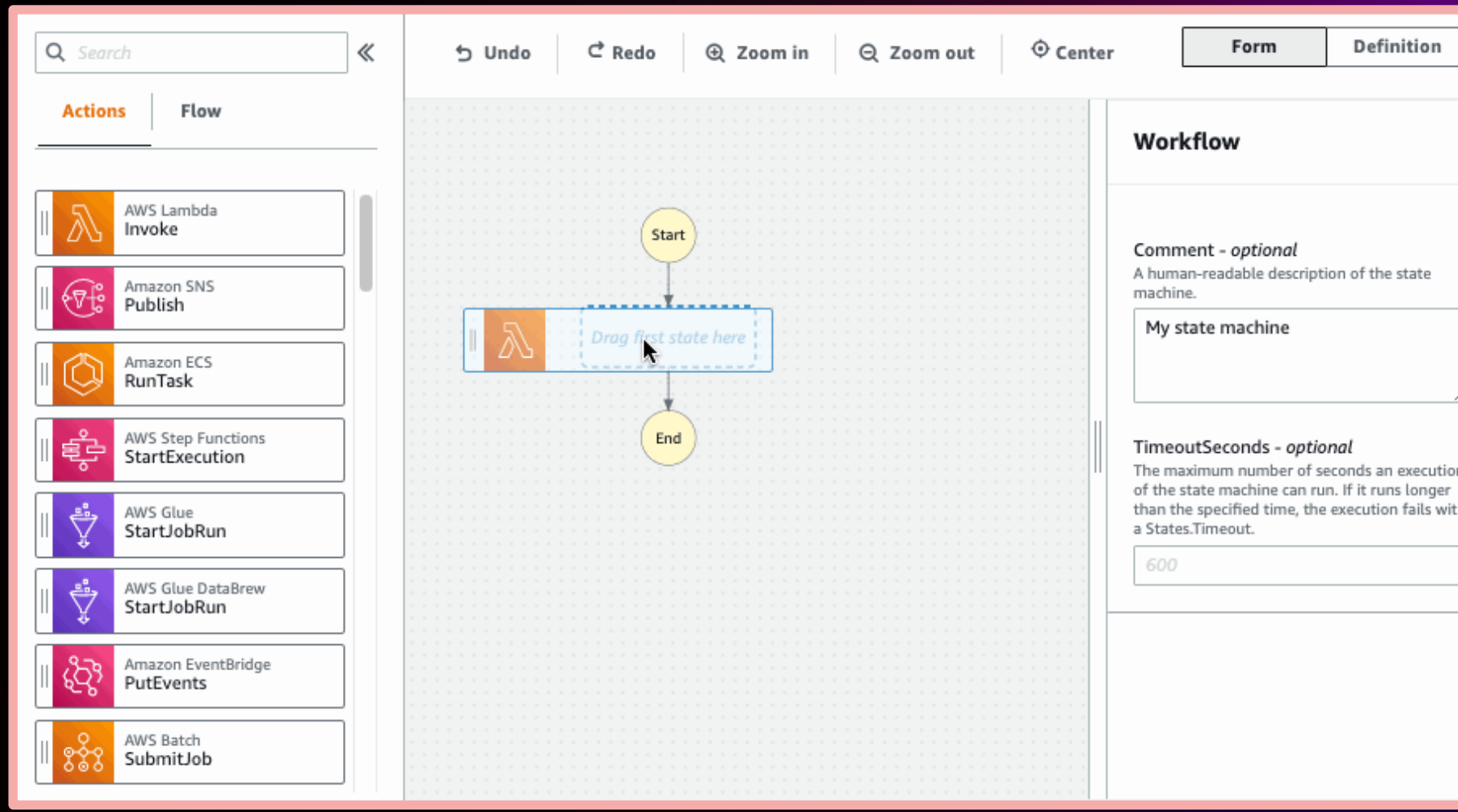
Amazon Kinesis

Amazon EKS

Amazon SQS

AWS IoT Core

AWS Fargate



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

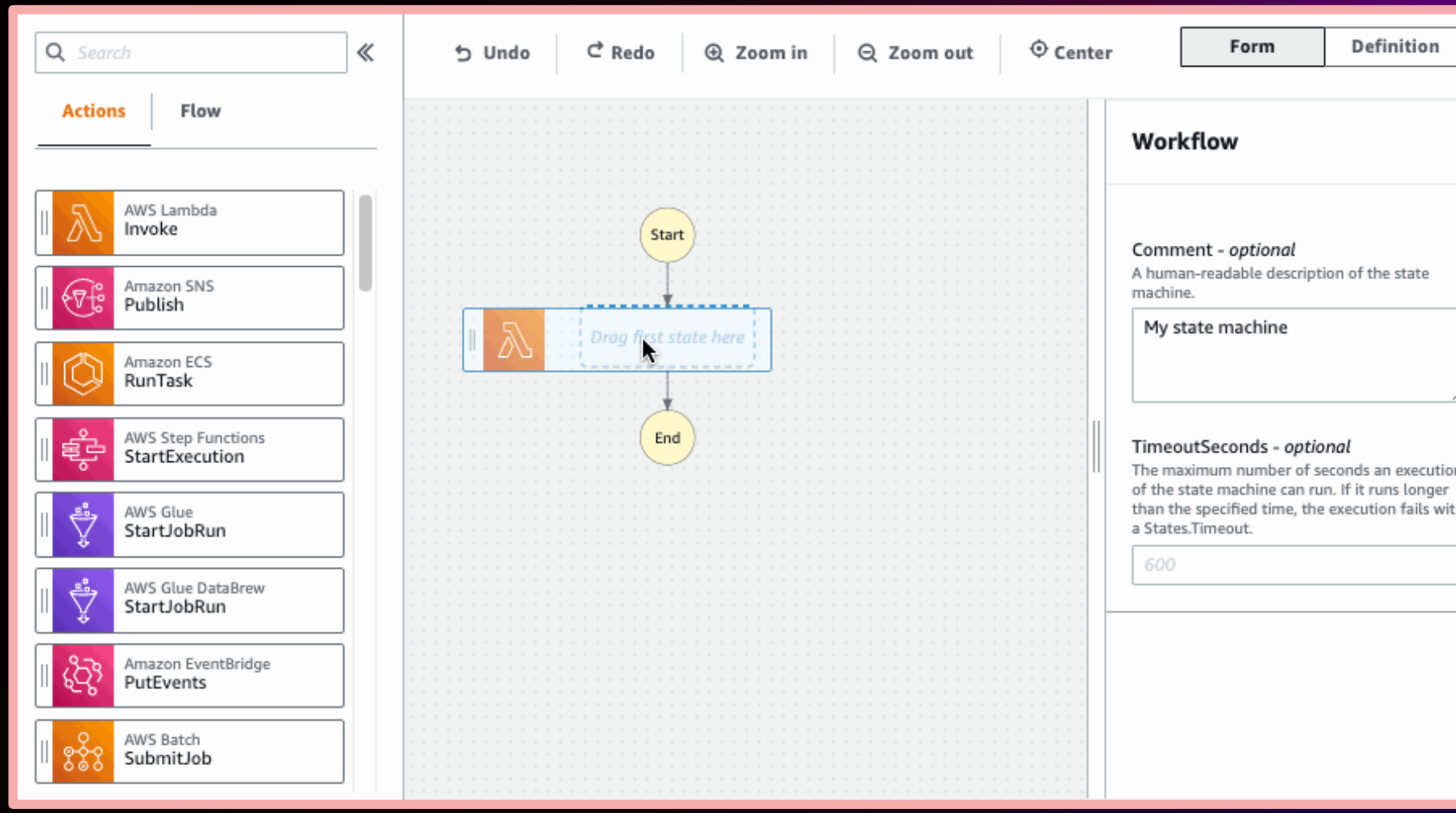
AWS Step Functions is different

SERVERLESS LOW-CODE VISUAL WORKFLOW SERVICE

- Pay-per-use
- Scales automatically
- Fully managed

- Drag and drop or ASL
- Built-in error handling

- Integrates with over 200 AWS services

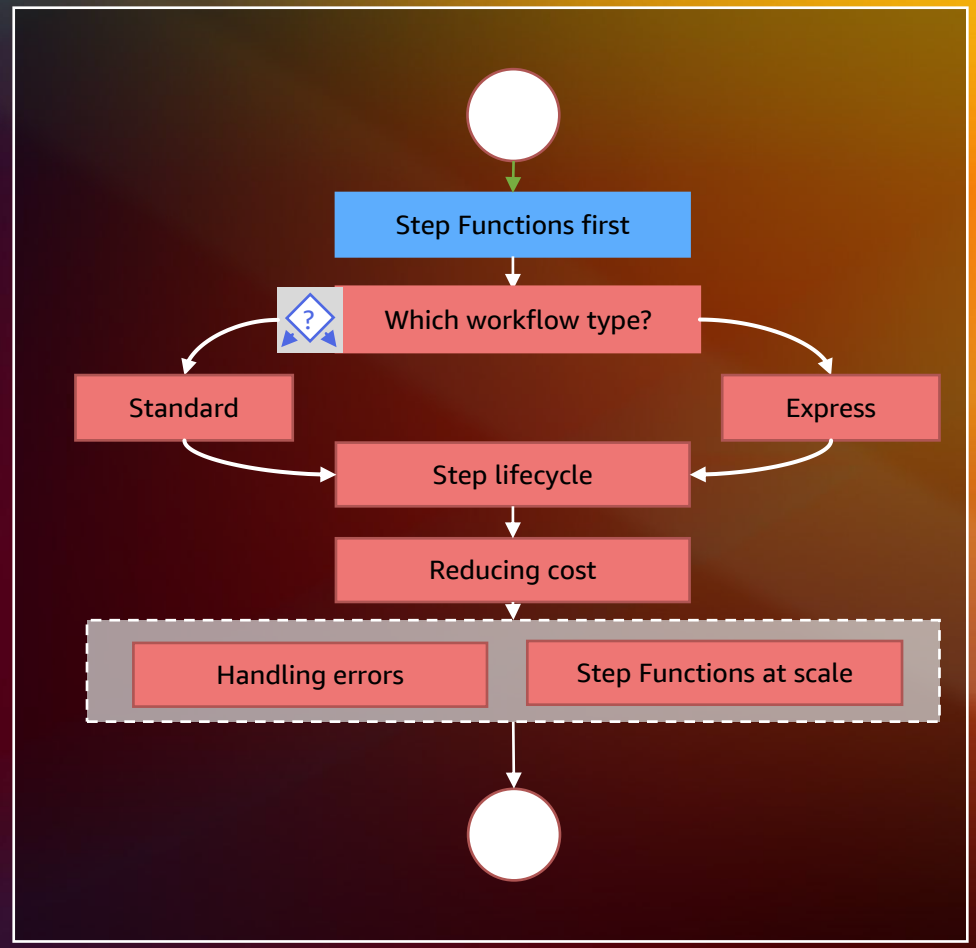


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“Step Functions first, Step Functions always.”

Ben Smith

Just now



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

From code to workflow

app.js

```
const AWS = require('aws-sdk');
const docClient = new AWS.DynamoDB.DocumentClient();


var params = {
  "TableName": "reinvent2022!",
  "Key": {
    "PK": {"S": "Wardrobe"},
    "SK": {"S": ”shoes"}
  }
}


async function queryItems(){
  try {
    const data = await docClient.getItem(params).promise()
    return data
  } catch (err) {
    return err
  }
}
exports.handler = async (event, context) => {
  try {
    const data = await queryItems()
    return { body: JSON.stringify(data) }
  } catch (err) {
    return { error: err }
  }
}
```

A Lambda function that queries Amazon DynamoDB has multiple lines of code



AWS Lambda

Amazon DynamoDB

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

From code to workflow

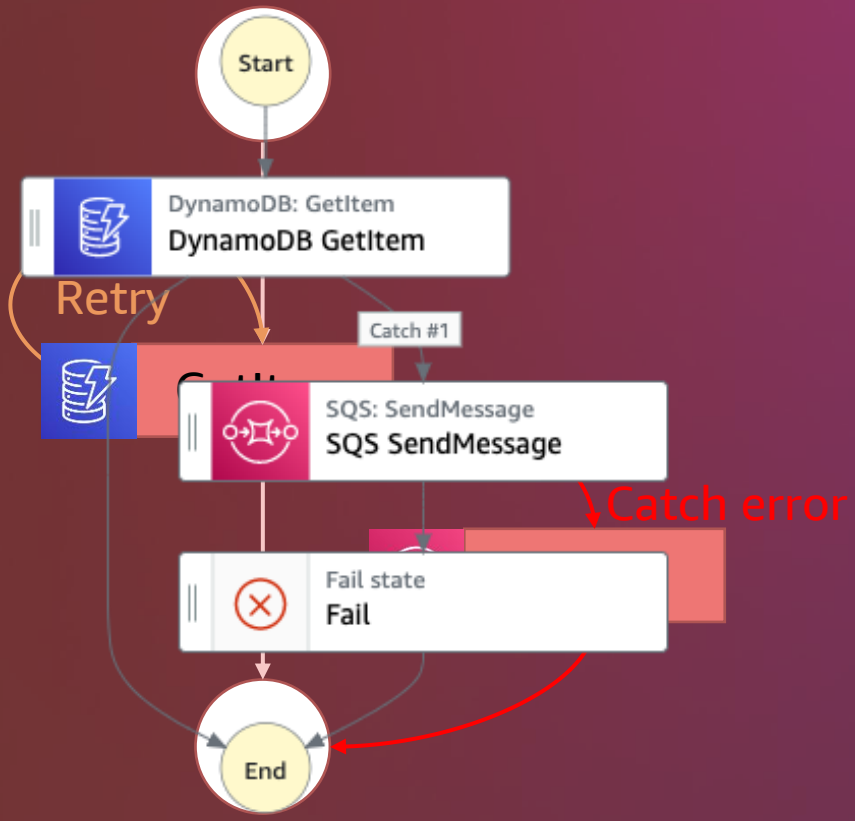
app.js

```
const AWS = require('aws-sdk');
const docClient = new AWS.DynamoDB.DocumentClient();


var params = {
  "TableName": "reinvent2022!",
  "Key": {
    "PK": {"S": "Wardrobe"},
    "SK": {"S": "shoes"}
  }
}

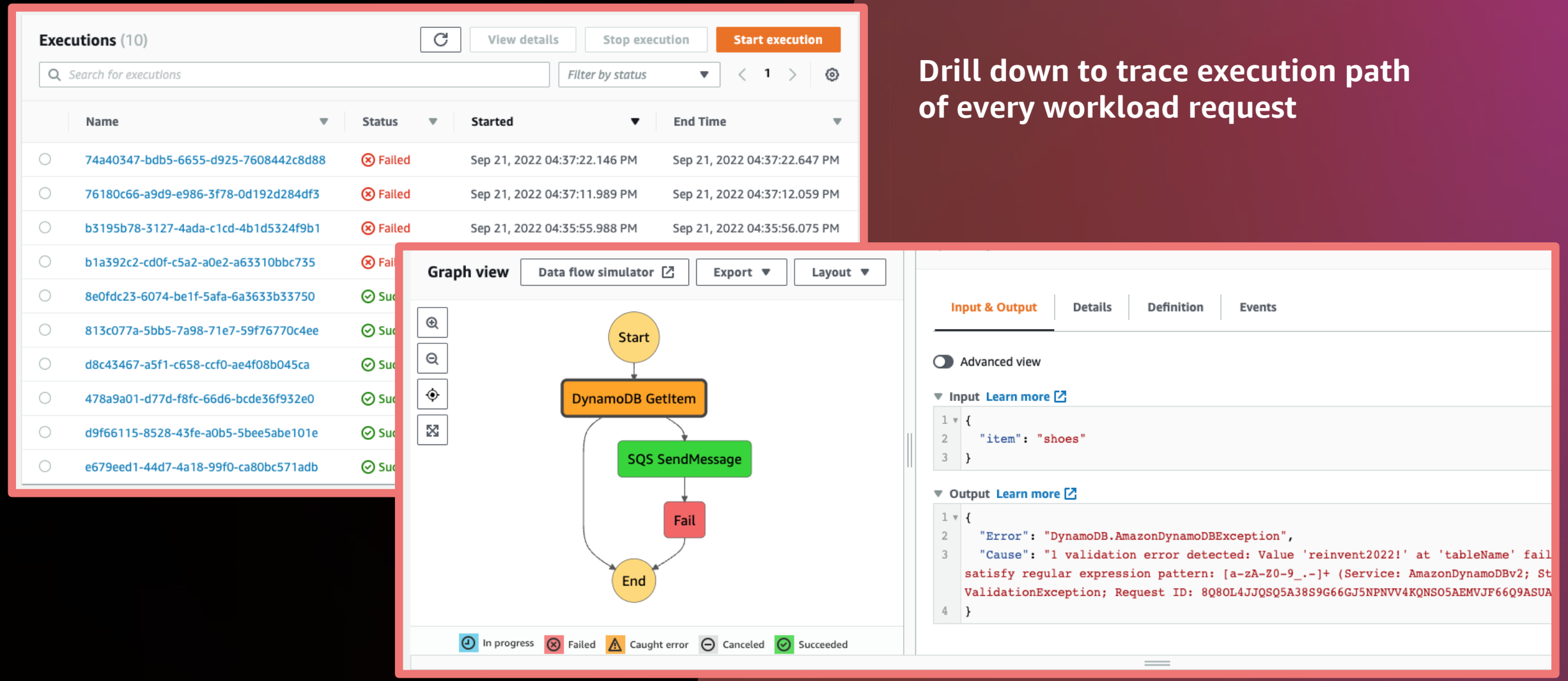

async function queryItems(){
  try {
    const data = await docClient.getItem(params).promise()
    return data
  } catch (err) {
    return err
  }
}
exports.handler = async (event, context) => {
  try {
    const data = await queryItems()
    return { body: JSON.stringify(data) }
  } catch (err) {
    return { error: err }
  }
}
```

Even a single task "workflow" adds value with built-in error handling, catch, retry, observability, reduction of custom code, and centralized logging of each workload



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

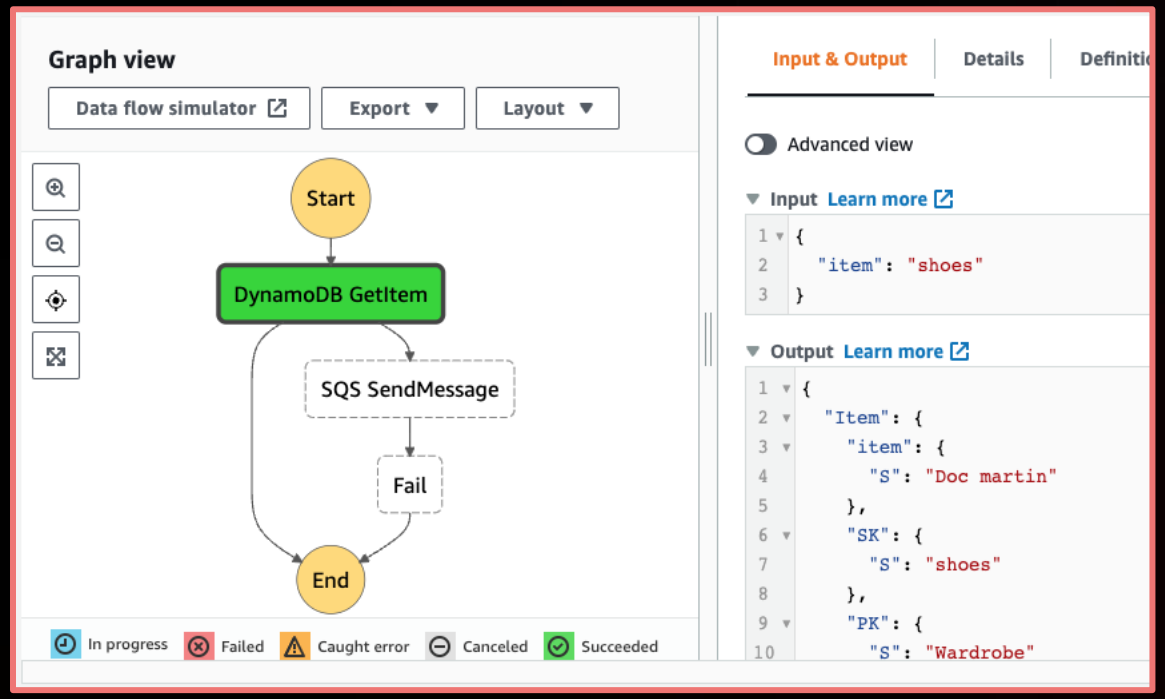
From code to workflow



Drill down to trace execution path of every workload request

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

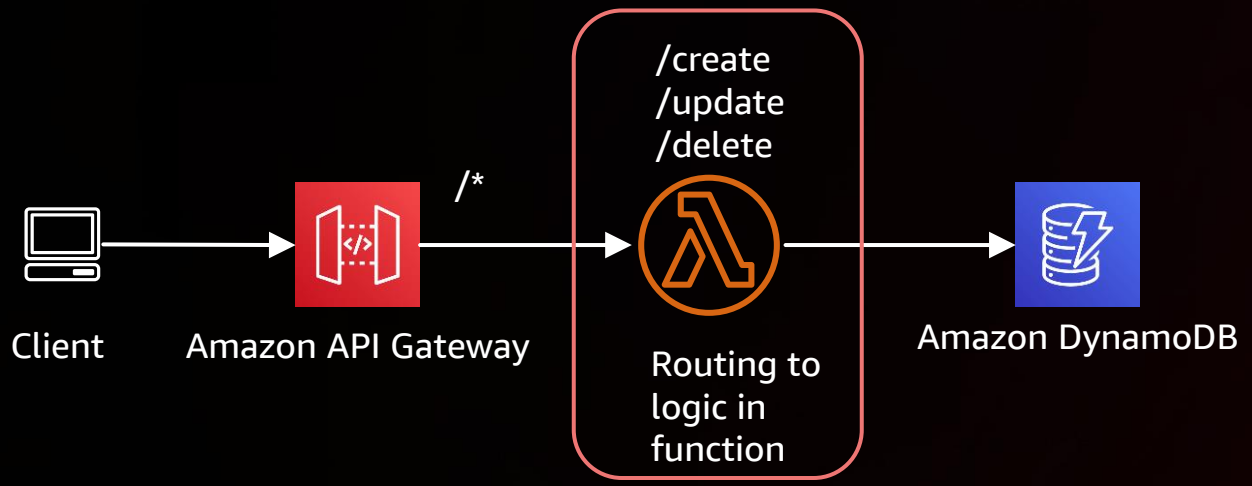
From code to workflow



Examine the input and output of each task for each request

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Breaking apart a “Lambda-lith”

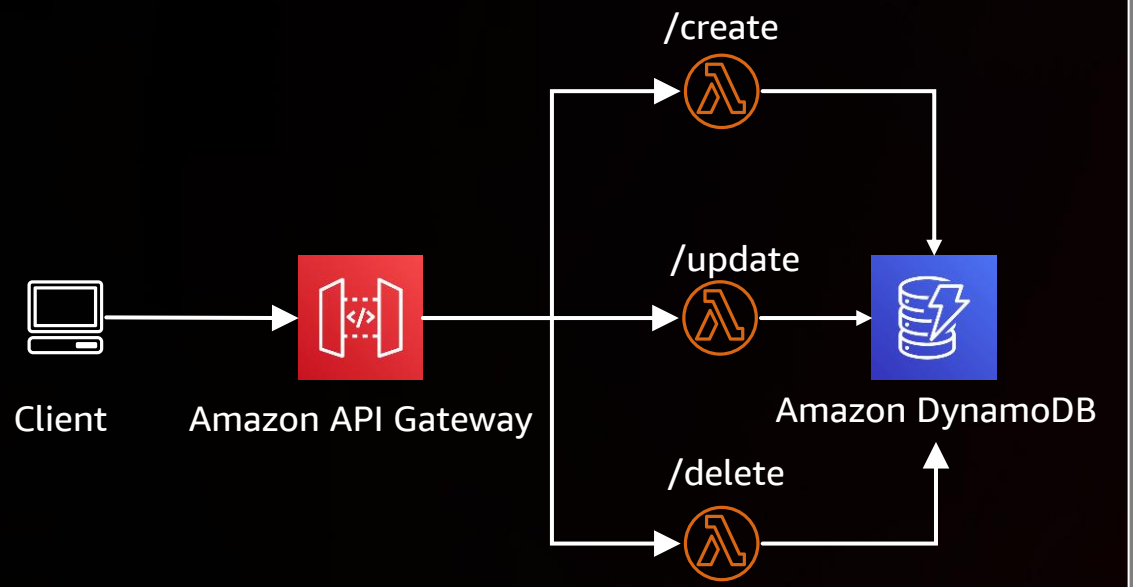


Amazon API Gateway routes all requests to a single Lambda function that runs the appropriate code based on its route configuration

- Security permissions applied to the whole
- Performance setting applied to the whole
- Duration and space limits apply to whole

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Micro Lambda



App-delete.js

App-update.js

app-create.js

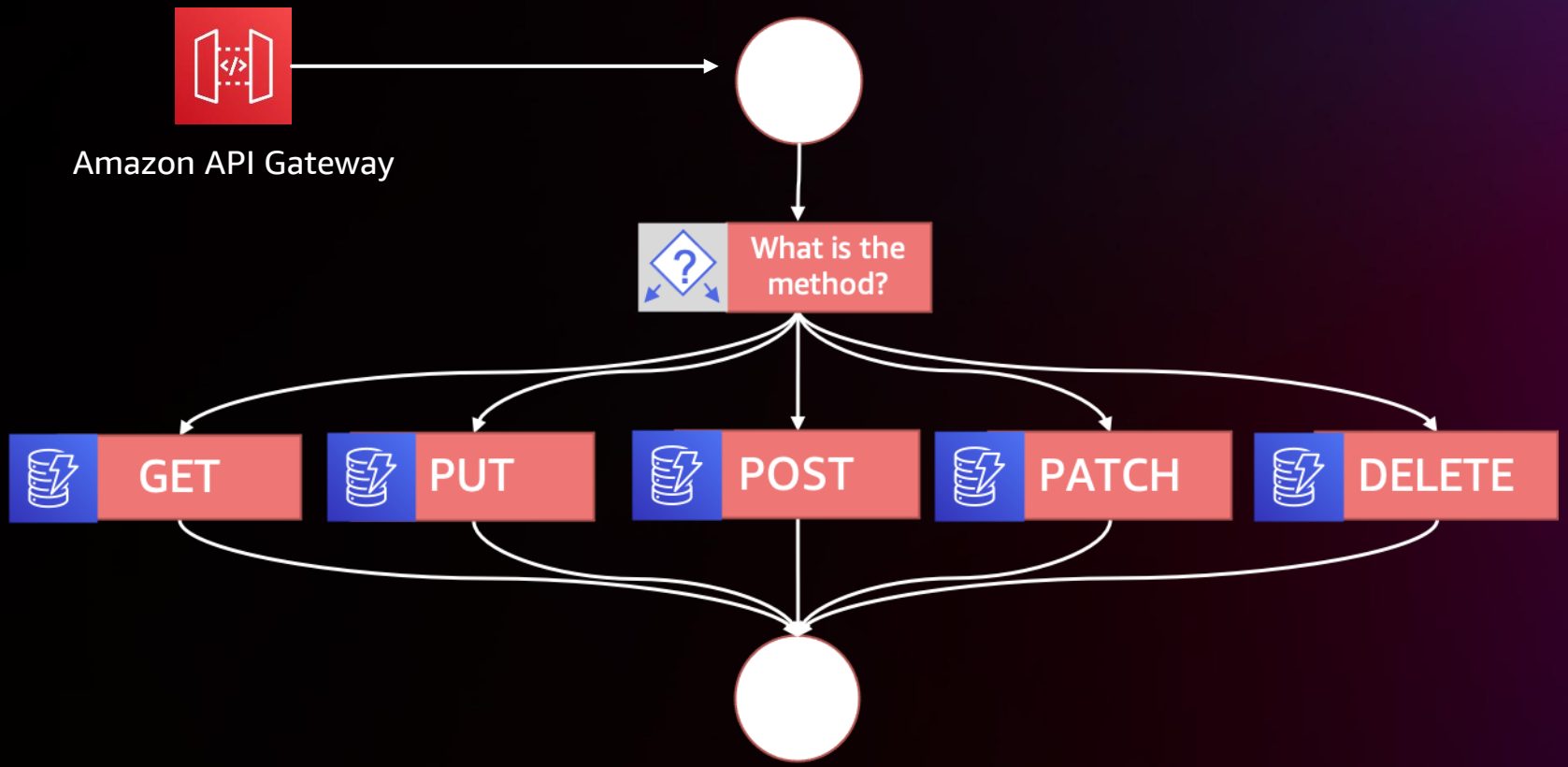
```
const AWS = require('aws-sdk');
const docClient = new AWS.DynamoDB.DocumentClient();

var params = {
  TableName: 'your-table-name',
  IndexName: 'some-index',
  KeyConditionExpression: '#name = :value',
  ExpressionAttributeValues: { ':value': 'shoes' },
  ExpressionAttributeNames: { '#name': 'name' }
}

async function queryItems(){
  try {
    const data = await docClient.query(params).promise()
    return data
  } catch (err) {
    return err
  }
}

exports.handler = async (event, context) => {
  try {
    const data = await queryItems()
    return { body: JSON.stringify(data) }
  } catch (err) {
    return { error: err }
  }
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



The “REST” easy

Combine with API Gateway and synchronous Express Workflows to create low-latency, scalable API backend

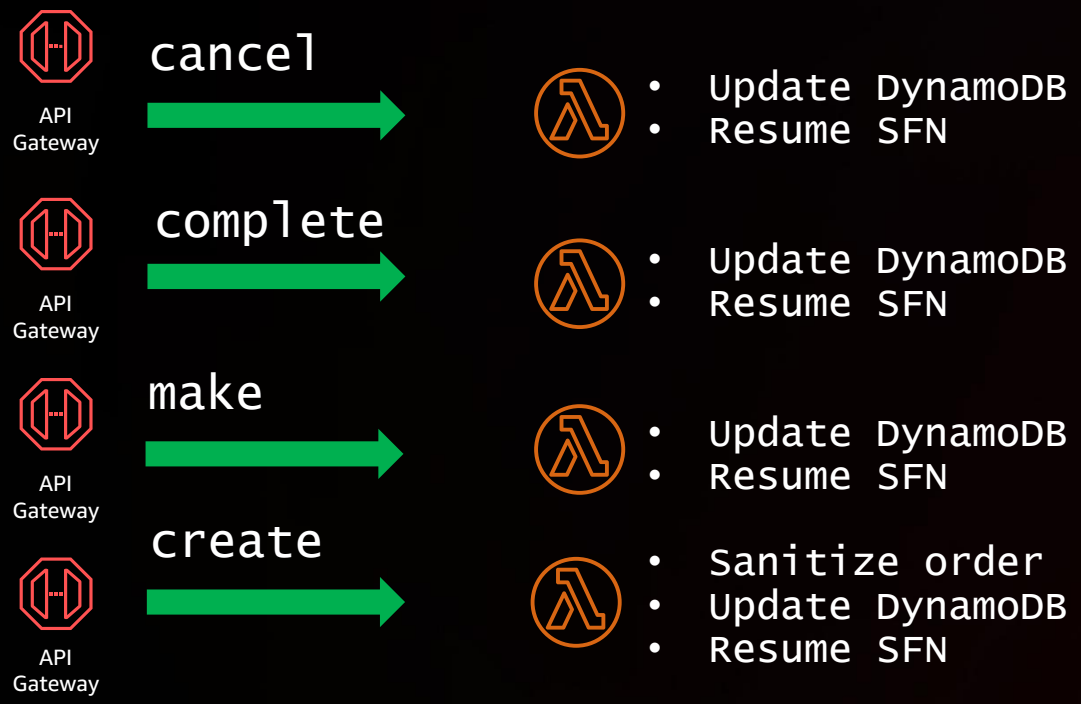
Simplicity

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Order manager service as a Lambda function

Version 1

Each operation invokes a Lambda function



Application grew more complex over time, performing multiple tasks to handle increasingly complex business logic, leading to

- Tightly coupled code base
- Slower release cadence
- Poor discoverability
- Additional complexity

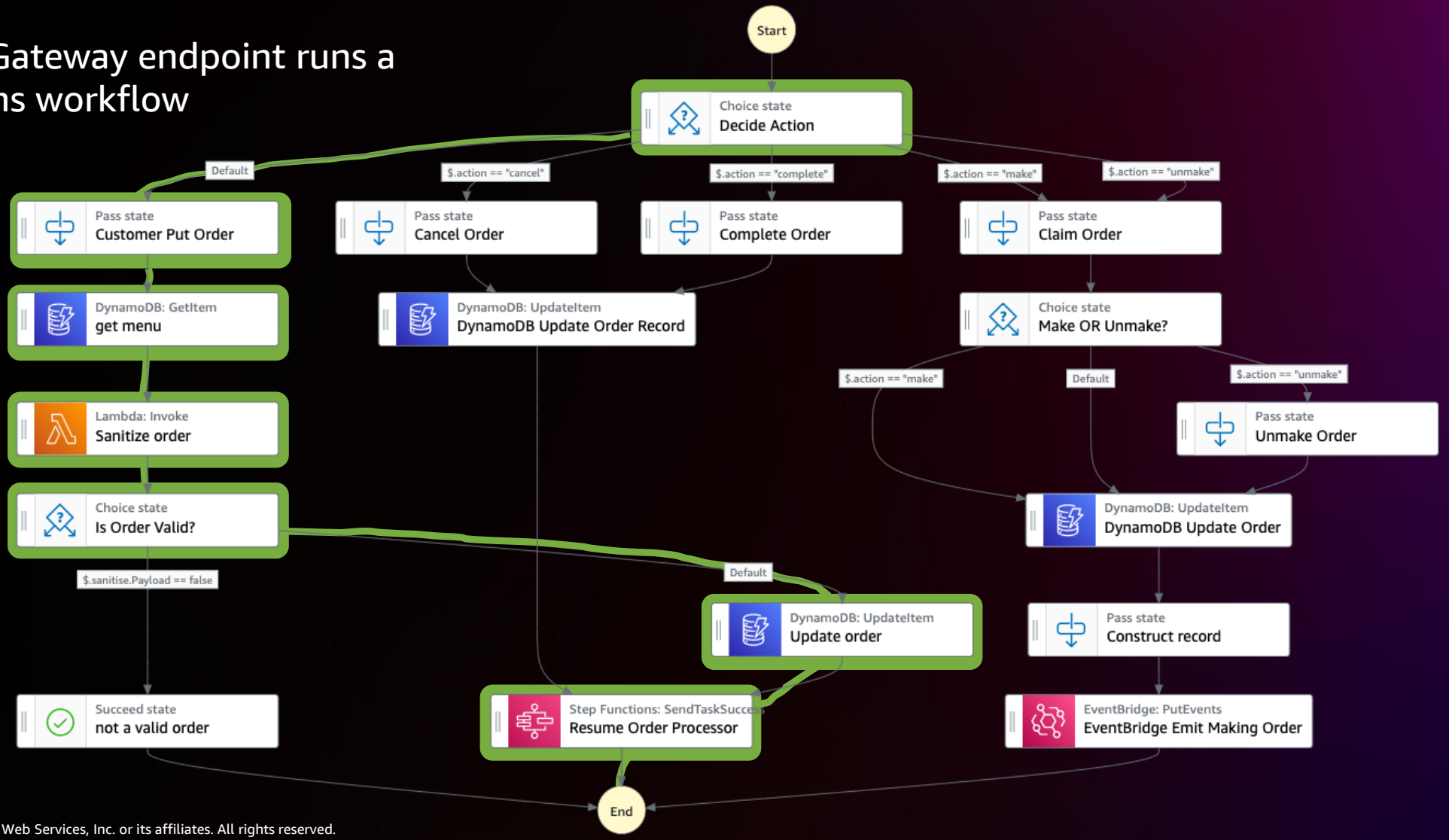
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Order manager service as a workflow

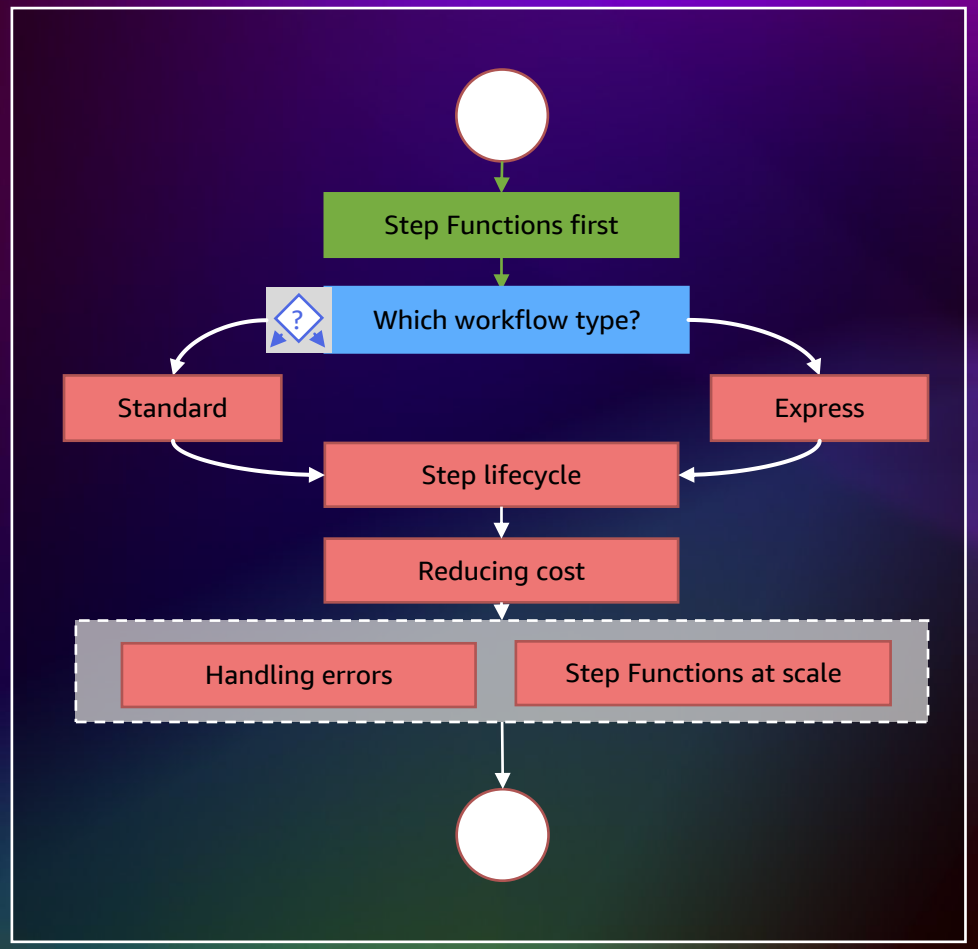
Version 2

A single API Gateway endpoint runs a Step Functions workflow



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Choosing Standard or Express Workflows

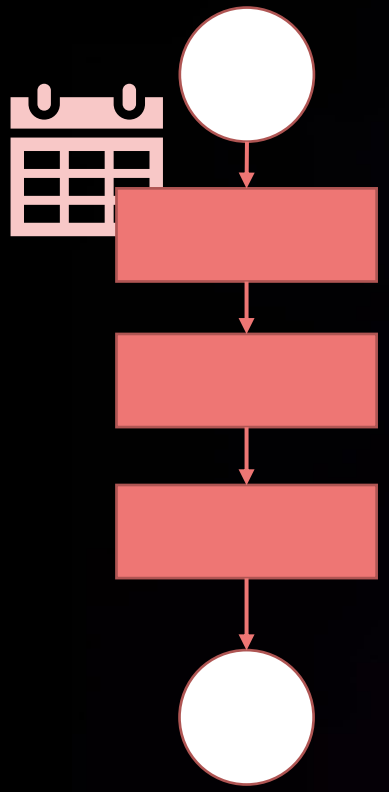


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Express

Standard vs

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



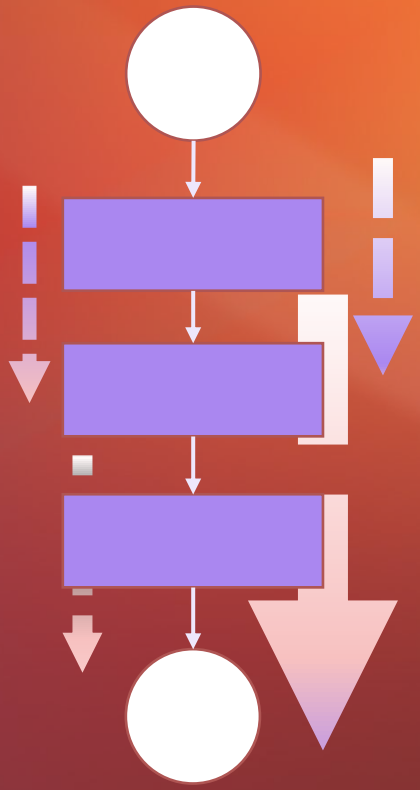
Standard

- Long-lasting
- Asynchronous
- Exactly once

vs

Express

- High throughput
- Cost effective
- At least once
- Synchronous
- Asynchronous
- Short duration



aws

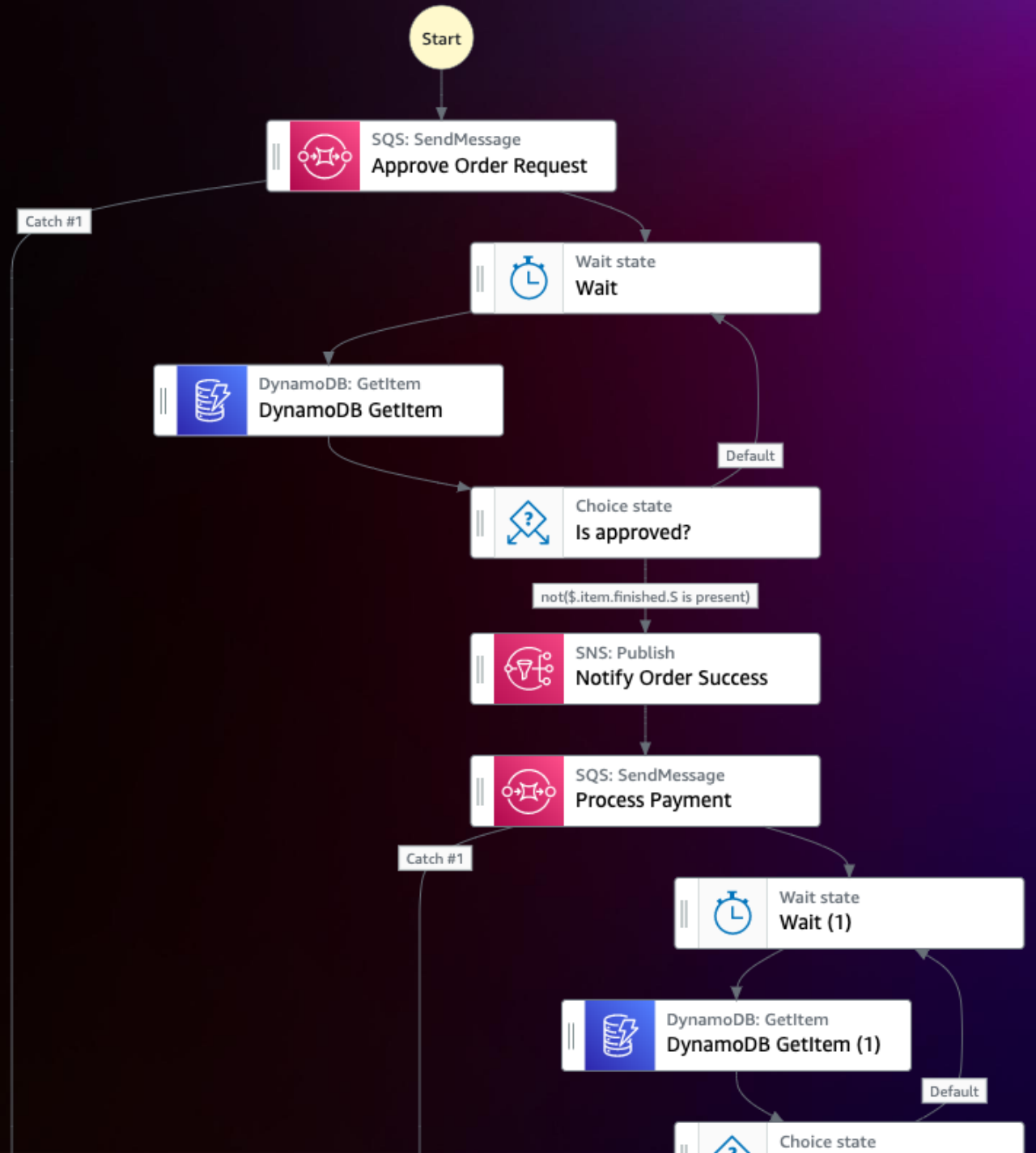
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Real-life example

An ecommerce workflow

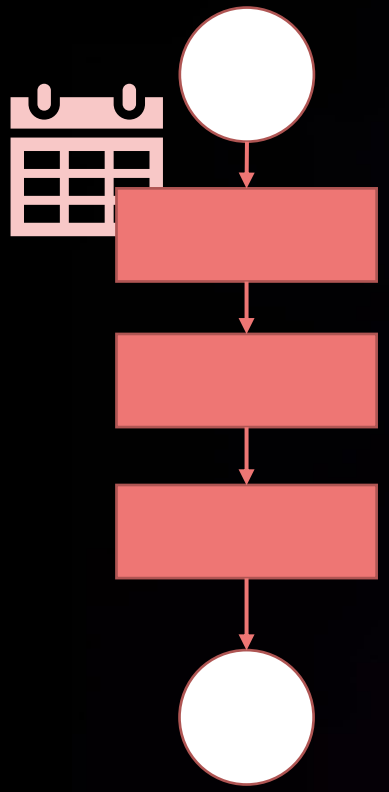
1,000 as a standard
1,000 as a express

I create an Amazon CloudWatch dashboard to display the average execution times



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Average duration



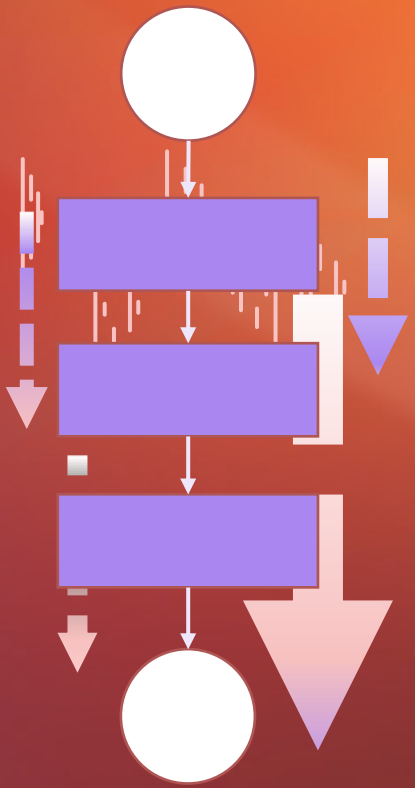
Standard

11.9s

vs

Express

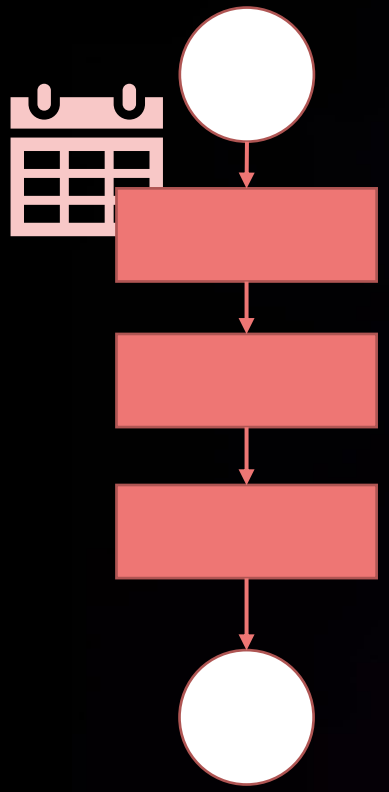
11.3s



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cost per thousand executions



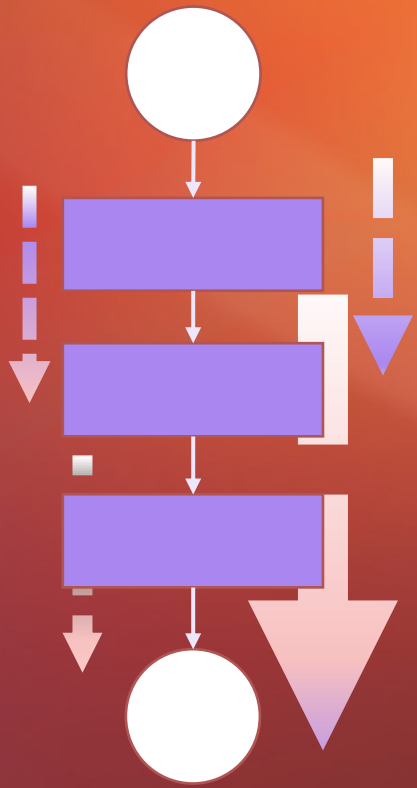
Standard

$ 0.42*

vs

Express

$ 0.01



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cost per thousand executions

CHARGED BASED ON THE NUMBER OF STATE TRANSITIONS

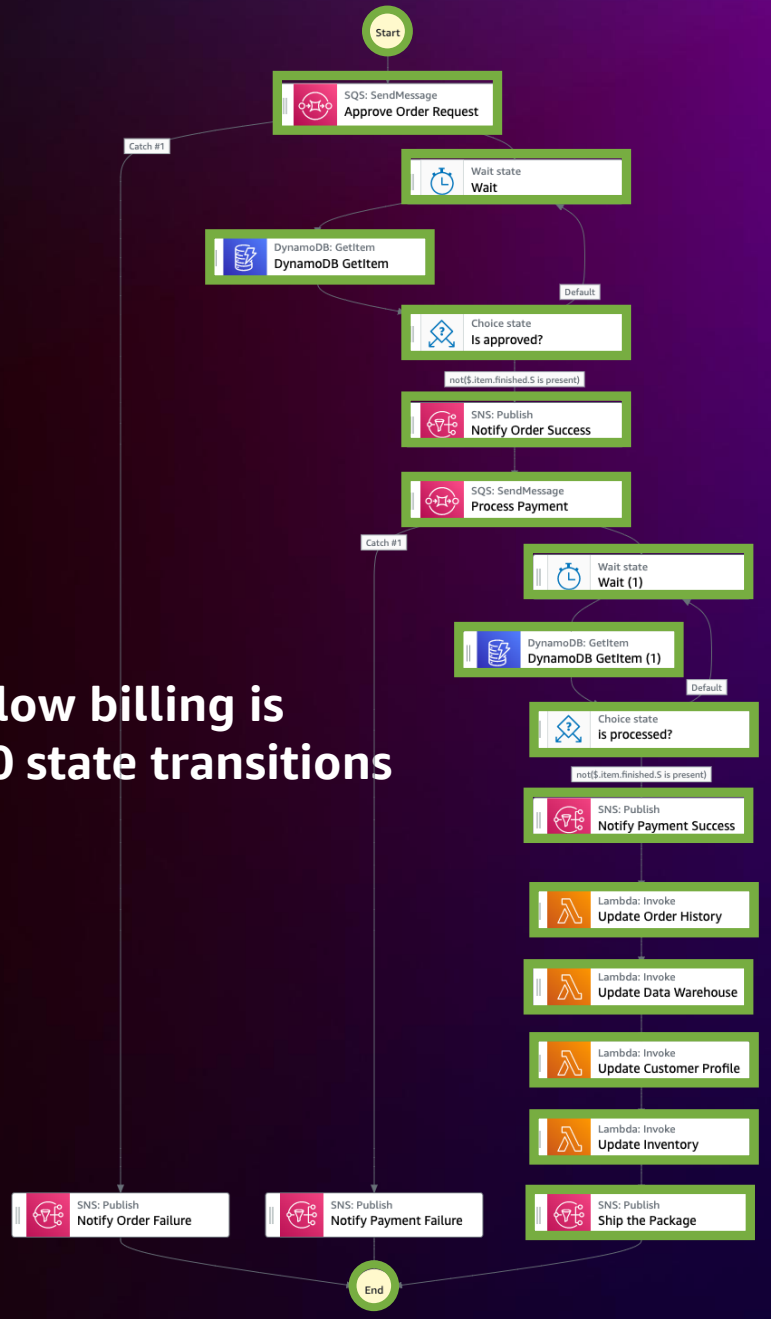
Total cost for a Standard Workflow

transitions x (# of executions x $0.000025)

Total cost = 17 x 0.025 = $0.42*

Total cost = $0.42 per 1000 executions

Standard Workflow billing is $0.025 per 1,000 state transitions



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cost per thousand executions

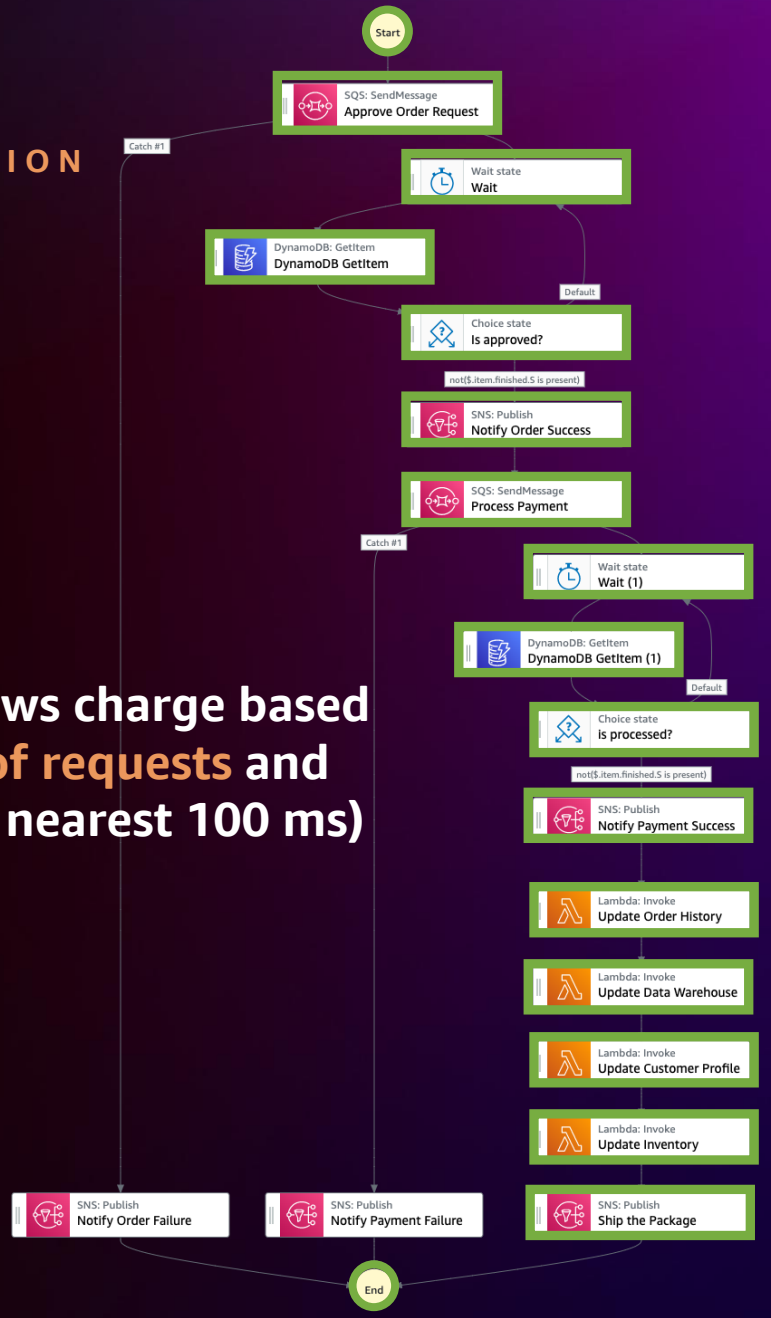
EXPRESS WORKFLOWS CHARGE BASED ON NUMBER OF REQUESTS AND DURATION

Total cost for a Express Workflow

(Execution cost + duration cost) x # of requests

Duration cost = execution duration ms / 100 * memory cost

Express Workflows charge based on the number of requests and duration (to the nearest 100 ms)



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cost per thousand executions

EXPRESS WORKFLOWS APPROXIMATE PRICE PER 100 MS FOR DIFFERENT MEMORY SIZES

Memory (MB)	Price per 100 ms ($) for first 1,000 GB-hours	Price per 100 ms ($) for next 4,000 GB-hours	Price per 100 ms ($) additional GB-hour
64	**0.0000001042**	0.0000000521	0.0000000285
128	0.0000002083	0.0000001042	0.0000000570
192	0.0000003125	0.0000001563	0.0000000856
256	0.0000004167	0.0000002083	0.0000001141
Each additional 64 MB	0.0000001042	0.0000000521	0.0000000285

Step Functions pricing page: https://aws.amazon.com/step-functions/pricing/

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cost per thousand executions

EXPRESS WORKFLOWS CHARGE BASED ON NUMBER OF REQUESTS AND DURATION

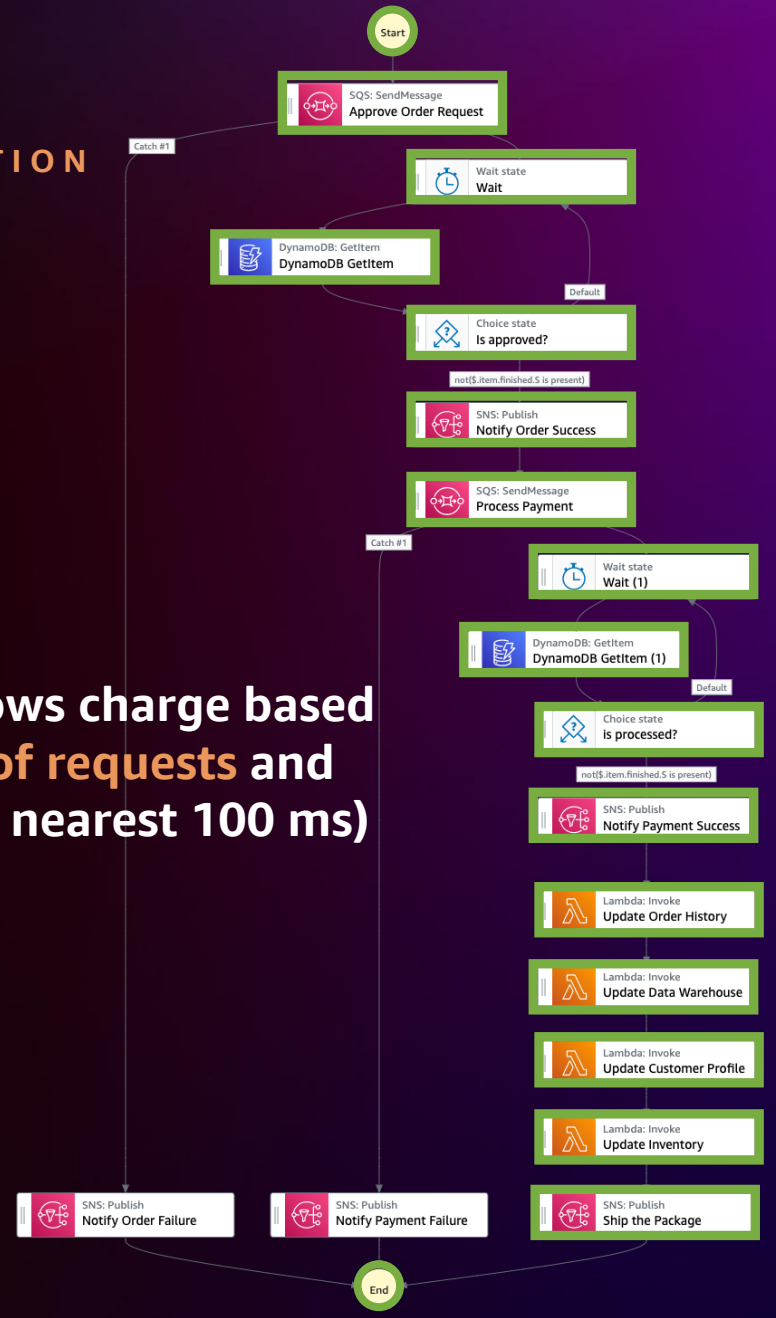
Total cost for a Express Workflow

(Execution cost + duration cost) x # of requests

Duration cost = (11,300 ms / 100) * $0.0000001042 = $0.0000117746

Execution cost = $0.000001 per request

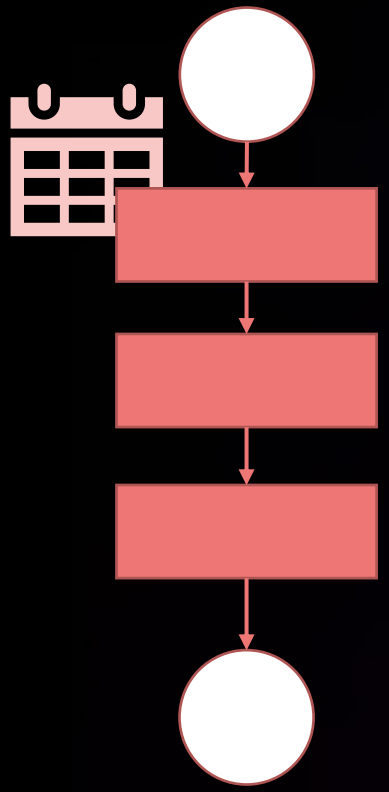
($0.000001 + $0.0000117746) x 1000 = $0.01



Express Workflows charge based on the number of requests and duration (to the nearest 100 ms)

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cost per thousand executions



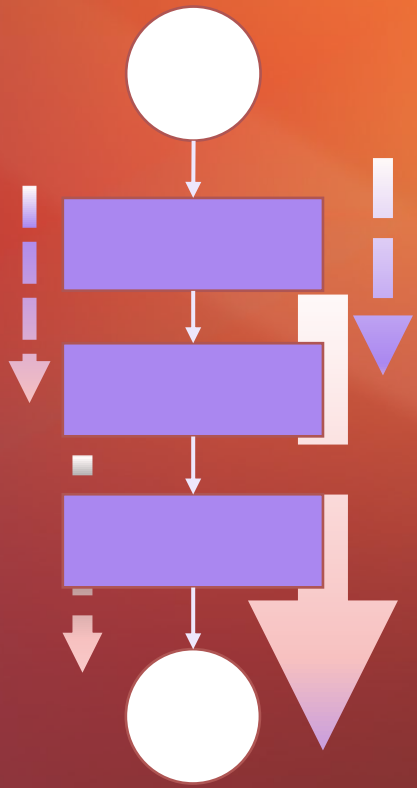
Standard

$ 0.42*

vs

Express

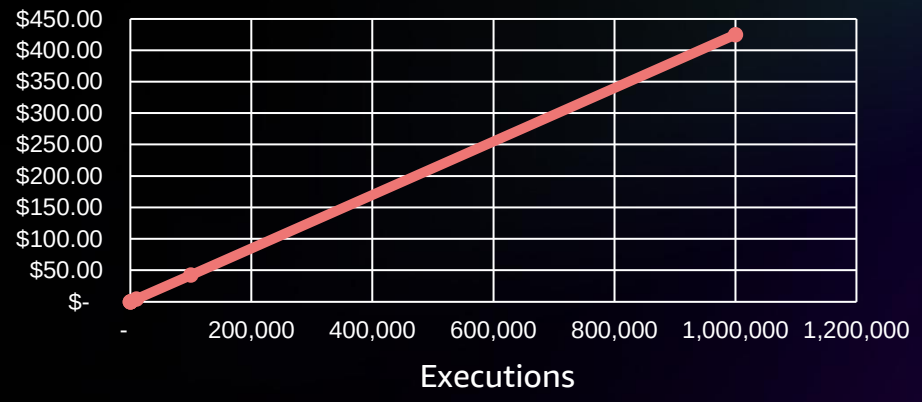
$ 0.01



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Standard Workflow cost



Standard

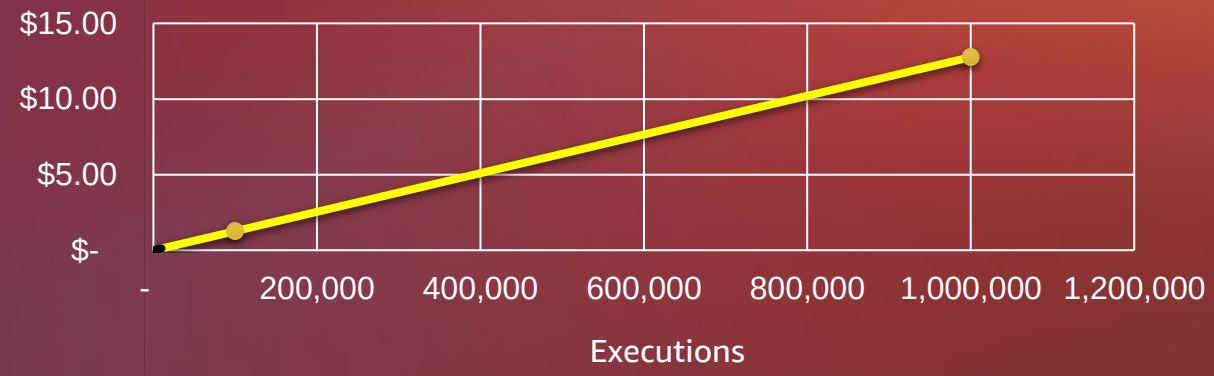
1 million executions = $420

VS

1 million executions = $12.77

Express

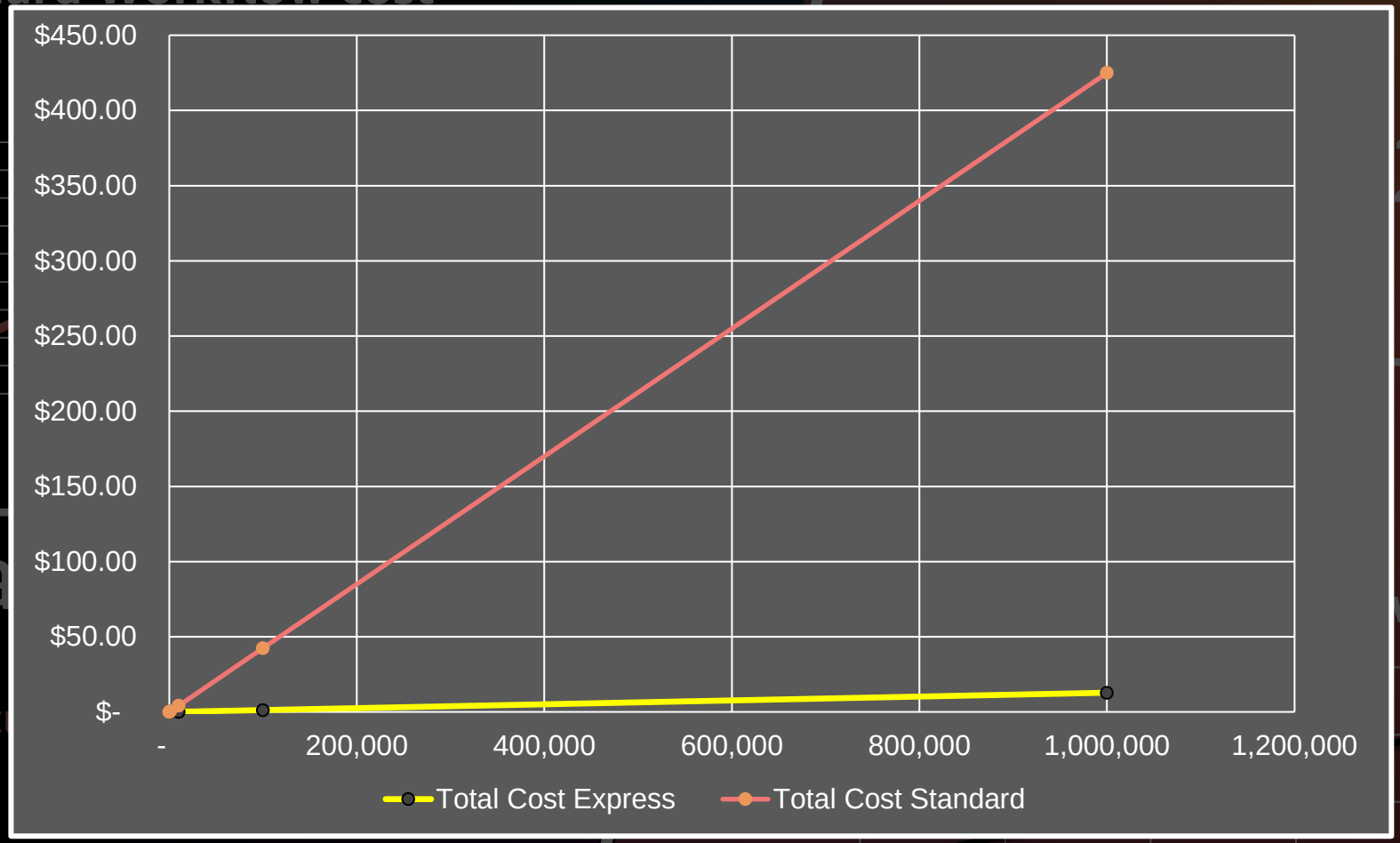
Express Workflow cost



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Standard vs Express Workflow: 1 million executions

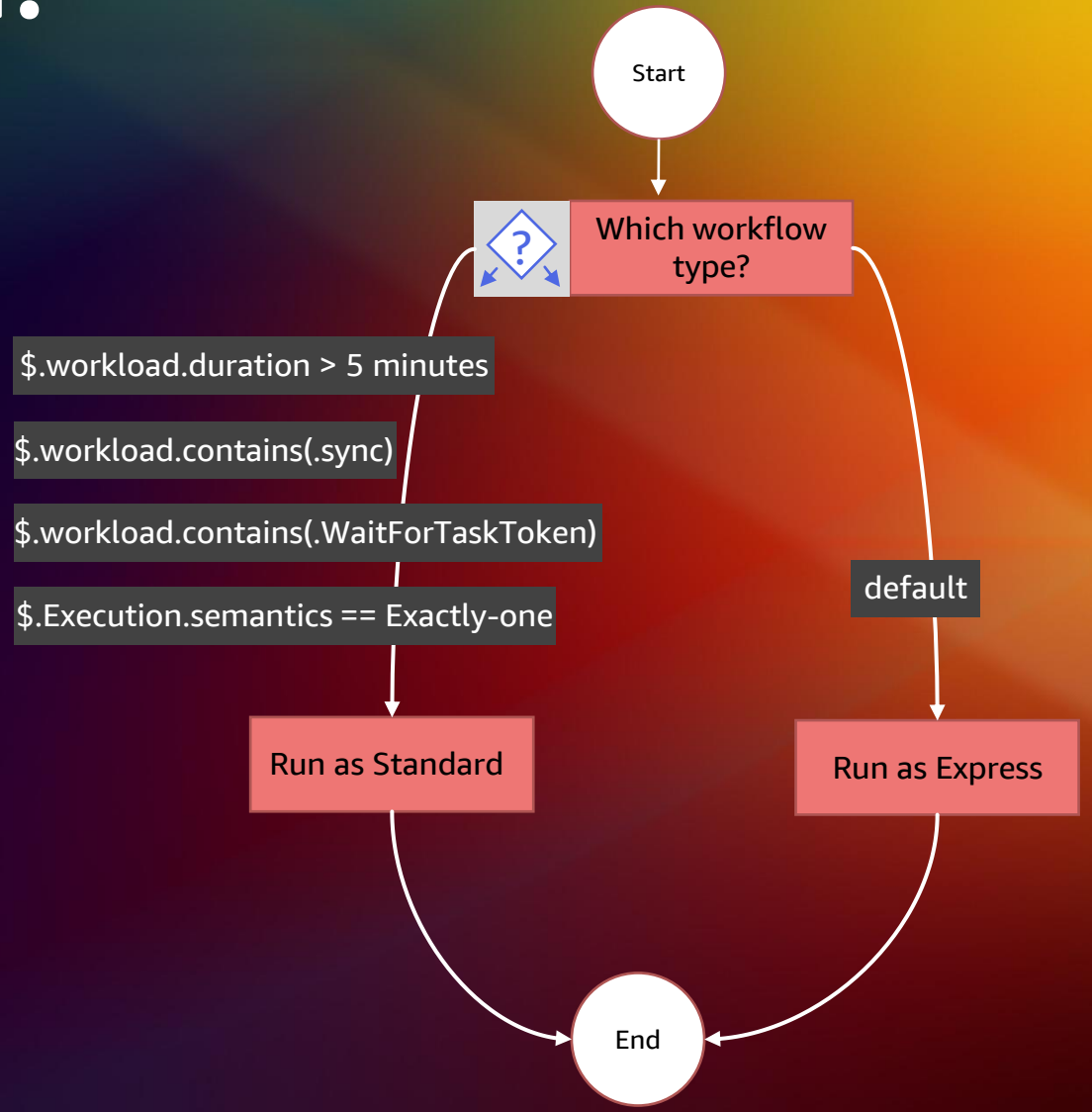


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

So why choose Standard?

When the workload

- Takes more than 5 minutes
- Uses callback (.WaitForTaskToken) or .Sync patterns
- Requires exactly-once execution



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

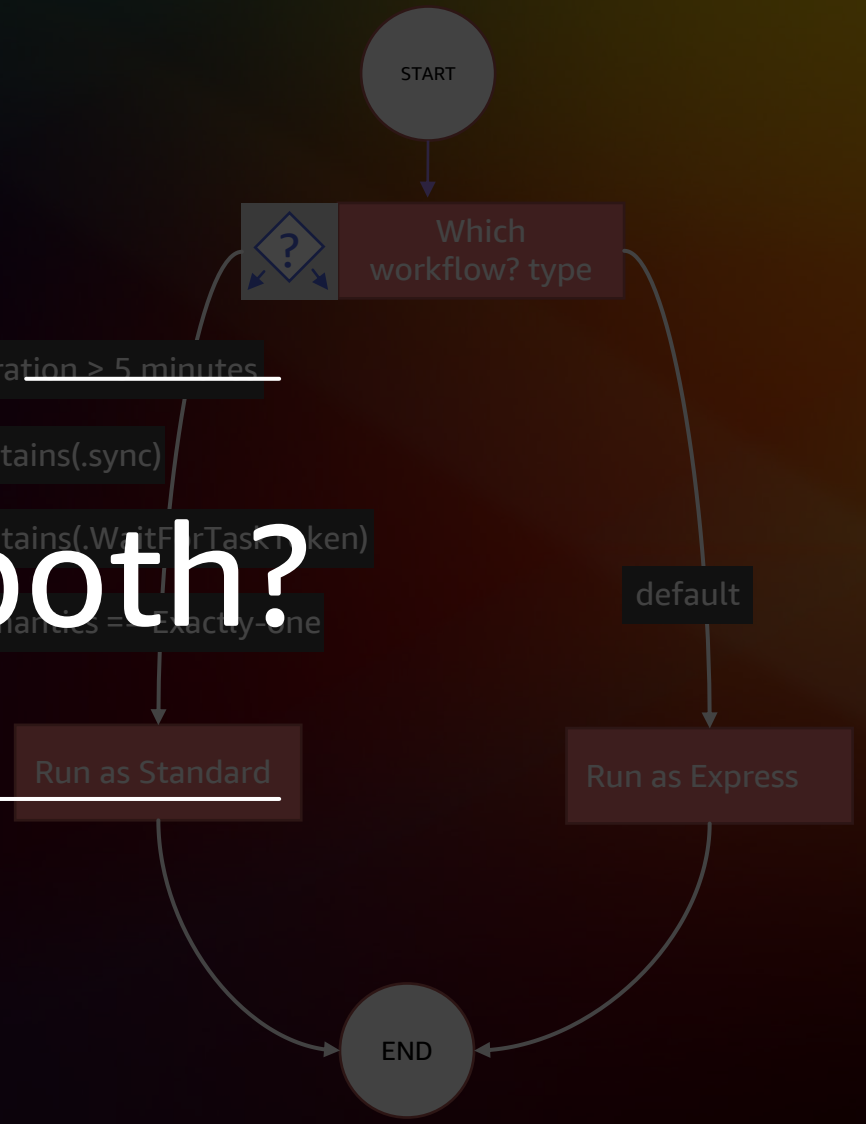
So why choose Standard?

When the workload:

- Takes more than 5 minutes
- Uses callback (.TaskToken) or .Sync patterns
- Required exactly-once execution

Standard or Express

How about both?



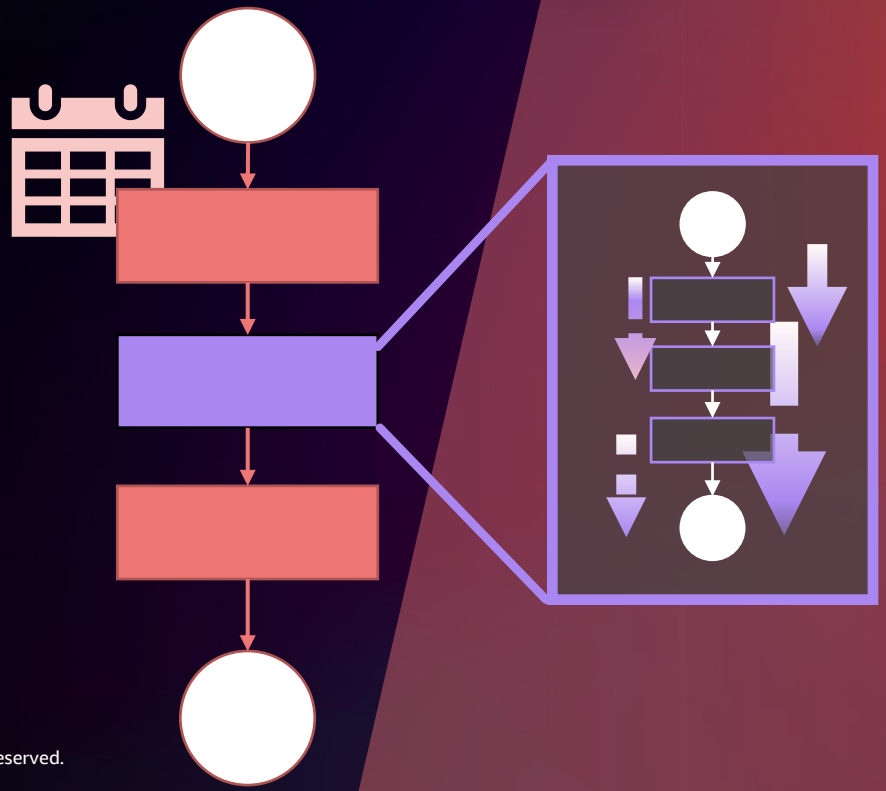
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pattern: The nester



Customer value

Standard & Express



Create **nested** workflows that combine workflow types to optimize for cost

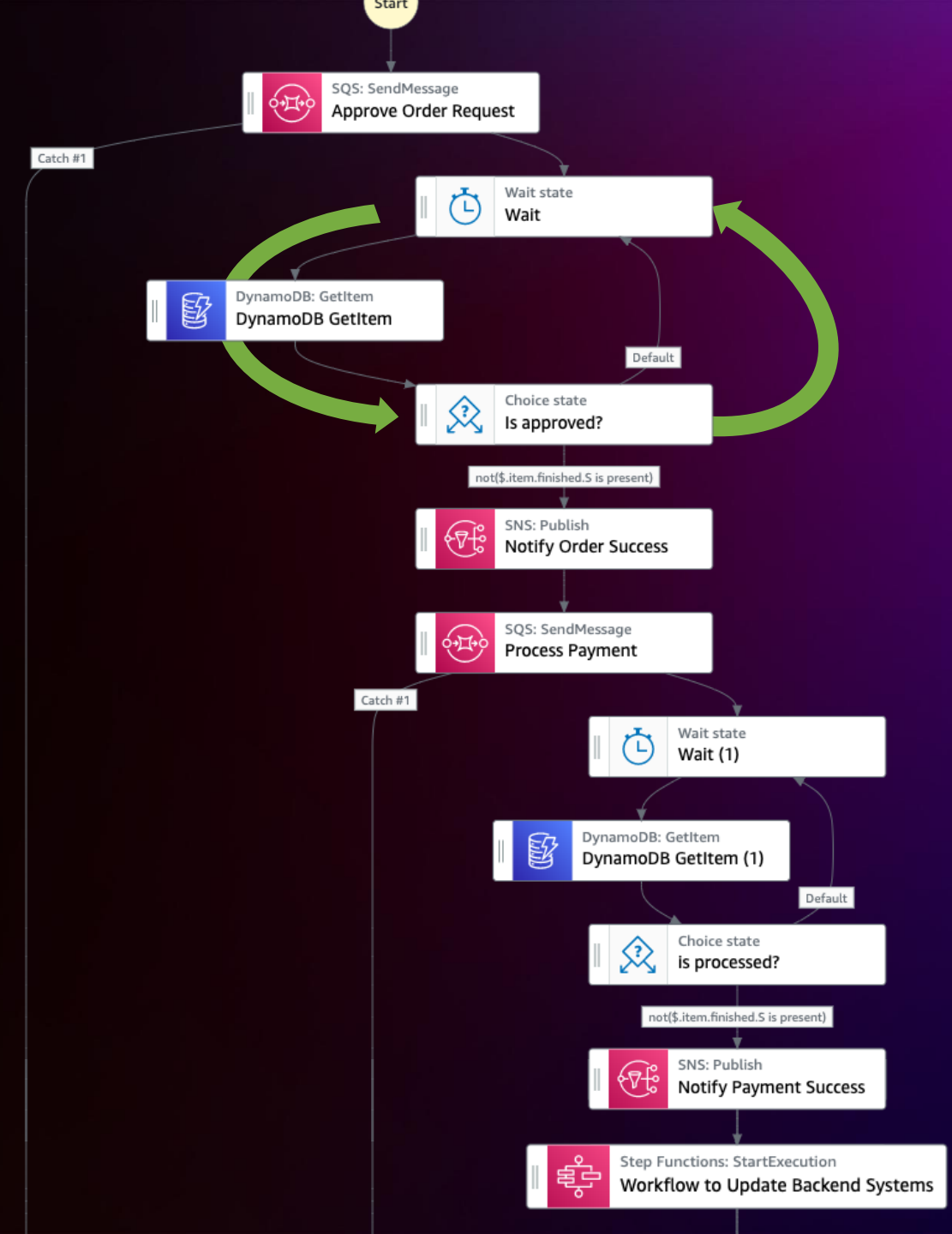
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How about both?

What if this sequence sometimes took longer than 5 minutes?

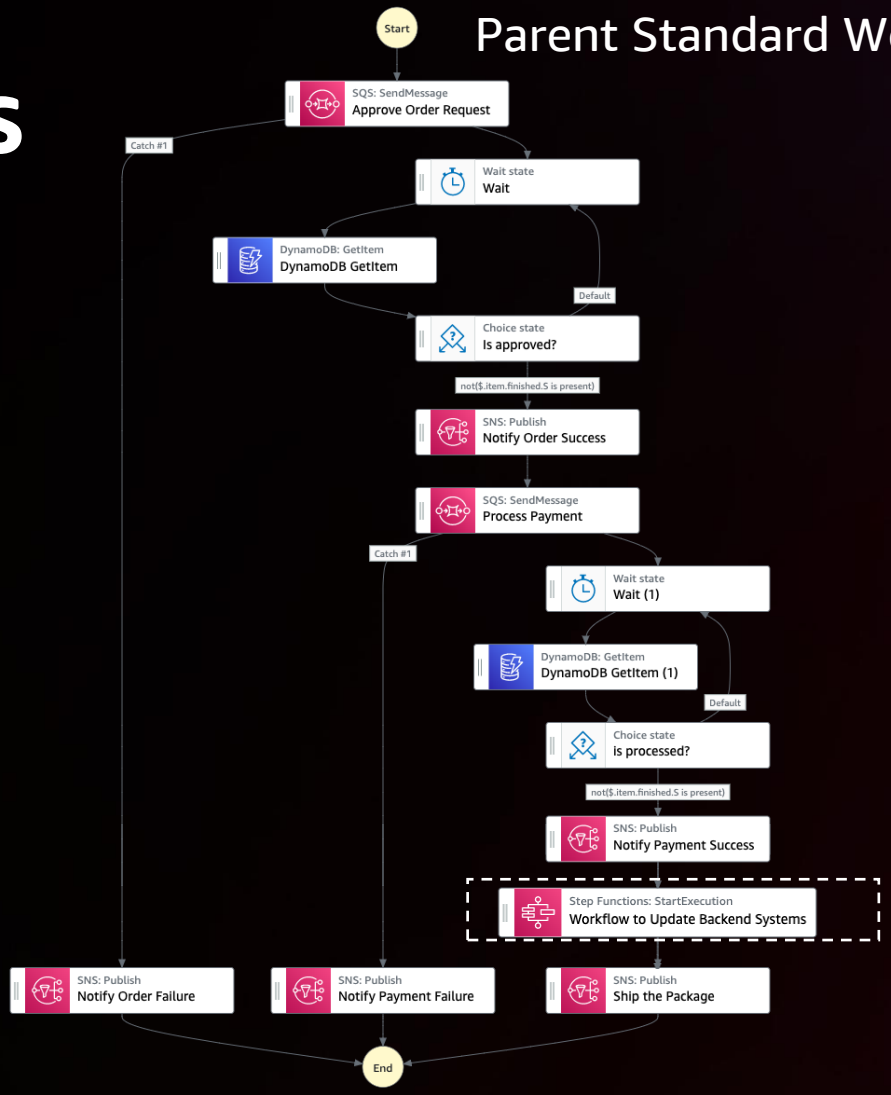
The workflow must run as a **standard** workflow



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Nested workflows

Parent Standard Workflow



.StartExecution

Place the idempotent states into a nested Express Workflow



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cost per thousand executions

EXPRESS WORKFLOWS CHARGE BASED ON NUMBER OF REQUESTS AND DURATION

Total cost for nest workflow

Cost of parent WF — Number of state transitions

(14*1000) * $0.000025 = $0.30

+

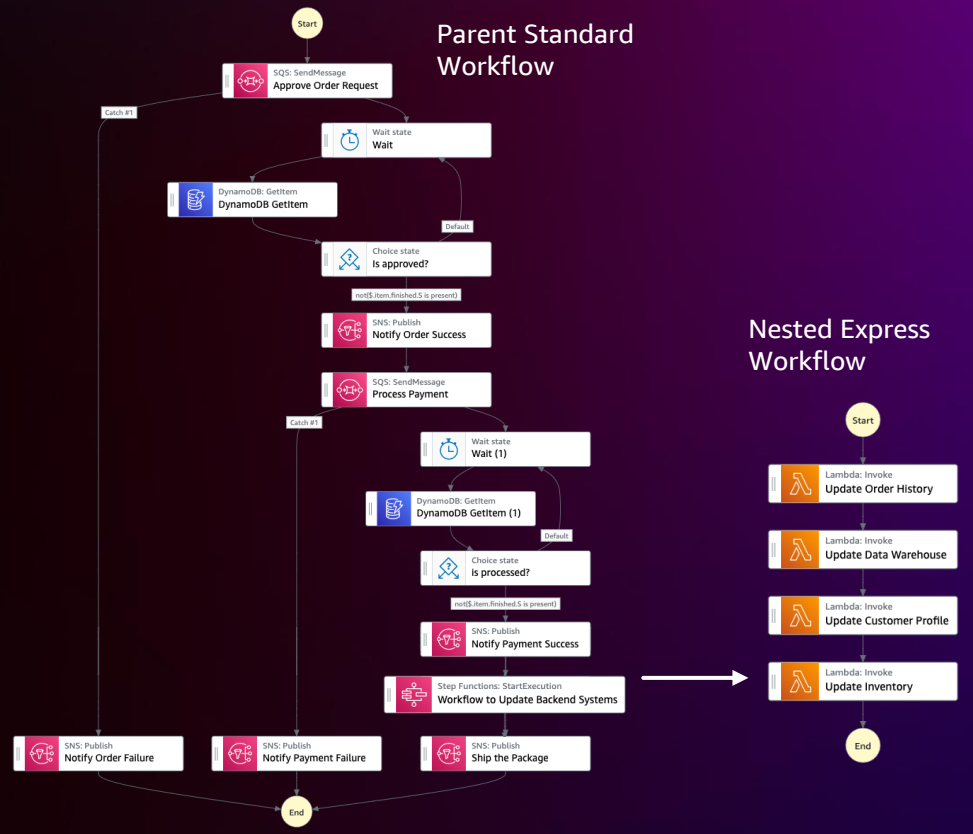
Cost of nested WF — Average duration — Price per 100 ms ($) for first 1,000 GB-hours

Duration cost = (1300 Ms /100) * $ 0.0000001042 = $0.000001

Execution cost = $0.000001

($0.000001 + $0.000001) x 1000 = $0.0002

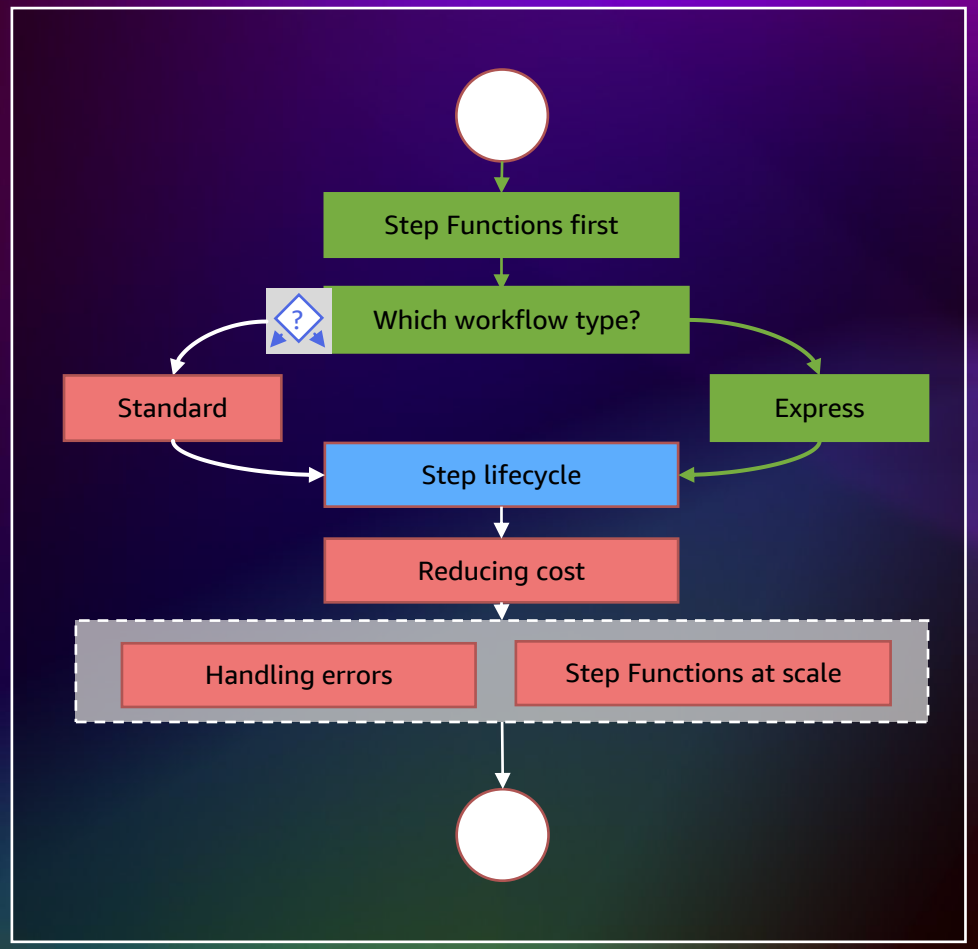
Total. = $0.30 per 1000 executions



This new workflow combination runs 1,000 times, costing a total cost of 30 cents

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step lifecycle



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

It all starts with an event...

```
{
    "SimpleKey": "SimpleValue",
    "ListKey": [
        "List value one",
        "List value two"
    ],
    "ObjectKey": {
        "NestedKey": "Nested value"
    }
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

...and some context

```
{
    "Execution": {
        "Id": "String",
        "Input": {},
        "StartTime": "Format: ISO 8601"
    },
    "State": {
        "EnteredTime": "Format: ISO 8601",
        "Name": "String",
        "RetryCount": Number
    },
    "StateMachine": {
        "Id": "String"
    },
    "Task": {
        "Token": "String"
    }
}
```

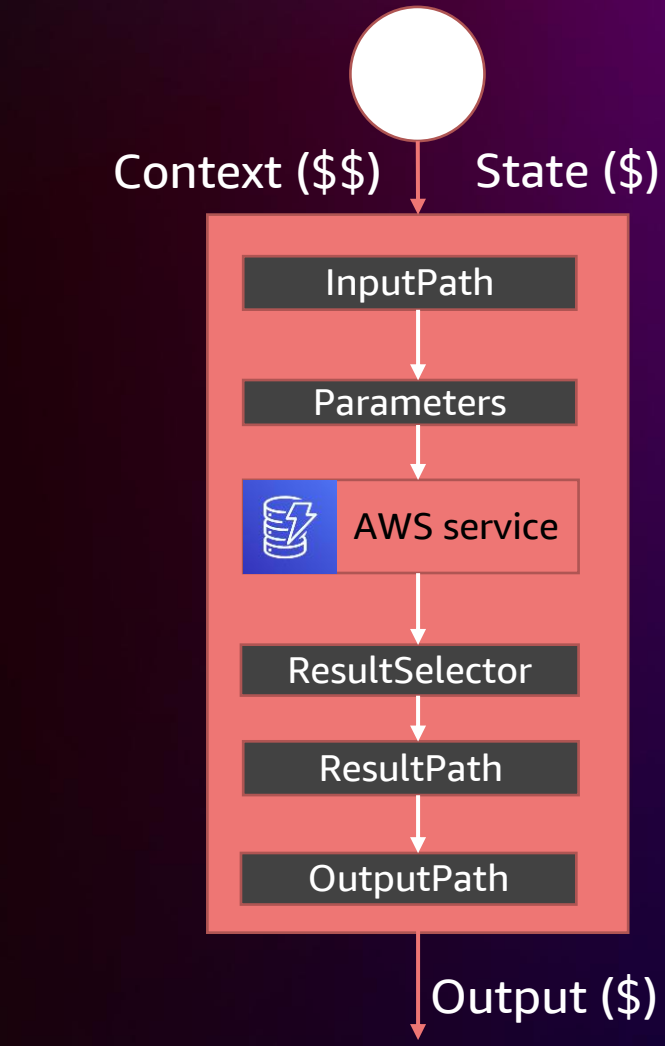
```
{
    "SimpleKey": "SimpleValue",
    "ListKey": [
        "List value one",
        "List value two"
    ],
    "ObjectKey": {
        "NestedKey": "Nested value"
    }
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Order of operations

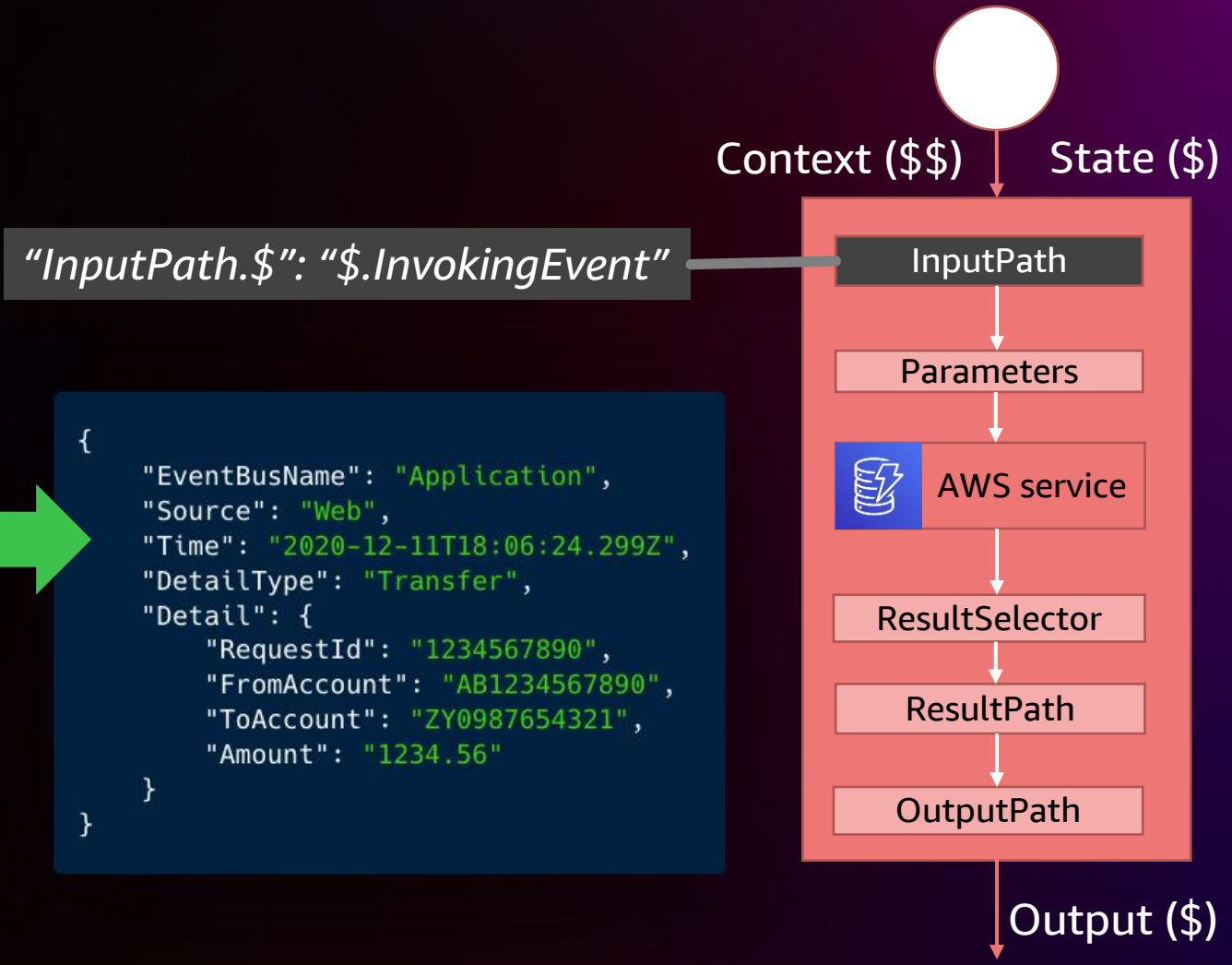
- InputPath
- Parameters
- Task state
- ResultSelector
- ResultPath
- OutputPath



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“InputPath”: “$.InvokingEvent”


```
{
    "ClusterId": "id-1234567890",
    "TableName": "SomeTable",
    "InvokingEvent": {
        "EventBusName": "Application",
        "Source": "Web",
        "Time": "2020-12-11T18:06:24.299Z",
        "DetailType": "Transfer",
        "Detail": {
            "RequestId": "1234567890",
            "FromAccount": "AB1234567890",
            "ToAccount": "ZY0987654321",
            "Amount": "1234.56"
        }
    }
}
```


```
{
    "EventBusName": "Application",
    "Source": "Web",
    "Time": "2020-12-11T18:06:24.299Z",
    "DetailType": "Transfer",
    "Detail": {
        "RequestId": "1234567890",
        "FromAccount": "AB1234567890",
        "ToAccount": "ZY0987654321",
        "Amount": "1234.56"
    }
}
```

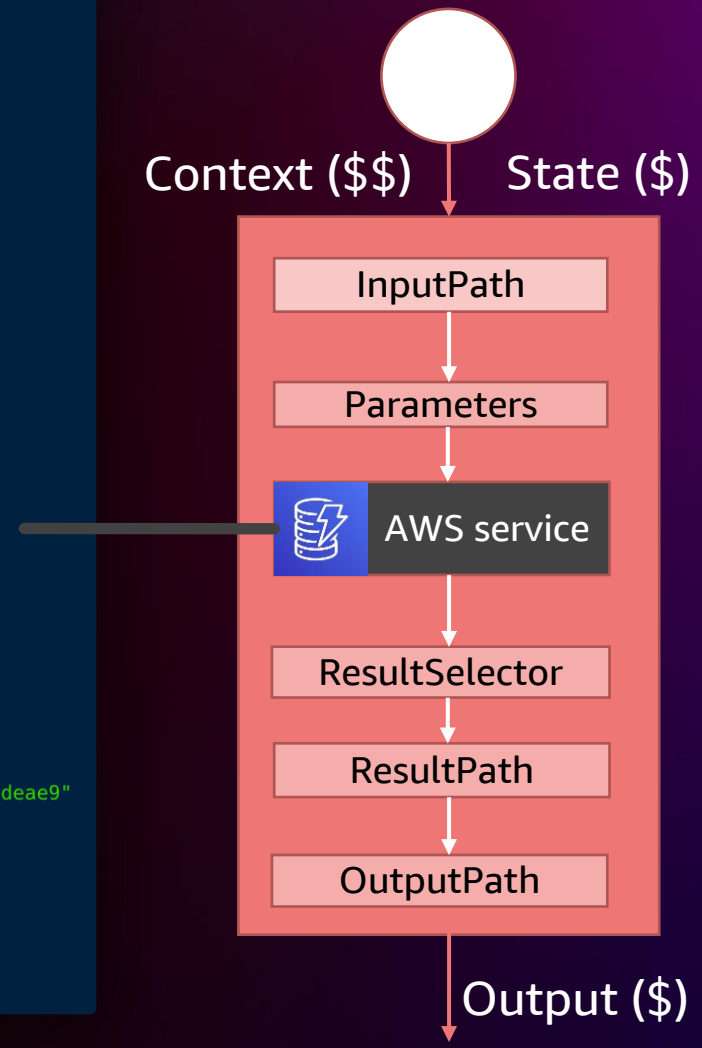

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Task state


```
{
    "EventBusName": "Application",
    "Source": "Web",
    "Time": "2020-12-11T18:06:24.299Z",
    "DetailType": "Transfer",
    "Detail": {
        "RequestId": "1234567890",
        "FromAccount": "AB1234567890",
        "ToAccount": "ZY0987654321",
        "Amount": "1234.56"
    }
}
```



```
{
    "SdkHttpMetadata": {
        "AllHttpHeaders": {
            "Server": [
                "Server"
            ],
            "Connection": [
                "keep-alive"
            ],
            "x-amzn-RequestId": [
                "01edf8c3-fadd-4368-b140-75026b7deae9"
            ],
            "x-amz-crc32": [
                "2745614147"
            ],
            "Content-Length": [
                "2"
            ],
            "Date": [
                "Fri, 11 Dec 2020 19:46:29 GMT"
            ],
            "Content-Type": [
                "application/x-amz-json-1.0"
            ]
        },
        "HttpHeaders": {
            "Connection": "keep-alive",
            "Content-Length": "2",
            "Content-Type": "application/x-amz-json-1.0",
            "Date": "Fri, 11 Dec 2020 19:46:29 GMT",
            "Server": "Server",
            "x-amz-crc32": "2745614147",
            "x-amzn-RequestId": "01edf8c3-fadd-4368-b140-75026b7deae9"
        },
        "HttpStatusCode": 200
    },
    "SdkResponseMetadata": {
        "RequestId": "01edf8c3-fadd-4368-b140-75026b7deae9"
    }
}
```

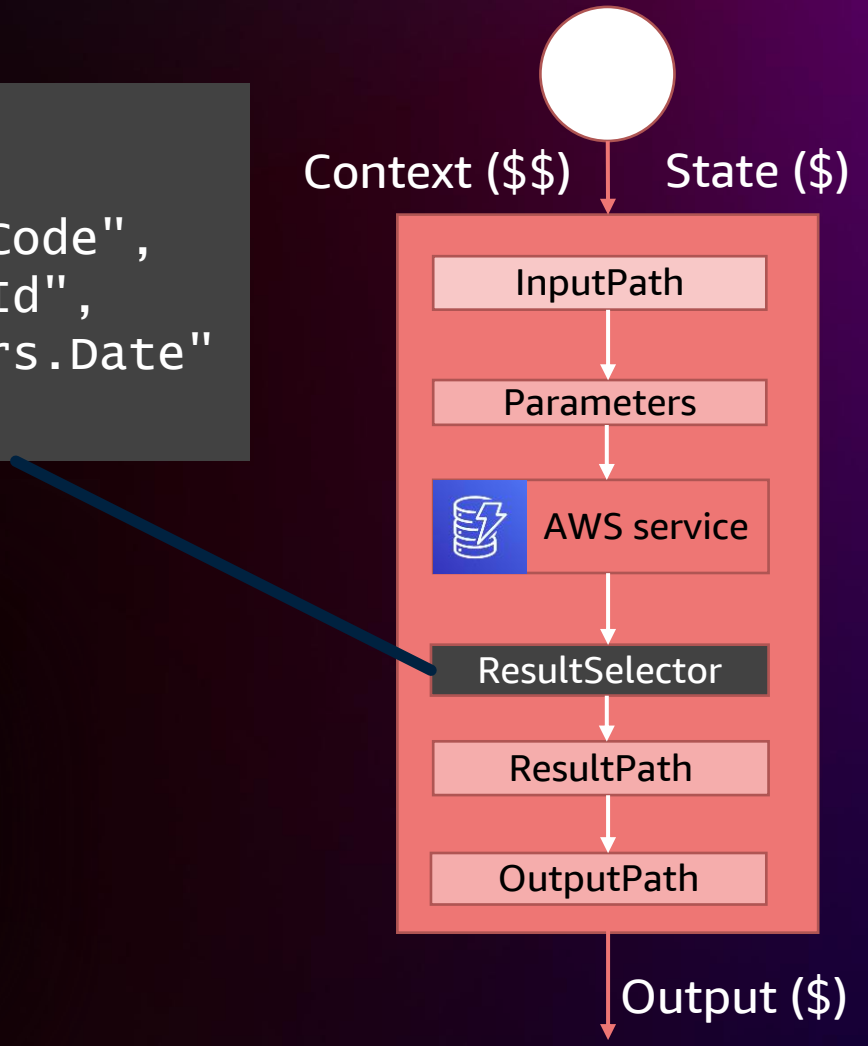
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

ResultSelector

```
Result Selector:
{
    "StatusCode.$": "$.SdkHttpMetadata.HttpStatusCode",
    "RequestId.$": "$.SdkResponseMetadata.RequestId",
    "RequestDate.$": "$.SdkHttpMetadata.HttpHeaders.Date"
}
```



```
{
    "StatusCode": 200,
    "RequestId": "01edf8c3-fadd-4368-b140-75026b7deae9",
    "RequestDate": "Fri, 11 Dec 2020 19:46:29 GMT"
}
```

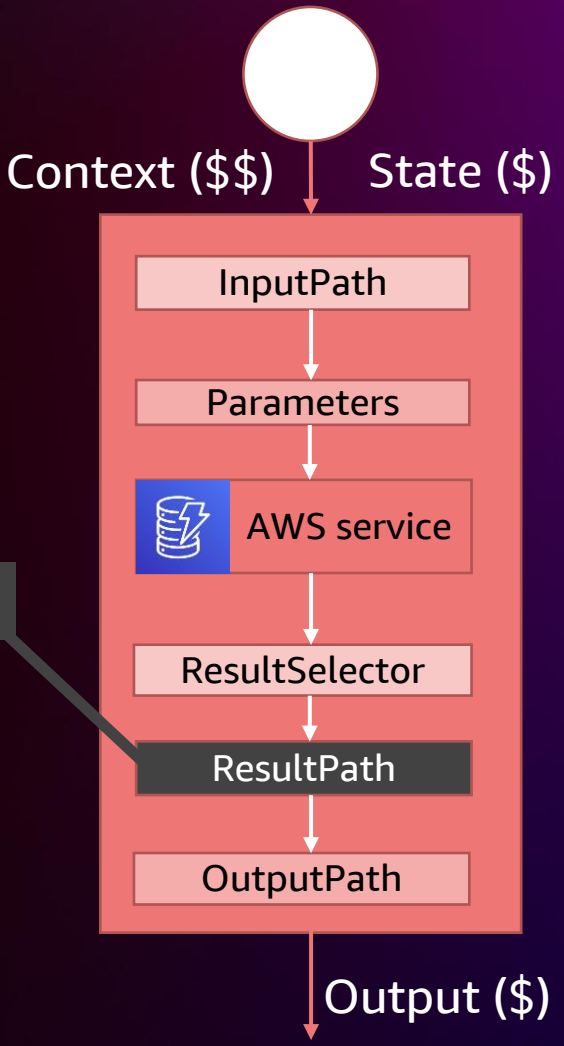


aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“ResultPath”: “$.Detail.LockInfo”


```
{
    "EventBusName": "Application",
    "Source": "Web",
    "Time": "2020-12-11T18:06:24.299Z",
    "DetailType": "Transfer",
    "Detail": {
        "RequestId": "1234567890",
        "FromAccount": "AB1234567890",
        "ToAccount": "ZY0987654321",
        "Amount": "1234.56",
        "LockInfo": {
            "StatusCode": 200,
            "RequestId": "01edf8c3-fadd-4368-b140-75026b7deae9",
            "RequestDate": "Fri, 11 Dec 2020 19:46:29 GMT"
        }
    }
}
```


“ResultPath.$”: “$.Detail.LockInfo”

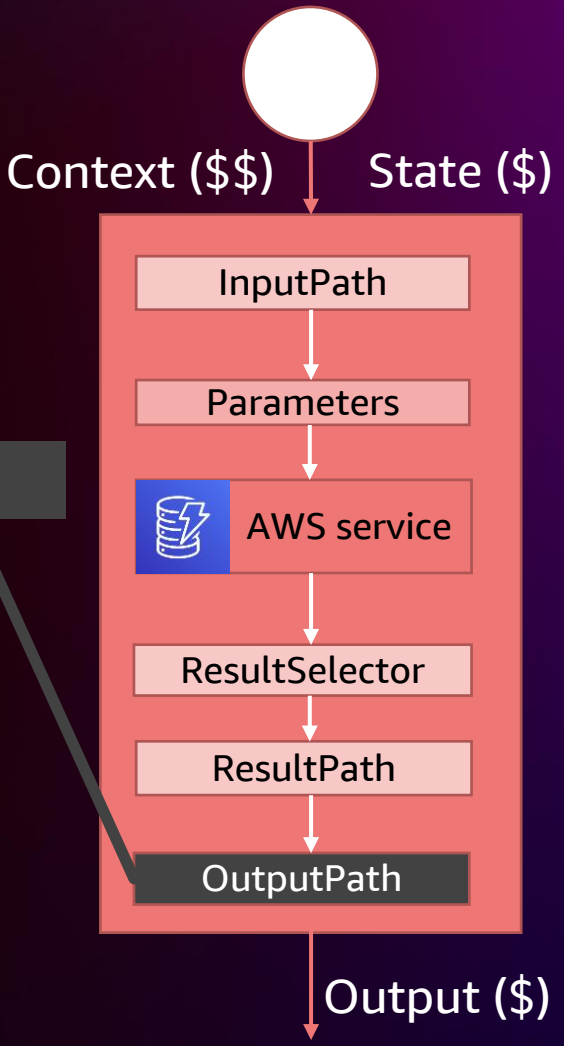


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“OutputPath”: “$.Detail”


```
{
    "RequestId": "1234567890",
    "FromAccount": "AB1234567890",
    "ToAccount": "ZY0987654321",
    "Amount": "1234.56",
    "LockInfo": {
        "StatusCode": 200,
        "RequestId": "01edf8c3-fadd-4368-b140-75026b7deae9",
        "RequestDate": "Fri, 11 Dec 2020 19:46:29 GMT"
    }
}
```

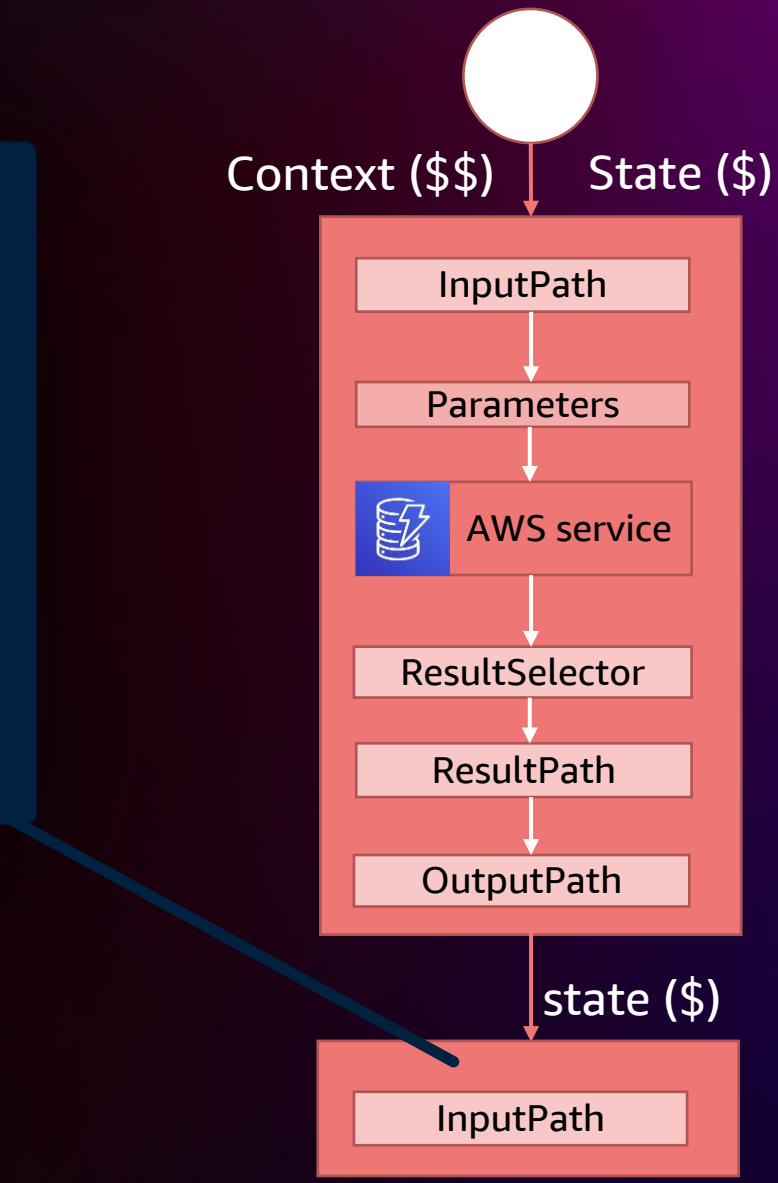

“OutputPath.$”: “$. Detail”



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

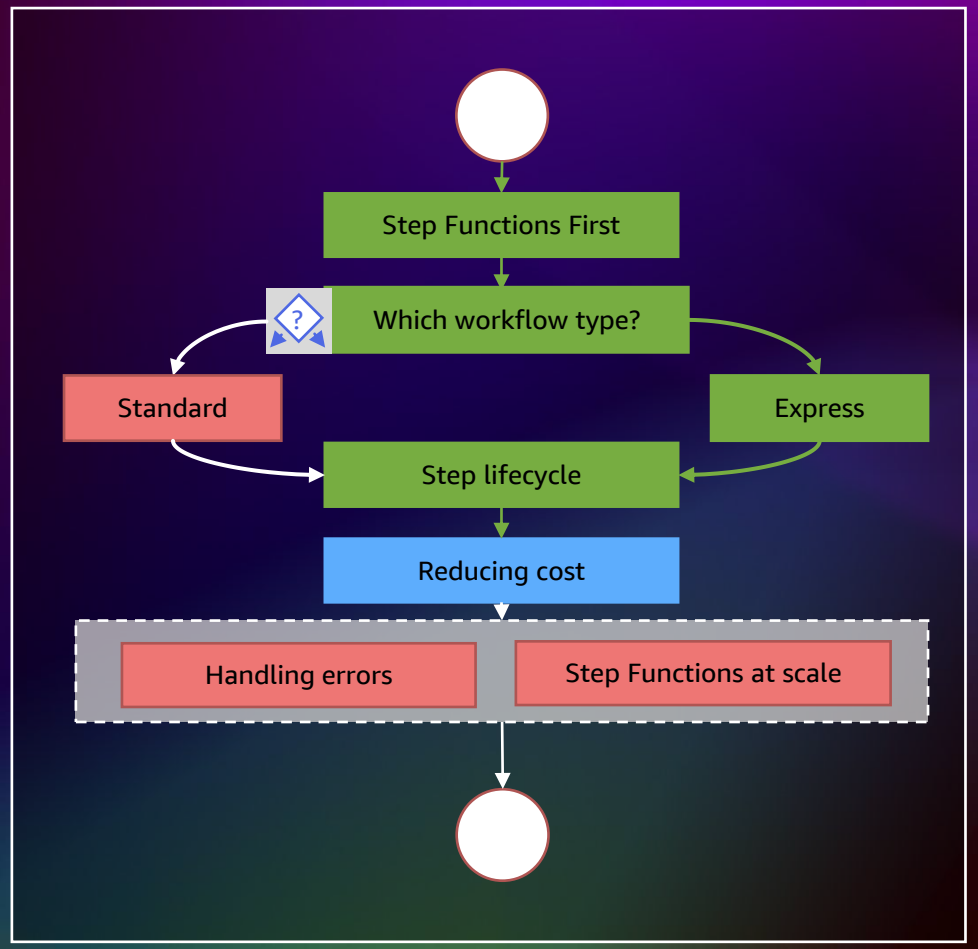
Now state is transformed


```
{
    "RequestId": "1234567890",
    "FromAccount": "AB1234567890",
    "ToAccount": "ZY0987654321",
    "Amount": "1234.56",
    "LockInfo": {
        "StatusCode": 200,
        "RequestId": "01edf8c3-fadd-4368-b140-75026b7deae9",
        "RequestDate": "Fri, 11 Dec 2020 19:46:29 GMT"
    }
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reducing cost

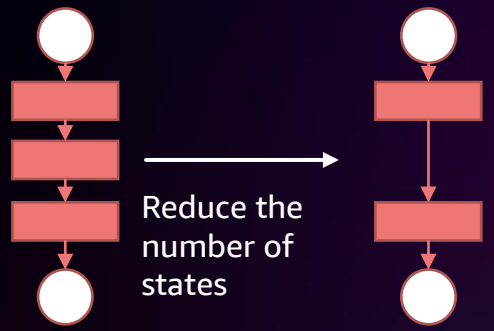


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reducing cost

Standard

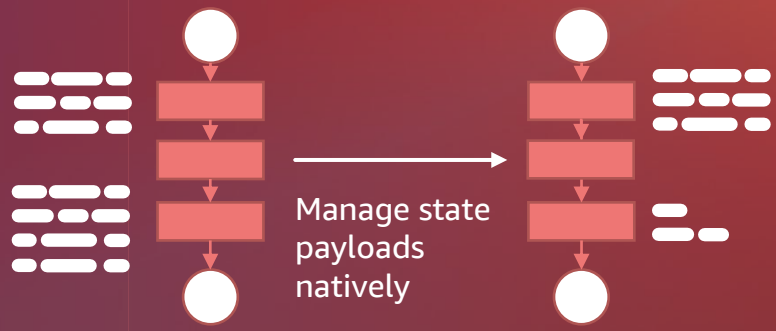
Reduce state transitions



VS

Express

Reduce execution duration
Reduce memory GB usage



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

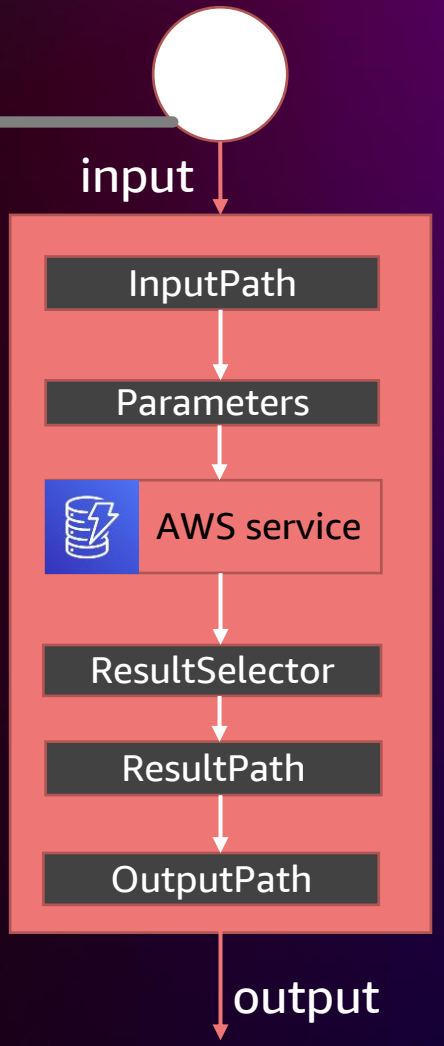
Reducing cost: Standard Workflows

REDUCE CUSTOM CODE AND STATE TRANSITIONS USING JSONPATH FILTERS

Reduce custom code and payload size by using JSONPath expressions instead of compute services to filter input payloads

JSON


```
{
  "version": 4,
  "library": {
    "movies": [
      {
        "genre": "crime",
        "director": "Quentin Tarantino",
        "title": "Reservoir Dogs",
        "year": 1992
      },
      {
        "genre": "action",
        "director": "Brian De Palma",
        "title": "Mission: Impossible",
        "year": 1996,
        "staring": [
          "Tom Cruise"
        ]
      }
    ],
    "metadata": {
      "lastUpdated": "2020-05-27T08:00:00.000Z"
    },
    "stringJson": "{\"arr\": [1, 2, 3, 4, 5], \"bool\": true, \"null\": null,
\"number\": 1}"
  },
  "TaskResult": {
    "modifiedPayload": {
      "body": "hello, world!",
      "statusCode": "200",
      "requestId": "88fba57b-adbe-467f-abf4-daca36fc9028"
    }
  }
}
```

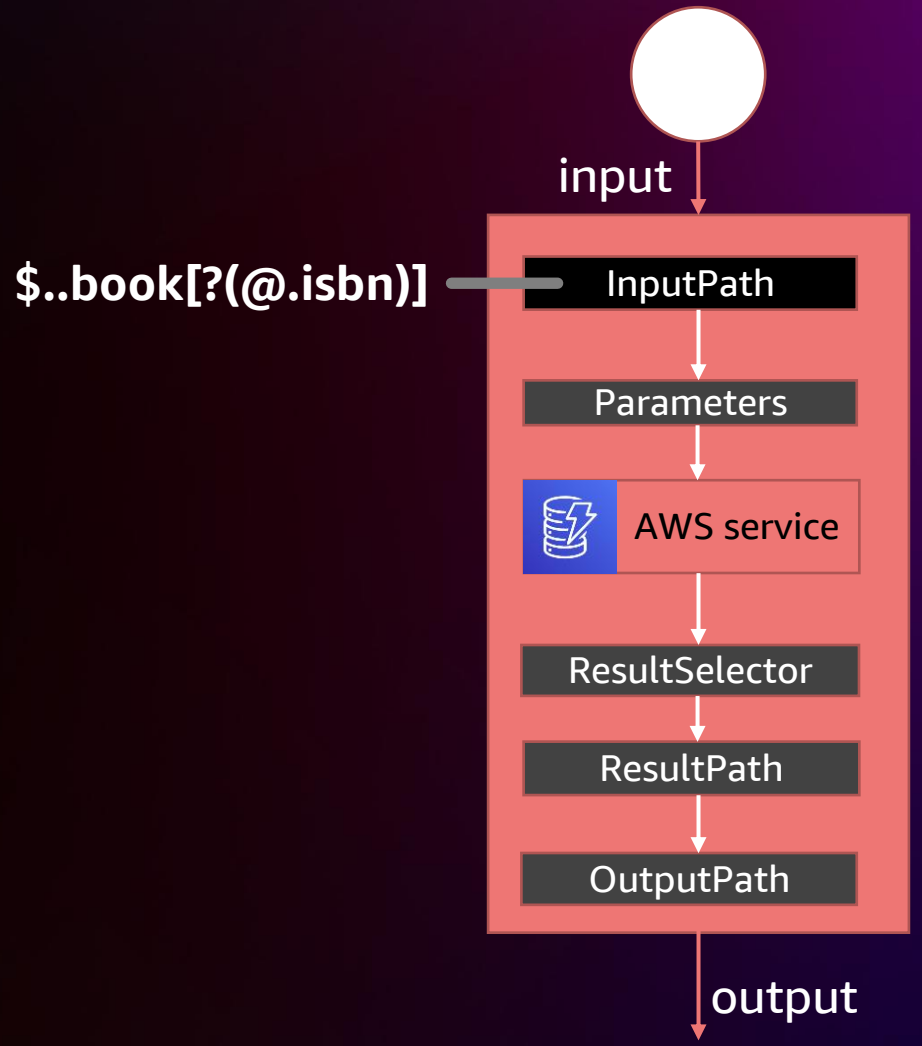



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reducing cost: Standard Workflows

REDUCE CUSTOM CODE AND STATE TRANSITIONS USING JSONPATH FILTERS

Reduce custom code and payload size by using JSONPath expressions instead of compute services to filter input payloads



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reducing cost: Standard Workflows

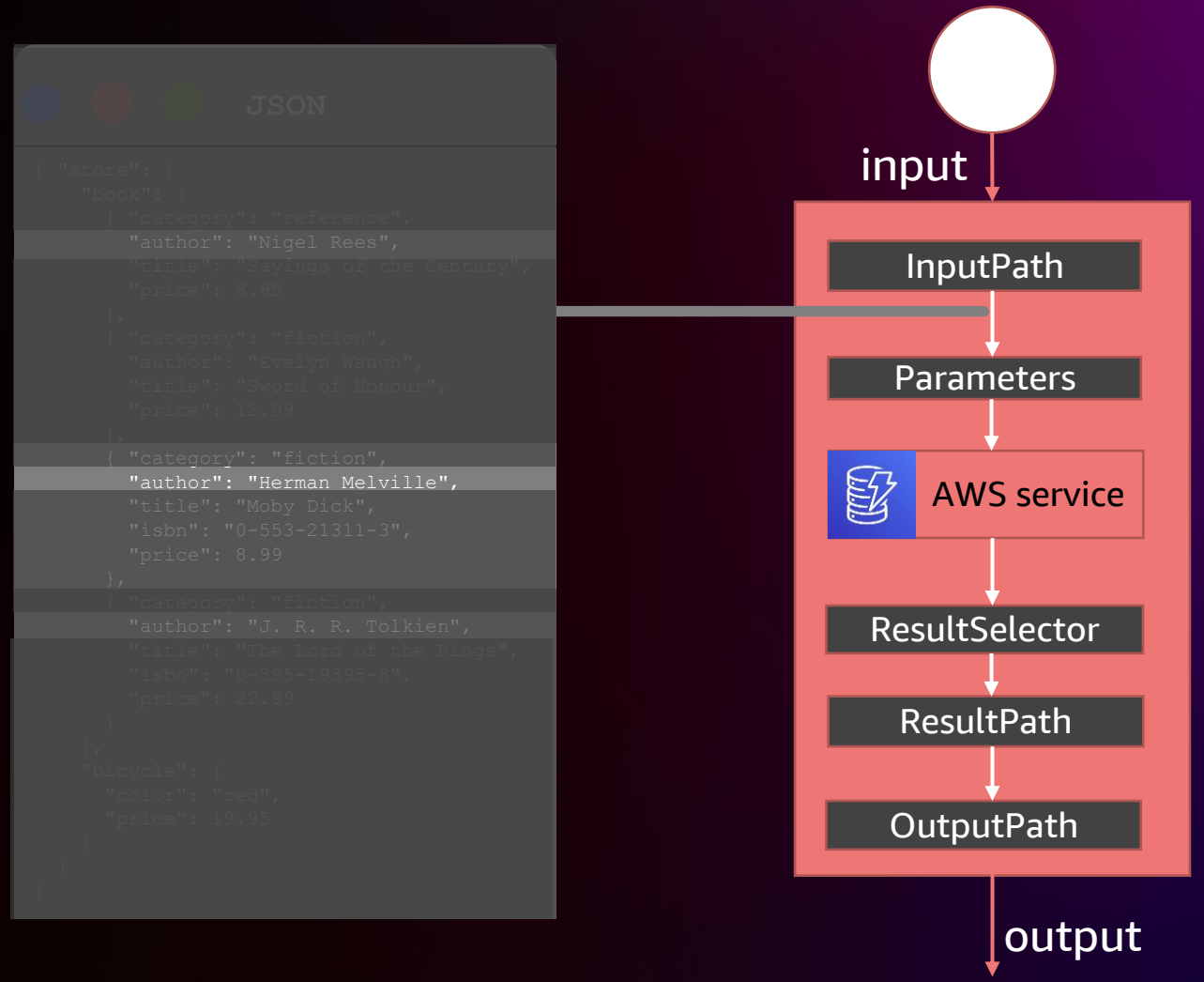
REDUCE CUSTOM CODE USING JSON FILTERS

Reduce custom code by using JSONPath expressions with the InputPath field

$..book[2] //the third book

$.store.book[*].author
// the authors of all books in the store

$..book[?(@.category==fiction &&
@.price < 10)] // filter all books
cheaper than 10 and category is fiction



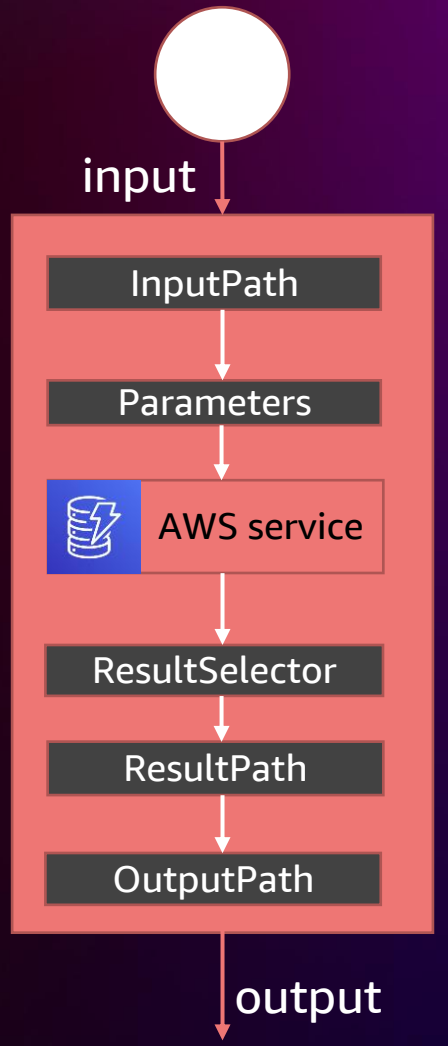
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reducing cost: Standard Workflows

REDUCE CUSTOM CODE AND STATE TRANSITIONS USING INTRINSIC FUNCTIONS

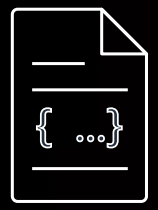
Use intrinsic functions instead of compute services to perform simple data transformations

Reduce the cost of running your workflows by decreasing the number of states, number of transitions, and total workflow duration



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Step Functions intrinsic functions



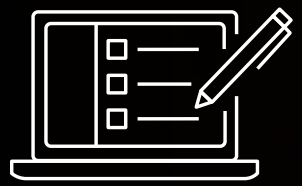
Arrays



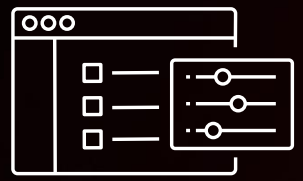
JSON data manipulation



Encoding and decoding



Math operations



String operations



Unique identifier generation

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

STATES._____

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Step Functions intrinsic functions

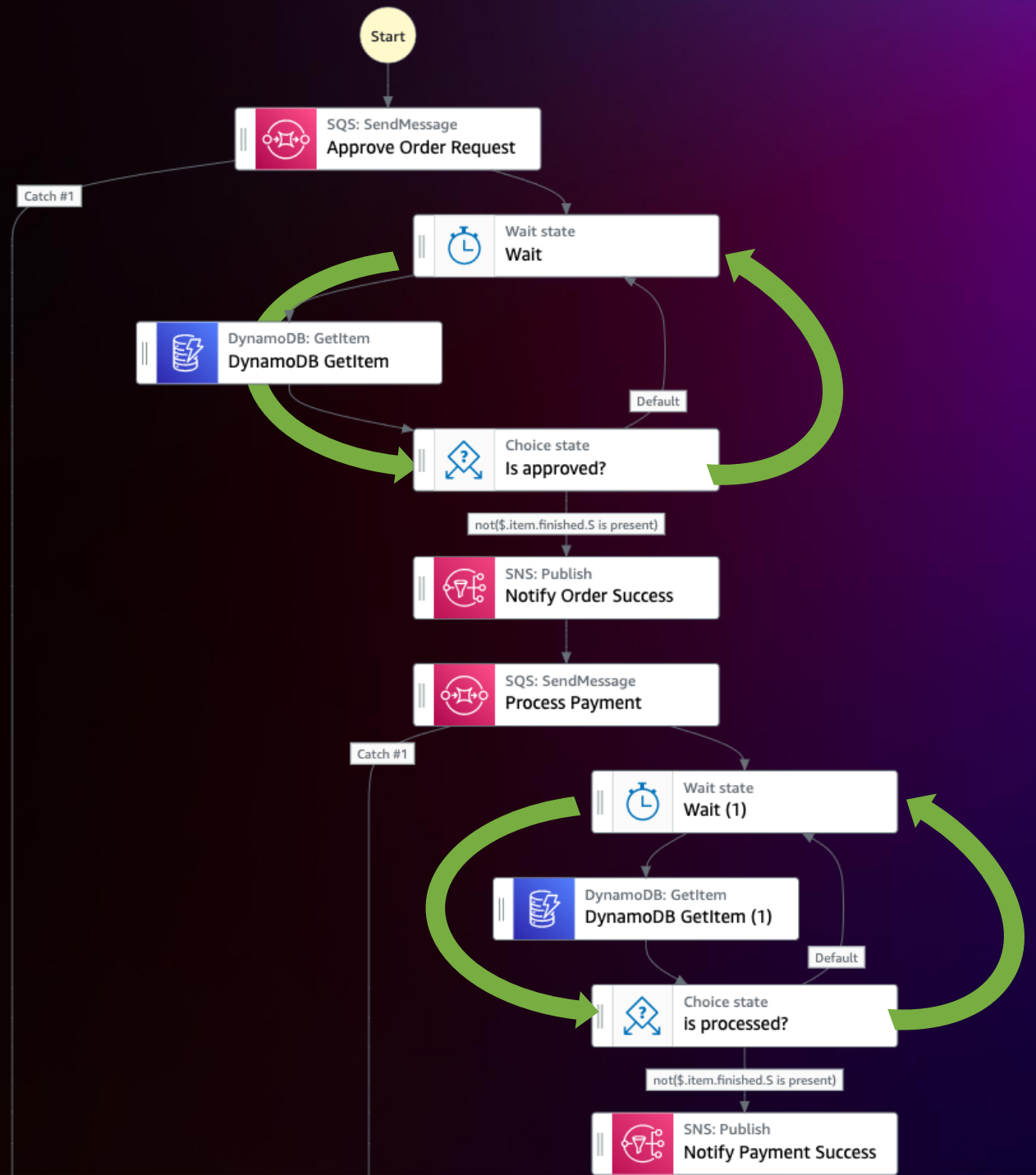
PERFORM BASIC DATA PROCESSING OPERATIONS WITHOUT USING A TASK

Function	Description
States.Array	Creates array with a given input
States.ArrayContains	Checks if given array contains your match
States.UUID	Generates universally unique identifier (v4 UUID)
States.Base64Encode	Encodes data based on MIME Base64 encoding scheme
States.Base64Decode	Decodes data based on MIME Base64 decoding scheme
States.Hash	Generates hash value of given input (MD5, SHA-1, SHA-256, SHA-384, SHA-512)

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

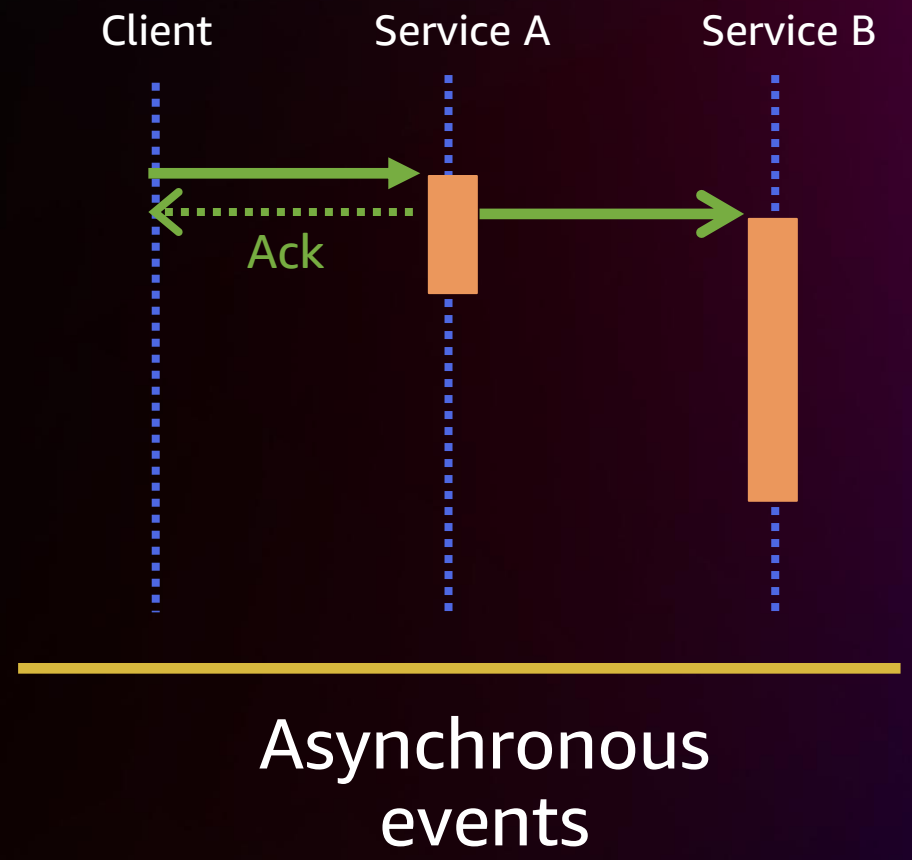
Reducing state transitions and duration with the callback pattern



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

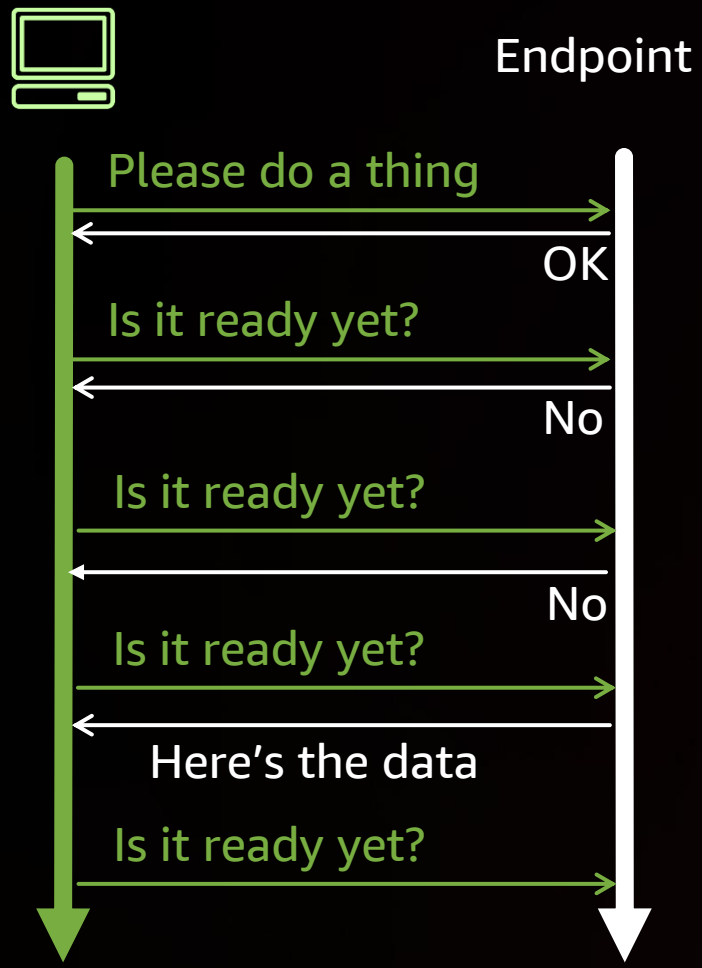
Handling response values and state for asynchronous requests

No return path to provide further information beyond the initial acknowledgement



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Tracking an inflight request – Polling



- Initial request returns a tracking identifier
- Create a second API endpoint for the frontend to check the status of the request, referencing the tracking ID
- Use DynamoDB or another data store to track the state of the request
- Simple mechanism to implement
- Can create many empty calls
- Delay between availability and front end notification

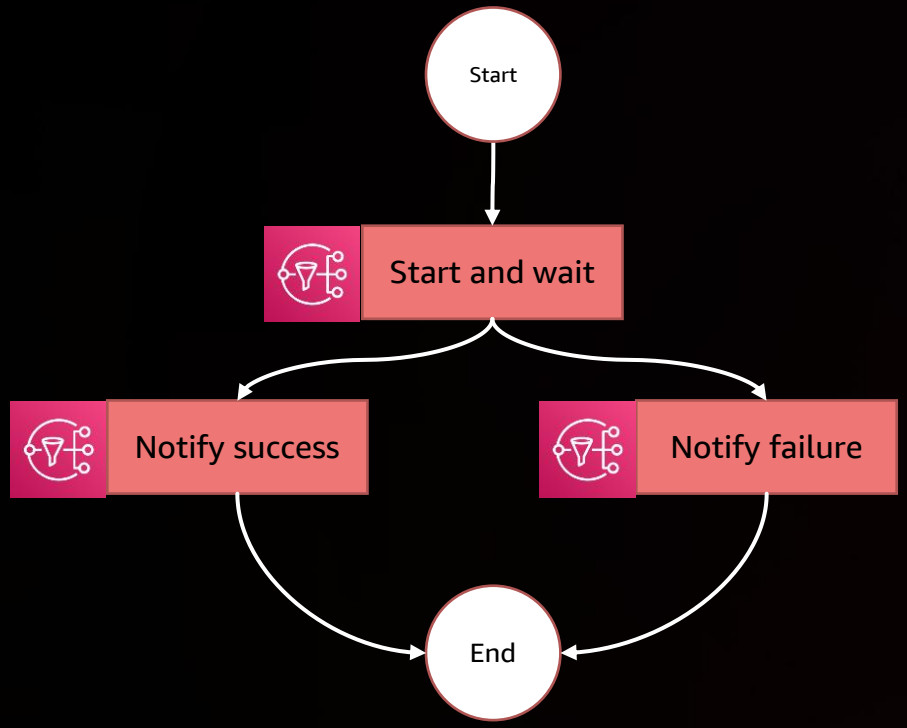
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Tracking an inflight request – Callback



- A bidirectional connection between the initiating service and the downstream service
- Your downstream services can send data back to the client directly
- Closer to real time
- Reduces the number of messages between the client and backend system
- Often more complex to implement

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

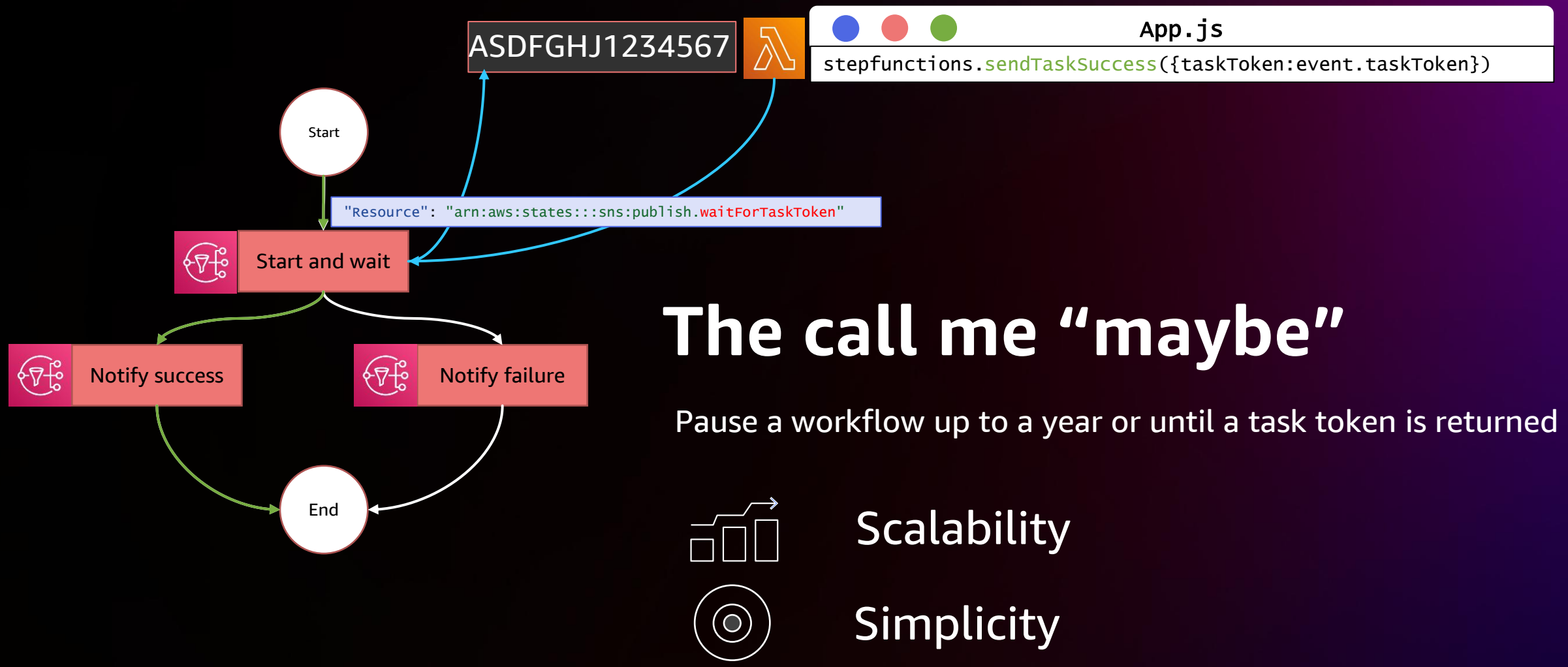


The call me “maybe”

Pause a workflow up to a year or until a task token is returned

- Scalability
- Simplicity

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

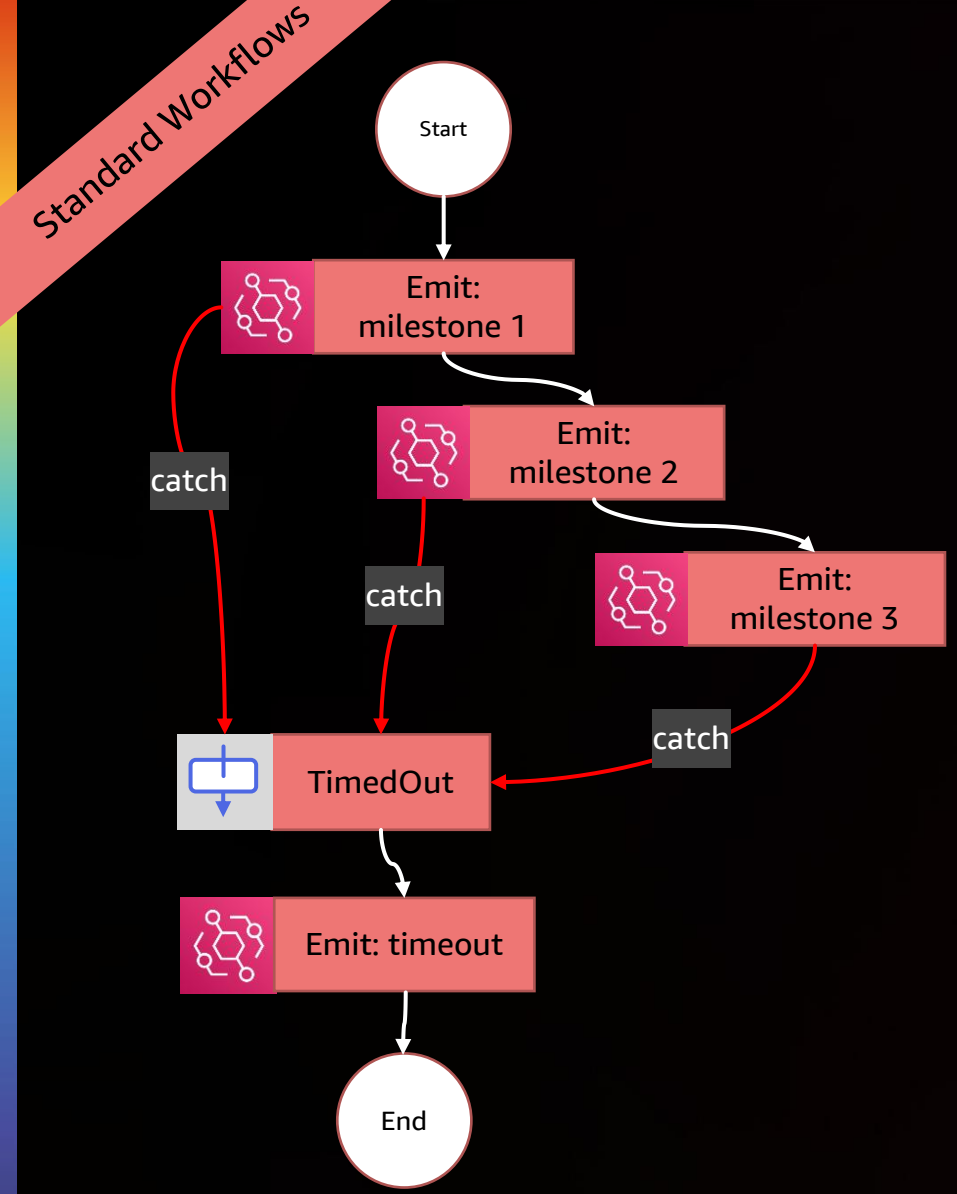


The call me “maybe”

Pause a workflow up to a year or until a task token is returned

- Scalability
- Simplicity

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



The “emit and wait”

Emit milestone events that invoke external decoupled microservices and emit error events if they don’t respond

Reliability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Standard Workflows



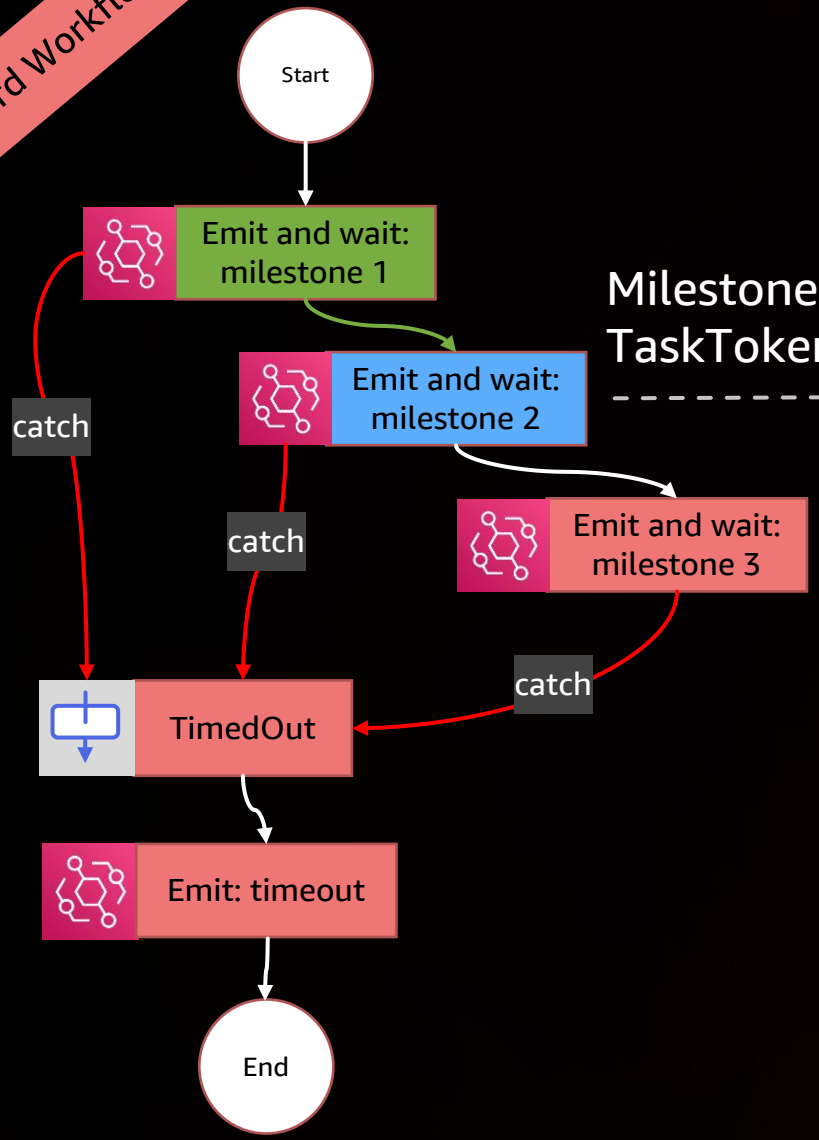
App.js

```
stepfunctions.sendTaskSuccess({
            “taskToken” :event.taskToken,
            “ExecutionArn” : event.arn
          })
```

Reliability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Standard Workflows



Milestone 2
TaskToken: 12345678



Set a time limit to wait for task token

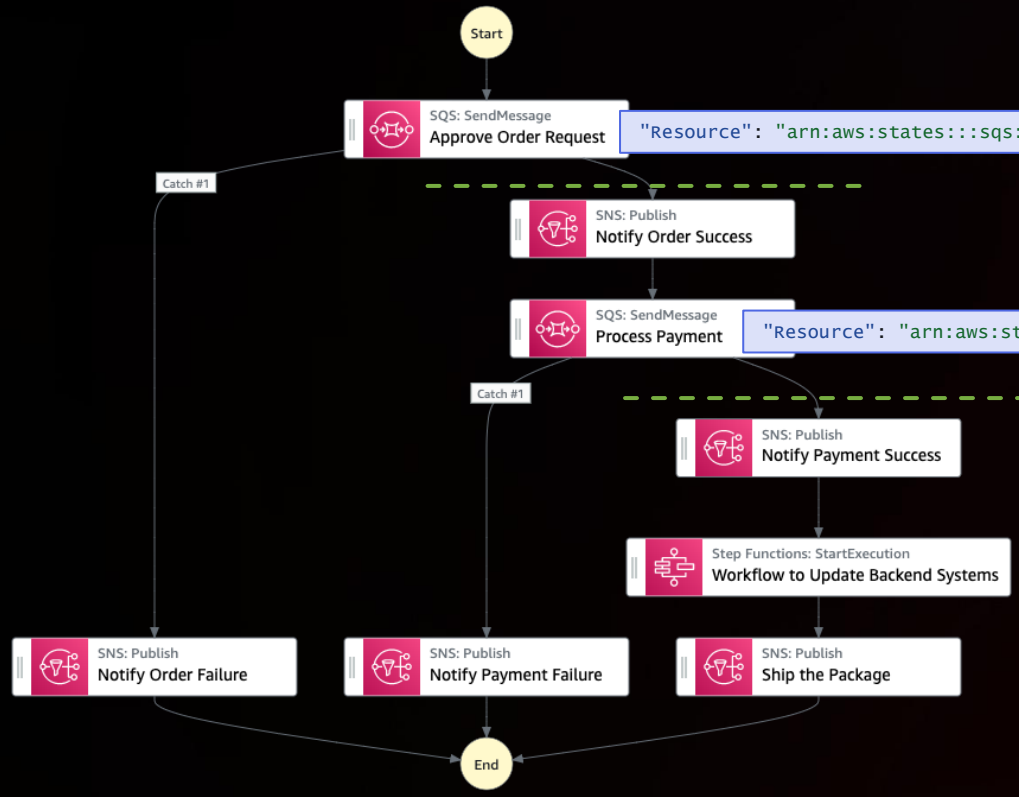
"HeartbeatSeconds": 20



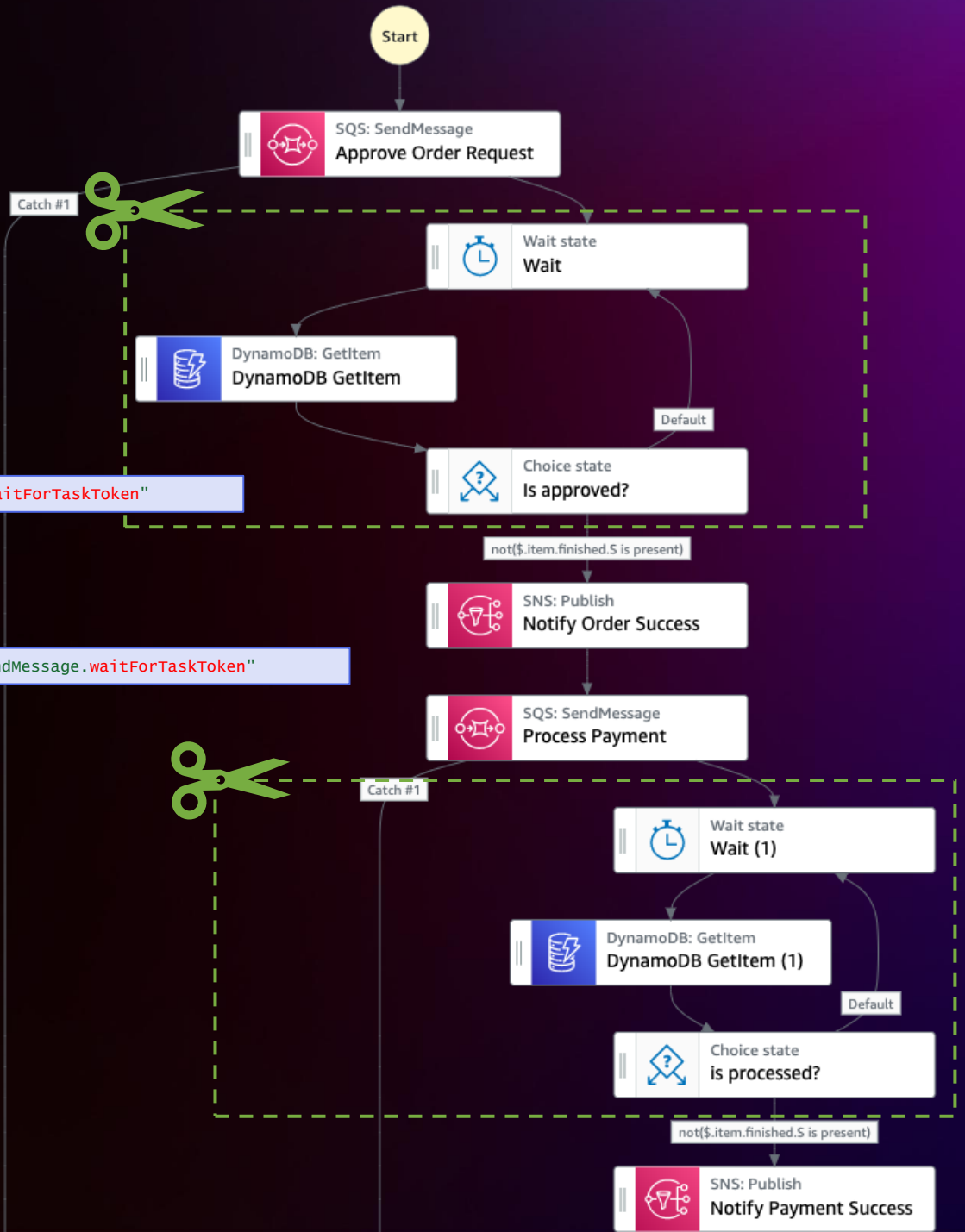
Reliability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Remove polling with call-back pattern



Standard Workflow



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cost per thousand executions

EXPRESS WORKFLOWS CHARGE BASED ON NUMBER OF REQUESTS AND DURATION

Total cost for calback optimized nested workflow

Cost of parent WF — Number of state transitions

(8*1000) * $0.000025 = $0.20

+

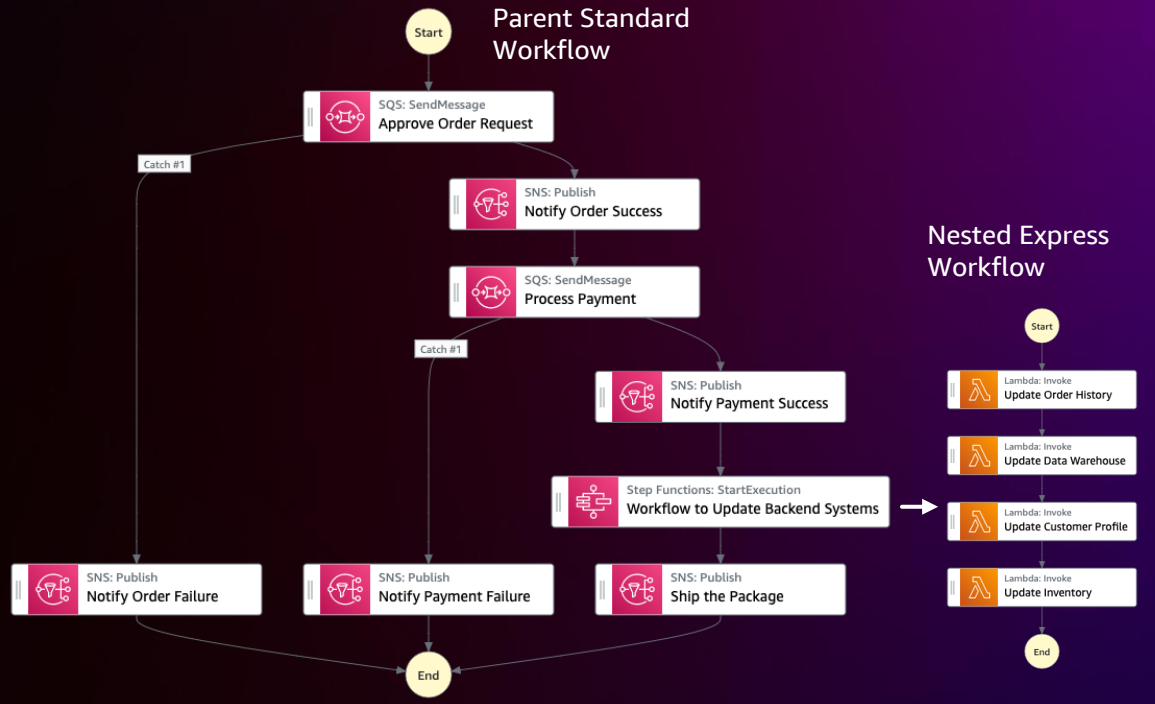
Cost of nested WF — Average duration — Price per 100 ms ($) for first 1,000 GB-hours

Duration cost = (1300 Ms /100) * $ 0.0000001042 = $0.000001

Execution cost = $0.000001

($0.000001 + $0.000001) x 1000 = $0.0002

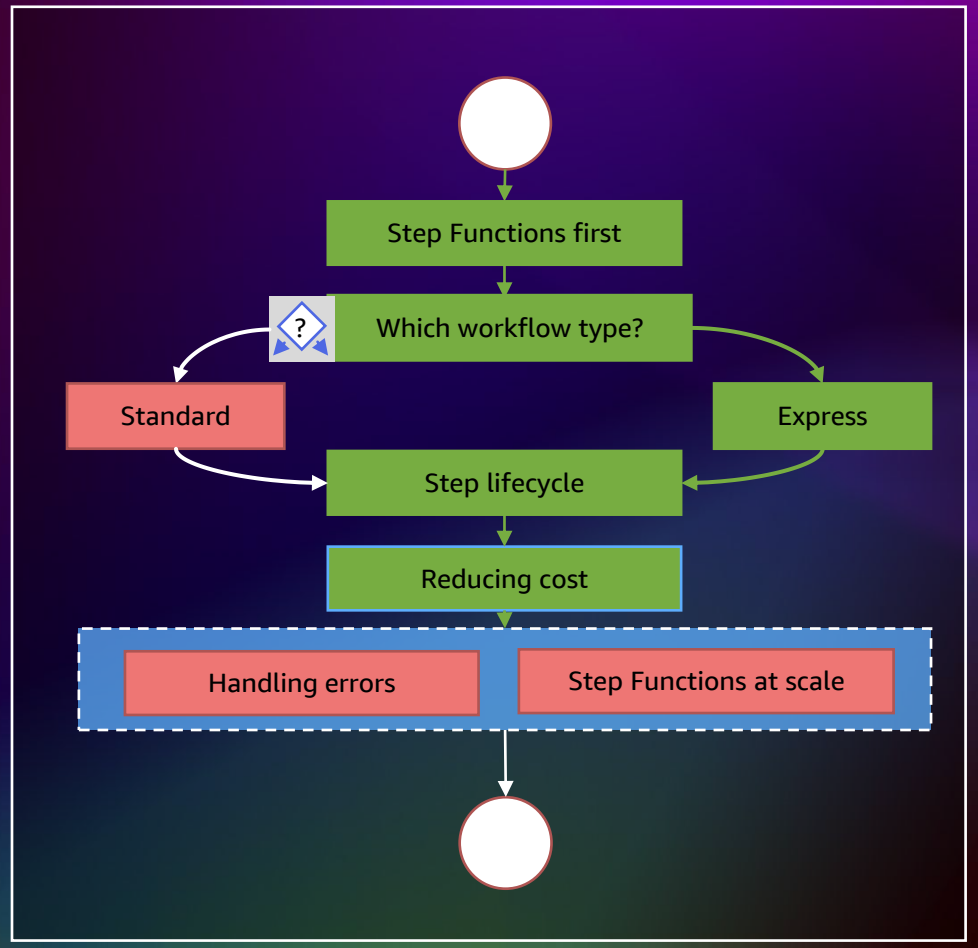
Total. = $0.20 per 1000 executions



This new workflow combination runs 1,000 times, costing a total cost of 20 cents

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

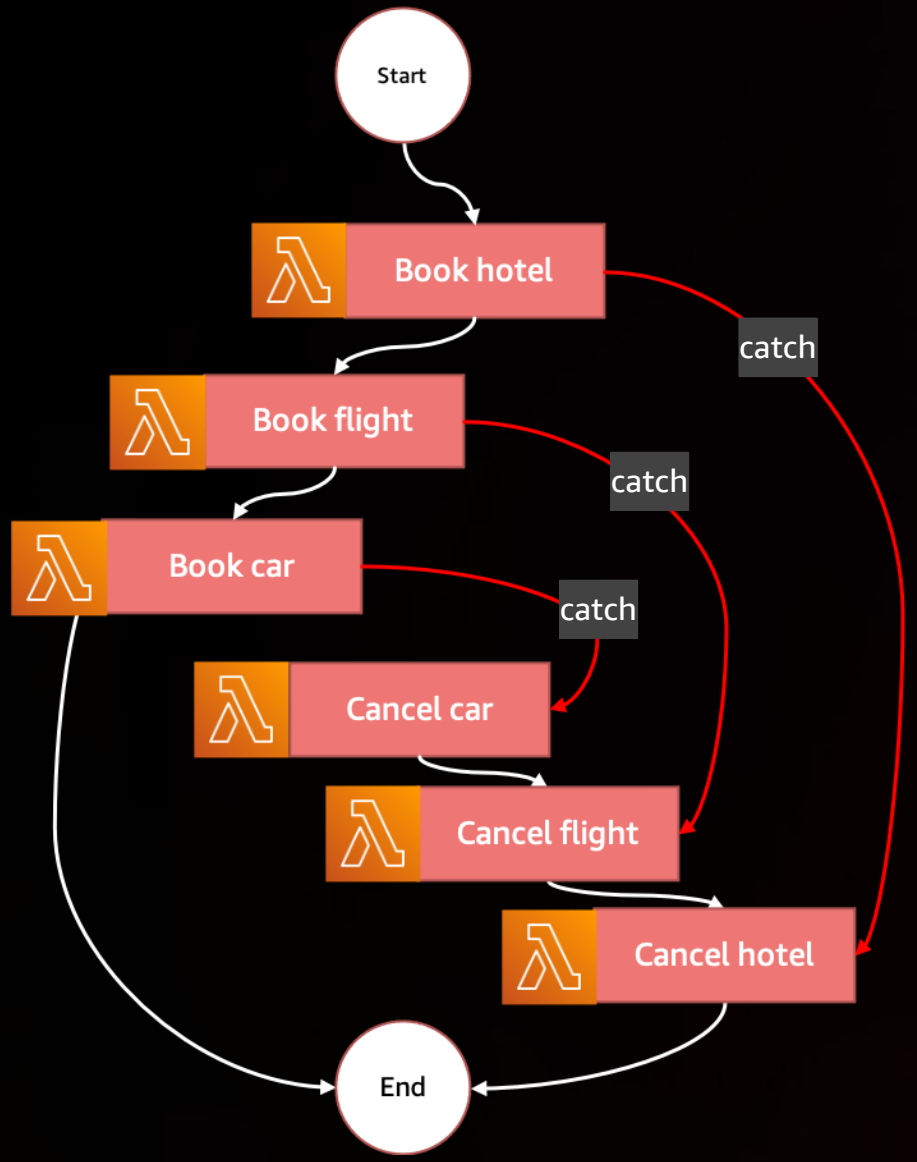
Managing failures



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

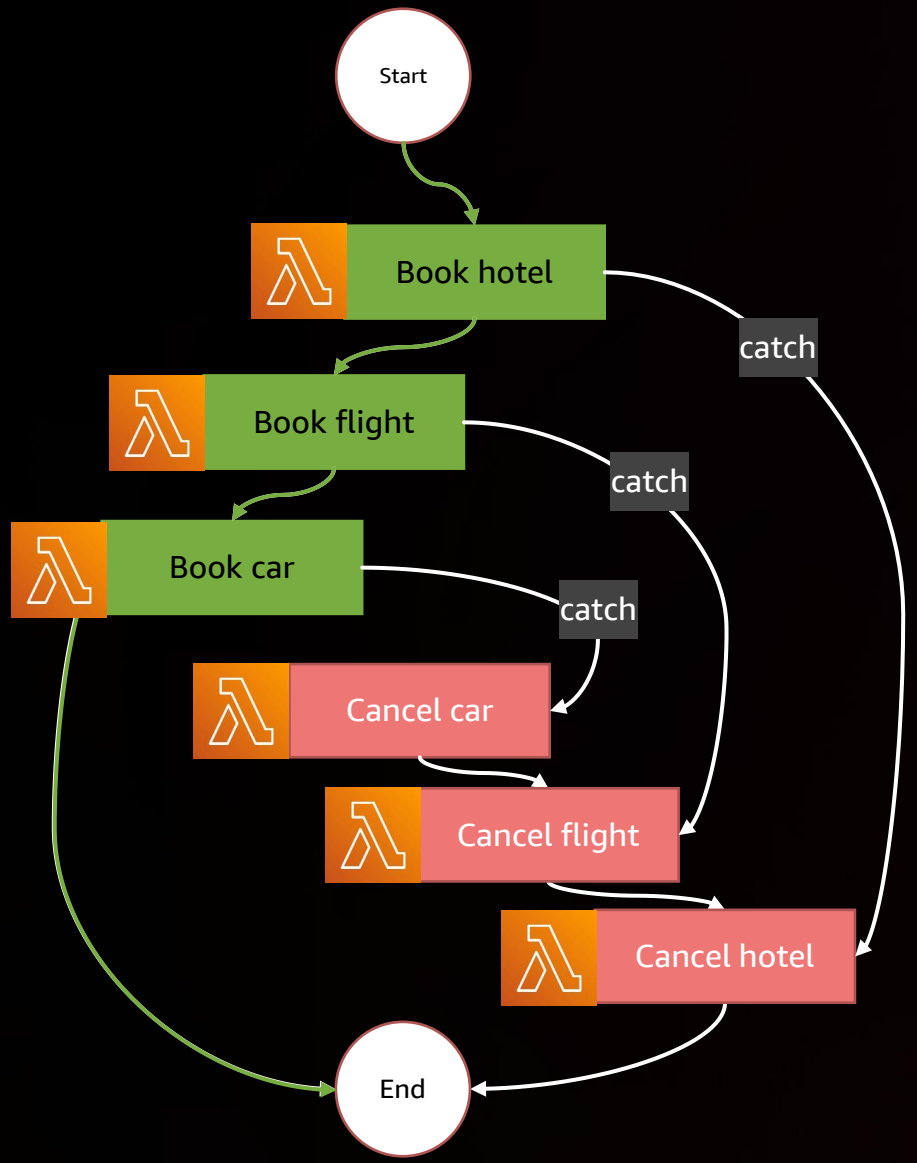


The “saga” continues

Use inherent error handling to roll back sequential system failures for long-running transactions

Resilience

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



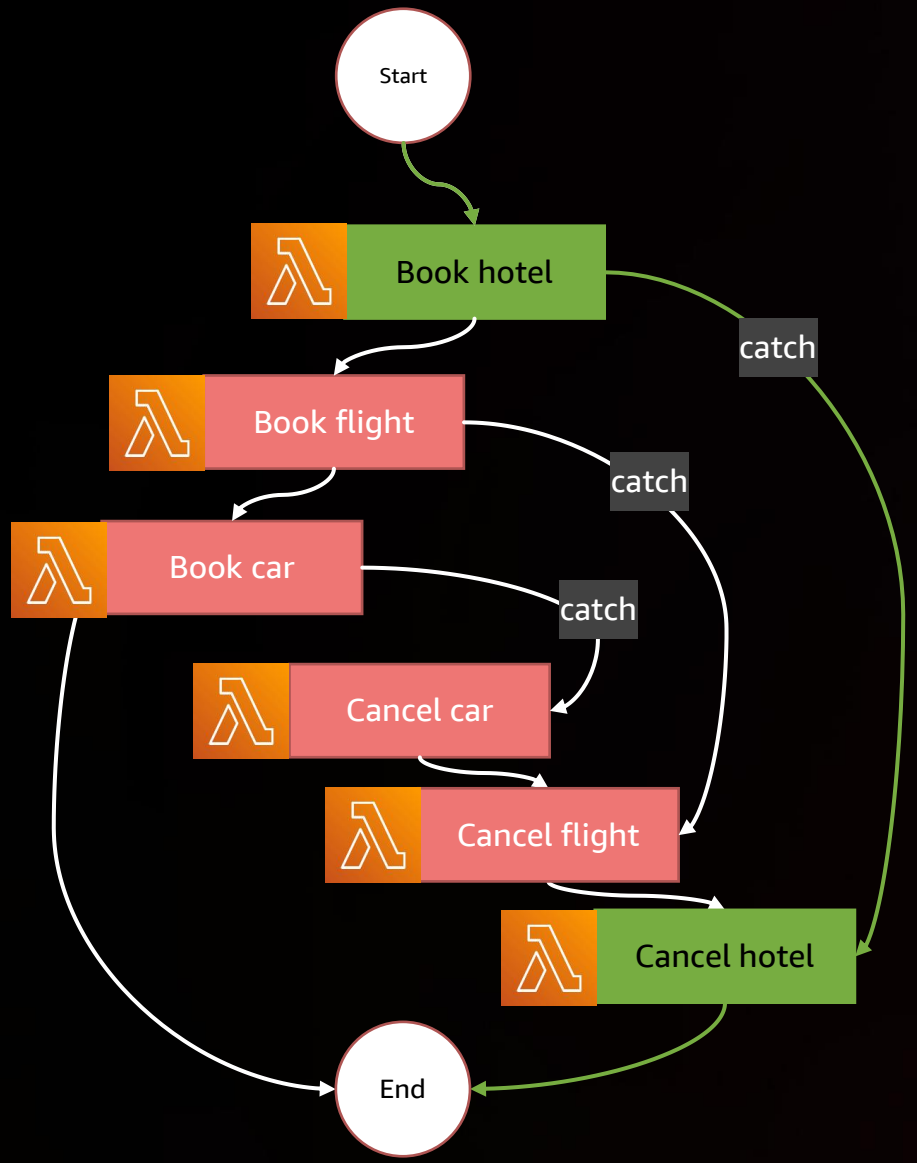
Success case

The “saga” continues

Use inherent error handling to roll back sequential system failures for long-running transactions

Resilience

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



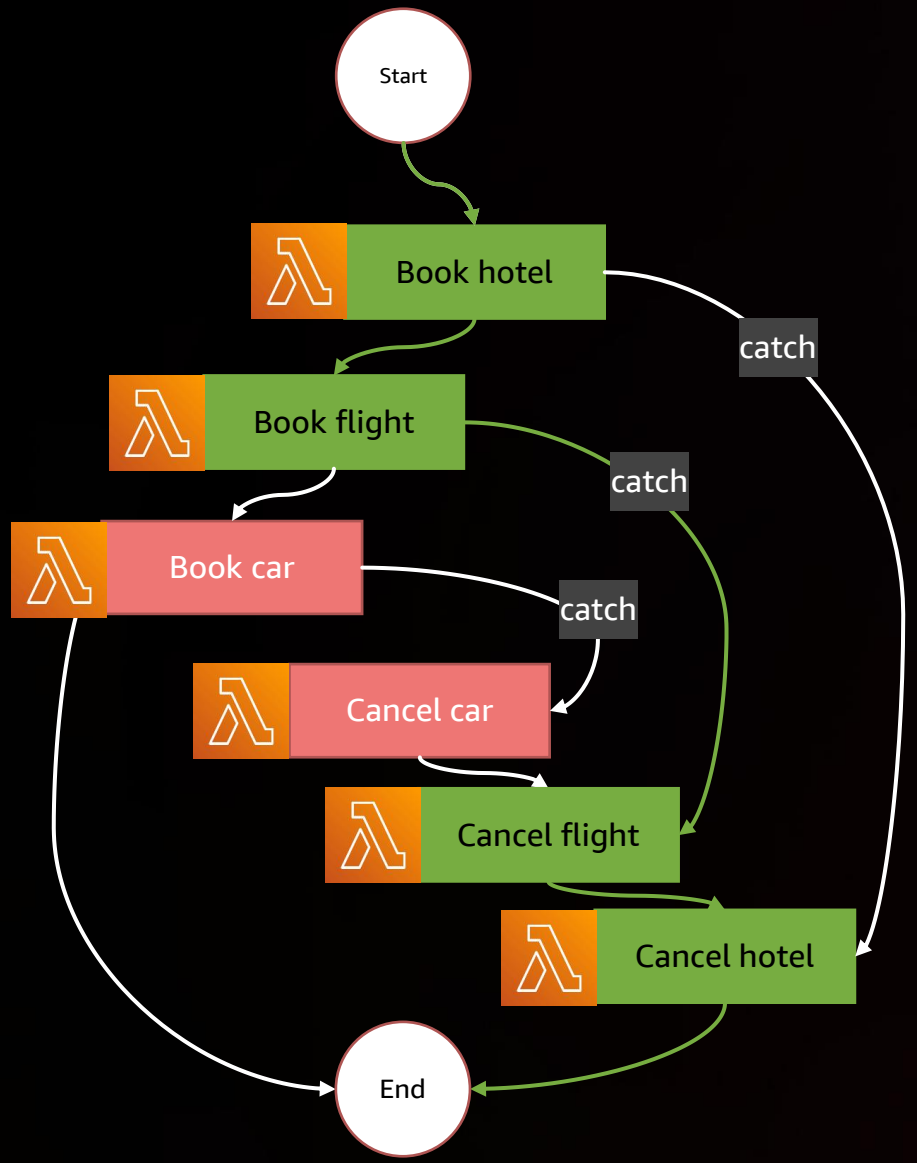
Failure cases

The “saga” continues

Use inherent error handling to roll back sequential system failures for long-running transactions

Resilience

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



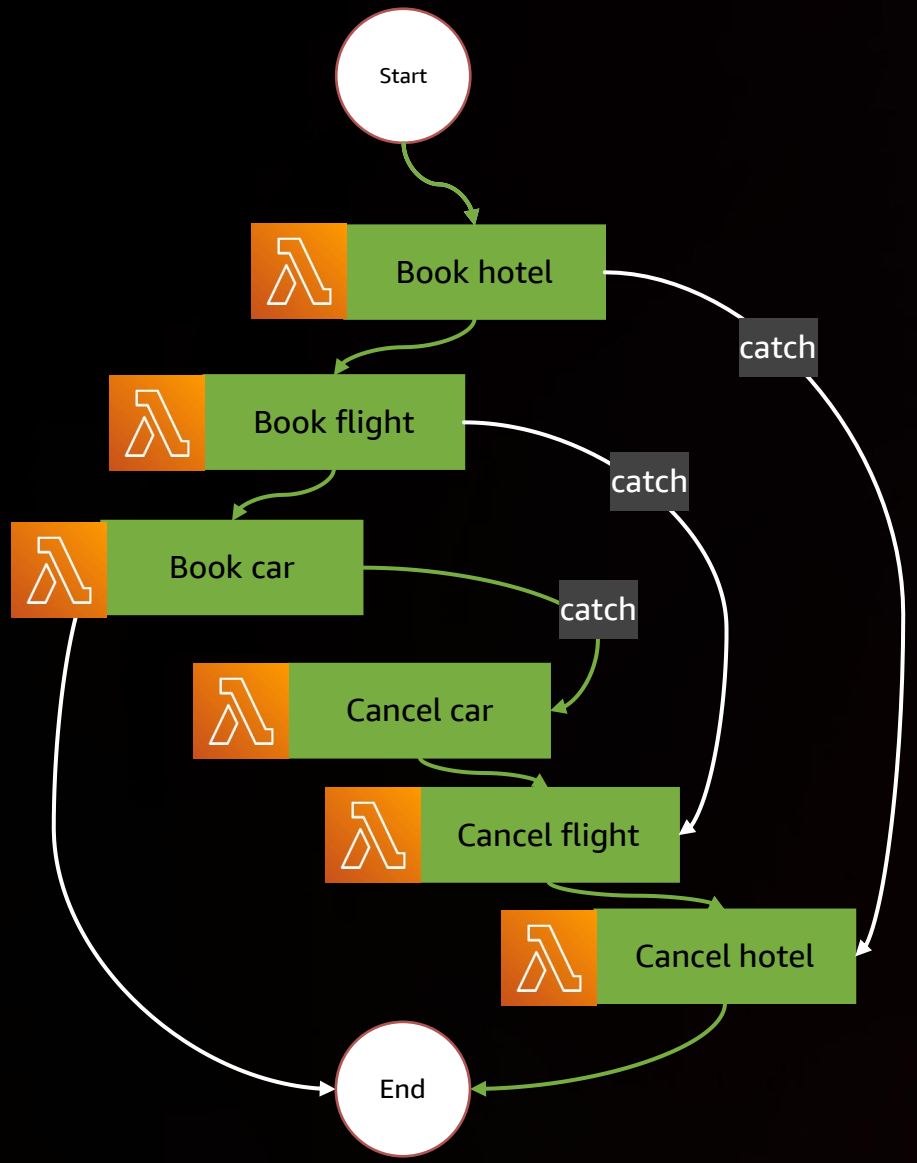
Failure cases

The “saga” continues

Use inherent error handling to roll back sequential system failures for long-running transactions

Resilience

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



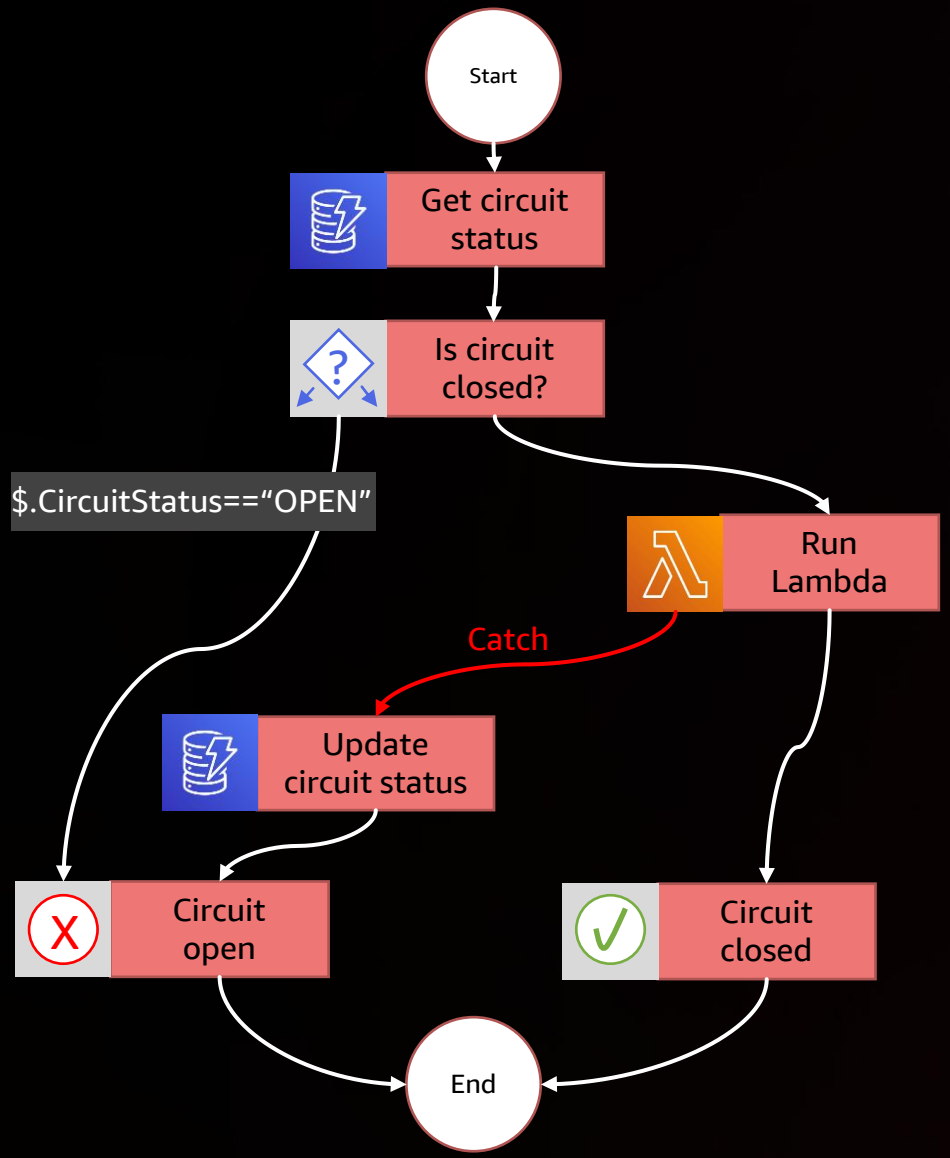
Failure cases

The “saga” continues

Use inherent error handling to roll back sequential system failures for long-running transactions

Resilience

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

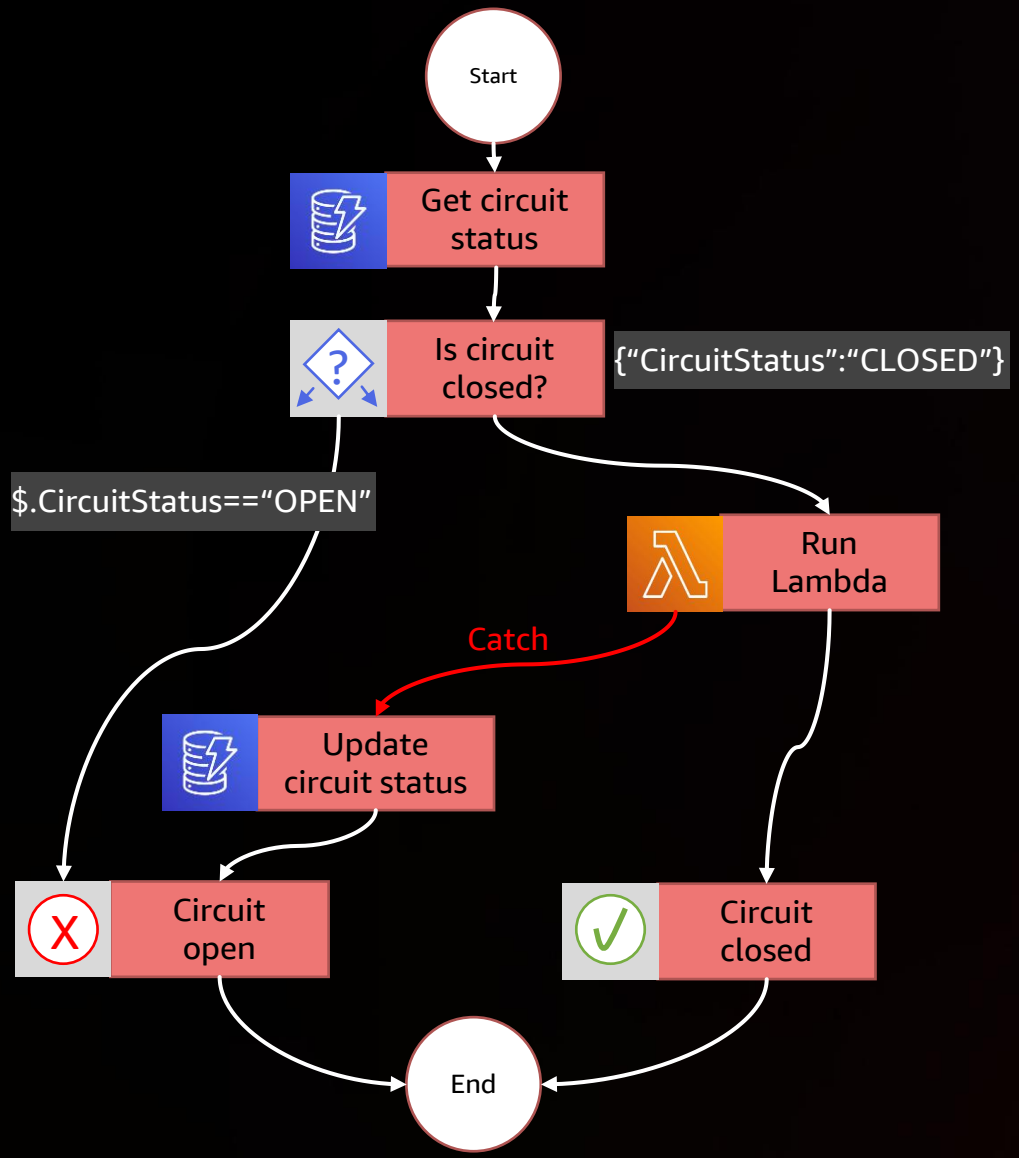


The "circuit breaker"

Prevent a caller service from retrying another callee service call that has previously caused repeated timeouts or failures

Reliability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



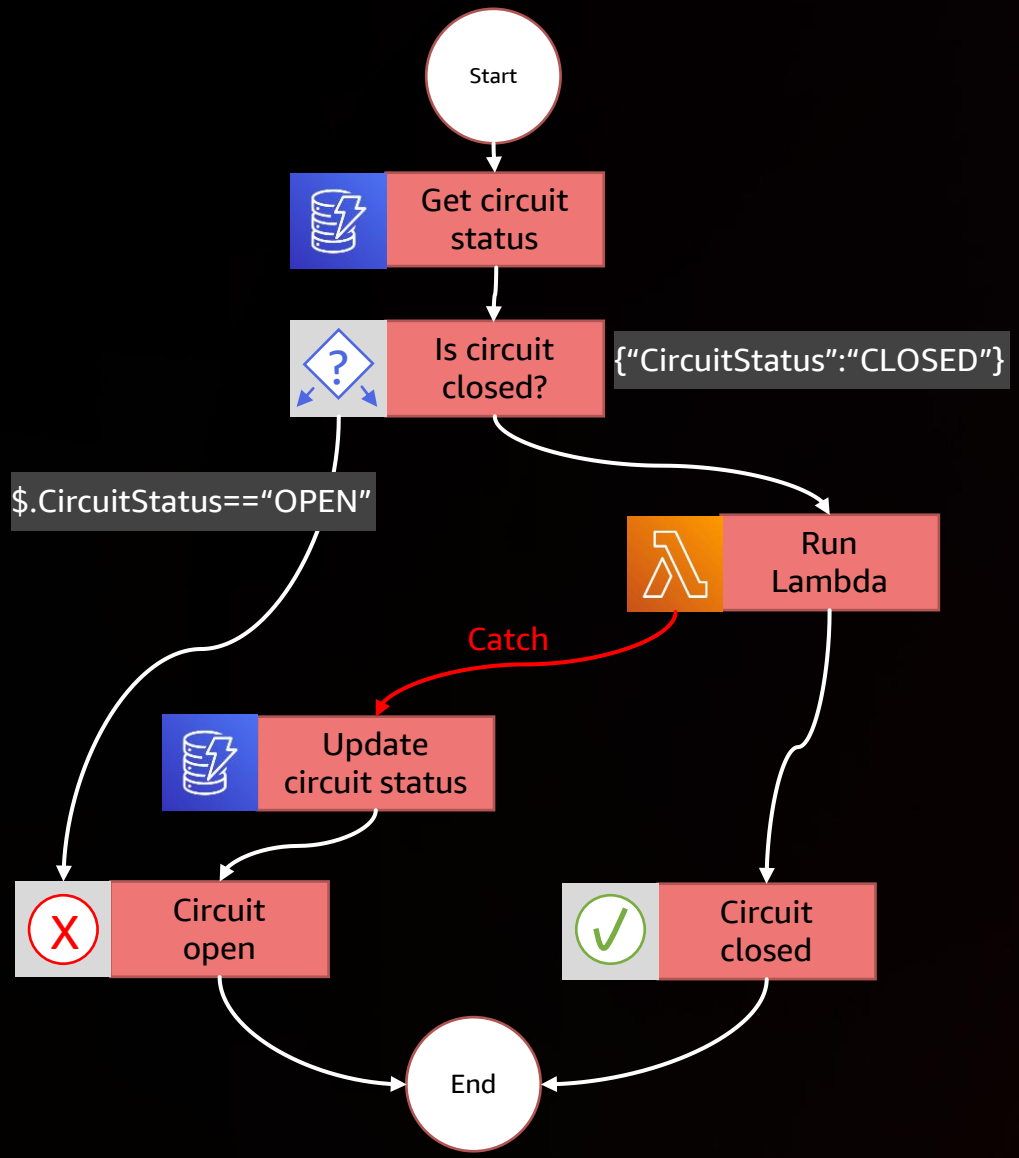
Success cases

The “circuit breaker”

Prevent a caller service from retrying another callee service call that has previously caused repeated timeouts or failures

Reliability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



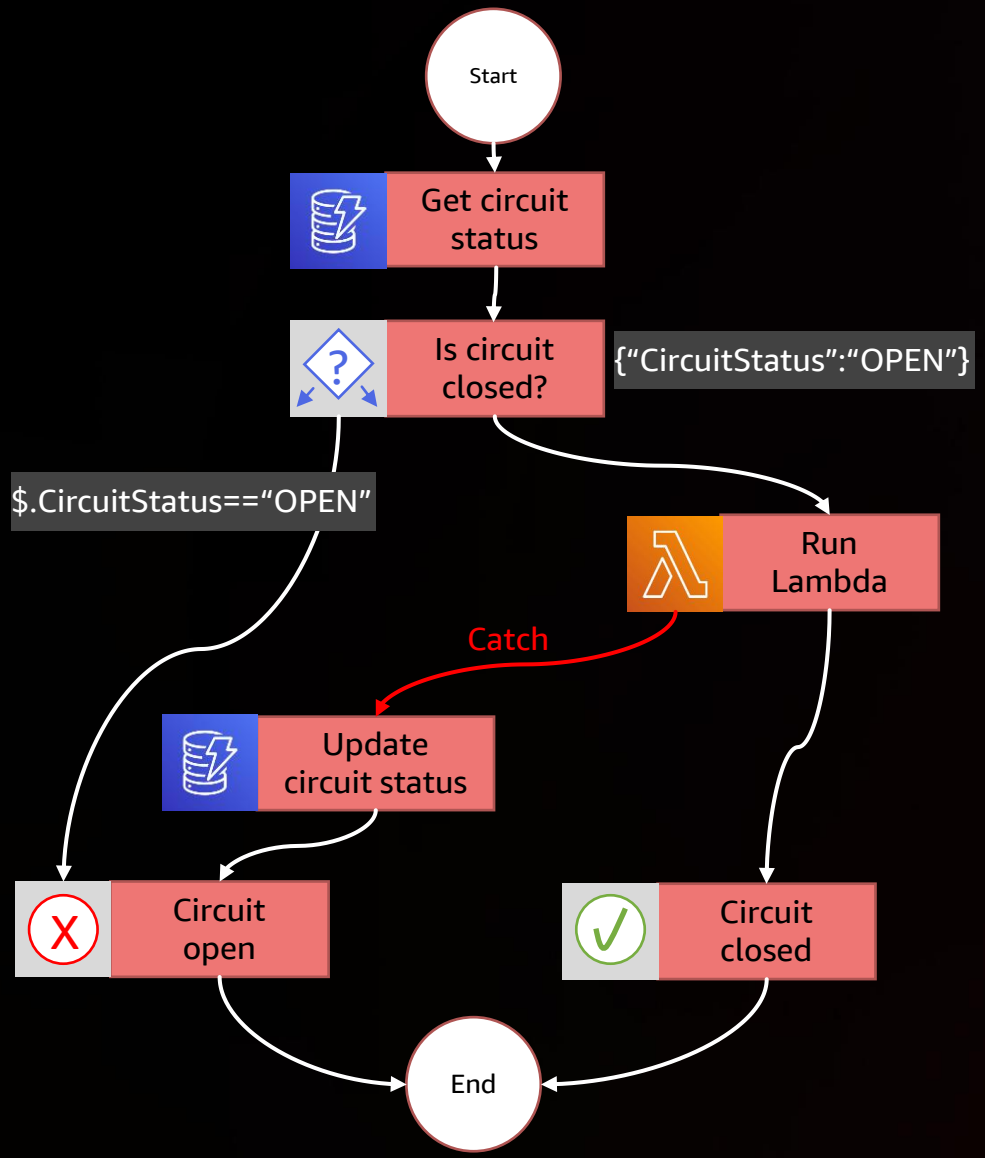
Success cases

The “circuit breaker”

Prevent a caller service from retrying another callee service call that has previously caused repeated timeouts or failures

Reliability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



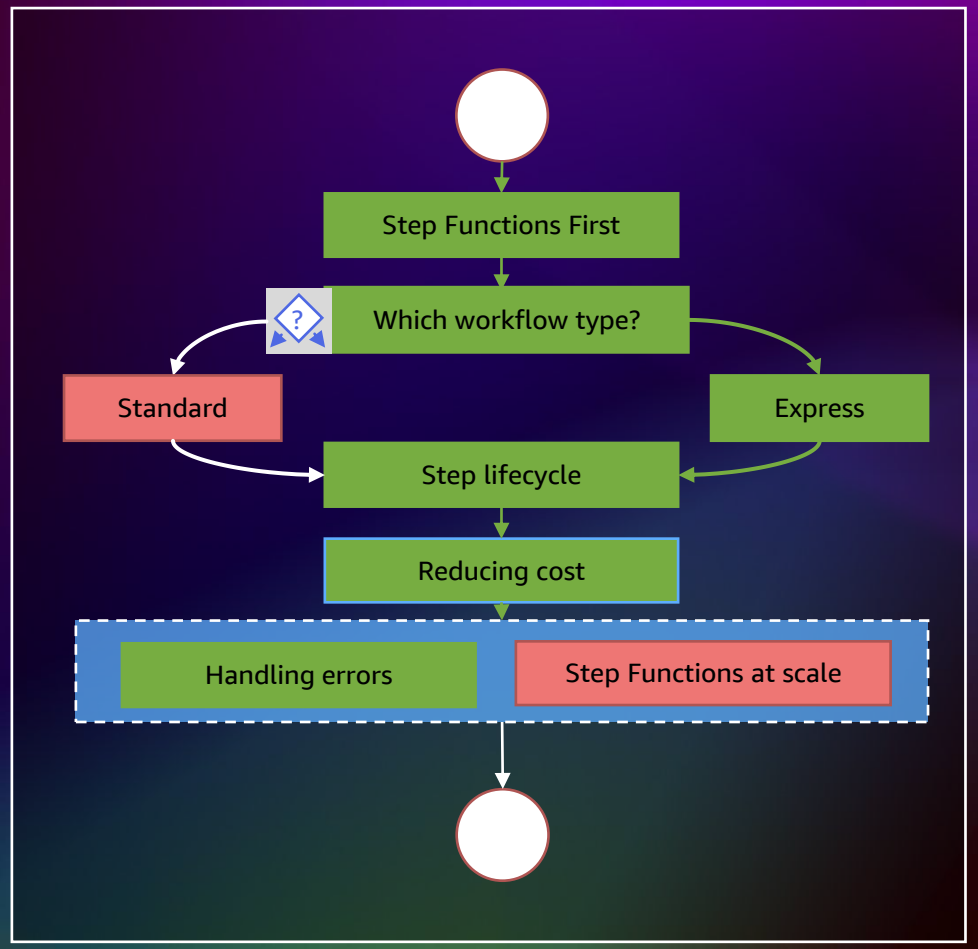
The “circuit breaker”

Prevent a caller service from retrying another callee service call that has previously caused repeated timeouts or failures

Reliability

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step Functions at scale



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step Functions at scale

Parallelism

Doing lots of things at once

Same input, different steps

VS

Concurrency

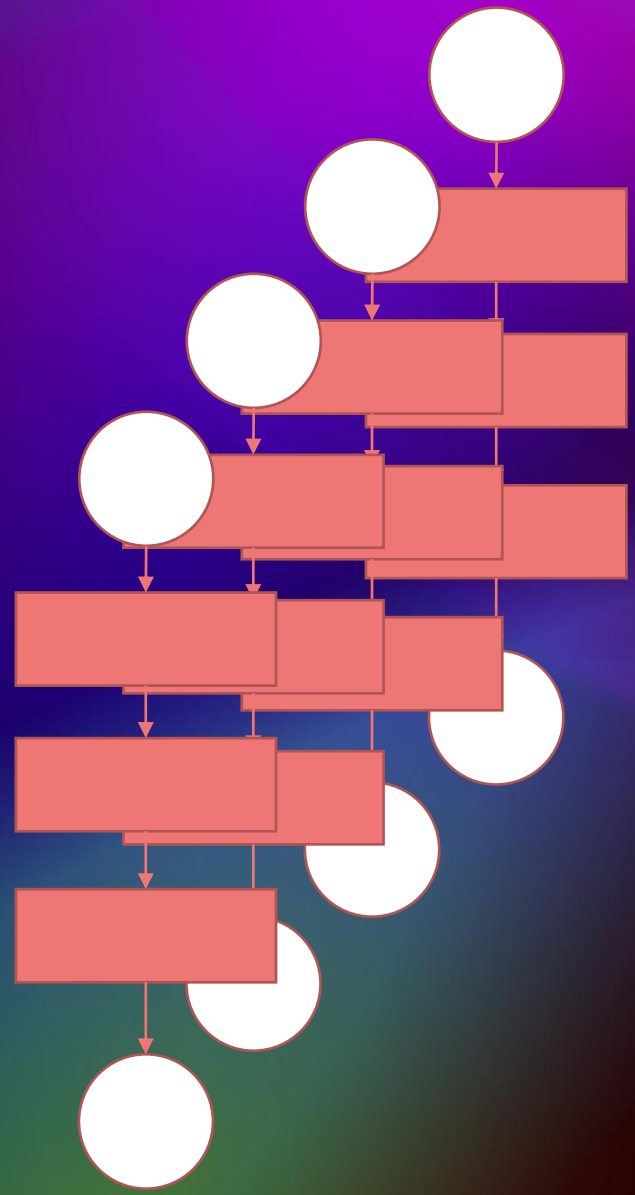
Dealing with lots of things at once

Same steps, different input

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Parallel invocations, lots of independent workflows doing their own thing



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

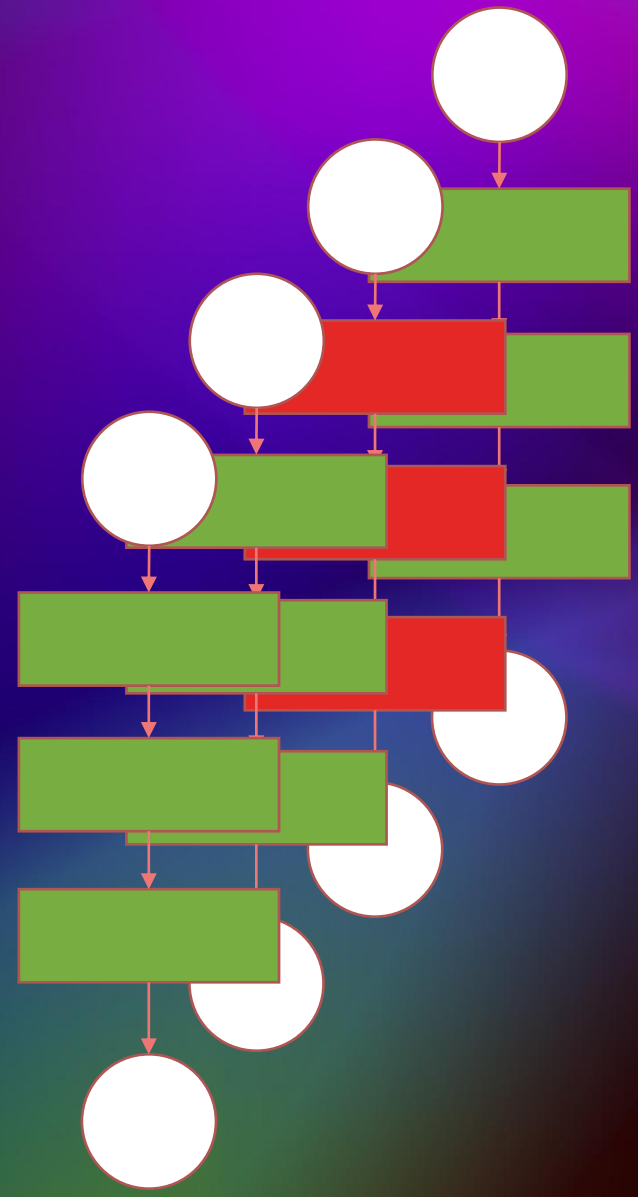
Same ASL definition executing at the same time, with different data

Each process is independent

Failure in one process doesn’t impact another

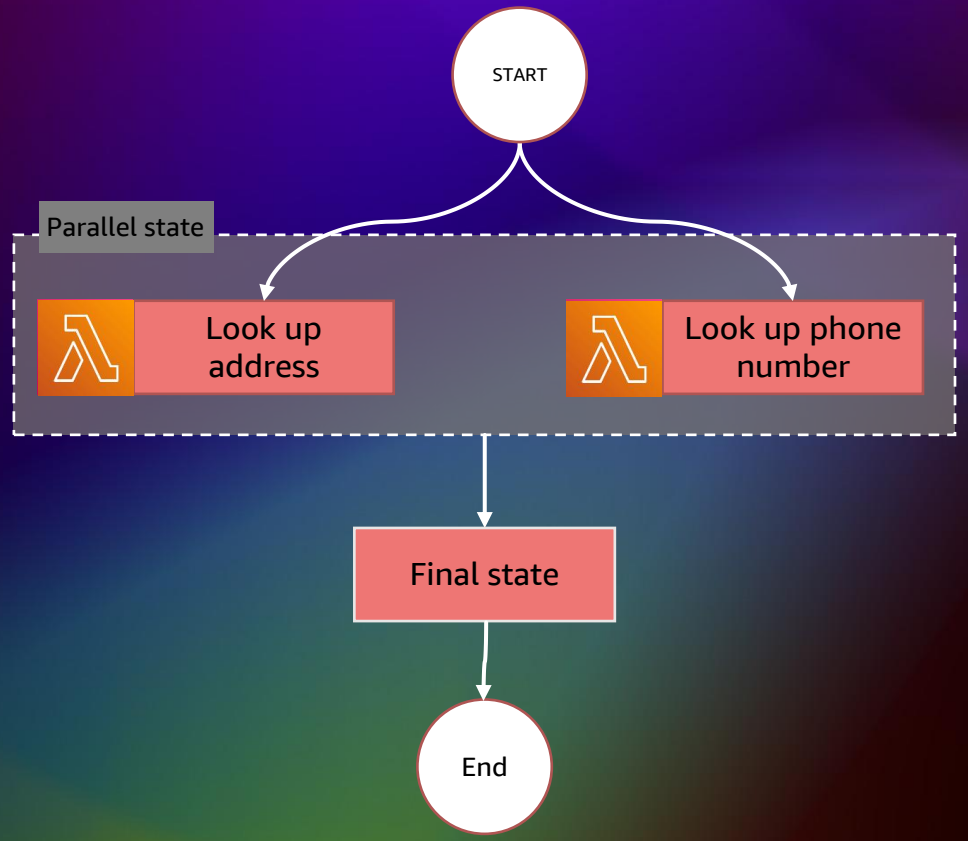
E.g., remediation workflow for non-compliant resource

Limitation: account quote (mostly soft limits)



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

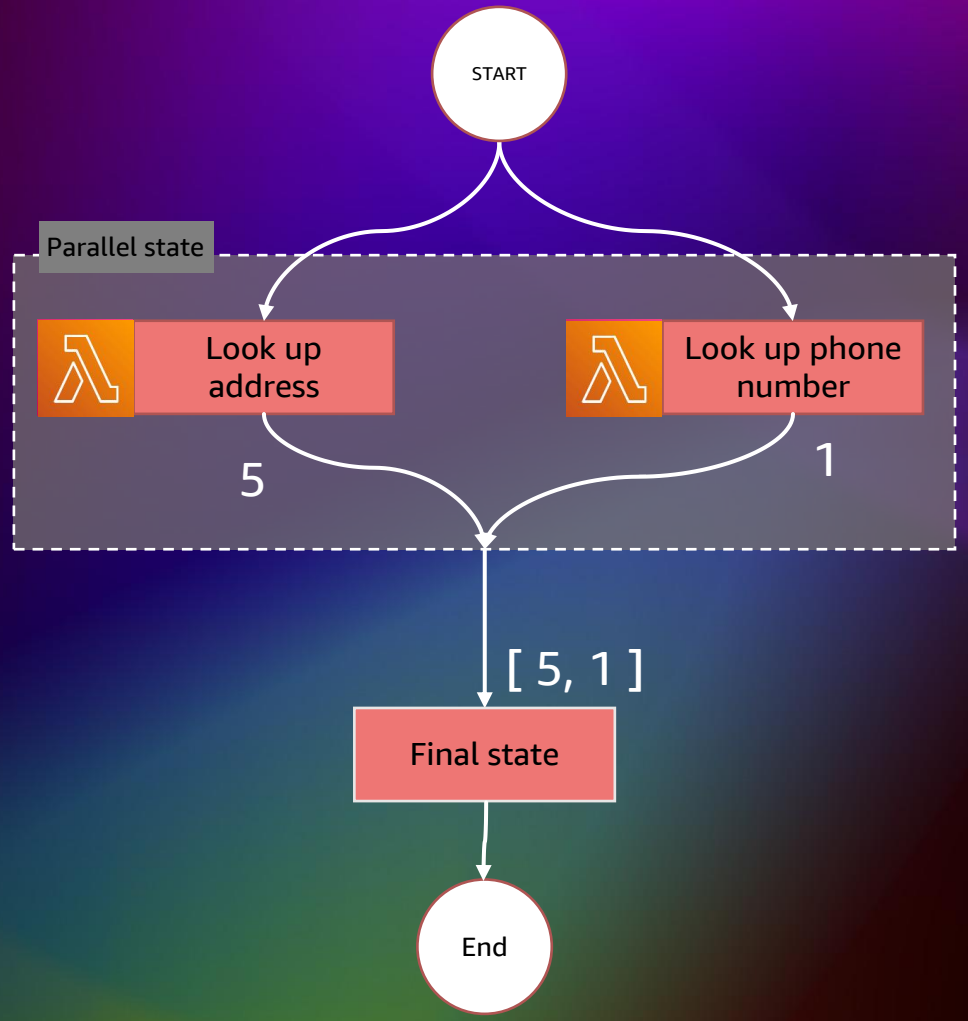
Parallel state: execute a fixed number of branches in parallel



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Executes multiple branches of steps using the same input

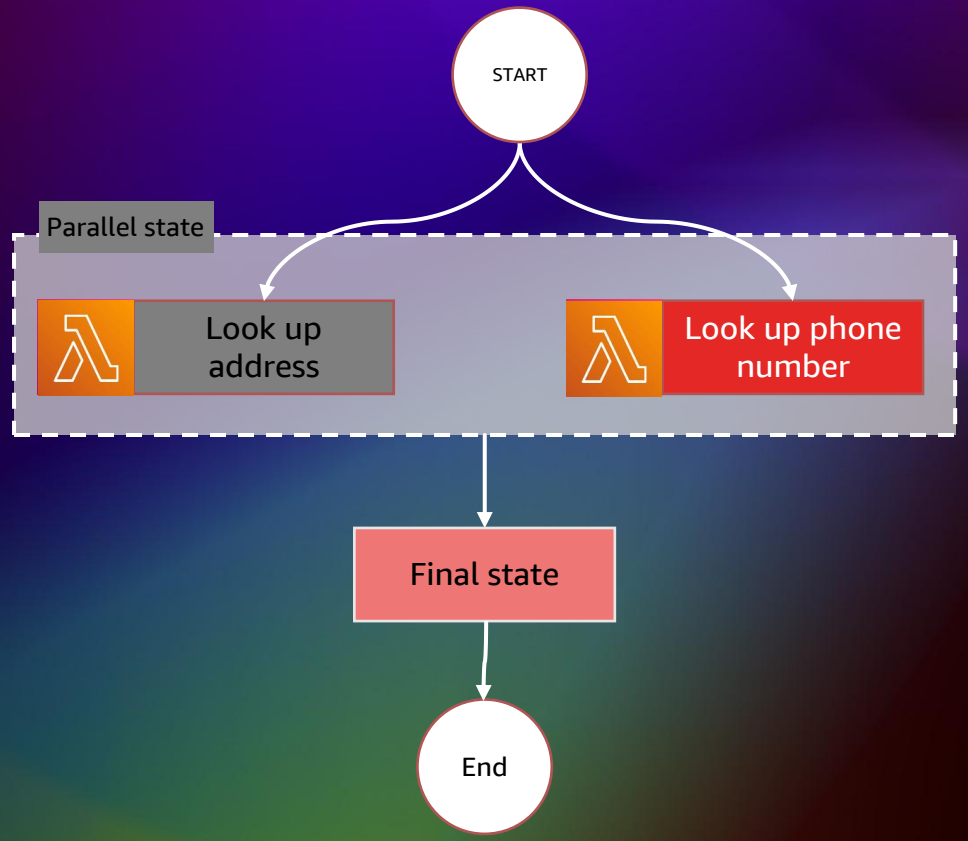
Map state output is an array with one element for each branch result



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Wait until all branches terminate (reach a terminal state) before processing the next state

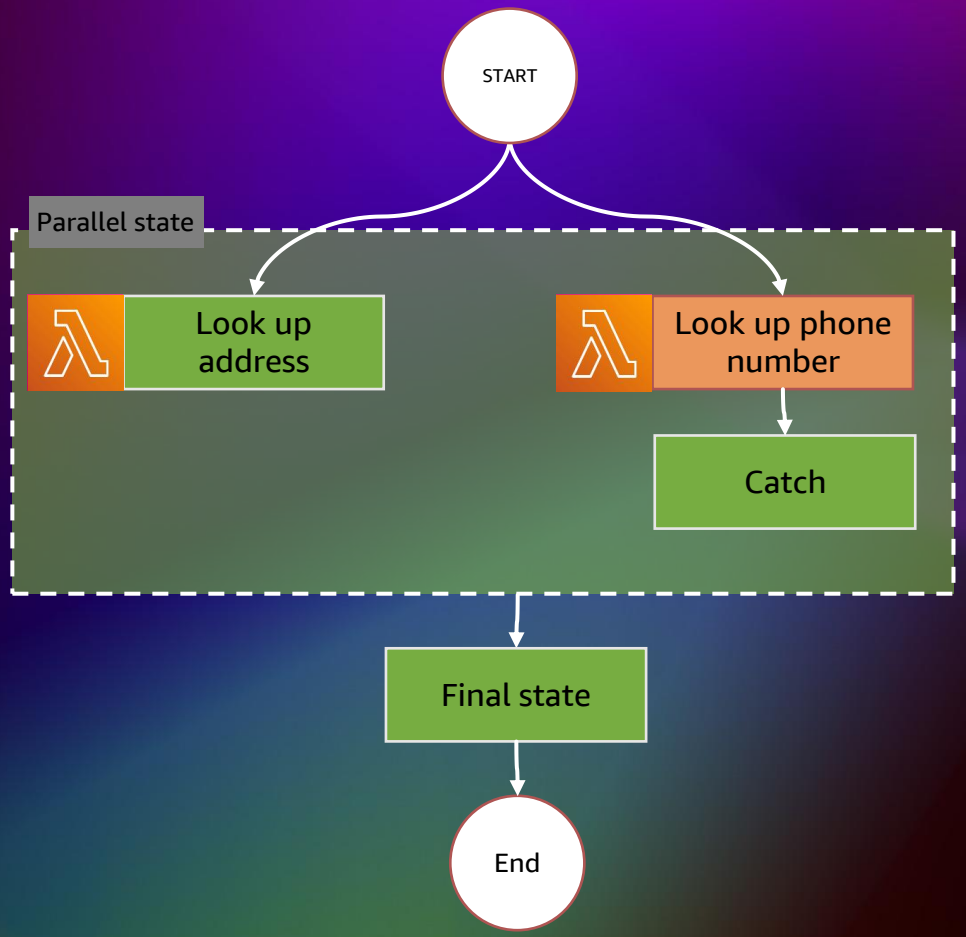
If any branch fails because of an unhandled error or by transitioning to a fail state, the entire parallel state is considered to have failed and all its branches are stopped



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

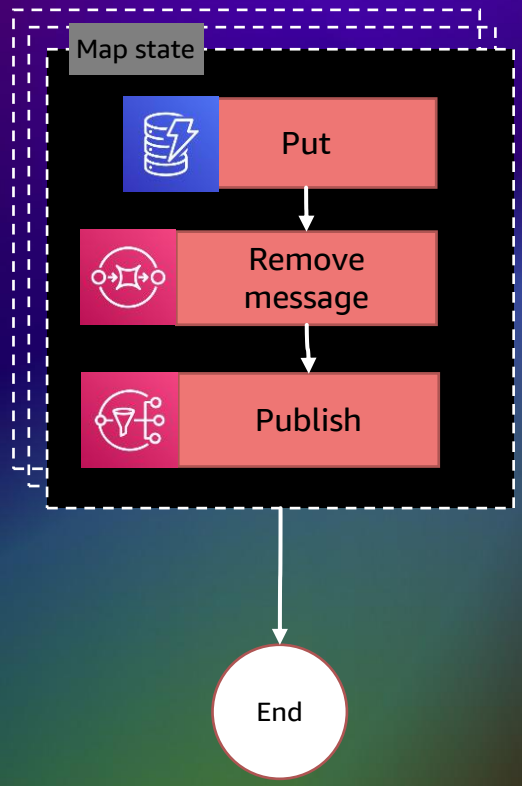
Catch errors to allow other branches to complete

All execution results are aggregated into the final state



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Dynamic parallelism: A sub-workflow to accomplish work faster by using more resources at the same time

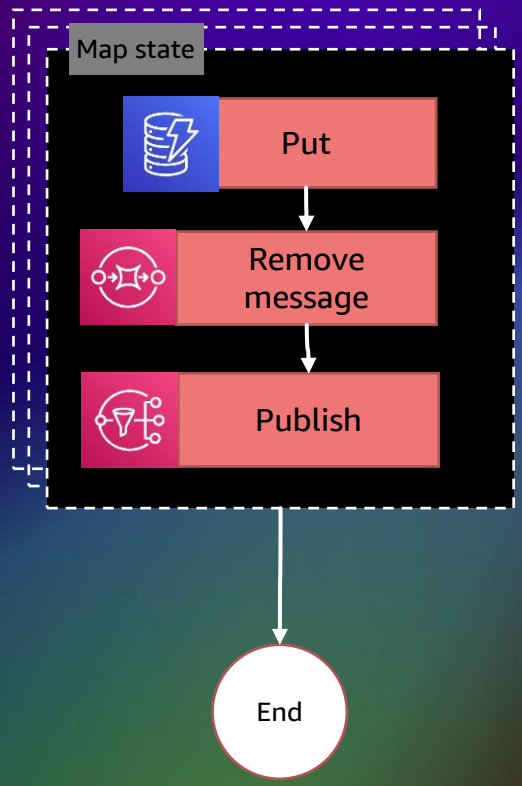


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

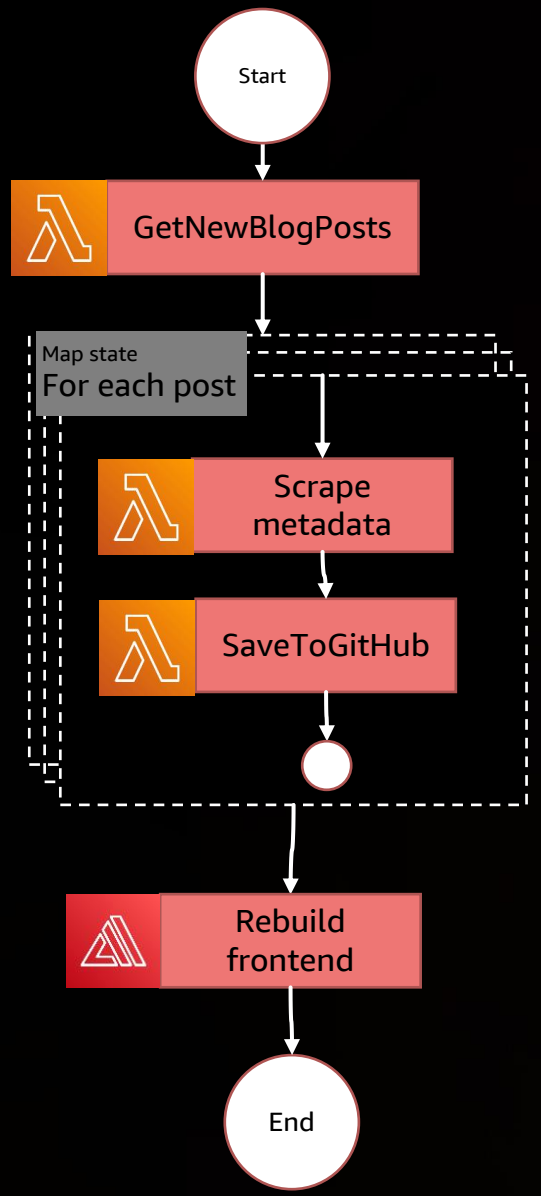
Execute the same steps for multiple entries of an input array

MaxConcurrency controls the number of iterations that will run in parallel

Set to 1 for single-threaded integrations;
set to 0 for maximum parallelism



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

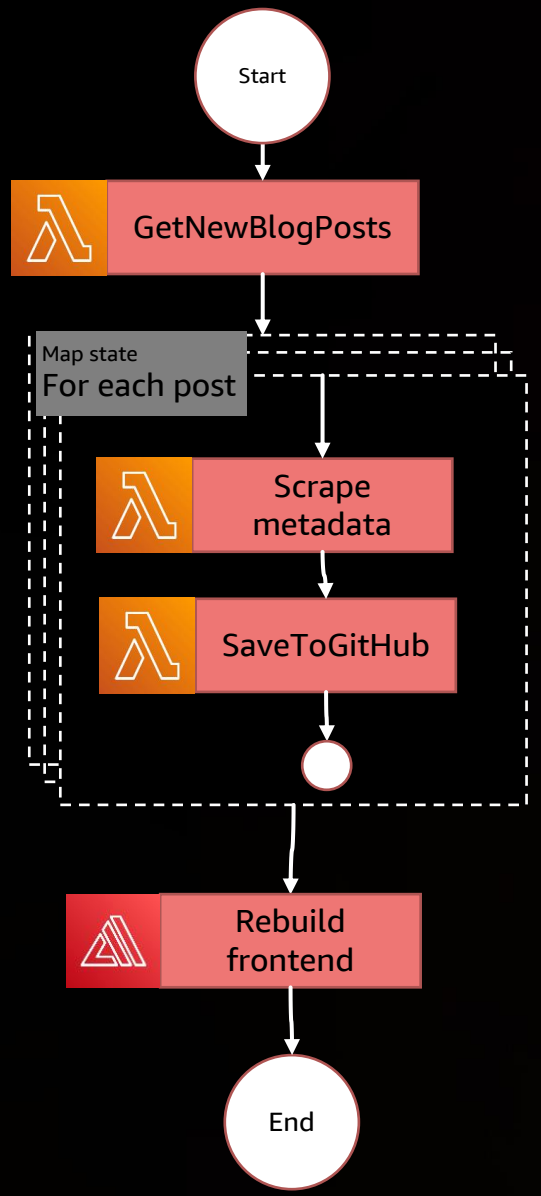


The big “fan” (fan out)

Deliver a workload to multiple destinations

Scalability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Scans RSS feed at https://aws.amazon.com/blogs/compute/feed
Returns an array of new compute blog post URLs

```
{
"body": [
    "https://aws.amazon.com/blogs/compute/integrating-amazon-memorydb-for-redis-with-java-based-aws-lambda/",
    "https://aws.amazon.com/blogs/compute/lifting-and-shifting-a-web-application-to-aws-serverless-part-2/",
    "https://aws.amazon.com/blogs/compute/lifting-and-shifting-a-web-application-to-aws-serverless-part-1" ,
    ……
  ]
}
```

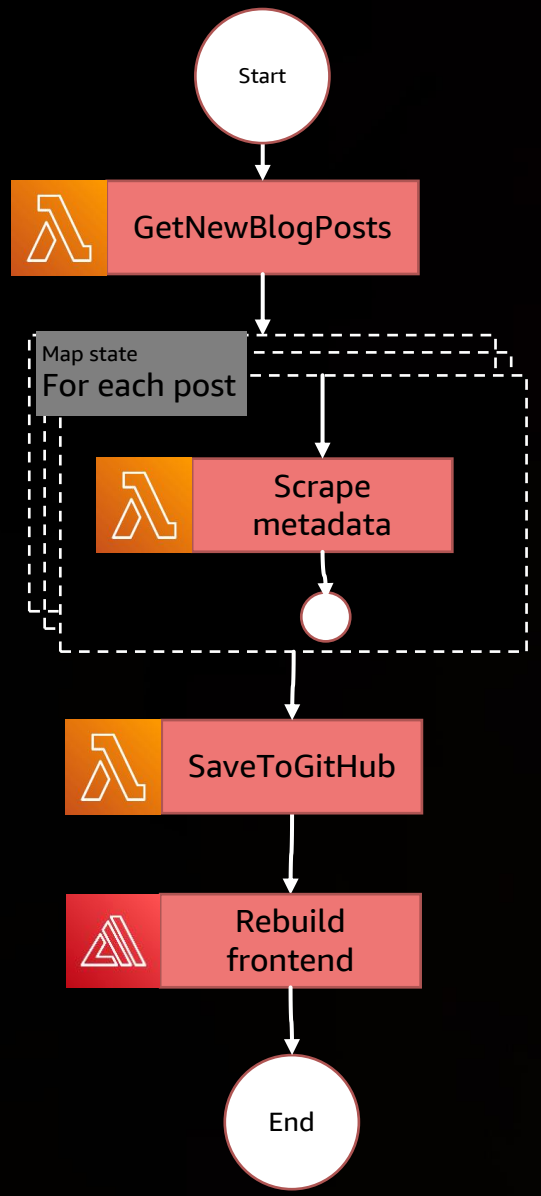
```
{
  "title":"Lifting and shifting a web application to AWS Serverless: Part 1",
  "Author":"Marcia Villalba",
  "desc":" Customers migrating to the cloud often want to get the benefits of serverless architecture.",
  "link":https://aws.amazon.com/blogs/compute/lifting-and-shifting-a-web-application-to-aws-serverless-part-1,
  "thumbnail":"/thumb1.png",
  "tags":["serverless","stepFunctions"]
  "date": "Sep 14, 2022",
}
```

out)

Deliver a payload to multiple destinations

Scalability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

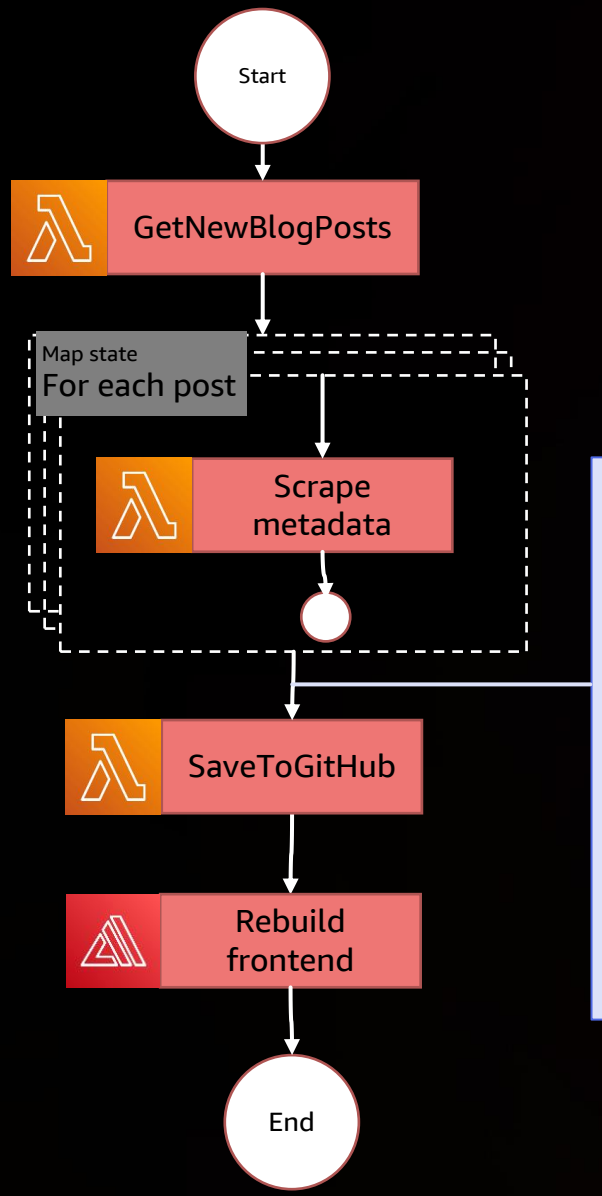


The scatter gather

Broadcasts a payload to multiple destinations (scatter) and then aggregates the responses back for the next steps (gather)

Scalability

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.




```
[
 {
  “title”:”Lifting and shifting a web application to AWS Serverless: Part 1”,
  “Author”:”Marcia Villalba”,
  “desc”:” Customers migrating to the cloud often want to get the benefits of serverless architecture.”,
  “link”:https://aws.amazon.com/blogs/compute/lifting-and-shifting-a-web-application-to-aws-serverless-part-1,
  “thumbnail”:”/thumb1.png”,
  “tags”:[”serverless”,”stepFunctions”]
  "date": "Sep 14, 2022",
 }
 ,
 {
  “title”:”Lifting and shifting a web application to AWS Serverless: Part 2”,
  “Author”:”Marcia Villalba”,
  “desc”:” Customers migrating to the cloud often want to get the benefits of serverless architecture part 2.”,
  “link”:https://aws.amazon.com/blogs/compute/lifting-and-shifting-a-web-application-to-aws-serverless-part-2,
  “thumbnail”:”/thumb1.png”,
  “tags”:[”serverless”,”stepFunctions”]
  "date": "Sep 16, 2022",
 }
]
```

ons (scatter)
or the next

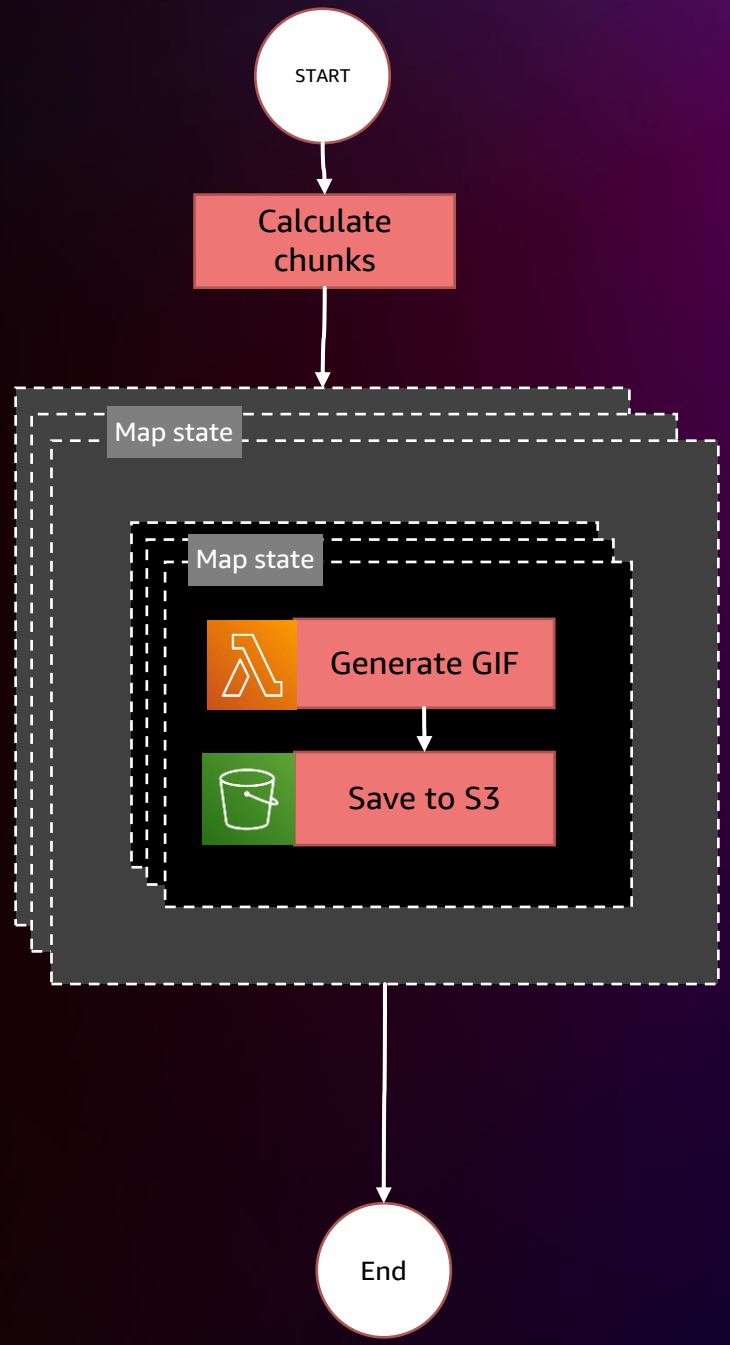
Scalability

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Map state limitations

When your input array has more than 40 items, some iterations may not begin until previous iterations have completed

To achieve greater concurrency, consider using nested state machines that cascade map states: for example, to achieve a concurrency of 1024, you could build a state machine that contains a map state that iterates 32 times, then nest that state machine into the map state of a higher level state machine that iterates 32 times



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step Functions at scale

Customers have been asking for more simple ways to dynamically process large arrays of data without building custom solutions

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

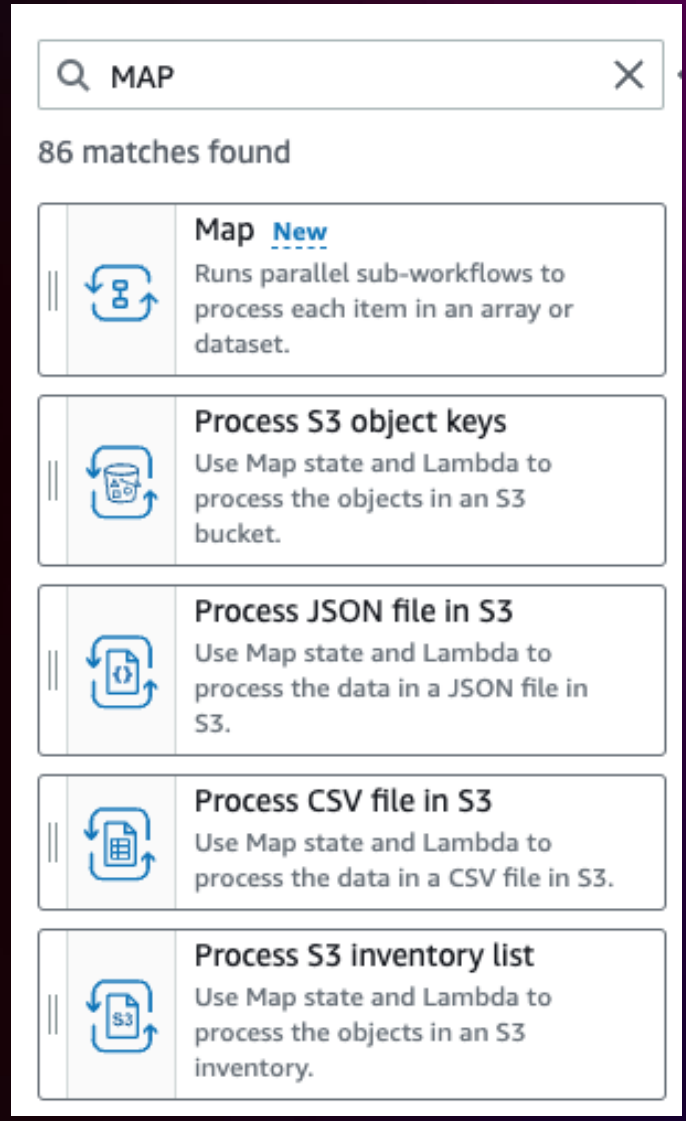
NEW

Introducing new Distributed Map State

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step Functions Distributed Map State

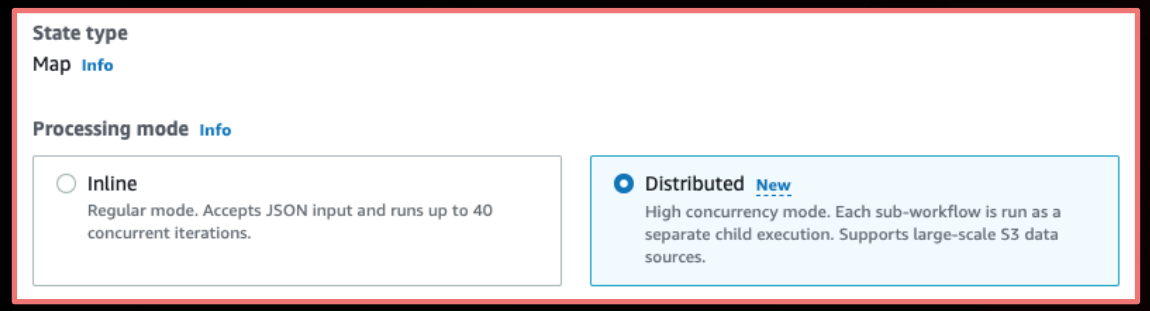
- Coordinate large-scale parallel workloads
- Iterate over millions of S3 objects such as logs, images, or JSON or CSV files
- Up to 10,000 parallel executions
- Invoke Lambda or Amazon ECS/ Fargate for large-scale on-demand serverless compute



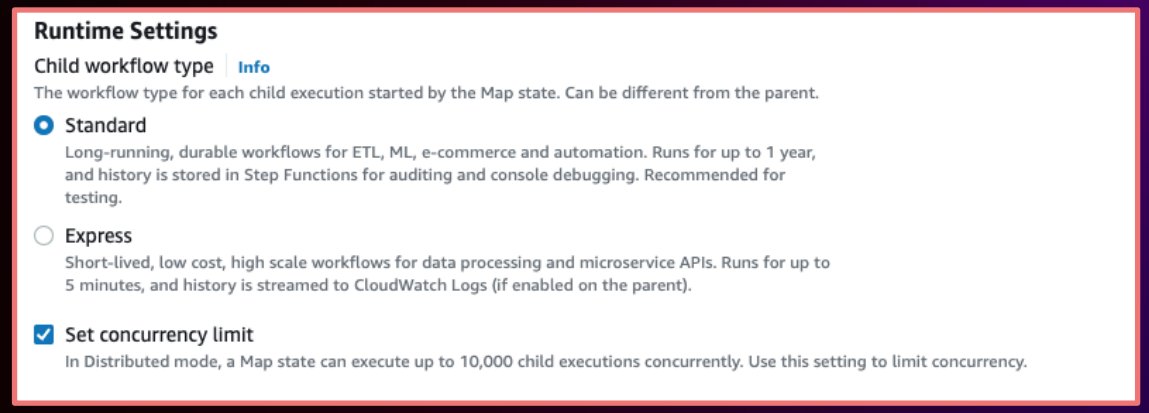
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step Functions Map State updates

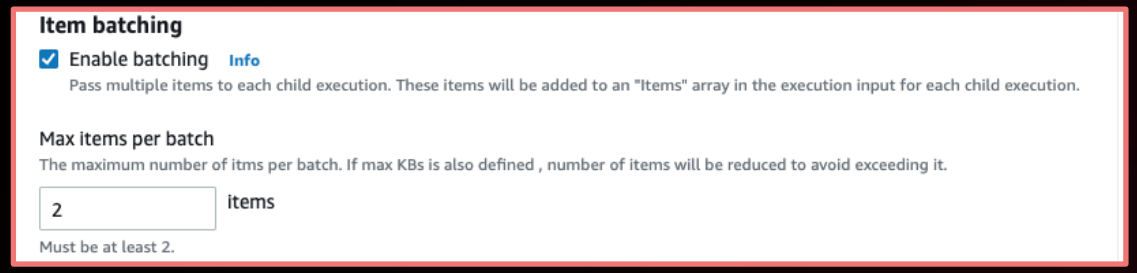
Map States have a Mode field that can be *INLINE* or *DISTRIBUTED*



If *DISTRIBUTED*, then you can set the ExecutionType as either STANDARD or EXPRESS



The ItemBatcher field specifies how items should be batched



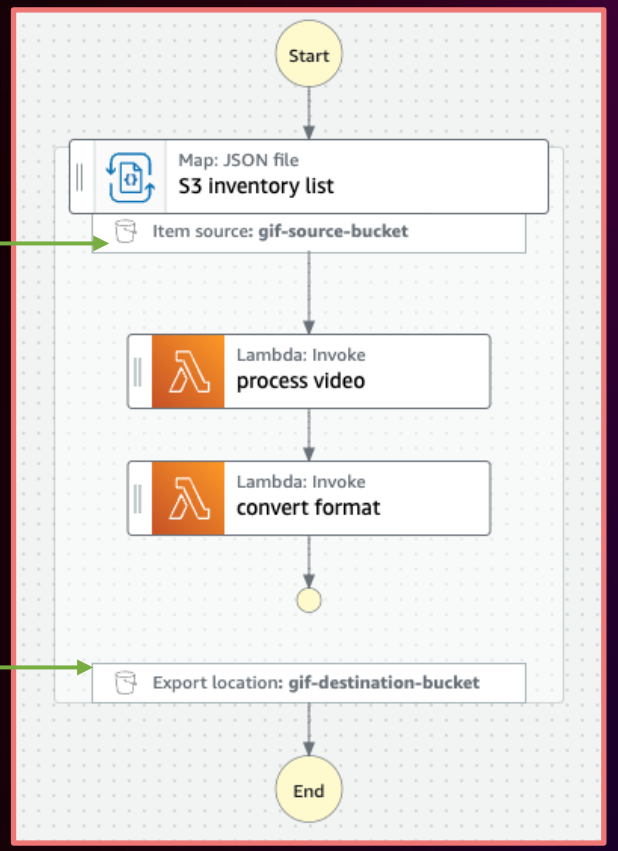
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step Functions Map State updates

ItemReader and ResultWriter fields specify where to read the items and write the result

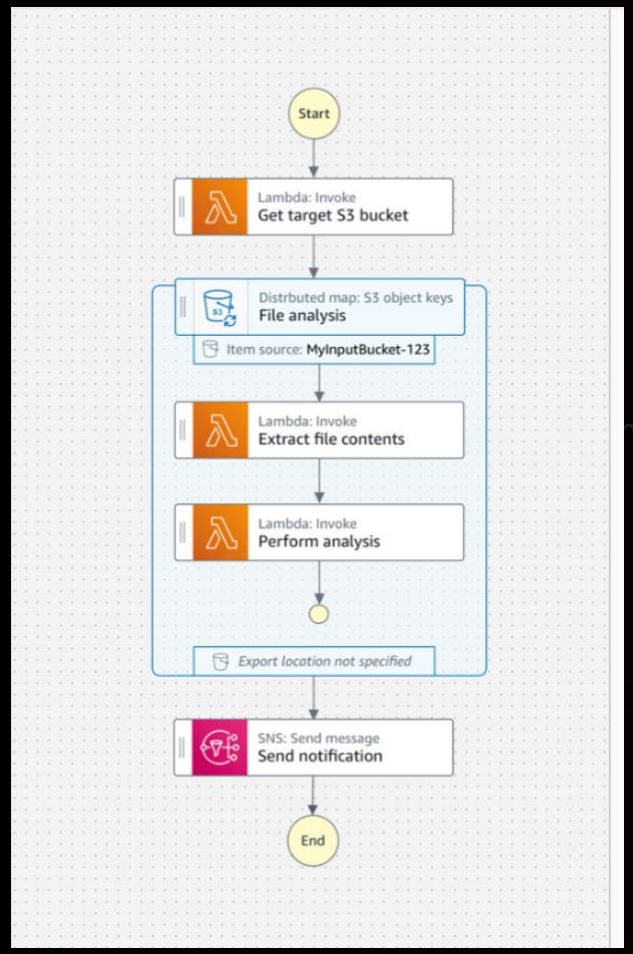
```
"ItemReader": {
  "ReaderConfig": {
    "InputType": "JSON"
  },
  "Parameters": {
    "Bucket": "gif-source-bucket",
    "Key": "demo12.mov"
   },
  "Resource": "arn:aws:states:::s3:getObject"
},
```

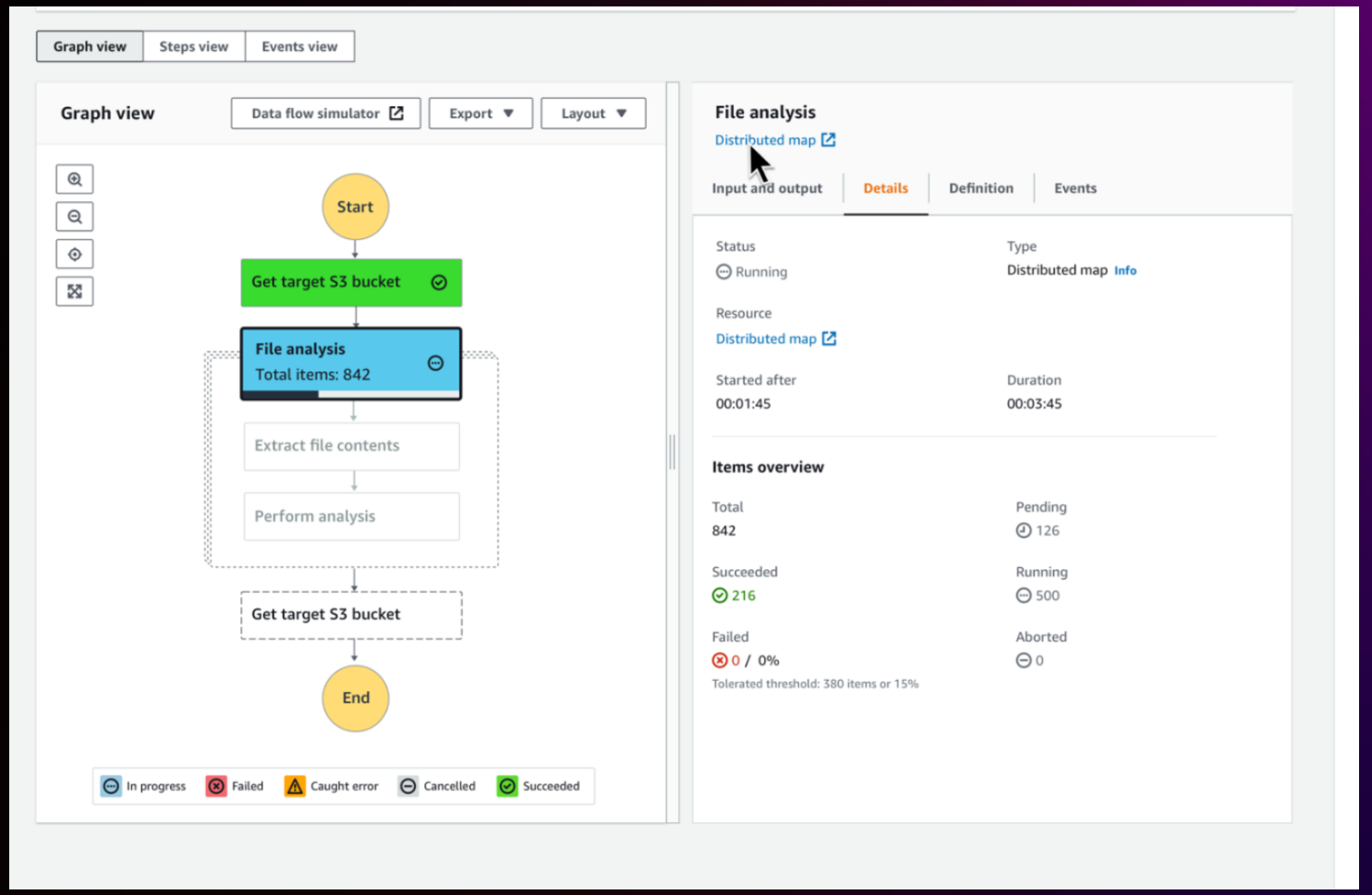
```
"ResultWriter": {
 "Resource": "arn:aws:states:::s3:putObject",
 "Parameters": {
   "Bucket": "gif-destination-bucket",
   "Prefix": "res/"
  }
}
```



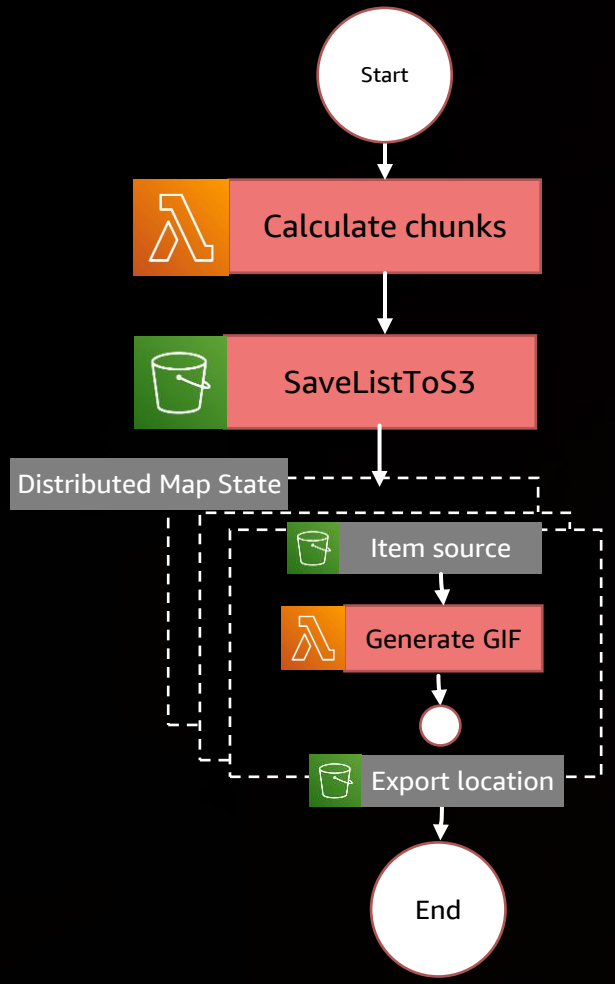
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step Functions Distributed Map State



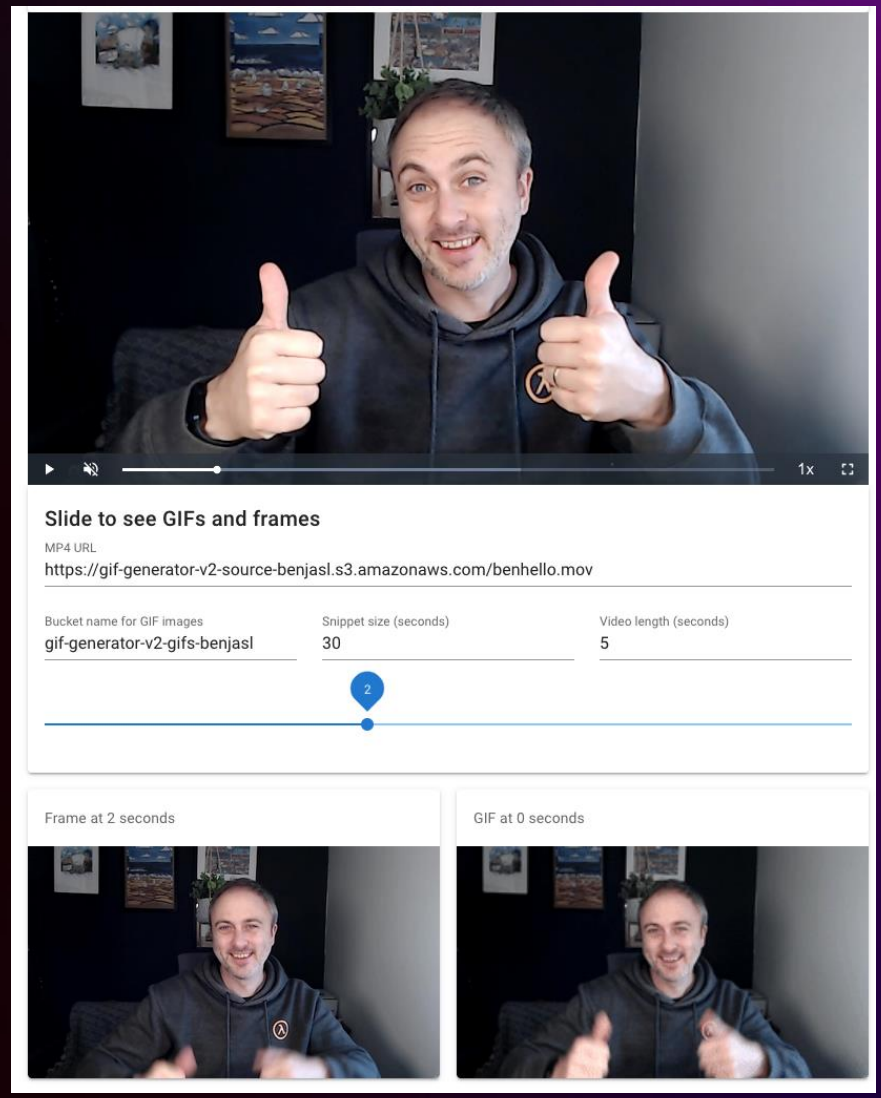


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



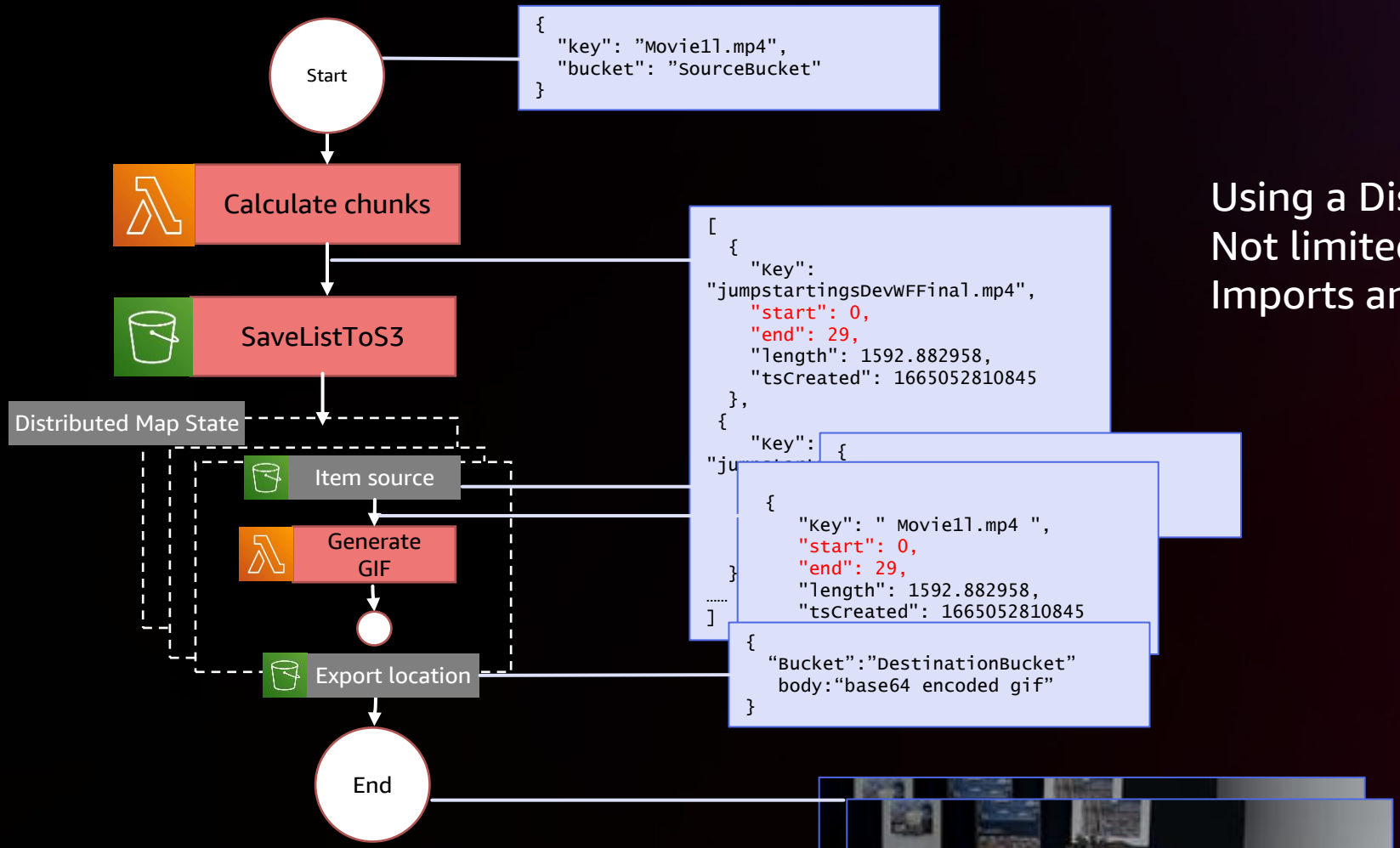
A serverless GIF generator

Convert an .mp4 from Amazon S3 into multiple GIFs for timeline scrubbing



Scalability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Using a Distributed Map State for this workload:
Not limited to 40 concurrent executions
Imports and exports directly from/to Amazon S3



Scalability

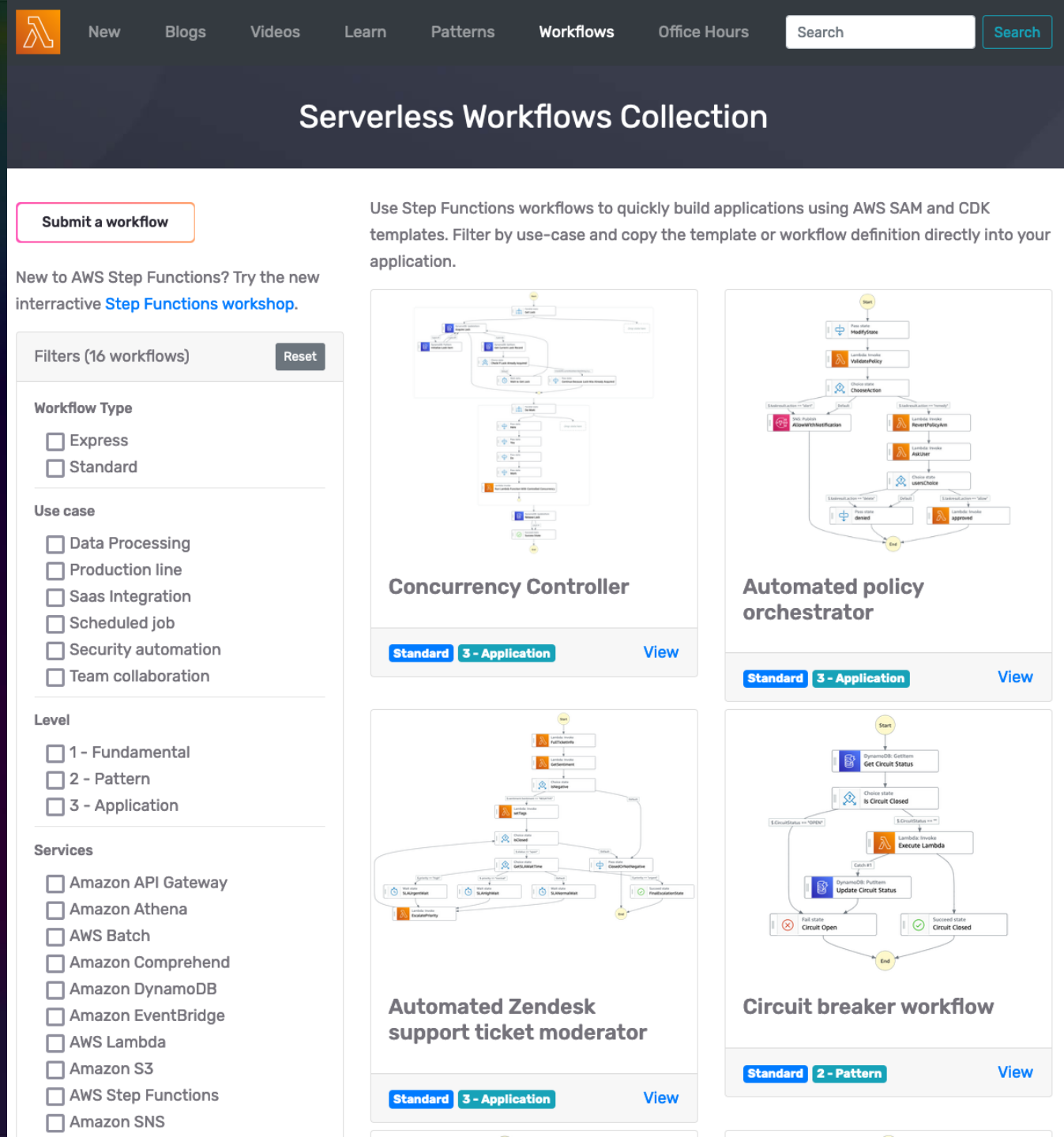
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing the Serverless Workflows collection

Discover, deploy, and share
Step Functions workflows

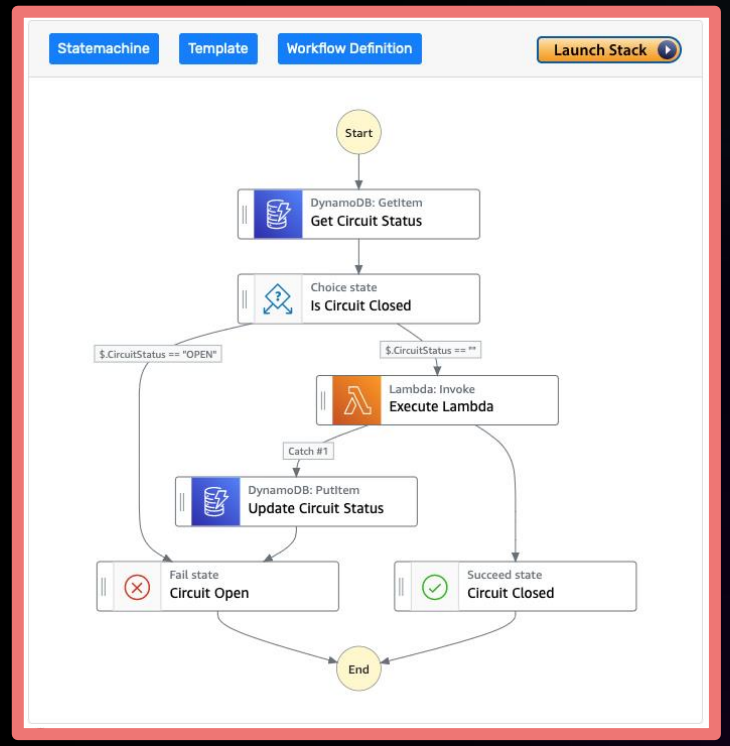
https://s12d.com/workflows

aws © 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

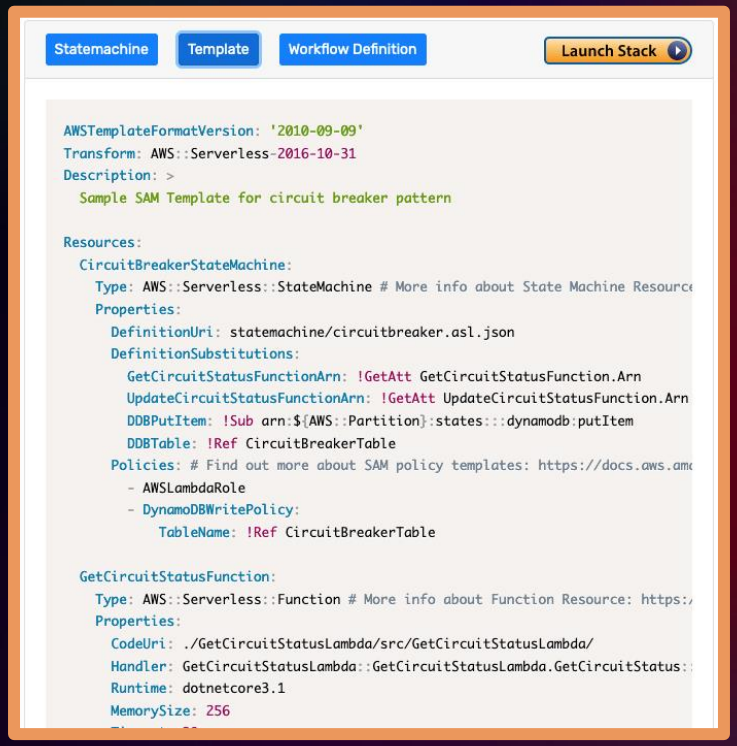


Discover workflow examples

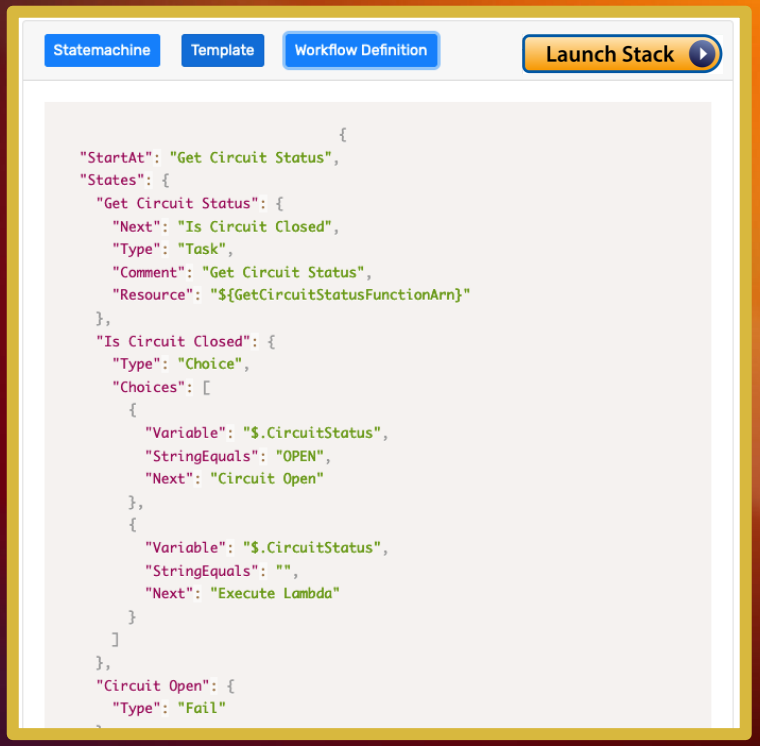
1. Workflow visual



2. IaC template



3. ASL definition



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



serverlesspresso

A re:Invent coffee bar built on serverless technologies
by the serverless DA team

In the Expo Hall

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Everything in one place



https://s12d.com/api309

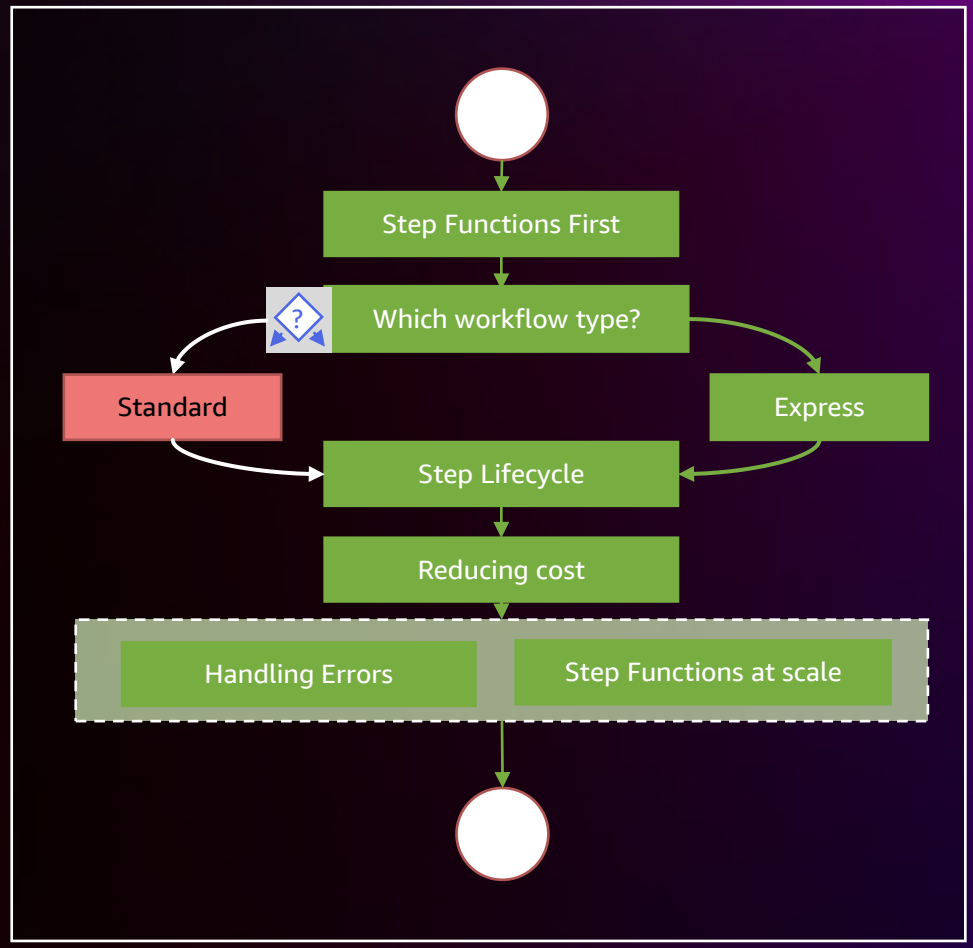
Resources for API309
Advanced serverless workflow patterns and best practices

- Downloadable templates
- Blog posts
- Videos
- Workshops
- Code samples

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Benjamin Smith
@Benjamin_l_s





Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.