

Step Functions,

Overview of new features

Jacob Verhoeks

Awsug.nl 3 October 2022

Jacob Verhoeks

Schuberg Philis

AWS Architect / Data Engineer

<https://www.linkedin.com/in/jacobverhoeks/>

<https://github.com/jverhoeks>

<https://jacob.verhoeks.org/>

AWS
community
builders

\SCHUBERG
PHILIS



Schuberg Philis

My colleagues will be having sessions too, so make sure you do not miss out

14.30 Enri Peters – “Build your first CRUD API in 60 minutes with AWS CDK!”

16.45 Martijn van Dongen – “Fireside Chat with Dr. Werner Vogels”

19.00 Costas Tyfoxylos and Sayantan Khanra – “What's your landing zone energy label? Visualising your risk level using Security Hub”

19.45 Enri Peters – “How I failed to become an AWS Community Builder (but how I will succeed next time and how you could as well)”

20.30 Martijn van Dongen – “Cloud Education At Scale”

20.30 Jos Vliegenthart – “Cloud exit as a service (CEaaS): strategized and stress-free”



We are hiring

Visit our booth



Step functions

Overview

History

Designs

For today



Before Step Functions

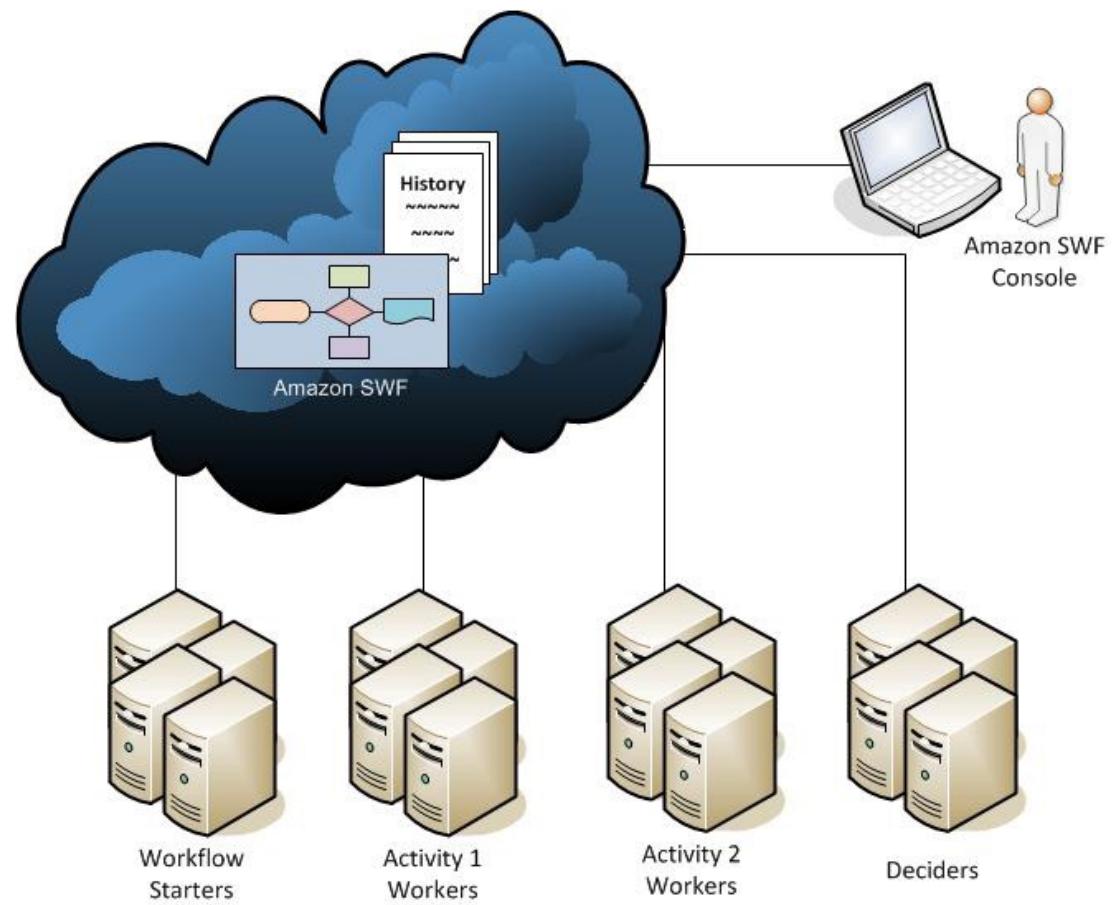
AWS Simple Workflow , released in Feb 2012.

One month after Dynamodb

Workflow orchestration tool

Decouple applications, based on tasks
Running on servers

Then came Lambda in Nov 2014



Step Functions was released at reinvent:2016

At the same time as CodeBuild, Rekognition, Athena and more

Serverless Orchestration



<https://www.awsgeek.com/AWS-History/>

Why

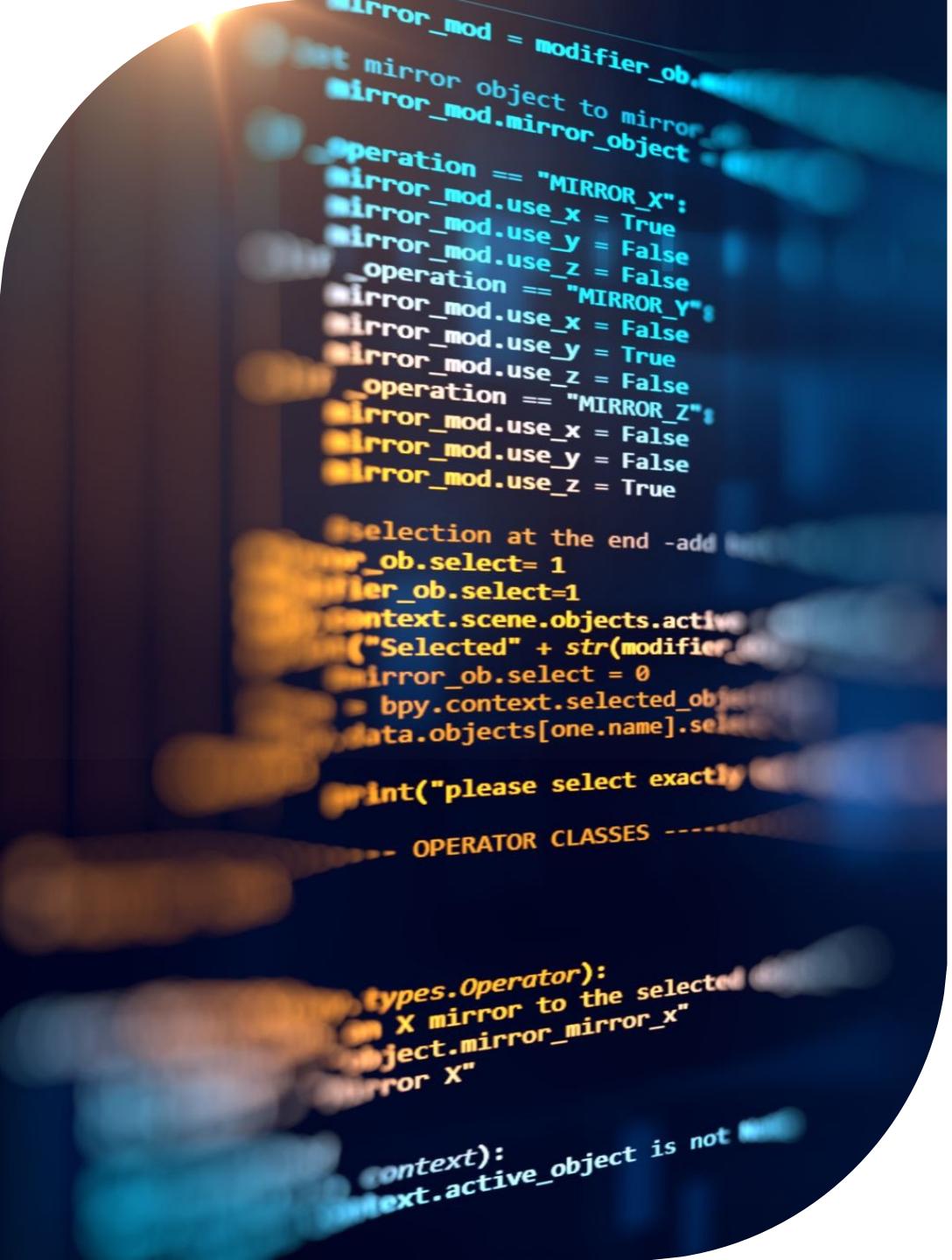
- **Serverless** Application across multiple AWS services
- Sequential actions
- Manage state between aws service calls
- Human intervention required

Use Case

- Data Processing
- Machine Learning
- Microservice Orchestration
- IT and security automation
-

DRY and Lean

- Move error Handling outside
- Don't Repeat Yourself
- Use less libraries = less memory
- Speed up



Patterns

Sequence

Retry failed

Parallel

Choose task based on result

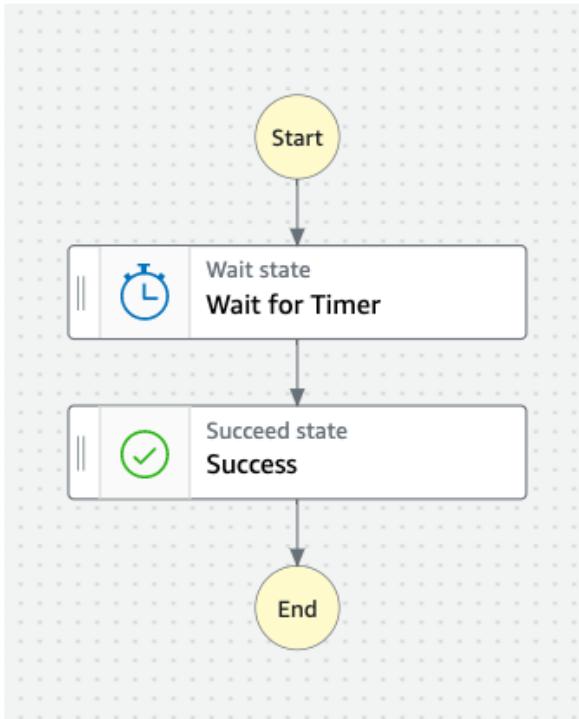
Error handling

Many more

Step Functions States

Pass	Forward Input to Output, for debugging
Task	A unit of work , calling a lambda, an API call etc
Task.Activity	Run work on a worker (ec2/ecs/etc)
Choice	Conditional Logic
Wait	Add a delay
Succeed	Stops with Success
Fail	Stops with Failure
Parallel	Create parallel execution branches
Map	Run a set of Step for each element in the input map

Sample Workflow



```
{  
    "Comment": "An example of the Amazon States Language for scheduling a task.",  
    "StartAt": "Wait for Timer",  
    "States": {  
        "Wait for Timer": {  
            "Type": "Wait",  
            "SecondsPath": "$.timer_seconds",  
            "Next": "Success"  
        },  
        "Success": {  
            "Type": "Succeed"  
        }  
    }  
}
```

Start execution

Start an execution using the latest definition of the state machine. [Learn more](#)

Name - optional

d062cb04-6032-8c1b-1f25-877a703fd40a

Input - optional

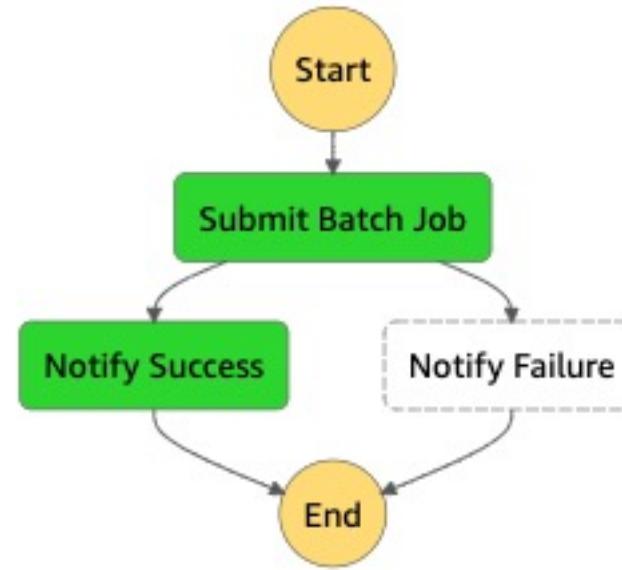
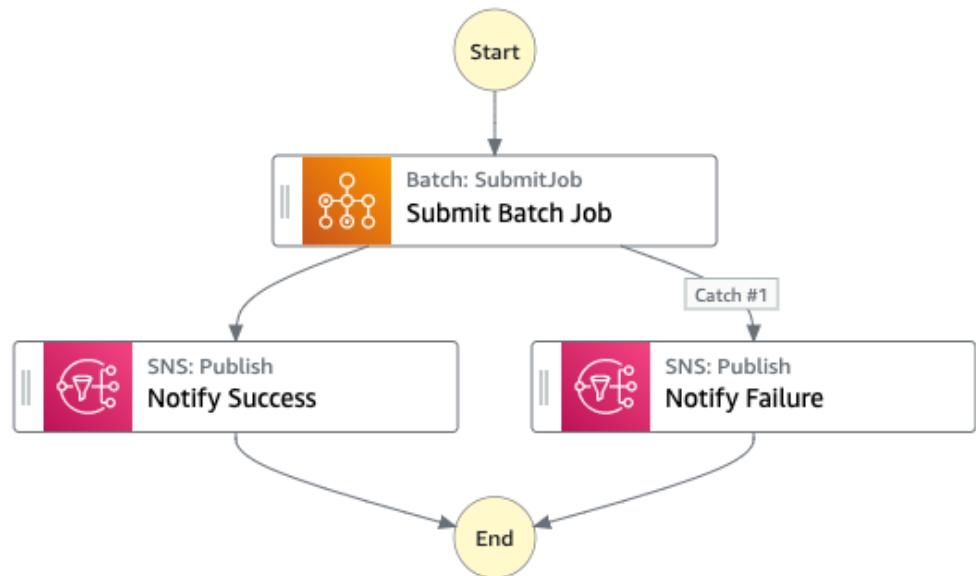
Enter input values for this execution in JSON format

1 { "timer_seconds": 5 }



<https://catalog.us-east-1.prod.workshops.aws/workshops/9e0368c0-8c49-4bec-a210-8480b51a34ac/en-US/module-1/step-2>

Sample Workflow



<https://docs.aws.amazon.com/step-functions/latest/dg/create-sample-projects.html>

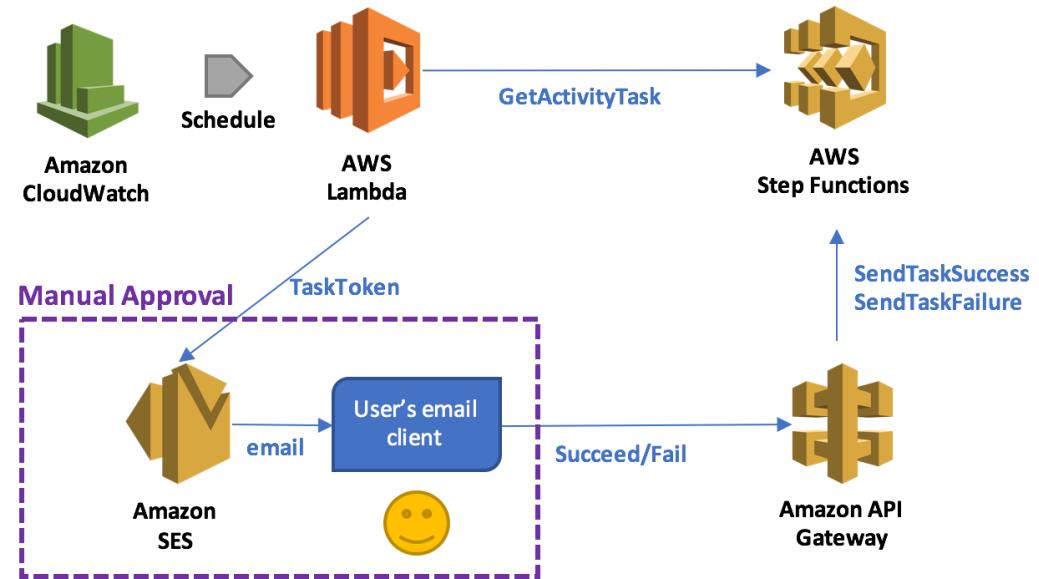
Back to the Future



2017

Cloud Formation Support
Api gateway Integration
Custom Errors for lambda
Cloudwatch Events as Target

Support for Updating state machines



<https://aws.amazon.com/blogs/compute/implementing-serverless-manual-approval-steps-in-aws-step-functions-and-amazon-api-gateway/>

2018

Higher Throughput

Integration:

- Amazon ECS
- AWS Fargate
- Amazon DynamoDB
- Amazon SNS
- Amazon SQS
- AWS Batch
- AWS Glue
- Amazon SageMaker.

<https://aws.amazon.com/about-aws/whats-new/2018/11/aws-step-functions-adds-eight-more-service-integrations/>

2019 Feb Develop and Test Local

Docker or Jar

Override Endpoints for other local testing

```
docker run -p 8083:8083 amazon/aws-stepfunctions-local
```

```
aws stepfunctions --endpoint-url http://localhost:8083 create-state-machine --definition "{\"Comment\": \"A Hello World example of the Amazon States Language using a Pass state\", \"StartAt\": \"HelloWorld\", \"States\": {\"HelloWorld\": {\"Type\": \"Pass\", \"End\": true}}}" --name "HelloWorld" --role-arn "arn:aws:iam::012345678901:role/DummyRole"
```

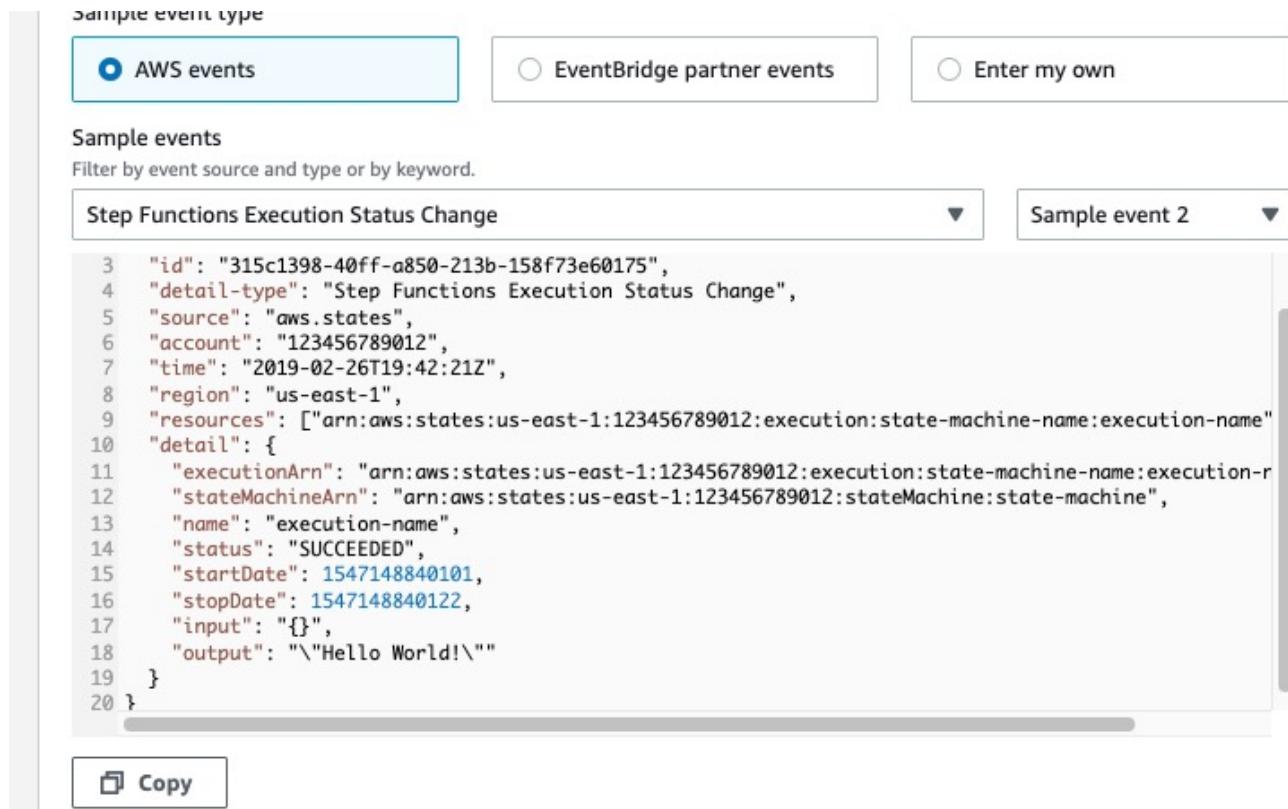
<https://aws.amazon.com/about-aws/whats-new/2019/02/develop-and-test-aws-step-functions-workflows-locally/>

2019 May Workflow execution Events

Integration with Cloudwatch Events , now Event bridge

Have an event if the Step Function

Works
Error
Timeout



The screenshot shows a configuration interface for a sample event. At the top, there are three radio buttons for 'Sample event type': 'AWS events' (selected), 'EventBridge partner events', and 'Enter my own'. Below this is a section titled 'Sample events' with a dropdown menu showing 'Step Functions Execution Status Change' (selected) and another dropdown showing 'Sample event 2'. The main area displays a JSON event payload with line numbers from 3 to 20. The payload details a step function execution status change for a specific execution ID, account, time, region, resources, and execution details.

```
3 "id": "315c1398-40ff-a850-213b-158f73e60175",
4 "detail-type": "Step Functions Execution Status Change",
5 "source": "aws.states",
6 "account": "123456789012",
7 "time": "2019-02-26T19:42:21Z",
8 "region": "us-east-1",
9 "resources": ["arn:aws:states:us-east-1:123456789012:execution:state-machine-name:execution-name"]
10 "detail": {
11     "executionArn": "arn:aws:states:us-east-1:123456789012:execution:state-machine-name:execution-r
12     "stateMachineArn": "arn:aws:states:us-east-1:123456789012:stateMachine:state-machine",
13     "name": "execution-name",
14     "status": "SUCCEEDED",
15     "startDate": 1547148840101,
16     "stopDate": 1547148840122,
17     "input": "{}",
18     "output": "\"Hello World!\""
19 }
20 }
```

<https://aws.amazon.com/about-aws/whats-new/2019/05/aws-step-functions-adds-support-for-workflow-execution-events/>

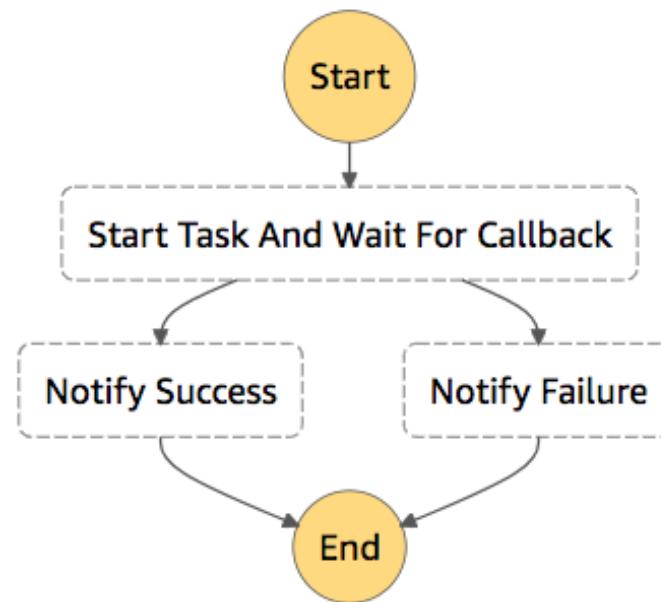
2019 May Callback Patterns

Pause the Workflow and pass the token to other service

Example:

pass token to SQS

Wait for the application to pass
back the token



<https://aws.amazon.com/about-aws/whats-new/2019/05/aws-step-functions-support-callback-patterns/>

2019 Aug Nested Workflows

Build larger and complexer workflows

Re-use reoccurring parts as building blocks

The screenshot shows the AWS Step Functions console interface. On the left, there is a visual workflow diagram consisting of three states: 'Start' (yellow circle at the top), 'Step Functions: StartExecution' (a rectangular state with a red header containing a flowchart icon and a blue footer labeled 'Step Functions StartExecution'), and 'End' (yellow circle at the bottom). An arrow points from 'Start' to the Step Functions state, and another arrow points from the Step Functions state to 'End'. To the right of the diagram, there is a configuration panel with the following sections:

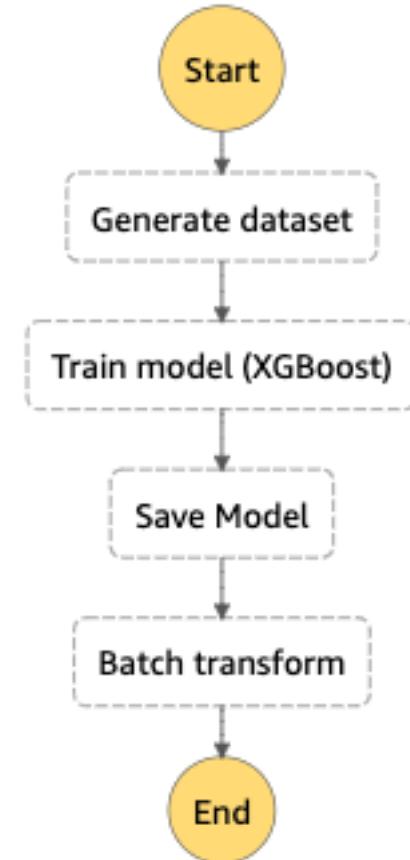
- Additional configuration**:
 - Integration pattern**: [Info](#). Specify how Step Functions manages the API call.
 - Call and continue**: Step Functions calls the API, receives an HTTP response, and continues to the next step.
 - Wait for child execution to complete**: Pause the parent execution at this state and monitor the child execution. Resume the parent execution when the child execution is complete.
 - Wait for callback**: Pause the execution at this state until the execution receives a callback from the SendTaskSuccess or SendTaskFailure APIs with the task token.
 - Response format**: Specify the response format of the child execution.
 - JSON

<https://aws.amazon.com/about-aws/whats-new/2019/08/aws-step-function-adds-support-for-nested-workflows/>

2019 Oct Sagemaker Integration

Move from Jupyter Notebooks to
Serverless machine learning workflow

ML-Ops



<https://aws.amazon.com/about-aws/whats-new/2019/10/aws-step-functions-expands-amazon-sagemaker-service-integration/>

2019 Nov Data Science SDK

Build a Step Function Workflow in Python
Predefined templates , TrainingPipeline, Inference Pipeline

Export to CloudFormation

Replaced by CDK?

Create the `TrainingStep` for the Workflow

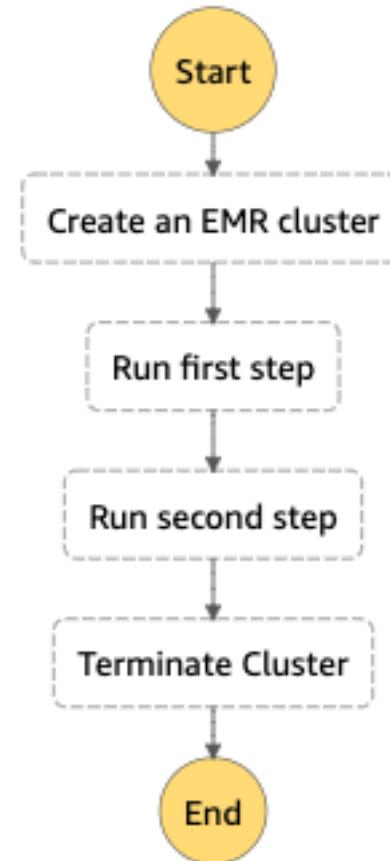
```
training_step = steps.TrainingStep(  
    "SageMaker Training Step",  
    estimator=sklearn,  
    data={"train": sagemaker.TrainingInput(preprocessed_training_data, content_type="text/csv")},  
    job_name=execution_input["TrainingJobName"],  
    wait_for_completion=True,  
)
```

<https://aws.amazon.com/about-aws/whats-new/2019/11/introducing-aws-step-functions-data-science-sdk-amazon-sagemaker/>

2019 Nov EMR Integration

Control EMR from Step Functions

On-Demand



<https://aws.amazon.com/about-aws/whats-new/2019/11/aws-step-functions-adds-amazon-emr-service-integration/>

2019 Re:invent Express Workflows

Cost Effective Faster

- Api-gateway
- IOT
- Event Bridge

	Standard	Express
Maximum duration	1 year	5 minutes
Supported execution start rate	Over 2,000 per second*	Over 100,000 per second*
Supported state transition rate	Over 4,000 per second*	Nearly unlimited*
Pricing	Priced per state transition. A state transition is counted each time a step in your execution is completed. You are charged \$25 per million state transitions.**	Priced by the number of executions you run, their duration, and memory consumption. You are charged \$1 per million executions, and duration price from \$0.000004 to \$0.00001 per GB-second.
Execution history	Executions can be listed and described with Step Functions APIs, and visually debugged in the console.	Executions can be inspected in CloudWatch Logs by enabling logging on your state machine.
Execution semantics	Exactly-once workflow execution.	At-least-once workflow execution.
Service integrations	Supports all service integrations and patterns.	Supports all service integrations. Does not support Job-run (.sync) or Callback (.waitForTaskToken) patterns.
Step Functions activities	Supports Step Functions activities.	Does not support Step Functions activities.

<https://aws.amazon.com/about-aws/whats-new/2019/12/introducing-aws-step-functions-express-workflows/>

2019 Re:invent Private Link

Reach step function from inside the VPC

No need to traverse
out to the internet

Number of VPC Interface endpoints per AWS region
1

Number of Availability Zones an Interface endpoint is deployed in
3

Total data processed by all VPCE Interface endpoints in the AWS region
Pricing Tiers -> 0-1PB @ 0.01/GB | 1-4PB @ 0.006/GB | 5+PB @ 0.004/GB

100 GB per month

▼ Show calculations

1 VPC endpoints x 3 ENIs per VPC endpoint x 730 hours in a month x 0.01 USD = 21.90 USD (Monthly cost for endpoint ENI)

Monthly cost for Interface endpoints: 21.90 USD

Tiered price for: 100 GB

100 GB x 0.0100000000 USD = 1.00 USD

Total tier cost = 1.0000 USD (PrivateLink data processing cost)

Total data processing cost: 1 USD

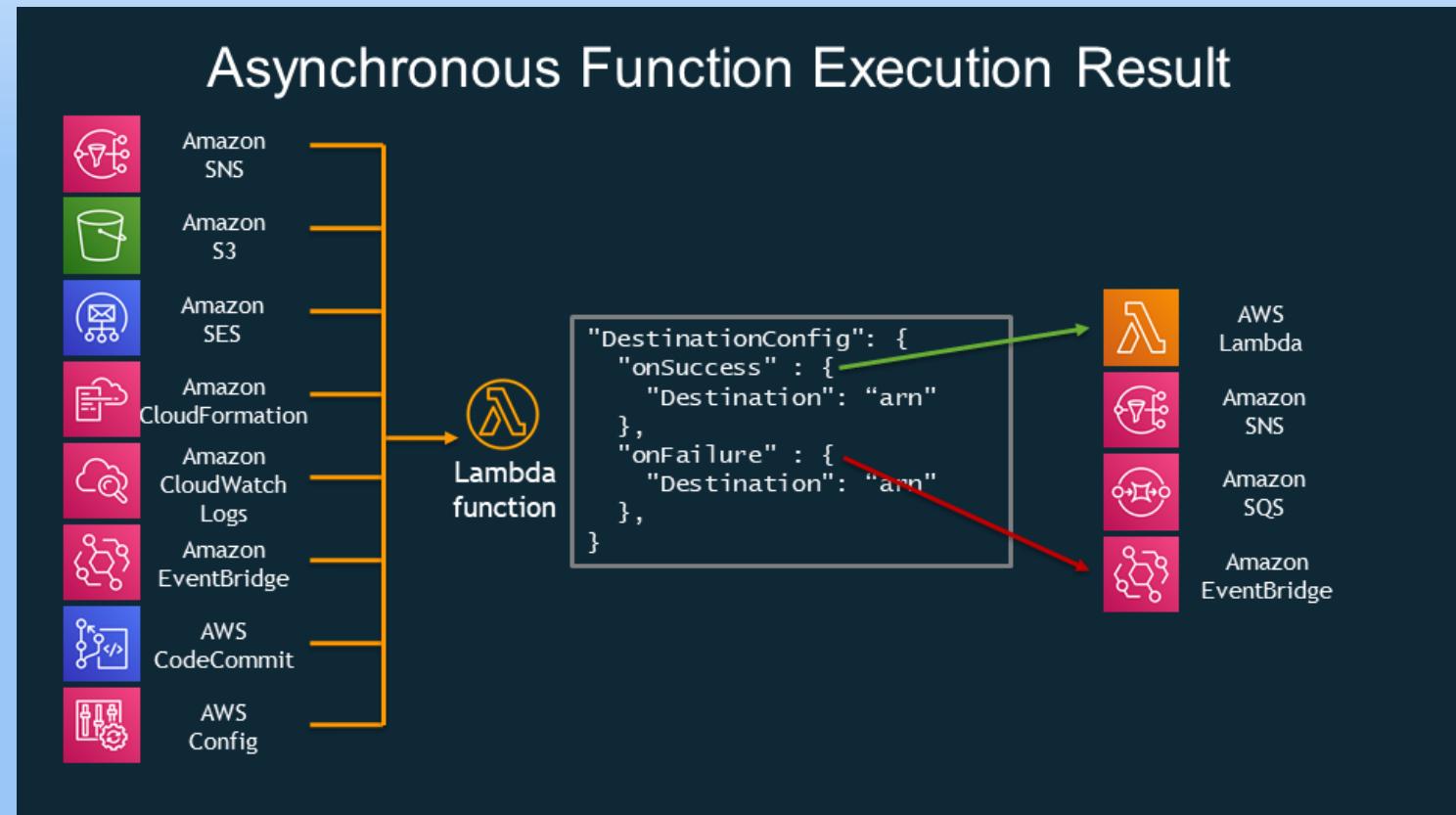
21.90 USD + 1 USD = 22.90 USD (Total PrivateLink Cost)

Total PrivateLink endpoints and data processing cost (monthly): 22.90 USD

<https://aws.amazon.com/about-aws/whats-new/2019/12/aws-step-functions-now-supports-aws-privatelink/>

2019 Re:invent Lambda Destinations

Simple workflow for Lambda's
Asynchronous



<https://aws.amazon.com/blogs/compute/introducing-aws-lambda-destinations/>

2020 Mar Visual Studio Code Support

Create, Edit, Debug

The screenshot displays the Visual Studio Code interface with the AWS Toolkit for Visual Studio Code extension installed. On the left, the 'Hello-World.asl.json' file is open, showing ASL (Amazon States Language) code. A code snippet for the 'Batch Task State' is highlighted, with a tooltip providing a detailed description: 'Code snippet for a Batch job Task state. Calls the AWS Batch SubmitJob API and resumes the execution once the job is complete.' On the right, the 'Graph: Hello-World.asl.json' view shows the state transition graph. It starts with a 'Start' state, followed by a 'Pass' state. From there, it branches to a decision state 'Hello World example?'. If the answer is 'Yes', it proceeds to a 'Wait 3 sec' state, then to two parallel states 'Hello' and 'World', which then merge back into a single path leading to an 'End' state. If the answer is 'No', it directly leads to the 'End' state. The bottom status bar indicates 'Previewing ASL document. View'.

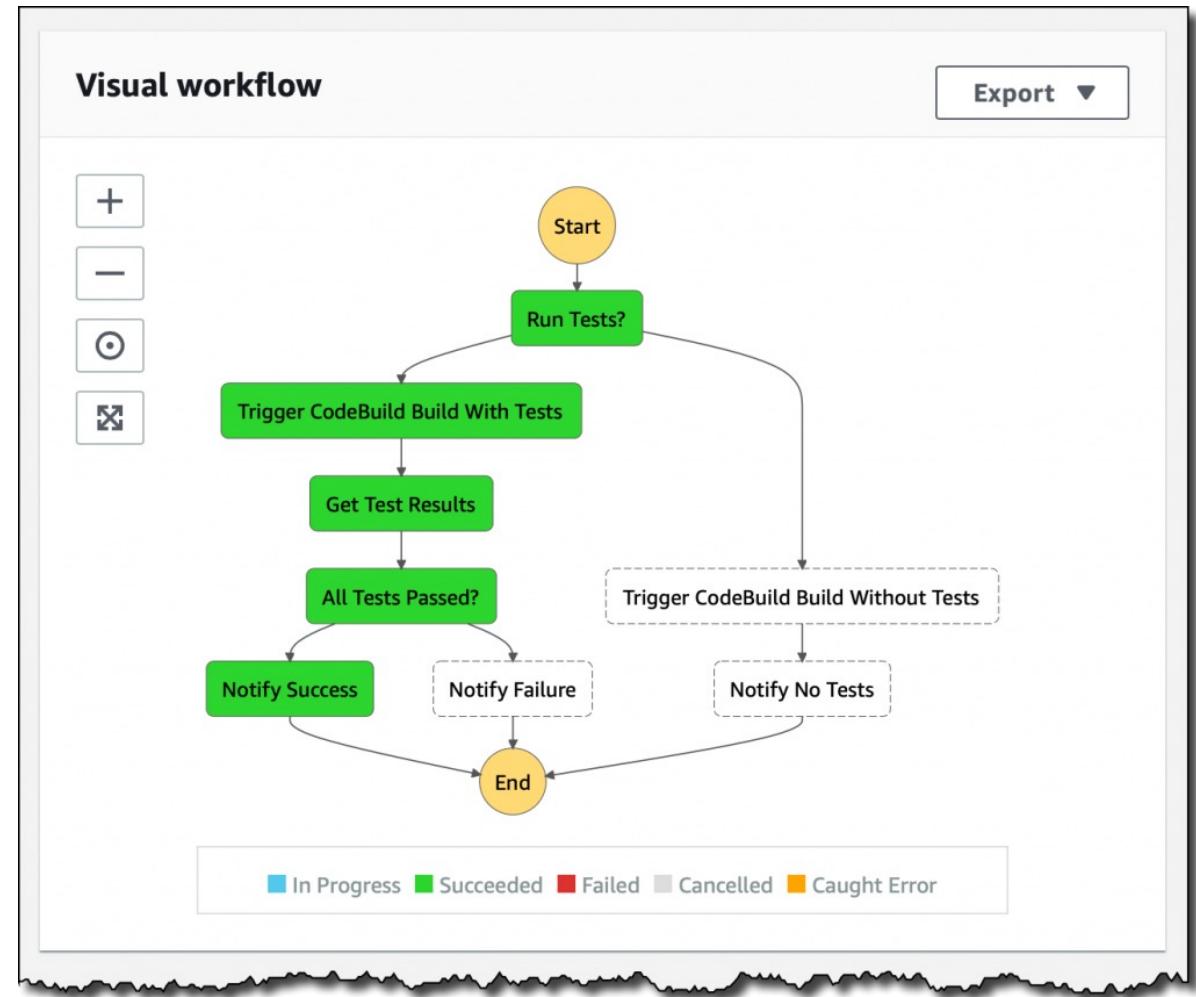
```
1 {
2     "Comment": "A Hello World example demonstrating various states and transitions in AWS Step Functions."
3     "StartAt": "Pass",
4     "States": [
5         "Pass": {
6             "Comment": "A Pass state passes its input to the next state without modification.",
7             "Type": "Pass",
8             "Next": "Hello World example?"
9         },
10        "Batch Task State", // This is the selected code snippet
11        "Choice State",
12        "ECS Task State",
13        "Fail State",
14        "Lambda Task State",
15        "Map State",
16        "Parallel State",
17        "Pass State",
18        "SNS Task State",
19        "SQS Task State",
20        "Succeed State",
21        "Wait State"
22    ],
23    "Next": "No"
24 },
25 ],
26 "Default": "Yes"
27 },
28 "Yes": {
```

<https://aws.amazon.com/about-aws/whats-new/2020/03/aws-toolkit-for-visual-studio-code-supports-aws-step-functions/>

2020 May Code Build Support

Create a webhook, called from Github to run CodeBuild

Use Cloudwatch Events/Event Bridge to schedule builds



<https://aws.amazon.com/about-aws/whats-new/2020/05/aws-step-functions-supports-aws-codebuild-service-integration/>

2020 May AWS Sam Support

Use SAM to deploy
your serverless workloads

```
YAML

Resources:
  SAMLogs:
    Type: AWS::Logs::LogGroup

  SimpleStateMachine:
    Type: AWS::Serverless::StateMachine
    Properties:
      Definition: {...}
      Logging:
        Destinations:
          - CloudWatchLogsLogGroup:
              LogGroupArn: !GetAtt SAMLogs.Arn
            IncludeExecutionData: true
            Level: ALL
      Policies:
        - CloudWatchLogsFullAccess
    Type: EXPRESS
```

<https://aws.amazon.com/about-aws/whats-new/2020/05/aws-sam-adds-support-for-aws-step-functions/>

2020 Aug

New Choice State Operators

- Test if Null
- Variable Exists
- Wildcarding
- Compare variables
- Dynamic Timeouts
- String and Array Construction
- String to Json
- Json to String

```
{  
  "Parameters": {  
    "foo.$": "States.Format('Hello, {} {}', $.firstName, $.lastName)"  
  }  
}  
  
{  
  "Type": "Task",  
  "Resource": "arn:aws:states:::some.future.integration:run.sync",  
  "Parameters": {  
    "FieldThatNeedsToBeAString.$":  
      "States.JsonToString($.JSONInputField)",  
  }  
}
```

<https://aws.amazon.com/about-aws/whats-new/2020/08/aws-step-functions-support-string-manipulation-comparison-operators-improved-output-processing/>

2020 Sep increase Payload to 256K

From 32K to 256K

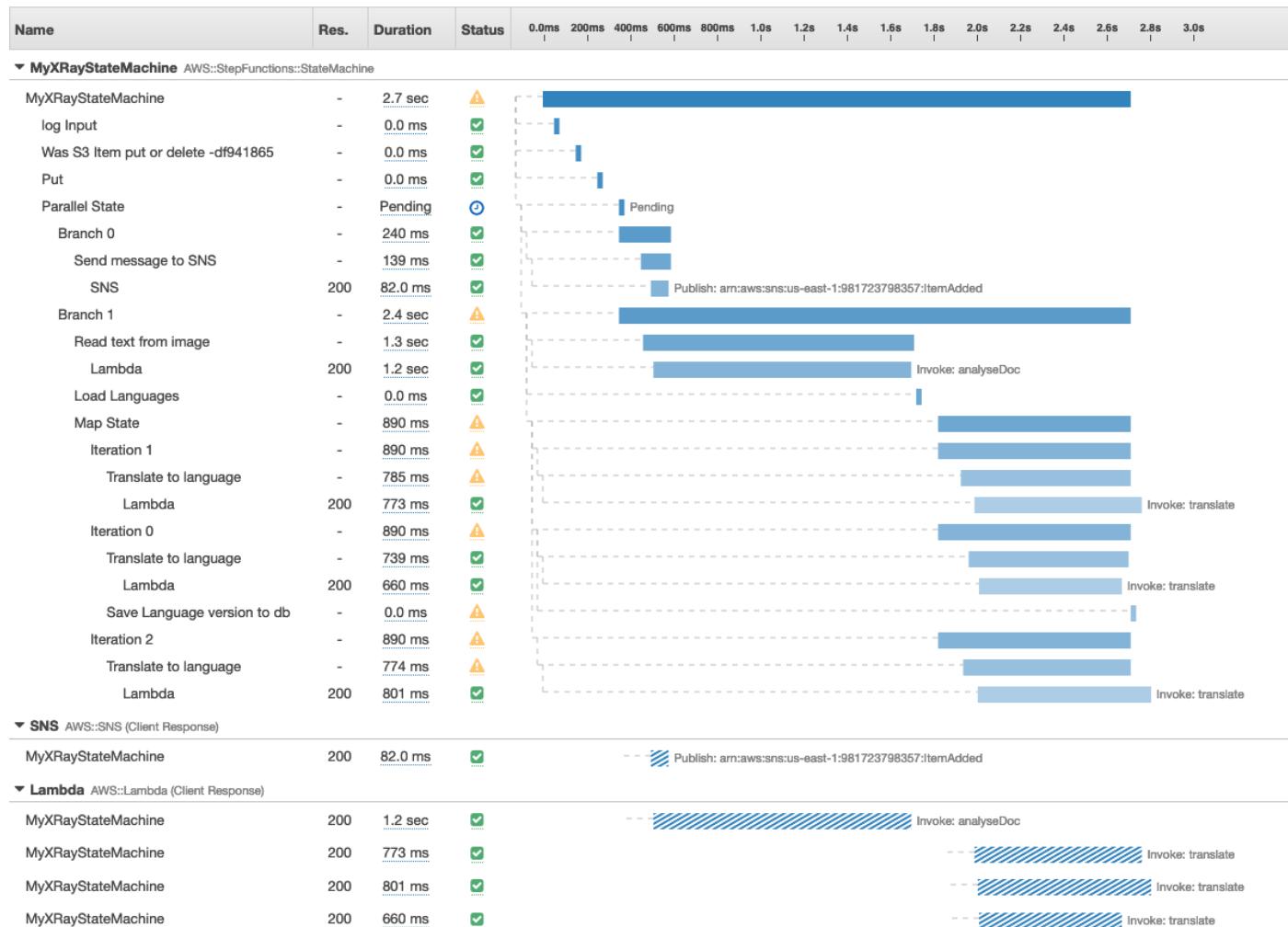
Handle more data in one execution

Same as services like DynamoDB, Lambda, SNS, SQS

2020 Sep AWS X-Ray Support

Detailed insights

Faster Troubleshooting



<https://aws.amazon.com/about-aws/whats-new/2020/09/aws-step-functions-adds-support-for-aws-x-ray/>

2020 Oct Amazon Athena Integration

Query Data
use for other processes

```
"Start an Athena query": {  
    "Type": "Task",  
    "Resource": "arn:aws:states:::athena:startQueryExecution.sync",  
    "Parameters": {  
        "QueryString": "SELECT * FROM \"myDatabase\".\"myTable\" limit 1",  
        "WorkGroup": "primary",  
        "ResultConfiguration": {  
            "OutputLocation": "s3://athenaQueryResult"  
        }  
    },  
    "Next": "Get results of the query"  
}
```

<https://aws.amazon.com/about-aws/whats-new/2020/10/aws-step-functions-now-supports-amazon-athena-service-integration/>

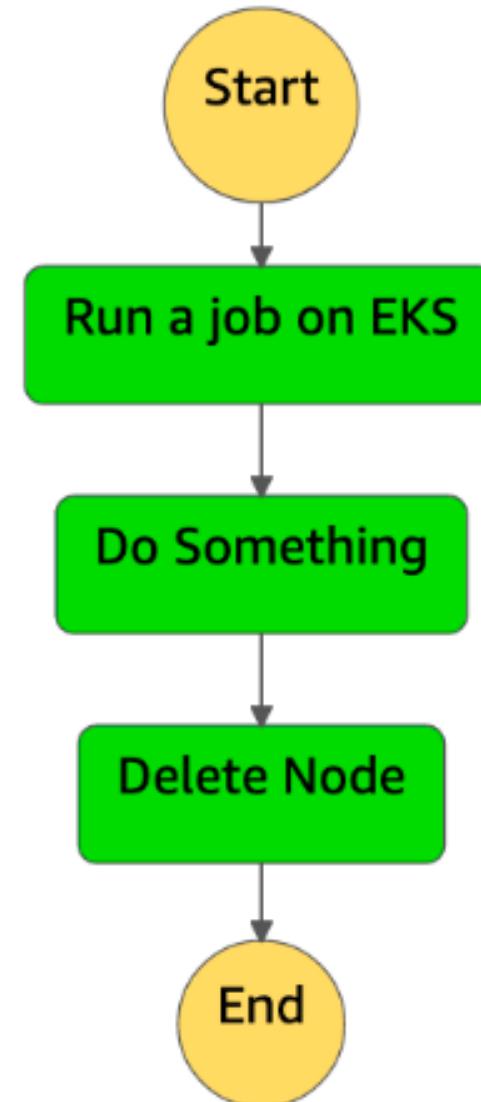
2020 Nov EKS Integration

Run Job

Call EKS API

Create/DeleteCluster

Create/DeleteNoteGroup

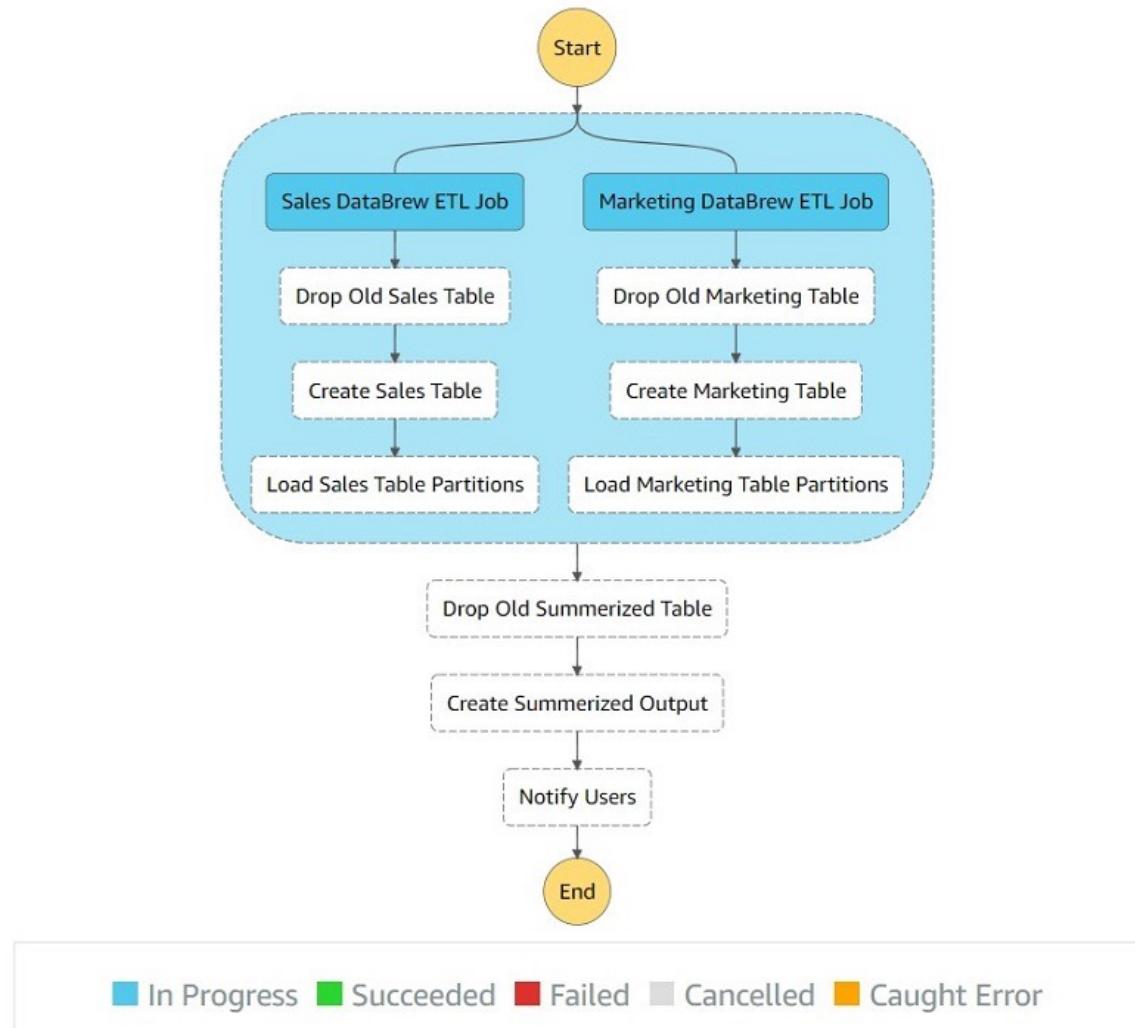


<https://aws.amazon.com/about-aws/whats-new/2020/11/aws-step-functions-now-supports-amazon-eks-service-integration/>

2021 Jan Glue DataBrew Support

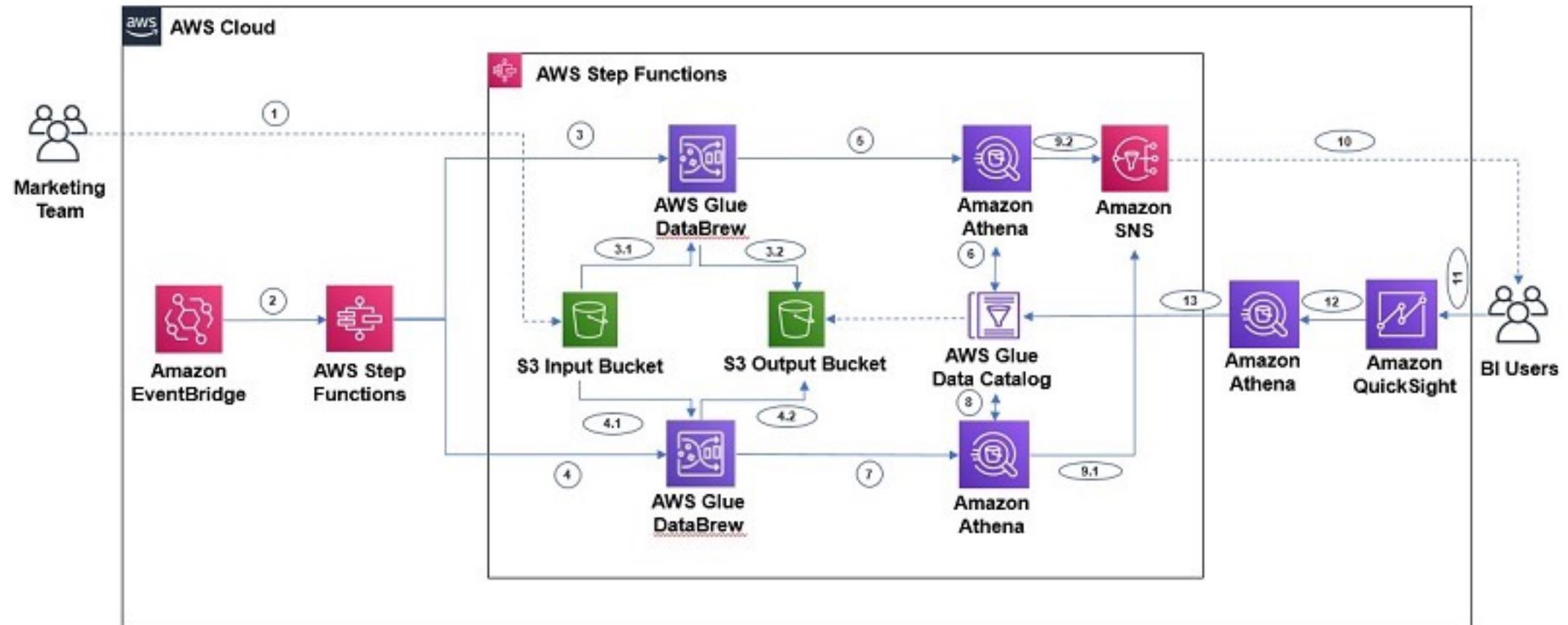
Visual Data Preparation

Separate Responsibility



<https://aws.amazon.com/blogs/big-data/orchestrating-an-aws-glue-databrew-job-and-amazon-athena-query-with-aws-step-functions/>

2021 Jan Glue DataBrew Support



<https://aws.amazon.com/about-aws/whats-new/2021/01/aws-step-functions-support-aws-glue-databrew-jobs-data-analytics-machine-learning-workflows/>

2021 Mar Yaml Support

Until now only JSON was support

```
Comment: "Example Workflow"
StartAt: exampleStep
States:
  exampleStep:
    Type: Task
    Resource: !GetAtt example-step-lambda.Arn
    TimeoutSeconds: 28
    End: true
```

<https://aws.amazon.com/about-aws/whats-new/2021/03/aws-step-functions-adds-tooling-support-for-yaml/>

2021 April Data Flow Simulator

Test your workflow

InputPath
Parameters
ResultSelector
OutputPath
ResultPath.

The screenshot shows the AWS Step Functions Data flow simulator page. The URL is eu-west-1.console.aws.amazon.com/states/home?region=eu-west-1#/simulator. The page has a sidebar with 'Step Functions' (highlighted with a red box), 'State machines', 'Activities', 'Data flow simulator' (highlighted with a red box and labeled '1.'), 'Feature spotlight', 'Local Development', and 'Join our feedback panel'. The main content area is titled 'Data flow simulator' and contains a brief description of how it simulates data processing. It lists two ways to use the tool: learning about optional filters and simulating data flow in a Task state. A large red box highlights the navigation bar at the top of the content area, which includes 'State Input', 'InputPath', 'Parameters', 'Task Result', 'ResultSelector', 'ResultPath', 'OutputPath', and 'State Output'. Below this, a section titled 'State input' shows a JSON object representing movie data. A red box highlights the 'Next' button at the bottom right of the page, labeled '3.'.

```
1. Data flow simulator
2. State Input > InputPath > Parameters > Task Result > ResultSelector > ResultPath > OutputPath > State Output
3. Next
```

1. Data flow simulator

In Step Functions, individual states receive JSON as input and usually pass JSON as output to the next state. Understanding how this data flows from state to state, and learning how to filter and manipulate it, is key to effectively designing and implementing workflows. [Learn more](#)

This tool simulates the order of data processing that occurs in a single [Task state](#) during execution.

You can use this tool:

- To learn about the various optional filters by reviewing the provided sample inputs and outputs of each step, and/or
- To simulate the data flow in a given Task state by populating the fields based on your own state machine.

State input

Each state in a state machine receives JSON as input and passes JSON as output. The state input can be reduced using one or more filters available to the state.

State input

```
1 v {
2   "version": 4,
3   "library": {
4     "movies": [
5       {
6         "genre": "crime",
7         "director": "Quentin Tarantino",
8         "title": "Reservoir Dogs",
9         "year": 1992
10      },
11      {
12        "genre": "action",
13        "director": "Brian De Palma",
14        "title": "Mission: Impossible",
15      }
16    ]
17  }
18 }
```

Must be a valid Json object.

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

<https://aws.amazon.com/about-aws/whats-new/2021/04/aws-step-functions-adds-new-data-flow-simulator-for-modelling-input-and-output-processing/>

2021 May Custom Event Bridge Support

YAML

```
Send an EventBridge custom event:  
  Type: Task  
  Resource: 'arn:aws:states::::events:  
  Parameters:  
    Entries:  
      - Detail:  
        Message: 'Hello from Step F  
        DetailType: MyDetailType  
        EventBusName: MyEventBusName  
        Source: MySource  
  Next: NEXT_STATE
```

YAML

```
Send an EventBridge custom event:  
  Type: Task  
  Resource: 'arn:aws:states::::events:putEvents.waitForTaskToken'  
  Parameters:  
    Entries:  
      - Detail:  
        Message: 'Hello from Step Functions!'  
        TaskToken.$: $$.Task.Token  
        DetailType: MyDetailType  
        EventBusName: MyEventBusName  
        Source: MySource  
  Next: NEXT_STATE
```

Fire and forget
Standard & Express

Wait for answer
Standard

<https://aws.amazon.com/about-aws/whats-new/2021/05/aws-step-functions-now-supports-amazon-custom-events-eventbridge/>

A photograph of a man and a woman playing foosball in a modern office lobby. They are both leaning over the table, focused on the game. The man is wearing a blue patterned shirt and jeans, while the woman is wearing a dark top. The foosball table has the brand name "eVolutio" on it. In the background, there are large windows with white curtains, a blue sofa, and a potted plant. The overall atmosphere is casual and professional.

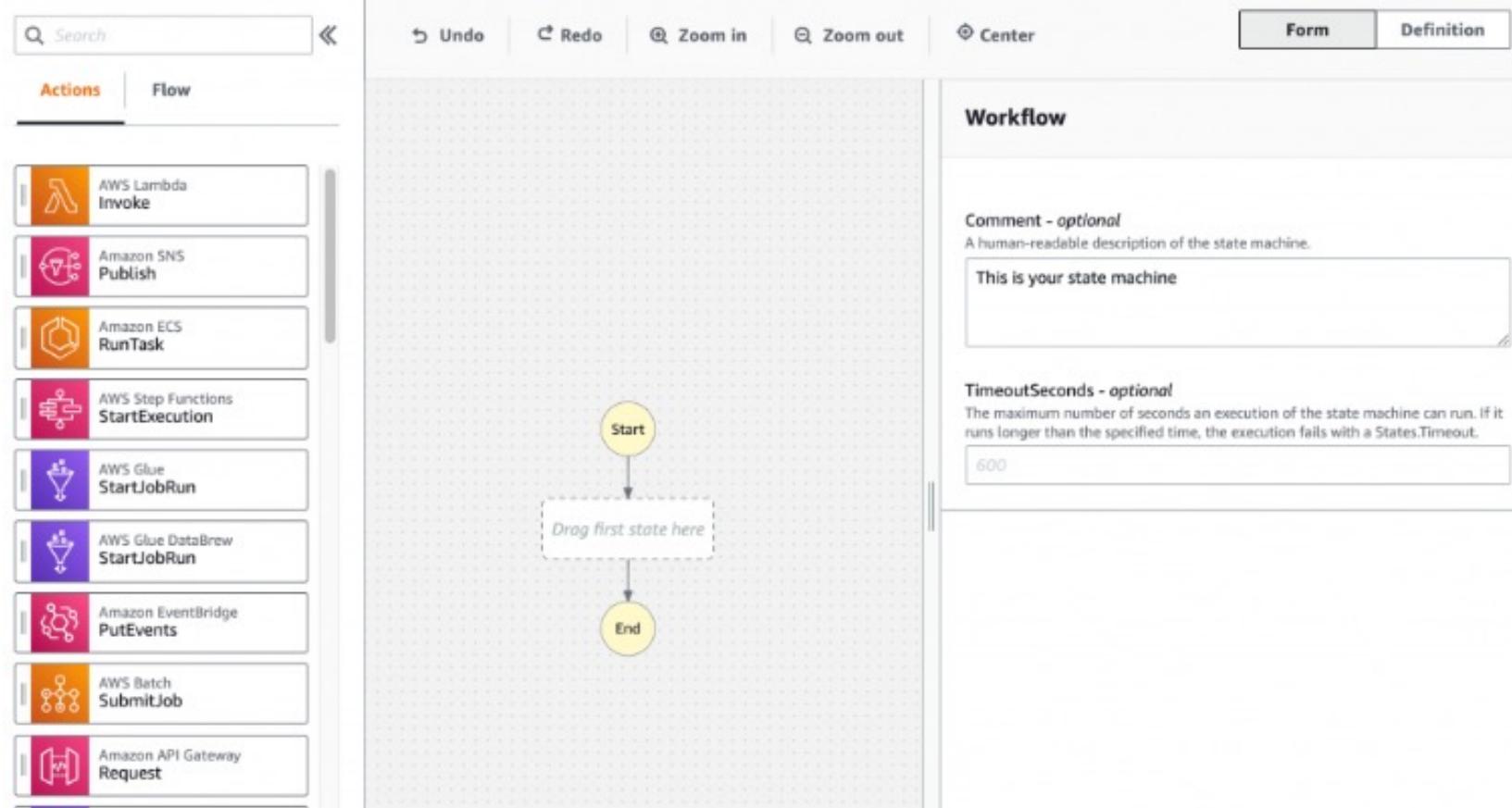
June 2021 Game Changer

\SCHUBERG
PHILIS

2021 June Workflow Studio

Drag & Drop

All parameters
visible



<https://aws.amazon.com/about-aws/whats-new/2021/07/announcing-workflow-studio-a-new-low-code-visual-workflow-designer-for-aws-step-functions/>

2021 Sept > 200 AWS Services SDK Integrations

Call API directly and work with the response

Example: Dynamodb GetItem, S3 GetObject

```
"GetTranscriptionText": {  
    "Type": "Task",  
    "Resource": "arn:aws:states:::aws-sdk:s3:getObject",  
    "Parameters": {  
        "Bucket.$": "$.S3BucketName",  
        "Key": "transcribe.json"  
    },  
    "ResultSelector": {  
        "filecontent.$": "States.StringToJson($.Body)"  
    },  
    "ResultPath": "$.transcription",  
    "Next": "PrepareTranscriptTest"  
},
```

arn:aws:states:::aws-sdk:serviceName:apiAction.[serviceIntegrationPattern]

<https://aws.amazon.com/about-aws/whats-new/2021/09/aws-step-functions-200-aws-sdk-integration/>

2022 Jan Local Mocking for Workflows

Create local mocking

Provide test cases for
All kinds of situations

CI/CD

```
[{"StateMachines": { "LocalTesting": { "TestCases": { "HappyPathTest": { "Check Identity": "CheckIdentityLambdaMockedSuccess", "Check Address": "CheckAddressLambdaMockedSuccess", "DetectSentiment": "DetectSentimentPositive", "Add to FollowUp": "AddToFollowUpSuccess", "CustomerAddedToFollowup": "CustomerAddedToFollowupSuccess" }, "NegativeSentimentTest": { "Check Identity": "CheckIdentityLambdaMockedSuccess", "Check Address": "CheckAddressLambdaMockedSuccess", "DetectSentiment": "DetectSentimentNegative", "NegativeSentimentDetected": "NegativeSentimentDetectedSuccess" } } } }]
```

<https://aws.amazon.com/about-aws/whats-new/2022/01/aws-step-functions-support-workflows/>

2022 May Observability Features

Advanced insights

Graph view Table view Event view

Table view

Filter by properties or search by keyword Filter by a date and time range Data flow simulator

	Name	Type	Status	Resource	Duration	Timeline	Started After
<input type="radio"/>	DynamoDB Get Shop st...	Task	✓ Succeeded	dynamodb	161 ms		38 ms
<input type="radio"/>	Shop Open?	Choice	✓ Succeeded	-	0 ms		199 ms
<input type="radio"/>	ListExecutions	Task	✓ Succeeded	aws-sdk:sfn	575 ms		199 ms
<input type="radio"/>	Is capacity available?	Choice	✓ Succeeded	-	0 ms		774 ms
<input checked="" type="radio"/>	Emit - Workflow Starte...	Task	⚠ Caught error	Eventbridge event bus	15 min		774 ms
<input type="radio"/>	Customer timeout	Pass	✓ Succeeded	-	0 ms		15 min
<input type="radio"/>	Emit - error timeout	Task	⌚ Not started		0 ms		15 min

<https://aws.amazon.com/about-aws/whats-new/2022/05/announcing-new-workflow-observability-features-aws-step-functions/>

2022 June Step Functions Workshop

AWS Workshops

This website lists workshops created by the teams at Amazon Web Services (AWS). Workshops are hands-on events designed to teach or introduce practical skills, techniques, or concepts which you can use to solve business problems. You can filter by topic using the toolbar above.



<https://aws.amazon.com/about-aws/whats-new/2022/06/aws-step-functions-interactive-workshop-building-deploying-application-workflows/>

- ▶ Module 1 - Hello World
- ▶ **Module 2 - Request Response**
- ▶ Module 3 - Run a Job (.sync)
- ▶ Module 4 - Wait for a Callback with the Task Token
- ▶ Module 5 - Choice State and Map State
- ▶ Module 6 - Input and Output Processing
- ▶ Module 7 - API Gateway, Parallel State, Express workflows
- ▶ Module 8 - Error Handling
- ▶ Module 9 - AWS SDK service integrations
- ▶ Module 10 - Deploy with AWS CDK
- ▶ Module 11 - Deploy with AWS SAM

Aug 2022 14 new intrinsic functions

Support for
220 AWS Services
10.000 API actions
18 intrinsic functions

Array
Json
Math
Strings
UUID
Template

```
States.Format('Hello, my name is {}.', $.name)
```

<https://aws.amazon.com/about-aws/whats-new/2022/08/aws-step-functions-14-new-intrinsic-features-process-data-workflows/>

Sep 2022 AWS Controllers for Kubernets (ACK)

Deploy AWS Resources from K8S with kubectl/helm

RDS,Lambda, managed Prometheus, KMS
S3, Sagemaker, Dynamodb, ECR, Api Gateway

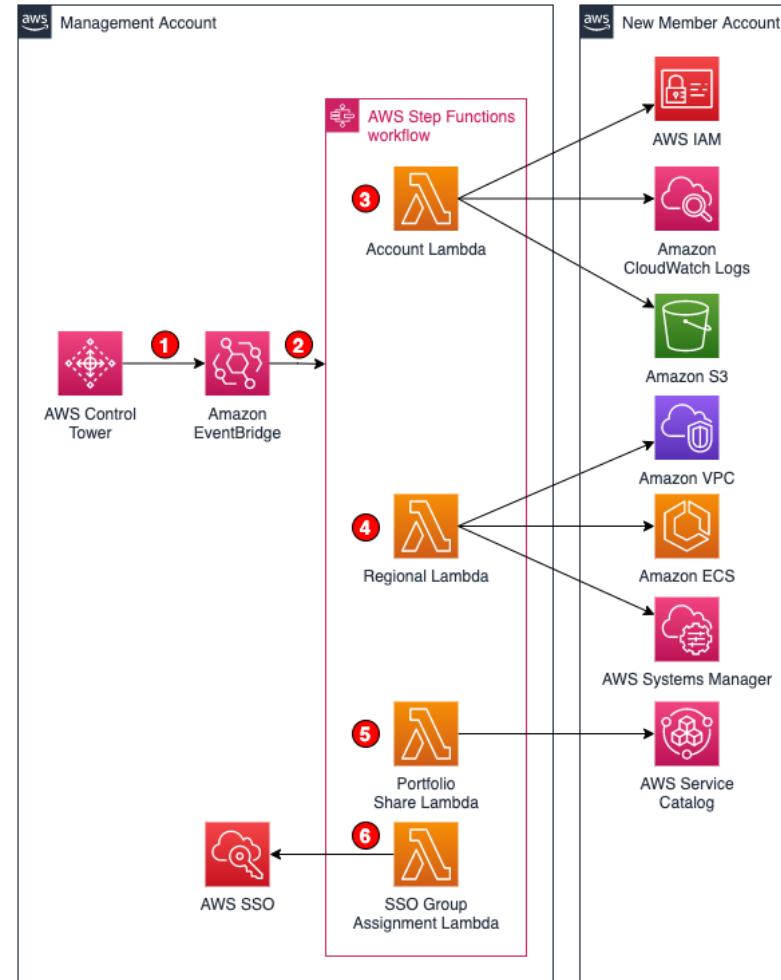
```
apiVersion: sfn.services.k8s.aws/v1alpha1
kind: StateMachine
metadata:
  name: $STATE_MACHINE_NAME
spec:
  name: $STATE_MACHINE_NAME
  roleARN: $SFN_EXECUTION_ROLE_ARN
  definition: "{ \"StartAt\": \"HelloWorld\", \"States\": { \"HelloWorld\": { \"Type\": \"Pass\", \"Result\": \"Hello World!\", \"End\": true }}}"
  tags:
    - key: k1
      value: v1
    - key: k2
      value: v2
    - key: k3
      value: v3
```

<https://aws.amazon.com/about-aws/whats-new/2022/09/aws-controllers-kubernetes-ack-rds-lambda-step-functions-prometheus-kms/>



Example: Account Setup

Run a set of actions
After account creation Event is
received



<https://github.com/aws-samples/aws-control-tower-account-setup-using-step-functions>

Functionless

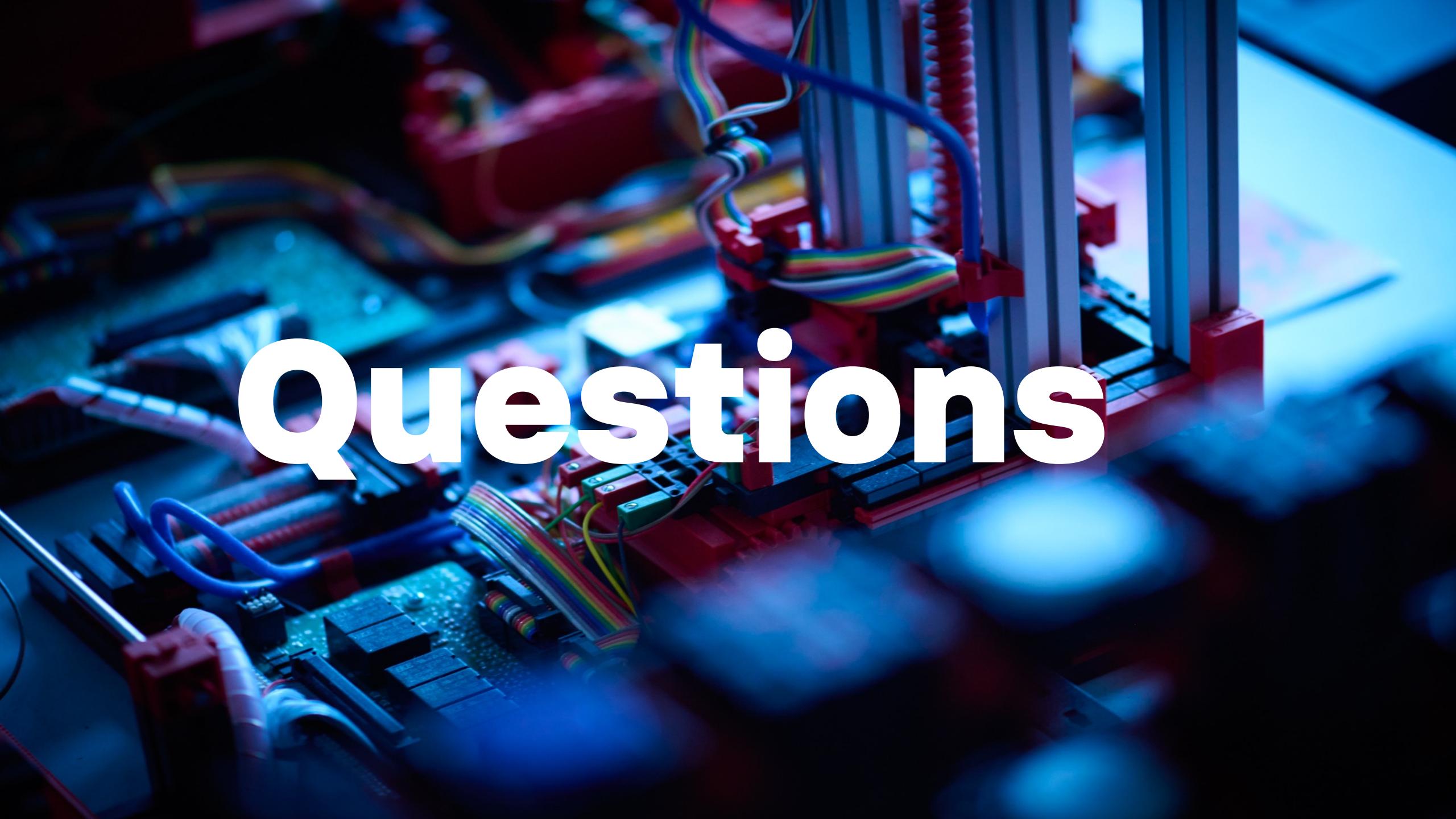
CDK Extension to infer domain specific Languages
Amazon State Language, Velocity Templates, Event Bridge Patterns

```
// Step Function workflow that validates the contents of a Post and deletes it if bad
const validatePostWorkflow = new StepFunction(
  this,
  "ValidatePostWorkflow",
  async (post: Post) => {
    const validationResult = await validatePost(post);
    if (validationResult.status === "Not Cool") {
      await $AWS.DynamoDB.DeleteItem({
        Table: postTable,
        Key: {
          postId: {
            S: post.postId,
          },
        },
      });
    }
  }
);
```

Resume

Conclusions





Questions