

AWS  
re:Invent

**SVS340-R**

# Serverless image processing workflows at scale with AWS Step Functions

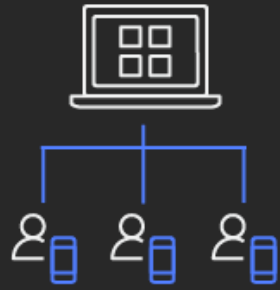
## **Ben Smith**

Senior Developer Advocate  
AWS Serverless  
Amazon Web Services

## **Rob Sutter**

Senior Developer Advocate  
AWS Serverless  
Amazon Web Services

# Agenda



The workshop environment

---



The modules and services used

---



Coding

# General housekeeping

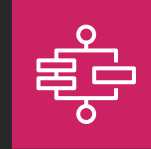
- Running time is 2h 15m
- AWS account will be provided – please use the access code
- This account is temporary
- Questions? We are here to help!

# Prerequisites

- Workshop cannot be completed with a tablet
- Compatible with Chrome/Firefox (not Safari)
- Familiarity with the AWS Management Console

# The AWS services you will be using today

## Orchestration layer



**AWS  
Step Functions**

---

## Compute layer



**AWS  
Lambda**

---

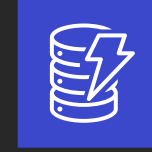
## Services layer



**Amazon  
Rekognition**

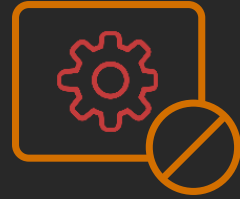


**Amazon S3**



**Amazon  
DynamoDB**

# What is serverless?

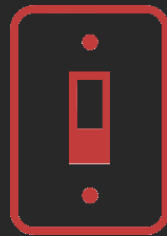


No infrastructure provisioning,  
no management



Automatic scaling

Pay for value



Highly available and secure



Event-driven compute

Functions as a service

Serverless FaaS





# Serverless applications

Event source



Function



Services



Changes in data state



Requests to endpoints



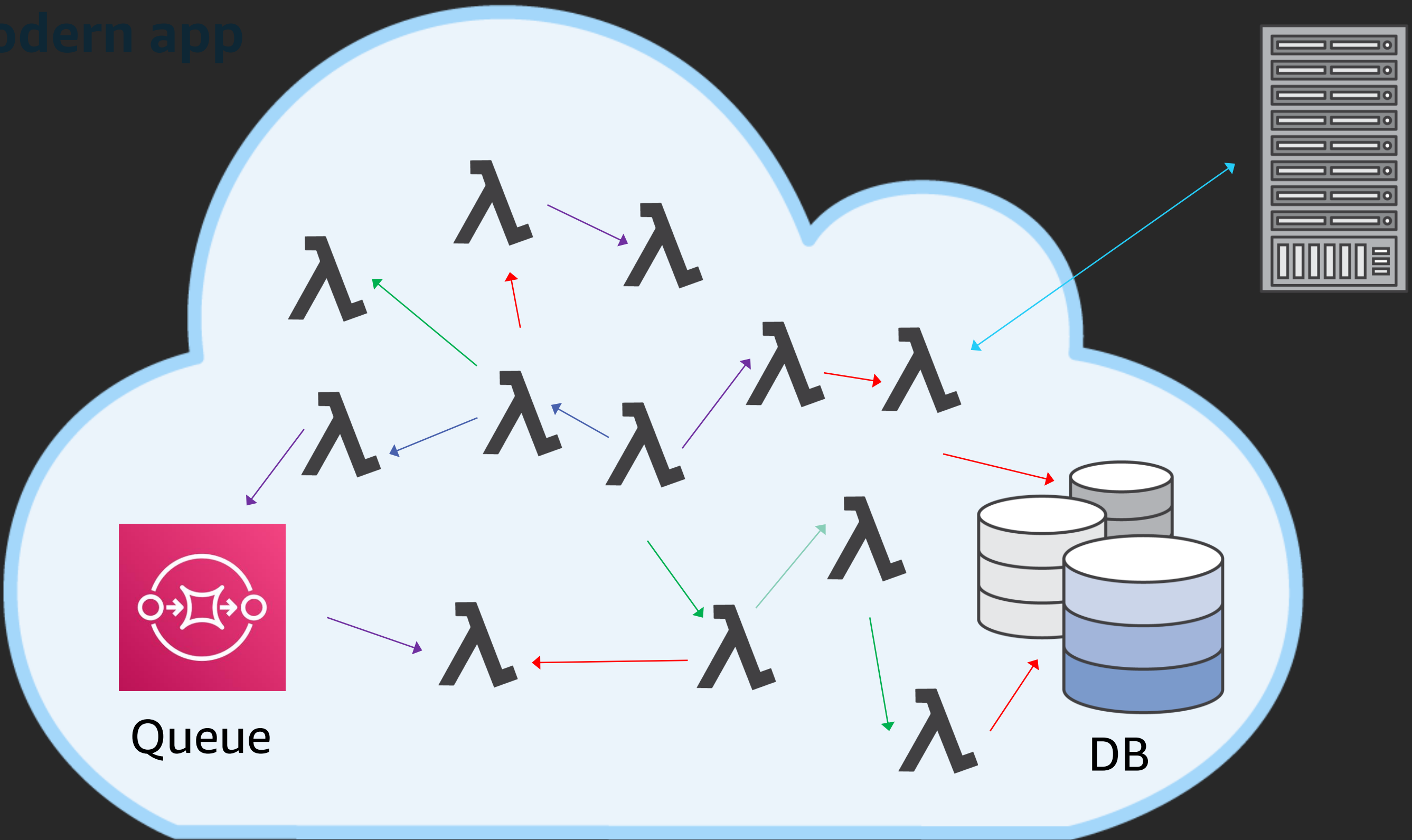
Changes in resource state

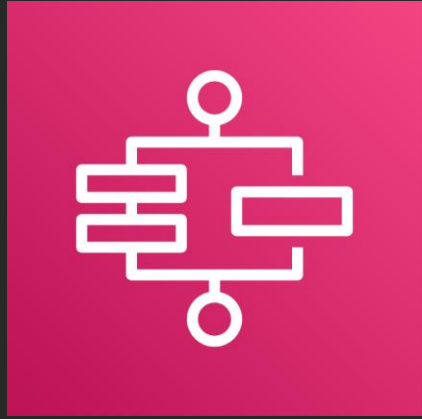


Node.js  
Python  
Java  
C#  
Go  
Ruby  
Runtime API

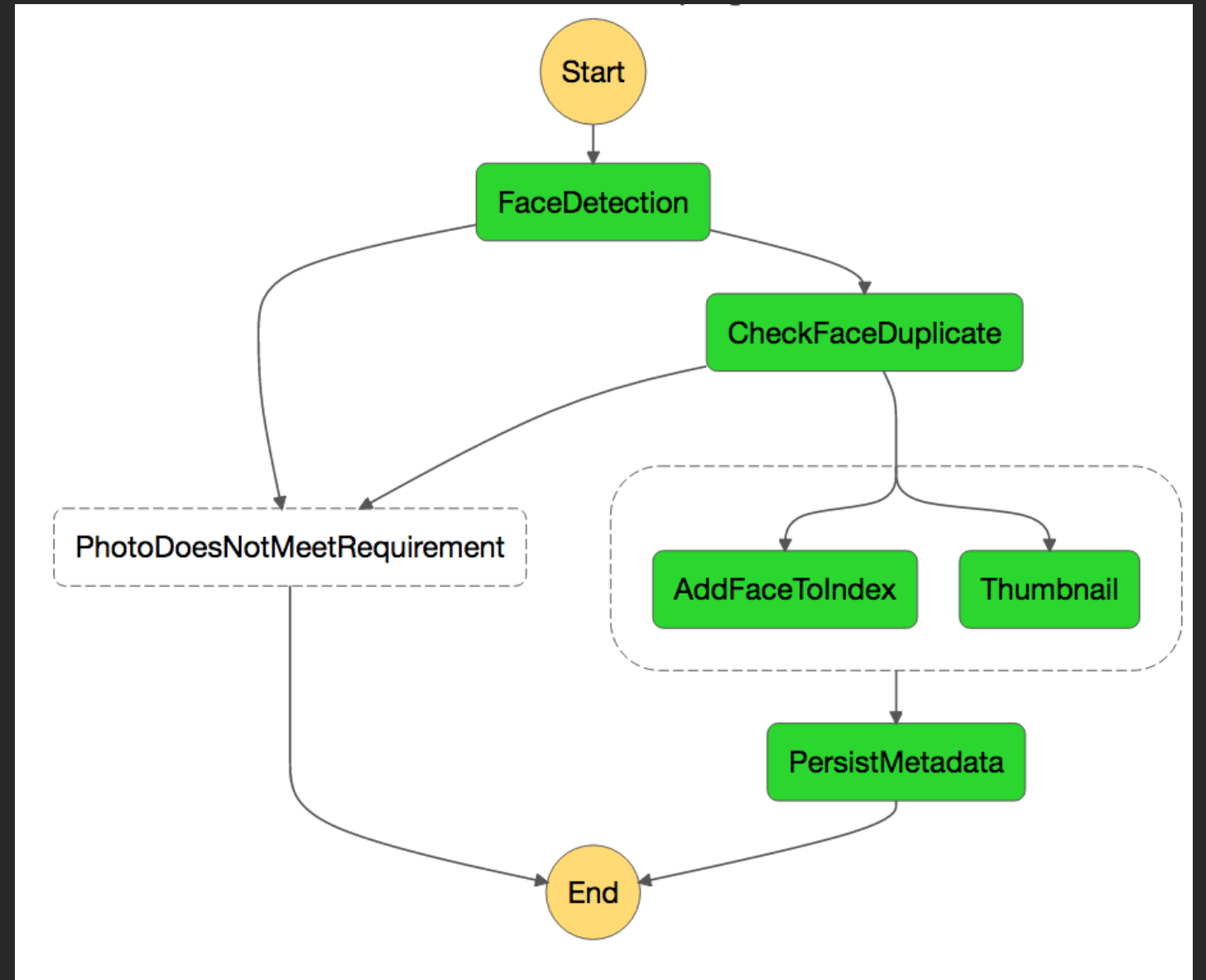


# Modern app





# AWS Step Functions

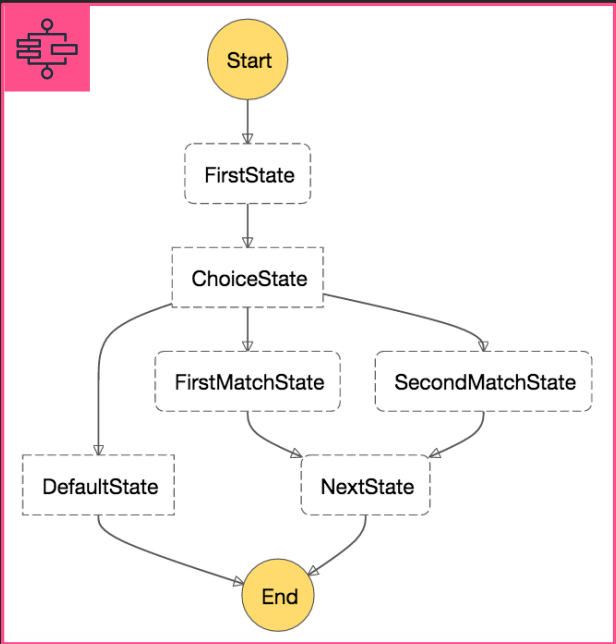


# Step Functions: Visual workflows

Define in JSON

```
Code
1 {
2   "Comment": "An AWL example using a choice state.",
3   "StartAt": "FirstState",
4   "States": {
5     "FirstState": {
6       "Type": "Task",
7       "Resource": "arn:aws:lambda:REGION:ACCOUNT_ID:function:FUNCTION_NAME",
8       "Next": "ChoiceState"
9     },
10    "ChoiceState": {
11      "Type": "Choice",
12      "Choices": [
13        {
14          "Variable": "$?.OrderType",
15          "Match": "Equals",
16          "Next": "FirstMatchState"
17        },
18        {
19          "Variable": "$?.OrderType",
20          "Match": "NotEquals",
21          "Next": "SecondMatchState"
22        },
23        "Default",
24        "DefaultState"
25      ]
26    },
27    "FirstMatchState": {
28      "Type": "Task",
29      "Resource": "arn:aws:lambda:REGION:ACCOUNT_ID:function:FUNCTION_NAME",
30      "Next": "NextState"
31    },
32    "SecondMatchState": {
33      "Type": "Task",
34      "Resource": "arn:aws:lambda:REGION:ACCOUNT_ID:function:FUNCTION_NAME",
35      "Next": "NextState"
36    },
37    "DefaultState": {
38      "Type": "Task",
39      "Resource": "arn:aws:lambda:REGION:ACCOUNT_ID:function:FUNCTION_NAME",
40      "Next": "NextState"
41    },
42    "NextState": {
43      "Type": "Task",
44      "Resource": "arn:aws:lambda:REGION:ACCOUNT_ID:function:FUNCTION_NAME",
45      "Next": "End"
46    },
47    "End": {
48      "Type": "End"
49    }
50  }
51 }
```

Visualize in the console



Monitor executions

Dashboard > Orderer > New\_Order Execution Arn: arn:aws:states:eu-central-1:492419596455:execution:Orderer:New\_Order

### New\_Order

Graph Code

Execution Status: Success Failed Needs retry In progress

Execution Details

Info Input Output

Execution Status: Succeeded

State Machine Arn: arn:aws:states:eu-central-1:492419596455:stateMachine:Orderer

Execution ID: arn:aws:states:eu-central-1:492419596455:execution:Orderer:New\_Order

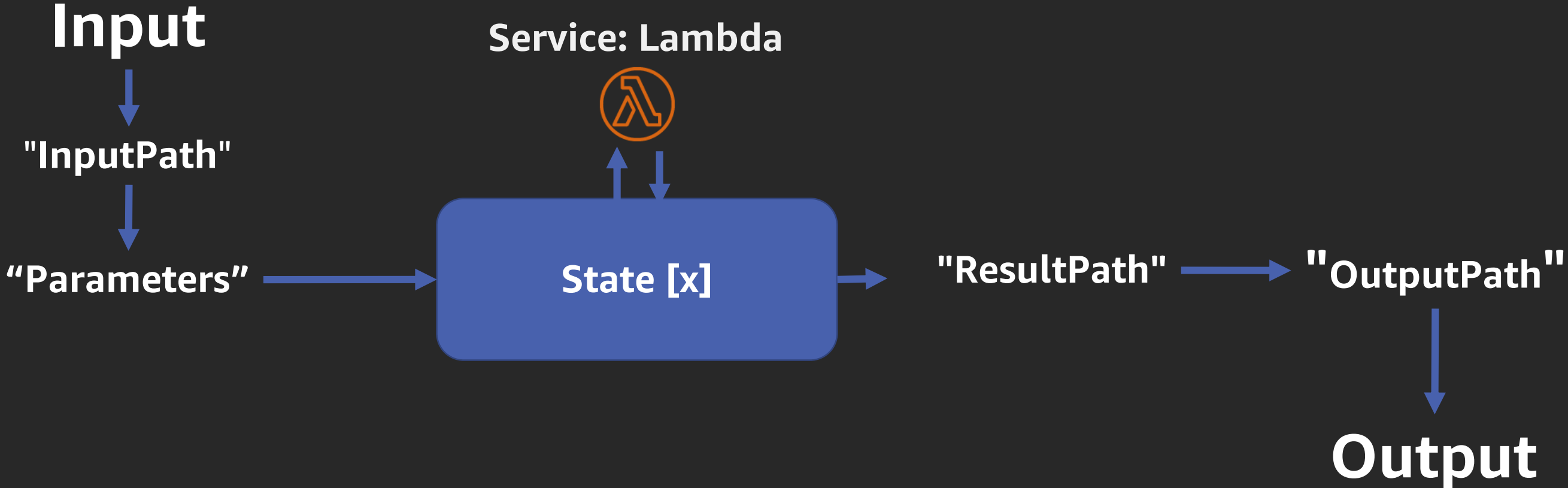
Started: Nov 20, 2016 9:58:28 AM

Closed: Nov 20, 2016 9:58:32 AM

Step Details

ID	Type	Timestamp
1	ExecutionStarted	Nov 20, 2016 9:58:28 AM
2	TaskStateEntered	Nov 20, 2016 9:58:28 AM
3	LambdaFunctionScheduled	Nov 20, 2016 9:58:28 AM

# Step Functions: Anatomy of state

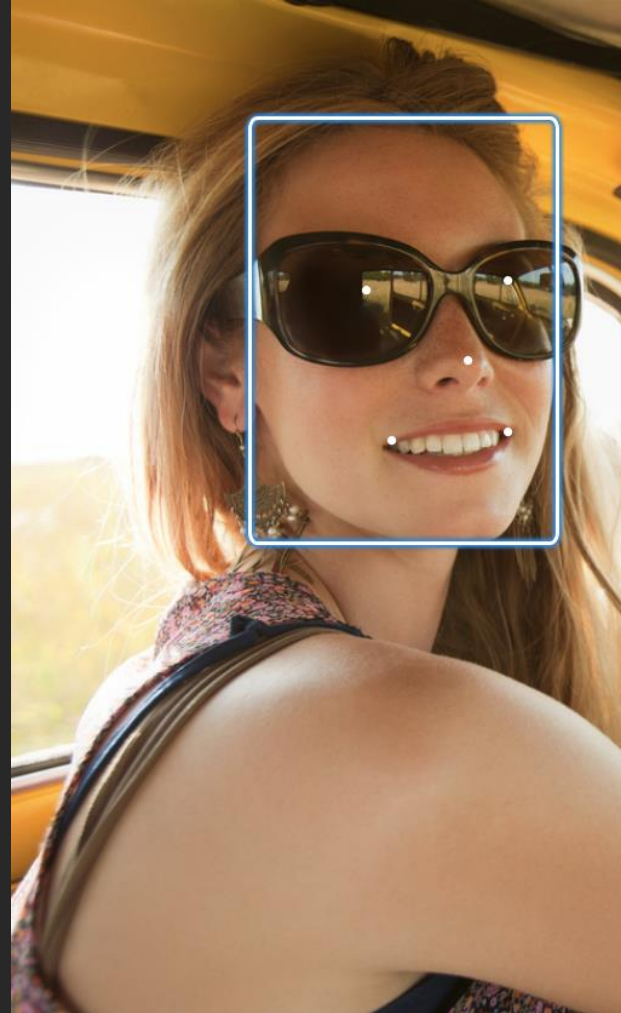


InputPath and OutputPath processes JSON


ResultPath places results into JSON

# Amazon Rekognition overview

Deep learning–based visual analysis service; search, verify, and organize millions of images and video

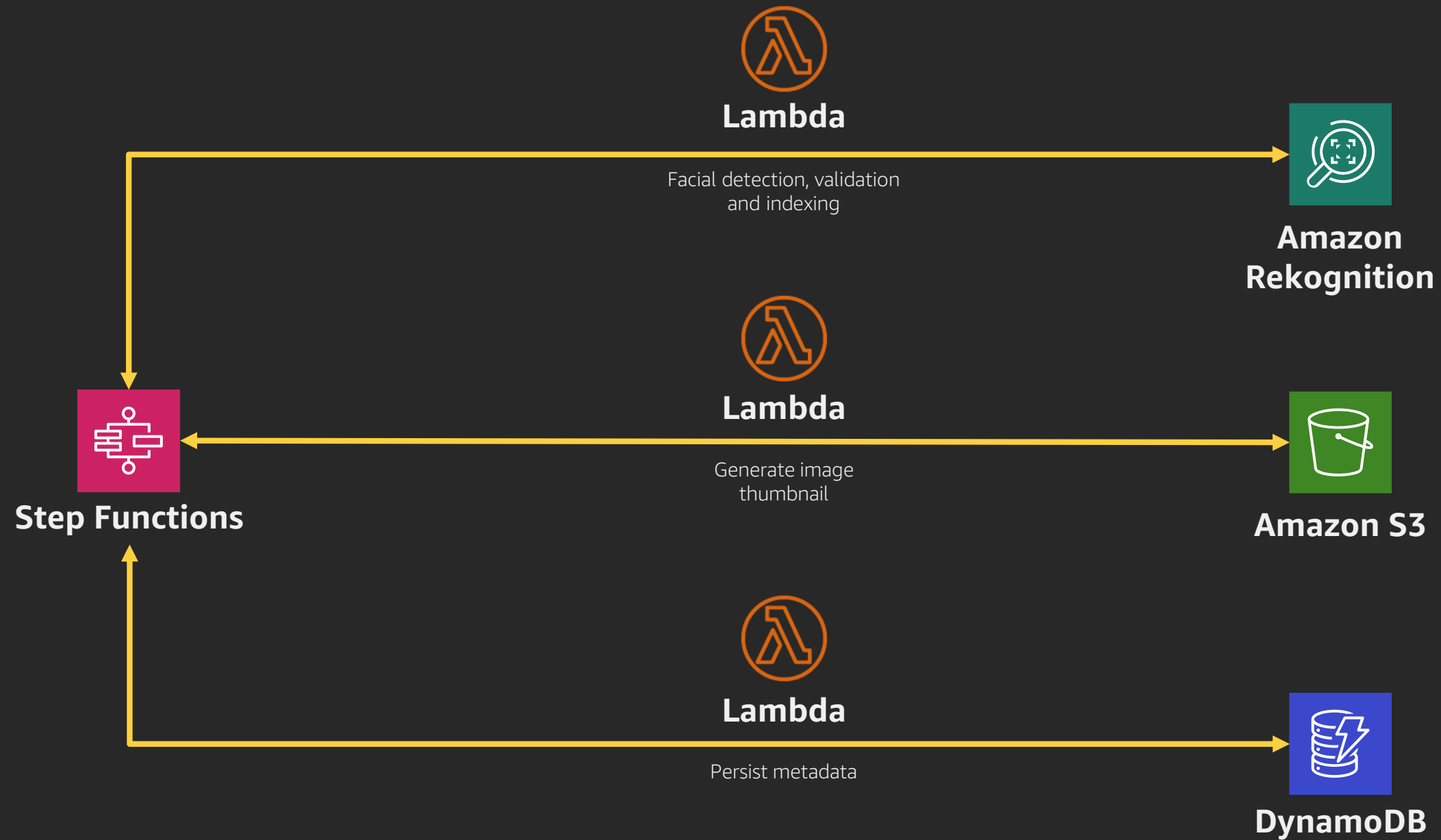


▼ Results

	
looks like a face	99.9 %
appears to be female	99.9 %
age range	17 - 29 years old
smiling	91.7 %
appears to be happy	99.5 %
wearing glasses	99.8 %

# Workshop overview

# The architecture





# Today's adventure

- **Module 1: Set up**
  - Event Engine, AWS Cloud9
- **Module 2: Implementation**
  - Face collection, state machine, de-duplication, parallel processing, database
- **Module 3: Validation**
  - Upload and test
- **Module 4: Extra credit**
  - Sending alert notifications

# AWS Cloud9



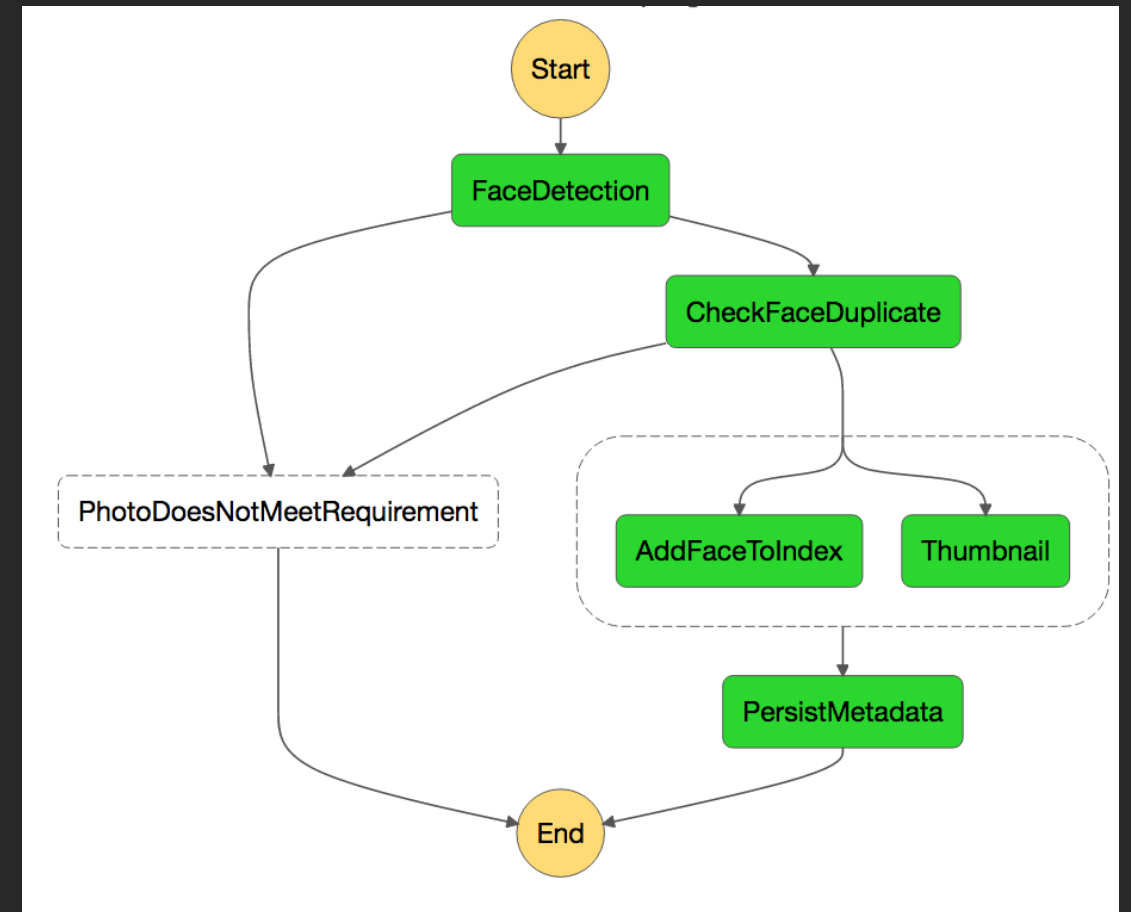
- Fully featured cloud-based IDE
- Integrated tools for serverless development
- Built-in terminal and local storage
- Provisioned bandwidth in AWS network (not dependent on Wi-Fi)
- This workshop is 100% browser-based – please do not use your local IDE, or terminal

# Today's adventure

- **Module 1: Set up**
  - Event engine, AWS Cloud9
- **Module 2: Implementation**
  - Face collection, state machine, de-duplication, parallel processing, database
- **Module 3: Validation**
  - Upload and test
- **Module 4: Extra credit**
  - Sending alert notifications

# Module 2 – Implementation

- Create a face collection
- Deploy the cloud formation stack
- Build the Step Functions state machine:
  - Face detection
  - Checking for missing/duplicates
  - Parallel processing
  - Saving metadata



# Today's adventure

- **Module 1: Set up**
  - Event engine, AWS Cloud9
- **Module 2: Implementation**
  - Face collection, state machine, de-duplication, parallel processing, database
- **Module 3: Validation**
  - Upload and test
- **Module 4: Extra credit**
  - Sending alert notifications

# Module 3 – Validation

- Test the final state machine:



Thumbnails generated?



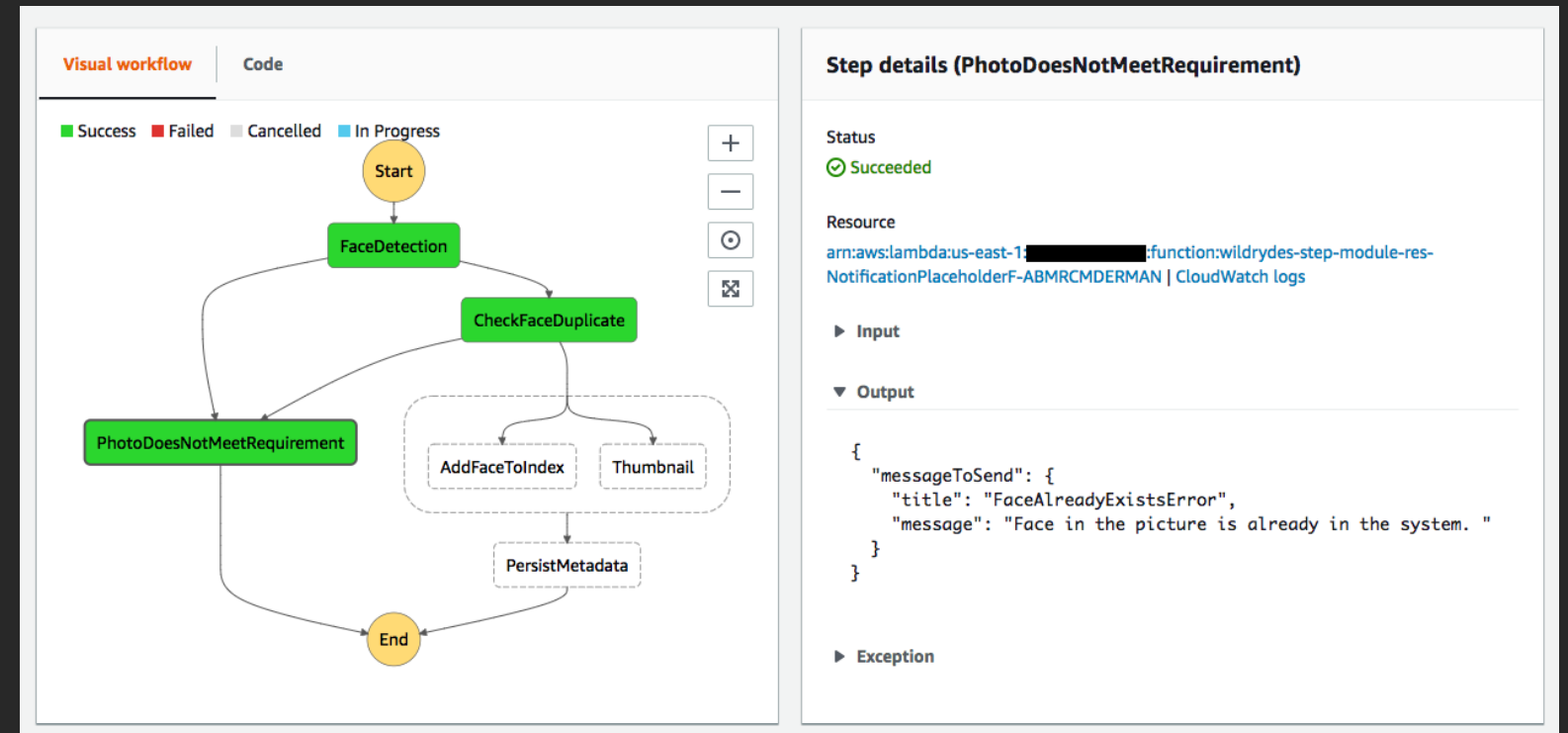
Database entry created?



Duplicates caught?



Input/output correct?

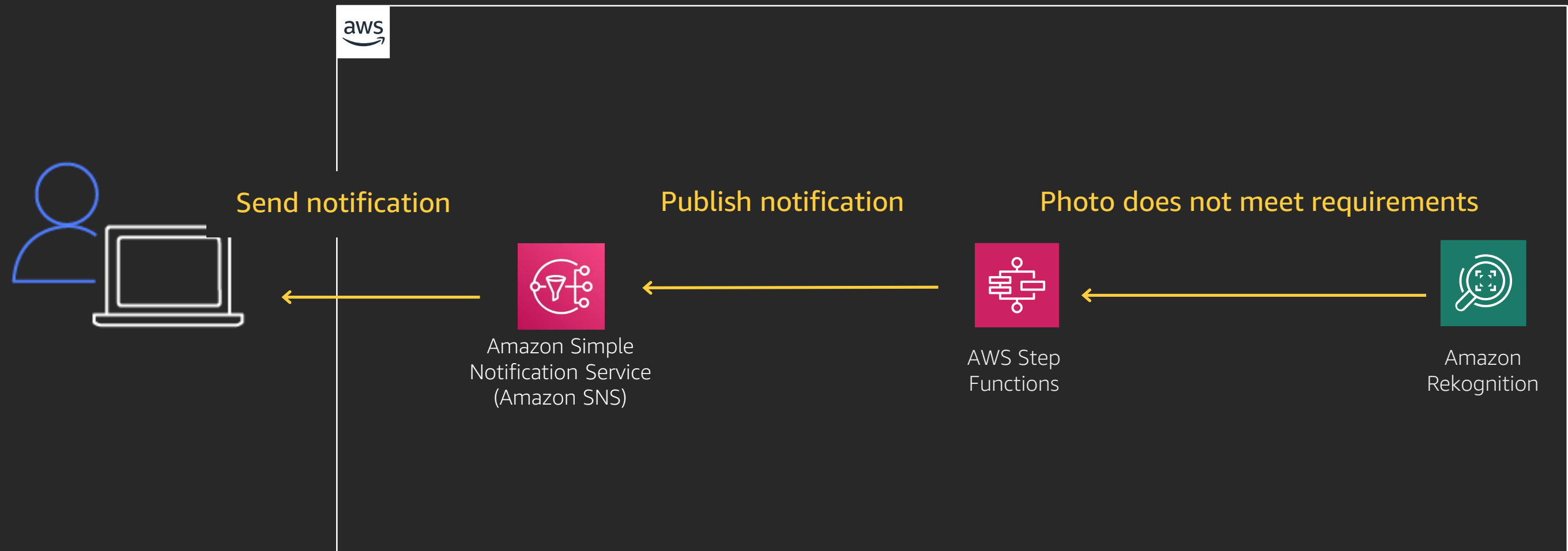


# Today's adventure

- **Module 1: Set up**
  - Event engine, AWS Cloud9
- **Module 2: Implementation**
  - Face collection, state machine, de-duplication, parallel processing, database
- **Module 3: Validation**
  - Upload and test
- **Module 4: Extra credit**
  - Sending alert notifications

# Module 4a – Extra credit

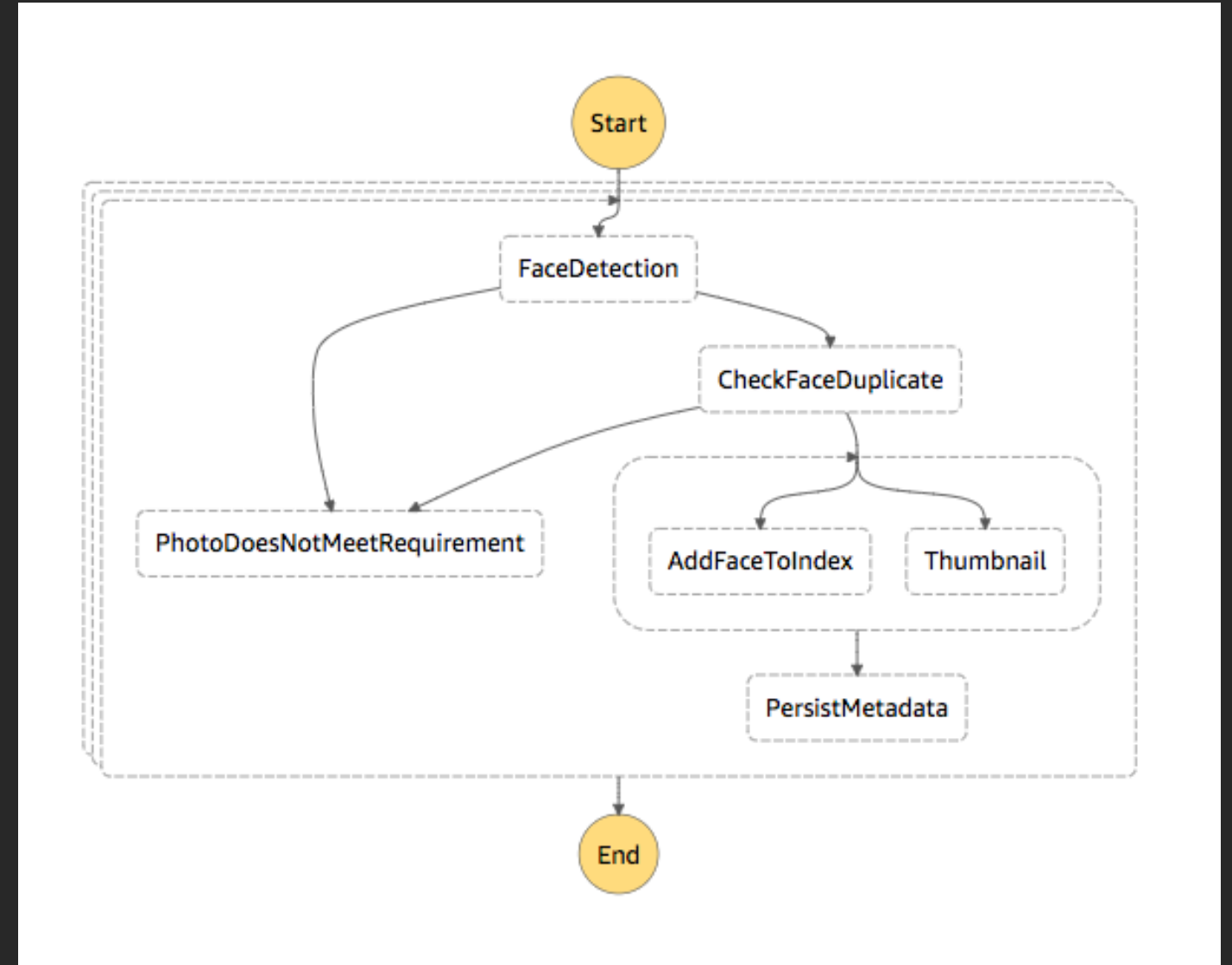
## Send error notification to user





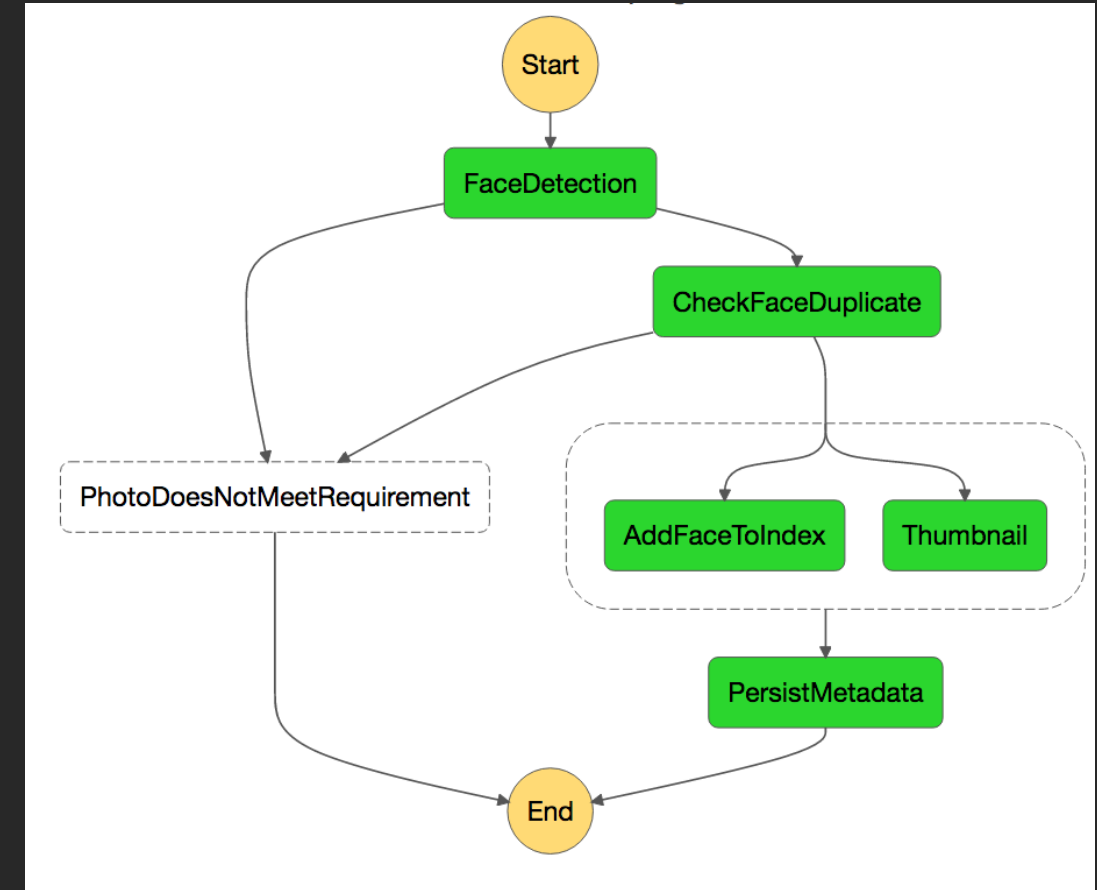
# Module 4b – Extra credit

Dynamically scaled state machine using the map state



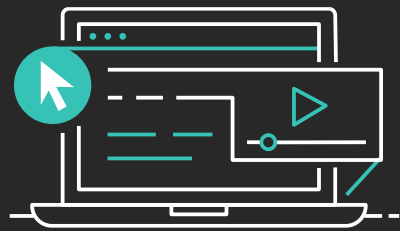
**Start your adventure:**

**<https://image-processing.serverlessworkshops.io>**



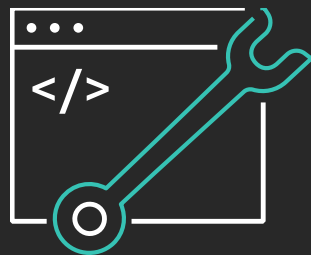
# Learn serverless with AWS Training and Certification

Resources created by the experts at AWS to help you learn modern application development



Free, on-demand courses on serverless, including

- Introduction to Serverless Development
- Getting into the Serverless Mindset
- AWS Lambda Foundations
- Amazon API Gateway for Serverless Applications
- Amazon DynamoDB for Serverless Architectures



Additional digital and classroom trainings cover modern application development and computing

Visit the Learning Library at <https://aws.training>

# Thank you!

## **Ben Smith**

benjasl@amazon.com  
@benjamin\_l\_s

## **Rob Sutter**

robsut@amazon.com  
@rts\_rob



Please complete the session survey in the mobile app.