

You may not need a Lambda.

Thomas Sabol
Sr. Solutions Architect
Lenus



Agenda

Introduction

Lambda use cases, what's new,
disadvantages

Path to serverless

Demo: Example of a functionless
application

Limitations, cost and when to use
Lambda

Key takeaways

Things I love

1. AWS Lambda

2. My girlfriend

3. Our dog



Lambda is AWSome



AWS Lambda

Run code without thinking
about servers or clusters

Responds to events

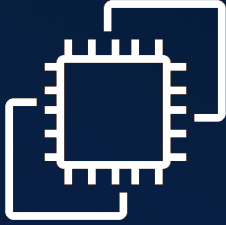
Automatically manages compute
resources

Cost efficiency

Quick deployments

Automatically handles scaling based on
incoming traffic

Total server abstraction



Customer Data

Application

Operating system

Networking & Firewall Configuration

Server-Side encryption

Compute, Storage, Networking

Hardware



Customer Data

Application

Operating system

Networking & Firewall Configuration

Server-Side encryption

Compute, Storage, Networking

Hardware



- You are responsible



- AWS is responsible

Common Lambda use-cases



API's



CloudFront edge
middleware



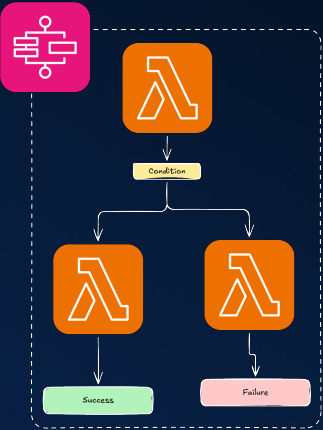
Schedule jobs



Message, Stream and
Event processing

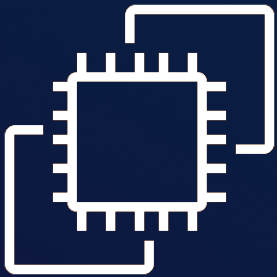


Retrieving and Uploading
Files



Workflows

What's new in Lambda



Managed Instances

Lambda on your own EC2
Many requests per instance
For steady, high volume



Durable functions

Checkpoints and resumes
Pause for up to one year
Python, Node.js, Java



MicroVMs

An isolated VM per user
Suspend, resume, keep state
For untrusted or AI code



Web Functions

Run a plain HTTP server
Streaming, scale to zero
Node.js only at launch

Event Functions

Web Functions **New**

Which kind of Lambda?

Event-driven logic



Event functions

Steady, high volume



Managed Instances

Existing web app



Web Functions

Untrusted or AI code



MicroVMs

Long-running flows



Durable functions

Disadvantages of AWS Lambda (Event Functions)

Cold start latency



Provisioned concurrency

Execution limit



Parallel processing, ECS Fargate Tasks

Unexpected costs



Rightsizing Resources, Compute Savings Plans, Lower Log Sampling Rate

Limited debugging



Structured Logging (Powertools), Log aggregation

Deployment of extra dependencies



Deployment tools (AWS CDK or SAM), Lambda Layers

Used as glue code between services



You may not even need it



Path to Serverless

Serverless manifesto advocates for:

- Resilience
- Security
- Pay per value pricing
- Ability to scale to 0
- Usage of managed services



Monolithic Lambda functions

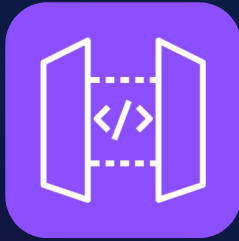


Serverless Microservices



Functionless

Core managed services



API Gateway

Serverless HTTP, REST
and WebSockets APIs



AppSync

Serverless GraphQL &
pub/sub APIs



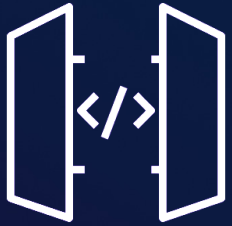
EventBridge

Serverless event router,
including rules,
schedules, pipes



Step Functions

Serverless workflow
orchestration service



API Gateway

Authorize, Validate,
Transform, Throttle,
Cache, Integrate

Request body validation with JSON
Schemas (REST API)

Required path, query string, and header
validation (REST API)

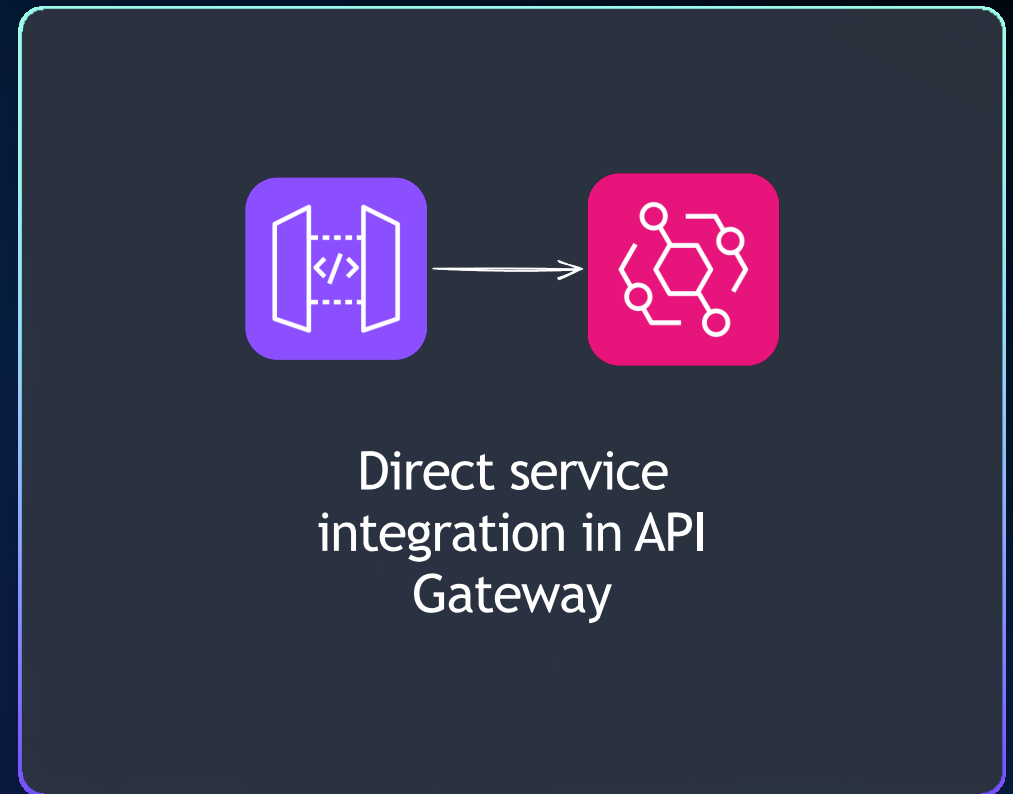
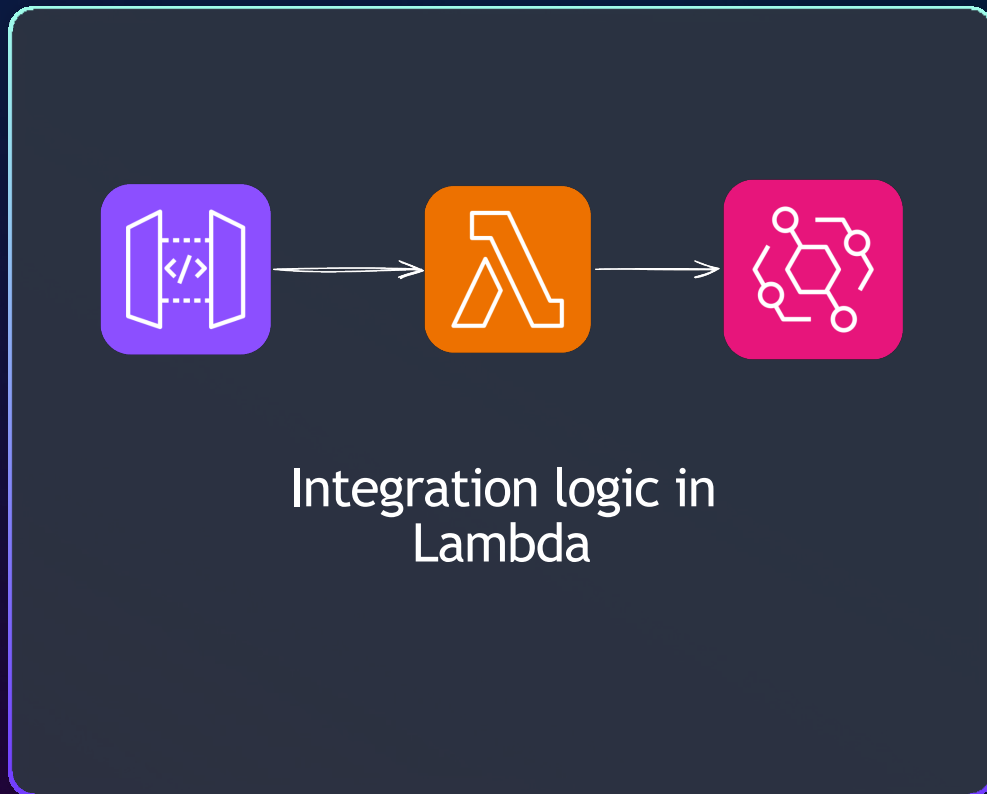
Route an event to EventBridge, SQS, SNS

Start a workflow with Lambda, Step
Functions or ECS RunTask

Do simple persistence with DynamoDB
or S3

Ingest data streams with Kinesis or
Firehose

Use Lambda for Logic, not glue



Use Lambda for Logic, not glue

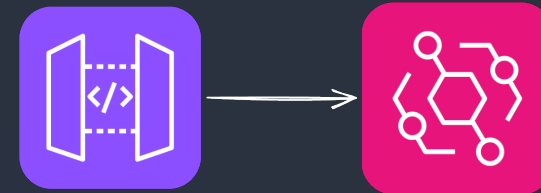
```
export const handler = async (
  event: APIGatewayProxyEventV2,
): Promise<APIGatewayProxyStructuredResultV2> => {
  try {
    const body = submissionSchema.safeParse(data: event.body)

    if (!body.success) {
      return {
        statusCode: 400,
        body: JSON.stringify(value: { message: "Invalid request body" }),
      }
    }

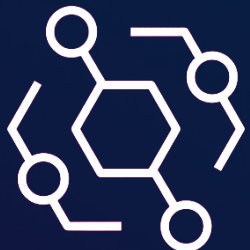
    const command = new PutEventsCommand(input: {
      Entries: [
        {
          EventBusName: process.env.EVENT_BUS_NAME,
          Source: "demo.api",
          DetailType: "submission.created",
          Detail: JSON.stringify(value: body.data),
        },
      ],
    })

    await eventBridgeClient.send(command)

    return {
      statusCode: 200,
      body: JSON.stringify(value: { message: "Submission created" }),
    }
  } catch (error) {
    return {
      statusCode: 500,
      body: JSON.stringify(value: { message: "Internal server error" }),
    }
  }
}
```



Direct service
integration in API
Gateway



EventBridge

Route, Filter, Transform,
Schedule and Integrate

Route events from AWS, your
applications and SaaS partners

Filter with event rules

Build point-to-point integrations
with event pipes

Create one time or recurring jobs
with event scheduler

Archive and replay events

Improve resilience with retries
and DLQs

EventBridge Integrations

Step Functions

Use when:

- The event triggers multiple steps
- Need workflow visibility
- Need retries, branching, wait steps, error handling

Examples:

- Payment processing
- Order processing
- Onboarding workflow
- Refund processing

SNS

Use when:

- Need simple pub/sub

Examples:

- Sending email
- Slack notifications
- Webhooks

CloudWatch

- Observability

SQS

Use when:

- Consumers might be slow or bursty
- Need retries controlled by queue
- Need batch processing

Examples:

- Image processing
- PDF generation

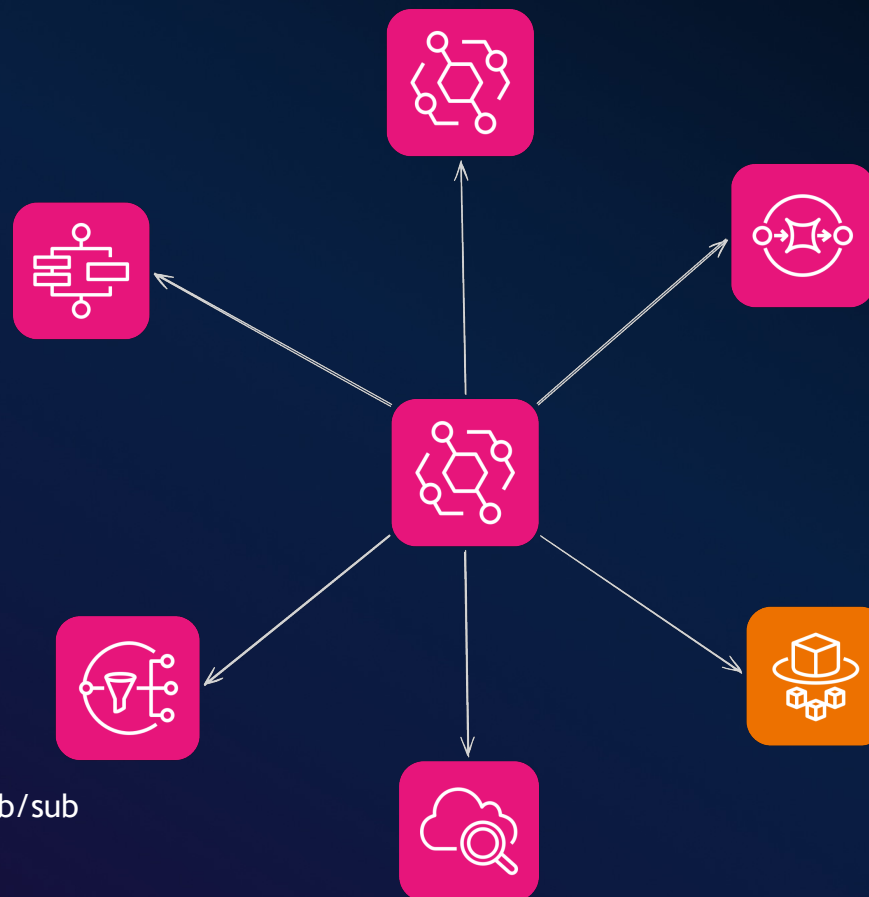
ECS Tasks

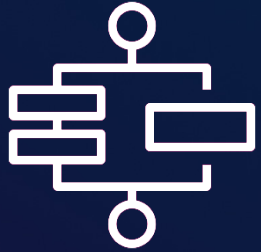
Use when:

- Processing long/heavy tasks

Examples:

- AI Agents
- Video conversion





Step Functions

Orchestrate, branch, wait,
retry, integrate

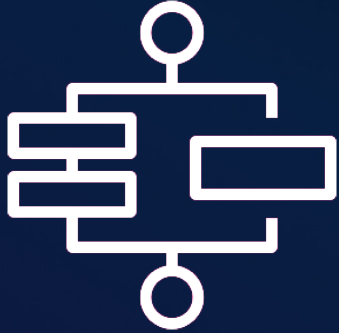
Orchestrate multi-step workflows

Direct integration with all 220+ AWS
Services

Visualize every step of the execution

Redrive from failures

Workflow types



Standard

Ideal for long-running workflows
Can run for up to one year
Full execution history for up to 90 days
Billed by state transitions (not duration)



Express

Ideal for high volume, event-processing workflows
Can run for up to five minutes
Billed by number of executions, duration of execution and the memory consumed

Integrate with everything



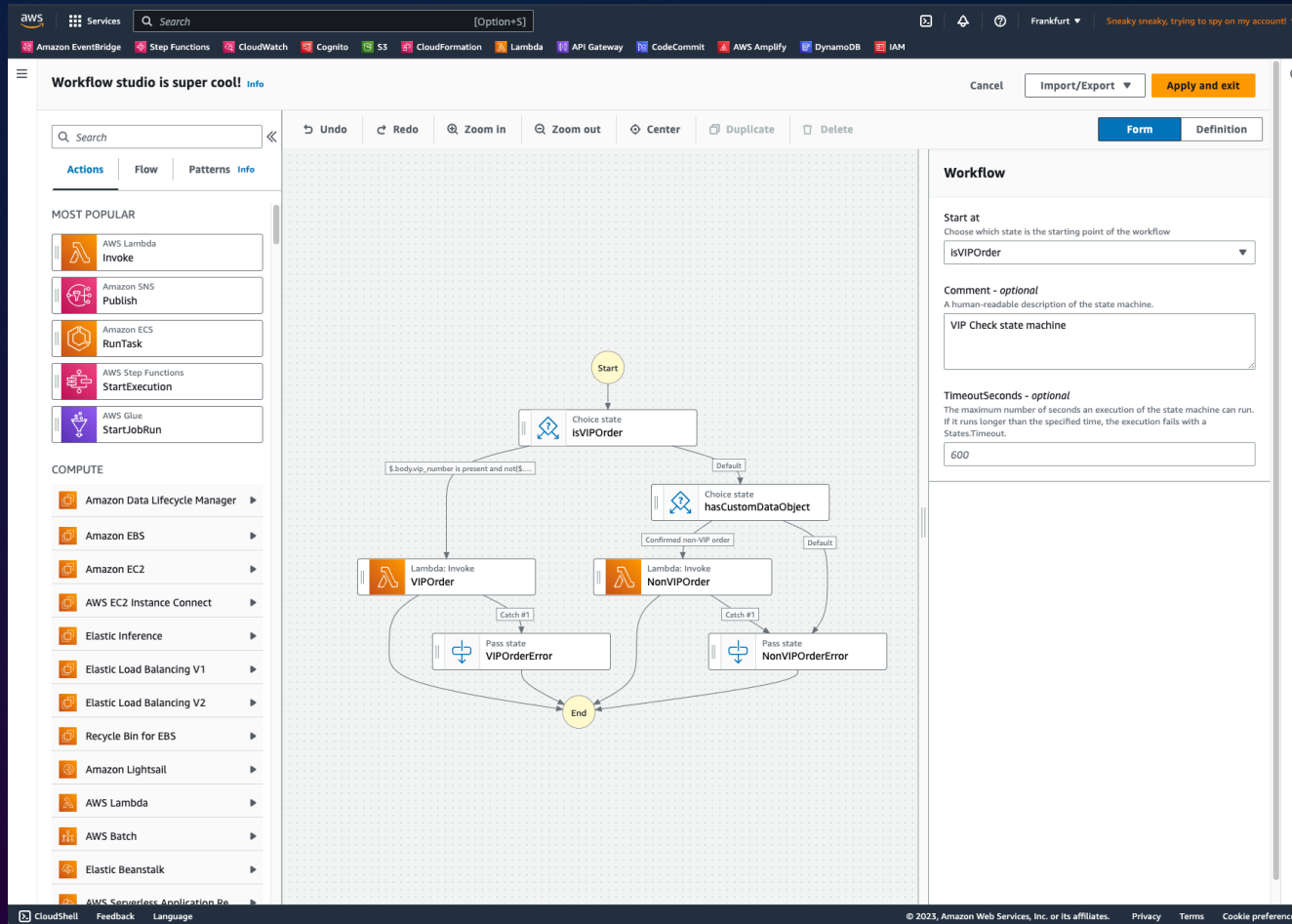
Integrations run directly in execution

Removes integration glue code

Familiar syntax

Use JSONata for simple data operations
in between steps

Workflow Studio



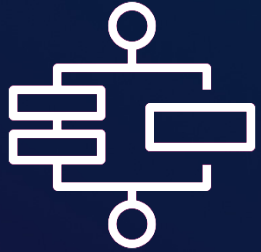
Visual workflow editor

Drag and drop

Auto generates code

Easy to review and export generated code

Export as JSON or YAML



Local testing

Test workflows in small pieces

Test one state at a time with the TestState API

Define everything as code with CDK or SAM

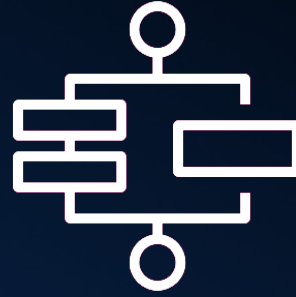
Iterate visually in Workflow Studio

Durable functions or Step Functions?



Durable functions

Workflow is plain code (JS, Python, GO, C#)



Step Functions

Workflow is a visual graph, defined in ASL

**Mostly gluing services? Step Functions.
Mostly your own code? Durable functions.**

DEMO: Return Processing With AI Triage

Example of a functionless application

Return Processing

A CUSTOMER RETURN PIPELINE FOR E-COMMERCE

Flow:

1. Return request arrives via API Gateway or EventBridge
2. DynamoDB checks eligibility and the return window
3. Rekognition analyzes customer images from S3
4. JSONata builds the decision model
5. Bedrock recommends: approve, review or reject
6. Translate localizes the notification
7. SNS notifies warehouse and support
8. DynamoDB stores the decision and history



Limitations of functionless

Complex logic
produces complex
workflows

Might be easier to
code in one of the
supported languages

Some services require
transforming JSON data
to the right format
(e.g. DynamoDB)

Harder local
development

Is functionless cheaper?

Lambda

Requests + duration. A glue function pays while it waits.

Step Functions

Standard: per state transition.
Express: per execution and duration.

Managed Instances

EC2 price plus a management fee. No per-duration charge.

Run the numbers for your volume.

So when do I still need Lambda?

Validate requests	➡	API Gateway
Route events	➡	EventBridge
Buffer or fan out	➡	SQS, SNS
Orchestrate steps	➡	Step Functions
Reshape JSON	➡	JSONata
Custom business logic	➡	Lambda

Key takeaways

If your Lambda only integrates AWS services and runs no custom logic, you may not need it

Keep functions small and single-purpose

Handle events with a variety of AWS services, not only Lambda

Orchestrate your applications with state machines, not functions

Thank you!

Thomas Sabol
tomassabol.com



Let's connect!



Rate this session



Community
Day **Malta**

