Meet Christine, Our Heroine

Noreen Hasan

Training Architect

LESSON BREAKDOWN

Meet Christine, Our Heroine

Christine’s Surprising Day

Lesson Summary

Noreen Hasan
Training Architect

Meet Christine, Our Heroine

Christine’s Surprising Day

— Placeholder —
A three minute animation will be added here

Lesson Summary

Meet Christine, Our Heroine

- Data engineer working on a startup

- Unpredictable spike in demand

- Cliffhanger: how will Christine respond to the spike?

Introduction to Unpacking Lambda

Noreen Hasan
Training Architect

SECTION BREAKDOWN

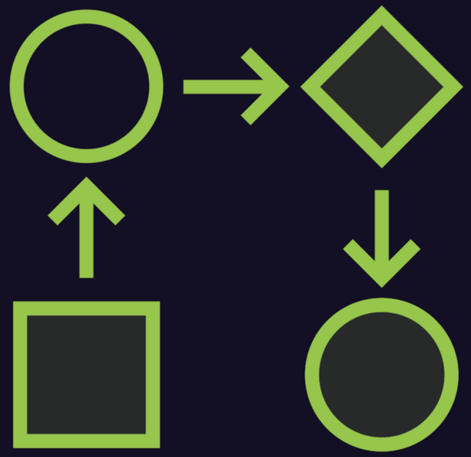
Section Overview

Introduction to Unpacking Lambda

Noreen Hasan
Training Architect

Section Overview

What Is Lambda?



How Lambda Works

Lambda Use Cases

Section Overview

Lambda is no

When to Use Lambda

Compare it against EC2, Elastic Beanstalk, and ECS

Accessing Lambda

CLI, console, SDK + demo

Let’s get started!

Unpacking Lambda: Recap

Noreen Hasan
Training Architect

SECTION BREAKDOWN

Lessons Summary

Unpacking Lambda: Recap

Noreen Hasan
Training Architect

Lessons Summary

What Is Lambda?

Event-driven serverless AWS service

Usages

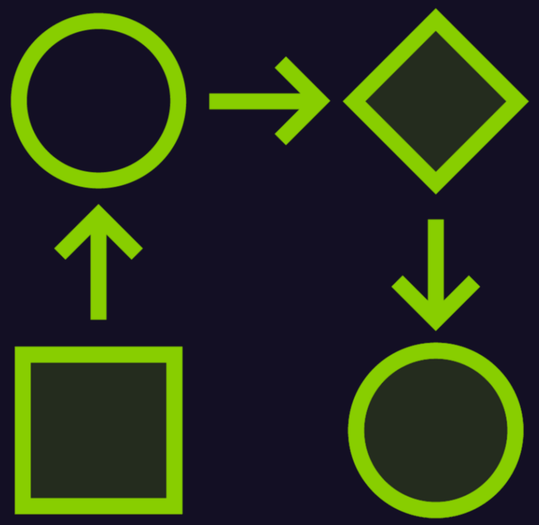
- IoT devices

- Serverless websites

High-Level Workflow

1. Create a **function**
2. Write or **upload code** to the function
3. Specify the **triggering event**

Lessons Summary



How Does Lambda Work?

Lambda gets invoked to run your code only when needed.

Main Building Blocks

1. Function
2. Trigger
3. Event
4. Execution Environment

Under the Hood

Load balancer and a fleet of EC2 instances

Lessons Summary

Lambda Use Cases

- Processing data
  - Processing S3 objects
  - Monitoring and analyzing logs
  - Processing streaming data
- Orchestrating workflows
- Operating serverless applications
- Processing messages

Lessons Summary

Lambda is no

Alternative Services

- AWS Elastic Beanstalk
- EC2
- ECS

When to Use Lambda

- If you want to focus more on the code and less on servers
- If your jobs take < 15 mins
- Never pay $ for idle time

Lessons Summary

Accessing Lambda

- AWS CLI
- AWS SDK
- AWS Lambda console

Lambda Demo

- Walkthrough of the console
- Create a Hello World application

See you in the next section!

What Is Lambda?

Noreen Hasan
Training Architect

LESSON BREAKDOWN

What Is Lambda?

Lambda in the Life of a Data Professional

Lambda and You

High-Level Workflow

Lesson Summary

Noreen Hasan
Training Architect

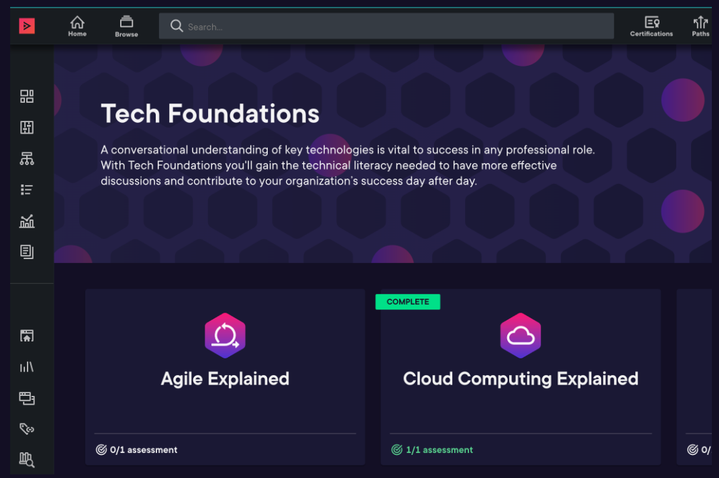
Lambda in the Life of a Data Professional

— Placeholder —
A three minute animation will be added here

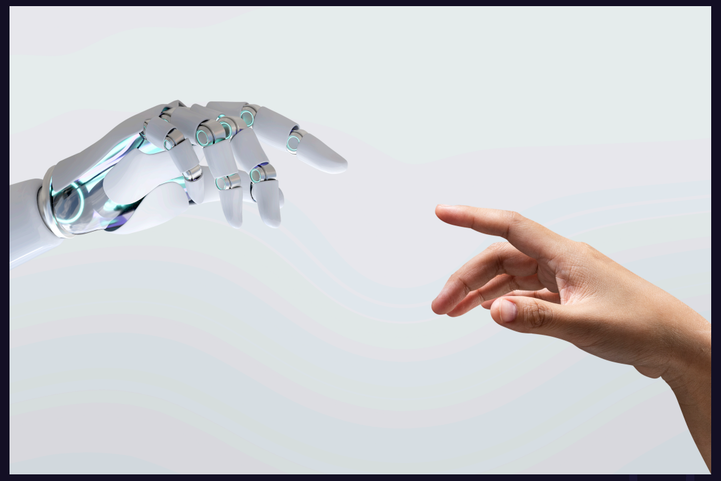
What Is Lambda?

Lambda and You

IoT Devices



Serverless Websites



Small Event Driven

Lambda and You

You're not in charge of provisioning the servers.

You're in charge of writing event-driven code.

You're in charge of persisting results.

Lambda and You

Amazon is in charge of everything hardware related.

Amazon is in charge of operating systems and security.

You pay for running code measured in milliseconds (ms).

High-Level Workflow

1. Create a function.
2. Write or upload code to the function.
3. Specify the triggering event.

High-Level Workflow

What?

Write data to DynamoDB.

What information to process?

The data inside the files uploaded to S3.

When?

Users upload to S3.

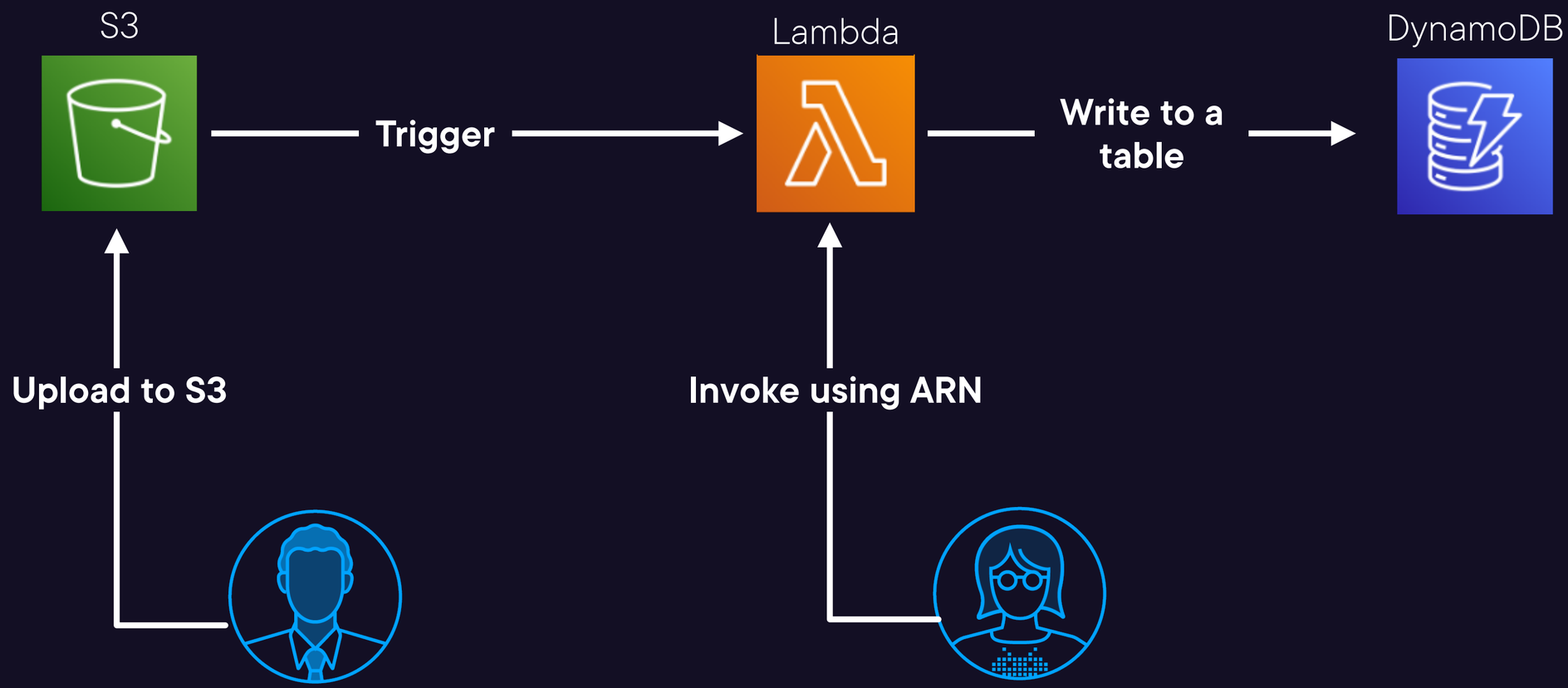
Where will the code run?

In an isolated runtime environment.

High-Level Workflow

Lambda Requirements

- The Lambda Function
- IAM Execution Role
- Memory Size Specification
- Execution Timeout
    - The limit is 15 mins
- Event Source Mapping


S3
Lambda
DynamoDB
Trigger
Write to a table
Upload to S3
Invoke using ARN

Lesson Summary

Lambda in the Life of a Data Professional

- Explored how Lambda serves Christine's startup

Lambda and You

- Everyday examples
- Your responsibilities in Lambda

High-Level Workflow

- High-level steps for working with Lambda
- Lambda requirements

How Lambda Works

Noreen Hasan
Training Architect

LESSON BREAKDOWN

How Lambda Works

- Overview of Lambda's Building Blocks
- First Block: Function
- Second Block: Trigger
- Third Block: Event
- Fourth Block: Execution Environment
- Under the Hood
- Lesson Summary

Noreen Hasan
Training Architect

How Lambda Works

Overview of Lambda's Building Blocks

Function

Trigger

Event

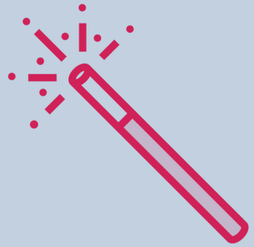
Execution environment

Overview of Lambda's Building Blocks

Function

A standalone unit of code that processes events

Trigger



A resource that invokes the function into a running state

Overview of Lambda's Building Blocks

Event

Relevant information that gets passed to functions to process the data

Execution Environment

An isolated and secure environment to run the function code

Overview of Lambda's Building Blocks

Function Code

What?

Write data to DynamoDB.

Trigger

When?

Users upload to S3.

Event

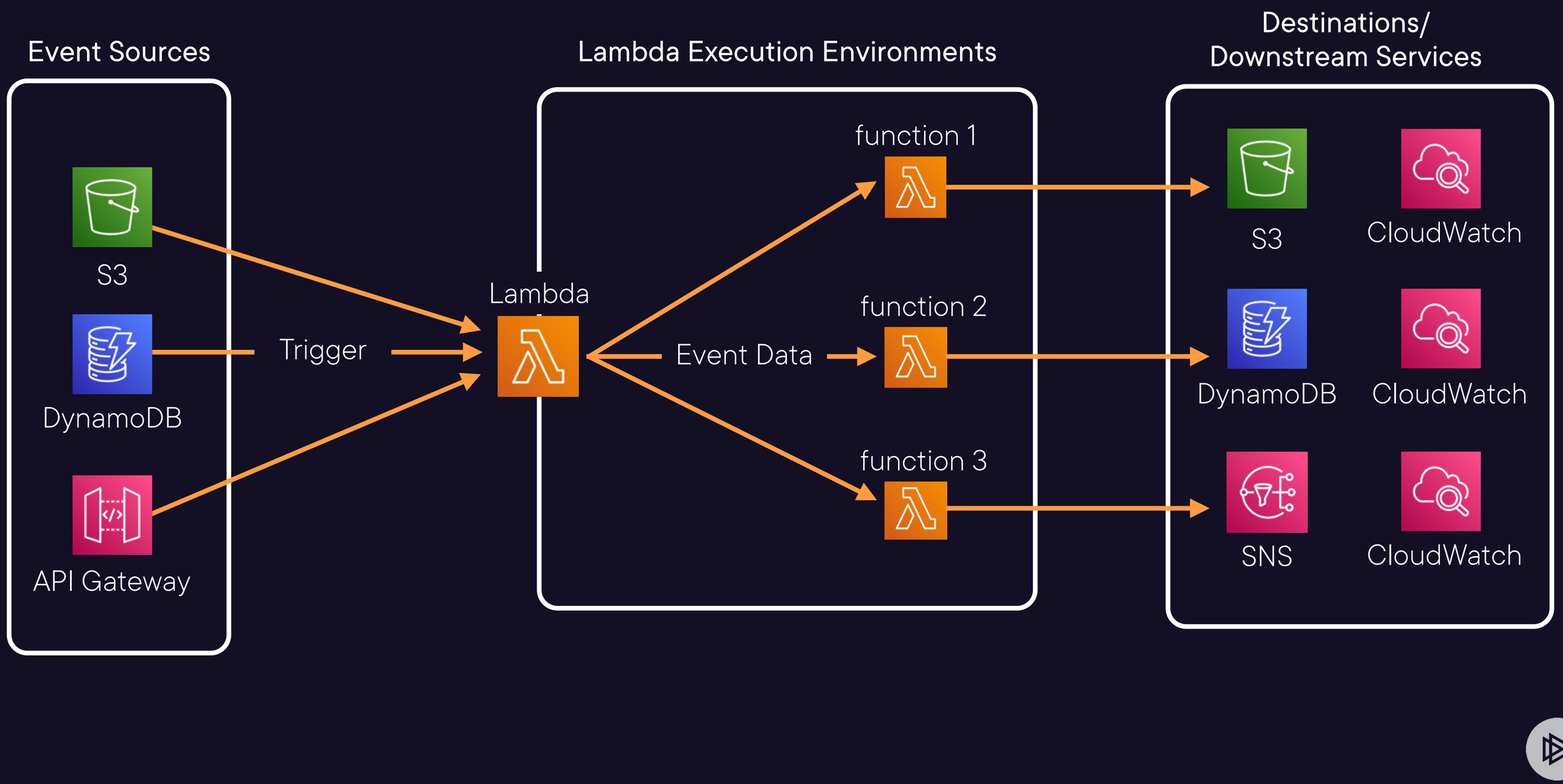
What information to process?

The data inside the files uploaded to S3.

Execution Environment

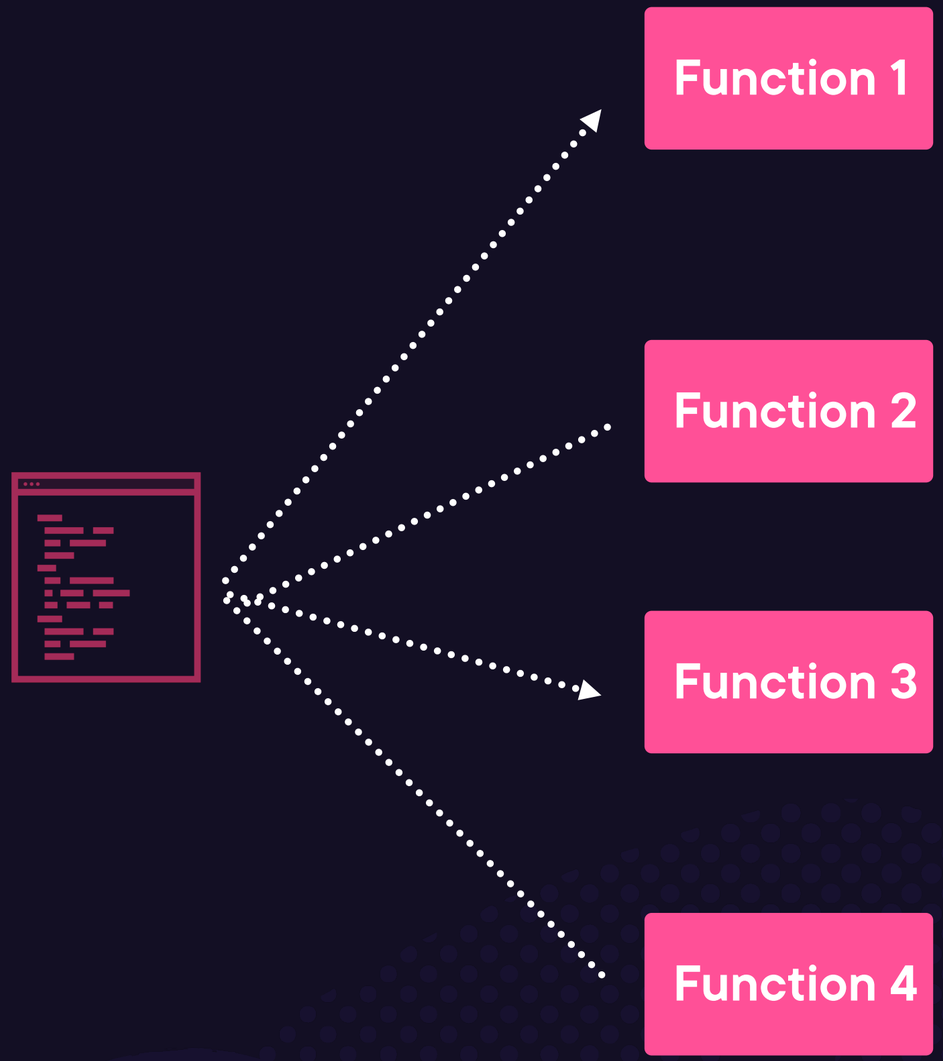
Where will the code run?

In an isolated runtime environment.


Event Sources
S3
DynamoDB
API Gateway
Trigger
Lambda Execution Environments
Lambda
Event Data
function 1
function 2
function 3
Destinations/
Downstream Services
S3
CloudWatch
DynamoDB
CloudWatch
SNS
CloudWatch

First Block: Function

In Lambda, code is organized and split into functions.



First Block: Function

You can configure your code to time out after a specific amount of time.

< 1 sec

< 3 secs

< 15 mins

First Block: Function

```
def handler_name(event, context)
. . .
return some_value
```

Lambda Function Handler

– A method to respond to events

Event handlers take two parameters:

– Event

– Context

How Lambda Works

First Block: Function

(Python)

lambda_function.py

```
import logging
import json
```

libraries

handler name

parameters

```
def lambda_handler(event, context)
    message = 'You might like the course {}'.format(event['courseName'])
    return {
            'message': message
            }
```

Parameters:

1. Event: has the details (usually type = dict) of the event that triggered the function.

2. Context: has the runtime information (type = context).

Second Block: Trigger

Trigger

In the classic tale from *The Arabian Nights*, the thieves hid their treasures in a cave behind an invisible door.

Uttering the magical words “Open, Sesame” was the event used to trigger the door to open.

Second Block: Trigger

Anything configured to invoke a Lambda function.

Other AWS Services

- S3 Upload
- Alexa Skill

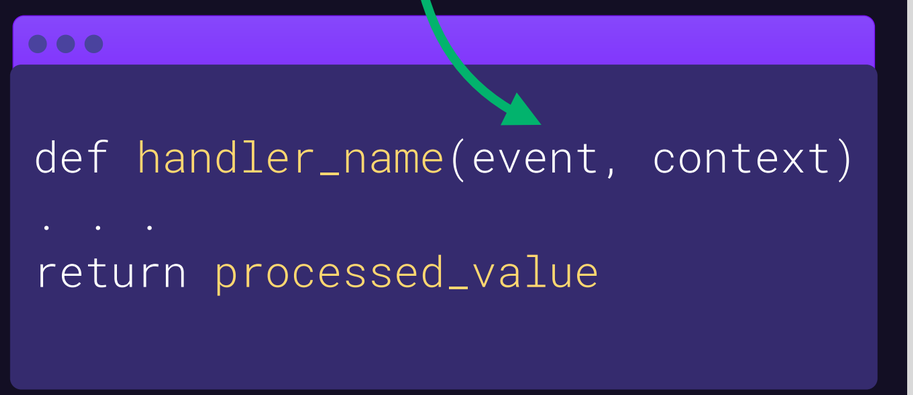
Lambda Resources

- The Lambda API
- The Lambda Console
- Event Source Mapping (to process queued items)

Functions can have multiple triggers.

Third Block: Event

```
{
  "FirstName": "Christine",
  "LastName": "Hernandez"
}
```




```
{
 "key":"CourseThumbnail.jpg",
 "size":1024,
 "eTag":"d41d8cd9f0b204e980ec",
 "versionId":"09fKTRTt9fV.nflo"
}
```

Events are JSON-formatted documents.

- They get passed into functions to pass in the relevant data.
- The runtime converts the document into an object.
- The structure and content depends on the invoking service.

Third Block: Event

How is an event object created?

lambda_function.py

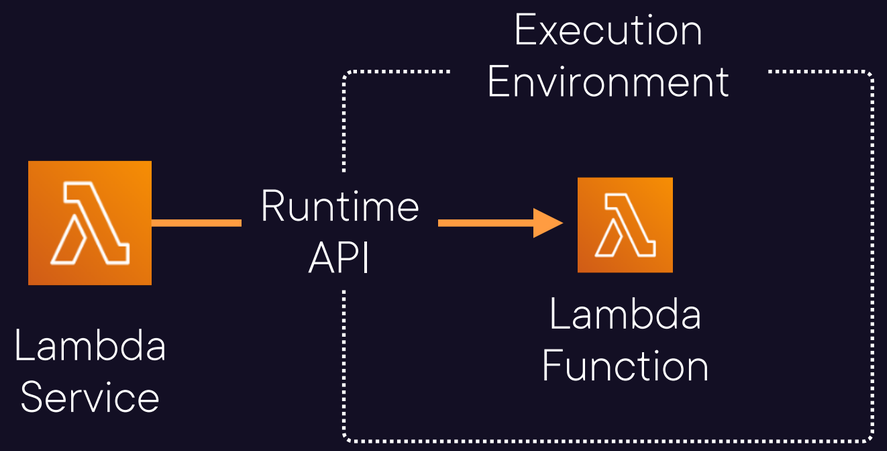
```
import pandas


def lambda_handler(event, context)
   message = 'You might like the course - {} -'.format(event['courseName'])
   return {
         'message': message
         }
```

```
{
   "Course": "Processing Serverless Data Using Lambda"
}
```

```
{
   "message":  "You might like the course — Processing Serverless Data Using Lambda"
}
```

Fourth Block: Execution Environment



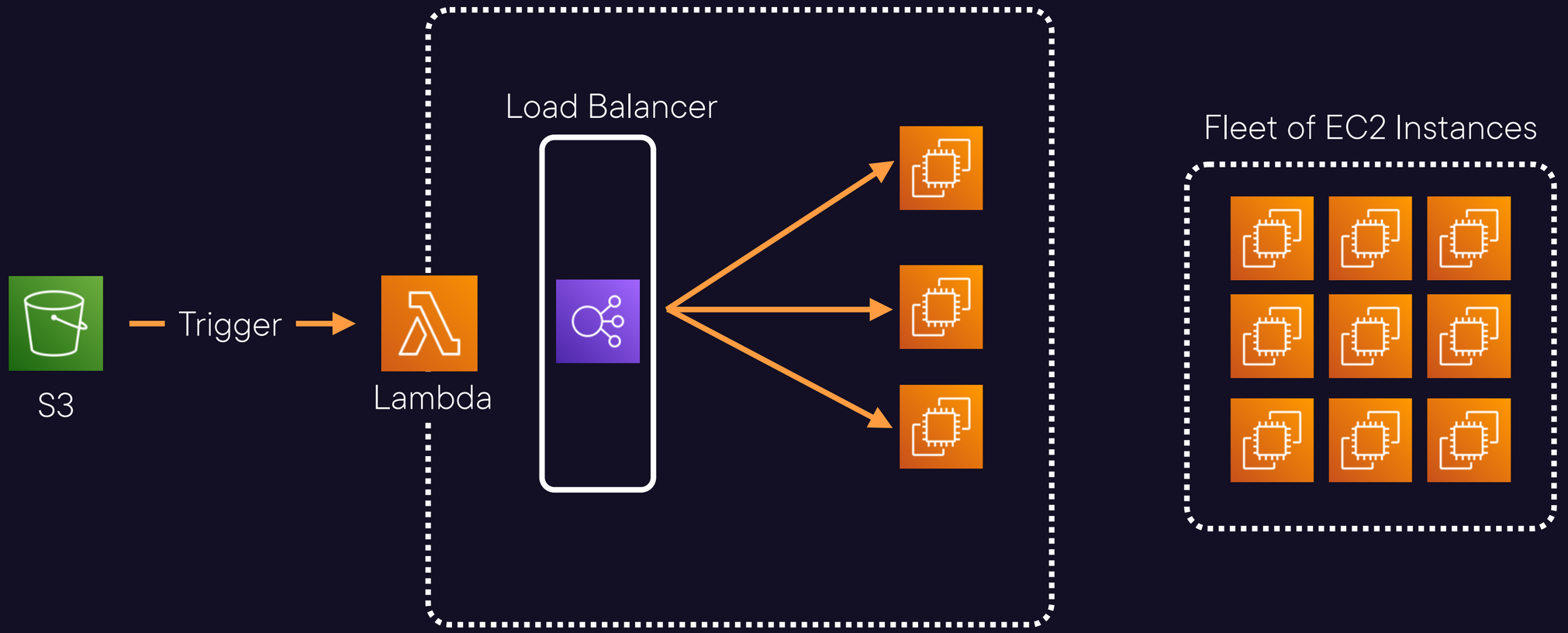
Each Lambda function runs in an isolated execution environment.

Execution environments:

- Manage the required resources
- Provide lifecycle support
- Language-specific

How Lambda Works

Under the Hood



Lesson Summary

Lambda's Building Blocks

- Function: A standalone unit of code
- Trigger: Resources that invoke Lambda
- Event: The data that is passed into functions
- Execution Environment: The runtime environment that manages the resources

Under the Hood

- Load balancer and a fleet of EC2 instances

Lambda Use Cases

Noreen Hasan
Training Architect

LESSON BREAKDOWN

Lambda Use Cases

- Processing Data
- Orchestrating Workflows
- Operating Serverless Applications
- Processing Messages
- Other Use Cases
- Lesson Summary

Noreen Hasan
Training Architect

Lambda Use Cases

Processing Data

Converting documents rapidly

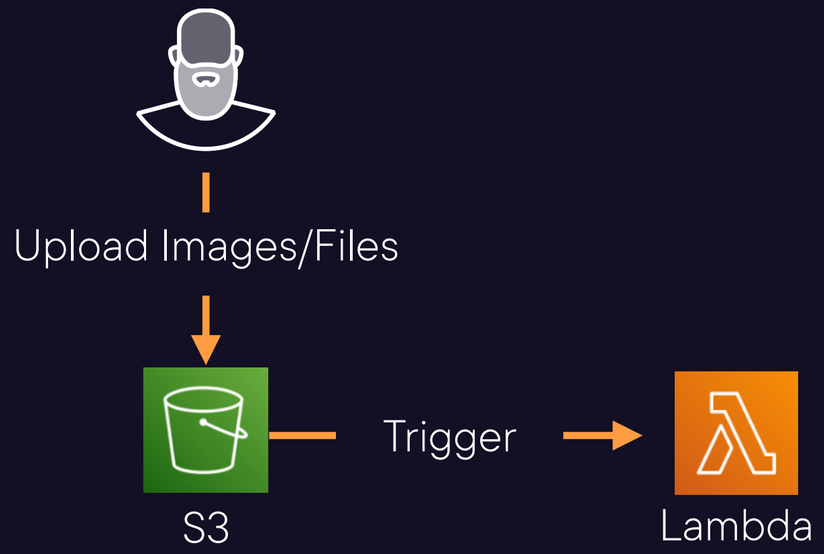
Processing S3 objects

Monitoring and analyzing logs

Processing streaming data

Lambda Use Cases

Processing Data



Processing S3 Objects

Trigger Lambda to modify files in S3:

Convert Files

Compress Files

Validate Data

Conform Data

Lambda Use Cases

Processing Data

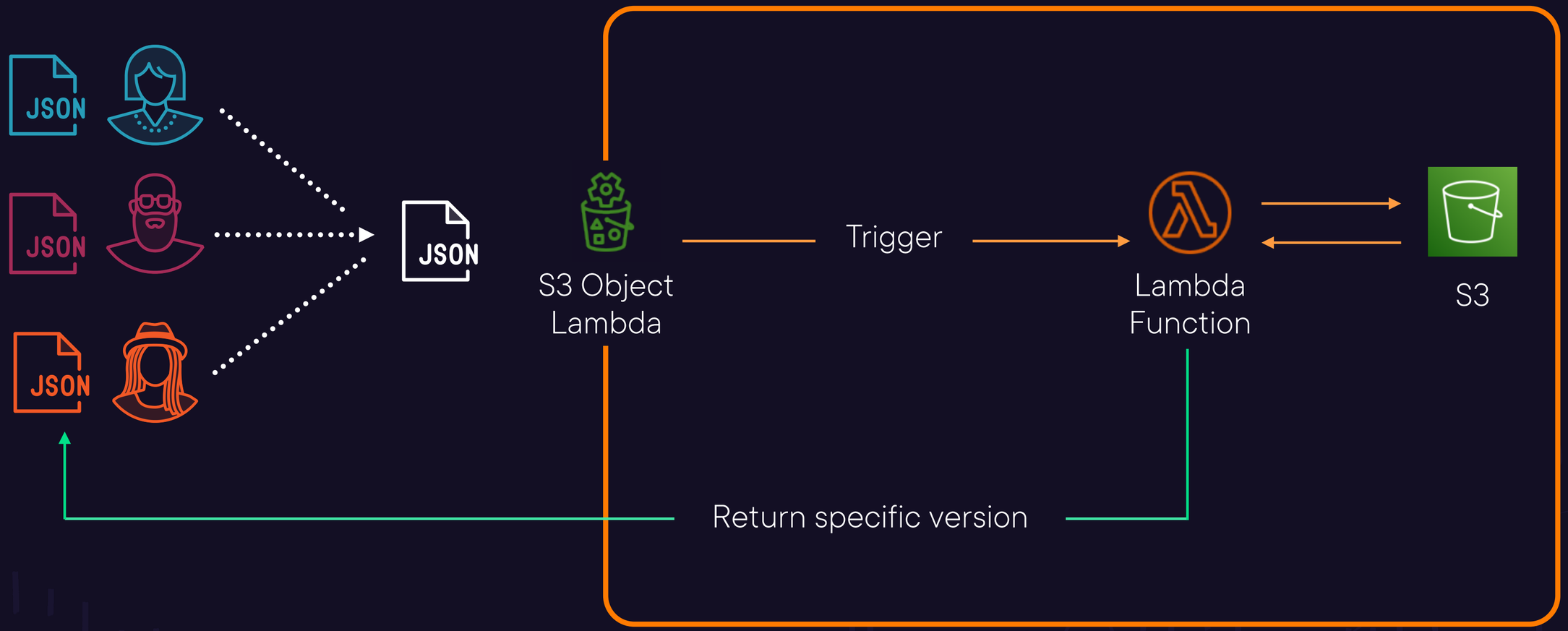
S3 Object Lambda

- Allows you to create different views of the data for different users

- Automatically transform your data as it is being retrieved from S3

Lambda Use Cases

Processing Data

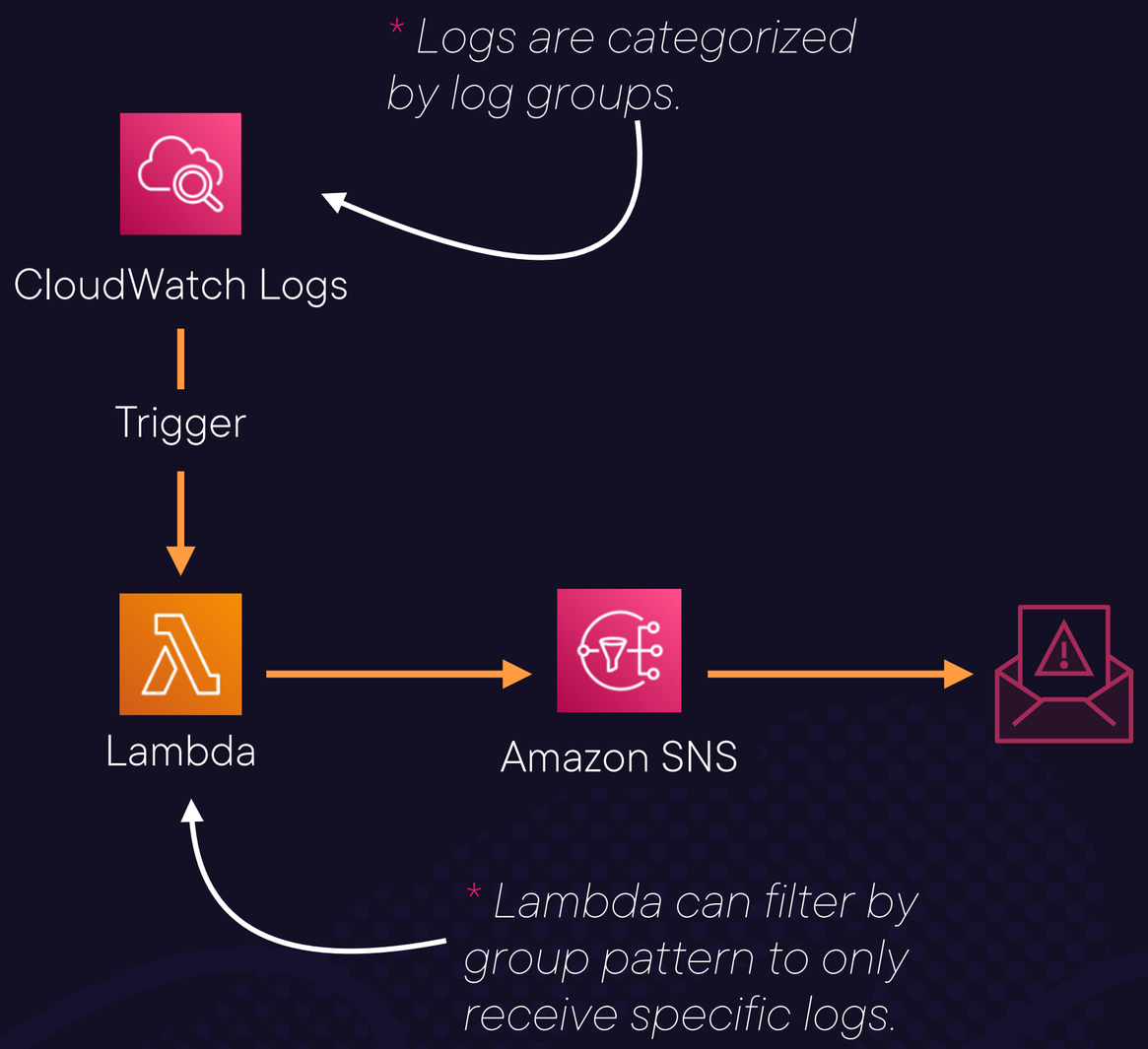


Lambda Use Cases

Processing Data

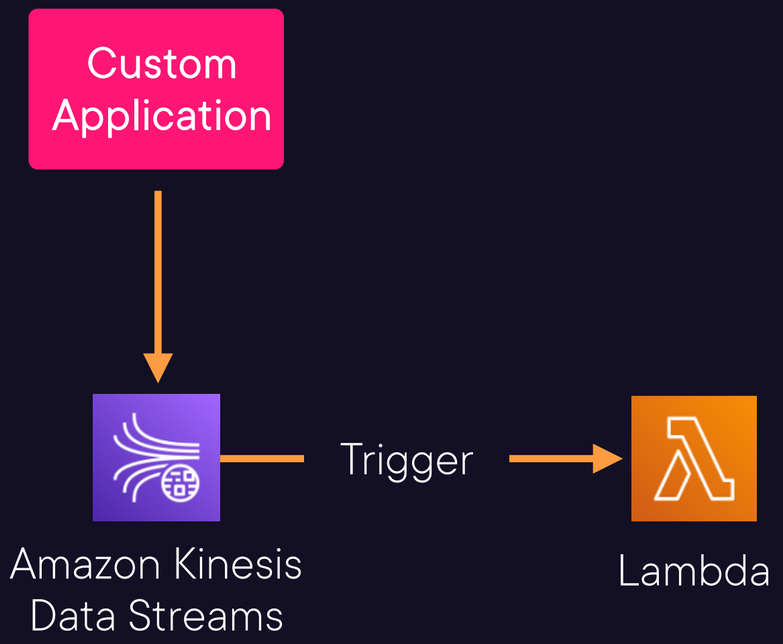
Monitoring and analyzing logs.

Subscribe to Lambda to receive log events from CloudWatch Logs.



Lambda Use Cases

Processing Data



Real-Time Stream Processing

Trigger Lambda to process data in real time one record at a time.

AWS Services

- Processing data in IoT
- Processing data from Kinesis Data Streams

Lambda Use Cases

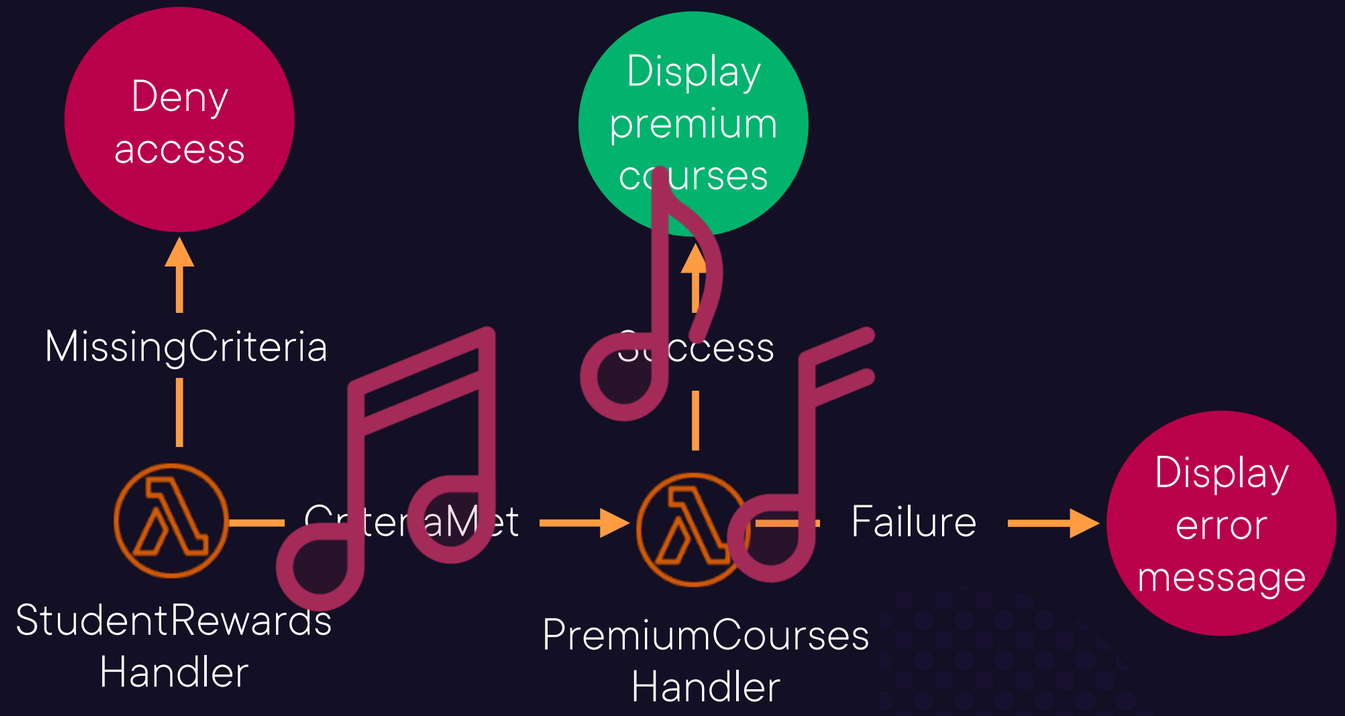
Orchestrating Workflows

Orchestration

- Workflows that involve branching logic

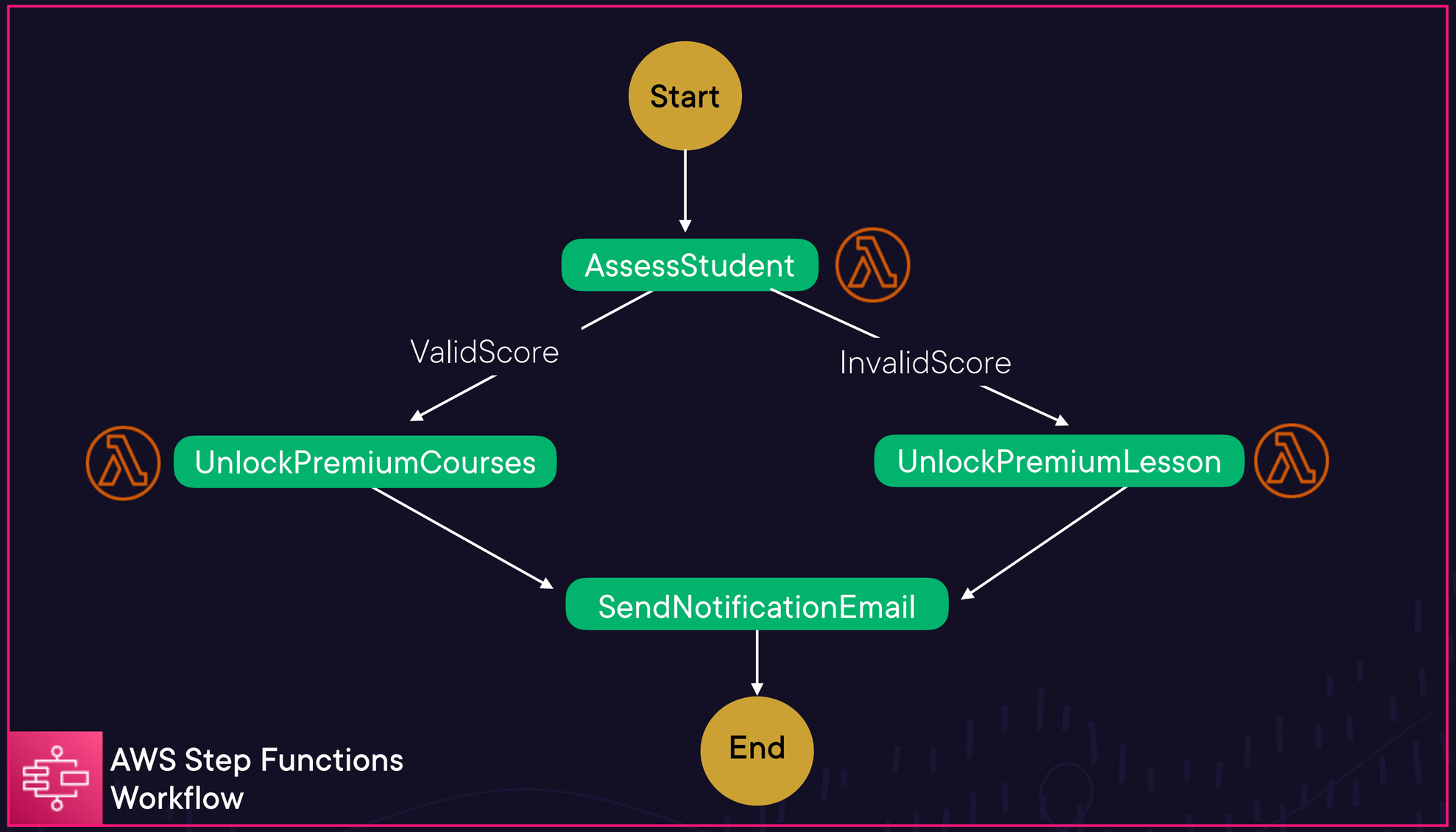
- Trigger Lambda to determine the next step in an orchestration workflow

- Integrates with AWS Step Functions



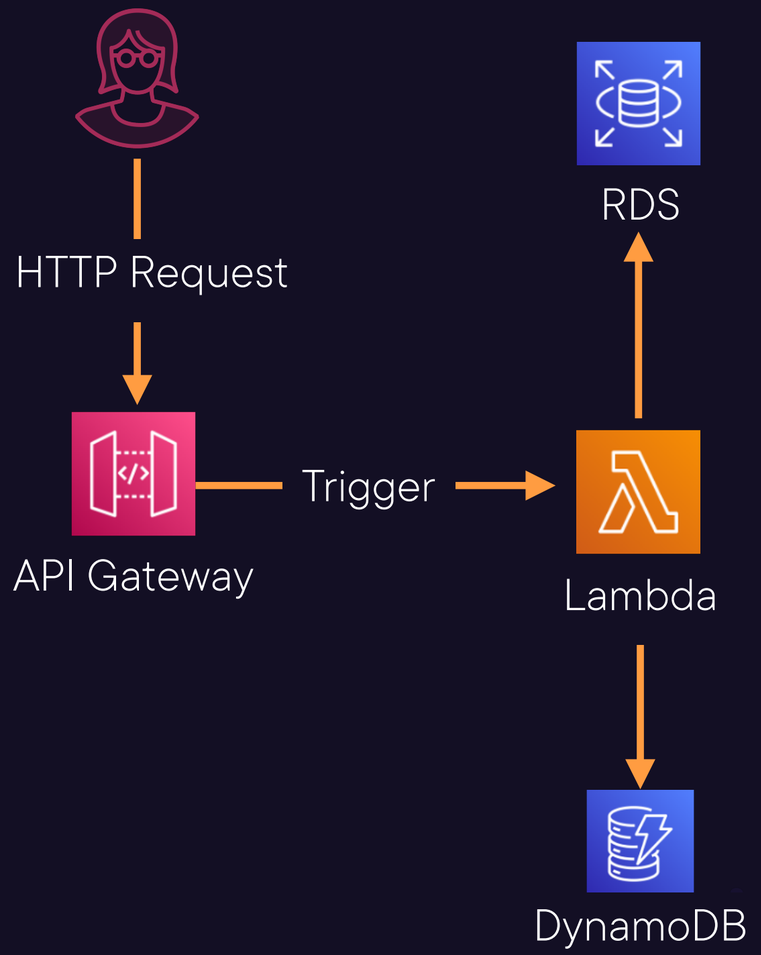
Lambda Use Cases

Orchestrating Workflows



Lambda Use Cases

Operating Serverless Applications

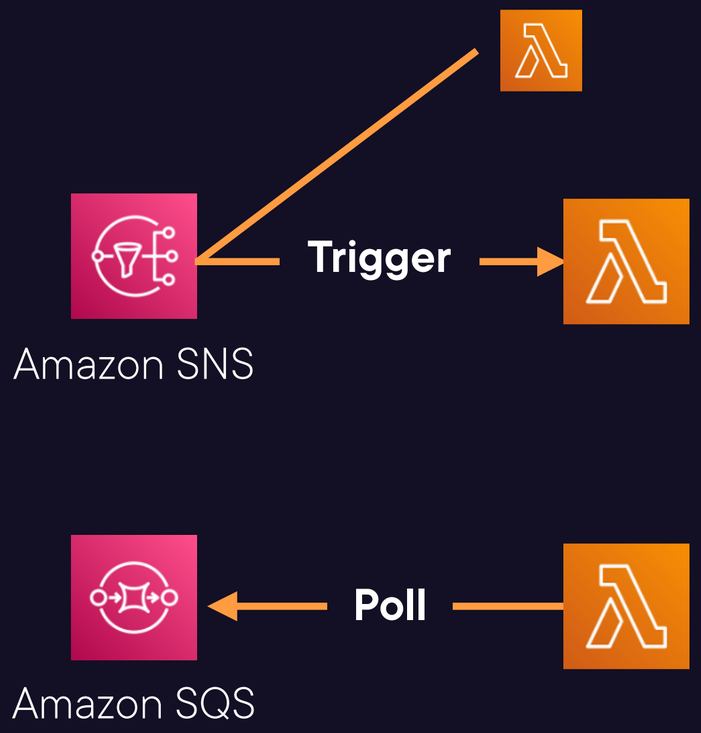


Serverless Applications

- Integrates with API Gateway to expose the Lambda function to external applications (e.g., web or mobile app).

Lambda Use Cases

Processing Messages



Simple Notification Service (SNS)

Trigger Lambda to process SNS messages.

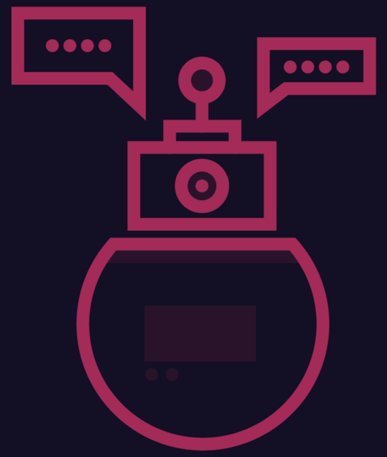
Simple Queue Service (SQS)

Get Lambda to **poll** SQS to look for messages in the queue.

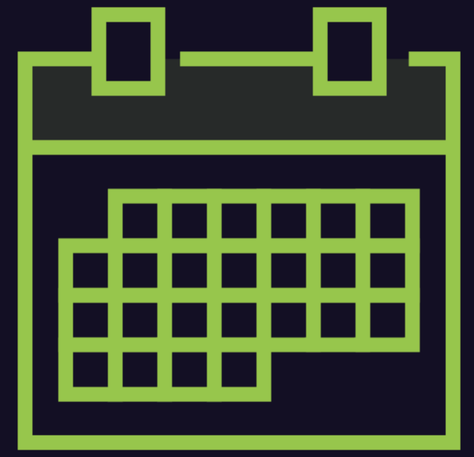
Other Use Cases

Alexa

Integrate with Alexa through Alexa Skills.



Build real-time chatbots using Lambda, SQS, SNS, and AWS Lex.



Schedule jobs (i.e., starting/stopping EC2 instances).

Lesson Summary

Processing Data

Integrating Lambda with S3 to:

- Convert documents rapidly
- Compress files
- Validate data
- Conform data

Orchestrating Workflows

Integrating Lambda with Step Functions for branching logic.

Processing Messages

- Getting triggered by SNS
- Polling from SQS

Other Use Cases

- Alexa
- Chatbots
- Scheduling jobs

When to Use Lambda

Noreen Hasan
Training Architect

LESSON BREAKDOWN

When to Use Lambda

- Lambda Constraints
- Alternative Services: Overview
- EC2
- Elastic Beanstalk
- ECS
- Deciding Factors
- Lesson Summary

Noreen Hasan
Training Architect

Lambda Constraints

Lambda is no

Execution Timeout:

< 15 mins

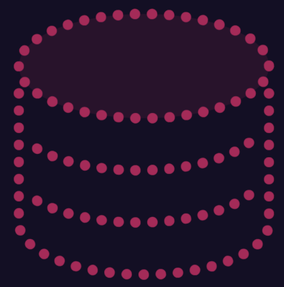
Lambda Constraints

Disk Space:

512 MB >  < 10 GB

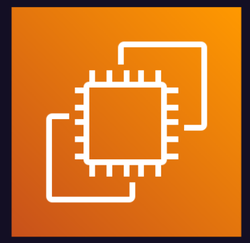
Lambda Constraints

Memory:

128 MB >  < **10 GB**

Alternative Services: Overview

AWS Elastic Beanstalk



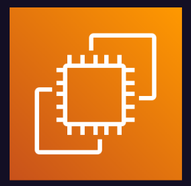
Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Container Service (ECS)

When to Use Lambda

Alternative Services

Lambda ⇢ Function as a Service (FaaS)



Elastic Beanstalk ⇢ Platform as a Service (PaaS)

EC2 ⇢ Infrastructure as a Service (IaaS)

ECS ⇢ Container as a Service (CaaS)

AWS Elastic Beanstalk

What is it? When to use it?

AWS Elastic Beanstalk

- Compute service for hosting applications
- Abstracts system administration

Use Cases

- Automation
- Hosting and deploying **fully functional** applications

When to Use Lambda

Elastic Beanstalk

Sys Admin

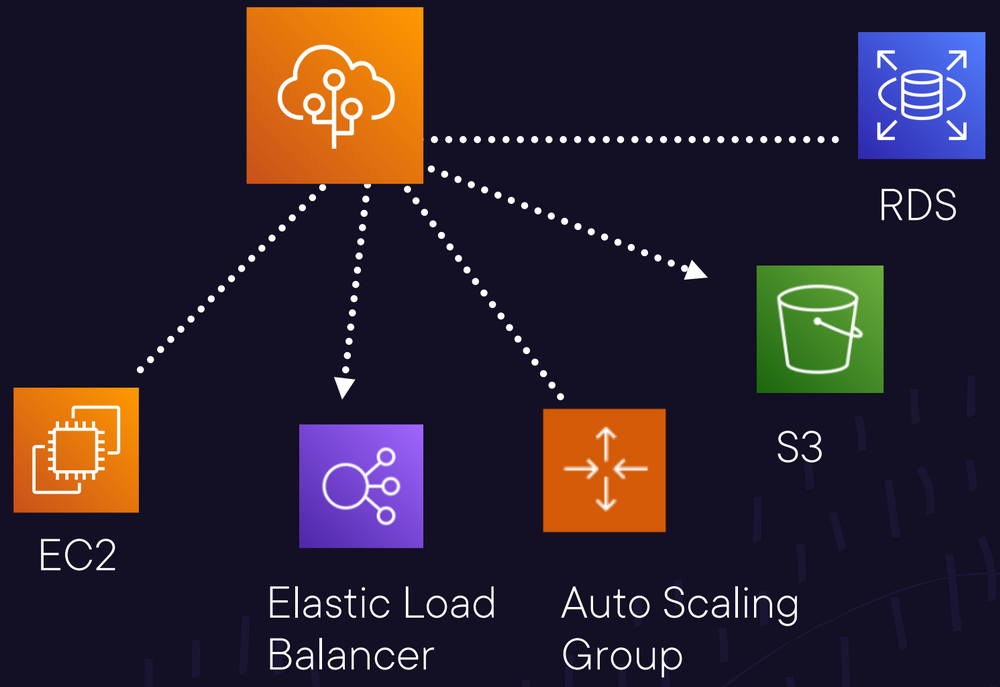
Provisioning

Load Balancing

Application Health Monitoring

Auto Scaling

Generated Resources



AWS Elastic Beanstalk

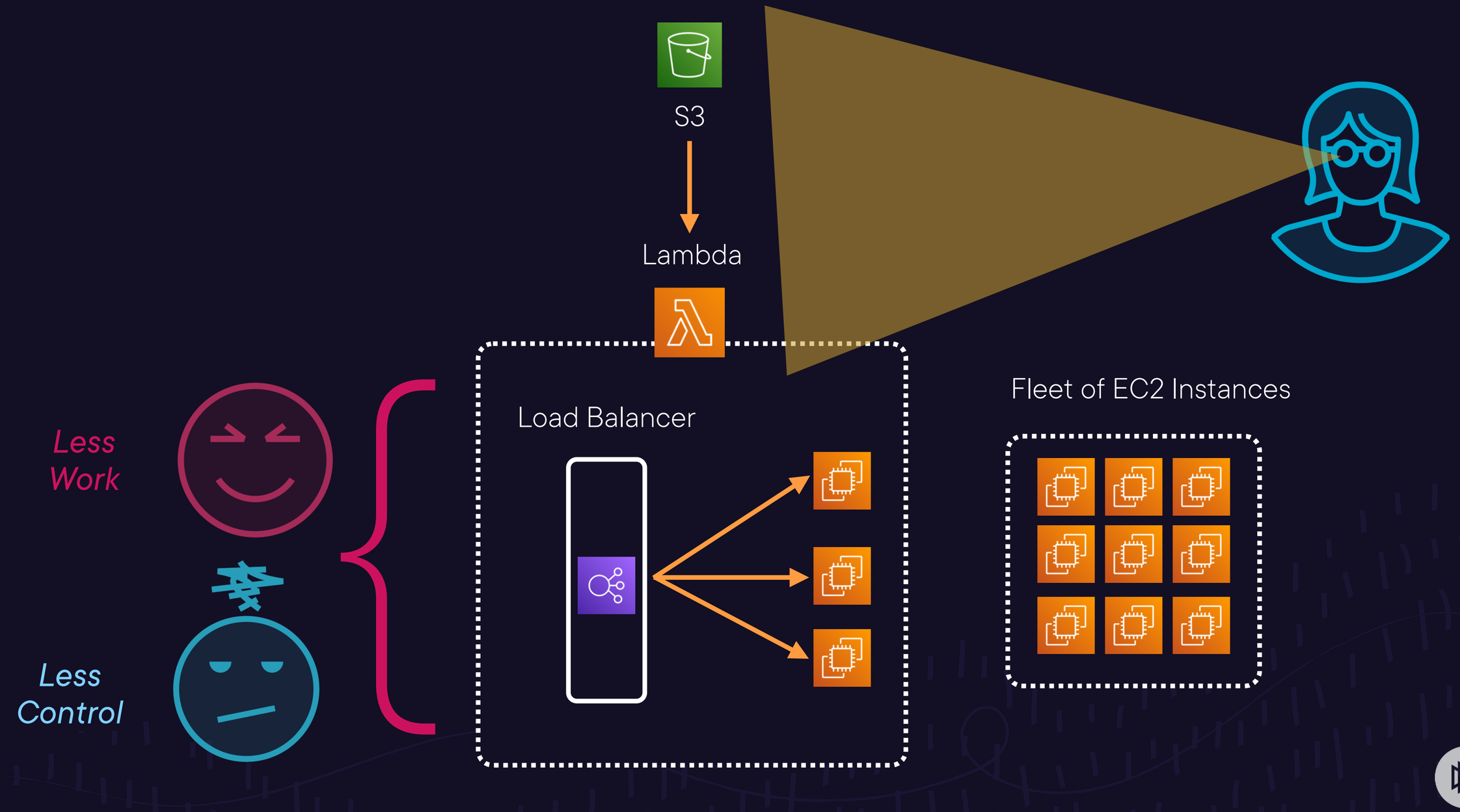
How does it differ from Lambda?

Elastic Beanstalk reduces complexity **without restricting** your choices and control.

Lambda reduces complexity by abstracting provisioning. Due to **limited choices**, you can set it and forget it.

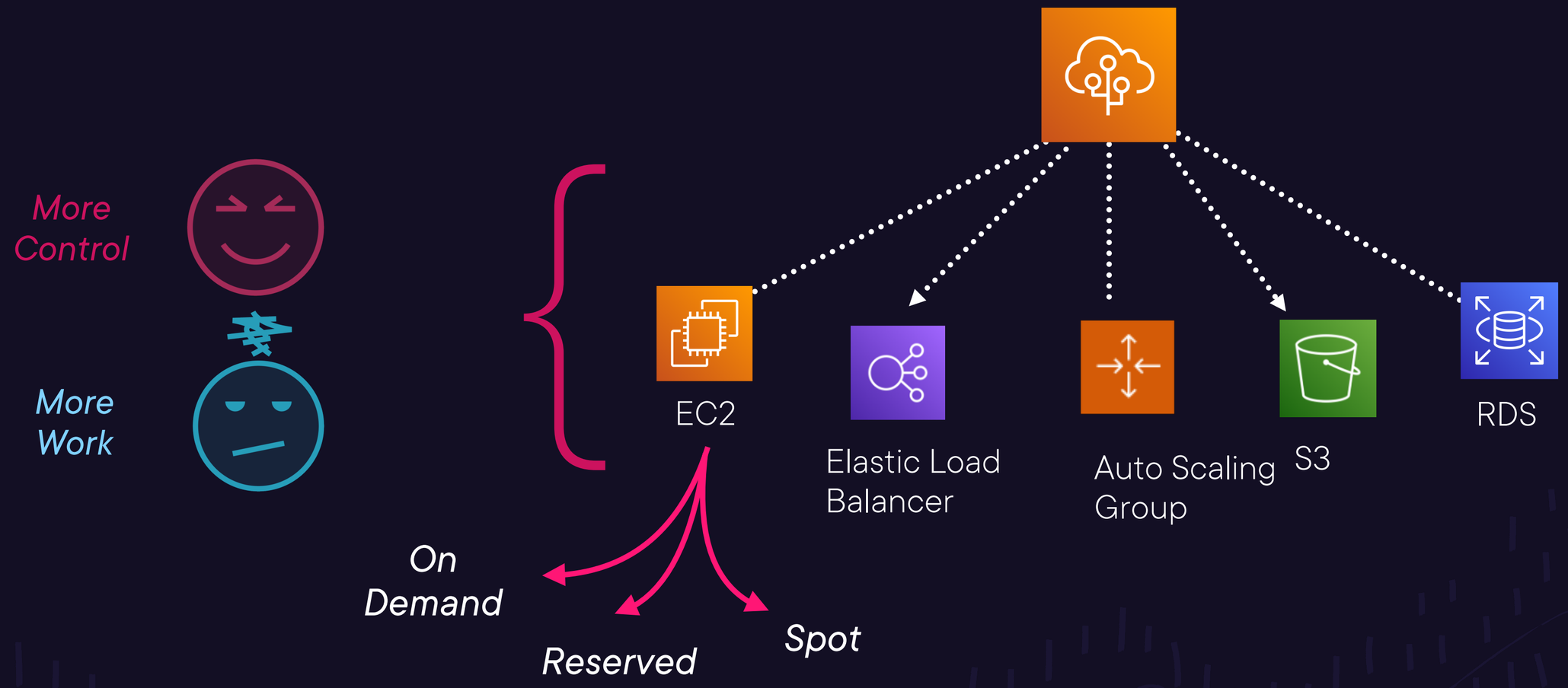
When to Use Lambda

Elastic Beanstalk vs. Lambda



When to Use Lambda

Elastic Beanstalk Offers Flexibility



Elastic Beanstalk: Pricing

- Elastic Beanstalk is free.

- You only get charged for the used resources.

EC2

What is it? When to use it?

EC2

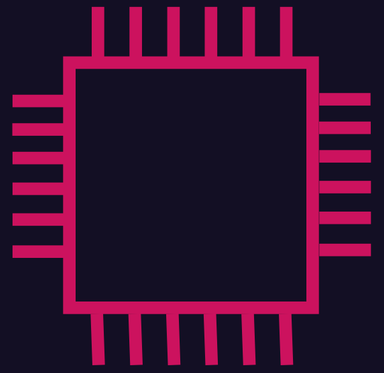
- Rentable virtual machines (instances)
- Allows you to spin up/down instances

Use Cases

- Hosting applications or databases (e.g., WordPress and MySQL)
- High flexibility needed and high expertise available

EC2 Instance Components

Amazon Machine Image (AMI)
(Windows or Linux)

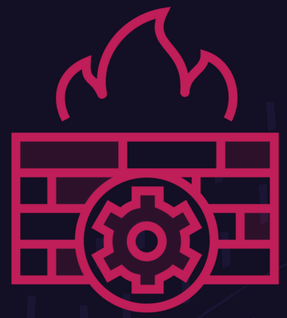


Instance Type
(Processing Power)

Memory
(RAM)

EBS
(Local Storage)

IP Addressing
(Access)



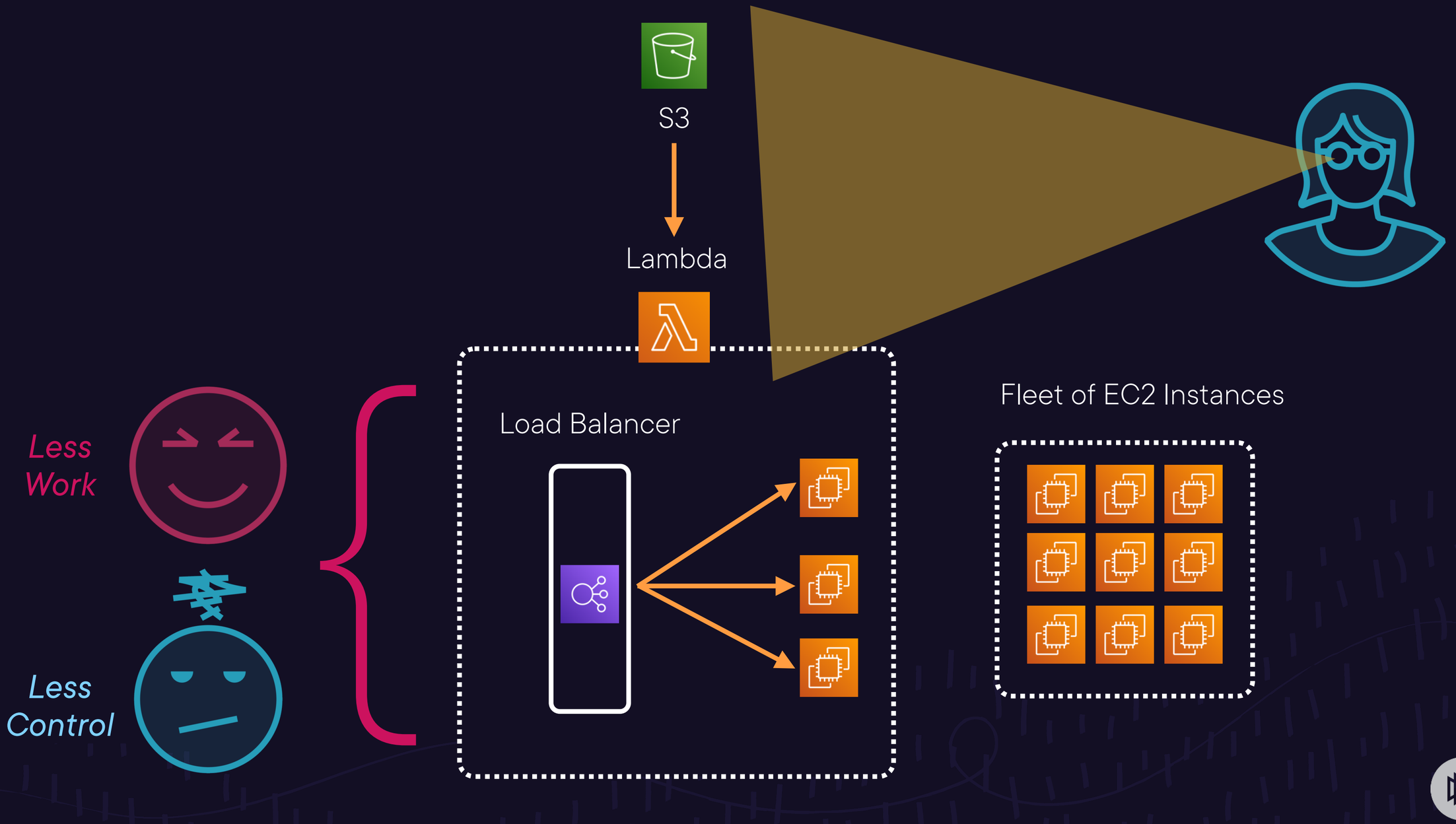
Security Groups

EC2

How does it differ from Lambda?

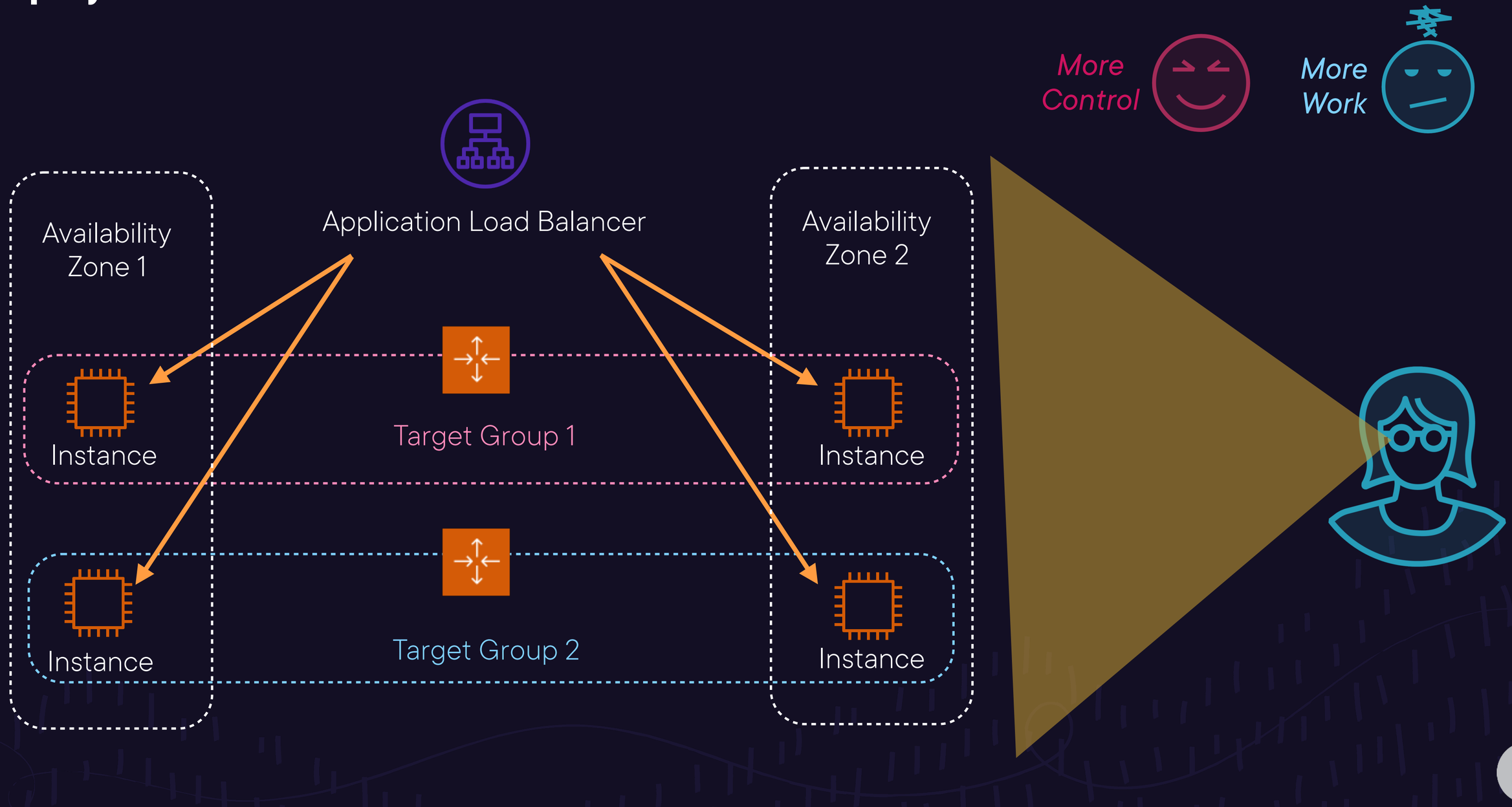
When to Use Lambda

EC2 vs. Lambda



When to Use Lambda

Deployment with EC2



EC2: Your Responsibilities with EC2

Pick an instance type for your use case (compute, memory, or general purpose).

Handle security through security groups and installing security patches.

Handle scaling and traffic distribution through setting up Auto Scaling groups and target groups.

Shut down the instance when no longer needed, either manually or via a shutdown task.

ECS

What is it? When to use it?

ECS

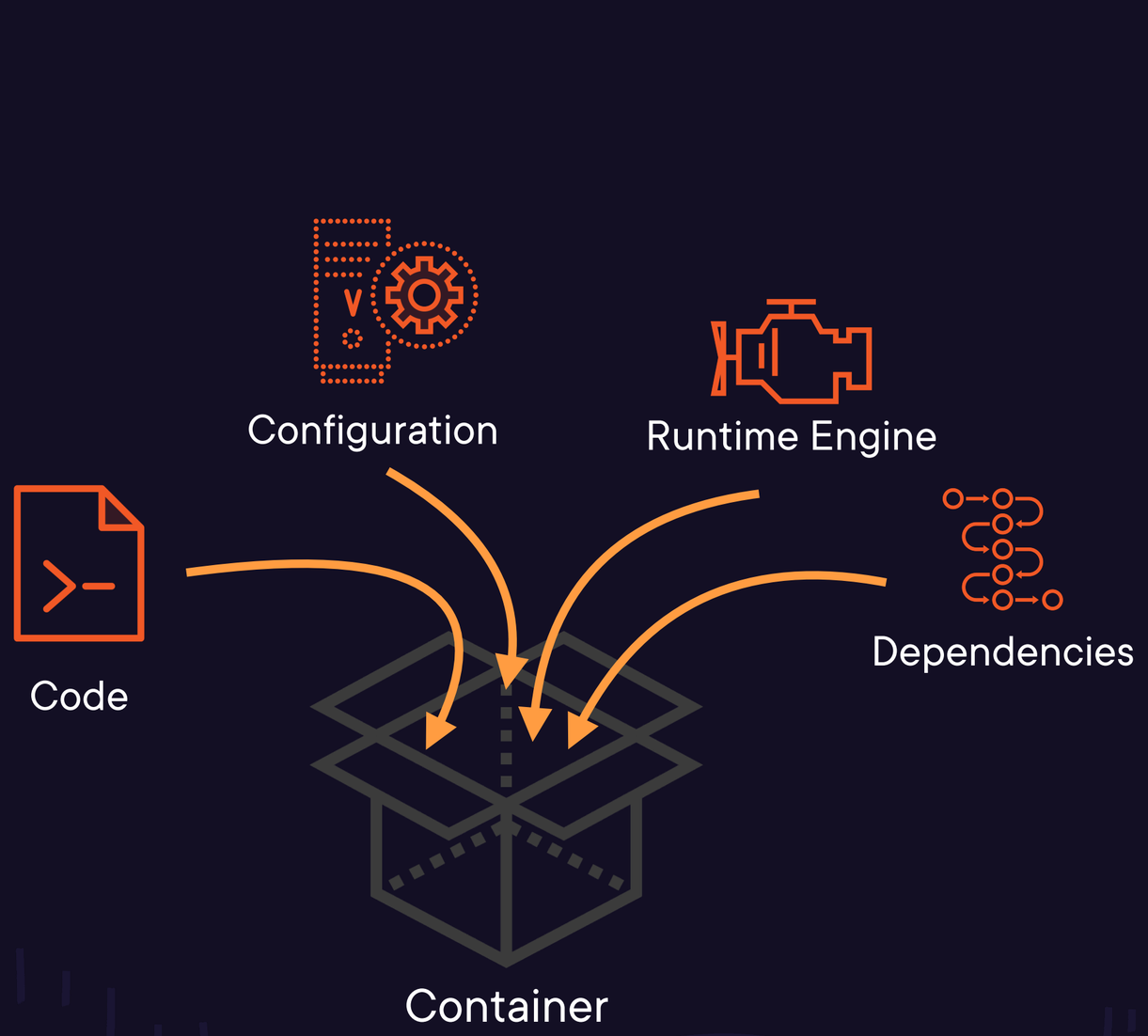
- Orchestrator containers (e.g., Docker)
- Operates on clusters (EC2 instances or Fargate) and tasks

Use Cases

- Full-scale applications or ad-hoc jobs
- Running pre-packaged images or third-party software
- Predictable traffic

When to Use Lambda

ECS and Containers



So, if  doesn't meet your runtime needs...

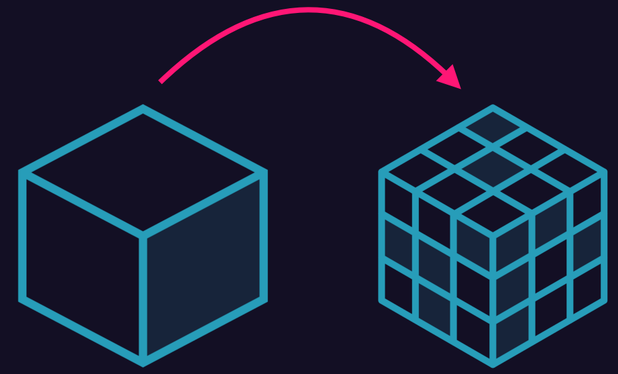
Consider Elastic Beanstalk

or EC2

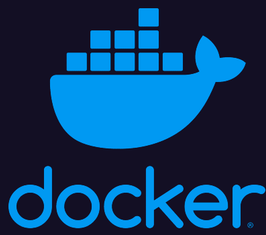
If your deployment package is

> 512 MB

break up your package
into **smaller**
packages and functions.



If

- You're working with 

- Your jobs take > 15 mins
- You need to schedule jobs

Consider  ECS.

If

- You want to focus more on the code
- You want to focus less on servers
- Your jobs take < 15 mins
- Never pay $ for idle time

Consider  Lambda.

Lesson Summary

Lambda Constraints

Execution timeout, disk space, and memory.

AWS Services

- Lambda: Smaller applications that take less than 15 mins to run
- EC2: Highly flexible, but high expertise needed
- Elastic Beanstalk: Fully managed customizable applications
- ECS: Container orchestration

Accessing Lambda

Noreen Hasan
Training Architect

LESSON BREAKDOWN

Accessing Lambda

Ways to Access Lambda's Resources

Lambda Console: Ways to Create Functions

Demo

Lesson Summary

Noreen Hasan
Training Architect

Accessing Lambda

Ways to Access Lambda's Resources

AWS Command Line Interface (CLI)

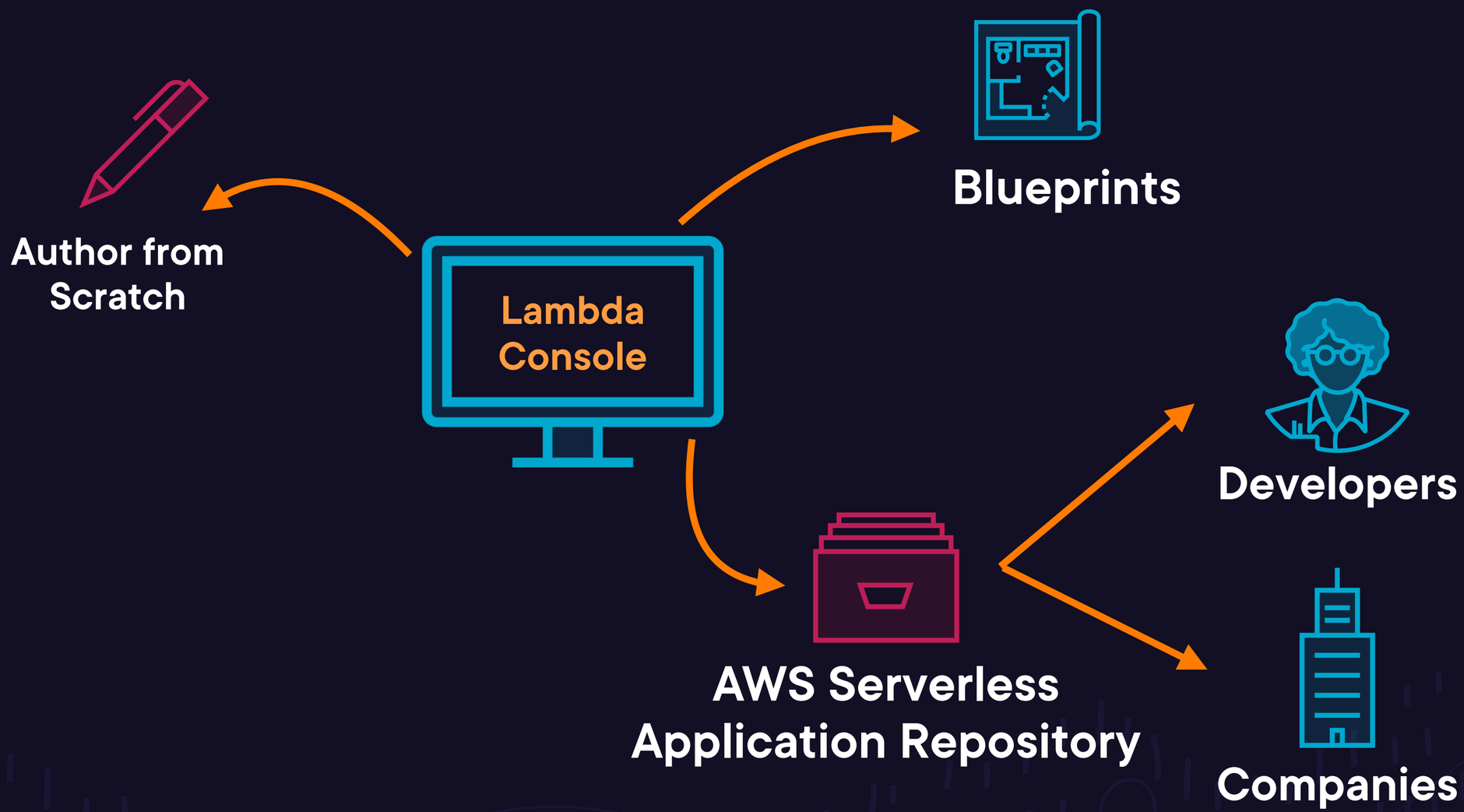
SDK

AWS Software Development Kit (SDK)

The AWS Console - Lambda Console

Accessing Lambda

Lambda Console: Ways to Create Functions



Demo

Lambda Console

- Walkthrough of the console
- Create a Hello World 👋 application

Lesson Summary

Ways to Access Lambda's Resources

- AWS CLI
- SDK
- Lambda console

Lambda Console: Ways to Create Functions

- Author from scratch
- Blueprints
- AWS Serverless Application Repository

Demo

- Create a Hello World application

Introduction to Processing Data with Lambda

Noreen Hasan
Training Architect

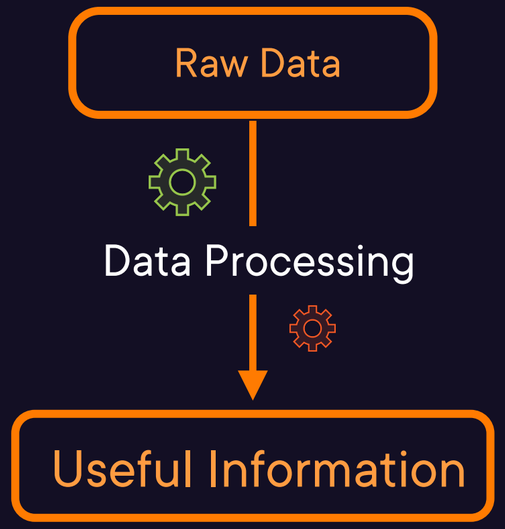
SECTION BREAKDOWN

Introduction to Processing Data with Lambda

Section Overview

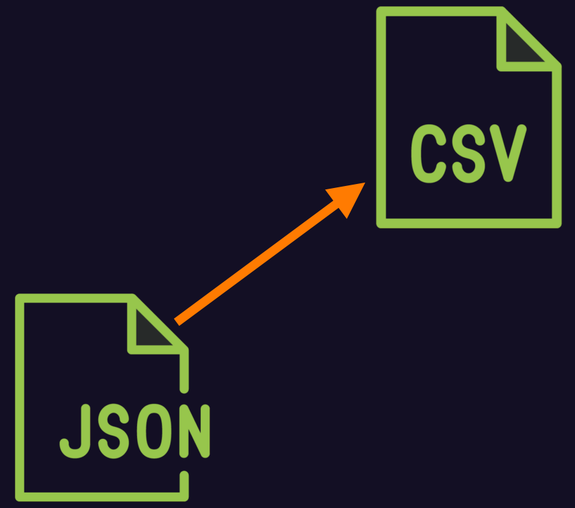
Noreen Hasan
Training Architect

Section Overview



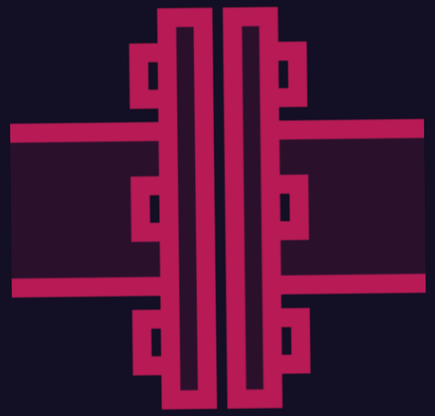
Processing Data: Overview

Understanding Lambda Layers



Demo: Converting JSON to CSV Using Lambda

Section Overview



Demo: Ingesting CSV Files from **S3** to **DynamoDB**

Processing Real-Time Data Using **Kinesis** and Lambda: Overview

Demo: Processing Real-Time Data Using **Kinesis** and Lambda

Testing and **Debugging** Lambda **Failures**

Let’s get started!

Processing Data: Overview

Noreen Hasan
Training Architect

LESSON BREAKDOWN

Processing Data: Overview

- What Is Data Processing?
- Data Lakes
- Extract, Transform, and Load (ETL)
- Extract, Load, and Transform (ELT)
- Types of Data Transformations
- Lesson Summary

Noreen Hasan
Training Architect

What Is Data Processing?

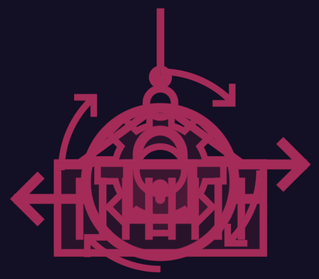
Data Processing

The process of converting raw data to meaningful information

How Is It Done?

- Manual: pen and paper
- Mechanical: typewriters
- Electronic: **computers programs**

Data Lakes



A central storage repository that holds raw data in native format

Structured Data

Semi-structured Data

Unstructured Data

Data Lake

Processing Data: Overview

Extract, Transform, and Load (ETL)

Extract

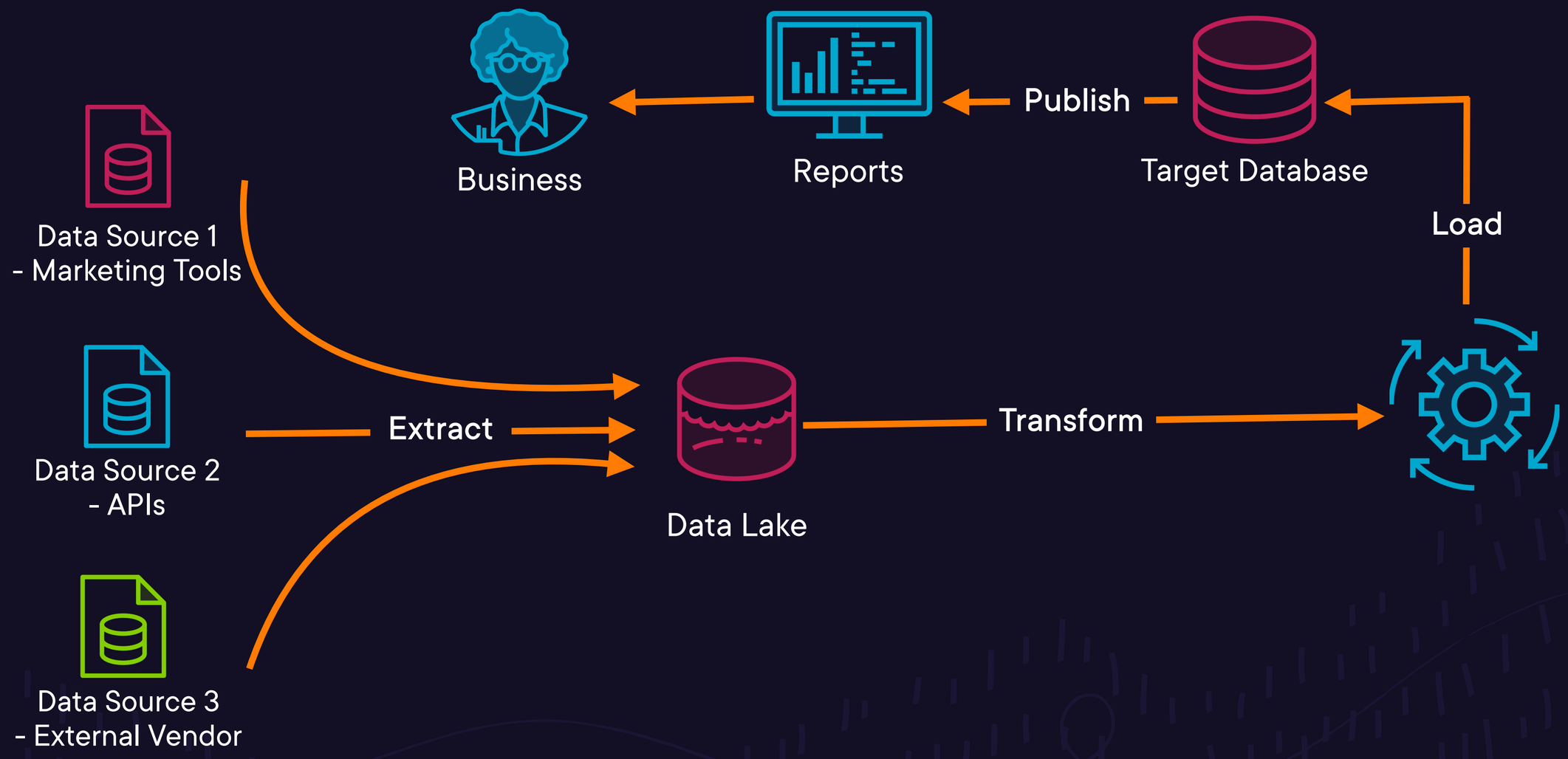
Transform

Load

Data Lake

Processing Data: Overview

Data Lakes and ETL



Processing Data: Overview

Extract, Load, and Transform (ELT)

Extract

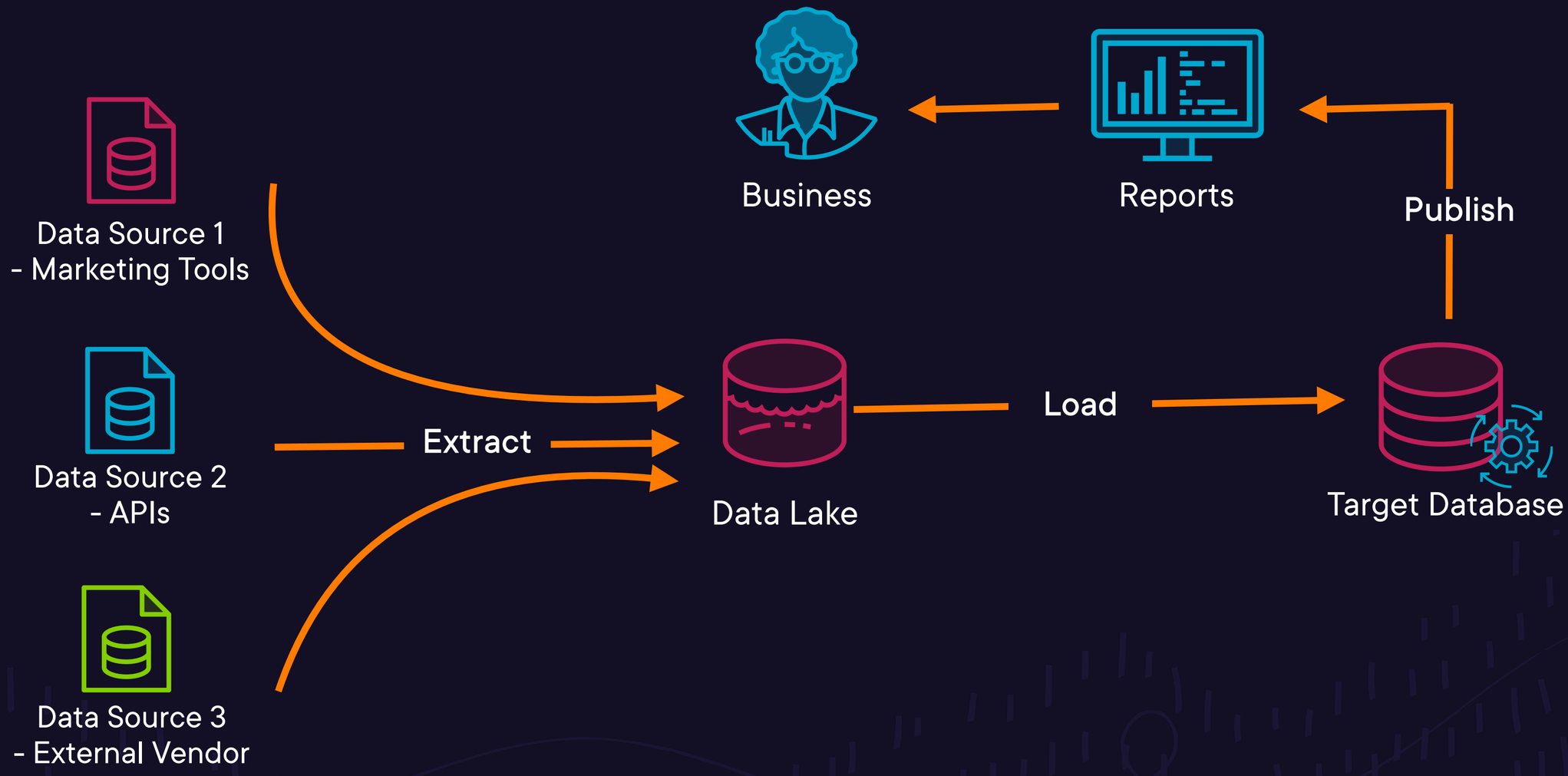
Load

Transform

Data Lake

Processing Data: Overview

Data Lakes and ELT



How Is ELT Made Possible?

How Is ELT Made Possible?

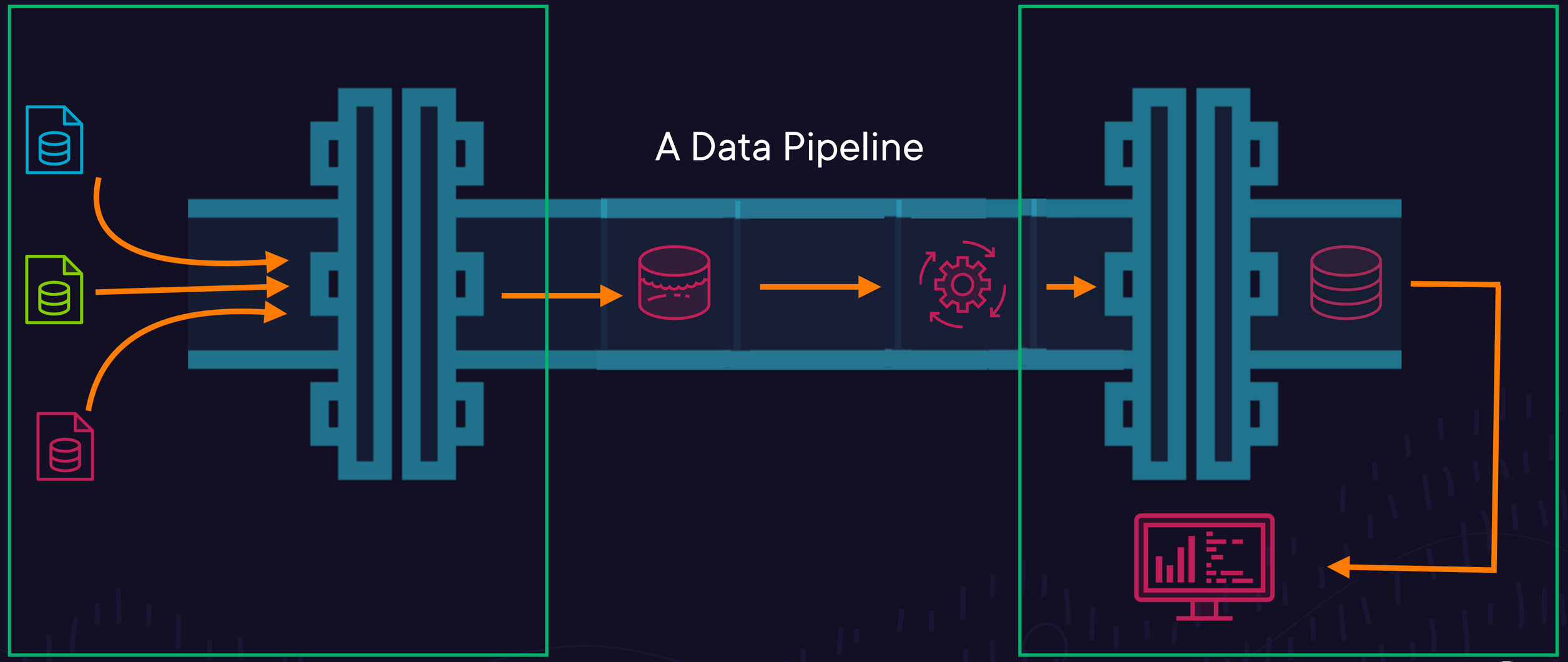
Amazon Redshift

Snowflake

Azure

Google BigQuery

Data Pipeline



Types of Data Transformations

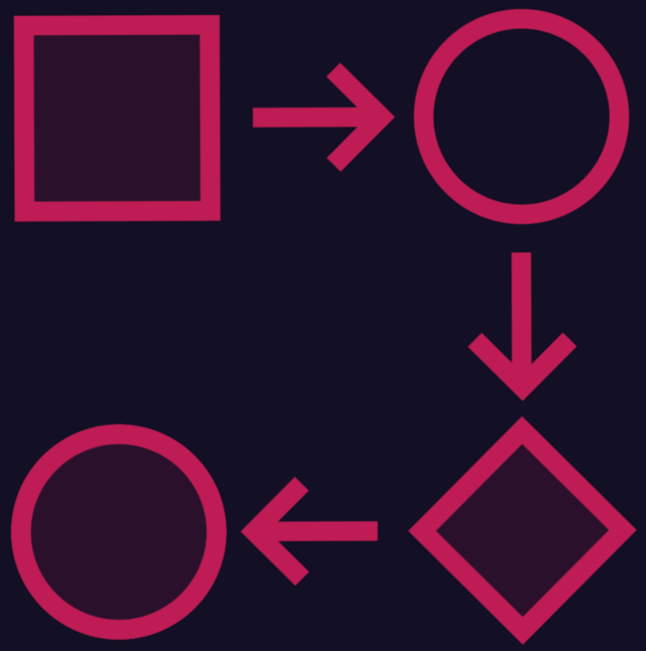
Filtering Data

Removing certain columns or rows before inserting the data into the final destination

Example

- Filtering out historical records to create the table certification_exams_2023

Types of Data Transformations



Mapping Data

Taking an input from a data source and changing (conforming) it into an equivalent in another data format

Example

- Transforming date fields from local time zones to a standardized timestamp

- Transforming textual fields to numerical fields (i.e., surveys)

  - Like= 1
  - Dislike= 0

Types of Data Transformations

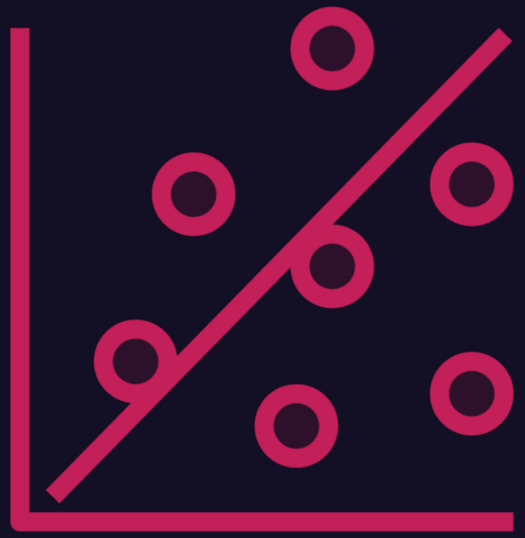
Deriving Variables

Variables that are computed based on other data variables

Example

- Creating a column, Final Price, that computes the final price of a course by subtracting the Discount from the Original Price

Types of Data Transformations



Aggregating Data

Creating a summarizing field that groups and aggregates data across a particular dimension

Example

- Calculating the sum, average, mean, or median

Types of Data Transformations

CourseID	CourseTitle	Author
837	Intro to Lambda	Tia Williams
837	Intro to Lambda	Tia Williams

Deduplicating Data

Removing duplicated data to decrease needed storage

It can be...

- part of the transform process before the data is inserted to the final database.

- a process that takes place after insertion.

Example

- Removing course records that are identical across all fields

Types of Data Transformations

Splitting Data

Refining unstructured data by splitting certain fields before inserting them into the final table

Example

- Splitting an address field into smaller fields, such as street, city, and zip_code

Lesson Summary

What Is Data Processing?

- Converting raw data to information

Data Lakes

- A central repo  with raw data
- ETL vs. ELT

Types of Data Transformation

- Data filtering
- Data mapping
- Deriving variables
- Aggregating data
- Deduplicating data
- Splitting data

Understanding Lambda Layers

Noreen Hasan
Training Architect

LESSON BREAKDOWN

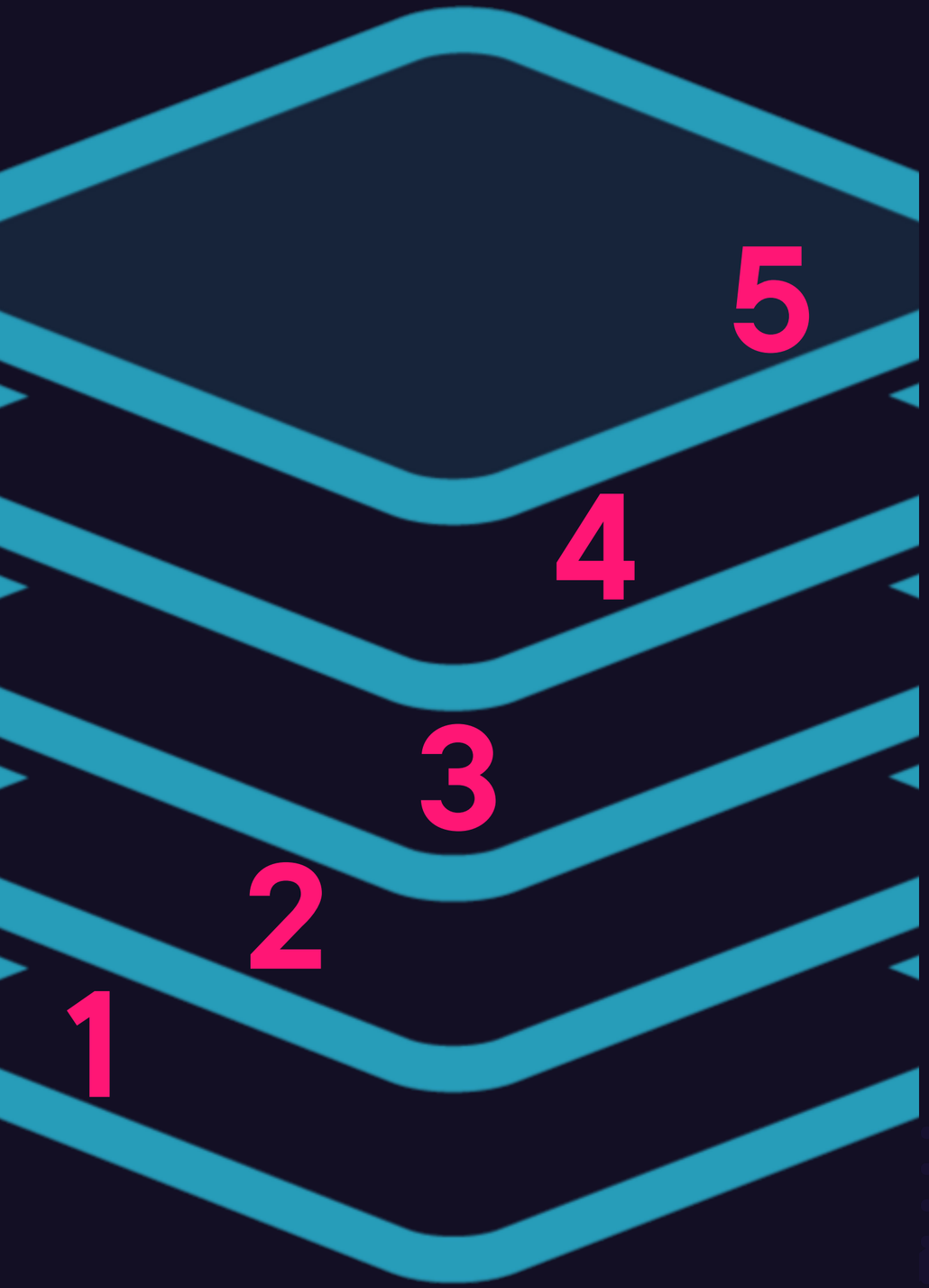
Understanding Lambda Layers

- What Are Lambda Layers?
- Why Layer Up?
- Challenges
- Demo
- Lesson Summary

Noreen Hasan
Training Architect

Lambda Layers

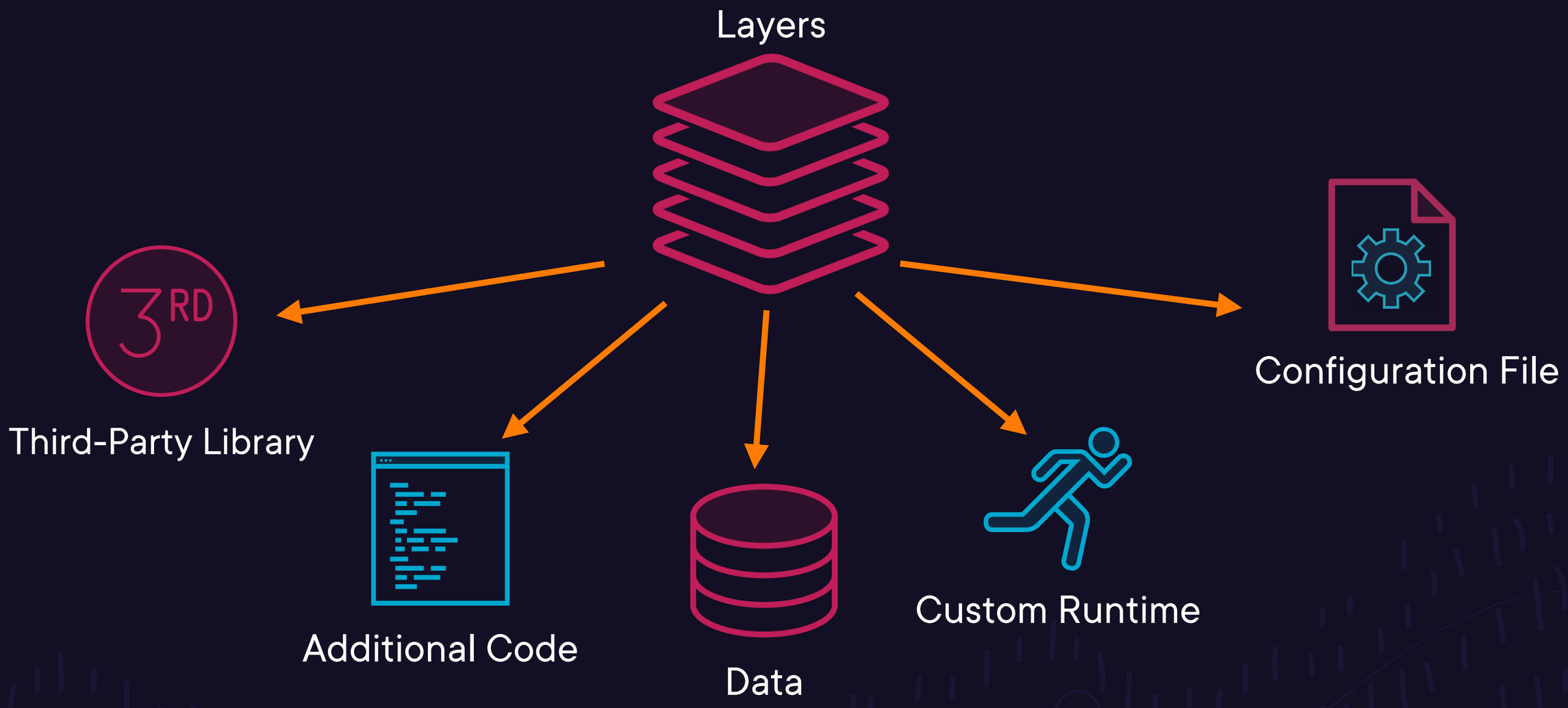
- A layer is a ZIP file with additional code or data.
- Layers allow you to share common code between functions and across accounts.
- To use a layer, you need to import it to your function.



Lambda Layers

- You can add up to five layers to a function.
- Each layer can potentially overwrite the previous layer; thus, the order is important.
  - First layer= custom runtime
  - Second layer= third-party library
  - Third layer= config file

What's in a Layer?



Why Layer Up? Story Time

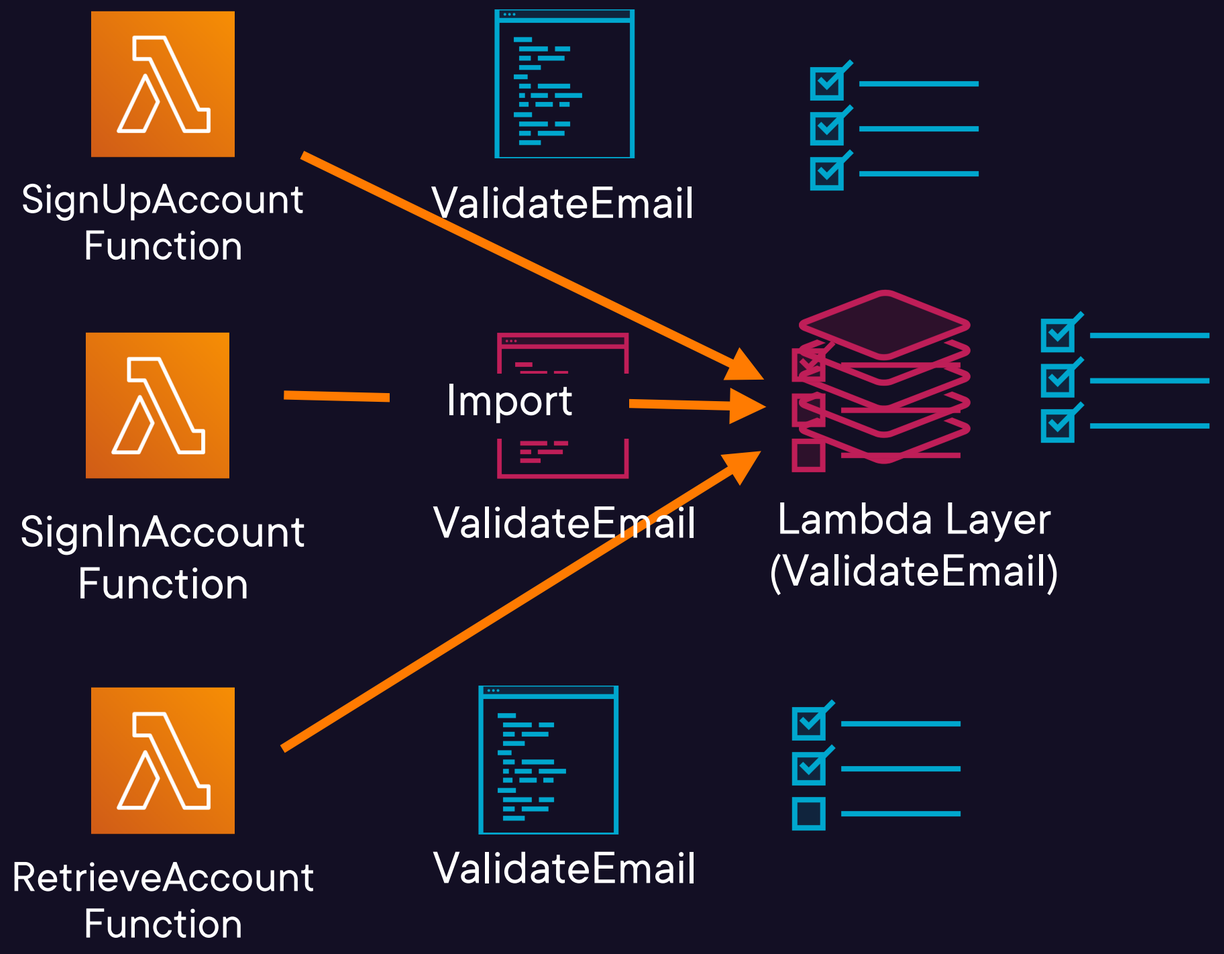
Grandma's Pizza

My Pizza

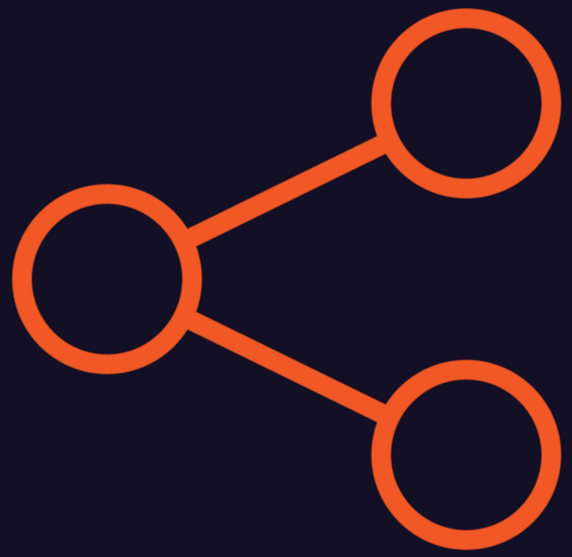
Grandma's Sauce

Any Pizza

Why Layer Up? Code Reusability



Why Layer Up? Code Reusability



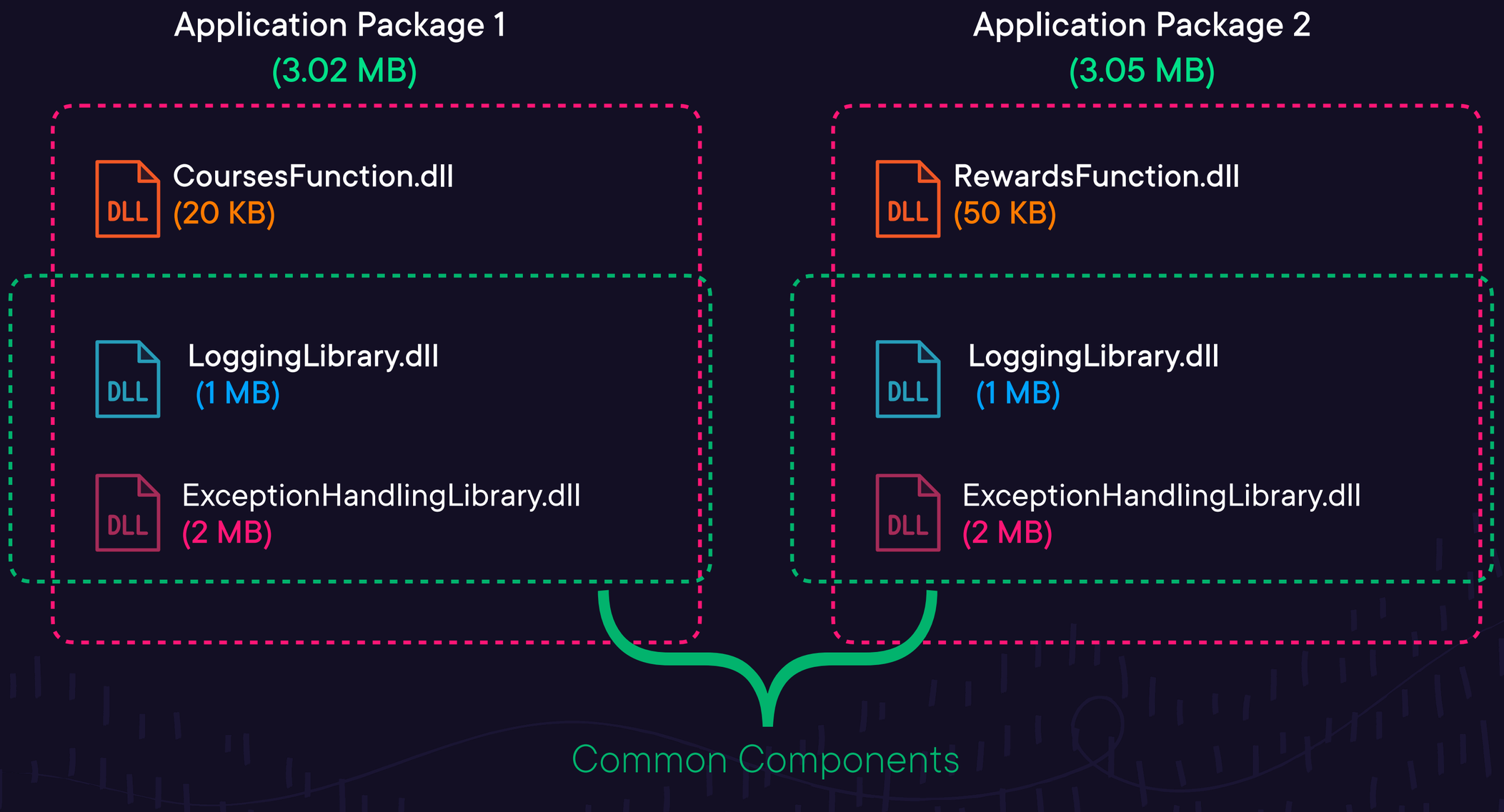
Layers can be shared...

- Within an **AWS account**
- Between **different AWS accounts**
- **Publicly** with the developer community

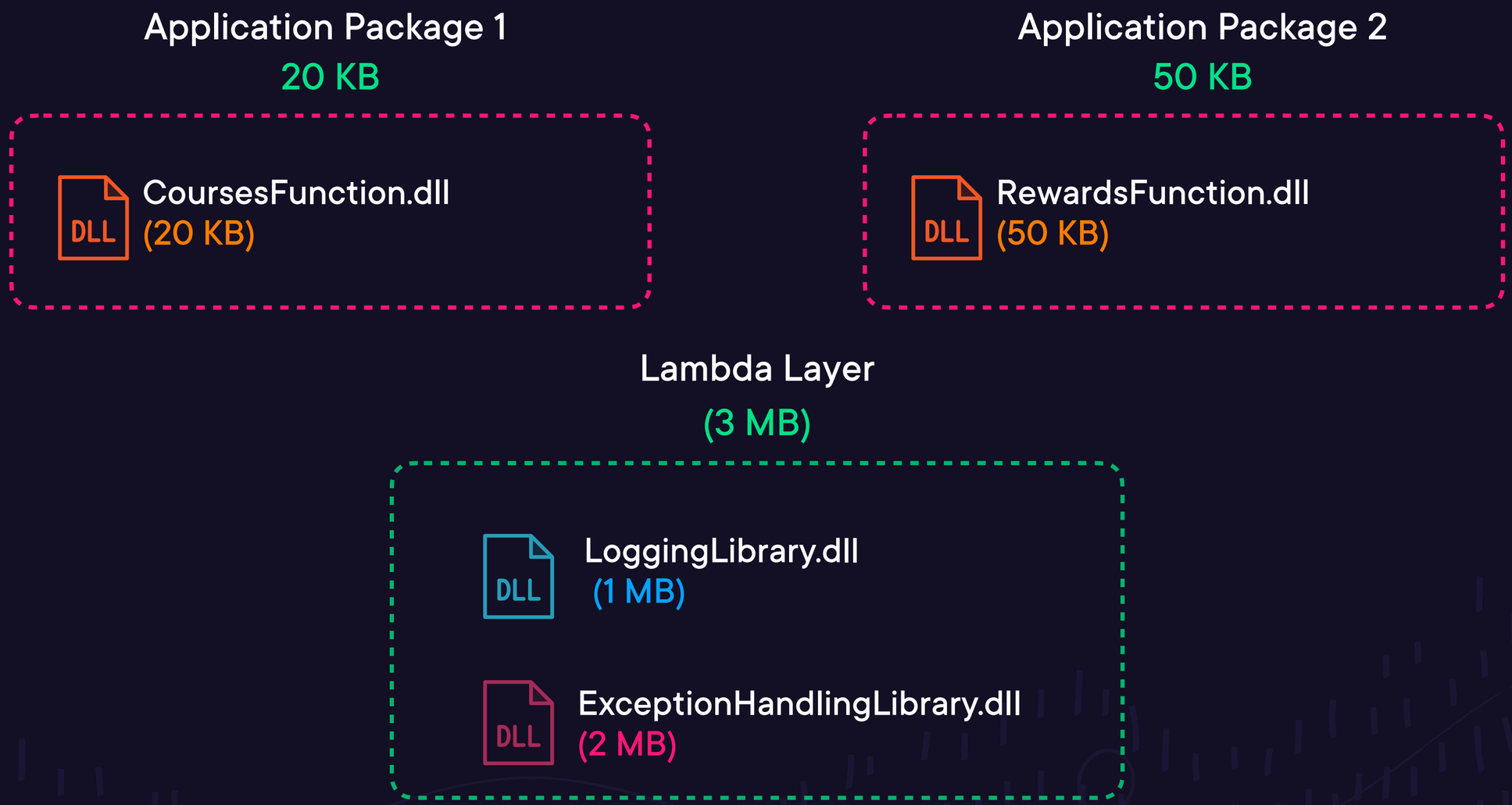
Code Reusability

- Don’t waste time rebuilding existing functionality.
- Simplify code updates.
- Bugs are contained in layers.
- Smaller deployment packages result in faster deployments.

Why Layer Up? Smaller Deployment Packages



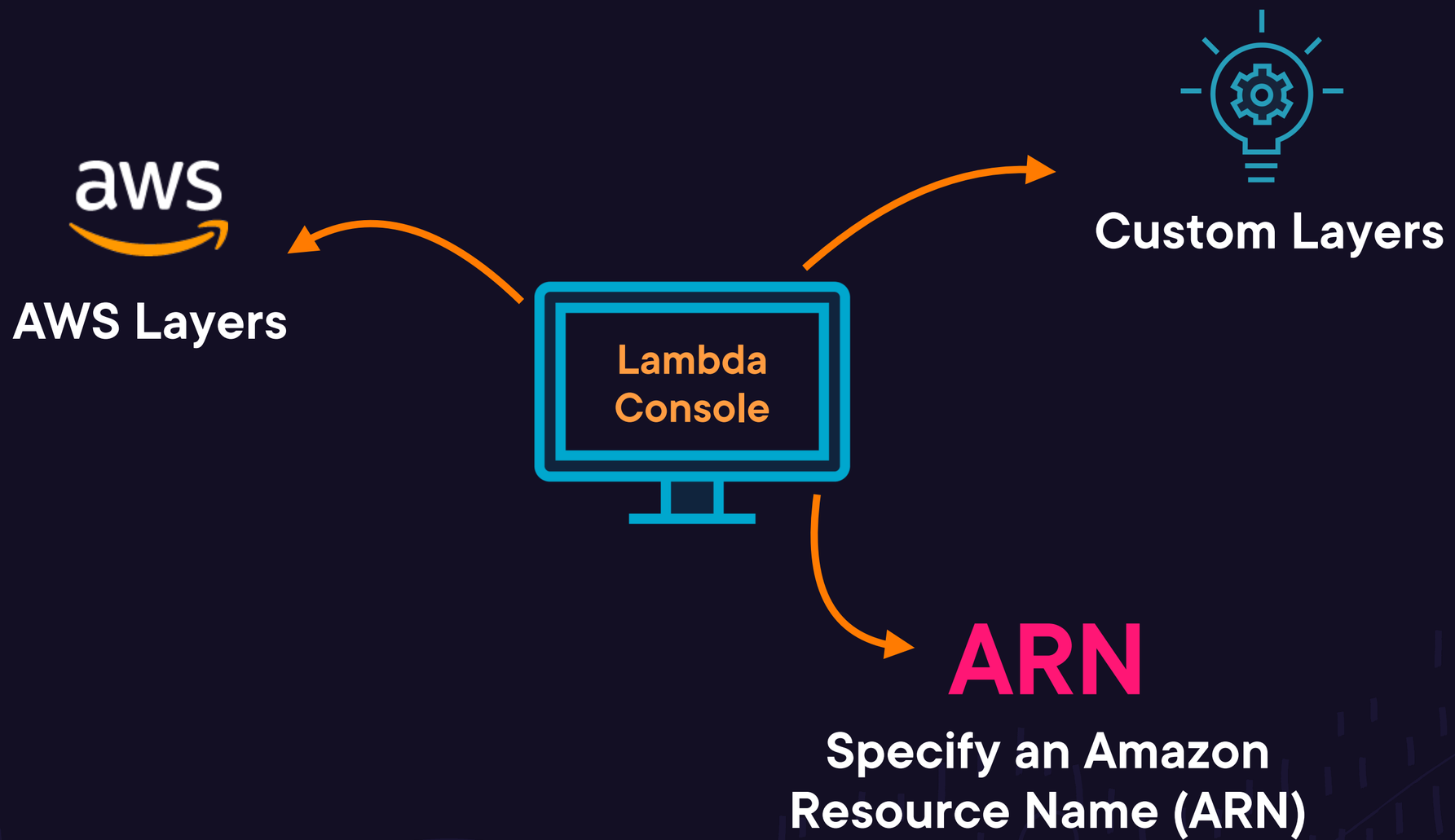
Why Layer Up? Smaller Deployment Packages



Challenges

- Invoking and testing functions locally becomes harder.
- You won't be able to update the function if:
  - The layer is deleted
  - Permissions to use the layer are revoked
  - Third party retires an older version of their library
- The total unzipped size of the function + all layers < 250 MB.

Lambda Console: Ways to Import Lambda Layers



Demo

Lambda Layers

- Download the requests library and create a ZIP folder.

- Use the package to create a new custom layer in Lambda.

Lesson Summary

What Are Lambda Layers?

- Enables importing packages into functions
- Packages can contain additional code, third-party libraries, config files, or custom runtime environments

Why Layer Up?

- Code reusability
- Smaller packages= faster deployments

Challenges

- Testing is harder
- Updating functions can get tricky
- Unzipped function + layers < 250 MB

Lesson Summary

Demo

- Created a custom layer for the requests library

Demo: Converting JSON to CSV Using Lambda

Noreen Hasan
Training Architect

LESSON BREAKDOWN

Demo: Converting JSON to CSV Using Lambda

What's in a File Type?

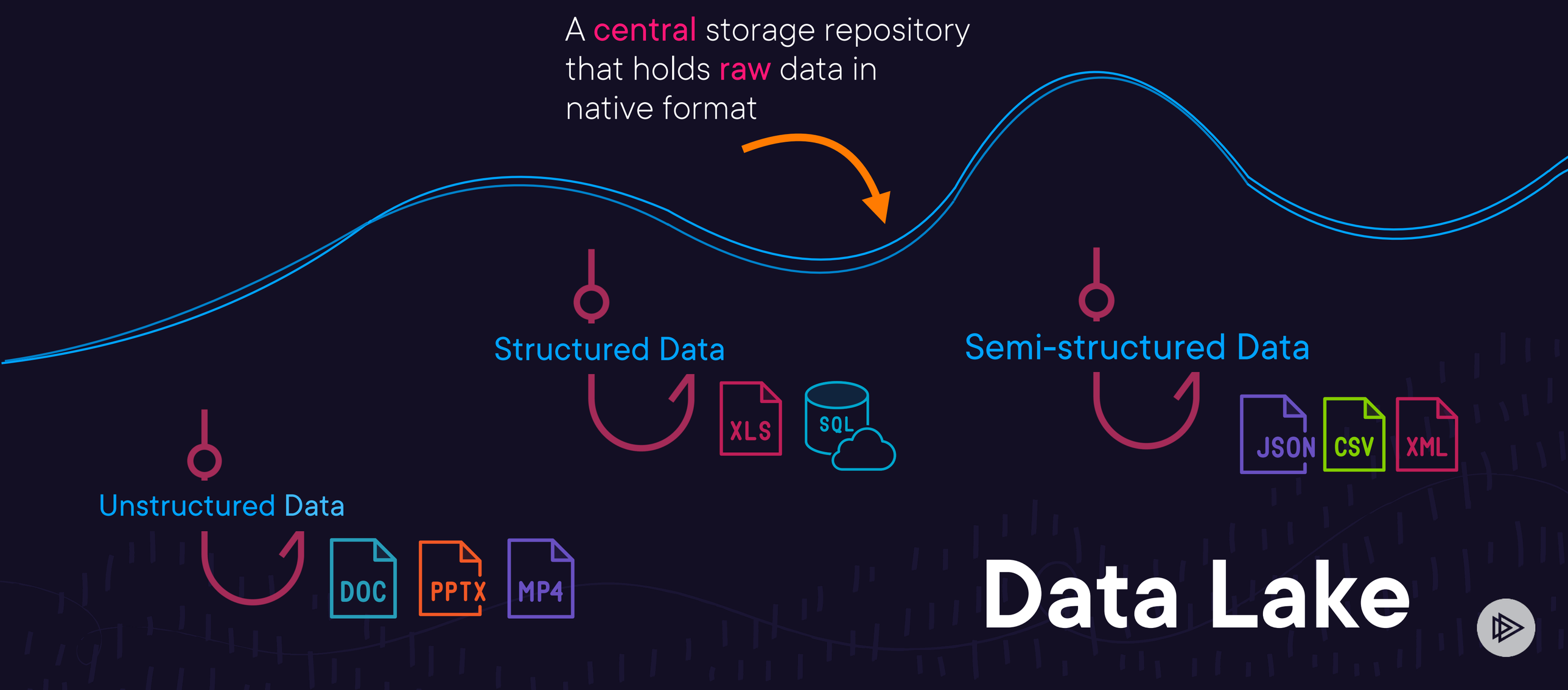
The Workflow

Demo

Lesson Summary

Noreen Hasan
Training Architect

What's in a File Type?



JSON vs. CSV

Semi-structured data

Larger file size

Less secure

Supports hierarchical relational data

vs.

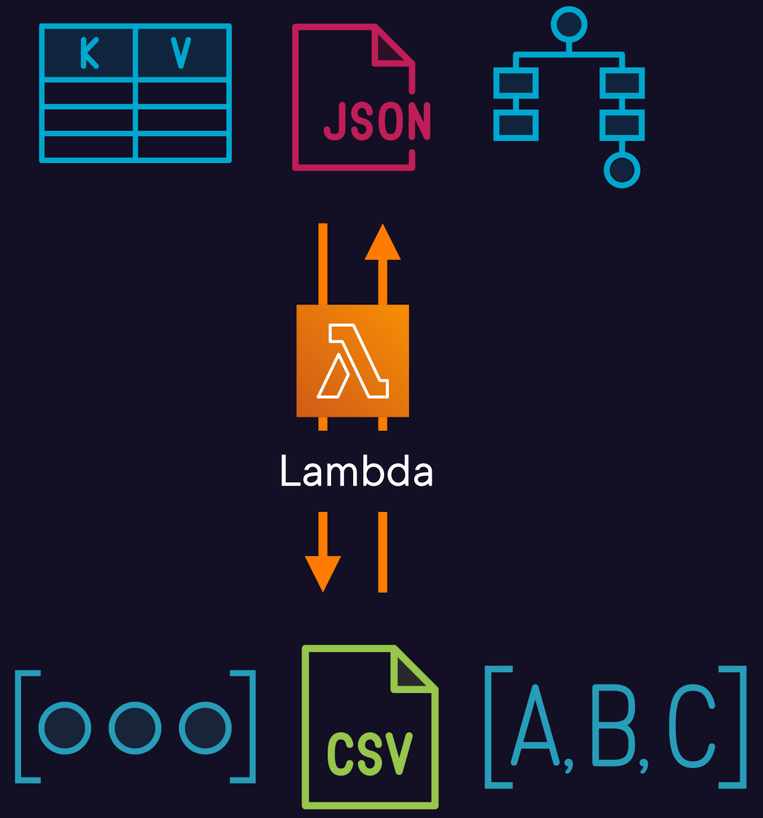
Semi-structured data

More compact

More secure

Doesn’t support hierarchical structure

JSON vs. CSV Use Cases



JSON

- Ideal for exchanging structured data

- Easily scalable and suitable for large datasets

- The compact size and readable nested structure makes it ideal for APIs

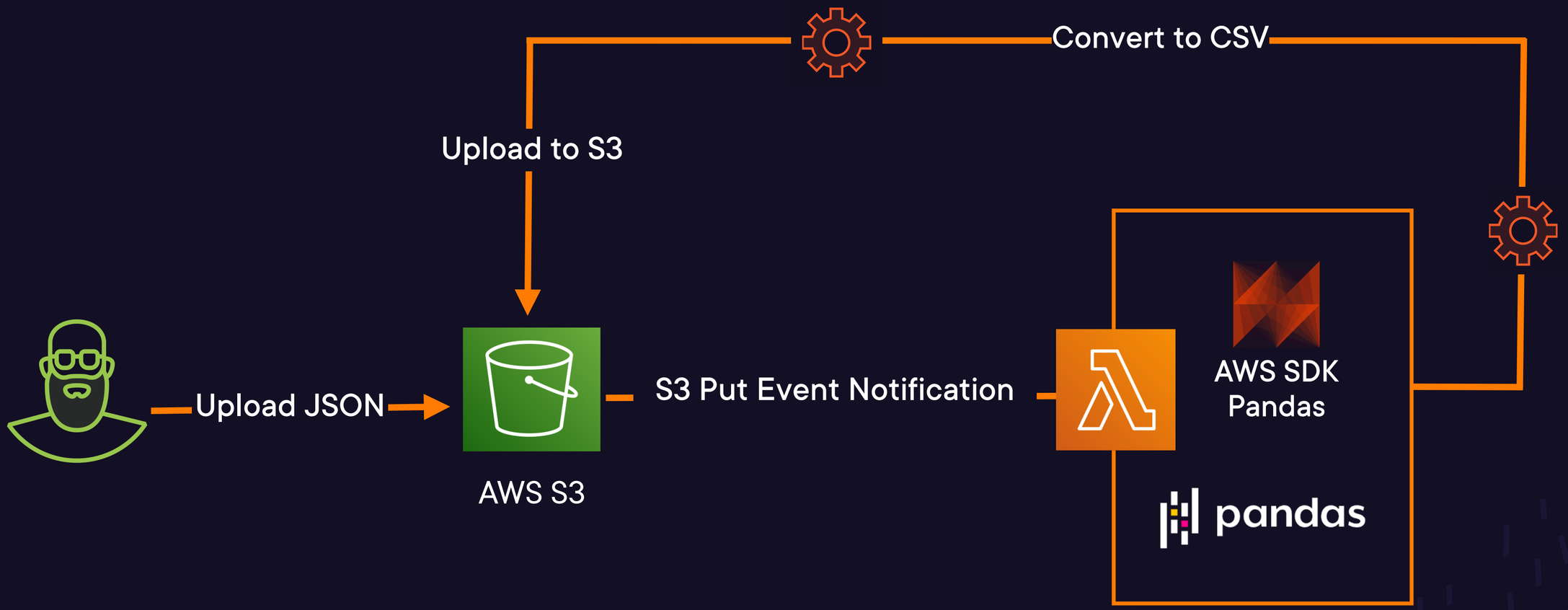
CSV

- Ideal for storing tabular data in a delimited text file

- Convenient for small datasets

Demo: Converting JSON to CSV Using Lambda

The Workflow



Demo

Lambda Function + Layer

The function code will use the AWS SDK for Pandas library to convert JSON files to CSV files.

IAM Policy

Allow Lambda to list the S3 buckets, read, and upload S3 objects to the bucket.

S3 Bucket

Add S3 Put Event Notification to trigger Lambda each time a JSON file is uploaded to S3.

Lesson Summary

What's in a File Type?

- How structured, unstructured, and semi-structured data fit in a data lake
- JSON versus CSV

The Workflow and Demo

- Trigger Lambda when {JSON} files are uploaded to S3
- Function: Convert JSON files to CSV

Demo: Ingesting CSV Files from S3 to DynamoDB

Noreen Hasan
Training Architect

LESSON BREAKDOWN

Demo: Ingesting CSV Files from S3 to DynamoDB

DynamoDB: Overview

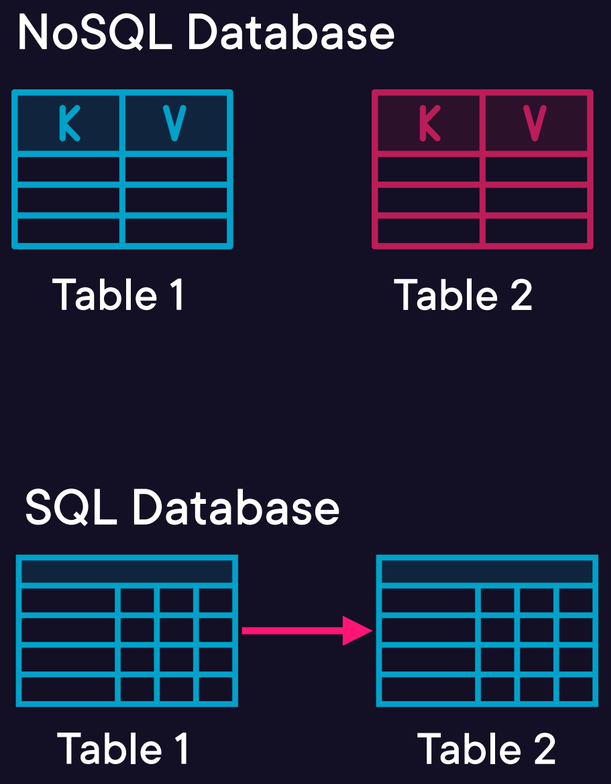
The Workflow

Demo

Lesson Summary

Noreen Hasan
Training Architect

DynamoDB: Overview



NoSQL Database

- Schema-less: it doesn't enforce relationships between the tables.

- It supports key-value and document data models, so every table must have a unique key.

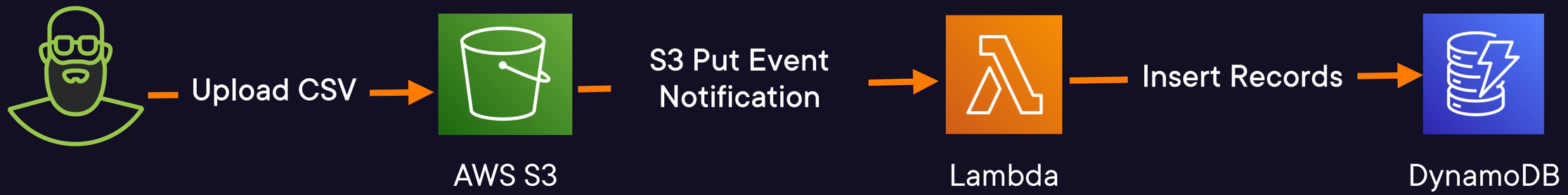
- DynamoDB can support semi-structured data such as CSV and JSON.

Fully Managed

- AWS takes care of operating and scaling the database behind the scenes.

Demo: Ingesting CSV Files from S3 to DynamoDB

The Workflow



Demo

DynamoDB

Create a DynamoDB table to store the records.

S3 Bucket

Create an S3 bucket and add an S3 Put event notification to trigger Lambda each time a CSV file is uploaded to S3.

Lambda Function

Create a function that inserts the records from the CSV file to the DynamoDB table.

IAM Policy

Allow Lambda to access S3 and DynamoDB.

Lesson Summary

DynamoDB: Overview

- NoSQL DB
- Fully managed

The Workflow and Demo

- Function: insert records to a DynamoDB table
- Trigger: a CSV file is uploaded to S3

Processing Real-Time Data Using Kinesis and Lambda: Overview

Noreen Hasan
Training Architect

LESSON BREAKDOWN

Processing Real-Time Data Using Kinesis and Lambda: Overview

Streaming Data: Use Cases

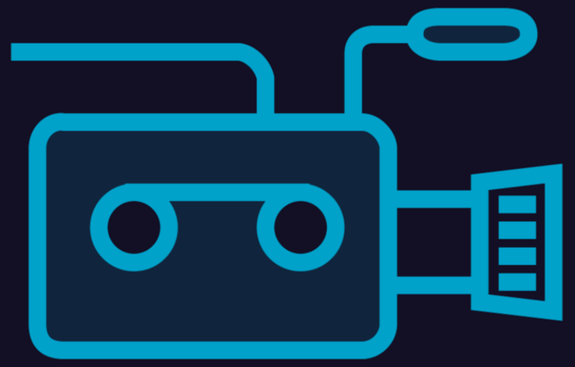
The Kinesis Umbrella

Kinesis Data Streams

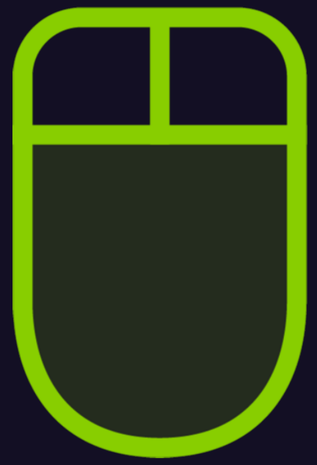
Lesson Summary

Noreen Hasan
Training Architect

Streaming Data: Use Cases



Video Processing



Clickstream Events

Data Lakes

The Kinesis Umbrella

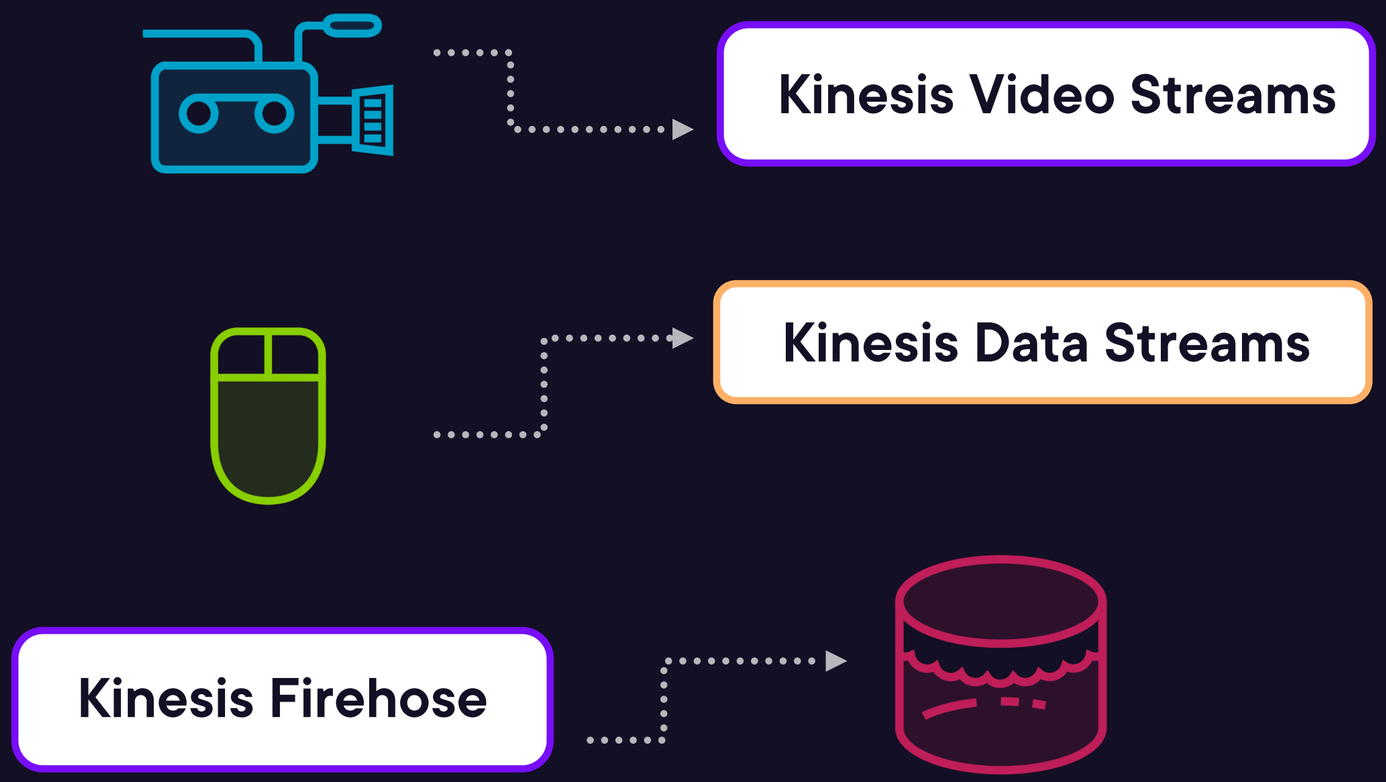
Amazon Kinesis Data Streams
Store, process, and aggregate data streams.

Amazon Kinesis Video Streams
Store and process video streams.

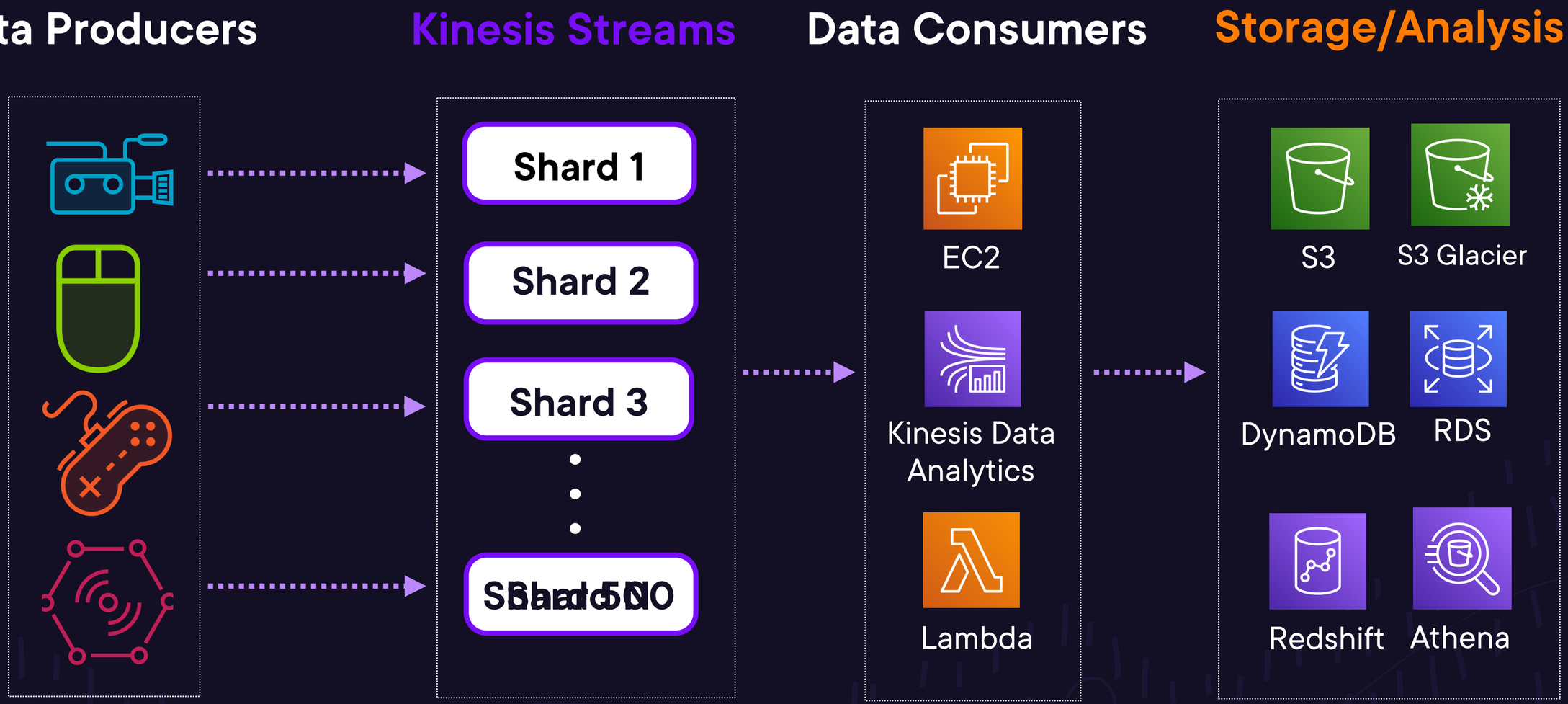
Amazon Kinesis Data Analytics
Use SQL to analyze data streams.

Amazon Kinesis Data Firehose
Load data streams into AWS data stores.

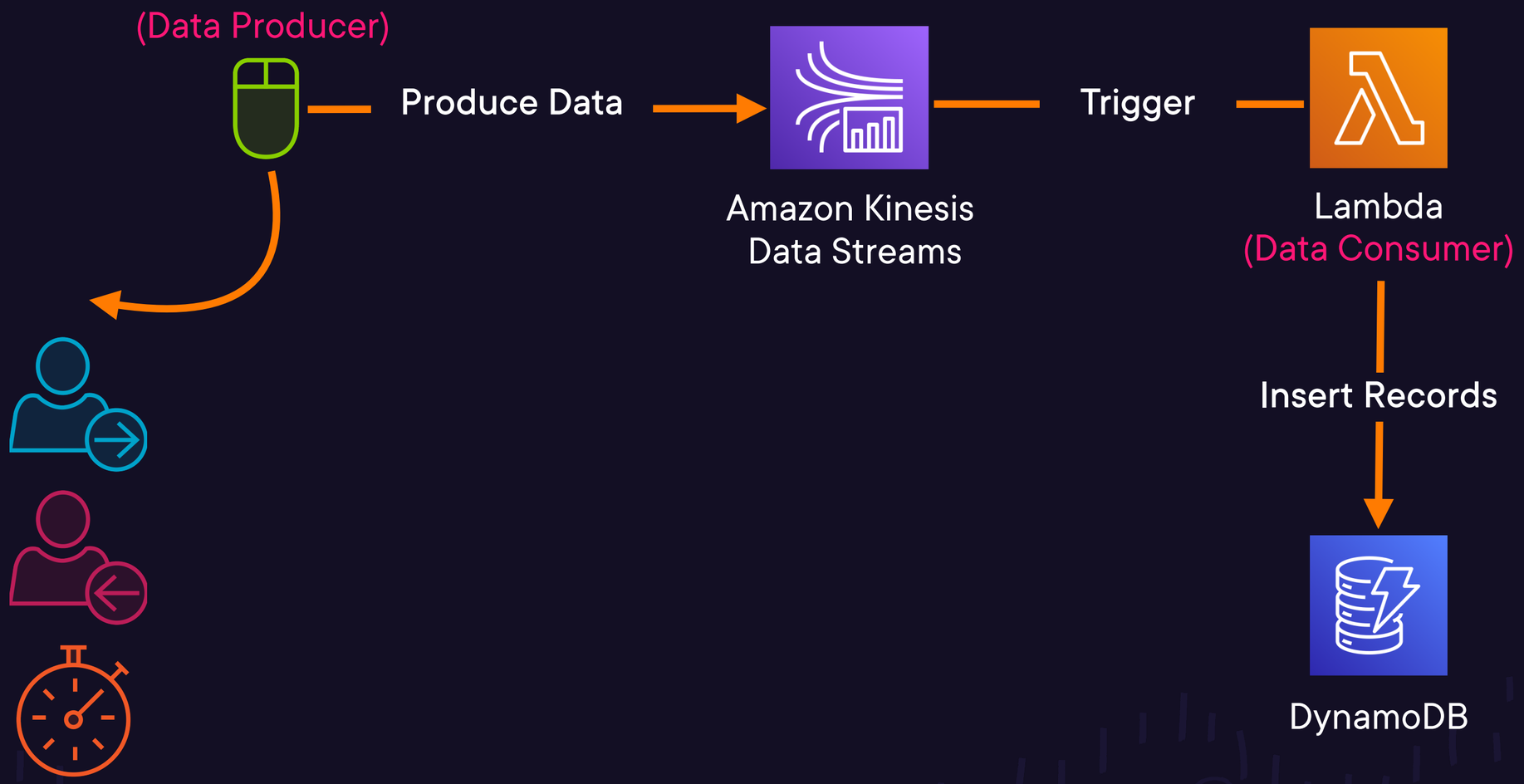
Use Cases and the Kinesis Umbrella



Kinesis Data Streams



Kinesis Data Streams: Sample Workflow



Lesson Summary

Streaming Data: Use Cases

- Video processing
- Clickstream events
- Data lakes

The Kinesis Umbrella

- Kinesis Data Streams
- Kinesis Video Streams
- Kinesis Data Analytics
- Kinesis Data Firehose

Main Components of Kinesis Data Streams

- Data producers
- Kinesis streams
- Data consumers

Demo: Processing Real-Time Data Using Kinesis and Lambda

Noreen Hasan
Training Architect

LESSON BREAKDOWN

Demo: Processing Real-Time Data Using Kinesis and Lambda

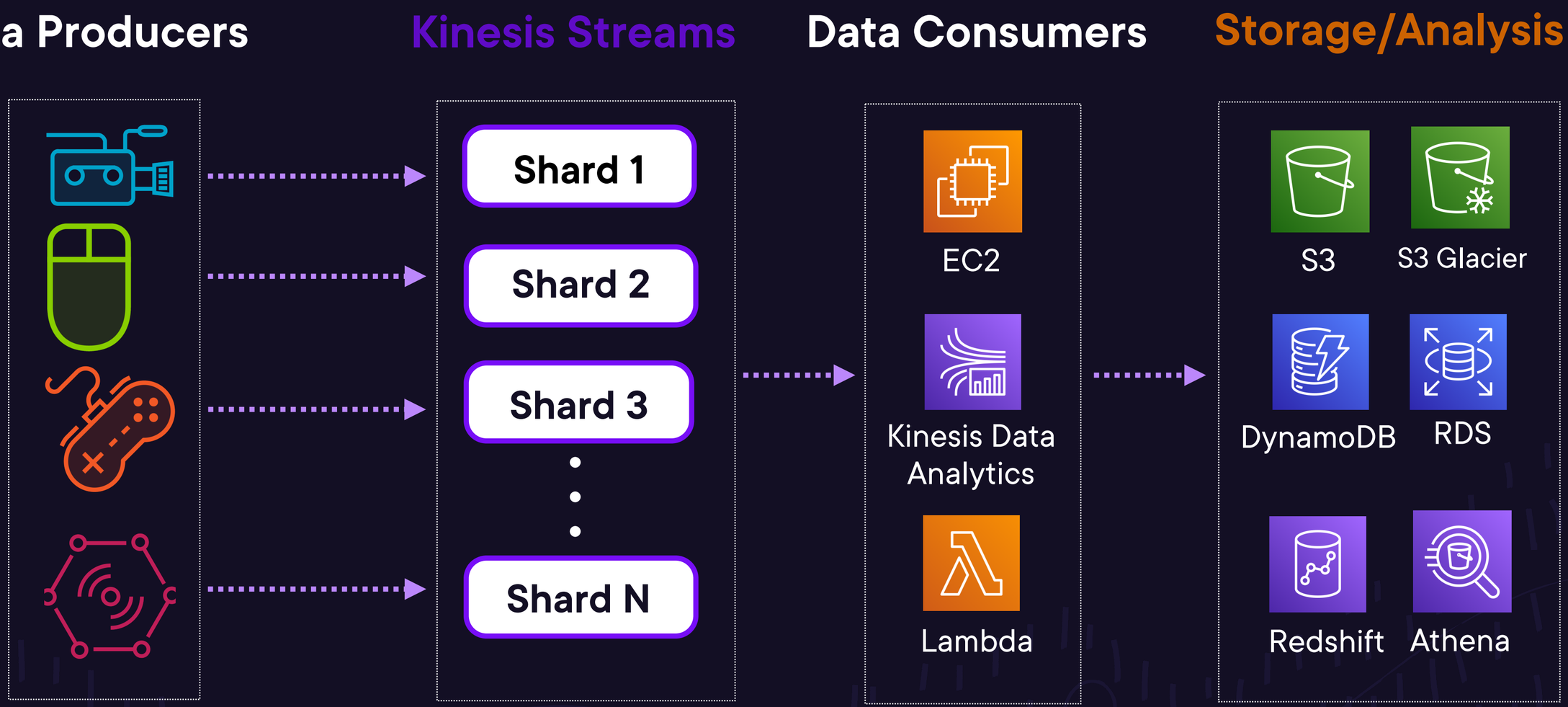
Kinesis, Lambda, and Shards

Demo

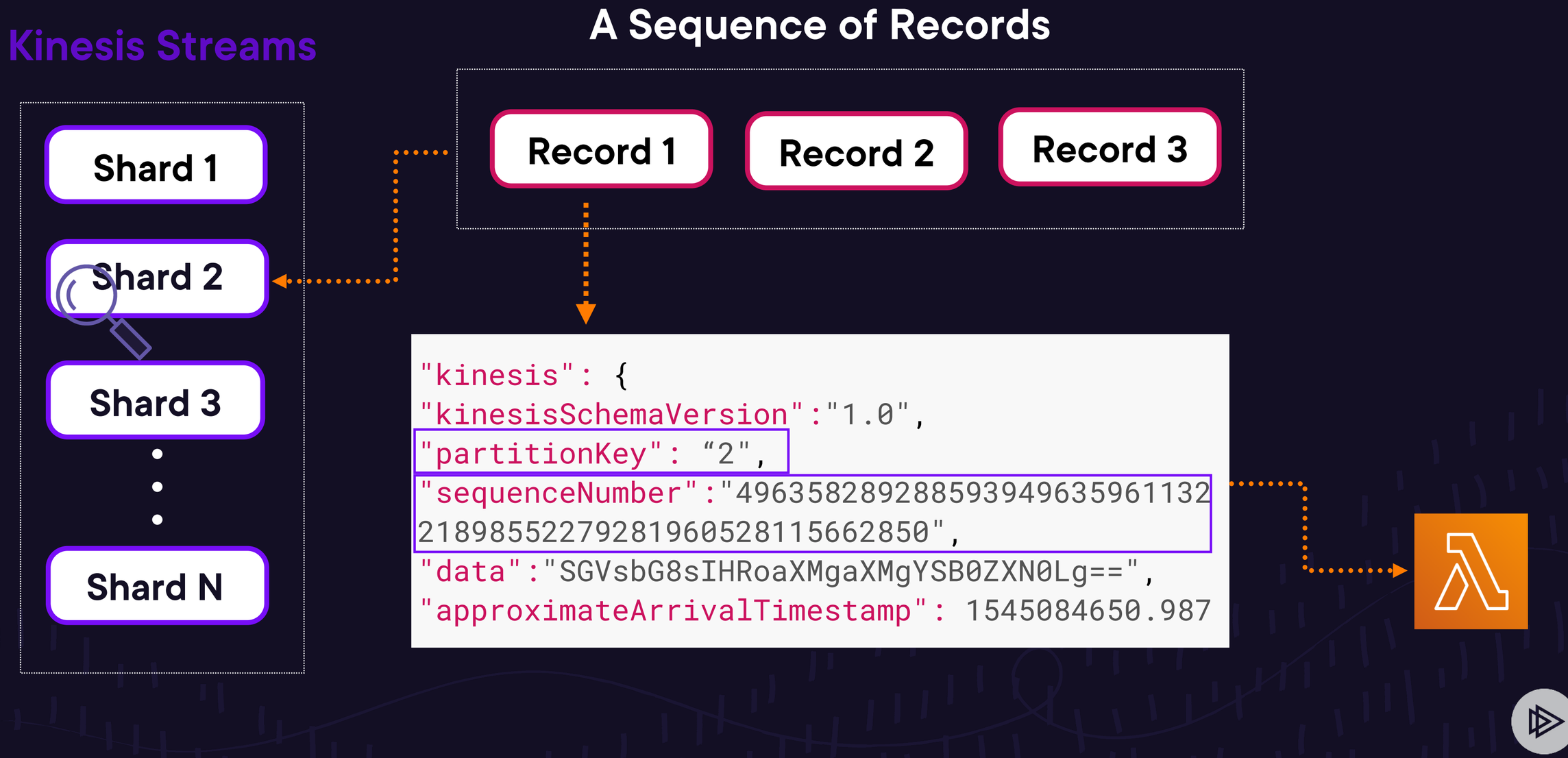
Lesson Summary

Noreen Hasan
Training Architect

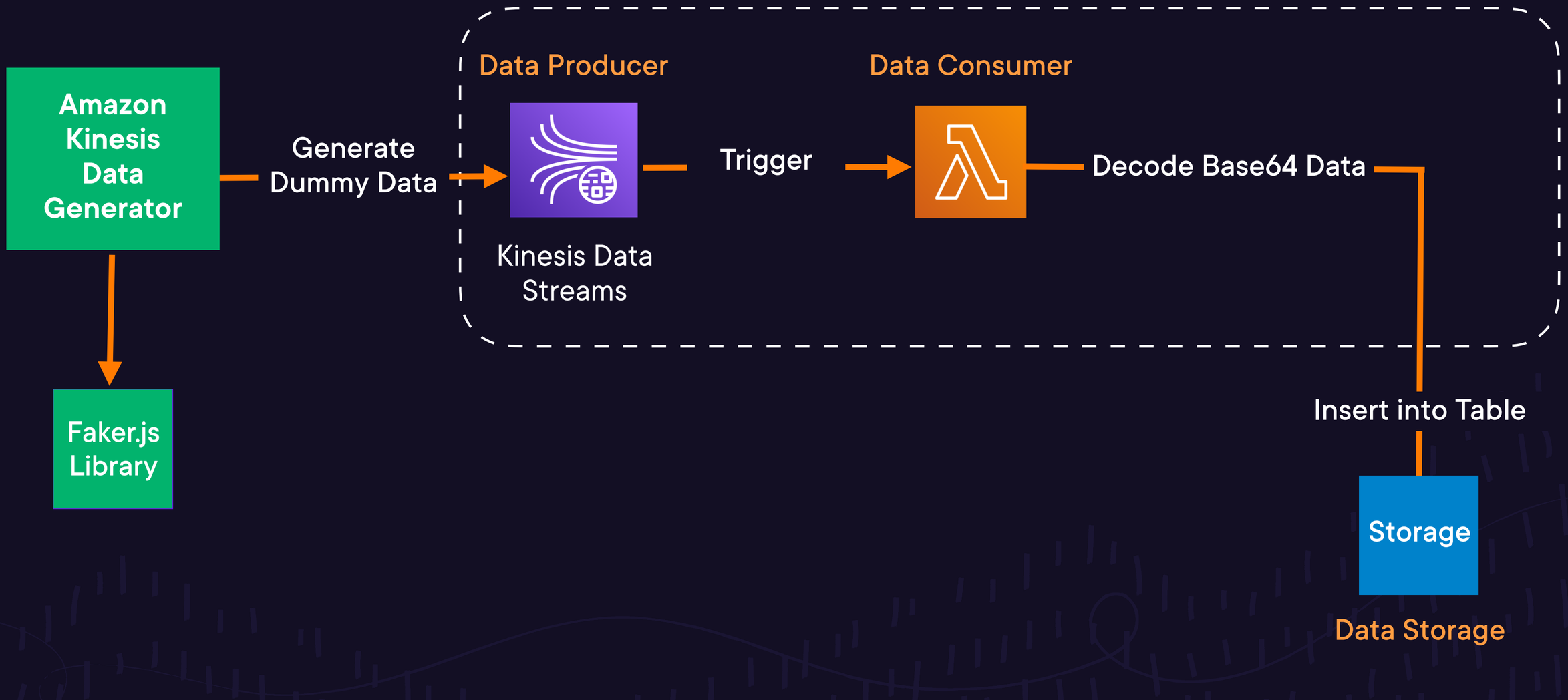
Kinesis, Lambda, and Shards



Kinesis Data Streams: Shards and Lambda



The Workflow: Decode Shards



Demo

Kinesis + Lambda

Create a Kinesis Data Stream and Lambda function to decode the **incoming records**.

IAM Policy

Allow Lambda access to Kinesis Data Streams.

CloudWatch Logs

Utilize CloudWatch to view the data stream records.

Lesson Summary

Kinesis, Lambda, and Shards

A Kinesis data stream consists of shards, which consist of a set of decodable records.

The Workflow and Demo

Created a Lambda function to decode Base64 data coming from Kinesis data streams.

Testing and Debugging Lambda Failures

Noreen Hasan
Training Architect

LESSON BREAKDOWN

Testing and Debugging Lambda Failures

- Invocation Types
- Types of Failures
- Testing
- Monitoring
- Lesson Summary

Noreen Hasan
Training Architect

Synchronous Invocation

- The invoking service waits for a response from Lambda.
- Lambda returns a response to the invoking service.

Lambda's Retry Behavior: None

The client (i.e., invoking service) is in charge of retries on failure.

Asynchronous Invocation

- The invoking service doesn't wait for response from Lambda.
- The invoking service is only in charge of delivering the event, Lambda handles the rest.

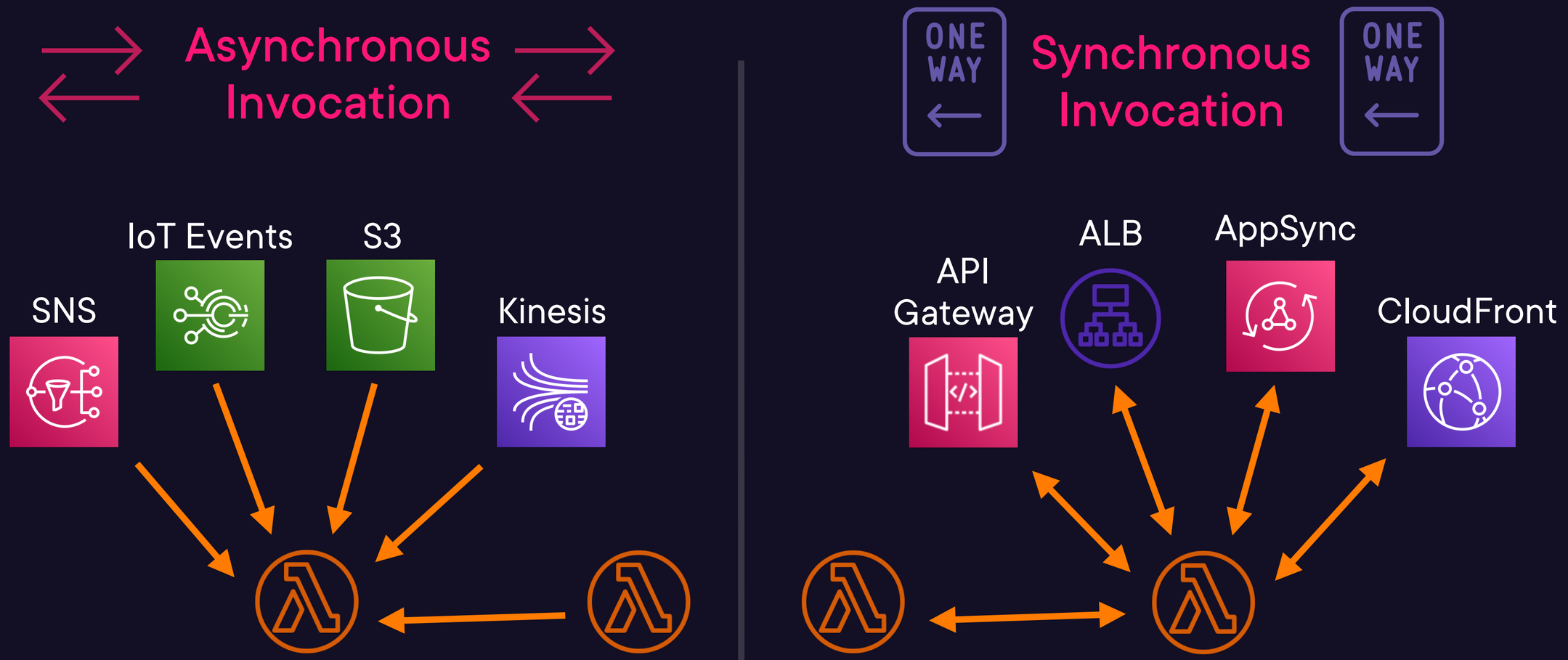
Lambda's Retry Behavior: 0-2 Attempts

- Lambda is in charge of retries.
- Extra charges.
- You can configure MaximumRetryAttempts.

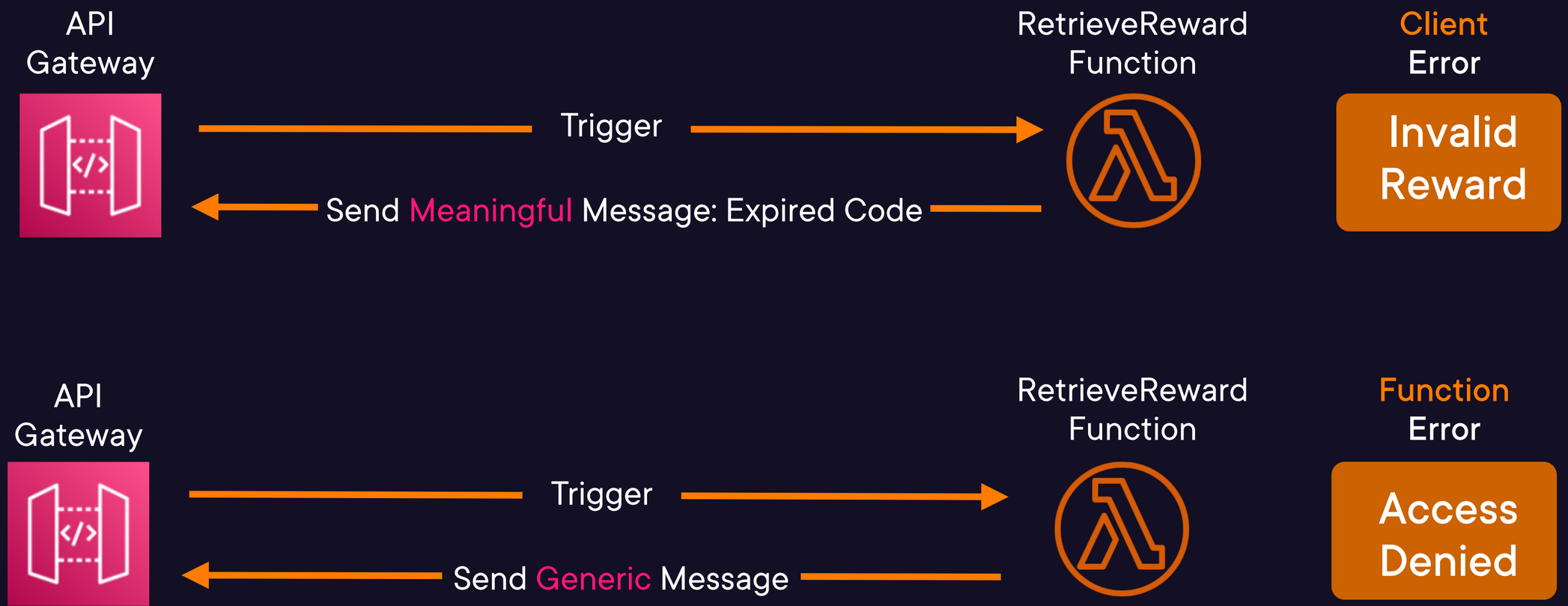
Asynchronous Invocation

- Offers the option to route results to destinations
  - OnSuccess
  - OnFailure
- No additional code needed
- Possible Destinations:
  - Amazon SNS
  - Amazon SQS
  - AWS Lambda

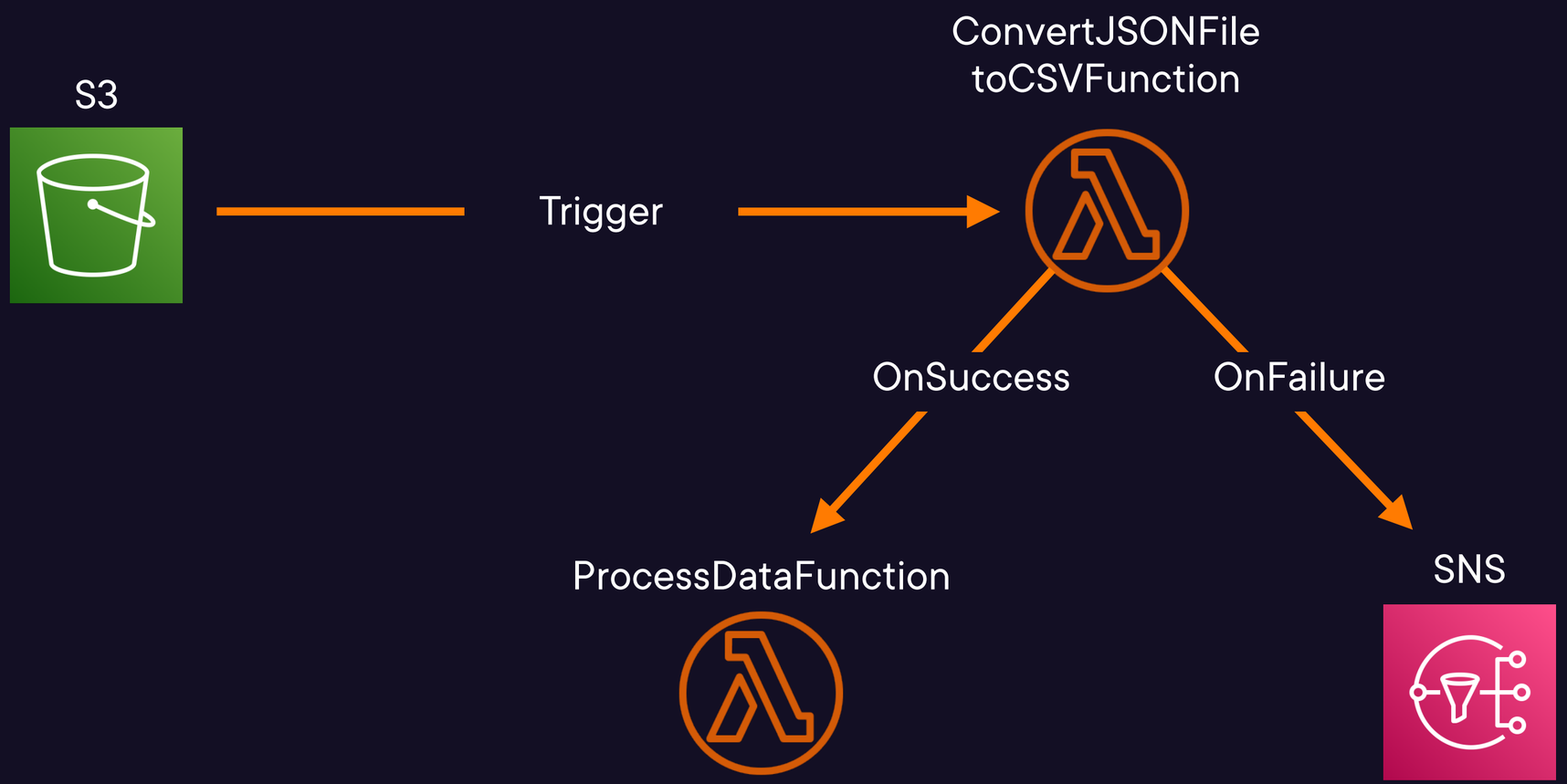
Invocation Types: Sample Services



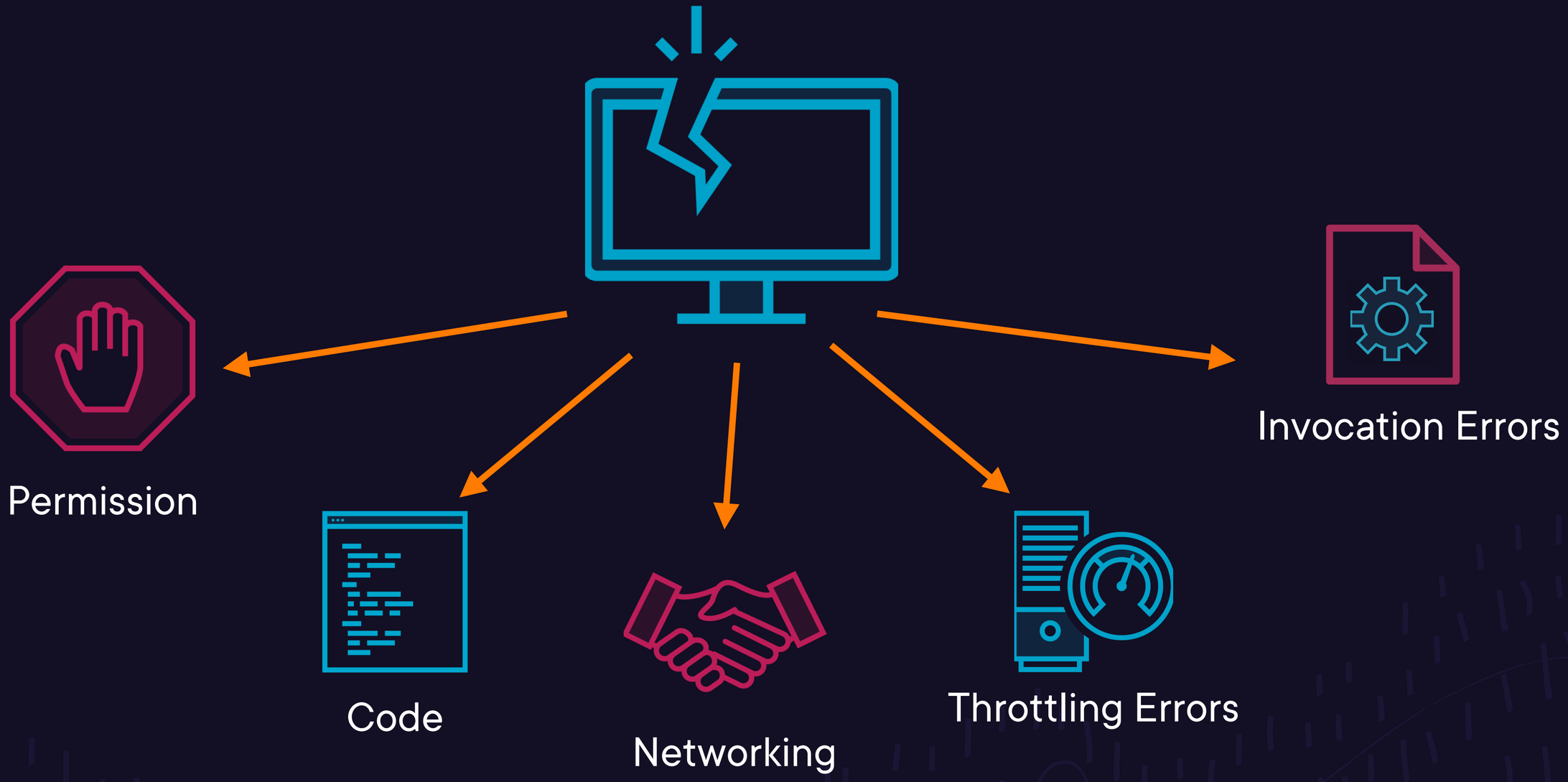
Synchronous Invocation: Handling Errors



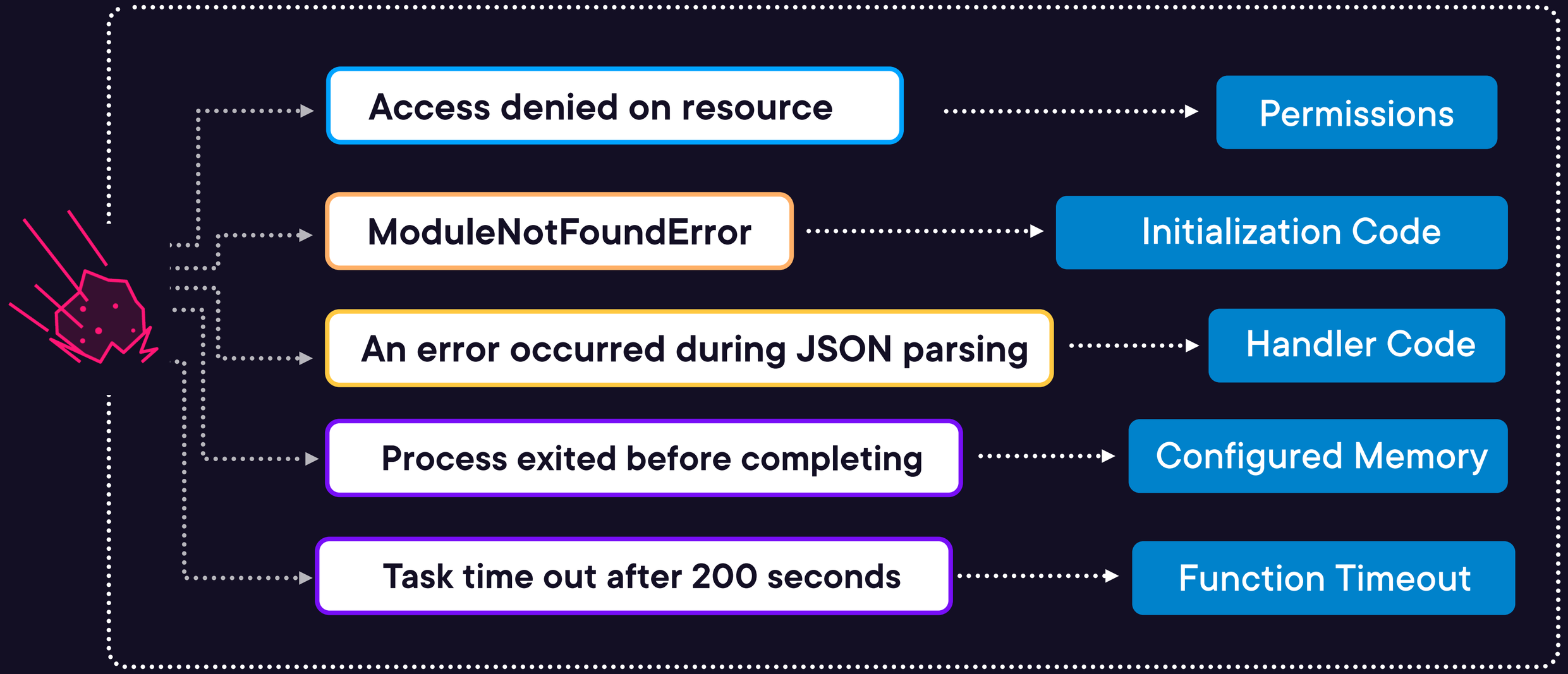
Asynchronous Invocation: Handling Errors



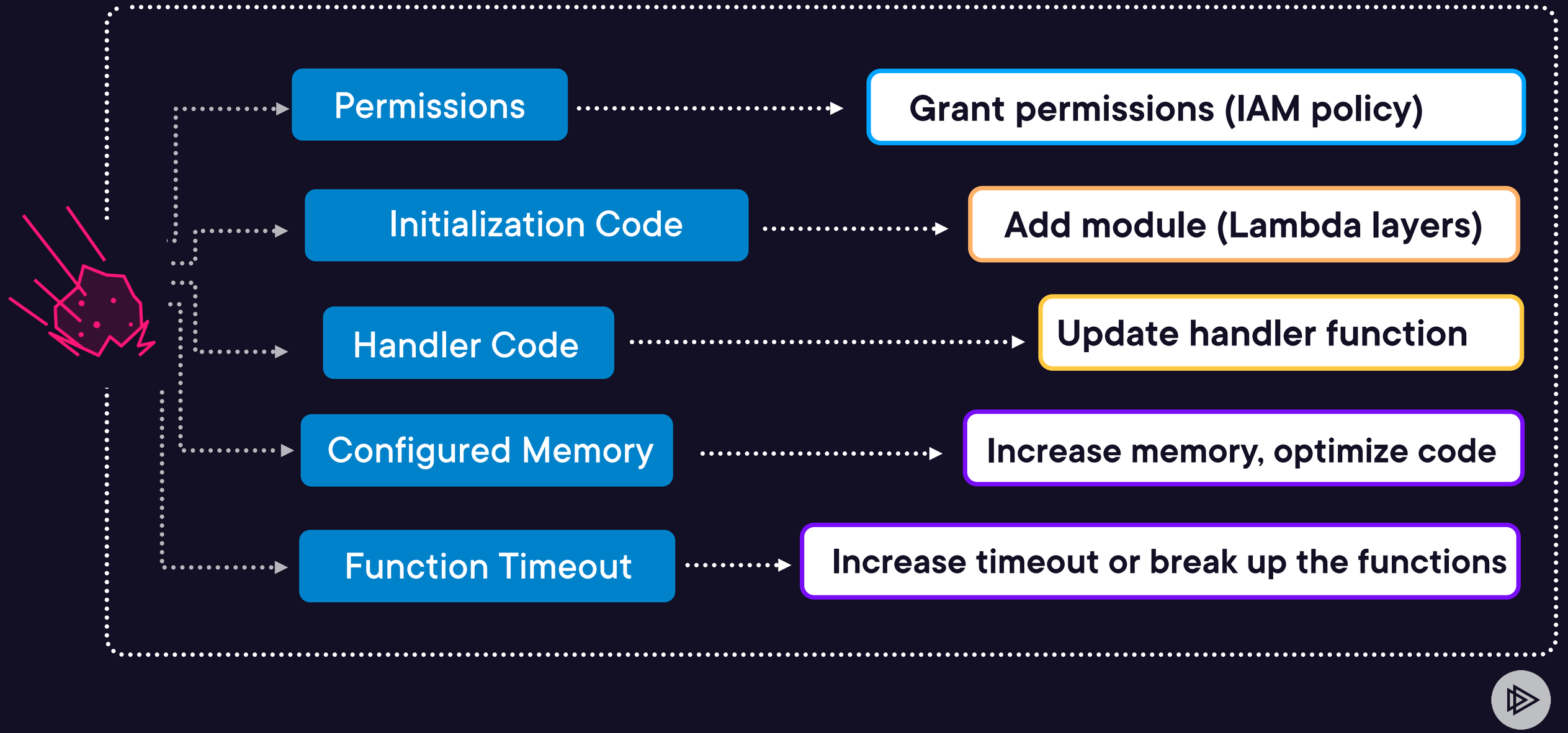
Types of Failures



Common Lambda Errors



Common Lambda Errors



Testing Environments

In a Deployed Lambda Function

Locally

Simulating Production

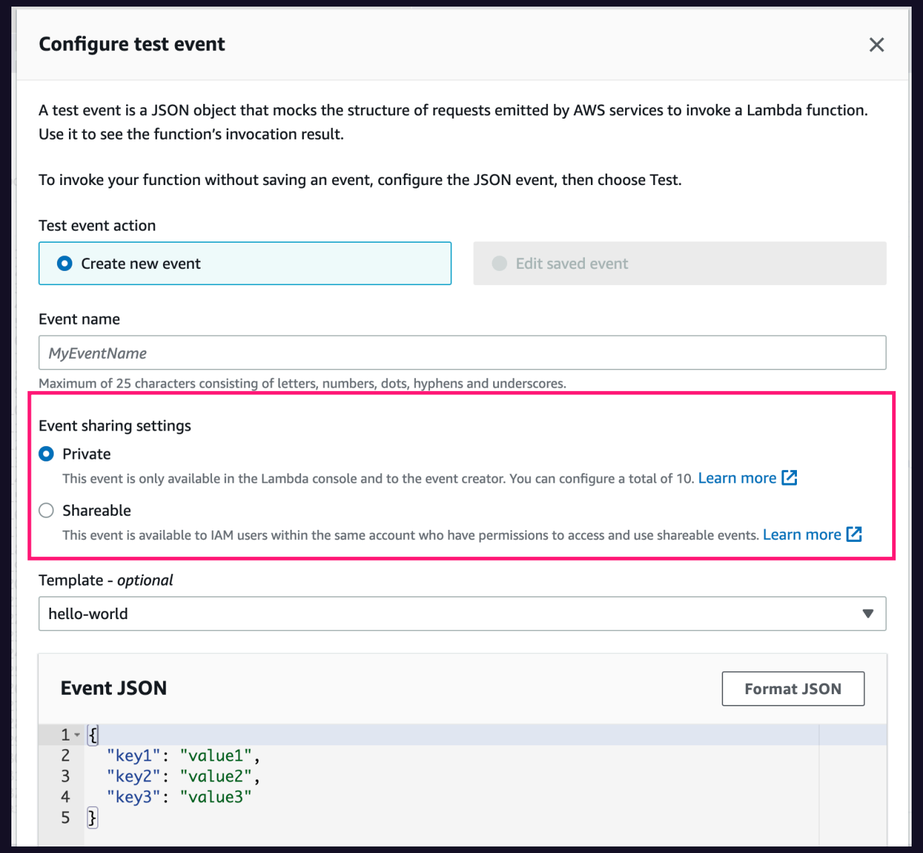
Using Test Events in Lambda Console

AWS SAM

Amazon EventBridge

Testing and Debugging Lambda Failures

Lambda Console: Testing



- from the AWS Lambda console

Test Events

Private

- Can only be accessed by the event creator
- You can create up to 10 private test events/function

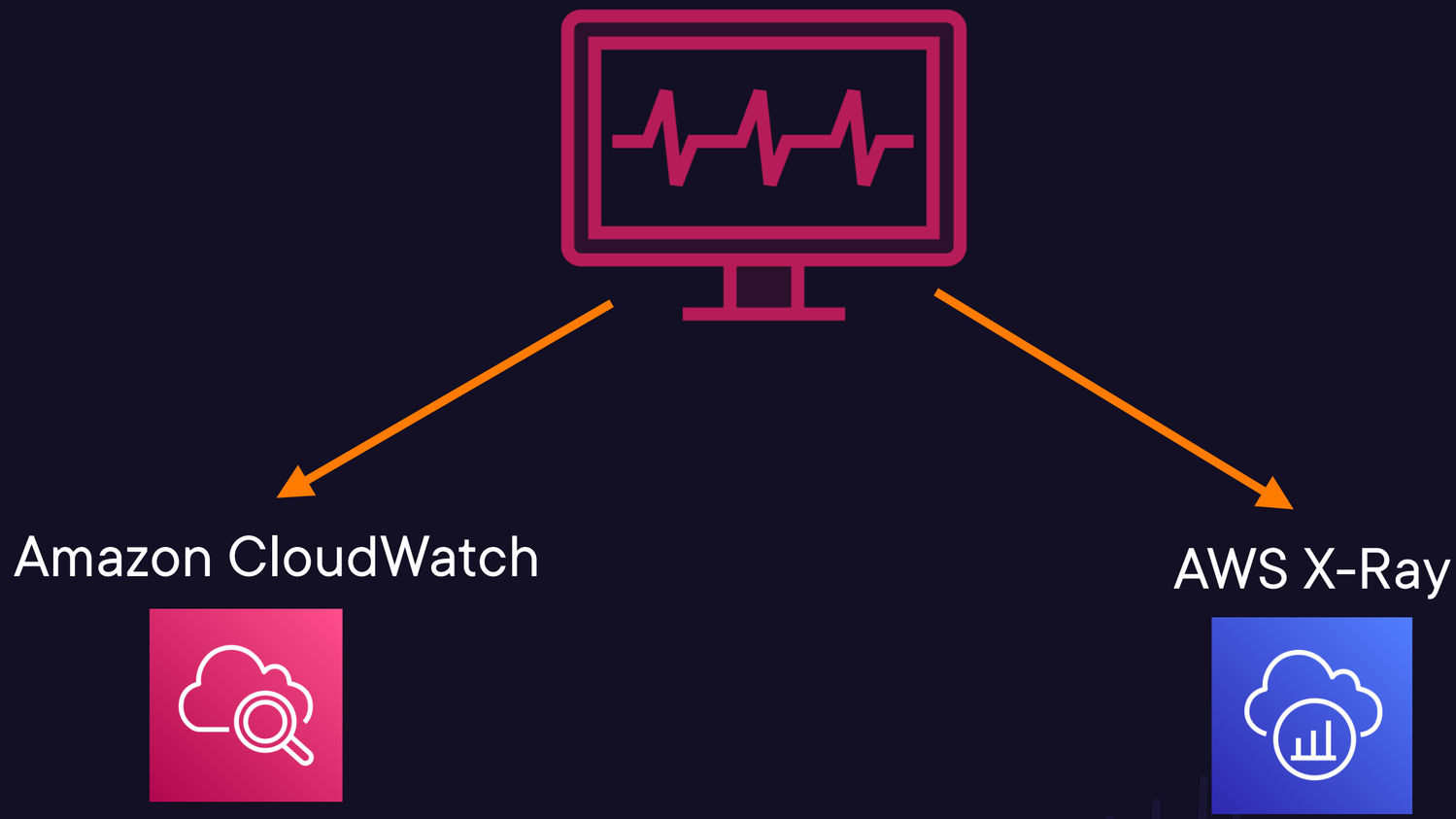
Shareable

- Shared with other IAM users in the same account
- Lambda saves shareable test events as schema in an Amazon EventBridge schema registry

Testing and Logging Goals

- Log errors in a consistent way.
- Enable alarms on necessary metrics.
- Synchronous Invocation: Send back sensible error messages to the invoking service.
    - Lambda sends back a meaningful error message when the client benefits from being informed about the issues.
    - Lambda sends back a generic error message when the issue can't be resolved by the client (i.e., throttles).

Monitoring Tools



Things Worth Tracking

Invocation Metrics

The sum of the below metrics:

- Errors
- Throttles
  - (i.e., TooManyRequestsException)
- DestinationDeliveryFailures

Performance Metrics

The average or max of the below metrics:

- Duration
- IteratorAge

Lesson Summary

Invocation Types

- Synchronous
- Asynchronous

Types of Failures

- Different types of failures (permission, code, networking, etc.)
- Common Lambda errors

Testing

- Environments (Lambda console, AWS SAM, and EventBridge)
- Testing goals and types of test events

Monitoring

- Monitoring tools
- Metrics

Processing Data with Lambda: Recap

Noreen Hasan
Training Architect

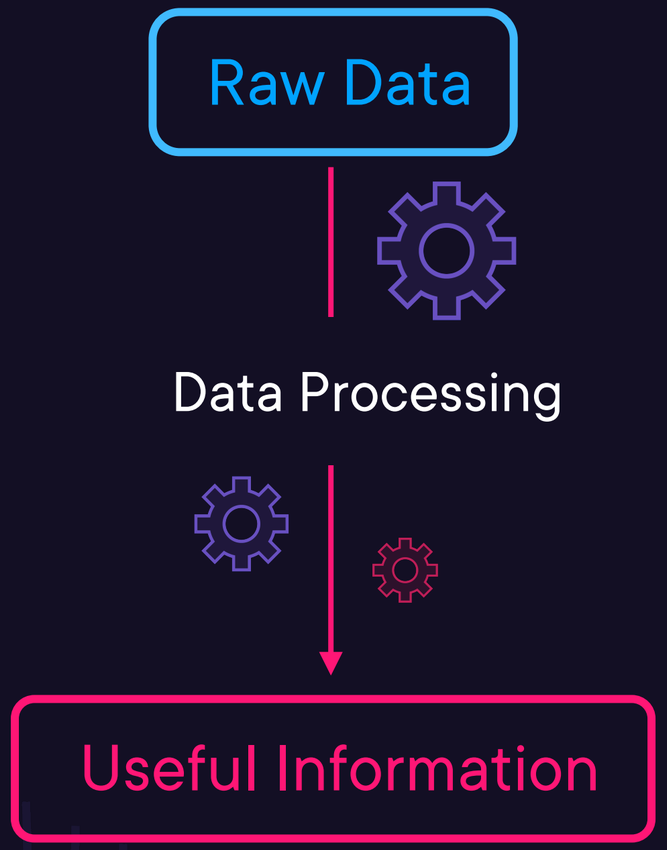
SECTION BREAKDOWN

Lessons Summary

Processing Data with Lambda: Recap

Noreen Hasan
Training Architect

Processing Data: Overview



What Is Data Processing?

- Converting raw data to information

Data Lakes

- A central repo with raw data
- ETL versus ELT

Types of Data Transformation

- Data filtering
- Data mapping
- Deriving variables
- Aggregating data
- Deduplicating data
- Splitting data

Understanding Lambda Layers

What Are Lambda Layers?

- Enables importing packages into functions
- Packages can contain additional code, third-party libraries, config files, or custom runtime environments

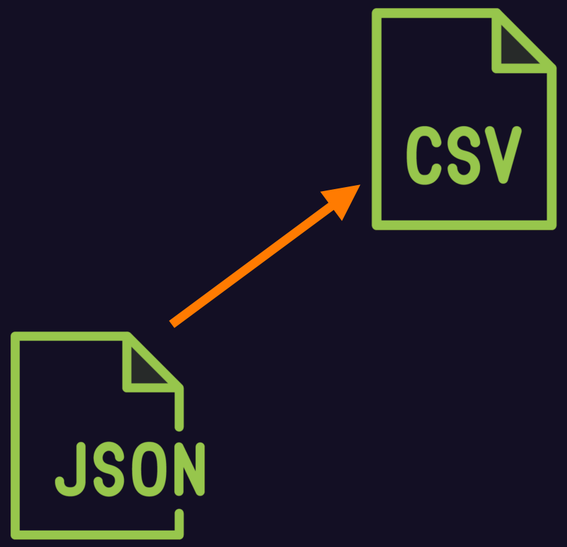
Why Layer Up?

- Code reusability
- Smaller packages= faster deployments

Challenges

- Testing is harder
- Updating functions can get tricky
- Unzipped function + layers < 250 MB

Demo: Converting JSON to CSV Using Lambda



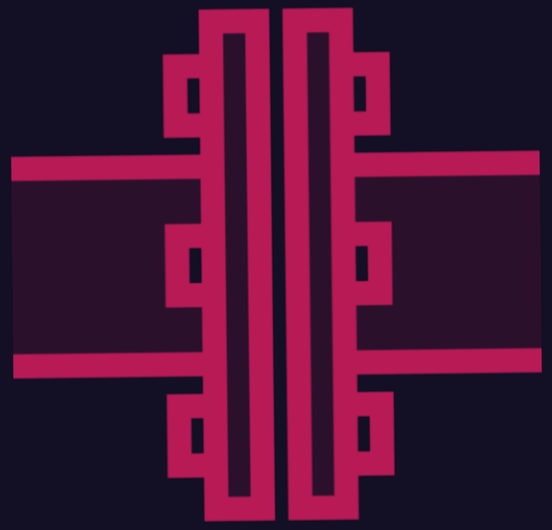
What's in a File Type?

- How structured, unstructured, and semi-structured data fit in a data lake
- JSON versus CSV

The Workflow and Demo

- Trigger Lambda when {JSON} files are uploaded to S3
- Function: Convert JSON files to CSV

Demo: Ingesting CSV Files from S3 to DynamoDB



DynamoDB: Overview

- NoSQL DB
- Fully managed

The Workflow and Demo

- Function: insert records to a DynamoDB table
- Trigger: a CSV file is uploaded to S3

Processing Real-Time Data Using Kinesis and Lambda: Overview

Streaming Data: Use Cases

- Video processing
- Clickstream events
- Data lakes

The Kinesis Umbrella

- Kinesis Data Streams
- Kinesis Video Streams
- Kinesis Data Analytics
- Kinesis Data Firehose

Main Components of Kinesis Data Streams

- Data producers
- Kinesis streams
- Data consumers

Demo: Processing Real-Time Data Using Kinesis and Lambda

Kinesis, Lambda, and Shards

A Kinesis data stream consists of shards, which consist of a set of decodable records.

The Workflow and Demo

Created a Lambda function to decode Base64 data coming from Kinesis data streams.

Testing and Debugging Lambda Failures

Invocation Types

- Synchronous
- Asynchronous

Types of Failures

- Different types of failures (permission, code, networking, etc.)
- Common Lambda errors

Testing

- Environments (Lambda console, AWS SAM, and EventBridge)
- Testing goals and types of test events

Monitoring

- Monitoring tools
- Metrics

See you in the next section!