AWS-Lambda

Angel

Lambda

What is Lambda?

A serverless compute service that runs code in response to events.

Key characteristics

- No servers to manage
- Pay only for execution time
- Event-driven
- Automatically scales

s3->SNS → SQS(queue)
(10)(“*********angeltech.csv uploaded, **********”)→ Lambda → S3 → Logs(cloudwatch)

Multiple sources(5) → SQS(5) → Lambda(

1. How many message in Sqs
2. If sqs = 5
3. Trigger downstream service

) → EMR CLUSTER (Airflow DAG)(ETL)

The service that trigger the lambda functions


Amazon S3


Amazon Simple Email Service


Amazon Kinesis Data Firehose


Amazon Kinesis Data Streams


Amazon DynamoDB


Amazon SNS


Amazon SQS


AWS Config


AWS IoT Button


Amazon Lex


Amazon CloudWatch


AWS CloudFormation


Amazon API Gateway


Amazon CloudFront


Amazon Cognito


AWS CodeCommit

Cost

How AWS Lambda charge?

- **Pay per request** Charged by number of function invocations
- **Pay per execution time** Billed by milliseconds (from start to finish)
- **Pay per memory allocation** CPU scales with memory size

Key Cost Factors

- Invocation count
- Execution duration
- Memory configuration

Important Notes

- No cost when Lambda is not running
- Short, fast executions are cheaper
- High-frequency triggers can become expensive

Generous Free Tier (per month)

- **1 million requests**
- **400,000 GB-seconds of compute time**
  Lambda: 1GB, 1s = 1GB-second
  i. **2GB, 3s = 2x3 = 6GB-second**

After Free Tier

- **$0.20 per 1 million requests**
- **$0.00001667 per GB-second**

1GB, 2s, 1000000

2GB-second

2x 1000000 = 20000000 x 0.00001667 = 334 x 30=

A: 521mb, 10 s

Create Lambda Function

Step 1: Choose Creation Method

- Author from scratch
- Use a blueprint
- Container image

Step 2: Basic Configuration

- Function name
- Runtime (Python / Node.js / Java, etc.)
- Architecture (x86_64 or arm64)

Step 3: Permissions

- Assign execution role
- Grant access to AWS services (S3, SQS, DynamoDB)

Step 4: Advanced Options

- Durable execution (optional)
- Environment variables
- Timeout & memory settings

Amazon SQS (Simple Queue Service)

What is SQS?

- Fully managed message queue service
- Decouples producers and consumers

Why use SQS?

- Buffer traffic spikes
- Improve system reliability
- Enable asynchronous processing

Core Concepts

- Message
- Queue
- Producer / Consumer
- Visibility Timeout

FIFO Queue

- First in, first out – not available in all regions
- Queue name must end with `.fifo`
- Messages are sent exactly once
- Lower throughput (up to 3000 messages/sec with batching)

SQS – Limitations

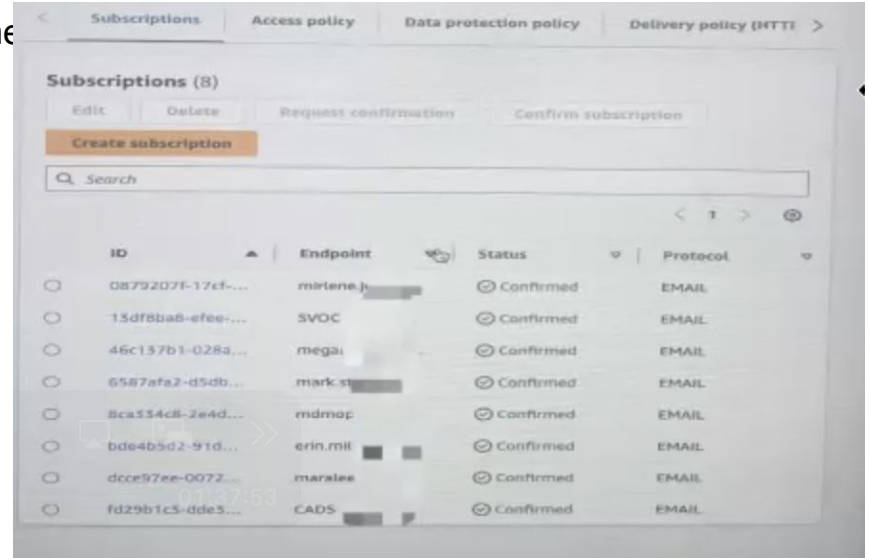
Maximize of 120000 in flight messages being processing by consumers
Batch request has a maximum of 10 messages - max 256 kb
Message content: XML JSON or Unformatted text
FIFO queue support up to 3000 messages per second using batching
Max message size: 256 KB
Data retention: 1min - 14 days
Pricing is calculated by per API request and network usage

Amazon SNS (Simple Notification Service)

- Fully managed **publish–subscribe (pub/sub)** messaging service
- One message → **fan-out** to multiple subscribers
- Decouples message producers from consume

Common Subscribers

- Email / SMS
- SQS
- Lambda
- HTTP / HTTPS endpoints



S3 Event–Driven Architecture

S3 can emit events when objects change

Common event types:

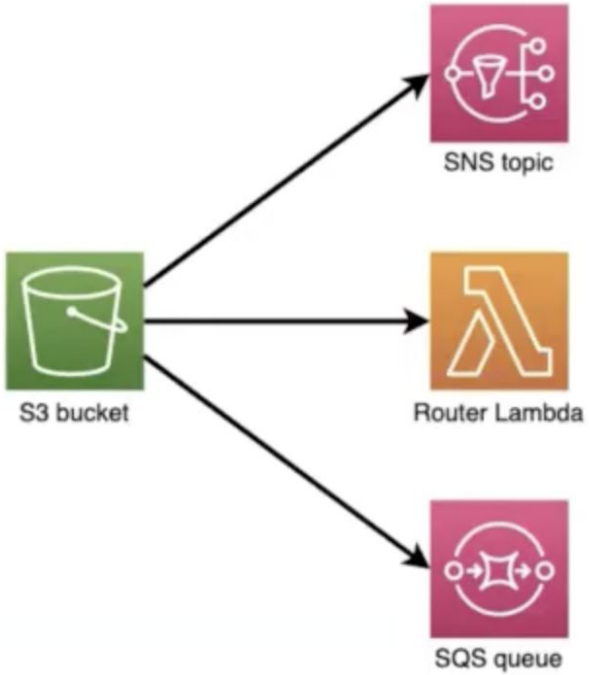
- ObjectCreated (PUT, POST, COPY, CompleteMultipartUpload)
- ObjectRemoved (DELETE)

Events can trigger downstream systems:

- AWS Lambda
- Amazon SQS
- Amazon SNS
- EventBridge

Typical use cases:

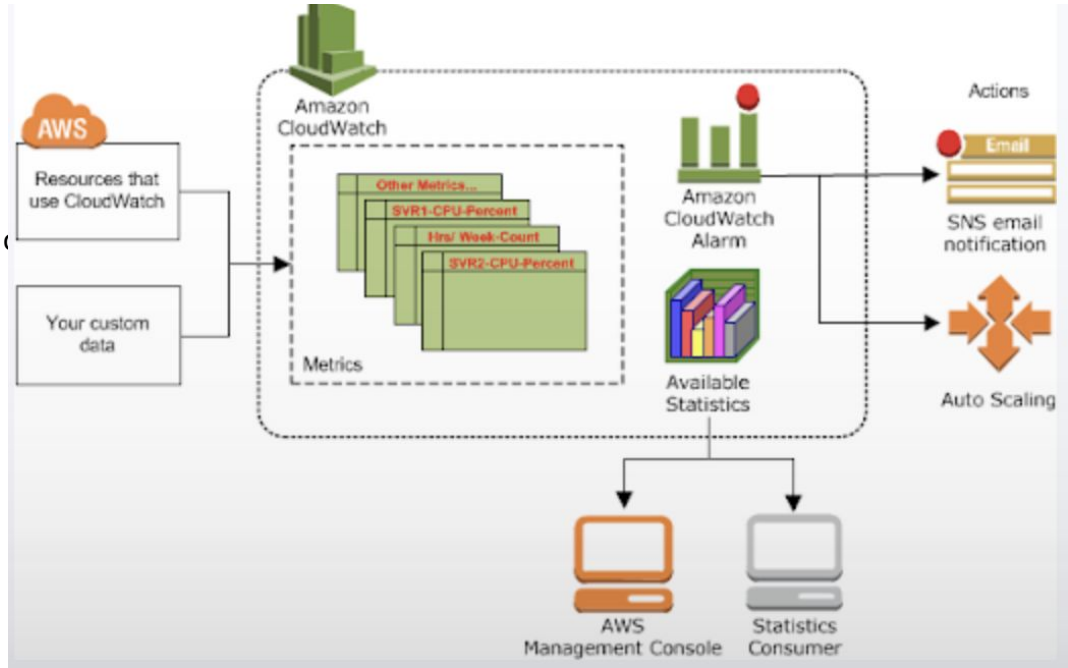
- Trigger data pipelines
- File-driven ETL
- Near real-time processing



Cloudwatch

What is CloudWatch?

- A **monitoring and observability service** provided by AWS
- Collects and tracks **metrics**
- Collects, stores, and searches **log files**
- Enables **alarms and automated actions**
- Provides visibility into the **health and performance**



Typical Workflow

- AWS resources emit metrics & logs
- CloudWatch evaluates thresholds
- Alarms trigger actions (SNS, Auto Scaling, Lambda)