Examarbete

# AWS Lambda

*Fastest way to build modern Applications with the lowest cost of ownership*

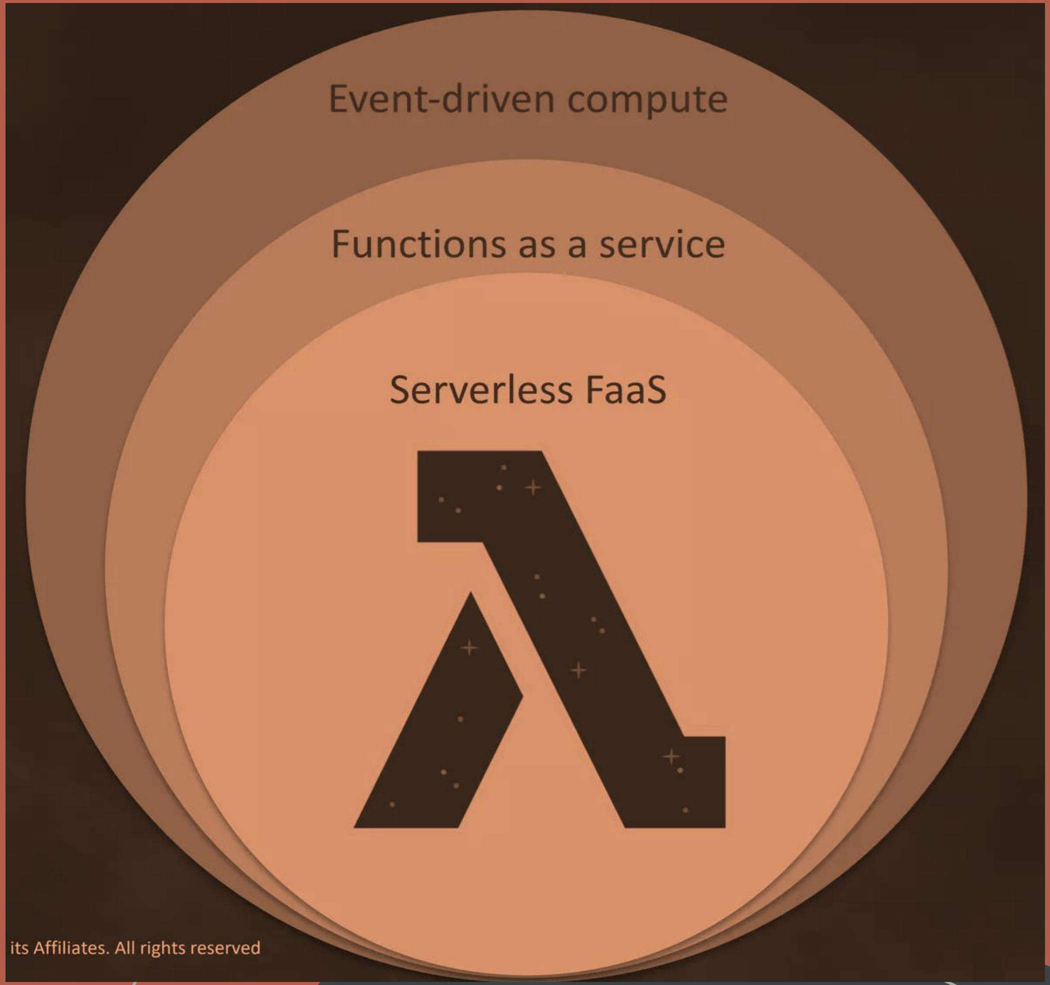
By Amna Sohail

# WHY AWS LAMBDA

# What is AWS Lambda?

*Fastest way to build modern Applications with the lowest cost of ownership*

- ✓ Provide your **Code** or **images**,we run it as an event when things happen
- ✓ Pay by the **millisecond**
- ✓ Packages as **.zip** file(250MB)or **container image**(10MB)
- ✓ **Built-in languages** (Java,Go,Node.js,.Net,Python and Ruby),or bring your own
- ✓ No provisioning or managing **Servers**
- ✓ **Scaleup** in milliseconds in response to traffic



its Affiliates. All rights reserved

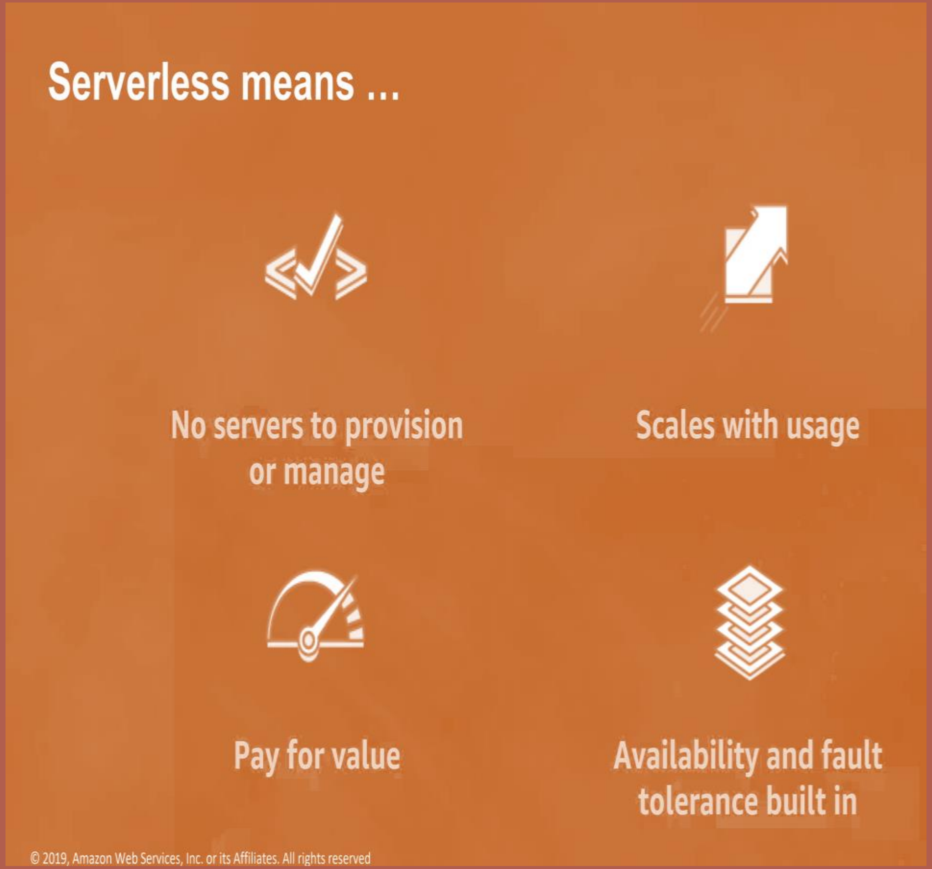
# Serverless adoption  is growing fast

More than a **MILLION** customers uses Lambda every month

More than **10 TRILLION** Lambda invocations per month

# Serverless means….

## General conception



Serverless means ...

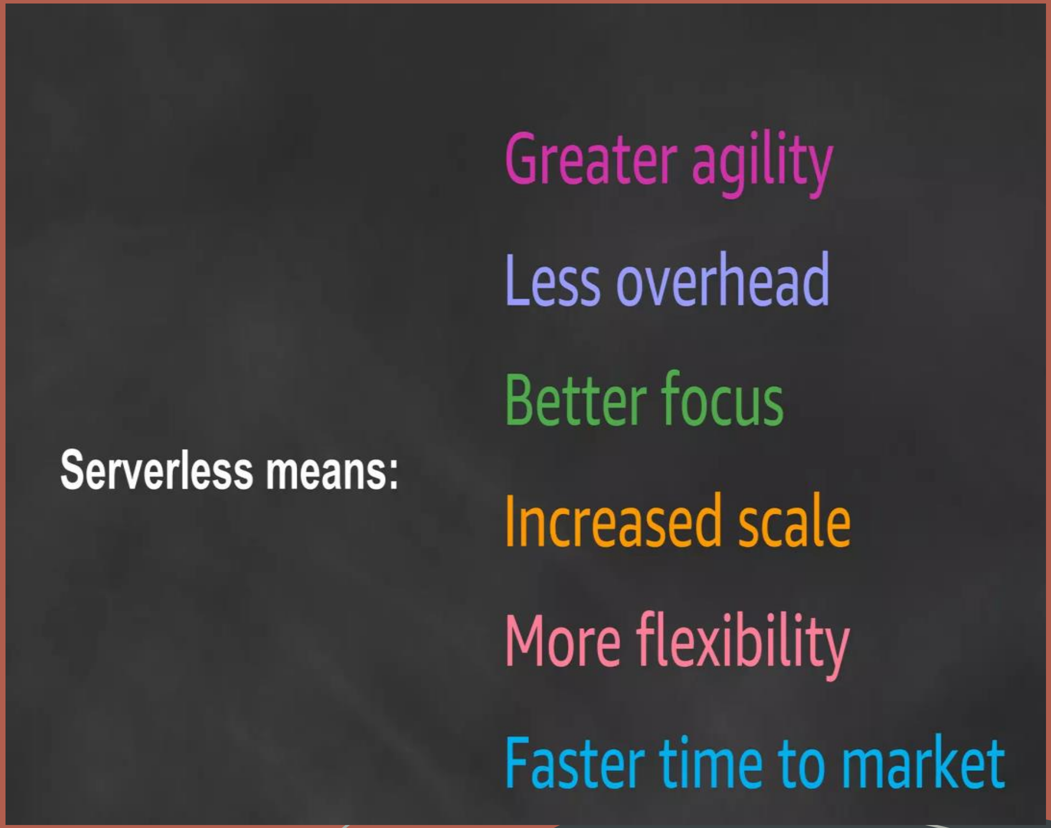
No servers to provision or manage

Scales with usage

Pay for value

Availability and fault tolerance built in

## Klient conception

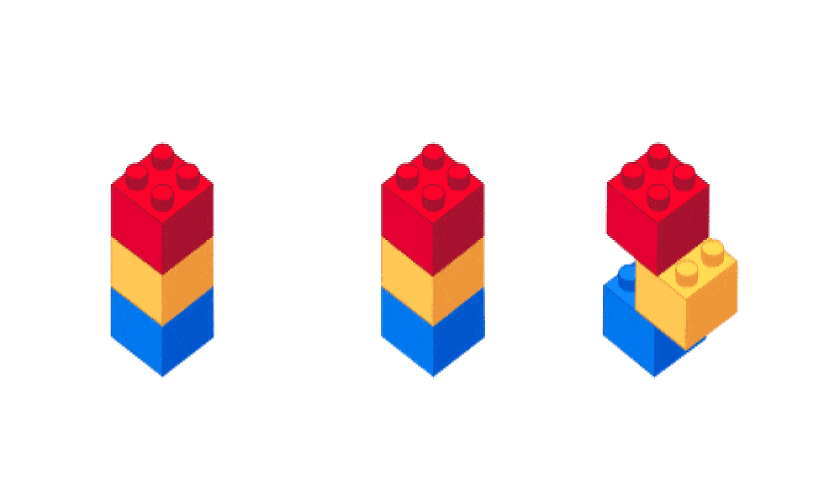


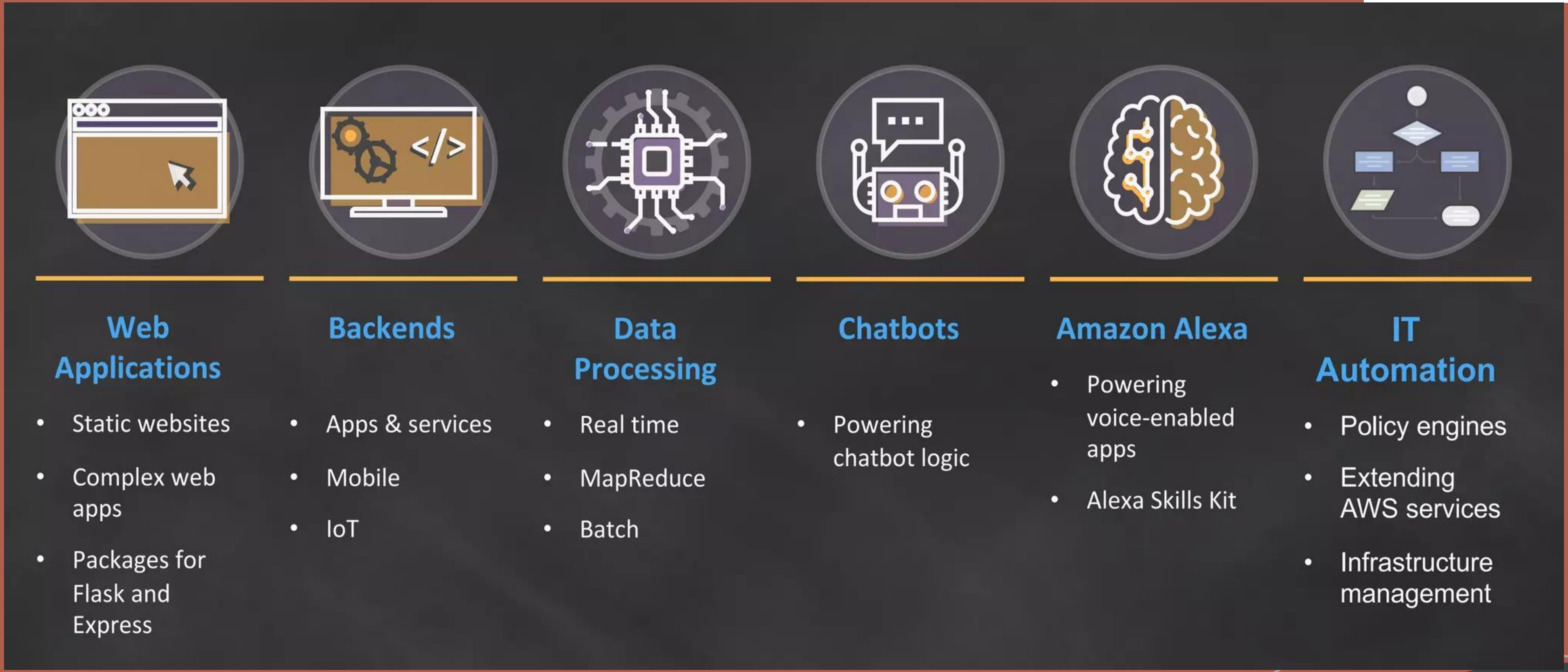
Serverless means:

- Greater agility
- Less overhead
- Better focus
- Increased scale
- More flexibility
- Faster time to market

SERVERLESS
IS "NOT" NO SERVER. IT'S NO SERVER FOR "YOU" TO MANAGE
generator.net

# What are customers building?





### Web Applications

- Static websites
- Complex web apps
- Packages for Flask and Express

### Backends

- Apps & services
- Mobile
- IoT

### Data Processing

- Real time
- MapReduce
- Batch

### Chatbots

- Powering chatbot logic

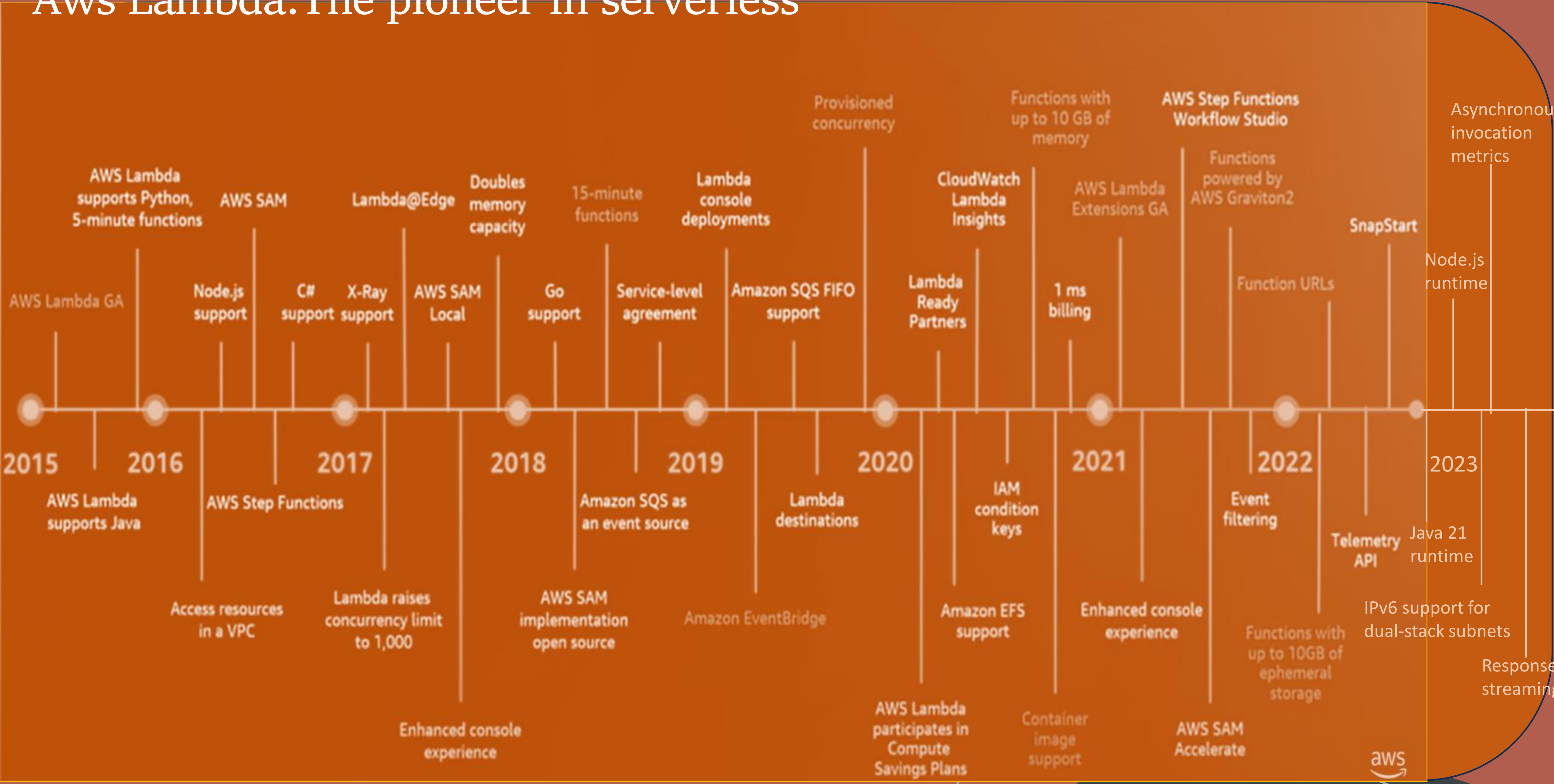
### Amazon Alexa

- Powering voice-enabled apps
- Alexa Skills Kit

### IT Automation

- Policy engines
- Extending AWS services
- Infrastructure management

# Aws Lambda:The pioneer in serverless



# Customers Love AWS Lambda

airbnb · mlbam · COMCAST · Experian · Coca-Cola · LexisNexis · New York Public Library · Benchling · Pentair

instacart · Zillow · ALT · VOGUE · 23andMe · SCHOLASTIC · CITRIX · Mc Graw Hill · The Seattle Times · ADITYA BIRLA GROUP

VidRoll · Localytics · asurion · Expedia · Atlassian · FireEye · vevo · EDIZIONI CONDÉ NAST S.P.A.

THE HONEST CO. · THOMSON REUTERS · AdRoll · SQUARE ENIX · BUSTLE · Nextdoor · ZAPPROVED · Vertafore Unleash your potential · DBRS · AirAsia

FUNNY OR DIE · Genome Institute of Singapore · MORNINGSTAR · Haier · iRobot · ISCS · La Mutuelle Générale · LA NACION · rachio · Discovery COMMUNICATIONS

TTI · FICO · twilio · FINRA · VERACODE · MADSACK MEDIENGRUPPE · BEACHBODY ON DEMAND · AUTODESK · WARBY PARKER

doppler labs · Cornelius · WHALEROCK · cimpress · Nishitetsu · Optimizely · ASTRO DIGITAL · SAISON INSURANCE · Serenova shine bright · nextel TÁ NAS SUAS MÃOS.

# Example: Teevolution is an innovative Swedish technology company

- Known for their golf-related products and services such as SmartGolfa.
- SmartGolfa, a successful web and mobile platform that allows its users to play golf at their own pace, secure golf tees, and purchase packages.

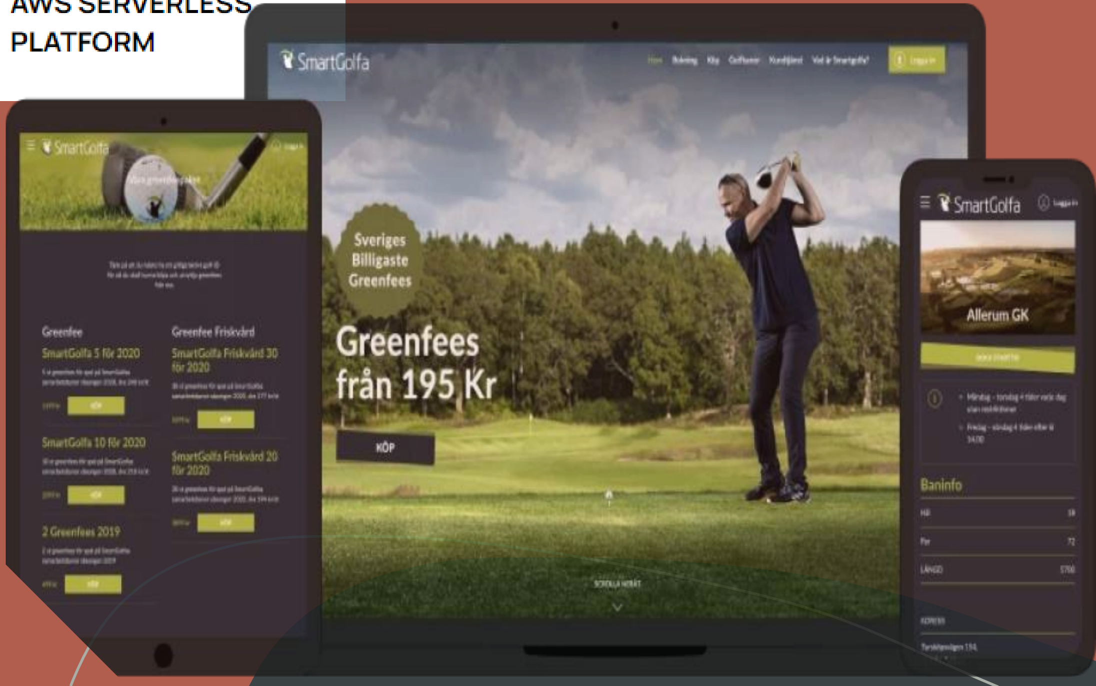
**50,000+**
ACTIVE USERS

**60+**
GOLF COURSES

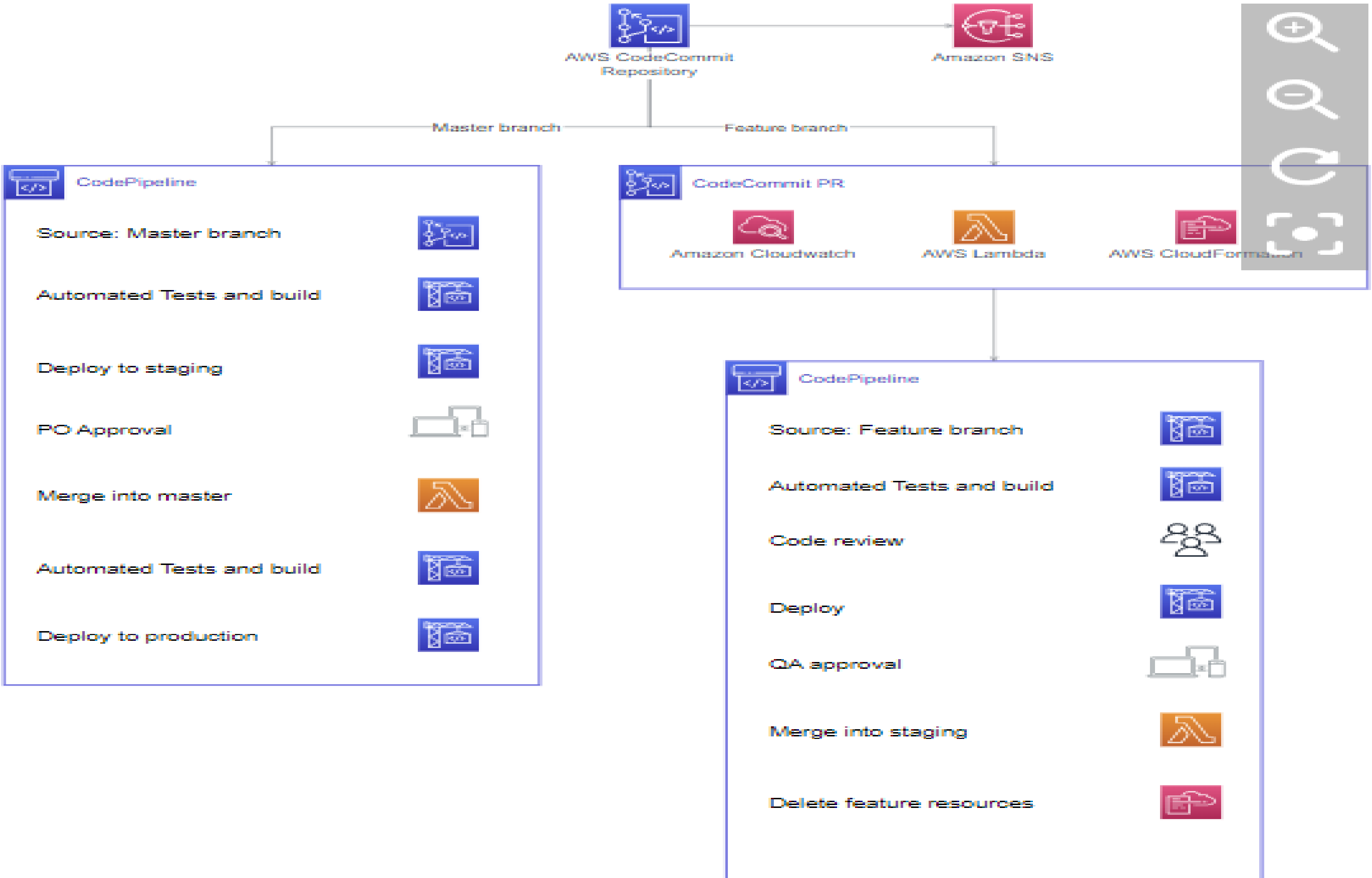
**100%**
AWS SERVERLESS PLATFORM

**AWS Services used as part of the solution**

- AWS Lambda
- Amazon Cognito
- AWS Step Functions
- AWS API Gateway
- AWS CodeCommit
- AWS CodePipeline
- AWS Pinpoint
- AWS DynamoDB & DAX



# Architecture diagram



# Lambda fundamentals

*to help customers become more productive we focus on:*

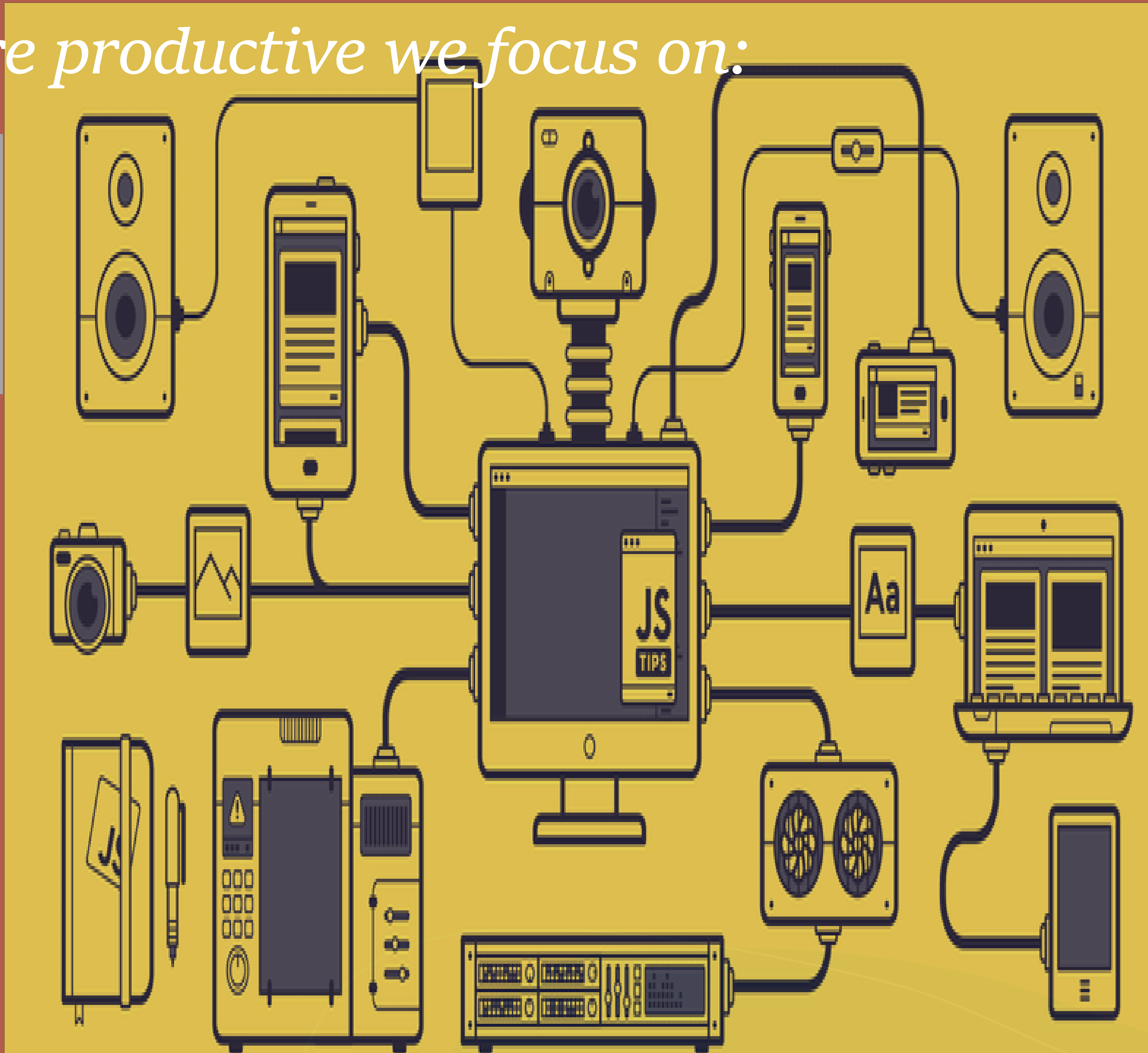
Performance

Security
Isolation

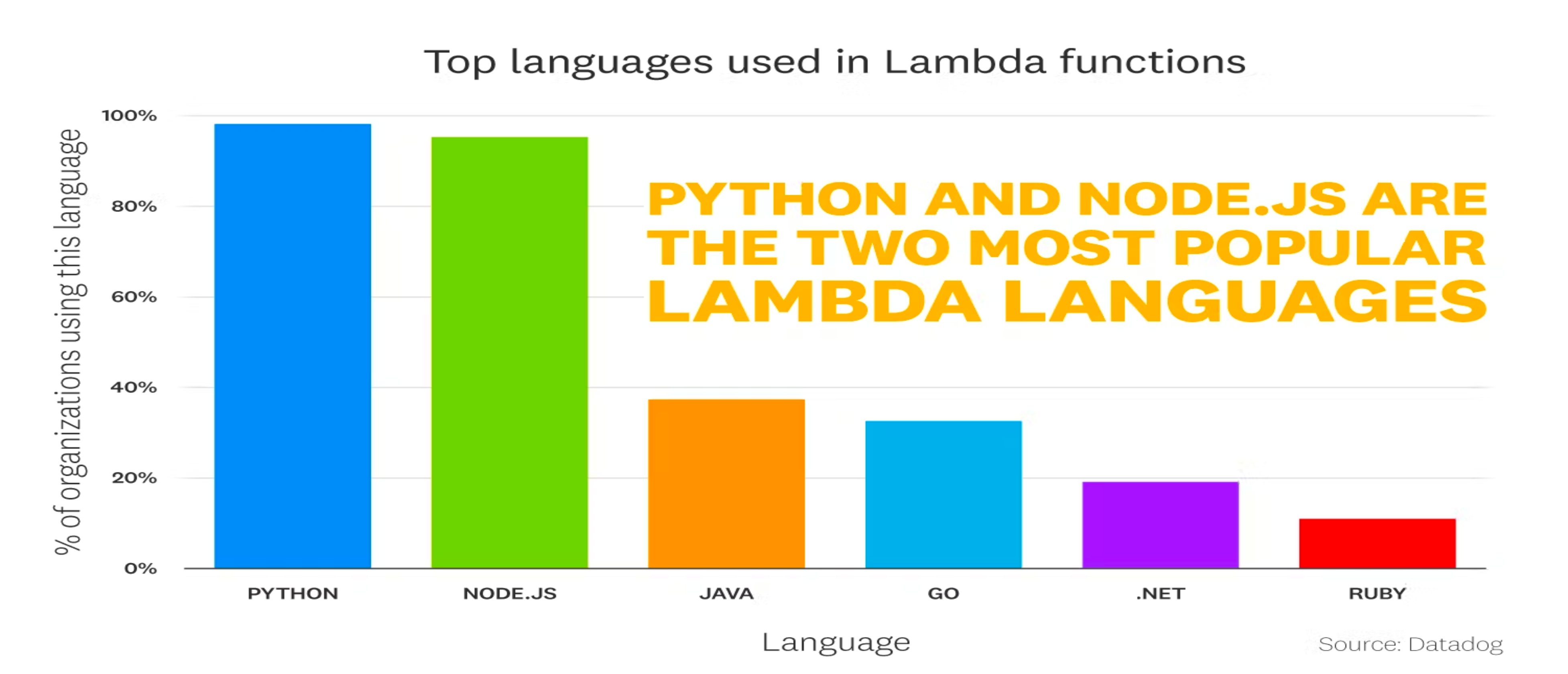
Availbility

Utilization

Scale



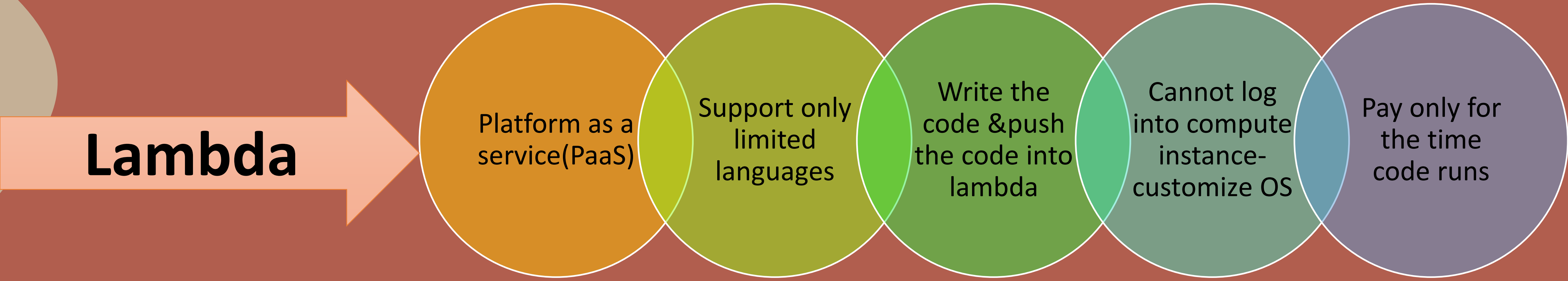
# Languages use in Lambda:



Amazon EC2

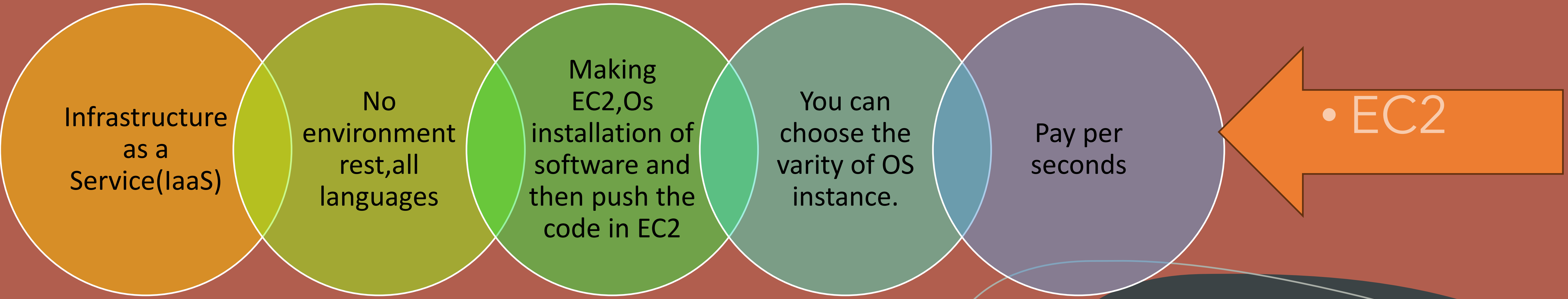
VS

AWS Lambda



## Lambda

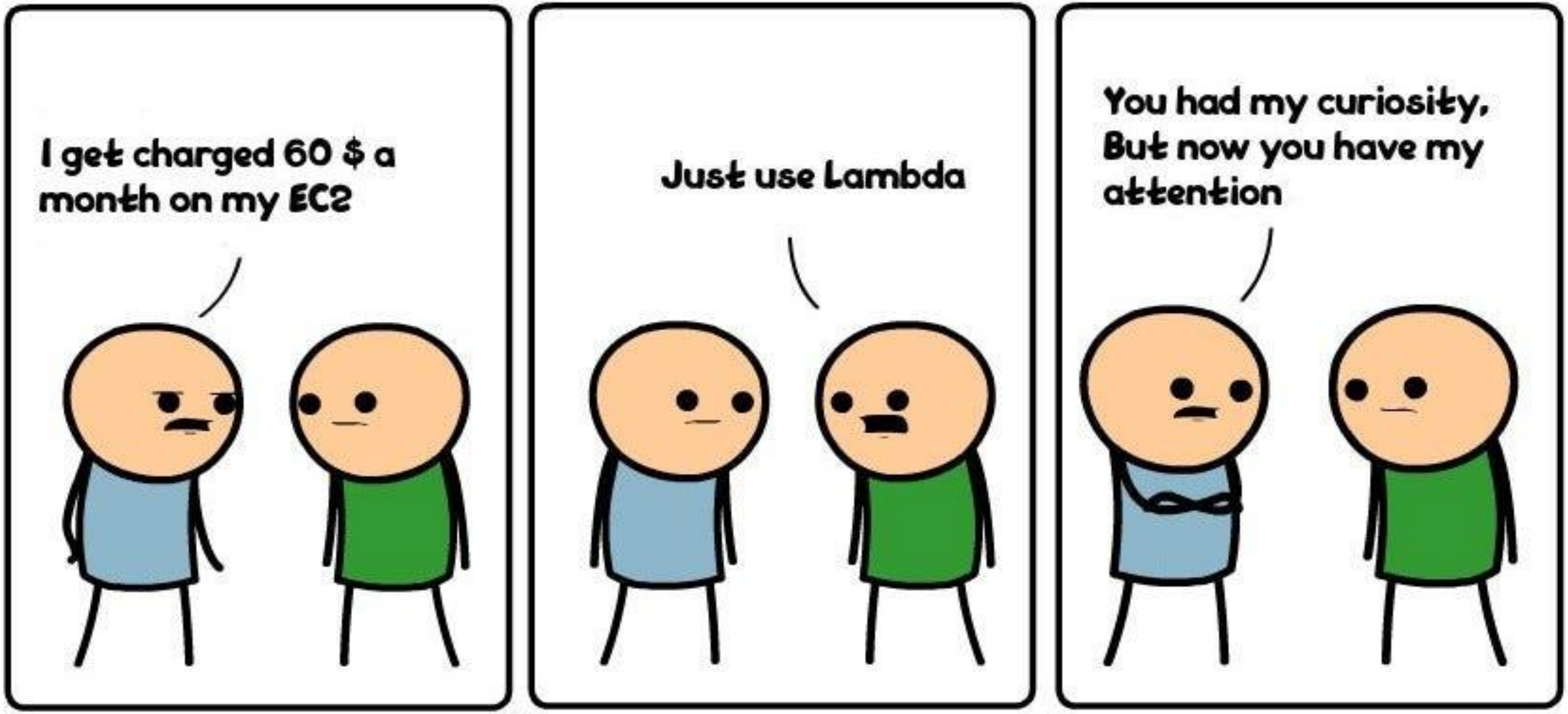
- Platform as a service(PaaS)
- Support only limited languages
- Write the code &push the code into lambda
- Cannot log into compute instance-customize OS
- Pay only for the time code runs

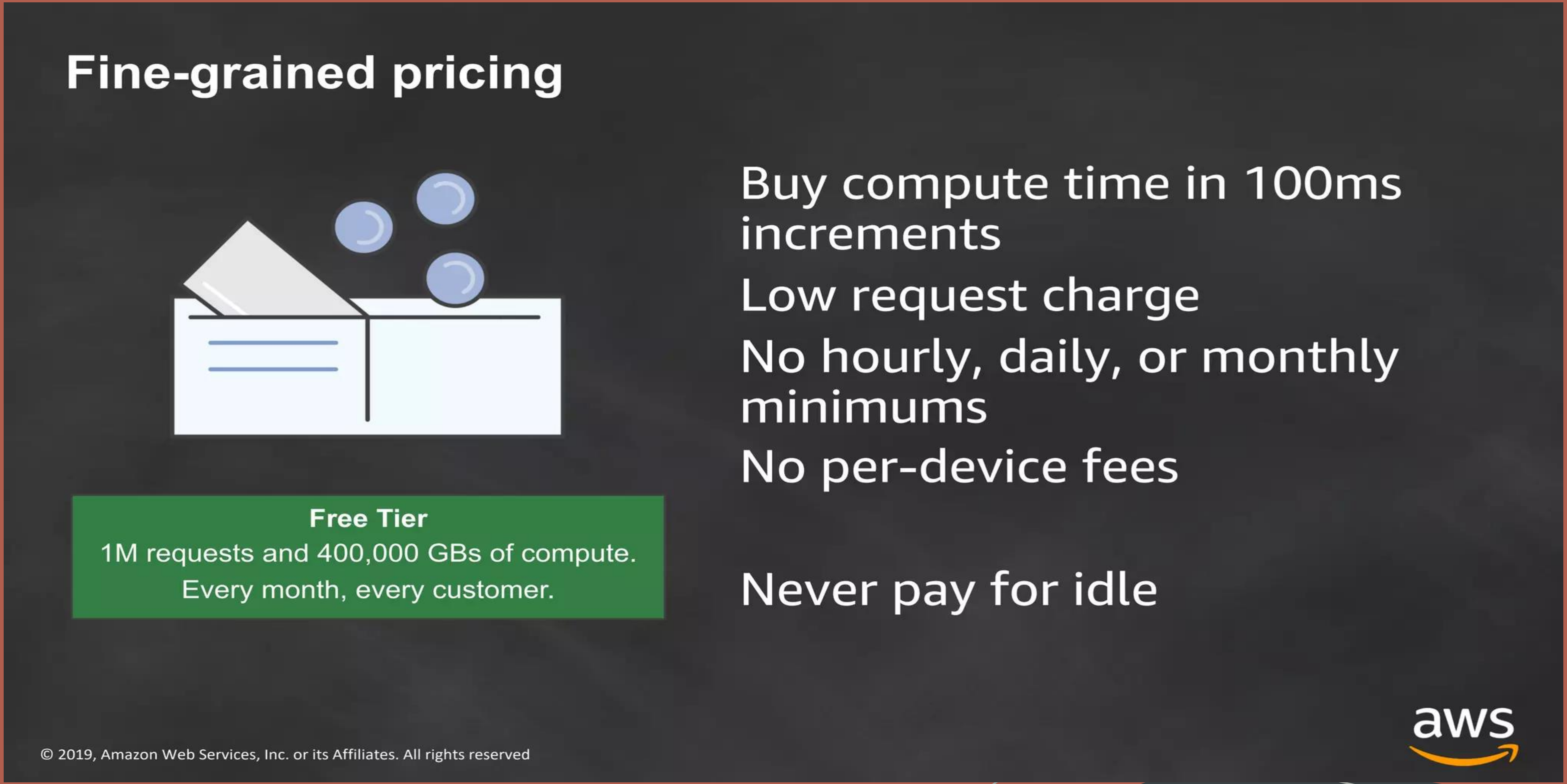


## EC2

- Infrastructure as a Service(IaaS)
- No environment rest,all languages
- Making EC2,Os installation of software and then push the code in EC2
- You can choose the varity of OS instance.
- Pay per seconds

# Fine-grained pricing in aws lambda


I get charged 60 $ a month on my EC2
Just use Lambda
You had my curiosity, But now you have my attention



# Fine-grained pricing

**Free Tier**
1M requests and 400,000 GBs of compute.
Every month, every customer.

Buy compute time in 100ms increments

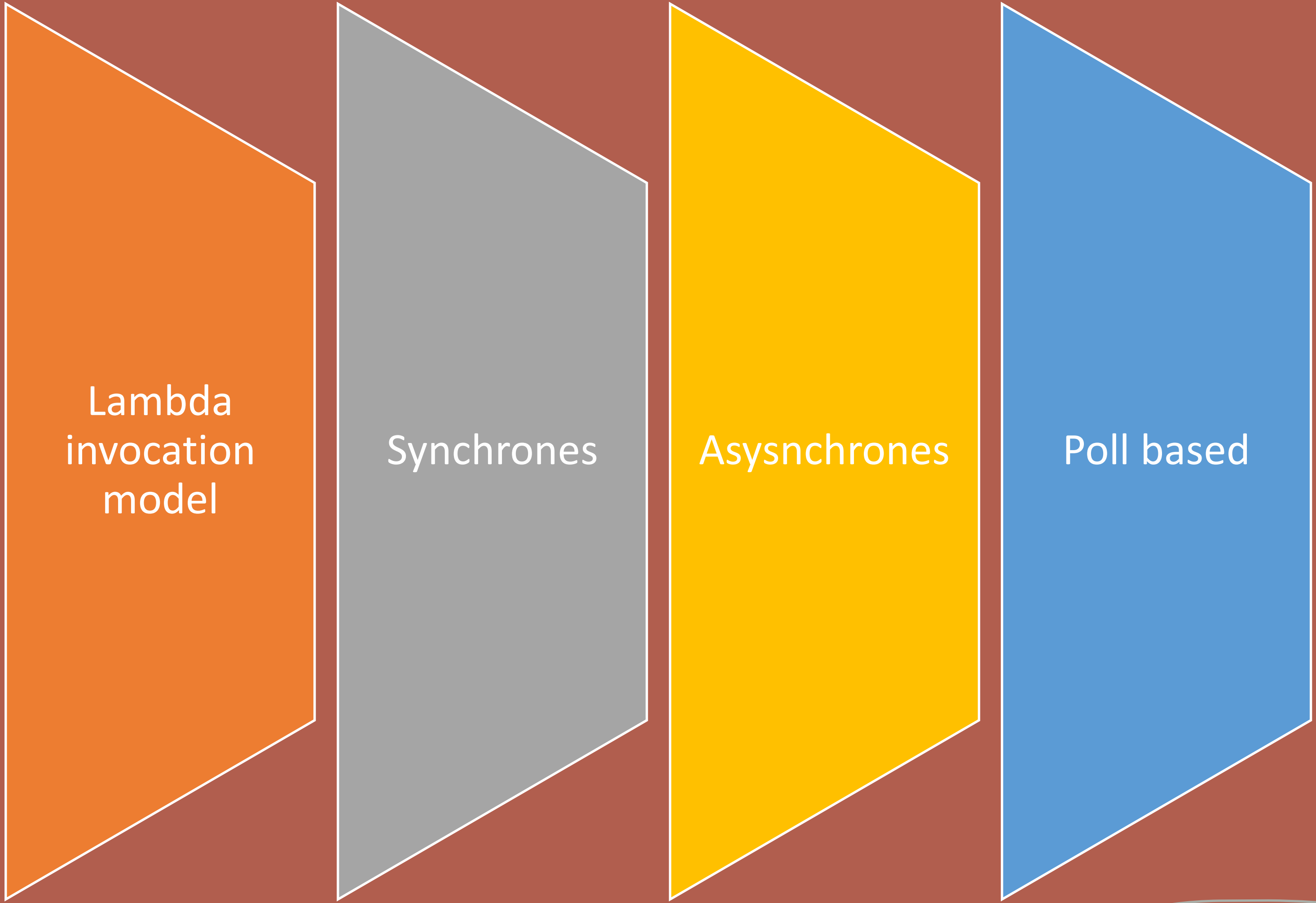
Low request charge

No hourly, daily, or monthly minimums

No per-device fees

Never pay for idle

# AWS Lambda invocation model:



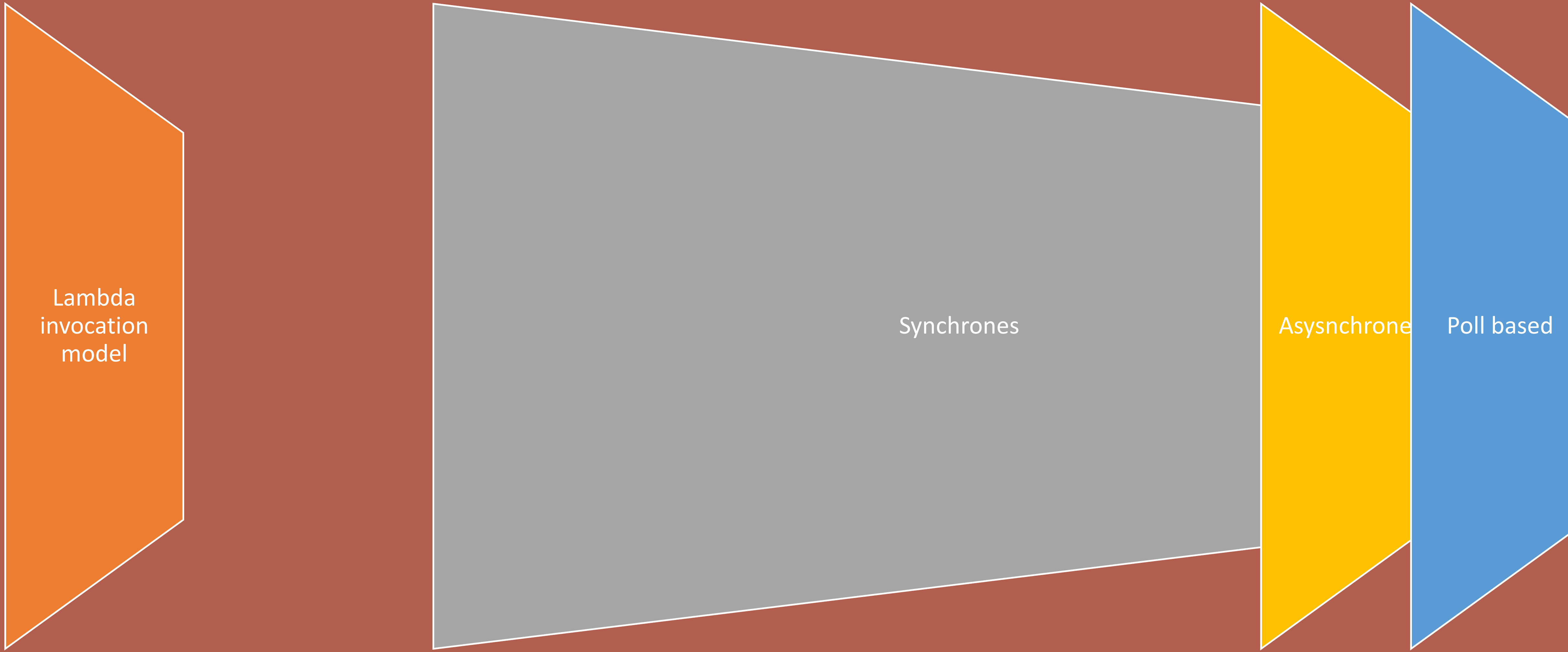
Lambda invocation model

Synchrones

Asysnchrones

Poll based

# AWS Lambda invocation model:



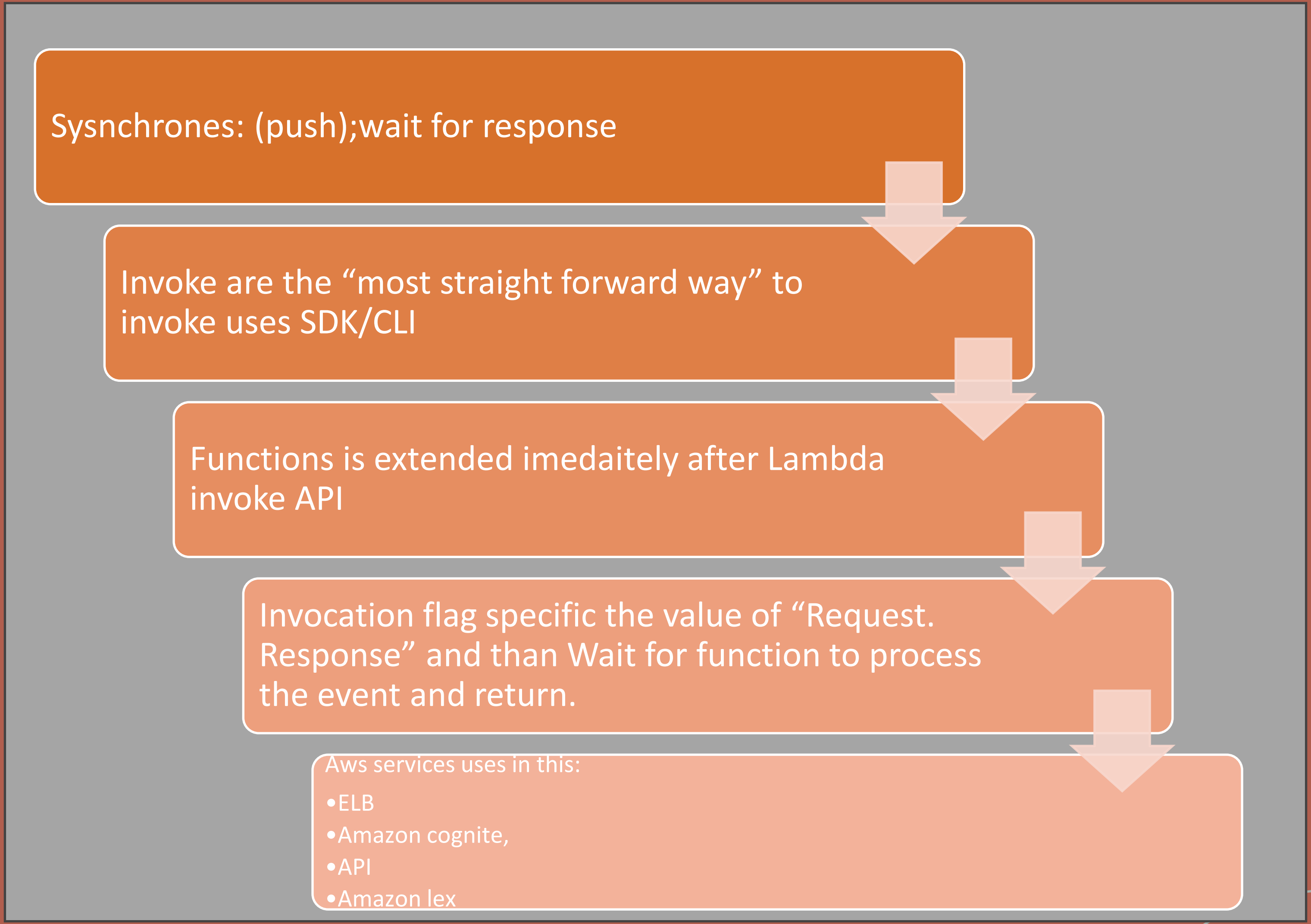
Lambda invocation model

Synchrones

Asysnchrone

Poll based

# AWS Lambda invocation model:



Sysnchrones: (push);wait for response

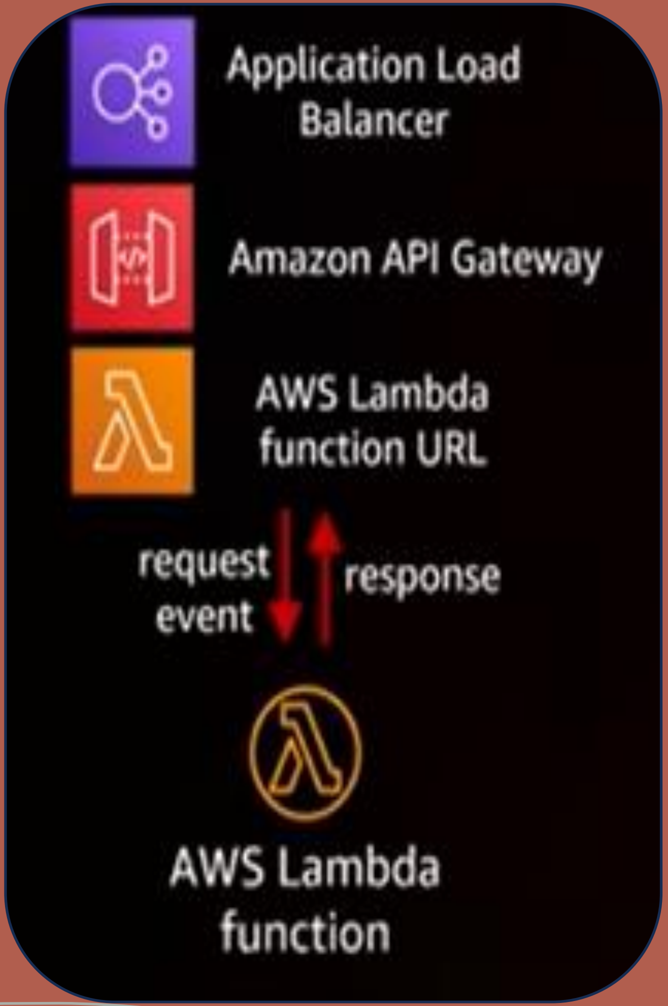
Invoke are the “most straight forward way” to invoke uses SDK/CLI

Functions is extended imedaitely after Lambda invoke API

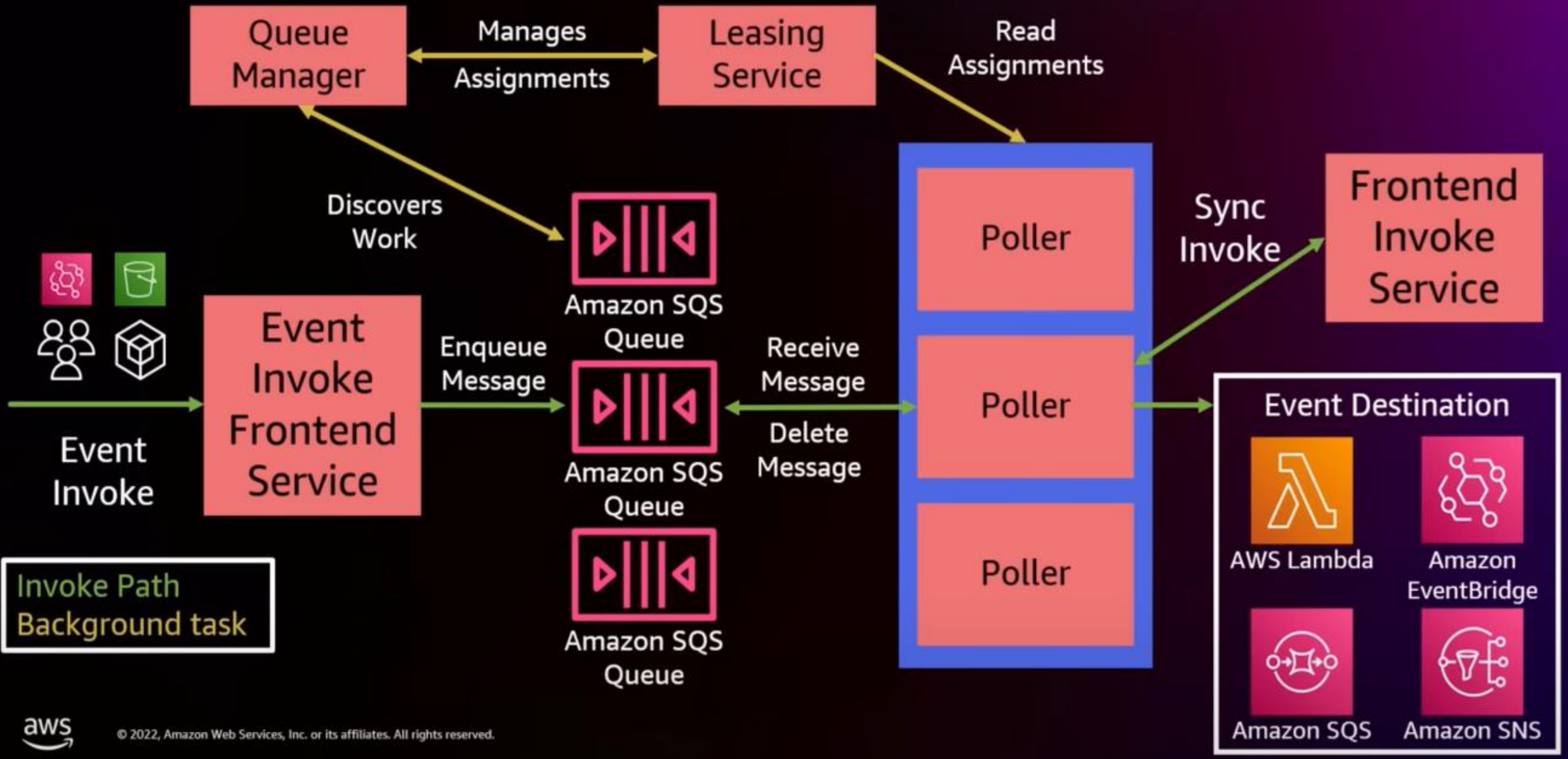
Invocation flag specific the value of “Request. Response” and than Wait for function to process the event and return.

Aws services uses in this:

- ELB
- Amazon cognite,
- API
- Amazon lex



# Lambda architecture: Async/event invoke



# AWS Lambda invocation model:

Asysnchrones: (Event); don’t wait for response

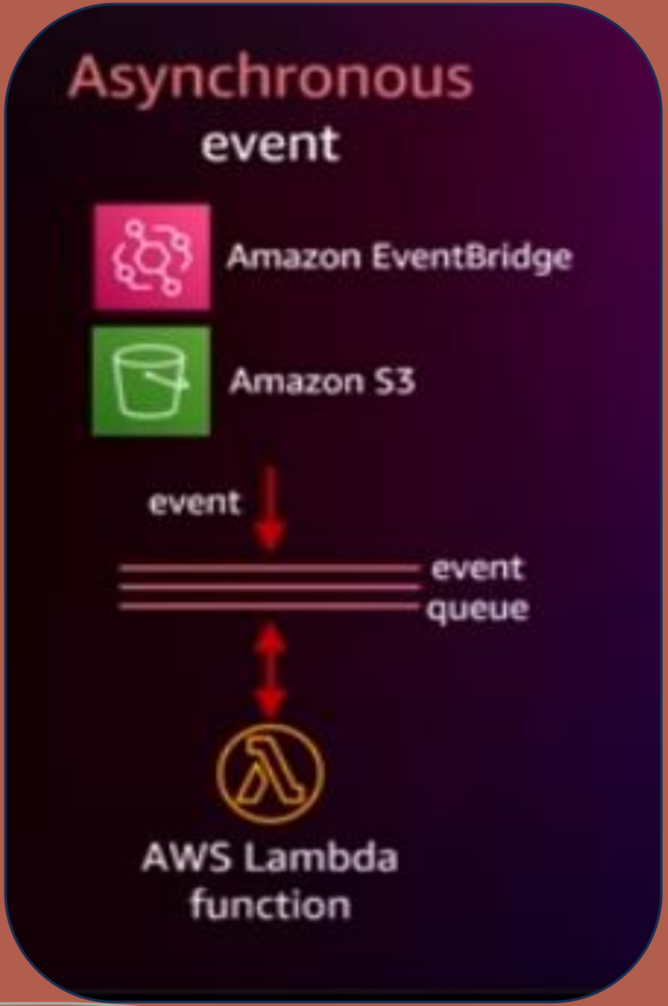
Lambda placed the “event” in a queue without response of success or failure.

Lambda queue the Event for processing & return a response imedateily.

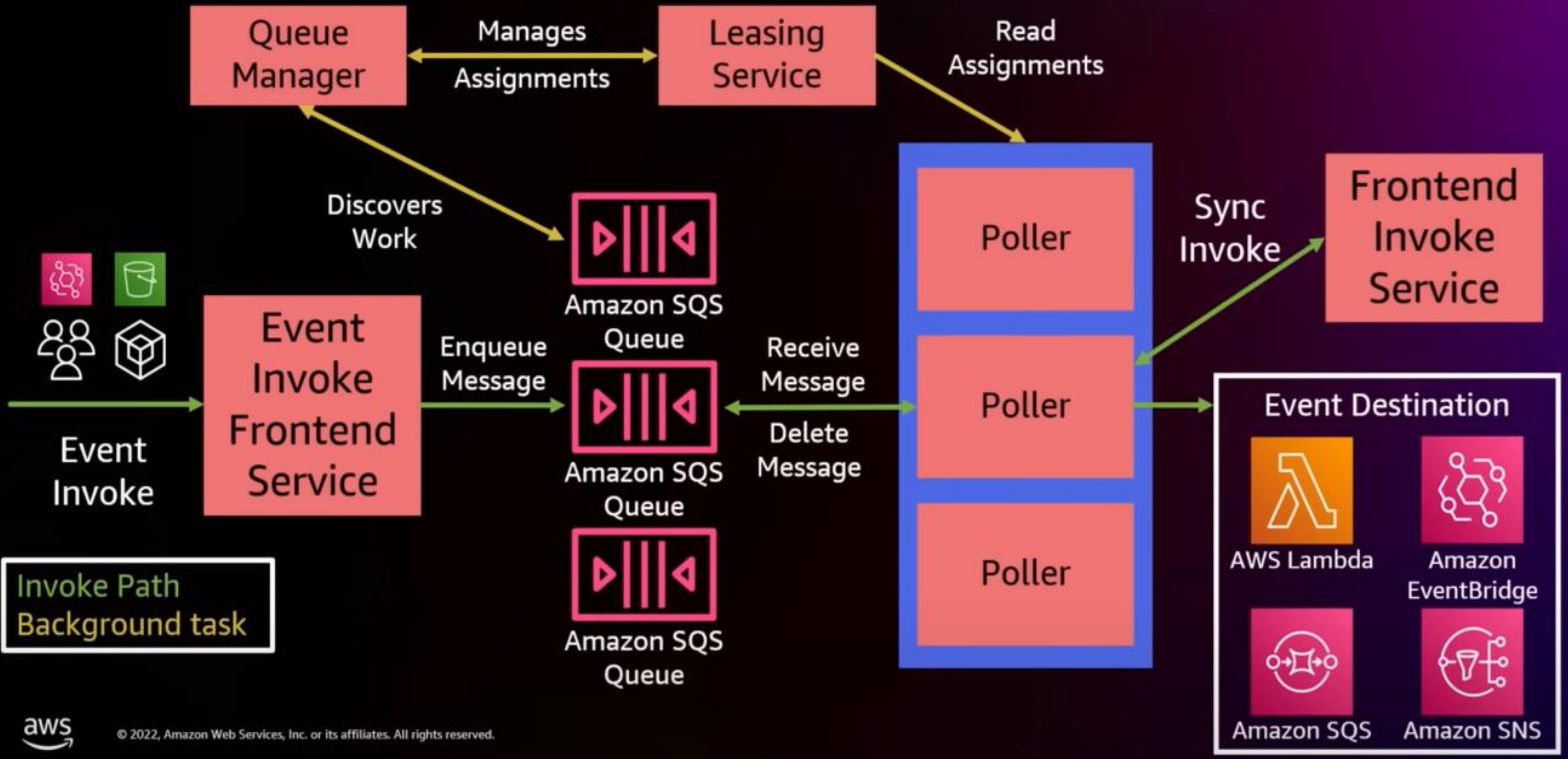
We can configure Lambda to send an invocation record to another server like SQS,SNS,Lambda and event build

Aws services uses in this:

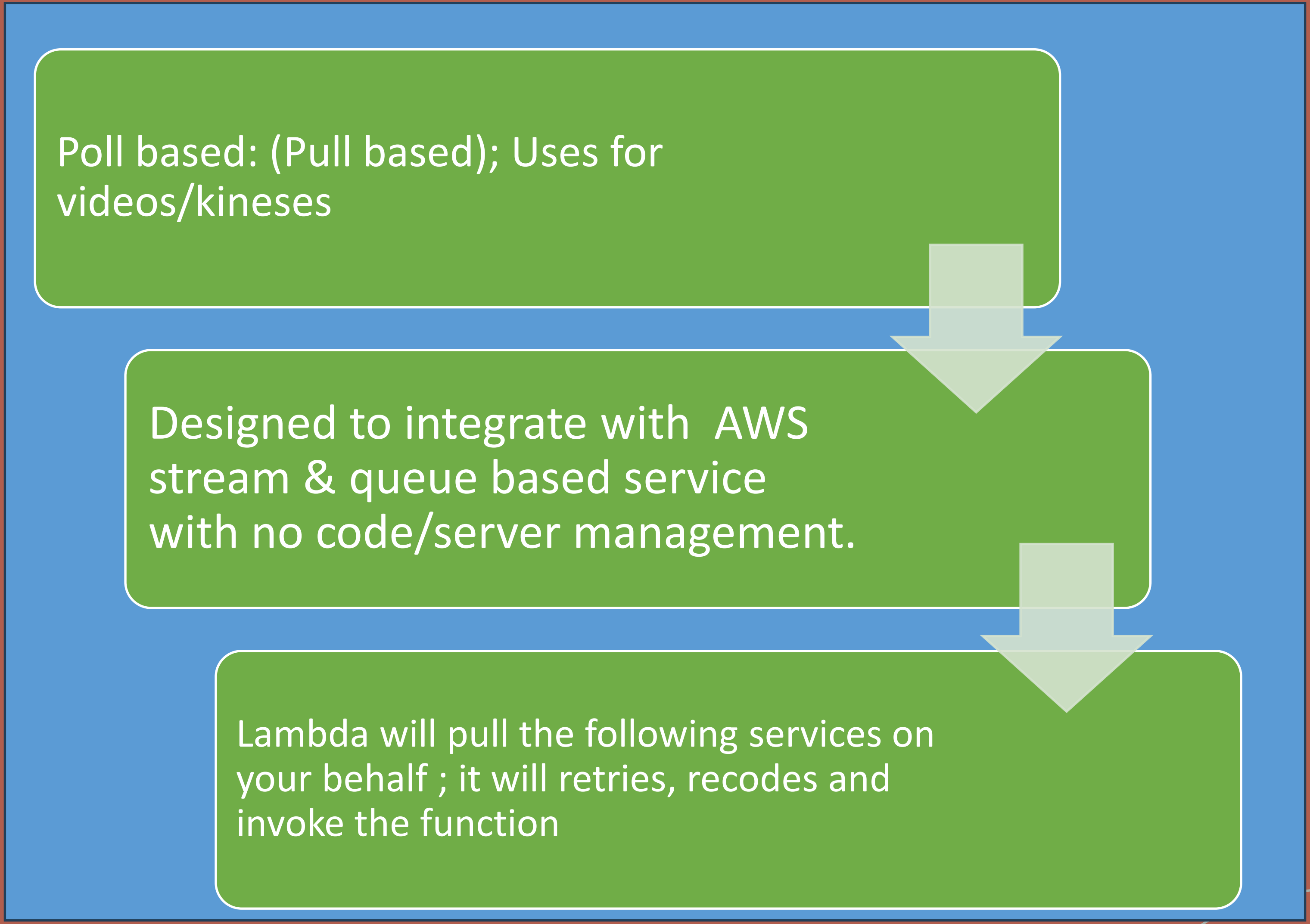
- S3 bucket
- Cloudwatch log
- Cloudformation
- Code commit



# Lambda architecture: Async/event invoke



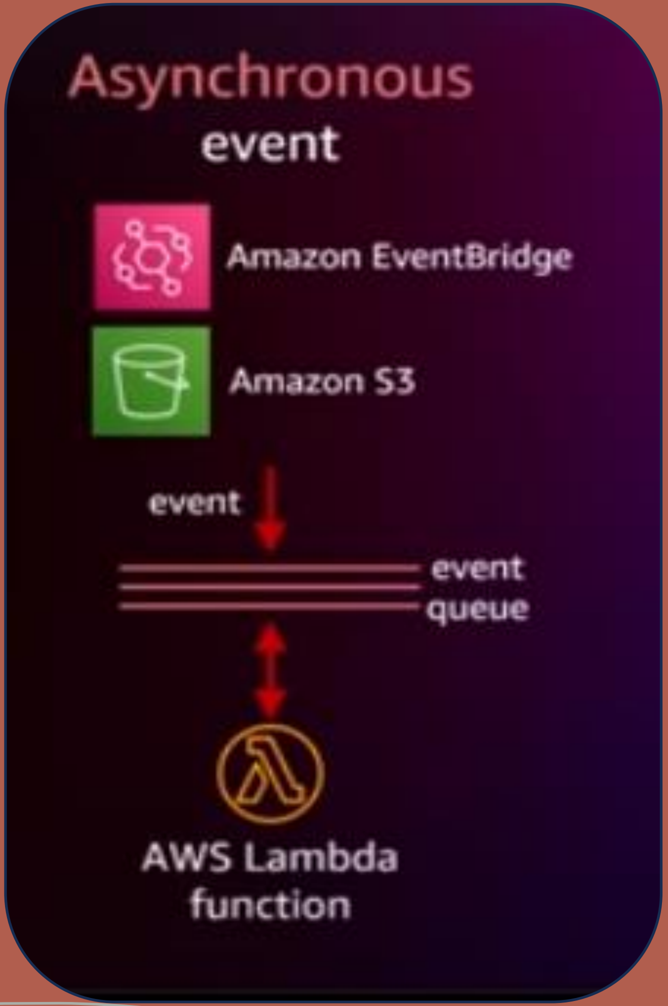
# AWS Lambda invocation model:



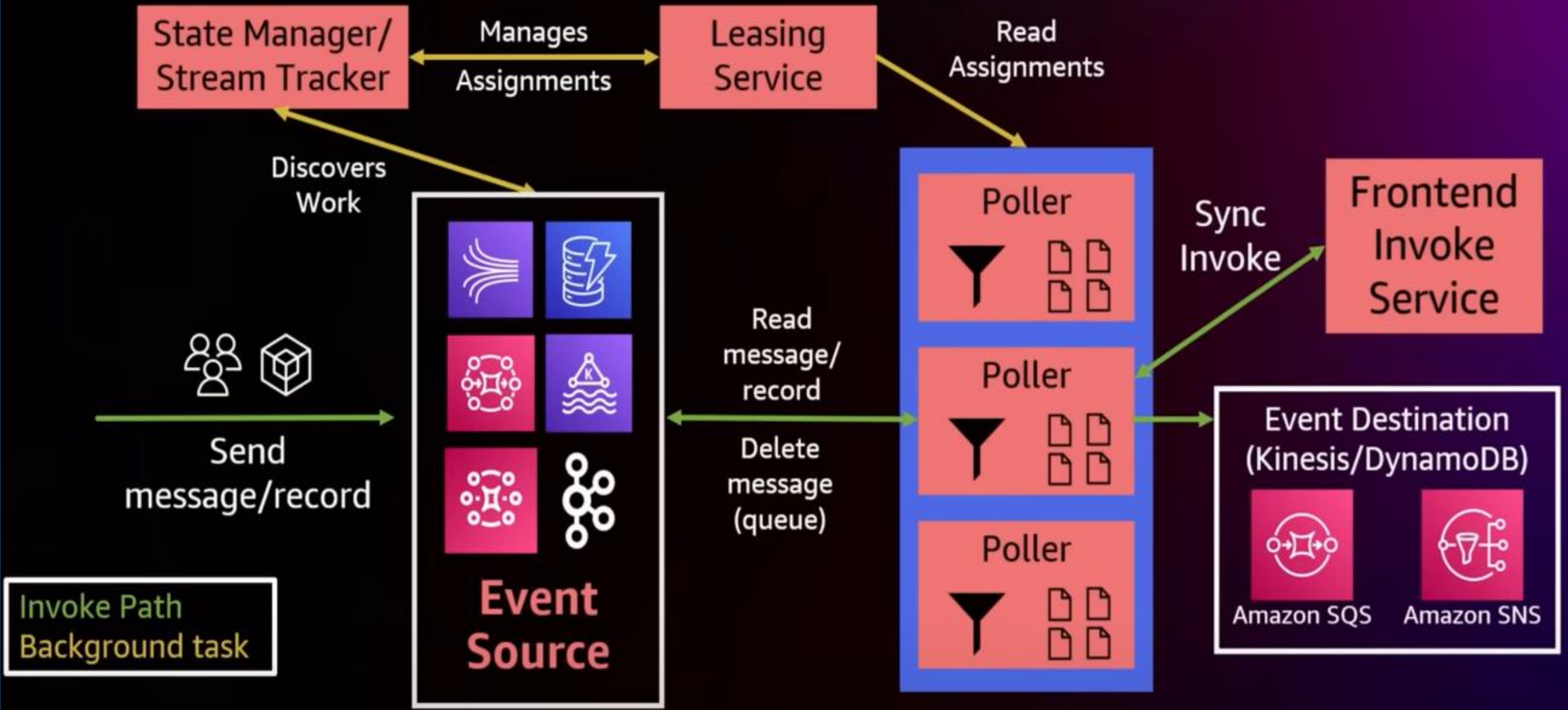
Poll based: (Pull based); Uses for videos/kineses

Designed to integrate with AWS stream & queue based service with no code/server management.

Lambda will pull the following services on your behalf ; it will retries, recodes and invoke the function



# Lambda architecture: Pollers



# Introducing: AWS Lambda runtime API and layers

NEW!

Features that allow developers to share, discover, and deploy both libraries and languages as part of their serverless applications

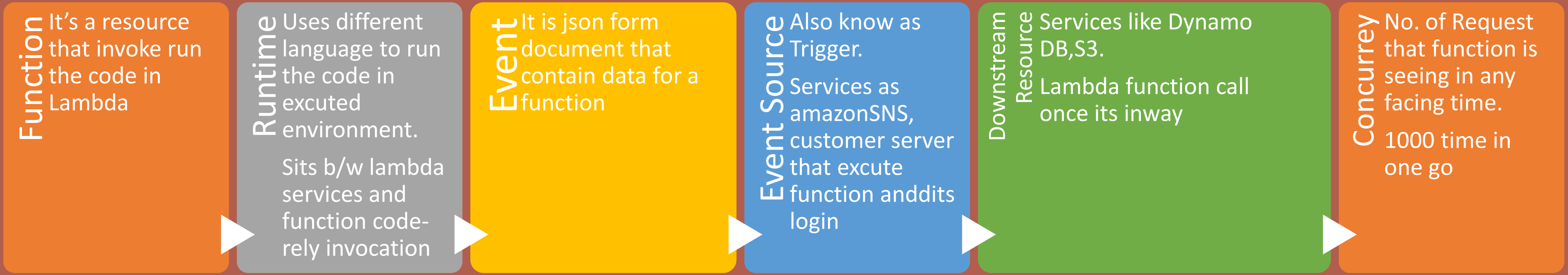
Runtime API enables developers to use Lambda with any programming language.

Layers let functions easily share code. Upload layer once, reference within any function.

Layers promote separation of responsibilities, lets developers focus on writing business logic.

Combined, runtime API and layers allow developers to share any programming language or language version with others

# Important terms used in Lambda:



**Function**

It's a resource that invoke run the code in Lambda

**Runtime**

Uses different language to run the code in excuted environment.

Sits b/w lambda services and function code-rely invocation

**Event**

It is json form document that contain data for a function

**Event Source**

Also know as Trigger.

Services as amazonSNS, customer server that excute function anddits login

**Downstream Resource**

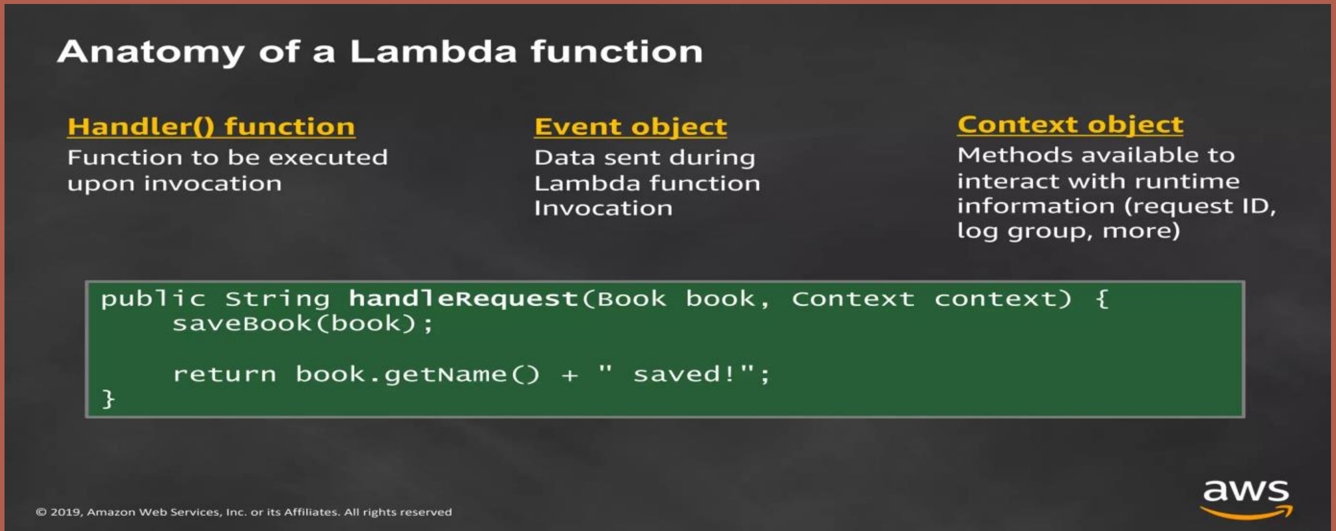
Services like Dynamo DB,S3.

Lambda function call once its inway

**Concurrey**

No. of Request that function is seeing in any facing time.

1000 time in one go



## Anatomy of a Lambda function

**Handler() function**
Function to be executed upon invocation

**Event object**
Data sent during Lambda function Invocation

**Context object**
Methods available to interact with runtime information (request ID, log group, more)

```
public String handleRequest(Book book, Context context) {
    saveBook(book);

    return book.getName() + " saved!";
}
```

# Using AWS Lambda

## Bring your own code

- Node.js, Java, Python, C#, Go
- Bring your own libraries (even native ones)

## Simple resource model

- Select power rating from 128 MB to 3 GB
- CPU and network allocated proportionately

## Flexible use

- Synchronous or asynchronous
- Integrated with other AWS services

## Flexible authorization

- Securely grant access to resources and VPCs
- Fine-grained control for invoking your functions

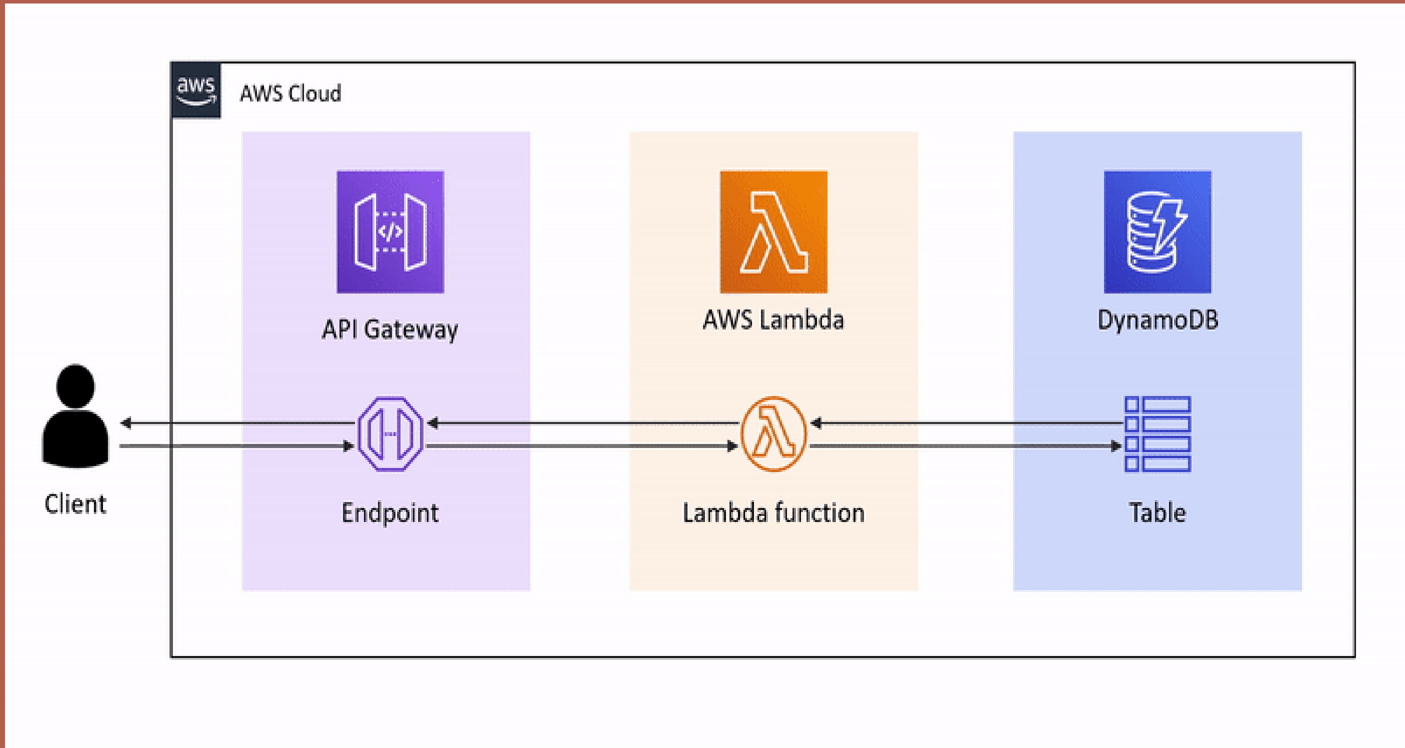
# When Lambda trigger

Code run in Repoistory

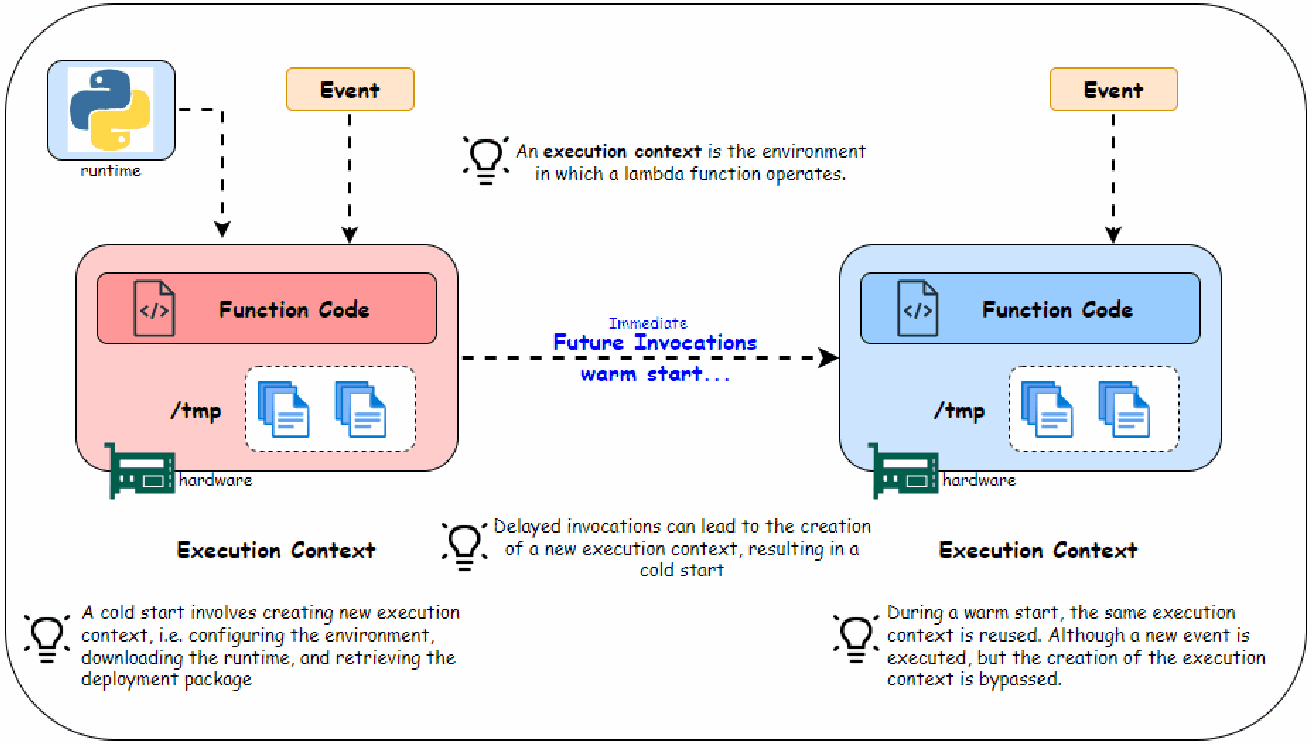
Event such change as data in S3 bucket

Run code in response to HTTP request: API gateway

Easily build data processing trigger for AWS swerver like S3 bucket,dynamo DB



# How Lambda Functions are Executed:


runtime
Event
Event
An execution context is the environment in which a lambda function operates.
Function Code
/tmp
hardware
Immediate
Future Invocations
warm start...
Function Code
/tmp
hardware
Execution Context
Delayed invocations can lead to the creation of a new execution context, resulting in a cold start
Execution Context
A cold start involves creating new execution context, i.e. configuring the environment, downloading the runtime, and retrieving the deployment package
During a warm start, the same execution context is reused. Although a new event is executed, but the creation of the execution context is bypassed.

**Execution Context:** Lambda code operates within a runtime environment known as the ***Execution Context***.

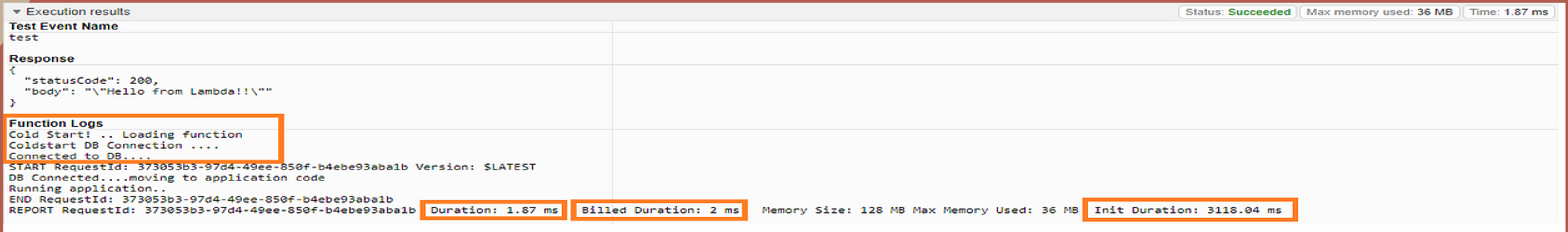
**Cold Start:** When you invoke a Lambda function, it's like creating a fresh *execution context* from scratch. This includes setting up the environment, downloading the runtime *(e.g., Python)*, and fetching the deployment package *(your code)*.

This process takes about 100's of milliseconds and is what we call a ***Cold Start***.

**Warm Start:** If you invoke the same Lambda function within a few seconds after the first invocation, there's a chance it will reuse the existing execution context. This is known as a *Warm Start.*

***During a Warm Start, there's no need to set up the execution context from scratch or download the deployment package***.

# Cold start



Execution results

Status: Succeeded | Max memory used: 36 MB | Time: 1.87 ms

Test Event Name

```
test
```

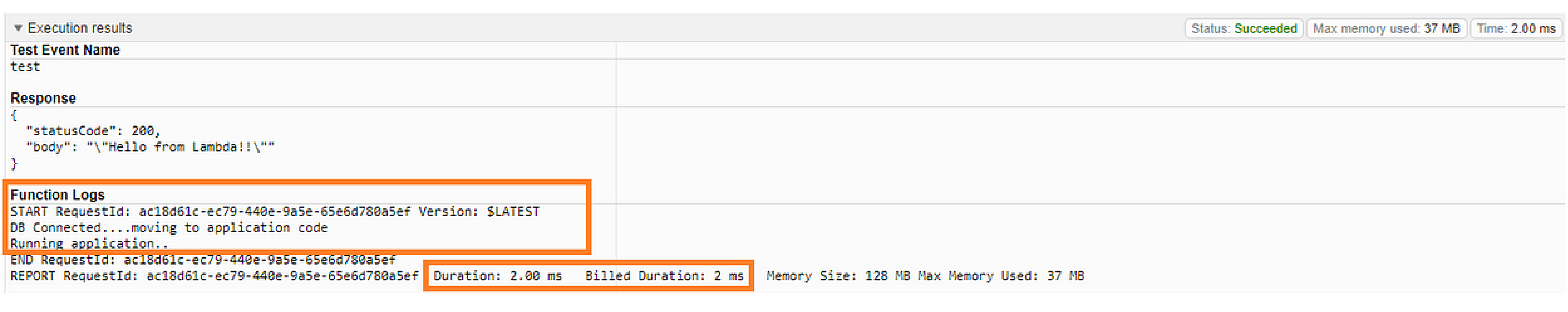
Response

```
{
  "statusCode": 200,
  "body": "\"Hello from Lambda!!\""
}
```

Function Logs

```
Cold Start! .. Loading function
Coldstart DB Connection ....
Connected to DB....
START RequestId: 373053b3-97d4-49ee-850f-b4ebe93aba1b Version: $LATEST
DB Connected....moving to application code
Running application..
END RequestId: 373053b3-97d4-49ee-850f-b4ebe93aba1b
REPORT RequestId: 373053b3-97d4-49ee-850f-b4ebe93aba1b  Duration: 1.87 ms  Billed Duration: 2 ms  Memory Size: 128 MB Max Memory Used: 36 MB  Init Duration: 3118.04 ms
```

# Warm start



Execution results

Status: Succeeded | Max memory used: 37 MB | Time: 2.00 ms

Test Event Name

```
test
```

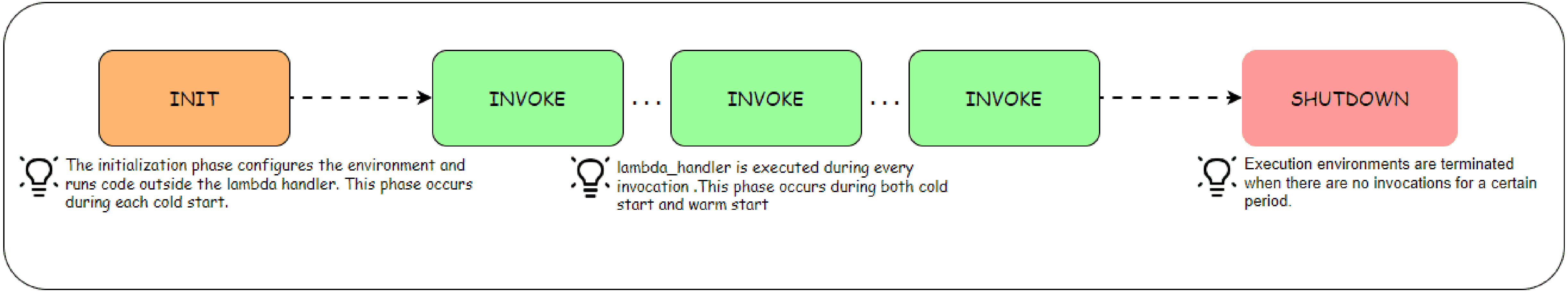
Response

```
{
  "statusCode": 200,
  "body": "\"Hello from Lambda!!\""
}
```

Function Logs

```
START RequestId: ac18d61c-ec79-440e-9a5e-65e6d780a5ef Version: $LATEST
DB Connected....moving to application code
Running application..
END RequestId: ac18d61c-ec79-440e-9a5e-65e6d780a5ef
REPORT RequestId: ac18d61c-ec79-440e-9a5e-65e6d780a5ef  Duration: 2.00 ms  Billed Duration: 2 ms  Memory Size: 128 MB Max Memory Used: 37 MB
```

# Lambda Handler Architecture:

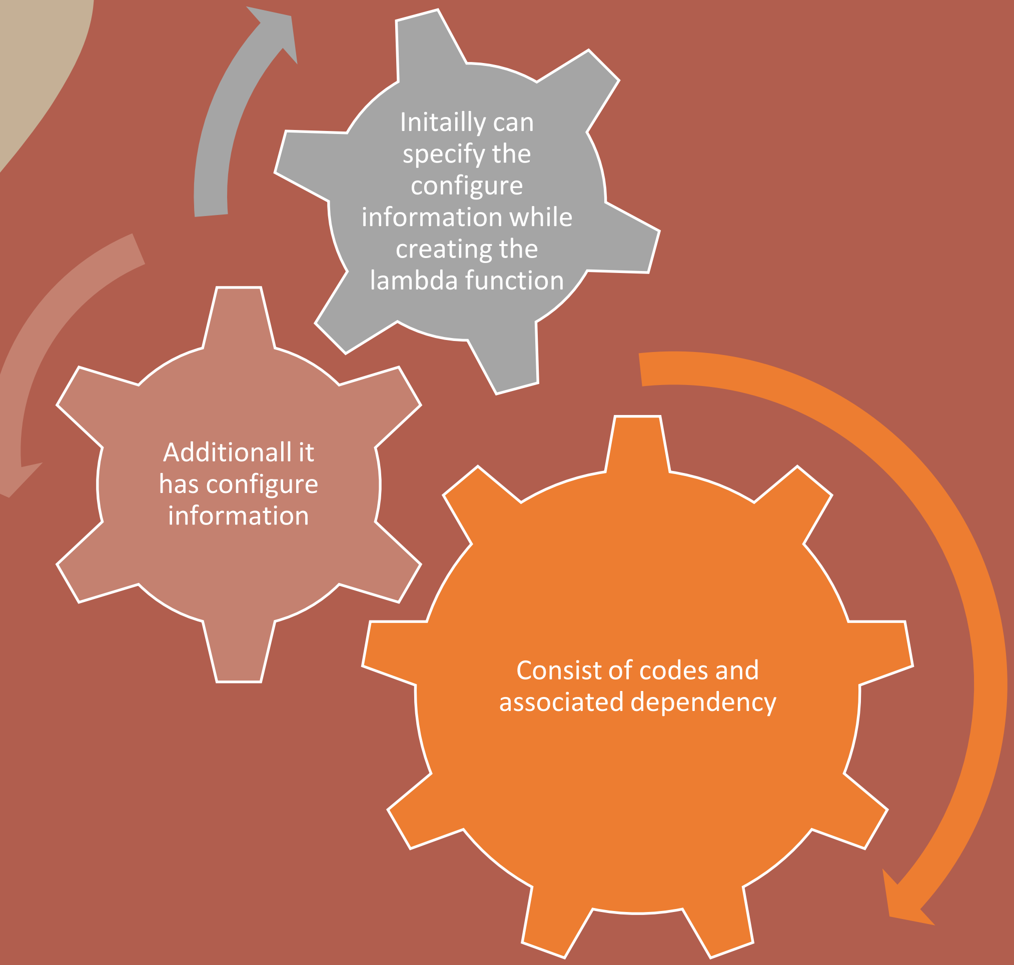


The initialization phase configures the environment and runs code outside the lambda handler. This phase occurs during each cold start.

lambda_handler is executed during every invocation .This phase occurs during both cold start and warm start

Execution environments are terminated when there are no invocations for a certain period.

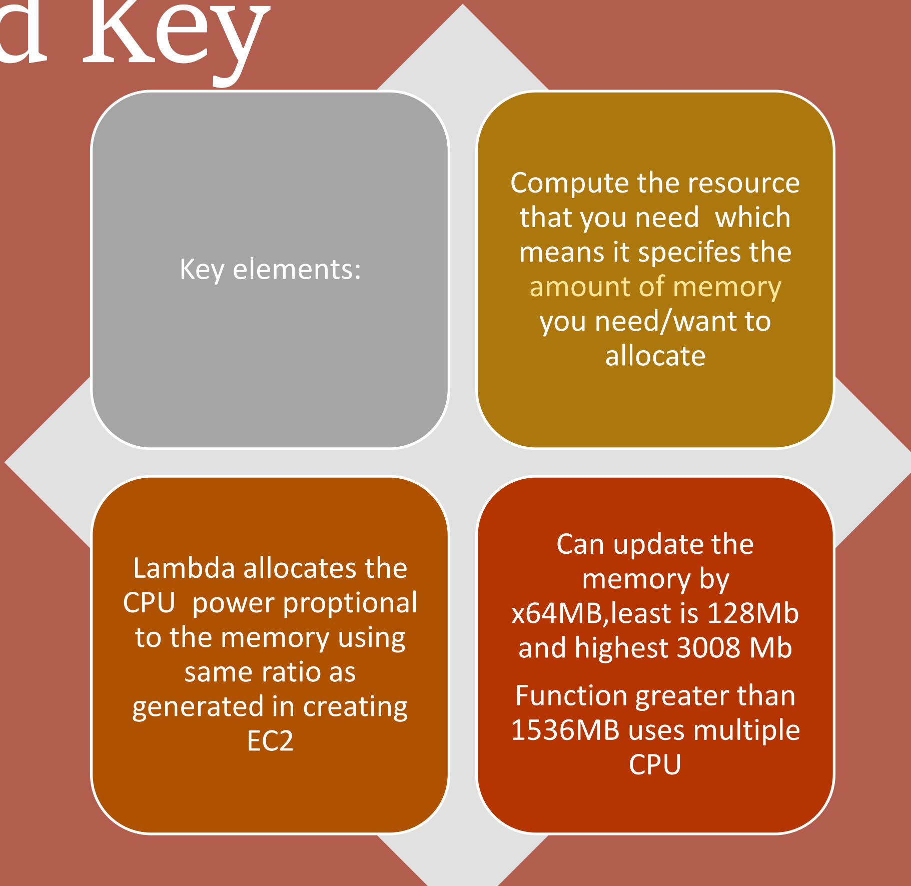
# AWS Lambda configuration and Key elements



Initailly can specify the configure information while creating the lambda function

Additionall it has configure information

Consist of codes and associated dependency



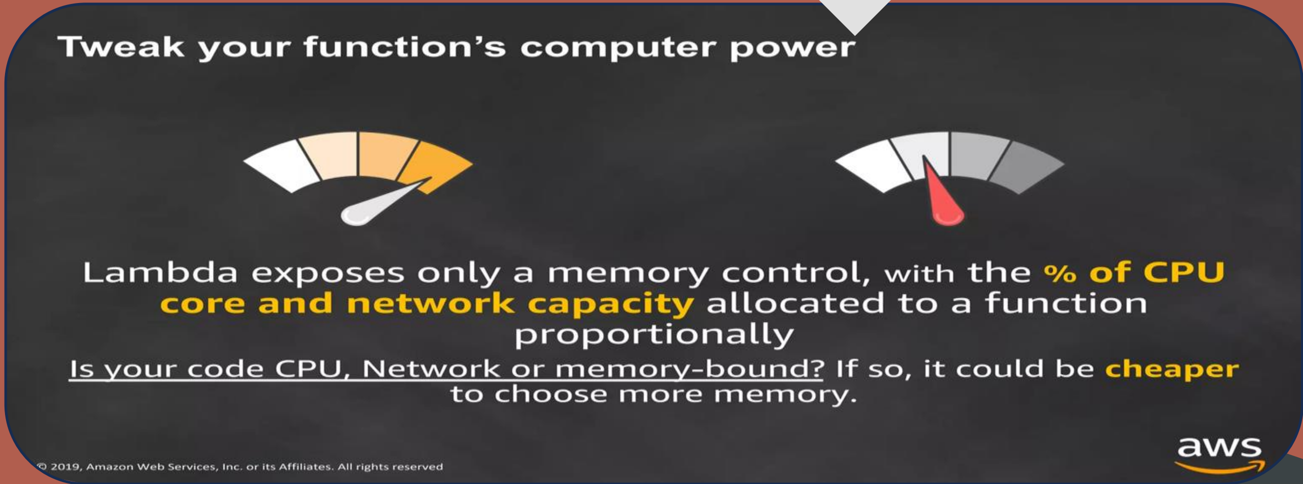
Key elements:

Compute the resource that you need which means it specifes the amount of memory you need/want to allocate

Lambda allocates the CPU power proptional to the memory using same ratio as generated in creating EC2

Can update the memory by x64MB,least is 128Mb and highest 3008 Mb

Function greater than 1536MB uses multiple CPU

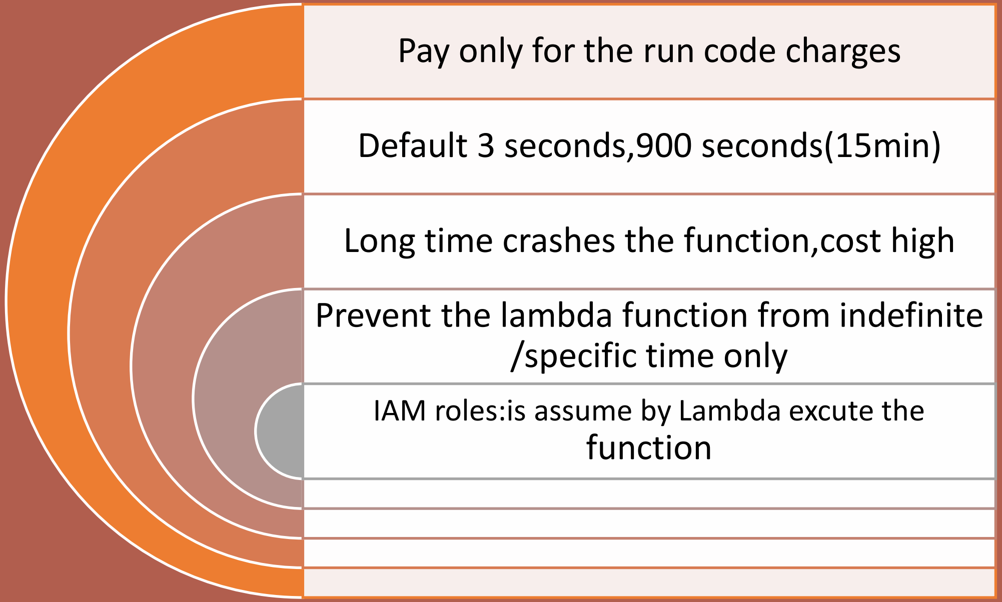


**Tweak your function's computer power**

Lambda exposes only a memory control, with the **% of CPU core and network capacity** allocated to a function proportionally

Is your code CPU, Network or memory-bound? If so, it could be **cheaper** to choose more memory.

# Maximum execution timeout and Lambda function-Services can access



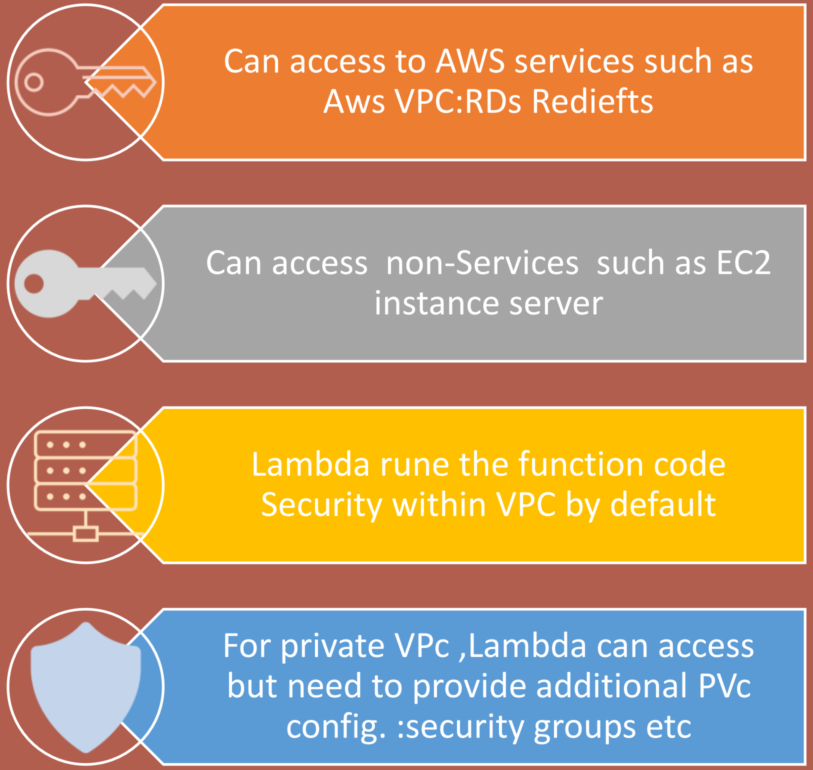
Pay only for the run code charges

Default 3 seconds,900 seconds(15min)

Long time crashes the function,cost high

Prevent the lambda function from indefinite /specific time only

IAM roles:is assume by Lambda excute the function



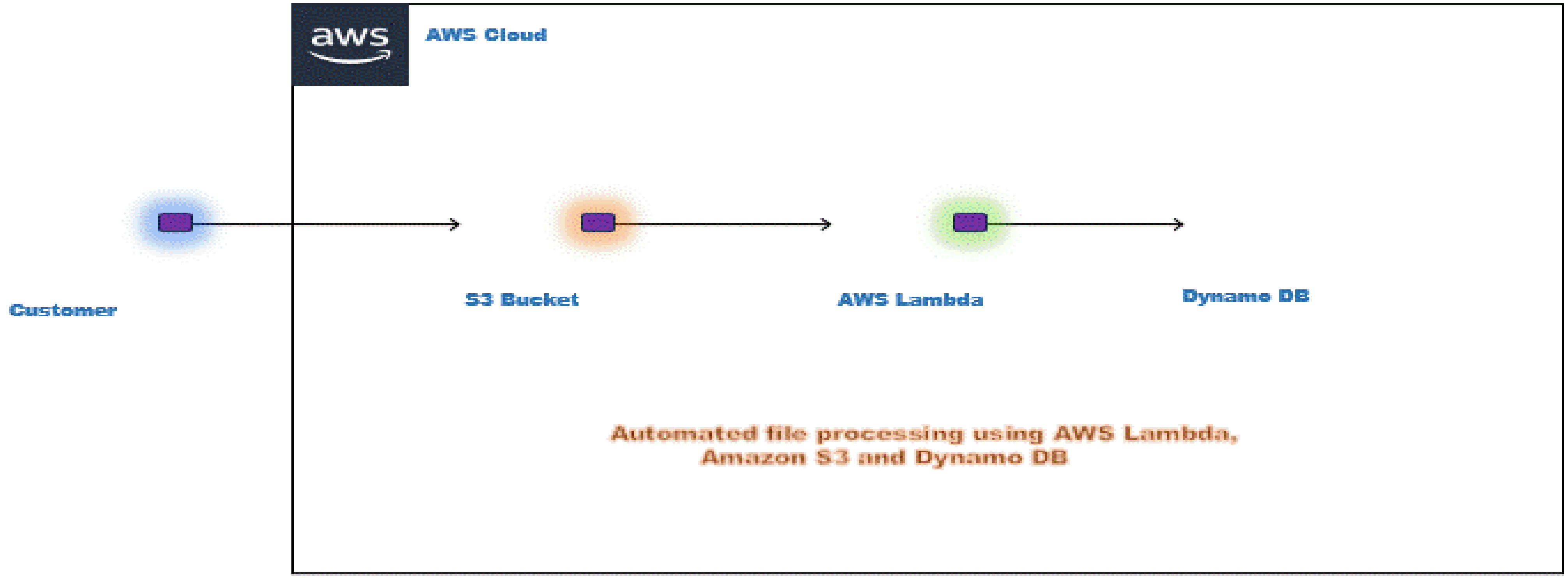
Can access to AWS services such as Aws VPC:RDs Rediefts

Can access non-Services such as EC2 instance server

Lambda rune the function code Security within VPC by default

For private VPc ,Lambda can access but need to provide additional PVc config. :security groups etc

# How the project looks like!



# Demo time

S3 bucket:
https://s3.console.aws.amazon.com/s3/buckets/contactlamb?region=eu-west-1&tab=objects

SES:
https://eu-west-1.console.aws.amazon.com/ses/home?region=eu-west-1#/verified-identities

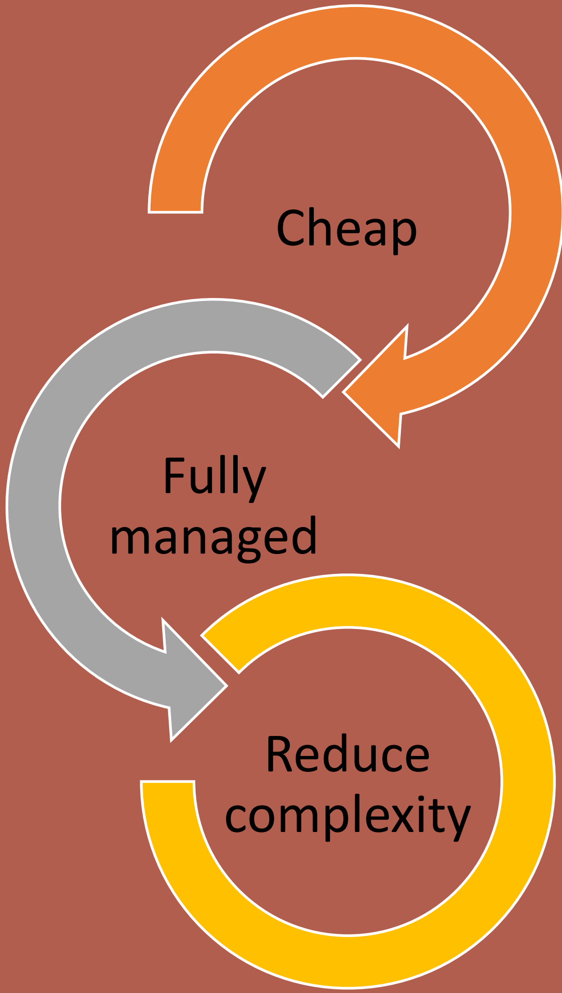
Emails I receive:
https://mail.yahoo.com/d/folders/1

Lambda function:
https://eu-west-1.console.aws.amazon.com/lambda/home?region=eu-west-1#/functions

https://eu-west-1.console.aws.amazon.com/lambda/home?region=eu-west-1#/functions/contactfunction?tab=code

Dynamdb:
https://eu-west-1.console.aws.amazon.com/dynamodbv2/home?region=eu-west-1#item-explorer?maximize=true&operation=SCAN&table=contactinfo

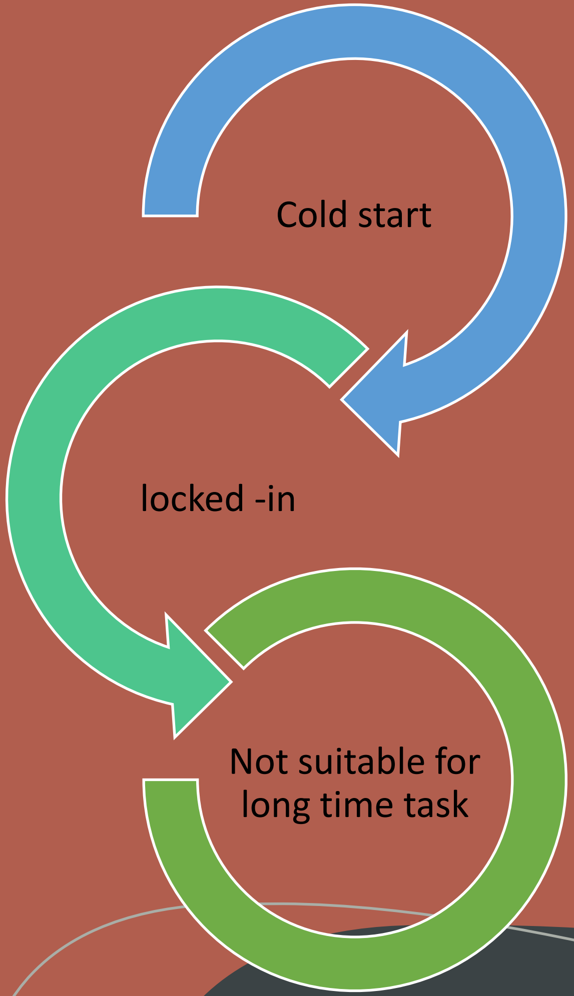
# Pros and cons



Cheap

Fully managed

Reduce complexity



Cold start

locked -in

Not suitable for long time task

*slaps logo of AWS Lambda*
THIS BAD BOY CAN DO ALL THAT FOR YOU
imgflip.com