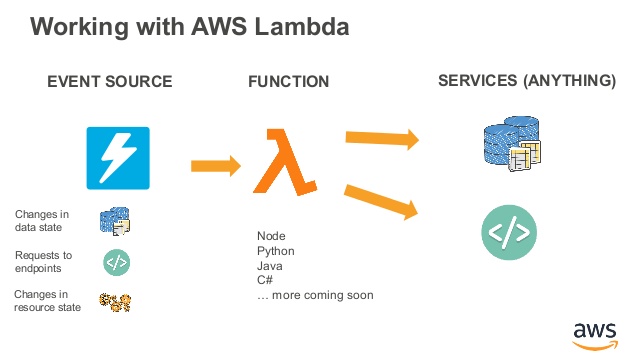
* **Lambda Function**
  + Là compute service để tính toán để run code mà không cần quản lý hệ điều hành, ... (giống docker)
  + Đặc điểm
    - Lambda sẽ quản lý tất cả các resource để run code

🡪 Không cần config HA, automatic scaling, monitoring, …, quản lý CPU, Networking, …

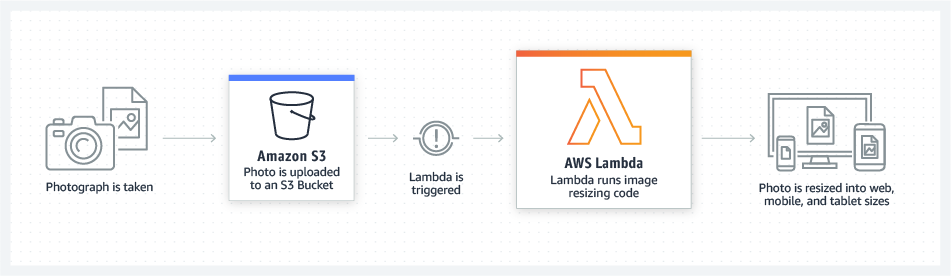
* + - Có thể connect tới các AWS service khác (S3, Database, SQS, …)
    - Mỗi 1 lần lambda sẽ chạy trên 1 thread riêng hoàn toàn và concurrency hàng nghìn lambda function cùng lúc



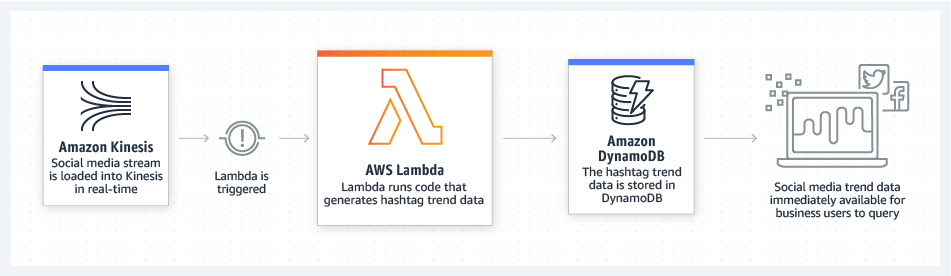
* + Usecase
    - Build data-processing triggers cho AWS services (S3, DynamoDB)

Phải trigger vào thì lambda mới chạy

VD: ngoài giờ làm việc thì sẽ stop hết instance thì sẽ tạo lambda

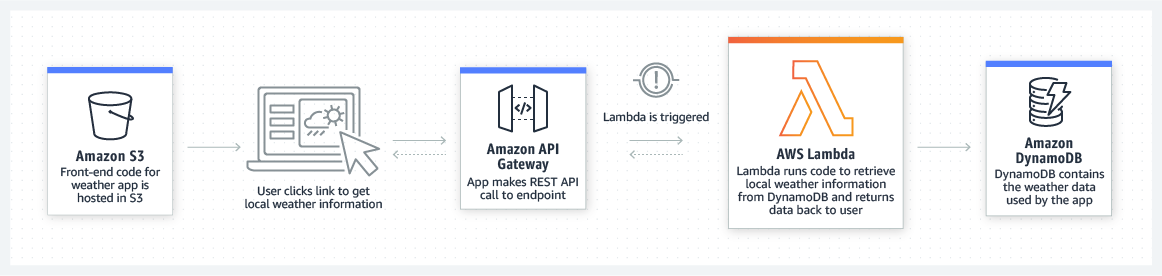


* + - Process streaming data stored in Amazon Kinesis

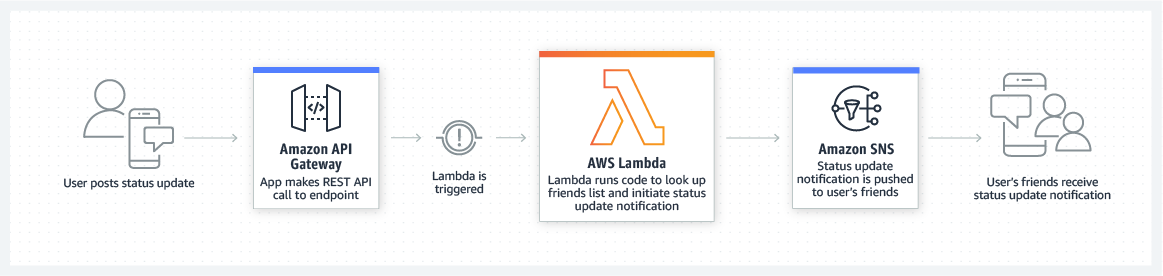


* + - Sử dụng như 1 backend, auto scaling, …

Lambda sẽ expose ra 1 API để gọi (trigger)

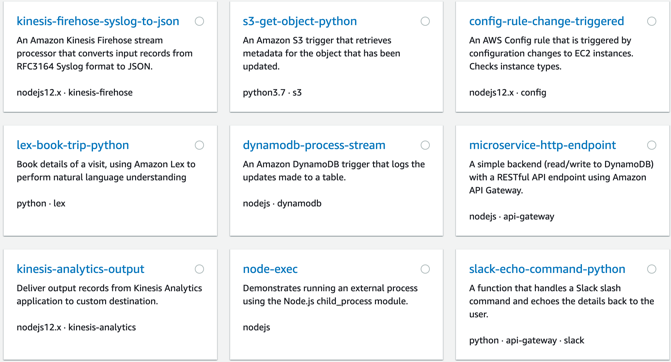


*Web application*



*Mobile application*

* + Hạn chế
    - Ổ cứng bị giới hạn là 512 MB
    - Code không quá 50MB (cả thư viện đi kèm)
    - RAM: 128MB 🡪 10GB
    - Timout: 15 minutes
  + Price
    - Tính tiền theo request hoặc theo thời gian (cực kì rẻ, gần như free)
    - <https://aws.amazon.com/lambda/pricing/>
* **Các thành phần**
  + **Lambda function**
    - Hoạt động trong 1 region
    - Các cách tạo Lambda Function
      * Author from scratch
        + Chạy các câu lệnh đơn giản
      * Blueprint
        + Có các mẫu code cho Node.js và Python



* + - * Container image (Docker)
        + Sử dụng các image có sẵn để build
      * Serverless app repository

<https://docs.aws.amazon.com/lambda/latest/dg/lambda-functions.html>

* + - Bao gồm
      * Runtime
        + Chọn language để run

Support: .NET, Go, Java, Python, PHP, …

* + - * + Architecture

x86\_64

arm64 (hiệu năng cao hơn)

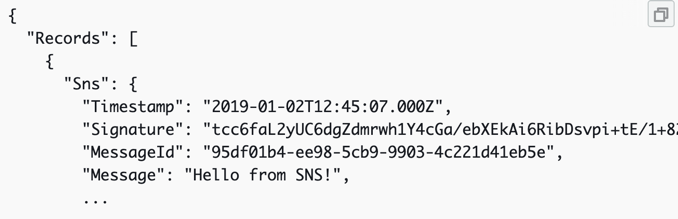
* + - * + Memory (128 MB 🡪 10,240 MB)
      * Source Code
        + Các thành phần

Event

Là parameter của Lambda Function

Đặc điểm

JSON-formatted



*Amazon SNS notification*

Contex

Callback

Return success hoặc error

Environment variable

* + - * + Có thể viết code trên

Editor

Tutorial: <https://docs.aws.amazon.com/lambda/latest/dg/foundation-console.html#code-editor>

Nén code vào file .zip rồi upload lên

Sử dụng Lambda layer để upload libraries, custom runtimes, other dependencies

Lưu file trên S3

Container image (docker)

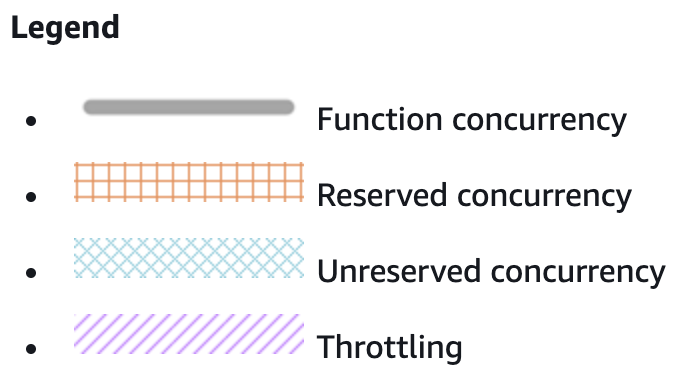


* + - * Concurrency
        + Config các lambda run đồng thời (hàng nghìn lambda có thể run đồng thời)
        + Có 2 loại

Reserved concurrency: Config max concurrency

Provisioned concurrency: Cung cấp execution environment, config response trực tiếp cho invocation

|  |  |
| --- | --- |
| Reserved concurrency allocated to two functions. | Scaling with provisioned concurrency. |



* + - * Autoscaling

Lambda tự scaling dựa trên workload

|  |  |
| --- | --- |
| Autoscaling provisioned concurrency with Application Auto Scaling. |  |

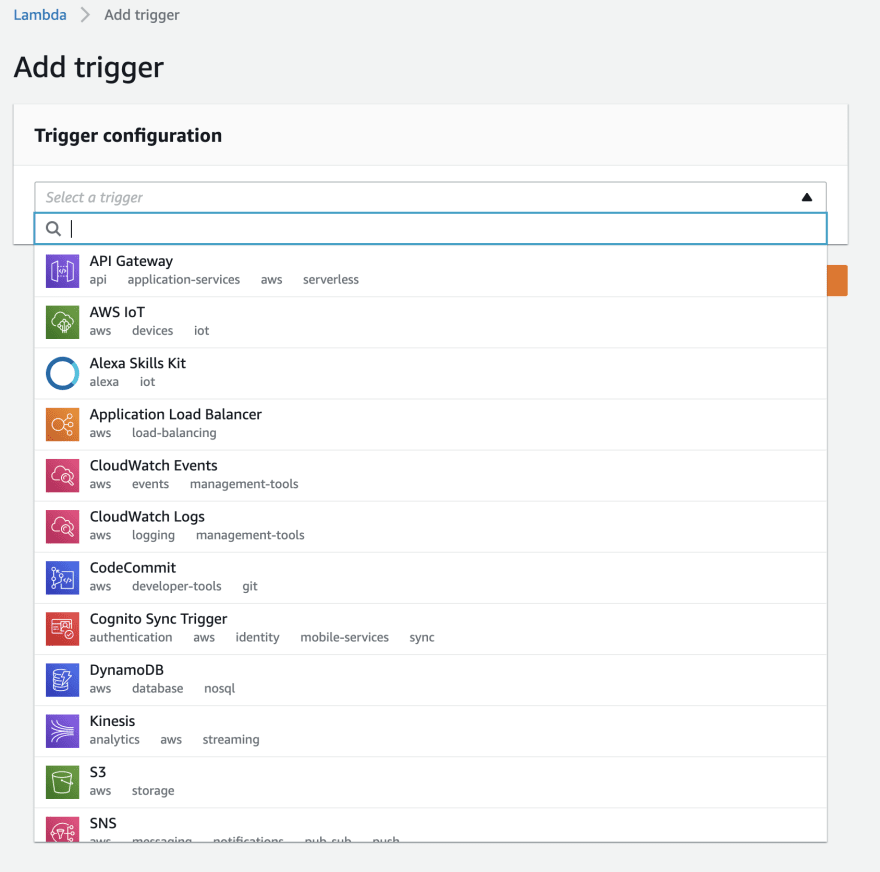
* + - * Network
        + Default thì Lambda public, cho phép truy cập qua internet
        + Có thể config để Lambda chỉ run trong VPC (private)
      * Database proxy / File system
        + Config proxy cho lambda sử dụng RDS/ File system


              The processor function processes Kinesis events and stores the events in a
                MySQL database.
            

* + - * State machine
        + Xử lý error handling cho lambda
      * Alias
      * Version
      * Security
        + Code signing
      * Monitor
        + CloudWatch

Check Log, Metric, Trace

* + - * + Cấp IAM Policies cho CloudWatch có thể ghi log
  + **Involking Function** 
    - Test
      * Max: 10 test / 1 function
    - Trigger
      * Lambda function được gọi khi xảy ra event trên các AWS Service



|  |  |
| --- | --- |
| aws lambda trigger api gateway 3 | An event source mapping reading records from a Kinesis stream. |

* + - Asynchronous / synchronous invocation
      * Synchronous invocation: đợi function xử lý xong event, sau đó mới return kết quả


      Clients invoke a function synchronously and wait for a response.
    

* + - * Asynchronous invocation: Lambda sẽ cho vào queue event để xử lý và return kết quả ngay lập tức


        Lambda queues asynchronous invocation events before sending them to the function.
      

* + - Destination
      * Khi Asynchronous invocation hoặc function xử lý record từ 1 stream xong (thành công hoặc thất bại) thì sẽ gửi invocation record tới destination
        + Invocation record bao gồm: thông tin event, response của function, reson
        + Destination có thể là

SNS Topic

SQS Queue

Lambda function

Event bus


        Lambda sends invocation records to a queue or event bus destination, depending on the result.
      

* **Config**
  + Setup

<https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html>

* **Reference**

<https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>