



CODESTAR

Lambda - Manage Service

CodeStar Academy

Nội dung chính

IaaS - PaaS - SaaS

Lambda



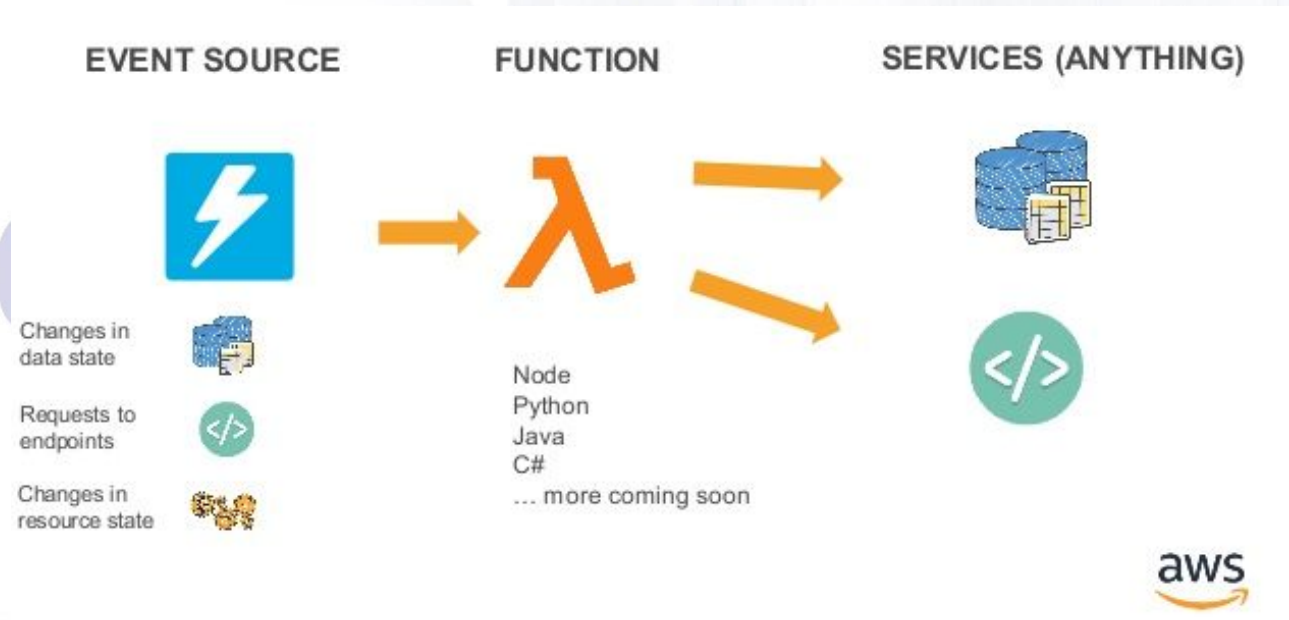
CODESTAR

IaaS - PaaS - SaaS

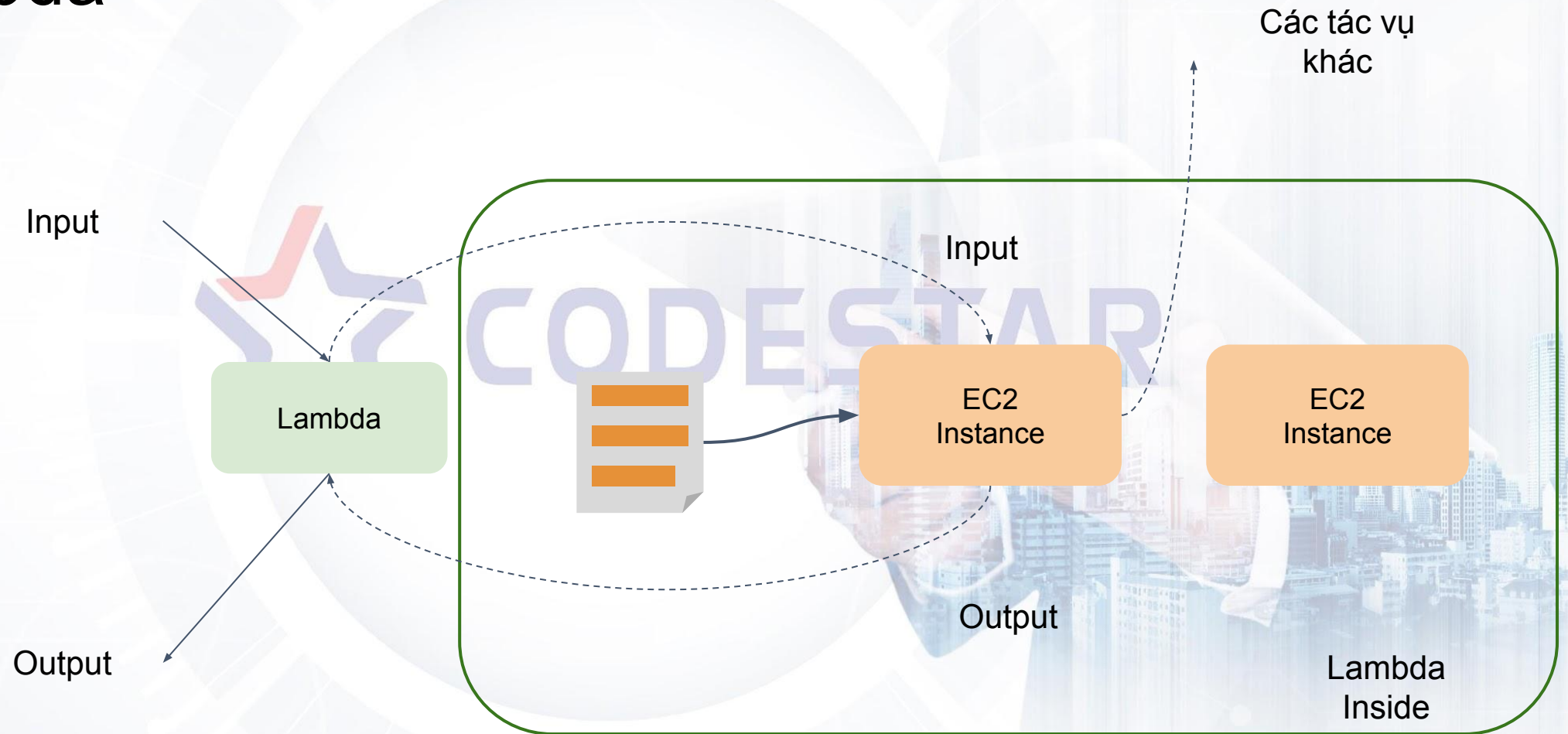
	On-premises	IaaS (Infrastructure-as-a-Service)	PaaS (Platform-as-a-Service)	SaaS (Software-as-a-Service)
Customer Responsibility	User Access/Identity	User Access/Identity	User Access/Identity	User Access/Identity
Cloud Service Provider Responsibility	Data	Data	Data	Data
	Application	Application	Application	Application
	Guest OS	Guest OS	Guest OS	Guest OS
	Virtualization	Virtualization	Virtualization	Virtualization
	Network	Network	Network	Network
	Infrastructure	Infrastructure	Infrastructure	Infrastructure
	Physical	Physical	Physical	Physical

Lambda

- Là Fully managed Service (dạng FaaS)
- Khởi tạo các concurrency để chạy và sử dụng các thao tác.
- Hiểu một cách đơn giản, Lambda sẽ giúp chúng ta tự động đưa Code vào một Instance, tính toán, biến Input thành Output.



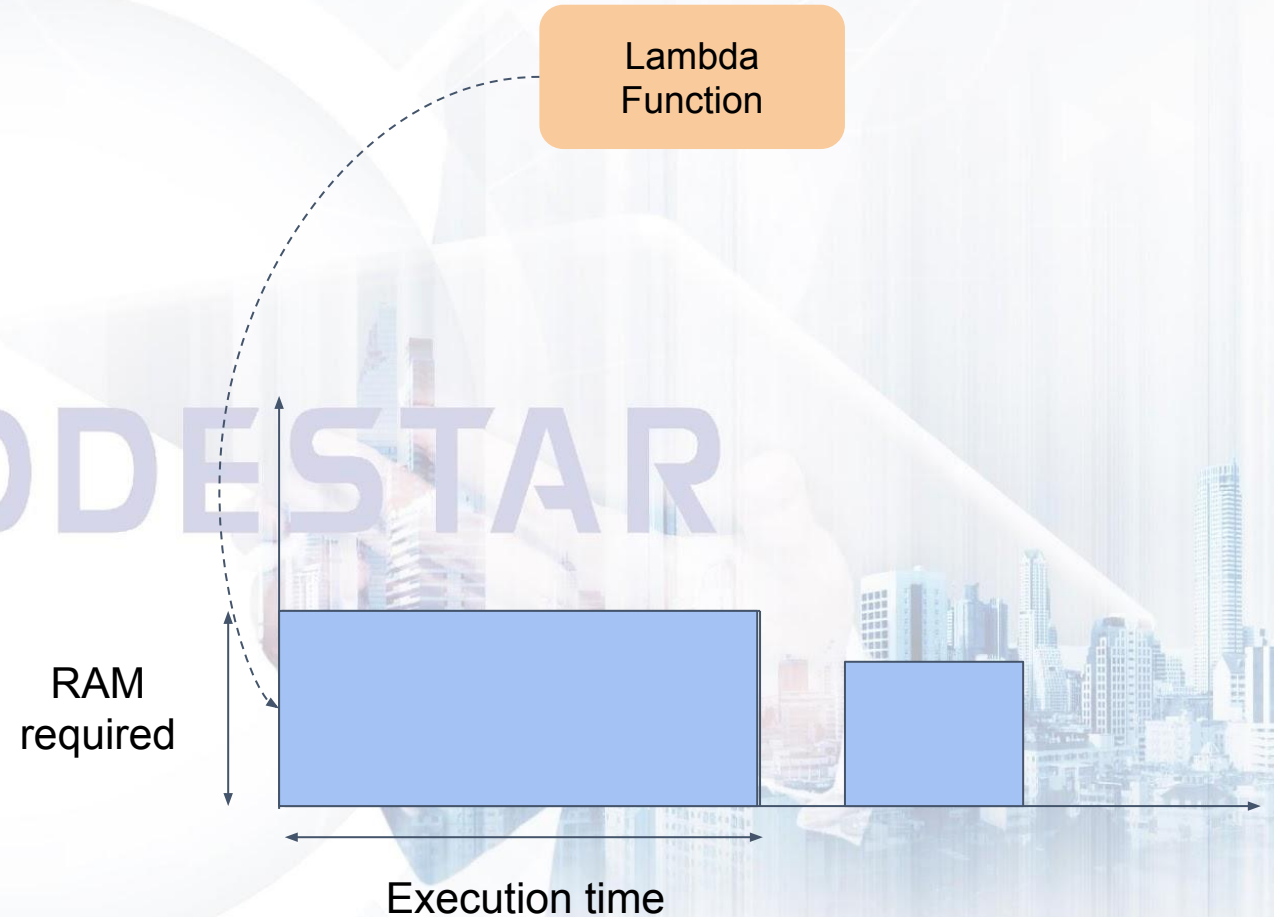
Lambda



Lambda

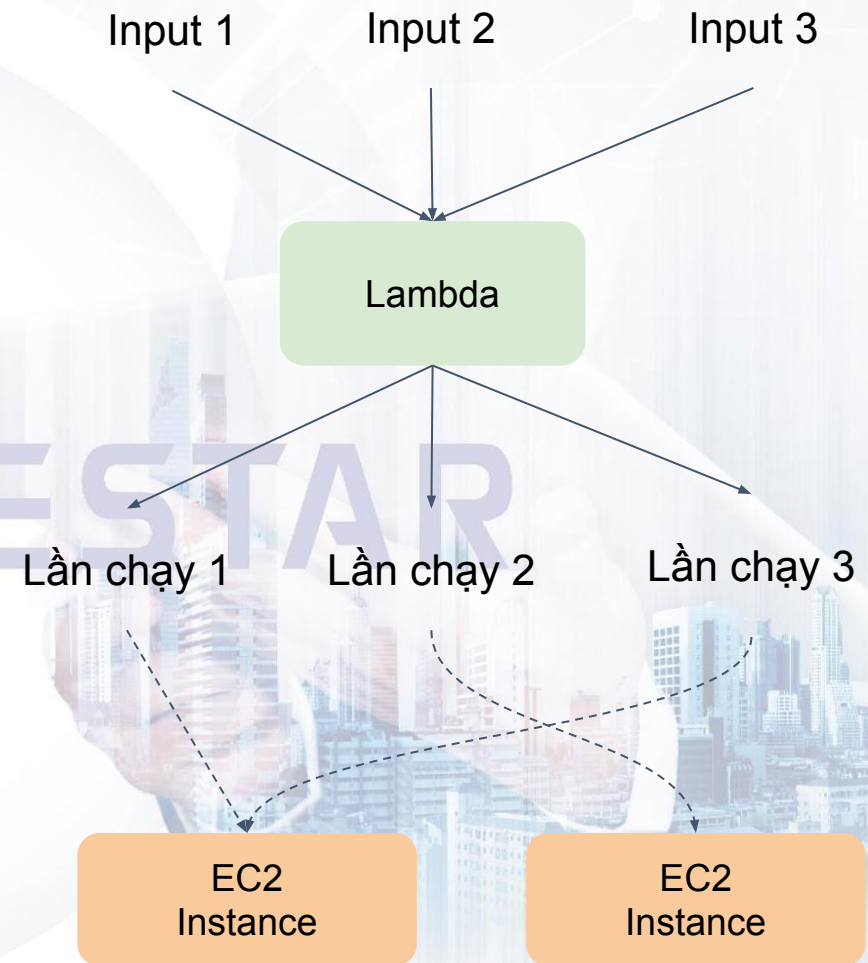
Lambda tính chi phí dựa trên:

- RAM cần sử dụng
- Thời gian chạy (kể cả thời gian chờ)



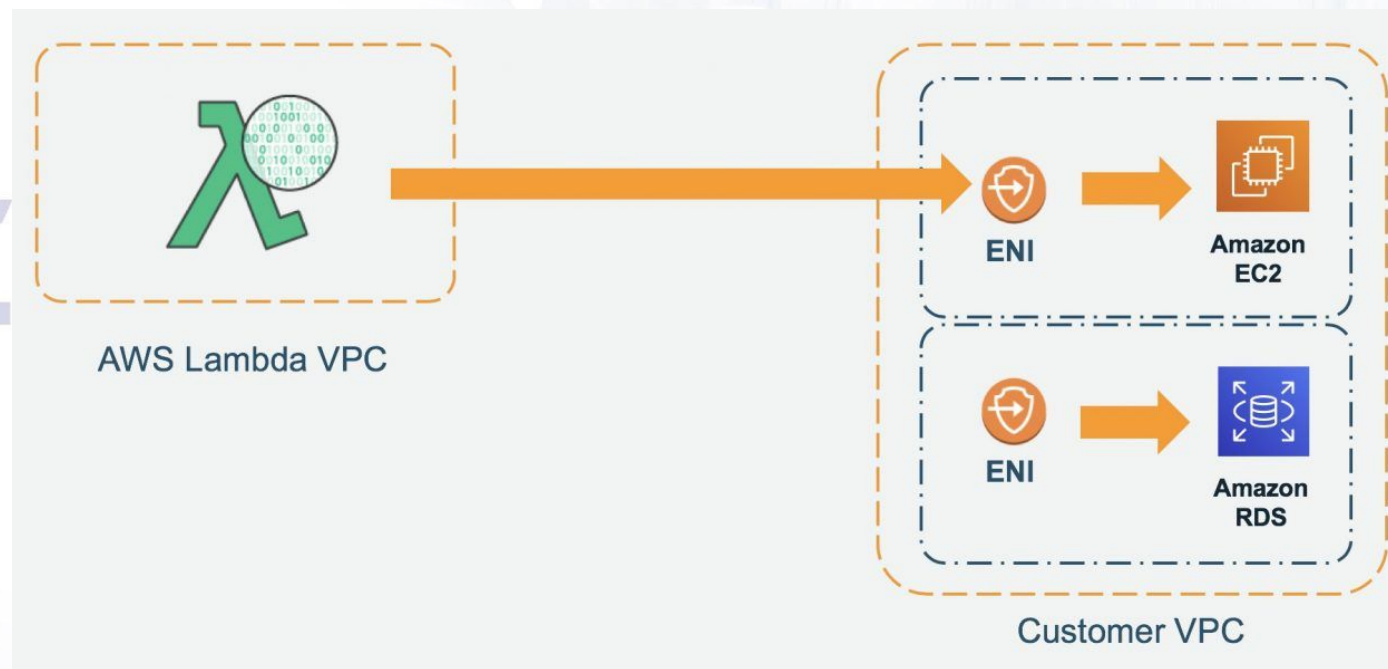
Lambda

- Mỗi lượt chạy trên Lambda, chúng ta gọi là một Concurrent Request.
- Nếu Các Request được gọi cùng 1 lúc, AWS cần chuẩn bị nhiều EC2 Instance để sử dụng
=> Giới hạn concurrency limit.



Lambda

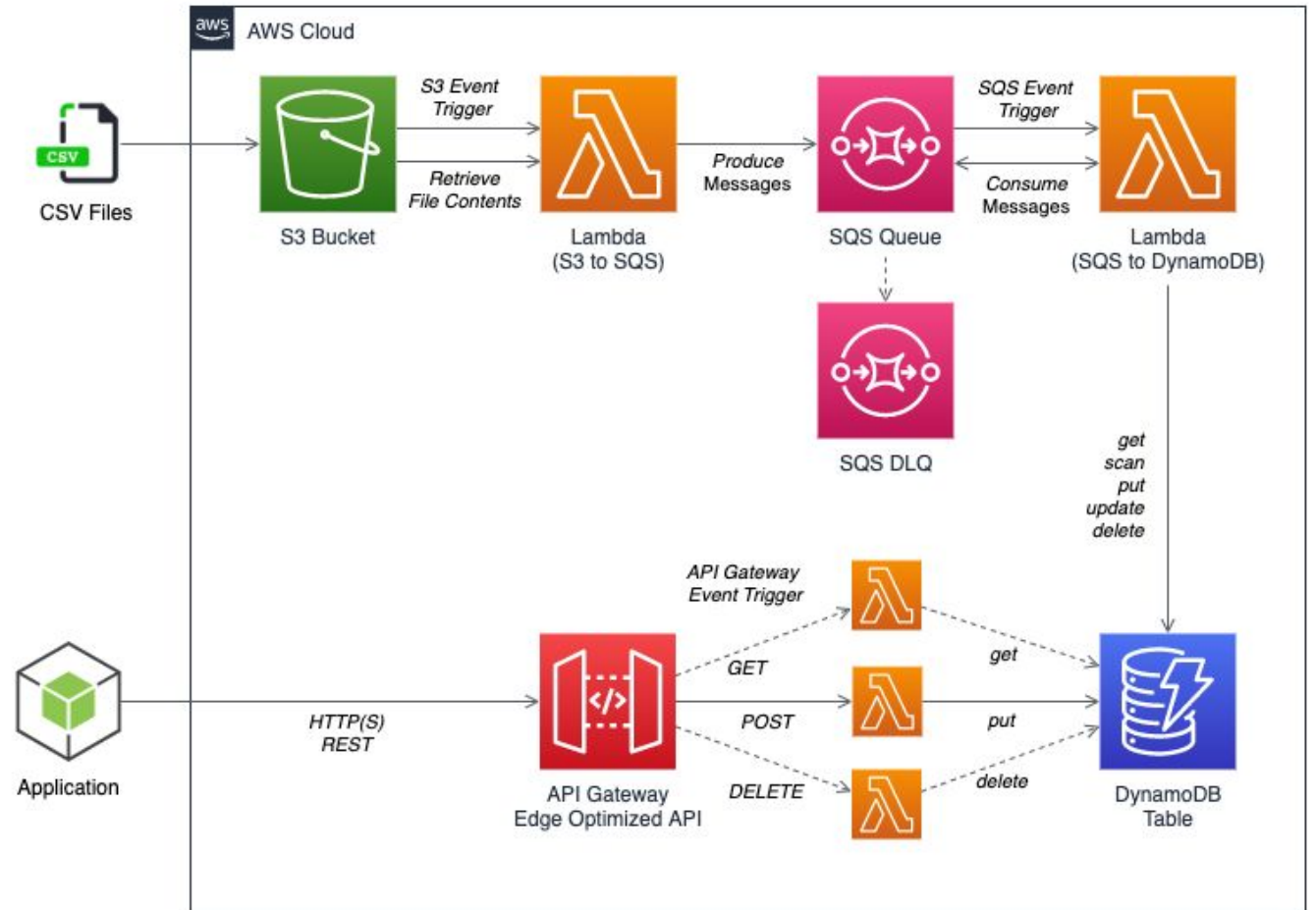
- Lambda nằm trong VPC của AWS, độc lập với các VPC chúng ta tạo ra.
- Ở chế độ thông thường, chúng ta sẽ Trigger thông qua API -> Lambda -> Thực hiện action -> Nhận về kết quả.
- Có thể sử dụng ENI để kết nối tới các thành phần nằm trong VPC.



Lambda

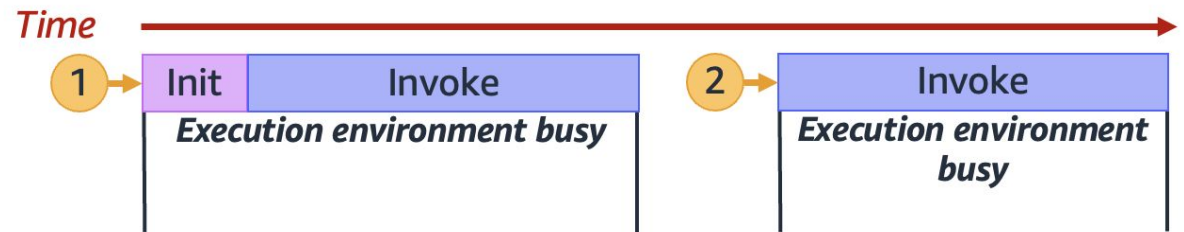
Lambda có thể được sử dụng để xây dựng kiến trúc hệ thống dạng Serverless.

Thường kết hợp với SQS, API GW, DynamoDB, S3



Lambda: Concurrency

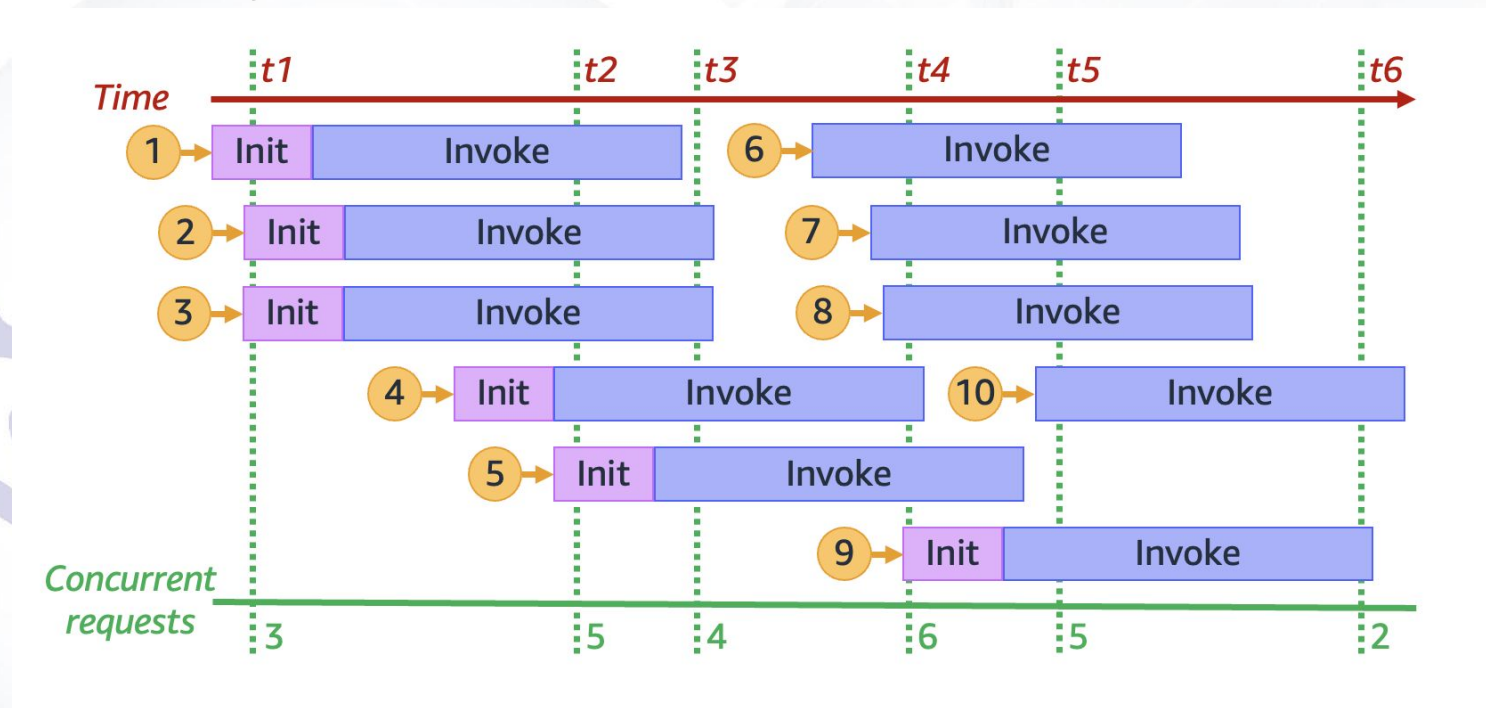
Execution time



Ở trạng thái chưa có instance, khi có request trigger lambda, sẽ có 1 khoảng thời gian (init time) để start environment.

Sau đó, nếu instance này chưa kết thúc (chưa timeout), mà vẫn có request đi tới, thì thời gian init này sẽ được bỏ qua (vẫn invoke trên instance đó)

Lambda: Concurrency



Số lượng Concurrency ở đây là 6 (1,2,3,4,5,9) ~ 6 instances sẽ chạy cùng lúc.

Request per sec \neq Concurrency

Lambda: Concurrency

Reserved Concurrency

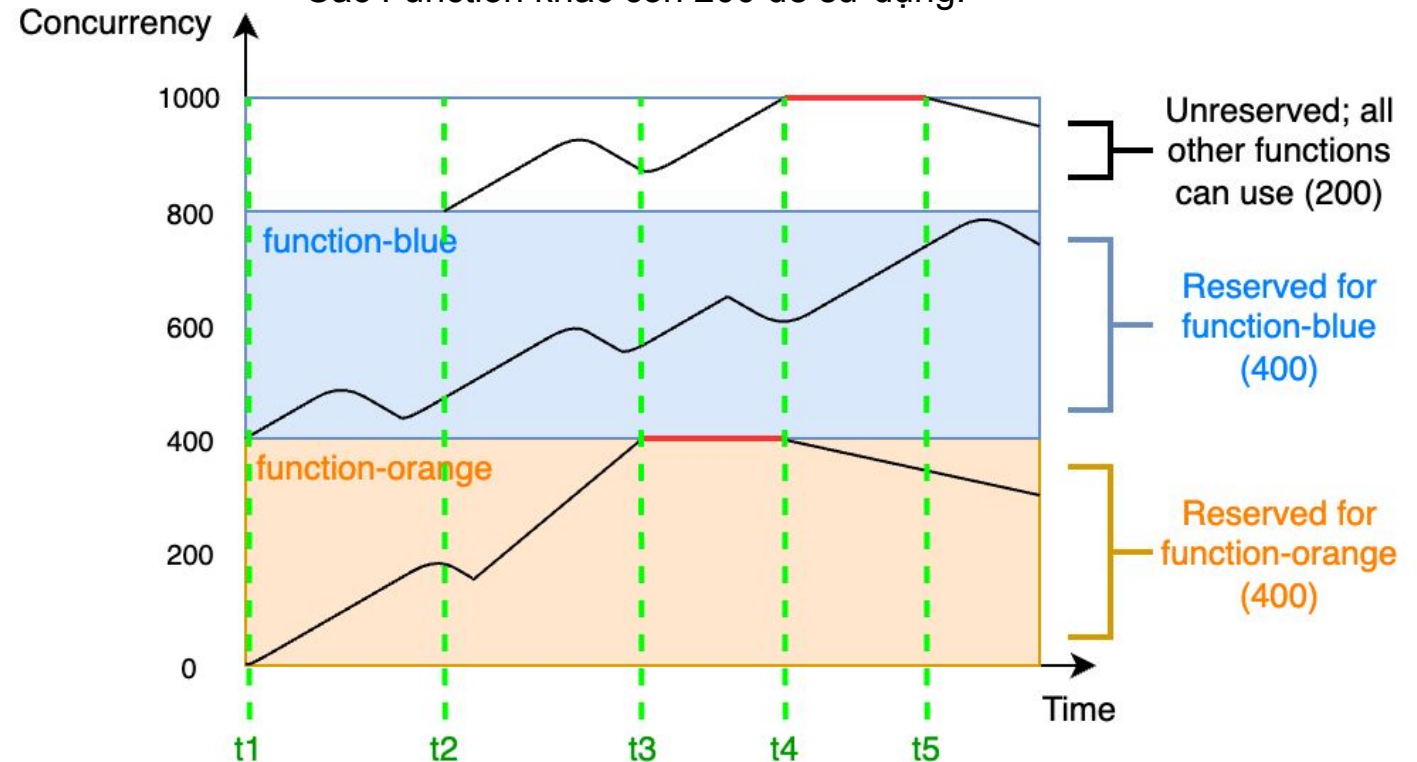
- Concurrency sẽ có limit cho từng Account.
- Phần reserved sẽ được dành riêng cho function cụ thể.
- Phần unreserved có thể được sử dụng bởi bất kỳ function nào.

Account có limit concurrency = 1000 (theo quota)

Function blue đặt RC = 400

Function orange đặt RC = 400

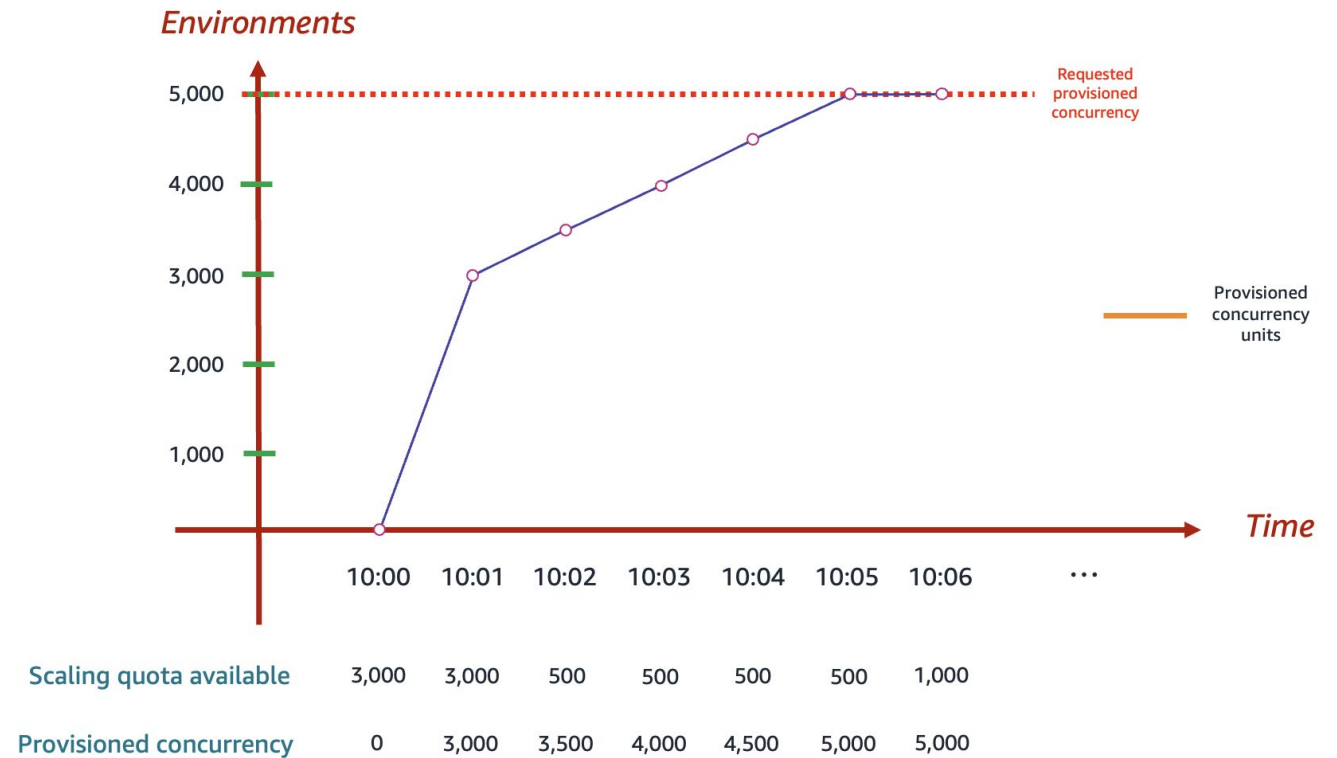
Các Function khác còn 200 để sử dụng.



Lambda: Concurrency

Bursting Concurrency

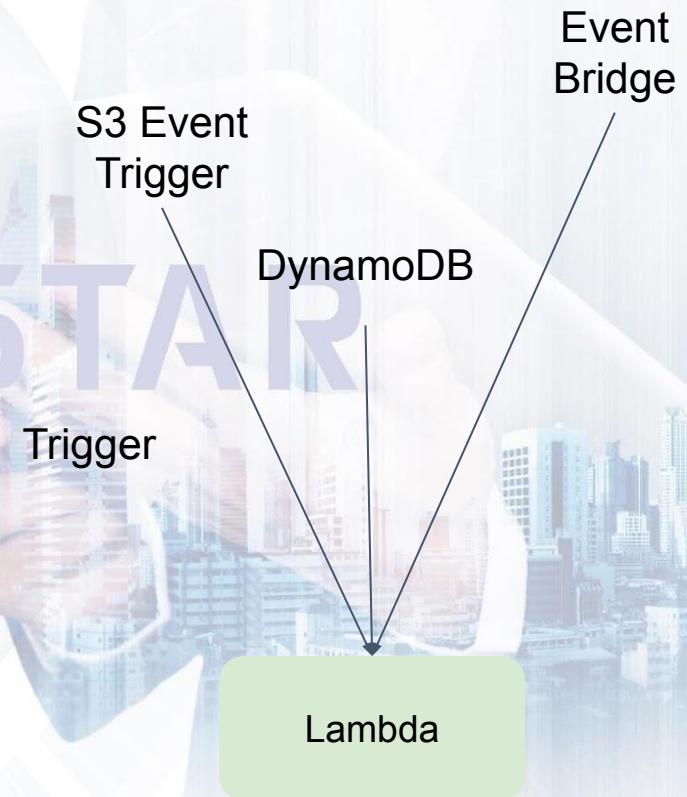
Provision có thể set max là 3000, sau đó, mỗi phút tăng thêm 500 cho tới khi đạt mốc concurrency đã request.



Lambda: Trigger

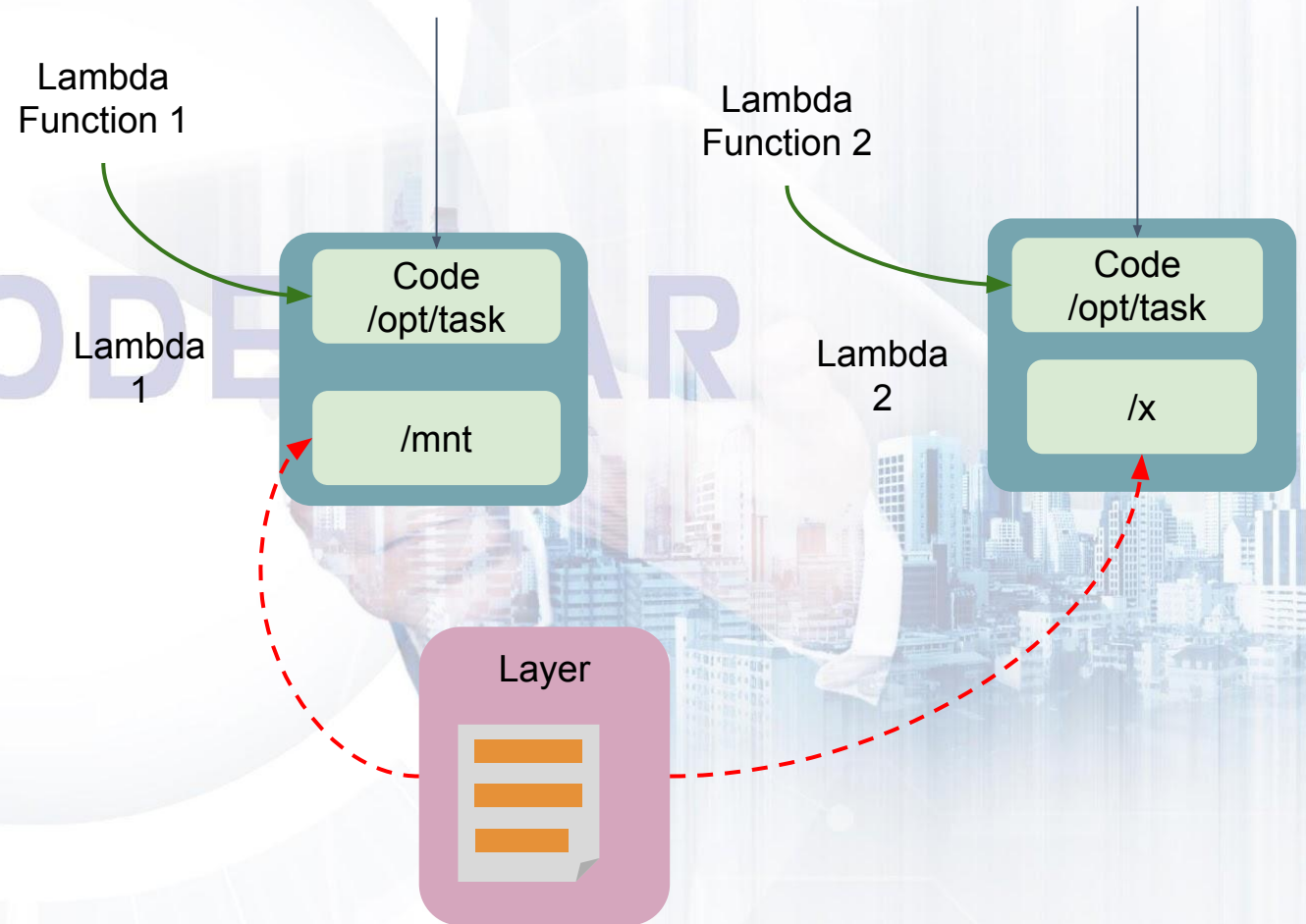
Lambda có thể được kích hoạt bởi 1 Event của một Service nào đó khác, như:

- S3
- DynamoDB Stream
- Aurora
- EventBridge
- ...



Lambda: Layer

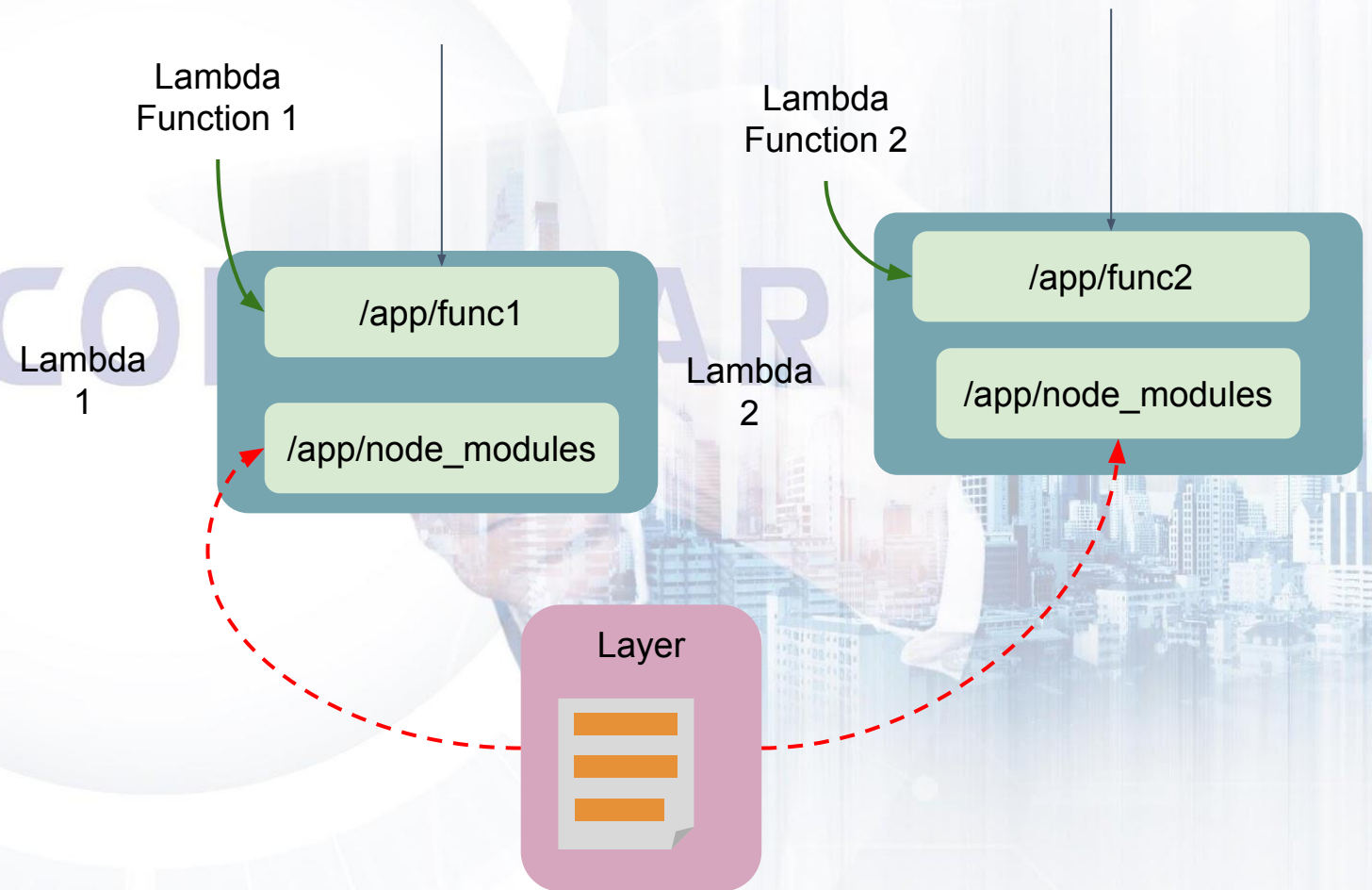
Layer cho phép đưa một/nhiều files đưa vào hệ thống để giảm thời gian download hoặc sử dụng một số module chung.



Lambda: Layer

Use case:

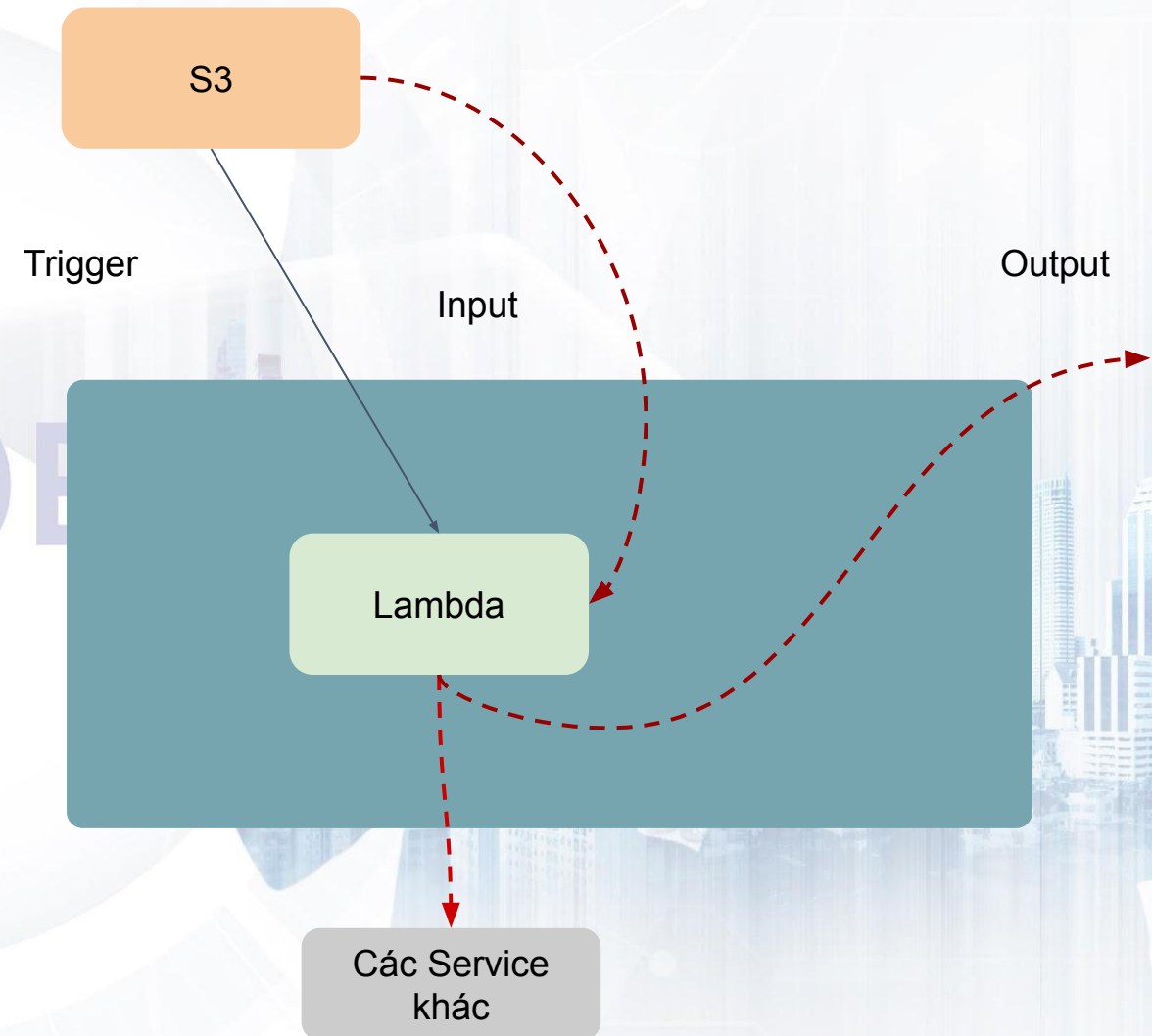
Các hàm sử dụng cùng thư viện.



Lambda: Use case

Lambda được sử dụng chủ yếu trong các use case cần thao tác với các resource AWS, Ví dụ

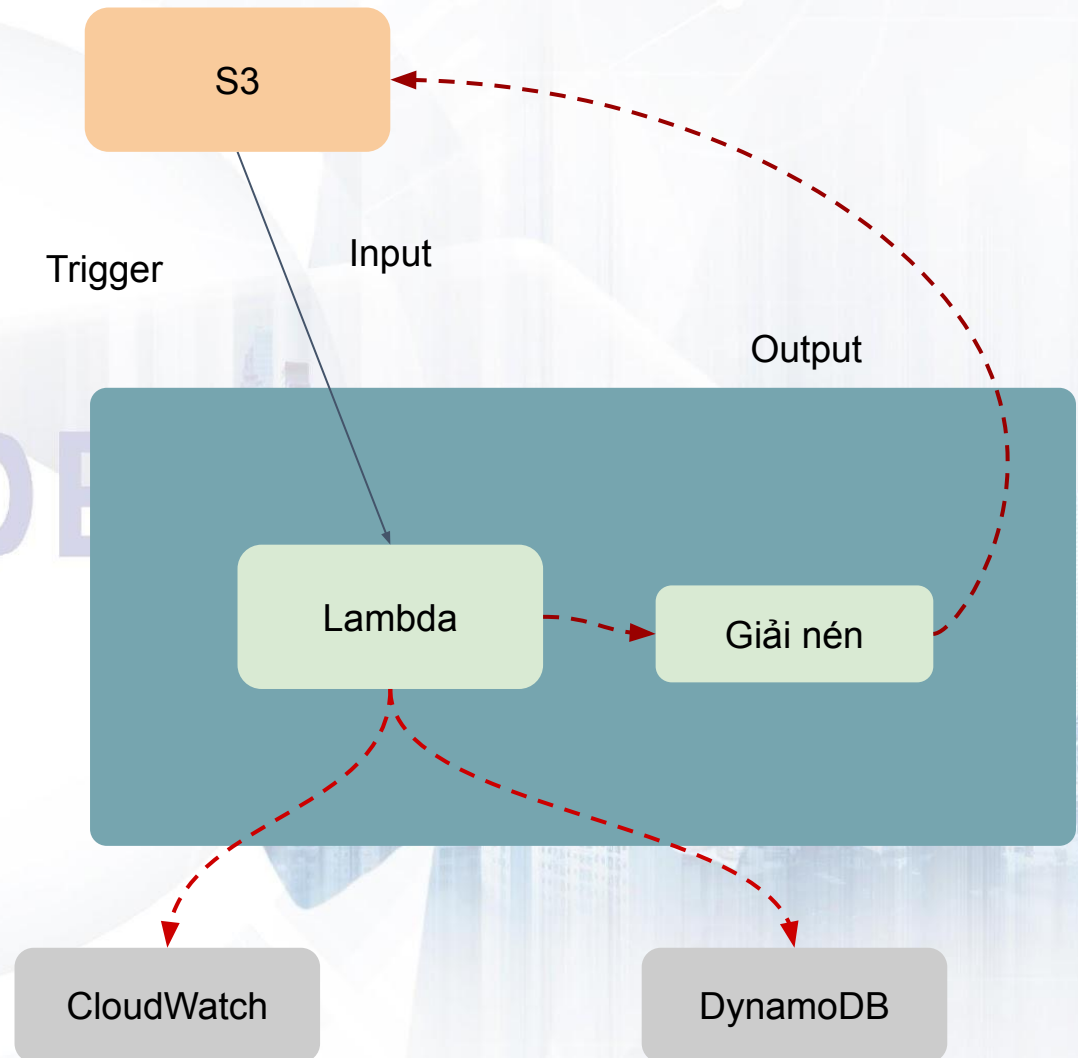
- Các thao tác cơ bản với các dịch vụ của AWS như export, resize ảnh, lưu trữ file name, đổi tên, ...
- Các thao tác tương tác với resource như bật/tắt EC2 instance, thêm IP vào blacklist trên WAF, Tạo IAM User/Role/SSO/ ...



Lambda: Use-case

Use-case:

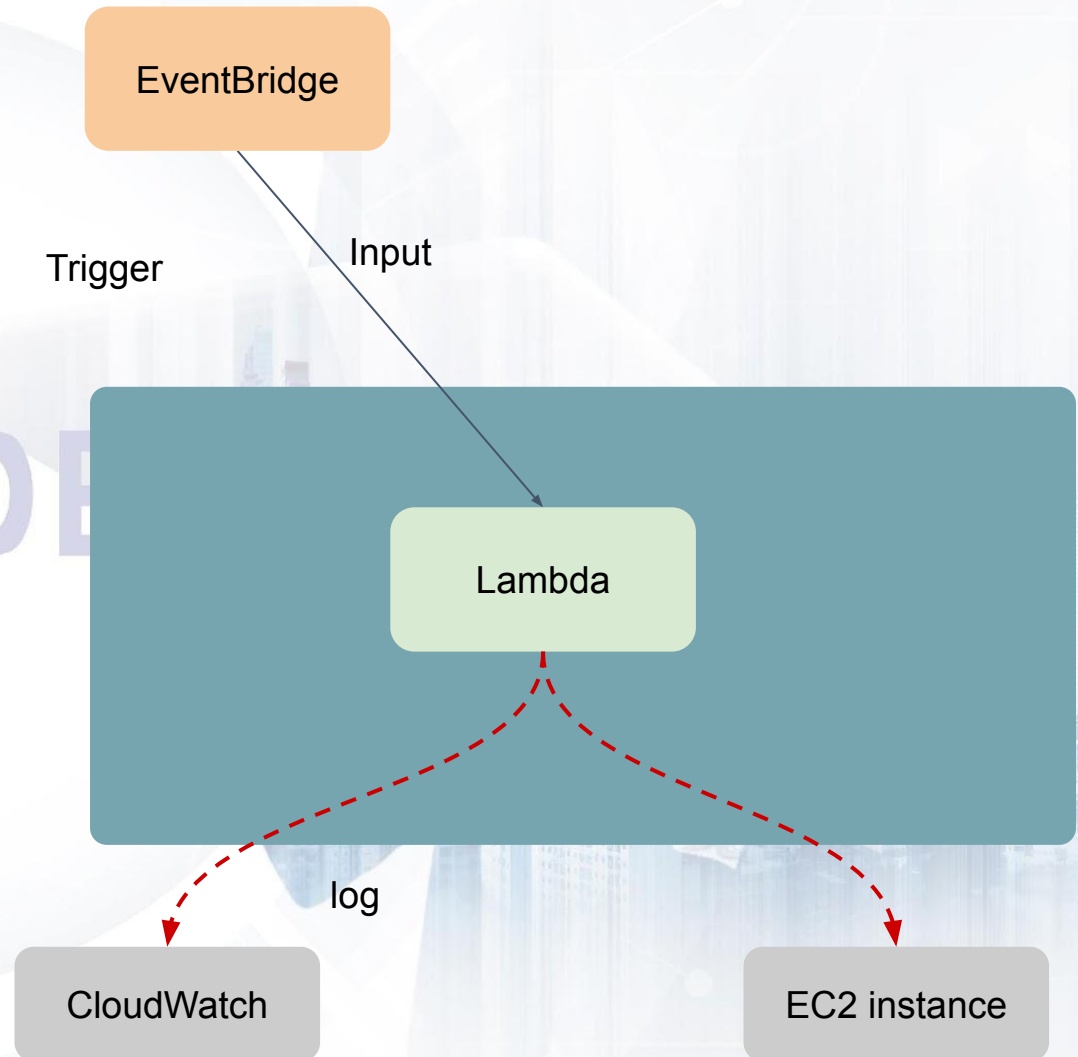
Sử dụng S3 Event Notification kích hoạt Lambda, Lambda lấy file zip vừa upload lên S3 giải nén ra, lưu trữ vào chính S3 vừa rồi và lưu lại thông tin trên DynamoDB.



Lambda: Use-case

Use-case:

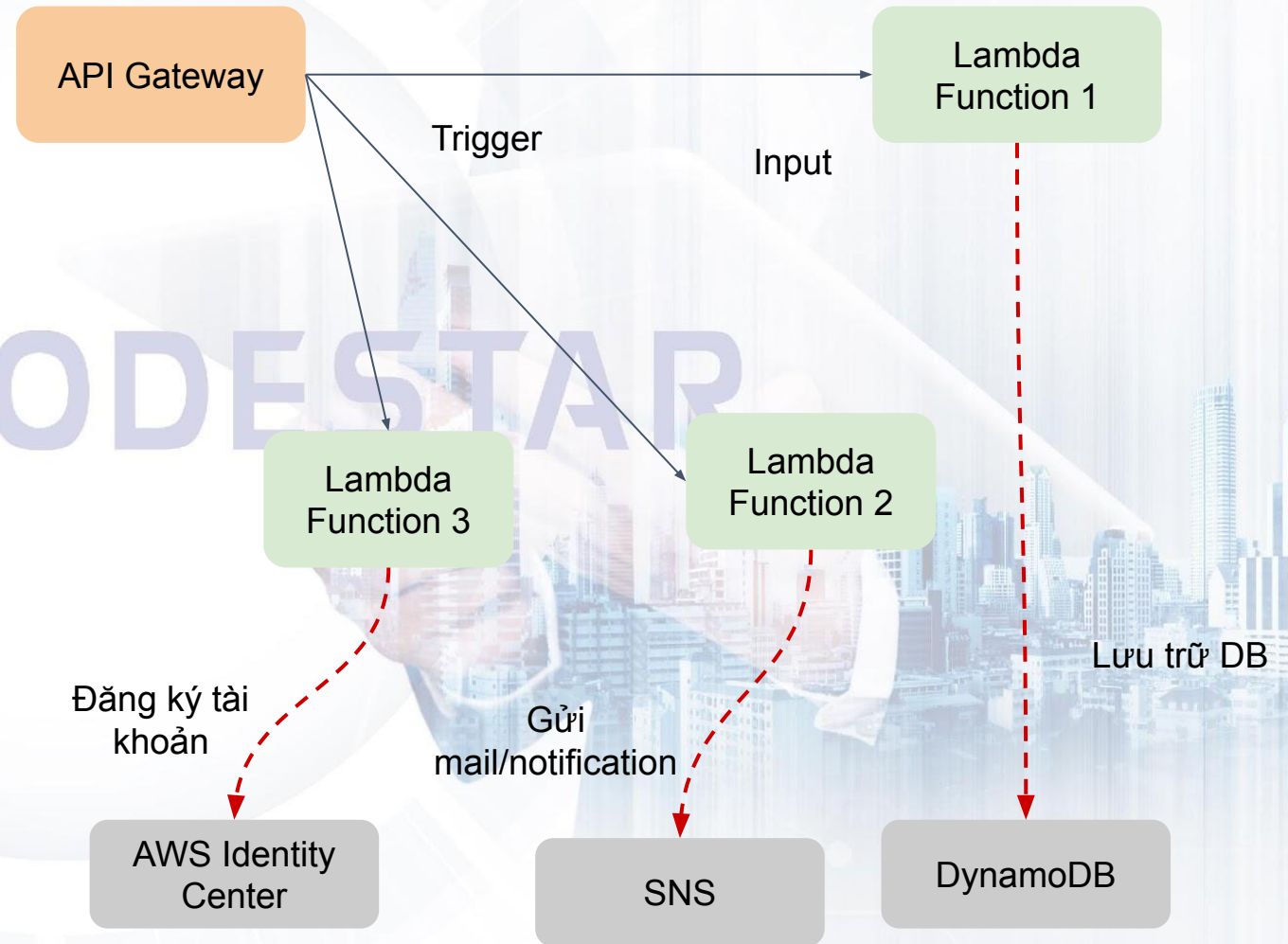
Sử dụng Event Bridge hẹn giờ
bật EC2 instance cho môi
trường Staging



Lambda: Use-case

Use-case:

Sử dụng làm API server cho hệ thống web có một số thao tác căn bản.





THANK YOU