

# AWS Cloud for beginner

Instructor: Linh Nguyen

(Engineering Consultant, AWS Cloud Solution Architect)

Level: Beginner

*“Không có việc gì khó, chỉ sợ không biết làm”*

# Content Delivery Network CloudFront

*“Một hệ thống tốt nhưng tốc độ chậm thì sẽ không có ai sử dụng. Trong một thế giới mà người dùng ngày càng thiếu kiên nhẫn, tốc độ luôn là mối quan tâm hàng đầu.”*

# Target

- Hiểu được CloudFront là gì? Khi nào sử dụng CloudFront.
- Các thành phần của CloudFront.
- Sử dụng CloudFront làm caching cho static website.
- Sử dụng CloudFront làm caching cho Application Load Balancer.
- Hiểu rõ về behavior, caching policy trên CloudFront.

Copyright@Linh Nguyen on Udemy

# What is CloudFront

Là một CDN service của AWS.

Vậy CDN là gì?

=>Viết tắt của Content Delivery Network, một mạng lưới giúp delivery nội dung tới người dùng cuối một cách nhanh chóng nhờ vào việc điều hướng request của họ tới các máy chủ chứa tài nguyên gần nhất.

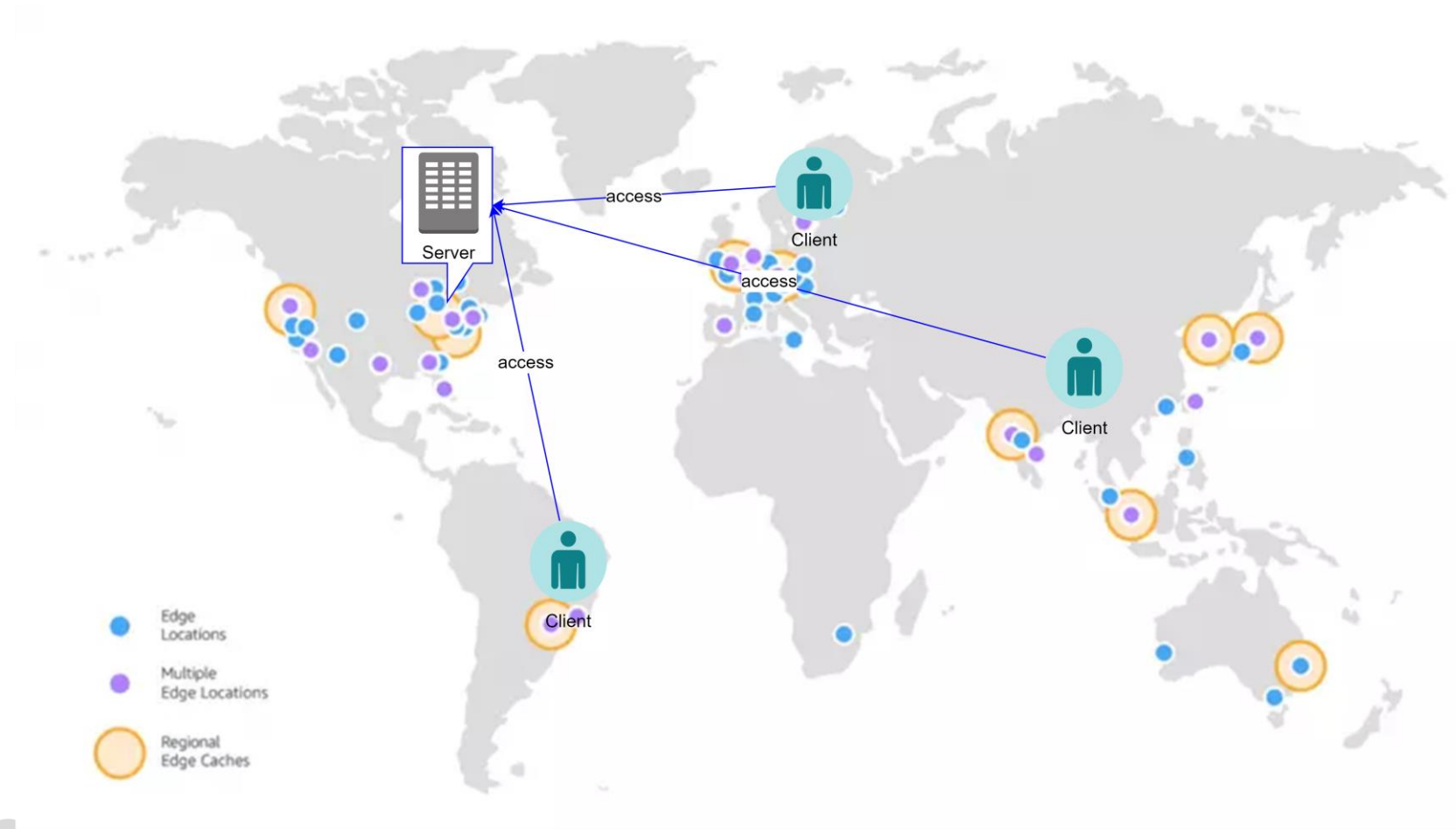


Amazon CloudFront

Copyright@Linh Nguyen on Udemy

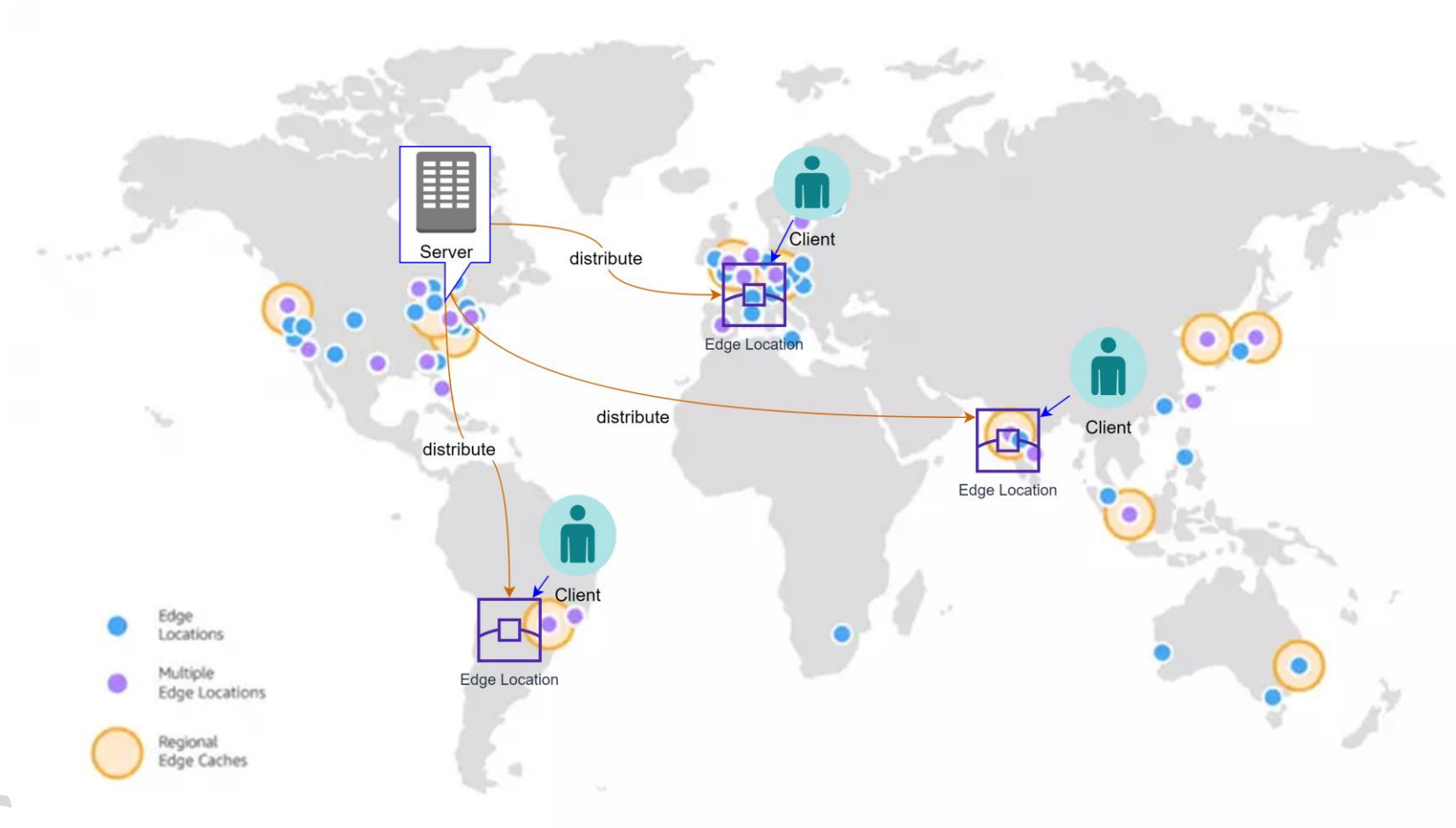
# Mô hình hoạt động của một mạng CDN

Khi không có CDN, tài nguyên của server sẽ được deliver tới client một cách trực tiếp.  
=> Tùy vào khoảng cách địa lý mà tốc độ truy cập sẽ nhanh hay chậm.



# Mô hình hoạt động của một mạng CDN

Khi có CDN, tài nguyên của server sẽ được cache trên các máy chủ Edge, request của user tới một tài nguyên trên CloudFront sẽ được redirect tới máy chủ Edge gần nhất.



# Các khái niệm cơ bản của CloudFront

- **Distribution:** AWS CloudFront phân phối nội dung từ origin đến các edge location thông qua một distribution. Một distribution định nghĩa cách CloudFront phân phối nội dung, bao gồm địa chỉ của origin, các edge location được sử dụng, các thiết lập security & caching.
- **Edge Location:** AWS CloudFront sử dụng một mạng lưới toàn cầu các edge location, là các điểm đặt máy chủ trên khắp thế giới, để phân phối nội dung đến người dùng ở gần nhất vị trí đó.
- **Origin:** Đây là nơi lưu trữ nội dung gốc (origin content) của bạn, bao gồm các tập tin, ứng dụng web, API và Database. Origin có thể là một Web server, S3 bucket hoặc các dịch vụ AWS khác.

Copyright@Linh Nguyen on Youtube

# Các khái niệm cơ bản của CloudFront

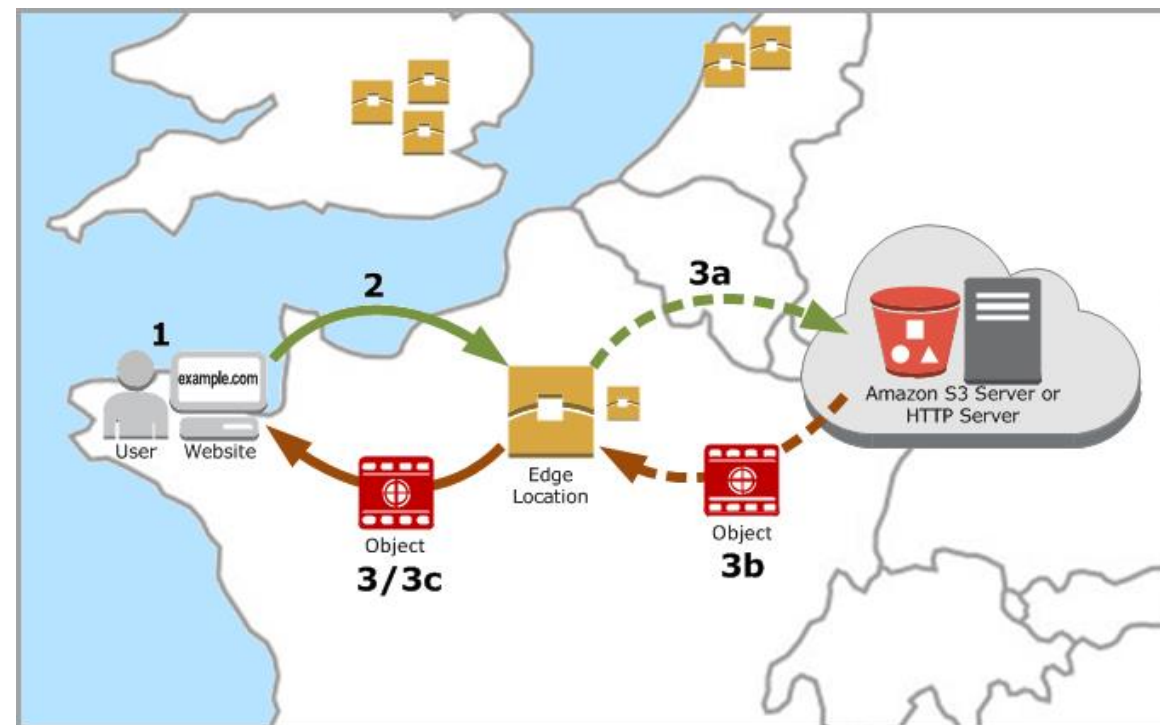
- **Cache:** AWS CloudFront lưu trữ các tài nguyên tại edge location để giảm thời gian phản hồi và tăng tốc độ tải trang web. Các tài nguyên này bao gồm hình ảnh, tập tin CSS và JavaScript.
- **Logging and Reporting:** AWS CloudFront cung cấp các báo cáo về hoạt động của distribution, bao gồm lưu lượng và số lần truy cập.
- **Security:** AWS CloudFront hỗ trợ nhiều tính năng bảo mật, bao gồm kết nối HTTPS, chữ ký số (Certificate) và xác thực người dùng.
- **Customize at the Edge:** thông qua cơ chế Lambda@Edge, cho phép bạn thực thi các function trên các sự kiện CloudFront. Lợi thế về tốc độ và hiệu suất so với thực thi ở Origin. Một số use-case có thể kể đến như: Authen/Author, xử lý tính toán đơn giản, SEO, Intelligent routing, Anti Bot, Realtime image transformation, A/B Testing, User prioritization.



# How CloudFront deliver content to user?

1. User access website, request một tài nguyên vd HTML file, Image, CSS..
2. DNS điều hướng request của user tới CloudFront edge location gần nhất (dựa theo độ trễ).
3.
  - a) CloudFront forward request tới Origin server (một HTTP server hoặc s3 bucket).
  - b) Origin server trả kết quả cho edge location.
  - c) Ngay sau khi nhận được firstbyte response, edge location forward object tới end-user đồng thời cache lại nội dung cho request lần sau.

\*Từ lần request thứ 2 trở đi, mặc định edge location sẽ trả về object được yêu cầu mà không gọi tới origin server -> tăng tốc độ truy cập.



Nguồn:

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/HowCloudFrontWorks.html>

# Usecase của CloudFront

CloudFront được sử dụng cho một số usecase sau:

- Tăng tốc website (Image, CSS, Document, Video,...)
- On demand video & video streaming.
- Field level encrypt: CloudFront tự động mã hoá data được gửi lên từ người dùng, chỉ có backend application có key có thể giải mã.
- Customize at the edge: Trả về mã lỗi khi server maintain hoặc authorize user trước khi forward request tới backend. Cần sử dụng Lambda@Edge.

Copyright@Linh Nguyen on Udemy

# CloudFront pricing

## **CloudFront tính phí dựa trên các mục sau:**

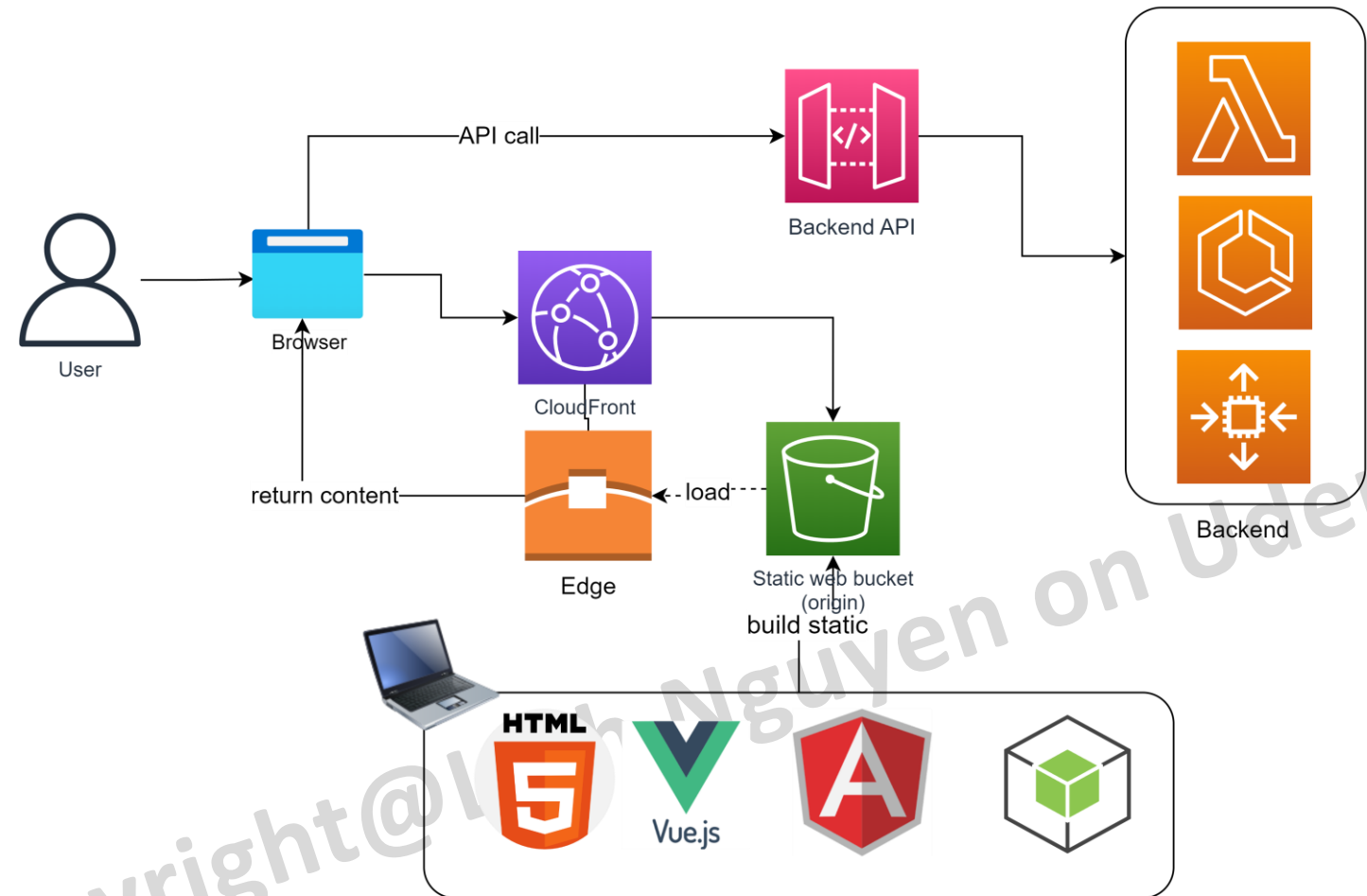
- No Up-front fee, người dùng chỉ trả tiền cho những gì sử dụng.
- Lượng data thực tế transfer out to internet.
- CloudFront function, Lambda@Edge.
- Invalidation Request (clear cache).
- Real-time log.
- Field level encrypt.

Copyright@Linh Nguyen on Udemy

# Usecase sample: hosting static website

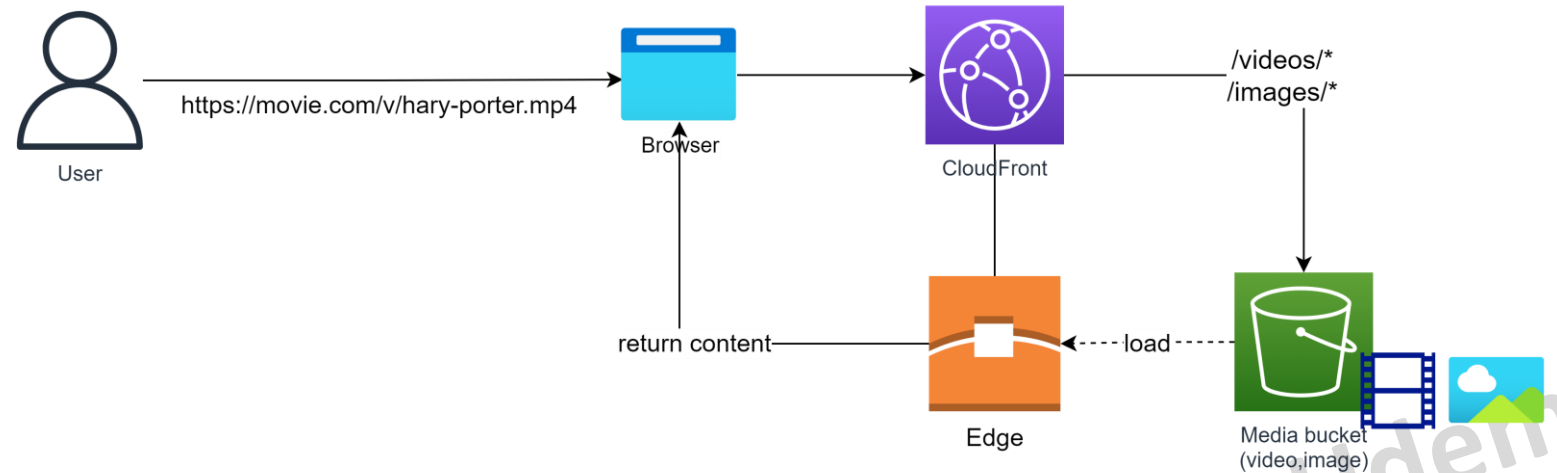
Website static (chỉ gồm HTML/css/js...) có thể được deploy lên S3 kết hợp với CloudFront.

Hầu hết các framework hiện nay như Angular, Vue, Nodejs đều hỗ trợ build website thành dạng static chỉ gồm HTML, css, javascript để deploy lên S3.



# Usecase sample: serve media (video, image)

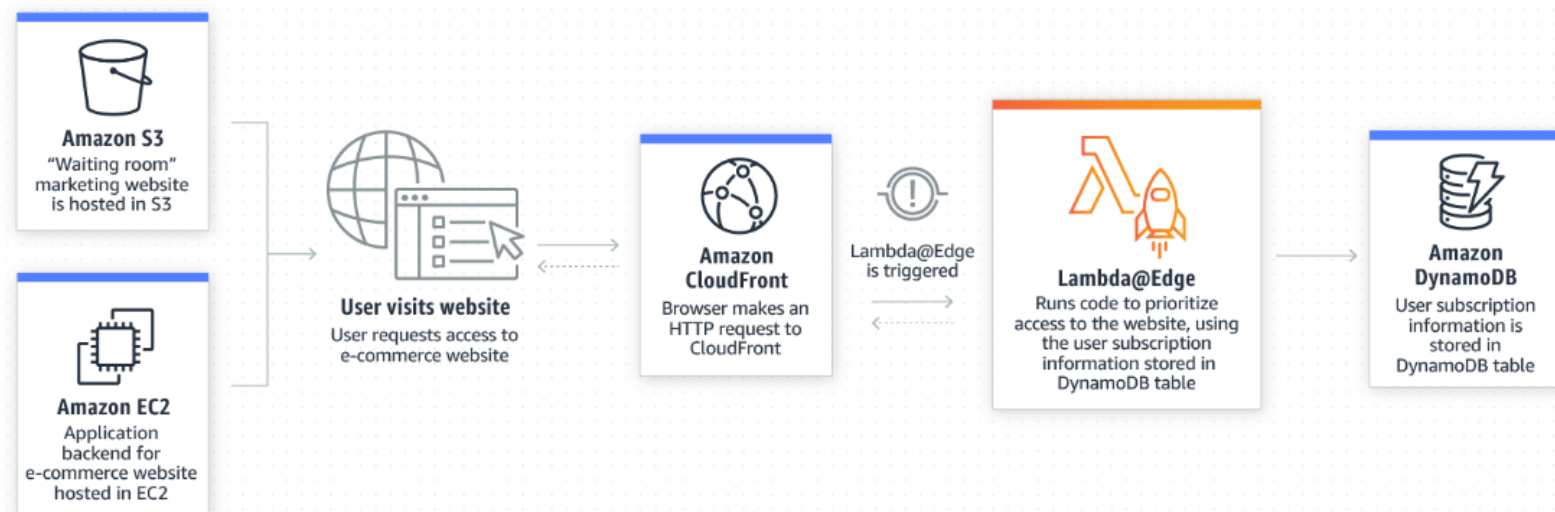
Media được phân phối tới end user thông qua CloudFront cho trải nghiệm tốt hơn do nội dung được redirect tới Edge Location gần với User nhất.



# Usecase sample: customize at the edge

Sử dụng Lambda@Edge để phân biệt user sử dụng subscription information lưu ở DynamoDB (có enable Global table).

Tùy theo subscription khác nhau (premium/normal) mà sẽ có các ưu tiên về xử lý khác nhau, vd user premium được redirect tới máy chủ mạnh hơn user normal.



\*Nguồn: <https://aws.amazon.com/lambda/edge>

# CloudFront behavior

Một CloudFront distribution có thể có nhiều hơn một Origin server phía sau.

VD: một hệ thống gồm có 1 S3 bucket host static website, 1 backend server gồm ALB + Container muốn sử dụng chung một CloudFront distribution.

=> Cần có cơ chế phân biệt request nào sẽ điều hướng tới đâu.

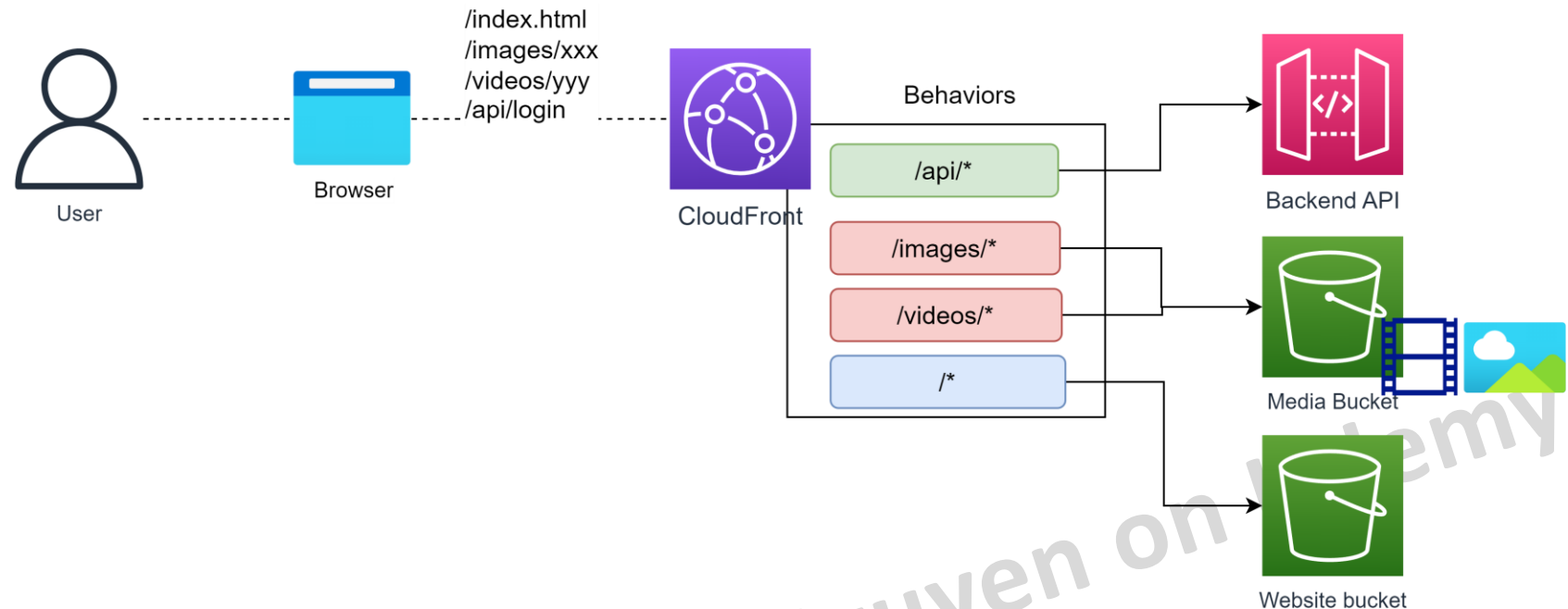
Behavior cho phép định nghĩa request với các pattern khác nhau sẽ được forward tới các origin khác nhau.

Copyright@Linh Nguyen on Udemy

# CloudFront behavior

Bằng việc cấu hình các behavior khác nhau, CloudFront giúp điều hướng request của người dùng tới đúng origin mong muốn.

\*Lưu ý khi setting behavior cần quan tâm tới thứ tự trước sau của các behavior, vì khi match 1 path rồi sẽ không đánh giá path tiếp theo.



\*Thứ tự của các behavior rất quan trọng, trong hình nếu để `/*` lên trên cùng, sẽ làm các behavior bên dưới bị sai lệch.



# CloudFront Cache Policy

- CloudFront cache policy là một cấu hình định nghĩa CloudFront sẽ cache và serves content như thế nào đối với User.
- Cache policy có thể được định nghĩa riêng cho từng distribution hoặc behavior. Một số rule có thể định nghĩa vd: thời gian content được cache, compress hay không?, forward cookies & query strings hay không?
- Việc apply caching policy khác nhau cho từng distribution hoặc behavior, cho phép tinh chỉnh việc control caching cho từng URL hoặc request pattern. Việc này giúp tối ưu hoá caching behavior cho từng loại content khác nhau, vd: Static assets (image, css, js, video...) hoặc dynamic API response.

Copyright@Linh Nguyen on Udemy

# CloudFront Origin request policy

- CloudFront origin request policy định nghĩa cách CloudFront xử lý request tới origin.
- Khi user request content trên CloudFront, request được forward tới các origin như S3, EC2, ALB. Origin Request Policy cho phép modify request trước khi forward tới origin.
- Bạn có thể add/modify/remove header hoặc query string để optimize caching, tăng performance. Ngoài ra bạn có thể cấu hình CloudFront sign hoặc encrypt request để bảo vệ backend khỏi những access unauthorized.
- Origin request policy hữu dụng khi có kiến trúc backend phức tạp vd như có nhiều loại origin server.

Copyright@Linh Nguyen on Udemy

# Lab 1 – Sử dụng CloudFront để host static web

*\*Yêu cầu đã học qua bài S3, biết cách host một website tĩnh lên S3 bucket.*

1. Chuẩn bị 1 website tĩnh, nếu có vài hình size lớn (>5mb) thì càng tốt.
2. Thực hiện host website tĩnh sử dụng S3.
3. Tạo CloudFront distribution, đăng ký S3 làm origin.
4. Thử truy cập thông qua CloudFront, so sánh tốc độ khi truy cập trực tiếp S3.
5. Disable public access cho S3, tạo OAI cho CloudFront, thiết lập policy chỉ cho phép truy cập resource thông qua CloudFront.
6. [Optional] phát hành SSL certificate và sử dụng CloudFront custom domain. Mua 1 tên miền và trỏ sang CloudFront, truy cập thông qua domain đã setting.

*\*Phần này bài **Route53** sẽ được trình bày rõ hơn.*

## Lab 2 – CloudFront + API Gateway + S3

\*Yêu cầu đã học qua bài S3, biết cách host một website tĩnh lên S3 bucket.

\*Yêu cầu đã học qua bài API Gateway, biết cách deploy một API đơn giản.

1. Chuẩn bị 1 website tĩnh deploy lên S3.
2. Tạo một API Gateway đơn giản, deploy thành một stage.
3. Tạo CloudFront distribution, đăng ký S3 làm origin (hoặc tái sử dụng lại bài lab 1)
4. Add thêm origin cho API Gateway.
5. Cấu hình behavior phù hợp cho API Gateway và S3.
6. Cấu hình Caching Behavior No-Cache cho API Gateway.
7. Modify code backend thử in ra request header (cần enable Proxy trên API GW)
8. Cấu hình forward một header vd “Source” = “CloudFront”

## Lab 3 – Sử dụng CloudFront cho single-page

\*Yêu cầu đã xem qua bài S3, biết cách host một website tĩnh lên S3 bucket.

1. Upload source lên S3, enable static website, liên kết với CloudFront, test truy cập thông qua cloudfront (nhớ thêm /index.html phía sau).
2. Cấu hình Lambda@Edge cho origin Website, nhiệm vụ của Lambda@Edge này là thêm index.html vào mỗi request rồi forward tiếp. \*Đặc trưng của single page là luôn luôn phải có **index.html**, nếu client không được redirect tới 1 file có trên origin, CloudFront sẽ trả về lỗi 403.
3. Thử truy cập website mà không cần có **index.html** ở phía sau.

Copyright@Linh Nguyen on Udemy

# Clear resources

Về cơ bản, các resource tạo ra trong bài lab có Zero Idle cost tuy nhiên các bạn có thể tiết kiệm chi phí bằng cách:

1. Xoá các file lớn trên S3 nếu có
2. Xoá CloudFront distribution.

Copyright@Linh Nguyen on Udemy

# Lab 2 – CloudFront with Wordpress

## **Yêu cầu:**

- Đã học qua bài S3, EC2, Elastic Load Balancer, RDS.
- Biết cách cài wordpress server lên EC2 (hoặc chọn AMI cài sẵn wordpress).
- Biết thao tác đơn giản với Linux, vim,...

## **Steps:**

1. Tạo EC2, cài wordpress hoặc sd AMI có sẵn Wordpress.
2. Tạo RDS, cấu hình cho Wordpress sử dụng RDS.
3. Cài plugin “W3 Total Cache” cho wordpress
4. Tạo S3 bucket, tạo CloudFront distribution link tới bucket.

## Lab 2 – CloudFront with Wordpress

5. Gán role S3 full access trên bucket cho Wordpress Instance.
6. Thiết lập CDN setting cho Wordpress, test connection.
7. Thử upload một số media (video, image) lên Wordpress, check xem có được sync qua S3 không?
8. Thử đăng 1 bài post với các media đã upload, test truy cập xem có link sang CloudFront không?

Copyright@Linh Nguyen on Udemy