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"

Load balancing and Autoscaling

Target

Tìm hiểu về mô hình hệ thống có Load Balancer (cân bằng tải).

Giới thiệu Elastic Load Balancer, các loại Load Balancer.

Tìm hiểu về EC2 Launch Template.

Auto Scaling Group là gì?

Kết hợp ELB & ASG để tạo 1 hệ thống đơn giản có thể scale & tự phục hồi.

Các option nâng cao của Load Balancer.

Tip and Trick for Auto Scaling Group

Load Balance

Load Balance là gì? Tại sao lại phải cần tới Load Balance?

Khái niệm “single point of failure”

Khi một sự cố xảy ra ở một thành phần nào đó có thể dẫn đến hệ thống bị dừng hoạt động, không thể phục vụ người dùng, ta gọi đó là single point of failure.

Ví dụ:

- Một chương trình bị lỗi và crash
- Database bị sập không response
- Hệ điều hành (OS) bị treo
- Một trong các phần cứng vật lý (RAM, CPU, Disk, Power,…) bị hỏng

Load Balance

Load Balance là gì? Tại sao lại phải cần tới Load Balance?

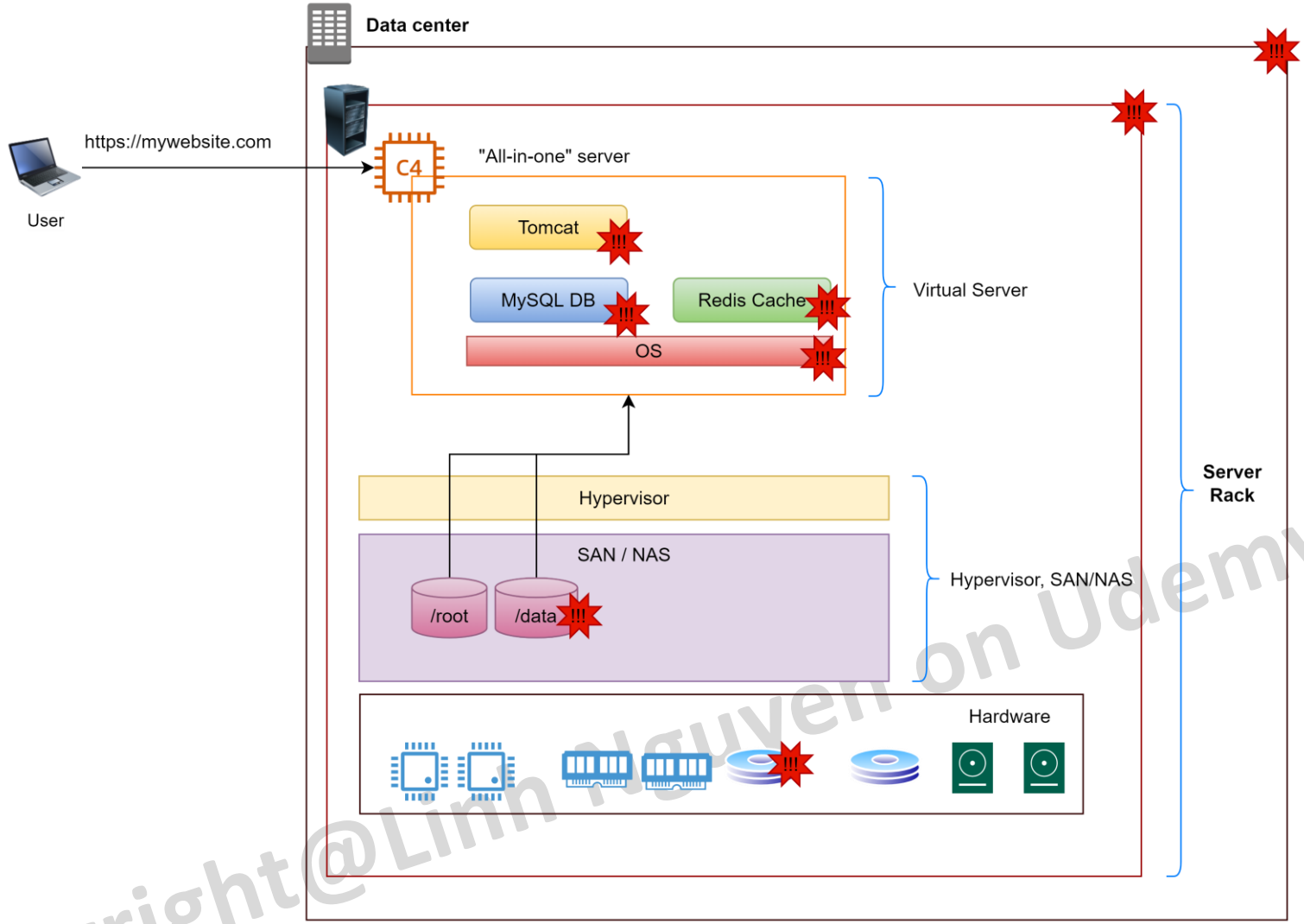
Lên cấp độ cao hơn chúng ta có các sự cố như:

- Bộ cung cấp nguồn cho cả 1 Server Rack bị sự cố.
- Xảy ra sự cố cúp điện cả data center (trong trường hợp không có nguồn dự phòng), bị lũ lụt, bão, đánh bom, thiên thạch rớt trúng, v.v..

=> Cần nhiều hơn 1 thành phần cho mỗi layer của hệ thống để tránh “single point of failure”.

Load Balance

Trong kiến trúc single point of failure như hình bên, bất kỳ sự cố hư hỏng ở 1 trong các cấp độ từ App, OS, Hypervisor(ảo hoá) cho tới hardware đều có thể ảnh hưởng tới ***Availability*** *(tính khả dụng) của hệ thống.*



Load Balance

Load Balance là gì? Tại sao lại phải cần tới Load Balance?

Nếu hệ thống có nhiều hơn 1 thành phần, cần có 1 cơ chế để phân phối request từ client đến các thành phần ở backend => **Sự ra đời của Load Balancer.**

AWS cho phép dễ dàng setup load balance tới nhiều target nằm ở các Availability Zone khác nhau.

Lưu ý: Mỗi Availability Zone (AZ) tương ứng với 1 data center cách nhau từ vài chục tới vài trăm km. Bằng việc phân bổ các instance nằm trên nhiều hơn 1 AZ, hệ thống có thể chịu được các sự cố ở cấp độ data center của AWS mà vẫn hoạt động bình thường.

Elastic Load Balancing

Elastic Load Balancing là gì?

⇒Một dịch vụ của AWS có nhiệm vụ điều hướng request từ client đến các target backend, đảm bảo request được cân bằng giữa các target.

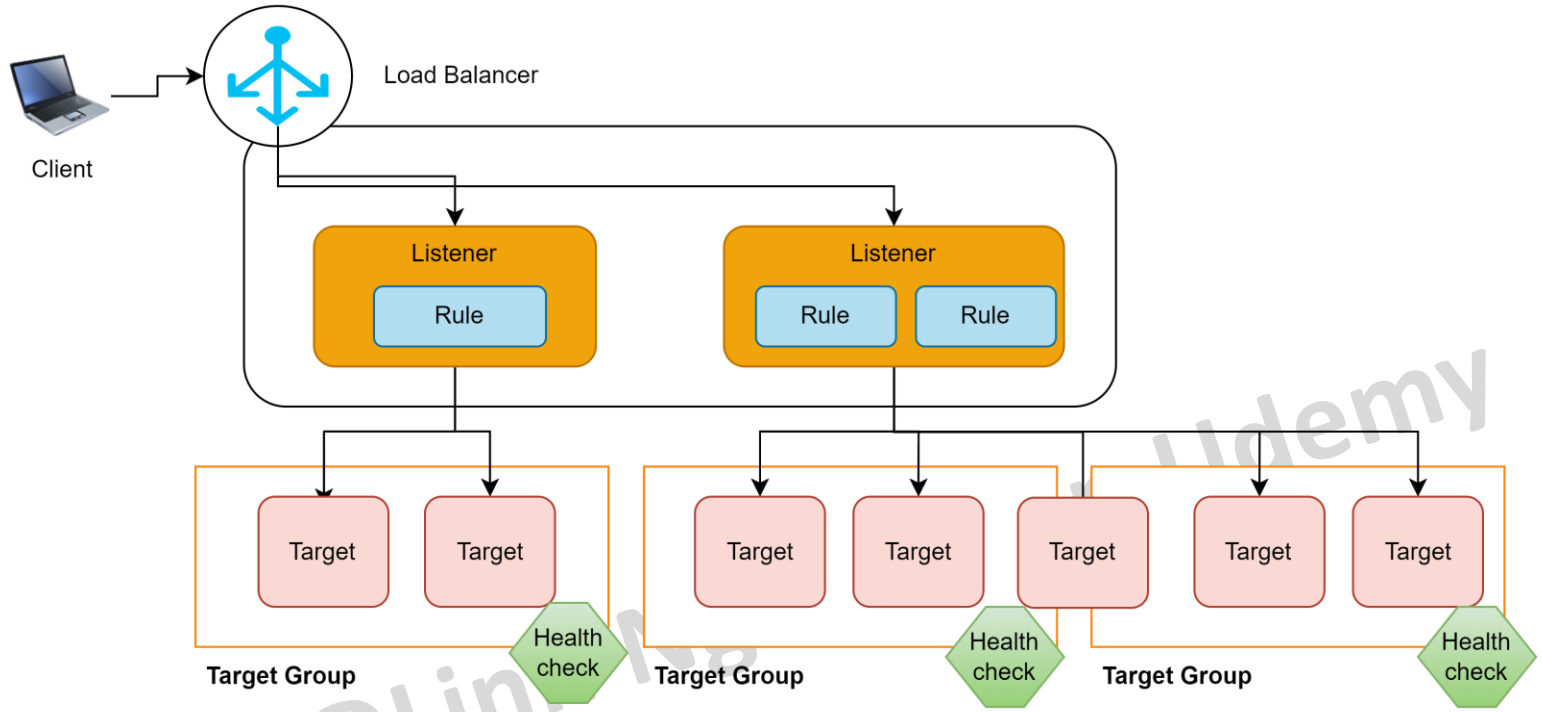
ELB là 1 dịch vụ managed hoàn toàn bởi AWS, dễ dàng setup, có đầy đủ các đặc tính cần thiết như:

- High Availability
- Scalability: về lý thuyết là không giới hạn
- High Security: nếu kết hợp với các dịch vụ khác như WAF, Security Group.

ELB có thể dễ dàng kết hợp với đa dạng backend sử dụng EC2, Container, Lambda.

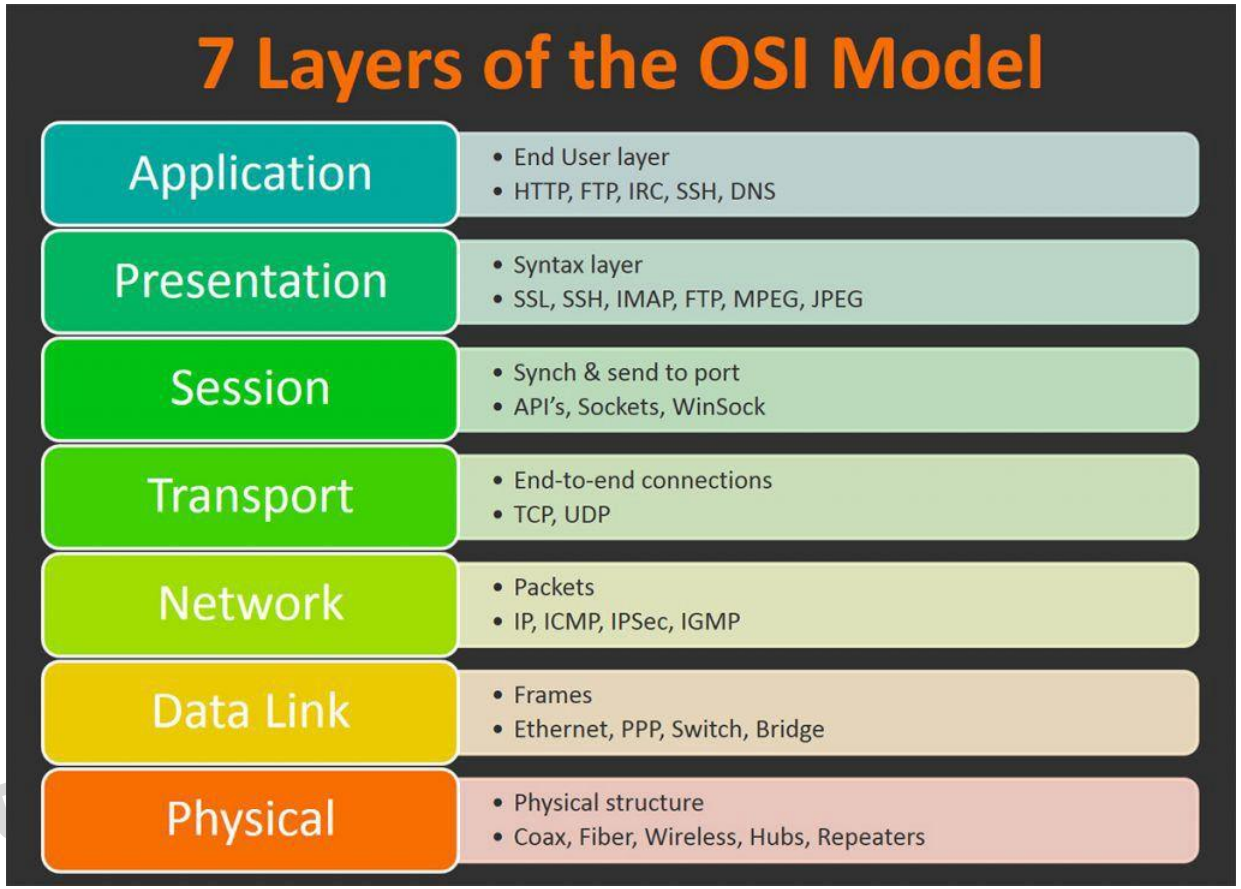
Các thành phần cơ bản của Load Balancer

- Load Balancer cho phép setting các listener (trên 1 port nào đó vd HTTP:80, HTTPS:443)
- Mỗi Listener cho phép cấu hình nhiều rule.
- Request sau khi đi vào listener, được đánh giá bởi các rule sẽ được forward tới target group phù hợp.
- Target group có nhiệm vụ health check để phát hiện và loại bỏ target un-healthy.



Type of Load Balancer

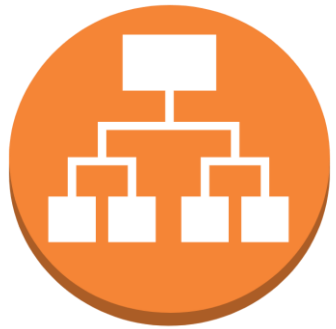
Các loại ELB được phân chia phụ thuộc vào việc nó hoạt động trên layer nào của mô hình OSI 7 layers.



Type of Load Balancer

Application Load Balancer

- Là loại Load Balancer thường dùng, phù hợp cho đa số các nhu cầu.
- Hoạt động trên layer 7 – Application.
- Do hoạt động trên layer 7 nên có một số ưu thể vượt trội so với các loại LB khác:
  - Hỗ trợ Path routing condition
  - Hỗ trợ host condition, cho phép dùng nhiều domain cùng trỏ vào 1 ALB
  - Hỗ trợ routing dựa trên thuộc tính của request (header, ip..)
  - Tích hợp được với Lambda, Container service
  - Hỗ trợ trả về custom HTTP response

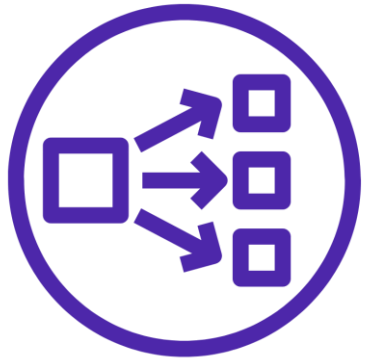


Application Load Balancer

Type of Load Balancer

Network Load Balancer

- Hoạt động trên layer 4 – Transport
- Hỗ trợ 2 giao thức TCP và UDP
- Không hỗ trợ nhiều hình thức rule routing.
- Thường dùng cho những hệ thống có workoad rất cao, lên tới hàng triệu request/s

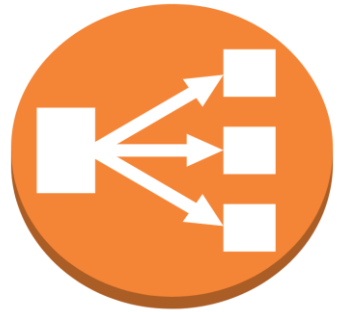


Type of Load Balancer

Classic Load Balancer

Sử dụng để điều phối traffic đi tới các Classic EC2 Instance.

*Hiện đã ngưng hỗ trợ EC2-Classic network từ 15/8/2022, do vậy để tạo ELB Loại này bạn phải có sẵn EC2 chạy dưới mode classic network.

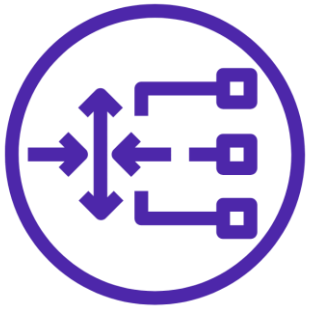


Classic Load Balancer

Type of Load Balancer

Gateway Load Balancer

- Giúp triển khai, scale và quản lý các Virtual Applicance (3rd party)
- Mục đích: Firewall, Phát hiện ngăn chặn xâm nhập (intrusion detection and prevention systems), kiểm tra gói tin chuyên sâu.
- Hoạt động trên layer 3 (Network) & Layer 4 (Transport).
- GLB listen trên tất cả các port và forward traffic đến các target group dựa trên các rule.
- GLB sử dụng GLB Endpoint để trao đổi traffic giữa VPC của service provider & VPC của consumer.
- Danh sách các provider có tại: https://aws.amazon.com/elasticloadbalancing/partners/



*Trong dự án thực tế thì đây là loại LB ít được sử dụng nhất.

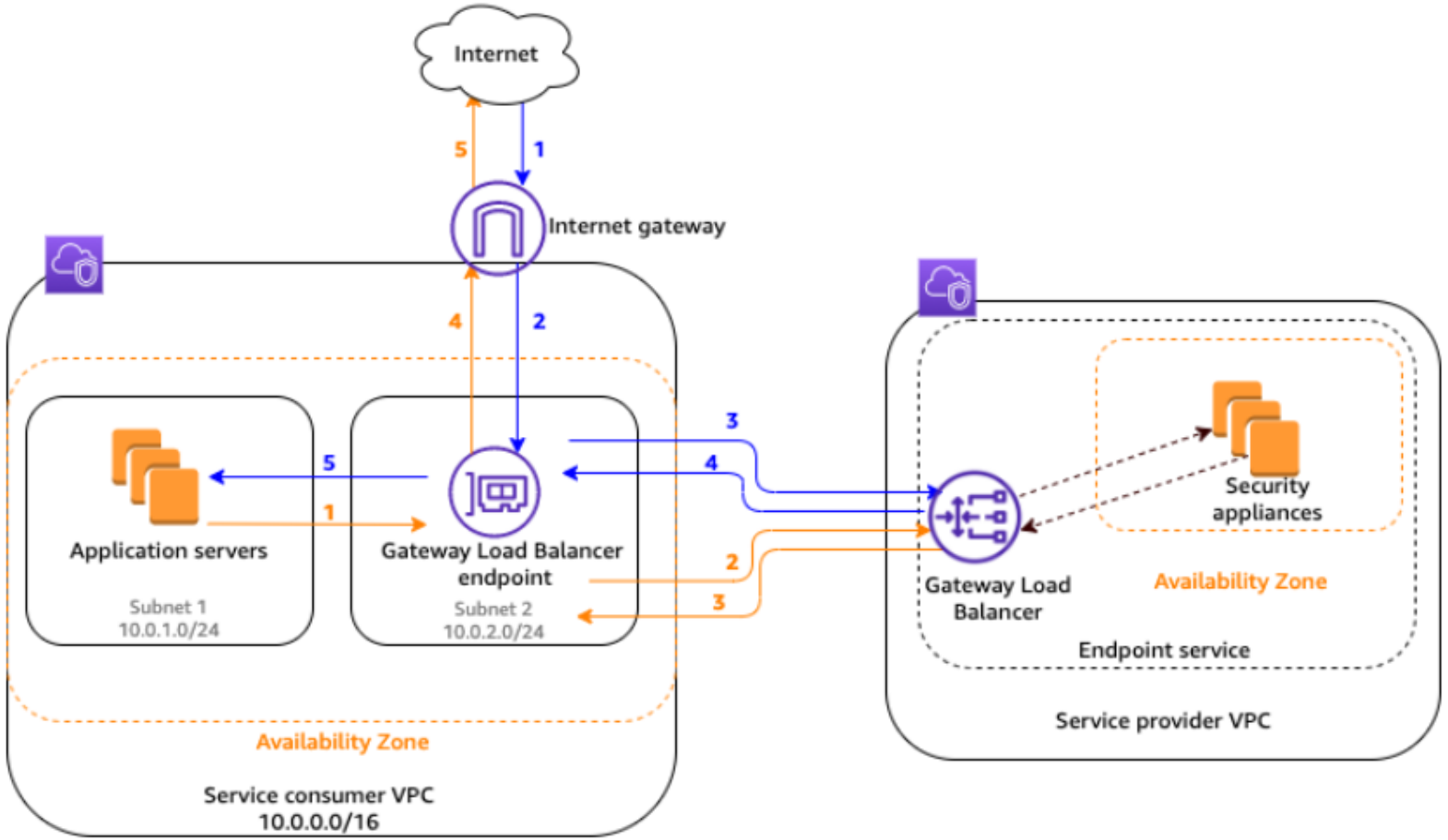
Type of Load Balancer

Gateway

Load Balancer

Example

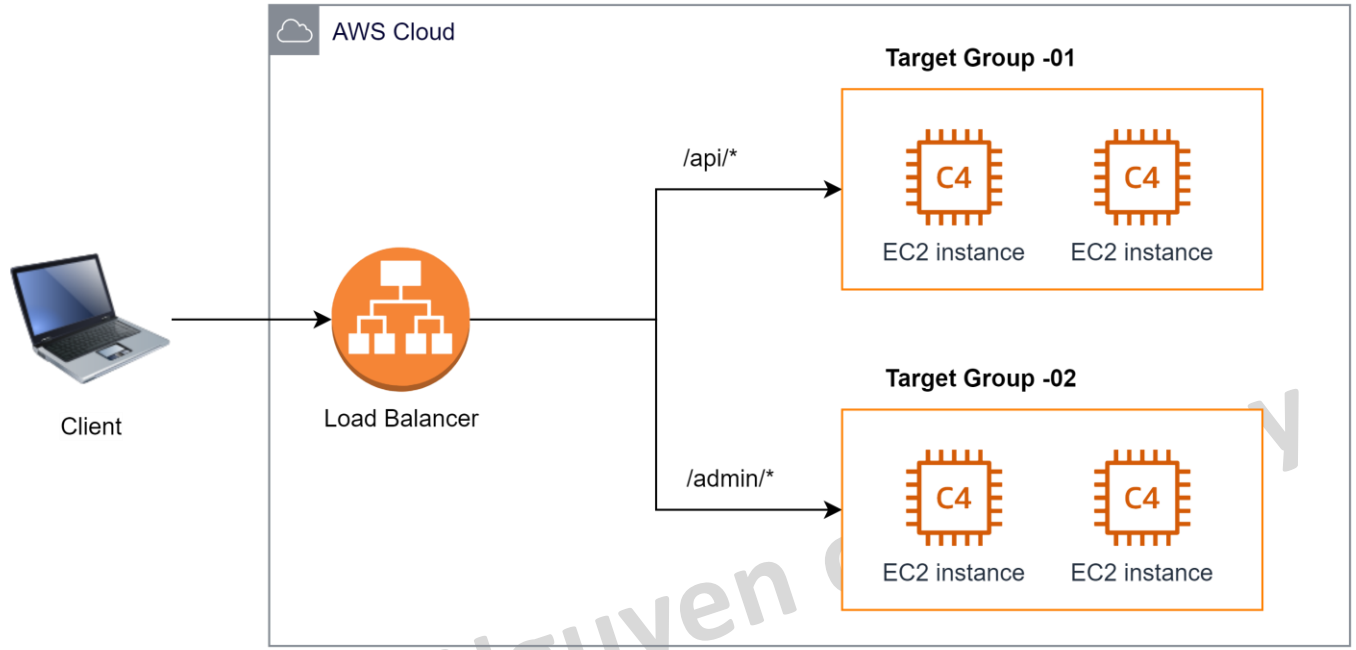
- Gateway LB được đặt bên trong VPC của Security Provider
- Traffic đi vào và đi ra hệ thống bên VPC của consumer được cấu hình routing để đi qua Gateway LB trước khi đến được target cần đến.
- Mũi tên màu xanh: traffic đi từ internet vào.
- Màu Cam: traffic từ bên trong đi ra.



Load Balancer hoạt động như thế nào?

Load Balancer có thể điều hướng tới nhiều hơn 1 target. Trong trường hợp multi-target, việc điều hướng tới target nào sẽ được quyết định bởi 1 số rule sau:

- Listener port
- Path pattern (Application LB only)
- Fixed ratio. VD Target Group 01 nhận 20%, Target Group 02 nhận 80% traffic.

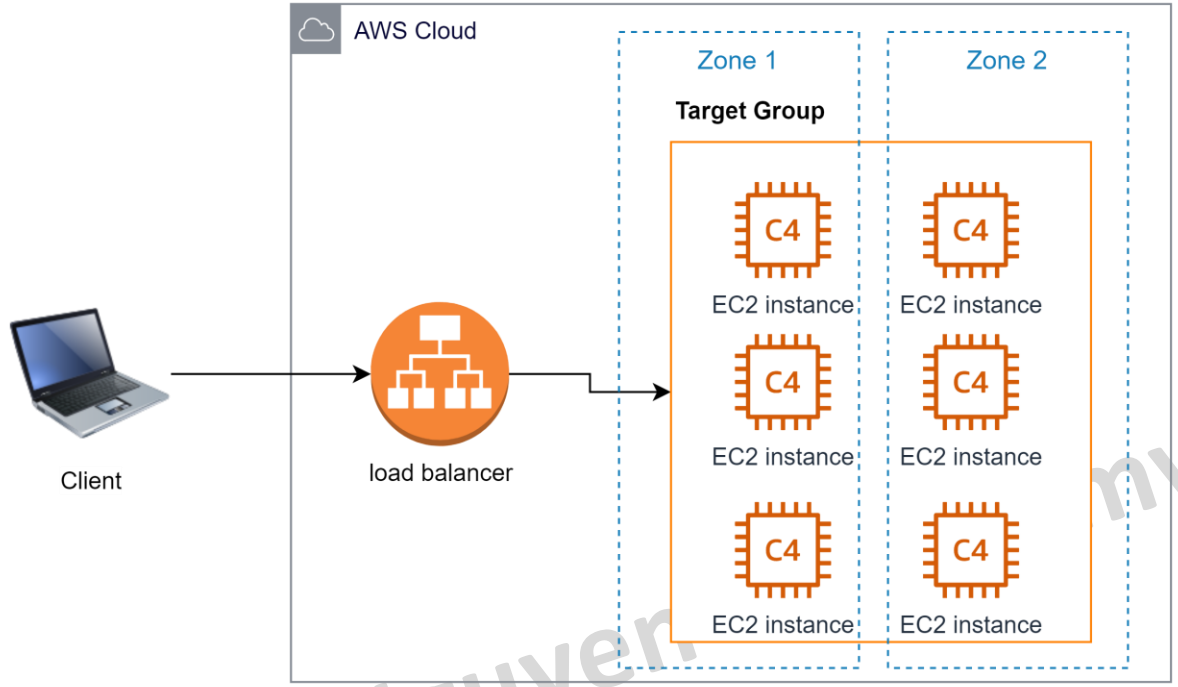


*Điều hướng tới 2 target làm nhiệm vụ khác nhau dựa vào path pattern của request đi tới.

Load Balancer hoạt động như thế nào?

Mặc định Load Balancer sẽ phân phối request từ client đến các target trong 1 Target Group theo tỷ lệ cân bằng (round robin), kể cả khi target đó nằm trong nhiều hơn 1 target group.

VD: trong hình bên mỗi instance sẽ nhận ~16.7% số request từ client.



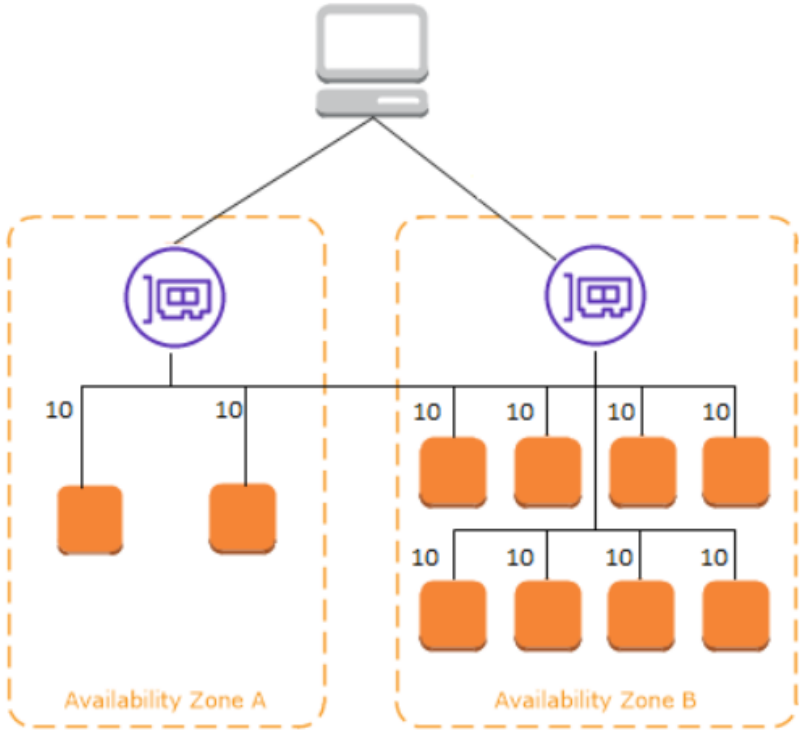
*Lưu ý các instance có thể nằm ở các zone khác nhau

Cross zone load balancer

ELB là 1 dịch vụ hoạt động cross zone (AWS suggest chọn tất cả các zone có thể khi khởi tạo ELB).

Nếu Cross zone load balance được enable, ELB sẽ điều hướng request từ client một cách cân bằng tới các target.

VD: Hình bên phải khi cross zone LB được bật, mỗi instance nhận 10% traffic mặc dù có sự chênh lệch phân bổ số lượng instances giữa 2 zone.

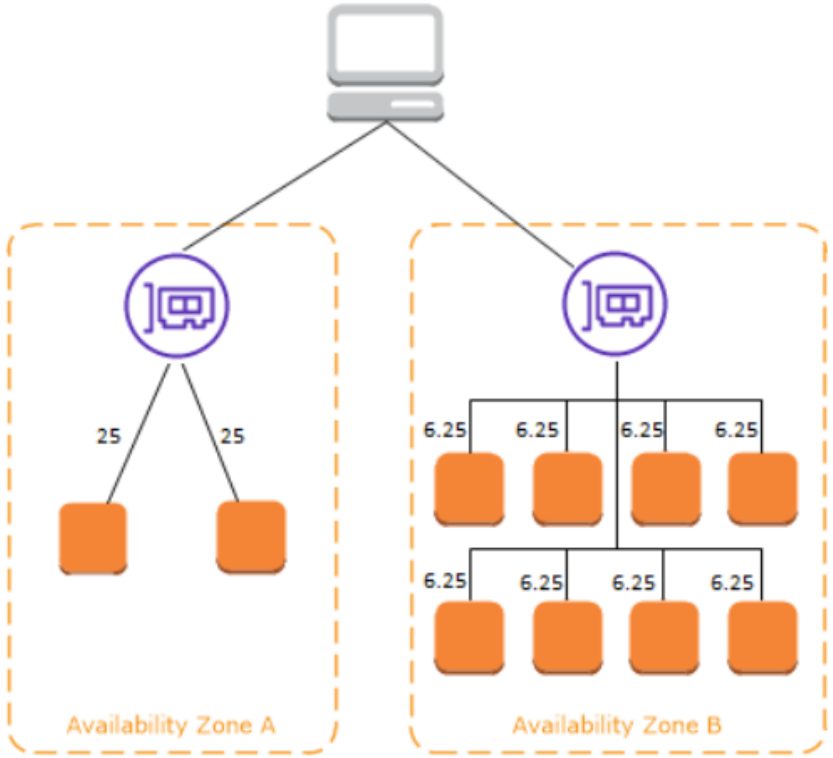


Phân bố request tới target trong trường hợp Cross zone enabled

Cross zone load balancer

Nếu Cross zone load balance được disable, ELB sẽ điều hướng request từ client một cách cân bằng tới mỗi zone, trong 1 zone sẽ chia đều tiếp cho các instances.

VD: Hình bên phải khi cross zone LB được tắt, mỗi instance trong zone A nhận 25% traffic trong khi mỗi instance trong zone B nhận 6.25% traffic.



Phân bố request tới target trong trường hợp Cross zone disabled

Cross zone load balancer

NOTE:

- By default, Application Load Balancer sẽ Enable cross zone, không thể tắt.
- By default, Network Load Balancer sẽ Disable cross zone. Cần enable sau khi tạo.

Load Balancer tính phí như thế nào?

- Mặc định Elastic Load Balancer tính tiền theo giờ ($/hour), giá phụ thuộc vào region.
- Ngoài ra còn tính phí dựa trên số lượng request, lượng data transfer qua Load Balancer quy đổi ra Load Balancer Capacity Units (LCUs)
- Khi estimate cho Load Balancer, user sẽ input các thông số như số lượng request per second (RPS), lưu lượng data (GB/TB per hour), số lượng connection new, thời gian trung bình cho một connection, số lượng rule. AWS sẽ quy đổi thành LCUs để ước tính chi phí.

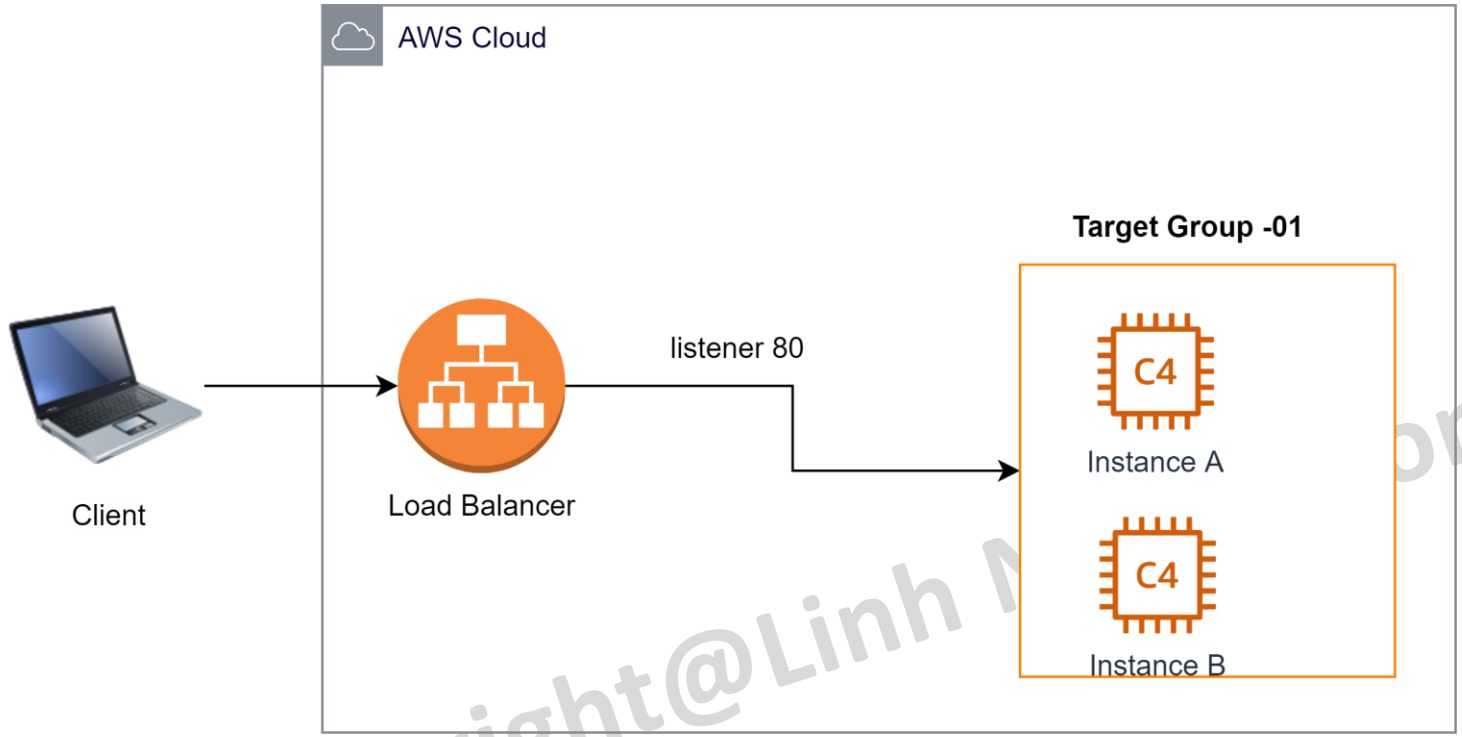
Lab1: Tạo 2 EC2, cấu hình loadbalancer

Login to AWS console, thực hiện nội dung sau:

1. Tạo 2 instance A và B, trong quá trình tạo sử dụng userdata đã dc chuẩn bị sẵn, mục đích là để có sự khác biệt về GUI khi truy cập 2 instances.
2. Tạo 1 target group tg-01, register 2 instance ở step trước.
3. Tạo 1 Application Load Balancer (ALB), cấu hình listener port 80 trỏ vào tg-01
4. Cấu hình Security Group
5. Truy cập ALB thông qua DNS link.

Lab1: Tạo 2 EC2, cấu hình loadbalancer

Lab1 part1 (Step 1-4)

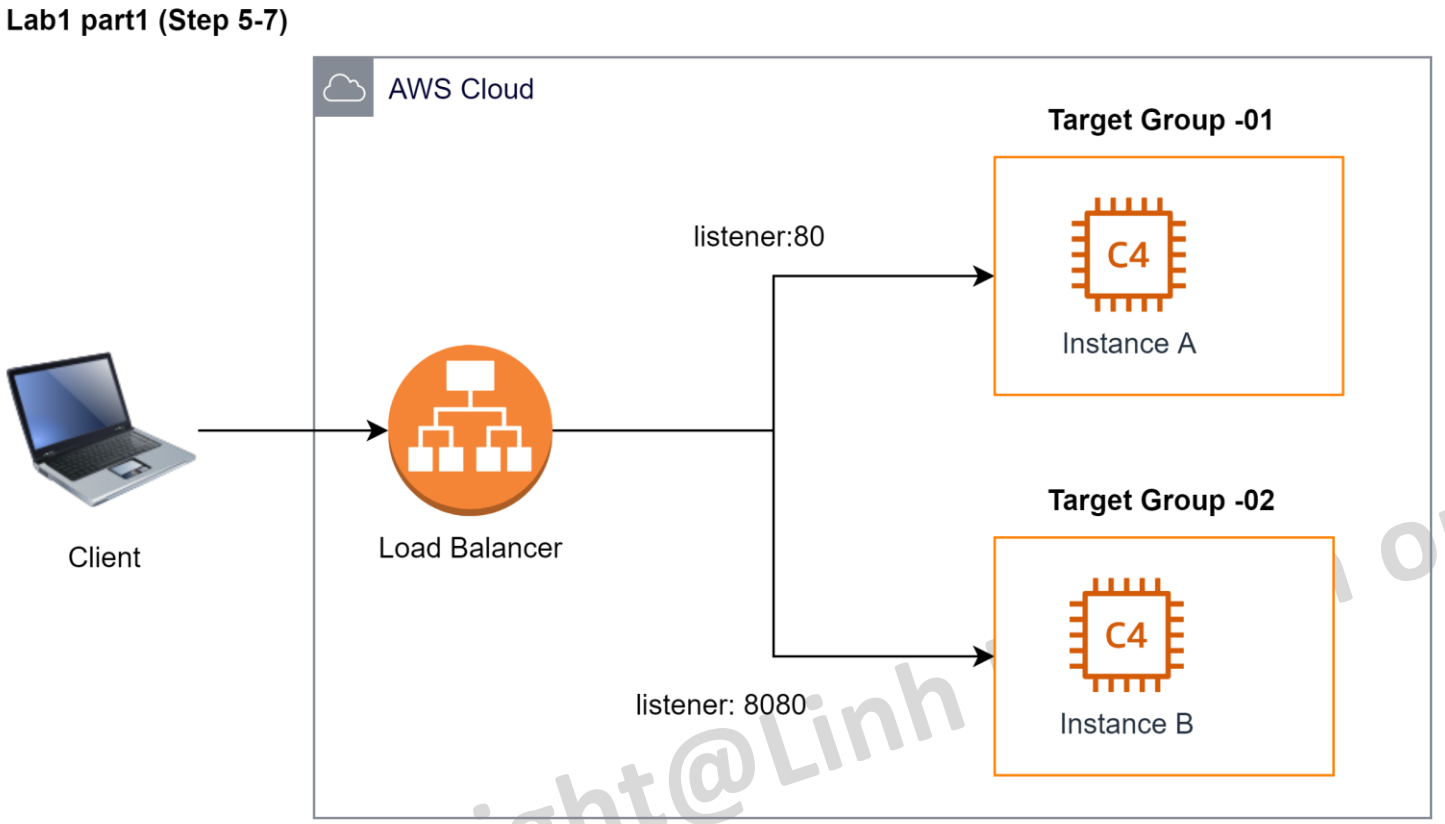


Lab1: Tạo 2 EC2, cấu hình loadbalancer

Login to AWS console, thực hiện nội dung sau:

5. Remove instance B khỏi tg-01. tạo 1 target group mới tg-02 và add instance B vào đó.
6. Tạo thêm 1 listener port 8080, trỏ tới tg-02.
7. Truy cập ALB thông qua DNS link với 2 port 80 và 8080
8. [Optional] Thiết lập tỷ lệ 1:3 cho traffic đến 2 target group tg-01 và tg-02 trên cùng 1 listener port 80, thử truy cập xem ALB có điều hướng tới 2 targets theo đúng tỷ lệ không?

Lab1: Tạo 2 EC2, cấu hình loadbalancer



Một số lưu ý về Load Balancer

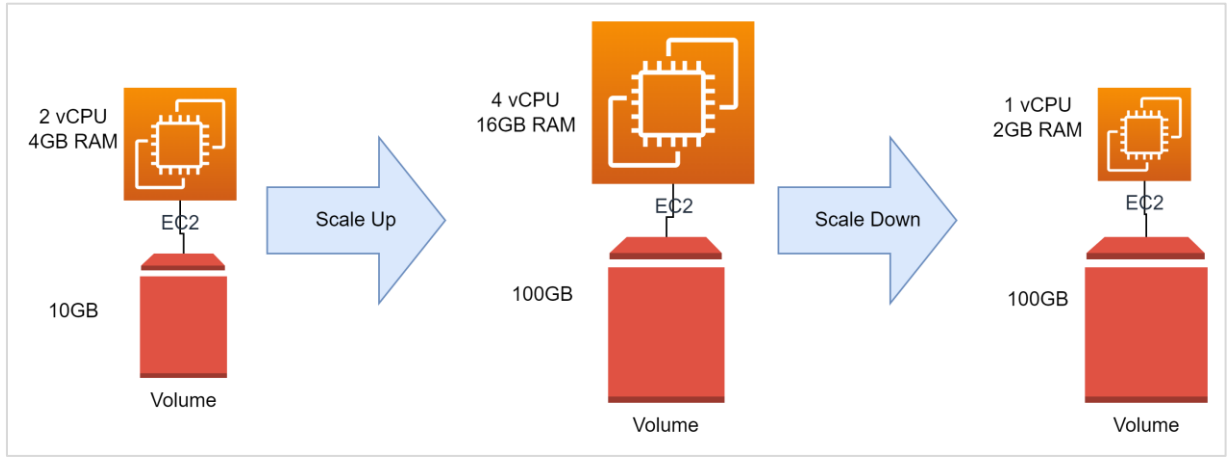
- Load Balancer là 1 dịch vụ Cross Zone, lưu ý khi tạo ELB nhớ chọn tối đa số zone có thể chọn.
- Nếu Load Balancer được tạo không chọn zone có chứa ec2 instance, khi access sẽ bị lỗi không kết nối được (502 Bad Gateway).

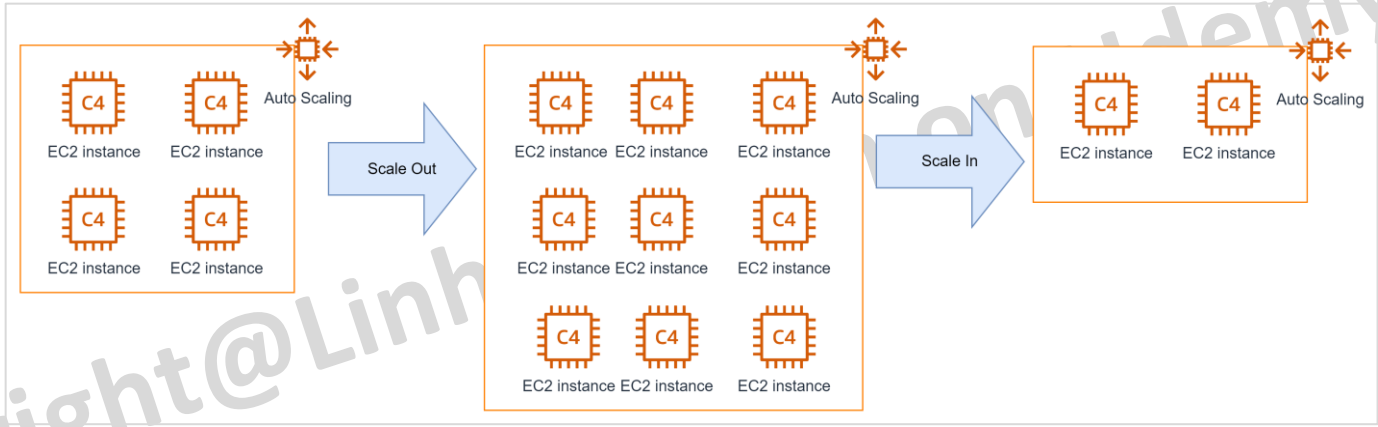
Scaling là gì?

Là việc điều chỉnh cấu hình của các tài nguyên để đáp ứng với nhu cầu workload (số request từ người dung, số lượng công việc phải xử lý,…)

Có 2 hình thức scale:

- Scale Up/Down: Tăng/Giảm cấu hình của resources (vd tăng CPU/Ram cho Server, database, tăng dung lượng ổ cứng,…)
- Scale Out/In: Tăng/giảm số lượng thành phần trong 1 cụm chức năng. (Vd add thêm server vào cụm application, add thêm node vào k8s cluster,…)



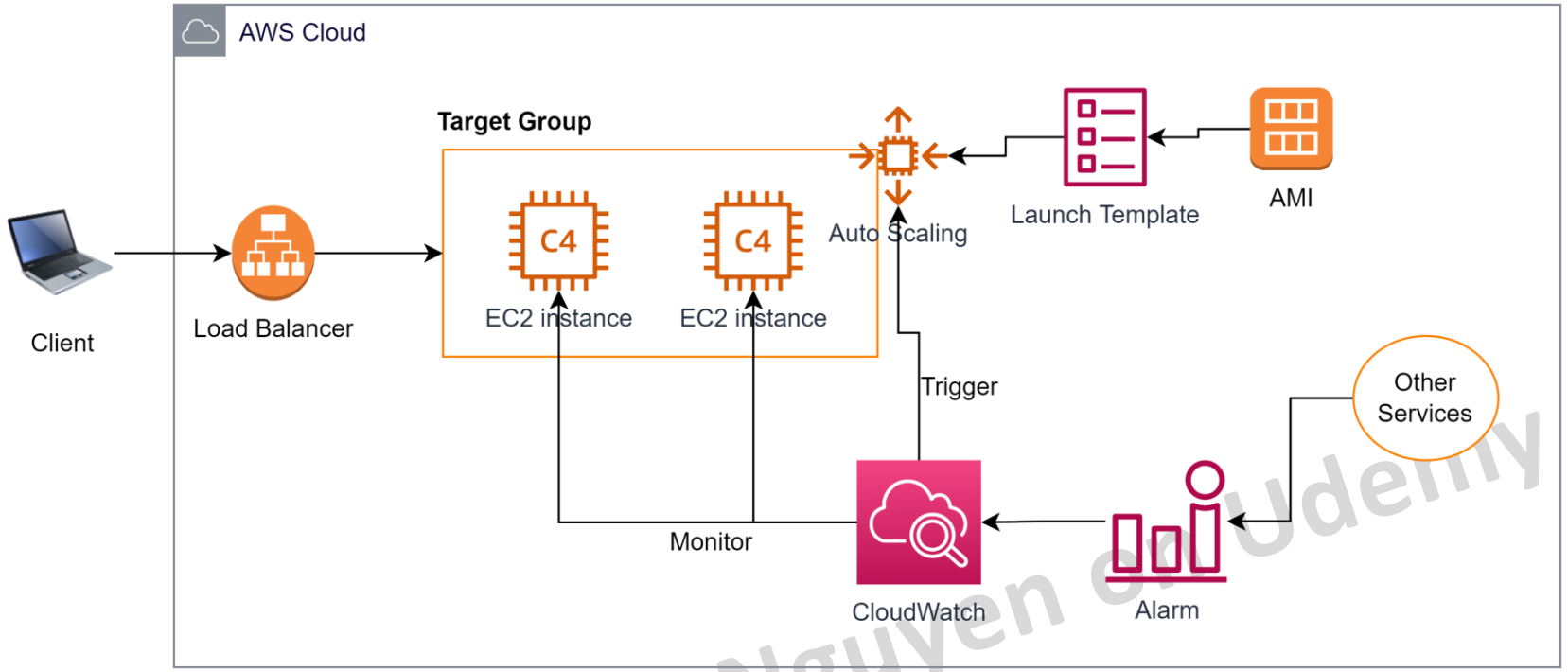


Auto Scaling Group

Có nhiệm vụ điều chỉnh số lượng của instance cho phù hợp với workload.
Mục đích:

- Tiết kiệm chi phí
- Tự động hóa việc mở rộng & phục hồi sự cố.

ASG Sử dụng Launch Template để biết được cần phải launch EC2 như thế nào (next slide)
Để thực hiện được việc scale, Auto Scaling Group phải kết hợp với việc monitor các thông số của các thành phần trong hệ thống để biết được khi nào cần scale-out, khi nào cần scale-in
*=> Sự cần thiết của **CloudWatch***



Auto Scaling Group

Launch Configuration and Launch Template

Mục đích: Chỉ dẫn cho Auto Scaling Group biết được cần phải launch instance như thế nào.
Các thông tin có thể định nghĩa trong launch template:

- AMI
- Instance Type
- Keypair (trong trường hợp bạn cần login vào instance sau khi tạo)
- Subnet (Thường không chọn mà để Auto Scaling Group quyết định)
- Security Group(s)
- Volume(s)
- Tag(s)
- Userdata (script tự động chạy khi instance start)

…

**Một số thông tin như Instance Type, Subnet, Security Group có thể được overwrite bởi Auto Scaling Group.*

**Launch Template thường được sử dụng hơn bởi nó có thể quản lý được version.*

Các phương pháp scale hệ thống

Có các option sau để scale một Auto Scaling Group

- **No Scale**: Duy trì 1 số lượng cố định instances (nếu instance die thì tạo con mới để bổ sung, ngoài ra không làm gì cả)
- **Manually Scaling**: điều chỉnh 3 thông số: min/max/desire để quyết định số lượng instance trong ASG.
- **Dynamic Scaling**: Scale tự động dựa trên việc monitor các thông số.
  - Target tracking scaling: Monitor thông số ngay trên chính cluster, vd CPU, Memory, Network in-out.
  - Step scaling: điều chỉnh số lượng instance (tăng/giảm) dựa trên 1 tập hợp các alarm (có thể đến từ các resource khác không phải bản thân cluster).
  - Simple scaling: Tương tự Step scaling tuy nhiên có apply “cool down period”

Các phương pháp scale hệ thống

- **Schedule Scaling**: Đặt lịch để tự động tăng giảm số instance theo thời gian, phù hợp với các hệ thống có workload tăng vào 1 thời điểm cố định trong ngày.
- **Predict Scaling**: AWS đưa ra dự đoán dựa vào việc học từ thông số hằng ngày, hằng tuần để điều chỉnh số lượng instance một cách tự động. Độ chính xác phụ thuộc vào thời gian application đã vận hành và tính ổn định của traffic đi vào hệ thống.

Auto Scaling Group

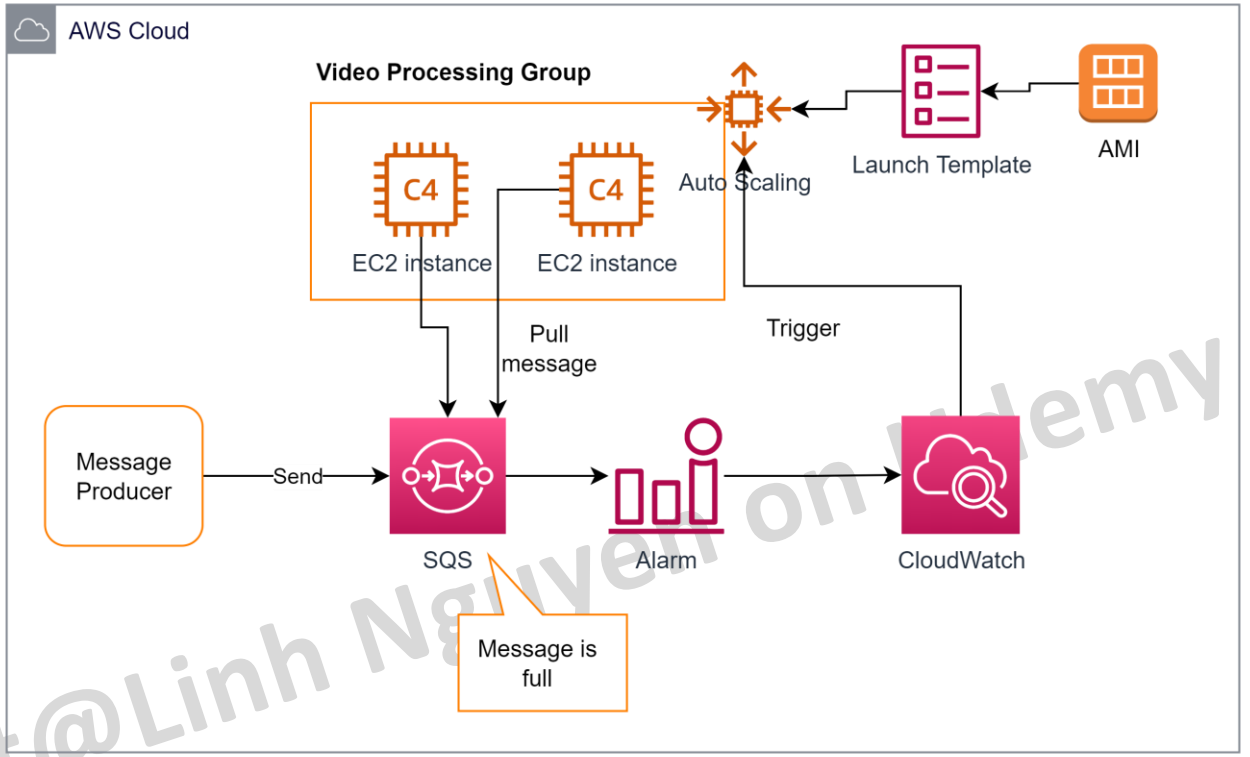
Ví dụ về trường hợp scale sử dụng metrics đến từ bên ngoài cluster

Hình bên mô tả 1 cụm server (cluster) có nhiệm vụ xử lý video encoding.

Danh sách video được lấy từ SQS (một dịch vụ message queue sẽ được trình bày sau).

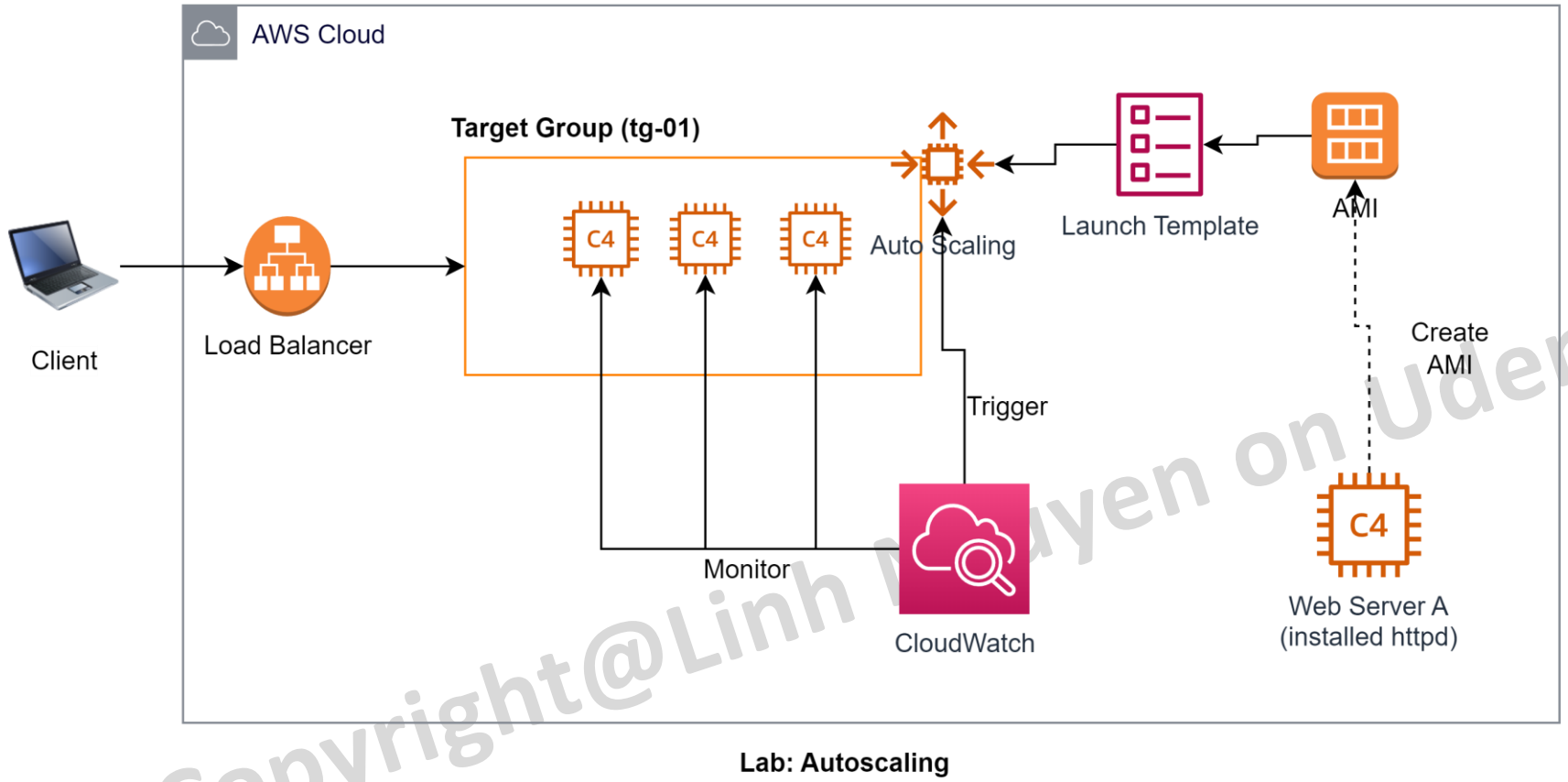
Message producer (1 process nào đó) có nhiệm vụ đăng ký video cần xử lý vào queue.

Nếu số lượng message trên queue quá nhiều hoặc message được ghi vào queue nhưng quá lâu không được xử lý xong, ta có thể monitor queue để ra quyết định có scale-out (add instance vào cụm cluster) hay không.



Lab2: Tạo hệ thống có Auto Scaling

Sơ đồ hệ thống của bài lab



Lab: Autoscaling

Lab2: Tạo hệ thống có Auto Scaling

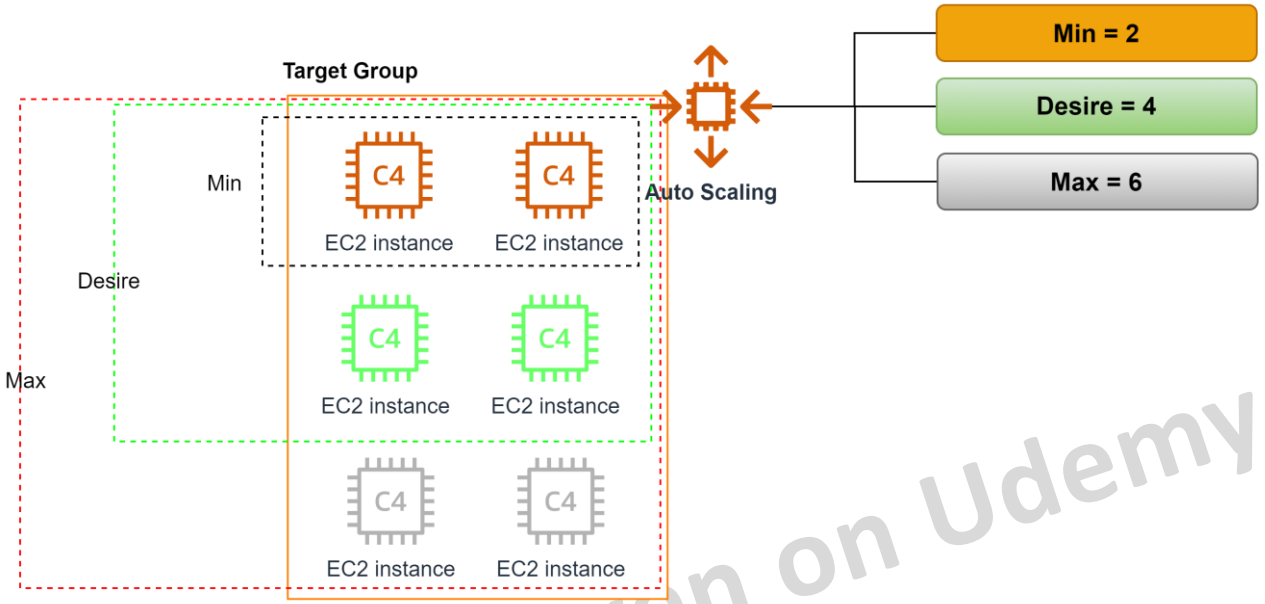
Login to AWS console, thực hiện nội dung sau:

1. Enable auto start for service **httpd**
2. Tạo AMI từ 1 instance đang chạy.
3. Tạo Launch Template
4. Tạo Auto Scaling Group, chọn target group cho ASG là tg-01
5. Cấu hình Application Load Balancer trỏ vào tg-01
6. Kiểm tra số lượng instance tạo ra có phù hợp chưa.
7. Thử terminate 1 instance xem ASG có tự tạo lại instance khác để bổ sung không?
8. Điều chỉnh số lượng instance trong ASG (tăng hoặc giảm size min + desire capacity)
9. Thử access liên tục xem ASG có tự add thêm instance không? (trước đó có cấu hình bật auto-scale).
10. Thử setting Schedule Auto Scaling (Lưu ý: chọn thời gian gần để test)

Auto Scaling Group

Hiểu rõ về 3 thông số Min, Max, Desire capacity
Về bản chất ASG nhìn vào thông số Desire capacity để biết được cần thêm hay bớt instance trong cluster.
Vd:
Cluster đang có 2 instances, set desire = 4 => ASG sẽ add thêm 2 instance.
Cluster đang có 4 instances, set desire = 3 => ASG sẽ terminate bớt 1 instance

Min: Sau khi số lượng instance bằng với min, ASG sẽ không terminate bớt instance vì bất cứ lý do gì.
Max: Sau khi số lượng instance bằng với max, ASG sẽ không add thêm instance vì bất cứ lý do gì.



Elastic Load Balancer stickiness session

- Cho phép điều hướng một client cụ thể tới target cố định trong một khoảng thời gian.
- Phù hợp cho các website sử dụng công nghệ cũ quản lý session của user trên RAM.

Clear resources

Login to AWS console, thực hiện nội dung sau:

1. Xóa Load Balancer
2. Xóa Auto Scaling Group (Hoặc chỉnh size về 0)
3. Terminate hết EC2 instances nếu còn sót lại.
4. Delete EBS Volume nếu còn sót lại.
5. Xoá (Deregister) AMI nếu không dùng tới.
6. Xoá hết các snapshot còn lại nếu không dùng tới.