EC2 Management

CodeStar Academy

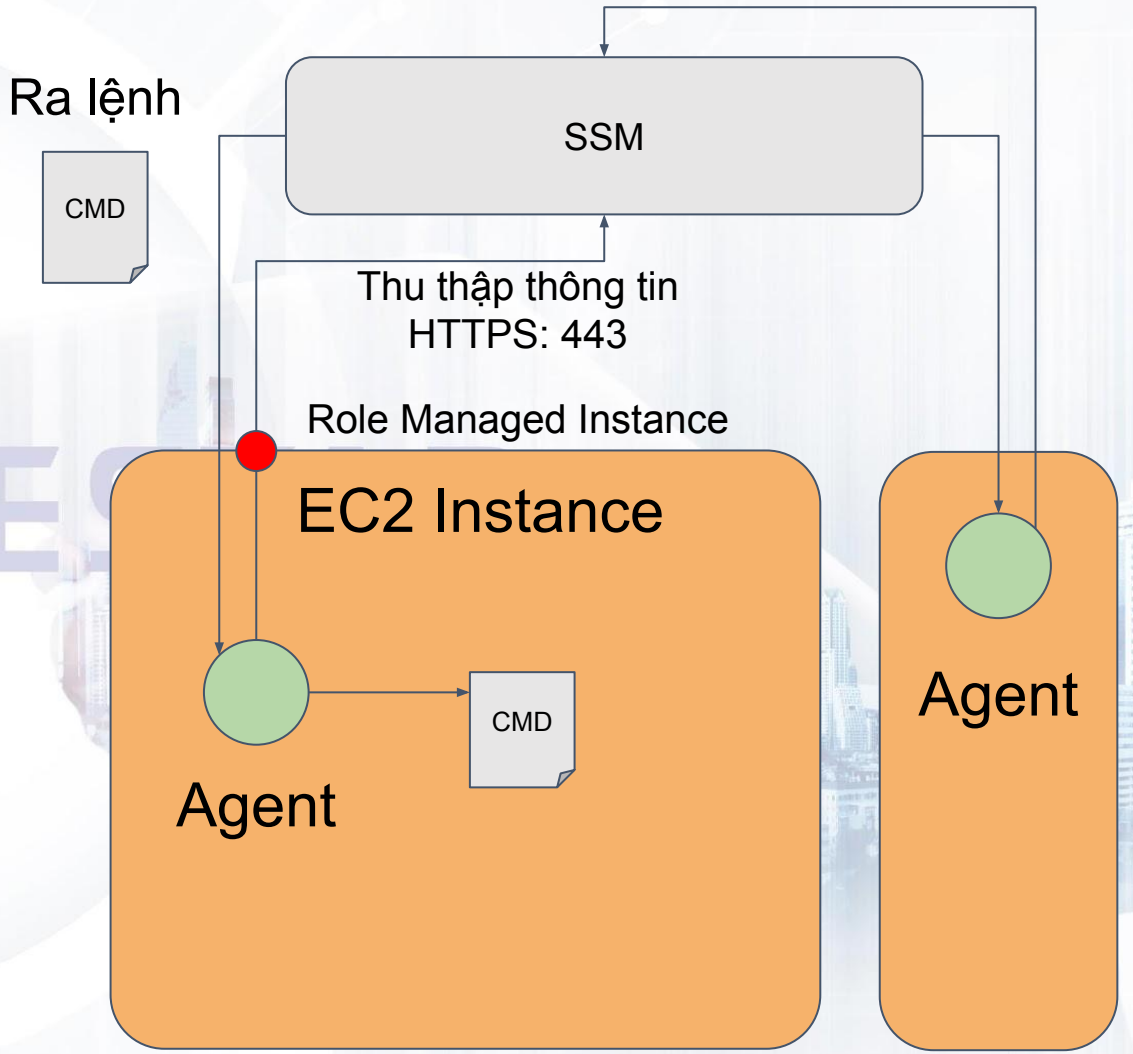
EC2 Management

- **System Manager**

Là một thành phần được cài đặt lên trên EC2 Instance để System Manager có thể quản lý và phục vụ cho các task khác.

Use case:

Chạy lệnh cho toàn bộ các EC2 Instance

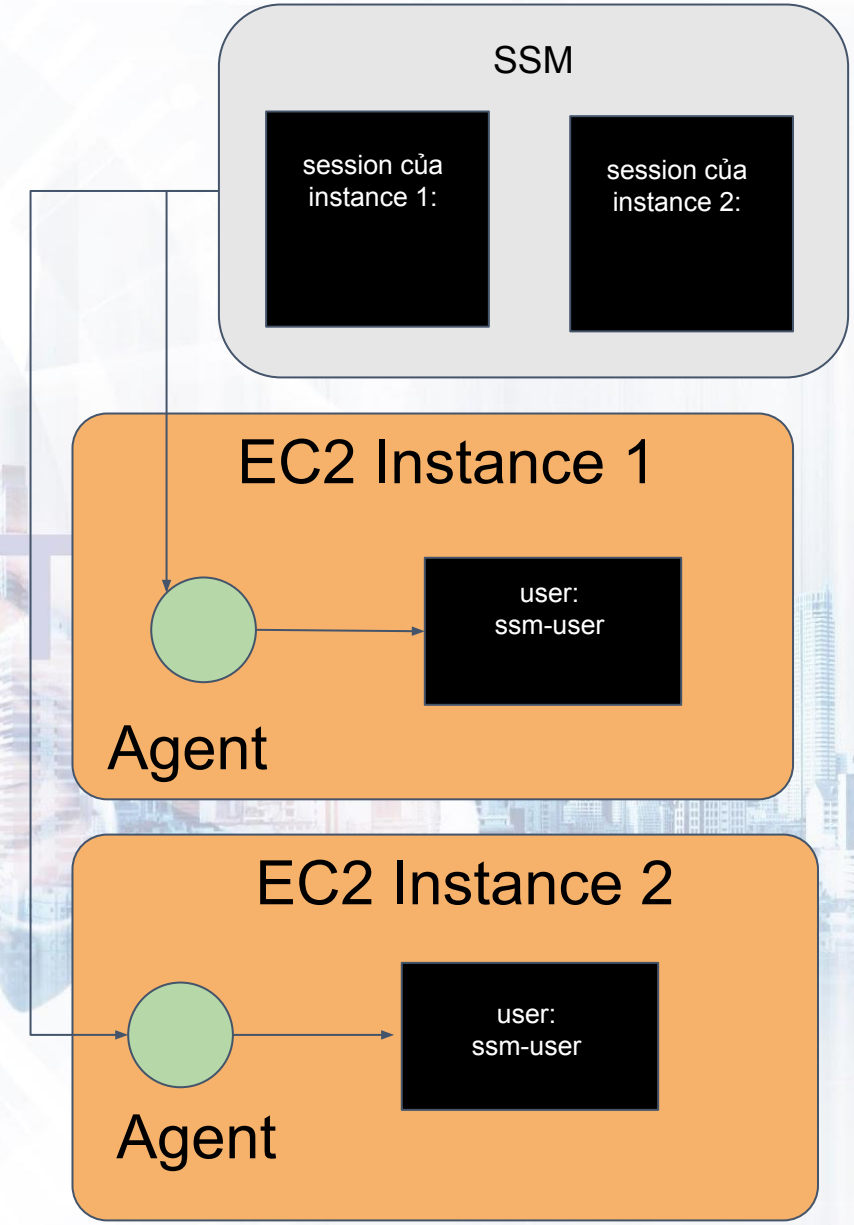


EC2 Management

- **System Manager**

user tại Session khi thực hiện trên các môi trường là **ssm-user**.

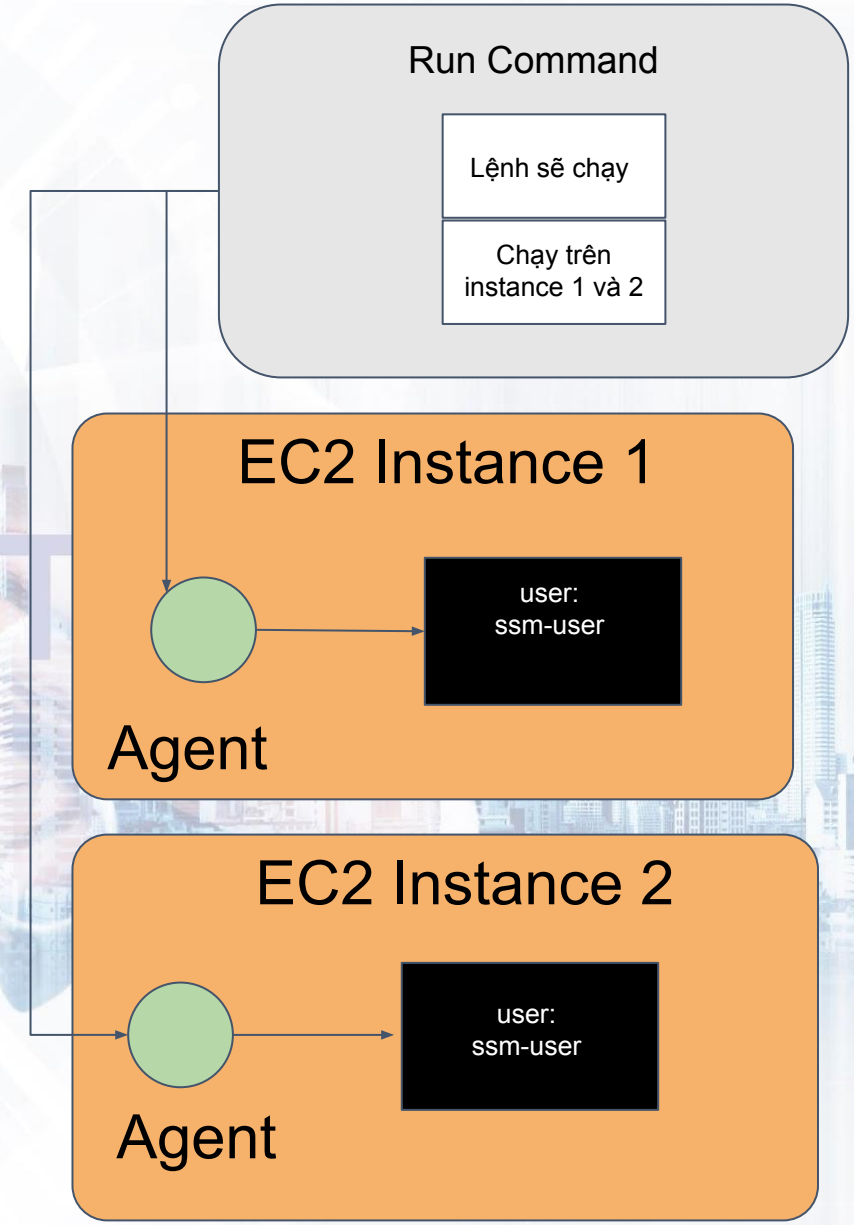
ssm-user được khởi tạo vào lần đầu tiên truy cập session vào EC2 Instance.



EC2 Management

- **System Manager Run Command**

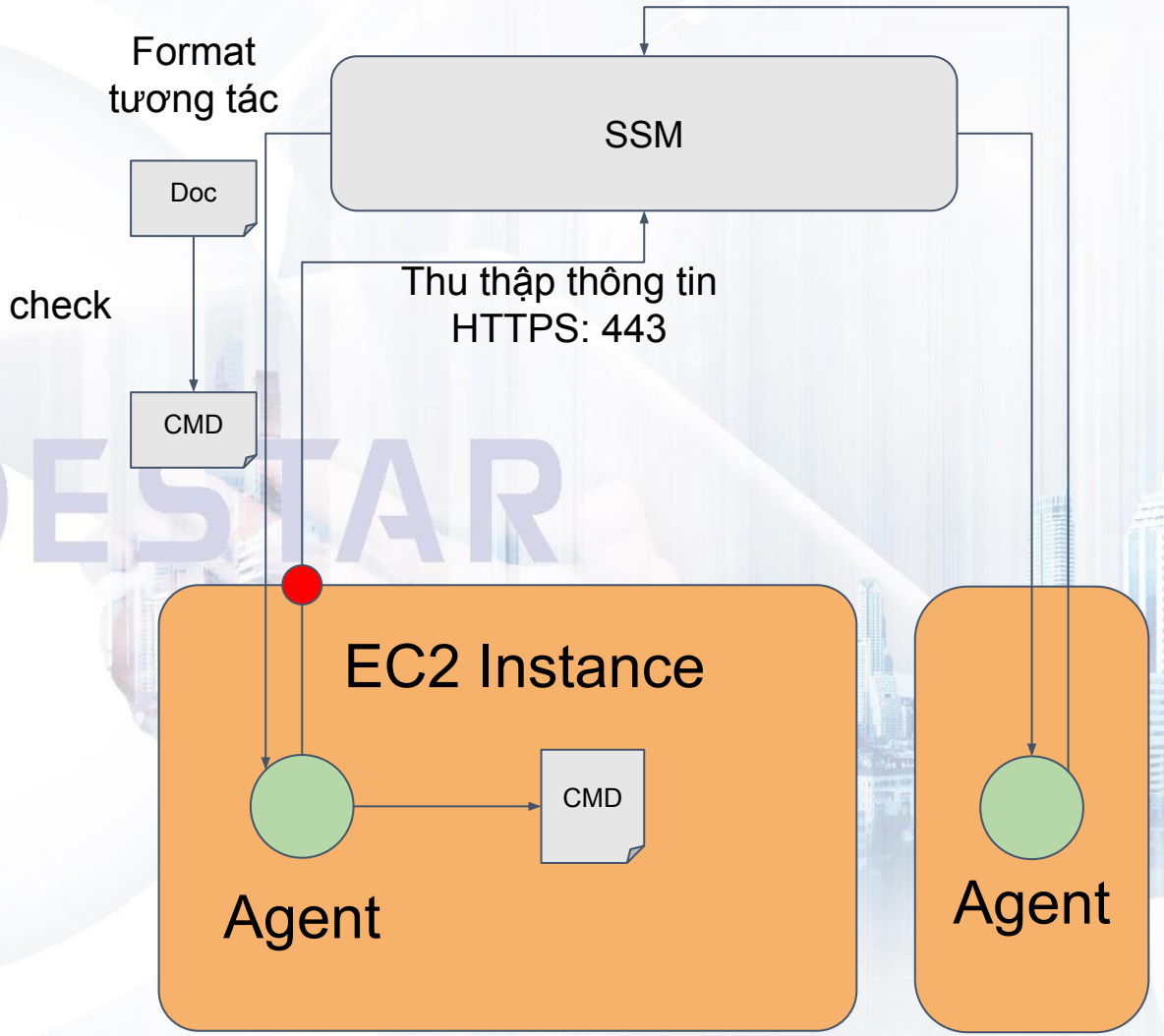
Run Command có thể áp dụng cho nhiều EC2 instance tùy theo tag/resource group hoặc instance id cụ thể.



EC2 Management

Document

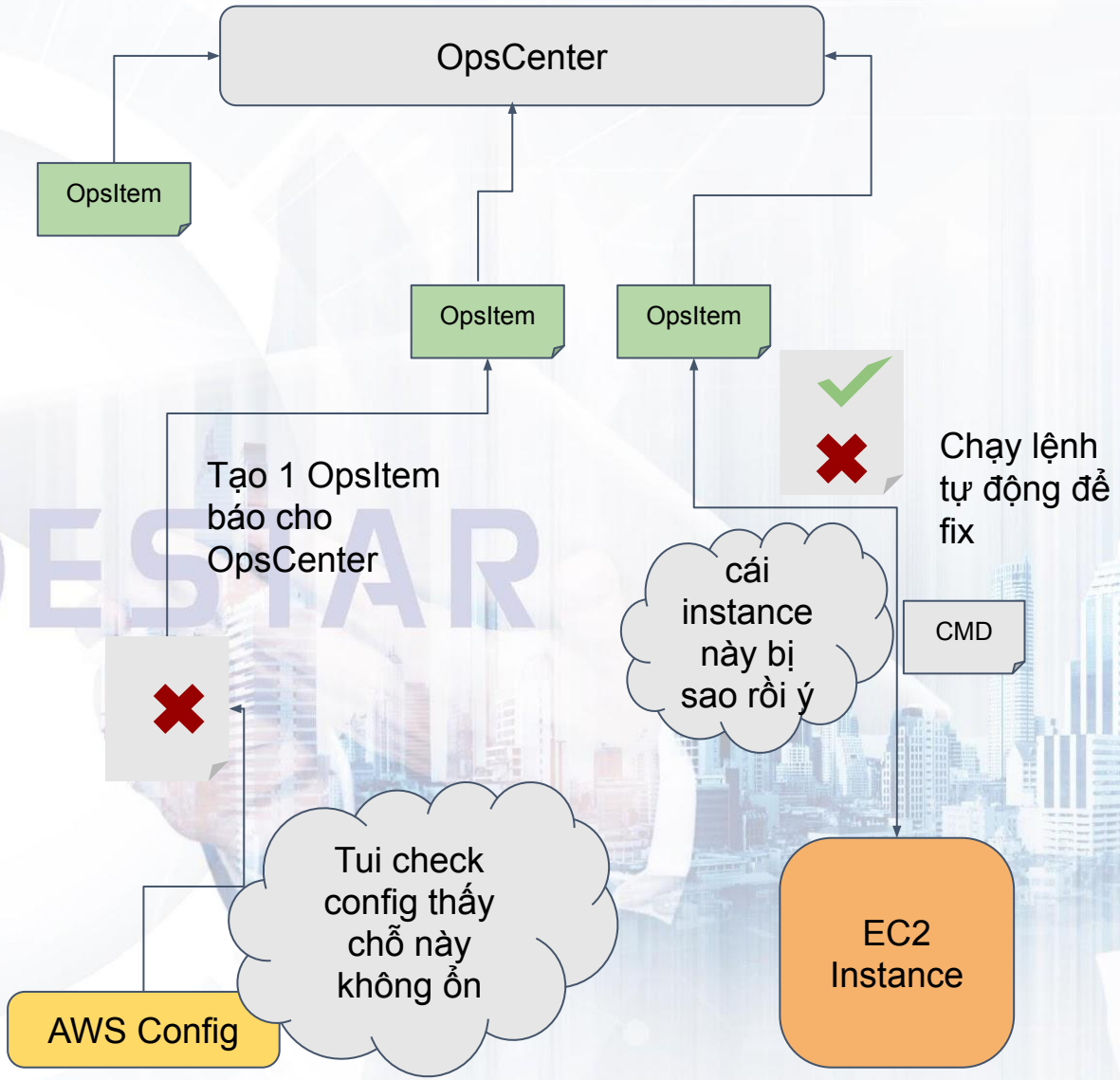
Document là một khái niệm đặc biệt cho 1 số dịch vụ, quy định cách thức và giao diện tương tác với EC2 Instance như ở dạng Terminal, giao diện, dòng lệnh CLI, …



EC2 Management

Operation Management:

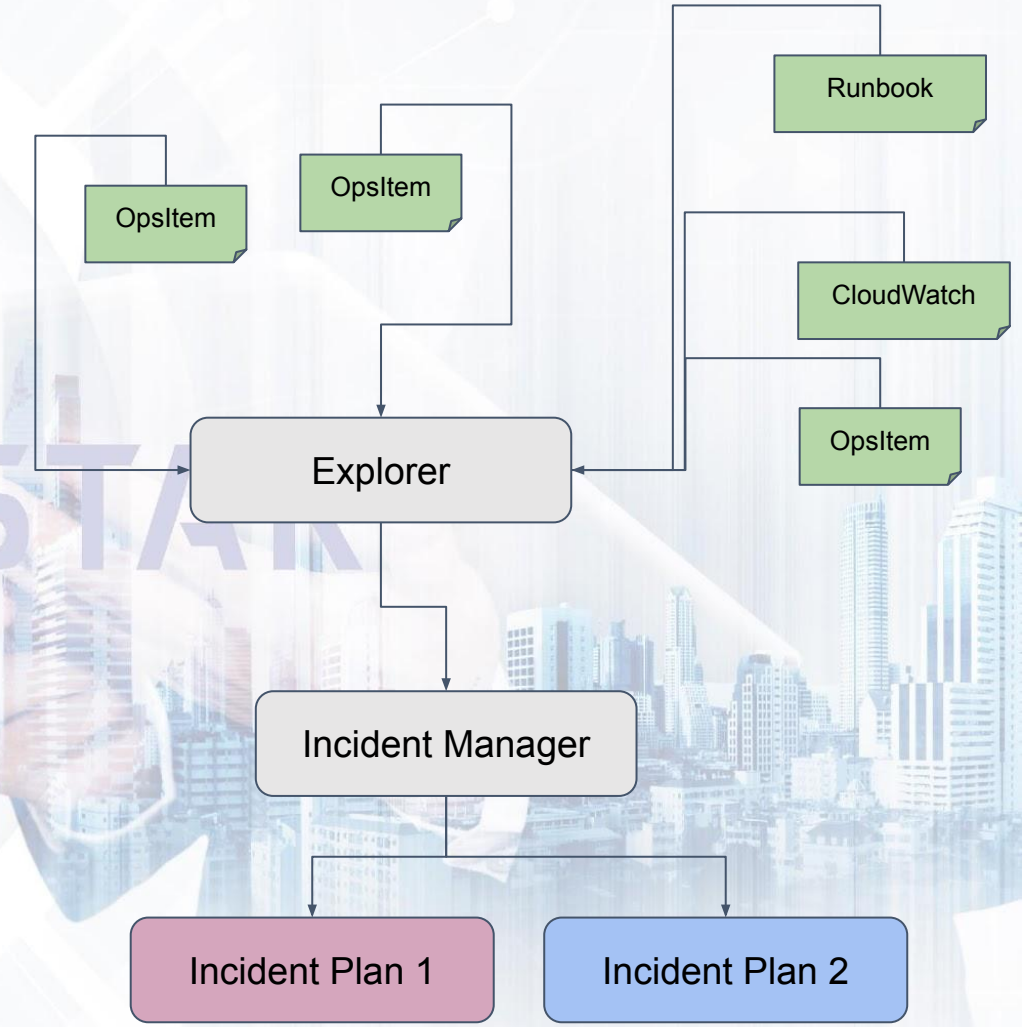
- **OpsCenter**: Cho phép theo dõi thông tin các vấn đề cần xử lý bằng OpsItem một cách tập trung (có thể theo dõi trên nhiều EC2, trên nhiều Region, của nhiều Account khác nhau)
- **OpsItem:** Một thành phần đang có vấn đề cần giải quyết.
- **Runbook:** Các thao tác tự động để giải quyết vấn đề của OpsItem.



EC2 Management

Operation Management:

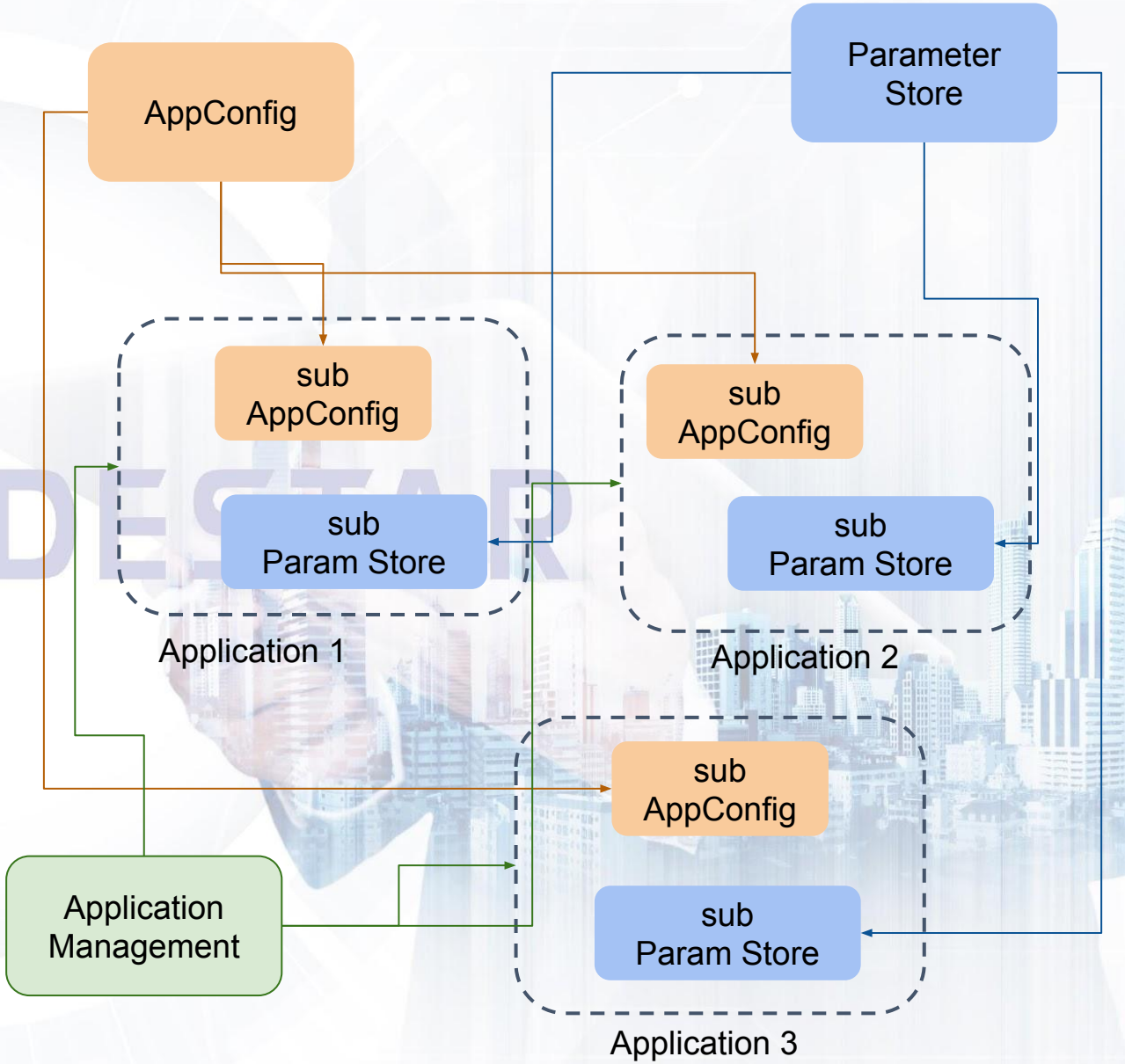
- **Explorer:** Tính năng cho phép theo dõi thông tin tổng hợp, bao gồm các OpsItem trên OpsCenter, AWS Config rules, CloudTrail, CloudWatch, runbooks, …
- **Incident Manager**: Theo dõi các Event, để tiến hành thực hiện các hành động tương ứng khi có Event xảy ra từ EventBridge hay CloudWatch -> chạy các runbook tự động trên Automations.



EC2 Management

Application Manager:

- **AppConfig**: Tính năng giúp dễ dàng quản lý các env, các biến môi trường thường xuyên thay đổi. Thường sử dụng kèm với CodePipeline.
- **Parameter Store**: Lưu trữ các giá trị dạng key value.
- **Application Management:** Đóng gói các thành phần của Application thành một App, cung cấp đầy đủ các OpsItem, Flag, … hiện tại của Application



EC2 Management

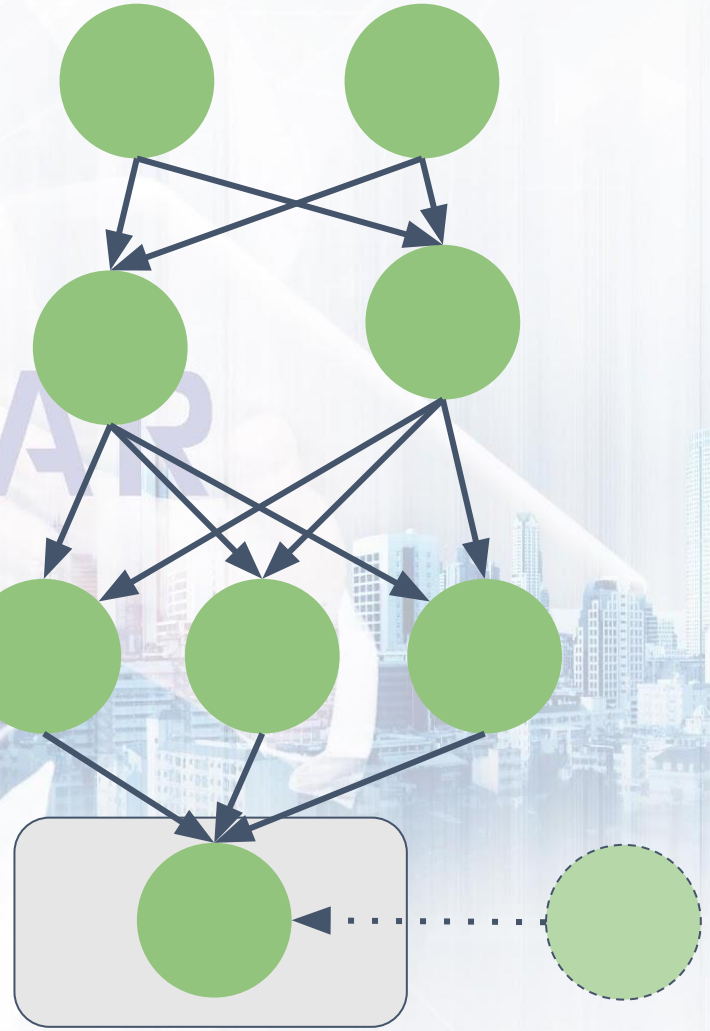
Application Manager:

- **Automation**: Cung cấp một bộ các thao tác thực hiện tác vụ nào đó, có thể viết dưới dạng các Script Python hoặc PowerShell.
- **PatchManager**: Tính năng giúp Chạy các bản Patch của hệ thống một cách tự động, thường là các bản Patch của hệ điều hành.
- **State Manager**: Chạy các Automation runbooks tự động, tuy nhiên State Manager tập trung vào các công việc configuration vào đầu lúc chạy, thiết lập các thư viện lúc đầu.
- **Manager Maintenance Window**: Thiết lập một schedule để chạy các thao tác runbooks.

High Availability (HA)

HA: Là khái niệm mô tả khả năng sẵn sàng cao của một hệ thống.

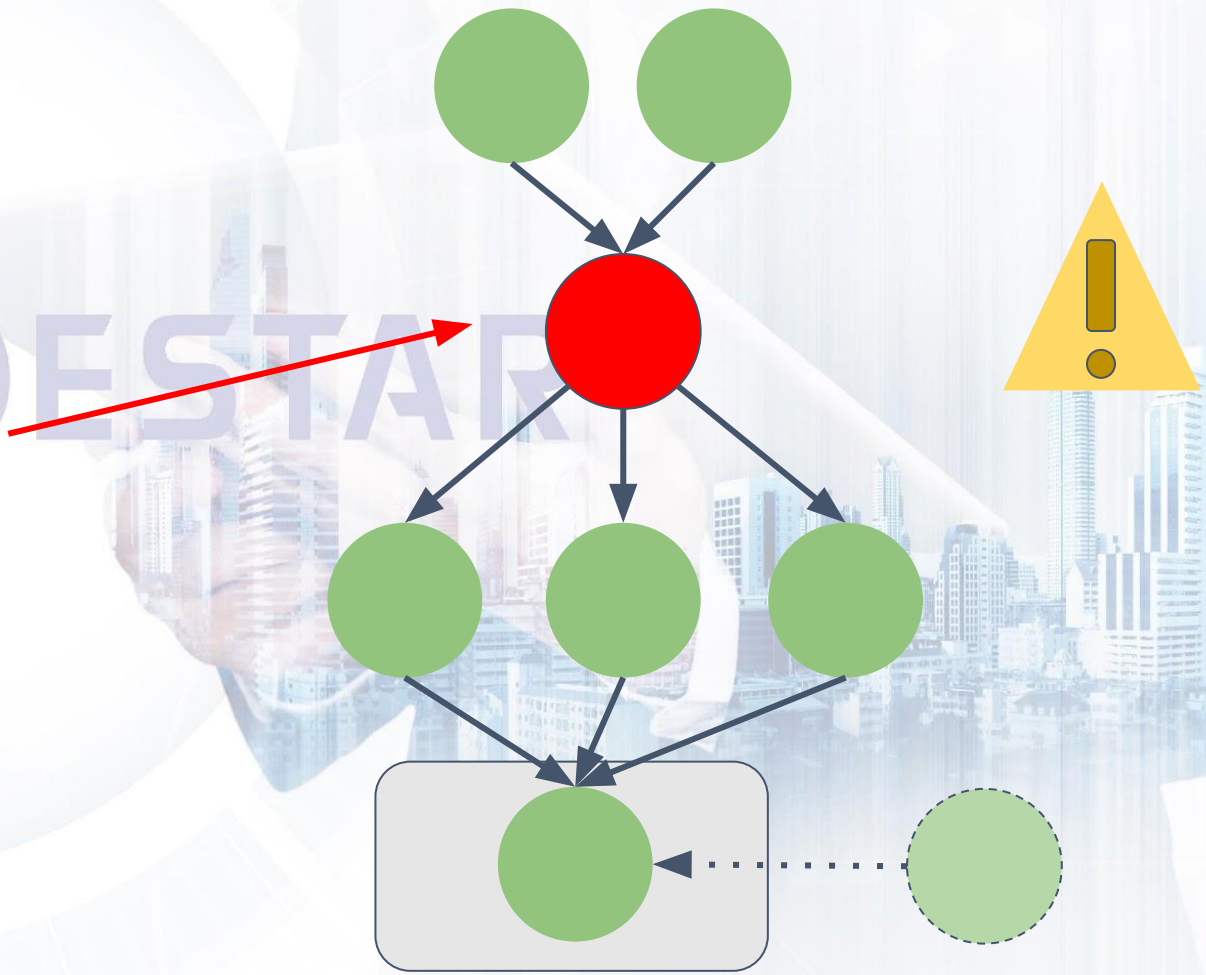
Hệ thống vẫn có thể hoạt động tương đối bình thường, đối với một số tình huống bị hỏng hóc hoặc chết server -> Hệ thống có tính HA.



High Availability (HA)

Single Point of Failure.

Hệ thống có một “điểm chết", nếu như tại điểm này, có vấn đề xảy ra, làm cho cả hệ thống bị trì trệ.



Khi xem xét hệ thống, chúng ta cần lưu ý không để hệ thống có những điểm chết thế này.

High Availability (HA)

Availability của hệ thống

Khi các thành phần chạy phụ thuộc vào nhau theo thứ tự, tính Availability sẽ giảm xuống

Avail cả hệ thống = 0.9 * 0.95 * 0.98 = 0.8379

Khi các thành phần làm cùng nhiệm vụ của nhau trong một hệ thống, tính Availability sẽ tăng lên.

Avail chức năng A
= 1 - (1 - 0.9)*(1 - 0.9)
= 0.99

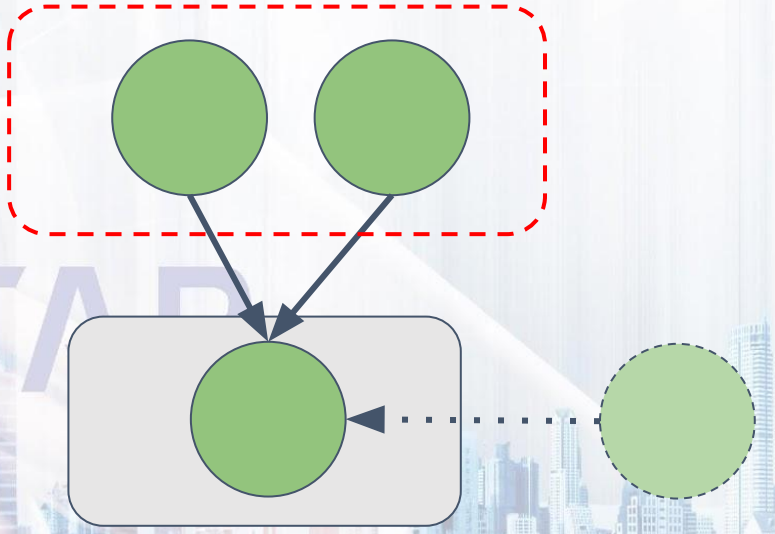
Avail chức năng B
= 1 - (1 - 0.95)*(1 - 0.95)
= 0.997

Avail cả hệ thống = 0.99*0.997 = 0.997 = 0.98703

High Availability (HA)

Một số phương án tăng availability

- Tăng số lượng thành phần làm cùng nhiệm vụ
- Sử dụng một dịch vụ tự manage và kiểm soát. Khi có vấn đề xảy ra, sẽ thay thế thành phần có sự cố.

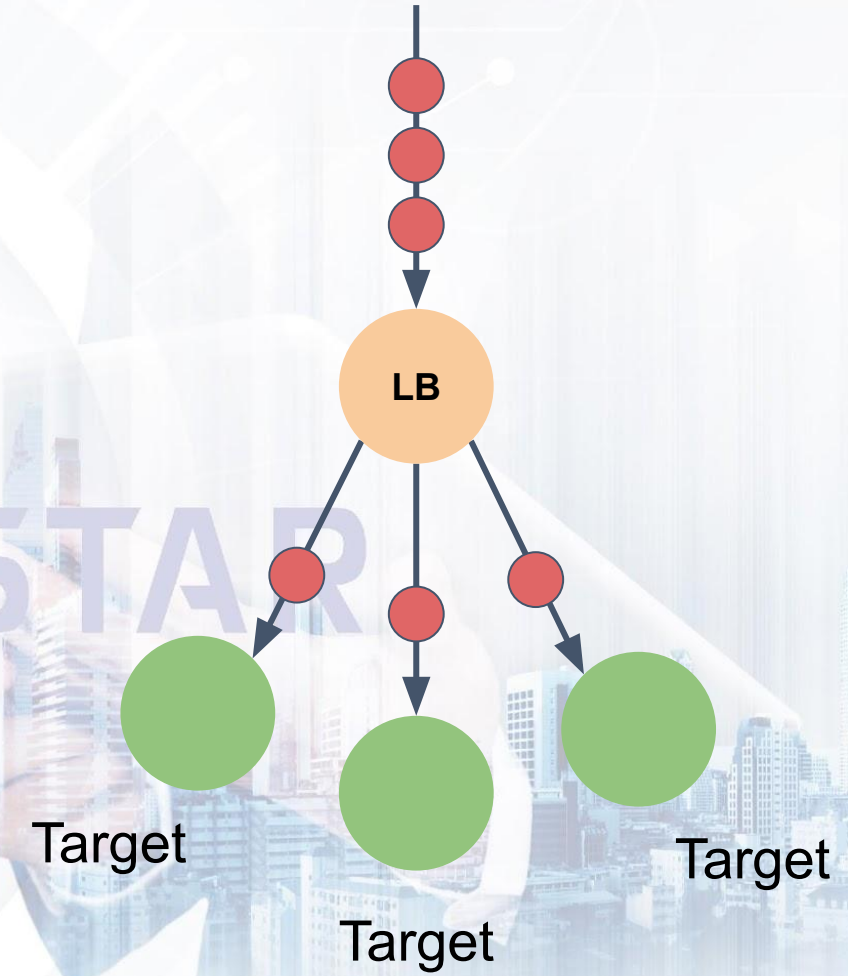


Lưu ý: Các phương án tăng availability nói chung đều làm tăng chi phí. Do đó, cần cân nhắc trước khi sử dụng.

Load Balancer

Thành phần giúp cho Request trỏ tới các target phía dưới.

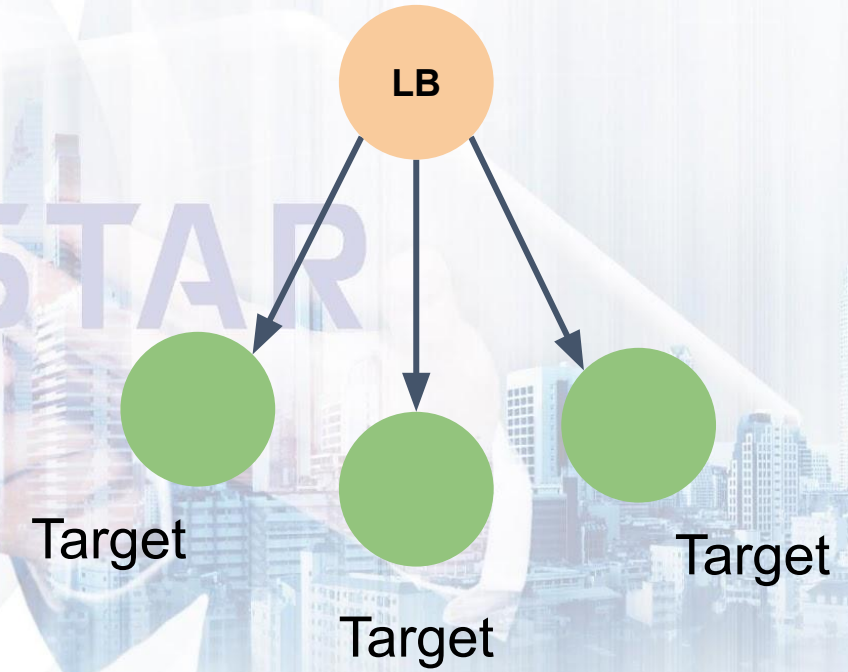
Phân phối các target được nhận request theo một tỷ lệ hoặc một rule nào đó.



Load Balancer

Các loại LB:

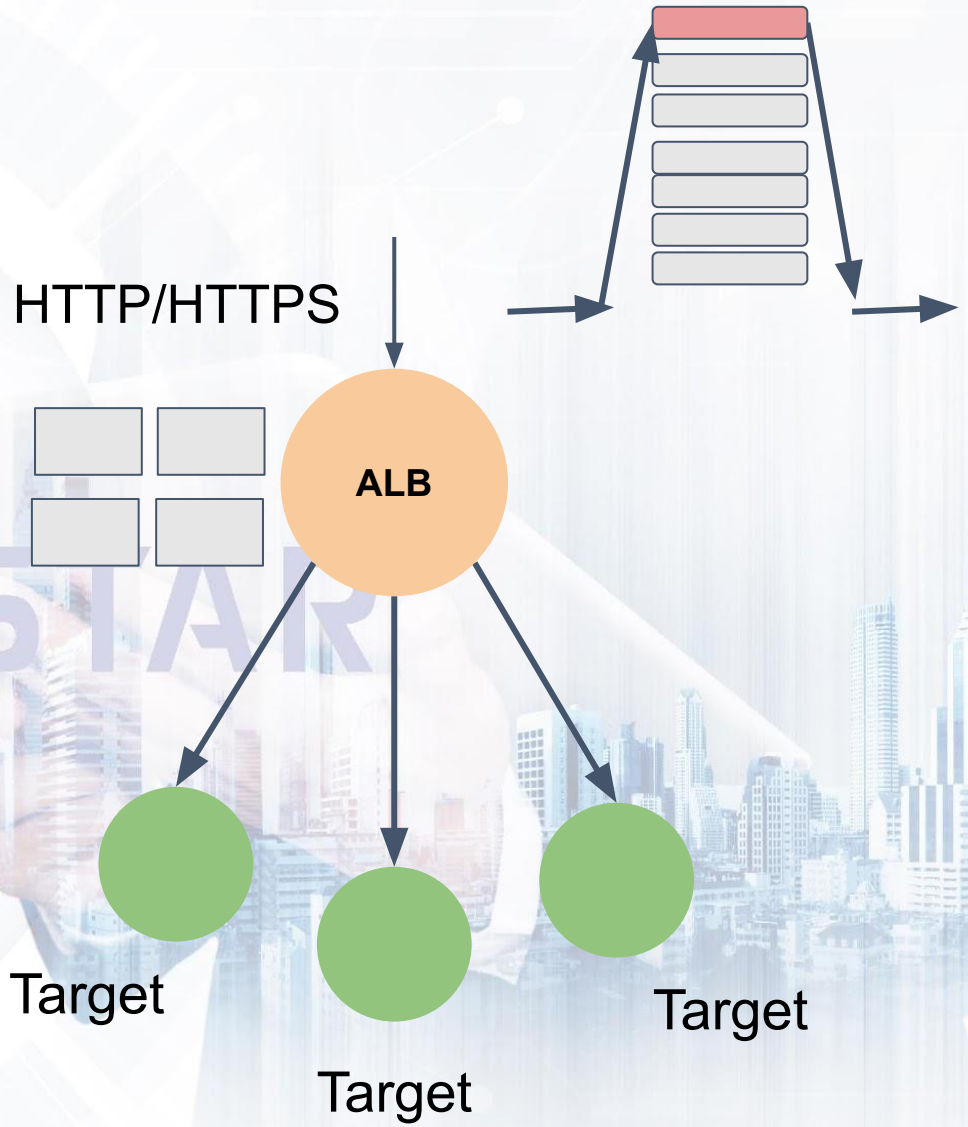
- Application LB
- Network LB
- Gateway LB



Load Balancer

Application Load Balancer

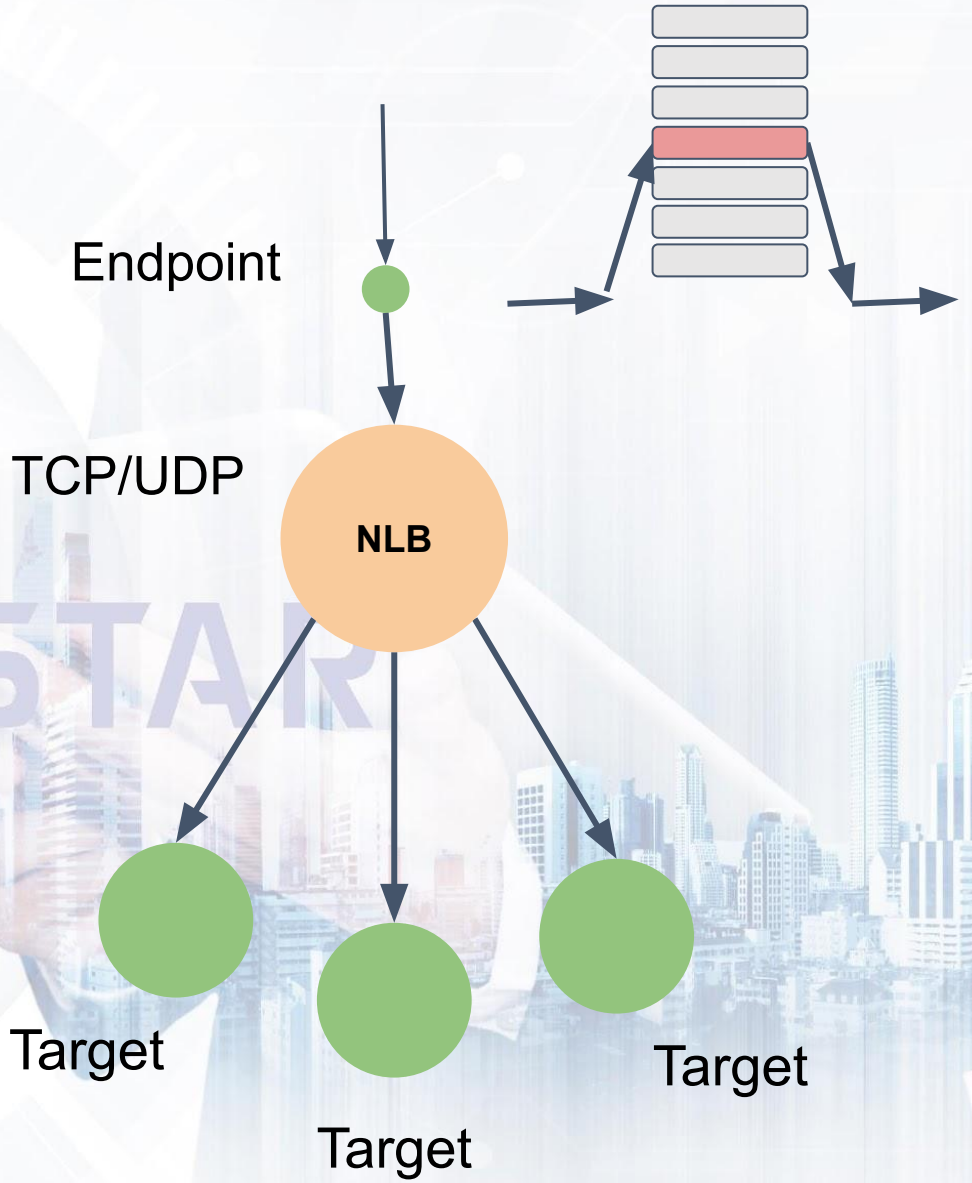
- ALB cho phép các protocol loại HTTP/HTTPS đi qua
- Có thể access layer 7 của gói tin đi qua (nên có thể truy xuất toàn bộ thông tin URL, Request, Method, …)
- Có thể integrate với rất nhiều các dịch vụ khác như ACM, WAF, …



Load Balancer

Network Load Balancer

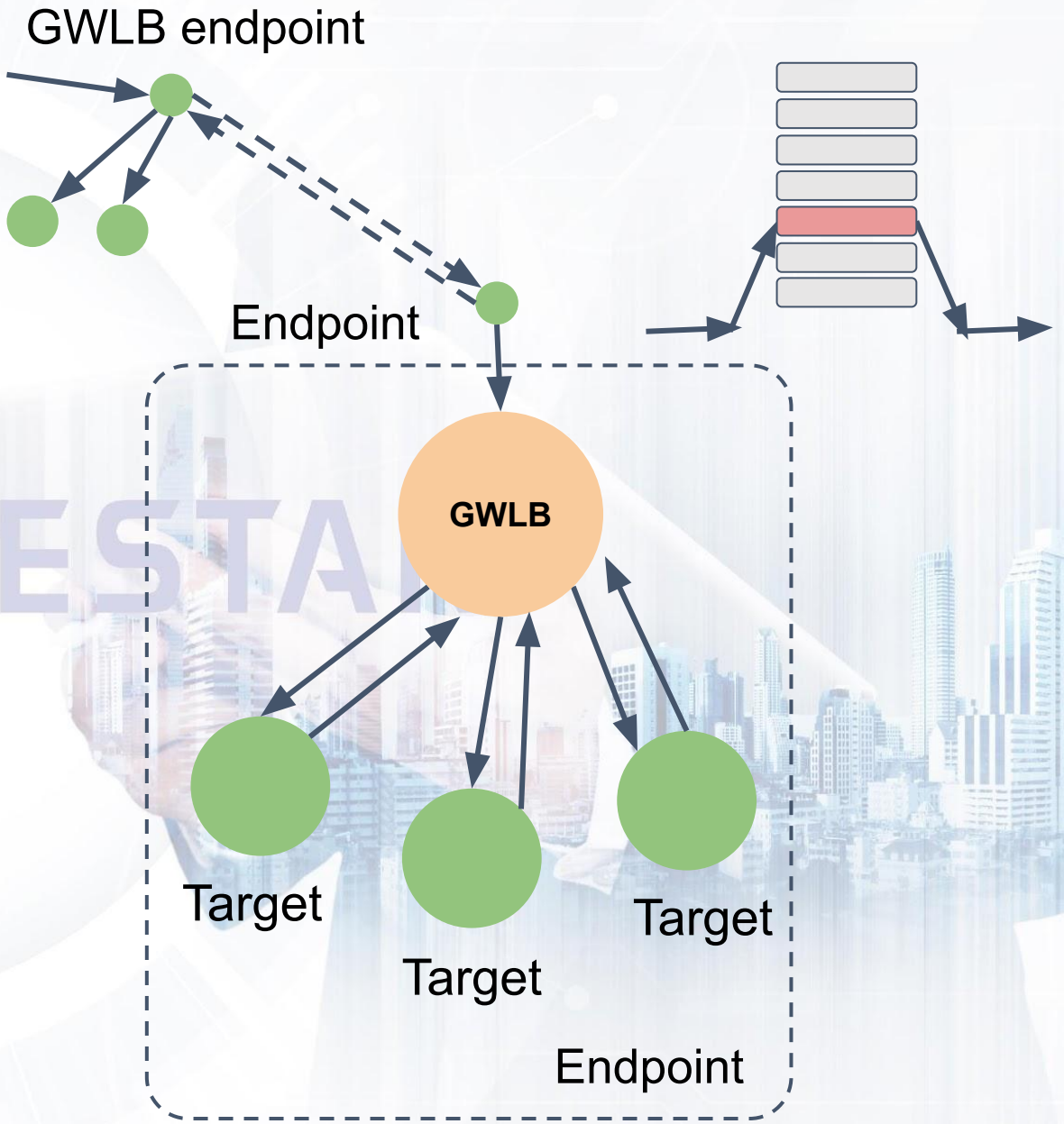
- NLB cho phép các protocol loại TCP/UDP đi qua
- Có thể access layer 4 của gói tin đi qua (có thể truy xuất được protocol, port của gói tin)
- Có thể gán IP tĩnh.



Load Balancer

Gateway Load Balancer

- GWLB là loại LB đặc biệt, cho phép đưa logic xử lý từ bên ngoài vào VPC hiện tại và thực hiện các thao tác monitor, filter request mà không chỉnh sửa thông tin đi qua nó.
- GWLB xử lý ở layer số 3 trong mô hình OSI.

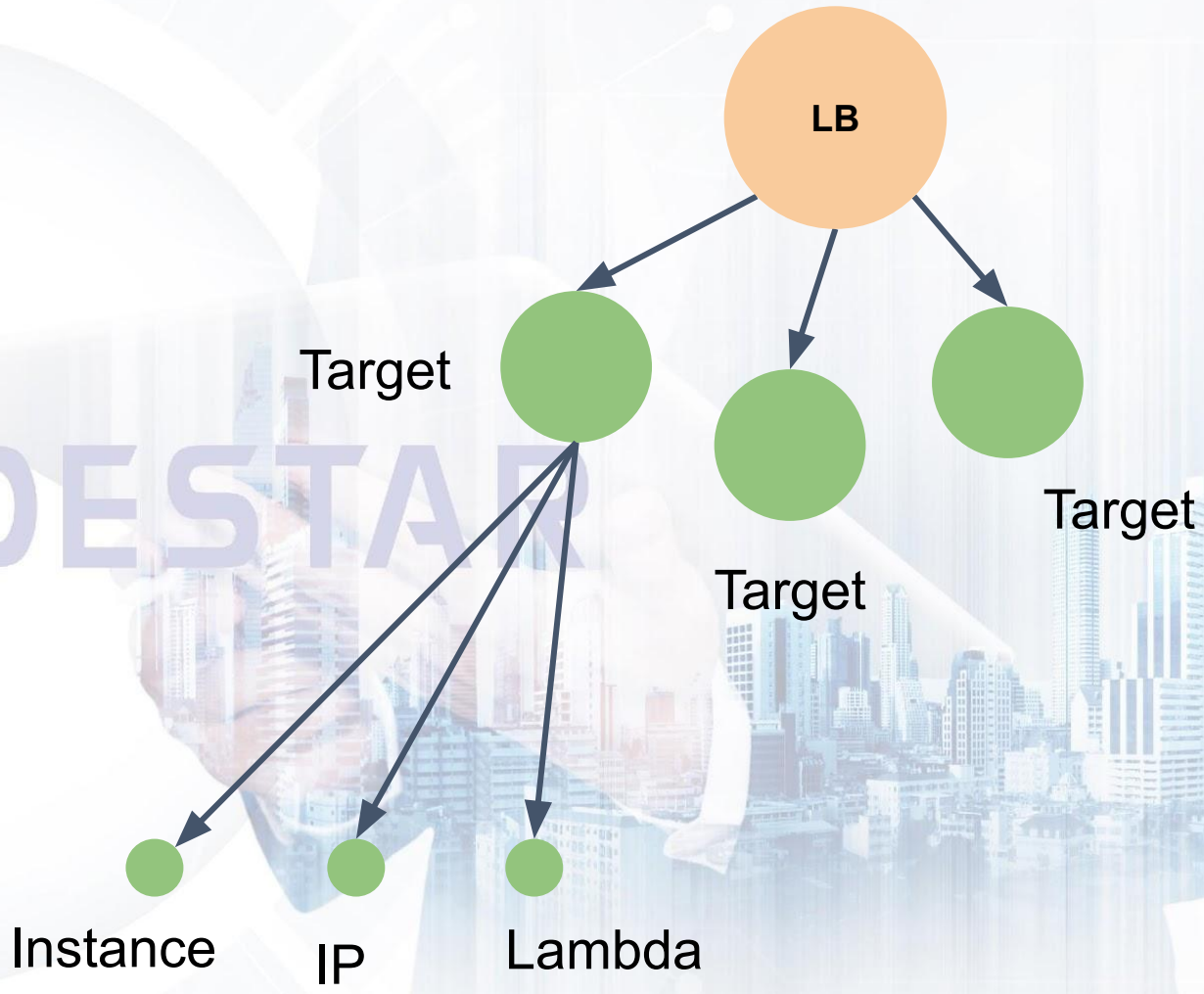


Load Balancer

Target

Các LB có thể đưa request tới các Target. Các target ở đây có thể là:

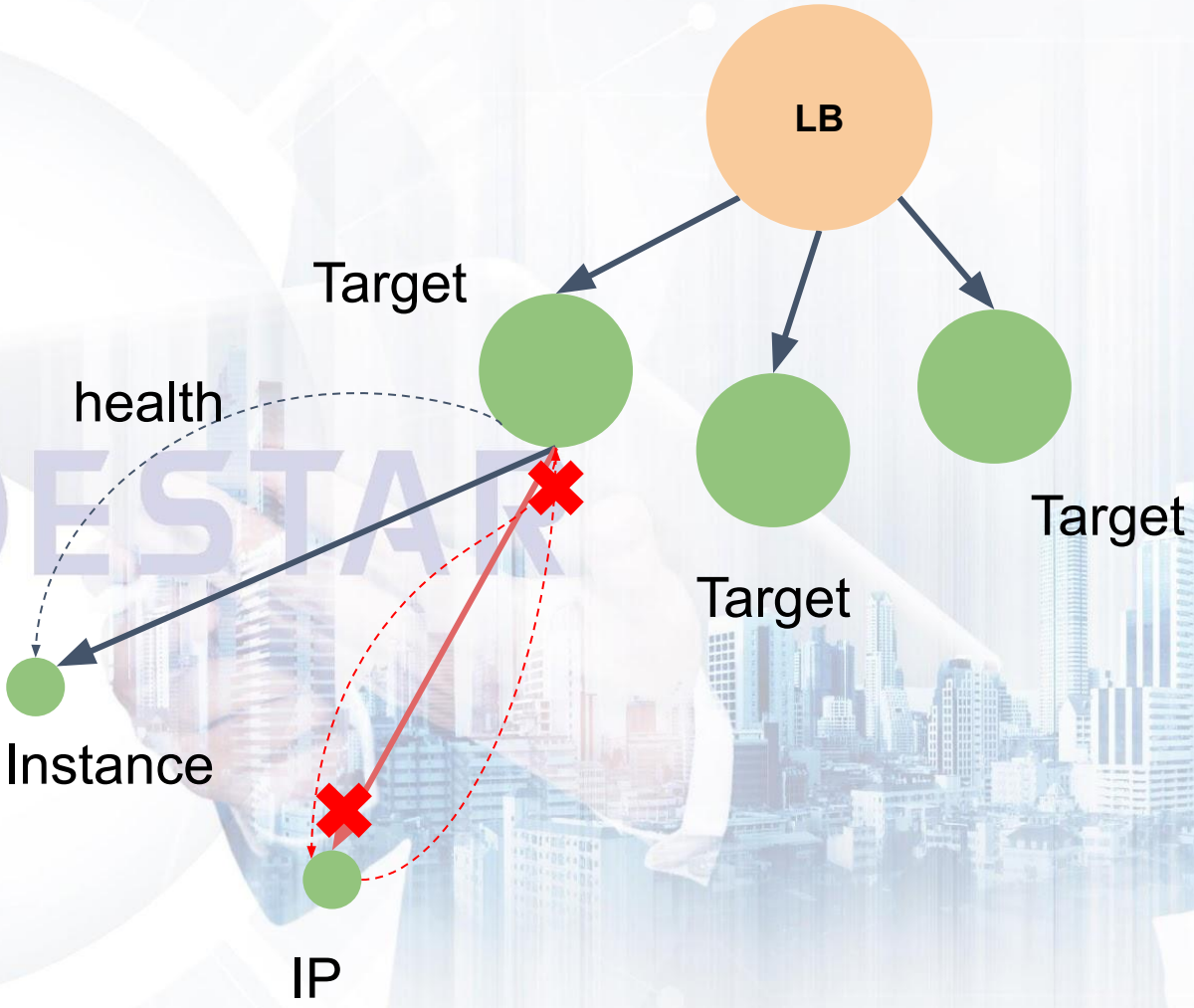
- IP
- Instance
- ALB khác
- Lambda



Load Balancer

Target

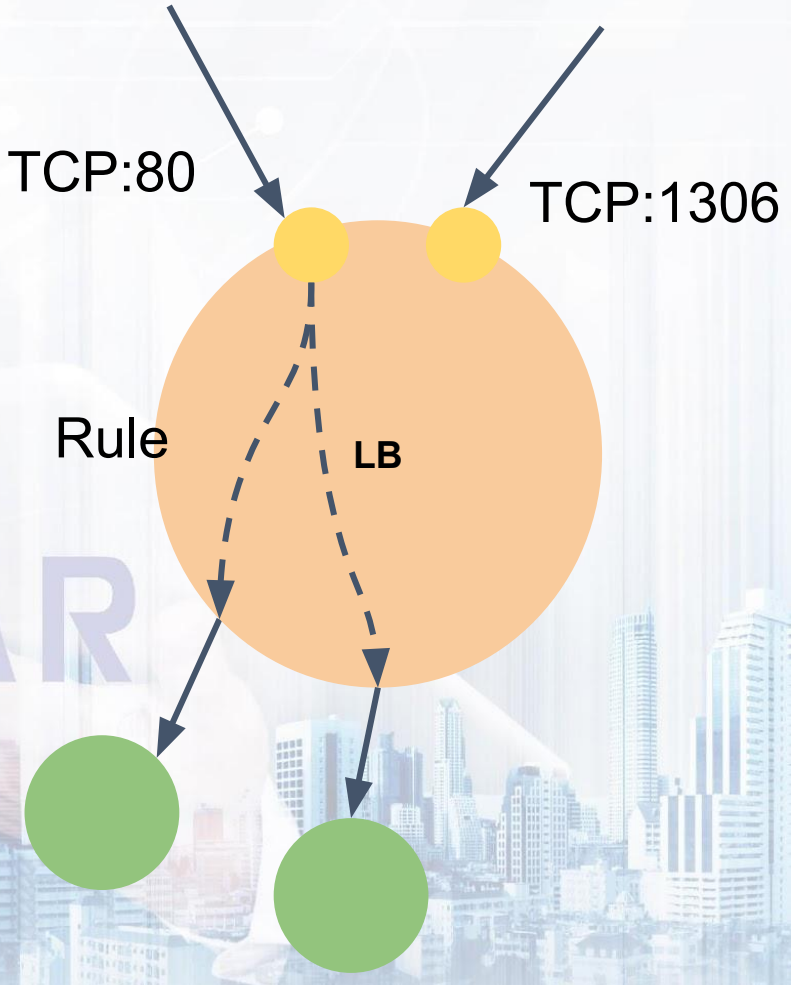
- Các target sẽ có các healthcheck để kiểm tra xem target có đang hoạt động hay không.
- Nếu healthcheck trả về là instance unhealthy, các request sẽ dừng đưa đến vị trí đó.



Load Balancer

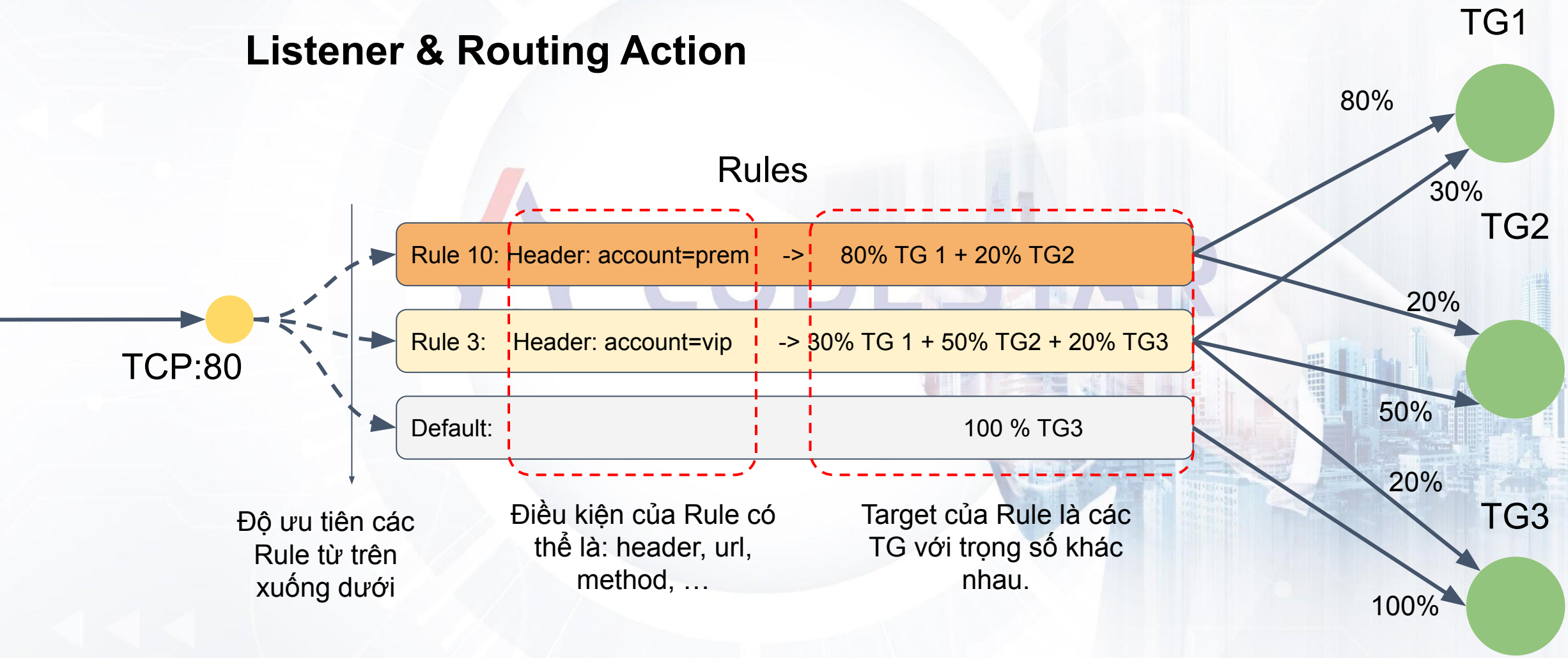
Listener & Routing Action

- Listener dùng nghe một cổng nào đó.
- Dựa vào Rule, chúng ta sẽ điều hướng traffic tới một target group.
- Rule có thể điều hướng Routing Action trỏ tới:
  - + Các Target Group
  - + Một URL khác
  - + Một response cố định.



Load Balancer

Listener & Routing Action



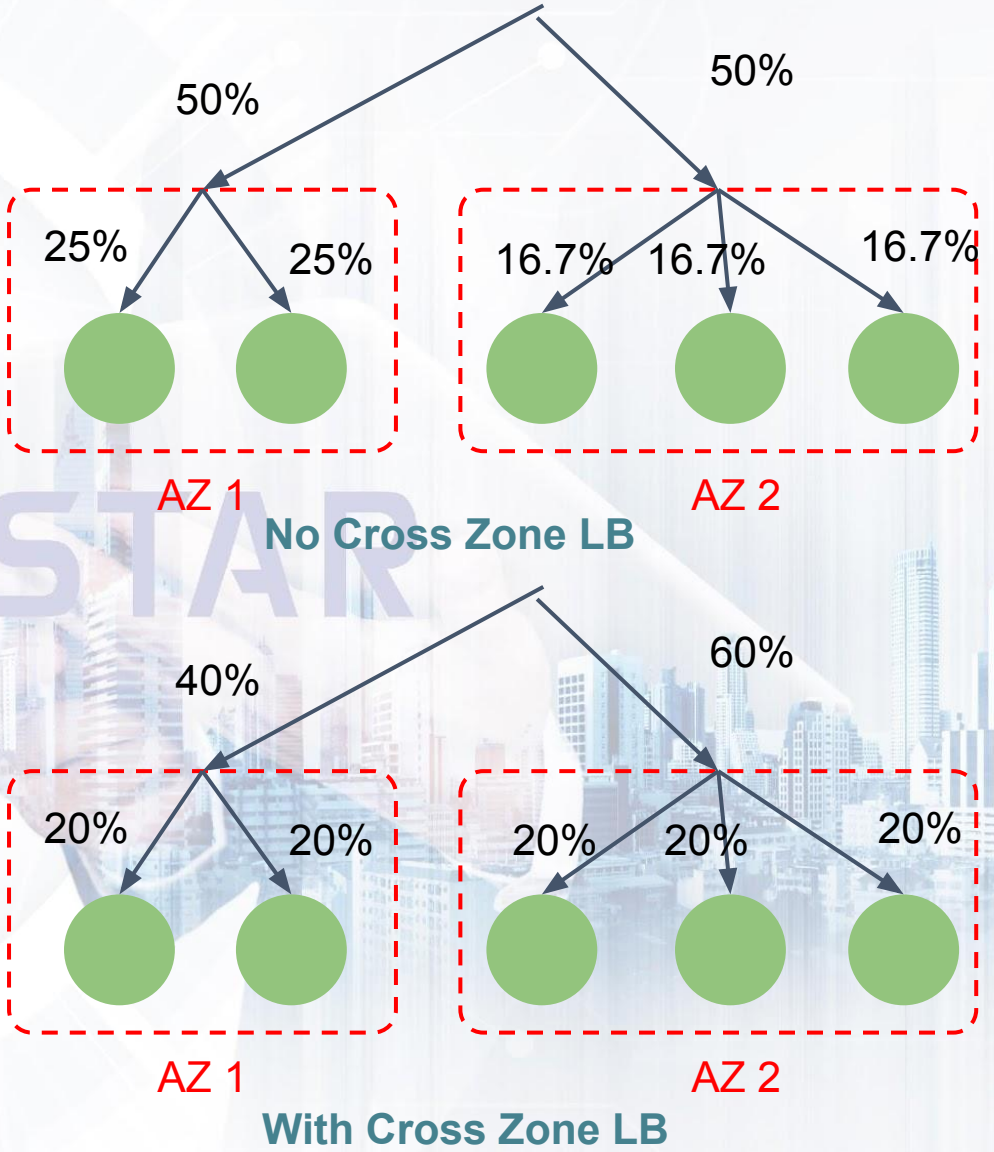
Load Balancer

Cross Zone Load Balancing

Đây là tính năng cho phép chúng ta cân bằng tải trên các Zone.

Khi không bật chức năng này, Traffic sẽ phân đều tới các AZ, sau đó tại mỗi AZ, phân đều tới các Target

Khi bật chức năng này, traffic sẽ coi toàn bộ Target là một khối và phân đều.

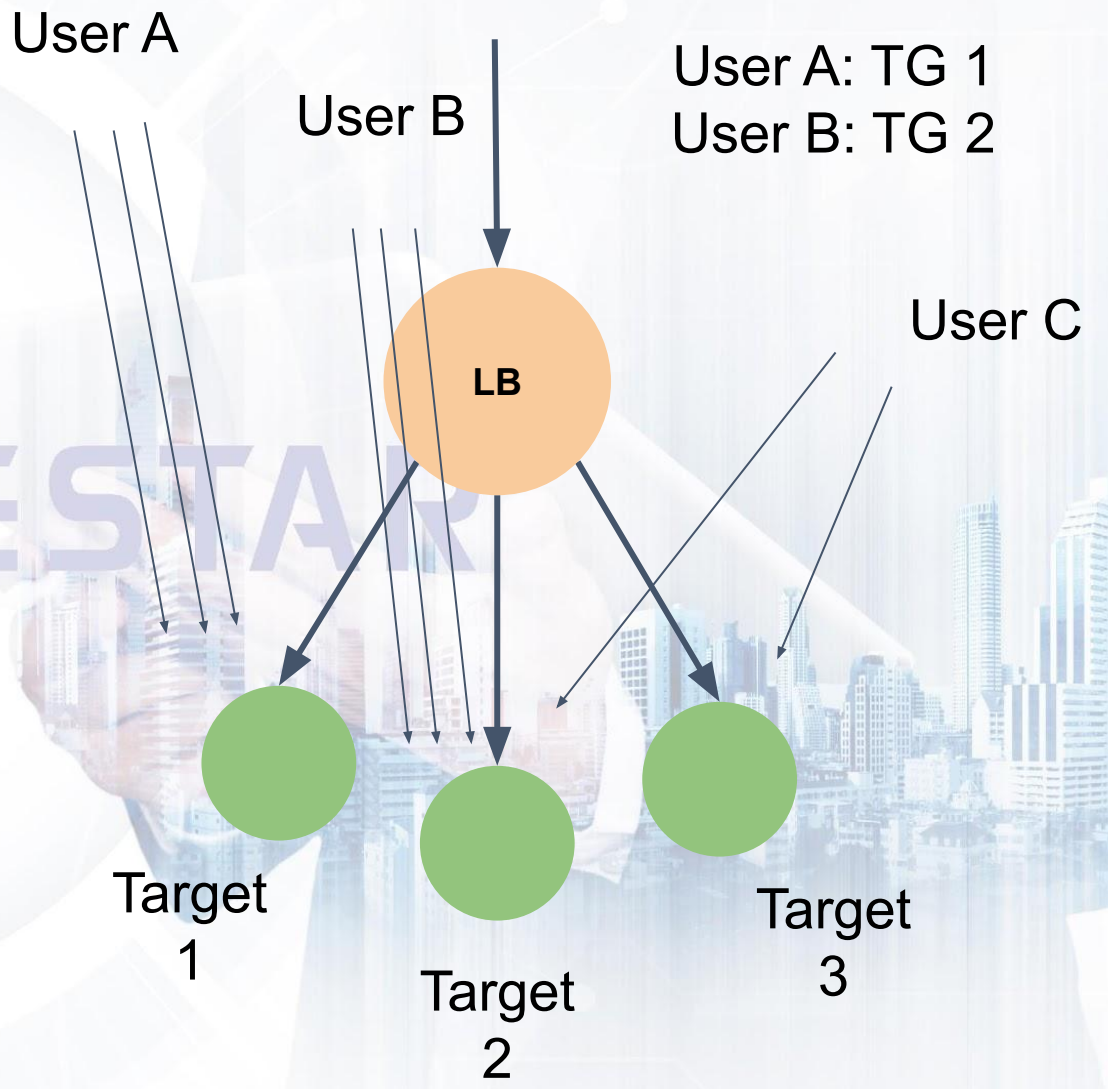


Load Balancer

Sticky Session

- Sticky Session cho phép cùng 1 User/Session, sử dụng cùng 1 target trong một khoảng thời gian 1s - 7 ngày.
- Nếu không sử dụng Sticky Session, request có thể chuyển đổi giữa các Target (như User C)

***Use case**: Sticky Session được sử dụng cho các hệ thống có lưu lại Session của người dùng như cho phép đăng nhập, lưu lại lịch sử thao tác trước đó, …*

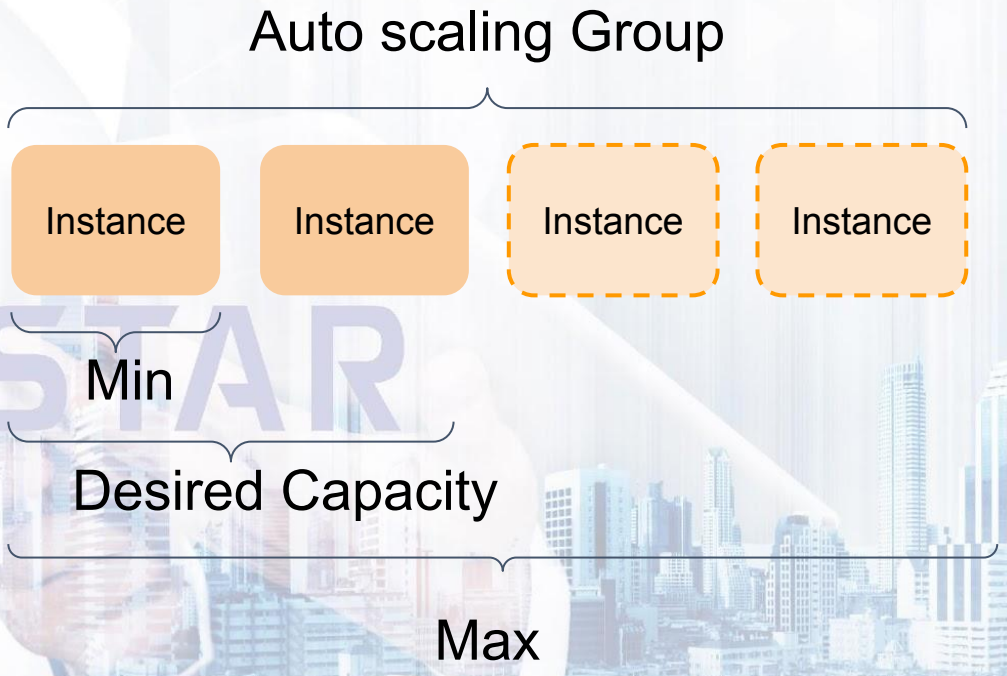


EC2 Auto Scaling

Auto Scaling Groups

Là một tập hợp các instance mục đích cho việc quản lý và tự động scale, đảm bảo duy trì số lượng instance.

Số lượng instance sẽ hướng tới con số desired capacity.

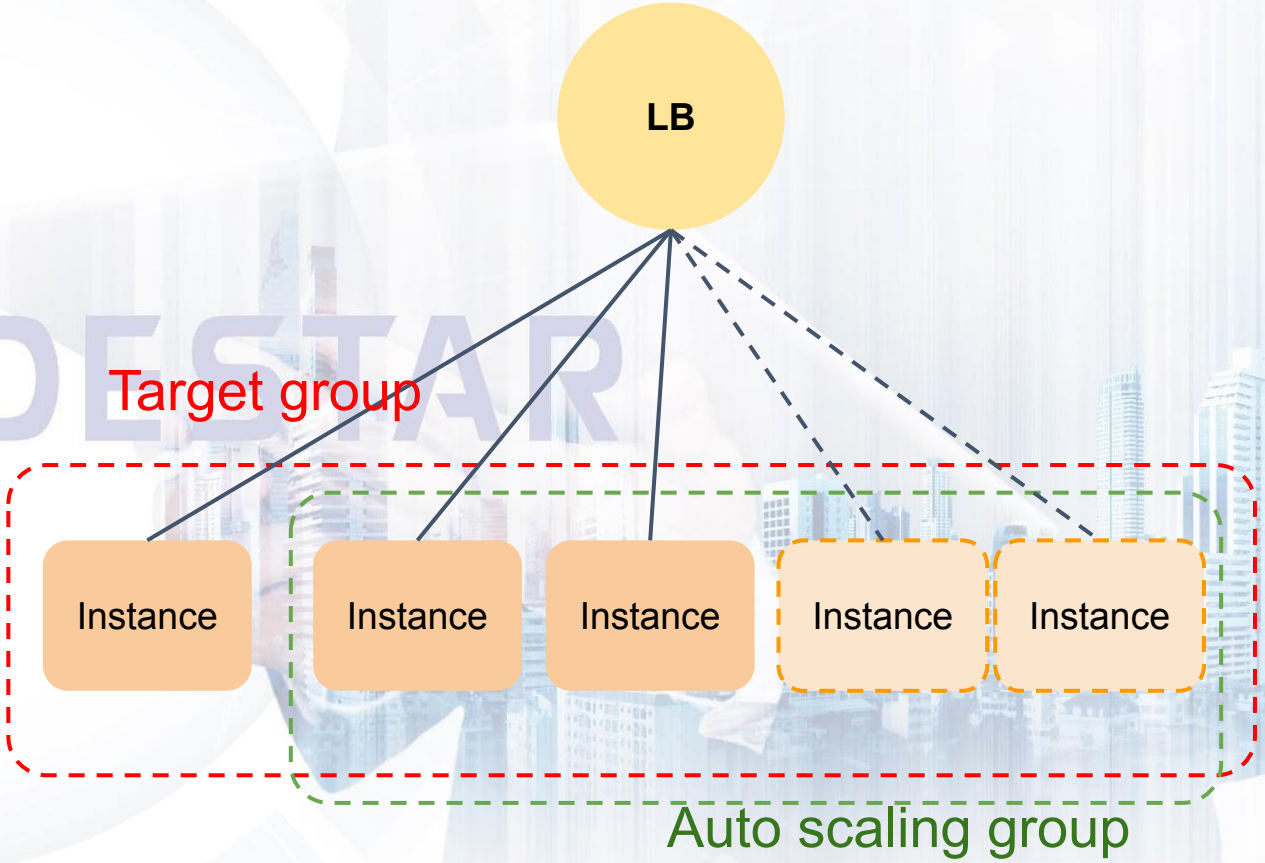


EC2 Auto Scaling

Kết hợp với LB

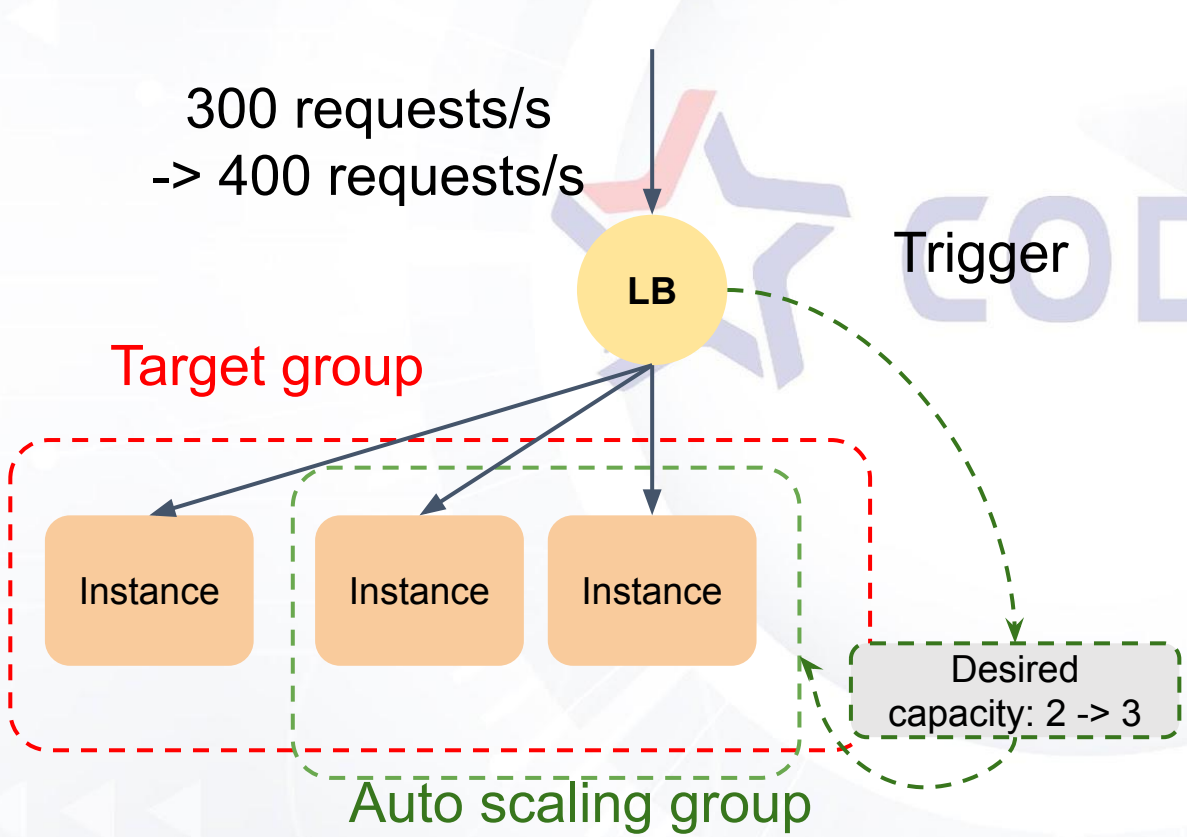
Auto Scaling thường kết hợp với Load Balancer để cân bằng tải cho toàn bộ instance nằm trong ASG.

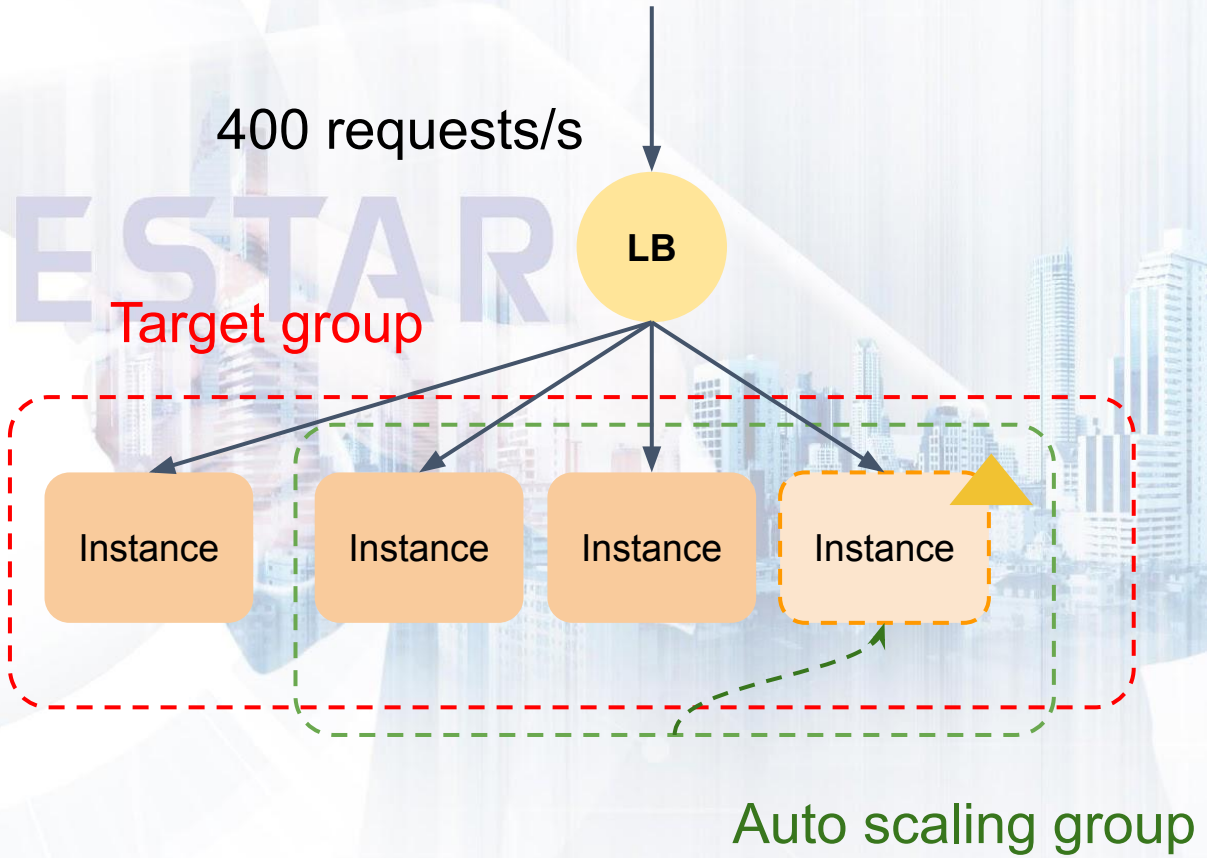
Các instance được tạo ra theo Template có thể nằm chung trong một Target của ALB.



EC2 Auto Scaling

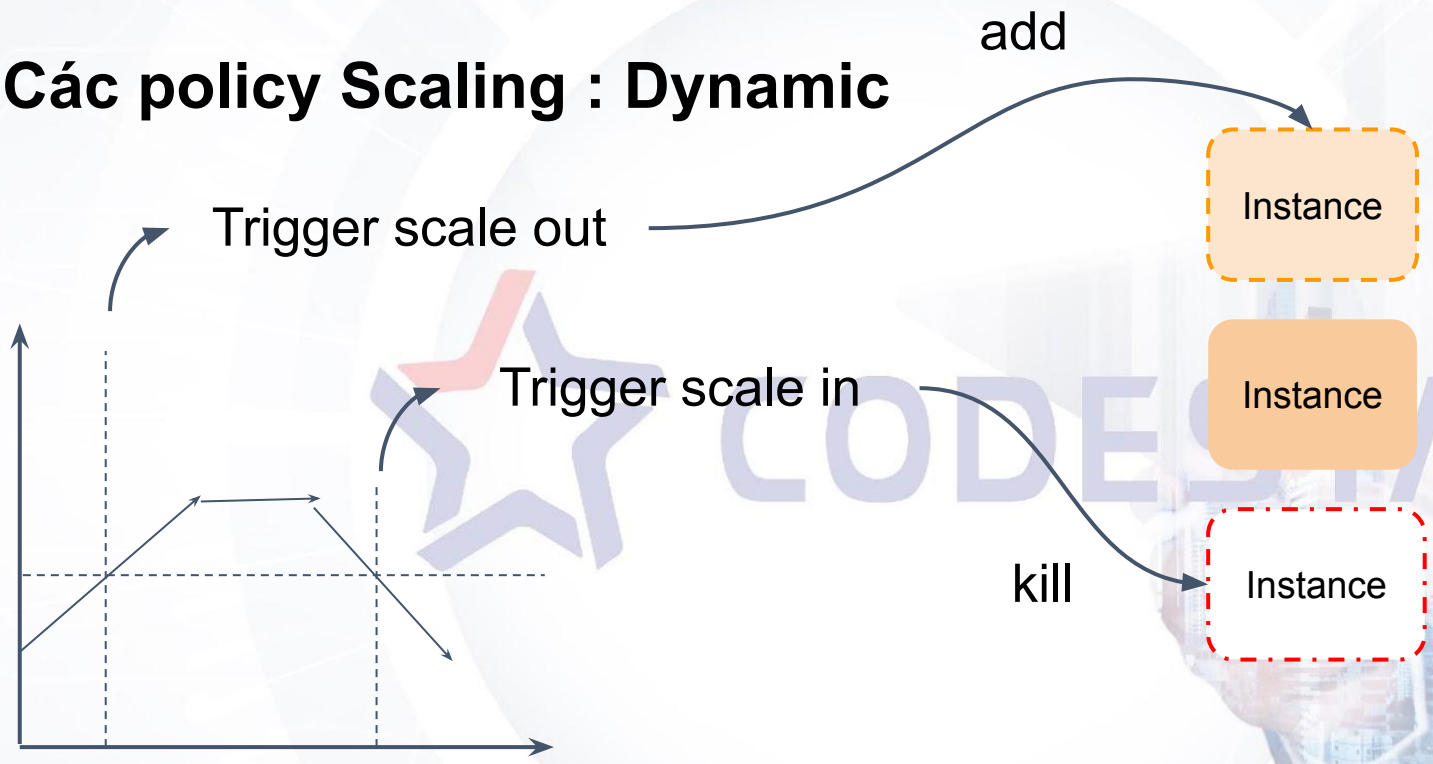
Hoạt động của Auto Scaling





EC2 Auto Scaling

Các policy Scaling : Dynamic

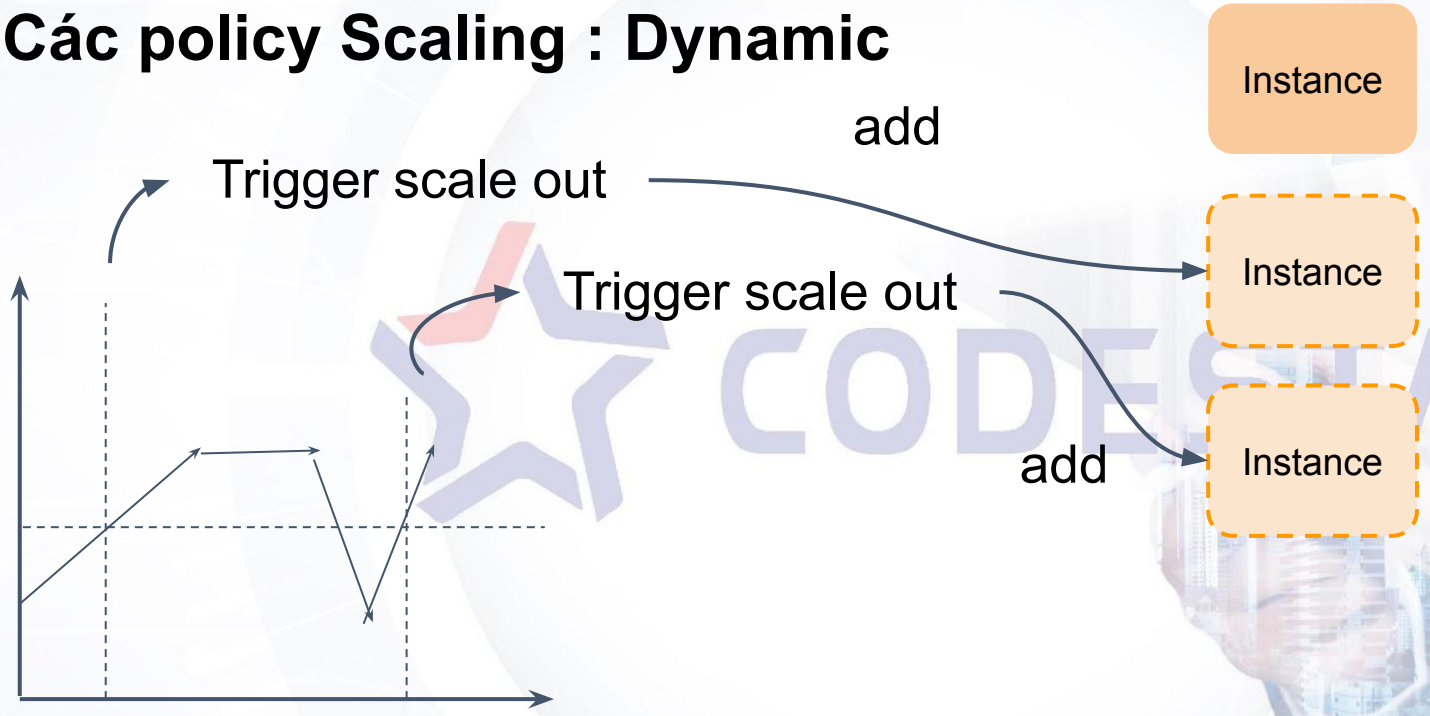


Target tracking policy

Sử dụng một mốc/một tham số trên CloudWatch để tăng, giảm số lượng instance theo mốc đó, với mục tiêu cân bằng tham số đó ở giá trị nhất định.

EC2 Auto Scaling

Các policy Scaling : Dynamic

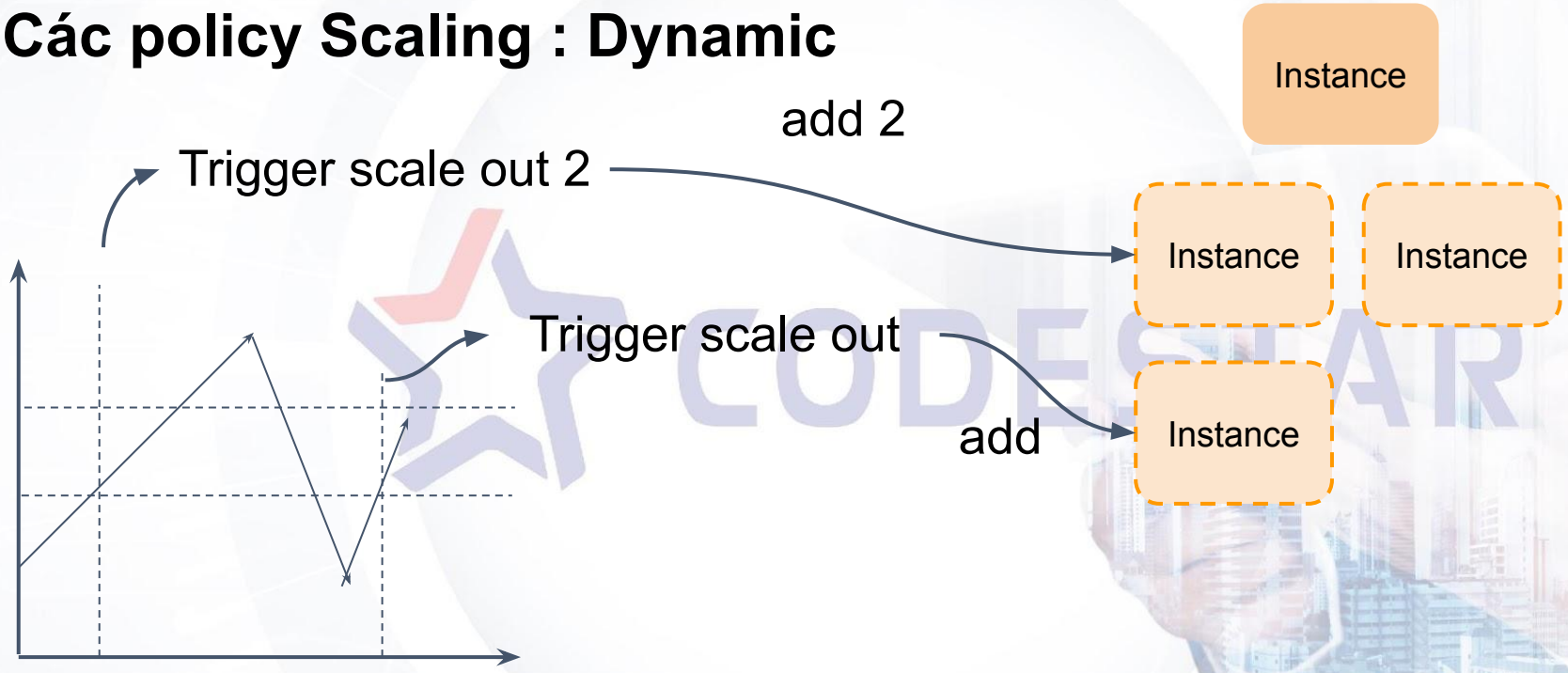


Simple scaling policy

Sử dụng scale tăng lên (hoặc giảm đi) mỗi khi vượt qua một mốc.

EC2 Auto Scaling

Các policy Scaling : Dynamic



Step scaling policy

Sử dụng scale tăng lên số instance dựa theo các mốc cần scale.

EC2 Auto Scaling

Các policy Scaling : Schedule

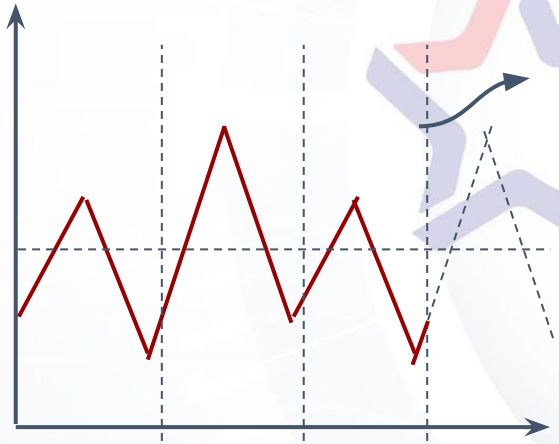
Trigger scale out

add

Schedule scaling trigger scaling action theo thời gian.

EC2 Auto Scaling

Các policy Scaling : Predictive

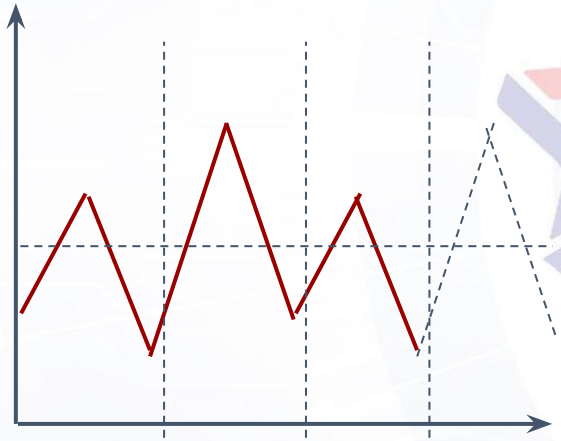


Dự đoán chuẩn bị tăng -> trigger add

Predictive Scaling sẽ dự đoán thời điểm chuẩn bị cần tăng scale để tăng trước

EC2 Auto Scaling vs AWS Auto Scaling

AWS Auto Scaling được dùng
để kích hoạt nhiều
Service/EC2 ASG cùng lúc



AWS Auto Scaling

Instance

Instance

EC2 Auto Scaling

Aurora read replicas

ECS

EMR cluster

SageMakers

Spot Fleet

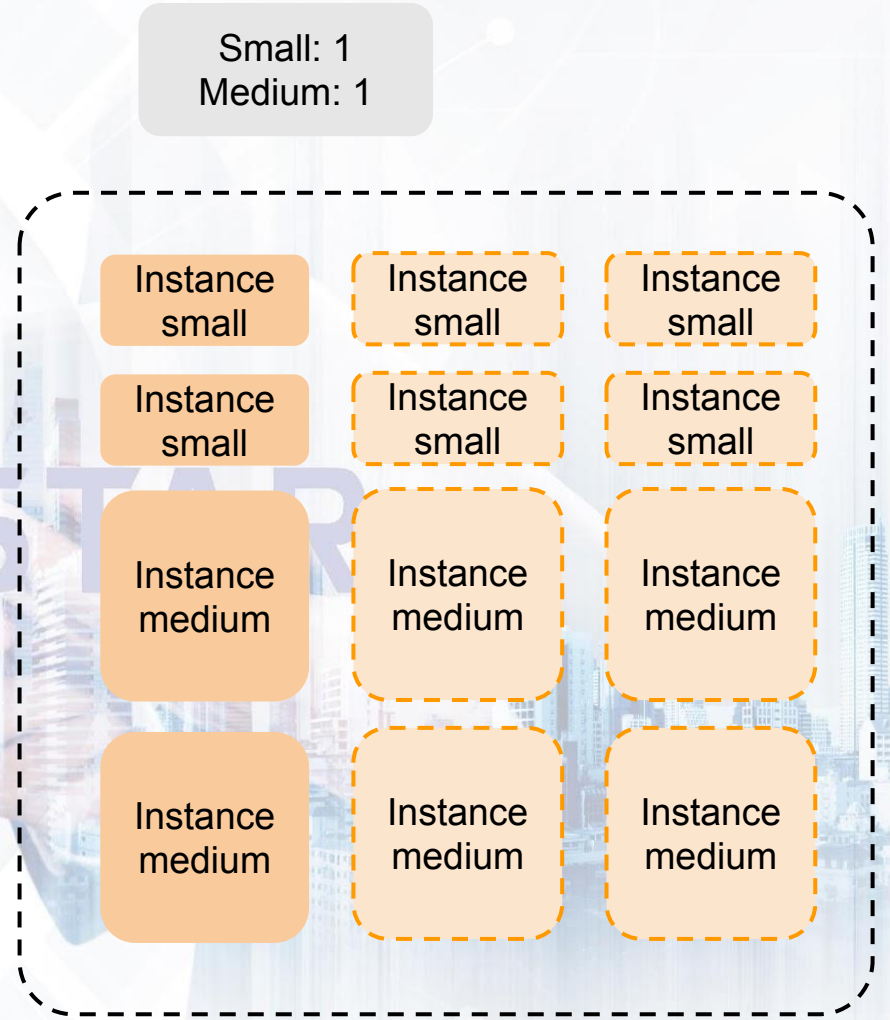
Application Auto Scaling API

(Scale những resource khác không phải EC2 instance)

EC2 Auto Scaling

Có thể thiết lập weight (trọng số) cho các loại instance type khác nhau.

Khi scale, Auto Scaling Groups sẽ tiến hành khởi tạo các resource theo như thiết weight tại Auto Scaling Group.



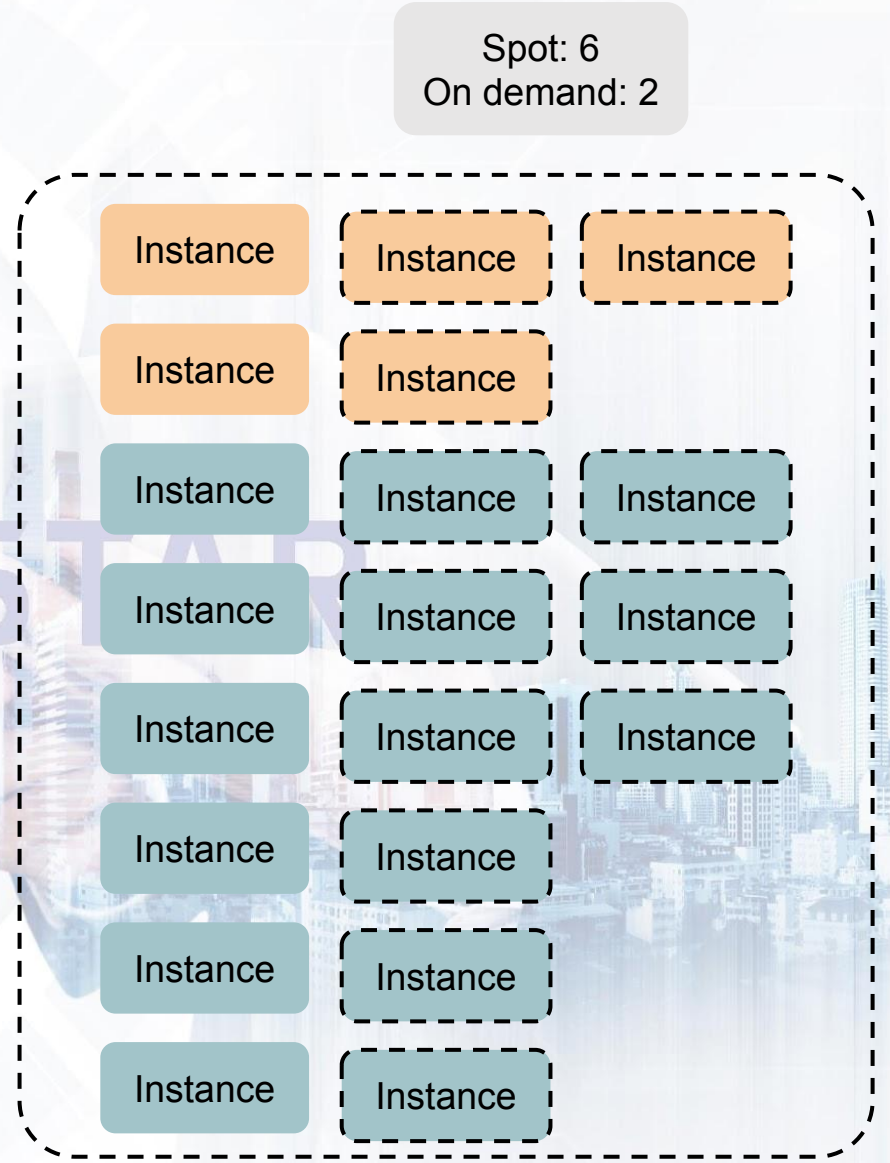
Scale từ 4 -> 12
(Có thể tính scale vCPU từ 12 -> 36)

EC2 Auto Scaling

Có thể thiết lập weight (trọng số) cho các loại purchase khác nhau.

Khi scale, Auto Scaling Groups sẽ tiến hành khởi tạo các resource theo theo thông số đã được định trước

Sử dụng Spot instance và On Demand instance sẽ giúp chúng ta tối ưu được chi phí.



Scale từ 8 -> 20

CODESTAR

THANK YOU