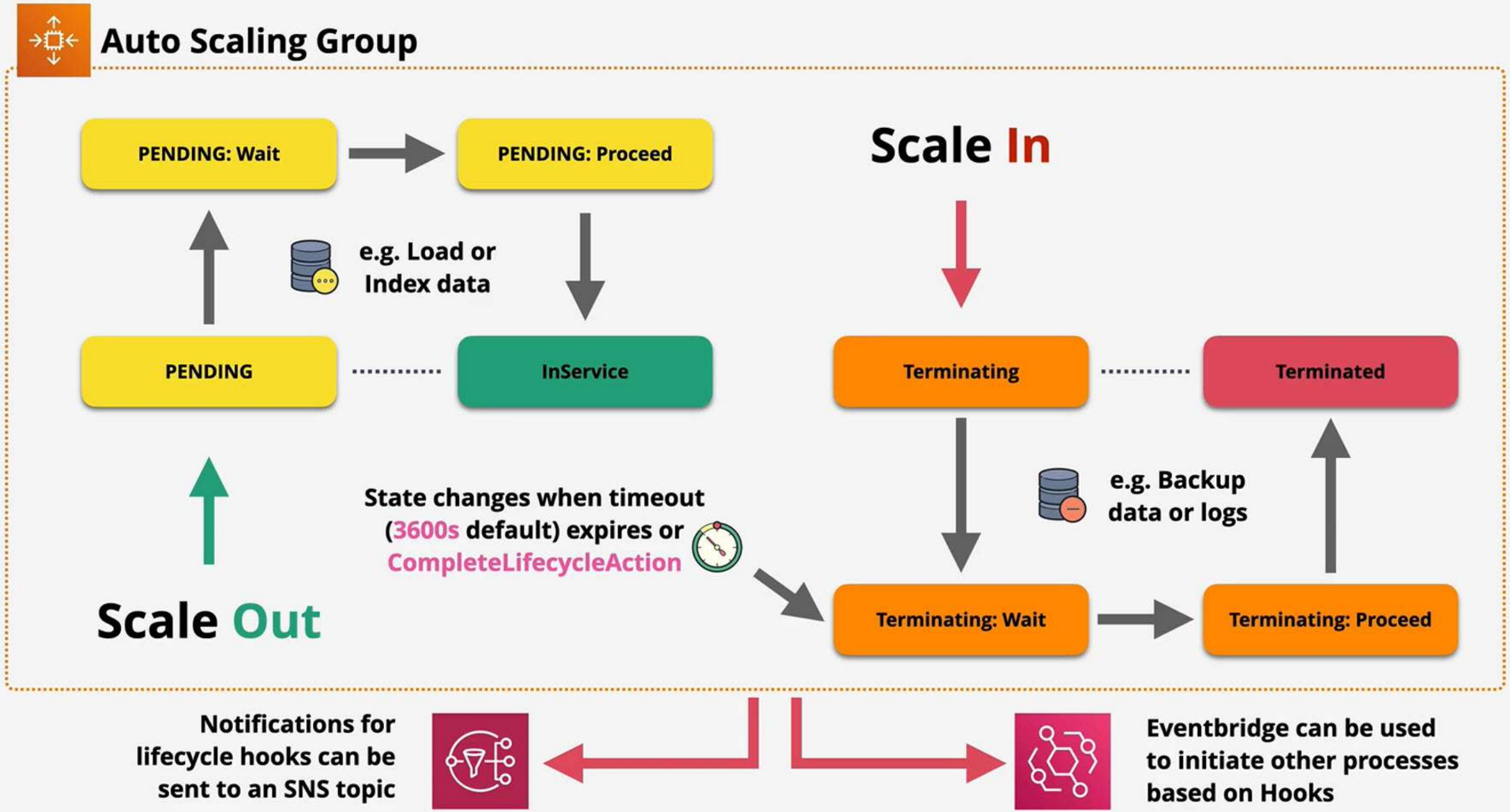
ASG Health Checks

- **EC2** (**Default**), **ELB** (**Can be enabled**) & **Custom**
- **EC2** - Stopping, Stopped, Terminated, Shutting Down or Impaired (not 2/2 status) = **UNHEALTHY**
- **ELB** - **HEALTHY** = **Running** & **passing ELB health check**
- … can be more **application aware** (Layer 7)
- **Custom** - Instances marked **healthy** & **unhealthy** by an external system.
- Health check grace period (Default **300s**) - **Delay before starting checks**
- … allows **system launch**, **bootstrapping** and **application start**

Auto Scaling Groups (ASG) Health checks

ASG Lifecycle Hooks



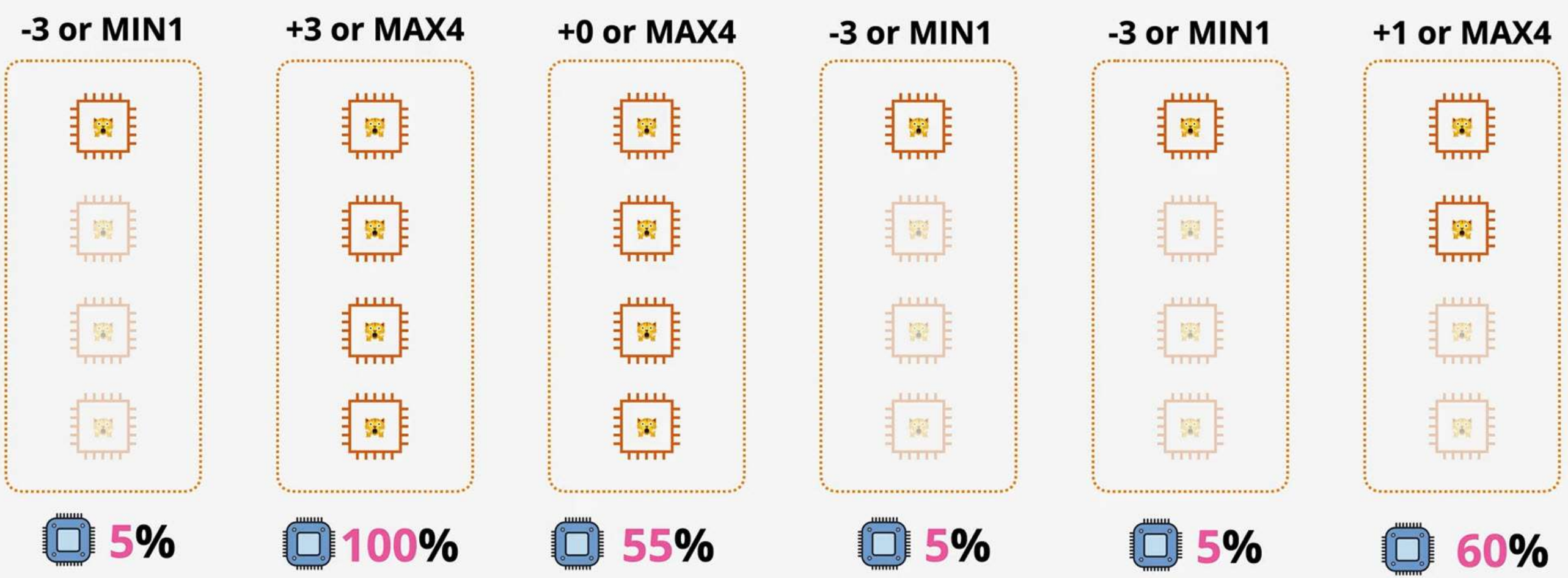
ASG Lifecycle Hooks

- **Custom Actions** on instances during **ASG actions**
- .. **Instance launch** or **Instance terminate** transitions
- Instances are paused within the flow .. they **wait**
- ... until a timeout (then either **CONTINUE** or **ABANDON**)
- ... or you resume the ASG process **CompleteLifecycleAction**
- EventBridge or SNS Notifications

ASG Lifecycle Hooks

ASG - Step Scaling

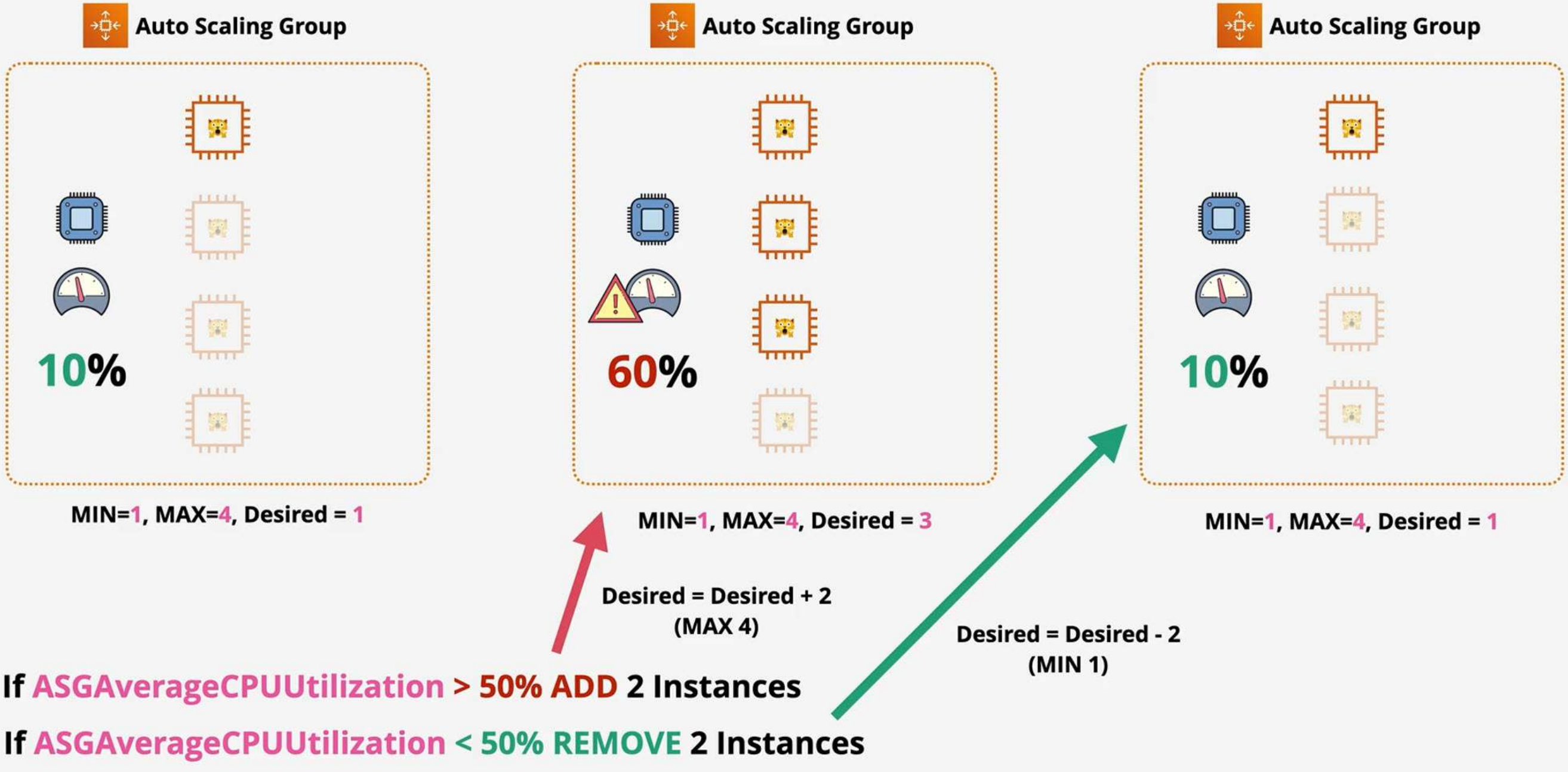
https://learn.cantrill.io adriancantrill



50%-59% DO NOTHING
60%-69% ADD 1
70%-79% ADD 2
80%-100% ADD 3

40%-49% DO NOTHING
30%-39% REMOVE 1
20%-29% REMOVE 2
0%-19% REMOVE 3

ASG - Simple Scaling



ASG Scaling Policies

- ASGs don't NEED scaling policies - they can have none
- **Manual** - **Min**, **Max** & **Desired** - Testing & Urgent
- **Simple** Scaling
- **Step** Scaling
- **Target** Tracking
- Scaling Based on **SQS** - **ApproximateNumberOfMessagesVisible**

ASG Scaling Policies

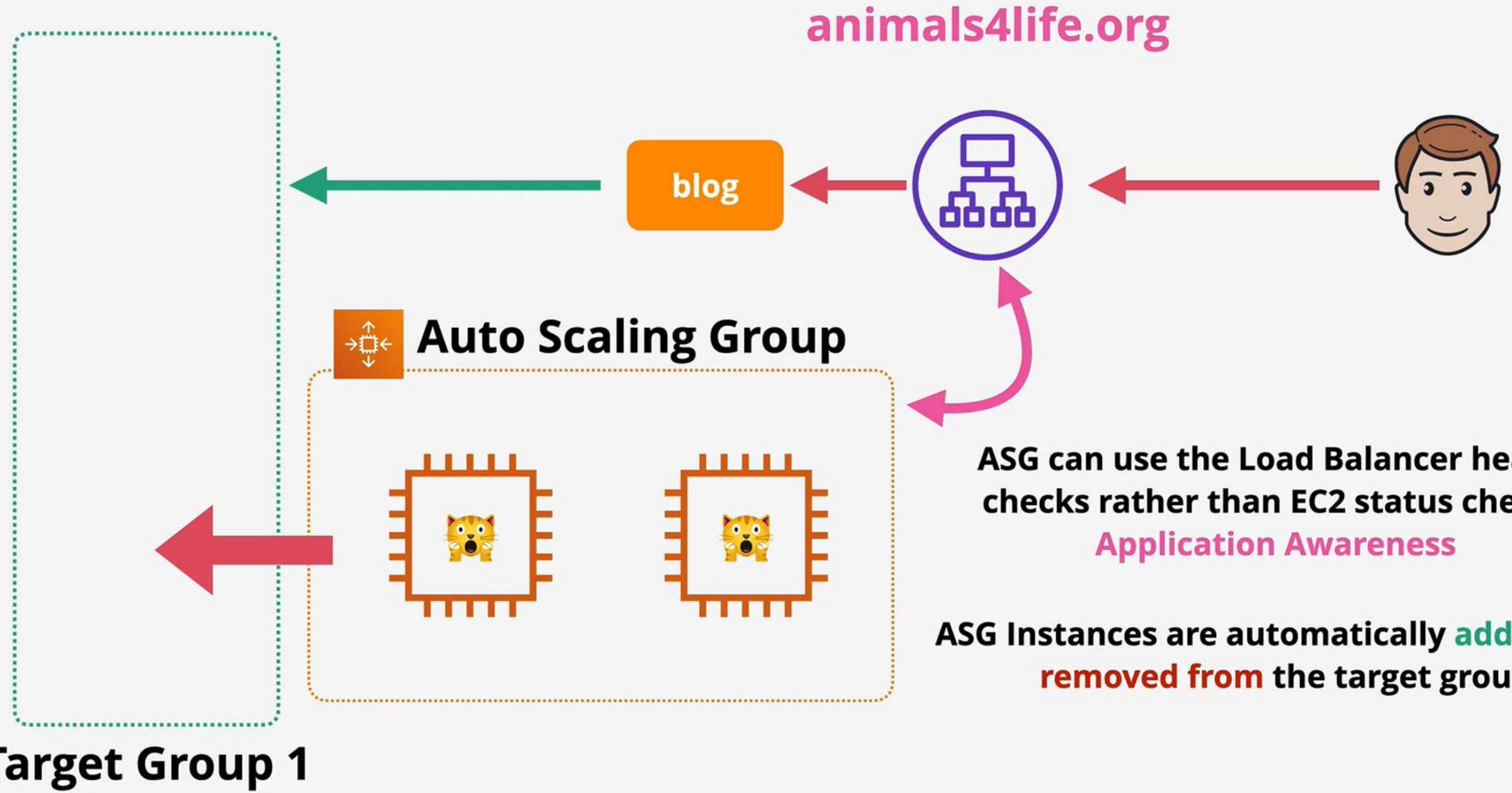
Final points

- Autoscaling Groups are free
- Only the resources created are billed ...
- Use cool downs to avoid rapid scaling
- Think about **more**, **smaller** instances - **granularity**
- Use with ALB's for elasticity - **abstraction**
- ASG defines **WHEN** and **WHERE**, LT defines **WHAT**

Scaling Processes

- **Launch** and **Terminate** - SUSPEND and RESUME
- **AddToLoadBalancer** - add to LB on launch
- **AlarmNotification** - accept notification from CW
- **AZRebalance** - Balances instances evenly across all of the AZs
- **HealthCheck** - instance health checks on/off
- **ReplaceUnhealthy** - Terminate unhealthy and replace
- **ScheduledActions** - Scheduled on/off
- **Standby** - use this for instances 'InService vs Standby'

ASG + Load Balancers

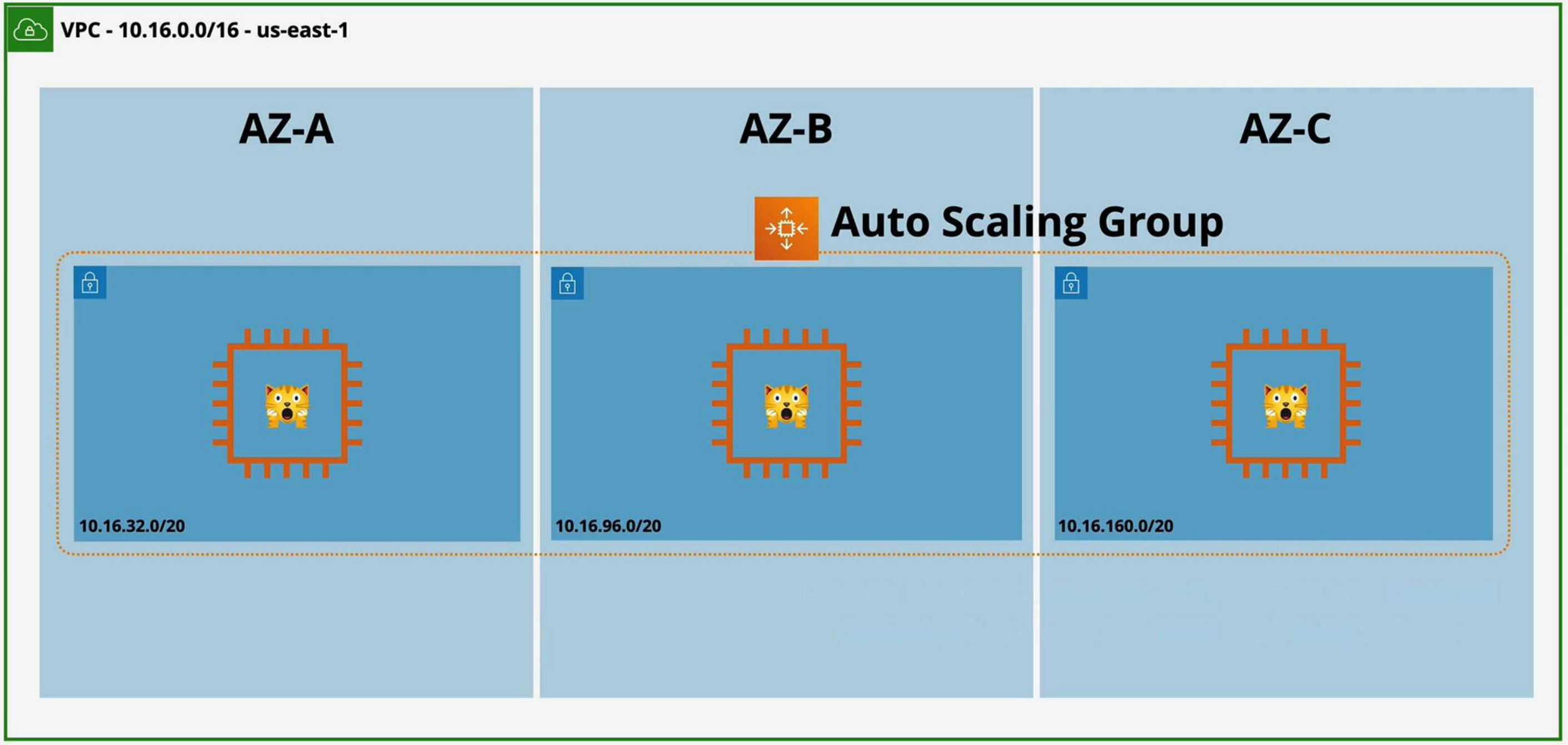


Scaling Policies

- **Manual** Scaling - Manually adjust the desired capacity
- **Scheduled** Scaling - Time based adjustment - e.g. Sales..
- **Dynamic** Scaling
  - **Simple** - “CPU above 50% +1”, “CPU Below 50 -1”
  - **Stepped** Scaling - Bigger +/- based on difference
  - **Target Tracking** - Desired Aggregate CPU = 40% ..ASG handle it
- **Cooldown Periods** ...

Auto Scaling Groups Architecture

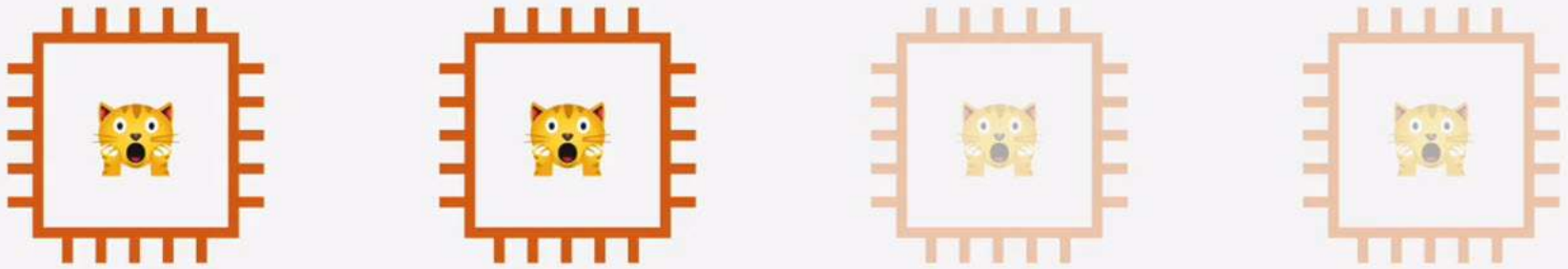
https://learn.cantrill.io adriancantrill



Auto Scaling Groups

Launch Template
provides EC2 Config

Auto Scaling Group



Minimum Size (1)

Desired Capacity (2)

Maximum Size (4)

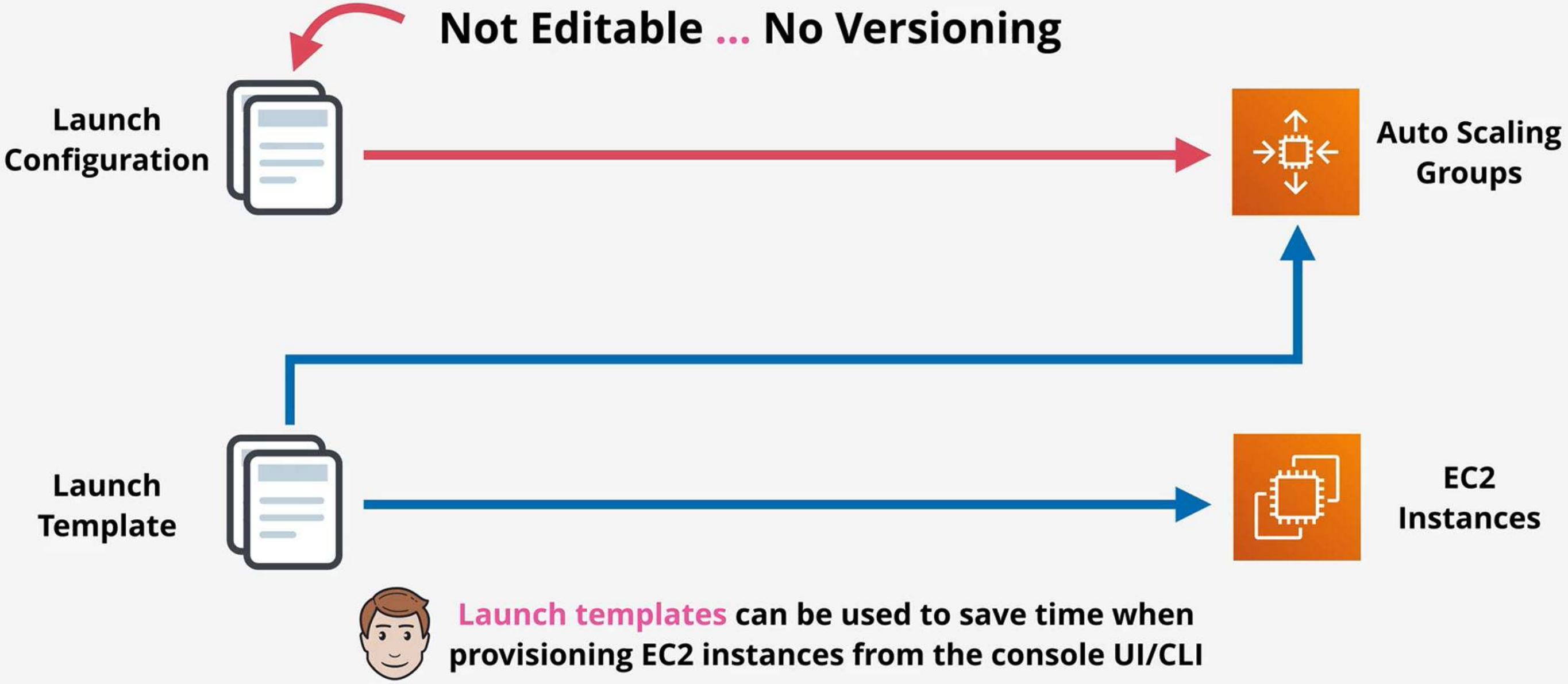
Scaling Policies automatically adjust the Desired Capacity between the MIN and MAX values

Auto Scaling Groups

- **Automatic Scaling** and ***Self-Healing*** for EC2
- Uses **Launch Templates** or **Configurations**
- Has a **Minimum**, **Desired** and **Maximum** Size (e.g 1:2:4)
- Keep running instances at the **Desired capacity** by **provisioning** or **terminating** instances
- **Scaling Policies** automate based on metrics

Auto Scaling Groups

LC and LT Architecture



LC and LT Key Concepts

- Allow you to define the configuration of an EC2 instance **in advance**
- AMI, Instance Type, Storage & Key pair
- Networking and Security Groups
- Userdata & IAM Role
- Both are NOT editable - defined once. LT has versions.
- LT provide **newer features** - including T2/T3 Unlimited, Placement Groups, Capacity Reservations, Elastic Graphics

Launch Configurations and Launch Templates

ALB vs NLB

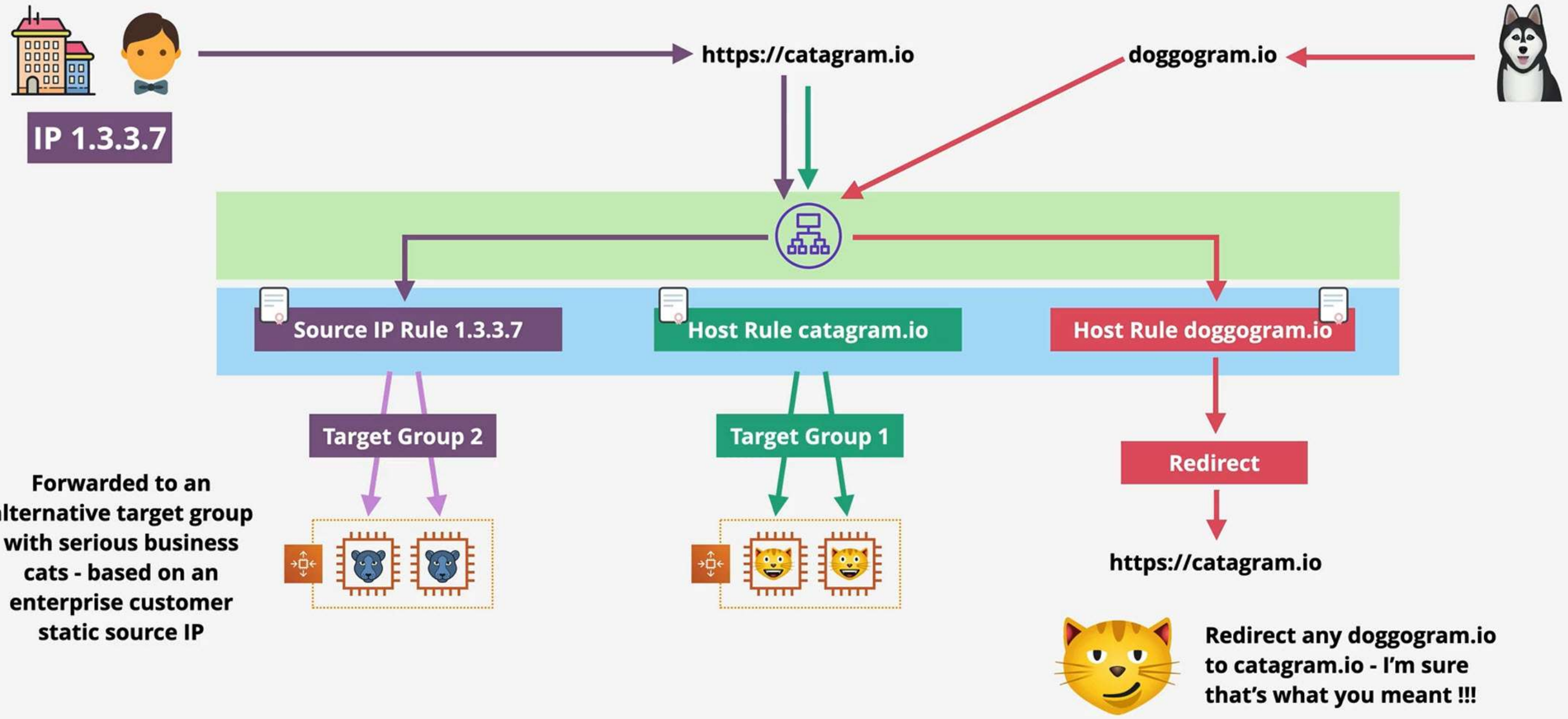
- Unbroken encryption ... NLB
- Static IP for whitelisting ... NLB
- The fastest performance ... NLB (millions rps)
- Protocols not HTTP or HTTPS ... NLB
- Privatelink ... NLB
- Otherwise ... ALB

Network Load Balancer (NLB)

- **Layer 4 load balancer** ... **TCP**, **TLS**, **UDP**, **TCP_UDP**
- **No visibility** or **understanding** of **HTTP** or **HTTPS**
- **No headers**, **no cookies**, **no session stickiness**
- Really Really Really Fast (**millions of rps**, **25%** of **ALB latency**)
- .. SMTP, SSH, Game Servers, financial apps (**not http/s**)
- Health checks **JUST** check ICMP / TCP Handshake .. **Not app aware**
- NLB's can have **static IP's** - useful for whitelisting
- **Forward TCP** to instances ... **unbroken encryption**
- Used with private link to provide services to other VPCs

Application Load Balancer (ALB) - Rules

https://learn.cantrill.io adriancantrill



Application Load Balancer (ALB) - Rules

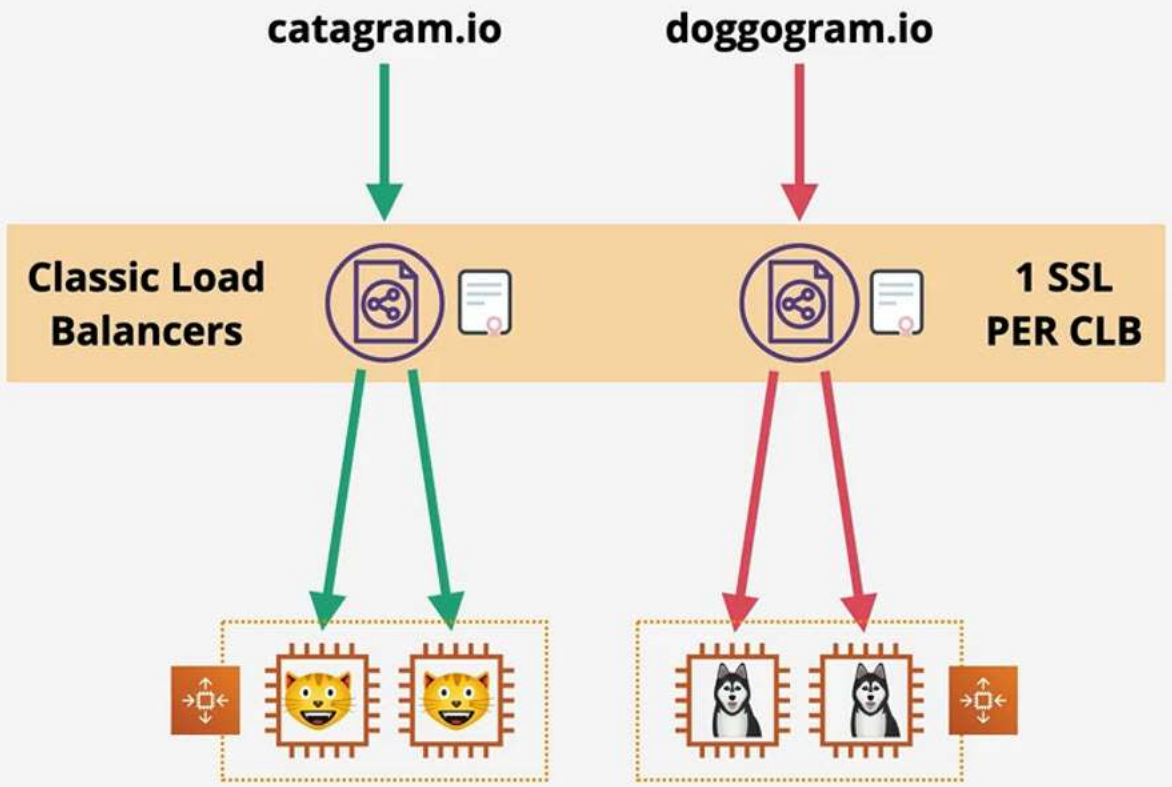
- Rules **direct connections** which **arrive** at a **listener**
- Processed in **priority order**
- **Default rule** = **catchall**
- **Rule Conditions**: host-header, http-header, http-request-method, path-pattern, query-string & source-ip
- **Actions** : forward, redirect, fixed-response, authenticate-oidc & authenticate-cognito

Application Load Balancer (ALB)

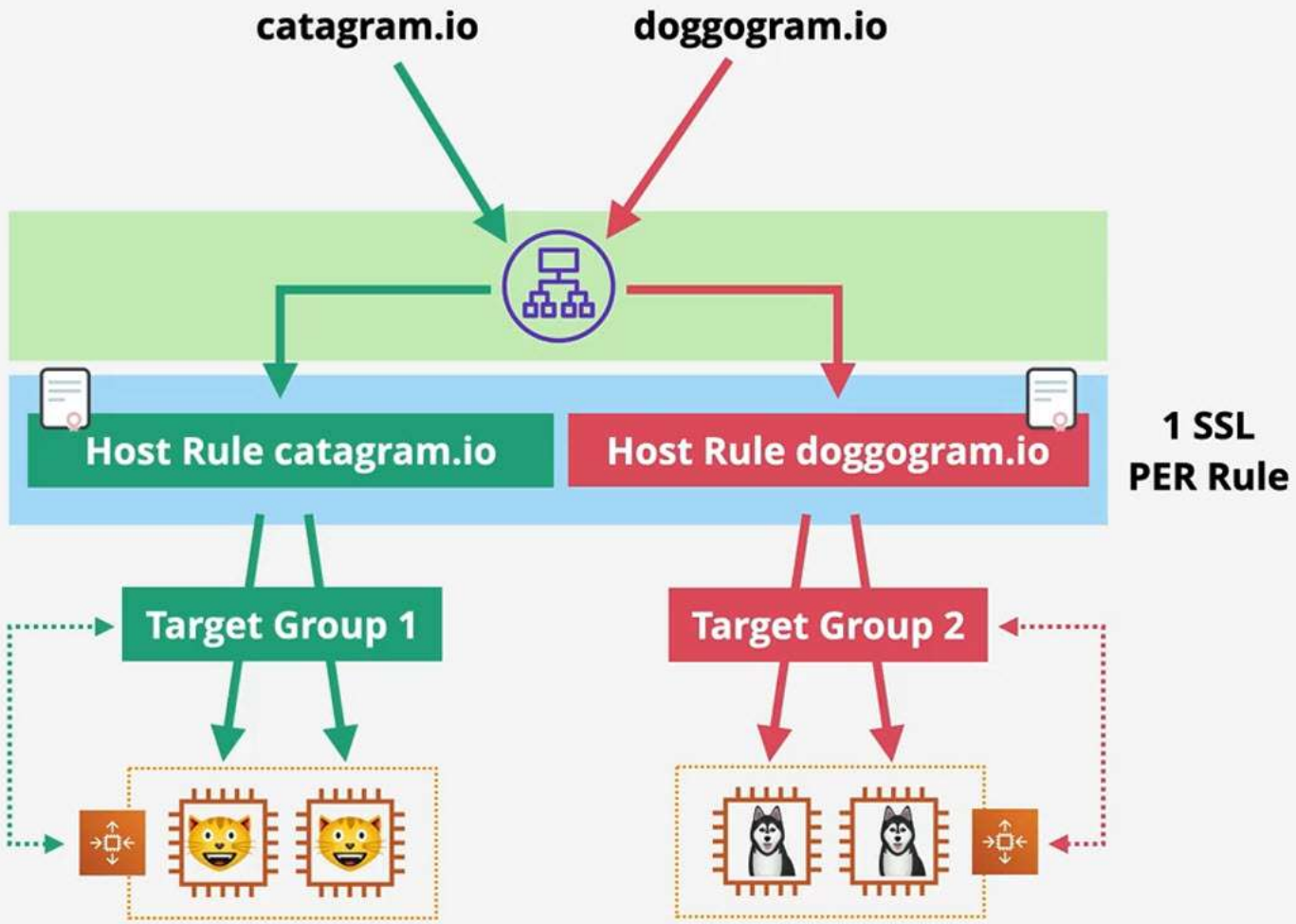
- **Layer 7** Load balancer .. listens on **HTTP** and/or **HTTPS**
- **No other Layer 7 protocols** (SMTP, SSH, Gaming ....)
- ....and **NO TCP/UDP/TLS Listeners**
- L7 content type, cookies, custom headers, user location and app behaviour
- HTTP HTTPS (SSL/TLS) always terminated on the ALB - **no unbroken SSL** (security teams!)
- ....**a new connection** is made to the application
- ALBs **MUST** have **SSL** certs if **HTTPS** is used
- ALBs are **slower** than **NLB** .. more levels of the network stack to process
- Health checks **evaluate application health** ... layer 7

Load Balancer Consolidation

https://learn.cantrill.io adriancantrill

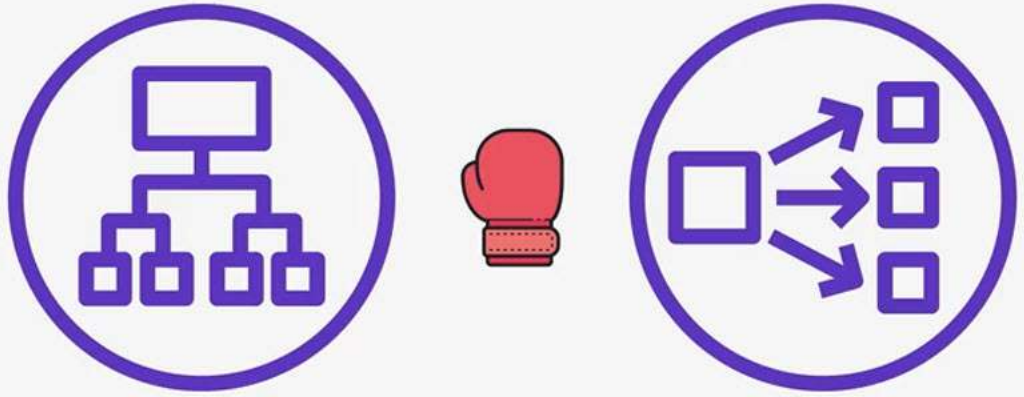


CLBs don't scale ... every unique HTTPS name requires an individual CLB because SNI isn't supported



v2 load balancers support rules and target groups. Host based rules using SNI and an ALB allows consolidation

Application and Network Load Balancer (ALB vs NLB)



ELB Architecture

- ELB is a **DNS A** Record pointing at **1+** Nodes per AZ
- Nodes (in one subnet per AZ) can scale
- **Internet-facing** means nodes have **public IPv4 IPs**
- **Internal** is **private only IPs**
- EC2 **doesn't need to be public** to work with a LB
- **Listener** Configuration controls **WHAT** the LB does
- **8+** Free IPs per subnet, and **/27** subnet to allow scaling

CROSS-ZONE LB

https://learn.cantrill.io adriancantrill

Each node gets 100% / Number of Nodes
e.g. 50% each in this example

LB DNS NAME

VPC - 10.16.0.0/16 - us-east-1

LB NODE

LB NODE

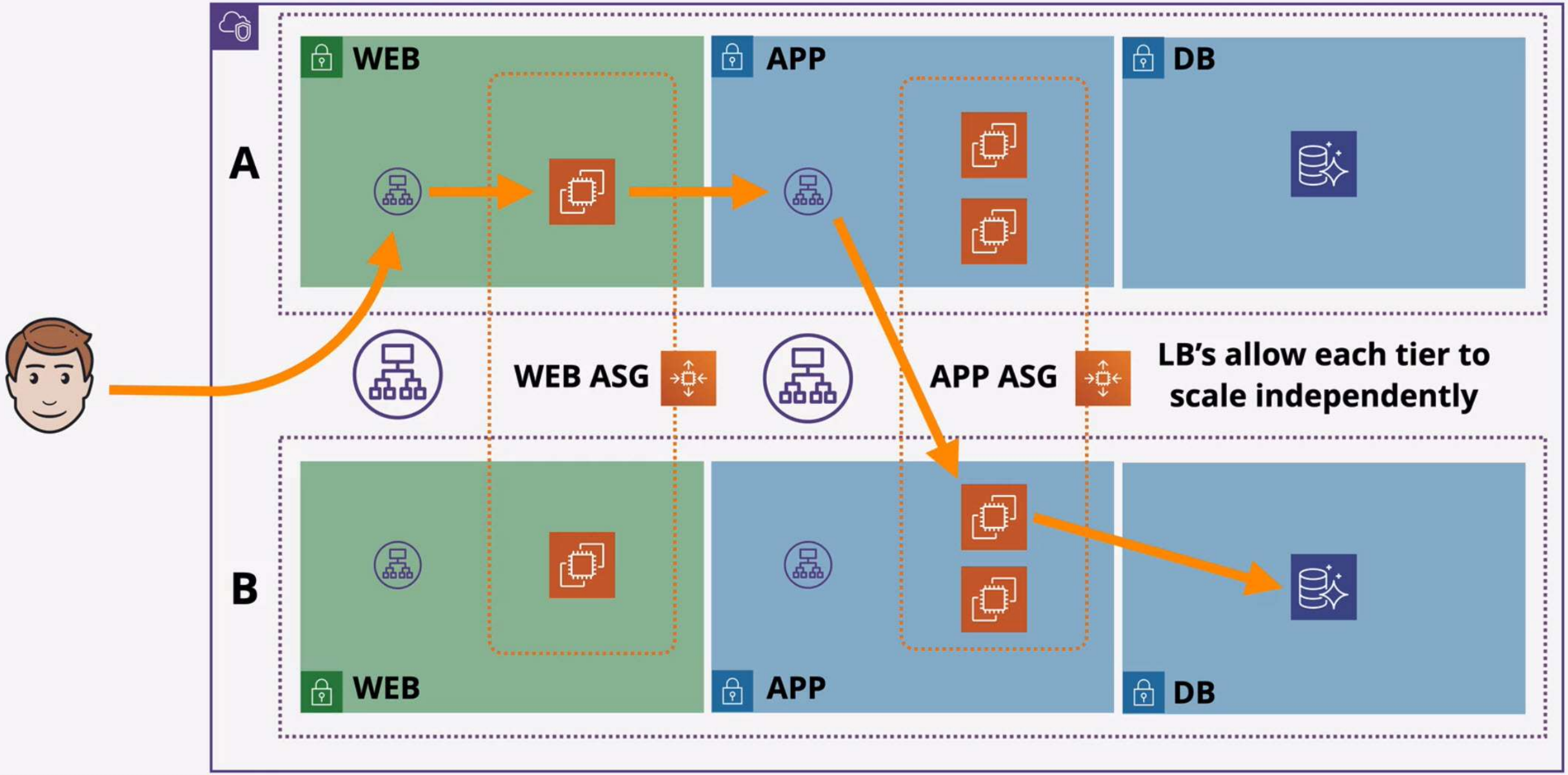
Cross-Zone Load Balancing

AZA

AZB

ELB Architecture

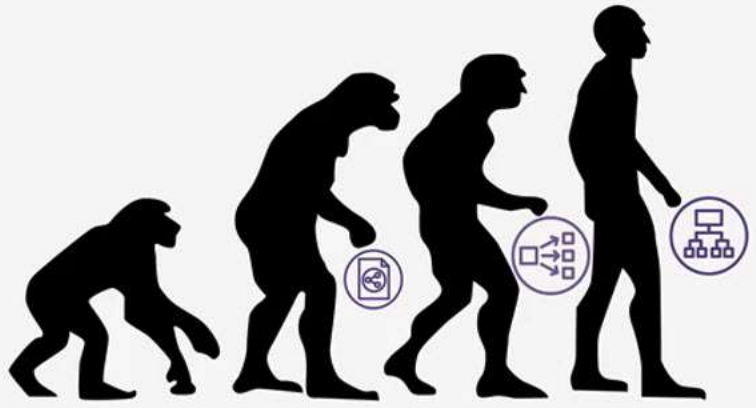
https://learn.cantrill.io adriancantrill

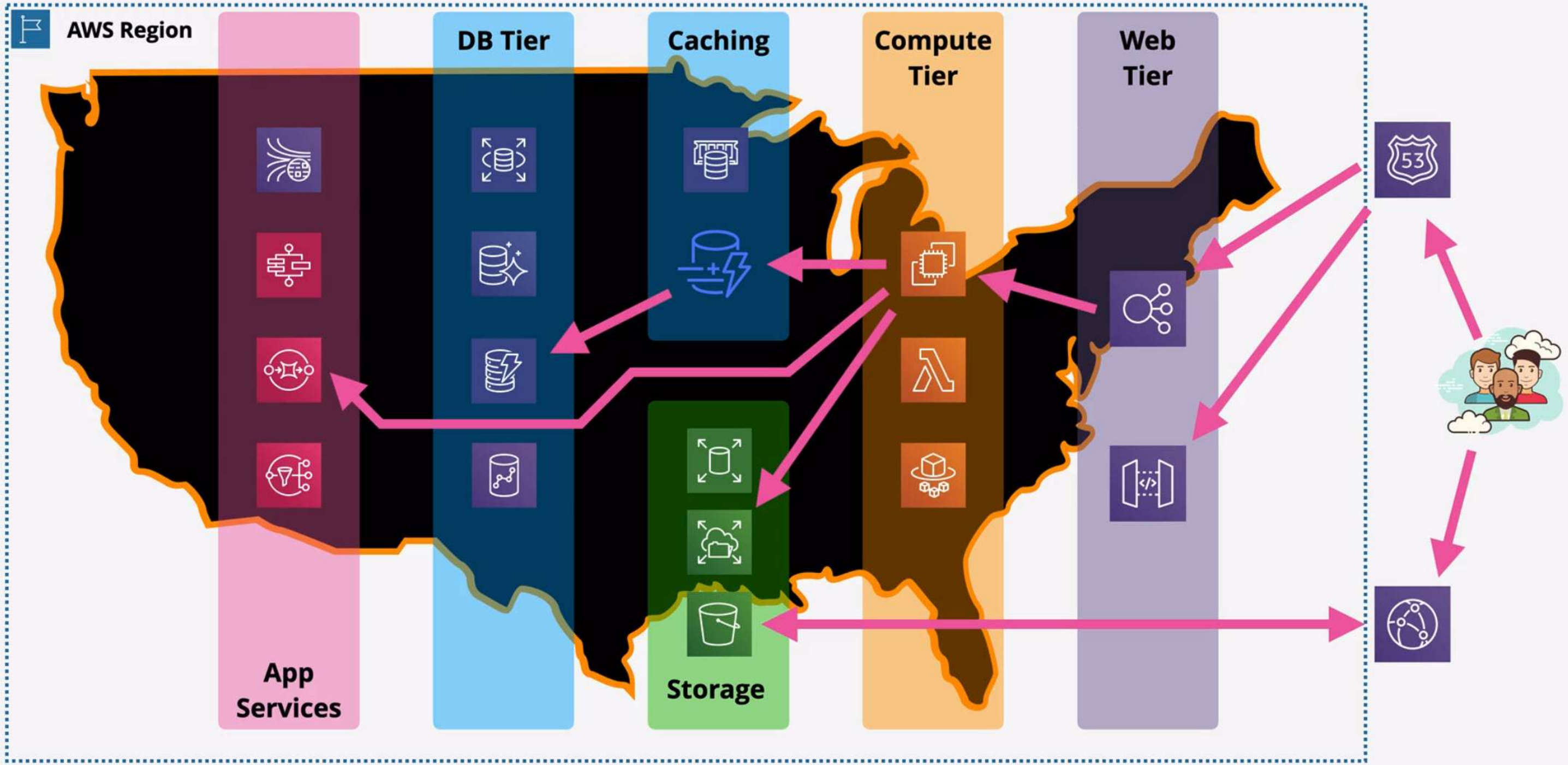


ELB Evolution

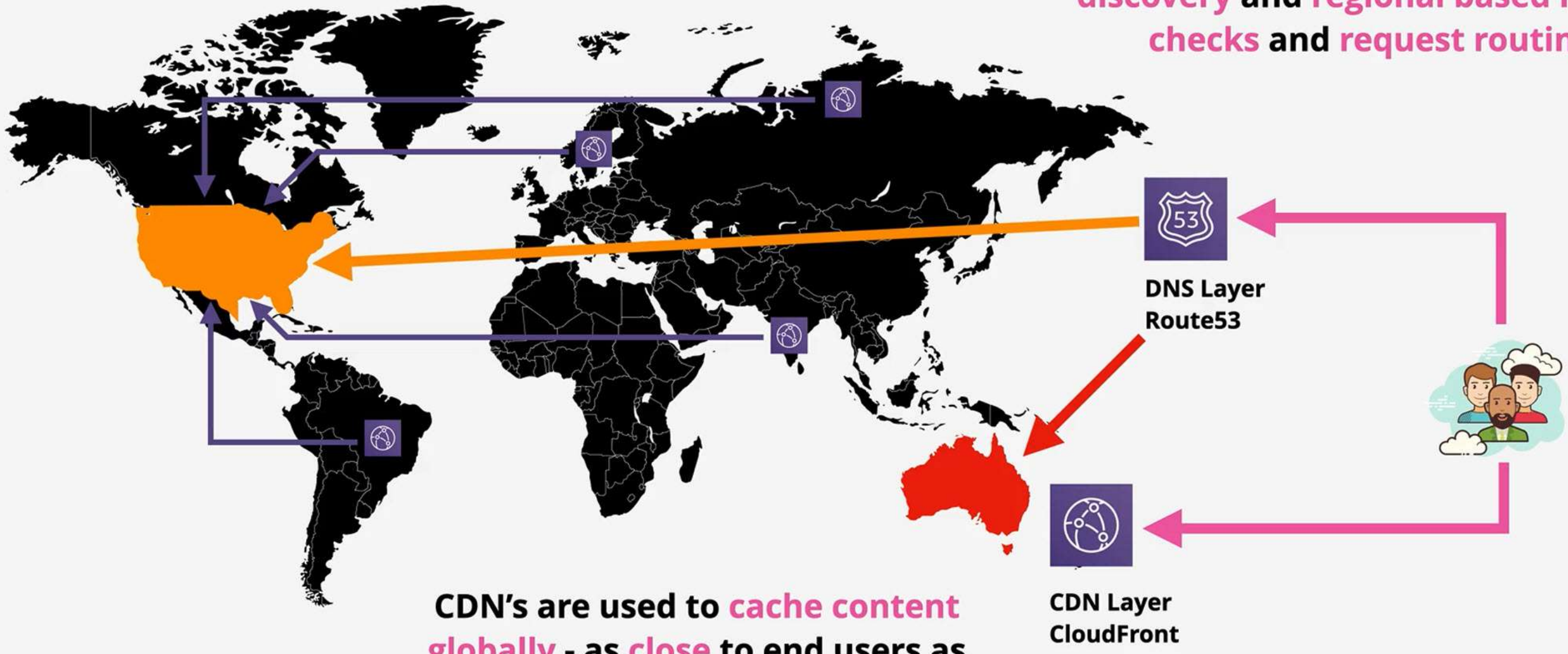
- 3 Types of load balancers (**ELB**) available within AWS
- Split between **v1** (**avoid** / **migrate**) and **v2** (**prefer**)
- Classic Load Balancer (**CLB**) - **v1** - Introduced in 2009
- Not really layer 7, lacking features, **1 SSL per CLB**
- Application Load Balancer (**ALB**) - v2 - HTTP/S/WebSocket
- Network Load Balancer (**NLB**) - v2 - TCP, TLS & UDP
- V2 = faster, cheaper, support target groups and rules

Evolution of Elastic Load Balancers (ELB)



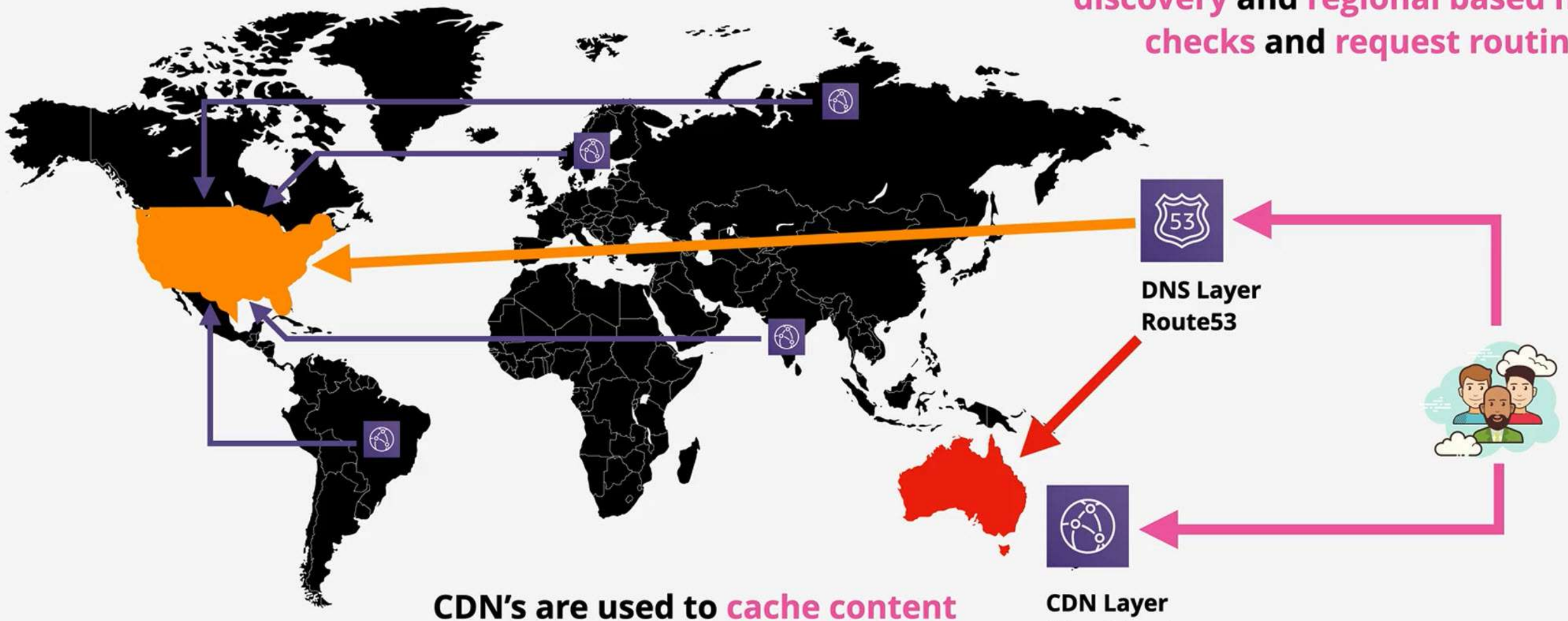

AWS Region
App Services
DB Tier
Caching
Compute Tier
Web Tier
Storage
53

Globally DNS is used for service discovery and regional based health checks and request routing



CDN's are used to cache content globally - as close to end users as possible to improve performance

Globally DNS is used for service discovery and regional based health checks and request routing



CDN's are used to cache content globally - as close to end users as possible to improve performance

customers will be directed towards a region

Regional and Global AWS Architecture

- Global **Service Location** & **Discovery**
- Content Delivery (**CDN**) and optimisation
- Global **health checks** & **Failover**
- Regional **entry point**
- **Scaling** & **Resilience**
- Application services and **components**