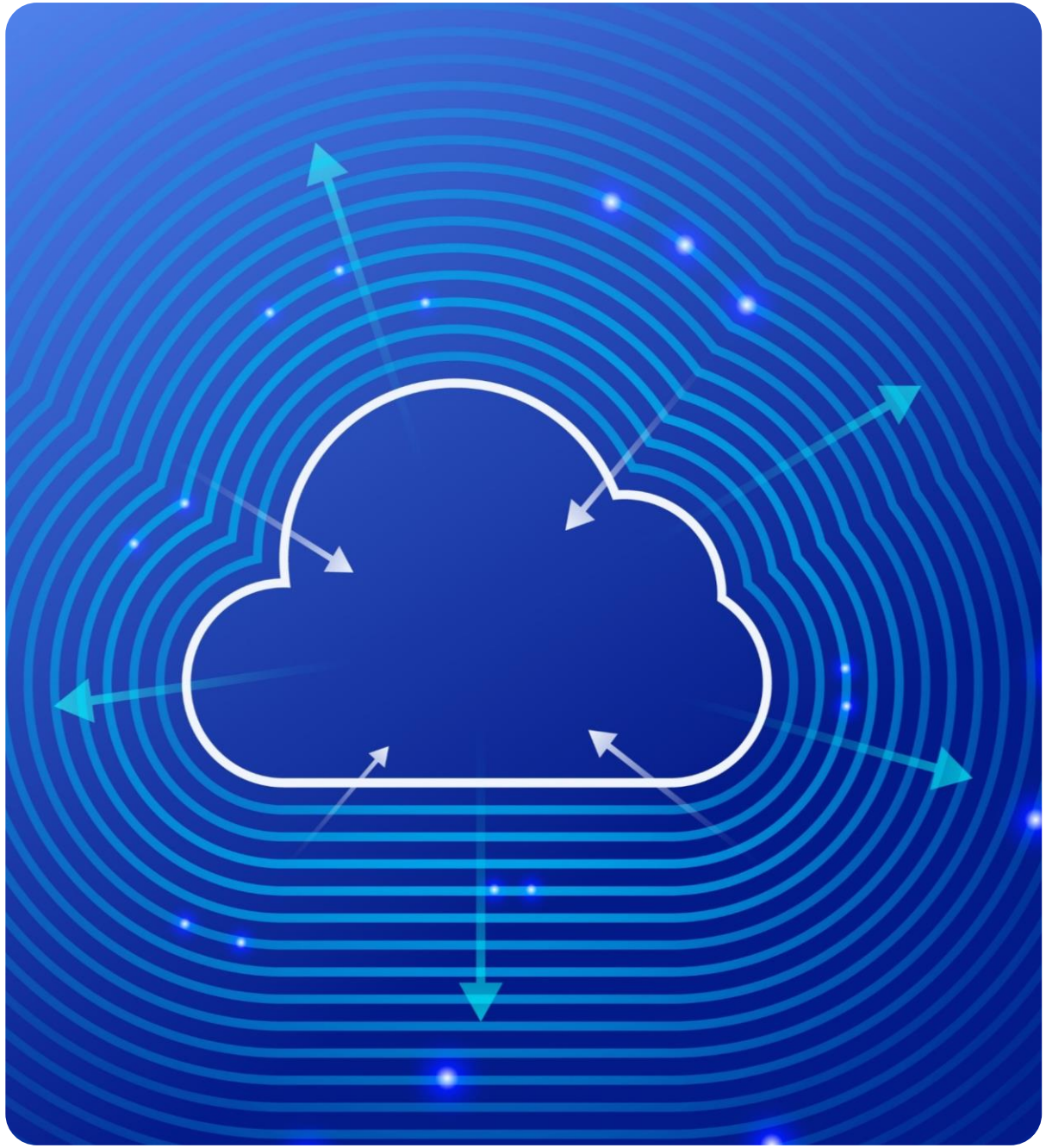
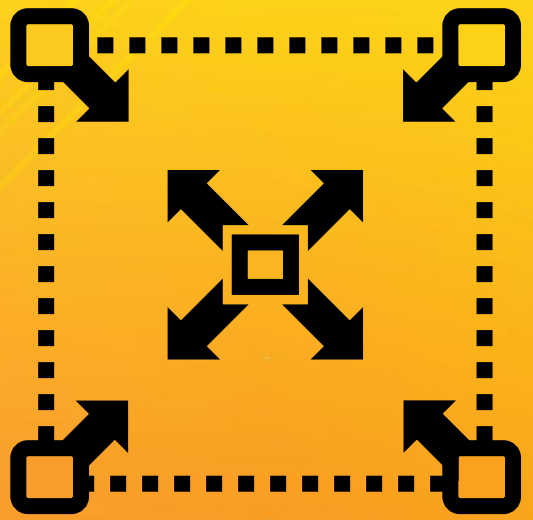
Scalability and Elasticity in Cloud

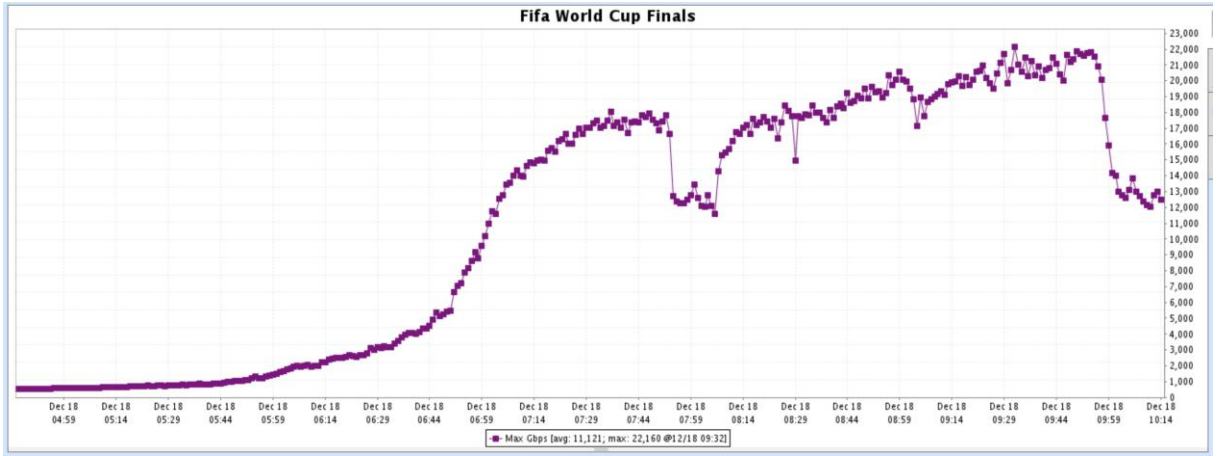




Scalability vs. Elasticity

Scalability vs. Elasticity – in Context of AWS

- Scalability – AWS expanding its global foot print
  - 2006 - 2011 4 Regions
  - 2011 - 2016 7 Regions
  - 2016 - 2018 11 Regions
  - 2018 - 2023 31 Regions
  - 2008 14 Edge Locations
  - 2018 150 Edge Locations
  - 2023 400+ Edge Locations
- Elasticity – Amazon CloudFront – FIFA World Cup 2022
  - Scaling up to support millions of concurrent viewers
  - Delivered more than 150 petabytes of streaming media content
  - Peaking at around 23 Terabits-per-second across
  - To 49 million unique client IP addresses

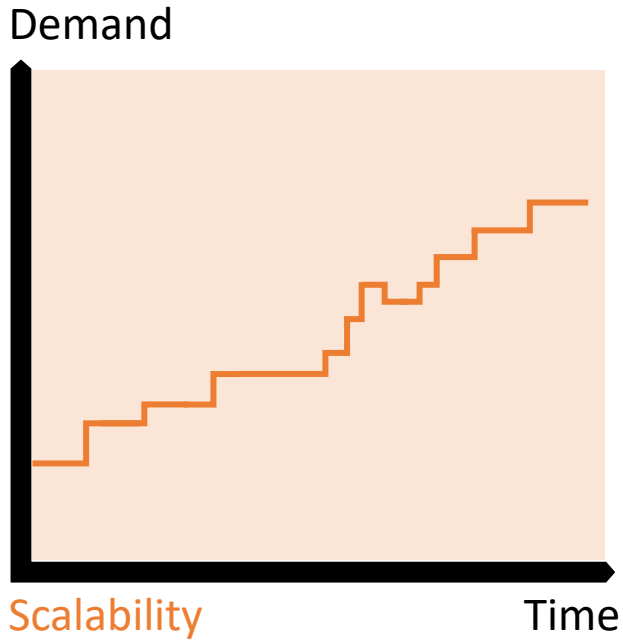


What is Scalability?

- Scalability is the ability of a system to grow or shrink in a graceful manner.

- Scalability describes a design or architecture that is growth friendly.
  - Some examples
    - How efficiently a system performs when user traffic is increased multifold?
    - How well a database responds to a higher volume of queries?
    - How an operating system performs on different type of hardware?

- A scalable system performs with the same stability even if the load on it increases.

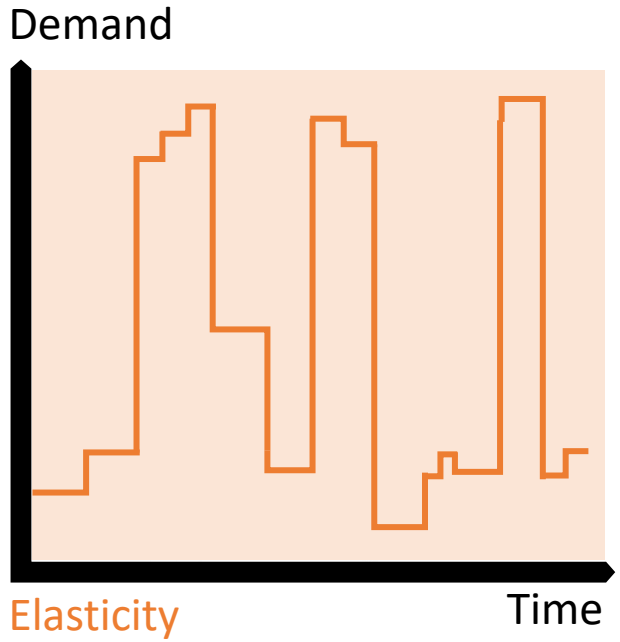


What is Elasticity?

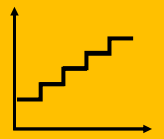
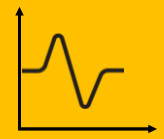
- Elasticity generally refers to dynamic increasing or decreasing of resources.

- An elastic system automatically adapts to match resources with demand as closely as possible, in real time.
  - Some examples
    - Adding more instances whenever a web application gets a lot of traffic.
    - Handling peak traffic like Black Friday sale
    - Scale infrastructure into for test and development activities and tear it down once test/dev work is complete

- Scalability gives you the ability to increase or decrease your resources, and elasticity lets those operations happen automatically according to configured rules.



Scalability vs. Elasticity

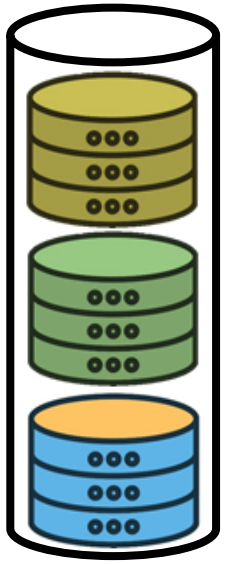
	Scalability 	Elasticity 
What is it?	Scalability is the ability of a system to uphold the functionality when the size or volume changes.	Elasticity is the ability to dynamically manage available resources for addressing the size or volume.
Use case	To meet the static/predictable increase in the workload.	To meet the dynamic/sudden increase in the workload.
Type	Strategic operation	Tactical approach
Focuses on	Design/architecture	Operations
Resource provisioning	To exceed future demands	To meet present demand
Consideration	Medium- and long-term predictions	Short-term demand
Execution by	Typically scheduled	Typically triggered by automation

Achieving Scalability

- Scale Up / Down (Vertical Scaling)
  - Replace the resource with a bigger size or add more capacity in the same resource
    - Suitable for – Stateful workloads

- Scale Out / In (Horizontal Scaling)
  - Adding more resource to our architecture and spread the workload across those resources.
    - Suitable for – Stateless workloads

Scale Up Storage

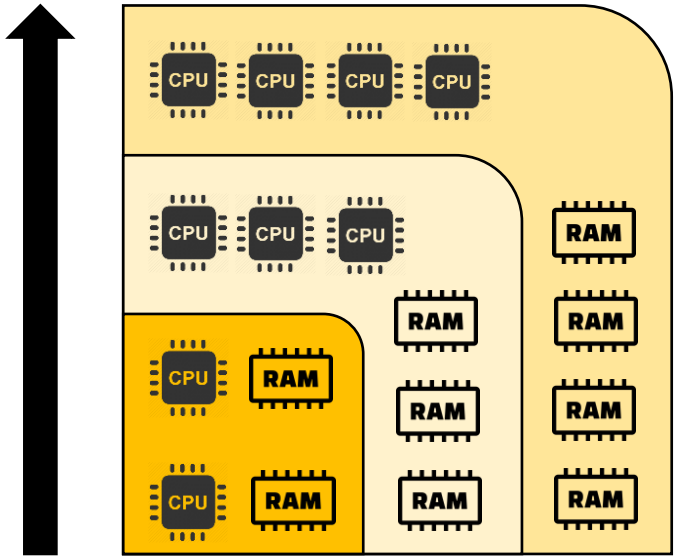


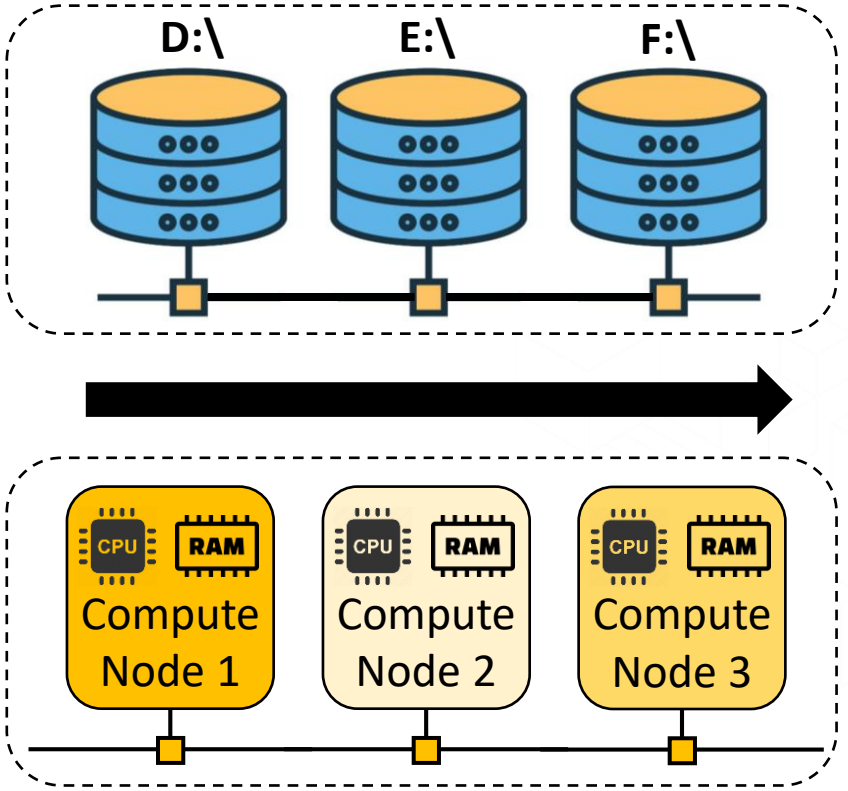
Additional storage

Additional storage

Initial storage

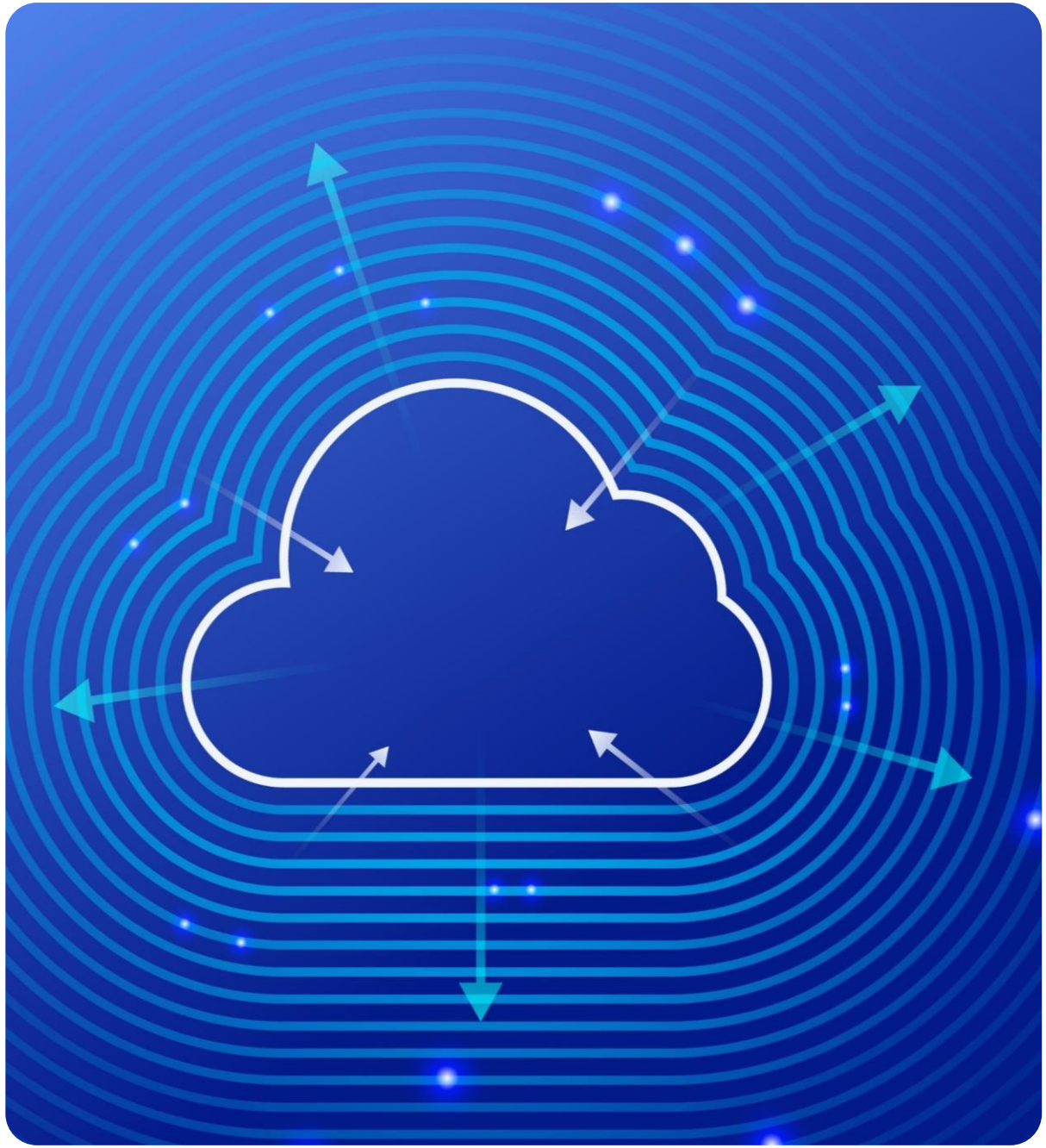
Scale Up Database Server

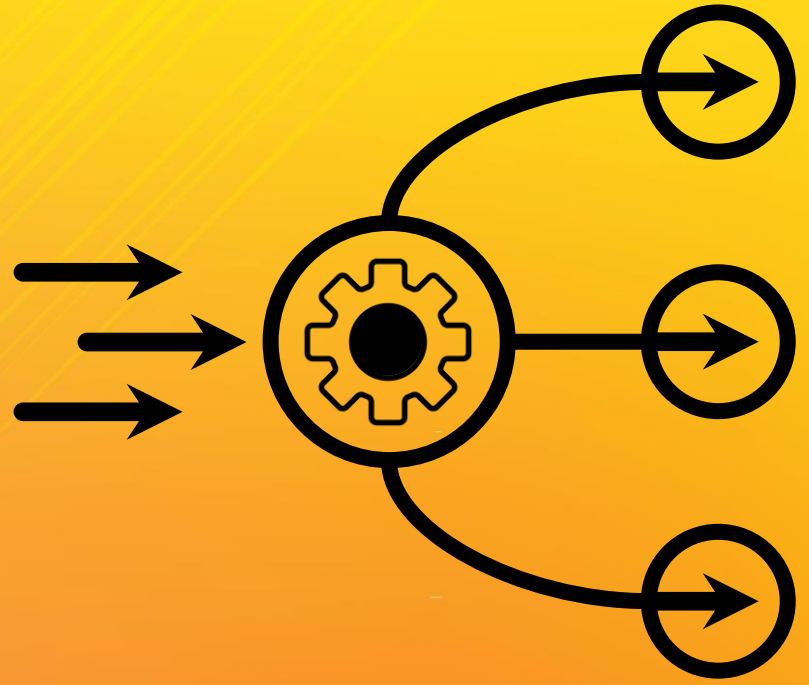




Scale Out Storage

Scale Out Compute Node





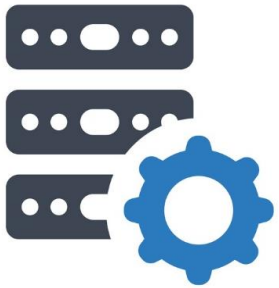
Load Balancing

Load Balancing

- Load balancing is the method of distributing network traffic equally across a pool of resources that support an application.

- A load balancer is a device that sits between the user and the server group and acts as an invisible facilitator, ensuring that all resource servers are used equally.

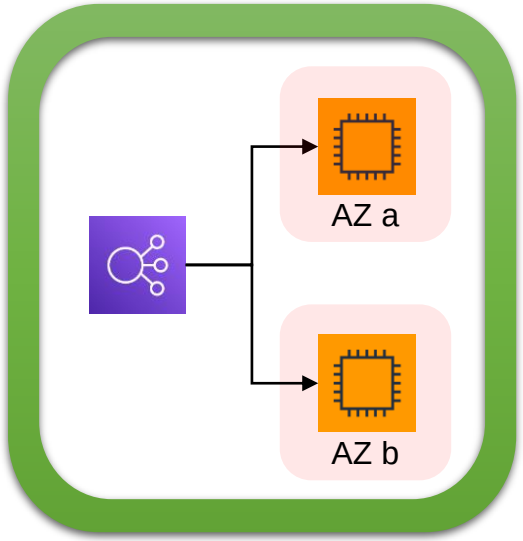
- Load balancers are of two types:



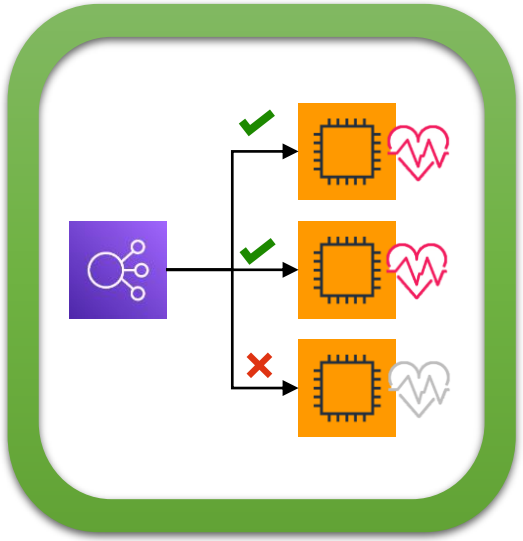
Hardware load balancers

Software load balancers

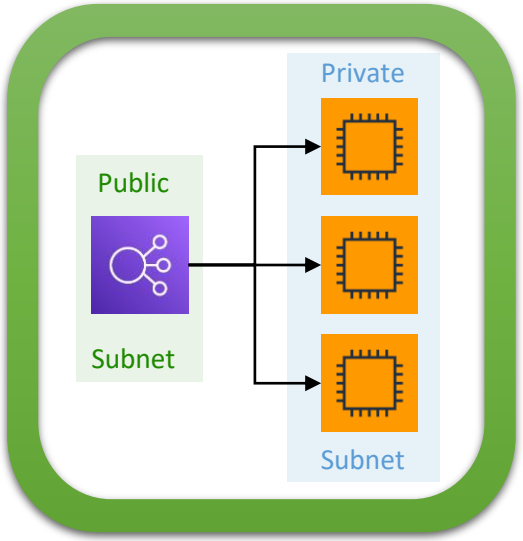
Elastic Load Balancer in AWS



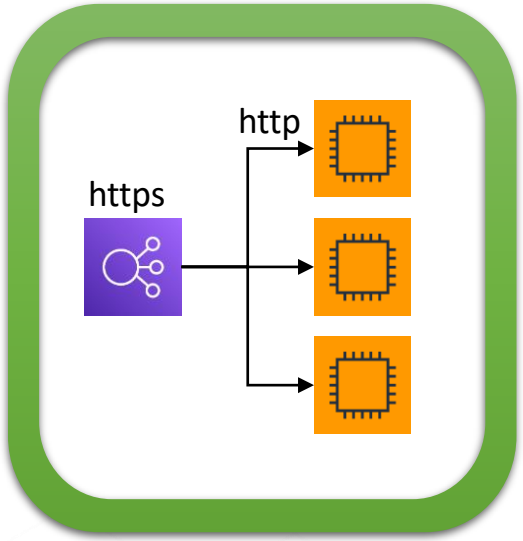
High availability



Health checks



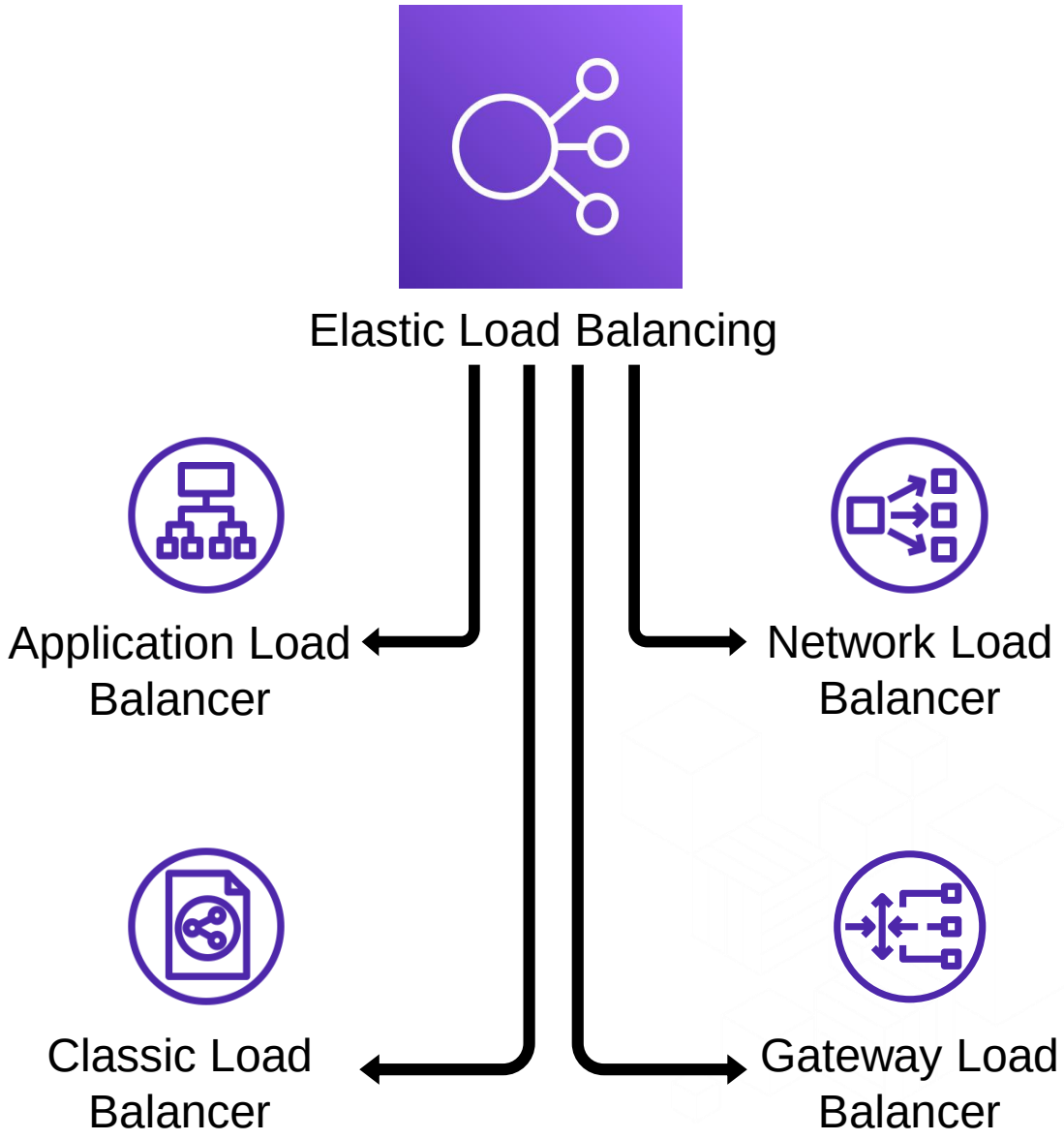
Security features



TLS termination

Load Balancing in AWS – Elastic Load Balancing

- Offered as a managed service
- Four different types of Load Balancers
- Support different target types
- Support different protocols
- No upfront cost



Load Balancer Types

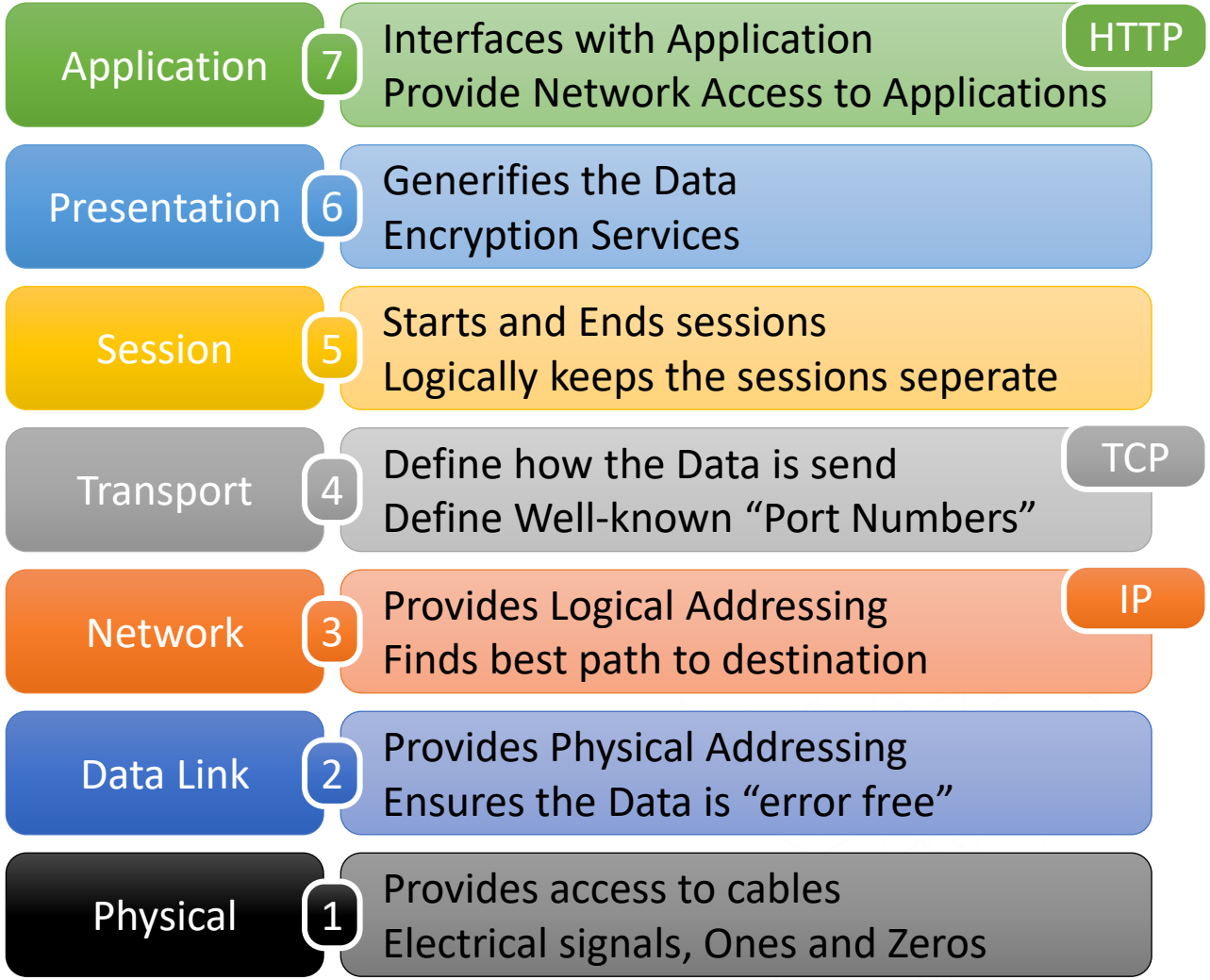
Application Load Balancer (ALB)

Networks Load Balancer (NLB)

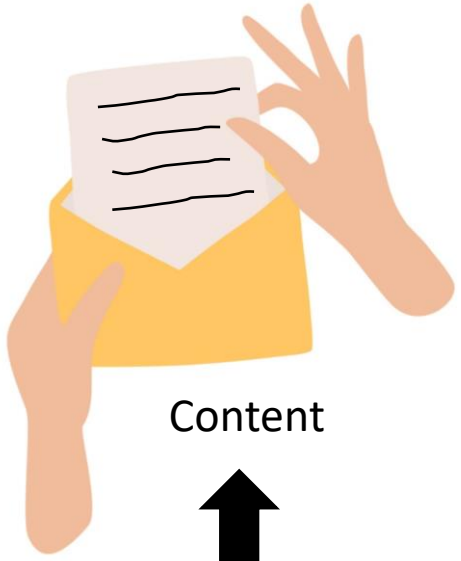
Open System Interconnection Model (OSI Model)

- In 1970 the International Standards Organization (ISO) developed the **Open Systems Interconnections** (OSI) reference model to define the basic standards for network communication.

- OSI Model is a 7 layer theoretical model
  - Reduces Complexity
  - Standardized interfaces
  - Facilities modular engineering
  - Ensure interoperability



Delivering letter to a king

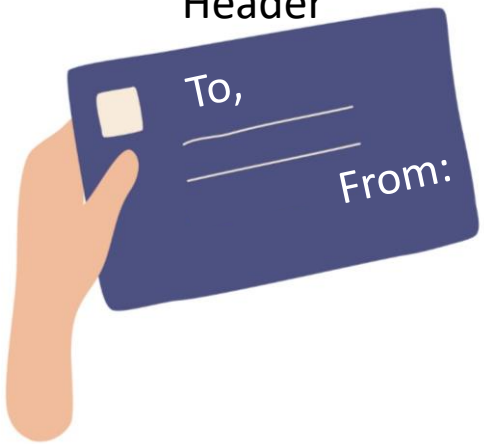


Content

Application Layer

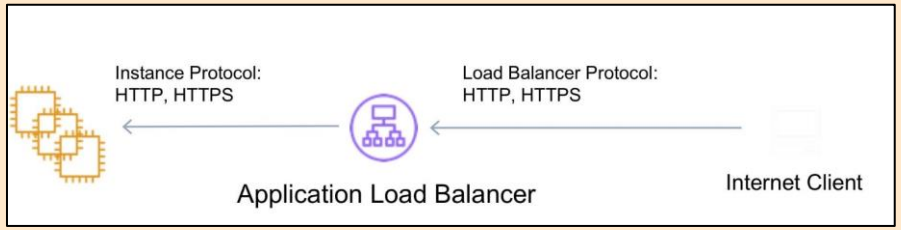
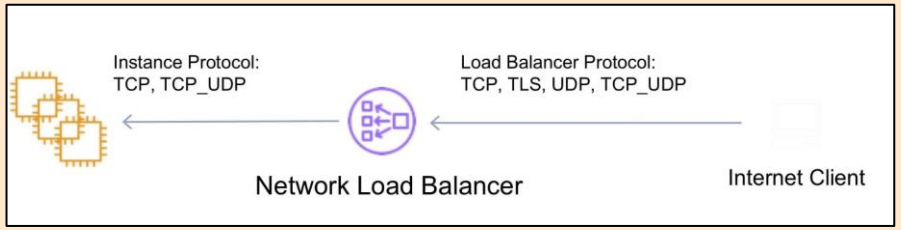
Filter based on

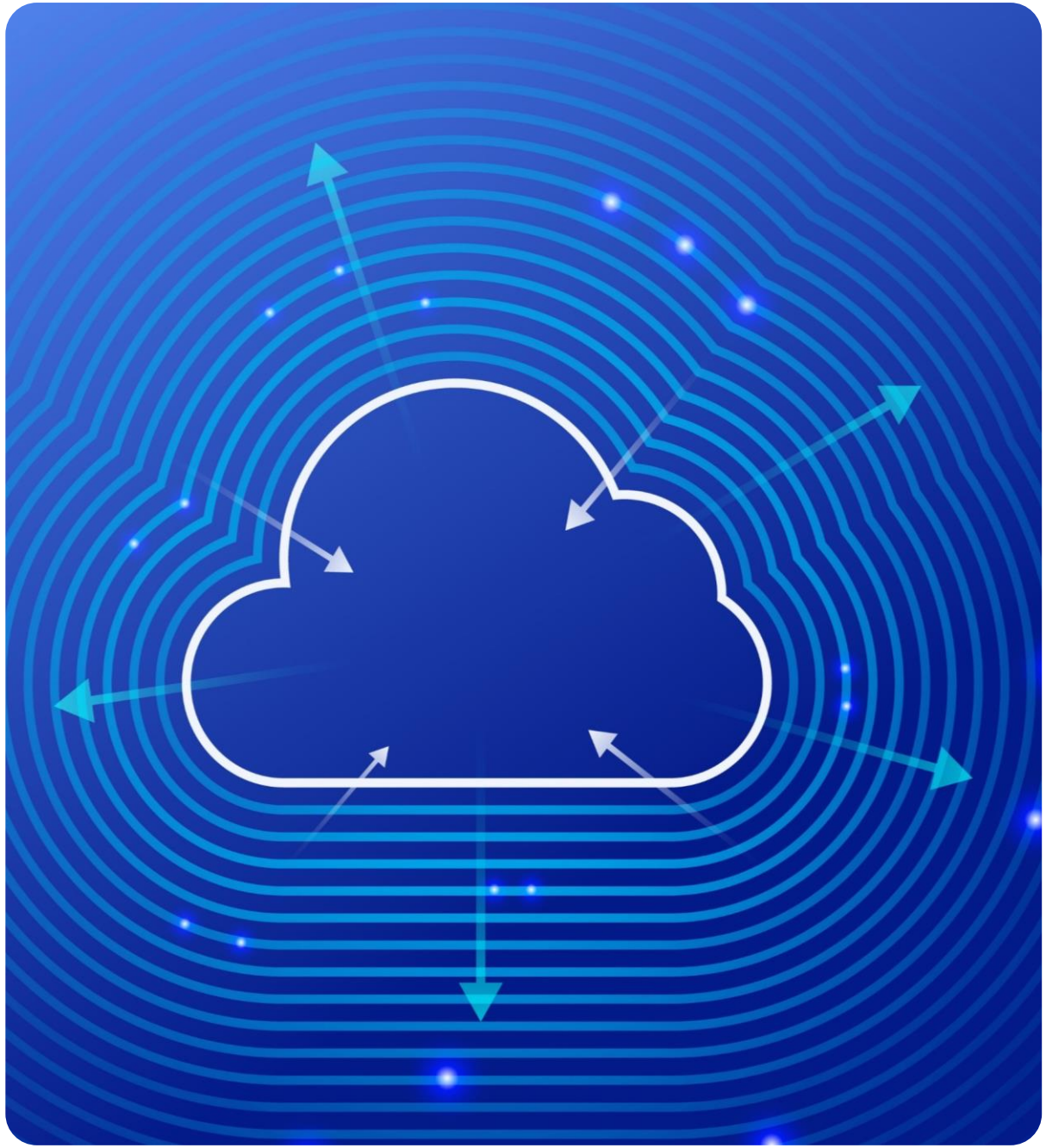
Header



Network Layer

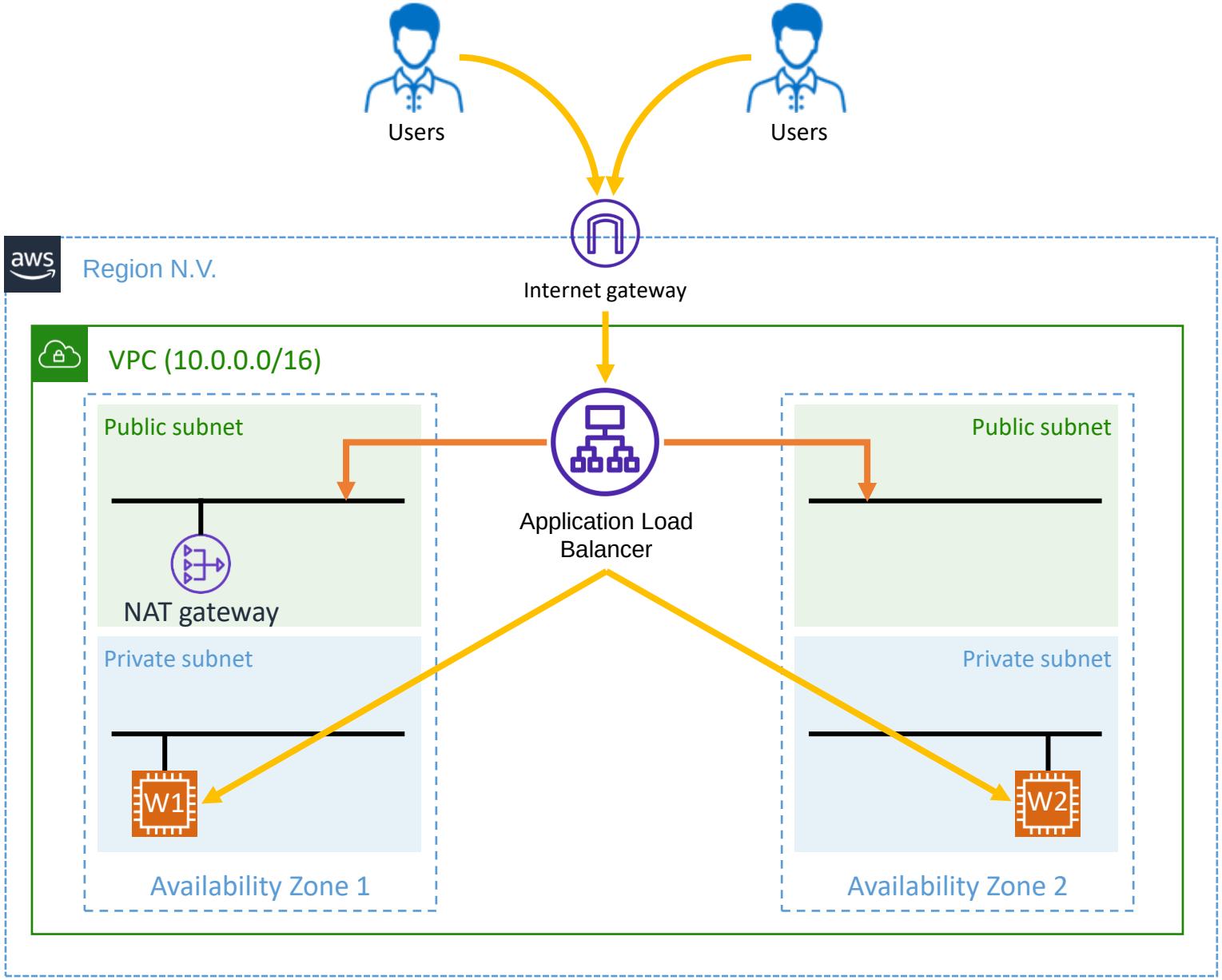
Load Balancer Types

	Application Load Balancer (ALB)	Networks Load Balancer (NLB)
Operates at	Request Level (Application Layer / Layer 7)	Connection Level (Network Layer / Layer 3)
Routes traffic based on	Content of the Packet	Header of the Packet
Supported Protocols	HTTP, HTTPS 	TCP, UDP, TLS 
Static and Elastic IP	No	Yes
Target Types	Instances, Containers, Lambda, IP Addresses	Instances, Containers, IP Addresses



Application Load Balancer (ALB)

Application Load Balancer in action



- Step 1
  - Deploy VPC
- Step 2
  - Deploy Web Servers
- Step 3
  - Deploy ALB
- Step 4
  - Test

UserData Script for Web Servers

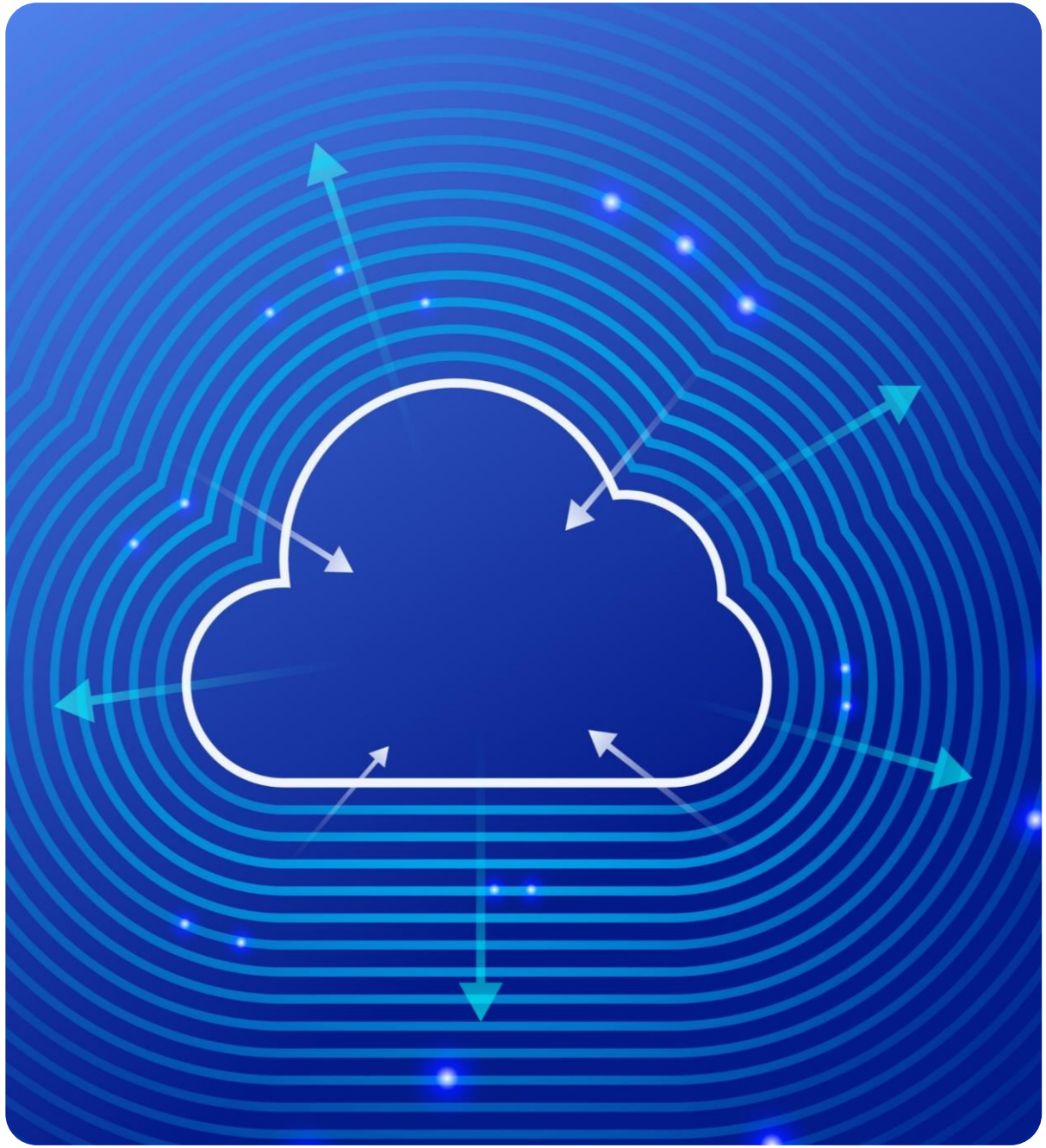
```
#!/bin/bash
yum update -y
yum install httpd -y
service httpd start
chkconfig httpd on
cd /var/www/html

echo "<html><body><h2><p><font color="red">IP Address</font></p>" > index.html
curl http://169.254.169.254/latest/meta-data/local-ipv4 >> index.html
echo "<br />"

echo "<br /> <br /> <p><font color="blue">Availablity Zone</font></p>" >> index.html
curl http://169.254.169.254/latest/meta-data/placement/availability-zone >> index.html

echo "</h2></body></html>" >> index.html
```

Application Load Balancer Terminologies

- Listener
  - A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

- Target Group
  - A target group tells a load balancer where to direct traffic to : EC2 instances, fixed IP addresses; or AWS Lambda functions, amongst others. Target groups route requests to one or more registered targets, such as EC2 instances, using the protocol and port number that you specify. You can register a target with multiple target groups.

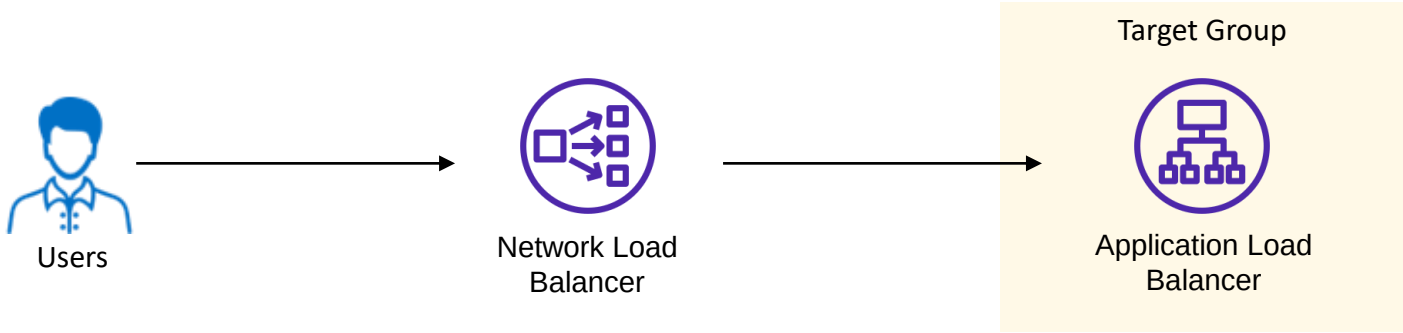
- Health Check
  - Application Load Balancer periodically sends requests to its registered targets to test their status. These tests are called health checks.

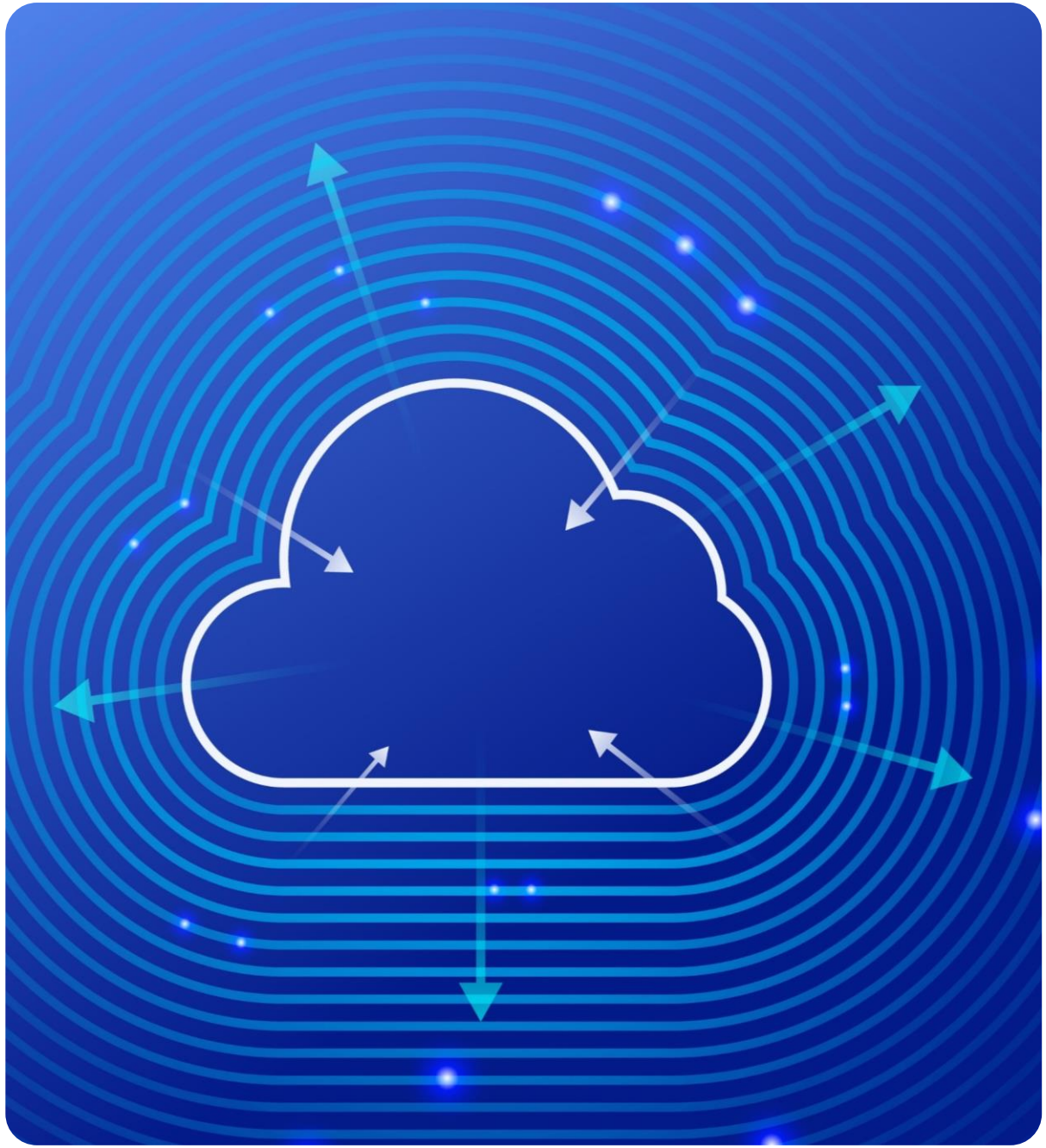


Network Load Balancer (NLB)

Network Load Balancer

- Connection-based Layer 4 load balancing
- Low latency
- PrivateLink support
- Elastic IP support
- Protocols
  - TCP, UDP, TLS
- Targets
  - Instance, Containers, IP, ALB
- Use cases
  - Chat sessions, Game connectivity, IoT service, Media and entertainment





Gateway Load Balancer (GWLB)

Third-party appliances

- Appliance = Any virtual network function (VNF) running on EC2 instance
- Example appliances
  - Load Balancers
  - NextGen firewalls
  - Intrusion detection / intrusion prevention systems (IDS / IPS)
  - DDoS mitigation
  - Visibility, performance, analytics tools
- Why Customers use it?
  - Specialized functions
  - Skill set
  - Operating processes
  - Existing investments

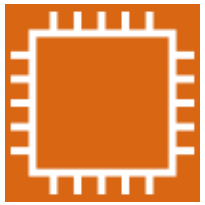
What challenges customer face?

- Fixed throughput / scale
- Single point of failure
- Difficult to manage
- Error prone configuration

Solution – Gateway Load Balancer

Source sends traffic to destination

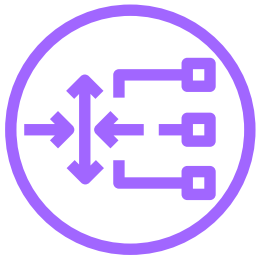
Destination



Traffic arrives at destination, appearing unchanged. The response follows the same path in revers – going back out through the same appliance.

Gateway Load Balancer

GWLB receives traffic and forwards to a healthy appliance



Third-party appliance fleet

L3 Gateway

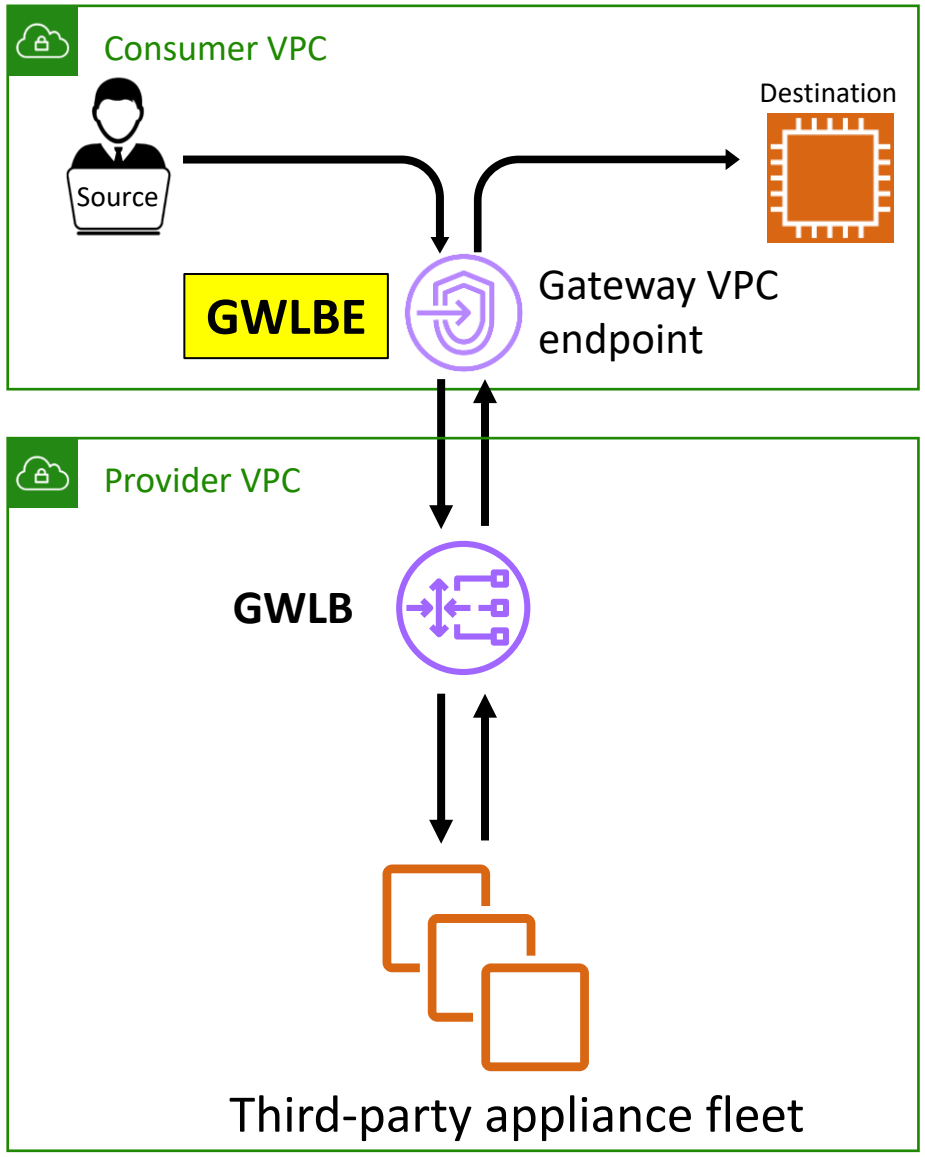
- Next-hop in route table
- No packet rewrite

L4 Load Balancer

- Scaling
- Stickiness
- Health checks
- Flow rerouting
- Encap original traffic

GENEVE - Generic Network Virtualization encapsulation

Gateway Load Balancer

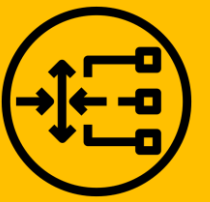


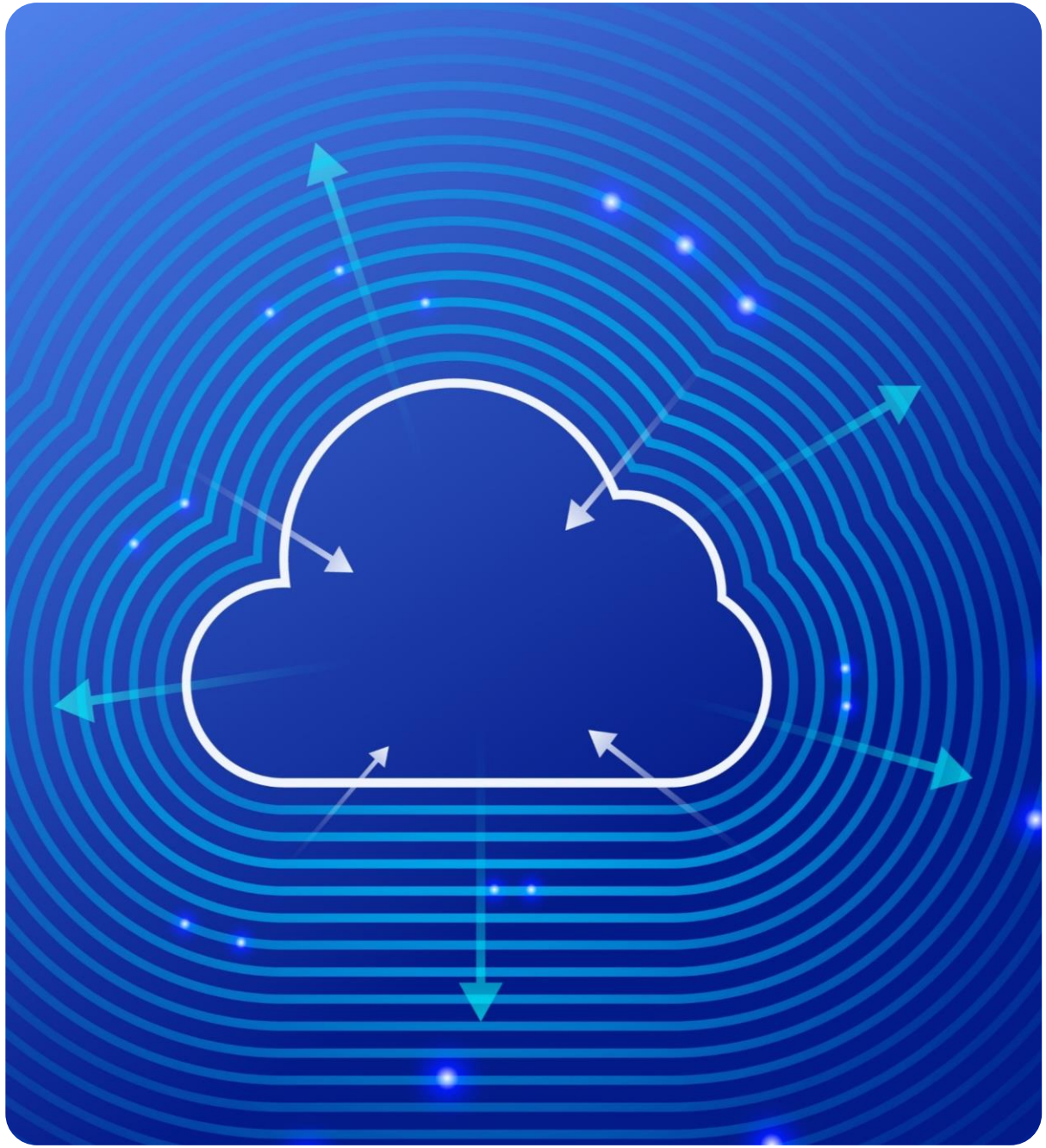
- What is GWLB/GWLBE?
  1. GWLB – Includes L3 Gateway + L4 Load Balancer capabilities
  2. **GWLBE** – VPC endpoint that can be a next-hop in route table
  3. Based on route table

- How to use GWLB?
  1. Create GWLB and appliance fleet using steps similar to ELB
  2. Send traffic to GWLB/GWLBE by updating route table

- Reference Architectures
- Integrate your custom logic

Elastic Load Balancer Types

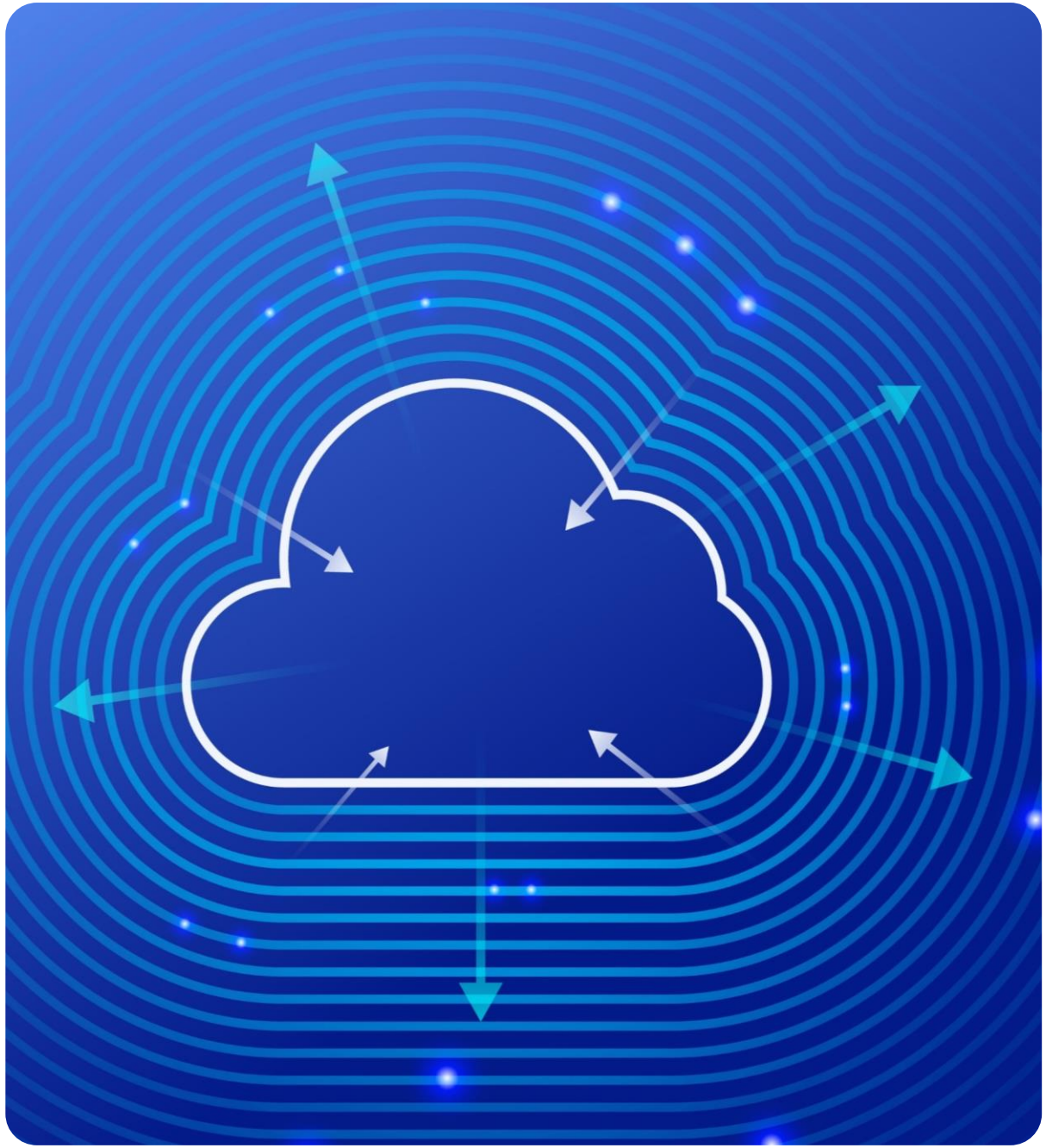
	 **Application Load Balancer (ALB)**	 **Network Load Balancer (NLB)**	 **Gateway Load Balancer (GWLB)**
Operates at	Layer 7	Layer 3	Layer 3 Gateway Layer 4 Load Balancer
Routes traffic based on	Content of the Packet	Header of the Packet	Header of the Packet
Supported Protocols	HTTP, HTTPS, gRPC	TCP, UDP, TLS	IP
Static and Elastic IP	No	Yes	Not Applicable
Target Types	Instances, Containers, Lambda, IP Addresses	Instances, Containers, IP Addresses	Instances, IP Addresses

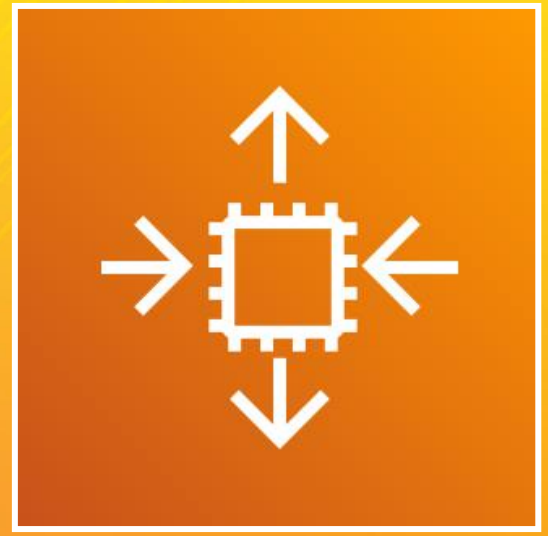


Autoscaling in AWS

Autoscaling in AWS

	 (Launched in 2009) Amazon EC2 Auto Scaling	 (Launched in 2016) AWS Auto Scaling	 (Launched in 2018) AWS Application Auto Scaling
What?	Focus on Amazon EC2 Instance Auto Scaling	Build scaling plans for application scaling for multiple resources across multiple services	Automatically scaling resources for individual AWS services beyond Amazon EC2
How?	Automatically add or remove EC2 instances according to conditions you define in an ASG.	Automatically discover scalable resources in your application and configure scaling for all the resources in a single place using predefined guidance or configure it individually.	Through a scaling plan you can track specific CloudWatch metrics and use AWS CloudFormation templates for custom resource.
Which?	• Amazon EC2 Instances	• Amazon EC2 • Amazon EC2 Spot Fleets • Amazon ECS • Amazon DynamoDB • Amazon Aurora	• AppStream 2.0 fleets • Amazon EMR clusters • Amazon Neptune clusters • SageMaker endpoint variants • Custom Resources • And many more…



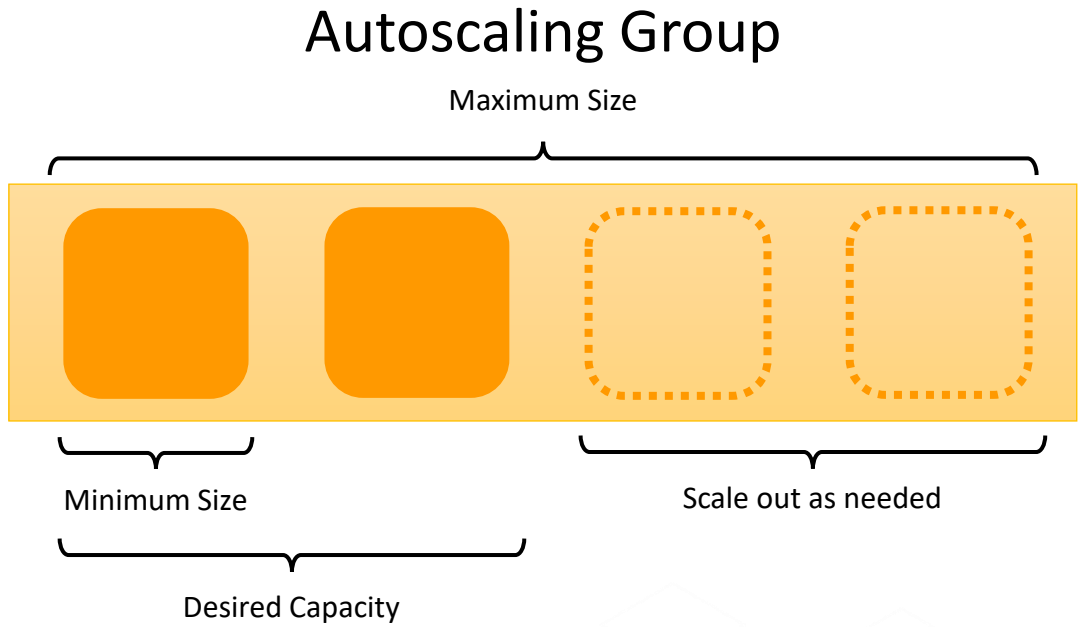


Amazon EC2 Auto Scaling

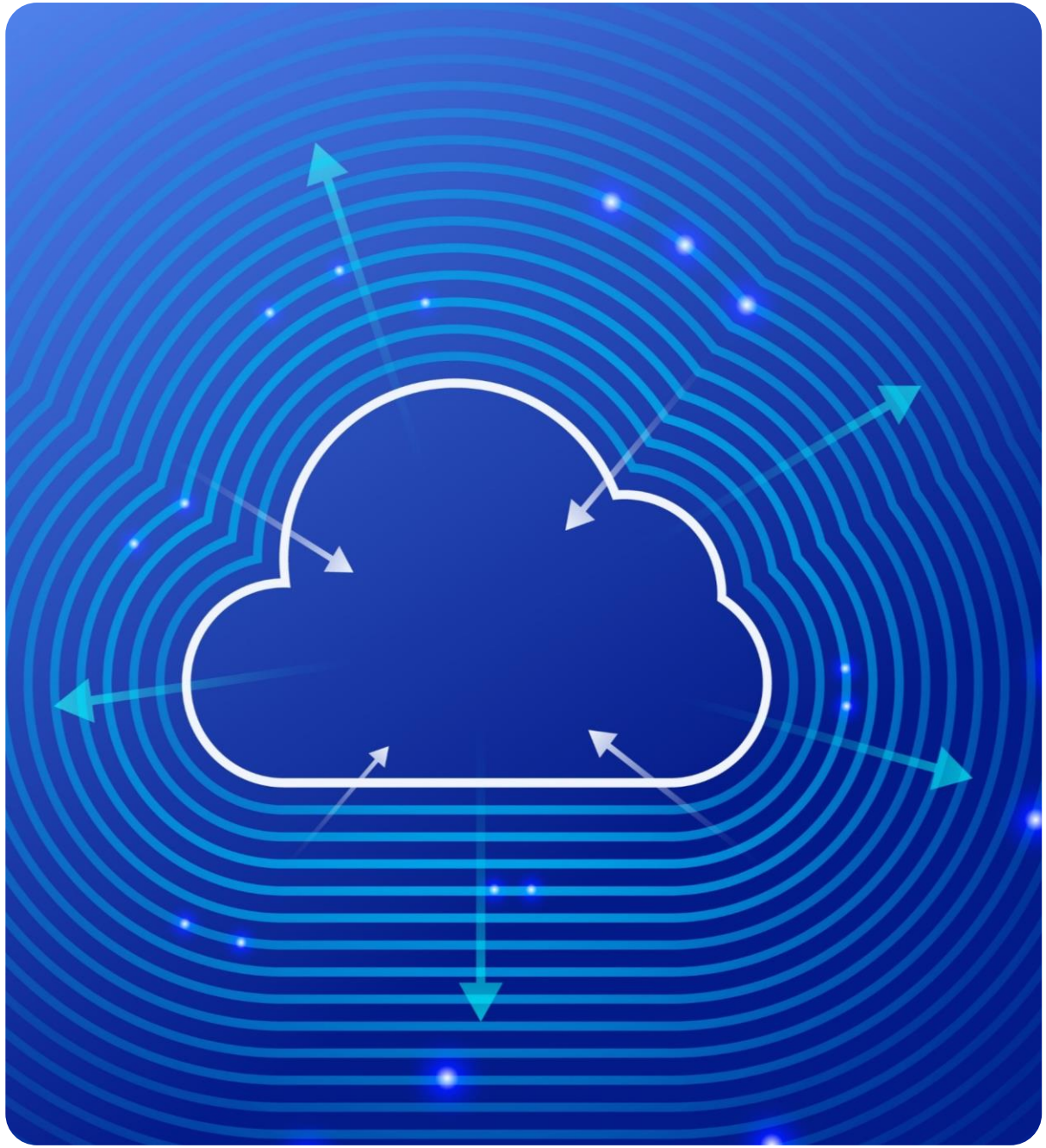
Amazon EC2 Auto Scaling

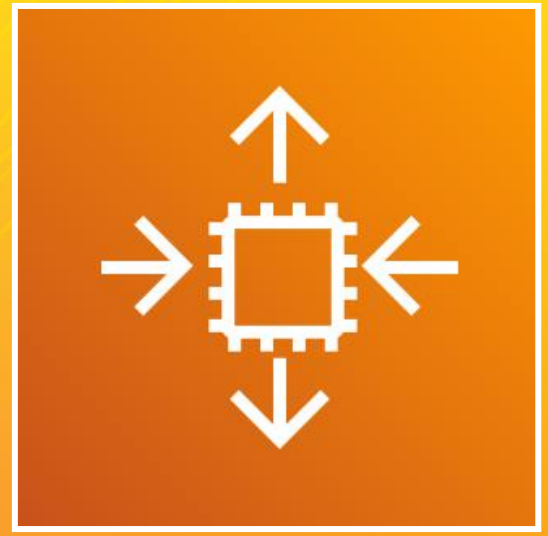
- Add or remove compute capacity to meet changing demand

- Two components
  - Auto Scaling Group
    - An Auto Scaling group contains a collection of EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management.
  - Launch Template
    - Launch Templates allow you to create and save an instance configuration that can be reused, shared and launched at a later time.



- AMI
- Instance Type
- Key Pair
- Network Setting
- User Data
- Etc…



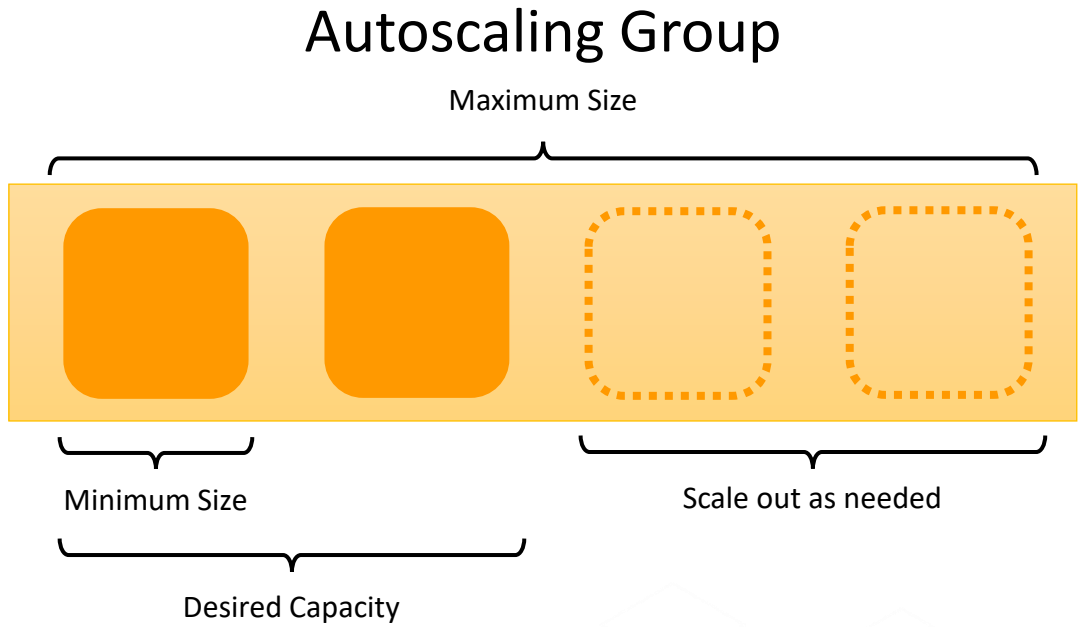


Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling

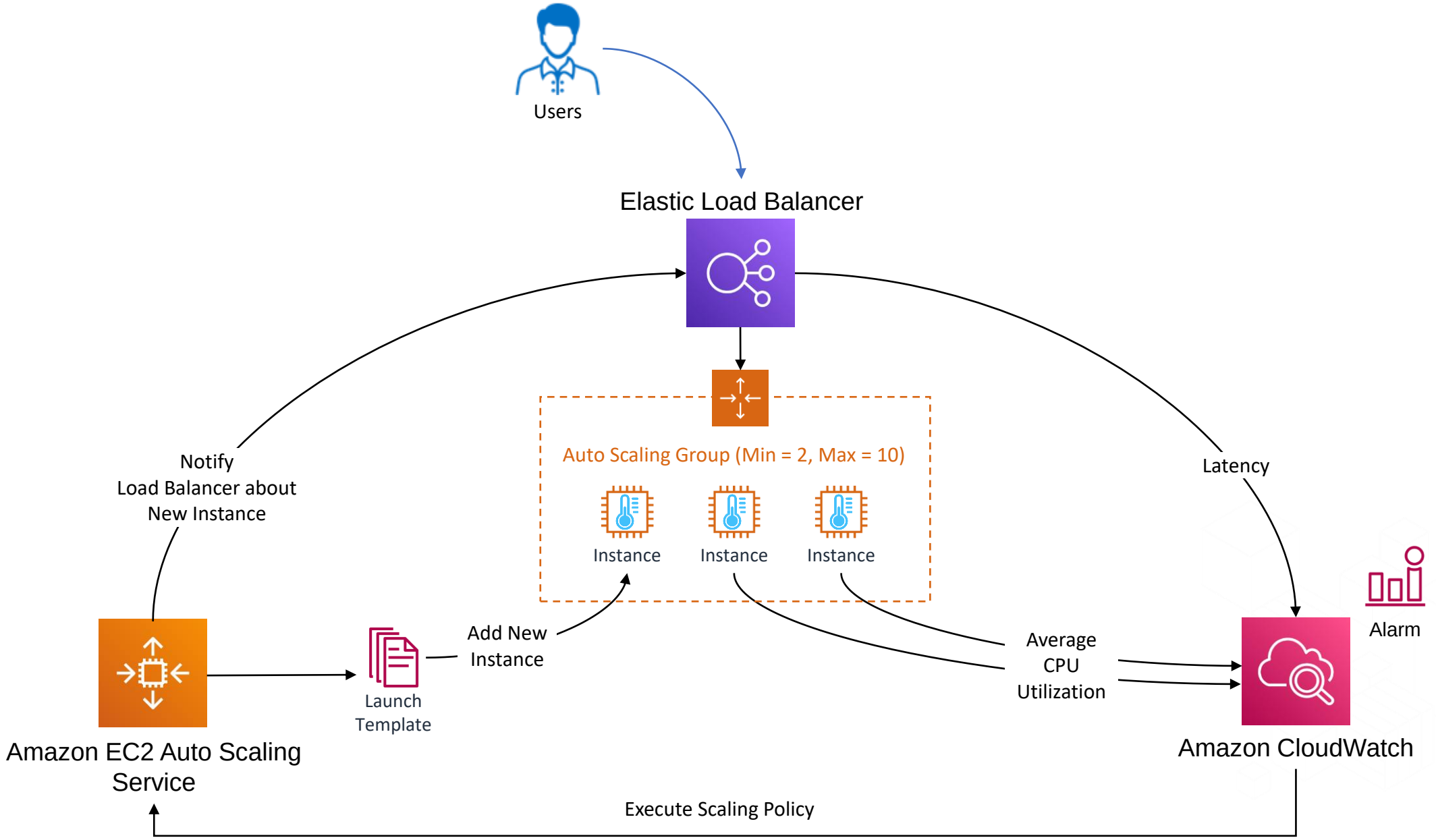
- Add or remove compute capacity to meet changing demand

- Two components
  - Auto Scaling Group
    - An Auto Scaling group contains a collection of EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management.
  - Launch Template
    - Launch Templates allow you to create and save an instance configuration that can be reused, shared and launched at a later time.

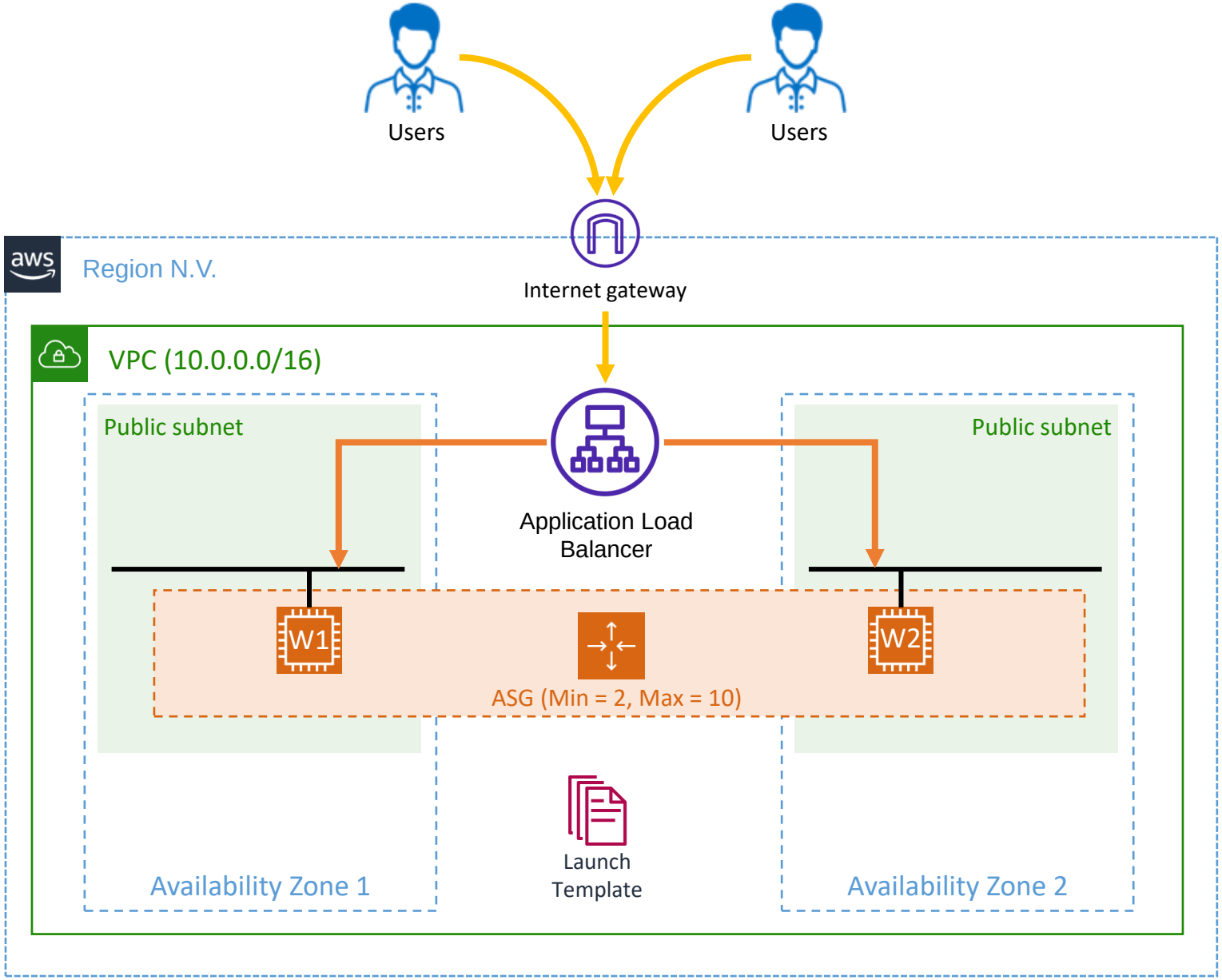


- AMI
- Instance Type
- Key Pair
- Network Setting
- User Data
- Etc…

EC2 Autoscaling



EC2 Auto Scaling in action



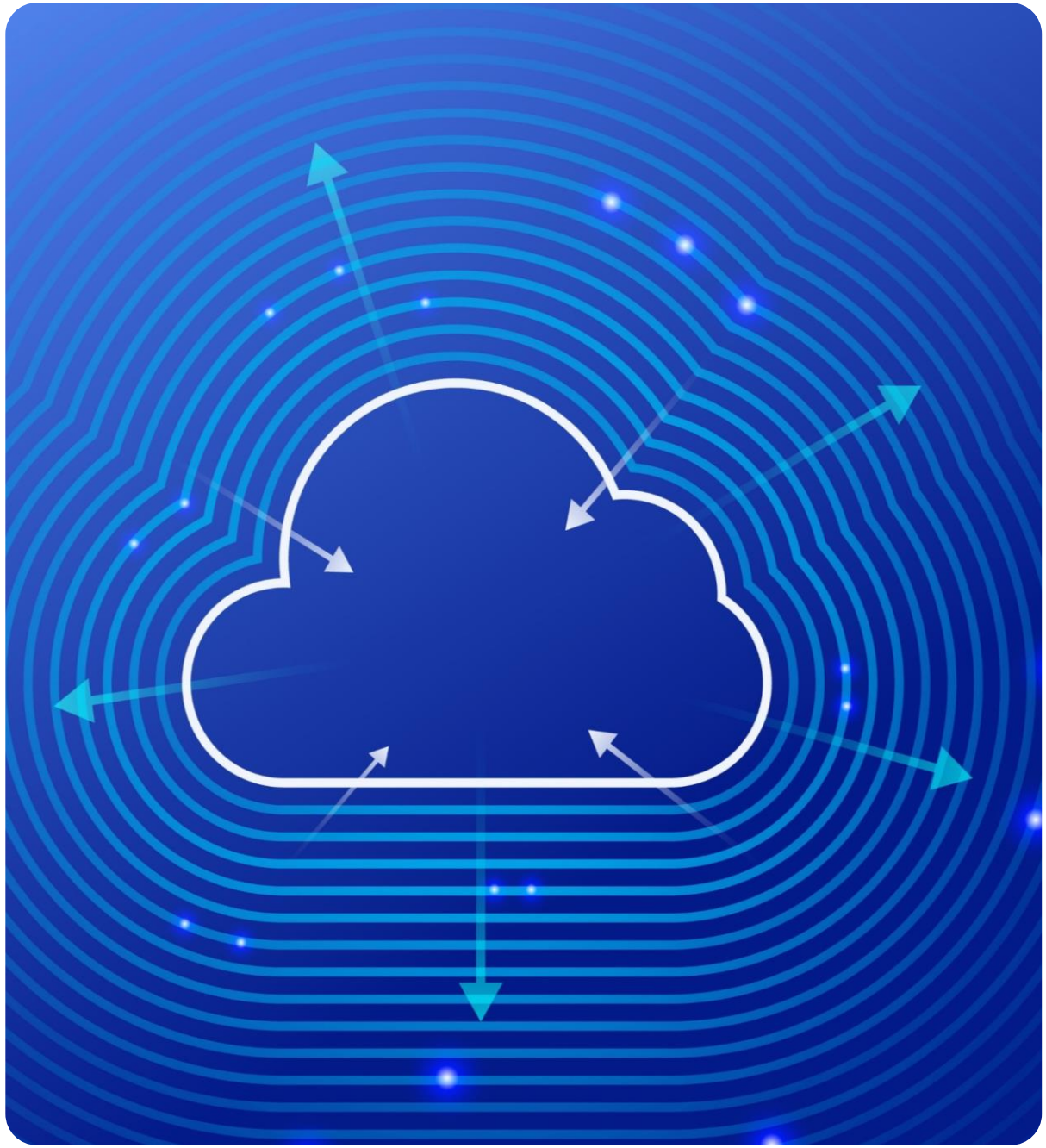
- Step 1
  - Create a Launch Template
- Step 2
  - Create an ALB
- Step 3
  - Create and Auto Scaling Group
- Step 4
  - Test (stress instances)

UserData Script for Web Servers

```
#!/bin/bash
yum update -y
yum install httpd -y
service httpd start
chkconfig httpd on
cd /var/www/html

echo "<html><body><h2><p><font color="red">IP Address</font></p>" > index.html
curl http://169.254.169.254/latest/meta-data/local-ipv4 >> index.html
echo "<br />"

echo "<br /> <br /> <p><font color="blue">Availablity Zone</font></p>" >> index.html
curl http://169.254.169.254/latest/meta-data/placement/availability-zone >> index.html

echo "</h2></body></html>" >> index.html
```

Generate CPU Load

- sudo amazon-linux-extras install epel
- sudo yum -y install stress
- uptime
- sudo stress --cpu 8 -v --timeout 30000s



Amazon EC2
Auto Scaling Policies

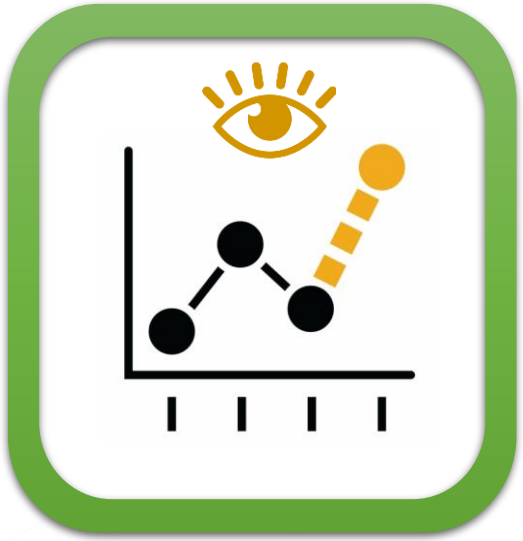
Scaling Policies

Manual Scaling

Scheduled Scaling

Dynamic Scaling

- Simple Scaling
- Step Scaling
- Target Tracking



Predictive Scaling

Manual Scaling

Group size Info

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity

2

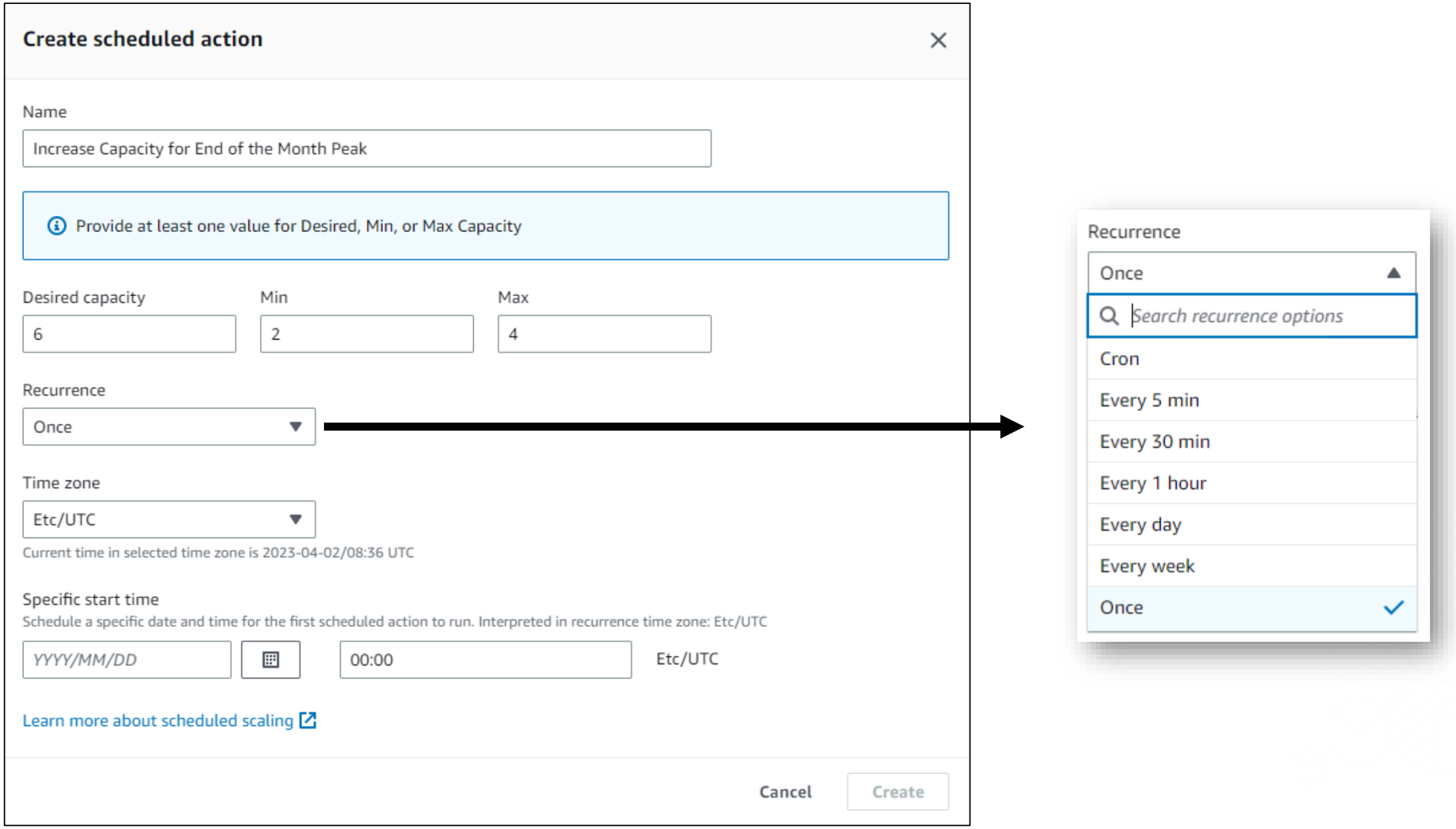
Minimum capacity

2

Maximum capacity

6

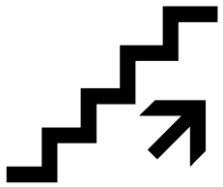
Scheduled Scaling



Dynamic Scaling

Dynamic Scaling

- Simple Scaling
- Step Scaling
- Target Tracking



Dynamic Scaling – Simple Scaling

Policy type

Simple scaling

Scaling policy name

Scale Based on CW Alarm

CloudWatch alarm

Choose an alarm that can scale capacity whenever:

High ELB Connections

Create a CloudWatch alarm

breaches the alarm threshold: RequestCount > 10 for 1 consecutive periods of 300 seconds for the metric dimensions:

LoadBalancer = app/ALB01/885ecb98bafdaba2

Take the action

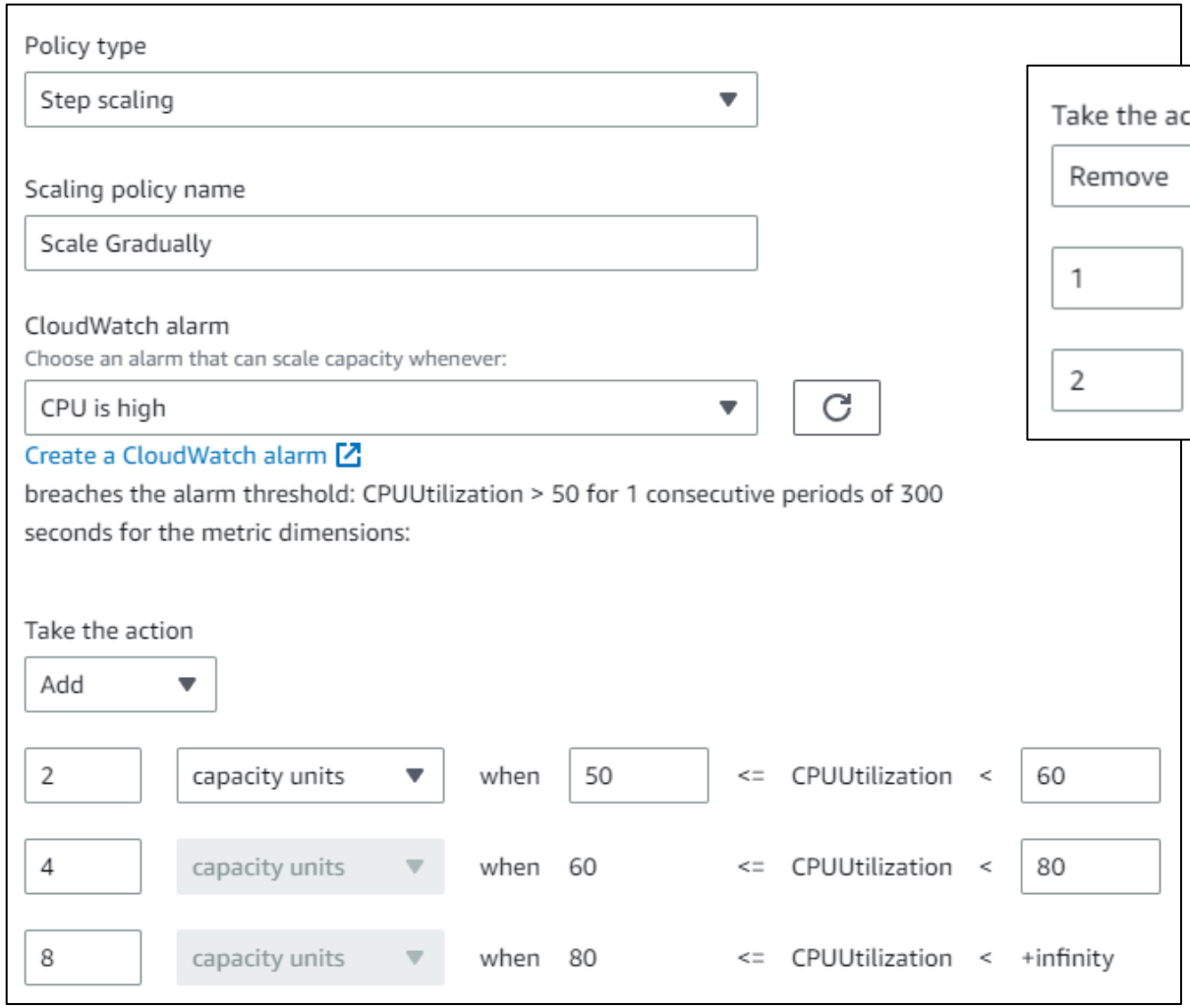
Add | 2 | capacity units

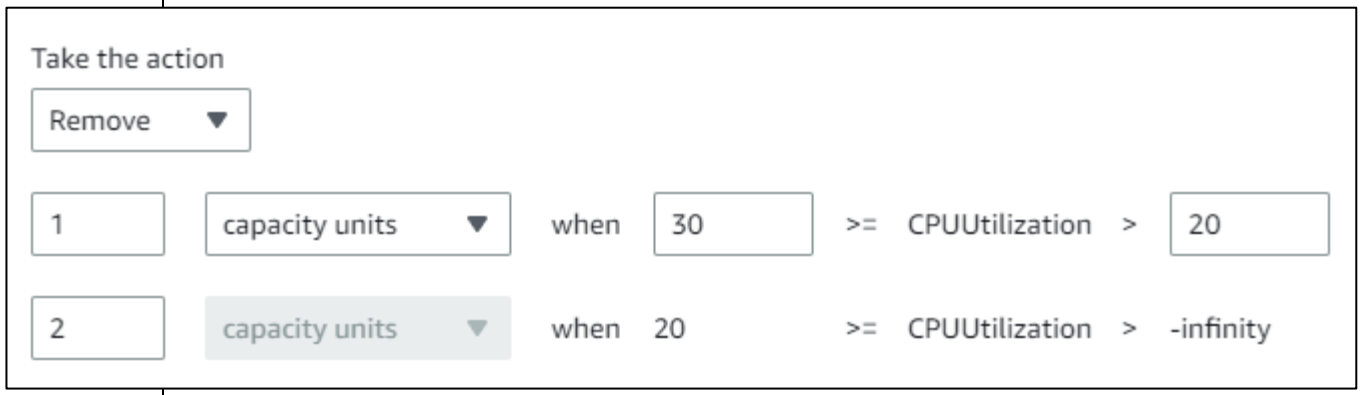
And then wait

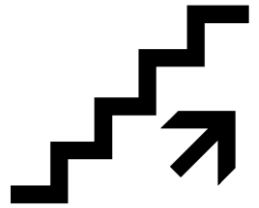
300 seconds before allowing another scaling activity

Dynamic Scaling – Step Scaling

- Bigger the breach, bigger the supply of resources





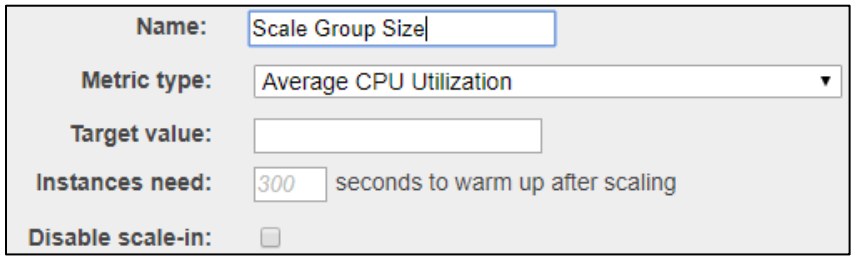


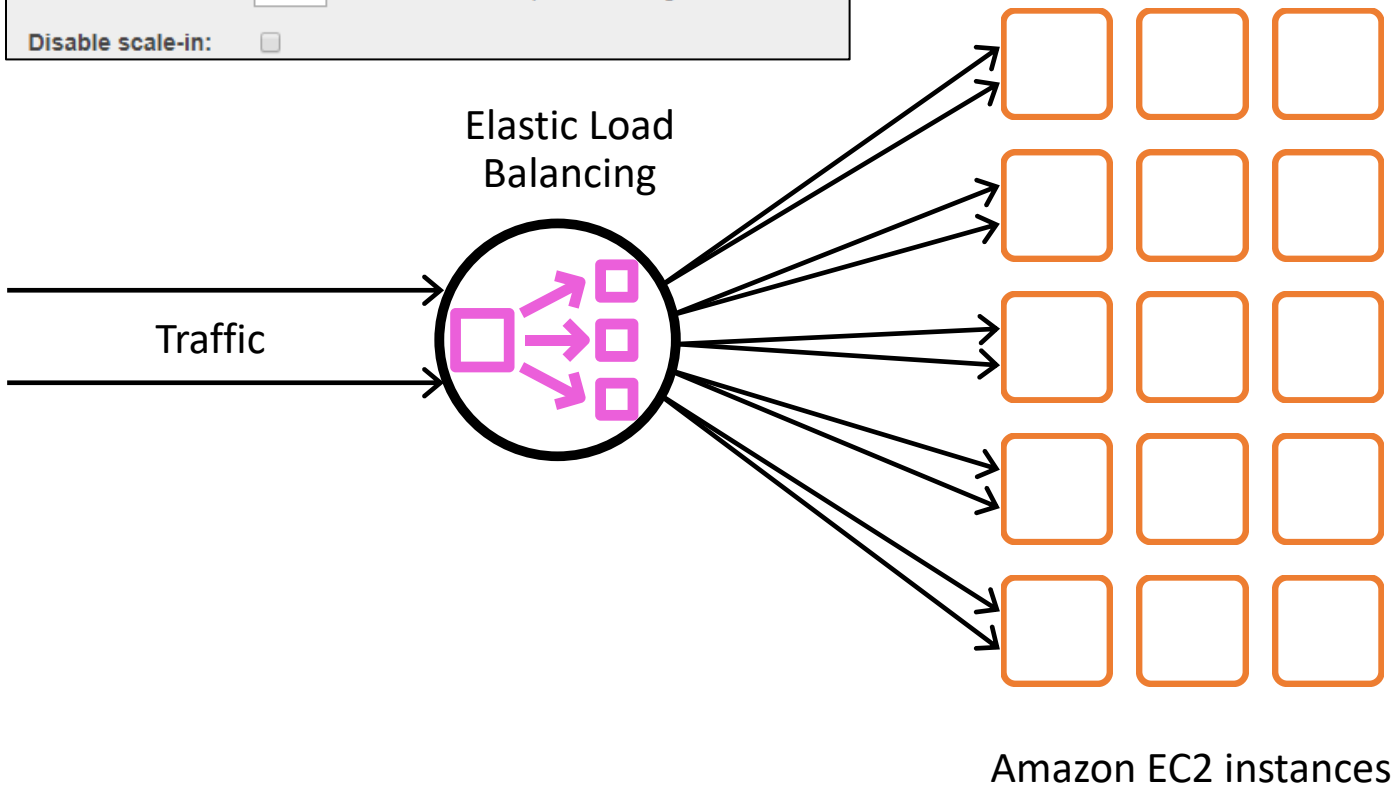
Dynamic Scaling – Target Tracking

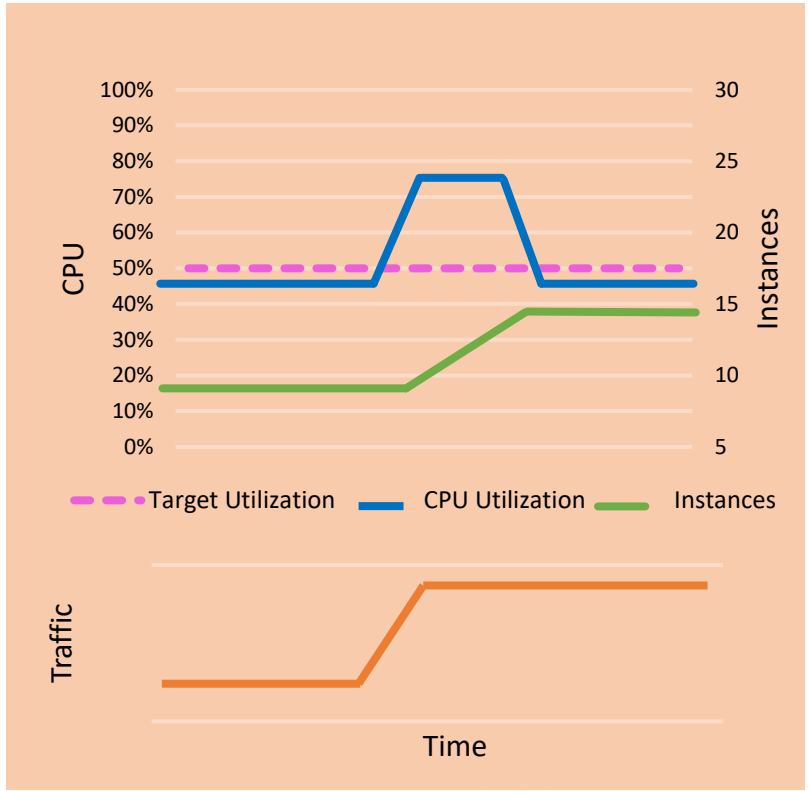
- Cruise Control in a Car

- A thermostat in house

Dynamic Scaling – Target Tracking





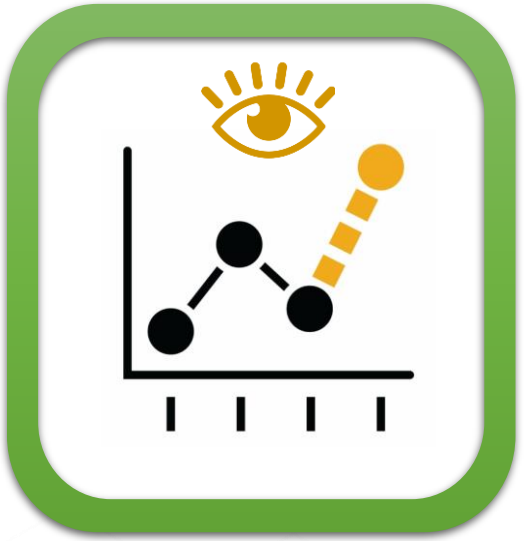


Scaling Policies

Manual Scaling

Scheduled Scaling

Dynamic Scaling



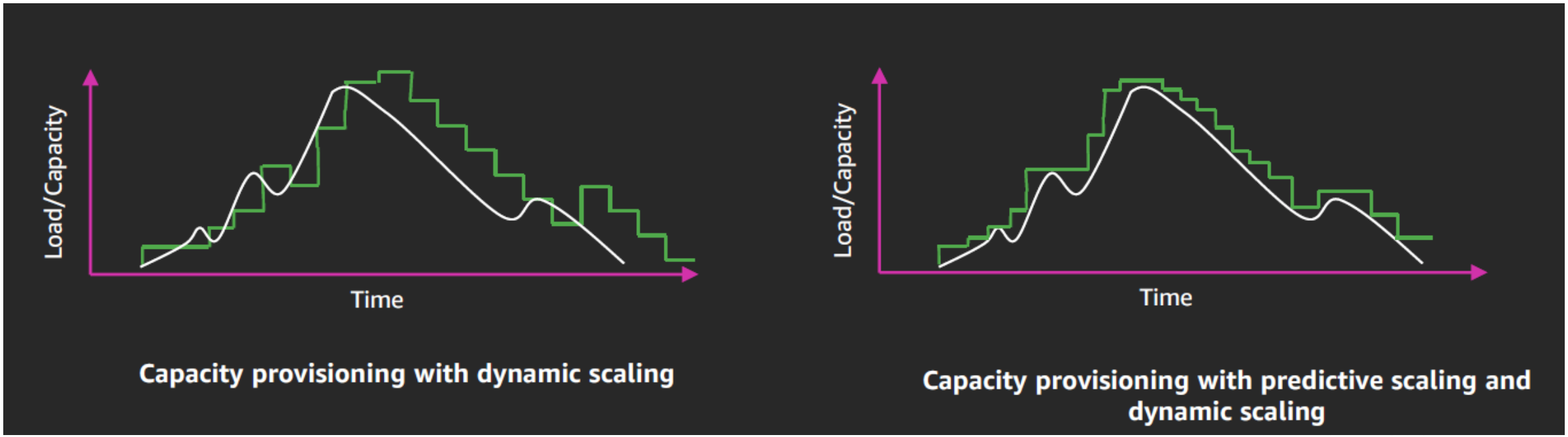
Predictive Scaling

Reactive

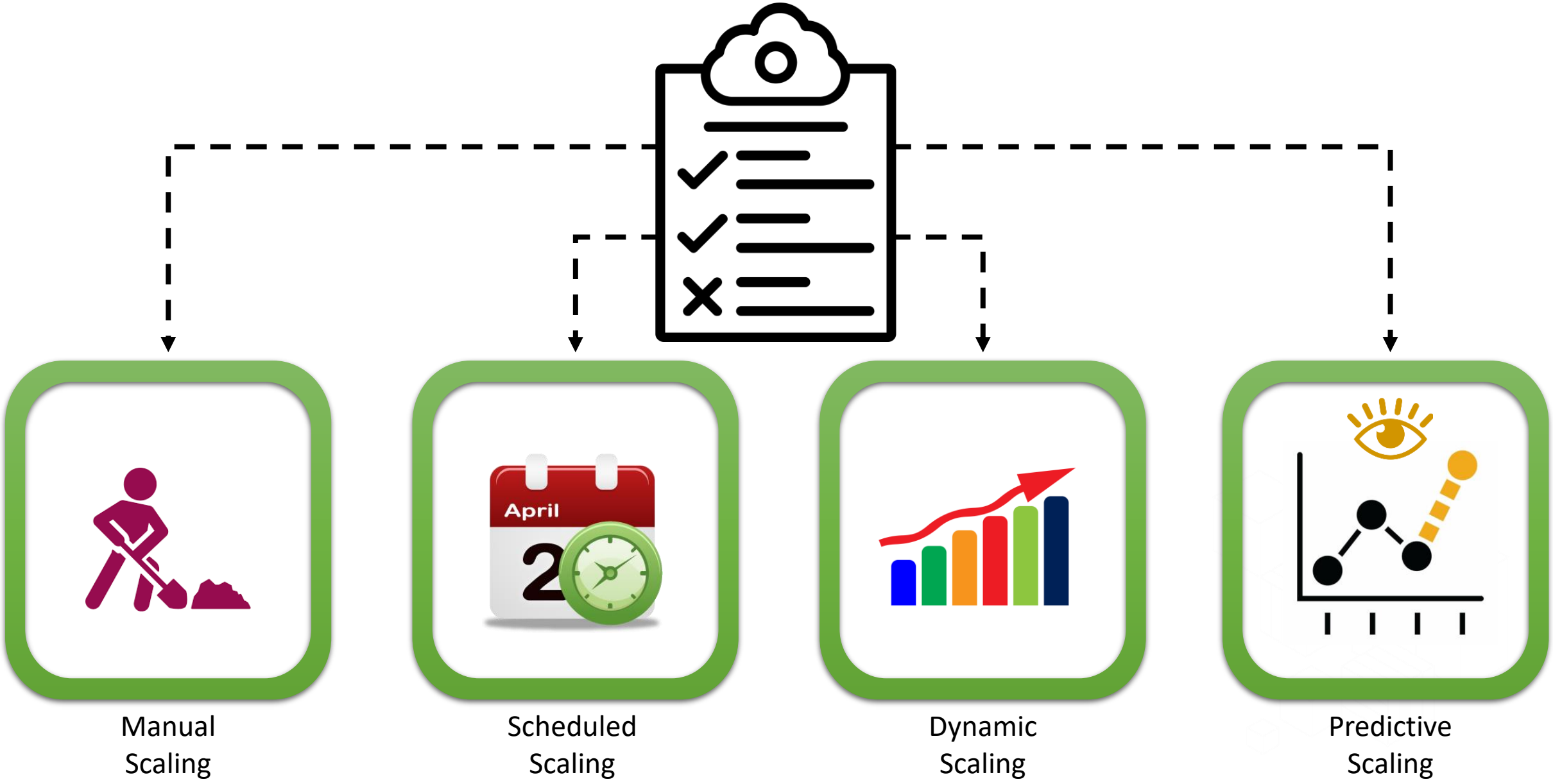
Pro-active

Predictive Scaling

- With predictive scaling, AWS Auto Scaling analyzes the history of the specified load metric from the past 14 days (minimum of 24 hours of data is required) to generate a forecast for two days ahead.
- It then schedules scaling actions to adjust the resource capacity to match the forecast for each hour in the forecast period.



Scaling Policies – Use any combinations



How Netflix uses Autoscaling?

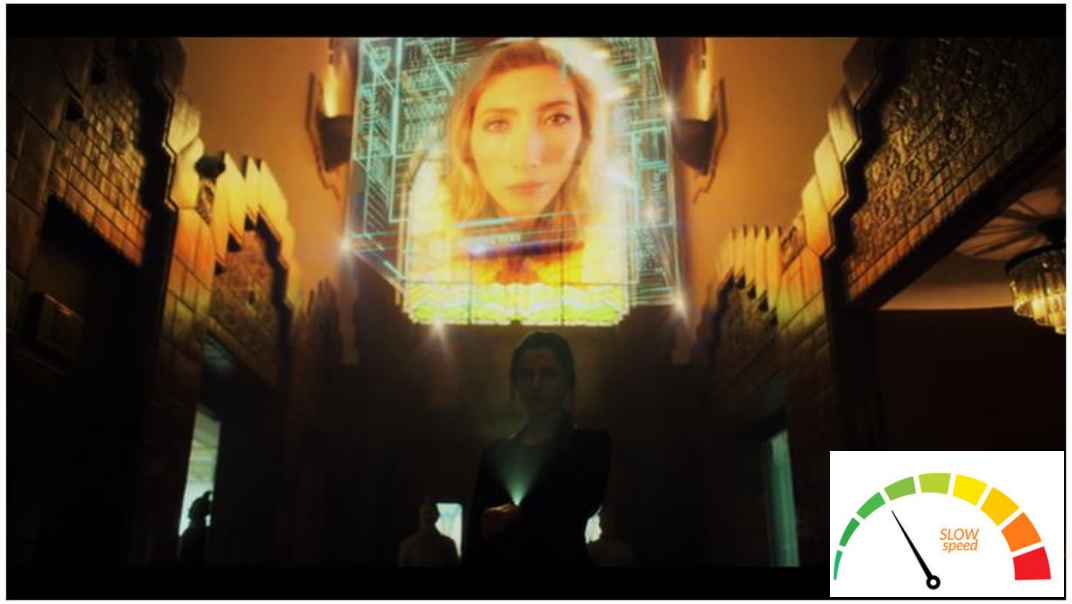
(2018 data)

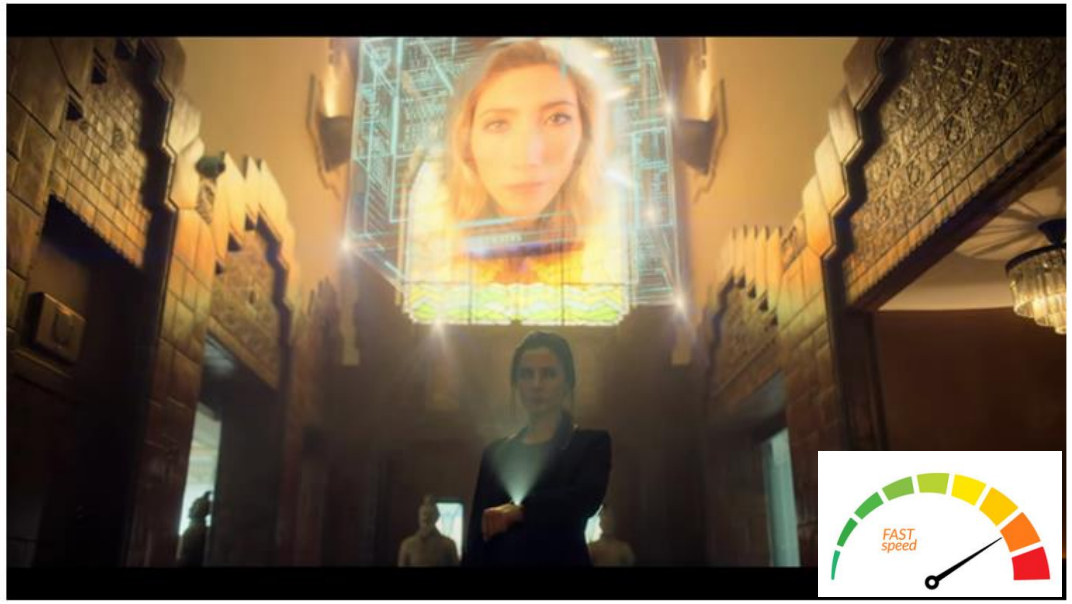
NETFLIX

- 137+ million subscribers
- 190+ countries
- 1700+ supported device types

AWS

- Three regions
- 250,000+ reserved instances
- 20,000 Auto Scaling groups





How Netflix uses Autoscaling?