FIBO
CLOUD
EDUCATION

AWS Solutions Architect

Load Balancers, Auto Scaling Groups

Н. Ганжигүүр
Fibo Cloud

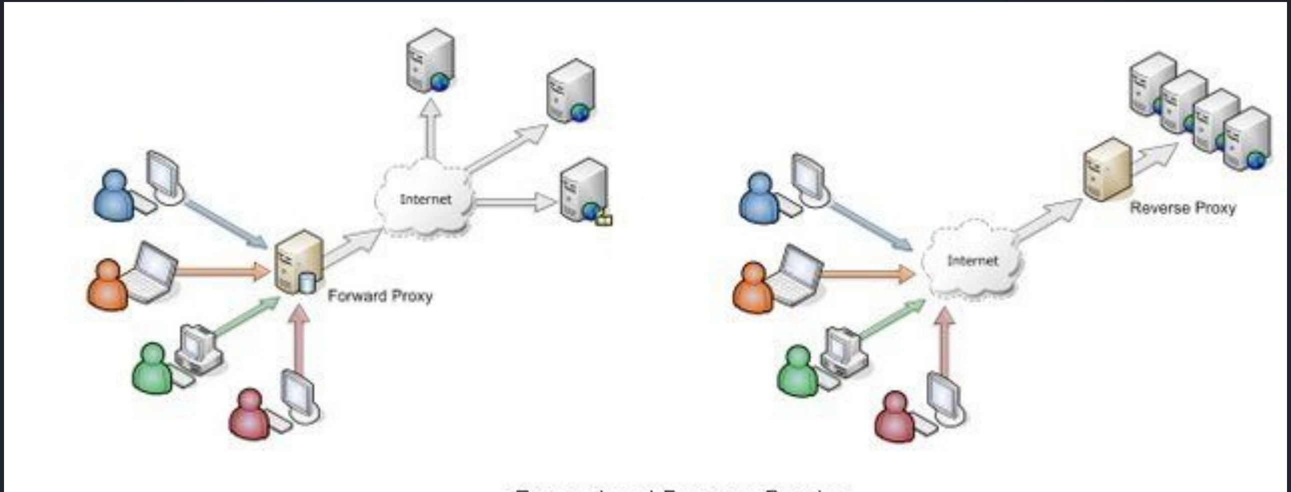
Агуулга

FIBO
CLOUD
EDUCATION

Proxy servers
Elastic Load Balancers
Auto Scaling Group
HA designs
Hands on

Proxy vs Reverse-proxy

FIBO CLOUD EDUCATION



Forward and Reverse Proxies

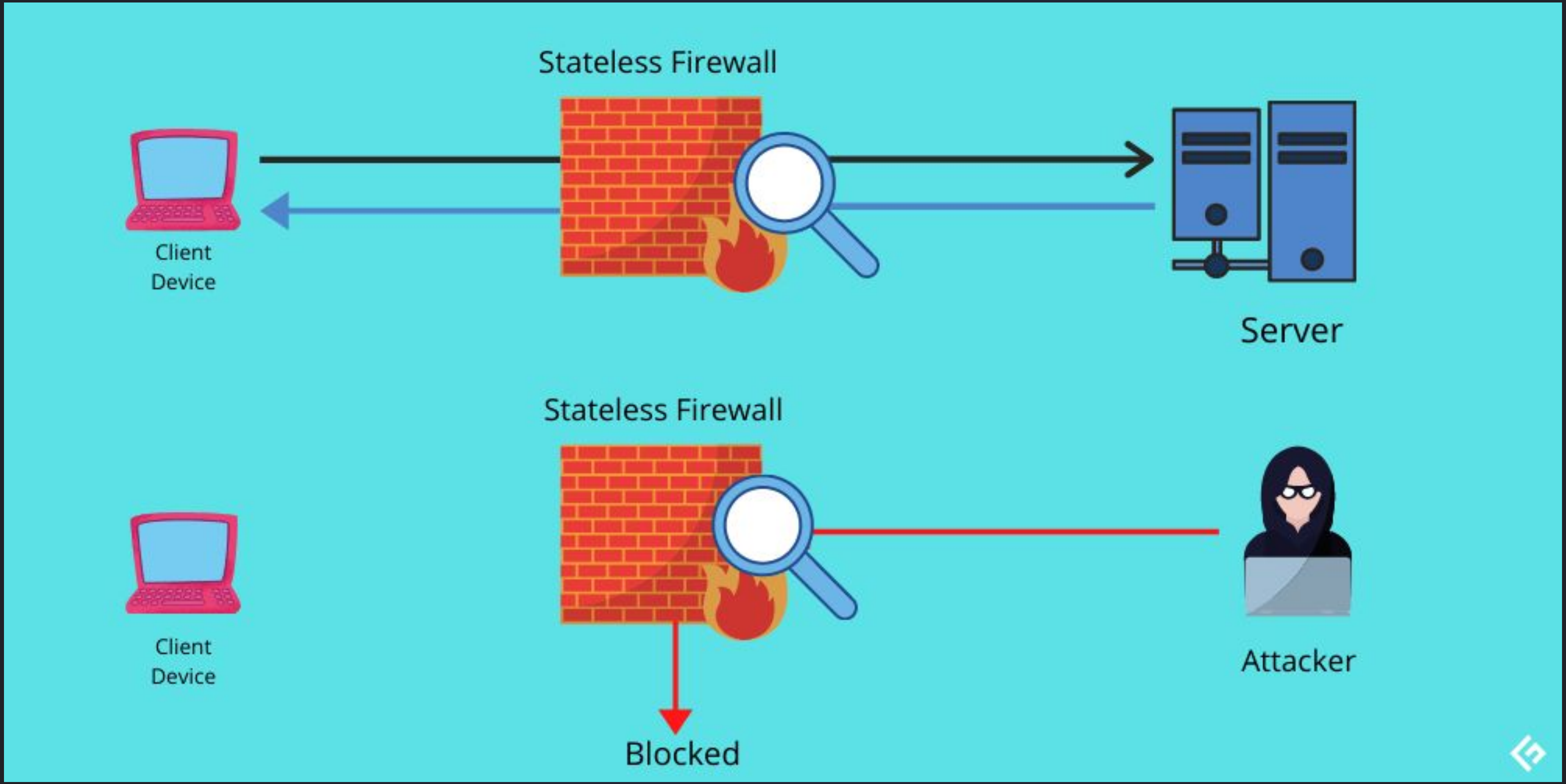
1. Firewall
2. Better Management
3. Security
4. Caching
5. Encryption/Decryption

1. Protect servers
2. Caching
3. Compress

Stateless vs Stateful app

FIBO CLOUD EDUCATION

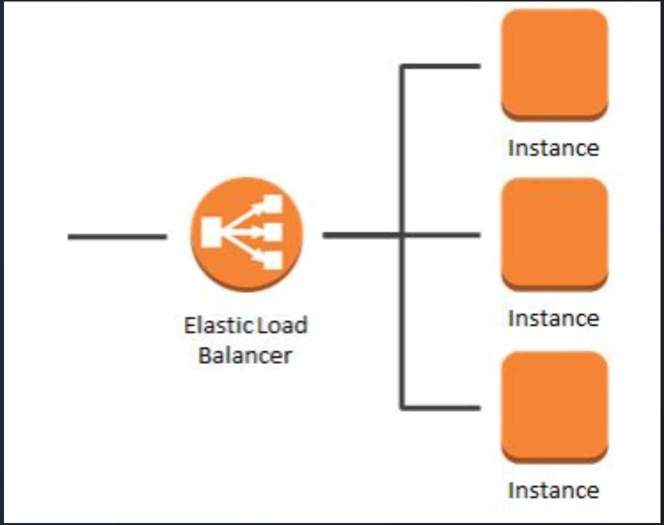
Stateless	Stateful
Does not require the server to retain information about the state.	Requires a server to save information about a session.
Server design, implementation and architecture is simple.	Server design, implementation and architecture is complicated.
Handles crashes well, as we can fail over to a completely new server. Servers are regarded as cheap commodity machines.	Does not handle crashes well. Servers are regarded as valuable and long-living. The user would probably be logged out and have to start from the beginning.
Scaling architecture is easy.	Scaling architectures is difficult and complex.


Stateless Firewall
Client
Device
Server
Stateless Firewall
Client
Device
Attacker
Blocked

Elastic Load balancers

FIBO CLOUD EDUCATION

- ELB Stands for Elastic Load Balancer.
- It distributes the incoming traffic to multiple targets such as Instances, Containers, Lambda Functions, IP Addresses etc.
- It spans in single or multiple availability zones.
- It provides high availability, scaling and security for the application



Types of ELB

- Classic Load Balancer (CLB) – this is the oldest of the three and provides basic load balancing at both layer 4 and layer 7.
- Application Load Balancer (ALB) – layer 7 load balancer that routes connections based on the content of the request.
- Network Load Balancer (NLB) – layer 4 load balancer that routes connections based on IP protocol data.

The Classic Load Balancer may be phased out over time and Amazon are promoting the ALB and NLB for most use cases within VPC.

FIBO
CLOUD
EDUCATION

Application Load Balancer

- It is best suited for load balancing of the web applications and websites.
- It routes traffic to targets within Amazon VPC based on the content of the request.

Network Load Balancer

- It is mostly for the application which has ultra-high performance.
- This load balancer also acts as a single point of contact for the clients.
- This Load Balancer distributes the incoming traffic to the multiple targets.
- The listener checks the connection request from the clients using the protocol and ports we specify.
- It supports TCP, UDP and TLS protocol.

Gateway Load Balancer (Newly Introduced)

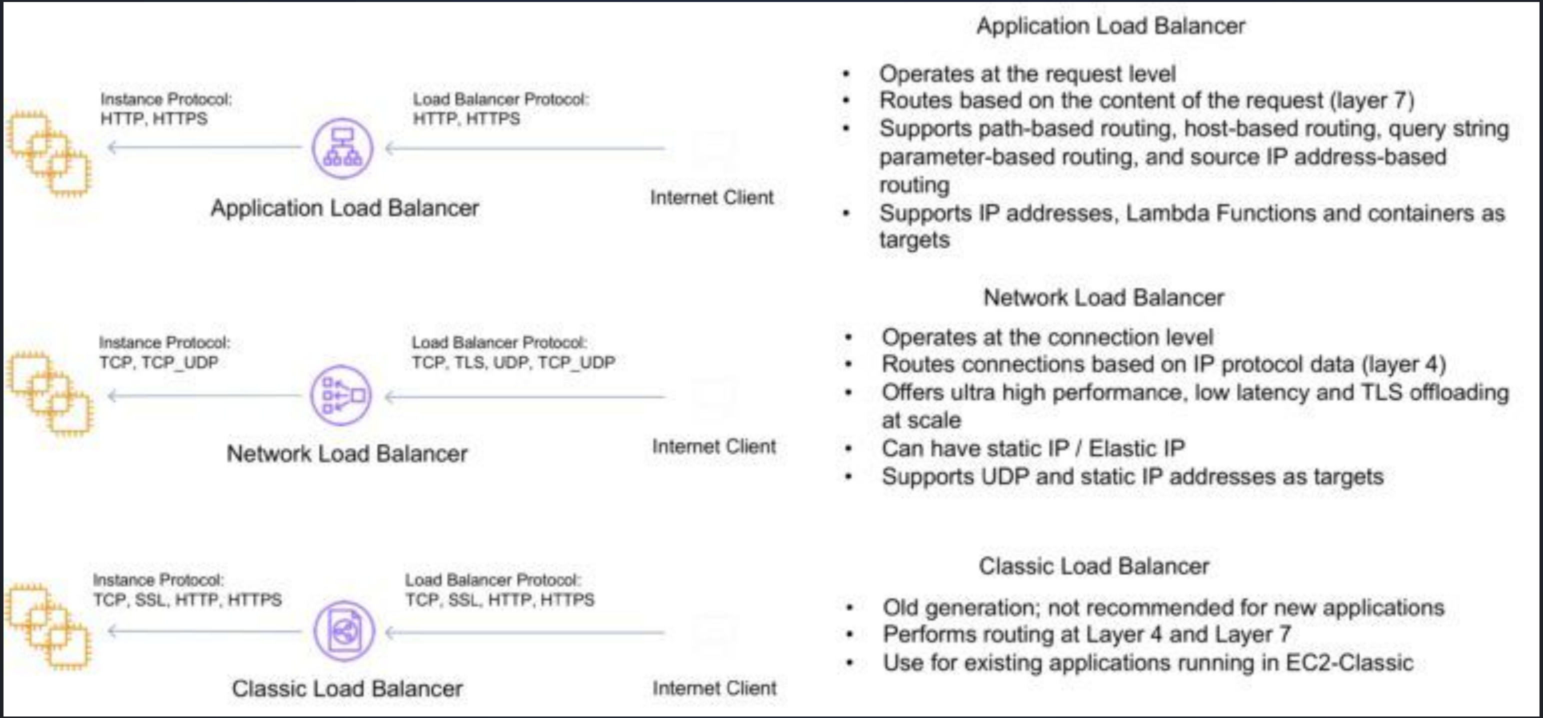
- It is like other load balancers but it is for third-party appliances.
- This provides load balancing and auto scaling for the fleet of third-party appliances.
- It is used for security, network analytics and similar use cases.

Classic Load Balancer

- It operates at request and connection level.
- It is for the EC2 Instance build in the old Classic Network.
- It is an old generation Load Balancer.
- AWS recommends to use Application or Network Load Balancer instead.

Types of ELB

FIBO CLOUD EDUCATION


Instance Protocol: HTTP, HTTPS
Load Balancer Protocol: HTTP, HTTPS
Application Load Balancer
Internet Client
Instance Protocol: TCP, TCP_UDP
Load Balancer Protocol: TCP, TLS, UDP, TCP_UDP
Network Load Balancer
Internet Client
Instance Protocol: TCP, SSL, HTTP, HTTPS
Load Balancer Protocol: TCP, SSL, HTTP, HTTPS
Classic Load Balancer
Internet Client

Application Load Balancer

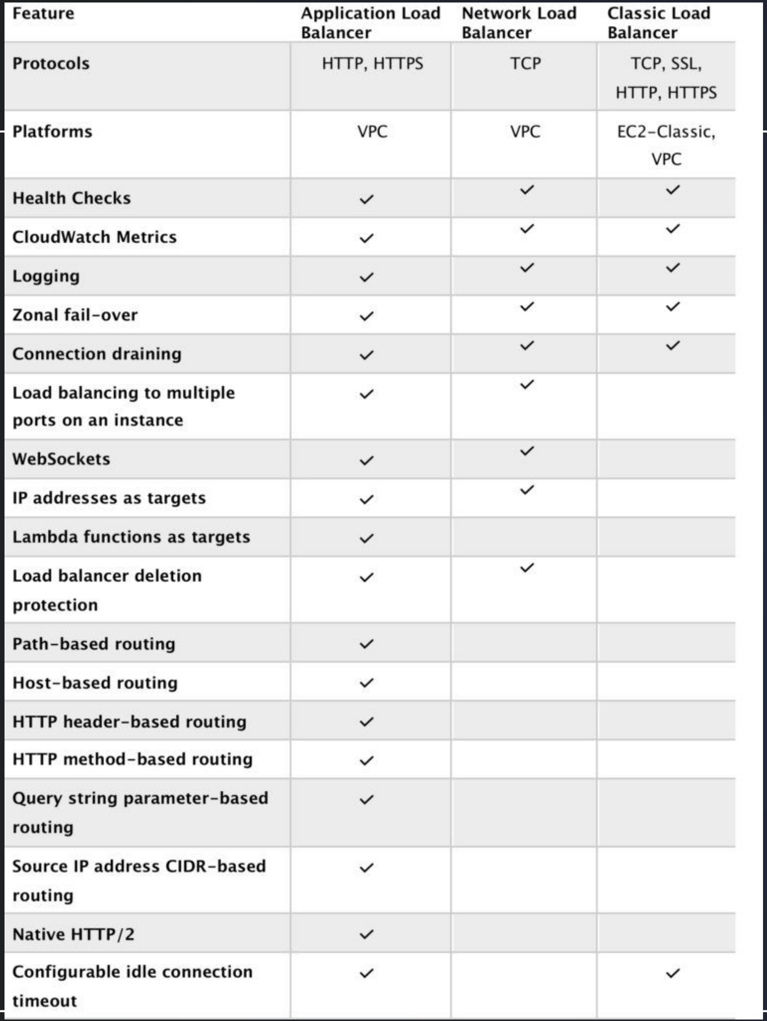
- Operates at the request level
- Routes based on the content of the request (layer 7)
- Supports path-based routing, host-based routing, query string parameter-based routing, and source IP address-based routing
- Supports IP addresses, Lambda Functions and containers as targets

Network Load Balancer

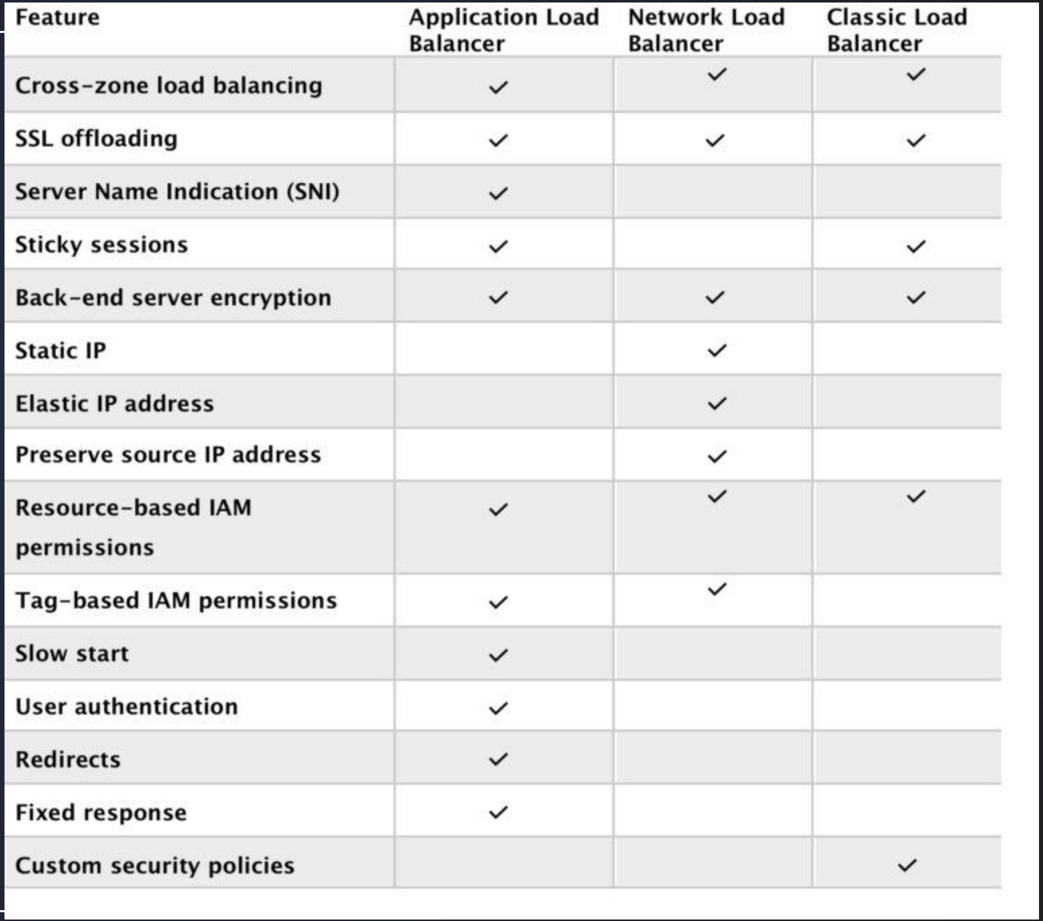
- Operates at the connection level
- Routes connections based on IP protocol data (layer 4)
- Offers ultra high performance, low latency and TLS offloading at scale
- Can have static IP / Elastic IP
- Supports UDP and static IP addresses as targets

Classic Load Balancer

- Old generation; not recommended for new applications
- Performs routing at Layer 4 and Layer 7
- Use for existing applications running in EC2-Classic



Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Protocols	HTTP, HTTPS	TCP	TCP, SSL, HTTP, HTTPS
Platforms	VPC	VPC	EC2-Classic, VPC
Health Checks	✓	✓	✓
CloudWatch Metrics	✓	✓	✓
Logging	✓	✓	✓
Zonal fail-over	✓	✓	✓
Connection draining	✓	✓	✓
Load balancing to multiple ports on an instance	✓	✓	
WebSockets	✓	✓	
IP addresses as targets	✓	✓	
Lambda functions as targets	✓		
Load balancer deletion protection	✓	✓	
Path-based routing	✓		
Host-based routing	✓		
HTTP header-based routing	✓		
HTTP method-based routing	✓		
Query string parameter-based routing	✓		
Source IP address CIDR-based routing	✓		
Native HTTP/2	✓		
Configurable idle connection timeout	✓		✓



Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Cross-zone load balancing	✓	✓	✓
SSL offloading	✓	✓	✓
Server Name Indication (SNI)	✓		
Sticky sessions	✓		✓
Back-end server encryption	✓	✓	✓
Static IP		✓	
Elastic IP address		✓	
Preserve source IP address		✓	
Resource-based IAM permissions	✓	✓	✓
Tag-based IAM permissions	✓	✓	
Slow start	✓		
User authentication	✓		
Redirects	✓		
Fixed response	✓		
Custom security policies			✓

Listeners & Target Group

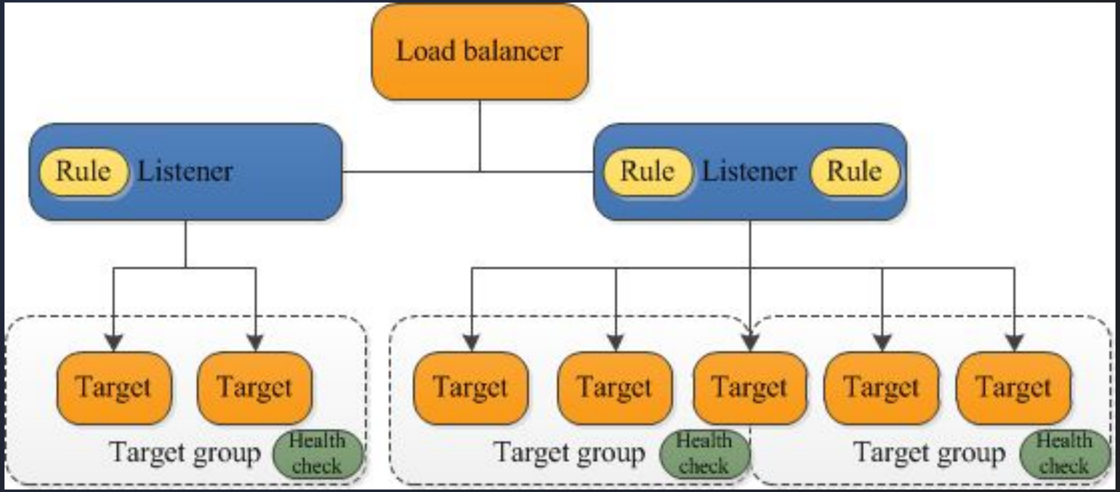
FIBO CLOUD EDUCATION

Listeners

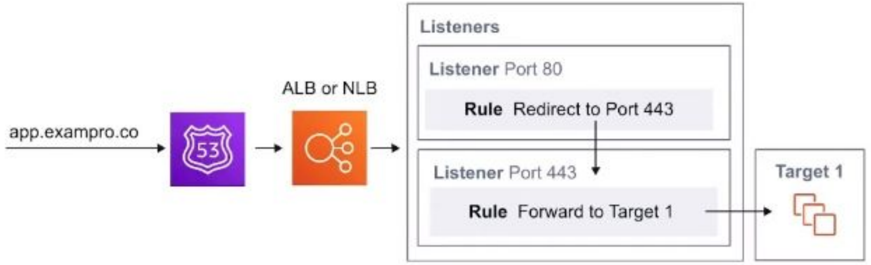
- A listener is a process that checks for connection requests, using the protocol and port that you configured.
- You can add HTTP, HTTPS or both.

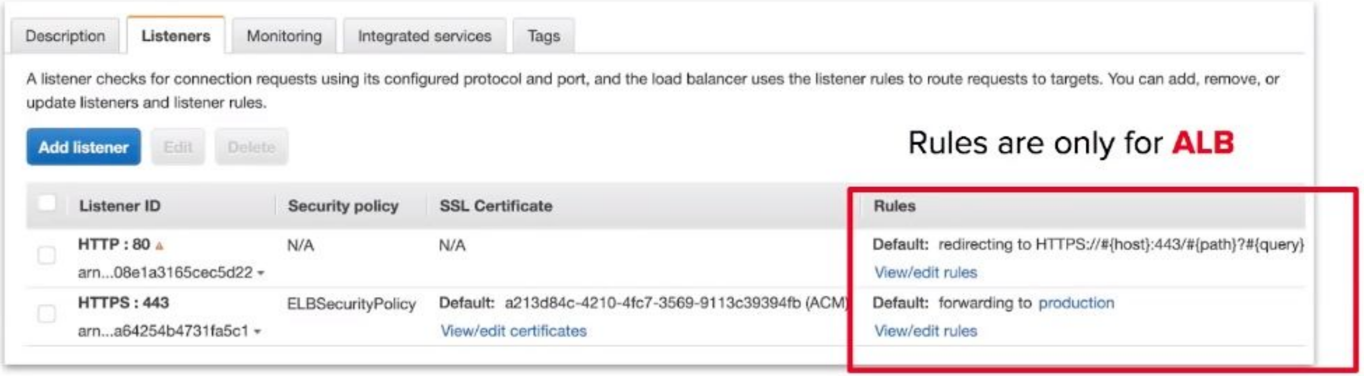
Target Group

- It is the destination of the ELB.
- Different target groups can be created for different types of requests.
- For example, one target group i.e., a fleet of instances will be handling the general request and other target groups will handle the other type of request such as micro services.
- Currently, three types of target supported by ELB: Instance, IP and Lambda Functions.



For Application Load Balancer (ALB) or Network Load Balancer (NLB) traffic is sent to the Listeners. When the port matches it then checks the rules what do to. The rules will forward the traffic to a Target Group. The target group will evenly distribute the traffic to instances registered to that target group.



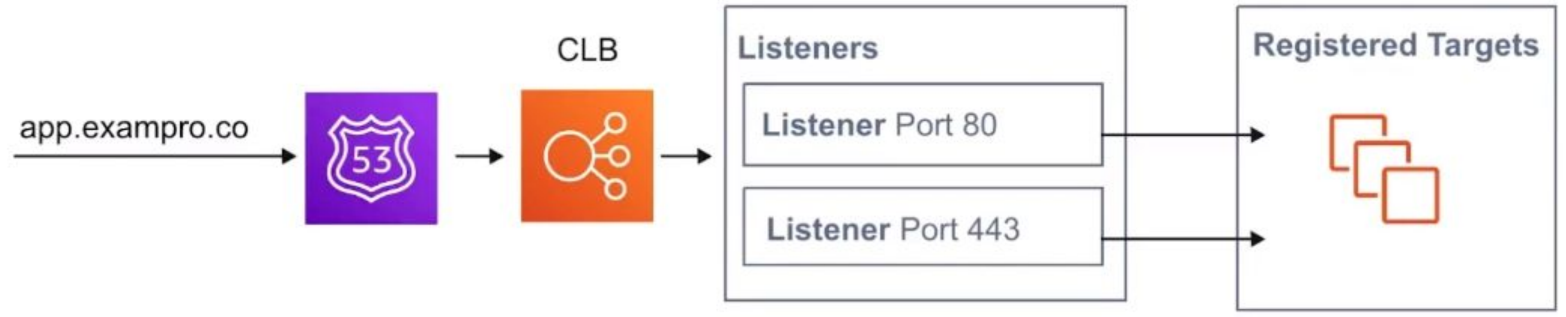


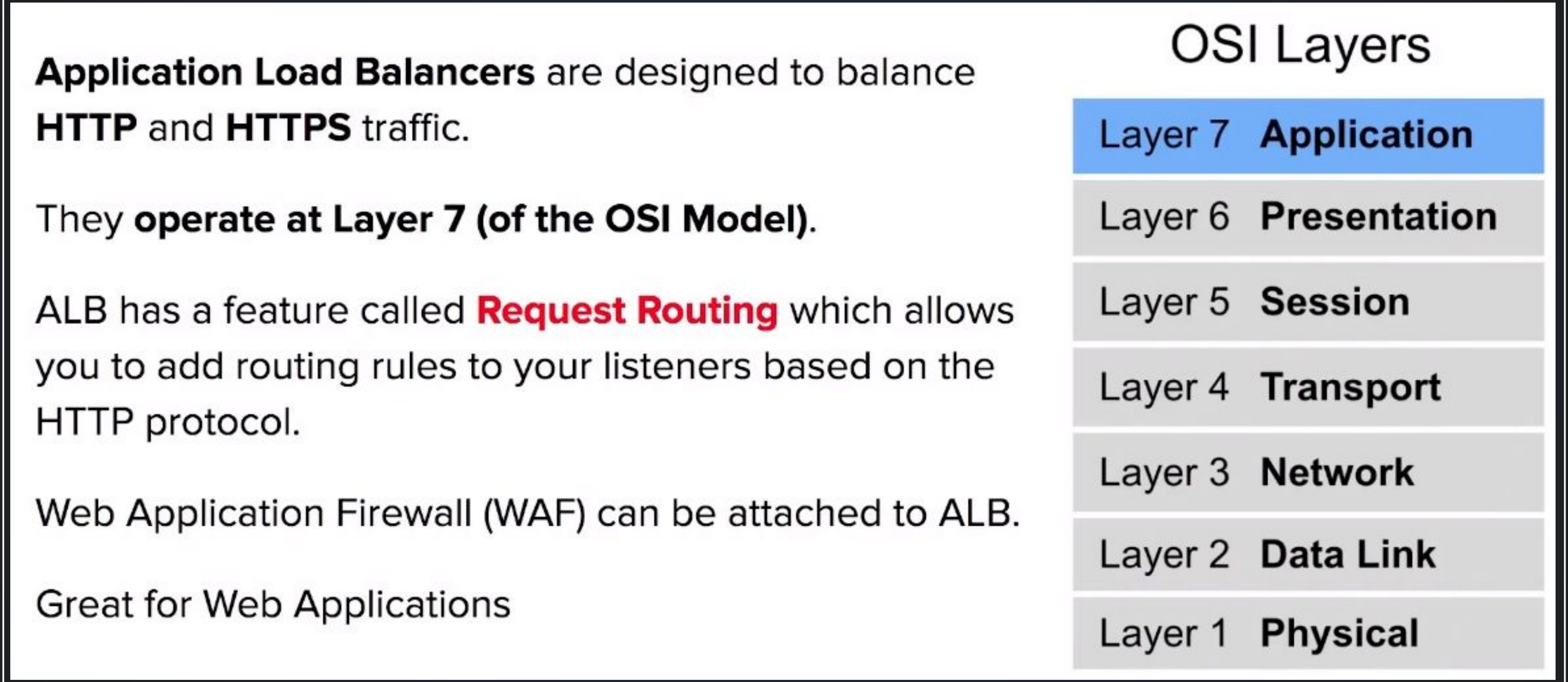
A listener checks for connection requests using its configured protocol and port, and the load balancer uses the listener rules to route requests to targets. You can add, remove, or update listeners and listener rules.

Rules are only for **ALB**

Listener ID	Security policy	SSL Certificate	Rules
HTTP : 80 arn...08e1a3165cec5d22	N/A	N/A	Default: redirecting to HTTPS://#{host}:443/#{path}?#{query} View/edit rules
HTTPS : 443 arn...a64254b4731fa5c1	ELBSecurityPolicy	Default: a213d84c-4210-4fc7-3569-9113c39394fb (ACM) View/edit certificates	Default: forwarding to production View/edit rules

For Classic Load Balancer (CLB) traffic is sent to the Listeners. When the port matches it then it forwards the traffic to any EC2 instances that are registered to the Classic Load Balancer. CLB does not allow you to apply rules to listeners





Application Load Balancers are designed to balance **HTTP** and **HTTPS** traffic.

They **operate at Layer 7 (of the OSI Model)**.

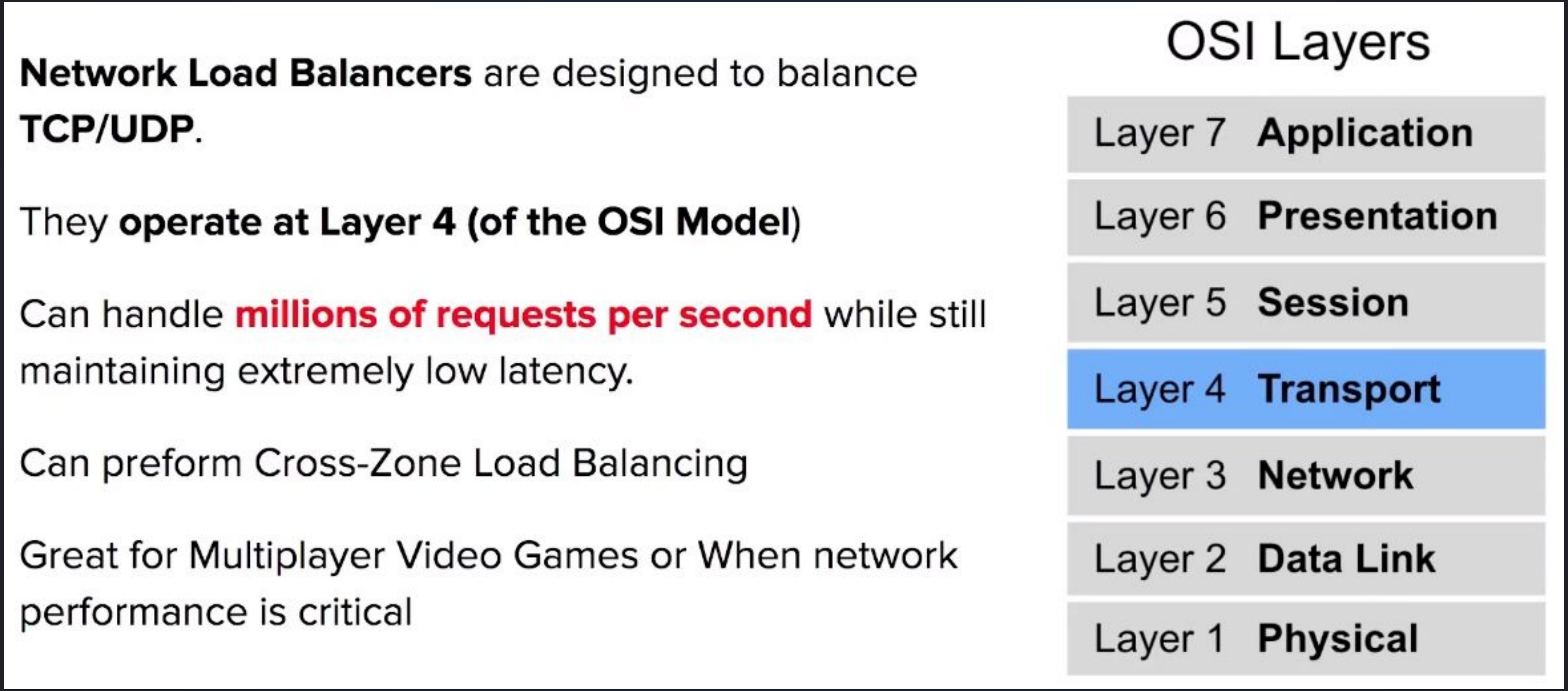
ALB has a feature called **Request Routing** which allows you to add routing rules to your listeners based on the HTTP protocol.

Web Application Firewall (WAF) can be attached to ALB.

Great for Web Applications

OSI Layers

Layer 7	**Application**
Layer 6	**Presentation**
Layer 5	**Session**
Layer 4	**Transport**
Layer 3	**Network**
Layer 2	**Data Link**
Layer 1	**Physical**



Network Load Balancers are designed to balance **TCP/UDP**.

They **operate at Layer 4 (of the OSI Model**)

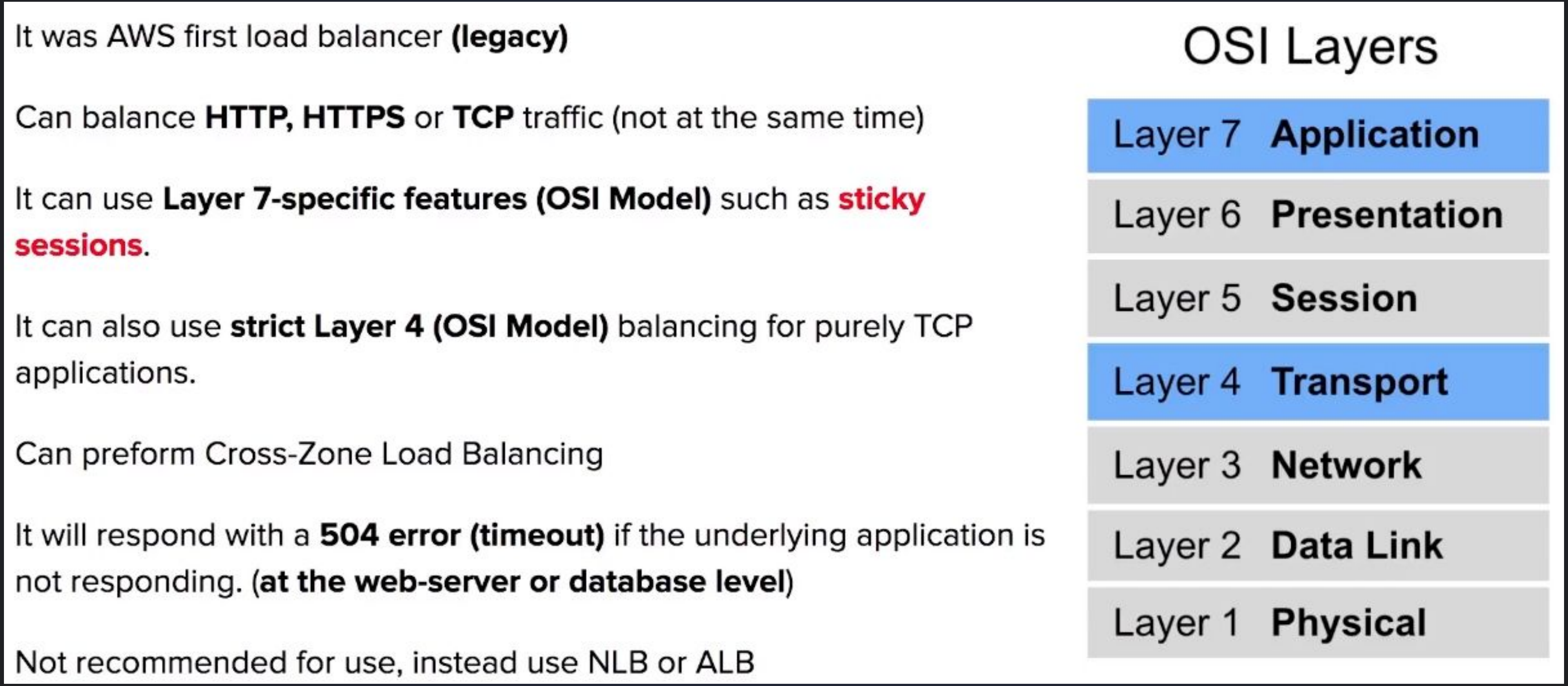
Can handle **millions of requests per second** while still maintaining extremely low latency.

Can preform Cross-Zone Load Balancing

Great for Multiplayer Video Games or When network performance is critical

OSI Layers

Layer 7	**Application**
Layer 6	**Presentation**
Layer 5	**Session**
Layer 4	**Transport**
Layer 3	**Network**
Layer 2	**Data Link**
Layer 1	**Physical**



It was AWS first load balancer **(legacy)**

Can balance **HTTP, HTTPS** or **TCP** traffic (not at the same time)

It can use **Layer 7-specific features (OSI Model)** such as **sticky sessions**.

It can also use **strict Layer 4 (OSI Model)** balancing for purely TCP applications.

Can preform Cross-Zone Load Balancing

It will respond with a **504 error (timeout)** if the underlying application is not responding. (**at the web-server or database level**)

Not recommended for use, instead use NLB or ALB

OSI Layers

Layer 7	**Application**
Layer 6	**Presentation**
Layer 5	**Session**
Layer 4	**Transport**
Layer 3	**Network**
Layer 2	**Data Link**
Layer 1	**Physical**

Use cases

FIBO CLOUD EDUCATION

Web Application Deployed in Multiple Servers:
If a web Application/Website is deployed in multiple EC2 Instances then we can distribute the traffic between the Application Load Balancers.
Building a Hybrid Cloud:
Elastic Load Balancing offers the ability to load balance across AWS and on-premises resources, using a single load balancer. You can achieve this by registering all of your resources to the same target group and associating the target group with a load balancer.
Migrating to AWS:
ELB supports the load balancing capabilities critical for you to migrate to AWS. ELB is well positioned to load balance both traditional as well as cloud native applications with auto scaling capabilities that eliminate the guess work in capacity planning.

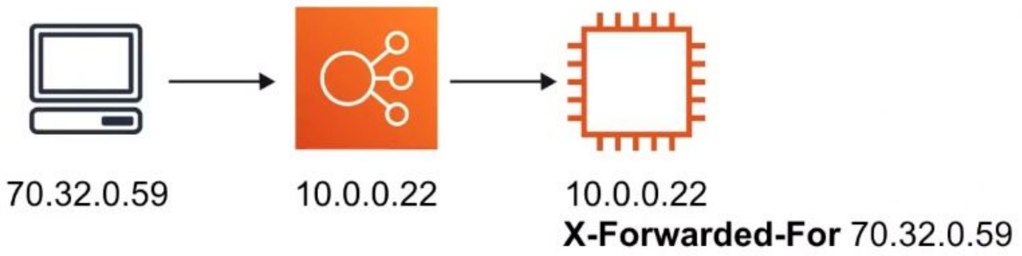
Charges:

- Charges will be based on each hour or partial hour that the ELB is running.
- Charges will also depend on the LCU (Load Balancer Units)
    - Number of new connections per second (up to 25 new connections per second is one LCU)
    - Number of active connections per minute (up to 3,000 active connections per minute is one LCU)
    - Bandwidth measured in Mbps (up to 2.22 Mbps is one LCU)

X-Forwarded-For

If you **need the IPv4 address** of a user, check the **X-Forwarded-For** header

The **X-Forwarded-For (XFF)** header is a command method for identifying the **originating IP address** of a client connecting to a web server through an HTTP proxy or a load balancer.



Sticky sessions

FIBO CLOUD EDUCATION

Sticky Sessions is an advanced load balancing method that allows you to **bind a user's session to a specific EC2 instance.**

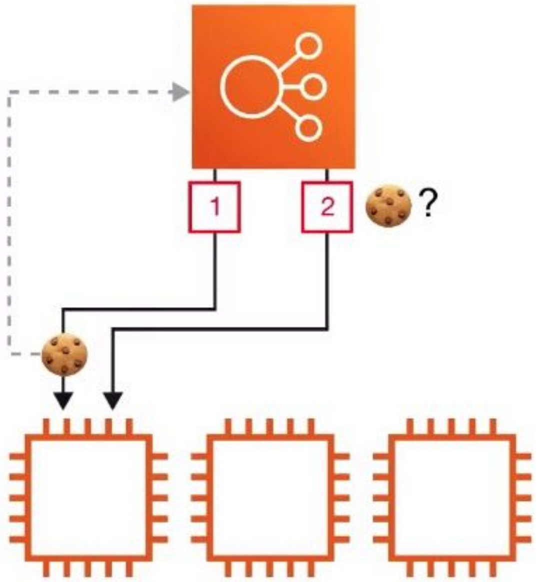
Ensures all **requests** from that session are **sent to the same instance.**

Typically **utilized** with a **Classic Load Balancer**

Can be enabled for ALB though can only be set on a Target Group not individual EC2 instances.

Cookies are used to remember which EC2 instance.

Useful when specific **information is only stored locally on a single instance**



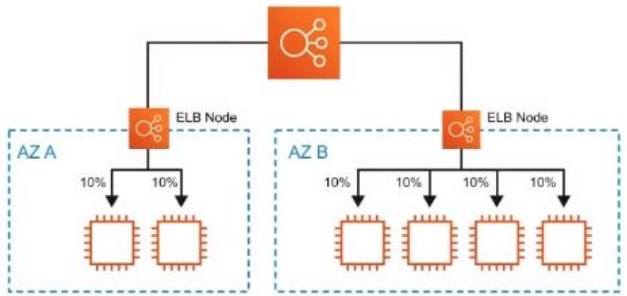
Cross-Zone Load Balancing

FIBO CLOUD EDUCATION

Only for **Classic** and **Network** Load Balancer

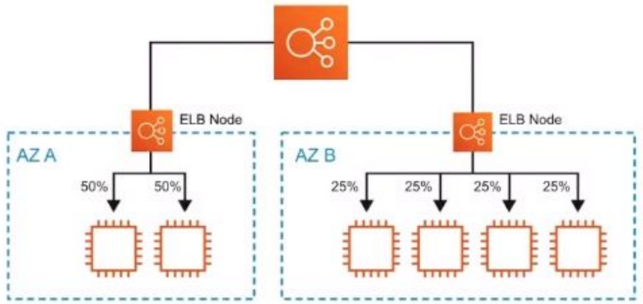
Cross-Zone Load Balancing Enabled

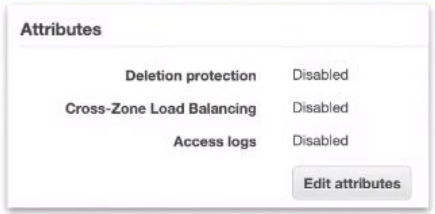
requests are distributed evenly across the instances **in all enabled** Availability Zones.



Cross-Zone Load Balancing Disabled

requests are distributed evenly across the instances **in only** its Availability Zone.

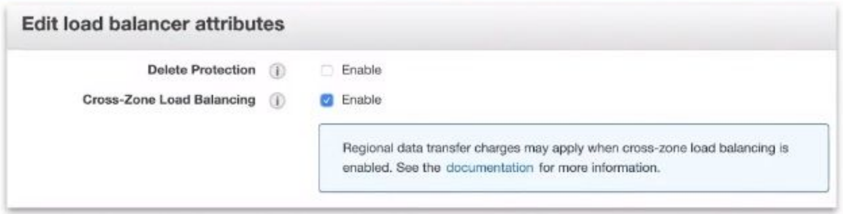




Attributes

Deletion protection	Disabled
Cross-Zone Load Balancing	Disabled
Access logs	Disabled

Edit attributes



Edit load balancer attributes

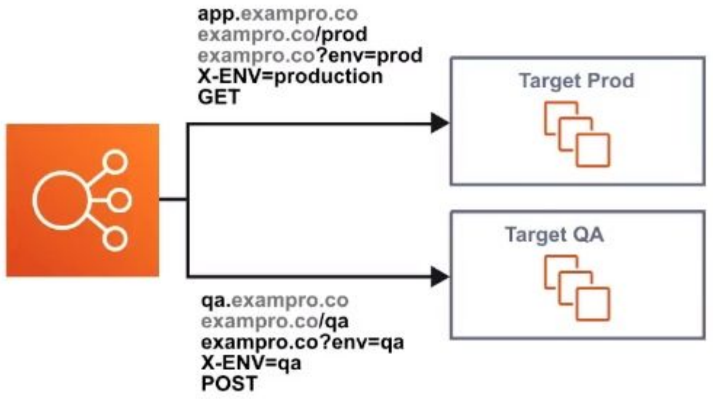
Delete Protection	☐ Enable
Cross-Zone Load Balancing	☑ Enable

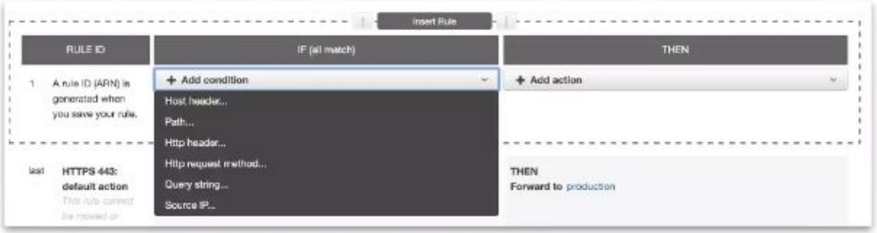
Regional data transfer charges may apply when cross-zone load balancing is enabled. See the documentation for more information.

ALB forwarding

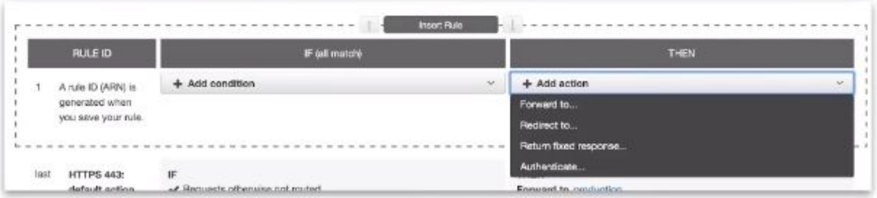
Apply rules to incoming request and then **forward** or **redirect** traffic.

- ✓ Host header
- ✓ Source IP
- ✓ Path
- ✓ Http header
- ✓ Http header method
- ✓ Query string

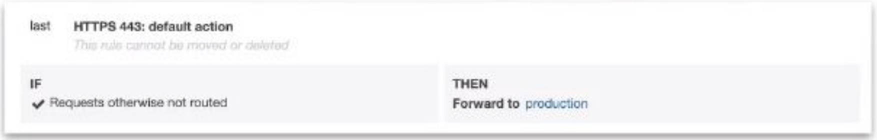




RULE ID	IF (all match)	THEN
1 A rule ID (ARN) is generated when you save your rule.	+ Add condition	+ Add action
last HTTPS 443: default action This rule cannot be moved or		THEN Forward to production



RULE ID	IF (all match)	THEN
1 A rule ID (ARN) is generated when you save your rule.	+ Add condition	+ Add action
last HTTPS 443: default action	IF ✓ Requests otherwise not routed	Forward to production



last HTTPS 443: default action

This rule cannot be moved or deleted

IF	THEN
✓ Requests otherwise not routed	Forward to production

FIBO
CLOUD
EDUCATION

Auto Scaling Group

AWS Auto-Scaling Group

Auto Scaling Groups (ASG) contains a collection of EC@ instances that are treated as a group for the purposes of automatic scaling and management

Concepts:

- Group (fleet)
  - Logical component. Webserver group or Application group or Database group etc.
- Configuration template
  - Groups uses a launch template or launch config as a config template for its EC2 instance. AMI ID, instance type, keypair, security groups, block device mappings...
- Scaling Policy
  - Scaling options provides several ways for you to scale your Auto Scaling groups.
- Health check

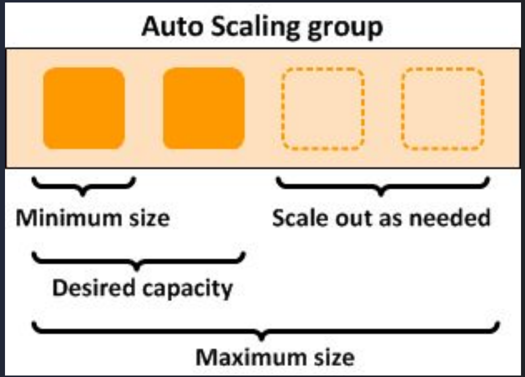
Scaling policies

1. Maintain current instance levels at all times

You can configure your ASG to maintain a specified number of running instance at all times.
When Amazon EC2 Auto scaling finds an unhealthy instance, it terminates it and launches a new one.

2. Scale manually

Manually scaling is the most basic way to scale your resources, where you specify only the change in the min, max or desired capacity of your ASG.



3. Scale based on a schedule

Scaling by schedule means that scalin action are performed automatically as a function of time and date.

This is useful when you know exactly when to increase or decrease the number of instance in your group, simple because the need arises on a predictable schedule.

Scaling policies

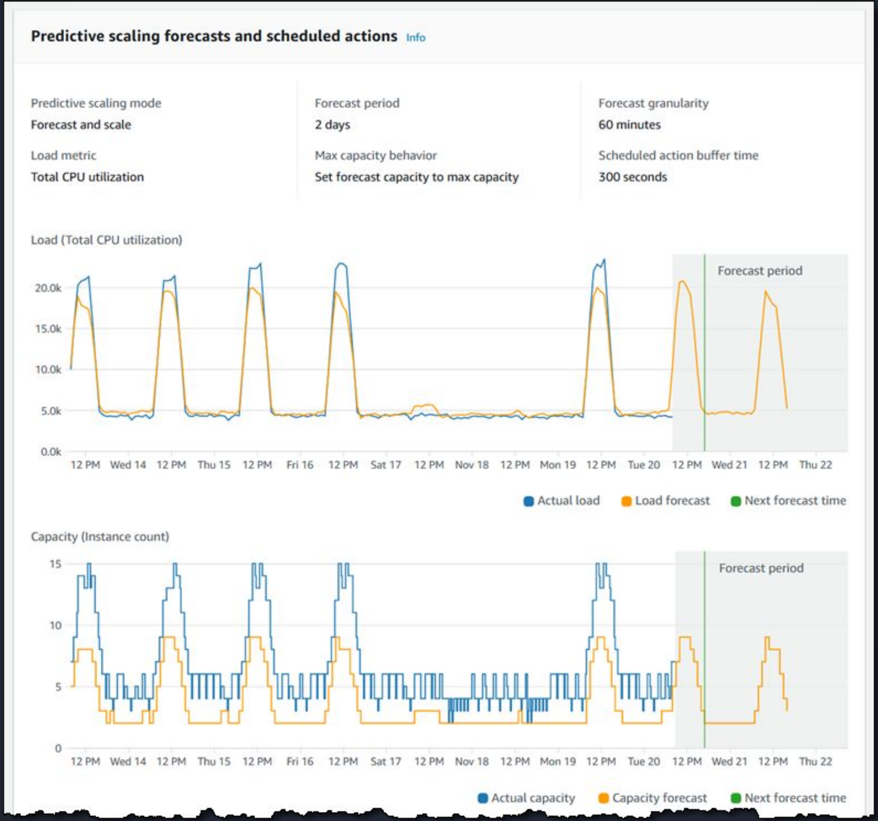
FIBO CLOUD EDUCATION

4. Scale based on demand

A more advanced way to scale your resources. Using scaling policies lets you define parameters that control that scaling process. Ex: CPU load > 75%

5. Use predictive scaling

Combination with ASG to scale resource across multiple services.



Predictive scaling forecasts and scheduled actions Info

Predictive scaling mode	Forecast period	Forecast granularity
Forecast and scale	2 days	60 minutes
Load metric	Max capacity behavior	Scheduled action buffer time
Total CPU utilization	Set forecast capacity to max capacity	300 seconds

Load (Total CPU utilization)
Forecast period
20.0k
15.0k
10.0k
5.0k
0.0k
12 PM Wed 14 12 PM Thu 15 12 PM Fri 16 12 PM Sat 17 12 PM Nov 18 12 PM Mon 19 12 PM Tue 20 12 PM Wed 21 12 PM Thu 22
● Actual load ● Load forecast ● Next forecast time
Capacity (Instance count)
Forecast period
15
10
5
0
12 PM Wed 14 12 PM Thu 15 12 PM Fri 16 12 PM Sat 17 12 PM Nov 18 12 PM Mon 19 12 PM Tue 20 12 PM Wed 21 12 PM Thu 22
● Actual capacity ● Capacity forecast ● Next forecast time

FIBO CLOUD EDUCATION

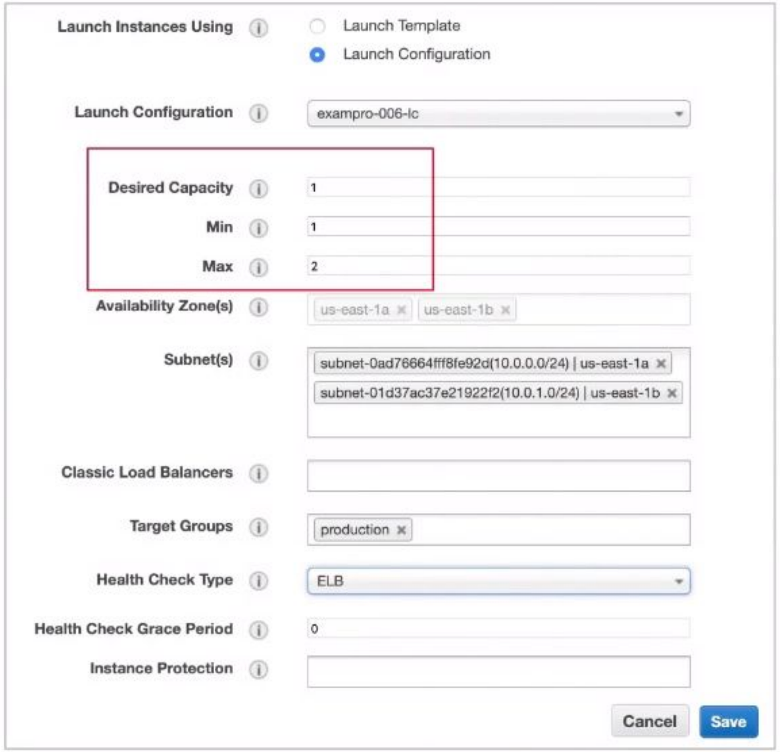
The size of an Auto Scaling Group is based on **Min**, **Max** and **Desired Capacity**.

Min is how many EC2 instances should at least be running.

Max is number EC2 instances allowed to be running.

Desired Capacity is how many EC2 instances you want to ideally run.

ASG will always launch instances to meet minimum capacity.

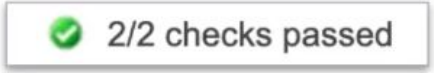


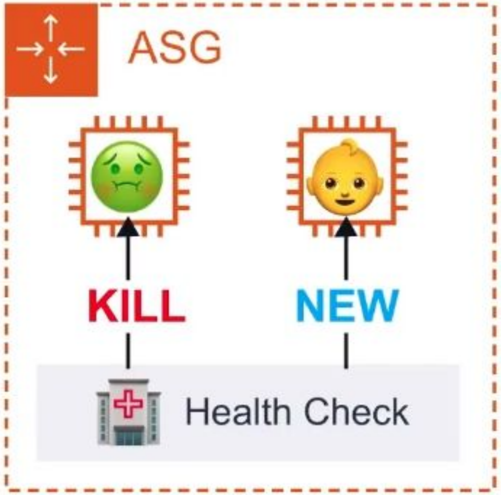
Launch Instances Using	Launch Template [✓ Launch Configuration]
Launch Configuration	exampro-006-lc
Desired Capacity	1
Min	1
Max	2
Availability Zone(s)	us-east-1a, us-east-1b
Subnet(s)	subnet-0ad76664fff8fe92d(10.0.0.0/24) | us-east-1a subnet-01d37ac37e21922f2(10.0.1.0/24) | us-east-1b
Classic Load Balancers	
Target Groups	production
Health Check Type	ELB
Health Check Grace Period	0
Instance Protection	

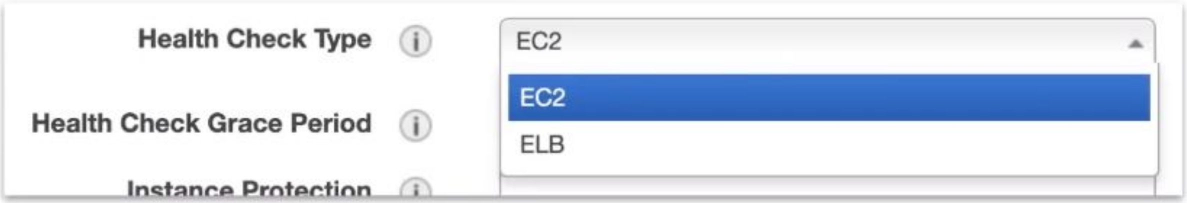
FIBO CLOUD EDUCATION

EC2 Health Check Type

ASG will perform a health check on EC2 instances to determine if there is a software or hardware issue. This is based on the **EC2 Status Checks**. If an instance is considered unhealthy. ASG will terminate and launch a new instance.


2/2 checks passed




Health Check Type: EC2

EC2

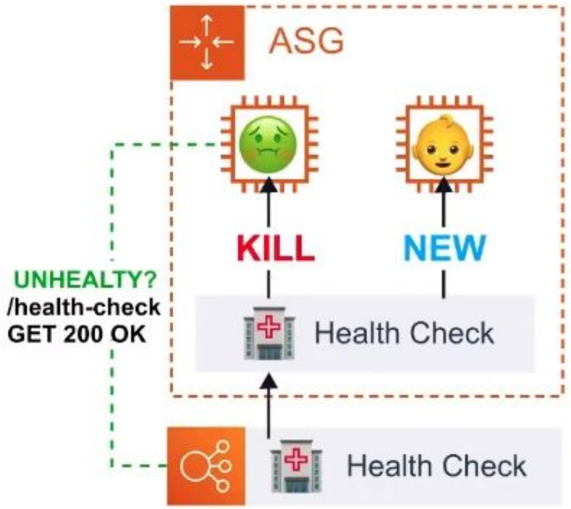
ELB

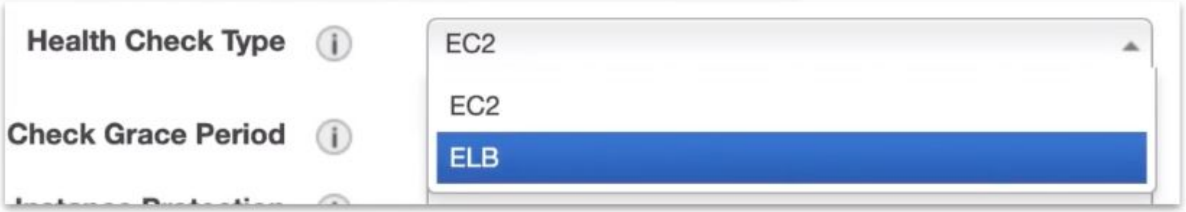
Health Check Grace Period

Instance Protection

ELB Health Check Type

ASG will perform a health check based on the ELB health check. ELB can perform health checks by pinging an HTTP(S) endpoint with an expected response. If ELB determines a instance is unhealthy it forwards this information to ASG which will terminate the unhealthy instance.





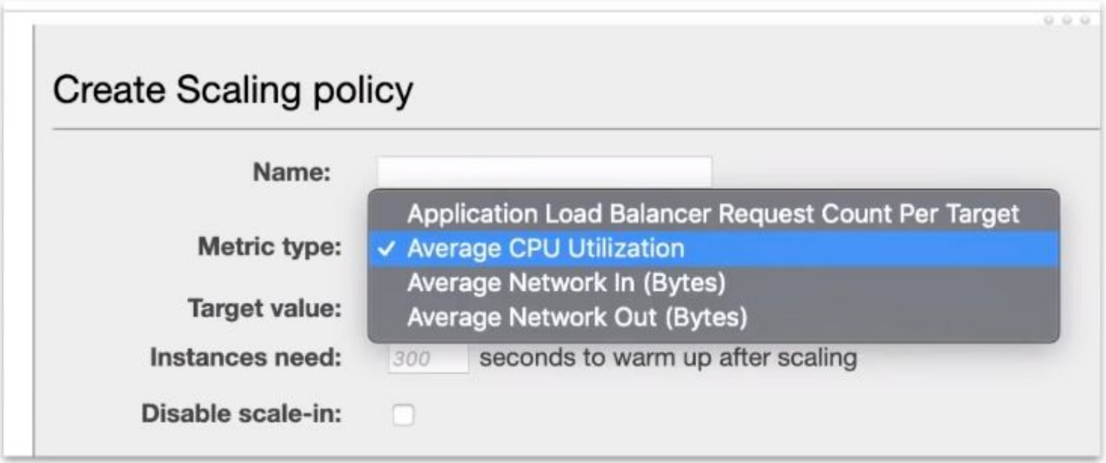
Health Check Type	EC2
Check Grace Period	

Scaling Out: Adding More Instances **Scaling In:** Removing Instances

Target Tracking Scaling Policy

Maintains a specific metric at a target value.

eg. If **Average CPU Utilization** exceeds 75% then add another server.



Create Scaling policy

Name:

Metric type:

Target value:

Instances need: 300 seconds to warm up after scaling

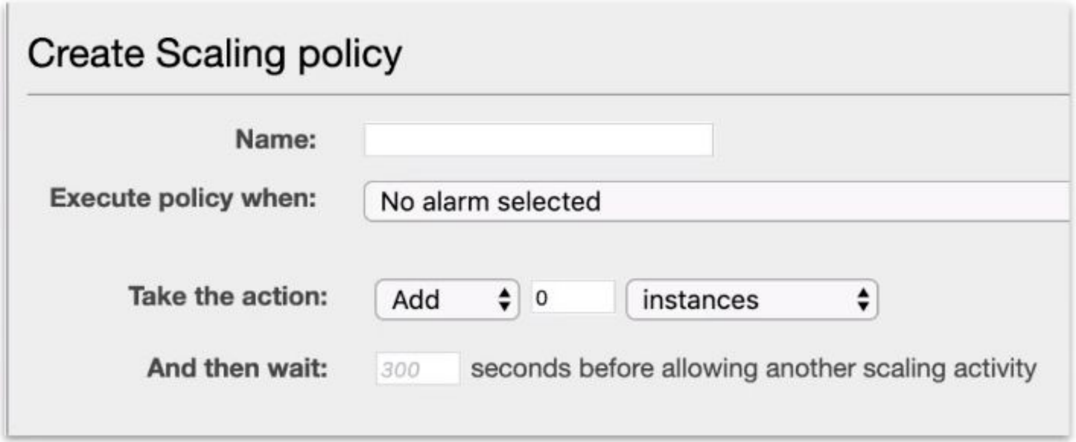
Disable scale-in: ☐

Application Load Balancer Request Count Per Target
✓ Average CPU Utilization
Average Network In (Bytes)
Average Network Out (Bytes)

FIBO
CLOUD
EDUCATION

Simple Scaling Policy

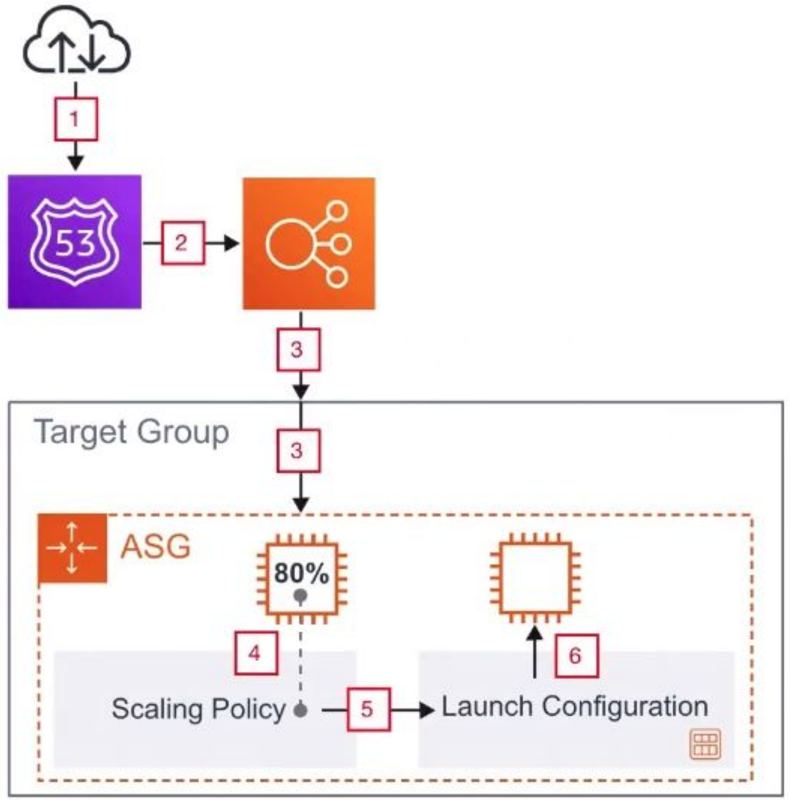
Scales when an **alarm is breached**.


Create Scaling policy

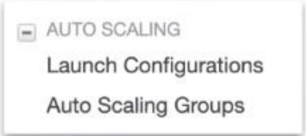
Name:	
Execute policy when:	No alarm selected
Take the action:	Add 0 instances
And then wait:	300 seconds before allowing another scaling activity

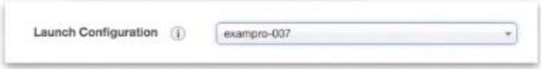
Not recommended, legacy scaling policy. Use scaling policies with steps now.

1. Burst of traffic from the internet hits our domain.
2. Route53 points that traffic to our load balancer.
3. Our load balancer passes the traffic to its target group.
4. The target group is associated with our ASG and sends the traffic to instances registered with our ASG
5. The ASG Scaling Policy will check if our instances are near capacity.
6. The Scaling Policy determines we need another instance, and it Launches an new EC2 instance with the associated Launch Configuration to our ASG

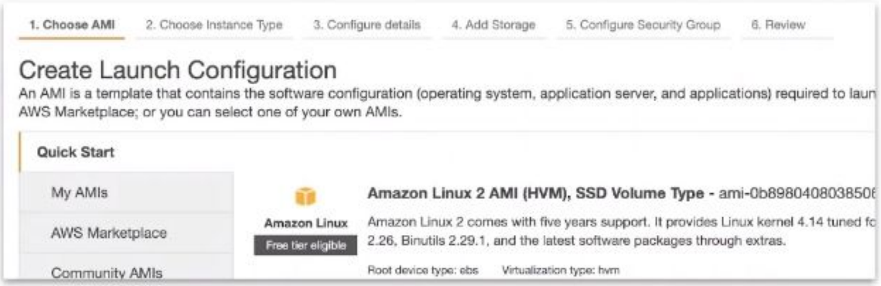


A launch configuration is an instance configuration template that an Auto Scaling group uses to launch EC2 instances.




Launch Configuration: exampro-007

A Launch Configuration is the same process as Launching an EC2 instance except you are saving that configuration to Launch an Instance for later. Hence "Launch Configuration.



Create Launch Configuration

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to laun
AWS Marketplace; or you can select one of your own AMIs.

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0b898040803850

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned fo
2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebs   Virtualization type: hvm

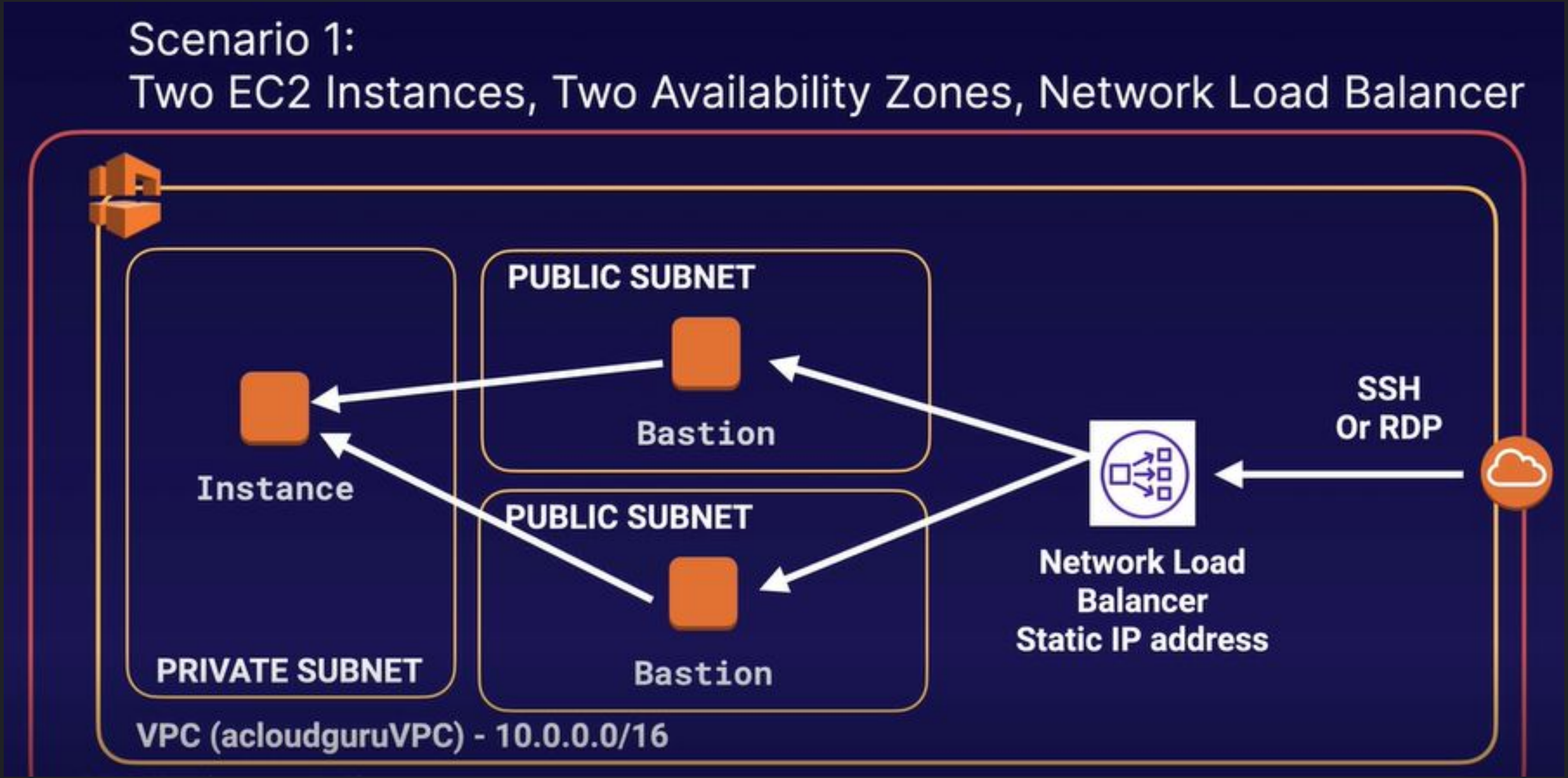
Launch Configurations **cannot be edited**, When you need to update your Launch Configuration you create a new one or clone the existing configuration and then manually associate that new Launch Configuration

Launch Templates are Launch Configurations with Versioning, Everyone appears to still use Launch Configurations

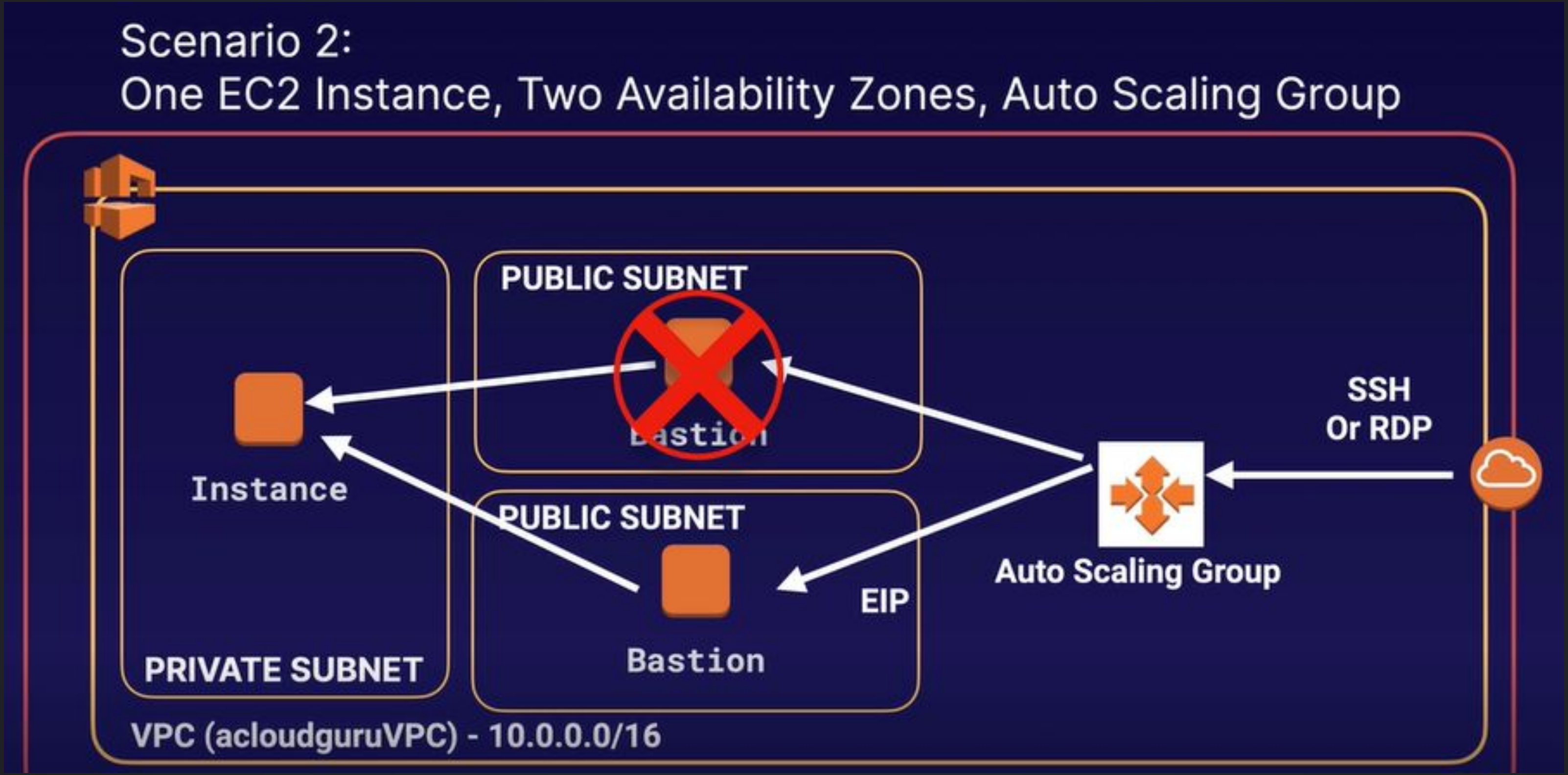
```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
EC2ID=$(curl -s http://169.254.169.254/latest/meta-data/instance-id)
EC2AZ=$(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone)
echo '<html><h1>My EC2: INID    AZ: AZID</h1></html>' > /var/www/html/index.txt
sed -e "s/AZID/$EC2AZ/g" -e "s/INID/$EC2ID/g" /var/www/html/index.txt >
/var/www/html/index.html
```

```
sudo amazon-linux-extras install epel -y
sudo yum install stress -y
```

HA Bastions Hosts

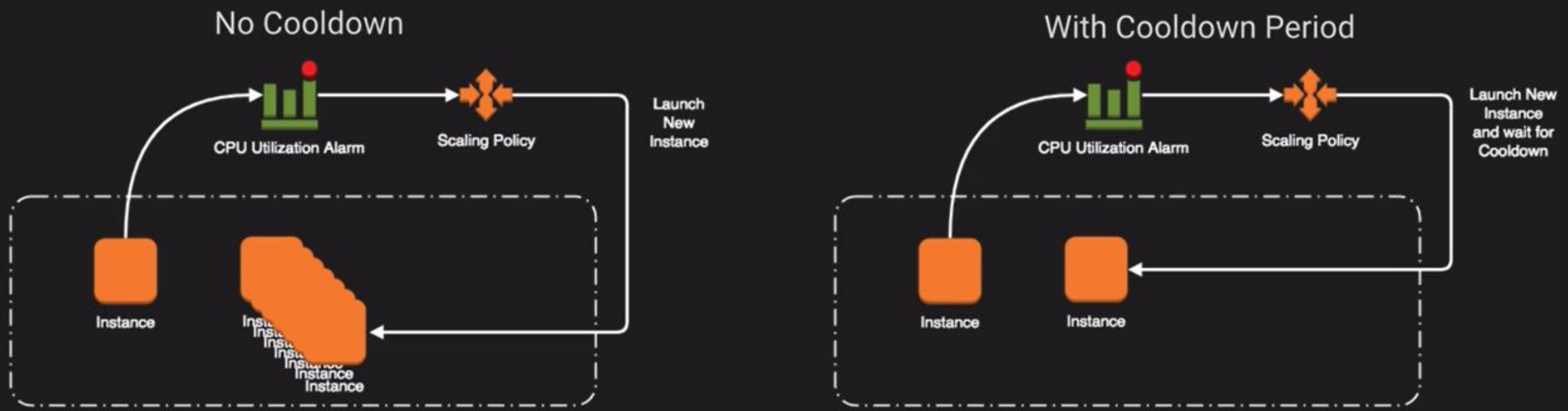

Scenario 1:
Two EC2 Instances, Two Availability Zones, Network Load Balancer
PUBLIC SUBNET
Bastion
PUBLIC SUBNET
Bastion
Instance
PRIVATE SUBNET
SSH Or RDP
Network Load Balancer Static IP address
VPC (acloudguruVPC) - 10.0.0.0/16

HA Bastions Hosts

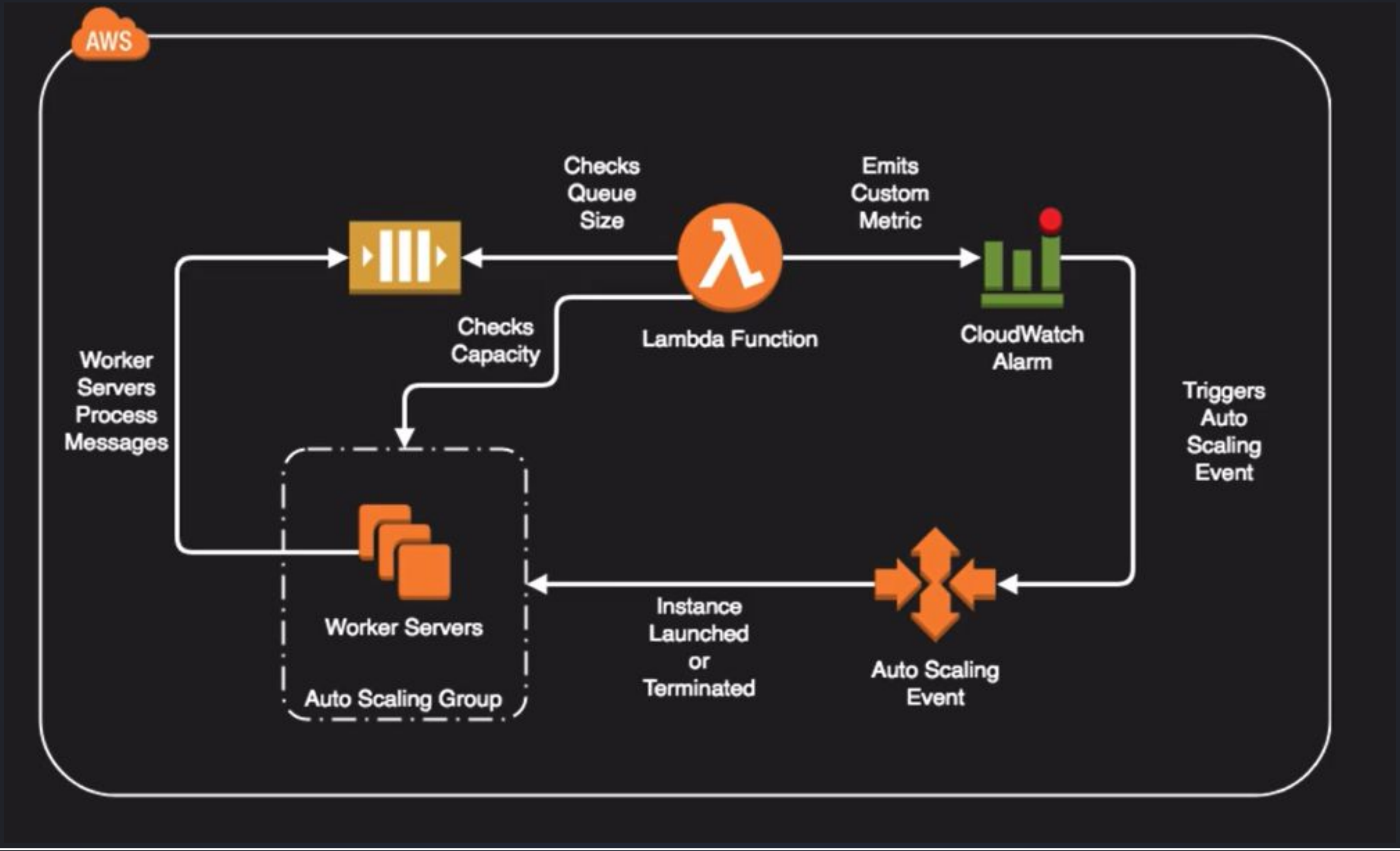


Cooldown Periods

- Configurable duration that gives your scaling a chance to “come up to speed” and absorb load.
- Default cooldown period is 300 seconds.
- Automatically applies to dynamic scaling and optionally to manual scaling but not supported for scheduled scaling.
- Can override default cooldown via scaling-specific cool down



Scaling with SQS


AWS
Checks Queue Size
Emits Custom Metric
Lambda Function
CloudWatch Alarm
Checks Capacity
Worker Servers Process Messages
Triggers Auto Scaling Event
Worker Servers
Auto Scaling Group
Instance Launched or Terminated
Auto Scaling Event