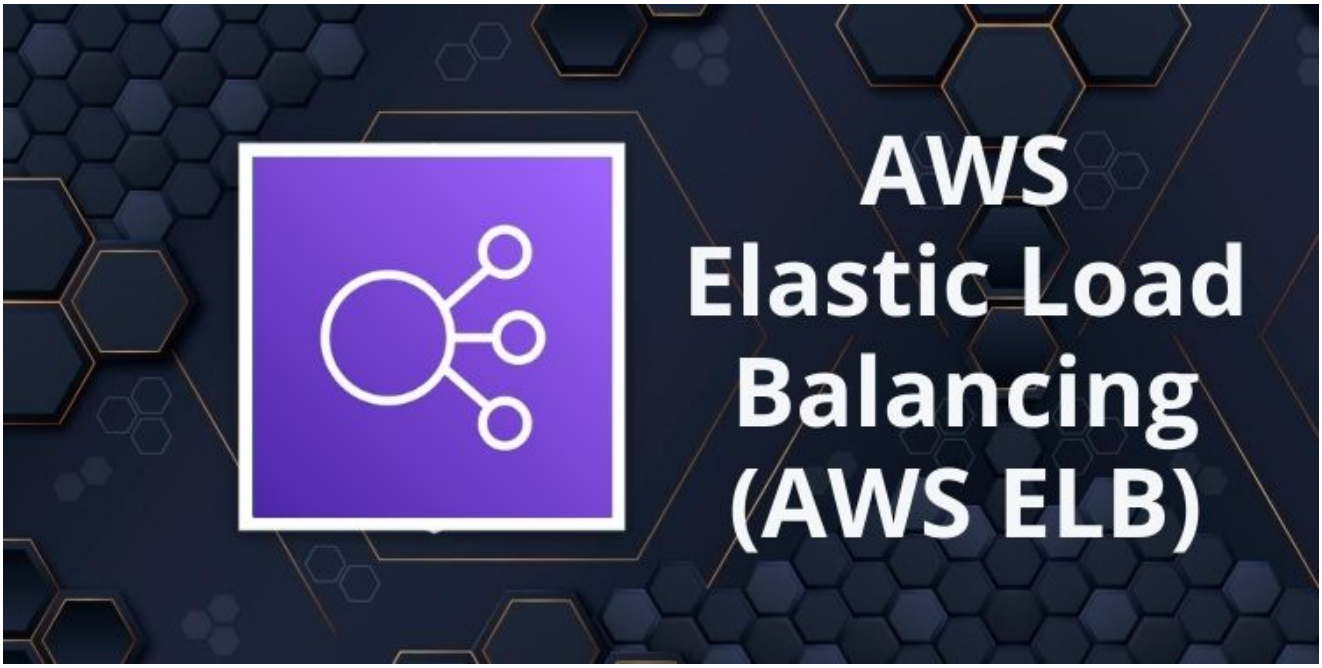
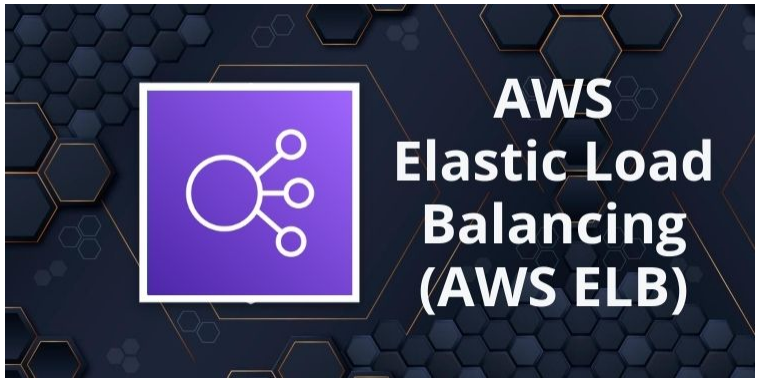
AWS Elastic Load Balancing (AWS ELB)

digitalcloud.training/aws-elastic-load-balancing-aws-elb/

January 5, 2022



Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses.



Network traffic can be distributed across a single or multiple Availability Zones (AZs) within an AWS Region.

There are four types of Elastic Load Balancer (ELB) on AWS:

- **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.
- **Gateway Load Balancer (GLB)** – layer 3/4 load balancer used in front of virtual appliances such as firewalls and IDS/IPS systems.

Note: *The CLB is not covered in detail on this page as it is on old generation load balancer and is no longer featured on most AWS exams.*

The following table provides a comparison of some of the key features relevant to AWS exams:

Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer	Gateway Load Balancer
OSI Layer	Layer 7	Layer 4	Layer 4/7	Layer 3 Gateway + Layer 4 Load Balancing
Target Type	IP, Instance, Lambda	IP, Instance, ALB	N/A	IP, Instance
Protocols	HTTP, HTTPS	TCP	TCP, SSL, HTTP, HTTPS	IP
WebSockets	✔	✔		✔
IP addresses as a target	✔	✔		
HTTP header-based routing	✔			
HTTP/2/gRPC	✔			
Configurable idle connection timeout	✔		✔	
Cross-zone load balancing	✔	✔	✔	✔
SSL Offloading	✔	✔	✔	
Server Name Indication (SNI)	✔	✔	✔	
Sticky sessions	✔	✔	✔	✔
Static / Elastic IP Address		✔		
Custom Security policies			✔	

Elastic Load Balancing provides fault tolerance for applications by automatically balancing traffic across targets –

Targets can be Amazon EC2 instances, containers, IP addresses, and Lambda functions.

ELB distributes traffic across Availability Zones while ensuring only healthy targets receive traffic.

Only 1 subnet per AZ can be enabled for each ELB.

Amazon Route 53 can be used for region load balancing with ELB instances configured in each region.

ELBs can be **Internet** facing or **internal-only.**

Internet facing ELB:

- ELB nodes have public IPs.
- Routes traffic to the private IP addresses of the EC2 instances.
- Need one public subnet in each AZ where the ELB is defined.
- ELB DNS name format: <name>-<id-number>.<region>.elb.amazonaws.com.

Internal only ELB:

- ELB nodes have private IPs.
- Routes traffic to the private IP addresses of the EC2 instances.
- ELB DNS name format: **internal**-<name>-<id-number>.<region>.elb.amazonaws.com.

Internal-only load balancers do not need an Internet gateway.

EC2 instances and containers can be registered against an ELB.

ELB nodes use IP addresses within your subnets, ensure at least a /27 subnet and make sure there are at least 8 IP addresses available for the ELB to scale.

An ELB forwards traffic to eth0 (primary IP address).

An ELB listener is the process that checks for connection requests:

- CLB listeners support the TCP and HTTP/HTTPS protocols.
- ALB listeners support the HTTP and HTTPS protocols.
- NLB listeners support the TCP, UDP and TLS protocols.
- GLB listeners support the IP protocol.

Deleting an ELB does not affect the instances registered against it (they won’t be deleted; they just won’t receive any more requests).

For ALB at least 2 subnets must be specified.

For NLB only one subnet must be specified (recommended to add at least 2).

ELB uses a DNS record TTL of 60 seconds to ensure new ELB node IP addresses are used to service clients.

By default the ELB has an idle connection timeout of 60 seconds, set the idle timeout for applications to at least 60 seconds.

Perfect Forward Secrecy (PFS) provides additional safeguards against the eavesdropping of encrypted data, using a unique random session key.

Server Order Preference lets you configure the load balancer to enforce cipher ordering, providing more control over the level of security used by clients to connect with your load balancer.

ELB does not support client certificate authentication (API Gateway does support this).

ELB Security Groups

Security groups control the ports and protocols that can reach the front-end listener.

In non-default VPCs you can choose which security group to assign.

You must assign a security group for the ports and protocols on the front-end listener.

You need to also allow the ports and protocols for the health check ports and back-end listeners.

Security group configuration for ELB:

Inbound to ELB (allow).

Internet-facing ELB:

- Source: 0.0.0.0/0.
- Protocol: TCP.
- Port: ELB listener ports.

sg-6b578e13 | Internet-Facing ELB

Summary | **Inbound Rules** | Outbound Rules | Tags

Edit

Type	Protocol	Port Range	Source	Description
HTTP (80)	TCP (6)	80	0.0.0.0/0	Inbound HTTP
HTTPS (443)	TCP (6)	443	0.0.0.0/0	Inbound HTTPS

Source is set to any address

Internal-only ELB:

- Source: VPC CIDR.
- Protocol: TCP.
- Port: ELB Listener ports.

sg-754e970d | Internal-Only ELB

Summary | **Inbound Rules** | Outbound Rules | Tags

Edit

Type	Protocol	Port Range	Source	Description
HTTP (80)	TCP (6)	80	10.0.0.0/16	HTTP Listener
HTTPS (443)	TCP (6)	443	10.0.0.0/16	HTTPS Listener

Source is set to the VPC CIDR

Outbound (allow, either type of ELB):

- Destination: EC2 registered instances security group.
- Protocol: TCP.
- Port: Health Check/Listener.

sg-6b578e13 | Internet-Facing ELB

Summary | Inbound Rules | **Outbound Rules** | Tags

Edit

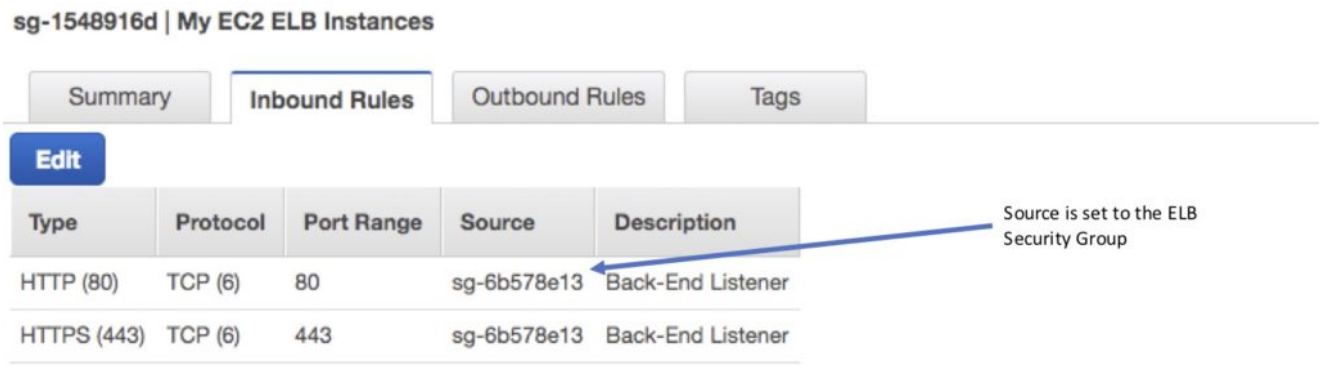
Type	Protocol	Port Range	Destination	Description
HTTP (80)	TCP (6)	80	sg-1548916d	Health Check/Listene...
HTTPS (443)	TCP (6)	443	sg-1548916d	HTTPS Listener

Destination is set to the EC2 registered instances Security Group

Security group configuration for registered instances:

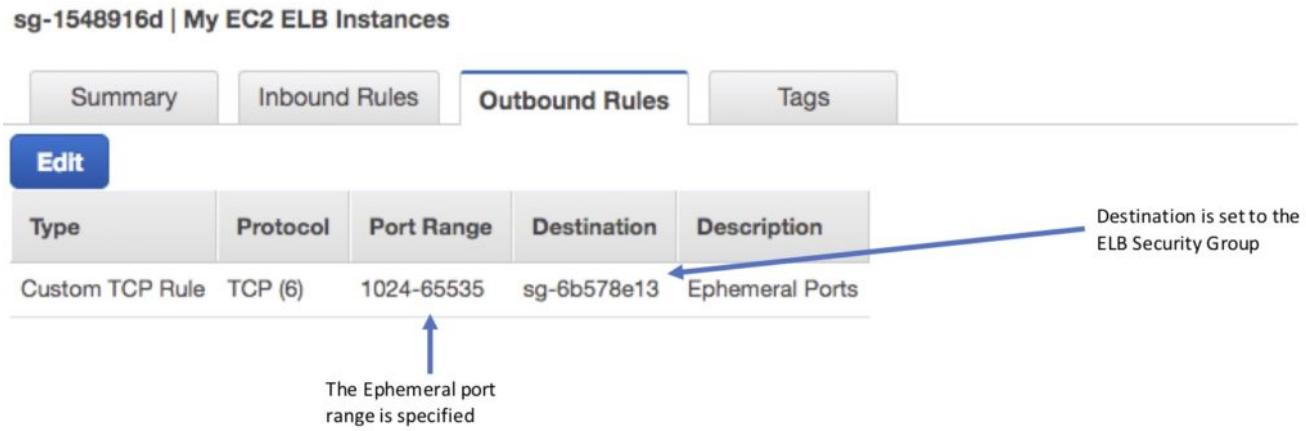
Inbound to registered instances (Allow, either type of ELB).

- Source: ELB Security Group.
- Protocol: TCP.
- Port: Health Check/Listener.



Outbound (Allow, for both types of ELB).

- Destination: ELB Security Group.
- Protocol: TCP.
- Port: Ephemeral.



It is also important to ensure NACL settings are set correctly.

Distributed Denial of Service (DDoS) protection:

ELB automatically distributes incoming application traffic across multiple targets, such as EC2 instances, containers, and IP addresses, and multiple Availability Zones, which minimizes the risk of overloading a single resource.

ELB only supports valid TCP requests so DDoS attacks such as UDP and SYN floods are not able to reach EC2 instances.

ELB also offers a single point of management and can serve as a line of defense between the internet and your backend, private EC2 instances.

You can also attach AWS Web Application Firewall (WAF) Web ACLs to Application Load Balancers to protect against web exploits.

ELB Monitoring

Monitoring takes place using:

- CloudWatch – every 1 minute.
  - ELB service only sends information when requests are active.
  - Can be used to trigger SNS notifications.
- Access Logs.
  - Disabled by default.
  - Includes information about the clients (not included in CloudWatch metrics).
  - Can identify requester, IP, request type etc.
  - Can be optionally stored and retained in S3.
- CloudTrail.
  - Can be used to capture API calls to the ELB.
  - Can be stored in an S3 bucket.

Target groups

Target groups are a logical grouping of targets and are used with ALB, NLB, and GLB.

Targets are the endpoints and can be EC2 instances, ECS containers, IP addresses, Lambda functions, and other load balancers.

Target groups can exist independently from the ELB.

A single target can be in multiple target groups.

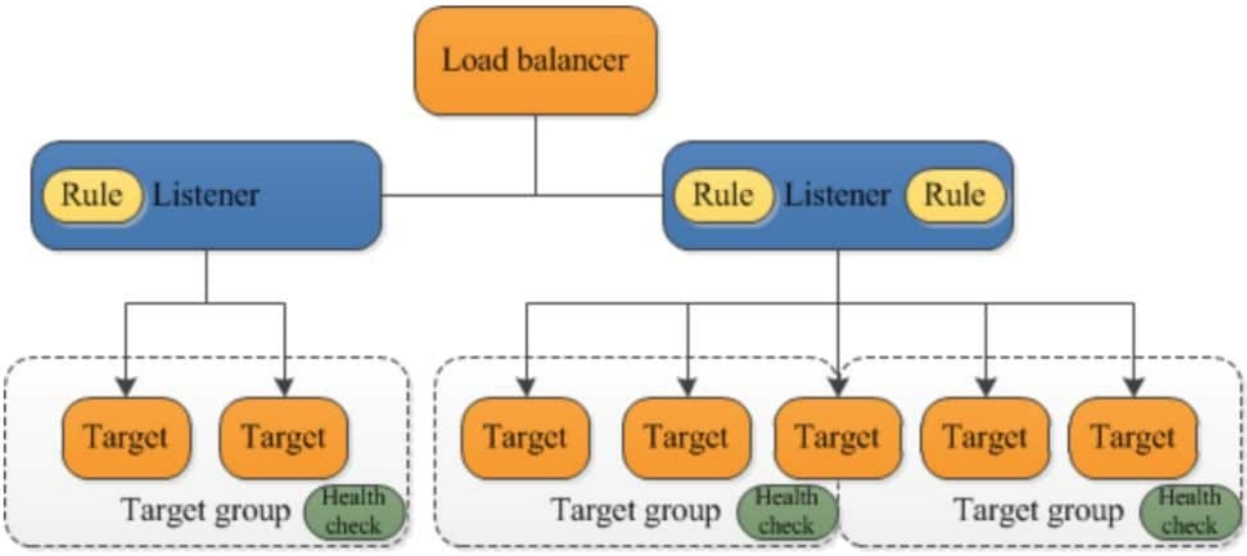
Only one protocol and one port can be defined per target group.

You cannot use public IP addresses as targets.

You cannot use instance IDs and IP address targets within the same target group.

A target group can only be associated with one load balancer.

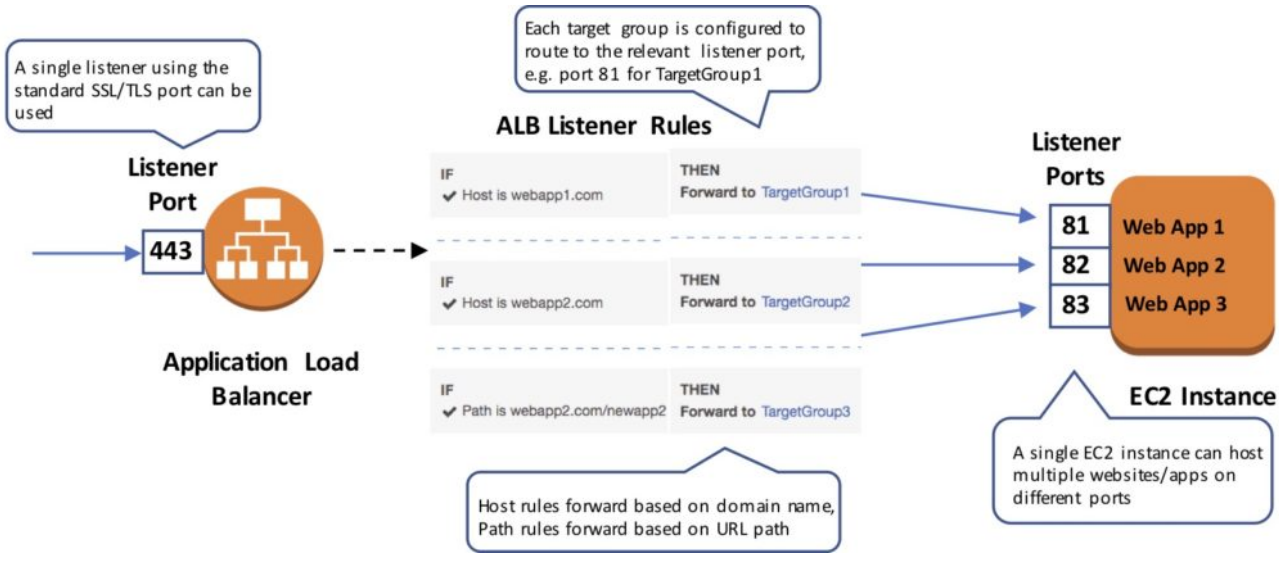
The following diagram illustrates the basic components. Notice that each listener contains a default rule, and one listener contains another rule that routes requests to a different target group. One target is registered with two target groups.



Target groups are used for registering instances against an ALB, NLB, or GLB.

Target groups are a regional construct (as are ELBs).

The following diagram shows how target groups can be used with an ALB using host-based and target-based routing to route traffic to multiple websites, running on multiple ports, on a single EC2 instance:

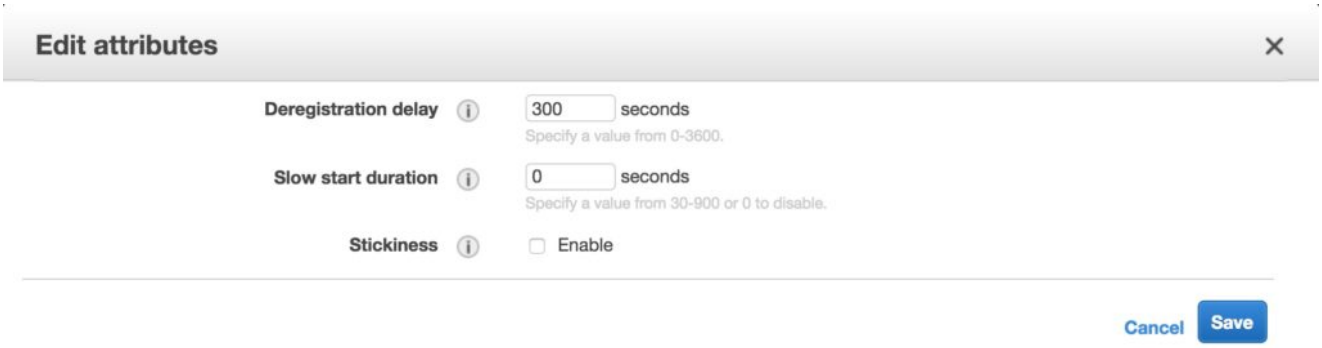


The following attributes can be defined:

- Deregistration delay – the amount of time for Elastic Load Balancing to wait before deregistering a target.

- Slow start duration – the time, in seconds, during which the load balancer sends a newly registered target a linearly increasing share of the traffic to the target group.
- Stickiness – indicates whether sticky sessions are enabled.

The default settings for attributes are shown below:



Auto Scaling groups can scale each target group individually.

You can only use Auto Scaling with the load balancer if using instance IDs in your target group.

Health checks are defined per target group.

ALB/NLB/GLB can route to multiple target groups.

You can register the same EC2 instance or IP address with the same target group multiple times using different ports (used for routing requests to microservices).

If you register by instance ID the traffic is routed using the primary private IP address of the primary network interface.

If you register by IP address you can route traffic to an instance using any private address from one or more network interfaces.

You cannot mix different types within a target group (EC2, ECS, IP, Lambda function).

IP addresses can be used to register:

- Instances in a peered VPC.
- AWS resources that are addressable by IP address and port.
- On-premises resources linked to AWS through Direct Connect or a VPN connection.

Application Load Balancer (ALB)

The Application Load Balancer operates at the request level (layer 7), routing traffic to targets – EC2 instances, containers and IP addresses based on the content of the request.

You can load balance HTTP/HTTPS applications and use layer 7-specific features, such as X-Forwarded-For headers.

The ALB supports HTTPS termination between the clients and the load balancer.

The ALB supports management of SSL certificates through AWS IAM and AWS Certificate Manager for predefined security policies.

The ALB supports Server Name Indication (SNI) which allows multiple secure websites to use a single secure listener.

With Server Name Indication (SNI) a client indicates the hostname to which it wants to connect.

IP addresses can be configured as targets which allows load balancing to applications hosted in AWS or on-premises using the IP addresses of the back-end instances/servers as targets.

You need at least two Availability Zones, and you can distribute incoming traffic across your targets in multiple Availability Zones.

The ALB automatically scales its request handling capacity in response to incoming application traffic.

You can configure an ALB to be Internet facing or create a load balancer without public IP addresses to serve as an internal (non-Internet-facing) load balancer.

The ALB supports content-based routing which allows the routing of requests to a service based on the content of the request. For example:

- **Host-based routing** routes client requests based on the Host field of the HTTP header allowing you to route to multiple domains from the same load balancer.
- **Path-based routing** routes a client request based on the URL path of the HTTP header (e.g. /images or /orders).

Support for microservices and containers with load balancing across multiple ports on a single EC2 instance.

Integration with Amazon Cognito for user authentication.

The slow start mode allows targets to “warm up” with a ramp-up period.

Health Checks:

- Can have custom response codes in health checks (200-399).
- Details provided in the API and management console for health check failures.

- Reason codes are returned with failed health checks.
- Health checks do not support WebSockets.
- Fail open means that if no AZ contains a healthy target the load balancer nodes route requests to all targets.

Detailed access log information is provided and saved to an S3 bucket every 5 or 6 minutes.

Deletion protection is possible.

Deregistration delay is like connection draining.

Sticky Sessions:

- Uses cookies to ensure a client is bound to an individual back-end instance for the duration of the cookie lifetime.
- ALB supports duration-based cookies and application-based cookies.
- For application-based cookies the cookie name is specified for each target group.
- For duration-based cookies the name of the cookie is always AWSALB.
- Sticky sessions are enabled at the target group level.
- WebSockets connections are inherently sticky (following the upgrade process).

Listeners and Rules

Listeners:

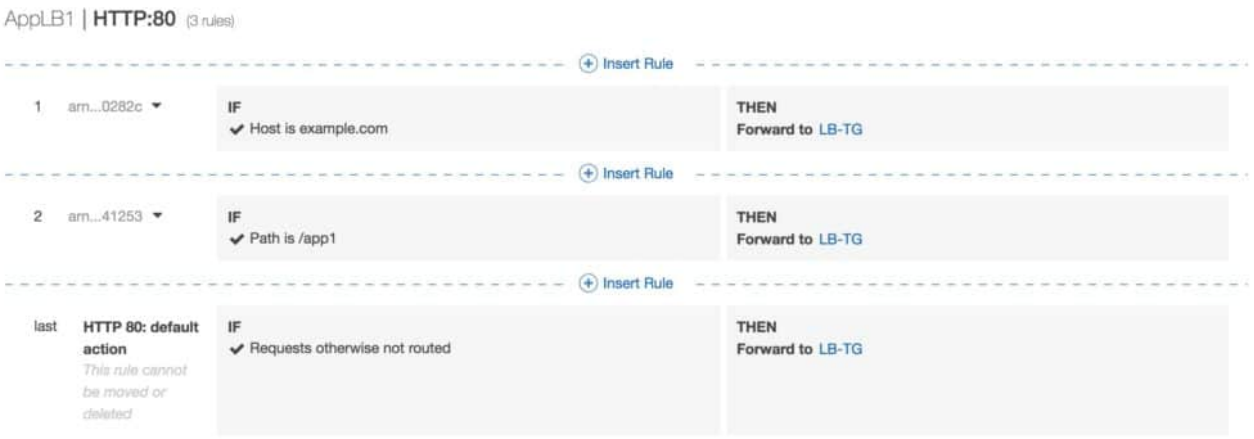
- Each ALB needs at least one listener and can have up to 10.
- Listeners define the port and protocol to listen on.
- Can add one or more listeners.
- Cannot have the same port in multiple listeners.

Listener rules:

- Rules determine how the load balancer routes requests to the targets in one or more target groups.
- Each rule consists of a priority, one or more actions, an optional host condition, and an optional path condition.
- Only one action can be configured per rule.
- One or more rules are required.
- Each listener has a default rule, and you can optionally define additional rules.
- Rules determine what action is taken when the rule matches the client request.
- Rules are defined on listeners.
- You can add rules that specify different target groups based on the content of the request (content-based routing).

- If no rules are found the default rule will be followed which directs traffic to the default target groups.

The image below shows a ruleset with a host-based and path-based entry and a default rule at the end:



Default rules:

- When you create a listener you define an action for the default rule.
- Default rules cannot have conditions.
- You can delete the non-default rules for a listener at any time.
- You cannot delete the default rule for a listener.
- When you delete a listener all its rules are deleted.
- If no conditions for any of a listener's rules are met, the action for the default rule is taken.

Rule priority:

- Each rule has a priority, and they are evaluated in order of lowest to highest.
- The default rule is evaluated last.
- You can change the value of a non-default rule at any time.
- You cannot change the value of the default rule.

Rule action:

- Only one target group per action.
- Each rule has a type and a target group.
- The only supported action type is forward, which forwards requests to the target group.
- You can change the target group for a rule at any time.

Rule conditions:

- There are two types of rule condition: host and path.

- When the conditions for a rule are met the action is taken.
- Each rule can have up to 2 conditions, 1 path condition and 1 host condition.
- Optional condition is the path pattern you want the ALB to evaluate for it to route requests.

Request routing:

- After the load balancer receives a request it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action using the round robin routing algorithm.
- Routing is performed independently for each target group even when a target is registered with multiple target groups.
- You can configure listener rules to route requests to different target groups based on the content of the application traffic.

Content-based routing:

- ALB can route requests based on the content of the request in the host field: host-based or path-based.
- Host-based is domain name-based routing e.g. example.com or app1.example.com.
- The host field contains the domain name and optionally the port number.
- Path-based is URL based routing e.g. example.com/images, example.com/app1.
- You can also create rules that combine host-based and path-based routing.
- Anything that doesn't match content routing rules will be sent to a default target group.
- The ALB can also route based on other information in the HTTP header including query string parameters, request method, and source IP addresses

ALB and ECS

ECS service maintains the "desired count" of instances.

Optionally a load balancer can distribute traffic across tasks.

All containers in a single task definition are placed on a single EC2 container instance.

ECS service can only use a single load balancer.

ALB does not support multiple listeners in a single task definition.

ALB supports dynamic host-port mapping which means that multiple ports from the same service are allowed on the same container host.

ALB supports path-based routing and priority rules.

The ALB integrates with the EC2 container service using service load balancing.

Federated authentication:

- ALB supports authentication from OIDC compliant identity providers such as Google, Facebook, and Amazon.
- Implemented through an authentication action on a listener rule that integrates with Amazon Cognito to create user pools.
- AWS SAM can also be used with Amazon Cognito.

Network Load Balancer

Network Load Balancer operates at the connection level (Layer 4), routing connections to targets – Amazon EC2 instances, containers and IP addresses based on IP protocol data.

The NLB is designed to handle millions of requests/sec and to support sudden volatile traffic patterns at extremely low latencies.

The NLB can be configured with a single static/Elastic IP address for each Availability Zone.

You can load balance any application hosted in AWS or on-premises using IP addresses of the application back-ends as targets.

The NLB supports connections from clients to IP-based targets in peered VPCs across different AWS Regions.

NLB supports both network and application target health checks.

NLB supports long-running/lived connections (ideal for WebSocket applications).

NLB supports failover between IP addresses within and across AWS Regions (uses Amazon Route 53 health checks).

The integration with Amazon Route 53 enables the removal of a failed load balancer IP address from service and subsequent redirection of traffic to an alternate NLB in another region.

NLB support cross-zone load balancing, but it is not enabled by default when the NLB is created through the console.

Target groups for NLBs support the following protocols and ports:

- **Protocols:** TCP, TLS, UDP, TCP_UDP.
- **Ports:** 1-65535.

The table below summarizes the supported listener and protocol combinations and target group settings:

Listener Protocol	Target Group Protocol	Target Group Type	Health Check Protocol
TCP	TCP | TCP_UDP	instance | ip	HTTP | HTTPS | TCP
TLS	TCP | TLS	Instance | ip	HTTP | HTTPS | TCP
UDP	UDP | TCP_UDP	instance	HTTP | HTTPS | TCP
TCP_UDP	TCP_UDP	instance	HTTP | HTTPS | TCP

Amazon CloudWatch reports Network Load Balancer metrics.

You can use VPC Flow Logs to record all requests sent to your load balancer.

Monitoring

AWS CloudTrail captures API calls for auditing.

You only pay for the S3 storage charges.

CloudTrail only monitors API calls.

Access logs can be used to monitor other actions such as the time the request was received, the client's IP address, request paths etc.

Access logging is optional and disabled by default.

You are only charged for the S3 storage with access logging.

With access logging, the ALB logs requests sent to the load balancer including requests that never reached targets.

The ALB does not log health check requests.

Related posts: