aws certified

Solutions Architect

Professional

DOLFINed

ELASTIC LOAD BALANCING

AWS Elastic Load Balancer (ELB)

This lecture focuses primarily on : **ELB Introduction**

Review Topic : Elastic Load Balancer

ELB Load Balancer Types in AWS

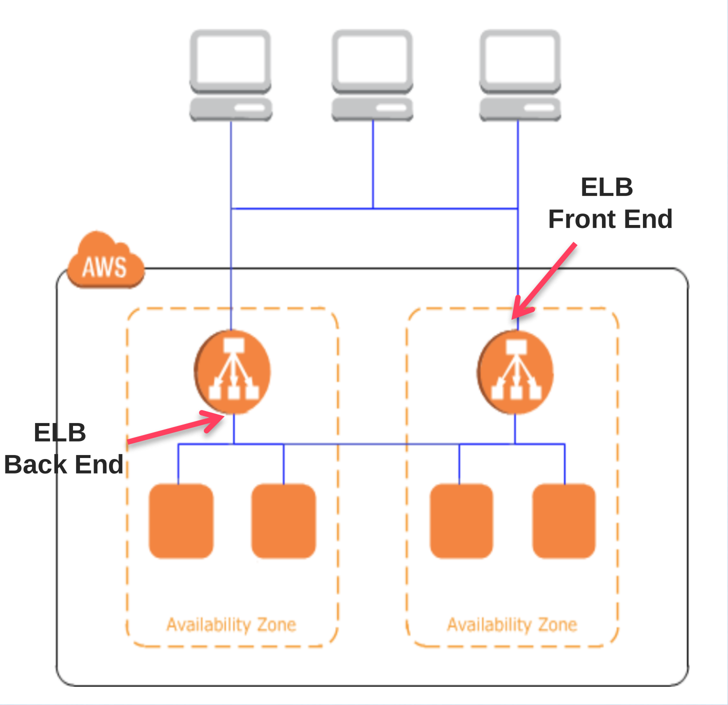
- There are multiple types of Load balancers in the AWS offerings,
  - Classical load balancer – CLB (focus of this section)
  - Application (layer7) load balancer – ALB (Will be explained in the next section)
  - Network Load Balancer – NLB (Will be explained in detail later)

- Classical load balancer (ELB) service supports:
  - HTTP, HTTPS, TCP, SSL (but not HTTP/2)
  - Protocols ports supported are : 1 -> 65535
  - It supports IPv4, IPv6 and Dual stack

Review Topic : Elastic Load Balancer

ELB - Listeners

- An ELB Listener, is the process that checks for connection requests
- You can configure the protocol/port on which your ELB listener listens for connection requests
- Frontend listeners check for traffic from clients to the ELB
- Backend listeners are configured with protocol/port to check for traffic from the ELB to the EC2 instances



Source: aws.amazon.com

Review Topic : Elastic Load Balancer

ELB

- Registered EC2 instances are those that are defined under the ELB
- ELB has “Nothing” to do with the outbound traffic that is initiated/generated from the registered EC2 instances destined directly to the internet, or to any other instances within the VPC.
- ELB has to do only with Inbound traffic destined to the EC2 registered instances (as the destination), and the respective return traffic
- You start to be charged hourly (also for partial hours) once your ELB is active
  - You are also charged for GBs transferred through your Classical Load Balancer
  - You are charged for Load Balancer Capacity Units (LCUs) per hour in case of ALB and NLB

AWS Elastic Load Balancer (ELB)

Classical Load Balancer (CLB) – How it Works

Review Topic : Elastic Load Balancer

ELB – Health Checks

- By default
  - AWS console uses ping HTTP (port 80) for health checks
  - AWS API uses Ping TCP (port 80) for health checks
- Registered instances must respond with a HTTP "200 OK" message within the timeout period, else, it will be considered as unhealthy
- Response timeout is 5 seconds (range 2 – 60 seconds)
- Health check interval:
  - Period of time between health checks
    - Default 30 (range 5 – 300 sec)

Review Topic : Elastic Load Balancer

ELB – Health Checks

- Unhealthy Threshold:
  - Number of consecutive failed health checks that should occur before the instance is declared unhealthy
    - Range 2-10
      - Default 2

- Healthy Threshold:
  - Number of consecutive successful health checks that must occur before the instance is considered healthy
    - Range 2-10
      - Default 10

Review Topic : Elastic Load Balancer

ELB – Cross-Zone Load Balancing

- **Cross-Zone load balancing:**
  - The CLB will distribute traffic evenly between registered EC2 instances in the different AZ’s it load balances to
    - This is to ensure that each registered /healthy instance gets an equal share of traffic from the CLB
    - If you have 5 EC2 instances in one AZ, and 3 in another, cross-zone load balancing will ensure that each registered EC2 instance will be getting around the same amount of traffic load from the ELB

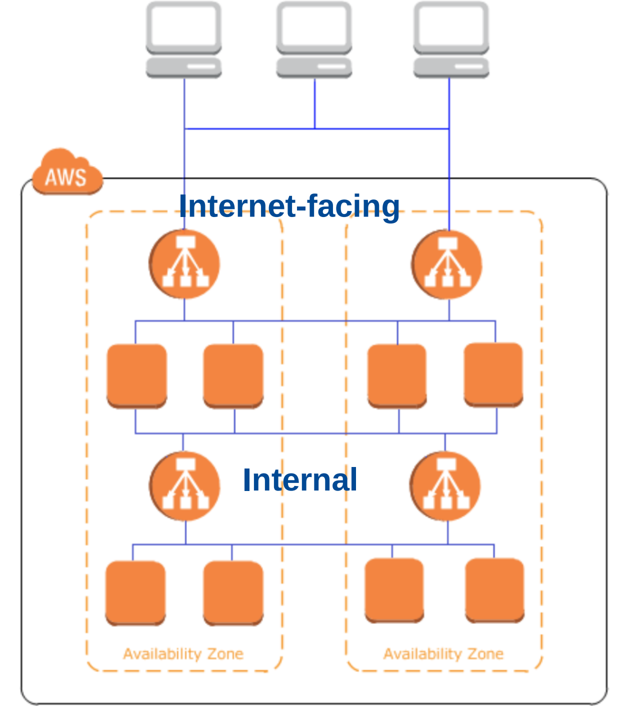
Clastic Load Balancer (CLB)

Positioning

Review Topic : Elastic Load Balancer

An ELB can be **Internet facing** or **Internal load balancer**

- **Internet facing:**
  - ELB nodes will have public IP addresses,
    - DNS will resolve the ELB DNS name to these IP addresses
  - It routes traffic to the private IP addresses of your registered EC2 instances,
    - Hence, why your instances do not have to have public IP addresses
  - You need one “Public” subnet in each AZ where the internet facing ELB will be defined, such that the ELB will be able to route internet traffic
    - Define this subnet in the ELB configuration
- **Internal ELB:**
  - ELB nodes will have private IP addresses, to which the DNS resolves ELB DNS name
  - It routes traffic to the Private IP addresses of your registered EC2 instances



Review Topic : Elastic Load Balancer

CLB – Security Groups

- If you create your CLB in a **default VPC**, you either choose an existing security group for your CLB, or create a new one

  - You must assign a security group to your ELB
    - This will control traffic that can reach your ELB's front end listeners
    - It must also allow health check protocol/ports & listener protocol/port (actual traffic) to reach your registered EC2 instances in the backend

  - You must also ensure that the subnets' N ACLs allow traffic to/from the ELB both ways (on the front and backend side)

Source: aws.amazon.com

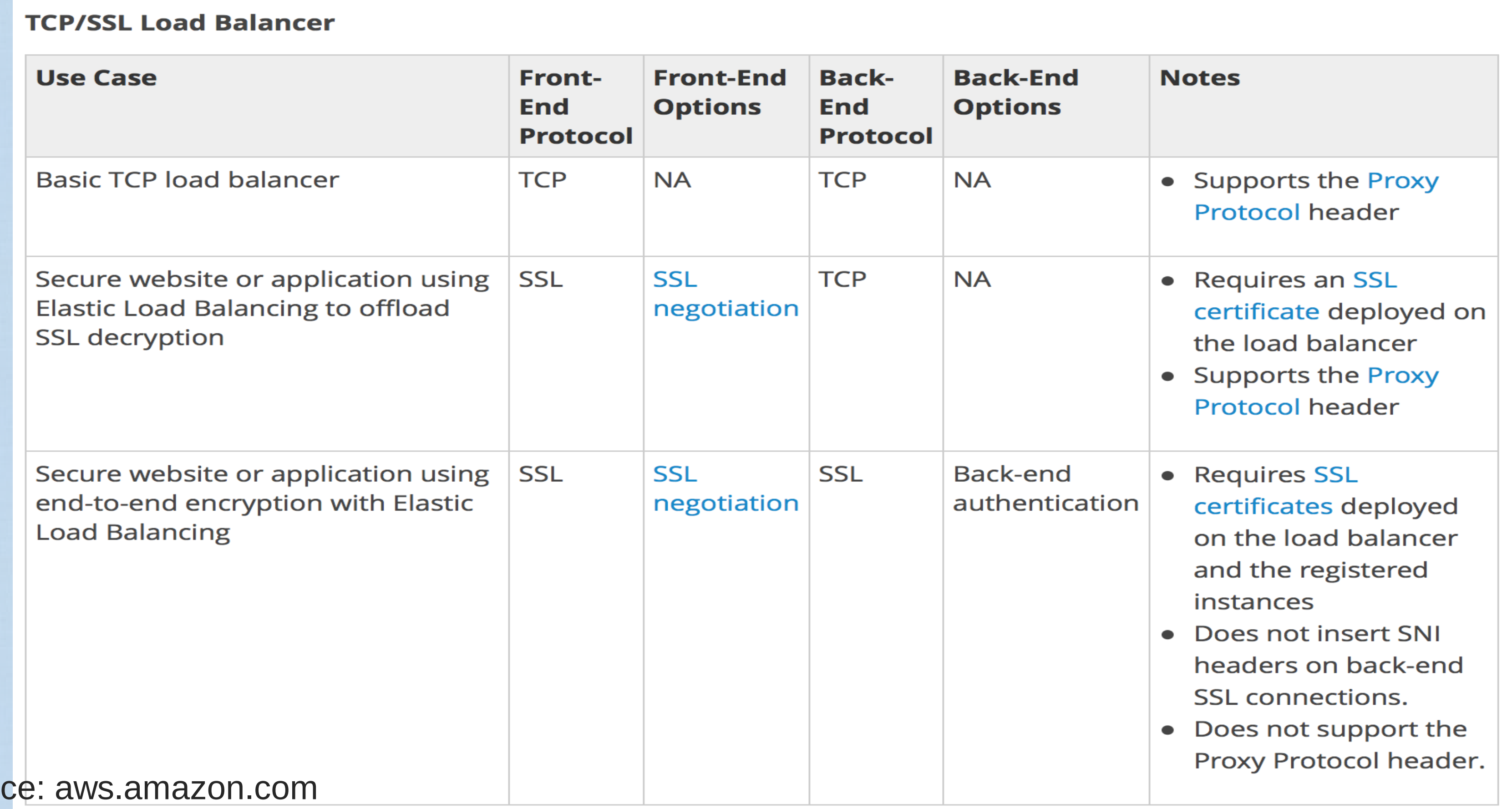
CLASSIC LOAD BALANCER (CLB)

Classic Load Balancer (CLB)

ELB Listeners

Review Topic : Elastic Load Balancer

ELB – TCP/SSL Support



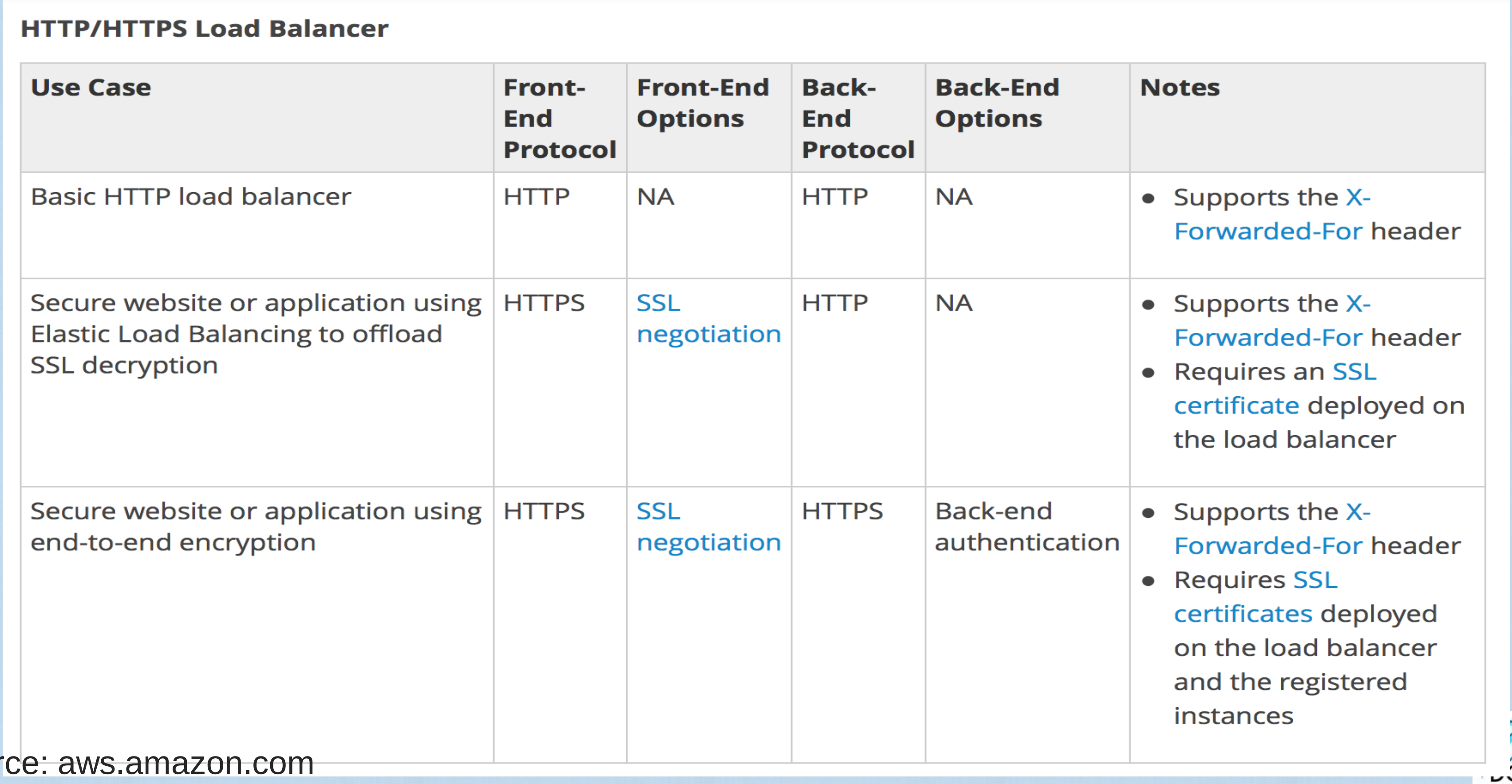
TCP/SSL Load Balancer

Use Case	Front-End Protocol	Front-End Options	Back-End Protocol	Back-End Options	Notes
Basic TCP load balancer	TCP	NA	TCP	NA	• Supports the Proxy Protocol header
Secure website or application using Elastic Load Balancing to offload SSL decryption	SSL	SSL negotiation	TCP	NA	• Requires an SSL certificate deployed on the load balancer • Supports the Proxy Protocol header
Secure website or application using end-to-end encryption with Elastic Load Balancing	SSL	SSL negotiation	SSL	Back-end authentication	• Requires SSL certificates deployed on the load balancer and the registered instances • Does not insert SNI headers on back-end SSL connections. • Does not support the Proxy Protocol header.

Source: aws.amazon.com

Review Topic : Elastic Load Balancer

ELB HTTP/HTTPS – Support



HTTP/HTTPS Load Balancer

Use Case	Front-End Protocol	Front-End Options	Back-End Protocol	Back-End Options	Notes
Basic HTTP load balancer	HTTP	NA	HTTP	NA	• Supports the X-Forwarded-For header
Secure website or application using Elastic Load Balancing to offload SSL decryption	HTTPS	SSL negotiation	HTTP	NA	• Supports the X-Forwarded-For header • Requires an SSL certificate deployed on the load balancer
Secure website or application using end-to-end encryption	HTTPS	SSL negotiation	HTTPS	Back-end authentication	• Supports the X-Forwarded-For header • Requires SSL certificates deployed on the load balancer and the registered instances

Source: aws.amazon.com

Classical Load Balancer (CLB)

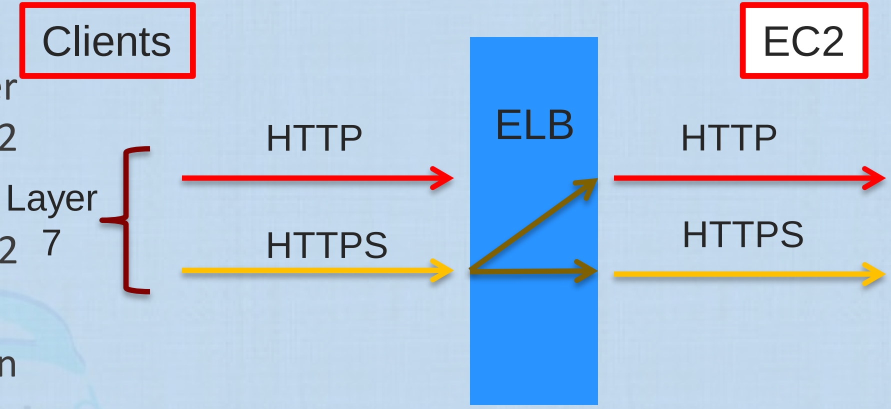
ELB Sticky Sessions

Review Topic : Elastic Load Balancer

CLB – HTTP/HTTPS Session Stickiness

Session Stickiness (Session Affinity)

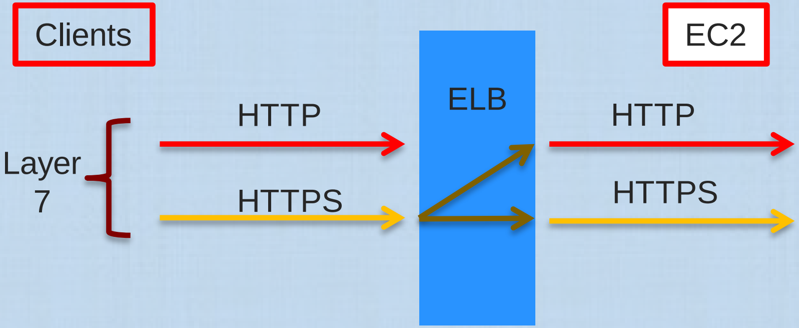
- Whereby the CLB binds a client/user session/requests to a specific backend EC2 instance
- It is not fault tolerant (in case the backend EC2 instance fails)
- It requires SSL termination (SSL Off-loading) on the CLB,
  - This in turn, requires an X.509 (SSL Server) certificate configured on the CLB



Review Topic : Elastic Load Balancer

ELB – HTTP/HTTPS Listeners

- Session Stickiness (Session Affinity)
  - You can upload the X.509 certificate if you have one using IAM, to be loaded on the ELB
  - The X.509 certificate MUST be in the same AWS Region as the ELB
- The duration of the session stickiness is either ELB duration based, or Application based.
- In either case the CLB requires a cookie inserted in the response of the first request, in order to be able to bind future requests from the same client to the same backend EC2 instance.



Review Topic : Elastic Load Balancer

Session Stickiness Duration

- **Application-Controlled Session Stickiness**
  - The load balancer uses a special cookie to associate the session with the instance that handled the initial request, but the duration of stickiness follows the lifetime of the application cookie specified in the policy configuration.

- **CLB duration-based stickiness:**
  - If the application does not have or provide its own cookies, then the ELB can be configured to create one and determine the stickiness duration
  - The ELB inserts a cookie in the response to bind subsequent requests from the user to the same backend instance
  - The cookie helps the ELB identify which user/session should be sticky to which backend instance

Review Topic : Elastic Load Balancer

ELB – Security Policy Components

Security Policy Components:

- SSL protocols
  - SSL or TLS, are cryptographic protocols
- SSL Ciphers (a set of ciphers is called a cipher suite)
  - Encryption algorithms
  - SSL can use different ciphers to encrypt data
- Server Order Preference
  - Enabled by default, the first match in the ELB cipher list with the Client list is used
- CLB does not support client-side certificates with HTTPS (Client side certificates are used to confirm the Identity of the client – or a two way authentication)

Review Topic : Elastic Load Balancer

SNI and CLB

- Classic Load Balancer **does not support Server Name Indication (SNI),**

- The ELB supports a single X.509 certificate

- For **multiple SSL certificates (**Like the case of multiple websites each with its own certificate):
  - Create multiple ELB instances (Expensive, Costly, Not scalable, Admin hassle), OR
  - Use TCP Passthrough configuration on the CLB where:
    - Front and back ends are configured with TCP listeners
    - The load balancer passes the request through, with the SNI certificate as is.
    - You then handle the HTTPS termination (SNI certificates) from the EC2 instance itself.
      - You loose X-Forwarded-For option once you go away from HTTP/HTTPS

ELB – Connection Draining

- When identifying unhealthy instances, the CLB will wait for a period of 300 seconds (by default), for the in-flight sessions to this EC2 back end instance to complete
  - If the in-flight sessions are not completed before the maximum time (300 seconds - configurable between 1 – 3600 seconds), the ELB will force termination of these sessions
  - During the connection draining, the Back end instance state will be “InService : Instance Deregistration Currently In Progress”
  - AWS Auto-Scaling would also honor the connection draining setting for unhealthy instances
  - During the connection draining period, ELB will not send new requests to the unhealthy Instance

AWS Classic Load Balancer

ELB Monitoring, Scaling & testing

Review Topic : Elastic Load Balancer

ELB – Monitoring

ELB monitoring can be achieved by:

- **AWS Cloud Watch:**
  - AWS ELB service sends ELB metrics to cloud watch every **"One minute"**
  - ELB service sends these metrics only if there are requests flowing through the ELB
  - AWS Cloud Watch can be used to trigger an SNS notification if a threshold you define is reached

- **Access Logs:**
  - Disabled by default
  - You can obtain request information such as requester, time of request, requester IP, request type…etc
  - Optional (disabled by default), you can choose to store the access logs in an S3 bucket that you specify

Review Topic : Elastic Load Balancer

ELB – Monitoring

- **Access Logs (Cont.):**
  - You are not charged extra for enabling access logs
    - You pay for S3 storage
  - You are not charged for data transfer of access logs from ELB to the S3 bucket
- **AWS Cloud Trail:**
  - You can use it to capture all API calls for your ELB
  - You can store these logs in an S3 bucket that you specify

Review Topic : Elastic Load Balancer

ELB Scaling & Load Testing your Applications

For efficient load testing of your ELB or applications hosted on backend instances

- Use multiple testing instances of client testing & try to launch the tests at the same time
  - You can also use global testing sites if possible
- **If using a single client for testing,** ensure your testing tool will enforce the **Re-resolving of DNS** with each testing/request initiated for testing
  - This will ensure that as ELB service launches new ELB nodes, the new nodes will be leveraged through DNS re-resolution

An alternative approach would be to implement a DNS round robin (e.g., with Amazon Route 53). In this case, DNS responses return an IP address from a list of valid hosts in a round robin fashion. While easy to implement, this approach does not always work well with the elasticity of cloud computing. This is because even if you can set low time to live (TTL) values for your DNS records, caching DNS resolvers are outside the control of Amazon Route 53 and might not always respect your settings.

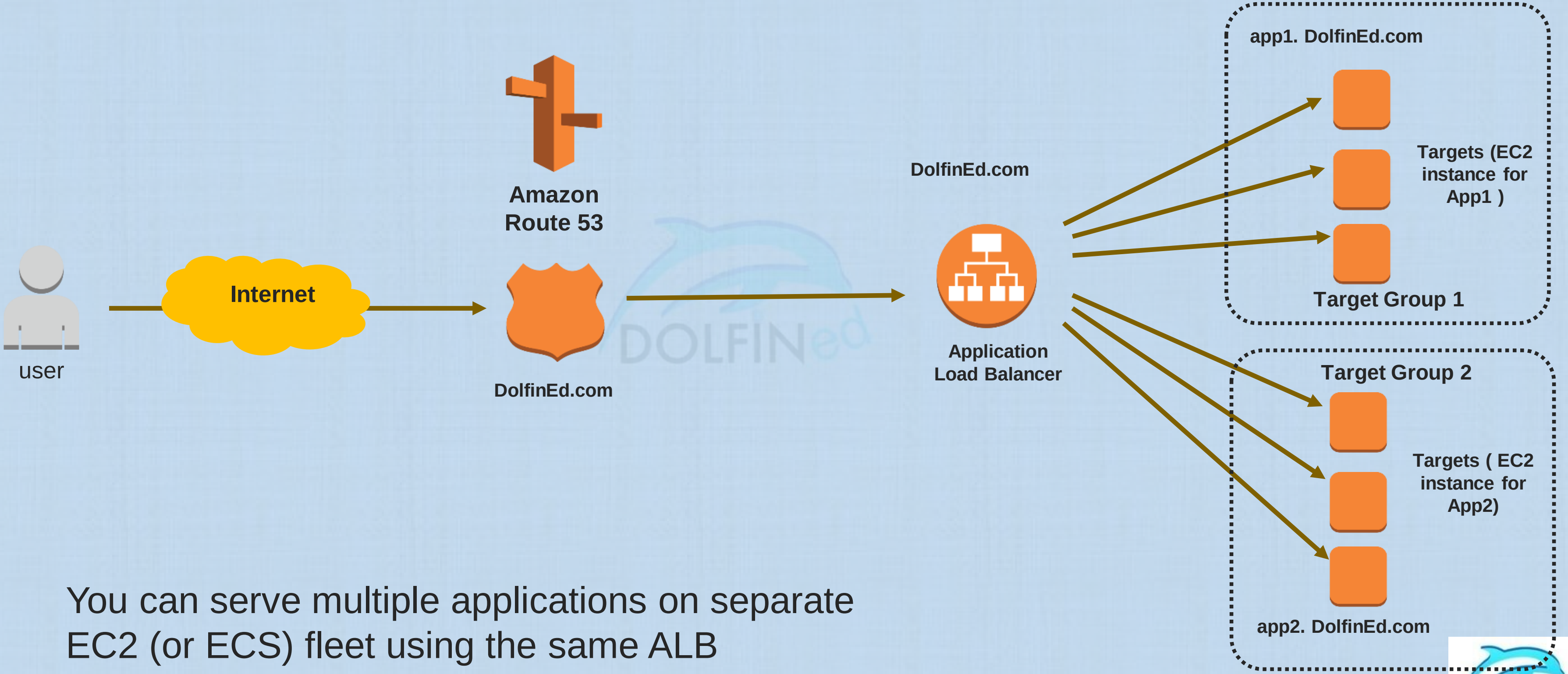
APPLICATION LOAD BALANCER (ALB)

AWS ALB

AWS Application Load Balancer (ALB) Introduction

AWS Application Load Balancer

AWS – Application Load Balancer (ALB) – One ALB, Multiple Applications

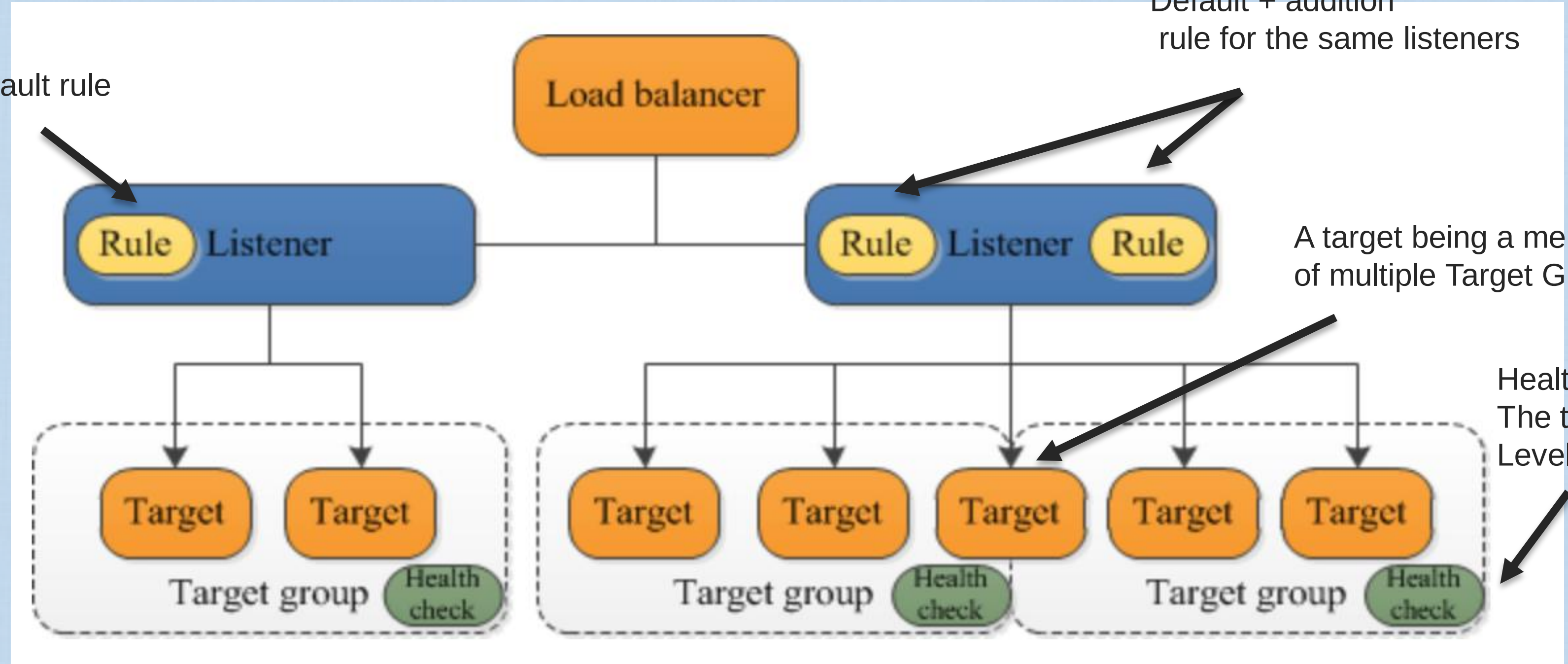


You can serve multiple applications on separate EC2 (or ECS) fleet using the same ALB

AWS Application Load Balancer

AWS ALB

- An Application Load Balancer functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model.
  - It supports HTTP, HTTPS, HTTP/2, and WebSockets



Default rule

Default + addition rule for the same listeners

A target being a member of multiple Target Groups

Health check at The target group Level

AWS Application Load Balancer

AWS ALB Components – Target Groups

Target Groups:

- Are regional constructs (confined to a region)
- A Target Group is a logical grouping of Targets
- Each target group can be associated **with only one load balancer**.
- AS Groups can scale each target group individually
- The **target group** is used to route requests to registered **targets** as part of an action for a rule.
- Health checks can be configured per target group.
- An ALB can route to multiple target groups.
- You define **one Protocol and one port per target group** which will be used to route/forward traffic to the registered targets.
- They can exist independently from the ALB

AWS Application Load Balancer

AWS ALB - Targets

Targets:

- **Targets** specify the endpoints and are registered with the ALB as part of a target group.
- Targets can be EC2 instances, a Microservice, and Application on an ECS Container, or IP addresses
  - You can't specify public internet-routable IP addresses as targets.
- You can register a target with multiple target groups.
- You can add and remove targets from your load balancer as your needs change, without disrupting the overall flow of requests to your application.
- You can use IP addresses as targets to register:
  - Instances in a peered VPC,
  - AWS resources that are addressable by IP address and port (for example, databases),
  - On-premises resources linked to AWS through AWS Direct Connect or a VPN connection.
- You can register each EC2 instance or IP address with the same target group multiple times using different ports, which enables the load balancer to route requests to microservices.

AWS Application Load Balancer

AWS ALB Components – Target Groups & Targets

- You CAN NOT mix targets of different types in one target group, i.e you can not mix EC2 with ECS and/or IP targets in one target group
  - You need to keep the endpoint type homogenous in each group
- You can configure health checks on a per target group basis.
  - Health checks are performed on all targets registered to a target group that is specified in a listener rule for your load balancer.
- By default, the load balancer sends requests to registered targets using the port and protocol that you specified for the target group.
  - You can override this port when you register each target with the target group.

AWS ALB Components – Rules (or Routing Rules)

Rules provide a link between listeners and target groups and consist of conditions and actions.

- Each rule consists of a priority, action, optional host condition, and optional path condition
- Each rule specifies a (optional) **condition**, **target group**, **action,** and a **priority**.
  - When the condition is met, the traffic is forwarded to the target group.
- You must define a default rule for each listener, and you can add rules that specify different target groups based on the content of the request **(also known as *content-based routing*).**
- If no rules are found, the request will follow the default rule, which forwards the request to the default target group.

Rule Priority

- Each rule has a priority.
- Rules are evaluated in priority order, from the lowest value to the highest value.

Rule Actions

- Each rule action has a type and a target group.
- You can change the target group for a rule at any time.

AWS ALB Components – Rules Conditions

Rule Conditions

- There are two types of rule conditions: host and path. When the conditions for a rule are met, then its 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 in order for it to route requests.

AWS ALB

AWS Application Load Balancer (ALB) – Content-based Routing

AWS Application Load Balancer

AWS ALB – Content-based Routing

- **Content-Based Routing**
  Application Load Balancer can route a request to a service based on the content of the request.
- Two types of content routing are supported on the ALB, they are host-based and path-based

- **Host-based (Domain name based) Routing**
  You can create ALB rules to route a client request based on the domain name Host field of the HTTP header allowing you to route to multiple domains from the same load balancer.
  - HTTP request – Host field:
    - Requests to blog.dolfinEd.com can be sent to a target group, while requests to content.dolfined.com are sent to to another.

AWS ALB – Path based Routing

Using Path-based Routing

You can route a client request based on the URL path of the HTTP header.

- It routes incoming HTTP and HTTPS traffic based on the path element of the URL in the request.
- This path-based routing allows you to route requests to, for example, **/images** to one target group ,and **/videos** to another target group.
- Segmenting your traffic in this way gives you the ability to control the processing environment for each category of requests.
  - Perhaps **/images** requests are best processed on a specific type of EC2 instances, while **/videos** requests are best handled by Graphics Optimized instances.
- You can also create rules that combine host-based routing and path-based routing.
  - This would allow you to route requests to **images.example.com/thumbnails** and **images.example.com/production** to distinct target groups.

AWS Application Load Balancer

AWS ALB - Features

- WebSockets protocol support:
  - Application Load Balancers provide native support for Websockets.
  - You can use WebSockets with both HTTP and HTTPS listeners.
  - Websockets allow for full duplex communication
  - Websockets protocol support is enabled by default
  - **CLB does not support it**
- **HTTP/2 Support:**
  - HTTP/2 allow multiple requests at the same time
  - HTTP/2 is supported by default
- Cross zone load balancing is enabled by default
- Supports enhanced health checks and enhanced CloudWatch metrics
- ALB provides health check improvements that allow detailed error codes from 200-399
- ALB provides additional information in Access Logs compared to CLB

AWS Application Load Balancer

AWS ALB

- **Web Application Firewall (WAF) Support**
  - You can use AWS WAF with your Application Load Balancer to allow or block requests based on the rules in a web access control list (web ACL).
- **Request Tracking**
  - The Application Load Balancer injects a new custom identifier “X-Amzn-Trace-Id” HTTP header on all requests coming into the load balancer.
- Internet facing ALB supports IPv4 and DualStack
  - However, the ALB will communicate with the Targets using IPv4
- Internal ALB uses IPv4 only (no dual stack support yet)
- **ALB does not support backend server authentication**
  - Back-end Server Authentication enables authentication of the instances.

AWS Application Load Balancer

AWS ALB – SNI, Connection Draining, and Sticky Sessions

- ALB supports Server Name Indication (SNI) certificates.
- You can serve multiple TLS secured applications (multiple domains) by the ALB, each with its own certificate.
- Integrates with AWS ACM
- ALB will choose the right certificates depending on the client request

Deregistration Delay (Connection Draining)

- Elastic Load Balancing stops sending requests to targets that are deregistering.
- By default, Elastic Load Balancing waits 300 seconds before completing the deregistration process, which can help in-flight requests to the target to complete.

- To use **sticky sessions,** the clients must support cookies.
- Application Load Balancers support load balancer-generated cookies only.
  - The name of the cookie is AWSALB.
  - You enable sticky sessions at the target group level.

AWS ALB

AWS Application Load Balancer (ALB) - Monitoring

AWS Application Load Balancer

AWS ALB - Monitoring

The following features can be used to monitor your load balancers, analyze traffic patterns, and troubleshoot issues with your load balancers and targets.

- **CloudWatch metrics**
  - Published every 1 minute if there are requests flowing through the ALB

- **Access logs**
  - You can use access logs to capture detailed information about the requests made to your load balancer and store them as log files in Amazon S3. You can use these access logs to analyze traffic patterns and to troubleshoot issues with your targets.

- **CloudTrail logs**
  - You can use AWS CloudTrail to capture detailed information about the calls made to the Elastic Load Balancing API and store them as log files in Amazon S3.

NETWORK LOAD BALANCER (NLB)

AWS NLB

AWS Network Load Balancer (NLB) – Features and How it Works

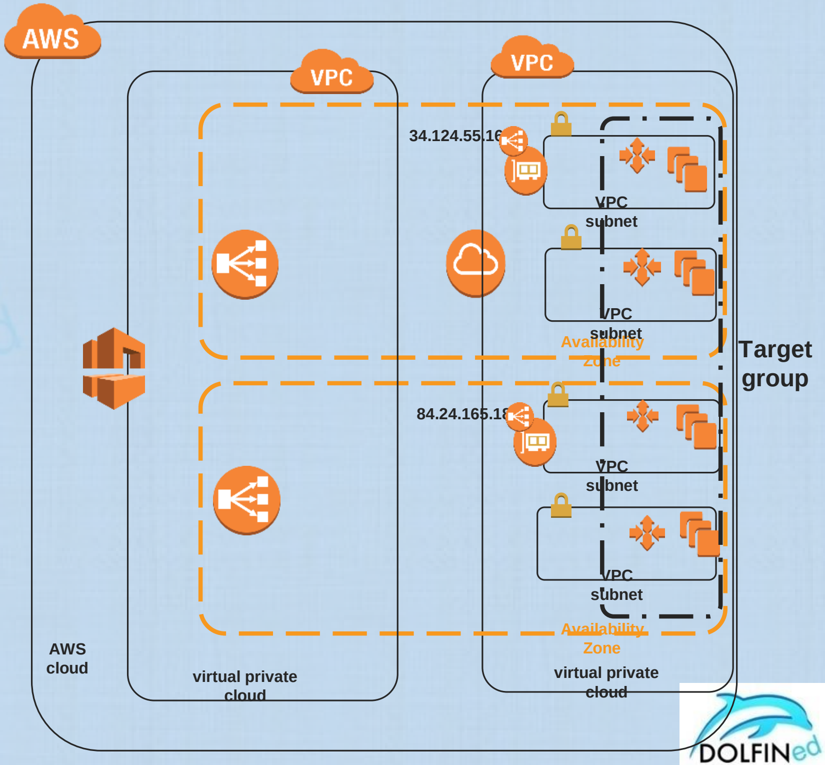
NLB Features and how it works

- A Network Load Balancer operates at the Transport layer (Layer 4) only of the OSI model.
- Supports TCP and TLS listeners for the client requests
- NLB has a higher connection rates per second compared to other ELBs, it can handle millions of requests per second.
  - Provider much lower latencies compared to other ELBs
- **Remember, NLB does not support security groups. You can't configure a security group for the NLB**

NLB Static IP (and EIP) support per AZ

Support for **static IP addresses** for the load balancer.

- You can optionally associate one Elastic IP address per NLB enabled subnet, as a static IPv4 address for the NLB node in that subnet.
- Any connections/requests sent to the NLB's IP address will spread traffic across the instances in all the VPC subnets in the AZ.
- You can also specify an existing Elastic IP for each AZ for even greater control.
- can be used in situations where:
  - IP addresses need to be hard-coded into DNS records,
  - Customer firewall rules (whitelisting) or similar needs.



NLB Features and how it works

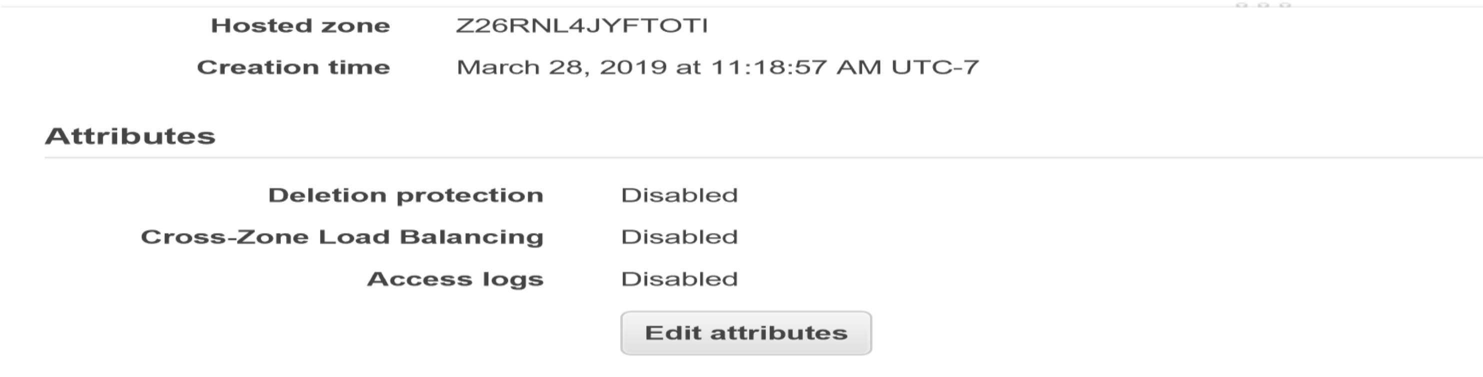
- Unlike other ELB types, NLB supports TCP and UDP
- Support for routing requests to multiple applications on a single EC2 instance.
  - You can register each instance or IP address with the same target group using multiple ports.
- NLB Supports load balancing to ECS containers.
- NLB supports monitoring the health of each service independently,
  - Health checks are defined at the target group level
- NLB can be used with Auto Scaling to achieve dynamic scaling of targets/services.
  - You can do this if you are registering targets by instance ID, not by IP address.
- Network Load Balancers support connections from clients over:
  - VPC peering,
  - AWS VPNs, and
  - Third-party VPN solutions.

AWS NLB

Multi-AZ, Access Logs, Delete Protection, Cross Zone Load balancing and TLS Listeners, Target Types

NLB in Multi-AZ

- You enable one or more Availability Zones for your load balancer when you create it.
- You cannot enable or disable Availability Zones for a Network Load Balancer after you create it.
- Access logs, Delete protection, and Cross Zone load balancing are disabled by default on NLB



Hosted zone	Z26RNL4JYFTOTI
Creation time	March 28, 2019 at 11:18:57 AM UTC-7

Attributes

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

TLS Listeners

- If the listener protocol is TLS, you must deploy exactly one SSL server certificate on the listener.
  - The certificate can be from ACM, uploaded to ACM, or IAM

- You can use WebSockets with your listeners.

- When you create a listener, you specify a rule for routing requests. This rule forwards requests to the specified target group.

Target Types

- NLB supports the either of the following target types (but not a mix in one target group):
  - IP Address: The Targets by IP address can be outside of the VPC.
  - Instance ID (Instances in a peered VPC must be referenced by IP not Instance ID)

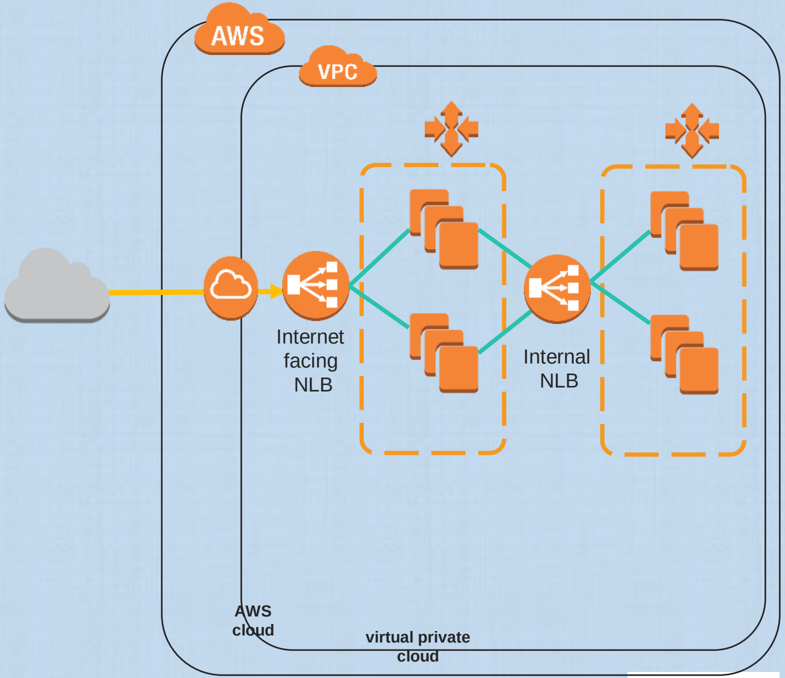
- If the target type is IP, it can be from one of the following CIDR ranges:
  - The subnets of the VPC for the target group
  - RFS1918 ranges: 10.0.0.0/8 , 172.16.0.0/12 , 192.168.0.0/16
    - 100.64.0.0/10 (RFC 6598)

- As in ALB, **target can not** be a publicly routable IP addresses.

- NLBs do not support the lambda target type,

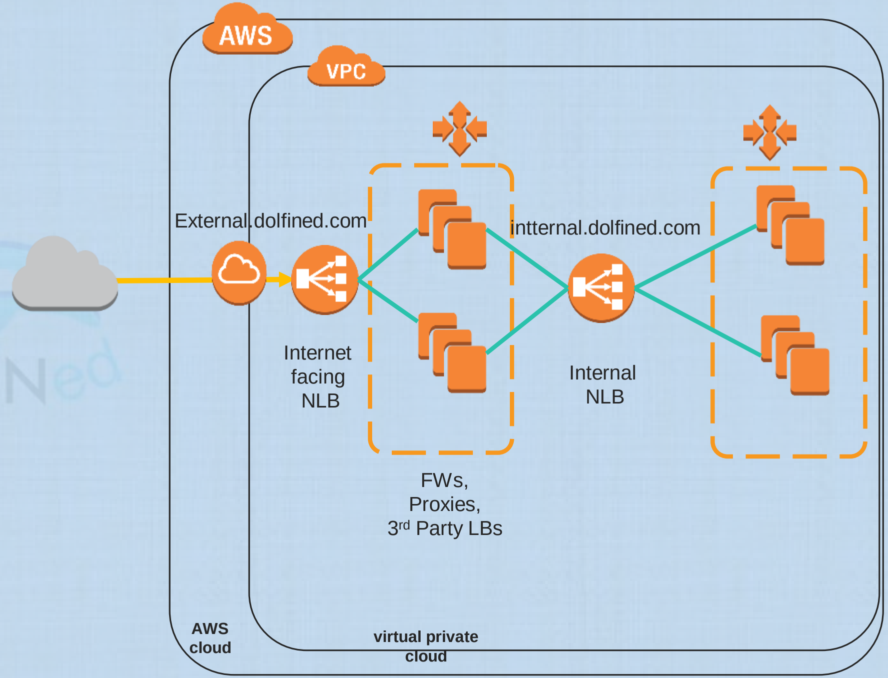
Client Source IP address Preservation

- If you use Instance ID as target type, NLB preserves the clients source IP addresses, and provides them to the targets.
- On the other hand, If you use the IP address as the target, the source IP addresses are the private IP addresses of the load balancer nodes.
  - ○ In this case if your applications require the Clients' source IP address, you can configure Proxy Protocol on the NLB.



Client Source IP address Preservation

- Two load balancers with a services layer sandwiched between them
- The services layer can have NGFWs, Proxies, or 3rd Party load balancers, or even a web tier
- This way you can scale this layer while using fewer Elastic IP addresses (or static IPs)
- Preserving source IP addresses helps Geo-Location services if this is a FW layer (or 3rd Party LBs),and Whitelisting



Proxy Protocol and Health Checks

Proxy Protocol

- NLB supports Proxy Protocol v2
  - Is configurable at the Target group level. It is disabled by default
- This comes in handy in case targets are referenced by IP address (Target types) and the clients' source IP address is required for the applications.
- When enabled, the NLB prepends a proxy protocol header to the TCP data.
  - The NLB will not discard or overwrite any existing data
- If the traffic/requests are coming to the NLB from ELB service consumers through a VPC endpoint service,
  - The source IP addresses provided to your applications is the NLB nodes' private IP addresses.
  - Use Proxy protocol if your applications needs the IP addresses of the AWS service consumers

Health checks

- Network Load Balancers use **active and passive (network) health checks**
  - With **active health checks,** the load balancer periodically sends a request to each registered target to check its status.
  - With **passive (Network) health checks,** the load balancer observes how targets respond to connections.

AWS NLB

Monitoring the NLB

CloudWatch Metrics , VPC Flow Logs, and CloudTrail

- ELB service publishes data points to Amazon CloudWatch for load balancers and targets.
  - And create alarms to monitor specific metrics
- ELB reports metrics to CloudWatch every one minute, but only when requests are flowing thr
- VPC Flow Logs can be used to capture detailed information about the traffic going to and from your NLB.
  - Create a flow log for each network interface for your load balancer.
  - There is one network interface per load balancer subnet.
- AWS CloudTrail can be used to capture detailed information about the calls made to the Elastic Load Balancing API and store them as log files in Amazon S3.
  - These CloudTrail logs can be used to determine which calls were made, the source IP address where the call came from, who made the call, when the call was made, and so on.

Monitoring the NLB – Access Logs

- Access logs can be used to capture detailed information about TLS requests made to the NLB.
  - The log files are stored in Amazon S3 buckets.
  - Access logs can be used to analyze traffic patterns and to troubleshoot issues with your targets.

- Access logs are created only if the load balancer has a TLS listener and they contain information only about TLS requests.

- ELB publishes a log file for each load balancer node every 5 minutes.
  - Log delivery is eventually consistent.

AWS NLB

Perfect Forward Secrecy (PFS)

Review Topic : Elastic Load Balancing

Elastic Load Balancing and PFS

- Amazon ELB now offer advanced cipher suites that use the Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) protocol.
- ECDHE allows SSL/TLS clients to provide Perfect Forward Secrecy, which uses session keys that are ephemeral and not stored anywhere.
  - This helps prevent the decoding of captured data by unauthorized third parties, even if the secret long-term key itself is compromised.