**ELB**

Q: Is HTTP/2 Supported on an Application Load Balancer?

A: Yes. HTTP/2 support is enabled natively on an Application Load Balancer. Clients that support HTTP/2 can connect to an Application Load Balancer over TLS.

Q: Is Request tracing supported on an Application Load Balancer?

A: Yes. Request tracing is enabled by default on your Application Load Balancer.

Q: Can I use the existing APIs that I use with my Classic Load Balancer with an Application Load Balancer?

A: No. Application Load Balancers require a new set of APIs.

Q: Can I convert my Classic Load Balancer to an Application Load Balancer (and vice versa)?

A: No, you cannot convert one load balancer type into another.

Q: Can I migrate to Application Load Balancer from Classic Load Balancer?

A: Yes. You can migrate to Application Load Balancer from Classic Load Balancer using one of the options listed in this [document](https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/migrate-to-application-load-balancer.html).

Q: Can I use a single Application Load Balancer for handling HTTP and HTTPS requests?

A: Yes, you can add listeners for HTTP port 80 and HTTPS port 443 to a single Application Load Balancer.

Q: Can I get a history of Application Load Balancing API calls made on my account for security analysis and operational troubleshooting purposes?

A: Yes. To receive a history of Application Load Balancing API calls made on your account, use [AWS CloudTrail](https://aws.amazon.com/cloudtrail/).

Q: Is back-end server authentication supported with an Application Load Balancer?

A: No, only encryption is supported to the back-ends with an Application Load Balancer.

Q: How can I enable Server Name Indication (SNI) for my Application Load Balancer?

A: SNI is automatically enabled when you associate more than one TLS certificate with the same secure listener on a load balancer. Similarly, SNI mode for a secure listener is automatically disabled when you have only one certificate associated to a secure listener.

Q: How do you set up rules on an Application Load Balancer?

A: You can configure rules for each of the listeners that you have on the load balancer. The rules include conditions and corresponding actions if the conditions are satisfied. The supported conditions are Host header, path, HTTP headers, methods, query parameters, and source IP CIDRs. The supported actions are redirect, fixed response, authenticate, and forward. Once you have set this up, the load balancer will use the rules to determine how a particular HTTP request should be routed. You can use multiple conditions and actions in a rule and in each condition can specify a match on multiple values.

Q. How can I protect my web applications behind a load balancer from web attacks?

A: You can integrate your Application Load Balancer with AWS WAF, a web application firewall that helps protect web applications from attacks by allowing you to configure rules based on IP addresses, HTTP headers, and custom URI strings. Using these rules, AWS WAF can block, allow, or monitor (count) web requests for your web application

Q: How can I load balance applications distributed across a VPC and on-premises location?

A: There are various ways to achieve hybrid load balancing. If an application runs on targets distributed between a VPC and an on-premises location, you can add them to the same target group using their IP addresses. To migrate to AWS without impacting your application, gradually add VPC targets to the target group and remove on-premises targets from the target group. If you have two different applications such that the targets for one application are in a VPC and the targets for other applications are in on-premises location, you can put the VPC targets in one target group and the on-premises targets in another target group and use content based routing to route traffic to each target group. You can also use separate load balancers for VPC and on-premises targets and use DNS weighting to achieve weighted load balancing between VPC and on-premises targets.

Q: When should I authenticate users using the Application Load Balancer’s integration with Amazon Cognito vs. the Application Load Balancers’ native support for OpenID Connect (IODC) identity providers (IdPs)?

A: You should use authentication through Amazon Cognito if:

* You want to provide flexibility to your users to authenticate via social network identities (Google, Facebook, and Amazon) or enterprise identities (SAML) or via your own user directories provided by Amazon Cognito’s User Pool.
* You are managing multiple identity providers including OpenID Connect and want to create a single authentication rule in Application Load Balancer (ALB), that can use Amazon Cognito to federate your multiple identity providers.
* You have a need to actively manage user profiles with one or more social or OpenID Connect identity providers from one central place. For example, you can put users in groups and add custom attributes to represent user status and control access for paid users.

Alternatively, if you have invested in developing custom IdP solutions and simply want to authenticate with a single identity provider that is OpenID Connect-compatible, you may prefer using Application Load Balancer’s native OIDC solution

Q: What type of redirects does Application Load Balancer support ?

A: The following three types of redirects are supported.

|  |  |
| --- | --- |
| **Types of redirects** | **Examples** |
| HTTP to HTTP | http://hostA to http://hostB |
| HTTP to HTTPS | http://hostA to https://hostB https://hostA:portA/pathA to https://hostB:portB/pathB |
| HTTPS to HTTPS | https://hostA to https://hostB |

Q: What content types does ALB support for the message body of fixed-response action?

A: The following content types are supported: text/plain, text/css, text/html, application/javascript, application/json.

Q: How does Lambda invocation via Application Load Balancer work?

A: HTTP(S) requests received by a load balancer are processed by the content-based routing rules. If the request content matches the rule with an action to forward it to a target group with a Lambda function as a target then the corresponding Lambda function is invoked. The content of the request (including headers and body) is passed on to the Lambda function in JSON format. The response from the Lambda function should be in JSON format. The response from the Lambda function is transformed into an HTTP response and sent to the client. The load balancer invokes your Lambda function using the AWS Lambda Invoke API and requires that you have provided invoke permissions for your Lambda function to Elastic Load Balancing service.

Q: Does Lambda invocation via Application Load Balancer support requests over both HTTP and HTTPS protocol?

A: Yes. Application Load Balancer supports Lambda invocation for requests over both HTTP and HTTPS protoco

Q: What are the key features available with the Network Load Balancer?

A: Network Load Balancer provides both TCP and UDP (Layer 4) load balancing. It is architected to handle millions of requests/sec, sudden volatile traffic patterns and provides extremely low latencies. In addition Network Load Balancer also supports TLS termination, preserves the source IP of the clients, and provides stable IP support and Zonal isolation. It also supports long-running connections that are very useful for WebSocket type applications.

Q: Can Network Load Balancer process both TCP and UDP protocol traffic on the same port?

A: Yes. To achieve this, you can use a TCP+UDP listener. For example, for a DNS services using both TCP and UDP you can create a TCP+UDP listener on port 53, and the load balancer will process traffic for both UDP and TCP requests on that port. You must associate a TCP+UDP listener with a TCP+UDP target group.

Q: Can I create my Network Load Balancer in a single Availability Zone?

A: Yes, you can create your Network Load Balancer in a single availability zone by providing a single subnet when you create the load balancer.

Q: Does Network Load Balancer support DNS regional and zonal fail-over?

A: Yes, you can use Amazon Route 53 health checking and DNS failover features to enhance the availability of the applications running behind Network Load Balancers. Using Route 53 DNS failover, you can run applications in multiple AWS Availability zones and designate alternate load balancers for failover across regions. In the event that you have your Network Load Balancer configured for multi-AZ, if there are no healthy EC2 instances registered with the load balancer for that Availability Zone or if the load balancer nodes in a given zone are unhealthy, then R-53 will fail away to alternate load balancer nodes in other healthy availability zones.

Q: Can I assign more than one EIP to my Network Load Balancer in each subnet?

A: No. For each associated subnet that a Network Load Balancer is in, the Network Load Balancer can only support a single public/internet facing IP address.

Q: If I remove/delete a Network Load Balancer what will happen to the Elastic IP addresses that were associated with it?

A: The Elastic IP Addresses that were associated with your load balancer will be returned to your allocated pool and made available for future use.

Q: Can the internal Network Load balancer support more than one private IP in each subnet?

A: No. For each associated subnet that a load balancer is in, the Network Load Balancer can only support a single private IP.

Q: Can I set up Websockets with my Network Load Balancer?

A: Yes, configure TCP listeners that route the traffic to the targets that implement WebSockets protocol (https://tools.ietf.org/html/rfc6455 ). Because WebSockets is a layer 7 protocol and Network Load Balancer is operating at layer 4, no special handling exists in Network Load Balancer for WebSockets or other higher level protocols.

Q: What benefit will I get by targeting containers behind a load balancer with IP addresses instead of instance IDs?

A: Each container on an instance can now have its own security group and does not need to share security rules with other containers. You can attach security groups to an ENI and each ENI on an instance can have a different security group. You can map a container to the IP address of a particular ENI to associate security group(s) per container. Load balancing using IP addresses also allows multiple containers running on an instance use the same port (say port 80). The ability to use the same port across containers allows containers on an instance to communicate with each other through well-known ports instead of random ports.

Q: Is there any impact of cross-zone load balancing on Network Load Balancer limits?

A: Yes. Network Load Balancer currently supports 200 targets per Availability Zone. For example, if you are in 2 Availability-Zones, you can have up to 400 targets registered with Network Load Balancer. If cross-zone load balancing is on, then the maximum targets reduces from 200 per Availability Zone to 200 per load balancer. So, in the example above when cross-zone load balancing is on, even though your load balancer is in 2 Availability Zones, you are limited to 200 targets that can be registered to the load balancer.

Q: Is source IP is preserved when terminating TLS on Network Load Balancer?

A: Source IP continues to be preserved even if you terminate TLS on the Network Load Balancer.

 If you use an HTTPS listener with SSL pass-through, then the EC2 instances would continue to be under heavy CPU load as they would still need to decrypt the secure traffic at the instance level.

* To use sticky sessions, the clients must support cookies. When a load balancer first receives a request from a client, it routes the request to a target, generates a cookie named AWSALB that encodes information about the selected target, encrypts the cookie, and includes the cookie in the response to the client.
* Long-lived TCP connections between clients and instances can potentially lead to unequal distribution of traffic by the load balancer. Long-lived TCP connections between clients and instances cause uneven traffic load distribution by design. As a result, new instances take longer to reach connection equilibrium.
* Enable session stickiness using elastic load balancers" is incorrect as this feature directs sessions from a specific client to a specific EC2 instances. Therefore, if the instance fails the user must be redirected to another EC2 instance and the session state data would be lost.
* Only 1 subnet per AZ can be enabled for each ELB.
* Route 53 can be used for region load balancing with ELB instances configured in each region.
* Need one public subnet in each AZ where the ELB is defined. >> Internet facing ELB:
* INTERNET ELB : ELB DNS name format: <name>-<id-number>.<region>.elb.amazonaws.com.
* INTERNAL ELB : ELB DNS name format: **internal**-<name>-<id-number>.<region>.elb.amazonaws.com.

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 in order for the ELB to scale.

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

* Listeners for CLB provide options for TCP and HTTP/HTTPS.
* Listeners for ALB only provide options for HTTP and HTTPS.
* Listeners for NLB only provide TCP as an option

For ALB at least 2 subnets must be specified.

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

For CLB you don’t need to specify any subnets unless you have “Enable advanced VPC configuration” enabled in which case you must specify two.

Perfect Forward Secrecy (PFS) provides additional safeguards against the eavesdropping of encrypted data, through the use of 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).

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

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

Q: Can I privately access Elastic Load Balancing APIs from my Amazon Virtual Private Cloud (VPC) without using public IPs?

A: Yes, you can privately access Elastic Load Balancing APIs from your Amazon Virtual Private Cloud (VPC) by creating [VPC endpoints](http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-endpoints.html). With VPC endpoints, the routing between the VPC and Elastic Load Balancing APIs is handled by the AWS network without the need for an Internet gateway, NAT gateway, or VPN connection. The latest generation of VPC Endpoints used by Elastic Load Balancing are powered by AWS PrivateLink, an AWS technology enabling the private connectivity between AWS services using Elastic Network Interfaces (ENI) with private IPs in your VPCs.

Q: Is HTTP/2 Supported on an Application Load Balancer?

A: Yes. HTTP/2 support is enabled natively on an Application Load Balancer. Clients that support HTTP/2 can connect to an Application Load Balancer over TLS.

Q: What TCP ports can I use to Application load balance?

A: You can perform load balancing for the following TCP ports: 1-65535

Q: Can I use the existing APIs that I use with my Classic Load Balancer with an Application Load Balancer?

A: No. Application Load Balancers require a new set of APIs.

Q: Can I convert my Classic Load Balancer to an Application Load Balancer (and vice versa)?

A: No, you cannot convert one load balancer type into another.

Q: Can I migrate to Application Load Balancer from Classic Load Balancer?

A: Yes. You can migrate to Application Load Balancer from Classic Load Balancer using one of the options listed in this [document](https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/migrate-to-application-load-balancer.html).

Q: Can I use a single Application Load Balancer for handling HTTP and HTTPS requests?

A: Yes, you can add listeners for HTTP port 80 and HTTPS port 443 to a single Application Load Balancer.

Q: Can I get a history of Application Load Balancing API calls made on my account for security analysis and operational troubleshooting purposes?

A: Yes. To receive a history of Application Load Balancing API calls made on your account, use [AWS CloudTrail](https://aws.amazon.com/cloudtrail/).

Q: Is back-end server authentication supported with an Application Load Balancer?

A: No, only encryption is supported to the back-ends with an Application Load Balancer.

Q: How can I enable Server Name Indication (SNI) for my Application Load Balancer?

A: SNI is automatically enabled when you associate more than one TLS certificate with the same secure listener on a load balancer. Similarly, SNI mode for a secure listener is automatically disabled when you have only one certificate associated to a secure listener.

Q: Can I associate multiple certificates for the same domain to a secure listener?

A: Yes, you can associate multiple certificates for the same domain to a secure listener. For example, you can associate:

ECDSA and RSA certificates

Certificates with different key sizes (e.g. 2K and 4K) for SSL/TLS certificates

Single-Domain, Multi-Domain (SAN) and Wildcard certificates

Q: Is IPv6 supported with an Application Load Balancer?

A: Yes, IPv6 is supported with an Application Load Balancer.

Q. How can I protect my web applications behind a load balancer from web attacks?

A: You can integrate your Application Load Balancer with AWS WAF, a web application firewall that helps protect web applications from attacks by allowing you to configure rules based on IP addresses, HTTP headers, and custom URI strings. Using these rules, AWS WAF can block, allow, or monitor (count) web requests for your web application. Please see AWS [WAF developer guide](http://docs.aws.amazon.com/console/waf) for more information.

Q: Can I load balance to any arbitrary IP address?

A: You can use any IP address from the load balancer’s VPC CIDR for targets within load balancer’s VPC and any IP address from RFC 1918 ranges (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16) or RFC 6598 range (100.64.0.0/10) for targets located outside the load balancer’s VPC (for example, targets in Peered VPC, EC2-Classic and on-premises locations reachable over AWS Direct Connect or VPN connection).

Q: How can I load balance applications distributed across a VPC and on-premises location?

A: There are various ways to achieve hybrid load balancing. If an application runs on targets distributed between a VPC and an on-premises location, you can add them to the same target group using their IP addresses. To migrate to AWS without impacting your application, gradually add VPC targets to the target group and remove on-premises targets from the target group. If you have two different applications such that the targets for one application are in a VPC and the targets for other applications are in on-premises location, you can put the VPC targets in one target group and the on-premises targets in another target group and use content based routing to route traffic to each target group. You can also use separate load balancers for VPC and on-premises targets and use DNS weighting to achieve weighted load balancing between VPC and on-premises targets.

Q: How can I load balance to EC2-Classic instances?

A: You cannot load balance to EC2-Classic Instances when registering their Instance IDs as targets. However if you link these EC2-Classic instances to the load balancer's VPC using ClassicLink and use the private IPs of these EC2-Classic instances as targets, then you can load balance to the EC2-Classic instances. If you are using EC2 Classic instances today with a Classic Load Balancer, you can easily migrate to an Application Load Balancer.

Q: When should I authenticate users using the Application Load Balancer’s integration with Amazon Cognito vs. the Application Load Balancers’ native support for OpenID Connect (IODC) identity providers (IdPs)?

A: You should use authentication through Amazon Cognito if:

* You want to provide flexibility to your users to authenticate via social network identities (Google, Facebook, and Amazon) or enterprise identities (SAML) or via your own user directories provided by Amazon Cognito’s User Pool.
* You are managing multiple identity providers including OpenID Connect and want to create a single authentication rule in Application Load Balancer (ALB), that can use Amazon Cognito to federate your multiple identity providers.
* You have a need to actively manage user profiles with one or more social or OpenID Connect identity providers from one central place. For example, you can put users in groups and add custom attributes to represent user status and control access for paid users.

Alternatively, if you have invested in developing custom IdP solutions and simply want to authenticate with a single identity provider that is OpenID Connect-compatible, you may prefer using Application Load Balancer’s native OIDC solution.

Q: What type of redirects does Application Load Balancer support ?

|  |  |
| --- | --- |
| **Types of redirects** | **Examples** |
| HTTP to HTTP | http://hostA to http://hostB |
| HTTP to HTTPS | http://hostA to https://hostB https://hostA:portA/pathA to https://hostB:portB/pathB |
| HTTPS to HTTPS | https://hostA to https://hostB |

Q: What content types does ALB support for the message body of fixed-response action?

A: The following content types are supported: text/plain, text/css, text/html, application/javascript, application/json.

Q: How does Lambda invocation via Application Load Balancer work?

A: HTTP(S) requests received by a load balancer are processed by the content-based routing rules. If the request content matches the rule with an action to forward it to a target group with a Lambda function as a target then the corresponding Lambda function is invoked. The content of the request (including headers and body) is passed on to the Lambda function in JSON format. The response from the Lambda function should be in JSON format. The response from the Lambda function is transformed into an HTTP response and sent to the client. The load balancer invokes your Lambda function using the AWS Lambda Invoke API and requires that you have provided invoke permissions for your Lambda function to Elastic Load Balancing service.

Q: Does Lambda invocation via Application Load Balancer support requests over both HTTP and HTTPS protocol?

A: Yes. Application Load Balancer supports Lambda invocation for requests over both HTTP and HTTPS protocol.

Q: Am I charged for regional AWS data-transfer for cross-zone load balancing in Application Load Balancer?

A: No. Since cross-zone load balancing is always on with Application Load Balancer, you are not charged for this type of regional data transfer.

Q: Is user authentication in Application Load Balancer charged separately?

A: No. There is no separate charge for enabling the authentication functionality in Application Load Balancer. When using Amazon Cognito with Application Load Balancer, Amazon Cognito pricing will apply.

Q: Can Network Load Balancer process both TCP and UDP protocol traffic on the same port?

A: Yes. To achieve this, you can use a TCP+UDP listener. For example, for a DNS services using both TCP and UDP you can create a TCP+UDP listener on port 53, and the load balancer will process traffic for both UDP and TCP requests on that port. You must associate a TCP+UDP listener with a TCP+UDP target group

Q: How does Network Load Balancer compare to what I get with the TCP listener on a Classic Load Balancer?

A: Network Load Balancer preserves the source IP of the client which in the Classic Load Balancer is not preserved. Customers can use proxy protocol with Classic Load Balancer to get the source IP. Network Load Balancer automatically provides a static IP per Availability Zone to the load balancer and also enables assigning an Elastic IP to the load balancer per Availability Zone. This is not supported with Classic Load Balancer.

Q: Can I create my Network Load Balancer in a single Availability Zone?

A: Yes, you can create your Network Load Balancer in a single availability zone by providing a single subnet when you create the load balancer.

Q: Does Network Load Balancer support DNS regional and zonal fail-over?

A: Yes, you can use Amazon Route 53 health checking and DNS failover features to enhance the availability of the applications running behind Network Load Balancers. Using Route 53 DNS failover, you can run applications in multiple AWS Availability zones and designate alternate load balancers for failover across regions. In the event that you have your Network Load Balancer configured for multi-AZ, if there are no healthy EC2 instances registered with the load balancer for that Availability Zone or if the load balancer nodes in a given zone are unhealthy, then R-53 will fail away to alternate load balancer nodes in other healthy availability zones.

Q: Can I have a Network Load Balancer with a mix of ELB-provided IPs and Elastic IPs or assigned private IPs?

A: No. A Network Load Balancer’s addresses must be completely controlled by you, or completely controlled by ELB. This is to ensure that when using Elastic IPs with a Network Load Balancer, all addresses known to your clients do not change.

Q: Can I assign more than one EIP to my Network Load Balancer in each subnet?

A: No. For each associated subnet that a Network Load Balancer is in, the Network Load Balancer can only support a single public/internet facing IP address.

Q: If I remove/delete a Network Load Balancer what will happen to the Elastic IP addresses that were associated with it?

A: The Elastic IP Addresses that were associated with your load balancer will be returned to your allocated pool and made available for future use.

Q: Can the internal Network Load balancer support more than one private IP in each subnet?

A: No. For each associated subnet that a load balancer is in, the Network Load Balancer can only support a single private IP.

Q: Can I set up Websockets with my Network Load Balancer?

A: Yes, configure TCP listeners that route the traffic to the targets that implement WebSockets protocol (https://tools.ietf.org/html/rfc6455 ). Because WebSockets is a layer 7 protocol and Network Load Balancer is operating at layer 4, no special handling exists in Network Load Balancer for WebSockets or other higher level protocols.

Q: Can I use Network Load Balancer to setup PrivateLink?

A: Yes, Network Load Balancers with TCP and TLS Listeners can be used to setup PrivateLink. You cannot setup PrivateLink with UDP listeners on Network Load Balancers.

Q: What is a UDP flow?

A: While UDP is connectionless, the load balancer maintains UDP flow state based on 5-tuple hash, making sure that packets sent in the same context are consistently forwarded to the same target. The flow is considered active as long as traffic is flowing and until the idle timeout is reached. Once the timeout threshold is reached, the load balancer will forget the affinity, and incoming UDP packet will be considered as a new flow and load-balanced to a new target.

Q: What is the idle timeout supported by Network Load Balancer?

A: Network Load Balancer idle timeout for TCP connections is 350 seconds. The idle timeout for UDP flows is 120 seconds.

Q: What benefit will I get by targeting containers behind a load balancer with IP addresses instead of instance IDs?

A: Each container on an instance can now have its own security group and does not need to share security rules with other containers. You can attach security groups to an ENI and each ENI on an instance can have a different security group. You can map a container to the IP address of a particular ENI to associate security group(s) per container. Load balancing using IP addresses also allows multiple containers running on an instance use the same port (say port 80). The ability to use the same port across containers allows containers on an instance to communicate with each other through well-known ports instead of random ports.

Q: How can I load balance applications distributed across a VPC and on-premises location?

A: There are various ways to achieve hybrid load balancing. If an application runs on targets distributed between a VPC and an on-premises location, you can add them to the same target group using their IP addresses. To migrate to AWS without impacting your application, gradually add VPC targets to the target group and remove on-premises targets from the target group. You can also use separate load balancers for VPC and on-premises targets and use DNS weighting to achieve weighted load balancing between VPC and on-premises targets.

Q: How can I load balance to EC2-Classic instances?

A: You cannot load balance to EC2-Classic Instances when registering their Instance IDs as targets. However if you link these EC2-Classic instances to the load balancer's VPC using ClassicLink and use the private IPs of these EC2-Classic instances as targets, then you can load balance to the EC2-Classic instances. If you are using EC2 Classic instances today with a Classic Load Balancer, you can easily migrate to a Network Load Balancer.

Q: Is source IP is preserved when terminating TLS on Network Load Balancer?

A: Source IP continues to be preserved even if you terminate TLS on the Network Load Balancer.

Q: Is back-end server authentication supported with Network Load Balancer?

A: No, only encryption is supported to the back-ends with Network Load Balancer.

Q: What are the certificate types supported by Network Load Balancer?

A: Network Load Balancer only supports RSA certificates with 2K key size. We currently do not support RSA certificate key sizes greater than 2K or ECDSA certificates on the Network Load Balancer.

Q: Which protocols does the Classic Load Balancer support?

A: The Classic Load Balancer supports load balancing of applications using HTTP, HTTPS (Secure HTTP), SSL (Secure TCP) and TCP protocols.

Q: Does the Classic Load Balancer support IPv6 traffic?

A: Yes. Each Classic Load Balancer has an associated IPv4, IPv6, and dualstack (both IPv4 and IPv6) DNS name. IPv6 is not supported in VPC. You can use an Application Load Balancer for native IPv6 support in VPC.

Q: Can I configure a security group for the front-end of Classic Load Balancers?

A: If you are using Amazon Virtual Private Cloud, you can configure security groups for the front-end of your Classic Load Balancers.

Q: Can I use a single Classic Load Balancer for handling HTTP and HTTPS requests?

A: Yes, you can map HTTP port 80 and HTTPS port 443 to a single Classic Load Balancer.