aws

Gateway Load Balancers

ELB

ELB: Gateway Load Balancers

Table of Contents

What is a Gateway Load Balancer?

ELB automatically distributes your incoming traffic across multiple targets, in one or more Availability Zones. It monitors the health of its registered targets, and routes traffic only to the healthy targets. ELB scales your load balancer as your incoming traffic changes over time. It can automatically scale to the vast majority of workloads.

ELB supports the following load balancers: Application Load Balancers, Network Load Balancers, Gateway Load Balancers, and Classic Load Balancers. You can select the type of load balancer that best suits your needs. This guide discusses Gateway Load Balancers. For more information about the other load balancers, see the User Guide for Application Load Balancers, the User Guide for Network Load Balancers, and the User Guide for Classic Load Balancers.

Gateway Load Balancer overview

Gateway Load Balancers enable you to deploy, scale, and manage virtual appliances, such as firewalls, intrusion detection and prevention systems, and deep packet inspection systems. It combines a transparent network gateway (that is, a single entry and exit point for all traffic) and distributes traffic while scaling your virtual appliances with the demand.

A Gateway Load Balancer operates at the third layer of the Open Systems Interconnection (OSI) model, the network layer. It listens for all IP packets across all ports and forwards traffic to the target group that's specified in the listener rule. It maintains flow stickiness to a specific target appliance using 5-tuple (default), 3-tuple, or 2-tuple. The Gateway Load Balancer and its registered virtual appliance instances exchange application traffic using the GENEVE protocol on port 6081.

Gateway Load Balancers use Gateway Load Balancer endpoints to securely exchange traffic across VPC boundaries. A Gateway Load Balancer endpoint is a VPC endpoint that provides private connectivity between virtual appliances in the service provider VPC and application servers in the service consumer VPC. You deploy the Gateway Load Balancer in the same VPC as the virtual appliances. You register the virtual appliances with a target group for the Gateway Load Balancer.

Traffic to and from a Gateway Load Balancer endpoint is configured using route tables. Traffic flows from the service consumer VPC over the Gateway Load Balancer endpoint to the Gateway Load Balancer in the service provider VPC, and then returns to the service consumer VPC. You must create the Gateway Load Balancer endpoint and the application servers in different subnets. This enables you to configure the Gateway Load Balancer endpoint as the next hop in the route table for the application subnet.

For more information, see Access virtual appliances through AWS PrivateLink in the *AWS PrivateLink Guide*.

Appliance vendors

You are responsible for choosing and qualifying software from appliance vendors. You must trust the appliance software to inspect or modify traffic from the load balancer. The appliance vendors listed as ELB Partners have integrated and qualified their appliance software with AWS. You can place a higher degree of trust in the appliance software from vendors in this list. However, AWS does not guarantee the security or reliability of software from these vendors.

Getting started

To create a Gateway Load Balancer using the AWS Management Console, see Getting started. To create a Gateway Load Balancer using the AWS Command Line Interface, see Getting started using the CLI.

Pricing

With your load balancer, you pay only for what you use. For more information, see ELB pricing.

Getting started with Gateway Load Balancers

Gateway Load Balancers make it easy to deploy, scale, and manage third-party virtual appliances, such as security appliances.

In this tutorial, we'll implement an inspection system using a Gateway Load Balancer and a Gateway Load Balancer endpoint.

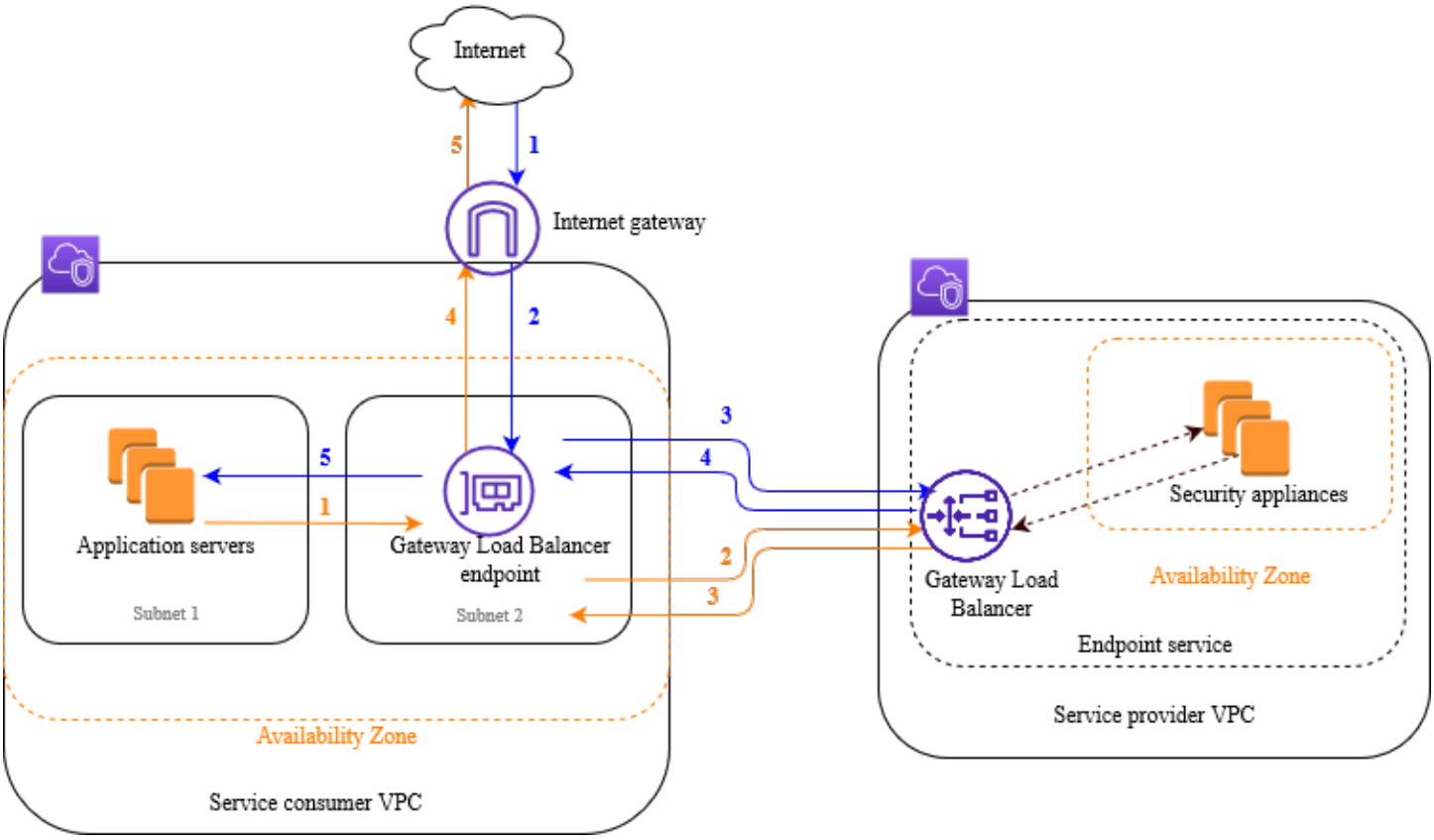
Contents

Overview

A Gateway Load Balancer endpoint is a VPC endpoint that provides private connectivity between virtual appliances in the service provider VPC, and application servers in the service consumer VPC. The Gateway Load Balancer is deployed in the same VPC as that of the virtual appliances. These appliances are registered as a target group of the Gateway Load Balancer.

The application servers run in one subnet (destination subnet) in the service consumer VPC, while the Gateway Load Balancer endpoint is in another subnet of the same VPC. All traffic entering the service consumer VPC through the internet gateway is first routed to the Gateway Load Balancer endpoint and then routed to the destination subnet.

Similarly, all traffic leaving the application servers (destination subnet) is routed to the Gateway Load Balancer endpoint before it is routed back to the internet. The following network diagram is a visual representation of how a Gateway Load Balancer endpoint is used to access an endpoint service.



The numbered items that follow, highlight and explain elements shown in the preceding image.

Traffic from the internet to the application (blue arrows):

1. Traffic enters the service consumer VPC through the internet gateway.
2. Traffic is sent to the Gateway Load Balancer endpoint, as a result of ingress routing.
3. Traffic is sent to the Gateway Load Balancer, which distributes the traffic to one of the security appliances.
4. Traffic is sent back to the Gateway Load Balancer endpoint after it is inspected by the security appliance.
5. Traffic is sent to the application servers (destination subnet).

Traffic from the application to the internet (orange arrows):

1. Traffic is sent to the Gateway Load Balancer endpoint as a result of the default route configured on the application server subnet.

2. Traffic is sent to the Gateway Load Balancer, which distributes the traffic to one of the security appliances.
3. Traffic is sent back to the Gateway Load Balancer endpoint after it is inspected by the security appliance.
4. Traffic is sent to the internet gateway based on the route table configuration.
5. Traffic is routed back to the internet.

Routing

The route table for the internet gateway must have an entry that routes traffic destined for the application servers to the Gateway Load Balancer endpoint. To specify the Gateway Load Balancer endpoint, use the ID of the VPC endpoint. The following example shows the routes for a dualstack configuration.

Destination	Target
VPC IPv4 CIDR	Local
VPC IPv6 CIDR	Local
Subnet 1 IPv4 CIDR	*vpc-endpoint-id*
Subnet 1 IPv6 CIDR	*vpc-endpoint-id*

The route table for the subnet with the application servers must have entries that route all traffic from the application servers to the Gateway Load Balancer endpoint.

Destination	Target
VPC IPv4 CIDR	Local
VPC IPv6 CIDR	Local
0.0.0.0/0	*vpc-endpoint-id*
::/0	*vpc-endpoint-id*

The route table for the subnet with the Gateway Load Balancer endpoint must route traffic that returns from inspection to its final destination. For traffic that originated from the internet, the local route ensures that it reaches the application servers. For traffic that originated from the application servers, add entries that route all traffic to the internet gateway.

Destination	Target
VPC IPv4 CIDR	Local
VPC IPv6 CIDR	Local
0.0.0.0/0	*internet-gateway-id*
::/0	*internet-gateway-id*

Prerequisites

- Ensure that the service consumer VPC has at least two subnets for each Availability Zone that contains application servers. One subnet is for the Gateway Load Balancer endpoint, and the other is for the application servers.
- The Gateway Load Balancer and the targets can be in the same subnet.
- You cannot use a subnet that is shared from another account to deploy the Gateway Load Balancer.
- Launch at least one security appliance instance in each security appliance subnet in the service provider VPC. The security groups for these instances must allow UDP traffic on port 6081.

Step 1: Create a Gateway Load Balancer

Use the following procedure to create your load balancer, listener, and target group.

To create the load balancer, listener, and target group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Choose **Create load balancer**.
4. Under **Gateway Load Balancer**, choose **Create**.

5. **Basic configuration**

   a. For **Load balancer name**, enter a name for your load balancer.

   b. For **IP address type**, choose **IPv4** to support IPv4 addresses only or **Dualstack** to support both IPv4 and IPv6 addresses.

6. **Network mapping**

   a. For **VPC**, select the service provider VPC.

   b. For **Mappings**, select all of the Availability Zones in which you launched security appliance instances, and one subnet per Availability Zone.

7. **IP listener routing**

   a. For **Default action**, select an existing target group to receive traffic. This target group must use the GENEVE protocol.

      If you don't have a target group, choose **Create target group**, which opens a new tab in your browser. Choose a target type, enter a name for the target group, and keep the GENEVE protocol. Select the VPC with your security appliance instances. Modify the health check settings as needed, and add any tags that you need. Choose **Next**. You can register your security appliance instances with the target group now, or after you finish this procedure. Choose **Create target group** and then return to the previous browser tab.

   b. (Optional) Expand **Listener tags** and add the tags that you need.

8. (Optional) Expand **Load balancer tags** and add the tags that you need.

9. Choose **Create load balancer**.

Step 2: Create a Gateway Load Balancer endpoint service

Use the following procedure to create an endpoint service using your Gateway Load Balancer.

To create a Gateway Load Balancer endpoint service

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

2. In the navigation pane, choose **Endpoint services**.

3. Choose **Create endpoint service** and do the following:

   a. For **Load balancer type**, choose **Gateway**.

b. For **Available load balancers**, select your Gateway Load Balancer.

c. For **Require acceptance for endpoint**, select **Acceptance required** to accept connection requests to your service manually. Otherwise, they are automatically accepted.

d. For **Supported IP address types**, do one of the following:

- Select **IPv4** – Enable the endpoint service to accept IPv4 requests.
- Select **IPv6** – Enable the endpoint service to accept IPv6 requests.
- Select **IPv4** and **IPv6** – Enable the endpoint service to accept both IPv4 and IPv6 requests.

e. (Optional) To add a tag, choose **Add new tag** and enter the tag key and tag value.

f. Choose **Create**. Note the service name; you'll need it when you create the endpoint.

4. Select the new endpoint service and choose **Actions**, **Allow principals**. Enter the ARNs of the service consumers that are allowed to create an endpoint to your service. A service consumer can be a user, IAM role, or AWS account. Choose **Allow principals**.

Step 3: Create a Gateway Load Balancer endpoint

Use the following procedure to create a Gateway Load Balancer endpoint that connects to your Gateway Load Balancer endpoint service. Gateway Load Balancer endpoints are zonal. We recommend that you create one Gateway Load Balancer endpoint per zone. For more information, see Access virtual appliances through AWS PrivateLink in the *AWS PrivateLink Guide*.

To create a Gateway Load Balancer endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create endpoint** and do the following:

a. For **Service category**, choose **Other endpoint services**.

b. For **Service name**, enter the service name that you noted earlier, and then choose **Verify service**.

c. For **VPC**, select the service consumer VPC.

d. For **Subnets**, select a subnet for the Gateway Load Balancer endpoint.

Note: You can only select one subnet within each Availability Zone when creating a Gateway Load Balancer endpoint.

e. For **IP address type**, choose from the following options:

- **IPv4** – Assign IPv4 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have IPv4 address ranges.
- **IPv6** – Assign IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets are IPv6 only subnets.
- **Dualstack** – Assign both IPv4 and IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have both IPv4 and IPv6 address ranges.

f. (Optional) To add a tag, choose **Add new tag** and enter the tag key and tag value.

g. Choose **Create endpoint**. The initial status is `pending acceptance`.

To accept the endpoint connection request, use the following procedure.

1. In the navigation pane, choose **Endpoint services**.
2. Select the endpoint service.
3. From the **Endpoint connections** tab, select the endpoint connection.
4. To accept the connection request, choose **Actions**, **Accept endpoint connection request**. When prompted for confirmation, enter **`accept`** and then choose **Accept**.

Step 4: Configure routing

Configure the route tables for the service consumer VPC as follows. This allows the security appliances to perform security inspection on inbound traffic that's destined for the application servers.

To configure routing

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Route tables**.
3. Select the route table for the internet gateway and do the following:

   a. Choose **Actions**, **Edit routes**.

b. Choose **Add route**. For **Destination**, enter the IPv4 CIDR block of the subnet for the application servers. For **Target**, select the VPC endpoint.

c. If you support IPv6, choose **Add route**. For **Destination**, enter the IPv6 CIDR block of the subnet for the application servers. For **Target**, select the VPC endpoint.

d. Choose **Save changes**.

4. Select the route table for the subnet with the application servers and do the following:

a. Choose **Actions**, **Edit routes**.

b. Choose **Add route**. For **Destination**, enter **`0.0.0.0/0`**. For **Target**, select the VPC endpoint.

c. If you support IPv6, choose **Add route**. For **Destination**, enter **`::/0`**. For **Target**, select the VPC endpoint.

d. Choose **Save changes**.

5. Select the route table for the subnet with the Gateway Load Balancer endpoint, and do the following:

a. Choose **Actions**, **Edit routes**.

b. Choose **Add route**. For **Destination**, enter **`0.0.0.0/0`**. For **Target**, select the internet gateway.

c. If you support IPv6, choose **Add route**. For **Destination**, enter **`::/0`**. For **Target**, select the internet gateway.

d. Choose **Save changes**.

Getting started with Gateway Load Balancers using the AWS CLI

Gateway Load Balancers make it easy to deploy, scale, and manage third-party virtual appliances, such as security appliances.

In this tutorial, we'll implement an inspection system using a Gateway Load Balancer and a Gateway Load Balancer endpoint.

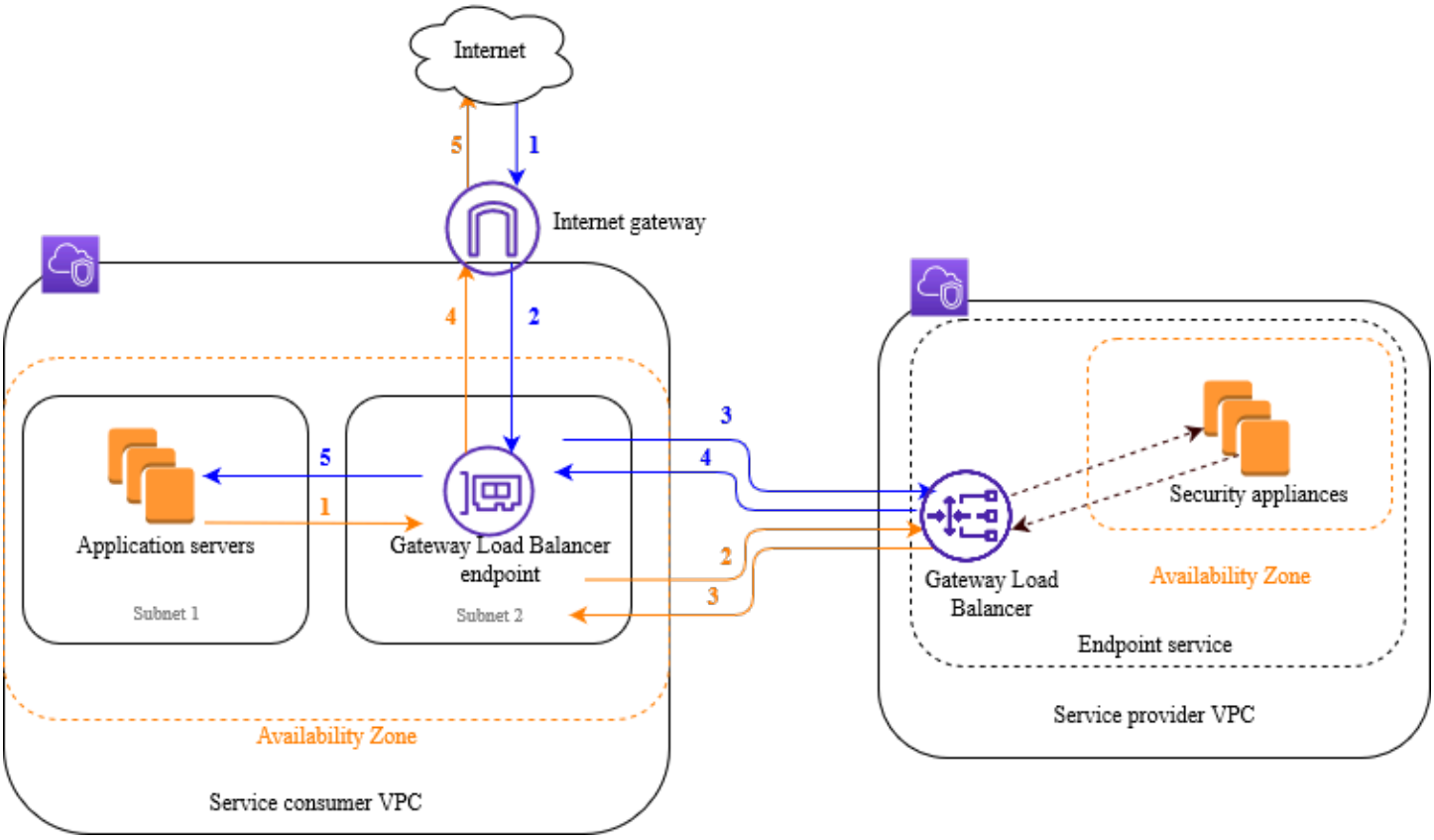
Contents

Overview

A Gateway Load Balancer endpoint is a VPC endpoint that provides private connectivity between virtual appliances in the service provider VPC, and application servers in the service consumer VPC. The Gateway Load Balancer is deployed in the same VPC as that of the virtual appliances. These appliances are registered as a target group of the Gateway Load Balancer.

The application servers run in one subnet (destination subnet) in the service consumer VPC, while the Gateway Load Balancer endpoint is in another subnet of the same VPC. All traffic entering the service consumer VPC through the internet gateway is first routed to the Gateway Load Balancer endpoint and then routed to the destination subnet.

Similarly, all traffic leaving the application servers (destination subnet) is routed to the Gateway Load Balancer endpoint before it is routed back to the internet. The following network diagram is a visual representation of how a Gateway Load Balancer endpoint is used to access an endpoint service.



The numbered items that follow, highlight and explain elements shown in the preceding image.

Traffic from the internet to the application (blue arrows):

1. Traffic enters the service consumer VPC through the internet gateway.
2. Traffic is sent to the Gateway Load Balancer endpoint, as a result of ingress routing.
3. Traffic is sent to the Gateway Load Balancer, which distributes the traffic to one of the security appliances.
4. Traffic is sent back to the Gateway Load Balancer endpoint after it is inspected by the security appliance.
5. Traffic is sent to the application servers (destination subnet).

Traffic from the application to the internet (orange arrows):

1. Traffic is sent to the Gateway Load Balancer endpoint as a result of the default route configured on the application server subnet.

2. Traffic is sent to the Gateway Load Balancer, which distributes the traffic to one of the security appliances.
3. Traffic is sent back to the Gateway Load Balancer endpoint after it is inspected by the security appliance.
4. Traffic is sent to the internet gateway based on the route table configuration.
5. Traffic is routed back to the internet.

Routing

The route table for the internet gateway must have an entry that routes traffic destined for the application servers to the Gateway Load Balancer endpoint. To specify the Gateway Load Balancer endpoint, use the ID of the VPC endpoint. The following example shows the routes for a dualstack configuration.

Destination	Target
VPC IPv4 CIDR	Local
VPC IPv6 CIDR	Local
Subnet 1 IPv4 CIDR	*vpc-endpoint-id*
Subnet 1 IPv6 CIDR	*vpc-endpoint-id*

The route table for the subnet with the application servers must have entries that route all traffic from the application servers to the Gateway Load Balancer endpoint.

Destination	Target
VPC IPv4 CIDR	Local
VPC IPv6 CIDR	Local
0.0.0.0/0	*vpc-endpoint-id*
::/0	*vpc-endpoint-id*

The route table for the subnet with the Gateway Load Balancer endpoint must route traffic that returns from inspection to its final destination. For traffic that originated from the internet, the local route ensures that it reaches the application servers. For traffic that originated from the application servers, add entries that route all traffic to the internet gateway.

Destination	Target
VPC IPv4 CIDR	Local
VPC IPv6 CIDR	Local
0.0.0.0/0	*internet-gateway-id*
::/0	*internet-gateway-id*

Prerequisites

- Install the AWS CLI or update to the current version of the AWS CLI if you are using a version that does not support Gateway Load Balancers. For more information, see Installing the AWS CLI in the *AWS Command Line Interface User Guide*.
- Ensure that the service consumer VPC has at least two subnets for each Availability Zone that contains application servers. One subnet is for the Gateway Load Balancer endpoint, and the other is for the application servers.
- Ensure that the service provider VPC has at least two subnets for each Availability Zone that contains security appliance instances. One subnet is for the Gateway Load Balancer, and the other is for the instances.
- Launch at least one security appliance instance in each security appliance subnet in the service provider VPC. The security groups for these instances must allow UDP traffic on port 6081.

Step 1: Create a Gateway Load Balancer and register targets

Use the following procedure to create your load balancer, listener, and target groups, and to register your security appliance instances as targets.

To create a Gateway Load Balancer and register targets

1. Use the create-load-balancer command to create a load balancer of type `gateway`. You can specify one subnet for each Availability Zone in which you launched security appliance instances.

```
aws elbv2 create-load-balancer --name my-load-balancer --type gateway --
subnets provider-subnet-id
```

   The default is to support IPv4 addresses only. To support both IPv4 and IPv6 addresses, add the `--ip-address-type dualstack` option.

   The output includes the Amazon Resource Name (ARN) of the load balancer, with the format shown in the following example.

```
arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/gwy/my-load-
balancer/1234567890123456
```

2. Use the create-target-group command to create a target group, specifying the service provider VPC in which you launched your instances.

```
aws elbv2 create-target-group --name my-targets --protocol GENEVE --port 6081 --
vpc-id provider-vpc-id
```

   The output includes the ARN of the target group, with the following format.

```
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/0123456789012345
```

3. Use the register-targets command to register your instances with your target group.

```
aws elbv2 register-targets --target-group-arn targetgroup-arn --targets
 Id=i-1234567890abcdef0 Id=i-0abcdef1234567890
```

4. Use the create-listener command to create a listener for your load balancer with a default rule that forwards requests to your target group.

```
aws elbv2 create-listener --load-balancer-arn loadbalancer-arn --default-actions
 Type=forward,TargetGroupArn=targetgroup-arn
```

The output contains the ARN of the listener, with the following format.

```
arn:aws:elasticloadbalancing:us-east-2:123456789012:listener/gwy/my-load-
balancer/1234567890123456/abc1234567890123
```

5. (Optional) You can verify the health of the registered targets for your target group using the following describe-target-health command.

```
aws elbv2 describe-target-health --target-group-arn targetgroup-arn
```

Step 2: Create a Gateway Load Balancer endpoint

Use the following procedure to create a Gateway Load Balancer endpoint. Gateway Load Balancer endpoints are zonal. We recommend that you create one Gateway Load Balancer endpoint per zone. For more information, see Access virtual appliances through AWS PrivateLink.

To create a Gateway Load Balancer endpoint

1. Use the create-vpc-endpoint-service-configuration command to create an endpoint service configuration using your Gateway Load Balancer.

```
aws ec2 create-vpc-endpoint-service-configuration --gateway-load-balancer-
arns loadbalancer-arn --no-acceptance-required
```

   To support both IPv4 and IPv6 addresses, add the `--supported-ip-address-types ipv4 ipv6` option.

   The output contains the service ID (for example, vpce-svc-12345678901234567) and the service name (for example, com.amazonaws.vpce.us-east-2.vpce-svc-12345678901234567).

2. Use the modify-vpc-endpoint-service-permissions command to allow service consumers to create an endpoint to your service. A service consumer can be a user, IAM role, or AWS account. The following example adds permission for the specified AWS account.

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-
svc-12345678901234567 --add-allowed-principals arn:aws:iam::123456789012:root
```

3. Use the create-vpc-endpoint command to create the Gateway Load Balancer endpoint for your service.

```
aws ec2 create-vpc-endpoint --vpc-endpoint-type GatewayLoadBalancer --service-
name com.amazonaws.vpce.us-east-2.vpce-svc-12345678901234567 --vpc-id consumer-vpc-
id --subnet-ids consumer-subnet-id
```

   To support both IPv4 and IPv6 addresses, add the `--ip-address-type dualstack` option.

   The output contains the ID of the Gateway Load Balancer endpoint (for example, vpce-01234567890abcdef).

Step 3: Configure routing

Configure the route tables for the service consumer VPC as follows. This allows the security appliances to perform security inspection on inbound traffic that's destined for the application servers.

To configure routing

1. Use the create-route command to add entries to the route table for the internet gateway that routes traffic that's destined for the application servers to the Gateway Load Balancer endpoint.

```
aws ec2 create-route --route-table-id gateway-rtb --destination-cidr-block Subnet 1
 IPv4 CIDR --vpc-endpoint-id vpce-01234567890abcdef
```

   If you support IPv6, add the following route.

```
aws ec2 create-route --route-table-id gateway-rtb --destination-cidr-block Subnet 1
 IPv6 CIDR --vpc-endpoint-id vpce-01234567890abcdef
```

2. Use the create-route command to add an entry to the route table for the subnet with the application servers that routes all traffic from the application servers to the Gateway Load Balancer endpoint.

```
aws ec2 create-route --route-table-id application-rtb --destination-cidr-block
 0.0.0.0/0 --vpc-endpoint-id vpce-01234567890abcdef
```

If you support IPv6, add the following route.

```
aws ec2 create-route --route-table-id application-rtb --destination-cidr-block ::/0
 --vpc-endpoint-id vpce-01234567890abcdef
```

3. Use the create-route command to add an entry to the route table for the subnet with the Gateway Load Balancer endpoint that routes all traffic that originated from the application servers to the internet gateway.

```
aws ec2 create-route --route-table-id endpoint-rtb --destination-cidr-block
 0.0.0.0/0 --gateway-id igw-01234567890abcdef
```

If you support IPv6, add the following route.

```
aws ec2 create-route --route-table-id endpoint-rtb --destination-cidr-block ::/0 --
gateway-id igw-01234567890abcdef
```

4. Repeat for each application subnet route table in each zone.

Gateway Load Balancers

Use a Gateway Load Balancer to deploy and manage a fleet of virtual appliances that support the GENEVE protocol.

A Gateway Load Balancer operates at the third layer of the Open Systems Interconnection (OSI) model. It listens for all IP packets across all ports and forwards traffic to the target group that's specified in the listener rule, using the GENEVE protocol on port 6081.

You can add or remove targets from your load balancer as your needs change, without disrupting the overall flow of requests. ELB scales your load balancer as traffic to your application changes over time. ELB can scale to the vast majority of workloads automatically.

Contents

Load balancer state

A Gateway Load Balancer can be in one of the following states:

`provisioning`

The Gateway Load Balancer is being set up.

`active`

The Gateway Load Balancer is fully set up and ready to route traffic.

`failed`

The Gateway Load Balancer could not be set up.

IP address type

You can set the types of IP addresses that the application servers can use to access your Gateway Load Balancers.

Gateway Load Balancers support the following IP address types:

`ipv4`

Only IPv4 is supported.

`dualstack`

Both IPv4 and IPv6 are supported.

Considerations

- The virtual private cloud (VPC) and subnets that you specify for the load balancer must have associated IPv6 CIDR blocks.
- The route tables for the subnets in the service consumer VPC must route IPv6 traffic, and the network ACLs for these subnets must allow IPv6 traffic.
- A Gateway Load Balancer encapsulates both IPv4 and IPv6 client traffic with an IPv4 GENEVE header and sends it to the appliance. The appliance encapsulates both IPv4 and IPv6 client traffic with an IPv4 GENEVE header and sends it back to the Gateway Load Balancer.

For more information about IP address types, see Update the IP address types for your Gateway Load Balancer.

Availability Zones

When you create a Gateway Load Balancer, you enable one or more Availability Zones, and specify the subnet that corresponds to each zone. When you enable multiple Availability Zones, it ensures that the load balancer can continue to route traffic even if an Availability Zone becomes unavailable. The subnets that you specify must each have at least 8 available IP addresses. Subnets cannot be removed after the load balancer is created. To remove a subnet, you must create a new load balancer.

Idle timeout

For each TCP request made through a Gateway Load Balancer, the state of that connection is tracked. If no data is sent through the connection by either the client or target for longer than the idle timeout, the connection is closed. After the idle timeout period elapses, the load balancer considers the next TCP SYN as a new flow and routes it to a new target. However, data packets sent after the idle timeout period elapses are dropped.

The default idle timeout value for TCP flows is 350 seconds, but can be updated to any value between 60-6000 seconds. Clients or targets can use TCP keepalive packets to reset the idle timeout.

> ⚠ **Stickiness limitation**
>
> Your Gateway Load Balancers idle timeout can only be updated when using 5-tuple stickiness. When using 3-tuple or 2-tuple stickness, the default idle timeout value is used. For more information, see Flow stickiness

While UDP is connectionless, the load balancer maintains UDP flow state based on the source and destination IP addresses and ports. This ensures that packets that belong to the same flow are consistently sent to the same target. After the idle timeout period elapses, the load balancer considers the incoming UDP packet as a new flow and routes it to a new target. Elastic Load Balancing sets the idle timeout value for UDP flows to 120 seconds. This cannot be changed.

EC2 instances must respond to a new request within 30 seconds in order to establish a return path.

For more information, see Update idle timeout.

Load balancer attributes

The following are the load balancer attributes for Gateway Load Balancers:

`deletion_protection.enabled`

Indicates whether deletion protection is enabled. The default is `false`.

`load_balancing.cross_zone.enabled`

Indicates whether cross-zone load balancing is enabled. The default is `false`.

For more information, see Edit load balancer attributes.

Network ACLs

If the application servers and the Gateway Load Balancer endpoint are in the same subnet, the NACL rules are evaluated for traffic from the application servers to the Gateway Load Balancer endpoint.

Asymmetric flows

Gateway Load Balancers support asymmetric flows when the load balancer processes the initial flow packet and the response flow packet is not routed through the load balancer. Asymmetric routing is not recommended, because it can result in reduced network performance. Gateway Load Balancers do not support asymmetric flows when the load balancer does not process the initial flow packet but the response flow packet is routed through the load balancer.

Network maximum transmission unit (MTU)

The maximum transmission unit (MTU) is the size of the largest data packet that can be transmitted through the network. The Gateway Load Balancer interface MTU supports packets up to 8,500 bytes. Packets with a size larger than 8500 bytes that arrive at the Gateway Load Balancer interface are dropped.

A Gateway Load Balancer encapsulates IP traffic with a GENEVE header and forwards it to the appliance. The GENEVE encapsulation process adds 68 bytes to the original packet. Therefore, to

support packets up to 8,500 bytes, ensure that the MTU setting of your appliance supports packets of at least 8,568 bytes.

Gateway Load Balancers do not support IP fragmentation. Additionally, Gateway Load Balancers do not generate ICMP message "Destination Unreachable: fragmentation needed and DF set". Due to this, Path MTU Discovery (PMTUD) is not supported.

Create a Gateway Load Balancer

A Gateway Load Balancer takes requests from clients and distributes them across targets in a target group, such as EC2 instances.

To create a Gateway Load Balancer using the AWS Management Console, complete the following tasks. Alternatively, to create a Gateway Load Balancer using the AWS CLI, see Getting started using the CLI.

Tasks

- Prerequisites
- Create the load balancer
- Important next steps

Prerequisites

Before you begin, ensure that the virtual private cloud (VPC) for your Gateway Load Balancer has at least one subnet in each Availability Zone where you have targets.

Create the load balancer

Use the following procedure to create your Gateway Load Balancer. Provide basic configuration information for your load balancer, such as a name and IP address type. Then provide information about your network, and the listener that routes traffic to your target groups. Gateway Load Balancers require target groups that use the GENEVE protocol.

To create the load balancer and listener using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.

3. Choose **Create load balancer**.
4. Under **Gateway Load Balancer**, choose **Create**.
5. **Basic configuration**

   a. For **Load balancer name**, enter a name for your load balancer. For example, **`my-glb`**. The name of your Gateway Load Balancer must be unique within your set of load balancers for the Region. It can have a maximum of 32 characters, can contain only alphanumeric characters and hyphens, and must not begin or end with a hyphen.

   b. For **IP address type**, choose **IPv4** to support IPv4 addresses only or **Dualstack** to support both IPv4 and IPv6 addresses.
6. **Network mapping**

   a. For **VPC**, select the service provider VPC.

   b. For **Mappings**, select all of the Availability Zones in which you launched security appliance instances, and the corresponding public subnets.
7. **IP listener routing**

   a. For **Default action**, select the target group to receive traffic. If you don't have a target group, choose **Create target group**. For more information, see Create a target group.

   b. (Optional) Expand **Listener tags** and add the tags that you need.
8. (Optional) Expand **Load balancer tags** and add the tags that you need.
9. Review your configuration, and then choose **Create load balancer**.

Important next steps

After creating your load balancer, verify that your EC2 instances have passed the initial health check. To test your load balancer, you must create a Gateway Load Balancer endpoint and update your route table to make the Gateway Load Balancer endpoint the next hop. These configurations are set within the Amazon VPC console. For more information, see the Getting started tutorial.

Update the IP address types for your Gateway Load Balancer

You can configure your Gateway Load Balancer so that application servers can access your load balancer using IPv4 addresses only, or using both IPv4 and IPv6 addresses (dualstack). The load balancer communicates with targets based on the IP address type of the target group. For more information, see IP address type.

To update the IP address type using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the load balancer.
4. Choose **Actions**, **Edit IP address type**.
5. For **IP address type**, choose **ipv4** to support IPv4 addresses only or **dualstack** to support both IPv4 and IPv6 addresses.
6. Choose **Save**.

To update the IP address type using the AWS CLI

Use the set-ip-address-type command.

Edit attributes for your Gateway Load Balancer

After you create a Gateway Load Balancer, you can edit its load balancer attributes.

Load balancer attributes

- Deletion protection
- Cross-zone load balancing

Deletion protection

To prevent your Gateway Load Balancer from being deleted accidentally, you can enable deletion protection. By default, deletion protection is disabled.

If you enable deletion protection for your Gateway Load Balancer, you must disable it before you can delete the Gateway Load Balancer.

To enable deletion protection using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the Gateway Load Balancer.

4. Choose **Actions**, **Edit attributes**.
5. On the **Edit load balancer attributes** page, select **Enable** for **Delete Protection**, and then choose **Save**.

To disable deletion protection using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the Gateway Load Balancer.
4. Choose **Actions**, **Edit attributes**.
5. On the **Edit load balancer attributes** page, clear **Enable** for **Delete Protection**, and then choose **Save**.

To enable or disable deletion protection using the AWS CLI

Use the modify-load-balancer-attributes command with the `deletion_protection.enabled` attribute.

Cross-zone load balancing

By default, each load balancer node distributes traffic across the registered targets in its Availability Zone only. If you enable cross-zone load balancing, each Gateway Load Balancer node distributes traffic across the registered targets in all enabled Availability Zones. For more information, see Cross-zone load balancing in the *Elastic Load Balancing User Guide*.

To enable cross-zone load balancing using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the Gateway Load Balancer.
4. Choose **Actions**, **Edit attributes**.
5. On the **Edit load balancer attributes** page, select **Enable** for **Cross-Zone Load Balancing**, and then choose **Save**.

To enable cross-zone load balancing using the AWS CLI

Use the modify-load-balancer-attributes command with the `load_balancing.cross_zone.enabled` attribute.

Tag a Gateway Load Balancer

Tags help you to categorize your load balancers in different ways, for example, by purpose, owner, or environment.

You can add multiple tags to each load balancer. Tag keys must be unique for each Gateway Load Balancer. If you add a tag with a key that is already associated with the load balancer, it updates the value of that tag.

When you are finished with a tag, you can remove it from your Gateway Load Balancer.

Restrictions

- Maximum number of tags per resource—50
- Maximum key length—127 Unicode characters
- Maximum value length—255 Unicode characters
- Tag keys and values are case-sensitive. Allowed characters are letters, spaces, and numbers representable in UTF-8, plus the following special characters: + - = . _ : / @. Do not use leading or trailing spaces.
- Do not use the `aws:` prefix in your tag names or values because it is reserved for AWS use. You can't edit or delete tag names or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

To update the tags for a Gateway Load Balancer using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the Gateway Load Balancer.
4. Choose **Tags**, **Add/Edit Tags**, and then do one or more of the following:
   a. To update a tag, edit the values of **Key** and **Value**.
   b. To add a new tag, choose **Create Tag**. For **Key** and **Value**, enter values.
   c. To delete a tag, choose the delete icon (X) next to the tag.

5. When you have finished updating tags, choose **Save**.

To update the tags for a Gateway Load Balancer using the AWS CLI

Use the add-tags and remove-tags commands.

Delete a Gateway Load Balancer

As soon as your Gateway Load Balancer becomes available, you are billed for each hour or partial hour that you keep it running. When you no longer need the Gateway Load Balancer, you can delete it. As soon as the Gateway Load Balancer is deleted, you stop incurring charges for it.

You can't delete a Gateway Load Balancer if it is in use by another service. For example, if the Gateway Load Balancer is associated with a VPC endpoint service, you must delete the endpoint service configuration before you can delete the associated Gateway Load Balancer.

Deleting a Gateway Load Balancer also deletes its listeners. Deleting a Gateway Load Balancer does not affect its registered targets. For example, your EC2 instances continue to run and are still registered to their target groups. To delete your target groups, see Delete a target group for your Gateway Load Balancer.

To delete a Gateway Load Balancer using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the Gateway Load Balancer.
4. Choose **Actions**, **Delete**.
5. When prompted for confirmation, choose **Yes, Delete**.

To delete a Gateway Load Balancer using the AWS CLI

Use the delete-load-balancer command.

Capacity reservations for your Gateway Load Balancer

Load balancer Capacity Unit (LCU) reservations allow you to reserve a static minimum capacity for your load balancer. Gateway Load Balancers automatically scale to support detected workloads and

meet capacity needs. When minimum capacity is configured, your load balancer continues scaling up or down based on the traffic received, but also prevents the capacity from going lower than the minimum capacity configured.

Consider using LCU reservation in following situations:

- You have an upcoming event that will have a sudden, unusual high traffic and want to ensure your load balancer can support the sudden traffic spike during the event.
- You have unpredictable spiky traffic due to the nature of your workload for a short period.
- You are setting up your load balancer to on-board or migrate your services at a specific start time and need start with a high capacity instead of waiting for auto-scaling to take effect.
- You are migrating workloads between load balancers and want to configure the destination to match the scale of the source.

Estimate the capacity that you need

When determining the amount of capacity you should reserve for your load balancer, we recommend performing load testing or reviewing historical workload data that represents the upcoming traffic you expect. Using the ELB console, you can estimate how much capacity you need to reserve based on the reviewed traffic.

Alternatively, you can refer to CloudWatch metric **ProcessedBytes** to determine the right level of capacity. Capacity for your load balancer is reserved in LCUs, with each LCU being equal to 2.2Mbps. You can use the **PeakBytesPerSecond** metric to see the maximum per-minute throughput traffic on the load balancer, then convert that throughput to LCUs using a conversion rate of 2.2Mbps equals 1 LCU.

If you don't have historical workload data to reference and cannot perform load testing, you can estimate capacity needed using the LCU reservation calculator. The LCU reservation calculator uses data based on historical workloads AWS observe and may not represent your specific workload. For more information, see Load Balancer Capacity Unit Reservation Calculator.

Supported Regions

This feature is available only in the following Regions:

- US East (N. Virginia)
- US East (Ohio)

- US West (Oregon)
- Asia Pacific (Hong Kong)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Europe (Frankfurt)
- Europe (Ireland)
- Europe (Stockholm)

Minimum and maximum values for an LCU reservation

The total reservation request must be at least 2,750 LCU per Availability Zone. The maximum value is determined by the quotas for your account. For more information, see the section called “Load Balancer Capacity Units”.

Request Load balancer Capacity Unit reservation for your Gateway Load Balancer

Before you use LCU reservation, review the following:

- LCU reservation only supports reserving throughput capacity for Gateway Load Balancers. When requesting a LCU reservation, convert your capacity needs from Mbps to LCUs using the conversion rate of 1 LCU to 2.2 Mbps.
- Capacity is reserved at the regional level and is evenly distributed across availability zones. Confirm you have enough evenly distributed targets in each availability zone before turning on LCU reservation.
- LCU reservation requests are fulfilled on a first come first serve basis, and depends on available capacity for a zone at that time. Most requests are typically fulfilled within an hour, but can take up to a few hours.
- To update an existing reservation, the previous request must be provisioned or failed. You can increase reserved capacity as many times as you need, however you can only decrease the reserved capacity two times per day.

Request a LCU reservation

The steps in this procedure explain how to request a LCU reservation on your load balancer.

To request a LCU reservation using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, choose **Load Balancers**.
3. Select the load balancer name.
4. On the **Capacity** tab, choose **Edit LCU Reservation**.
5. Select **Historic reference based estimate**, then select the load balancer from the dropdown list.
6. Select the reference period to view the recommended reserved LCU level.
7. If you do not have historic reference workload, you can choose **Manual estimate** and enter the number of LCUs to be reserved.
8. Choose **Save**.

To request a LCU reservation using AWS CLI

Use the modify-capacity-reservation command.

Update or terminate Load balancer Capacity Unit reservations for your Gateway Load Balancer

Update or terminate a LCU reservation

The steps in this procedure explain how to update or terminate a LCU reservation on your load balancer.

To update or terminate a LCU reservation using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, choose **Load Balancers**.
3. Select the load balancer name.
4. On the **Capacity** tab, confirm the status of reservation is Provisioned.
    a. To update the LCU reservation choose **Edit LCU Reservation**.
    b. To terminate the LCU reservation, choose **Cancel Capacity**.

To update or terminate a LCU reservation using the AWS CLI

Use the modify-capacity-reservation command.

Monitor Load balancer Capacity Unit reservation for your Gateway Load Balancer

Reservation Status

LCU reservation has four available status:

- pending - Indicates the reservation it is in the process of provisioning.
- provisioned - Indicates the reserved capacity is ready and available to use.
- failed - Indicates the request cannot be completed at the time.
- rebalancing - Indicates an availability zone has been added and the load balancer is rebalancing capacity.

Reserved LCU

To determine reserved LCU utilization, you can compare the per-minute **PeakBytesPerSecond** metric with the per-hour Sum(ReservedLCUs). To convert bytes per minute to LCU per hour, use (bytes per min)*8/60/ (10^6)/2.2.

Monitor reserved capacity

The steps in this process explain how to check the status of a LCU reservation on your load balancer.

To view the status of a LCU reservation using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, choose **Load Balancers**.
3. Select the load balancer name.
4. On the **Capacity** tab, you can view the **Reservation Status** and **Reserved LCU** value.

To monitor the status of the LCU reservation using AWS CLI

Use the describe-capacity-reservation command.

Listeners for your Gateway Load Balancers

When you create your Gateway Load Balancer, you add a *listener*. A listener is a process that checks for connection requests.

Listeners for Gateway Load Balancers listen for all IP packets across all ports. You cannot specify a protocol or port when you create a listener for a Gateway Load Balancer.

When you create a listener, you specify a rule for routing requests. This rule forwards requests to the specified target group. You can update the listener rule to forward requests to a different target group.

Listener attributes

The following are the listener attributes for Gateway Load Balancers:

`tcp.idle_timeout.seconds`

The tcp idle timeout value, in seconds. The valid range is 60-6000 seconds. The default is 350 seconds.

For more information, see Update idle timeout.

Update the target group for your Gateway Load Balancer listener

When you create a listener, you specify a rule for routing requests. This rule forwards requests to the specified target group. You can update the listener rule to forward requests to a different target group.

To update your listener using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the load balancer and choose **Listeners**.
4. Choose **Edit listener**.

5. For **Forwarding to target group**, choose a target group.
6. Choose **Save**.

To update your listener using the AWS CLI

Use the modify-listener command.

Update the TCP idle timeout for your Gateway Load Balancer listener

For each TCP request made through a Gateway Load Balancer, the state of that connection is tracked. If no data is sent through the connection by either the client or target for longer than the idle timeout, the connection is closed. The default idle timeout value for TCP flows is 350 seconds, but can be updated to any value between 60-6000 seconds.

To update the TCP idle timeout using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the Gateway Load Balancer.
4. On the listeners tab choose **Actions**, **View listener details**.
5. On the listener details page, in the **Attributes** tab, select **Edit**.
6. On the **Edit listener attributes** page, in the **Listener attributes** section, enter a value for **TCP idle timeout**.
7. Choose **Save changes**

To update the TCP idle timeout using the AWS CLI

Use the modify-listener-attributes command with the `tcp.idle_timeout.seconds` attribute.

Target groups for your Gateway Load Balancers

Each *target group* is used to route requests to one or more registered targets. When you create a listener, you specify a target group for its default action. Traffic is forwarded to the target group that's specified in the listener rule. You can create different target groups for different types of requests.

You define health check settings for your Gateway Load Balancer on a per target group basis. Each target group uses the default health check settings, unless you override them when you create the target group or modify them later on. After you specify a target group in a rule for a listener, the Gateway Load Balancer continually monitors the health of all targets registered with the target group that are in an Availability Zone enabled for the Gateway Load Balancer. The Gateway Load Balancer routes requests to the registered targets that are healthy. For more information, see Health checks for Gateway Load Balancer target groups.

Contents

Routing configuration

Target groups for Gateway Load Balancers support the following protocol and port:

- **Protocol**: GENEVE
- **Port**: 6081

The Gateway Load Balancer encapsulates the original packets using GENEVE. The GENEVE header uses a Type-Length-Value (TLV) format to store information, using Option Class 0x0108. Appliances must decapsulate the TLV pairs to process the original packets. For more information, see the following blog post: Integrate your appliance with a Gateway Load Balancers.

Target type

When you create a target group, you specify its target type, which determines how you specify its targets. After you create a target group, you cannot change its target type.

The following are the possible target types:

`instance`

The targets are specified by instance ID.

`ip`

The targets are specified by IP address.

When the target type is `ip`, you can specify IP addresses from one of the following CIDR blocks:

- The subnets of the VPC for the target group
- 10.0.0.0/8 (RFC 1918)
- 100.64.0.0/10 (RFC 6598)
- 172.16.0.0/12 (RFC 1918)
- 192.168.0.0/16 (RFC 1918)

Important

You can't specify publicly routable IP addresses.

Registered targets

Your Gateway Load Balancer serves as a single point of contact for clients, and distributes incoming traffic across its healthy registered targets. Each target group must have at least one registered

target in each Availability Zone that is enabled for the Gateway Load Balancer. You can register each target with one or more target groups.

If demand increases, you can register additional targets with one or more target groups in order to handle the demand. The Gateway Load Balancer starts routing traffic to a newly registered target as soon as the registration process completes.

If demand decreases, or you need to service your targets, you can deregister targets from your target groups. Deregistering a target removes it from your target group, but does not affect the target otherwise. The Gateway Load Balancer stops routing traffic to a target as soon as it is deregistered. The target enters the `draining` state until in-flight requests have completed. You can register the target with the target group again when you are ready for it to resume receiving traffic.

Target group attributes

You can use the following attributes with target groups:

`deregistration_delay.timeout_seconds`

The amount of time for ELB to wait before changing the state of a deregistering target from `draining` to `unused`. The range is 0-3600 seconds. The default value is 300 seconds.

`stickiness.enabled`

Indicates whether configurable flow stickiness is enabled for the target group. The possible values are `true` or `false`. The default is false. When the attribute is set to `false`, 5_tuple is used.

`stickiness.type`

Indicates the type of the flow stickiness. The possible values for target groups associated to Gateway Load Balancers are:

- `source_ip_dest_ip`
- `source_ip_dest_ip_proto`

`target_failover.on_deregistration`

Indicates how the Gateway Load Balancer handles existing flows when a target is deregistered. The possible values are `rebalance` and `no_rebalance`. The default is

no_rebalance. The two attributes (target_failover.on_deregistration and target_failover.on_unhealthy) can't be set independently. The value you set for both attributes must be the same.

target_failover.on_unhealthy

Indicates how the Gateway Load Balancer handles existing flows when a target is unhealthy. The possible values are rebalance and no_rebalance. The default is no_rebalance. The two attributes (target_failover.on_deregistration and target_failover.on_unhealthy) cannot be set independently. The value you set for both attributes must be the same.

For more information, see Edit target group attributes.

Create a target group for your Gateway Load Balancer

You register targets for your Gateway Load Balancer using a target group.

To route traffic to the targets in a target group, create a listener and specify the target group in the default action for the listener. For more information, see Listeners.

You can add or remove targets from your target group at any time. For more information, see Register targets. You can also modify the health check settings for your target group. For more information, see Modify health check settings.

To create a target group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Choose **Create target group**.
4. **Basic configuration**
   a. For **Choose a target type**, select **Instances** to specify targets by instance ID, or select **IP addresses** to specify targets by IP address.
   b. For **Target group name**, enter a name for the target group. This name must be unique per Region per account, can have a maximum of 32 characters, must contain only alphanumeric characters or hyphens, and must not begin or end with a hyphen.

c. Verify that **Protocol** is GENEVE and **Port** is 6081. No other protocols or ports are supported.

d. For **VPC**, select the virtual private cloud (VPC) with the security appliance instances to include in your target group.

5. (Optional) For **Health checks**, modify the settings and advanced settings as needed. If health checks consecutively exceed the **Unhealthy threshold** count, the load balancer takes the target out of service. If health checks consecutively exceed the **Healthy threshold** count, the load balancer puts the target back in service. For more information, see Health checks for Gateway Load Balancer target groups.
6. (Optional) Expand **Tags** and add the tags that you need.
7. Choose **Next**.
8. For **Register targets** add one or more targets as follows:
   - If the target type is **Instances**, select one or more instances, enter one or more ports, and then choose **Include as pending below**.
   - If the target type is **IP addresses**, select the network, enter the IP address and ports, and then choose **Include as pending below**.
9. Choose **Create target group**.

To create a target group using the AWS CLI

Use the create-target-group command to create the target group, the add-tags command to tag your target group, and the register-targets command to add targets.

Health checks for Gateway Load Balancer target groups

You register your targets with one or more target groups. Your Gateway Load Balancer starts routing requests to a newly registered target as soon as the registration process completes. It can take a few minutes for the registration process to complete and for health checks to start.

The Gateway Load Balancer periodically sends a request to each registered target to check its status. After each health check is complete, the Gateway Load Balancer closes the connection that was established for the health check.

Health check settings

You configure active health checks for the targets in a target group by using the following settings. If the health checks exceed the specified number of **UnhealthyThresholdCount** consecutive failures, the Gateway Load Balancer takes the target out of service. When the health checks exceed the specified number of **HealthyThresholdCount** consecutive successes, the Gateway Load Balancer puts the target back in service.

Setting	Description
HealthCheckProtocol	The protocol that the load balancer uses when performing health checks on targets. The possible protocols are HTTP, HTTPS, and TCP. The default is TCP.
HealthCheckPort	The port that Gateway Load Balancer uses when performing health checks on targets. The range is 1 to 65535. The default is 80.
HealthCheckPath	[HTTP/HTTPS health checks] The health check path that is the destination on the targets for health checks. The default is /.
HealthCheckTimeoutSeconds	The amount of time, in seconds, during which no response from a target means a failed health check. The range is 2 to 120. The default is 5.
HealthCheckIntervalSeconds	The approximate amount of time, in seconds, between health checks of an individual target. The range is 5 to 300. The default is 10 seconds. This value must be greater than or equal to **HealthCheckTimeoutSeconds**. **Important** Health checks for Gateway Load Balancers are distributed and use a

Setting	Description
	consensus mechanism to determine target health. Therefore, you should expect target appliances to receive several health checks within the configured time interval.
HealthyThresholdCount	The number of consecutive successful health checks required before considering an unhealthy target healthy. The range is 2 to 10. The default is 5.
UnhealthyThresholdCount	The number of consecutive failed health checks required before considering a target unhealthy. The range is 2 to 10. The default is 2.
Matcher	[HTTP/HTTPS health checks] The HTTP codes to use when checking for a successful response from a target. This value must be 200-399.

Target health status

Before the Gateway Load Balancer sends a health check request to a target, you must register it with a target group, specify its target group in a listener rule, and ensure that the Availability Zone of the target is enabled for the Gateway Load Balancer.

The following table describes the possible values for the health status of a registered target.

Value	Description
`initial`	The Gateway Load Balancer is in the process of registering the target or performing the initial health checks on the target.

Value	Description
	Related reason codes: `Elb.RegistrationIn Progress` | `Elb.InitialHealthChecking`
`healthy`	The target is healthy. Related reason codes: None
`unhealthy`	The target did not respond to a health check or failed the health check. Related reason code: `Target.FailedHealt hChecks`
`unused`	The target is not registered with a target group, the target group is not used in a listener rule, the target is in an Availability Zone that is not enabled, or the target is in the stopped or terminated state. Related reason codes: `Target.NotRegistered` | `Target.NotInUse` | `Target.InvalidState` | `Target.IpUnusable`
`draining`	The target is deregistering and connection draining is in process. Related reason code: `Target.Deregistrat ionInProgress`
`unavailable`	Target health is unavailable. Related reason code: `Elb.InternalError`

Health check reason codes

If the status of a target is any value other than `Healthy`, the API returns a reason code and a description of the issue, and the console displays the same description. Reason codes that begin

with `Elb` originate on the Gateway Load Balancer side and reason codes that begin with `Target` originate on the target side.

Reason code	Description
`Elb.InitialHealthChecking`	Initial health checks in progress
`Elb.InternalError`	Health checks failed due to an internal error
`Elb.RegistrationInProgress`	Target registration is in progress
`Target.DeregistrationInProgress`	Target deregistration is in progress
`Target.FailedHealthChecks`	Health checks failed
`Target.InvalidState`	Target is in the stopped state Target is in the terminated state Target is in the terminated or stopped state Target is in an invalid state
`Target.IpUnusable`	The IP address cannot be used as a target, as it is in use by a load balancer
`Target.NotInUse`	Target group is not configured to receive traffic from the Gateway Load Balancer Target is in an Availability Zone that is not enabled for the Gateway Load Balancer
`Target.NotRegistered`	Target is not registered to the target group

Gateway Load Balancer target failure scenarios

Existing flows: By default, existing flows go to the same target unless the flow times out or is reset, regardless of the health and registration status of the target. This approach facilitates connection

draining, and accommodates 3rd party firewalls that are sometimes unable to respond to health checks due to high CPU usage. For more information, see the section called “Target failover”.

New flows: New flows are sent to a healthy target. When a load balancing decision for a flow has been made, the Gateway Load Balancer will send the flow to the same target even if that target becomes unhealthy, or other targets become healthy.

When all targets are unhealthy, the Gateway Load Balancer picks a target at random and forwards traffic to it for the life of the flow, until it is either reset or has timed out. Because traffic is being forwarded to an unhealthy target, traffic is dropped until that target becomes healthy again.

TLS 1.3: If a target group is configured with HTTPS health checks, its registered targets fail health checks if they support only TLS 1.3. These targets must support an earlier version of TLS, such as TLS 1.2.

Cross-zone load balancing: By default, load balancing across Availability Zones is disabled. If load balancing across zones is enabled, each Gateway Load Balancer is able to see all targets in all Availability Zones, and they are all treated the same, regardless of their zone.

Load balancing and health check decisions are always independent among zones. Even when load balancing across zones is enabled, the behavior for existing flows and new flows is the same as described above. For more information, see Cross-zone load balancing in the *Elastic Load Balancing User Guide*.

Check the health of your targets

You can check the health status of the targets registered with your target groups.

To check the health of your targets using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Choose the name of the target group to open its details page.
4. On the **Targets** tab, the **Status** column indicates the status of each target.
5. If the target status is any value other than `Healthy`, the **Status details** column contains more information.

To check the health of your targets using the AWS CLI

Use the describe-target-health command. The output of this command contains the target health state. It includes a reason code if the status is any value other than `Healthy`.

To receive email notifications about unhealthy targets

Use CloudWatch alarms to trigger a Lambda function to send details about unhealthy targets. For step-by-step instructions, see the following blog post: Identifying unhealthy targets of your load balancer.

Modify health check settings

You can modify some of the health check settings for your target group.

To modify health check settings for a target group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Choose the name of the target group to open its details page.
4. On the **Group details** tab, in the **Health check settings** section, choose **Edit**.
5. On the **Edit health check settings** page, modify the settings as needed, and then choose **Save changes**.

To modify health check settings for a target group using the AWS CLI

Use the modify-target-group command.

Edit target group attributes for your Gateway Load Balancer

After you create a target group for your Gateway Load Balancer, you can edit its target group attributes.

Target group attributes

- Target failover
- Deregistration delay
- Flow stickiness

Target failover

With target failover, you specify how the Gateway Load Balancer handles existing traffic flows after a target becomes unhealthy or when the target is deregistered. By default, the Gateway Load Balancer continues to send existing flows to the same target, even if the target has failed or is deregistered. You can manage these flows by either rehashing them (`rebalance`) or leaving them at the default state (`no_rebalance`).

No rebalance:

The Gateway Load Balancer continues to send existing flows to failed or drained targets. If the Gateway Load Balancer cannot reach the target, the traffic is dropped.

However, new flows are sent to healthy targets. This is the default behavior.

Rebalance:

The Gateway Load Balancer rehashes existing flows and sends them to healthy targets after the deregistration delay timeout.

For deregistered targets, the minimum time to failover will depend on the deregistration delay. The target is not marked as deregistered until deregistration delay is completed.

For unhealthy targets, the minimum time to failover will depend on the target group health check configuration (interval times threshold). This is the minimum time before which a target is flagged as unhealthy. After this time, the Gateway Load Balancer can take several minutes due to additional propagation time and TCP retransmission backoff before it reroutes new flows to healthy targets.

To update the target failover attribute using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Choose the name of the target group to open its details page.
4. On the **Group details** page, in the **Attributes** section, choose **Edit**.
5. On the **Edit attributes** page, change the value of **Target failover** as needed.
6. Choose **Save changes**.

To update the target failover attribute using the AWS CLI

Use the modify-target-group-attributes command, with the following key value pairs:

- Key=`target_failover.on_deregistration` and Value= `no_rebalance` (default) or `rebalance`
- Key=`target_failover.on_unhealthy` and Value= `no_rebalance` (default) or `rebalance`

> **Note**
>
> Both attributes (`target_failover.on_deregistration` and `target_failover.on_unhealthy`) must have the same value.

Deregistration delay

When you deregister a target, the Gateway Load Balancer manages flows to that target as follows:

New flows

The Gateway Load Balancer stops sending new flows.

Existing flows

The Gateway Load Balancer handles existing flows based on the protocol:

- **TCP**: Existing flows are closed if they are idle for more than 350 seconds.
- **Other protocols**: Existing flows are closed if they are idle for more than 120 seconds.

To help drain existing flows, you can enable flow rebalancing for your target group. For more information, see the section called “Target failover”.

A deregistered target shows that it is `draining` until the timeout expires. After the deregistration delay timeout expires, the target transitions to an `unused` state.

To update the deregistration delay attribute using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Choose the name of the target group to open its details page.

4. On the **Group details** page, in the **Attributes** section, choose **Edit**.
5. On the **Edit attributes** page, change the value of **Deregistration delay** as needed.
6. Choose **Save changes**.

To update the deregistration delay attribute using the AWS CLI

Use the modify-target-group-attributes command.

Flow stickiness

By default, the Gateway Load Balancer maintains stickiness of flows to a specific target appliance using 5-tuple (for TCP/UDP flows). 5-tuple includes source IP, source port, destination IP, destination port, and transport protocol. You can use the stickiness type attribute to modify the default (5-tuple) and choose either 3-tuple (source IP, destination IP, and transport protocol) or 2-tuple (source IP and destination IP).

Flow stickiness considerations

- Flow stickiness is configured and applied at the target group level, and it applies to all traffic that goes to the target group.
- 2-tuple and 3-tuple flow stickiness are not supported when AWS Transit Gateway appliance mode is turned on. To use appliance mode on your AWS Transit Gateway, use 5-tuple flow stickiness on your Gateway Load Balancer
- Flow stickiness can lead to uneven distribution of connections and flows, which can impact the availability of the target. It is recommended that you terminate or drain all existing flows before modifying the stickiness type of the target group.

To update the flow stickiness attribute using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Choose the name of the target group to open its details page.
4. On the **Group details** page, in the **Attributes** section, choose **Edit**.
5. On the **Edit attributes** page, change the value of **Flow stickiness** as needed.
6. Choose **Save changes**.

To update the flow stickiness attribute using the AWS CLI

Use the modify-target-group-attributes command with the `stickiness.enabled` and `stickiness.type` target group attributes.

Register targets for your Gateway Load Balancer

When your target is ready to handle requests, you register it with one or more target groups. You can register targets by instance ID or by IP address. The Gateway Load Balancer starts routing requests to the target as soon as the registration process completes and the target passes the initial health checks. It can take a few minutes for the registration process to complete and health checks to start. For more information, see Health checks for Gateway Load Balancer target groups.

If demand on your currently registered targets increases, you can register additional targets in order to handle the demand. If demand on your registered targets decreases, you can deregister targets from your target group. It can take a few minutes for the deregistration process to complete and for the Gateway Load Balancer to stop routing requests to the target. If demand increases subsequently, you can register targets that you deregistered with the target group again. If you need to service a target, you can deregister it and then register it again when servicing is complete.

Contents

Considerations

- Each target group must have at least one registered target in each Availability Zone that is enabled for the Gateway Load Balancer.
- The target type of your target group determines how you register targets with that target group. For more information, see Target type.
- You can't register targets across an inter-Region VPC peering.

- You can't register instances by instance ID across an intra-Region VPC peering, but you can register them by IP address.

Target security groups

When you register EC2 instances as targets, you must ensure that the security groups for these instances allow inbound and outbound traffic on port 6081.

Gateway Load Balancers do not have associated security groups. Therefore, the security groups for your targets must use IP addresses to allow traffic from the load balancer.

Network ACLs

When you register EC2 instances as targets, you must ensure that the network access control lists (ACL) for the subnets for your instances allow traffic on port 6081. The default network ACL for a VPC allows all inbound and outbound traffic. If you create custom network ACLs, verify that they allow the appropriate traffic.

Register targets by instance ID

An instance must be in the `running` state when you register it.

To register targets by instance ID using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Choose the name of the target group to open its details page.
4. On the **Targets** tab, choose **Register targets**.
5. Select the instances, and then choose **Include as pending below**.
6. When you are finished adding instances, choose **Register pending targets**.

To register targets by instance ID using the AWS CLI

Use the register-targets command with the IDs of the instances.

Register targets by IP address

An IP address that you register must be from one of the following CIDR blocks:

- The subnets of the VPC for the target group
- 10.0.0.0/8 (RFC 1918)
- 100.64.0.0/10 (RFC 6598)
- 172.16.0.0/12 (RFC 1918)
- 192.168.0.0/16 (RFC 1918)

To register targets by IP address using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Chose the name of the target group to open its details page.
4. On the **Targets** tab, choose **Register targets**.
5. Choose the network, IP addresses, and ports, and then choose **Include as pending below**.
6. When you are finished specifying addresses, choose **Register pending targets**.

To register targets by IP address using the AWS CLI

Use the register-targets command with the IP addresses of the targets.

Deregister targets

When you deregister a target, ELB waits until in-flight requests have completed. This is known as *connection draining*. The status of a target is `draining` while connection draining is in progress. After deregistration is complete, status of the target changes to `unused`. For more information, see Deregistration delay.

To deregister targets using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Choose the name of the target group to open its details page.
4. Choose the **Targets** tab.
5. Select the targets and then choose **Deregister**.

To deregister targets using the AWS CLI

Use the deregister-targets command to remove targets.

Tag a target group for your Gateway Load Balancer

Tags help you to categorize your target groups in different ways, for example, by purpose, owner, or environment.

You can add multiple tags to each target group. Tag keys must be unique for each target group. If you add a tag with a key that is already associated with the target group, it updates the value of that tag.

When you are finished with a tag, you can remove it.

Restrictions

- Maximum number of tags per resource—50
- Maximum key length—127 Unicode characters
- Maximum value length—255 Unicode characters
- Tag keys and values are case sensitive. Allowed characters are letters, spaces, and numbers representable in UTF-8, plus the following special characters: + - = . _ : / @. Do not use leading or trailing spaces.
- Do not use the `aws:` prefix in your tag names or values because it is reserved for AWS use. You can't edit or delete tag names or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

To update the tags for a target group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Choose the name of the target group to open its details page.
4. On the **Tags** tab, choose **Manage tags** and do one or more of the following:
   a. To update a tag, enter new values for **Key** and **Value**.
   b. To add a tag, choose **Add tag** and enter values for **Key** and **Value**.
   c. To delete a tag, choose **Remove** next to the tag.
5. When you have finished updating tags, choose **Save changes**.

To update the tags for a target group using the AWS CLI

Use the add-tags and remove-tags commands.

Delete a target group for your Gateway Load Balancer

You can delete a target group if it is not referenced by the forward actions of any listener rules. Deleting a target group does not affect the targets registered with the target group. If you no longer need a registered EC2 instance, you can stop or terminate it.

To delete a target group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Select the target group and choose **Actions**, **Delete**.
4. When prompted for confirmation, choose **Yes, delete**.

To delete a target group using the AWS CLI

Use the delete-target-group command.

Monitor your Gateway Load Balancers

You can use the following features to monitor your Gateway Load Balancers to analyze traffic patterns, and to troubleshoot issues. However, the Gateway Load Balancer does not generate access logs since it is a transparent layer 3 load balancer that does not terminate flows. To receive access logs, you must enable access logging on Gateway Load Balancer target appliances such as firewalls, IDS/IPS, and security appliances. In addition, you can also choose to enable VPC flow logs on Gateway Load Balancers.

CloudWatch metrics

You can use Amazon CloudWatch to retrieve statistics about data points for your Gateway Load Balancers and targets as an ordered set of time-series data, known as *metrics*. You can use these metrics to verify that your system is performing as expected. For more information, see CloudWatch metrics for your Gateway Load Balancer.

VPC Flow Logs

You can use VPC Flow Logs to capture detailed information about the traffic going to and from your Gateway Load Balancer. For more information, see VPC flow logs in the *Amazon VPC User Guide*.

Create a flow log for each network interface for your Gateway Load Balancer. There is one network interface per subnet. To identify the network interfaces for a Gateway Load Balancer, look for the name of the Gateway Load Balancer in the description field of the network interface.

There are two entries for each connection through your Gateway Load Balancer, one for the frontend connection between the client and the Gateway Load Balancer, and the other for the backend connection between the Gateway Load Balancer and the target. If the target is registered by instance ID, the connection appears to the instance as a connection from the client. If the security group of the instance doesn't allow connections from the client but the network ACLs for the subnet allow them, the logs for the network interface for the Gateway Load Balancer show "ACCEPT OK" for the frontend and backend connections, while the logs for the network interface for the instance show "REJECT OK" for the connection.

CloudTrail logs

You can use AWS CloudTrail to capture detailed information about the calls made to the ELB API, and store them as log files in Amazon S3. You can use these CloudTrail logs 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. For more information, see Log API calls for ELB using CloudTrail.

CloudWatch metrics for your Gateway Load Balancer

ELB publishes data points to Amazon CloudWatch for your Gateway Load Balancers and your targets. CloudWatch enables you to retrieve statistics about those data points as an ordered set of time-series data, known as *metrics*. Think of a metric as a variable to monitor, and the data points as the values of that variable over time. For example, you can monitor the total number of healthy targets for a Gateway Load Balancer over a specified time period. Each data point has an associated time stamp and an optional unit of measurement.

You can use metrics to verify that your system is performing as expected. For example, you can create a CloudWatch alarm to monitor a specified metric and initiate an action (such as sending a notification to an email address) if the metric goes outside of what you consider an acceptable range.

ELB reports metrics to CloudWatch only when requests are flowing through the Gateway Load Balancer. If there are requests flowing, ELB measures and sends its metrics in 60-second intervals. If there are no requests flowing or no data for a metric, the metric is not reported.

For more information, see the Amazon CloudWatch User Guide.

Contents

Gateway Load Balancer metrics

The `AWS/GatewayELB` namespace includes the following metrics.

Metric	Description
`ActiveFlowCount`	The total number of concurrent flows (or connections) from clients to targets.

Metric	Description
	Reporting criteria: There is a nonzero value **Statistics**: The most useful statistics are `Average`, `Maximum`, and `Minimum`. **Dimensions** • `LoadBalancer` • `AvailabilityZone ,LoadBalancer`
`ConsumedLCUs`	The number of load balancer capacity units (LCU) used by your load balancer. You pay for the number of LCUs that you use per hour. For more information, see ELB Pricing. **Reporting criteria**: Always reported **Statistics**: All **Dimensions** • `LoadBalancer`
`HealthyHostCount`	The number of targets that are considered healthy. **Reporting criteria**: Reported if health checks are enabled **Statistics**: The most useful statistics are `Maximum` and `Minimum`. **Dimensions** • `LoadBalancer ,TargetGroup` • `AvailabilityZone ,LoadBalancer ,TargetGroup`

Metric	Description
NewFlowCount	The total number of new flows (or connections) established from clients to targets in the time period. **Reporting criteria**: There is a nonzero value **Statistics**: The most useful statistic is Sum. **Dimensions** • LoadBalancer • AvailabilityZone ,LoadBalancer
PeakBytesPerSecond	The highest average bytes processed per second, calculated every 10 seconds during the sampling window. This metric does not include health check traffic. **Reporting criteria**: Always reported **Statistics**: The most useful statistic is Maximum. **Dimensions** • LoadBalancer • AvailabilityZone ,LoadBalancer
PeakPacketsPerSecond	The highest average packet rate (packets processed per second), calculated every 10 seconds during the sampling window. This metric includes health check traffic. **Reporting criteria**: Always reported **Statistics**: The most useful statistic is Maximum. **Dimensions** • LoadBalancer • AvailabilityZone ,LoadBalancer

Metric	Description
ProcessedBytes	The total number of bytes processed by the load balancer. This count includes traffic to and from targets, but not health check traffic. **Reporting criteria**: There is a nonzero value **Statistics**: The most useful statistic is Sum. **Dimensions** • LoadBalancer • AvailabilityZone ,LoadBalancer
ProcessedPackets	The total number of packets processed by the load balancer. This count includes traffic to and from targets, including health check traffic. **Reporting criteria**: Always reported. **Statistics**: The most useful statistic is Sum. **Dimensions** • LoadBalancer • AvailabilityZone ,LoadBalancer
RejectedFlowCount	The total number of flows (or connections) rejected by the load balancer. **Reporting criteria**: Always reported. **Statistics**: The most useful statistics are Average, Maximum, and Minimum. **Dimensions** • LoadBalancer • AvailabilityZone ,LoadBalancer

Metric	Description
RejectedFlowCount_TCP	The number of TCP flows (or connections) rejected by the load balancer. **Reporting criteria**: There is a nonzero value. **Statistics**: The most useful statistic is Sum. **Dimensions** • LoadBalancer • AvailabilityZone ,LoadBalancer
UnHealthyHostCount	The number of targets that are considered unhealthy. **Reporting criteria**: Reported if health checks are enabled **Statistics**: The most useful statistics are Maximum and Minimum. **Dimensions** • LoadBalancer ,TargetGroup • AvailabilityZone ,LoadBalancer ,TargetGroup

Metric dimensions for Gateway Load Balancers

To filter the metrics for your Gateway Load Balancer, use the following dimensions.

Dimension	Description
AvailabilityZone	Filters the metric data by Availability Zone.
LoadBalancer	Filters the metric data by Gateway Load Balancer. Specify the Gateway Load Balancer as follows: gateway/*load-balancer-name*/*1234567890123456* (the final portion of the ARN).

Dimension	Description
`TargetGroup`	Filters the metric data by target group. Specify the target group as follows: targetgroup/*target-group-name*/*1234567890123456* (the final portion of the target group ARN).

View CloudWatch metrics for your Gateway Load Balancer

You can view the CloudWatch metrics for your Gateway Load Balancers by using the Amazon EC2 console. These metrics are displayed as monitoring graphs. The monitoring graphs show data points if the Gateway Load Balancer is active and receiving requests.

Alternatively, you can view metrics for your Gateway Load Balancer using the CloudWatch console.

To view metrics using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. To view metrics filtered by target group, do the following:
   a. In the navigation pane, choose **Target Groups**.
   b. Select your target group and choose **Monitoring**.
   c. (Optional) To filter the results by time, select a time range from **Showing data for**.
   d. To get a larger view of a single metric, select its graph.
3. To view metrics filtered by Gateway Load Balancer, do the following:
   a. In the navigation pane, choose **Load Balancers**.
   b. Select your Gateway Load Balancer and choose **Monitoring**.
   c. (Optional) To filter the results by time, select a time range from **Showing data for**.
   d. To get a larger view of a single metric, select its graph.

To view metrics using the CloudWatch console

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Select the **GatewayELB** namespace.

4. (Optional) To view a metric across all dimensions, enter its name in the search field.

To view metrics using the AWS CLI

Use the following list-metrics command to list the available metrics:

```
aws cloudwatch list-metrics --namespace AWS/GatewayELB
```

To get the statistics for a metric using the AWS CLI

Use the following get-metric-statistics command get statistics for the specified metric and dimension. Note that CloudWatch treats each unique combination of dimensions as a separate metric. You can't retrieve statistics using combinations of dimensions that were not specially published. You must specify the same dimensions that were used when the metrics were created.

```
aws cloudwatch get-metric-statistics --namespace AWS/GatewayELB \
--metric-name UnHealthyHostCount --statistics Average  --period 3600 \
--dimensions Name=LoadBalancer,Value=net/my-load-balancer/50dc6c495c0c9188 \
Name=TargetGroup,Value=targetgroup/my-targets/73e2d6bc24d8a067 \
--start-time 2017-04-18T00:00:00Z --end-time 2017-04-21T00:00:00Z
```

The following is example output.

```
{
    "Datapoints": [
        {
            "Timestamp": "2020-12-18T22:00:00Z",
            "Average": 0.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2020-12-18T04:00:00Z",
            "Average": 0.0,
            "Unit": "Count"
        },
        ...
    ],
    "Label": "UnHealthyHostCount"
}
```

Quotas for your Gateway Load Balancers

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, and other quotas cannot be increased.

To request a quota increase, see Requesting a quota increase in the *Service Quotas User Guide*. If the quota is not yet available in Service Quotas, submit a request for a service quota increase.

Quotas

- Load balancers
- Target groups
- Load Balancer Capacity Units

Load balancers

Your AWS account has the following quotas related to Gateway Load Balancers.

Name	Default	Adjustable
Gateway Load Balancers per Region	100	Yes
Gateway Load Balancers per VPC	100	Yes
Gateway Load Balancer ENIs per VPC	300 *	Yes
Listeners per Gateway Load Balancer	1	No

* Each Gateway Load Balancer uses one network interface per zone.

Target groups

The following quotas are for target groups.

Name	Default	Adjustable
GENEVE target groups per Region	100	Yes
Targets per Availability Zone per GENEVE target group	300	No
Targets per Availability Zone per Gateway Load Balancer	300	No
Targets per Gateway Load Balancer	300	No

Load Balancer Capacity Units

The following quotas are for Load Balancer Capacity Units (LCUs).

Name	Default	Adjustable
Reserved Gateway Load Balancer Capacity Units (LCU) per Region	0	Yes

Document history for Gateway Load Balancers

The following table describes the releases for Gateway Load Balancers.

Change	Description	Date
Capacity Unit reservation	This release adds support to set a minimum capacity for your load balancer.	April 10, 2025
IPv6 support	You can configure your Gateway Load Balancer to support both IPv4 and IPv6 addresses.	December 12, 2022
Flow rebalancing	This release adds support to define the flow handling behaviour for Gateway Load Balancers when targets fail or deregister.	October 13, 2022
Configurable flow stickiness	You can configure the hashing that maintains the stickines s of flows to a specific target appliance.	August 25, 2022
Available in new regions	This release adds support for Gateway Load Balancers in the AWS GovCloud (US) regions.	June 17, 2021
Available in new regions	This release adds support for Gateway Load Balancers in the Canada (Central), Asia Pacific (Seoul), and Asia Pacific (Osaka) region.	March 31, 2021

Available in new regions	This release adds support for Gateway Load Balancers in the US West (N. Californi a), Europe (London), Europe (Paris), Europe (Milan), Africa (Cape Town), Middle East (Bahrain), Asia Pacific (Hong Kong), Asia Pacific (Singapor e), and Asia Pacific (Mumbai) region.	March 19, 2021
Initial release	This release of ELB introduces Gateway Load Balancers.	November 10, 2020