Elastic Load Balancing

Gateway Load Balancers

Elastic Load Balancing: Gateway Load Balancers

Table of Contents

What is a Gateway Load Balancer?

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.

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. The Gateway Load Balancer and its registered virtual appliance instances exchange application traffic using the GENEVE protocol on port 6081. It supports a maximum transmission unit (MTU) size of 8500 bytes.

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 Gateway Load Balancer endpoints (AWS PrivateLink) in the *Amazon VPC User 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 Elastic Load Balancing 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 (p. 2).
To create a Gateway Load Balancer using the AWS Command Line Interface, see Getting started using the CLI (p. 6).

Pricing

With your load balancer, you pay only for what you use. For more information, see Elastic Load Balancing 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.

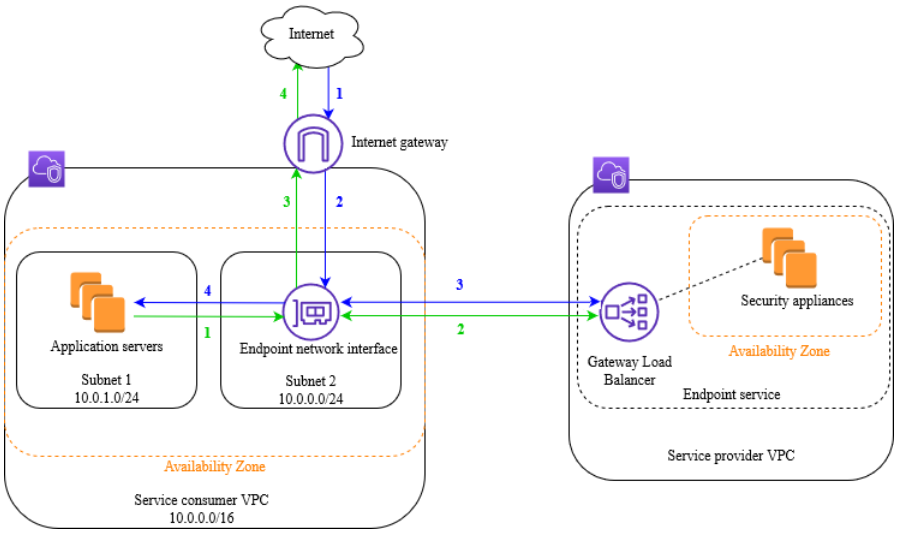
In this tutorial, we'll implement an inspection system using a Gateway Load Balancer and a Gateway Load Balancer endpoint.

Contents

Overview

The security appliances run in a subnet of the service provider VPC. The Gateway Load Balancer is in another subnet of the service provider VPC.

The application servers run in a subnet of the service consumer VPC. The network interface for the Gateway Load Balancer endpoint is in another subnet of the service consumer VPC. All traffic entering the service consumer VPC through the internet gateway is routed to the Gateway Load Balancer endpoint for inspection before it's routed to the destination subnet. Similarly, all traffic leaving the EC2 instance is routed to the Gateway Load Balancer endpoint for inspection before it's routed 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.

Destination	Target
10.0.0.0/16	Local
10.0.1.0/24	*vpc-endpoint-id*

The route table for the subnet with the application servers must have an entry that routes all traffic (0.0.0.0/0) from the application servers to the Gateway Load Balancer endpoint.

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	*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 an entry that routes all traffic (0.0.0.0/0) to the internet gateway.

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	*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.
- 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 group, and to register your security appliance instances as targets.

To create a Gateway Load Balancer and register targets

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. For **Gateway Load Balancer**, choose **Create**.
5. For **Name**, enter a name for your load balancer. For example, **`my-load-balancer`**.
6. For **Availability Zones**, select the service provider VPC. For each Availability Zone in which you launched security appliance instances, select the Availability Zone and then select a public subnet.
7. (Optional) Expand **Tags** and add tags.
8. Choose **Next: Configure Routing**.
9. For **Name**, enter a name for your target group. For example, **`my-targets`**.
10. For **Target type**, select `instance` to specify targets by instance ID or `ip` to specify targets by IP address.
11. **Protocol** must be `GENEVE`, and **Port** must be `6081`.
12. (Optional) For **Health checks**, modify the health check settings as needed.
13. Choose **Next: Register Targets**.
14. Add your instances or IP addresses to the list, and then choose **Next: Review**.
15. Choose **Create**.

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 Gateway Load Balancer endpoints (AWS PrivateLink).

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 **Endpoint Services**.
3. Choose **Create Endpoint Service** and do the following:

   a. For **Associate Load Balancers**, select your Gateway Load Balancer.
   b. For **Require acceptance for endpoint**, select **Acceptance required** to accept connection requests to your service manually. Otherwise, endpoint connections are automatically accepted.
   c. (Optional) To add a tag, choose **Add tag** and then specify the key and value for the tag.
   d. Choose **Create service**. Choose the service ID. Save the service name from the **Details** tab; you'll need it when you create the endpoint.
   e. Choose **Actions**, **Add principals to whitelist**. Enter the ARNs of the service consumers that are allowed to create an endpoint to your service. A service consumer can be an IAM user, IAM role, or AWS account.
4. In the navigation pane, choose **Endpoints**.
5. Choose **Create Endpoint** and do the following:

   a. For **Service category**, choose **Find service by name**.
   b. For **Service name**, enter the service name that you saved earlier, and then choose **Verify**. If the name is found, proceed to the next step. Otherwise, be sure that you used the correct service name.
   c. For **VPC**, select the service consumer VPC.

d. For **Subnets**, select a subnet for the Gateway Load Balancer endpoint.
e. (Optional) To add a tag, choose **Add tag** and specify the key and value for the tag.
f. Choose **Create endpoint**. The initial status is `pending acceptance`.

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. 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 CIDR block of the subnet for the application servers (for example, 10.0.1.0/24). For **Target**, select the VPC endpoint.
   c. Choose **Save routes**.
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. Choose **Save routes**.
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. Choose **Save routes**.

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.

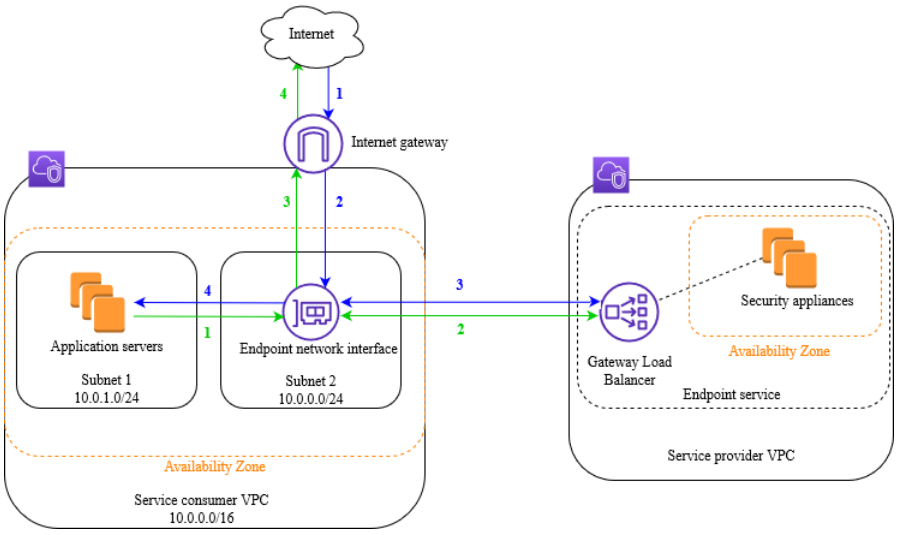
In this tutorial, we'll implement an inspection system using a Gateway Load Balancer and a Gateway Load Balancer endpoint.

Contents

Overview

The security appliances run in a subnet of the service provider VPC. The Gateway Load Balancer is in another subnet of the service provider VPC.

The application servers run in a subnet of the service consumer VPC. The network interface for the Gateway Load Balancer endpoint is in another subnet of the service consumer VPC. All traffic entering the service consumer VPC through the internet gateway is routed to the Gateway Load Balancer endpoint for inspection before it's routed to the destination subnet. Similarly, all traffic leaving the EC2 instance is routed to the Gateway Load Balancer endpoint for inspection before it's routed 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.

Destination	Target
10.0.0.0/16	Local
10.0.1.0/24	*vpc-endpoint-id*

The route table for the subnet with the application servers must have an entry that routes all traffic (0.0.0.0/0) from the application servers to the Gateway Load Balancer endpoint.

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	*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 an entry that routes all traffic (0.0.0.0/0) to the internet gateway.

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	*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 Command Line Interface 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 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 Gateway Load Balancer endpoints (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
```

   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 an IAM 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
```

   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 an entry 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 10.0.1.0/24
 --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
```

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
```

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. Elastic Load Balancing scales your load balancer as traffic to your application changes over time. Elastic Load Balancing 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.

Load balancer attributes

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

`deletion_protection.enabled`

Indicates whether deletion protection (p. 12) is enabled. The default is `false`.

`load_balancing.cross_zone.enabled`

Indicates whether cross-zone load balancing (p. 12) is enabled. The default is `false`.

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, cross-zone load balancing is disabled, so each Gateway Load Balancer node distributes traffic across the registered targets in its Availability Zone only. If a target becomes unhealthy, the load balancer node continues to send existing flows to the unhealthy target and traffic is nulled. The load balancer node sends new flows to healthy targets in the same Availability Zone. If no healthy targets are found, all traffic is nulled.

After you enable cross-zone load balancing, if the load balancer node does not find a healthy target in the same Availability Zone, it can send new flows to healthy targets in a different Availability Zone. If a target becomes unhealthy, the load balancer continues to send existing flows to the unhealthy target until the client resets them.

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.

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.

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.

To create a Gateway Load Balancer using the AWS CLI, see Getting started using the CLI (p. 6).

To create a Gateway Load Balancer using the AWS Management Console, complete the following tasks.

Tasks

Step 1: Configure a load balancer and a listener

First, provide some basic configuration information for your Gateway Load Balancer, such as a name, and a network. The listener for the load balancer listens for all IP packets across all ports.

To configure your load balancer and listener

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. Choose **Create Load Balancer**.
4. For **Gateway Load Balancer**, choose **Create**.
5. For **Name**, enter a name for your Gateway Load Balancer. For example, **`my-load-balancer`**.
6. For **Availability Zones**, select the VPC that you used for your appliance instances. For each Availability Zone that you used to launch your instances, select an Availability Zone and then select the subnet for that Availability Zone.
7. (Optional) Expand **Tags** and add tags.
8. Choose **Next: Configure Routing**.

Step 2: Configure a target group

You register targets, such as EC2 instances, with a target group. The target group that you configure in this step is used as the target group in the listener rule, which forwards requests to the target group. For more information, see Target groups (p. 18).

To configure your target group

1. For **Target group**, keep the default, **New target group**.

2. For **Name**, enter a name for the target group.
3. For **Target type**, select `instance` to specify targets by instance ID or `ip` to specify targets by IP address.
4. **Protocol** must be `GENEVE`, and **Port** must be `6081`.
5. (Optional) For **Health checks**, modify the health check settings as needed.
6. Choose **Next: Register Targets**.

Step 3: Register targets with the target group

You can register EC2 instances as targets in a target group. The target type of the target group determines how you register targets with that target group.

To register targets by instance ID

1. For **Instances**, select one or more instances and choose **Add to registered**.
2. When you have finished adding instances to the list, choose **Next: Review**.

To register targets by IP address

1. For each IP address to register, do the following:
   a. For **Network**, if the IP address is from a subnet of the target group VPC, select the VPC. Otherwise, select **Other private IP address**.
   b. For **IP**, enter the address.
   c. Choose **Add to list**.
2. When you have finished adding IP addresses to the list, choose **Next: Review**.

Step 4: Create the load balancer

After creating your load balancer, you can verify that your EC2 instances have passed the initial health check and then test that the load balancer is sending traffic to your EC2 instances. When you are finished with your load balancer, you can delete it. For more information, see Delete a load balancer (p. 15).

To create the load balancer

1. On the **Review** page, choose **Create**.
2. After the load balancer is created, choose **Close**.
3. On the navigation pane, under **LOAD BALANCING**, choose **Target Groups**.
4. Select the newly created target group.
5. Choose **Targets** and verify that your instances are ready. If the status of an instance is `initial`, it's probably because the instance is still in the process of being registered, or it has not passed the minimum number of health checks to be considered healthy. After the status of at least one instance is healthy, you can test your load balancer.

Tags for your 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 (p. 29).

To delete n 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.

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. You cannot delete the 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.

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.

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 your target groups (p. 21).

Contents

Routing configuration

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

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

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

The following are the target group attributes:

`deregistration_delay.timeout_seconds`

The amount of time for Elastic Load Balancing 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.

Deregistration delay

When you deregister an instance, the Gateway Load Balancer stops creating new connections to the instance. The Gateway Load Balancer uses connection draining to ensure that in-flight traffic completes on the existing connections. If the deregistered instance stays healthy and an existing connection is not idle, the Gateway Load Balancer can continue to send traffic to the instance. To ensure that existing connections are closed, you can verify that the instance is unhealthy before you deregister it, or you can periodically close client connections.

The initial state of a deregistering target is `draining`. By default, the Gateway Load Balancer changes the state of a deregistering target to `unused` after 300 seconds. To change the amount of time that the Gateway Load Balancer waits before changing the state of a deregistering target to `unused`, update the deregistration delay value. We recommend that you specify a value of at least 120 seconds to ensure that requests are completed.

New console

To update the deregistration delay value using the new 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**.

Old console

To update the deregistration delay value using the old 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. Select the target group.
4. Choose **Description**, **Edit attributes**.
5. Change the value of **Deregistration delay** as needed, and then choose **Save**.

To update the deregistration delay value using the AWS CLI

Use the modify-target-group-attributes command.

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 (p. 17).

You can add or remove targets from your target group at any time. For more information, see Register targets (p. 25). You can also modify the health check settings for your target group. For more information, see Modify health check settings (p. 25).

New console

To create a target group using the new 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. For **Choose a target type**, select **Instances** to register targets by instance ID or **IP addresses** to register targets by IP address.
5. 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.

6. For **Protocol**, use **GENEVE**. With the GENEVE protocol, **Port** must be 6081.
7. For **VPC**, select a virtual private cloud (VPC).
8. (Optional) In the **Health checks** section, modify the default settings as needed.
9. (Optional) Expand the **Tags** section and add one or more tags. To add a tag, choose **Add tag** and enter the tag key and tag value.
10. Choose **Next**.
11. (Optional) 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**.
12. Choose **Create target group**.

Old console

To create a target group using the old 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. 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.
5. For **Protocol**, use **GENEVE**. With the GENEVE protocol, **Port** must be 6081.
6. For **Target type**, select `instance` to specify targets by instance ID or `ip` to specify targets by IP address.
7. For **VPC**, select a virtual private cloud (VPC).
8. (Optional) For **Health check settings** and **Advanced health check settings**, modify the default settings as needed. Choose **Create**.
9. (Optional) Add one or more tags as follows:
    a. Select the newly created target group.
    b. Choose **Tags**, **Add/Edit Tags**.
    c. On the **Add/Edit Tags** page, for each tag that you add, choose **Create Tag** and then specify the tag key and tag value. When you have finished adding tags, choose **Save**.
10. (Optional) To add targets to the target group, see Register targets with your target group (p. 25).

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 your 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. Each node checks the health of each target, using the health check settings for the target group with which the target is registered. After each health check is completed, the node 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 ping 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**.
HealthyThresholdCount	The number of consecutive successful health checks required before considering an unhealthy target healthy. The range is 2 to 10. The default is 3.
UnhealthyThresholdCount	The number of consecutive failed health checks required before considering a target unhealthy. The range is 2 to 10. The default is 3.
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. Related reason codes: `Elb.RegistrationInProgress` | `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.FailedHealthChecks`
`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.DeregistrationInProgress`
`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

Reason code	Description
`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

Check the health of your targets

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

New console

To check the health of your targets using the new 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.

Old console

To check the health of your targets using the old 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.
4. Choose **Targets**, and view the status of each target in the **Status** column. If the status is any value other than `Healthy`, the console displays 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.

New console

To modify health check settings for a target group using the new 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**.

Old console

To modify health check settings for a target group using the old 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.
4. Choose **Health checks**, **Edit**.
5. On the **Edit target group** page, modify the settings as needed, and then choose **Save**.

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

Use the modify-target-group command.

Register targets with your target group

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 your target groups (p. 21).

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.

When you deregister a target, Elastic Load Balancing 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 (p. 19).

Target security groups

When you register EC2 instances as targets, you must ensure that the security groups for these instances allow 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 or deregister targets

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 (p. 18).

Requirements

- You cannot register instances by instance ID if they are in a VPC that is peering to the load balancer VPC (same Region or different Region). You can register these instances by IP address.

Contents

Register or deregister targets by instance ID

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

New console

To register or deregister targets by instance ID using the new 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. To register instances, choose **Register targets**. Select one or more instances, and then choose **Include as pending below**. When you are finished adding instances, choose **Register pending targets**.
6. To deregister instances, select the instance and then choose **Deregister**.

Old console

To register or deregister targets by instance ID using the old 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.

4. Choose **Targets**, **Edit**.
5. (Optional) For **Registered instances**, select any instances to be deregistered and choose **Remove**.
6. (Optional) For **Instances**, select any running instances to be registered and then choose **Add to registered**.
7. Choose **Save**.

Register or deregister 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)

New console

To register or deregister targets by IP address using the new 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. Choose the **Targets** tab.
5. To register IP addresses, choose **Register targets**. For each IP address, select the network, Availability Zone, IP address, and port, and then choose **Include as pending below**. When you are finished specifying addresses, choose **Register pending targets**.
6. To deregister IP addresses, select the IP addresses and then choose **Deregister**. If you have many registered IP addresses, you might find it helpful to add a filter or change the sort order.

Old console

To register or deregister targets by IP address using the old 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 **Targets**, **Edit**.
4. To register IP addresses, choose the **Register targets** icon (the plus sign) in the menu bar. For each IP address, specify the network, Availability Zone, IP address, and port, and then choose **Add to list**. When you are finished specifying addresses, choose **Register**.
5. To deregister IP addresses, choose the **Deregister targets** icon (the minus sign) in the menu bar. If you have many registered IP addresses, you might find it helpful to add a filter or change the sort order. Select the IP addresses and choose **Deregister**.
6. To leave this screen, choose the **Back to target group** icon (the back button) in the menu bar.

Register or deregister targets using the AWS CLI

Use the register-targets command to add targets and the deregister-targets command to remove targets.

Tags for your target group

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.

New console

To update the tags for a target group using the new 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**.

Old console

To update the tags for a target group using the old 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. Select the target group.
4. On the **Tags** tab, choose **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** and then enter values for **Key** and **Value**.
   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 target group using the AWS CLI

Use the add-tags and remove-tags commands.

Delete a target group

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.

New console

To delete a target group using the new 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**.

Old console

To delete a target group using the old 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**.

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.

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 (p. 30).

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 Elastic Load Balancing 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 Logging API calls for your Gateway Load Balancer using AWS CloudTrail (p. 34).

CloudWatch metrics for your Gateway Load Balancer

Elastic Load Balancing 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.

Elastic Load Balancing reports metrics to CloudWatch only when requests are flowing through the Gateway Load Balancer. If there are requests flowing, Elastic Load Balancing 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. **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 Elastic Load Balancing 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`
`NewFlowCount`	The total number of new flows (or connections) established from clients to targets in the time period.

Metric	Description
	Reporting criteria: There is a nonzero value **Statistics**: The most useful statistic is `Sum`. **Dimensions** • `LoadBalancer` • `AvailabilityZone`, `LoadBalancer`
`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`
`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).
`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 Amazon EC2 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"
}
```

Logging API calls for your Gateway Load Balancer using AWS CloudTrail

Elastic Load Balancing is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Elastic Load Balancing. CloudTrail captures all API calls for Elastic Load Balancing as events. The calls captured include calls from the AWS Management Console and code calls to the Elastic Load Balancing API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Elastic Load Balancing. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Elastic Load Balancing, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Elastic Load Balancing information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Elastic Load Balancing, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail event history.

For an ongoing record of events in your AWS account, including events for Elastic Load Balancing, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple Regions and Receiving CloudTrail log files from multiple accounts

All Elastic Load Balancing actions for Gateway Load Balancers are logged by CloudTrail and are documented in the Elastic Load Balancing API Reference version 2015-12-01. For example, calls to the `CreateLoadBalancer` and `DeleteLoadBalancer` actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity element.

Understanding Elastic Load Balancing log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The log files include events for all AWS API calls for your AWS account, not just Elastic Load Balancing API calls. You can locate calls to the Elastic Load Balancing API by checking for `eventSource` elements with the value `elasticloadbalancing.amazonaws.com`. To view a record for a specific action, such as `CreateLoadBalancer`, check for `eventName` elements with the action name.

The following are example CloudTrail log records for Elastic Load Balancing for a user who created a Gateway Load Balancer and then deleted it using the AWS CLI. You can identify the CLI using the `userAgent` elements. You can identify the requested API calls using the `eventName` elements. Information about the user (`Alice`) can be found in the `userIdentity` element.

Example Example: CreateLoadBalancer

```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2020-12-11T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "CreateLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "subnets": ["subnet-8360a9e7","subnet-b7d581c0"],
        "name": "my-load-balancer",
        "type": "gateway"
    },
    "responseElements": {
        "loadBalancers":[{
            "type": "gateway",
            "loadBalancerName": "my-load-balancer",
```

```
            "vpcId": "vpc-3ac0fb5f",
            "state": {"code":"provisioning"},
            "availabilityZones": [
               {"subnetId":"subnet-8360a9e7","zoneName":"us-west-2a"},
               {"subnetId":"subnet-b7d581c0","zoneName":"us-west-2b"}
            ],
            "createdTime": "Dec 11, 2020 5:23:50 PM",
            "loadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/gateway/my-load-balancer/ffcddace1759e1d0",
        }]
    },
    "requestID": "b9960276-b9b2-11e3-8a13-f1ef1EXAMPLE",
    "eventID": "6f4ab5bd-2daa-4d00-be14-d92efEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-12-01",
    "recipientAccountId": "123456789012"
}
```


Example Example: DeleteLoadBalancer


```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2020-12-12T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "DeleteLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "loadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/gateway/my-load-balancer/ffcddace1759e1d0"
    },
    "responseElements": null,
    "requestID": "349598b3-000e-11e6-a82b-298133eEXAMPLE",
    "eventID": "75e81c95-4012-421f-a0cf-babdaEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-12-01",
    "recipientAccountId": "123456789012"
}
```

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 view the quotas for your Gateway Load Balancers, open the Service Quotas console. In the navigation pane, choose **AWS services** and select **Elastic Load Balancing**.

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, use the limit increase form.

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

Load balancer

- Gateway Load Balancers per Region: 20
- Gateway Load Balancers per VPC: 10

Target group

- Target groups with GENEVE protocol: 100
- Targets per Availability Zone per target group with GENEVE protocol: 300

Document history for Gateway Load Balancers

The following table describes the releases for Gateway Load Balancers.

update-history-change	update-history-description	update-history-date
Initial release (p. 38)	This release of Elastic Load Balancing introduces Gateway Load Balancers.	November 10, 2020