aws

Developer Guide

AWS Global Accelerator

AWS Global Accelerator: Developer Guide

Table of Contents

What is AWS Global Accelerator? 1
Components 2
AWS Regions 5
How it works 7
Overview of how it works 9
Types of accelerators 10
Idle timeout 11
Global static IP addresses 11
Health checks 12
Traffic dials and endpoint weights 13
ICMP response messages 14
IP address ranges 15
Use cases 16
Speed Comparison Tool 17
How to get started 18
Tagging 19
Tagging support in Global Accelerator 20
Adding, editing, and deleting tags in Global Accelerator 20
Pricing 21
Getting started 22
Create a standard accelerator 22
Before you begin 23
Step 1: Create a standard accelerator 23
Step 2: Add listeners 24
Step 3: Add endpoint groups 24
Step 4: Add endpoints 25
Step 5: Test your accelerator 26
Step 6 (optional): Delete your accelerator 26
Create a custom routing accelerator 27
Before you begin 28
Step 1: Create a custom routing accelerator 28
Step 2: Add listeners 29
Step 3: Add endpoint groups 29
Step 4: Add VPC subnet endpoints 30

Step 5 (optional): Delete your accelerator 31
API actions 33
Working with standard accelerators 37
Standard accelerators 38
Create accelerator 39
Update accelerator 40
Delete accelerator 41
View accelerators 42
Integrate Global Accelerator with load balancer creation 42
Compare global and regional addresses 44
Listeners for standard accelerators 45
Add listener 45
Edit listener 46
Remove listener 47
How client affinity works 47
Endpoint groups for standard accelerators 48
Add endpoint group 49
Edit endpoint group 50
Remove endpoint group 50
Adjust traffic flow with traffic dials 51
Override listener ports 52
Ensure health check access 54
Endpoints for standard accelerators 56
Endpoint requirements 57
Add endpoint 59
Edit endpoint 61
Remove endpoint 62
How endpoint weights work 62
Failover for unhealthy endpoints 63
Avoid TCP connection time delays 64
Working with custom routing accelerators 67
How custom routing accelerators work 68
Custom routing example 69
Custom routing guidelines 72
Custom routing accelerators 75
Create a custom routing accelerator 76

Edit a custom routing accelerator ... 77
View custom routing accelerators ... 77
Delete a custom routing accelerator ... 78
Listeners for custom routing accelerators ... 79
Add listener ... 79
Edit listener ... 80
Remove listener ... 81
Endpoint groups for custom routing accelerators ... 81
Add endpoint group ... 82
Edit endpoint group ... 83
Remove endpoint group ... 84
VPC subnet endpoints ... 84
Add an Amazon VPC subnet endpoint ... 85
Edit an Amazon VPC subnet endpoint ... 87
Remove an Amazon VPC subnet endpoint ... 88
Configure cross-account access ... 89
How cross-account works ... 90
Work with cross-account attachments ... 90
Create cross-account attachments ... 91
Edit cross-account attachments ... 91
Delete cross-account attachments ... 93
Work with cross-account resources ... 93
Add cross-account BYOIP addresses ... 94
Add cross-account endpoints ... 95
Remove cross-account endpoints ... 96
Identify cross-account resources ... 96
Owner: Identify cross-account resources ... 96
Principal: Identify cross-account resources ... 97
Responsibilities and permissions ... 99
Permissions for resource owners ... 99
Permissions for principals ... 99
Billing costs ... 100
Quotas ... 100
DNS addressing and custom domains ... 101
Support for DNS addressing ... 101
Route custom domain traffic to your accelerator ... 102

Bring your own IP addresses ... 103
Requirements ... 104
IP address range authorization ... 105
Provision the address range ... 108
Advertise the address range ... 109
Deprovision the address range ... 111
Use your BYOIP address with an accelerator ... 111
Update an IP address ... 112
Preserve client IP addresses ... 115
Guidelines and restrictions ... 116
Requirements for client IP address preservation ... 118
How the client IP address is preserved ... 120
Benefits of client IP address preservation ... 121
Best practices for ENIs and security ... 122
Transition endpoints ... 124
Transitioning endpoints ... 125
Logging and monitoring ... 128
CloudWatch monitoring ... 129
Global Accelerator metrics ... 130
Metric dimensions for accelerators ... 139
Troubleshooting Global Accelerator TCP reset issues ... 141
Statistics for Global Accelerator metrics ... 142
View CloudWatch metrics for your accelerators ... 143
Flow logs ... 144
Enable flow logs ... 146
Process flow log records ... 146
Publishing to Amazon S3 ... 146
Timing of log files ... 151
Flow log record syntax ... 152
CloudTrail logging ... 155
Global Accelerator information in CloudTrail ... 155
Viewing Global Accelerator events in event history ... 156
Understanding Global Accelerator log file entries ... 156
Security ... 165
Identity and Access Management ... 165
Audience ... 166

Authenticating with identities ... 167
Managing access using policies ... 170
How Global Accelerator works with IAM ... 173
Identity-based policy examples ... 179
Service-linked role ... 184
AWS managed policies ... 187
Tag-based policies ... 190
Troubleshooting ... 192
Secure VPC connections ... 194
Logging and monitoring ... 195
Compliance validation ... 196
Resilience ... 197
Infrastructure security ... 197
Quotas ... 199
General quotas ... 199
Quotas for endpoints per endpoint group ... 200
Related quotas ... 201
Related information ... 203
API Reference and product information for AWS Global Accelerator ... 203
Getting support ... 203
Tips from the AWS blog website ... 204
Document history ... 205

What is AWS Global Accelerator?

AWS Global Accelerator is a service in which you create *accelerators* to improve the performance of your applications for local and global users. Depending on the type of accelerator you choose, you can gain additional benefits:

- With a standard accelerator, you can improve availability of your internet applications that are used by a global audience. With a standard accelerator, Global Accelerator directs traffic over the AWS global network to endpoints in the nearest Region to the client.
- With a custom routing accelerator, you can map one or more users to a specific destination among many destinations.

Global Accelerator is a global service that supports endpoints in multiple AWS Regions. To determine if Global Accelerator or other services are currently supported in a specific AWS Region, see the AWS Regional Services List.

By default, Global Accelerator provides you with static IP addresses that you associate with your accelerator. The static IP addresses are anycast from the AWS edge network. For IPv4, Global Accelerator provides two static IPv4 addresses. For dual-stack, Global Accelerator provides a total of four addresses: two static IPv4 addresses and two static IPv6 addresses. For IPv4, instead of using the addresses that Global Accelerator provides, you can configure these entry points to be IPv4 addresses from your own IP address ranges that you bring to Global Accelerator (BYOIP).

Important

The static IP addresses remain assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you *delete* an accelerator, you lose the static IP addresses that are assigned to it, so you can no longer route traffic by using them. You can use IAM policies, like tag-based permissions with Global Accelerator, to limit the users who have permissions to delete an accelerator. For more information, see ABAC with Global Accelerator.

For standard accelerators, Global Accelerator uses the AWS global network to route traffic to the optimal regional endpoint based on health, client location, and policies that you configure, which increases the availability of your applications. Endpoints for standard accelerators can be Network

Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses that are located in one AWS Region or multiple Regions.

The service reacts instantly to changes in health or configuration to ensure that internet traffic from clients is always directed to healthy endpoints. Global Accelerator also respects traffic redirection or blocking by several other AWS services:

- Global Accelerator respects ARC traffic redirection for supported endpoints, to reroute traffic from a potentially impaired Availability Zone with a zonal shift or zonal autoshift. For more information, see Multi-AZ recovery in Amazon Application Recovery Controller (ARC).
- Global Accelerator respects VPC Block Public Access, including both modes (bidirectional and ingress-only). When the feature is enabled for an AWS account, Global Accelerator does not serve traffic in the Region for the VPC, except for excluded VPCs and subnets, if any. Note, however, that egress-only exclusions are still blocked for excluded VPCs and subnets. VPC and subnet exclusions must allow ingress traffic. For more information, see Block public access to VPCs and subnets.

Custom routing accelerators only support Amazon VPC (VPC) subnet endpoint types and route traffic to private IP addresses in that subnet.

Contents

- AWS Global Accelerator components
- AWS Region availability for AWS Global Accelerator
- How AWS Global Accelerator works
- Location and IP address ranges of Global Accelerator Edge servers
- Understanding AWS Global Accelerator use cases
- AWS Global Accelerator Speed Comparison Tool
- How to get started with AWS Global Accelerator
- Tagging in AWS Global Accelerator
- Pricing for AWS Global Accelerator

AWS Global Accelerator components

AWS Global Accelerator includes the following components:

Static IP addresses

By default, Global Accelerator provides you with static IP addresses that you associate with your accelerator. The static IP addresses are anycast from the AWS edge network. For IPv4, Global Accelerator provides two static IPv4 addresses. For dual-stack, Global Accelerator provides a total of four addresses: two static IPv4 addresses and two static IPv6 addresses. If you bring your own IP address range to AWS (BYOIP) to use with Global Accelerator (IPv4 only), you can instead assign IPv4 addresses from your own pool to use with your accelerator. For more information, see Bring your own IP addresses (BYOIP) in Global Accelerator.

The IP addresses serve as single fixed entry points for your clients. If you already have Elastic Load Balancing load balancers, Amazon EC2 instances, or Elastic IP address resources set up for your applications, you can easily add those to a standard accelerator in Global Accelerator. This allows Global Accelerator to use static IP addresses to access the resources. If you'd like to access an API Gateway by using Global Accelerator static IP addresses, see the following blog post for more information: Accessing an Amazon API Gateway via static IP addresses provided by AWS Global Accelerator.

The static IP addresses remain assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you *delete* an accelerator, you lose the static IP addresses that are assigned to it, so you can no longer route traffic by using them. You can use IAM policies, such as tag-based permissions, with Global Accelerator to limit the users who have permissions to delete an accelerator. For more information, see ABAC with Global Accelerator.

Accelerator

An accelerator directs traffic to endpoints over the AWS global network to improve the performance of your internet applications. Each accelerator includes one or more listeners.

There are two types of accelerators:

- A *standard* accelerator directs traffic to the optimal AWS endpoint based on several factors, including the user's location, the health of the endpoint, and the endpoint weights that you configure. This improves the availability and performance of your applications. Endpoints can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses.
- A *custom routing* accelerator lets you deterministically route multiple users to a specific EC2 destination behind your accelerator, as is required for some use cases. You do this by

directing users to a unique IP address and port on your accelerator, which Global Accelerator has mapped to the destination. Note that custom routing accelerators do not support dual-stack for IP addresses.

For more information, see Types of accelerators.

DNS name

Global Accelerator assigns each accelerator a default Domain Name System (DNS) name, similar to `a1234567890abcdef.awsglobalaccelerator.com`, that points to the static IP addresses that Global Accelerator assigns to you or that you choose from your own IP address range. If you have a dual-stack accelerator, Global Accelerator also assigns you a dual-stack DNS name, similar to `a1234567890abcdef.dualstack.awsglobalaccelerator.com` that points to the four static IP addresses for your dual-stack accelerator.

Depending on the use case, you can use your accelerator's static IP addresses or DNS name to route traffic to your accelerator, or set up DNS records to route traffic using your own custom domain name. For more information, see Support for DNS addressing in AWS Global Accelerator.

Network zone

Similar to an AWS Availability Zone, a network zone is an isolated unit with its own set of physical infrastructure. When you create an accelerator, Global Accelerator provides you with a set of static IP addresses: two static IPv4 addresses for an accelerator with an IPv4 IP address type or four static IP addresses for a dual-stack accelerator (two IPv4 addresses and two IPv6 addresses). Global Accelerator serves one static IP address per network zone from a unique IP subnet for each IP address family. If one address from a network zone becomes unavailable, due to IP address blocking by certain client networks or network disruptions, client applications can retry on the healthy static IP address from the other isolated network zone.

Listener

A listener processes inbound connections from clients to Global Accelerator, based on the port (or port range) and protocol (or protocols) that you configure. A listener can be configured for TCP, UDP, or both TCP and UDP protocols. Each listener has one or more endpoint groups associated with it, and traffic is forwarded to endpoints in one of the groups. You associate endpoint groups with listeners by specifying the Regions that you want to distribute traffic to. With a standard accelerator, traffic is distributed to optimal endpoints within the endpoint groups associated with a listener.

Endpoint group

Each endpoint group is associated with a specific AWS Region. Endpoint groups include one or more endpoints in the Region. With a standard accelerator, you can increase or reduce the percentage of traffic that would be otherwise directed to an endpoint group by adjusting a setting called a *traffic dial*. The traffic dial lets you easily do performance testing or blue/green deployment testing, for example, for new releases across different AWS Regions.

Endpoint

An endpoint is the resource that Global Accelerator directs traffic to.

Endpoints for standard accelerators can be Network Load Balancers, Application Load Balancers, EC2 instances, or Elastic IP addresses. An Application Load Balancer endpoint can be an internet-facing or internal. Traffic for standard accelerators is routed to endpoints based on the health of the endpoint along with configuration options that you choose, such as endpoint weights. For each endpoint, you can configure weights, which are numbers that you can use to specify the proportion of traffic to route to each one. This can be useful, for example, to do performance testing within a Region.

Endpoints for custom routing accelerators are Amazon VPC (VPC) subnets with one or many Amazon EC2 instances that are the destinations for traffic.

AWS Region availability for AWS Global Accelerator

For detailed information about Regional support and service endpoints for AWS Global Accelerator, see AWS Global Accelerator endpoints and quotas in the *Amazon Web Services General Reference*.

> **Note**
>
> AWS Global Accelerator is a global service. However, you must specify the US West (Oregon) Region (that is, specify the parameter `--region us-west-2`) in Regional Global Accelerator AWS CLI commands. That is, when you create resources, such as accelerators.

Global Accelerator is currently available in the following AWS Regions. Availability Zone (AZ) exceptions are noted. Adding endpoints in AWS Local Zones is not supported.

Region Name	Region
US East (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1 (except AZ usw1-az2)
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Hyderabad)	ap-south-2
Asia Pacific (Jakarta)	ap-southeast-3
Asia Pacific (Melbourne)	ap-southeast-4
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Malaysia)	ap-southeast-5
Asia Pacific (Tokyo)	ap-northeast-1 (except AZ apne1-az3)
Asia Pacific (Seoul)	ap-northeast-2
Canada (Central)	ca-central-1 (except AZ cac1-az3)
Canada West (Calgary)	ca-west-1
Europe (Frankfurt)	eu-central-1

Region Name	Region
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Milan)	eu-south-1
Europe (Paris)	eu-west-3
Europe (Spain)	eu-south-2
Europe (Stockholm)	eu-north-1
Europe (Zurich)	eu-central-2
Israel (Tel Aviv)	il-central-1
Mexico (Central)	mx-central-1
Middle East (Bahrain)	me-south-1
Middle East (UAE)	me-central-1
South America (São Paulo)	sa-east-1

How AWS Global Accelerator works

The static IP addresses provided by AWS Global Accelerator serve as single fixed entry points for your clients. When you set up your accelerator with Global Accelerator, you associate the static IP addresses to regional endpoints in one or more AWS Regions. For standard accelerators, the endpoints are Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses. For custom routing accelerators, endpoints are Amazon VPC (VPC) subnets with one or more EC2 instances. The static IP addresses accept incoming traffic onto the AWS global network from the edge location that is closest to your users.

> **Note**
>
> If you bring your own IP address range to AWS (BYOIP) to use with Global Accelerator, you can instead assign static IP addresses from your own pool to use with your accelerator. For more information, see Bring your own IP addresses (BYOIP) in Global Accelerator.

From the edge location, traffic for your application is routed based on the type of accelerator that you configure.

- For standard accelerators, traffic is routed to the optimal AWS endpoint based on several factors, including the user's location, the health of the endpoint, and the endpoint weights that you configure.
- For custom routing accelerators, each client is routed to a specific Amazon EC2 instance and port in a VPC subnet, based on the external static IP address and listener port that you provide.

Be aware of the following when you use Global Accelerator:

- **Overriding endpoint weights:** In specific, limited scenarios, Global Accelerator overrides the endpoint weights that you set, to help ensure availability. When Global Accelerator is load balancing traffic across endpoints in an endpoint group, it must, in certain circumstances, choose between preserving availability for client traffic and abiding by endpoint weights. For example, with accelerators where the client IP address is preserved, Global Accelerator might need to override an endpoint weight setting to help avoid connection collisions.
- **Security groups and rules:** When you add an accelerator, security groups and AWS WAF rules that you have already configured continue to work as they did before you added the accelerator.
- **IP fragmentation:** IP packets that are too large to fit into a standard Ethernet frame (1500+ bytes) when transmitted across the internet or other large networks are fragmented by intermediate routers and sent individually. The TCP protocol does not require IP fragmentation because clients and endpoints automatically negotiate a smaller Maximum Segment Size (MSS). However, the UDP protocol requires IP fragmentation. When packets are fragmented, Global Accelerator forwards UDP fragments to the configured endpoint, which reassembles the original IP packet. Global Accelerator drops TCP fragments at the edge, because they are not supported by the AWS network.

Topics

- Overview of how AWS Global Accelerator works
- Types of accelerators
- Understanding idle timeout in AWS Global Accelerator
- Using static IP addresses in AWS Global Accelerator
- How Global Accelerator uses health checks
- How you can manage traffic flow with traffic dials and endpoint weights
- ICMP response messages and AWS Global Accelerator

Overview of how AWS Global Accelerator works

Traffic travels over the well-monitored, congestion-free, redundant AWS global network to the endpoint. By maximizing the time that traffic is on the AWS network, Global Accelerator ensures that traffic is always routed over the optimum network path. Global Accelerator terminates TCP connections from clients at AWS edge locations and, almost concurrently, establishes a new TCP connection with your endpoints. This gives clients faster response times (lower latency) and increased throughput.

Global Accelerator always preserves client IP addresses for endpoints on custom routing accelerators. With standard accelerators, you have the option to preserve and access the client IP address for some endpoint types. For detailed information about the endpoint types and configurations that Global Accelerator supports, including client IP address preservation support, see Requirements for resources you add as accelerator endpoints.

With standard accelerators, Global Accelerator continuously monitors the health of all endpoints, and instantly begins directing traffic for all new connections to another available endpoint when it determines that an active endpoint is unhealthy. This allows you to create a high-availability architecture for your applications on AWS. Health checks aren't used with custom routing accelerators and there is no failover, because you specify the destination to route traffic to.

If you want fine-grained control over your global traffic, you can configure *weights* for your endpoints in a standard accelerator. In addition, you can use the *traffic dial* in Global Accelerator to increase (dial up) or decrease (dial down) the percentage of traffic to a specific endpoint group, for example, for performance testing or stack upgrades.

Types of accelerators

There are two types of accelerators that you can use with AWS Global Accelerator: *standard accelerators* and *custom routing accelerators*. Both types of accelerators route traffic over the AWS global network to improve performance and stability, but they're each designed for different application needs.

Standard accelerator

By using a standard accelerator, you can improve the availability and performance of your applications running on Application Load Balancers, Network Load Balancers, or Amazon EC2 instances. With a standard accelerator, Global Accelerator routes client traffic across regional endpoints based on geo-proximity and endpoint health. It also allows customers to shift client traffic across endpoints based on controls such as traffic dials and endpoint weights. This works for a wide variety of use cases, including blue/green deployment, A/B testing, and multi-Region deployment. To see more use cases, see Understanding AWS Global Accelerator use cases.

To learn more, see Working with standard accelerators in AWS Global Accelerator.

Custom routing accelerator

Custom routing accelerators work well for scenarios where you want to use custom application logic to direct one or more users to a specific destination and port among many, while still gaining the performance benefits of Global Accelerator. One example is VoIP applications that assign multiple callers to a specific media server to start voice, video, and messaging sessions. Another example is online real-time gaming applications where you want to assign multiple players to a single session on a game server based on factors such as geographic location, player skill, and game mode.

> **Note**
>
> Custom routing accelerators support only the IPv4 IP address type.

To learn more, see Working with custom routing accelerators in AWS Global Accelerator.

Based on your specific needs, you create one of these types of accelerators to accelerate your customer traffic.

Understanding idle timeout in AWS Global Accelerator

AWS Global Accelerator sets an idle timeout period that applies to its connections. If no data has been sent or received by the time that the idle timeout period elapses, Global Accelerator closes the connection. The idle timeout periods are not customizable.

To prevent connection timeout, Global Accelerator requires that you send a packet with a minimum of one byte of data, in the ingress or egress direction, within the TCP connection timeout window. You cannot use TCP keep-alive packets to maintain an open connection.

The Global Accelerator idle timeout for a network connection depends on the type of connection:

- The timeout is 340 seconds for TCP connections.
- The timeout is 30 seconds for UDP connections.

Global Accelerator continues to direct traffic for established connections to an endpoint until the idle timeout is met, even if the endpoint is marked as unhealthy or if it is removed from the accelerator. Global Accelerator selects a new endpoint, if needed, only when a new connection starts or after an idle timeout.

Using static IP addresses in AWS Global Accelerator

By default, Global Accelerator provides you with static IP addresses that are associated with your accelerator. You use the static IP addresses that Global Accelerator assigns to your accelerator—or that you specify from your own IP address pool, for standard accelerators—to route internet traffic to the AWS global network close to where your users are, regardless of their location. For standard accelerators, you associate the addresses with Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses that run in a single AWS Region or multiple Regions. For custom routing accelerators, you direct traffic to EC2 destinations in VPC subnets in one or more Regions. Routing traffic through the AWS global network improves availability and performance because traffic doesn't have to take multiple hops over the public internet. Using static IP addresses also lets you distribute incoming application traffic across multiple endpoint resources in multiple AWS Regions.

In addition, using static IP addresses makes it easier to add your application to more Regions or to migrate applications between Regions. Using fixed IP addresses means that users have a consistent way to connect to your application as you make changes.

If you like, you can associate your own custom domain name with the static IP addresses for your accelerator. For more information, see Route custom domain traffic to your accelerator.

The static IP addresses are anycast from the AWS edge network.

For IPv4, Global Accelerator provides two static IPv4 addresses. For dual-stack, Global Accelerator provides a total of four addresses: two static IPv4 addresses and two static IPv6 addresses. If you bring your own IP address range to AWS (BYOIP) to use with Global Accelerator (IPv4 only), you can instead assign IPv4 addresses from your own pool to use with your accelerator. For more information, see Bring your own IP addresses (BYOIP) in Global Accelerator.

For accelerators with dual-stack, Global Accelerator allocates the IPv6 addresses from the same two /64 CIDR prefixes. This can help simplify steps for allow-listing and setting ACL controls.

You can add IPv4-only endpoints to standard accelerators that are configured for IPv4 IP address types, but accelerators that you configure as dual-stack require that you add only endpoints that also support dual-stack. For information about endpoints that are supported for dual-stack accelerators, see Requirements for resources you add as accelerator endpoints.

Global Accelerator provides the static IP addresses for you from the Amazon pool of IP addresses, unless you bring your own IP address range to AWS, and then specify the static IP addresses from that pool. (For more information, see Bring your own IP addresses (BYOIP) in Global Accelerator.) To create an accelerator on the console, the first step is to prompt Global Accelerator to provision the static IP addresses by entering a name for your accelerator or choose your own static IP addresses. To see the steps for creating an accelerator, see Getting started with AWS Global Accelerator.

The static IP addresses remain assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you *delete* an accelerator, you lose the static IP addresses that are assigned to it, so you can no longer route traffic by using them. You can use IAM policies like tag-based permissions with Global Accelerator to limit the users who have permissions to delete an accelerator. For more information, see ABAC with Global Accelerator.

How Global Accelerator uses health checks

For standard accelerators, AWS Global Accelerator automatically checks the health of the endpoints that are associated with your static IP addresses, and then directs user traffic only to healthy endpoints.

Global Accelerator includes default health checks that are run automatically, but you can configure the timing for the checks and other options. If you've configured custom health check settings, Global Accelerator uses those settings in specific ways, depending on your configuration. You configure those settings in Global Accelerator for Amazon EC2 instance or Elastic IP address endpoints or by configuring settings on the Elastic Load Balancing console for Network Load Balancers or Application Load Balancers. For more information, see Ensure health check access for your accelerator.

When you add an endpoint to a standard accelerator, it must pass a health check to be considered healthy before traffic is directed to it. If Global Accelerator doesn't have any healthy endpoints to route traffic to in a standard accelerator, it routes requests to all endpoints.

How you can manage traffic flow with traffic dials and endpoint weights

There are two ways that you can customize how AWS Global Accelerator sends traffic to your endpoints with a standard accelerator:

- Change the traffic dial to limit the traffic for one or more endpoint groups
- Specify weights to change the proportion of traffic to the endpoints in a group

How traffic dials work

For each endpoint group in a standard accelerator, you can set a traffic dial to control the percentage of traffic that is sent to the endpoint group. The percentage is applied only to traffic that is already directed to the endpoint group, not to all listener traffic.

The traffic dial limits the portion of traffic that an endpoint group accepts, expressed as a percentage of traffic directed to that endpoint group. For example, if you set the traffic dial for an endpoint group in `us-east-1` to 50 (that is, 50%) and the accelerator directs 100 user requests to that endpoint group, only 50 requests are accepted by the group. The accelerator directs the remaining 50 requests to endpoint groups in other Regions.

For more information, see Use traffic dials to adjust traffic flow to Regions.

How weights work

For each endpoint in a standard accelerator, you can specify weights, which are numbers that change the proportion of traffic that the accelerator routes to each endpoint. This can be useful, for example, to do performance testing within a Region.

A weight is a value that determines the proportion of traffic that the accelerator directs to an endpoint. By default, the weight for an endpoint is 128—that is, half of the maximum value for a weight, 255.

The accelerator calculates the sum of the weights for the endpoints in an endpoint group, and then directs traffic to the endpoints based on the ratio of each endpoint's weight to the total. For an example of how weights work, see How endpoint weights work to manage traffic volume.

Traffic dials and weights affect how the standard accelerator serves traffic in different ways:

- You configure traffic dials for *endpoint groups*. The traffic dial lets you cut off a percentage of traffic—or all traffic—to the group, by "dialing down" traffic that the accelerator has already directed to it based on other factors, such as proximity.
- You use weights, on the other hand, to set values for *individual endpoints* within an endpoint group. Weights provide a way to divide up traffic within the endpoint group. For example, you can use weights to do performance testing for specific endpoints in a Region.

For more information about how traffic dials and weights affect failover, see How failover works for unhealthy endpoints.

ICMP response messages and AWS Global Accelerator

ICMP response messages, such as `ICMP Packet Too Big` or `Fragmentation Needed`, help to ensure availability on the internet. AWS Global Accelerator responds to ICMP echo messages (pings) at the edge for all global IP addresses. These pings are not forwarded to customers' endpoints. To accurately test performance with Global Accelerator, use a deeper protocol for your tests.

Here's a brief summary of how ICMP helps to ensure internet availability. The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The larger the MTU of a connection, the more data that can be passed in a single packet. Path MTU Discovery (PMTUD) is used to determine the *path MTU* between two devices. The path MTU is the maximum packet size that's supported on the path between the originating host and the receiving host. When there is a difference in the MTU size in the network between two hosts, packets that are bigger than the MTU get dropped, and

the receiving host that dropped the packet notifies the sender with an ICMP message. For more information, see Path MTU Discovery.

You cannot block ICMP traffic at your accelerator in Global Accelerator. Blocking all ICMP traffic would also drop ICMP messages such as `ICMPv6 Packet Too Big (PTB)` (Type 2) and `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set` (Type 3, Code 4). These messages are necessary for traffic to successfully make it back to the originating host. In turn, these dropped messages would cause TCP and protocols that are built on top of Global Accelerator to drop traffic from clients that are on networks with smaller-than-typical MTU, preventing PMTUD.

Note that for PMTUD to work, the security groups of your endpoints must also allow ICMP traffic. If you have availability issues that are specific to certain end-user networks, confirm that your endpoint security groups allow ICMP traffic.

Location and IP address ranges of Global Accelerator Edge servers

For a list of Global Accelerator edge server locations, see *Global Edge Network* on the AWS Global Accelerator features page.

AWS publishes its current IP address ranges in JSON format. To view the current ranges, download ip-ranges.json. For more information, see AWS IP address ranges in the *Amazon Web Services General Reference*.

Before you work with the `ip-ranges.json` file, first review the following information:

- To find the IP address ranges that are associated with AWS Global Accelerator Edge servers, search `ip-ranges.json` for the following string:

  ```
  "service": "GLOBALACCELERATOR"
  ```

- Global Accelerator entries that include `"region": "GLOBAL"` refer to the static IP addresses that are allocated to accelerators. If you want to filter for traffic through your accelerator that comes from points of presence (POPs) in one area, filter for entries that include a specific geographical area, such as `us-*` or `eu-*`. So, for example, if you filter for `us-*`, you will see only traffic coming through POPs in the United States (U.S.).
- Global Accelerator supports two ways of routing traffic: using client IP address preservation or using network address translation (NAT). The way that traffic is routed determines the client IP

address that AWS WAF can apply rules to. When you use client IP address preservation, AWS WAF rules target the client IP address—that is, the IP address of the clients who access your service. When you use NAT, AWS WAF rules are applied to the global IP addresses that Global Accelerator uses to route traffic.

Understanding AWS Global Accelerator use cases

Using AWS Global Accelerator can help you accomplish a variety of goals. This section lists some of them, to give you an idea how you can use Global Accelerator to meet your needs.

Scale for increased application utilization

When application usage grows, the number of IP addresses and endpoints that you need to manage also increases. Global Accelerator enables you to scale your network up or down. It lets you associate regional resources, such as load balancers and Amazon EC2 instances, to two static IPv4 addresses or, for dual-stack, to two static IPv4 addresses and two IPv6 addresses. You include these addresses on allow lists just once in your client applications, firewalls, and DNS records. With Global Accelerator, you can add or remove endpoints in AWS Regions, run blue/green deployment, and do A/B testing without having to update the IP addresses in your client applications. This is especially useful for IoT, retail, media, automotive, and healthcare use cases where you can't easily update client applications frequently.

Acceleration for latency-sensitive applications

Many applications, especially in areas such as gaming, media, mobile apps, ad-tech, and financials, require very low latency for a great user experience. To improve the user experience, Global Accelerator directs user traffic to the application endpoint that is nearest to the client, which reduces internet latency and jitter. Global Accelerator routes traffic to the closest edge location by using Anycast, and then routes it to the closest regional endpoint over the AWS global network. Global Accelerator quickly reacts to changes in network performance to improve your users' application performance.

Disaster recovery and multi-Region resiliency

You must be able to rely on your network to be available. You might be running your application across multiple AWS Regions to support disaster recovery, higher availability, lower latency, or compliance. If Global Accelerator detects that your application endpoint is failing in the primary AWS Region, it instantly triggers traffic re-routing to your application endpoint in the next available, closest AWS Region.

For more information about how Global Accelerator supports resiliency inherently and in applications that use the service, read the following blog post: Maximising application resiliency with AWS Global Accelerator.

Protect your applications

Exposing your AWS origins, such as Application Load Balancers or Amazon EC2 instances, to public internet traffic creates an opportunity for malicious attacks. Global Accelerator decreases the risk of attack by masking your origin behind two static entry points. These entry points are protected by default from Distributed Denial of Service (DDoS) attacks with AWS Shield. Global Accelerator creates a peering connection with your Amazon Virtual Private Cloud using private IP addresses, keeping connections to your internal Application Load Balancers or private EC2 instances off the public internet.

Improve performance for VoIP or online gaming applications

Using a custom routing accelerator, you can leverage the performance benefits of Global Accelerator for your VoIP or gaming applications. For example, you can use Global Accelerator for online gaming applications that assign multiple players to a single gaming session. Use Global Accelerator to reduce latency and jitter globally for applications that require custom logic to map users to specific endpoints, such as multiplayer games or VoIP calls. You can use a single accelerator to connect clients to thousands of Amazon EC2 instances running in a single or multiple AWS Regions, while retaining full control over which client is directed to which EC2 instance and port.

AWS Global Accelerator Speed Comparison Tool

You can use the AWS Global Accelerator Speed Comparison Tool to see Global Accelerator download speeds compared to direct internet downloads, across AWS Regions. This tool enables you to use your browser to see the performance difference when you transfer data using Global Accelerator. You choose a file size to download, and the tool downloads files over HTTPS/TCP from Application Load Balancers in different Regions to your browser. For each Region, you see a direct comparison of the download speeds.

To access the Speed Comparison Tool, copy the following URL into your browser:

```
https://speedtest.globalaccelerator.aws
```

> **Important**
>
> Results may differ when you run the test multiple times. Download times can vary based on factors that are external to Global Accelerator, such as the quality, capacity, and distance of the connection in the last-mile network that you're using.

How to get started with AWS Global Accelerator

You can get started with setting up AWS Global Accelerator by using the API or by using the AWS Global Accelerator console. Because Global Accelerator is a global service, it's not tied to a specific AWS Region. Note that Global Accelerator is a global service that supports endpoints in multiple AWS Regions but you must specify the US West (Oregon) Region to create or update accelerators.

To get started using Global Accelerator, you follow these general steps:

1. **Choose the type of accelerator that you want to create:** A standard accelerator or a custom routing accelerator.
2. **Configure the initial setup for Global Accelerator:** Provide a name for your accelerator, then choose the type of accelerator and the address type.
3. **Configure one or more listeners for your accelerator:** Listeners process inbound connections from clients, based on the protocol and port (or port range) that you specify.
4. **Configure regional endpoint groups for your accelerator:** You can select one or more regional endpoint groups to add to your listener. The listener routes requests to the endpoints that you've added to an endpoint group.

   For a standard accelerator, Global Accelerator monitors the health of endpoints within the group by using the health check settings that are defined for each of your endpoints. For each endpoint group in a standard accelerator, you can configure a *traffic dial* percentage to control the percentage of traffic that an endpoint group will accept. The percentage is applied only to traffic that is already directed to the endpoint group, not all listener traffic. By default, the traffic dial is set to 100% for all regional endpoint groups.

   For a custom routing accelerators, traffic is deterministically routed to a specific destination in a VPC subnet, based on the listener port that the traffic is received on.
5. **Add endpoints to endpoint groups:** The endpoints that you add depend on the type of accelerator.

- For a standard accelerator, you can add one or more regional resources, such as load balancers or EC2 instances endpoints, to each endpoint group. Next, you can decide how much traffic you want to route to each endpoint by setting endpoint weights.
- For a custom routing accelerator, you add one or more Amazon VPC (VPC) subnets with up to thousands of Amazon EC2 instance destinations.

For detailed steps about how to create a standard accelerator or a custom routing accelerator using the AWS Global Accelerator console, see Getting started with AWS Global Accelerator. To work with API operations, see Common API actions for AWS Global Accelerator and the AWS Global Accelerator API Reference.

Tagging in AWS Global Accelerator

Tags are words or phrases (metadata) that you use to identify and organize your AWS resources. You can add multiple tags to each resource, and each tag includes a key and a value that you define. For example, the key might be `environment` and the value might be `production`. You can search and filter your resources based on the tags you add. In AWS Global Accelerator, you can tag accelerators.

The following are two examples of how it can be useful to work with tags in Global Accelerator:

- Use tags to track billing information in different categories. To do this, apply tags to accelerators or other AWS resources (such as Network Load Balancers, Application Load Balancers, or Amazon EC2 instances) and activate the tags. Then AWS generates a cost allocation report as a comma-separated value (CSV file) with your usage and costs aggregated by your active tags. You can apply tags that represent business categories (such as cost centers, application names, or owners) to organize your costs across multiple services. For more information, see Using Cost Allocation Tags in the *AWS Billing User Guide*.
- Use tags to enforce tag-based permissions for accelerators. To do this, create IAM policies that specify tags and tag values to allow or disallow actions. For more information, see ABAC with Global Accelerator.

For usage conventions and links to other resources about tagging, see Tagging your AWS resources in the *AWS General Reference*. For tips on using tags, see Tagging Best Practices: AWS Resource Tagging Strategy in the *AWS Whitepapers* blog.

For the maximum number of tags that you can add to a resource in Global Accelerator, see Quotas for AWS Global Accelerator.

You can add and update tags by using the AWS console, AWS CLI, or Global Accelerator API. This chapter includes steps for working with tagging in the console. For more information about working with tags by using the AWS CLI and the Global Accelerator API, including CLI examples, see the following operations in the *AWS Global Accelerator API Reference*:

- CreateAccelerator
- CreateCrossAccountAttachment
- TagResource
- UntagResource
- ListTagsForResource

Tagging support in Global Accelerator

AWS Global Accelerator supports tagging for accelerators and cross-account attachments.

Global Accelerator supports the tag-based access control feature of AWS Identity and Access Management (IAM). For more information, see ABAC with Global Accelerator.

Adding, editing, and deleting tags in Global Accelerator

The following procedure explains how to add, edit, and delete tags for accelerators in the Global Accelerator console.

You can add or remove tags using the console, the AWS CLI, or Global Accelerator API operations. For more information, including CLI examples, see TagResource in the *AWS Global Accelerator API Reference*.

To add tags, edit, or delete tags in Global Accelerator

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. Choose the accelerator that you want to add or update tags for.
3. In the **Tags** section, you can do the following:

Add a tag

Choose **Add tag**, then enter a key and, optionally, a value for the tag.

Edit a tag

Update the text for a key, value, or both. You can also clear the value for a tag, but the key is required.

Delete a tag

Choose **Remove** on the right side of the value field.

4. Choose **Save changes**.

Pricing for AWS Global Accelerator

With AWS Global Accelerator, you are charged a *fixed hourly fee* for each accelerator that is provisioned in your account (whether it's enabled or disabled), and an *incremental charge*, in addition to standard data transfer rates, for every hour of traffic in the dominant direction that flows through the accelerator. The incremental rate depends on the AWS Region that serves the request (the source) and the AWS edge location where the responses are directed (the destination). Customers typically create one accelerator for each application, but customers with complex applications might require more accelerators.

In addition, you will incur standard public IPv4 address charges for IPv4 addresses used with your accelerators.

For details about pricing, information about pricing by source and destination Regions, and a pricing example, see AWS Global Accelerator pricing.

Getting started with AWS Global Accelerator

To help you get started with AWS Global Accelerator, this chapter provides tutorials for setting up a standard accelerator and a custom routing accelerator.

To learn more about the two types of accelerators that you can create in Global Accelerator, see Working with standard accelerators in AWS Global Accelerator and Working with custom routing accelerators in AWS Global Accelerator.

The tutorials provide steps that primarily use AWS Management Console. Note that when you set up a custom routing accelerator, you must use the API for certain configuration steps.

You can also use Global Accelerator API operations with the AWS Command Line Interface (AWS CLI) or AWS SDKs to create and customize your accelerators. The following are resources for working with the Global Accelerator APIs.

- For a list of API operations, see Common API actions for AWS Global Accelerator.
- For detailed information about working with AWS Global Accelerator API operations, see the AWS Global Accelerator API Reference.

Global Accelerator is a global service that supports endpoints in multiple AWS Regions. Supported Regions are listed in the AWS Region Table.

Contents

- Getting started with a standard accelerator
- Getting started with a custom routing accelerator

Getting started with a standard accelerator

This section provides steps for creating a standard accelerator, which routes traffic to an optimal endpoint.

Tasks

- Before you begin
- Step 1: Create a standard accelerator

- Step 2: Add listeners
- Step 3: Add endpoint groups
- Step 4: Add endpoints
- Step 5: Test your accelerator
- Step 6 (optional): Delete your accelerator

Before you begin

Before you create an accelerator, create at least one resource that you can add as an endpoint to direct traffic to. For example, create one of the following:

- Launch at least one Amazon EC2 instance to add as an endpoint. For more information, see Create your EC2 resources and launch your EC2 instance in the *Amazon EC2 User Guide*.
- Optionally, create one or more Network Load Balancers or Application Load Balancers that include EC2 instances. For more information, see Create a Network Load Balancer in the *User Guide for Network Load Balancers*.

When you create a resource to add to Global Accelerator, be aware of the following:

- When you add an internal Application Load Balancer or an EC2 instance endpoint in Global Accelerator, you enable internet traffic to flow directly to and from the endpoint in virtual private clouds (VPCs) by targeting it in a private subnet. The VPC that contains the load balancer or EC2 instance must have an internet gateway attached to it, to indicate that the VPC accepts internet traffic. For more information, see Secure VPC connections in AWS Global Accelerator.
- Global Accelerator requires your router and firewall rules to allow inbound traffic from the IP addresses associated with Amazon Route 53 health checkers to complete health checks for EC2 instance or Elastic IP address endpoints. You can find information about the IP address ranges associated with Route 53 health checkers in IP address ranges of Amazon Route 53 servers in the *Amazon Route 53 Developer Guide*.

Step 1: Create a standard accelerator

When you create a standard accelerator, you can choose IPv4 or dual-stack for the static IP addresses Global Accelerator assigns to your accelerator. Dual-stack supports both IPv4 and IPv6 IP addresses.

To create an accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. Choose **Create accelerator**.
3. Provide a name for your accelerator.
4. For **Accelerator type**, select **Standard**.
5. For **IP address type**, select **IPv4** or **Dual-stack**.
6. Optionally, add one or more tags to help you identify your Global Accelerator resources.
7. Choose **Next**.

Step 2: Add listeners

Create a listener to process inbound connections from your users to Global Accelerator.

To create a listener

1. On the **Add listener** page, enter the ports or port ranges that you want to associate with the listener. Listeners support ports 1-65535.
2. Choose the protocol or protocols for the ports that you entered.
3. Optionally, choose to enable client affinity. Client affinity for a listener means that Global Accelerator ensures that connections from a specific source (client) IP address are always routed to the same endpoint. To enable this behavior, in the dropdown list, choose **Source IP**.

   The default is **None**, which means that client affinity is not enabled and Global Accelerator distributes traffic equally between the endpoints in the endpoint groups for the listener.

   For more information, see How client affinity works in Global Accelerator.
4. Optionally, choose **Add listener** to add an additional listener.
5. When you're finished adding listeners, choose **Next**.

Step 3: Add endpoint groups

Add one or more endpoint groups, each of which is associated with a specific AWS Region.

To add an endpoint group

1. On the **Add endpoint groups** page, in the section for a listener, choose a **Region** from the dropdown list.
2. Optionally, for **Traffic dial**, enter a number from 0 to 100 to set a percentage of traffic for this endpoint group. The percentage is applied only to the traffic already directed to this endpoint group, not all listener traffic. By default, the traffic dial for an endpoint group is set to 100 (that is, 100%).
3. Optionally, for custom health check values, choose **Configure health checks**. When you configure health check settings, Global Accelerator uses the settings for health checks for EC2 instance and Elastic IP address endpoints. For Network Load Balancer and Application Load Balancer endpoints, Global Accelerator uses the health check settings that you've already configured for the load balancers themselves. For more information, see Ensure health check access for your accelerator.
4. Optionally, choose **Add endpoint group** to add additional endpoint groups for this listener or other listeners.
5. Choose **Next**.

Step 4: Add endpoints

Add one or more endpoints that are associated with specific endpoint groups. This step isn't required, but no traffic is directed to endpoints in a Region unless the endpoints are included in an endpoint group.

To add endpoints

1. On the **Create endpoints** page, in the section for an endpoint, choose an **Endpoint**.
2. Optionally, for **Weight**, enter a number from 0 to 255 to set a weight for routing traffic to this endpoint. When you add weights to endpoints, you configure Global Accelerator to route traffic based on proportions that you specify. By default, all endpoints have a weight of 128. For more information, see How endpoint weights work to manage traffic volume.
3. Optionally, under **Preserve client IP address**, select **Preserve address**. (For some endpoint types, this option is selected and can't be cleared.) For more information, see Preserve client IP addresses in AWS Global Accelerator.
4. Optionally, choose **Add endpoint** to add more endpoints.
5. Choose **Next**.

After you choose **Next**, on the Global Accelerator dashboard you'll see a message that your accelerator is in progress. When the process is finished, the accelerator status in the dashboard is **Active**.

Step 5: Test your accelerator

Take steps to test your accelerator to make sure that traffic is being directed to your endpoints. For example, run a curl command such as the following, substituting one of your accelerator's static IP addresses, to show the AWS Regions where requests are processed. This is especially helpful if you set different weights for endpoints or adjust the traffic dial on endpoint groups.

Run a curl command like the following, substituting one of your accelerator's static IP addresses, to call the IP address 100 times and then output a count of where each request was processed.

```
for ((i=0;i<100;i++)); do  curl http://198.51.100.0/ >> output.txt; done; cat
 output.txt | sort | uniq -c ; rm output.txt;
```

If you've adjusted the traffic dial on any endpoint groups, this command can help you confirm that your accelerator is directing the correct percentages of traffic to different groups. For more information, see the detailed examples in the following blog post, Traffic management with AWS Global Accelerator.

Step 6 (optional): Delete your accelerator

If you created an accelerator as a test or if you're no longer using an accelerator, you can delete it. On the console, disable the accelerator, and then you can delete it. You don't have to remove listeners and endpoint groups from the accelerator.

To delete an accelerator by using an API operation instead of the console, you must first remove all listeners and endpoint groups that are associated with the accelerator as well as disable it. For more information, see the DeleteAccelerator operation in the *AWS Global Accelerator API Reference*.

Be aware of the following when you remove endpoints or endpoint groups, or delete an accelerator:

- When you create an accelerator, Global Accelerator provides you with a set of two static IP addresses. The IP addresses are assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you *delete* an accelerator, you lose the static IP addresses that are assigned to the accelerator, so you can no

longer route traffic by using them. As a best practice, ensure that you have permissions in place to avoid inadvertently deleting accelerators. You can use IAM policies with Global Accelerator, for example, tag-based permissions, to limit the users who have permissions to delete an accelerator. For more information, see ABAC with Global Accelerator.

- If you terminate an EC2 instance before you remove it from an endpoint group in Global Accelerator, and then you create another instance with the same private IP address, and health checks pass, Global Accelerator will route traffic to the new endpoint. If you don't want this to happen, remove the EC2 instance from the endpoint group before you terminate the instance.

To delete an accelerator

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. Choose the accelerator that you want to delete.
3. Choose **Edit**.
4. Choose **Disable accelerator**, and then choose **Save**.
5. Choose the accelerator that you want to delete.
6. Choose **Delete accelerator**.
7. In the confirmation dialog box, choose **Delete**.

Getting started with a custom routing accelerator

This section provides steps for creating a custom routing accelerator, which routes traffic deterministically to Amazon EC2 instance destinations in virtual private cloud (VPC) subnet endpoints.

Tasks

- Before you begin
- Step 1: Create a custom routing accelerator
- Step 2: Add listeners
- Step 3: Add endpoint groups
- Step 4: Add endpoints
- Step 5 (optional): Delete your accelerator

Before you begin

Before you create a custom routing accelerator, create a resource that you can add as an endpoint to direct traffic to. A custom routing accelerator endpoint must be a virtual private cloud (VPC) subnet, which can include multiple Amazon EC2 instances. For instructions for creating the resources see the following:

- Create a VPC subnet. For more information, see Create and Configure Your VPC in the *AWS Directory Service Administration Guide*.
- Optionally, launch one or more Amazon EC2 instances in your VPC. For more information, see Create your EC2 resources and launch your EC2 instance in the *Amazon EC2 User Guide*.

When you create a resource to add to Global Accelerator, be aware of the following:

- When you add an EC2 instance endpoint in Global Accelerator, you enable internet traffic to flow directly to and from the endpoint in a VPC by targeting it in a private subnet. The VPC that contains the EC2 instance must have an internet gateway attached to it, to indicate that the VPC accepts internet traffic. For more information, see Secure VPC connections in AWS Global Accelerator.

Before you create a custom routing accelerator, make sure that you review the best practices described in Guidelines and restrictions for custom routing accelerators.

Step 1: Create a custom routing accelerator

To create an accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. Provide a name for your accelerator.
3. For **Accelerator type**, select **Custom routing**.
4. Optionally, add one or more tags to help you identify your accelerator resources.
5. Choose **Next** to add listeners, endpoint groups, and VPC subnet endpoints.

Step 2: Add listeners

Create a listener to process inbound connections from your users to Global Accelerator.

The range that you specify when you create a listener defines how many listener port and destination IP address combinations that you can use with your custom routing accelerator. For maximum flexibility, we recommend that you specify a large port range. Each listener port range that you specify must include a minimum of 16 ports.

To create a listener

1. On the **Add listener** page, enter the ports or port ranges that you want to associate with the listener. Listeners support ports 1-65535.
2. Choose the protocol or protocols for the ports that you entered.
3. Optionally, choose **Add listener** to add an additional listener.
4. When you're finished adding listeners, choose **Next**.

Step 3: Add endpoint groups

Add one or more endpoint groups, each of which is associated with a specific AWS Region. For each endpoint group, specify one or more sets of port ranges and protocols. Global Accelerator uses these to direct traffic to Amazon EC2 instances in subnets in the Region.

For each port range that you provide, you also specify the protocol to use: UDP, TCP, or both UDP and TCP.

To add an endpoint group

1. On the **Add endpoint groups** page, in the section for a listener, choose a **Region**.
2. For **Ports and protocols sets**, enter port ranges and protocols for your Amazon EC2 instances.

   - Enter a **From port** and a **To port** to specify a range of ports.
   - For each port range, specify the protocol or protocols for that range.

   The port range doesn't have to be a subset of your listener port range, but there must be enough total ports in the listener port range to support the total number of ports that you specify.

3. Choose **Save**.
4. Optionally, choose **Add endpoint group** to add additional endpoint groups for this listener or other listeners.
5. Choose **Next**.

Step 4: Add VPC subnet endpoints

Add one or more virtual private cloud (VPC) subnet endpoints for this regional endpoint group. Endpoints for custom routing accelerators define the VPC subnets that can receive traffic through a custom routing accelerator. Each subnet can contain one or many Amazon EC2 instance destinations.

When you add a VPC subnet endpoint, Global Accelerator generates new port mappings that you can use to route traffic to the destination EC2 instance IP addresses in the subnet. Then you can use the Global Accelerator API to get a static list of all the port mappings for the subnet, and use the mapping to deterministically direct traffic to specific EC2 instances.

To add endpoints

1. On the **Add endpoints** page, in the section for the endpoint group that you want to add the endpoint to, choose a subnet ID for **Endpoint**.
2. Optionally, do one of the following to enable traffic to EC2 instance destinations in the subnet:

   - To allow traffic to be directed to all EC2 endpoints and ports on the subnet, select **Allow all traffic**
   - To allow traffic to specific EC2 endpoints and ports on the subnet, select **Allow traffic to specific destination socket addresses**. Then specify the IP addresses and ports or port ranges to allow. Finally, choose **Allow these destinations**.

   By default, no traffic is allowed to subnet endpoints. If you don't select an option to allow traffic, traffic is denied to all destinations in the subnet.

> **Note**
>
> If you want to enable traffic to specific EC2 instances and ports in the subnet, you can do that programmatically. For more information, see AllowCustomRoutingTraffic in the *AWS Global Accelerator API Reference*.

3. Choose **Next**.

After you choose **Next**, on the Global Accelerator, dashboard you'll see a message that your accelerator is in progress. When the process is finished, the accelerator status in the dashboard is **Active**.

Step 5 (optional): Delete your accelerator

If you created an accelerator as a test or if you're no longer using an accelerator, you can delete it. On the console, disable the accelerator, and then you can delete it. You don't have to remove listeners and endpoint groups from the accelerator.

To delete an accelerator by using an API operation instead of the console, you must first remove all listeners and endpoint groups that are associated with the accelerator as well as disable it. For more information, see the DeleteCustomRoutingAccelerator operation in the *AWS Global Accelerator API Reference*.

Be aware of the following when you delete an accelerator:

- When you create an accelerator, Global Accelerator provides you with a set of two static IP addresses. The IP addresses are assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you *delete* an accelerator, you lose the static IP addresses that are assigned to the accelerator, so you can no longer route traffic by using them. As a best practice, ensure that you have permissions in place to avoid inadvertently deleting accelerators. You can use IAM policies like tag-based permissions with Global Accelerator to limit the users who have permissions to delete an accelerator. For more information, see ABAC with Global Accelerator.

To delete an accelerator

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. Choose the accelerator that you want to delete.
3. Choose **Edit**.
4. Choose **Disable accelerator**, and then choose **Save**.
5. Choose the accelerator that you want to delete.
6. Choose **Delete accelerator**.
7. In the confirmation dialog box, choose **Delete**.

Common API actions for AWS Global Accelerator

This section lists common AWS Global Accelerator actions that you can use with Global Accelerator resources, with links to relevant documentation.

Actions to use with standard accelerators

The following table lists common Global Accelerator actions that you can use with standard accelerators, with links to relevant documentation.

Action	Using the Global Accelerator Console	Using the Global Accelerator API
Create a standard accelerator	See Getting started with a standard accelerator	See CreateAccelerator
Create a listener for a standard accelerator	See Listeners for standard accelerators in AWS Global Accelerator	See CreateListener
Create a endpoint group for a standard accelerator	See Endpoint groups for standard accelerators in AWS Global Accelerator	See CreateEndpointGroup
Update a standard accelerator	See Standard accelerators in AWS Global Accelerator	See UpdateAccelerator
Update an endpoint group	See Add a standard endpoint group	See UpdateEndpointGroup
Add an endpoint	See Add a standard endpoint	See AddEndpoints
Remove an endpoint	See Add a standard endpoint	See RemoveEndpoints
List standard accelerators	See View your accelerators	See ListAccelerator
Get all information about an accelerator	See View your accelerators	See DescribeAccelerator

Action	Using the Global Accelerator Console	Using the Global Accelerator API
Delete an accelerator	See Create accelerator	See DeleteAccelerator

Actions to use with custom routing accelerators

The following table lists common Global Accelerator actions that you can use with custom routing accelerators, with links to relevant documentation.

Action	Using the Global Accelerator Console	Using the Global Accelerator API
Create a custom routing accelerator	See Getting started with a custom routing accelerator	See CreateCustomRoutingAccelerator
Create a listener for a custom routing accelerator	See Listeners for custom routing accelerators in Global Accelerator	See CreateCustomRoutingListener
Create an endpoint group for a custom routing accelerator	See Endpoint groups for custom routing accelerators in Global Accelerator	See CreateCustomRoutingEndpointGroup
Update a custom routing accelerator	See Custom routing accelerators in AWS Global Accelerator	See UpdateCustomRoutingAccelerator
List your custom routing accelerators	See View custom routing accelerators in Global Accelerator	See ListCustomRoutingAccelerator
Get all information about a custom routing accelerator	See View custom routing accelerators in Global Accelerator	See DescribeCustomRoutingAccelerator
Delete a custom routing accelerator	See Create a custom routing accelerator in Global Accelerator	See DeleteCustomRoutingAccelerator

Action	Using the Global Accelerator Console	Using the Global Accelerator API
Get the static port mapping for a custom routing accelerator	N/A	See ListCustomRoutingPortMappings
Allow all destination traffic for a subnet in a custom routing accelerator	See Add a VPC subnet endpoint for a custom routing accelerator	See AllowCustomRoutingTraffic
Deny all destination traffic for a subnet in a custom routing accelerator	See Add a VPC subnet endpoint for a custom routing accelerator	See DenyCustomRoutingTraffic
Allow traffic to specific destinations in a custom routing accelerator	See Add a VPC subnet endpoint for a custom routing accelerator	See AllowCustomRoutingTraffic
Deny traffic to specific destinations in a custom routing accelerator	See Add a VPC subnet endpoint for a custom routing accelerator	See DenyCustomRoutingTraffic

Actions to use with cross-account support in Global Accelerator

The following table lists common Global Accelerator actions that you can use with cross-account support in Global Accelerator, with links to relevant documentation.

Action	Using the Global Accelerator Console	Using the Global Accelerator API
Create a cross-account attachment	See Create a cross-account attachment in AWS Global Accelerator	See CreateCrossAccountAttachment
Delete a cross-account attachment	See Create a cross-account attachment in AWS Global Accelerator	See DeleteCrossAccountAttachment

Action	Using the Global Accelerator Console	Using the Global Accelerator API
Describe the information in a cross-account attachment	See Identify your cross-account resources in Global Accelerator	See DescribeCrossAccountAttachment
List cross-account attachments in an account	See Identify your cross-account resources in Global Accelerator	See ListCrossAccountAttachments
Update a cross-account attachment	See Create a cross-account attachment in AWS Global Accelerator	See UpdateCrossAccountAttachment

Working with standard accelerators in AWS Global Accelerator

This chapter includes procedures and recommendations for creating standard accelerators in AWS Global Accelerator, including configuring accelerators, listeners, endpoint groups, and endpoints. With a standard accelerator, Global Accelerator chooses the closest healthy endpoint for your traffic.

If instead you want to use custom application logic to direct one or more users to a specific endpoint among many endpoints, create a custom routing accelerator. For more information, see Working with custom routing accelerators in AWS Global Accelerator.

To set up a standard accelerator, do the following:

1. Create an accelerator, and choose the standard accelerator option.
2. For **Address type**, select **IPv4** or **Dual-stack**.
3. Optionally, configure static IP addresses with bring your own IP address.
4. Add a listener with a specific set of ports or port range, and choose the protocol to accept: TCP or UDP.
5. Add one or more endpoint groups, one for each AWS Region in which you have endpoint resources.
6. Add one or more endpoints to the endpoint groups. This isn't required, but traffic won't be routed if you don't have any endpoints. To learn about the types of endpoints and requirements, see Requirements for resources you add as accelerator endpoints.

The following sections provide steps for adding, deleting, and configuring standard accelerators and their components, including listeners, endpoint groups, and endpoints.

Topics

- Standard accelerators in AWS Global Accelerator
- Listeners for standard accelerators in AWS Global Accelerator
- Endpoint groups for standard accelerators in AWS Global Accelerator
- Endpoints for standard accelerators in AWS Global Accelerator

Standard accelerators in AWS Global Accelerator

A *standard accelerator* in AWS Global Accelerator directs traffic over the AWS global network to endpoints that you include in specified AWS Regions. Each accelerator includes one or more listeners. A listener processes inbound connections from clients to Global Accelerator, based on the protocol (or protocols) and port (or port range) that you configure.

For standard accelerators, Global Accelerator directs traffic to the optimal regional endpoint based on health, client location, and policies that you configure, which increases the availability of your applications. Endpoints for standard accelerators can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses that are located in one AWS Region or multiple Regions.

Important

By default, Global Accelerator provides you with static IP addresses that are associated with your accelerator The IP addresses are assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you *delete* an accelerator, you lose the Global Accelerator static IP addresses that are assigned to the accelerator, so that you can no longer route traffic by using them. As a best practice, ensure that you have permissions in place to avoid inadvertently deleting accelerators. You can use IAM policies with Global Accelerator, for example, tag-based permissions, to limit the users who have permissions to delete an accelerator. For more information, see ABAC with Global Accelerator.

This section includes procedures for working with a standard accelerator on the Global Accelerator console. If you want to use API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

Contents

- Create accelerator
- Update accelerator
- Delete accelerator
- View your accelerators
- Adding an accelerator when you create a load balancer

- Compare using global static IP addresses to regional static IP addresses

Create accelerator

This section explains how to create a standard accelerator on the console. To work with Global Accelerator programmatically, see the AWS Global Accelerator API Reference.

To create a standard accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. Choose **Create accelerator**.
3. Provide a name for your accelerator.
4. For **Accelerator type**, select **Standard**.
5. For **IP address type**, select **IPv4** or **DUAL-STACK**.
6. Optionally, if you brought your own IP address ranges to AWS (BYOIP), you can specify a static IP address for your accelerator, one from each address pool. Make this choice for each of the two static IP addresses for your accelerator.

   - For each static IP address, choose the IP address pool to use.

     > **Note**
     >
     > You must choose a different IP address pool for each static IP address. This restriction is because Global Accelerator assigns each address range to a different network zone, for high availability.

   - If you chose your own IP address pool, also choose a specific IP address from the pool. If you choose the default Amazon IP address pool, Global Accelerator assigns a specific IP address to your accelerator.

   For more information about the requirements for specifying or updating static IP addresses with BYOIP, see Requirements when you update an accelerator to change the IP address.
7. Optionally, add one or more tags to help you identify your accelerator resources.
8. Choose **Next** to add listeners, endpoint groups, and endpoints.

Update accelerator

This section explains how to update a standard accelerator on the console. To work with Global Accelerator programmatically, see the AWS Global Accelerator API Reference.

To update a standard accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. In the list of accelerators, choose one, and then choose **Edit**.
3. On the **Edit accelerator** page, make changes, such as the following:
   - Change the name of the accelerator.
   - Disable the accelerator so that it no longer accepts or routes traffic, or so that you can delete it.
   - Enable the accelerator, if it is disabled.
   - Update the IP address type. If it's set to IPv4, change it to dual-stack. Or if it's dual-stack, change it to IPv4.
   - Update tags.
4. Choose **Save changes**.

If you disable an accelerator, be aware of the following:

- Global Accelerator static IP addresses remain assigned to your accelerator even if you disable the accelerator and it no longer accepts or routes traffic. Your accelerator retains the same static IP addresses for as long as the accelerator exists.
- If you delete the accelerator, however, you lose the Global Accelerator static IP addresses that are assigned to it. At that time, you can no longer route traffic by using the addresses.

If you make changes to the IP address type, be aware of the following:

- Only an accelerator that has dual-stack endpoints can be changed to an IP address type of dual-stack.
- If you change the IP address type for an accelerator from dual-stack to IPv4, Global Accelerator saves the IPv6 IP addresses that are assigned to the accelerator. This means that if you change

the IP address type for the accelerator back to dual-stack, the original IPv6 static IP addresses are restored for the accelerator.

If you want to change other functionality for your accelerator, such as adding or removing endpoints, updating traffic dials, or adjusting endpoint weights, see the specific sections that cover those topics, such as the following:

- Add a standard listener
- Add a standard endpoint group
- Add a standard endpoint

Delete accelerator

If you created an accelerator as a test or if you're no longer using an accelerator, you can delete it. On the console, disable the accelerator, and then you can delete it. You don't have to remove listeners and endpoint groups from the accelerator.

To delete an accelerator by using an API operation instead of the console, you must first remove all listeners and endpoint groups that are associated with the accelerator, and then disable it. For more information, see the DeleteAccelerator operation in the *AWS Global Accelerator API Reference*.

To disable an accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. In the list, choose an accelerator that you want to disable.
3. Choose **Edit**.
4. Choose **Disable accelerator**, and then choose **Save**.

To delete an accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. In the list, choose an accelerator that you want to delete.
3. Choose **Delete**.

> **Note**
>
> If you haven't disabled the accelerator, **Delete** is unavailable.

4. In the confirmation dialog box, choose **Delete**.

> **Important**
>
> When you delete an accelerator, you lose the static IP addresses that are assigned to the accelerator, so you can no longer route traffic by using them.

View your accelerators

You can view information about your accelerators on the console. To see descriptions of your accelerators programmatically, see ListAccelerators and DescribeAccelerator in the *AWS Global Accelerator API Reference*.

To view information about your accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. To see details about an accelerator, in the list, choose an accelerator, and then choose **View**.

Adding an accelerator when you create a load balancer

When you create an Application Load Balancer or Network Load Balancer in the AWS Management Console, you can optionally add an accelerator at the same time. Elastic Load Balancing and Global Accelerator work together to transparently add the accelerator for you. The accelerator is created in your account, with the load balancer as an endpoint. Using an accelerator provides static IP addresses and improves the availability and performance of your applications. (Learn more about accelerators by reading What is AWS Global Accelerator?.)

> **Important**
>
> To create an accelerator, you must have the correct permissions in place. For more information, see Identity-based policy examples for AWS Global Accelerator.

Configure and view your accelerator

You must update your DNS configuration to direct traffic to the static IP addresses or DNS name for the accelerator. Traffic won't go through the accelerator to your load balancer until your configuration changes are complete.

After you create your load balancer by choosing the Global Accelerator add-on on the Amazon EC2 console, go to the **Integrated services** tab to see the static IP addresses and Domain Name System (DNS) name for your accelerator. You use this information to start routing user traffic to the load balancer over the AWS global network. For more information about the DNS name assigned to your accelerator, see DNS addressing and custom domains in AWS Global Accelerator.

You can view and configure your accelerator by navigating to Global Accelerator in the AWS Management Console. For example, you can see the accelerators that are associated with your account or add additional load balancers to your accelerator. For more information, see View your accelerators and Create accelerator.

Pricing

With AWS Global Accelerator, you pay only for what you use. You are charged an hourly rate and data transfer costs for each accelerator in your account. For more information, see AWS Global Accelerator Pricing.

Stop using the accelerator

If you'd like to stop routing traffic through Global Accelerator to your load balancer, do the following:

1. Update your DNS configuration to point your traffic directly to the load balancer.
2. Delete the load balancer from the accelerator. For more information, see *To remove an endpoint* in Add a standard endpoint.
3. Delete the accelerator. For more information, see Delete accelerator.

Compare using global static IP addresses to regional static IP addresses

If you want to use a static IP address in front of an AWS resource, such as an Amazon EC2 instance, you have several options. For example, you can allocate an Elastic IP address, which is a static IPv4 or IPv6 address that you can associate with an Amazon EC2 instance or network interface in a single AWS Region.

If you have a global audience, you can create an accelerator with Global Accelerator to get global static addresses that are announced from AWS edge locations around the world. For IPv4, Global Accelerator provides two global static IPv4 addresses. For dual-stack, Global Accelerator provides a total of four global static IP addresses: two IPv4 addresses and two IPv6 addresses. If you already have AWS resources set up for your applications, in one or multiple Regions, including Amazon EC2 instances, Network Load Balancers, and Application Load Balancers, you can easily add those to Global Accelerator to front them with global static IP addresses. For more information, see Requirements for resources you add as accelerator endpoints.

Opting to use global static IP addresses provisioned by Global Accelerator can also improve the availability and performance of your applications. With Global Accelerator, static IP addresses accept incoming traffic onto the AWS global network from the edge location that is closest to your users. Maximizing time that traffic is on the AWS network can provide a faster and better customer experience. For more information, see How AWS Global Accelerator works .

You can add an accelerator from the AWS Management Console or by using API operations with the AWS CLI or SDKs. For more information, see Create accelerator.

Note the following when you add an accelerator:

- The global static IP addresses provisioned by Global Accelerator remain assigned to you for as long as your accelerator exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, if you delete an accelerator, you lose the static IP addresses that are assigned to it. For more information, see Delete accelerator.
- With Global Accelerator, you pay only for what you use. You are charged an hourly rate and data transfer costs for each accelerator in your account. For more information, see AWS Global Accelerator Pricing.

Listeners for standard accelerators in AWS Global Accelerator

With AWS Global Accelerator, you add listeners that process inbound connections from clients based on the ports and protocols that you specify. Listeners support TCP and UDP protocols.

You define a standard listener when you create your standard accelerator, and you can add more listeners at any time. You associate each listener with one or more endpoint groups, and you associate each endpoint group with one AWS Region.

Optionally, you can configure *client affinity* for a listener. With client affinity, Global Accelerator directs all requests from a user at a specific source (client) IP address to the same endpoint resource. Choosing this option maintains client affinity for your users.

Contents

- Add a standard listener
- Edit a standard listener
- Remove a standard listener
- How client affinity works in Global Accelerator

Add a standard listener

This section provides the steps to create a standard listener on the AWS Global Accelerator console. To complete this task by using an API operation instead of the console, see `CreateListener`, in the *AWS Global Accelerator API Reference*.

To add a listener

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. On the **accelerators** page, choose an accelerator.
3. Choose **Add listener**.
4. On the **Add listener** page, enter the ports or port ranges that you want to associate with the listener. Listeners support ports 1-65535.
5. Choose the protocol for the ports that you entered.

6. Optionally, choose to enable client affinity. Client affinity for a listener means that Global Accelerator ensures that connections from a specific source (client) IP address are always routed to the same endpoint. To enable this behavior, in the dropdown list, choose **Source IP**.

   The default is **None**, which means that client affinity is not enabled and Global Accelerator distributes traffic equally between the endpoints in the endpoint groups for the listener.

   For more information, see How client affinity works in Global Accelerator.
7. Choose **Add listener**.

Edit a standard listener

This section provides the steps to edit a standard listener on the AWS Global Accelerator console. To complete this task by using an API operation instead of the console, see `UpdateListener` in the *AWS Global Accelerator API Reference*.

To edit a standard listener

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. On the **accelerators** page, choose an accelerator.
3. Choose a listener, and then choose **Edit listener**.
4. On the **Edit listener** page, change the ports, port ranges, or protocols that you want to associate with the listener.
5. Optionally, choose to enable client affinity. Client affinity for a listener means that Global Accelerator ensures that connections from a specific source (client) IP address are always routed to the same endpoint. To enable this behavior, in the dropdown list, choose **Source IP**.

   The default is **None**, which means that client affinity is not enabled and Global Accelerator distributes traffic equally between the endpoints in the endpoint groups for the listener.

   For more information, see How client affinity works in Global Accelerator.
6. Choose **Save**.

Remove a standard listener

This section provides the steps to remove a standard listener on the AWS Global Accelerator console. To complete this task by using an API operation instead of the console, see `DeleteListener` in the *AWS Global Accelerator API Reference*.

To remove a listener

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. On the **accelerators** page, choose an accelerator.
3. Choose a listener, and then choose **Remove**.
4. In the confirmation dialog box, choose **Remove**.

How client affinity works in Global Accelerator

If you have stateful applications that you use with a standard accelerator, you can configure *client affinity* to have Global Accelerator direct all requests from a user at a specific source (client) IP address to the same endpoint resource. Choosing this option maintains client affinity for your users.

By default, client affinity for a standard listener is set to **None** and Global Accelerator distributes traffic equally between the endpoints in the endpoint groups for the listener.

Global Accelerator uses a consistent-flow hashing algorithm to choose the optimal endpoint for a user's connection. If you configure client affinity for your Global Accelerator resource to be **None**, Global Accelerator uses the 5-tuple properties—source IP, source port, destination IP, destination port, and protocol—to select the hash value. Next, it chooses the endpoint that provides the best performance. If a given client uses different ports to connect to Global Accelerator and you've specified this setting, Global Accelerator can't ensure that connections from the client are always routed to the same endpoint.

If you want to maintain client affinity by routing a specific user—identified by their source IP address—to the same endpoint each time they connect, set client affinity to **Source IP**. When you specify this option, Global Accelerator uses the 2-tuple properties—source IP and destination IP—to select the hash value and route the user to the same endpoint whenever they connect. Additionally, Global Accelerator honors client affinity by routing all connections with the same source IP address to the same endpoint group.

On occasion, network maintenance or disruptions created by variations in internet traffic routing can cause client traffic to shift to different Global Accelerator edge locations. When this happens, if the edge location that now serves the client traffic prefers a different AWS Region, then client affinity is not guaranteed to be maintained.

In addition, be aware that when you've set endpoint weights in your accelerator, in specific, limited scenarios, Global Accelerator overrides those weights, to help ensure availability. When Global Accelerator is load balancing traffic across endpoints in an endpoint group, it must, in certain circumstances, choose between preserving availability for client traffic and abiding by endpoint weights. For example, with accelerators where the client IP address is preserved, Global Accelerator might need to override an endpoint weight setting to help avoid connection collisions.

Endpoint groups for standard accelerators in AWS Global Accelerator

An endpoint group routes requests to one or more registered endpoints in AWS Global Accelerator. When you add a listener in a standard accelerator, you specify the endpoint groups for Global Accelerator to direct traffic to. An endpoint group, and all the endpoints in it, must be in one AWS Region. You can add different endpoint groups for different purposes, for example, for blue/green deployment testing.

Global Accelerator directs traffic to endpoint groups in standard accelerators based on the location of the client and the health of the endpoint group. If you like, you can also set the percentage of traffic to send to an endpoint group. You do that by using the traffic dial to increase (dial up) or decrease (dial down) traffic to the group. The percentage is applied only to the traffic that Global Accelerator is already directing to the endpoint group, not all traffic coming to a listener.

You can define health check settings for Global Accelerator for each endpoint group. By updating health check settings, you can change your requirements for polling and verifying the health of Amazon EC2 instance and Elastic IP address endpoints. For Network Load Balancer and Application Load Balancer endpoints, configure health check settings on the Elastic Load Balancing console.

Global Accelerator continually monitors the health of all endpoints that are included in a standard endpoint group, and routes requests only to the active endpoints that are healthy. For more information, see Ensure health check access for your accelerator If there aren't any healthy endpoints to route traffic to, Global Accelerator routes requests to all endpoints.

This section explains how to work with endpoint groups for standard accelerators on the AWS Global Accelerator console. If you want to use API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

Contents

- Add a standard endpoint group
- Edit a standard endpoint group
- Remove a standard endpoint group
- Use traffic dials to adjust traffic flow to Regions
- Override listener ports for restricted ports or connection collisions
- Ensure health check access for your accelerator

Add a standard endpoint group

You work with endpoint groups on the AWS Global Accelerator console or by using an API operation. You can add or remove endpoints from an endpoint group at any time.

This section explains how to add a standard endpoint groups on the AWS Global Accelerator console. If you want to use API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

To add a standard endpoint group

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. On the **Accelerators** page, choose an accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of the listener that you want to add an endpoint group to.
4. Choose **Add endpoint group**.
5. In the section for a listener, specify a Region for the endpoint group by choosing one from the dropdown list.
6. Optionally, for **Traffic dial**, enter a number from 0 to 100 to set a percentage of traffic for this endpoint group. The percentage is applied only to the traffic that is already directed to this endpoint group, not all listener traffic. By default, the traffic dial is set to 100.

7. Optionally, to override the listener port used for routing traffic to endpoints and reroute traffic to specific ports on your endpoints, choose **Configure port overrides**. For more information, see Override listener ports for restricted ports or connection collisions.
8. Optionally, to specify custom health check values to be applied to EC2 instance and Elastic IP address endpoints, choose **Configure health checks**. For more information, see Ensure health check access for your accelerator.
9. Optionally, choose **Add endpoint group** to add additional endpoint groups for this listener or other listeners.
10. Choose **Add endpoint group**.

Edit a standard endpoint group

This section explains how to edit a standard endpoint groups on the AWS Global Accelerator console. If you want to use API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

To edit an endpoint group

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. On the **Accelerators** page, choose an accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of the listener that the endpoint group is associated with.
4. Choose **Edit endpoint group**.
5. On the **Edit endpoint group** page, change the Region, adjust the traffic dial percentage, or choose **Configure health checks** to modify the health check settings.
6. Choose **Save**.

Remove a standard endpoint group

This section explains how to remove a standard endpoint groups on the AWS Global Accelerator console. If you want to use API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

To remove a standard endpoint group

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. On the **Accelerators** page, choose an accelerator.
3. In the **Listeners** section, choose a listener.
4. In the **Endpoint groups** section, choose an endpoint group, and then choose **Remove**.
5. On the confirmation dialog box, choose **Remove**.

Use traffic dials to adjust traffic flow to Regions

For each standard endpoint group, you can set a traffic dial to control the percentage of traffic that is directed to the endpoint group (AWS Region). The percentage is applied only to traffic that is already directed to the endpoint group, not to all listener traffic.

Note that when you change a traffic dial, the updated setting applies to only new connections. Existing connections are not terminated to adjust the current traffic flow.

By default, the traffic dial is set to 100 (that is, 100%) for all regional endpoint groups in an accelerator. The traffic dial lets you easily do performance testing or blue/green deployment testing for new releases across different AWS Regions, for example.

Here are a few examples to illustrate how you can use traffic dials to change the traffic flow to endpoint groups.

Upgrade your application by Region

If you want to upgrade an application in a Region or do maintenance, first set the traffic dial to 0 to cut off traffic for the Region. When you complete the work and you're ready bring the Region back into service, adjust the traffic dial to 100 to dial the traffic back up.

Mix traffic between two Regions

This example shows how traffic flow works when you change the traffic dials for two regional endpoint groups at the same time. Let's say that you have two endpoint groups for your accelerator—one for the `us-west-2` Region and one for the `us-east-1` Region—and you've set the traffic dials to 50% for each endpoint group.

Now, say you have 100 requests coming to your accelerator, with 50 from the East Coast of the United States and 50 from the West Coast. The accelerator directs the traffic as follows:

- The first 25 requests on each coast (50 requests in total) are served from their nearby endpoint group. That is, 25 requests are directed to the endpoint group in `us-west-2` and 25 are directed to the endpoint group in `us-east-1`.
- The next 50 requests are directed to the opposite Regions. That is, the next 25 requests from the East Coast are served by `us-west-2`, and the next 25 requests from the West Coast are served by `us-east-1`.

The result in this scenario is that both endpoint groups serve the same amount of traffic. However, each one receives a mix of traffic from both Regions.

Load sharing multi-Region architectures

You can configure the traffic dial and endpoint weights to implement complex scenarios as well, to configure load sharing between application endpoints. With these Global Accelerator features, you can deploy and run applications in multi-Region architectures, including active-active and active-standby setups. For more information and detailed examples, see the following blog post: Deploying multi-Region applications in AWS using AWS Global Accelerator

Override listener ports for restricted ports or connection collisions

By default, an accelerator routes user traffic to endpoints in AWS Regions using the protocol and port ranges that you specify when you create a listener. For example, if you define a listener that accepts TCP traffic on ports 80 and 443, the accelerator routes traffic to those ports on an endpoint.

However, when you add or update an endpoint group, you can override the listener port used for routing traffic to endpoints. For example, you can create a port override in which the listener receives user traffic on ports 80 and 443, but your accelerator routes that traffic to ports 1080 and 1443, respectively, on the endpoints.

One benefit of using port overrides can be to help avoid connection collisions, which can cause intermittent connectivity issues in Global Accelerator, resulting in TCP connection time delays, in certain scenarios. These collisions can occur when users (with the same source IP and source port) access resources in Global Accelerator. You can prevent the collisions, and thus avoid the delays, by configuring port overrides in your accelerators. For more information, see How to avoid connection collisions that result in TCP connection time delays.

Overriding a port can also help you avoid issues with listening on restricted ports. It's safer to run applications that don't require superuser (root) privileges on your endpoints. However, in Linux and

other UNIX-like systems, you must have superuser privileges to listen on restricted ports (TCP or UDP ports below 1024). By mapping a restricted port on a listener to a non-restricted port on an endpoint, you avoid this issue. You can accept traffic on restricted ports while running applications without root access on your endpoints behind Global Accelerator. For example, you can override a listener port 443 to an endpoint port 8443.

For each port override, you specify a listener port that accepts traffic from users and the endpoint port that Global Accelerator will route that traffic to. For more information, see Add a standard endpoint group.

When you create a port override, keep the following in mind:

- **Endpoint ports can't overlap listener port ranges.** The endpoint ports that you specify in a port override cannot be included in any of the listener port ranges that you've configured for the accelerator. For example, say that you have two listeners for an accelerator, and you've defined the port ranges for those listeners as 100-199 and 200-299, respectively. When you create a port override, you can't define one from listener port 100 to endpoint port 210, for example, because the endpoint port (210) is included in a listener port range that you defined (200-299).
- **No duplicate endpoint ports.** If one port override in an accelerator specifies an endpoint port, you can't specify the same endpoint port with port override from a different listener port. For example, you can't specify a port override from listener port 80 to endpoint port 90 together with an override from listener port 81 to endpoint port 90.
- **Health check continues to use the original port.** If you specify a port override for a port that is configured as a health check port, the health check still uses the original port, not the override port. For example, say that you specify health checks on listener port 80, and you also specify a port override from listener port 80 to endpoint port 480. Health checks continue to use endpoint port 80. However, user traffic that comes in through port 80 goes to port 480 on the endpoint.

  This behavior maintains consistency between Network Load Balancer, Application Load Balancer, EC2 instance, and Elastic IP address endpoints. Because Network Load Balancers and Application Load Balancers don't map health check ports to a different endpoint ports when you specify a port override in Global Accelerator, it would be inconsistent for Global Accelerator to map health check ports to different endpoint ports for EC2 instance and Elastic IP address endpoints.
- **Security group settings must allow port access.** Make sure that your security groups allow traffic to arrive at endpoint ports that you've designated in port overrides. For example, if you override listener port 443 to endpoint port 1433, make sure that any port restrictions set in your

security group for that Application Load Balancer or Amazon EC2 endpoint allow inbound traffic on port 1433.

Ensure health check access for your accelerator

Each listener for a standard accelerator routes requests only to healthy, active endpoints. When you add an endpoint, it must pass a health check to be considered healthy. AWS Global Accelerator also regularly sends health check requests to all endpoints on standard accelerators, to test their status. Global Accelerator automatically runs these regular health checks. After each health check is completed, the listener closes the connection that was established for the health check.

Note that if there aren't any healthy endpoints to route traffic to, Global Accelerator routes incoming client requests to *all* endpoints in the endpoint group. For more information, see How failover works for unhealthy endpoints.

Details about how health checks work, and guidance about using health checks, depends on the type of endpoint resource. This topic provides information about how to work with health checks for different endpoint types, including steps for updating health check options in Global Accelerator (applies to EC2 instance or Elastic IP address endpoints).

Ensure access for your accelerator health checks

To ensure access for health checks to complete successfully for EC2 instance or Elastic IP address endpoints, make sure that your router and firewall rules allow inbound traffic from the IP addresses associated with Amazon Route 53 health checkers. To see the list of IP address ranges associated with Route 53 health checkers, see IP address ranges of Route 53 servers in the *Amazon Route 53 Developer Guide*.

Global Accelerator health checks work by receiving traffic for Route 53 health checks, which is forwarded to the configured health check port for the endpoint group. Typically, the ports configured for health checks match the listener configuration. If you configure a different port for health checks instead, review your security group configuration to make sure that you don't allow public traffic on the port.

For example, if your listener is configured on port 80, then your health check port is also 80. If you choose to configure health ports on another port, for example, port 83, then make sure that you configure your security groups to allow traffic on port 83 only from IP addresses that are in the IP address range for Route 53 health checks.

Health check guidance for different endpoint types

Review the information in this section for guidelines about the health checks that you specify for each endpoint type for your accelerator.

In addition, make sure that the health checks that you choose for endpoints with HTTP workloads are representative of the overall health of your application, and that you follow the guidance for ensuring access to health checks that is described in the preceding section, Ensure security and access for health checks.

The following guidelines apply to each specified endpoint type:

- For Network Load Balancer or Application Load Balancer endpoints, be aware of the following:
  - The health check options that you choose in Global Accelerator do not affect Network Load Balancers or Application Load Balancers that you've added as endpoints. That is, health check options that you specify in Global Accelerator are used for Amazon EC2 and Elastic IP address health checks, but not for health checks on load balancer endpoints.

    For load balancer endpoints, configure health checks by using Elastic Load Balancing configuration options. For more information, see Health checks for your target groups.
  - Global Accelerator considers an Application Load Balancer healthy if every target group has at least one healthy target. For more information, see Health checks for your target groups.
  - Global Accelerator considers a Network Load Balancer healthy if there is at least one healthy Availability Zone. An Availability Zone is healthy if it has a healthy target in all load balancer target groups that it is in.

    Be aware that when you enable cross-zone load balancing, healthy targets in Network Load Balancer target groups contribute to the health of the target group in all Availability Zones. This is how Global Accelerator evaluates the health of an AZ in a target group regardless of which AZs are actually healthy. This means that with cross-zone load balancing, if every target group contains a healthy target, Global Accelerator considers the Network Load Balancer to be healthy. However, at all times, Global Accelerator only considers a Network Load Balancer healthy if the number of healthy targets meets the Network Load Balancer minimum healthy target count setting, `minimum_healthy_targets`.
- For EC2 instance or Elastic IP address endpoints, be aware of the following:
  - When you add EC2 instance or Elastic IP address endpoints to a listener configured with TCP, you can specify the port to use for health checks. By default, if you don't specify a port for health checks, Global Accelerator uses the listener port that you specified for your accelerator.

- When you add these endpoint types with a UDP listener, Global Accelerator uses the listener port and the TCP protocol for health checks, so you must have a TCP server on your endpoint.

  Make sure to check that the port that you've configured for the TCP server on each endpoint is the same as the port that you specify for the health check in Global Accelerator. If the port numbers aren't the same, or if you haven't set up a TCP server for the endpoint, Global Accelerator marks the endpoint as unhealthy, regardless of the endpoint's health.
- Make sure to follow the guidance for security and access when you configure ports for health checks for your EC2 instance or Elastic IP address endpoints.

Set health check options

To set health check options for your accelerator, specify one or more of the following options when you create an accelerator or when you edit an endpoint group.

You can add the following health check options for an endpoint group.

Health check port

The port to use when Global Accelerator performs health checks on endpoints that are part of this endpoint group.

Note that you can't set a port override for health check ports.

Health check protocol

The protocol to use when Global Accelerator performs health checks on endpoints that are part of this endpoint group.

Health check interval

The interval, in seconds, between each health check for an endpoint.

Threshold count

The number of consecutive health checks required before considering an unhealthy target healthy or a healthy target unhealthy.

Endpoints for standard accelerators in AWS Global Accelerator

Endpoints for standard accelerators in AWS Global Accelerator can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses. In AWS Global

Accelerator, static IP addresses serve as a single point of contact for clients, and, with a standard accelerator, Global Accelerator distributes incoming traffic across healthy endpoints. Global Accelerator directs traffic to endpoints by using the port (or port range) that you specify for the listener that the endpoint group for the endpoint belongs to.

Each endpoint group can have multiple endpoints. You can add each endpoint to multiple endpoint groups, but the endpoint groups must be associated with different listeners. A resource must be valid and active when you add it as an endpoint.

> **Important**
>
> Accelerators that you configure as dual-stack (that is, accelerators that you want to support IPv4 and IPv6) require that you add only endpoints that also support dual-stack. Network Load Balancers, Application Load Balancers, and Amazon EC2 instances can be added as dual-stack endpoints.

Global Accelerator continually monitors the health of all endpoints that are included in a standard endpoint group. It routes traffic only to the active endpoints that are healthy. If Global Accelerator doesn't have any healthy endpoints to route traffic to, it routes traffic to all endpoints in the AWS Region.

Contents

- Requirements for resources you add as accelerator endpoints
- Add a standard endpoint
- Edit a standard endpoint
- Remove a standard endpoint
- How endpoint weights work to manage traffic volume
- How failover works for unhealthy endpoints
- How to avoid connection collisions that result in TCP connection time delays

Requirements for resources you add as accelerator endpoints

Be aware of the following requirements and limitations for different types of resources that you can add as endpoints for standard accelerators in AWS Global Accelerator. Some requirements apply regardless of the type of resource that you add.

All resource types

- Before you enable client IP address preservation for an endpoint, there are additional requirements to keep in mind. For more information, see Transition endpoints with client IP address preservation.
- To add an endpoint to a dual-stack accelerator, the endpoint must have client IP address preservation enabled.
- When you add resources as endpoints behind Global Accelerator, we recommend that you don't also send traffic directly to the same endpoints over the internet. Sending direct traffic can lead to connection collision issues. For more information, see How to avoid connection collisions that result in TCP connection time delays.
- The resources that you add as endpoints for an accelerator and the accelerator itself must be owned by the same account, unless you configure cross-account support. However, the target instances behind a load balancer endpoint can be owned by different accounts. In this scenario, the accounts that own the target instances must be given permission to access a subnet owned by the account that owns the load balancer and accelerator. For more information, see Configure cross-account access in Global Accelerator.
- Before you terminate or delete a resource that you've added as an endpoint behind an accelerator, we recommend that you remove the endpoint from Global Accelerator endpoint groups.

Application Load Balancer endpoints

- An Application Load Balancer endpoint can be internet-facing or internal.
- Dual-stack Application Load Balancers can be added as endpoints.
- Global Accelerator only supports Application Load Balancers running inside an AWS Region. Global Accelerator does not support an Application Load Balancer running as an endpoint in a Local Zone.

Network Load Balancer endpoints

- A Network Load Balancer endpoint can be internet-facing or internal.
- Dual-stack Network Load Balancers can be added as endpoints, but there are a few restrictions:
  - For dual-stack accelerators, when you add a dual-stack Network Load Balancer, the Network Load Balancer cannot have a target group with a target type of `ip`, or a target type of `instance` and IP address type of `ipv6`.
  - For IPv4 accelerators, when you add a dual-stack Network Load Balancer, you cannot enable client IP address preservation for the endpoint in Global Accelerator.

- Global Accelerator only supports Network Load Balancers running inside an AWS Region. Global Accelerator does not support a Network Load Balancer running as an endpoint in a Local Zone.
- For Network Load Balancer endpoints, we recommend that you disable cross-zone traffic for the load balancers to avoid connection collisions, which can result in increased TCP connection time. For more information, see How to avoid connection collisions that result in TCP connection time delays.
- Global Accelerator does not support using shared subnets to target Network Load Balancer endpoints with client IP address preservation.

Amazon EC2 instance endpoints

- An EC2 instance endpoint can't be one of the following types: C1, CC1, CC2, CG1, CG2, CR1, CS1, G1, G2, HI1, HS1, M1, M2, M3, or T1.
- EC2 instances are supported as endpoints in specific AWS Regions. For more information, see AWS Region availability for AWS Global Accelerator.

  Global Accelerator only supports EC2 instances inside an AWS Region. Global Accelerator does not support routing to an Elastic IP address as an endpoint in a Local Zone.
- We recommend that you remove an EC2 instance from Global Accelerator endpoint groups before you terminate the instance. If you terminate an EC2 instance before you remove it from an endpoint group in Global Accelerator, and then you create another instance in the same VPC with the same private IP address, and health checks pass, Global Accelerator will route traffic to the new endpoint.
- Dual-stack EC2 instances can be added as endpoints. However, the instances must have a primary IPv6 elastic network interface (ENI) attached to them. For more information, see Work with network interfaces in the Amazon Elastic Compute Cloud User Guide.

Elastic IP addresses

- Dual-stack Elastic IP addresses cannot be added as endpoints.

Add a standard endpoint

You add endpoints to endpoint groups so that traffic can be directed to your resources. You can edit a standard endpoint to change the weight for the endpoint. Or you can remove an endpoint from your accelerator by removing it from an endpoint group. Removing an endpoint doesn't affect the endpoint itself, but Global Accelerator can no longer direct traffic to that resource.

You must create a resource first, and then you can add it as an endpoint in Global Accelerator. A resource must be valid and active when you add it as an endpoint. For detailed information about the endpoint types and configurations that Global Accelerator supports, see Requirements for resources you add as accelerator endpoints.

One reason that you might add or remove endpoints from endpoint groups is usage. For example, if demand on your application increases, you can create more resources. Then, you can add more endpoints to one or more endpoint groups to handle the increased traffic. Global Accelerator starts routing requests to an endpoint as soon as you add it and the endpoint passes the initial health checks.

You can manage traffic to endpoints by adjusting the weights on an endpoint, to send proportionally more or less traffic to the endpoint. For more information, see How endpoint weights work to manage traffic volume.

Note: if you're considering adding an endpoint with client IP address preservation, first review the information in Preserve client IP addresses in AWS Global Accelerator.

This section explains how to add endpoints on the AWS Global Accelerator console. If you want to use API operations with AWS Global Accelerator, see the AWS Global Accelerator API Reference.

To add a standard endpoint

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. On the **Accelerators** page, choose an accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of a listener.
4. In the **Endpoint groups** section, for **Endpoint group ID**, choose the ID of the endpoint group that you want to add an endpoint to.
5. Choose **Edit**.
6. In the **Endpoints** section, choose **Add endpoint**.
7. On the **Add endpoints** page, choose a resource from the dropdown list.

   If you don't have any AWS resources, there aren't any items in the list. To continue, create AWS resources such as load balancers, Amazon EC2 instances, or Elastic IP addresses. Then come back to the steps here, and choose a resource from the list.

> **Note**
>
> If you have a dual-stack accelerator, you must add a dual-stack endpoint. Network Load Balancers, Application Load Balancers, and Amazon EC2 instances can be added as dual-stack endpoints.

8. Optionally, for **Weight**, enter a number from 0 to 255 to set a weight for routing traffic to this endpoint. When you add weights to endpoints, you configure Global Accelerator to route traffic based on proportions that you specify. By default, all endpoints have a weight of 128. For more information, see How endpoint weights work to manage traffic volume.
9. Optionally, enable client IP address preservation for the endpoint. Under **Preserve client IP address**, select **Preserve address**. For more information, see Preserve client IP addresses in AWS Global Accelerator.

> **Note**
>
> Before you add and begin to route traffic to endpoints that preserve the client IP address, make sure that all your required security configurations, for example, security groups, are updated to include the user client IP address on allow lists.

10. Choose **Add endpoint**.

Edit a standard endpoint

This section explains how to edit an endpoint on the AWS Global Accelerator console. If you want to use API operations with AWS Global Accelerator, see the AWS Global Accelerator API Reference.

To edit a standard endpoint

You can edit an endpoint configuration to change the weight. For more information, see How endpoint weights work to manage traffic volume.

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. On the **accelerators** page, choose an accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of a listener.
4. In the **Endpoint groups** section, for **Endpoint group ID**, choose the ID of the endpoint group.

5. Choose **Edit endpoint**.
6. On the **Edit endpoint** page, make updates, and then choose **Save**.

Remove a standard endpoint

This section explains how to remove an endpoint on the AWS Global Accelerator console. If you want to use API operations with AWS Global Accelerator, see the AWS Global Accelerator API Reference.

You can remove endpoints from your endpoint groups, for example, if you need to service your endpoints. Removing an endpoint takes it out of the endpoint group, but does not affect the endpoint otherwise. Global Accelerator stops directing traffic to an endpoint as soon as you remove it from an endpoint group. The endpoint goes into a state where it waits for all current requests to be completed so there's no interruption for client traffic that is in progress. You can add the endpoint back to the endpoint group when you're ready for it to resume receiving requests.

Note: Before you terminate or delete a resource that you've added as an endpoint behind an accelerator, we recommend that you remove the endpoint from Global Accelerator endpoint groups.

To remove an endpoint

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. On the **accelerators** page, choose an accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of a listener.
4. In the **Endpoint groups** section, for **Endpoint group ID**, choose the ID of the endpoint group.
5. Choose **Remove endpoint**.
6. In the confirmation dialog box, choose **Remove**.

How endpoint weights work to manage traffic volume

Weighted routing lets you choose how much traffic is routed to a specific resource (endpoint) in an endpoint group. This can be useful in several ways, including for load balancing and for testing new versions of your application.

A weight is a value you can set that determines the proportion of traffic that Global Accelerator directs to an endpoint in a standard accelerator. Endpoints can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses. Global Accelerator calculates the sum of the weights for the endpoints in an endpoint group, and then directs traffic to the endpoints based on the ratio of each endpoint's weight to the total. By default, the weight for an endpoint is set to 128, which is half of the maximum value of 255.

How endpoint weights work

To use weights, you assign each endpoint in an endpoint group a relative weight that corresponds with how much traffic that you want to send to it. By default, the weight for an endpoint is 128—that is, half of the maximum value for a weight, 255. Global Accelerator sends traffic to an endpoint based on the weight that you assign to it as a proportion of the total weight for all endpoints in the group:

$$\frac{\text{Weight for a specified endpoint}}{\text{Sum of the weights for all endpoints}}$$

For example, if you want to send a tiny portion of your traffic to one endpoint and the rest to another endpoint, you might specify weights of 1 and 255, respectively. The endpoint with a weight of 1 gets 1/256 of the traffic (1/1+255), and the other endpoint gets 255/256 (255/1+255). You can gradually change the balance of traffic volume to each endpoint by changing the weights. If you want Global Accelerator to stop sending traffic to an endpoint, you can change the weight for that resource to 0.

Be aware that even when you've set endpoint weights in your accelerator, in specific, limited scenarios, Global Accelerator overrides those weights, to help ensure availability. That is, when Global Accelerator is load balancing traffic across endpoints in an endpoint group, it must, in certain circumstances, choose between preserving availability for client traffic and abiding by endpoint weights. For example, with accelerators where the client IP address is preserved, Global Accelerator might need to override an endpoint weight setting to help avoid connection collisions.

How failover works for unhealthy endpoints

If there are no healthy endpoints in an endpoint group that have a weight greater than zero, Global Accelerator tries to fail over to a healthy endpoint with a weight greater than zero in another endpoint group. Note that for this failover, Global Accelerator ignores the traffic dial setting. So if, for example, an endpoint group has a traffic dial set to zero, Global Accelerator still includes that endpoint group in the failover attempt.

If Global Accelerator doesn't find a healthy endpoint with a weight greater than zero after trying the three closest endpoint groups (that is, AWS Regions), it routes traffic to a random endpoint in the endpoint group that is closest to the client. That is, it *fails open*.

Note the following:

- The endpoint group chosen for failover might be one that has a traffic dial set to zero.
- The nearest endpoint group might not be the original endpoint group. This is because Global Accelerator considers account traffic dial settings when it chooses the original endpoint group.

For example, let's say your configuration has two endpoints, one healthy and one unhealthy, and you've set the weight for each of them to be greater than zero. In this case, Global Accelerator routes traffic to the healthy endpoint. However, now say you set the weight of the only healthy endpoint to zero. Global Accelerator then tries three additional endpoint groups to find a healthy endpoint with a weight greater than zero. If it doesn't find one, Global Accelerator routes traffic to a random endpoint in the endpoint group that is closest to the client.

When recovery occurs, that is, Regions are healthy again, Global Accelerator returns to regular routing behavior. This means that, typically, routing will start back to healthy endpoints with traffic dials that aren't set to zero in about 30 seconds or so. However, note that established active connections are not moved. They continue to route to the zero weight Region until the connection is reset by the client or the server, or until the client makes a new connection.

How to avoid connection collisions that result in TCP connection time delays

Intermittent connectivity issues can be caused by connection collisions in AWS Global Accelerator. These can occur when users (with the same source IP and source port) access resources in Global Accelerator in certain scenarios. The collisions can result in TCP connection time delays for traffic that goes through your accelerators.

You can avoid these delays by configuring your accelerators with *port overrides*, a feature in Global Accelerator that enables you to route incoming traffic to a different destination ports on your accelerator endpoints. Follow the guidance in this section to learn about how to use port overrides to prevent the connection collisions and avoid potential TCP connection time delays.

Scenarios that can cause connection collisions

There are three scenarios in Global Accelerator that can lead to connection collisions, and thus to TCP connection time delays:

- You configure the same resource as an endpoint with multiple accelerators.
- You configure resources as endpoints behind Global Accelerator, and you also send traffic directly over the internet from your end users to the same resources.
- You configure Network Load Balancer endpoints for cross-zone traffic.

For Network Load Balancer endpoints, we recommend that you disable cross-zone traffic for the load balancers to avoid connection collisions. For more information, see TCP Connection Delays in the *User Guide for Network Load Balancers*.

For the other scenarios, we recommend that you use the port override feature with the endpoint group to prevent collisions. Using port overrides, you can map Global Accelerator listener ports to different destination port numbers on an endpoint resource. Listener ports default to using the same port numbers on endpoint resources. By using port overrides, accelerators can route traffic from the same users (with the source IP and source port) to the same endpoint, but use different destination port numbers, which avoids collisions.

The next section provides specific examples for each of the scenarios of how you can configure port overrides to avoid connection collisions. For more information about configuring port overrides, see Override listener ports for restricted ports or connection collisions.

How to prevent connection collisions by using port overrides

By default, an accelerator routes user traffic to endpoints in AWS Regions using the same protocol and the same destination port ranges that you specify when you create a listener. However, you can optionally choose to override the port number mapping for the listener port. That is, you can map a listener port number to route traffic to a different destination port number on an endpoint.

For example, if you define a listener that accepts TCP traffic on ports 80 and 443, by default, the accelerator routes traffic to those same ports, 80 and 443, on endpoints. However, using the port override feature, the accelerator can route traffic coming in on those ports to different ports on endpoints, such as 8080 and 8443.

By creating different port mappings for listeners in two (or more) accelerators that have the same resources configured behind them, you can use separate destination port numbers for each accelerator and avoid collisions.

For example, say you have Accelerator-A and Accelerator-B, and each one has a listener configured for TCP and port 443. You can set up a port override for the listener for Accelerator-A to map port 443 to 8443, and the listener for Accelerator-B to map port 443 to 9443. Now you configure an Application Load Balancer endpoint, ALB-1234, for example, to listen on both ports 8443 and 9443. Then traffic coming in on port 443 (to the listeners for both accelerators) from the same user IP address will arrive at ALB-1234, without connection collisions or TCP connection time delays.

You can see the traffic paths for this example illustrated in the following:

```
Accelerator-A [listener: tcp,443] # Endpoint-Group [port-override:
443#8443] # ALB-1234 (listener: HTTPS,8443)

Accelerator-B [listener: tcp,443] # Endpoint-Group [port-override:
443#9443] # ALB-1234 (listener: HTTPS,9443)
```

You can use a port override in a similar way to prevent connection collisions for resources that are accessed by both direct user traffic and through an accelerator by overriding the default mapping for the accelerator's listener port number. To prevent collisions in this scenario, do the following:

1. Determine the port that you want the resource to listen on for your direct traffic.
2. Configure the listener for your accelerator to override the default port, and configure the listener on your resource to listen on that port for accelerator traffic.

For example, you could set up a port override for the listener for your accelerator to map port 443 to port 8443. Now, you could configure an Application Load Balancer endpoint, for example, to listen for your accelerator traffic on port 8443 and for direct traffic on port 443. With this configuration, you avoid connection collisions on the Application Load Balancer for traffic coming from the same user IP address.

Working with custom routing accelerators in AWS Global Accelerator

This chapter includes information about how a custom routing accelerator in AWS Global Accelerator works, and how to configure accelerators, listeners, endpoint groups, and VPC subnet endpoints for a custom routing accelerator.

A custom routing accelerator lets you use application logic to directly map one or more users to a specific Amazon EC2 instance among many destinations, while gaining the performance improvements of routing your traffic through Global Accelerator. This is useful when you have an application that requires a group of users to interact with each other on the same session running on a specific EC2 instance and port, such as gaming applications or Voice over IP (VoIP) sessions.

Endpoints for custom routing accelerators must be Amazon VPC (VPC) subnets, and a custom routing accelerator can only route traffic to Amazon EC2 instances in those subnets. When you create a custom routing accelerator, you can include thousands of Amazon EC2 instances running in a single or multiple VPC subnets. To learn more, see How custom routing accelerators work in Global Accelerator.

> **Note**
>
> If you want Global Accelerator to instead automatically choose the closest healthy endpoint to your clients, create a standard accelerator. For more information, see Working with standard accelerators in AWS Global Accelerator.

To set up custom routing accelerator, you do the following:

1. Review the guidelines and requirements for creating a custom routing accelerator. See Guidelines and restrictions for custom routing accelerators.
2. Create a VPC subnet. You can add EC2 instances to the subnet at any time after adding the subnet to Global Accelerator.
3. Create an accelerator in Global Accelerator. Select the option for a custom routing accelerator.
4. Add a listener in which you specify a range of ports for Global Accelerator to listen on. Make sure that you include a large range with enough ports for Global Accelerator to map to all the destinations that you expect to have. These ports are distinct from destination ports, which you

specify in the next step. For more information about listener port requirements, see Guidelines and restrictions for custom routing accelerators.

5. Add one or more endpoint groups for AWS Regions in which you have VPC subnets. You specify the following for each endpoint group:
   - An endpoint port range, which represents the ports on your destination EC2 instances that will be able to receive traffic.
   - The protocol for each destination port range: UDP, TCP, or both UDP and TCP.
6. For the endpoint subnet, select a subnet ID. You can add multiple subnets in each endpoint group and subnets can be different sizes (up to /17).

The following sections explain how custom routing accelerators work, and provide steps for creating and working with custom routing accelerators and their components, including listeners, endpoint groups, and VPC subnet endpoints.

Topics

- How custom routing accelerators work in Global Accelerator
- Example of how custom routing works in Global Accelerator
- Guidelines and restrictions for custom routing accelerators
- Custom routing accelerators in AWS Global Accelerator
- Listeners for custom routing accelerators in Global Accelerator
- Endpoint groups for custom routing accelerators in Global Accelerator
- Amazon VPC subnet endpoints for custom routing accelerators in Global Accelerator

How custom routing accelerators work in Global Accelerator

By using a custom routing accelerator in AWS Global Accelerator, you can use application logic to directly map one or more users to a specific destination among many destinations while still gaining the performance benefits of Global Accelerator. A custom routing accelerator maps listener port ranges to EC2 instance destinations in Amazon VPC (VPC) subnets. This allows Global Accelerator to deterministically route traffic to a specific Amazon EC2 private IP address and port destination in your subnet.

For example, you can use a custom routing accelerator with an online real-time gaming application in which you assign multiple players to a single session on an Amazon EC2 game server based on

factors that you choose, such as geographic location, player skill, and game mode. Or you might have a VoIP or social media application that assigns multiple users to a specific media server for voice, video, and messaging sessions.

Your application can call a Global Accelerator API and receive a full static mapping of Global Accelerator ports and their associated destination IP addresses and ports. You can save that static mapping, and then your matchmaking service use it to route users to specific destination EC2 instances. You don't have to make any modifications to your client software to start using Global Accelerator with your application.

To configure a custom routing accelerator, you select a VPC subnet endpoint. Then you define a destination port range that incoming connections will be mapped to, so your software can listen on the same set of ports across all instances. Global Accelerator creates a static mapping that allows your matchmaking service to translate a destination IP address and port number for a session to an external IP address and port that you give to users.

Your application's network stack might operate over a single transport protocol, or perhaps instead you use UDP for fast delivery and TCP for reliable delivery. You can set UDP, TCP, or both UDP and TCP for each destination port range, to give you maximum flexibility without having to duplicate your configuration for each protocol.

> **Note**
>
> By default, all VPC subnet destinations in a custom routing accelerator aren't allowed to receive traffic. This is to be secure by default, and also to give you granular control over which private EC2 instance destinations in your subnet are allowed to receive traffic. You can allow or deny traffic to the subnet, or to specific IP address and port combinations (destination sockets). For more information, see Add a VPC subnet endpoint for a custom routing accelerator. You can also specify destinations by using the Global Accelerator API. For more information, see AllowCustomRoutingTraffic and DenyCustomRoutingTraffic.

Example of how custom routing works in Global Accelerator

As an example, let's say that you want to support 10,000 sessions where groups of users interact, such as gaming sessions or VoIP call sessions, across 1,000 Amazon EC2 instances behind Global Accelerator. In this example, we'll specify a listener port range of 10001–20040 and a destination

port range of 81–90. We'll say that we have the four VPC subnets in us-east-1: subnet-1, subnet-2, subnet-3, and subnet-4.

In our example configuration, each VPC subnet has a block size of /24 so it can support 251 Amazon EC2 instances. (Five addresses are reserved and unavailable from each subnet, and these addresses are not mapped.) Each server running on each EC2 instance serves the following 10 ports, that we specified for the destination ports in our endpoint group: 81-90. This means that we have 2510 ports (10 x 251) associated with each subnet. Each port can be associated with a session.

Because we've specified 10 destination ports on each EC2 instance in our subnet, Global Accelerator internally associates them with 10 listener ports that you can use to access EC2 instances. To illustrate this simply, we'll say that there's a block of listener ports that starts with the first IP address of the endpoint subnet for the first set of 10, and then moves to the next IP address for the next set of 10 listener ports.

> **Note**
>
> The mapping is actually not predictable like this, but we're using a sequential mapping here to help to show how the port mapping works. To determine the actual mapping for your listener port ranges, use the following API operations: ListCustomRoutingPortMappings and ListCustomRoutingPortMappingsByDestination.

In our example, the first listener port is 10001. That port is associated with the first subnet IP address, 192.0.2.4, and the first EC2 port, 81. The next listener port, 10002, is associated with the first subnet IP address, 192.0.2.4, and the second EC2 port, 82. The following table illustrates how this example mapping continues through the last IP address of the first VPC subnet, and then on to the first IP address of the second VPC subnet.

Global Accelerator listener port	VPC subnet	EC2 instance port
10001	192.0.2.4	81
10002	192.0.2.4	82
10003	192.0.2.4	83
10004	192.0.2.4	84

Global Accelerator listener port	VPC subnet	EC2 instance port
10005	192.0.2.4	85
10006	192.0.2.4	86
10007	192.0.2.4	87
10008	192.0.2.4	88
10009	192.0.2.4	89
10010	192.0.2.4	90
10011	192.0.2.5	81
10012	192.0.2.5	82
10013	192.0.2.5	83
10014	192.0.2.5	84
10015	192.0.2.5	85
10016	192.0.2.5	86
10017	192.0.2.5	87
10018	192.0.2.5	88
10019	192.0.2.5	89
10020	192.0.2.5	90
...	...	...
12501	192.0.2.244	81
12502	192.0.2.244	82
12503	192.0.2.244	83

Global Accelerator listener port	VPC subnet	EC2 instance port
12504	192.0.2.244	84
12505	192.0.2.244	85
12506	192.0.2.244	86
12507	192.0.2.244	87
12508	192.0.2.244	88
12509	192.0.2.244	89
12510	192.0.2.244	90
12511	192.0.3.4	81
12512	192.0.3.4	82
12513	192.0.3.4	83
12514	192.0.3.4	84
12515	192.0.3.4	85
12516	192.0.3.4	86
12517	192.0.3.4	87
12518	192.0.3.4	88
12519	192.0.3.4	89
12520	192.0.3.4	90

Guidelines and restrictions for custom routing accelerators

When you create and work with custom routing accelerators in AWS Global Accelerator, keep the following guidelines and restrictions in mind.

Supported endpoint destinations

The virtual public cloud (VPC) subnet endpoints in a custom routing accelerator can only include EC2 instances. No other resources, such as load balancers, are supported for custom routing accelerators. The types of EC2 instances that are supported with Global Accelerator are listed in Endpoints for standard accelerators in AWS Global Accelerator.

With custom routing accelerators, Global Accelerator can only route traffic to private IP endpoints on Amazon EC2 instances on VPC subnets. However, gaming customers who want to use custom routing might need to connect to stateful sessions. To do this, the customers run their game servers on Amazon Elastic Kubernetes Service (EKS), with sessions hosted on a specific container running inside a Kubernetes Pod.

To use custom routing in this scenario, you can configure a VPC-CNI plugin to send traffic to Kubernetes Pods through an elastic network interface (ENI) that Global Accelerator creates for each subnet where an endpoint is present. This is a way to use a custom routing accelerator with EKS. The same configuration works to use a custom routing accelerator with Amazon Elastic Container Service (ECS). To learn more, see the detailed steps provided in the following blog post: AWS Global Accelerator Custom Routing with Amazon Elastic Kubernetes Service.

Port mappings

When you add a VPC subnet, Global Accelerator creates a static port mapping of listener port ranges to the port ranges supported by the subnet. The port mapping for a specific subnet never changes.

You can view the port mapping list for a custom routing accelerator programmatically. For more information, see `ListCustomRoutingPortMappings`.

VPC subnet size

VPC subnets that you add to a custom routing accelerator must be a minimum of /28 and a maximum of /17.

IP address type

Custom routing accelerators support only the IPv4 IP address type.

Listener port ranges

You must specify enough listener ports, by specifying listener port ranges, to accommodate the number of destinations included in the subnets that you plan to add to your custom

routing accelerator. The range that you specify when you create a listener determines how many listener port and destination IP address combinations that you can use with your custom routing accelerator. For maximum flexibility and to reduce the possibility of getting an error that you don't have enough listener ports available, we recommend that you specify a large port range.

Global Accelerator allocates port ranges in blocks when you add a subnet to a custom routing accelerator. We recommend that you allocate listener port ranges linearly and make the ranges large enough to support the number of destination ports that you intend to have. That is, the number of ports you should allocate should be at least the subnet size times the number of destination ports and protocols (destination configurations) that you will have in the subnet.

> **Note**
>
> The algorithm that Global Accelerator uses to allocate port mappings might require you to add more listener ports, beyond this total.

After you create a listener, you can edit it to add additional port ranges and associated protocols, but you can't decrease existing port ranges. For example, if you have a listener port range of 5,000–10,000, you can't change the port range to be 5900–10,000 and you can't change the port range to be 5,000–9,900.

Each listener port range must include a minimum of 16 ports. Listeners support ports 1-65535.

Destination port ranges

There are two places that you specify port ranges for a custom routing accelerator: the port ranges that you specify when you add a listener and the destination port ranges and protocols that you specify for an endpoint group.

- **Listener port ranges:** The listener ports on the Global Accelerator static IP addresses that your clients connect to. Global Accelerator maps each port to a unique destination IP address and port on a VPC subnet behind the accelerator.
- **Destination port ranges:** The sets of destination port ranges that you specify for an endpoint group (also called the destination configurations) are the EC2 instance ports that receive traffic. To receive traffic on destination ports, the Security Groups associated with your EC2 instances must permit traffic on them.

Health checks and failover

Global Accelerator does not perform health checks for custom routing accelerators and does not failover to healthy endpoints. Traffic for custom routing accelerators is routed deterministically, regardless of the health of a destination resource.

All traffic is denied by default

By default, traffic directed through a custom routing accelerator is denied to all destinations in your subnet. To enable destination instances to receive traffic, you must specifically allow all traffic to the subnet or, alternatively, allow traffic to specific instance IP addresses and ports in the subnet.

Updating a subnet or specific destination to allow or deny traffic takes time to propagate across the internet. To determine if a change has propagated, you can call the `DescribeCustomRoutingAccelerator` API action to check the accelerator status. For more information, see DescribeCustomRoutingAccelerator.

AWS CloudFormation is not supported

AWS CloudFormation is not supported for custom routing accelerators.

Custom routing accelerators in AWS Global Accelerator

A *custom routing accelerator* in AWS Global Accelerator lets you use custom application logic to direct one or more users to a specific destination among many destinations, while using the AWS global network to improve the availability and performance of your application.

A custom routing accelerator routes traffic only to ports on Amazon EC2 instances that are running in virtual private cloud (VPC) subnets. With a custom routing accelerator, Global Accelerator does not route traffic based on the geoproximity or health of the endpoint. To learn more, see How custom routing accelerators work in Global Accelerator.

When you create an accelerator, by default, Global Accelerator provides you with a set of two static IPv4 addresses. Custom routing accelerators support only the IPv4 IP address type. If you bring your own IP address range to AWS (BYOIP), you can assign static IPv4 addresses from your own pool to use with your accelerator. For more information, see Bring your own IP addresses (BYOIP) in Global Accelerator.

Important

The IP addresses are assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you *delete* an accelerator, you lose the Global Accelerator static IP addresses that are assigned to the accelerator, so you can no longer route traffic by using them. As a best practice, ensure that you have permissions in place to avoid inadvertently deleting accelerators. You can use IAM policies such as tag-based permissions with Global Accelerator to limit the users who have permissions to delete an accelerator. For more information, see ABAC with Global Accelerator.

This section explains how to work with a custom routing accelerator on the Global Accelerator console. To learn about using API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

Contents

- Create a custom routing accelerator in Global Accelerator
- Edit a custom routing accelerator in Global Accelerator
- View custom routing accelerators in Global Accelerator
- Delete a custom routing accelerator in Global Accelerator

Create a custom routing accelerator in Global Accelerator

This section provides steps for how to create a custom accelerator on the console. To work with Global Accelerator programmatically, see the AWS Global Accelerator API Reference.

To create a custom routing accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. Choose **Create accelerator**.
3. Provide a name for your accelerator.
4. For **Accelerator type**, select **Custom routing**.

5. Optionally, if you have brought your own IP address range to AWS (BYOIP), you can specify static IP addresses for your accelerator from that address pool. Make this choice for each of the two static IP addresses for your accelerator.

   - For each static IP address, choose the IP address pool to use.
   - If you chose your own IP address pool, also choose a specific IP address from the pool. If you chose the default Amazon IP address pool, Global Accelerator assigns a specific IP address to your accelerator.
6. Optionally, add one or more tags to help you identify your accelerator resources.
7. Choose **Next** to go to the next pages in the wizard to add listeners, endpoint groups, and VPC subnet endpoints.

Edit a custom routing accelerator in Global Accelerator

This section provides steps for how to update a custom accelerator on the console. To work with Global Accelerator programmatically, see the AWS Global Accelerator API Reference.

To edit a custom routing accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. In the list of custom routing accelerators, choose one, and then choose **Edit**.
3. On the **Edit accelerator** page, make any changes that you like. For example, you can disable the accelerator so that you can delete it.
4. Choose **Save**.

View custom routing accelerators in Global Accelerator

This section provides steps to view information about your custom routing accelerators on the console. To see descriptions of your custom routing accelerators programmatically, see ListCustomRoutingAccelerator and DescribeCustomRoutingAccelerator in the AWS Global Accelerator API Reference.

To view information about your custom routing accelerators

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.

2. To see details about an accelerator, choose an accelerator, and then choose **View**.

Delete a custom routing accelerator in Global Accelerator

If you created a custom routing accelerator as a test, or if you're no longer using an accelerator, you can delete it. On the console, disable the accelerator, and then you can delete it. You don't have to remove listeners and endpoint groups from the accelerator.

To delete a custom routing accelerator by using an API operation instead of the console, you must first remove all listeners and endpoint groups that are associated with the accelerator, and then disable it. For more information, see the DeleteAccelerator operation in the *AWS Global Accelerator API Reference*.

To disable a custom routing accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. In the list, choose an accelerator that you want to disable.
3. Choose **Edit**.
4. Choose **Disable accelerator**, and then choose **Save**.

To delete a custom routing accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. In the list, choose an accelerator that you want to delete.
3. Choose **Delete**.

> **Note**
>
> If you haven't disabled the accelerator, **Delete** is unavailable. To disable the accelerator, see the previous procedure.

4. In the confirmation dialog box, choose **Delete**.

> **Important**
>
> When you delete an accelerator, you lose the static IP addresses that are assigned to the accelerator, so you can no longer route traffic by using them.

Listeners for custom routing accelerators in Global Accelerator

For a custom routing accelerator in AWS Global Accelerator, you configure a listener that specifies a range of listener ports with associated protocols that Global Accelerator maps to specific destination Amazon EC2 instances in your VPC subnet endpoints. When you add a VPC subnet endpoint, Global Accelerator creates a static port mapping between the port ranges that you define for your listener and the destination IP addresses and ports in the subnet. Then you can use the port mapping to specify your accelerator static IP addresses together with a listener port and protocol to direct user traffic to specific destination Amazon EC2 instance IP addresses and ports in your VPC subnet.

You define a listener when you create your custom routing accelerator, and you can add more listeners at any time. Each listener can have one or more endpoint groups, one for each AWS Region in which you have VPC subnet endpoints. A listener in a custom routing accelerator supports both TCP and UDP protocols. You specify the protocol or protocols for each destination port range that you define: UDP, TCP, or both UDP and TCP.

For more information, see How custom routing accelerators work in Global Accelerator.

Contents

- Add a listener for a custom routing accelerator in Global Accelerator
- Edit a listener for a custom routing accelerator in Global Accelerator
- Remove a listener for a custom routing accelerator in Global Accelerator

Add a listener for a custom routing accelerator in Global Accelerator

This section explains how to add a listener for a custom routing accelerator on the AWS Global Accelerator console. To learn about using API operations with AWS Global Accelerator, see the AWS Global Accelerator API Reference.

To add a listener for a custom routing accelerator

The range that you specify when you create a listener defines how many listener port and destination IP address combinations that you can use with your custom routing accelerator. For maximum flexibility, we recommend that you specify a large port range. Each listener port range that you specify must include a minimum of 16 ports.

> **Note**
>
> After you create a listener, you can edit it to add additional port ranges and associated protocols, but you can't decrease existing port ranges.

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. On the **Accelerators** page, choose a custom routing accelerator.
3. Choose **Add listener**.
4. On the **Add listener** page, enter the listener port range that you want to associate with the accelerator.

   Listeners support ports 1-65535. For maximum flexibility with a custom routing accelerator, we recommend that you specify a large port range.
5. Choose **Add listener**.

Edit a listener for a custom routing accelerator in Global Accelerator

This section explains how to edit a listener for a custom routing accelerator on the AWS Global Accelerator console. To learn about using API operations with AWS Global Accelerator, see the AWS Global Accelerator API Reference.

To edit a listener for a custom routing accelerator

When you edit a listener for a custom routing accelerator, be aware that you can add additional port ranges and associated protocols, increase existing port ranges, or change protocols, but you can't decrease existing port ranges.

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.

2. On the **Accelerators** page, choose an accelerator.
3. Choose a listener, and then choose **Edit listener**.
4. On the **Edit listener** page, make the changes that you want to existing port ranges or protocols, or add new port ranges.

   Be aware that you cannot decrease the range of an existing port range.
5. Choose **Save**.

Remove a listener for a custom routing accelerator in Global Accelerator

This section explains how to remove a listener for a custom routing accelerator on the AWS Global Accelerator console. To learn about using API operations with AWS Global Accelerator, see the AWS Global Accelerator API Reference.

To remove a listener

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. On the **Accelerators** page, choose an accelerator.
3. Choose a listener, and then choose **Remove**.
4. In the confirmation dialog box, choose **Remove**.

Endpoint groups for custom routing accelerators in Global Accelerator

With a custom routing accelerator in AWS Global Accelerator, an endpoint group defines the ports and protocols that destination Amazon EC2 instances in your virtual private cloud (VPC) subnets accept traffic on.

You create an endpoint group for your custom routing accelerator for each AWS Region in which your VPC subnets and EC2 instances are located. Each endpoint group in a custom routing accelerator can have multiple VPC subnet endpoints. Similarly, you can add each VPC to multiple endpoint groups, but the endpoint groups must be associated with different listeners.

For each endpoint group, you specify a set of one or more port ranges that include the ports that you want to direct traffic to on the EC2 instances in the Region. For each endpoint group port range, you specify the protocol to use: UDP, TCP, or both UDP and TCP. This provides maximum flexibility for you, without having to duplicate sets of port ranges for each protocol. For example, you might have a game server with gaming traffic running over UDP on ports 8080-8090 while you also have a server listening for chat messages over TCP on port 80.

To learn more, see How custom routing accelerators work in Global Accelerator.

Contents

- Add an endpoint group for a custom routing accelerator in Global Accelerator
- Edit an endpoint group for a custom routing accelerator in Global Accelerator
- Remove an endpoint group for a custom routing accelerator in Global Accelerator

Add an endpoint group for a custom routing accelerator in Global Accelerator

You work with an endpoint group for your custom routing accelerator on the AWS Global Accelerator console or by using an API operation. You can add or remove VPC subnet endpoints from an endpoint group at any time.

This section explains how to create endpoint groups for your custom routing accelerator on the AWS Global Accelerator console. To learn about using API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

To add an endpoint group for a custom routing accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. On the **Accelerators** page, choose a custom routing accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of the listener that you want to add an endpoint group to.
4. Choose **Add endpoint group**.
5. In the section for a listener, specify a Region for the endpoint group.
6. For **Ports and protocols sets**, enter port ranges and protocols for your Amazon EC2 instances.

- Enter a **From port** and a **To port** to specify a range of ports.
- For each port range, specify the protocol or protocols for that range.

The port range doesn't have to be a subset of your listener port range, but there must be enough total ports in the listener port range to support the total number of ports that you specify for the endpoint groups in your custom routing accelerator.

7. Choose **Save**.
8. Optionally, choose **Add endpoint group** to add additional endpoint groups for this listener. You can also choose another listener and add endpoint groups.
9. Choose **Add endpoint group**.

Edit an endpoint group for a custom routing accelerator in Global Accelerator

You work with an endpoint group for your custom routing accelerator on the AWS Global Accelerator console or by using an API operation. You can add or remove VPC subnet endpoints from an endpoint group at any time.

This section explains how to edit endpoint groups for your custom routing accelerator on the AWS Global Accelerator console. To learn about using API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

To edit an endpoint group for a custom routing accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. On the **Accelerators** page, choose a custom routing accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of the listener that the endpoint group is associated with.
4. Choose **Edit endpoint group**.
5. On the **Edit endpoint group** page, change the Region, the range of ports, or the protocol for a range of ports.
6. Choose **Save**.

Remove an endpoint group for a custom routing accelerator in Global Accelerator

You work with an endpoint group for your custom routing accelerator on the AWS Global Accelerator console or by using an API operation.

This section explains how to remove endpoint groups for your custom routing accelerator on the AWS Global Accelerator console. To learn about using API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

To remove a custom routing accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/globalaccelerator/home#GlobalAcceleratorHome:.
2. On the **Accelerators** page, choose an accelerator.
3. In the **Listeners** section, choose a listener, and then choose **Remove**.
4. In the **Endpoint groups** section, choose an endpoint group, and then choose **Remove**.
5. On the confirmation dialog box, choose **Remove**.

Amazon VPC subnet endpoints for custom routing accelerators in Global Accelerator

Endpoints for custom routing accelerators are Amazon Virtual Private Cloud (VPC) subnets that can receive traffic through an accelerator. Each subnet can contain one or many Amazon EC2 instance destinations. When you add a subnet endpoint, Global Accelerator generates new port mapping. Then, you can use the Global Accelerator API to get a static list of all the port mappings for the subnet, which you can use to route traffic to destination EC2 instance IP addresses in the subnet. For more information, see ListCustomRoutingPortMappings.

Be aware of the following when you add VPC subnets and destinations for your custom routing accelerator:

- You can only direct traffic to EC2 instances in the subnets, not other resources, like load balancers (in contrast to standard accelerators).
- An EC2 instance destination in a subnet endpoint can't be one of the following types: C1, CC1, CC2, CG1, CG2, CR1, CS1, G1, G2, HI1, HS1, M1, M2, M3, or T1.

- By default, traffic directed through a custom routing accelerator can't arrive at any destinations in your subnet. To enable destination instances to receive traffic, you must choose to allow all traffic to the subnet or, alternatively, enable traffic to specific instance IP addresses and ports (destination sockets) in the subnet.

  > **Important**
  >
  > Updating a subnet or specific destination to allow or deny traffic takes time to propagate across the internet. To determine if a change has propagated, you can use the `DescribeCustomRoutingAccelerator` API action to check the accelerator status. For more information, see DescribeCustomRoutingAccelerator.

- Because VPC subnets preserve the client IP address, you should review the relevant security and configuration information when you add subnets as endpoints for custom routing accelerators. For more information, see Requirements for endpoints with client IP address preservation.
- When you configure resources as endpoints behind Global Accelerator, we recommend that you don't also send traffic directly to the same endpoints over the internet. Sending direct traffic can lead to connection collision issues.

To learn more, see How custom routing accelerators work in Global Accelerator.

Contents

- Add a VPC subnet endpoint for a custom routing accelerator
- Edit a VPC subnet endpoint for a custom routing accelerator
- Remove a VPC subnet endpoint for a custom routing accelerator

Add a VPC subnet endpoint for a custom routing accelerator

You add Amazon Virtual Private Cloud (VPC) subnet endpoints to endpoint groups in your custom routing accelerators so that you can direct user traffic to destination Amazon EC2 instances in the subnet.

When you add and remove EC2 instances from the subnet, or enable or disable traffic to EC2 destinations, you change whether those destinations can receive traffic. However the Global Accelerator port mapping doesn't change.

To allow traffic to some destinations in the subnet, but not all, enter IP addresses for each EC2 instance that you want to allow, along with the ports on the instance that you want to receive traffic. The IP addresses that you specify must be for EC2 instances in the subnet. You can specify a port or range of ports, from the ports that are mapped for the subnet.

You can remove the VPC subnet from your accelerator by removing it from an endpoint group. Removing a subnet doesn't affect the subnet itself, but Global Accelerator can no longer direct traffic to the subnet or to the Amazon EC2 instances in it. In addition, Global Accelerator will reclaim the port mapping for the VPC subnet to potentially use them for new subnets that you add.

The steps in this section explain how to add VPC subnet endpoints on the AWS Global Accelerator console. To learn about using API operations with AWS Global Accelerator, see the AWS Global Accelerator API Reference.

To add a VPC subnet endpoint

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. On the **Accelerators** page, choose a custom routing accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of a listener.
4. In the **Endpoint groups** section, for **Endpoint group ID**, choose the ID of the endpoint group (AWS Region) that you want to add the VPC subnet endpoint to.
5. In the **Endpoints** section, choose **Add endpoint**.
6. On the **Add endpoints** page, for **Endpoint**, choose a VPC subnet.

   If you don't have any VPCs, there aren't any items in the list. To continue, add at least one VPC, then come back to the steps here, and choose a VPC from the list.
7. For VPC subnet endpoint that you add, you can choose to allow or deny traffic to all destinations in the subnet, or you can allow traffic to only specific EC2 instances and ports. The default is to deny traffic to all destinations in the subnet.
8. Choose **Add endpoint**.

Edit a VPC subnet endpoint for a custom routing accelerator

You can edit Amazon Virtual Private Cloud (VPC) subnet endpoints for your custom routing accelerators so that you can change where you direct user traffic to destination Amazon EC2 instances, or allow or deny traffic to all destinations in the subnet.

When you add and remove EC2 instances from the subnet, or enable or disable traffic to EC2 destinations, you change whether those destinations can receive traffic. However the Global Accelerator port mapping doesn't change.

The steps in this section explain how to edit VPC subnet endpoints on the AWS Global Accelerator console. To learn about using API operations with AWS Global Accelerator, see the AWS Global Accelerator API Reference.

To allow or deny traffic to specific destinations

You can edit the subnet port mapping for a VPC endpoint to allow or deny traffic to specific EC2 instances and ports (destination sockets) in a subnet.

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. On the **Accelerators** page, choose a custom routing accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of a listener.
4. In the **Endpoint groups** section, for **Endpoint group ID**, choose the ID of the endpoint group (AWS Region) of the VPC subnet endpoint that you want to edit.
5. Choose an endpoint subnet, and then choose **View details**.
6. On the **Endpoint** page, under **Port mappings**, choose an IP address, and then choose **Edit**.
7. Enter the ports that you want to enable traffic for, and then choose **Allow these destinations**.

To allow or deny ALL traffic to a subnet

You can update an endpoint to allow or deny traffic to all destinations in the VPC subnet.

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. On the **Accelerators** page, choose a custom routing accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of a listener.

4. In the **Endpoint groups** section, for **Endpoint group ID**, choose the ID of the endpoint group (AWS Region) of the VPC subnet endpoint that you want to update.
5. Choose **Allow/Deny all traffic**.
6. Choose an option, to allow all traffic or deny all traffic, and then choose **Save**.

Remove a VPC subnet endpoint for a custom routing accelerator

You can remove an Amazon Virtual Private Cloud (VPC) subnet endpoint from your custom routing accelerator so that user traffic no longer goes to destination Amazon EC2 instances in the subnet.

The steps in this section explain how to remove a VPC subnet endpoint on the AWS Global Accelerator console. To learn about using API operations with AWS Global Accelerator, see the AWS Global Accelerator API Reference.

To remove an endpoint

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. On the **Accelerators** page, choose a custom routing accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of a listener.
4. In the **Endpoint groups** section, for **Endpoint group ID**, choose the ID of the endpoint group (AWS Region) of the VPC subnet endpoint that you want to remove.
5. Choose **Remove endpoint**.
6. In the confirmation dialog box, choose **Remove**.

Configure cross-account access in Global Accelerator

By using cross-account support, you can use AWS Global Accelerator as a fixed entry point to your application that accesses resources in multiple accounts, or choose IP addresses for your accelerator from shared CIDR blocks. Using cross-account permissions for allowing access to resources in different accounts is an AWS best practice. With cross-account support for bring your own IP (BYOIP) address CIDR blocks, you can use the same address pool for accelerators in different accounts in your organization. You can also organize AWS resources under one account that controls internet access to your applications, which can simplify monitoring and security, as well as provide visibility to inbound connections.

Cross-account support in Global Accelerator enables you to do the following:

- Add endpoints, such as Network Load Balancers, from other accounts to an accelerator.
- Choose a BYOIP address pool for IP addresses, and then select IP addresses from the pool for accelerators in different accounts. By sharing a BYOIP address pool, you can use more addresses from the same CIDR block, reducing the number of CIDR blocks that you require.

You can work with cross-account attachments and resources in the Global Accelerator console, or by using Global Accelerator API operations with the AWS Command Line Interface (AWS CLI) or an AWS SDK. For example, as a principal, you can use the UpdateEndpoints operation to add a cross-account resource as an endpoint for an accelerator. When you use the API operation, you specify the cross-account attachment ARN and the endpoint ID. For more information, see the AWS Global Accelerator API Reference Guide.

Contents

- How cross-account works in Global Accelerator
- Work with cross-account attachments in Global Accelerator
- Work with cross-account resources in Global Accelerator
- Identify your cross-account resources in Global Accelerator
- Responsibilities and permissions for cross-account resources in Global Accelerator
- Billing costs for cross-account resources in Global Accelerator
- Quotas for cross-account resources in Global Accelerator

How cross-account works in Global Accelerator

With cross-account support in Global Accelerator, resource owners control whether their resources are shared with accelerators owned by other accounts. To enable resource sharing for your resources, you—as a resource owner—create a Global Accelerator *cross-account attachment* to authorize resources in your account to be added to an accelerator by another account.

You create the cross-account attachment in Global Accelerator. The attachment lists the *resources* that you want to share, and the *principals*—other accounts or specific accelerator ARNs— that are authorized to use the resources. Resources can be AWS resources, like Network Load Balancers, that you add as endpoints to accelerator endpoint groups, or resources can be IP address ranges that you've brought to Global Accelerator with the bring your own IP address (BYOIP) process.

> **Important**
>
> Before you can add a BYOIP IP address range to a cross-account attachment to share with principals, you must complete the process to *provision* and *advertise* the address range. For more information, see Bring your own IP addresses (BYOIP) in Global Accelerator.

After you, as a resource owner, create an attachment, principals listed in the attachment can work with resources that are listed in the attachment. That is, they can add as endpoints AWS resources that are listed, or select as a static IP address a BYOIP address from CIDR prefixes that are listed. When a principal wants to add a cross-account resource for an accelerator, they must specify the cross-account attachment that authorizes them as a principal with permission to use the resource.

Work with cross-account attachments in Global Accelerator

To allow someone to add a resource from another account as an endpoint or a BYOIP address for an accelerator, the owner of the resource must create a *cross-account attachment* in Global Accelerator. In the attachment, the resource owner specifies one or more accelerators or accounts—principals— that are allowed to add resources, along with the specific resources that the principals can add to accelerators.

As a resource owner, be aware that to specify a resource in a cross-account attachment, you must own the resource in your AWS account. That is, the resource must be allocated or provisioned in your account; you cannot specify a resource that has been shared with *you*, such as a shared subnet.

Contents

- Create a cross-account attachment in AWS Global Accelerator
- Edit a cross-account attachment in AWS Global Accelerator
- Delete a cross-account attachment in Global Accelerator

Create a cross-account attachment in AWS Global Accelerator

Follow the steps in this section to create a cross-account attachment using the AWS Global Accelerator console.

This section explains how to create a cross-acount attachment by using the AWS Global Accelerator console. To learn about using API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

To create a cross-account attachment

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. Choose **Create cross-account attachment**.
3. On the **Create cross-account attachment** page, enter a name for the attachment.
4. Add the AWS accounts or the ARNs for the accelerators, or both, that you want to allow to add your resources.
5. Select the resources that you want to allow to be used. For example, to add resources that can added as endpoints, for each resource, choose an AWS Region. Then, from the drop-down menus, select an endpoint type (resource type) and the endpoint (resource) to add.
6. Choose **Create attachment**.

Note: To see the new cross-account attachment in your list of attachments, refresh the **Cross-account attachments** page.

Edit a cross-account attachment in AWS Global Accelerator

Follow the steps in this section to edit a cross-account attachment using the AWS Global Accelerator console.

This section explains how to edit a cross-acount attachment by using the AWS Global Accelerator console. To learn about using API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

You can edit a cross-account attachment to add or remove principals or resources, rename the attachment, or delete the attachment.

Be aware of the following when you remove principals or resources, or delete an attachment:

- To remove a principal or CIDR from an attachment, the principal must first remove shared IP addresses from all accelerators that use them. Then, you can remove the principal, or CIDRs, from the attachment.
- Before you can remove shared IP addresses or remove authorization for principals to access a shared CIDR from an attachment, the shared IP addresses for the CIDR must not be currently used by any accelerators.
- If you remove a principal from a cross-account attachment that enables the principal to add one or more shared endpoints, Global Accelerator removes those cross-account endpoints from any accelerator that uses that permission for cross-account resources listed in the attachment.
- If you remove an endpoint resource from a cross-account attachment, Global Accelerator removes the cross-account endpoint from any accelerator where it was added as an endpoint based on the permissions in the attachment.
- If you delete a cross-account attachment, Global Accelerator removes all cross-account endpoints listed in the attachment from all accelerators where the resources were added as endpoints based on the permissions in the attachment.
- If there are multiple cross-account attachments that include a principal, or that include a resource, Global Accelerator continues to allow the access that any existing attachment provides. So, for example, if you remove a principal from one attachment but the principal still has permission to access a resource that's granted by a second attachment, Global Accelerator continues to allow the principal access to the cross-account resource.

To edit a cross-account attachment

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. Choose **Cross-account attachments**.
3. Choose a cross-account attachment to update, and then choose **Edit**.

4. Modify the attachment to make the desired changes. For example, you can add or remove principals, rename the attachment, or add or remove resources.
5. Choose **Save changes**.

Delete a cross-account attachment in Global Accelerator

Follow the steps in this section to delete a cross-account attachment using the AWS Global Accelerator console.

This section explains how to delete a cross-account attachment by using the AWS Global Accelerator console. To learn about using API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

To delete a cross-account attachment

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. Choose **Cross-account attachments**.
3. Choose a cross-account attachment, and then choose **Delete**.
4. In the dialog box, type **delete** in the text box, to confirm that you want to delete the cross-account attachment.
5. Choose **Delete**.

Work with cross-account resources in Global Accelerator

If your account, or an accelerator that you have permission to access, is specified as a principal in a cross-account attachment in AWS Global Accelerator, you can use resources that have been shared with you from another account.

For example, you can select bring your own IP (BYOIP) addresses as static IP addresses when you create an accelerator, or you can add endpoints to accelerator endpoint groups for an accelerator. The resources that you can add must also be specified in the attachment.

The following sections include the steps to add or remove cross-account attachments in Global Accelerator.

Contents

- Add a cross-account BYOIP address in Global Accelerator
- Add cross-account endpoints in AWS Global Accelerator
- Remove a cross-account endpoint in Global Accelerator

Add a cross-account BYOIP address in Global Accelerator

Follow the steps in this section to configure cross-account bring your own IP (BYOIP) ID addresses using the Global Accelerator console.

This section explains how to use a BYOIP IP address by using the AWS Global Accelerator console. To learn about using API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

You can change the BYOIP addresses that you use for your accelerator, but some restrictions apply. For more information, see How to update an accelerator to change an IP address.

To use a cross-account BYOIP IP address

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. Choose **Create accelerator**.
3. Provide a name for your accelerator.
4. Select an **Accelerator type**.
5. For **IP address type**, select **IPv4**.
6. Select the **Use a static IP address from a CIDR authorized for cross-account** check box.
7. Select the account ID for the owner of the cross-account attachment that specifies you as a principal and that includes the BYOIP address block that has been shared with you.

   Note that because you must choose one account to select addresses from, if you select two BYOIP IP addresses when you create an accelerator, the IP addresses must have the same owner and be authorized in the same cross- account attachment.
8. Specify one or both static IP addresses for your accelerator.

   - For each static IP address, choose the IP address pool to use.

> **Note**
>
> You must choose a different IP address pool for each static IP address. This restriction is because Global Accelerator assigns each address range to a different network zone, for high availability.

- If you chose your own IP address pool, also choose a specific IP address from the pool. If you choose the default Amazon IP address pool, Global Accelerator assigns a specific IP address to your accelerator.

9. Optionally, add one or more tags to help you identify your accelerator resources.
10. Choose **Next** to add listeners, endpoint groups, and endpoints.

Add cross-account endpoints in AWS Global Accelerator

Follow the steps in this section to add a cross-account endpoints using the Global Accelerator console.

This section explains how to add cross-account endpoints by using the AWS Global Accelerator console. To learn about using API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

To add a cross-account endpoint

1. When you create or update an accelerator, in the **Endpoints** section, choose **Add endpoint**.
2. On the **Add endpoints** page, select **Add a resource specified in a cross-account attachment**.
3. In the drop-down menu, select an AWS account that has created a cross-account attachment that includes you or the accelerator as a principal.
4. For **Endpoint type**, choose the type of resource that you want to add.

   Note that only the resource types included in the cross-account attachment appear in the drop-down menu.
5. For **Endpoint**, choose resource that you want to add.

   Note that only resources that are included in the cross-account attachment appear in the drop-down menu. To see resources that are not enabled by a cross-account attachment, clear the **Add a resource specified in a cross-account attachment** check box.

Remove a cross-account endpoint in Global Accelerator

Follow the steps in this section to remove a cross-account endpoints using the Global Accelerator console.

This section explains how to remove cross-account endpoints by using the AWS Global Accelerator console. To learn about using API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

To remove a cross-account endpoint

1. When you create or update an accelerator, on the **Endpoint group** details page, choose the endpoint that you want to remove.
2. Choose **Remove**.

Identify your cross-account resources in Global Accelerator

Resource owners and principals can identify shared resources by using the AWS Global Accelerator console or by using the AWS CLI with Global Accelerator operations. For example, you can do the following:

- As an owner, you can see a list of your cross-account attachments, and view the principals and resources in each attachment.
- As a principal, you can view all cross-account attachments that you're listed in, and you can list the resources that you can add as endpoints or IP address ranges for an accelerator, for a specific attachment.

For more information about using API operations to view cross-account attachments and shared resources, see AWS Global Accelerator API Reference Guide.

As an owner: Identify your cross-account resources in Global Accelerator

As an owner, you can view your cross-account attachments in the AWS Management Console, or by using the AWS Command Line Interface with Global Accelerator API operations.

To see your cross-account attachments

- In the Global Accelerator console, choose **Cross-accounts attachments**.

To see the information included in a cross-account attachment

1. In the Global Accelerator console, on the **Cross-accounts attachments** page, choose an attachment, and then choose **View details**.

   —OR—

2. Use the API operation ListCrossAccountResources, for example, by using the AWS Command Line Interface. This operation returns a list of unique attachment-resource pairs, for every resource, in every attachment, in the account.

   For example, if you have two cross-account attachments, and the first includes two endpoints and a CIDR block, while the second includes three endpoints, `ListCrossAccountResources` returns six attachment-resource pairs: attachment1-endpoint1, attachment1-endpoint2, attachment1-CIDR, attachment2-endpoint3, attachment2-endpoint4, and attachment2-endpoint5.

As a principal: Identify your cross-account resources in Global Accelerator

As a principal, after you're authorized by a cross-account attachment to add a resource to an accelerator as an endpoint, there is no additional action to take before you can add a resource as an endpoint.

You can see the AWS accounts that have created a cross-account attachment that you're listed as a principal in. You can also see the resources specified in the attachment that each account has created, that you can add as endpoints or IP address ranges for an accelerator.

To see the accounts that have created a cross-account attachment that you're listed as a principal in

1. In the Global Accelerator console, on the **Endpoint details** page for an accelerator, choose **Add endpoint**.
2. On the **Add endpoints** page, select **Add a resource specified in a cross-account attachment**.

3. In the drop-down menu for **Select account ID of the cross-account attachment owner**, view the account or accounts that give you permission in a cross-account attachment to add resources to the accelerator.

To see the endpoint resources specified in the attachment that each account has created

1. In the Global Accelerator console, on the **Endpoint details** page for an accelerator, choose **Add endpoint**.
2. On the **Add endpoints** page, select **Add a resource specified in a cross-account attachment**.
3. In the drop-down menu, select an account that gives you permission in a cross-account attachment to add resources to the accelerator.
4. For **Endpoint type**, choose a type of resource.

   Note that only the resource types included in the cross-account attachment appear in the drop-down menu.
5. In the **Endpoint** drop-down menu is a list of the resources. These are the resources that you are authorized by the account that created the cross-account attachment to add as endpoints, for a specific resource type.
6. To see the resources that you can add that are specified in the cross-account attachment created by a different account, do the following: In the drop-down menu for **Select account ID of the cross-account attachment owner**, select a different AWS account.

To see the IP address resources specified in the attachment that an account has created

1. In the Global Accelerator console, choose **Create accelerator**.
2. On the **Enter name** page, for IP address type, select **IPv4**.
3. Under IP address pool selection, select **Use a shared IP address pool specified in a cross-account attachment**.
4. Select an account that gives you permission in a cross-account attachment to choose IP addresses from a shared IP address pool.
5. For **IP address pool**, in the drop-down list, you can view shared IP address pools.

   Note that only the shared IP address pools included in a cross-account attachment that you are permitted to use appear in the drop-down menu.

Responsibilities and permissions for cross-account resources in Global Accelerator

The following sections list the permissions you have as a resource owner or as a principal for cross-account access in AWS Global Accelerator.

Permissions for resource owners

When you, as a resource owner, authorize principals to add resources from your AWS account to their accelerators, or to a specific accelerator, principals can add any resources that you have listed in the cross-account attachment.

As a resource owner, you are responsible for creating, managing, and deleting your resources. You can't add or remove resources in accelerators unless you have a role that is authorized to do so.

If you have an accelerator and you need to add or remove cross-account resources, a principal can set up a role in IAM with permission to access the resources, and add your account to the role.

You can add or remove principals or resources from a cross-account attachment, to manage whether resources that you own are used as endpoints or shared IP address pools for accelerators.

Permissions for principals

In general, principals can add resources that are listed in a cross-account attachment for an accelerator that the attachment provides permission for. They can only view, add, or remove endpoints, or select shared IP addresses from BYOIP address pools, for the cross-account resources that they have permission for.

The following applies for principals:

- Principals can only view, add, or remove resources as endpoints or shared IP address pools for an accelerator that they have been granted permission for in a cross-account attachment.
- Principals can only modify resources, such as load balancers, that they own themselves. They cannot modify resources specified in a cross-account attachment, because the resources belong to the resource owner.

Although principals cannot modify the actual cross-account resources, based on a cross-account attachment, the resource owner can create an IAM role that provides permission to access the

resource. Then, the owner can grant a principal permissions to assume the role, so that the principal can access the resource, however the owner has specified through the role's permissions.

Billing costs for cross-account resources in Global Accelerator

The owner of an accelerator in AWS Global Accelerator is billed for costs associated with the accelerator. There are no additional costs, for accelerator owners or for resource owners, for adding cross-account resources as endpoints or as bring your own IP address (BYOIP) pools for an accelerator.

For more information about pricing, see Pricing for AWS Global Accelerator.

Quotas for cross-account resources in Global Accelerator

The following applies when you work with cross-account attachments and cross-account resources in AWS Global Accelerator:

- All cross-account resources, and other resources, that are added as endpoints for an accelerator —including resources added by all principals with cross-account permission—count toward quotas in effect for the accelerator.
- Quotas for accelerators are enforced for principals.
- Quotas for cross-account attachments in Global Accelerator are enforced for resource owners.

For more information about quotas, see Quotas for AWS Global Accelerator.

DNS addressing and custom domains in AWS Global Accelerator

This chapter explains how AWS Global Accelerator does DNS routing and includes information about using a custom domain with Global Accelerator. It also includes the steps for configuring bring your own IP (BYOIP) addresses to use with accelerators in Global Accelerator.

- **DNS addressing**: When you create an accelerator, Global Accelerator assigns a default Domain Name System (DNS) name to your accelerator.
- **Custom domain name**: You can configure DNS to use your custom domain name (such as `www.example.com`) with your accelerator, instead of using the assigned static IP addresses or the default DNS name.
- **BYOIP IP addresses**: You can bring your own IP addresses to AWS to add to an accelerator instead of, or together with, the static IP addresses that Global Accelerator assigns to you.

Contents

- Support for DNS addressing in AWS Global Accelerator
- Route custom domain traffic to your accelerator
- Bring your own IP addresses (BYOIP) in Global Accelerator

Support for DNS addressing in AWS Global Accelerator

When you create an accelerator with an IPv4 IP address type, Global Accelerator provisions two static IPv4 addresses for you. It also assigns a default Domain Name System (DNS) name to your accelerator, similar to `a1234567890abcdef.awsglobalaccelerator.com`, that points to the static IP addresses.

For accelerators with dual-stack IP address types, Global Accelerator provides a total of four addresses: two static IPv4 addresses and two static IPv6 addresses. Global Accelerator creates a new DNS name that points to both the A record and the AAAA record that points to all four IP addresses. The new DNS record enables Global Accelerator to upgrade an accelerator to dual-stack without affecting clients that currently reference the original DNS record that is not dual-stack. An example DNS name for an accelerator with dual-stack IP addresses is the following: `a1234567890abcdef.dualstack.awsglobalaccelerator.com`

The static addresses are advertised globally using anycast from the AWS edge network to your endpoints. You can use your accelerator's static addresses or DNS name to route traffic to your accelerator. DNS servers and DNS resolvers use the round-robin DNS process to resolve the DNS name for an accelerator, so the name resolves to the static IP addresses for the accelerator, returned by Amazon Route 53 in random order. Clients typically use the first IP address that is returned.

> **Note**
>
> For each IPv4 and IPv6 address associated with your accelerator, Global Accelerator creates a Pointer (PTR) record that maps an accelerator's static IP address to the corresponding DNS name generated by Global Accelerator, to support reverse DNS lookup. This is also known as a reverse hosted zone. Be aware that the DNS name that Global Accelerator generates for you isn't configurable, and you can't create PTR records that point to your custom domain name. Global Accelerator also does not create PTR records for static IP addresses from an IP address range that you bring to AWS (BYOIP).

Route custom domain traffic to your accelerator

In most scenarios, you can configure DNS to use your custom domain name (such as `www.example.com`) with your accelerator, instead of using the assigned static IP addresses or the default DNS name. First, using Amazon Route 53 or another DNS provider, create a domain name, and then add or update DNS records with your Global Accelerator IP addresses. Or you can associate your custom domain name with the DNS name for your accelerator. Complete the DNS configuration and wait for the changes to propagate over the internet. Now when a client makes a request using your custom domain name, the DNS server resolves it to the IP addresses, in random order, or to the DNS name for your accelerator.

To use your custom domain name with Global Accelerator when you use Route 53 as your DNS service, you create an alias record that points your custom domain name to the DNS name assigned to your accelerator. An alias record is a Route 53 extension to DNS. It's similar to a CNAME record, but you can create an alias record both for the root domain, such as `example.com`, and for subdomains, such as `www.example.com`. For more information, see Choosing Between Alias and Non-Alias Records in the Amazon Route 53 Developer Guide.

To set up Route 53 with an alias record for an accelerator, follow the guidance included in the *Value/route traffic to* section in the Values that are common for alias records for all routing policies

in the Amazon Route 53 Developer Guide. To see the information for Global Accelerator, scroll down to "AWS Global Accelerator accelerators".

Bring your own IP addresses (BYOIP) in Global Accelerator

You can bring part or all of your public IPv4 address ranges from your on-premises network to your AWS account to use with AWS Global Accelerator. You continue to own the address ranges, but AWS advertises them on the internet. BYOIP with IPv6 is not supported at this time.

Global Accelerator uses static IP addresses as entry points for your accelerators. These IP addresses are anycast from AWS edge locations. By default, Global Accelerator provides static IP addresses from the Amazon IP address pool. Instead of using the IP addresses that Global Accelerator provides, you can configure these entry points to be IPv4 addresses from your own address ranges. This topic explains how to use your own IP address ranges with Global Accelerator.

You can't use the IP addresses that you bring to AWS for one AWS service with another service. The steps in this chapter describe how to bring your own IP address range for use in AWS Global Accelerator only. For steps to bring your own IP address range for use in Amazon EC2, see Bring your own IP addresses (BYOIP) in the Amazon EC2 User Guide.

> **Important**
>
> You must stop advertising your IP address range from other locations before you advertise it through AWS. If an IP address range is multihomed (that is, the range is advertised by multiple service providers at the same time), we can't guarantee that traffic to the address range will enter our network or that your BYOIP advertising workflow will complete successfully.

After you bring an address range to AWS, it appears in your account as an address pool. When you create an accelerator, you can assign one IP address from your range to it. Global Accelerator assigns you a second static IP address from an Amazon IP address range. If you bring two IP address ranges to AWS, you can assign one IP address from each range to your accelerator. This restriction is because Global Accelerator assigns each address range to a different network zone, for high availability.

To use your own IP address range with Global Accelerator, review the requirements, and then follow the steps provided in this topic.

Contents

- Requirements
- Prepare to bring your IP address range to your AWS account: Authorization
- Provision the address range for use with Global Accelerator
- Advertise the address range through AWS
- Deprovision the address range
- Use your BYOIP address with an accelerator in Global Accelerator
- Update an accelerator to change your IP addresses

Requirements

You can bring up to two qualifying IP address ranges to AWS Global Accelerator per AWS account.

To qualify, your IP address range must meet the following requirements:

- The IP address range must be registered with one of the following regional internet registries (RIRs): the American Registry for Internet Numbers (ARIN), Réseaux IP Européens Network Coordination Centre (RIPE), or Asia-Pacific Network Information Centre (APNIC). The address range must be registered to a business or institutional entity. It can't be registered to an individual.
- The only address range that you can bring is /24. The first 24 bits of the IP address specify the network number. For example, 198.51.100 is the network number for IP address 198.51.100.0.
- The IP addresses in the address range must have a clean history. That is, they can't have a poor reputation or be associated with malicious behavior. We reserve the right to reject the IP address range if we investigate the reputation of the IP address range and find that it contains an IP address that doesn't have a clean history.

Also, we require the following allocation and assignment network types or statuses, depending on where you registered your IP address range:

- ARIN: `Direct Allocation` and `Direct Assignment` network types
- RIPE: `ALLOCATED PA`, `LEGACY`, and `ASSIGNED PI` allocation statuses
- APNIC: `ALLOCATED PORTABLE` and `ASSIGNED PORTABLE` allocation statuses

Prepare to bring your IP address range to your AWS account: Authorization

To ensure that only you can bring your IP address space to Amazon, we require two authorizations:

- You must authorize Amazon to advertise the IP address range.
- You must provide proof that you own the IP address range and so have the authority to bring it to AWS.

> **Note**
>
> When you use BYOIP to bring an IP address range to AWS, you can't transfer ownership of that address range to a different account or company while we're advertising it. You also can't directly transfer an IP address range from one AWS account to another account. To transfer ownership or to transfer between AWS accounts, you must deprovision the address range, and then the new owner must follow the steps to add the address range to their AWS account.

To authorize Amazon to advertise the IP address range, you provide Amazon with a signed authorization message. Use a Route Origin Authorization (ROA) to provide this authorization. A ROA is a cryptographic statement about your route announcements that you create through your Regional Internet Registry (RIR). A ROA contains the IP address range, the Autonomous System Numbers (ASN) that are allowed to advertise the IP address range, and an expiration date. The ROA authorizes Amazon to advertise an IP address range under a specific Autonomous System (AS).

A ROA does not authorize your AWS account to bring the IP address range to AWS. To provide this authorization, you must publish a self-signed X.509 certificate in the Registry Data Access Protocol (RDAP) remarks for the IP address range. The certificate contains a public key, which AWS uses to verify the authorization-context signature that you provide. Keep your private key secure and use it to sign the authorization-context message.

The following sections provide detailed steps for completing these authorization tasks. The commands in these steps are supported on Linux. If you use Windows, you can access the Windows Subsystem for Linux to run Linux commands.

Steps to provide authorization

- Step 1: Create a ROA object
- Step 2: Create a self-signed X.509 certificate
- Step 3: Create a signed authorization message

Step 1: Create a ROA object

Create a ROA object to authorize Amazon ASN 16509 to advertise your IP address range as well as the ASNs that are currently authorized to advertise the IP address range. The ROA must contain the /24 IP address that you want to bring to AWS and you must set the maximum length to /24.

For more information about creating a ROA request, see the following sections, depending on where you registered your IP address range:

- ARIN: ROA Requests
- RIPE: Managing ROAs
- APNIC: Route Management

Step 2: Create a self-signed X.509 certificate

Create a key pair and a self-signed X.509 certificate, and then add the certificate to the RDAP record for your RIR. The following steps describe how to perform these tasks.

> **Note**
>
> The `openssl` commands in these steps require OpenSSL version 1.0.2 or later.

To create and add an X.509 certificate

1. Generate an RSA 2048-bit key pair using the following command.

```
openssl genrsa -out private.key 2048
```

2. Create a public X.509 certificate from the key pair using the following command.

```
openssl req -new -x509 -key private.key -days 365 | tr -d "\n" > publickey.cer
```

In this example, the certificate expires in 365 days, after which time it can't be trusted. When you run the command, make sure that you set the `–days` option to the desired value for the correct expiration. When you're prompted for other information, you can accept the default values.

3. Update the RDAP record for your RIR with the X.509 certificate by using the following steps, depending on your RIR.

   1. View your certificate using the following command.

      ```
      cat publickey.cer
      ```

   2. Add the certificate that you previously created to the RDAP record for your RIR. Be sure to include the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` strings before and after the encoded portion. All of this content must be on a single, long line. The procedure for updating RDAP depends on your RIR:

      - For ARIN, use the Account Manager portal to add the certificate in the "Public Comments" section for the "Network Information" object representing your address range. Do not add it to the comments section for your organization.
      - For RIPE, add the certificate as a new "descr" field to the "inetnum" or "inet6num" object representing your address range. These can usually be found in the "My Resources" section of the RIPE Database portal. Do not add it to the comments section for your organization or the "remarks" field of the above objects.
      - For APNIC, email the certificate to helpdesk@apnic.net to manually add it to the "remarks" field for your address range. Send the email using the APNIC authorized contact for the IP addresses.

      You can remove the certificate from your RIR's record after the provisioning stage below has been completed.

Step 3: Create a signed authorization message

Create the signed authorization message to allow Amazon to advertise your IP address range.

The format of the message is as follows, where the `YYYYMMDD` date is the expiration date of the message.

```
1|aws|aws-account|address-range|YYYYMMDD|SHA256|RSAPSS
```

To create the signed authorization message

1. Create a plaintext authorization message and store it in a variable named `text_message`, as the following example shows. Replace the example account number, IP address range, and expiration date with your own values.

   ```
   text_message="1|aws|123456789012|203.0.113.0/24|20191201|SHA256|RSAPSS"
   ```

2. Sign the authorization message in `text_message` using the key pair that you created in the previous section.
3. Store the message in a variable named `signed_message`, as the following example shows.

   ```
   signed_message=$(echo $text_message | tr -d "\n" | openssl dgst -sha256 -sigopt
         rsa_padding_mode:pss -sigopt rsa_pss_saltlen:-1 -sign private.key -keyform
    PEM | openssl base64 |
         tr -- '+=/' '-_~' | tr -d "\n")
   ```

Provision the address range for use with Global Accelerator

When you provision an address range for use with AWS, you are confirming that you own the address range and authorize Amazon to advertise it. We'll verify that you own the address range.

You must provision your address range using the CLI or Global Accelerator API operations. This functionality is not available in the AWS console.

To provision the address range, use the following ProvisionByoipCidr command. The `--cidr-authorization-context` parameter uses the variables that you created in the previous section, not the ROA message.

```
aws globalaccelerator --region us-west-2 provision-byoip-cidr --cidr address-range --
cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

The following is an example of provisioning an address range.

```
aws globalaccelerator --region us-west-2 provision-byoip-cidr
    --cidr 203.0.113.0/24
```

```
    --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

Provisioning an address range is an asynchronous operation, so the call returns immediately. However, the address range is not ready to use until its state changes from `PENDING_PROVISIONING` to `READY`. It can take up to 3 weeks to complete the provisioning process. To monitor the state of the address ranges that you've provisioned, use the following ListByoipCidrs command:

```
aws globalaccelerator --region us-west-2 list-byoip-cidrs
```

To see a list of the states for an IP address range, see ByoipCidr.

When your IP address range is provisioned, the `State` returned by `list-byoip-cidrs` is `READY`. For example:

```
{
    "ByoipCidrs": [
        {
            "Cidr": "203.0.113.0/24",
            "State": "READY"
        }
    ]
}
```

Advertise the address range through AWS

After the address range is provisioned, it's ready to be advertised. You must advertise the exact address range that you provisioned. You can't advertise only a portion of the provisioned address range. In addition, you must stop advertising your IP address range from other locations before you advertise it through AWS.

You must advertise (or stop advertising) your address range using the CLI or Global Accelerator API operations. This functionality is not available in the AWS console.

> **Important**
>
> Make sure that your IP address range is advertised by AWS before you use an IP address from your pool with Global Accelerator.

To advertise the address range, use the following AdvertiseByoipCidr command.

```
aws globalaccelerator --region us-west-2 advertise-byoip-cidr --cidr address-range
```

The following is an example of requesting Global Accelerator to advertise an address range.

```
aws globalaccelerator --region us-west-2 advertise-byoip-cidr --cidr 203.0.113.0/24
```

To monitor the state of the address ranges that you've advertised, use the following ListByoipCidrs command.

```
aws globalaccelerator --region us-west-2 list-byoip-cidrs
```

When your IP address range is advertised, the `State` returned by `list-byoip-cidrs` is `ADVERTISING`. For example:

```
{
    "ByoipCidrs": [
        {
            "Cidr": "203.0.113.0/24",
            "State": "ADVERTISING"
        }
    ]
}
```

To stop advertising the address range, use the following `withdraw-byoip-cidr` command.

Important

To stop advertising your address range, you first must remove any accelerators that have static IP addresses that are allocated from the address pool. To delete an accelerator using the console or using API operations, see Delete accelerator.

```
aws globalaccelerator --region us-west-2 withdraw-byoip-cidr --cidr address-range
```

The following is an example of requesting Global Accelerator to withdraw an address range.

```
aws globalaccelerator --region us-west-2 withdraw-byoip-cidr
    --cidr 203.0.113.0/24
```

Deprovision the address range

To stop using your address range with AWS, you first must remove any accelerators that have static IP addresses that are allocated from the address pool and stop advertising your address range. After you complete those steps, you can deprovision the address range.

You must stop advertising and deprovision your address range using the CLI or Global Accelerator API operations. This functionality is not available in the AWS console.

Step 1: Delete any associated accelerators. To delete an accelerator using the console or using API operations, see Delete accelerator.

Step 2. Stop advertising the address range. To stop advertising the range, use the following WithdrawByoipCidr command.

```
aws globalaccelerator --region us-west-2 withdraw-byoip-cidr --cidr address-range
```

Step 3. Deprovision the address range. To deprovision the range, use the following DeprovisionByoipCidr command.

```
aws globalaccelerator --region us-west-2 deprovision-byoip-cidr --cidr address-range
```

Use your BYOIP address with an accelerator in Global Accelerator

After you complete the steps to add an address range with BYOIP, you can create an accelerator with your BYOIP IP addresses, or you can use your BYOIP IP addresses with an existing accelerator. If you brought one address range to AWS, you can assign one IP address to your accelerator. If you brought two address ranges, you can assign one IP address from each address range to your accelerator.

You can also update an existing accelerator to use one or more of your BYOIP IP addresses. For more information, see Update an accelerator to change your IP addresses

Another option is to use a shared BYOIP address. If one or more additional CIDR addresses have been shared with you from another account, you can choose from a shared BYOIP CIDR when you select one or both BYOIP IP addresses. Note that if you choose to use two shared BYOIP addresses, they must both come from CIDRs owned by the same account. For more information, see Configure cross-account access in Global Accelerator.

You have several options for creating an accelerator using your own IP addresses for the static IP addresses:

- **Use Global Accelerator console to create an accelerator.** For more information, see the following:
  - Create accelerator
  - Create a custom routing accelerator in Global Accelerator
  - Add cross-account endpoints in AWS Global Accelerator
- **Use the Global Accelerator API to create an accelerator.** For more information, including examples of using the CLI, see the following in the AWS Global Accelerator API Reference:
  - CreateAccelerator
  - CreateCustomRoutingAccelerator

Update an accelerator to change your IP addresses

After you assign BYOIP addresses as static IP addresses for an accelerator in AWS Global Accelerator, you can update the accelerator later to use different IP addresses from your address ranges. You can also update an accelerator that uses your own IP addresses to instead use IP addresses provided by AWS Global Accelerator.

After an Amazon-owned static IP address is changed, you can revert to the original static IP address, but you must do so *within 10 days* of when it was changed. After 10 days, the original static IP address is returned to the Amazon IP address pool and reused. After that, if you update your accelerator to change a BYOIP address to a Global Accelerator-assigned IP address, you are assigned a new IP address from the Amazon IP address pool. To learn more about reverting your IP address, see Revert a static IP address change.

The following sections provide information about how to change IP addresses when you use bring your own IP address (BYOIP) with Global Accelerator, and list the requirements and things to know when you change static IP addresses.

How to update an accelerator to change an IP address

To change an IP address for an accelerator, edit the accelerator and then, under **IP addresses**, select a new IP address. Your options for whether you can select an address from your own BYOIP address pool or the Amazon IP address pool depend on what your accelerator already has for static IP addresses, and other factors.

Make sure that you review the requirements and things to be aware of for changing accelerator static IP addresses before you begin.

The following topics provide procedures for updating accelerators.

- **Use Global Accelerator console to update an accelerator.** For more information, see the following:
  - Update accelerator
  - Edit a custom routing accelerator in Global Accelerator
- **Use the Global Accelerator API to update an accelerator.** For more information, including examples of using the CLI, see the following in the AWS Global Accelerator API Reference:
  - UpdateAccelerator
  - UpdateCustomRoutingAccelerator

Requirements when you update an accelerator to change IP addresses

When you update an accelerator to change one or both static IP addresses, keep in mind the following:

- You can change the BYOIP address for both standard accelerators and custom routing accelerators. After you create an accelerator with one or two BYOIP addresses, that accelerator must always have at least one BYOIP address. However, you can update the accelerator to change one or both static IP addresses to use a BYOIP addresses or to change the BYOIP address
- If you have an accelerator with two BYOIP static IP addresses, you can change only one of them to use a static IP address assigned by Global Accelerator. Note the following about changing a BYOIP static IP address for an accelerator to a Global Accelerator-assigned static IP address:
  - You can only change the address back to one of your original Global Accelerator static IP addresses if you make the change *within 10 days* of when you changed it to a BYOIP address. After 10 days, the original static IP address is returned to the Global Accelerator IP address pool and reused. After that, if you update your accelerator to change a BYOIP address to a

Global Accelerator-assigned IP address, you are assigned a new IP address from the Global Accelerator IP address pool.

- You can't change both BYOIP static IP addresses to use Global Accelerator static IP addresses instead. To use two static IP addresses that are assigned by Global Accelerator with an accelerator, create a new accelerator.

- If you have an accelerator that is using two BYOIP addresses, you can change either of them to a different BYOIP address. The same restrictions apply as when you add BYOIP addresses when you create an accelerator, however. For example, if you update an accelerator to use two different BYOIP addresses, the addresses must be from different BYOIP address ranges that you've added to Global Accelerator.
- If you've configured cross-account BYOIP addresses, when you update the static IP addresses for an accelerator, you can use a cross-account address.
- In one specific scenario, when you update a BYOIP address, Global Accelerator might need to change your Amazon static IP address so that it can complete the update successfully. The Amazon static IP address can only be impacted when 1) you update a BYOIP static IPv4 address for your accelerator to use a BYOIP address from another account (that is, a cross-account BYOIP address), and 2) your second static IP address on the accelerator is from the Amazon pool.

  If you didn't want the Amazon static IP address to change, you can revert to the previous Amazon IP address, but only if no more than 10 days has elapsed since you made the update. When you revert the change, the original Amazon IP address is restored for your accelerator. After 10 days has elapsed, however, the Amazon IP address is released back into the available IP addresses pool and can't be restored.

Revert a static IP address change

To revert to the original Amazon IP address for your accelerator, do the following:

- Update the accelerator with the original BYOIP static IP address that you changed to a new address.

When you make this update, Global Accelerator will restore the original Amazon static IP address as well.

Preserve client IP addresses in AWS Global Accelerator

Your options for preserving and accessing the client IP address for AWS Global Accelerator depend on the endpoints that you've set up with your accelerator. When client IP address preservation is enabled, the source IP address of the original client is preserved for packets that arrive at the load balancer.

Endpoints on custom routing accelerators always have the client IP address preserved. There are three types of endpoints for standard accelerators that can preserve the source IP address of the client in incoming packets: Application Load Balancers, Amazon EC2 instances, and Network Load Balancers with security groups. There are requirements and limitations for specific resources that you add as endpoint with client IP address preservation. For more information, see Transition endpoints with client IP address preservation.

Note that Global Accelerator does not support client IP address preservation for the following endpoint types:

- Network Load Balancers without security groups
- Elastic IP addresses

For details about endpoint requirements, see Requirements for resources you add as accelerator endpoints.

Contents

- Guidelines and restrictions for client IP address preservation in Global Accelerator
- Requirements for endpoints with client IP address preservation
- How the client IP address is preserved in AWS Global Accelerator
- Benefits of client IP address preservation
- Best practices for ENIs and security groups with client IP address preservation
- Transition endpoints with client IP address preservation

Guidelines and restrictions for client IP address preservation in Global Accelerator

As you prepare for and use client IP address preservation in AWS Global Accelerator, be aware of the following guidelines and restrictions.

General caveats

When you plan for adding client IP address preservation, be aware of the following:

- Global Accelerator does not support Gateway Load Balancer inspection between Global Accelerator and an endpoint when client IP address preservation is enabled.
- Before you add and begin to route traffic to endpoints that preserve the client IP address, make sure that all your required security configurations, for example, security groups, are updated to include the user client IP address on allow lists.
- You might see client IP addresses in AWS WAF, instead of Global Accelerator IP addresses. Client IP addresses appear in AWS WAF when you configure Global Accelerator for client IP address preservation and you enable AWS WAF to block connections from your Application Load Balancers that don't come from Global Accelerator.
- Client IP address preservation is supported in all AWS Regions where Global Accelerator is supported. For a list of supported Regions, see AWS Region availability for AWS Global Accelerator.

Defaults for client IP address preservation

When you create a new accelerator using the console, client IP address preservation is enabled, by default, for supported endpoints. You can choose to disable the option for some endpoints, depending on the endpoint type:

- When you use an internet-facing Application Load Balancer or a Network Load Balancer with security groups as an endpoint with Global Accelerator, client IP address preservation is enabled by default for new accelerators. You can choose to disable the option when you create the accelerator or by editing the accelerator later.
- When you use an internal Application Load Balancer or an EC2 instance with Global Accelerator, the endpoint always has client IP address preservation enabled.

Be aware of the following:

- Internal Application Load Balancers and EC2 instances always have client IP address preservation enabled. You can't disable the option for these endpoints.

- When you use the AWS console to create a new accelerator, the option for client IP address preservation is enabled by default for Application Load Balancer endpoints. The option is not enabled by default for Network Load Balancer with security groups endpoints. You can update the option for client IP address preservation for these endpoints at any time after you add it.
- When you use the AWS CLI or an API action to create a new accelerator and you don't specify the option for client IP address preservation, the following is the default setting for client IP address preservation:
  - Internet-facing Application Load Balancer endpoints have client IP address preservation enabled by default.
  - Network Load Balancer with security group endpoints do *not* have client IP address preservation enabled by default.

Transitioning endpoints without client IP address preservation

For existing accelerators, you can transition endpoints without client IP address preservation to endpoints that do preserve the client IP address. For example, existing Application Load Balancer endpoints can be transitioned to new Application Load Balancer endpoints. To transition to the new endpoints, we recommend that you move traffic slowly from an existing endpoint to a new endpoint that has client IP address preservation by doing the following:

- For existing Application Load Balancer or Network Load Balancer with security groups endpoints, first add to Global Accelerator a duplicate load balancer endpoint that targets the same backends, and make sure that client IP address preservation is enabled for it. Then adjust the weights on the endpoints to slowly move traffic from the load balancer that does *not* have client IP address preservation enabled to the load balancer *with* client IP address preservation.
- For an existing Elastic IP address endpoint, you can move traffic to an EC2 instance endpoint with client IP address preservation. First add an EC2 instance endpoint to Global Accelerator, and then adjust the weights on the endpoints to slowly move traffic from the Elastic IP address endpoint to the EC2 instance endpoint.

For step-by-step transition guidance, see Transitioning endpoints to use client IP address preservation.

Requirements for endpoints with client IP address preservation

There are specific requirements for endpoint types that you can use with client IP address preservation. >You can use this feature with endpoints that are Application Load Balancers, Network Load Balancers with security groups, and Amazon EC2 instances, subject to the additional requirements described in this section. Endpoints on custom routing accelerators always have the client IP address preserved.

This section provides information that is specific to endpoints that you want to add with client IP address preservation enabled. For information about overall requirements for endpoints, see Requirements for resources you add as accelerator endpoints.

In addition, for more information about best practices with client IP address preservation, see Best practices for ENIs and security groups with client IP address preservation.

If you intend to use the client IP address preservation feature, be aware of the following when you add endpoints to Global Accelerator, in addition to the overall requirements for endpoints in Global Accelerator.

Elastic IP addresses

Client IP address preservation is not supported for Elastic IP address endpoints in Global Accelerator.

Network Load Balancer endpoints

If you want to enable client IP address preservation when you add Network Load Balancer resources as endpoints to Global Accelerator, be aware that client IP address preservation is not supported for the following:

- Network Load Balancers without security groups
- Network Load Balancers with security groups that have TLS listeners attached
- Network Load Balancers with security groups that perform IPv4 to IPv6 NAT translation to their EC2 targets

In addition, for Network Load Balancers, client IP address preservation is supported only when targets are in the same VPC as the Network Load Balancer. Traffic must flow directly from the Network Load Balancer to the target.

These requirements apply only to Network Load Balancer endpoints, not to other load balancing endpoints, such as Application Load Balancers.

Elastic network interfaces

To support client IP address preservation, Global Accelerator creates elastic network interfaces in your AWS account—one for each subnet where an endpoint is present. For more information about how Global Accelerator works with elastic network interfaces, see Best practices for ENIs and security groups with client IP address preservation.

Endpoints in private subnets

You can target an Application Load Balancer, Network Load Balancer, or an EC2 instance in a private subnet using Global Accelerator but you must have an internet gateway attached to the VPC that contains the endpoints. For more information, see Secure VPC connections in AWS Global Accelerator.

As a best practice, use private subnets if you want to ensure that traffic is delivered only by Global Accelerator. Also, make sure that inbound security group rules are configured appropriately to correctly allow or deny traffic for your applications.

Add the client IP address to the allow list

Before you add and begin to route traffic to endpoints that preserve the client IP address, make sure that all your required security configurations, for example, security groups, are updated to include the user client IP address on the allow list. Network access control lists (ACLs) only apply to egress (outbound) traffic. If you need to filter ingress (inbound) traffic, you must use security groups.

Configure network access control lists (ACLs)

Network ACLs associated with your VPC subnets apply to egress (outbound) traffic when client IP address preservation is enabled on your accelerator. However, for traffic to be allowed to exit through Global Accelerator, you must configure the ACL as both an inbound and outbound rule.

For example, to allow TCP and UDP clients using an ephemeral source port to connect to your endpoint through Global Accelerator, associate the subnet of your endpoint with a Network ACL that allows outbound traffic destined to an ephemeral TCP or UDP port (port range 1024-65535, destination 0.0.0.0/0). In addition, create a matching inbound rule (port range 1024-65535, source 0.0.0.0/0).

Be aware of the following for security groups and WAF:

- Security group and AWS WAF rules are an additional set of capabilities that you can apply to protect your resources. For example, the inbound security group rules associated with your Amazon EC2 instances and Application Load Balancers allow you to control the destination

ports that clients can connect to through Global Accelerator, such as port 80 for HTTP or port 443 for HTTPS.

- Amazon EC2 instance security groups apply to any traffic that arrives to your instances, including traffic from Global Accelerator and any public or Elastic IP address that is assigned to your instance.

How the client IP address is preserved in AWS Global Accelerator

AWS Global Accelerator preserves the source IP address of the client differently for Amazon EC2 instances, Network Load Balancers, and Application Load Balancers:

- For an EC2 instance endpoint, the client's IP address is preserved for all traffic.
- For a Network Load Balancer endpoint with client IP address preservation, Global Accelerator works together with the Network Load Balancer to include the IP address of the original client in the IP header of the packet so that your application can access it.
- For an Application Load Balancer endpoint with client IP address preservation, Global Accelerator works together with the Application Load Balancer to provide an `X-Forwarded` header, `X-Forwarded-For`, that includes the IP address of the original client so that your web tier can access it.

HTTP requests and HTTP responses use header fields to send information about the HTTP messages. Header fields are colon-separated name-value pairs that are separated by a carriage return (CR) and a line feed (LF). A standard set of HTTP header fields is defined in RFC 2616, Message Headers. There are also non-standard HTTP headers available that are widely used by the applications. Some of the non-standard HTTP headers have an `X-Forwarded` prefix.

Because an Application Load Balancer terminates incoming TCP connections and creates new connections to your backend targets, it does not preserve client IP addresses all the way to your target code (such as instances, containers, or Lambda code). The source IP address that your targets see in the TCP packet is the IP address of the Application Load Balancer. However, an Application Load Balancer does preserve the original client IP address by removing it from the original packet's reply address and inserting it into an HTTP header before it sends the request to your backend over a new TCP connection.

The `X-Forwarded-For` request header is formatted like this:

```
X-Forwarded-For: client-ip-address
```

The following example shows an `X-Forwarded-For` request header for a client with an IP address of 203.0.113.7.

```
X-Forwarded-For: 203.0.113.7
```

The following example shows an `X-Forwarded-For` request header for a client with an IPv6 address of 2001:DB8::21f:5bff:febf:ce22:8a2e.

```
X-Forwarded-For: 2001:DB8::21f:5bff:febf:ce22:8a2e
```

Benefits of client IP address preservation

You can configure client IP address preservation for specific endpoints in Global Accelerator. For some applications that you configure with AWS Global Accelerator, you might want to access the original client IP address by using endpoints with client IP address preservation.

For example, when you have the client IP address, you can gather statistics based on client IP addresses. You can also use IP-address-based filters, such as security groups on Application Load Balancers, to filter traffic. You can apply logic that is specific to a user's IP address in your applications that run on the web tier servers behind that Application Load Balancer endpoint by using the load balancer's `X-Forwarded-For` header, which contains the original client IP address information. You can also use client IP address preservation in security group rules in the security groups associated with your Application Load Balancer or Network Load Balancer. For more information, see How the client IP address is preserved in AWS Global Accelerator. For EC2 instance endpoints, the original client IP address is preserved.

The client IP address can be used by the following, for example:

- The security group associated with an Application Load Balancer or Network Load Balancer
- Application Load Balancer listener rules
- AWS WAF rules

For endpoints that don’t have client IP address preservation enabled, the IP addresses used by the Global Accelerator service at the edge network replace the requesting user's IP address as the

source address in the arriving packets. The original client's connection information—such as the IP address of the client and the client's port—is not preserved as traffic travels to systems behind an accelerator. This works fine for many applications, especially those that are available to all users such as public websites.

For endpoints that don't have client IP address preservation, you can filter for the source IP address that Global Accelerator uses when it forwards traffic from the edge. You can see information about the source IP addresses (which are also client IP addresses, when client IP address preservation is enabled) of incoming packets by reviewing your Global Accelerator flow logs. For more information, see Location and IP address ranges of Global Accelerator Edge servers and Configuring and using flow logs in AWS Global Accelerator.

Best practices for ENIs and security groups with client IP address preservation

When you use client IP address preservation in AWS Global Accelerator, keep in mind the information and best practices in this section for elastic network interfaces (ENIs) and security groups.

To support client IP address preservation, Global Accelerator creates elastic network interfaces in your AWS account—one for each subnet where an endpoint is present. An elastic network interface is a logical networking component in a VPC that represents a virtual network card. Global Accelerator uses these elastic network interfaces to route traffic to the endpoints configured behind an accelerator. Client IP address preserved traffic still respects your subnet's route table rules and NACLs. The supported endpoints for routing traffic this way are Application Load Balancers (internal and internet-facing), Network Load Balancers with security groups, and Amazon EC2 instances.

> **Note**
>
> When you add an internal Application Load Balancer or an EC2 instance endpoint in Global Accelerator, you enable internet traffic to flow directly to and from the endpoint in Virtual Private Clouds (VPCs) by targeting it in a private subnet. For more information, see Secure VPC connections in AWS Global Accelerator.

How Global Accelerator uses elastic network interfaces

When you have an Application Load Balancer or Network Load Balancer endpoint with client IP address preservation enabled, the number of subnets that the load balancer is in determines the number of elastic network interfaces that Global Accelerator creates in your account. Global Accelerator creates one elastic network interface for each subnet that has at least one elastic network interface of the Application Load Balancer or Network Load Balancer in it that is fronted by an accelerator in your account.

The following examples illustrate how this works:

- **Example 1:** If an Application Load Balancer has elastic network interfaces in subnet A and subnet B, and then you add the load balancer as an accelerator endpoint, Global Accelerator creates two elastic network interfaces, one in each subnet.
- **Example 2:** If you add, for example, an ALB1 that has elastic network interfaces in subnetA and subnetB to Accelerator1, and then add an ALB2 with elastic network interfaces in subnet A and subnet B to Accelerator2, Global Accelerator creates only two elastic network interfaces: one in subnetA and one in subnetB.
- **Example 3:** If you add an ALB1 that has elastic network interfaces in subnetA and subnetB to Accelerator1, and then add an ALB2 with elastic network interfaces in subnetA and subnetC to Accelerator2, Global Accelerator creates three elastic network interfaces: one in subnetA, one in subnetB, and one in subnetC. The elastic network interface in subnetA delivers traffic on for both Accelerator1 and Accelerator2.

As shown in Example 3, elastic network interfaces are reused across accelerators if endpoints in the same subnet are placed behind multiple accelerators.

The logical elastic network interfaces that Global Accelerator creates do not represent a single host, a throughput bottleneck, or a single point of failure. Like other AWS services that appear as a single elastic network interface in an Availability Zone or subnet—services like a network address translation (NAT) gateway or a Network Load Balancer—Global Accelerator is implemented as a horizontally scaled, highly available service.

Evaluate the number of subnets that are used by endpoints in your accelerators to determine the number of elastic network interfaces that Global Accelerator will create. Before you create an accelerator, make sure that you have enough IP address space capacity for the required elastic network interfaces: that is, at least one free IP address per relevant subnet. If you don't have enough free IP address space, you must create or use a subnet that has adequate free IP

address space for your Application Load Balancer or Network Load Balancer and associated Global Accelerator elastic network interfaces.

When Global Accelerator determines that an elastic network interface is not being used by any of the endpoints in accelerators in your account, Global Accelerator deletes the interface.

Security groups created by Global Accelerator

Review the following information and best practices when you work with Global Accelerator and security groups.

- You can use the security groups created by Global Accelerator as a source group in other security groups that you maintain, but Global Accelerator only forwards traffic to the targets that you specify in your VPC.
- If you modify the security group rules created by Global Accelerator, the endpoint might become unhealthy. If that happens, contact AWS Support for assistance.
- Global Accelerator creates a specific security group for each VPC. Elastic network interfaces that are created for the endpoints within a specific VPC all use the same security group, no matter which subnet an elastic network interface is associated with.

> **Important**
>
> Global Accelerator creates security groups that are associated with its elastic network interfaces. Although the system doesn't prevent you from doing so, you shouldn't edit any of the security group settings for these groups.

Transition endpoints with client IP address preservation

If you haven't yet configured client IP address preservation for the endpoints in your accelerator, follow the guidance in this section add and transition one or more endpoints to endpoints that preserve the user's client IP address. You can choose to transition an Application Load Balancer, Network Load Balancer with security groups, or an Elastic IP address endpoint to a corresponding endpoint—a corresponding load balancer endpoint or an EC2 instance endpoint—that has client IP address preservation.

This section explains how to add and transition endpoints by using the AWS Global Accelerator console. If you want to use API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

Transitioning endpoints to use client IP address preservation

We recommend that you transition endpoints to using client IP address preservation slowly.

- **Add the new endpoint:** First, add to Global Accelerator the new load balancer or EC2 instance endpoints that you enable to preserve the client IP address.
- **Slowly increase traffic:** Then, slowly move traffic from existing endpoints to the new endpoints by configuring weights on the endpoints.
- **Test as you go:** After you move a small amount of traffic to the new endpoint with client IP address preservation, test to make sure that your configuration is working as you expect it to. Then gradually increase the proportion of traffic to the new endpoint by adjusting the weights on the corresponding endpoints.

Follow the steps in the following sections to transition your endpoints.

Client IP address preservation is supported in all AWS Regions where Global Accelerator is supported. For a list of supported Regions, see AWS Region availability for AWS Global Accelerator.

> **Important**
>
> Before you begin to route traffic to endpoints that preserve the client IP address, make sure that all the configurations in which you've included Global Accelerator client IP addresses on allow lists are updated to include the user client IP address instead.

To add an endpoint with client IP address preservation

1. Open the Global Accelerator console at https://console.aws.amazon.com/globalaccelerator/home.
2. On the Accelerators page, choose an accelerator.
3. In the **Listeners** section, choose a listener.
4. In the **Endpoint group** section, choose an endpoint group.
5. In the **Endpoints** section, choose **Add endpoint**.
6. On the **Add endpoints** page, in the **Endpoints** drop-down menu, choose an endpoint that supports client IP address preservation.

7. In the **Weight** field, choose a low number compared to the weights that are set for your existing endpoints. For example, if the weight for a corresponding Application Load Balancer is 255, you could enter a weight of 5 for the new Application Load Balancer, to start with. For more information, see How endpoint weights work to manage traffic volume.
8. If needed, under **Preserve client IP address**, select **Preserve address**.
9. Choose **Save changes**.

Next, follow the steps here to edit the corresponding existing endpoints (that you're replacing with the new endpoints with client IP address preservation) to reduce the weights for existing endpoints so that less traffic goes to them.

To reduce traffic for the existing endpoints

1. On the **Endpoint group** page, choose an existing endpoint that doesn't have client IP address preservation.
2. Choose **Edit**.
3. On the **Edit endpoint** page, in the **Weight** field, enter a lower number than the current number. For example, if the weight for an existing endpoint is 255, you could enter a weight of 220 for the new endpoint (with client IP address preservation).
4. Choose **Save changes**.

After you've tested with a small portion of the original traffic by setting the weight for the new endpoint to a low number, you can slowly transition all the traffic by continuing to adjust the weights for the original and new endpoints.

For example, say you start with an existing Application Load Balancer with a weight set to 200, and you add a new Application Load Balancer endpoint with client IP address preservation enabled with a weight set to 5. Gradually shift traffic from the original Application Load Balancer to the new Application Load Balancer by increasing the weight for the new Application Load Balancer and decreasing the weight for the original Application Load Balancer. For example:

- Original weight → 190 new weight 10
- Original weight 180 → new weight 20
- Original weight 170 → new weight 30, and so on.

When you have decreased the weight to 0 for the original endpoint, all traffic (in this example scenario) goes to the new Application Load Balancer endpoint, which includes client IP address preservation.

If you have additional endpoints—load balancers or EC2 instances—that you want to transition to use client IP address preservation, repeat the steps in this section to transition them.

If you need to revert your configuration for an endpoint so that traffic to the endpoint doesn't preserve the client IP address, you can do that at any time: increase the weight for the endpoint that does *not* have client IP address preservation to the original value, and decrease the weight for the endpoint *with* client IP address preservation to 0.

Logging and monitoring in AWS Global Accelerator

You can use Amazon CloudWatch, flow logs, and AWS CloudTrail to monitor your accelerator in AWS Global Accelerator. For example, you can troubleshoot issues with your listeners and endpoints, analyze traffic patterns, and get information that's required for audits.

These logging and monitoring methods can have some overlap. The following are typical uses for each method:

- **CloudWatch metrics** provide real time information, without additional setup, that can help you troubleshoot setup. You can also create alarms to alert you, for example, when there are production issues.
- **Flow logs** provide detailed information about traffic coming into an accelerator and going back to clients. Flow logs are useful for troubleshooting reachability issues and for providing information for comprehensive audits. (Note that flow logs require setup and use Amazon S3 storage.)
- **CloudTrail** automatically tracks actions that you take that call Global Accelerator APIs, which can be useful for audits, for example.

> **Note**
>
> You must view CloudWatch metrics and logs for Global Accelerator in the US West (Oregon) Region, both in the console or when using the AWS CLI. When you use the AWS CLI, specify the US West (Oregon) Region for your command by including the following parameter: `--region us-west-2`.

Topics

- Using Amazon CloudWatch with AWS Global Accelerator
- Configuring and using flow logs in AWS Global Accelerator
- Using AWS CloudTrail to log AWS Global Accelerator API calls

Using Amazon CloudWatch with AWS Global Accelerator

AWS Global Accelerator publishes data points to Amazon CloudWatch for your accelerators. 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 traffic through an accelerator over a specified time period. Each data point has an associated time stamp and an optional unit of measurement.

> **Note**
>
> You must view CloudWatch metrics and logs for Global Accelerator in the US West (Oregon) Region, both in the console or when using the AWS CLI. When you use the AWS CLI, specify the US West (Oregon) Region for your command by including the following parameter: `--region us-west-2`.

You can use metrics to troubleshoot an initial Global Accelerator setup, to help determine whether traffic is arriving at an endpoint, and then responses are returning. View the CloudWatch metrics, which are logged automatically, to see if traffic is making it to your endpoints, such as a Network Load Balancer. There should be metrics for outbound from Global Accelerator towards the endpoints, and then from Global Accelerator back to the client, and the same for an endpoint, such as a load balancer. Traffic flowing in from Global Accelerator but not back out, or not reaching the load balancer, can indicate that you need to verify that your configuration allows traffic to flow through the expected ports and that your security group settings allow access.

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

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

For more information, see the Amazon CloudWatch User Guide.

Contents

- Global Accelerator metrics
- Metric dimensions for accelerators
- Troubleshooting Global Accelerator TCP reset issues
- Statistics for Global Accelerator metrics
- View CloudWatch metrics for your accelerators

Global Accelerator metrics

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

Metric	Description
`ActiveFlowCount`	The total number of concurrent TCP and UDP connections from clients to endpoints for an accelerator in Global Accelerator. For TCP connections, which are terminated at the accelerator, a client opening a TCP connection to an endpoint counts as a single flow. You can use this metric to better understand how many active users (connection count) are accessing an endpoint, or to determine if your resources need to be scaled to handle traffic. **Reporting criteria**: Reported for accelerators that are configured and enabled. **Statistics**: The only useful statistic is `Sum`. **Dimensions** • `Accelerator` • `Accelerator, Listener` • `Accelerator, Listener, EndpointGroup` • `Accelerator, SourceRegion` • `Accelerator, DestinationEdge` • `Accelerator, TransportProtocol` • `Accelerator, AcceleratorIPAddress`

Metric	Description
`Flows_Dropped_No_Endpoint_Found`	The total number of TCP IPv6 packet flows that were dropped because no IPv6 endpoints were available. This could happen, for example, if you had an accelerator with a dual-stack IP address type and you changed the IP address type to IPv4 for an endpoint for the accelerator. **Reporting criteria**: Reported for accelerators with dual-stack IP address types that are receiving IPv6 traffic when one of the following occurs: • An accelerator with IPv6 endpoints serving traffic reports a 0 metric • An accelerator with misconfigured endpoints reports the total number of flows dropped **Statistics**: The only useful statistic is `Sum`. **Dimensions** • `Accelerator` • `Accelerator, Listener` • `Accelerator, AcceleratorIPAddress`

Metric	Description
HealthyEndpointCount	The total number of endpoints that are considered healthy. Global Accelerator regularly checks the status of endpoints on standard accelerators. These health checks run automatically. How and when these health checks run depends on the type of endpoint and the health check options for the endpoint. To learn more, see Ensure health check access for your accelerator. **Reporting criteria**: Reported for accelerators that are configured and enabled. **Statistics**: The most useful statistics are `Minimum` and `Maximum`. **Dimensions** • `Accelerator` • `Accelerator, Listener` • `Accelerator, Listener, EndpointGroup`
NewFlowCount	The total number of new TCP and UDP flows (or connections) established from clients to endpoints in the time period. **Reporting criteria**: There is a nonzero value. **Statistics**: The only useful statistic is `Sum`. **Dimensions** • `Accelerator` • `Accelerator, Listener` • `Accelerator, Listener, EndpointGroup` • `Accelerator, SourceRegion` • `Accelerator, DestinationEdge` • `Accelerator, TransportProtocol` • `Accelerator, AcceleratorIPAddress` • `Accelerator, NetworkProtocol`

Metric	Description
ProcessedBytesIn	The total number of incoming bytes processed by the accelerator, including TCP/IP headers. This count includes all traffic to endpoints . **Reporting criteria**: There is a nonzero value. **Statistics**: The only useful statistic is Sum. **Dimensions** • Accelerator • Accelerator, Listener • Accelerator, Listener, EndpointGroup • Accelerator, SourceRegion • Accelerator, DestinationEdge • Accelerator, TransportProtocol • Accelerator, AcceleratorIPAddress • Accelerator, NetworkProtocol

Metric	Description
ProcessedBytesOut	The total number of outgoing bytes processed by the accelerator, including TCP/IP headers. This count includes traffic from endpoints , minus health check traffic. **Reporting criteria**: There is a nonzero value. **Statistics**: The only useful statistic is Sum. **Dimensions** • Accelerator • Accelerator, Listener • Accelerator, Listener, EndpointGroup • Accelerator, SourceRegion • Accelerator, DestinationEdge • Accelerator, TransportProtocol • Accelerator, AcceleratorIPAddress • Accelerator, NetworkProtocol

Metric	Description
PacketsProcessed	The total number of packets processed by Global Accelerator for an accelerator, including traffic to and from endpoints, including health check traffic. This metric can help you to benchmark traffic volumes within a specific time period. **Reporting criteria**: Reported for accelerators that are configured and enabled. **Statistics**: The only useful statistic is Sum. **Dimensions** • Accelerator • Accelerator, Listener • Accelerator, Listener, EndpointGroup • Accelerator, SourceRegion • Accelerator, DestinationEdge • Accelerator, TransportProtocol • Accelerator, AcceleratorIPAddress

Metric	Description
UnhealthyEndpointCount	The total number of endpoints that are considered unhealthy. Global Accelerator regularly checks the status of endpoints on standard accelerators. These health checks run automatically. How and when these health checks run depend on the type of endpoint and the health check options for the endpoint. To learn more, see Ensure health check access for your accelerator. **Reporting criteria**: Reported for accelerators that are configured and enabled. **Statistics**: The most useful statistics are `Minimum` and `Maximum`. **Dimensions** • `Accelerator` • `Accelerator, Listener` • `Accelerator, Listener, EndpointGroup`

Metric	Description
TCP_AGA_Reset_Count	The total number of reset (RST) packets generated by AWS Global Accelerator ("AGA"). By using this metric, you can determine whether Global Accelerator is terminating client connections and sending resets back to the client endpoint. For more information about evaluating and troubleshooting TCP RST generated by Global Accelerator, see Troubleshooting Global Accelerator TCP reset issues. **Reporting criteria**: Reported when there is traffic and there is a nonzero value. **Statistics**: The only useful statistic is Sum. **Dimensions** • Accelerator • Accelerator, Listener • Accelerator, Listener, EndpointGroup • Accelerator, SourceRegion • Accelerator, DestinationEdge • Accelerator, AcceleratorIPAddress

Metric	Description
`TCP_Client_Reset_Count`	The total number of reset (RST) packets sent from a client to an endpoint. By using this metric, you can determine whether a client can keep a connection open with Global Accelerator or if the connection is reset unexpectedly early. This is useful, for example, when you configure Global Accelerator initially, and for visibility when you make a change to clients that create connection resets. For more information about evaluating and troubleshooting TCP RST generated by Global Accelerator, see Troubleshooting Global Accelerator TCP reset issues. **Reporting criteria**: Reported when there is traffic and there is a nonzero value. **Statistics**: The only useful statistic is `Sum`. **Dimensions** • `Accelerator` • `Accelerator, Listener` • `Accelerator, Listener, EndpointGroup` • `Accelerator, SourceRegion` • `Accelerator, DestinationEdge` • `Accelerator, AcceleratorIPAddress`

Metric	Description
TCP_Endpoint_Reset_Count	The total number of reset (RST) packets sent from an endpoint to a client. Using this metric, can help you determine when your client endpoints are overloaded. For more information about evaluating and troubleshooting TCP RST generated by Global Accelerator, see Troubleshooting Global Accelerator TCP reset issues. **Reporting criteria**: Reported when there is traffic and there is a nonzero value. **Statistics**: The only useful statistic is Sum. **Dimensions** • Accelerator • Accelerator, Listener • Accelerator, Listener, EndpointGroup • Accelerator, SourceRegion • Accelerator, DestinationEdge • Accelerator, AcceleratorIPAddress

Metric dimensions for accelerators

To filter the metrics for your accelerator, use the following dimensions.

Dimension	Description
Accelerator	Filters the metric data by accelerator. Specify the accelerator by the accelerator id (the final portion of the accelerator ARN). For example, if the ARN is arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh , you specify the following: **1234abcd-abcd-1234-abcd-1234abcdefgh** .

Dimension	Description
`Listener`	Filters the metric data by listener. Specify the listener by the listener id (the final portion of the listener ARN). For example, if the ARN is `arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123wxyz`, you specify the following: **`0123wxyz`**.
`EndpointGroup`	Filters the metric data by endpoint group. Specify the endpoint group by the AWS Region, for example, **`us-east-1`** (all lowercase).
`SourceRegion`	Filters the metric data by source region, which is the geographic area of the AWS Regions where your application endpoints are running. Source region is one of the following: • NA – United States and Canada • EU – Europe • AP – Asia Pacific* • KR – South Korea • IN – India • AU – Australia • ME – Middle East • SA – South America • ZA – South Africa *Excluding South Korea and India

Dimension	Description
DestinationEdge	Filters the metric data by destination edge, which is the geographic area of the AWS edge locations that serve your client traffic. Destination edge is one of the following: • NA – United States and Canada • EU – Europe • AP – Asia Pacific* • KR – South Korea • IN – India • AU – Australia • ME – Middle East • SA – South America • ZA – South Africa *Excluding South Korea and India
TransportProtocol	Filters the metric data by transport protocol: UDP or TCP.
AcceleratorIPAddress	Filters the metric data by the IP address of the accelerator: that is, one of the static IP addresses assigned to an accelerator.

Troubleshooting Global Accelerator TCP reset issues

Each accelerator reports the number of TCP Resets (TCP RSTs) that were generated and sent from Global Accelerator. The following are common reasons that Global Accelerator sends a TCP Reset:

- Global Accelerator marks a TCP connection as closed when either the client or the endpoint closes the connection, using FIN handshake or Reset. If the client or endpoint sends data packets on a closed TCP connection, then Global Accelerator generates a TCP Reset to indicate that the connection is closed and cannot accept traffic.
- If a client or endpoint sends data after the idle timeout period elapses, it receives a TCP Reset packet from Global Accelerator to indicate that the connection is no longer valid.

- If Global Accelerator receives an unexpected packet while building the connection with either the client or the endpoint during the TCP handshake, Global Accelerator generates a TCP Reset.

If you see a stable number of `AGA_Reset_Count` metrics for an accelerator, this is because the client or the endpoint sent data towards Global Accelerator to a closed or expired connection.

If you notice a sharp increase in `AGA_Reset_Count` metrics and the increase aligns with related metrics changes on the endpoint side, such as a scale up, scale down, or an unhealthy endpoint, the endpoint might have become unreachable and triggered the Global Accelerator TCP Reset. For help investigating this issue, contact AWS support.

Statistics for Global Accelerator metrics

CloudWatch provides statistics based on the metric data points published by Global Accelerator. Statistics are aggregations of metric data over a specified period of time. When you request statistics, the returned data stream is identified by the metric name and dimension. A dimension is a name/value pair that uniquely identifies a metric. For example, you can request the processed bytes out for an accelerator where the bytes are served from AWS edge locations in Europe (destination edge is "EU").

The following are examples of metric/dimension combinations that you might find useful:

- View the amount of traffic served (such as ProcessedBytesOut) by each of your two accelerator IP addresses to validate that your DNS configuration is correct.
- View the geographical distribution of your user traffic and monitor how much of it is local (for example, North America to North America) or global (for example, Australia or India to North America). To determine this, view the metrics ProcessedBytesIn or ProcessedBytesOut with the dimensions DestinationEdge and SourceRegion set to specific values.
- View the number of unhealthy endpoints across your accelerator, and determine which endpoint groups they belong to. If you have a large number of endpoint groups, this is especially useful to help you quickly find endpoint groups with endpoints that are experiencing issues. To determine this, view the metric UnhealthyEndpointCount with the dimensions Accelerator, Listener, and EndpointGroup.

View CloudWatch metrics for your accelerators

You can view the CloudWatch metrics for your accelerators using the CloudWatch console or the AWS CLI. In the console, metrics are displayed as monitoring graphs. The monitoring graphs show data points only if the accelerator is active and receiving requests.

You must view CloudWatch metrics for Global Accelerator in the US West (Oregon) Region, both in the console or when using the AWS CLI. When you use the AWS CLI, specify the US West (Oregon) Region for your command by including the following parameter: `--region us-west-2`.

To view metrics using the CloudWatch console, follow the steps in the Amazon CloudWatch User Guide and select the **GlobalAccelerator** namespace. To learn more, see View available metrics.

To get the statistics for a metric using the AWS CLI

Use the following get-metric-statistics command to get statistics for a 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 specifically published. You must specify the same dimensions that were used when the metrics were created.

The following example lists the total processed bytes in, per minute, for your accelerator serving from the North America (NA) destination edge.

```
aws cloudwatch get-metric-statistics --namespace AWS/GlobalAccelerator \
--metric-name ProcessedBytesIn \
--region us-west-2 \
--statistics Sum --period 60 \
--dimensions Name=Accelerator,Value=1234abcd-abcd-1234-abcd-1234abcdefgh
 Name=DestinationEdge,Value=NA \
--start-time 2019-12-18T20:00:00Z --end-time 2019-12-18T21:00:00Z
```

The following is example output from the command:

```
{
    "Label": "ProcessedBytesIn",
    "Datapoints": [
        {
            "Timestamp": "2019-12-18T20:45:00Z",
            "Sum": 2410870.0,
            "Unit": "Bytes"
        },
        {
```

```
            "Timestamp": "2019-12-18T20:47:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:46:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:42:00Z",
            "Sum": 1560.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:48:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:43:00Z",
            "Sum": 1343.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:49:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:44:00Z",
            "Sum": 35791560.0,
            "Unit": "Bytes"
        }
    ]
}
```


Configuring and using flow logs in AWS Global Accelerator

Flow logs enable you to capture information about the IP address traffic going to and from network interfaces in your accelerator in AWS Global Accelerator. Flow log data is published to Amazon S3, where you can retrieve and view your data after you've created a flow log.

> **Note**
>
> You must view CloudWatch metrics and logs for Global Accelerator in the US West (Oregon) Region, both in the console or when using the AWS CLI. When you use the AWS CLI, specify the US West (Oregon) Region for your command by including the following parameter: `--region us-west-2`.

Flow logs can help you with a number of tasks. For example, you can troubleshoot why specific traffic is not reaching an endpoint, which in turn helps you diagnose overly restrictive security group rules. You can also use flow logs as a security tool to monitor the traffic that is reaching your endpoints.

A flow log record represents a network flow in your flow log. Each record captures the network flow for a specific 5-tuple, for a specific capture window. A 5-tuple is a set of five different values that specify the source, destination, and protocol for an IP flow. The capture window is a duration of time during which the flow logs service aggregates data before publishing flow log records. The capture window is up to 1 minute. That is, logs might be published more frequently than every minute, but will be published at least every minute.

CloudWatch Logs charges apply when using flow logs, even when logs are published directly to Amazon S3. For more information, see *Vended Logs* under the *Logs* tab at Amazon CloudWatch Pricing.

> **Tip**
>
> Using Amazon Athena and Amazon QuickSight with your Global Accelerator flow log data can help you troubleshoot reachability issues for your application, identify security vulnerabilities, and get an overview of how users access your application. To learn more, see the following AWS blog post: Analyzing and visualizing AWS Global Accelerator flow logs using Amazon Athena and QuickSight.

Contents

- Enable publishing flow logs to Amazon S3
- Processing flow log records in Amazon S3
- Publishing flow logs to Amazon S3

- Timing of log file delivery
- Flow log record syntax

Enable publishing flow logs to Amazon S3

To enable flow logs in AWS Global Accelerator, follow the steps in this procedure. Additional sections in this chapter provide the steps to configure your Amazon S3 bucket and set up permissions, so that flow logs can be published and accessed.

To enable flow logs in AWS Global Accelerator

1. Create an Amazon S3 bucket for your flow logs in your AWS account.
2. Add the required IAM policy for the AWS user who is enabling the flow logs. For more information, see IAM roles for publishing flow logs to Amazon S3.
3. Run the following AWS CLI command, with the Amazon S3 bucket name and prefix that you want to use for your log files:

```
aws globalaccelerator update-accelerator-attributes
       --accelerator-arn
 arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh
       --region us-west-2
       --flow-logs-enabled
       --flow-logs-s3-bucket s3-bucket-name
       --flow-logs-s3-prefix s3-bucket-prefix
```

Processing flow log records in Amazon S3

The log files are compressed. If you open the log files using the Amazon S3 console, they are decompressed and the flow log records are displayed. If you download the files, you must decompress them to view the flow log records.

Publishing flow logs to Amazon S3

Flow logs for AWS Global Accelerator are published to Amazon S3 to an existing S3 bucket that you specify. Flow log records are published to a series of log file objects that are stored in the bucket.

To create an Amazon S3 bucket for use with flow logs, see Create your first S3 bucket in the *Amazon Simple Storage Service User Guide*.

Flow logs files

Flow logs collect flow log records, consolidate them into log files, and then publish the log files to the Amazon S3 bucket at 5-minute intervals. That is, log files are written every five minutes, and each log file contains flow log records for the IP address traffic recorded in the previous five minutes.

The maximum file size for a log file is 75 MB. If the log file reaches the file size limit within the 5-minute period, the flow log stops adding flow log records to it, publishes it to the Amazon S3 bucket, and then creates a new log file.

Log files are saved to the specified Amazon S3 bucket using a folder structure that is determined by the flow log's ID, Region, and the date on which they are created. The bucket folder structure uses the following format:

```
s3-bucket_name/s3-bucket-prefix/AWSLogs/aws_account_id/globalaccelerator/region/yyyy/mm/dd/
```

Similarly, the log file name is determined by the flow log's ID, Region, and the date and time it was created. File names use the following format:

```
aws_account_id_globalaccelerator_accelerator_id_flow_log_id_timestamp_hash.log.gz
```

Note the following about the folder and file name structure for log files:

- The timestamp uses the YYYYMMDDTHHmmZ format.
- If you specify slash (/) for the S3 bucket prefix, the log file bucket folder structure will include a double slash (//), like the following:

  ```
  s3-bucket_name//AWSLogs/aws_account_id
  ```

The following example shows the folder structure and file name of a log file for a flow log created by AWS account 123456789012 for an accelerator with an ID of 1234abcd-abcd-1234-abcd-1234abcdefgh, on November 23, 2018 at 00:05 UTC:

```
amzn-s3-demo-bucket/prefix1/AWSLogs/123456789012/globalaccelerator/us-
west-2/2018/11/23/123456789012_globalaccelerator_1234abcd-abcd-1234-
abcd-1234abcdefgh_20181123T0005Z_1fb1234.log.gz
```

A single flow log file contains interleaved entries with multiple 5-tuple records; that is, `client_ip`, `client_port`, `accelerator_ip`, `accelerator_port`, `protocol`. To see all the flow log files for your accelerator, look for entries aggregated by the `accelerator_id` and your `account_id`.

IAM roles for publishing flow logs to Amazon S3

An IAM principal, such as an IAM role or user, must have sufficient permissions to publish flow logs to the Amazon S3 bucket. The IAM policy must include the following permissions. To learn more about how to create policies, including where and how to add JSON similar to the following to an IAM policy, see Identity-based policy examples for AWS Global Accelerator

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DeliverLogs",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowGlobalAcceleratorService",
            "Effect": "Allow",
            "Action": [
                "globalaccelerator:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "s3Perms",
            "Effect": "Allow",
            "Action": [
```

```
                "s3:GetBucketPolicy",
                "s3:PutBucketPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon S3 bucket permissions for flow logs

By default, Amazon S3 buckets and the objects that they contain are private. Only the bucket owner can access the bucket and the objects stored in it. The bucket owner, however, can grant access to other resources and users by writing an access policy.

If the user creating the flow log owns the bucket, the service automatically attaches the following policy to the bucket to give the flow log permission to publish logs to it:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::bucket_name/optional_folder/
AWSLogs/account_id/*",
            "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-
control"}}
        },
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::bucket_name"
        }
    ]
}
```

If the user creating the flow log does not own the bucket, or does not have the `GetBucketPolicy` and `PutBucketPolicy` permissions for the bucket, the flow log creation fails. In this case, the bucket owner must manually add the preceding policy to the bucket and specify the flow log creator's AWS account ID. For more information, see Adding a bucket policy by using the Amazon S3 console in the *Amazon Simple Storage Service User Guide*. If the bucket receives flow logs from multiple accounts, add a `Resource` element entry to the `AWSLogDeliveryWrite` policy statement for each account.

For example, the following bucket policy allows AWS accounts 123123123123 and 456456456456 to publish flow logs to a folder named `flow-logs` in a bucket named `log-bucket`:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": [
             "arn:aws:s3:::log-bucket/flow-logs/AWSLogs/123123123123/*",
             "arn:aws:s3:::log-bucket/flow-logs/AWSLogs/456456456456/*"
             ],
            "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-
control"}}
        },
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::log-bucket"
        }
    ]
}
```

> **Note**
>
> We recommend that you grant the `AWSLogDeliveryAclCheck` and `AWSLogDeliveryWrite` permissions to the log delivery service principal instead of individual AWS account ARNs.

Required CMK key policy for use with SSE-KMS buckets

If you enabled server-side encryption for your Amazon S3 bucket using AWS KMS-managed keys (SSE-KMS) with a customer-managed CMK, you must add the following to the key policy for your CMK so that flow logs can write log files to the bucket:

```
{
    "Sid": "Allow AWS Global Accelerator Flow Logs to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "delivery.logs.amazonaws.com"
        ]
    },
    "Action": "kms:GenerateDataKey*",
    "Resource": "*"
}
```

Amazon S3 log file permissions

In addition to the required bucket policies, Amazon S3 uses access control lists (ACLs) to manage access to the log files created by a flow log. By default, the bucket owner has `FULL_CONTROL` permissions on each log file. The log delivery owner, if different from the bucket owner, has no permissions. The log delivery account has `READ` and `WRITE` permissions. For more information, see Access control list (ACL) overview in the *Amazon Simple Storage Service User Guide*.

Timing of log file delivery

AWS Global Accelerator delivers log files for your configured accelerator up to several times an hour. In general, a log file contains information about the requests that your accelerator received during a given time period. Global Accelerator usually delivers the log file for that time period to your Amazon S3 bucket within an hour of the events that appear in the log. Some or all log file entries for a time period can sometimes be delayed by up to 24 hours. When log entries are

delayed, Global Accelerator saves them in a log file for which the file name includes the date and time of the period in which the requests occurred, not the date and time when the file was delivered.

When creating a log file, Global Accelerator consolidates information for your accelerator from all the edge locations that received requests during the time period that the log file covers.

Global Accelerator begins to reliably deliver log files about four hours after you enable logging. You might get a few log files before that time.

> **Note**
>
> If no users connect to your accelerator during the time period, you don't receive any log files for that period.

Flow log record syntax

A flow log record is a space-separated string that has the following format:

```
<version> <aws_account_id> <accelerator_id> <client_ip>
<client_port> <accelerator_ip> <accelerator_port> <endpoint_ip>
<endpoint_port> <protocol> <ip_address_type> <packets>
<bytes> <start_time> <end_time> <action> <log-status>
<globalaccelerator_source_ip> <globalaccelerator_source_port>
<endpoint_region> <globalaccelerator_region> <direction> <vpc_id>
<reject_reason>
```

The Version 1.0 format does not include the VPC identifier, `vpc_id`. The Version 2.0 format, which includes `vpc_id`, is generated when Global Accelerator sends traffic to an endpoint with client IP address preservation.

The following table describes the fields of a flow log record.

Field	Description
`version`	The flow logs version.
`aws_account_id`	The AWS account ID for the flow log.

Field	Description
accelerator_id	The ID of the accelerator for which the traffic is recorded.
client_ip	The source IPv4 or IPv6 address.
client_port	The source port.
accelerator_ip	The accelerator's IP address.
accelerator_port	The accelerator's port.
endpoint_ip	The destination IP address of the traffic.
endpoint_port	The destination port of the traffic.
protocol	The IANA protocol number of the traffic. For more information, see Assigned Internet Protocol Numbers.
ip_address_type	IPv4 or IPv6.
packets	The number of packets transferred during the capture window. When the number of packets is 0 (zero), the flow is alive but no packets were seen in that direction during a capture window.
bytes	The number of bytes transferred during the capture window.
start_time	The time, in Unix seconds, of the start of the capture window.
end_time	The time, in Unix seconds, of the end of the capture window.
action	The action associated with the traffic: • ACCEPT: The recorded traffic was permitted by the security groups or network ACLs. The value is currently always ACCEPT.

Field	Description
log-status	The logging status of the flow log: • OK: Data is logging normally to the chosen destinations. • SKIPDATA: Some flow log records were skipped during the capture window. This can be because of an internal capacity constraint, or an internal error.
globalaccelerator_source_ip	The IP address used by the Global Accelerator network interface. If client IP address preservation is enabled, this value is set to - (hyphen). For more information, see Preserve client IP addresses in AWS Global Accelerator.
globalaccelerator_source_port	The port used by the Global Accelerator network interface. If client IP address preservation is enabled, this value is set to 0 (zero). For more information, see Preserve client IP addresses in AWS Global Accelerator.
endpoint_region	The AWS Region where the endpoint is located.
globalaccelerator_region	The edge location (point of presence) that served the request. Each edge location has a three-letter code and an arbitrarily assigned number, for example, DFW3. The three-letter code typically corresponds with the International Air Transport Association airport code for an airport near the edge location. (These abbreviations might change in the future.)
direction	The direction of the traffic. Denotes traffic coming into the Global Accelerator network (INGRESS) or returning to the client (EGRESS).
vpc_id	The VPC identifier. Included with Version 2.0 flow logs when Global Accelerator sends traffic to an endpoint with client IP address preservation.

Field	Description
`reject_reason`	The reason traffic was not served. The value is BPA if traffic was not served due to VPC Block Public Access feature settings. Otherwise, the field is empty.

If a field does not apply for a specific record, the record displays a '-' symbol for that entry.

Using AWS CloudTrail to log AWS Global Accelerator API calls

AWS Global Accelerator is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Global Accelerator. CloudTrail captures all API calls for Global Accelerator as events, including calls from the Global Accelerator console and from code calls to the Global Accelerator API. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Global Accelerator. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**.

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

Global Accelerator information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Global Accelerator, 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 Global Accelerator, 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 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 topics:

- 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 Global Accelerator actions are logged by CloudTrail and are documented in the AWS Global Accelerator API Reference. For example, calls to the `CreateAccelerator`, `ListAccelerators` and `UpdateAccelerator` operations 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.

Viewing Global Accelerator events in event history

CloudTrail lets you view recent events in **Event history**. To view events for Global Accelerator API requests, you must choose **US West (Oregon)** in the Region selector at the top of the console. For more information, see Viewing events with CloudTrail event history in the *AWS CloudTrail User Guide*.

Understanding Global Accelerator log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. Each JSON-formatted CloudTrail log file contains one or more log entries. A log entry represents a single request from any source and includes information about the requested action, including any parameters, the date and time of the action, 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 following example shows a CloudTrail log entry that includes the following Global Accelerator actions:

- Listing the accelerators for an account: `eventName` is `ListAccelerators`.
- Creating a listener: `eventName` is `CreateListener`.
- Updating a listener: `eventName` is `UpdateListener`.

- Describing a listener: `eventName` is `DescribeListener`.
- Listing the listeners for an account: `eventName` is `ListListeners`.
- Deleting a listener: `eventName` is `DeleteListener`.


```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:03:14Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "ListAccelerators",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": null,
      "responseElements": null,
      "requestID": "083cae81-28ab-4a66-862f-096e1example",
      "eventID": "fe8b1c13-8757-4c73-b842-fe2a3example",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
```

```
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:04:49Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "CreateListener",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": {
        "acceleratorArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-
a114-5d7fexample",
        "portRanges": [
          {
            "fromPort": 80,
            "toPort": 80
          }
        ],
        "protocol": "TCP"
      },
      "responseElements": {
        "listener": {
          "listenerArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-
a114-5d7fexample/listener/abcde1234",
```

```
        "portRanges": [
          {
            "fromPort": 80,
            "toPort": 80
          }
        ],
        "protocol": "TCP",
        "clientAffinity": "NONE"
      }
    },
    "requestID": "6090509a-5a97-4be6-8e6a-7d73example",
    "eventID": "9cab44ef-0777-41e6-838f-f249example",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  },
  {
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "A1B2C3D4E5F6G7EXAMPLE",
      "arn": "arn:aws:iam::111122223333:user/smithj",
      "accountId": "111122223333",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "sessionContext": {
        "attributes": {
          "mfaAuthenticated": "false",
          "creationDate": "2018-11-17T21:02:36Z"
        },
        "sessionIssuer": {
          "type": "Role",
          "principalId": "A1B2C3D4E5F6G7EXAMPLE",
          "arn": "arn:aws:iam::111122223333:user/smithj",
          "accountId": "111122223333",
          "userName": "smithj"
        }
      }
    },
    "eventTime": "2018-11-17T21:03:52Z",
    "eventSource": "globalaccelerator.amazonaws.com",
    "eventName": "CreateAccelerator",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.50",
    "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
    "requestParameters": {
```

```
        "name": "cloudTrailTest"
      },
      "responseElements": {
        "accelerator": {
          "acceleratorArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-
a114-5d7fexample",
          "name": "cloudTrailTest",
          "ipAddressType": "IPV4",
          "enabled": true,
          "ipSets": [
            {
              "ipAddressFamily": "IPv4",
              "ipAddresses": [
                "192.0.2.213",
                "192.0.2.200"
              ]
            }
          ],
          "status": "IN_PROGRESS",
          "createdTime": "Nov 17, 2018 9:03:52 PM",
          "lastModifiedTime": "Nov 17, 2018 9:03:52 PM"
        }
      },
      "requestID": "d2d7f300-2f0b-4bda-aa2d-e67d6e4example",
      "eventID": "11f9a762-8c00-4fcc-80f9-848a29example",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
```

```
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:05:27Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "UpdateListener",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": {
        "listenerArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-
a114-5d7fexample/listener/abcde1234",
        "portRanges": [
          {
            "fromPort": 80,
            "toPort": 80
          },
          {
            "fromPort": 81,
            "toPort": 81
          }
        ]
      },
      "responseElements": {
        "listener": {
          "listenerArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-
a114-5d7fexample/listener/abcde1234",
          "portRanges": [
            {
              "fromPort": 80,
              "toPort": 80
            },
            {
              "fromPort": 81,
              "toPort": 81
            }
          ],
          "protocol": "TCP",
```

```
          "clientAffinity": "NONE"
        }
      },
      "requestID": "008ef93c-b3a3-44b4-afb3-768example",
      "eventID": "85958f0d-63ff-4a2c-99e3-6ffbexample",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:06:05Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "DescribeListener",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": {
        "listenerArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-
a114-5d7fexample/listener/abcde1234"
      },
      "responseElements": null,
      "requestID": "9980e368-82fa-40da-95a3-4b0example",
      "eventID": "885a02e9-2a60-4626-b1ba-57285example",
```

```
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:05:47Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "ListListeners",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": {
        "acceleratorArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-
a114-5d7fexample"
      },
      "responseElements": null,
      "requestID": "08e4b0f7-689b-4c84-af2d-47619example",
      "eventID": "f4fb8e41-ed21-404d-af9d-037c4example",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.05",
```

```
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:06:24Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "DeleteListener",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": {
        "listenerArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-
a114-5d7fexample/listener/abcde1234"
      },
      "responseElements": null,
      "requestID": "04d37bf9-3e50-41d9-9932-6112example",
      "eventID": "afedb874-2e21-4ada-b1b0-2ddb2example",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    }
  ]
}
```

Security in AWS Global Accelerator

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to AWS Global Accelerator, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Global Accelerator. The following topics show you how to configure Global Accelerator to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Global Accelerator resources.

Contents

- Identity and Access Management for AWS Global Accelerator
- Secure VPC connections in AWS Global Accelerator
- Logging and monitoring in AWS Global Accelerator
- Compliance validation for AWS Global Accelerator
- Resilience in AWS Global Accelerator
- Infrastructure security in AWS Global Accelerator

Identity and Access Management for AWS Global Accelerator

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Global Accelerator resources. IAM is an AWS service that you can use with no additional charge.

Contents

- Audience
- Authenticating with identities
- Managing access using policies
- How AWS Global Accelerator works with IAM
- Identity-based policy examples for AWS Global Accelerator
- Service-linked role for AWS Global Accelerator
- AWS managed policies for AWS Global Accelerator
- Using tag-based policies with AWS Global Accelerator
- Troubleshooting AWS Global Accelerator identity and access

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Global Accelerator.

Service user – If you use the Global Accelerator service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Global Accelerator features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Global Accelerator, see Troubleshooting AWS Global Accelerator identity and access.

Service administrator – If you're in charge of Global Accelerator resources at your company, you probably have full access to Global Accelerator. It's your job to determine which Global Accelerator features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Global Accelerator, see How AWS Global Accelerator works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Global Accelerator. To view example Global Accelerator identity-based policies that you can use in IAM, see Identity-based policy examples for AWS Global Accelerator.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and AWS Multi-factor authentication in IAM in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your

root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. To temporarily assume an IAM role in the AWS Management Console, you can switch from a user to an IAM role (console). You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Methods to assume a role in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Create a role for a third-party identity provider (federation) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

- **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.
- **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Use an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – RCPs are JSON policies that you can use to set the maximum available permissions for resources in your accounts without updating the IAM policies attached to each resource that you own. The RCP limits permissions for resources in member accounts and can impact the effective permissions for identities, including the AWS account root user, regardless of whether they belong to your organization. For more information about Organizations and RCPs, including a list of AWS services that support RCPs, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS Global Accelerator works with IAM

Before you use IAM to manage access to Global Accelerator, learn what IAM features are available to use with Global Accelerator.

To see tables showing a similar high-level view of how AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

IAM features you can use with AWS Global Accelerator

IAM feature	Global Accelerator support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	Yes
ABAC (tags in policies)	Partial
Temporary credentials	Yes
Principal permissions	Yes
Service roles	No
Service-linked roles	Yes

Identity-based policies for Global Accelerator

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

To view examples of Global Accelerator identity-based policies, see Identity-based policy examples for AWS Global Accelerator.

Resource-based policies within Global Accelerator

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM role trust policies and Amazon S3 bucket policies. In services that support resource-based policies, service administrators can use them to control access to a specific resource.

Policy actions for Global Accelerator

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Global Accelerator actions, see Actions defined by AWS Global Accelerator in the *Service Authorization Reference*.

Policy actions in Global Accelerator use the following prefix before the action:

```
aws-globalaccelerator
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "aws-globalaccelerator:action1",
      "aws-globalaccelerator:action2"
         ]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action:

```
"Action": "aws-globalaccelerator:Describe*"
```

To view examples of Global Accelerator identity-based policies, see Identity-based policy examples for AWS Global Accelerator.

Policy resources for Global Accelerator

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

In the *Service Authorization Reference*, you can see the following information related to Global Accelerator:

- To see a list of Global Accelerator resource types and their ARNs, see Resources defined by AWS Global Accelerator.
- To learn the actions that you can specify with the ARN of each resource, see Actions defined by AWS Global Accelerator.

To view examples of Global Accelerator identity-based policies, see Identity-based policy examples for AWS Global Accelerator.

Policy condition keys for Global Accelerator

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of Global Accelerator condition keys, see Condition keys for AWS Global Accelerator in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by AWS Global Accelerator.

To view examples of Global Accelerator identity-based policies, see Identity-based policy examples for AWS Global Accelerator.

ACLs in Global Accelerator

Supports ACLs: Yes

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with Global Accelerator

Supports ABAC (tags in policies): Partial

Global Accelerator has *partial* support for tags in policies. It supports tagging for one resource, accelerators. For more information about using tags in policy statement conditions, and to view an example policy for limiting access to a resource based on tags on the resource, see Using tag-based policies with AWS Global Accelerator.

For more information about tagging Global Accelerator resources, see Tagging in AWS Global Accelerator.

To learn more about using tags in policies, review the following information.

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with Global Accelerator

Supports temporary credentials: Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switch from a user to an IAM role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Cross-service principal permissions for Global Accelerator

Supports forward access sessions (FAS): Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Service roles for Global Accelerator

Supports service roles: No

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

Service-linked role for Global Accelerator

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For more information about the service-linked role for Global Accelerator, see Service-linked role for AWS Global Accelerator.

For details about creating or managing service-linked roles in general in AWS, see AWS services that work with IAM. Find a service in the table that includes a Yes in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for AWS Global Accelerator

By default, users and roles don't have permission to create or modify Global Accelerator resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by Global Accelerator, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS Global Accelerator in the *Service Authorization Reference*.

Topics

- Policy best practices
- Creating a Global Accelerator accelerator
- Using the Global Accelerator console

- Using a Global Accelerator API action
- Allow users to view their own permissions

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Global Accelerator resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Creating a Global Accelerator accelerator

To create an AWS Global Accelerator accelerator, users must have permission to create service-linked roles that are associated with Global Accelerator.

To ensure that users have the correct permissions to create accelerators in Global Accelerator, attach a policy to the user such as the following.

> **Note**
>
> If you create an identity-based permissions policy that is more restrictive than the following policy, users with the more restrictive policy won't be able to create an accelerator.

```
{
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "globalaccelerator.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
      ],
      "Resource": "arn:aws:iam::*:role/aws-service-role/
globalaccelerator.amazonaws.com/AWSServiceRoleForGlobalAccelerator*"
    }
```

Using the Global Accelerator console

To access the AWS Global Accelerator console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Global Accelerator resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the Global Accelerator console, also attach the Global Accelerator `GlobalAcceleratorReadOnlyAccess` or `GlobalAcceleratorFullAccess` AWS managed policy to the entities.

Attach the first policy, `GlobalAcceleratorReadOnlyAccess`, if users only need to view information in the console or make calls to the AWS Command Line Interface or the API that use `List*` or `Describe*` operations.

Attach the second policy, `GlobalAcceleratorFullAccess`, to users who need to create or make updates to accelerators. The full access policy includes *full* permissions for Global Accelerator as well as *describe* permissions for Amazon EC2 and Elastic Load Balancing.

> **Note**
>
> If you create an identity-based permissions policy that does not include the required permissions for Amazon EC2 and Elastic Load Balancing, users with that policy will not be able to add Amazon EC2 and Elastic Load Balancing resources to accelerators.

For more information, see the Global Accelerator AWS managed policies page or Adding permissions to a user in the *IAM User Guide*.

Using a Global Accelerator API action

AWS Global Accelerator supports using actions in a policy. This allows an administrator to control whether an entity can complete an operation in Global Accelerator.

For example, the following policy allows a user to perform the `CreateAccelerator` operation to programmatically create an accelerator in an AWS account:

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "globalaccelerator:CreateAccelerator"
         ],
         "Resource":"*"
      }
   ]
}
```

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
```

```
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Service-linked role for AWS Global Accelerator

AWS Global Accelerator uses an AWS Identity and Access Management (IAM) service-linked role. A service-linked role is a unique type of IAM role that is linked directly to Global Accelerator. The service-linked role is predefined by Global Accelerator and includes all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Global Accelerator easier because you don't have to manually add the necessary permissions. Global Accelerator defines the permissions of its service-linked role, and unless defined otherwise, only Global Accelerator can assume its role. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting its related resources. This protects your Global Accelerator resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS services that work with IAM and look for the services that have **Yes** in the **Service-linked role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for Global Accelerator

AWS Global Accelerator uses a service-linked role named **AWSServiceRoleForGlobalAccelerator**. This role allows Global Accelerator to access resources in your account, such as load balancers

and other endpoints, to help make sure, for example, that you can add only resources that are configured to work with Global Accelerator. The AWSServiceRoleForGlobalAccelerator role also allows Global Accelerator to create and manage resources necessary for client IP address preservation.

Global Accelerator automatically creates a role named AWSServiceRoleForGlobalAccelerator when the role is first required to support a Global Accelerator API operation. This role is required for using accelerators in Global Accelerator. The ARN for the AWSServiceRoleForGlobalAccelerator role looks like this:

```
arn:aws:iam::123456789012:role/aws-service-role/
globalaccelerator.amazonaws.com/AWSServiceRoleForGlobalAccelerator
```

Service-linked role permissions

Global Accelerator uses the service-linked role named **AWSServiceRoleForGlobalAccelerator** to access resources and configurations to check readiness. This service-linked role uses the managed policy `AWSGlobalAcceleratorSLRPolicy`.

The AWSServiceRoleForGlobalAccelerator service-linked role trusts the following service to assume the role:

- `globalaccelerator.amazonaws.com`

To view the permissions for this policy, see AWSGlobalAcceleratorSLRPolicy in the *AWS Managed Policy Reference*.

You must configure permissions to allow an IAM entity (such as a user, group, or role) to delete the Global Accelerator service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating the service-linked role for Global Accelerator

You don't manually create the service-linked role for Global Accelerator. The service creates the role for you automatically the first time that you create an accelerator. If you remove your Global Accelerator resources and delete the service-linked role, the service creates the role again automatically when you create a new accelerator.

Editing the Global Accelerator service-linked role

Global Accelerator does not allow you to edit the AWSServiceRoleForGlobalAccelerator service-linked role. After the service has created a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of a role by using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting the Global Accelerator service-linked role

If you no longer need to use Global Accelerator, we recommend that you delete the service-linked role. That way you don't have unused entities that are not actively monitored or maintained. However, you must clean up the Global Accelerator resources in your account before you can manually delete the roles.

After you have disabled and deleted your accelerators, then you can delete the service-linked role. For more information about deleting accelerators, see Create accelerator.

> **Note**
>
> If you have disabled and deleted your accelerators but Global Accelerator hasn't finished updating, service-linked role deletion might fail. If that happens, wait for a few minutes, and then try the service-linked role deletion steps again.

To manually delete the AWSServiceRoleForGlobalAccelerator service-linked role

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane of the IAM console, choose **Roles**. Then select the check box next to the role name that you want to delete, not the name or row itself.
3. For **Role** actions at the top of the page, choose **Delete role**.
4. In the confirmation dialog box, review the service last accessed data, which shows when each of the selected roles last accessed an AWS service. This helps you to confirm whether the role is currently active. If you want to proceed, choose **Yes, Delete** to submit the service-linked role for deletion.
5. Watch the IAM console notifications to monitor the progress of the service-linked role deletion. Because the IAM service-linked role deletion is asynchronous, after you submit the

role for deletion, the deletion task can succeed or fail. For more information, see Deleting a service-linked role in the *IAM User Guide*.

Updates to the policy for the Global Accelerator service-linked role

For updates to `AWSGlobalAcceleratorSLRPolicy`, the AWS managed policy for the Global Accelerator service-linked role, see the AWS managed policies updates table. You can also subscribe to automatic RSS alerts on the AWS Global Accelerator Document history page.

AWS managed policies for AWS Global Accelerator

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWS managed policy: AWSServiceRoleForGlobalAccelerator

You can't attach `AWSServiceRoleForGlobalAccelerator` to your IAM entities. This policy is attached to a service-linked role that allows AWS Global Accelerator to access AWS services and resources that are used or managed by Global Accelerator. For more information, see Service-linked role for AWS Global Accelerator.

AWS managed policy: GlobalAcceleratorReadOnlyAccess

You can attach `GlobalAcceleratorReadOnlyAccess` to your IAM entities. This policy grants read-only access to actions for working with accelerators in Global Accelerator. It's useful for

users who only need to view information in the console or make calls to the AWS Command Line Interface or the API that use `List*` or `Describe*` operations.

To view the permissions for this policy, see GlobalAcceleratorReadOnlyAccess in the *AWS Managed Policy Reference*.

AWS managed policy: GlobalAcceleratorFullAccess

You can attach `GlobalAcceleratorFullAccess` to your IAM entities. This policy grants full access to actions for working with accelerators in Global Accelerator. Attach it to IAM users and other principals who need full access to Global Accelerator actions.

> **Note**
>
> If you create an identity-based permissions policy that does not include the required permissions for Amazon EC2 and Elastic Load Balancing, users with that policy will not be able to add Amazon EC2 and Elastic Load Balancing resources to accelerators.

To view the permissions for this policy, see GlobalAcceleratorFullAccess in the *AWS Managed Policy Reference*.

Global Accelerator updates to AWS managed policies

View details about updates to AWS managed policies for Global Accelerator since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Global Accelerator Document history page.

Change	Description	Date
AWSGlobalAcceleratorSLRPolicy – Updated policy	Global Accelerator added a new permission to describe target groups on load balancers. Global Accelerator uses `elasticloadbalancing:DescribeTargetGroups` to identify load	October 20, 2023

Change	Description	Date
	balancers with target type `ip`, which is not a supported target type for dual-stack load balancer endpoints in Global Accelerator.	
AWSGlobalAcceleratorSLRPolicy – Updated policy	Global Accelerator added new permissions to describe listeners on load balancers and describe addresses on EC2 instances. Global Accelerator uses `elasticloadbalancing:DescribeListeners` to support making listener management decisions for load balancers, based on listener configurations. Global Accelerator uses `ec2:DescribeAddresses` to add Elastic IP address endpoints to accelerators.	May 23, 2023

Change	Description	Date
AWSGlobalAcceleratorSLRPolicy – Updated policy	Global Accelerator added new permissions to support IPv6 addresses. Global Accelerator uses `ec2:AssignIpv6Addresses` to update the Global Accelerator ENI on a customer subnet with an IPv6 address for sending and receiving IPv6 traffic, and uses `UnassignIpv6Addresses` to remove the IPv6 address when it's no longer needed.	November 15, 2021
AWSGlobalAcceleratorSLRPolicy – Updated policy	Global Accelerator added a new permission to help Global Accelerator to diagnose errors. Global Accelerator uses `ec2:DescribeRegions` to determine the AWS Region that a customer is in, which can help Global Accelerator to troubleshoot errors.	May 18, 2021
Global Accelerator started tracking changes	Global Accelerator started tracking changes for its AWS managed policies.	May 18, 2021

Using tag-based policies with AWS Global Accelerator

When you design IAM policies, you might set granular permissions by granting access to specific resources. However, as the number of resources that you manage grows, this task becomes more

difficult. Tagging a resource, and then using tags in policy statement conditions can make this task easier. You can grant access in bulk to any resource that has a certain tag. You can repeatedly apply this tag to relevant resources, when you create the resource or by updating the resource later.

Using tags in conditions is one way to control access to resources and requests. Tags can be attached to a resource or passed in the request to services that support tagging. In Global Accelerator, only accelerators can include tags. For more information about tagging in Global Accelerator, see Tagging in AWS Global Accelerator.

When you create an IAM policy, you can use tag condition keys to control:

- Which users can perform actions on an accelerator, based on tags that it already has.
- What tags can be passed in an action's request.
- Whether specific tag keys can be used in a request.

For example, the AWS `GlobalAcceleratorFullAccess` managed user policy gives users unlimited permission to perform any Global Accelerator action on any resource. The following policy limits this power and denies unauthorized users permission to perform any Global Accelerator action on any *production* accelerators. A customer's administrator must attach this IAM policy to unauthorized IAM users, in addition to the managed user policy.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Deny",
         "Action":"*",
         "Resource":"*",
         "Condition":{
            "ForAnyValue:StringEquals":{
               "aws:RequestTag/stage":"prod"
            }
         }
      },
      {
         "Effect":"Deny",
         "Action":"*",
         "Resource":"*",
```

```
            "Condition":{
               "ForAnyValue:StringEquals":{
                  "aws:ResourceTag/stage":"prod"
               }
            }
         }
      ]
}
```

For the complete syntax and semantics of tag condition keys, see Control access using IAM tags in the *IAM User Guide*.

Troubleshooting AWS Global Accelerator identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Global Accelerator and IAM.

Topics

- I am not authorized to perform an action in Global Accelerator
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my Global Accelerator resources

I am not authorized to perform an action in Global Accelerator

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `aws-globalaccelerator:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: aws-globalaccelerator:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `aws-globalaccelerator:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Global Accelerator.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Global Accelerator. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my Global Accelerator resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Global Accelerator supports these features, see How AWS Global Accelerator works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.

- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Secure VPC connections in AWS Global Accelerator

When you add a Network Load Balancer, an internal Application Load Balancer, or an Amazon EC2 instance endpoint in AWS Global Accelerator, you enable internet traffic to flow directly to and from the endpoint in Virtual Private Clouds (VPCs) by targeting it in a private subnet. The VPC that contains the load balancer or EC2 instance must have an internet gateway attached to it, to indicate that the VPC accepts internet traffic. However, you don't need public IP addresses on the load balancer or EC2 instance. You also don't need an associated internet gateway route for the subnet.

This is different from the typical internet gateway use case in which both public IP addresses and internet gateway routes are required for internet traffic to flow to instances or load balancers in a VPC. Even if the elastic network interfaces of your targets are present in a public subnet (that is, a subnet with an internet gateway route), when you use Global Accelerator with client IP address preservation enabled, Global Accelerator overrides the typical internet route and all logical connections that arrive through the Global Accelerator also return through Global Accelerator rather than through the internet gateway.

> **Note**
>
> Using public IP addresses and using a public subnet for your Amazon EC2 instances are not typical, though it's possible to set up your configuration with them. Security groups apply to any traffic that arrives to your instances, including traffic from Global Accelerator and any public or Elastic IP address that is assigned to your instance ENI. Use private subnets to ensure that traffic is delivered only by Global Accelerator.
> To learn more about working with ENIs, security groups, and Global Accelerator, see Requirements for endpoints with client IP address preservation.

Keep this information in mind when considering network perimeter issues and configuring IAM privileges related to internet access management. For more information about controlling internet access to your VPC, see this service control policy example.

Logging and monitoring in AWS Global Accelerator

Monitoring is an important part of maintaining the availability and performance of Global Accelerator and your AWS solutions. You should collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your Global Accelerator resources and activity, and responding to potential incidents:

Global Accelerator provides the following three main avenues for logging and tracking:

Amazon CloudWatch metrics and alarms

Using CloudWatch, you can monitor, in real time, your AWS resources and the applications that you run on AWS. As soon as your accelerator is deployed, CloudWatch begins collect and tracking metrics for Global Accelerator. Metrics are variables that you can view for confirmation that traffic is flowing, or that you can measure over time.

You can use metrics, for example, to verify that traffic is flowing through Global Accelerator to your endpoints, and back out to clients, and to help troubleshoot issues. You can also create alarms that watch specific metrics, and then send notifications or automatically make changes to the resources you are monitoring when the metric exceeds a certain threshold for a period of time.

For more information, see Using Amazon CloudWatch with AWS Global Accelerator.

Global Accelerator flow logs

Server flow logs are logs that you set up in Global Accelerator that provide detailed records about traffic that flows through an accelerator to an endpoint. Server flow logs are useful for many applications, for example, for security and access audits. For more information, see Configuring and using flow logs in AWS Global Accelerator.

AWS CloudTrail logs

CloudTrail provides a record of actions taken by a user, role, or an AWS service in Global Accelerator. CloudTrail captures all API calls for Global Accelerator as events, including calls from the Global Accelerator console and from code calls to the Global Accelerator API. For more information, see Using AWS CloudTrail to log AWS Global Accelerator API calls.

Compliance validation for AWS Global Accelerator

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security Compliance & Governance – These solution implementation guides discuss architectural considerations and provide steps for deploying security and compliance features.
- HIPAA Eligible Services Reference – Lists HIPAA eligible services. Not all AWS services are HIPAA eligible.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.

- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in AWS Global Accelerator

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

In addition to the support of AWS global infrastructure, Global Accelerator offers the following features that help support data resiliency:

- Similar to an Availability Zone in AWS, a network zone is an isolated unit with its own set of physical infrastructure. When you create an accelerator, Global Accelerator provides you with a set of static IP addresses: two static IPv4 addresses for an accelerator with an IPv4 IP address type or four static IP addresses for a dual-stack accelerator (two IPv4 addresses and two IPv6 addresses). Global Accelerator serves one static IP address per network zone from a unique IP subnet for each IP address family. If one address from a network zone becomes unavailable, due to IP address blocking by certain client networks or network disruptions, client applications can retry on the healthy static IP address from the other isolated network zone.
- Global Accelerator continuously monitors the health of all endpoints. When it determines that an active endpoint is unhealthy, Global Accelerator instantly begins directing traffic to another available endpoint. This allows you to create a high-availability architecture for your applications on AWS.

Infrastructure security in AWS Global Accelerator

As a managed service, AWS Global Accelerator is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access Global Accelerator through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Quotas for AWS Global Accelerator

Your AWS account has specific quotas, also known as limits, related to AWS Global Accelerator.

The Service Quotas console provides information about Global Accelerator quotas. Along with viewing the default quotas, you can use the Service Quotas console to request quota increases for adjustable quotas.

You must be in the US East (N. Virginia) (us-east-1) Region to manage service limits and request quota increases for Global Accelerator in the Service Quotas console. Global Accelerator service quotas are managed in the US East (N. Virginia) Region because that's where AWS Global Service quotas are defined. In any other AWS Region, you won't see Global Accelerator quotas and can't change the quotas. Note, however, that all Global Accelerator API operations must be run in the US West (Oregon) (us-west-2) Region.

Topics

- General quotas
- Quotas for endpoints per endpoint group
- Related quotas

General quotas

The following are overall quotas for Global Accelerator.

Entity	Quota
Standard accelerators per AWS account	20 You can request a quota increase.
Custom routing accelerators per AWS account	10 You can request a quota increase.
Listeners per accelerator	10 You can request a quota increase.

Entity	Quota
Endpoint groups per accelerator, across all listeners	42
AWS Regions that Global Accelerator can point to, across all listeners and endpoint groups	42 If your accelerator has one listener, you can point to all Global Accelerator supported Regions with your accelerator's endpoint group configuration. Note that the maximum number of Regions that you can reference in an accelerator using endpoint groups decreases proportionally as you increase the number of listeners. Your (total # of listeners) x (# of endpoint groups) must not exceed 42.
Port ranges per listener	10
Port overrides per endpoint group	10 You can request a quota increase.
Principals per cross-account attachment	10 You can request a quota increase.
Resources per cross-account attachment	500

Quotas for endpoints per endpoint group

The following are Global Accelerator quotas that apply to the number of endpoints in endpoint groups.

Entity	Description	Quota
Endpoint groups with more than one endpoint type	Number of endpoints in an endpoint group containing more than one endpoint type.	10
Endpoint groups with just Application Load Balancers	Number of Application Load Balancers in an endpoint group containing only Application Load Balancer endpoints.	10
Endpoint groups with just Network Load Balancers	Number of Network Load Balancers in an endpoint group containing only Network Load Balancer endpoints.	10 You can request a quota increase.
Endpoint groups with just Amazon EC2 instances	Number of EC2 instances in an endpoint group containing only EC2 instance endpoints.	10 You can request a quota increase.
Endpoint groups with just Elastic IP addresses	Number of Elastic IP addresses in an endpoint group containing only Elastic IP address endpoints.	10 You can request a quota increase.
Endpoint groups with just Amazon Virtual Private Cloud subnets	Number of Amazon VPC subnets in an endpoint group containing only subnet endpoints.	10 You can request a quota increase.

Related quotas

In addition to quotas in Global Accelerator, there are quotas that apply to the resources that you use as endpoints for an accelerator. For more information, see the following:

- Elastic IP address quotas in the *Amazon EC2 User Guide*.
- Amazon EC2 service quotas in the *Amazon EC2 User Guide*.
- Quotas for your Network Load Balancers in the *User Guide for Network Load Balancers*.

- Quotas for your Application Load Balancers in the *User Guide for Application Load Balancers*.
- Amazon VPC quotas in the *Amazon VPC User Guide*.

AWS Global Accelerator related information

The information and resources listed here can help you learn more about Global Accelerator.

Topics

- API Reference and product information for AWS Global Accelerator
- Getting support
- Tips from the AWS blog website

API Reference and product information for AWS Global Accelerator

The following related resources can help you as you work with this service.

- AWS Global Accelerator API Reference – Gives complete descriptions of the API actions, parameters, and data types, and a list of errors that the service returns.
- Global Accelerator What's New – Announcements of new Global Accelerator features and recently added edge locations.
- AWS Global Accelerator product information – The primary web page for information about Global Accelerator, including features and pricing information.
- Terms of Use – Detailed information about our copyright and trademark; your account, license, and site access; and other topics.

Getting support

Support for Global Accelerator is available in several forms.

- Discussion forums – A community-based forum for developers to discuss technical questions related to Global Accelerator.
- Support Center – This site brings together information about your recent support cases and results from AWS Trusted Advisor and health checks, as well as providing links to discussion forums, technical FAQs, the service health dashboard, and information about AWS support plans.

- AWS Premium Support Information – The primary web page for information about AWS Premium Support, a one-on-one, fast-response support channel to help you build and run applications on AWS Infrastructure Services.
- Contact Us – Links for inquiring about your billing or account. For technical questions, use the discussion forums or support links above.

Tips from the AWS blog website

The AWS blog website has a number of posts to help you use AWS services, including the following blog posts about Global Accelerator:

- Use AWS Global Accelerator to improve application performance
- Best practices for deployment with AWS Global Accelerator
- Announcing cross-account support for AWS Global Accelerator
- Accessing an Amazon API Gateway via static IP addresses provided by AWS Global Accelerator
- AWS Global Accelerator Custom Routing with Amazon Elastic Kubernetes Service
- Deploying multi-Region applications in AWS using AWS Global Accelerator
- Maximising application resiliency with AWS Global Accelerator
- Starting Small with AWS Global Accelerator
- Traffic management with AWS Global Accelerator
- Analyzing and visualizing AWS Global Accelerator flow logs using Amazon Athena and QuickSight

For a complete list of AWS Global Accelerator blogs, see AWS Global Accelerator in the Networking & Content Delivery category of AWS blog posts.

Document history

The following entries describe important changes made to the AWS Global Accelerator documentation.

- **API version: latest**
- **Latest documentation update:** March 27, 2024

Change	Description	Date
Adds cross-account support for BYOIP	Global Accelerator now supports five additional CloudWatch metrics that you can use to more easily detect issues with your accelerator endpoints. For more infor mation, see Using Amazon CloudWatch with AWS Global Accelerator.	March 27, 2024
Adds cross-account support for BYOIP	Global Accelerator now supports using bring your own IP (BYOIP) addresses across AWS accounts. For more information, see Working with cross-account attachments and resources in AWS Global Accelerator.	March 25, 2024
Adds support for dual-stack for Network Load Balancers	Global Accelerator now supports adding dual-stac k Network Load Balancers to standard accelerators. For more information, see Endpoint requirements for	November 2, 2023

Change	Description	Date
	accelerators in AWS Global Accelerator.	
Adds support for cross-account resources	Global Accelerator now supports adding cross-account resources to accelerators. To add permissions for a cross-account resource, you create a cross-account attachment in Global Accelerator. For more information, see Working with cross-account attachments and endpoints in AWS Global Accelerator.	November 1, 2023
Adds support in four AWS Regions	Support in the following AWS Regions has been added for Global Accelerator: Asia Pacific (Melbourne), Europe (Spain), Europe (Zurich), and Israel (Tel Aviv). For more information, see AWS Region availability for AWS Global Accelerator.	September 26, 2023

Change	Description	Date
Updates service linked role	Adds a new permission, `elasticloadbalancing:DescribeTargetGroups` , to the service. Global Accelerator uses the permission to identify load balancers with target type `ip`, which is not a supported target type for dual-stack load balancer endpoints in Global Accelerator. For more information, see Service-linked role for AWS Global Accelerator.	September 12, 2023
Adds support for client IP address preservation for Network Load Balancers	Global Accelerator now supports enabling client IP address preservation for Network Load Balancers with security groups. For more information, see Adding or updating endpoints with client IP address preservation.	August 22, 2023
Adds IPv6 support for EC2 instances	Global Accelerator now supports adding Amazon EC2 instances to dual-stack accelerators, to enable both IPv4 and IPv6 traffic to EC2 endpoints. For a full list of supported endpoint types and more information, see Endpoints for standard accelerators in AWS Global Accelerator.	August 8, 2023

Change	Description	Date
Added a new Region	Global Accelerator now supports Asia Pacific (Jakarta) . For a full list of supported Regions, see AWS Region availability for AWS Global Accelerator.	June 15, 2023
Added two new Regions	Global Accelerator now supports Asia Pacific (Hyderabad) and Middle East (UAE). For a full list of supported Regions, see AWS Region availability for AWS Global Accelerator.	May 23, 2023
Updates service linked role	Adds new permissions, `elasticloadbalancing:DescribeListeners` and `ec2:DescribeAddresses` , to the service linked role for Global Accelerator, to support making listener management decisions for load balancers , based on listener configurations, and to add Elastic IP address endpoints to accelerators. For more information, see Service-linked role for AWS Global Accelerator.	May 23, 2023

Change	Description	Date
Adds custom routing accelerator quotas	Adds custom routing accelerator quotas. Global Accelerator also has quotas for standard accelerators. For more information, see Quotas for AWS Global Accelerator.	February 13, 2023
Updates the IAM guidance in the guide	Updated guide to align with the IAM best practices . For more information, see Security best practices in IAM.	February 10, 2023
Updates for `AddEndpoints` and `RemoveEndpoints`	Global Accelerator now supports adding and removing endpoints separately from using the `UpdateEndpointGroup` API operation, by using the new `AddEndpoints` and `RemoveEndpoints` API operations. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/global-accelerator-actions.html.	October 20, 2022

Change	Description	Date
Updates for dual-stack accelerators	Global Accelerator now supports dual-stack accelerators. For IPv4, Global Accelerator provides two static IPv4 addresses. For dual-stack, Global Accelerator provides a total of four addresses: two static IPv4 addresses and two static IPv6 addresses. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/what-is-global-accelerator.html.	July 27, 2022
Update to the Global Accelerator existing service-linked role	Global Accelerator added new permissions, `ec2:AssignIpv6Addresses` and `ec2:UnassignIpv6Addresses`, to support IPv6 addresses. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/security-iam-awsmanpol-updates.html.	November 2, 2021
Added new CloudWatch metrics	Global Accelerator has added two new CloudWatch metrics. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/cloudwatch-monitoring.html.	October 28, 2021

Change	Description	Date
Update to the flow logs capture window	Global Accelerator has expanded the flow log capture window from 10 seconds to 60 seconds. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/monitoring-global-accelerator.flow-logs.html.	July 30, 2021
Update to the Global Accelerator existing service-linked role	Global Accelerator added a new permission, `ec2:DescribeRegions`, to allow Global Accelerator to get AWS Region information to help diagnose errors. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/security-iam-awsmanpol-updates.html.	May 7, 2021

Change	Description	Date
Added custom routing accelerators	Global Accelerator introduced a new type of accelerator custom routing accelerators. Custom routing accelerators work well for scenarios where you want to use custom application logic to direct one or more users to a specific destination and port among many, while still gaining the performance benefits of Global Accelerator. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/work-with-custom-routing-accelerators.html.	December 9, 2020
Added port overrides support	Global Accelerator now supports overriding the listener port used for routing traffic to endpoints so you can reroute traffic to specific ports on your endpoints. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/about-endpoint-groups-port-override.html.	October 21, 2020

Change	Description	Date
Added two new Regions	Global Accelerator now supports Africa (Cape Town) and Europe (Milan). For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/preserve-client-ip-address.regions.html.	May 20, 2020
Tagging and BYOIP	This release adds support for adding tags to accelerators and bringing your own IP address to AWS Global Accelerator (BYOIP). For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/tagging-in-global-accelerator.html and https://docs.aws.amazon.com/global-accelerator/latest/dg/using-byoip.html.	February 27, 2020
Updated Security chapter	Added content for compliance, resilience, and infrastructure security. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/security.html.	December 20, 2019

Change	Description	Date
Support for EC2 instances and default DNS name	AWS Global Accelerator now supports adding EC2 instances in supported AWS Regions. In addition, Global Accelerator creates a default DNS name that is mapped to the static IP addresses for your accelerator. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/introduction-how-it-works-client-ip.html and https://docs.aws.amazon.com/global-accelerator/latest/dg/about-accelerators.html#about-accelerators.dns-addressing.	October 29, 2019
Client IP address preservation for Application Load Balancers	You can now choose to have AWS Global Accelerator preserve the client IP address for Application Load Balancers in supported AWS Regions. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/introduction-how-it-works-client-ip.html.	August 28, 2019

Change	Description	Date
Release of AWS Global Accelerator service	The AWS Global Accelerator Developer Guide provides information about setting up and using accelerators—network layer traffic managers—that improve availability and performance for your internet applications that have a global audience.	November 26, 2018