AWS Global Accelerator
Developer Guide
# Table of Contents

What is AWS Global Accelerator? .......................................................................................................................... 1
Components .......................................................................................................................................................... 2
How it works ......................................................................................................................................................... 3
  Idle timeout .................................................................................................................................................... 4
  Static IP addresses ........................................................................................................................................ 5
  Traffic dials and endpoint weights ...................................................................................................................... 6
  Health checks ................................................................................................................................................ 6
Types of accelerators ........................................................................................................................................... 7
Location and IP address ranges of edge servers ................................................................................................. 7
Use cases .......................................................................................................................................................... 8
Speed Comparison Tool ..................................................................................................................................... 9
How to get started ............................................................................................................................................. 9
Tagging ............................................................................................................................................................ 10
  Tagging support in Global Accelerator .......................................................................................................... 10
Adding, editing, and deleting tags in Global Accelerator ......................................................................................... 10
Pricing ............................................................................................................................................................. 11
Getting started .................................................................................................................................................... 12
Getting started with a standard accelerator .......................................................................................................... 12
  Before you begin ............................................................................................................................................ 12
  Step 1: Create an accelerator ......................................................................................................................... 13
  Step 2: Add listeners .................................................................................................................................... 13
  Step 3: Add endpoint groups .......................................................................................................................... 14
  Step 4: Add endpoints .................................................................................................................................. 14
  Step 5: Test your accelerator .......................................................................................................................... 15
  Step 6 (optional): Delete your accelerator ................................................................................................... 15
Getting started with a custom routing accelerator ............................................................................................... 16
  Before you begin ............................................................................................................................................ 16
  Step 1: Create a custom routing accelerator .................................................................................................. 16
  Step 2: Add listeners .................................................................................................................................... 17
  Step 3: Add endpoint groups .......................................................................................................................... 17
  Step 4: Add VPC subnet endpoints .................................................................................................................. 18
  Step 5 (optional): Delete your accelerator ................................................................................................... 18
Actions .............................................................................................................................................................. 20
Work with standard accelerators ......................................................................................................................... 22
  Standard accelerators .................................................................................................................................. 22
  Creating or updating a standard accelerator .................................................................................................... 23
  Deleting an accelerator .................................................................................................................................. 24
  Viewing your accelerators ............................................................................................................................... 24
  Add an accelerator when you create a load balancer .................................................................................... 25
  Using global static IP addresses instead of regional static IP addresses .................................................... 25
  Listeners for standard accelerators ................................................................................................................. 26
  Adding, editing, or removing a standard listener ........................................................................................... 26
  Client affinity ................................................................................................................................................. 27
Endpoint groups for standard accelerators ......................................................................................................... 28
  Adding, editing, or removing a standard endpoint group ................................................................................ 28
  Using traffic dials ......................................................................................................................................... 29
  Port overrides ............................................................................................................................................. 30
  Health check options ................................................................................................................................... 31
Endpoints for standard accelerators ...................................................................................................................... 32
  Adding, editing, or removing a standard endpoint ........................................................................................ 33
  Endpoint weights ......................................................................................................................................... 34
  Adding endpoints with client IP address preservation .................................................................................. 35
  Transitioning endpoints to use client IP address preservation ................................................................... 36
Work with custom routing accelerators .............................................................................................................. 39
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.

- By using 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.
- By using 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, which are listed in the AWS Region Table.

By default, Global Accelerator provides you with two static IP addresses that you associate with your accelerator. With a standard accelerator, instead of using the IP 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. The static IP addresses are anycast from the AWS edge network.

**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 [Tag-based policies](p. 88).

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.

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

For a list of the AWS Regions where Global Accelerator and other services are currently supported, see the AWS Region Table.

**Topics**
- AWS Global Accelerator components (p. 2)
- How AWS Global Accelerator works (p. 3)
- Types of accelerators (p. 7)
- Location and IP address ranges of Global Accelerator edge servers (p. 7)
- AWS Global Accelerator use cases (p. 8)
- AWS Global Accelerator Speed Comparison Tool (p. 9)
- How to get started with AWS Global Accelerator (p. 9)
AWS Global Accelerator components

AWS Global Accelerator includes the following components:

Static IP addresses

Global Accelerator provides you with a set of two static IP addresses that are anycast from the AWS edge network. If you bring your own IP address range to AWS (BYOIP) to use with Global Accelerator, you can instead assign IP addresses from your own pool to use with your accelerator. For more information, see Bring your own IP addresses (BYOIP) in AWS Global Accelerator (p. 53).

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.

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 Tag-based policies (p. 88).

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.

For more information, see Types of accelerators (p. 7).

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. 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.

Network zone

A network zone services the static IP addresses for your accelerator from a unique IP subnet. Similar to an AWS Availability Zone, a network zone is an isolated unit with its own set of physical infrastructure. When you configure an accelerator, by default, Global Accelerator allocates two
IPv4 addresses for it. If one IP address from a network zone becomes unavailable due to IP address blocking by certain client networks, or network disruptions, then 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 virtual private cloud (VPC) subnets with one or many Amazon EC2 instances that are the destinations for traffic.

## 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 virtual private cloud (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 AWS Global Accelerator (p. 53).

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.
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.

With some endpoint types (in some AWS Regions (p. 63)), you have the option to preserve and access the client IP address. Two types of endpoints can preserve the source IP address of the client in incoming packets: Application Load Balancers and Amazon EC2 instances. Global Accelerator does not support client IP address preservation for Network Load Balancer and Elastic IP address endpoints. Endpoints on custom routing accelerators always have the client IP address preserved.

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.

In standard accelerators, Global Accelerator continuously monitors the health of all endpoints, and instantly begins directing traffic 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.

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.

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

Be aware of the following when you use Global Accelerator:

• AWS Direct Connect does not advertise IP address prefixes for AWS Global Accelerator over a public virtual interface. We recommend that you do not advertise IP addresses that you use to communicate with Global Accelerator over your AWS Direct Connect public virtual interface. If you advertise IP addresses that you use to communicate with Global Accelerator over your AWS Direct Connect public virtual interface, it will result in an asymmetric traffic flow: your traffic toward Global Accelerator goes to Global Accelerator over the internet, but return traffic coming to your on-premises network comes over your AWS Direct Connect public virtual interface.

• Global Accelerator does not support adding as an endpoint a resource that belongs to another AWS account.

Topics
• Idle timeout in AWS Global Accelerator (p. 4)
• Static IP addresses in AWS Global Accelerator (p. 5)
• Traffic flow management with traffic dials and endpoint weights (p. 6)
• Health checks for AWS Global Accelerator  (p. 6)

Idle timeout in AWS Global Accelerator

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

• The timeout is 350 seconds for TCP connections to endpoints with client IP address preservation enabled (Application Load Balancers and EC2 instances).

• The timeout is 90 seconds for TCP connections to endpoints without client IP address preservation (Network Load Balancers and Elastic IP addresses).
• The timeout is 30 seconds for UDP connections.

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

To ensure that a TCP connection stays alive, the client must continue to send at least 1 byte of data to the endpoint before the idle timeout period ends, to start a new idle timeout period.

**Note**

For each TCP connection in Global Accelerator, Global Accelerator maintains two connections: a front-end connection (between client and accelerator) and a back-end connection (between accelerator and endpoint). The idle timeout applies to both connections. If no data has been sent or received by the time that the idle timeout period ends, Global Accelerator closes the connections to the client and to the endpoint.

To ensure that a UDP connection stays alive, Global Accelerator must detect new packets for the connection before the idle timeout period ends. For UDP connections, the idle timeout in Global Accelerator affects how long connections are kept in a flow table and marked as active at edge locations. In each edge location, for each new network connection, Global Accelerator maintains a record of the UDP connection in the flow table, as long as packets continue to come through. If Global Accelerator doesn't detect new packets for a connection after the idle timeout period, it removes the connection entry from the flow table.

If more UDP packets arrive after the timeout is exceeded for a specific flow, Global Accelerator re-selects an endpoint target based on health, client location, and policies that you configure.

### Static IP addresses in AWS Global 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 (p. 52).

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 AWS Global Accelerator (p. 53).) 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 (p. 12).

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 Tag-based policies (p. 88).
Traffic flow management 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 Adjusting traffic flow with traffic dials (p. 29).

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 Endpoint weights (p. 34).

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.

Note

For more information about how traffic dials and weights affect failover, see Failover for unhealthy endpoints (p. 35).

Health checks for AWS Global Accelerator

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 Health check options (p. 31).

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.

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 AWS Global Accelerator use cases (p. 8).

To learn more, see Work with standard accelerators in AWS Global Accelerator (p. 22).

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.

To learn more, see Work with custom routing accelerators in AWS Global Accelerator (p. 39).

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

Location and IP address ranges of Global Accelerator edge servers

For a list of Global Accelerator edge server locations, see the Where is AWS Global Accelerator deployed today? section on the AWS Global Accelerator FAQs 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.

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

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 IP 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 particularly useful for IoT, retail, media, automotive, and healthcare use cases in which you can’t easily update client applications frequently.

Acceleration for latency-sensitive applications

Many applications, especially in areas such as gaming, media, mobile apps, 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.

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 configure one or more listeners to process inbound connections from clients, based on the protocol and port (or port range) that you specify.

3. **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.

4. **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 virtual private cloud (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 (p. 12)**. To work
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 and Cost Management 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 Tag-based policies (p. 88).

For usage conventions and links to other resources about tagging, see Tagging 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 (p. 108).

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
- TagResource
- UntagResource
- ListTagsForResource

Tagging support in Global Accelerator

AWS Global Accelerator supports tagging for accelerators.

Global Accelerator supports the tag-based access control feature of AWS Identity and Access Management (IAM). For more information, see Tag-based policies (p. 88).

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.
Note
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

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 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.
Getting started with AWS Global Accelerator

These tutorials provide the steps for getting started with AWS Global Accelerator using the console. You can also use AWS Global Accelerator API operations to create and customize your accelerators. At each step in this tutorial, there's a link to the corresponding API operation for completing the task programmatically. (When you set up a custom routing accelerator, you must use the API for certain configuration steps.) For more information about working with AWS Global Accelerator API operations, see the AWS Global Accelerator API Reference.

Tip
To explore how you can use Global Accelerator to improve performance and availability for web applications, check out the following self-paced workshop: AWS Global Accelerator Workshop.

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

This chapter includes two tutorials: one for creating a standard accelerator and one for creating a custom routing accelerator. To learn more about the two types of accelerators, see Work with standard accelerators in AWS Global Accelerator (p. 22) and Work with custom routing accelerators in AWS Global Accelerator (p. 39).

Topics
- Getting started with a standard accelerator (p. 12)
- Getting started with a custom routing accelerator (p. 16)

Getting started with a standard accelerator

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

Tasks
- Before you begin (p. 12)
- Step 1: Create an accelerator (p. 13)
- Step 2: Add listeners (p. 13)
- Step 3: Add endpoint groups (p. 14)
- Step 4: Add endpoints (p. 14)
- Step 5: Test your accelerator (p. 15)
- Step 6 (optional): Delete your accelerator (p. 15)

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 for Linux Instances.
• Optionally, create one or more Network Load Balancers or Application Load Balancers that includes EC2 instances. For more information, see Create a Network Load Balancer Application 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 (p. 105).

• Global Accelerator requires your router and firewall rules to allow inbound traffic from the IP addresses associated with 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 Amazon Route 53 health checkers in Health Checks for Your Target Groups in the Amazon Route 53 Developer Guide.

Important
Make sure that you're in the AWS Region US West (Oregon). You must be in this Region to create or update accelerators.

Step 1: Create an accelerator

To create your accelerator, you enter a name.

Note
To complete this task by using an API operation instead of the console, see CreateAccelerator in the AWS Global Accelerator API Reference.

To create an accelerator

2. Choose Create accelerator.
3. Provide a name for your accelerator.
4. Optionally, add one or more tags to help you identify your Global Accelerator resources.
5. Choose Next.

Step 2: Add listeners

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

Note
To complete this task by using an API operation instead of the console, see CreateListener in the AWS Global Accelerator API Reference.

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.
Step 3: Add endpoint groups

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

**Note**
To complete this task by using an API operation instead of the console, see CreateEndpointGroup in the AWS Global Accelerator API Reference.

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 Health check options (p. 31).
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.

**Note**
If you're creating your accelerator programmatically, you add endpoints as part of adding endpoint groups. For more information, see CreateEndpointGroup in the AWS Global Accelerator API Reference.

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 Endpoint weights (p. 34).
3. Optionally, for an Application Load Balancer endpoint, under Preserve client IP address, select Preserve address. For more information, see Preserve client IP addresses in AWS Global Accelerator (p. 59).
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 Tag-based policies (p. 88).
- 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

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 that routes traffic deterministically to Amazon EC2 instance destinations in virtual private cloud (VPC) subnet endpoints.

Tasks
- Before you begin (p. 16)
- Step 1: Create a custom routing accelerator (p. 16)
- Step 2: Add listeners (p. 17)
- Step 3: Add endpoint groups (p. 17)
- Step 4: Add endpoints (p. 18)
- Step 5 (optional): Delete your accelerator (p. 18)

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 for Linux Instances.

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 VPCs 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 (p. 105).

Important
Make sure that you’re in the AWS Region US West (Oregon). You must be in this Region to create or update accelerators.

Step 1: Create a custom routing accelerator

Note
To complete this task by using an API operation instead of the console, see CreateCustomRoutingAccelerator in the AWS Global Accelerator API Reference.

To create an accelerator
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.

**Note**
To complete this task by using an API operation instead of the console, see **CreateCustomRoutingListener** in the *AWS Global Accelerator API Reference*.

**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.

**Note**
To complete this task by using an API operation instead of the console, see **CreateCustomRoutingEndpointGroup** in the *AWS Global Accelerator API Reference*.

**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.

Note

The steps here show how to add endpoints in the console. If you're creating your accelerator programmatically, you add endpoints with endpoint groups. For more information, see CreateCustomRoutingEndpointGroup in the AWS Global Accelerator API Reference.

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 Tag-based policies (p. 88).

To delete an accelerator

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 actions that you can use with 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 resources**

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

<table>
<thead>
<tr>
<th>Action</th>
<th>Using the Global Accelerator Console</th>
<th>Using the Global Accelerator API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a standard accelerator</td>
<td>See Getting started with a standard accelerator (p. 12)</td>
<td>See CreateAccelerator</td>
</tr>
<tr>
<td>Create a listener for a standard accelerator</td>
<td>See Listeners for standard accelerators in AWS Global Accelerator (p. 26)</td>
<td>See CreateListener</td>
</tr>
<tr>
<td>Create a endpoint group for a standard accelerator</td>
<td>See Endpoint groups for standard accelerators in AWS Global Accelerator (p. 28)</td>
<td>See CreateEndpointGroup</td>
</tr>
<tr>
<td>Update a standard accelerator</td>
<td>See Standard accelerators in AWS Global Accelerator (p. 22)</td>
<td>See UpdateAccelerator</td>
</tr>
<tr>
<td>List your accelerators</td>
<td>See Viewing your accelerators (p. 24)</td>
<td>See ListAccelerator</td>
</tr>
<tr>
<td>Get all information about an accelerator</td>
<td>See Viewing your accelerators (p. 24)</td>
<td>See DescribeAccelerator</td>
</tr>
<tr>
<td>Delete an accelerator</td>
<td>See Creating or updating a standard accelerator (p. 23)</td>
<td>See DeleteAccelerator</td>
</tr>
</tbody>
</table>

**Actions to use with custom routing resources**

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

<table>
<thead>
<tr>
<th>Action</th>
<th>Using the Global Accelerator Console</th>
<th>Using the Global Accelerator API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a custom routing accelerator</td>
<td>See Getting started with a custom routing accelerator (p. 16)</td>
<td>See CreateCustomRoutingAccelerator</td>
</tr>
<tr>
<td>Action</td>
<td>Using the Global Accelerator Console</td>
<td>Using the Global Accelerator API</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Create a listener for a custom routing accelerator</td>
<td>See Listeners for custom routing accelerators in AWS Global Accelerator (p. 46)</td>
<td>See CreateCustomRoutingListener</td>
</tr>
<tr>
<td>Create an endpoint group for a custom routing accelerator</td>
<td>See Endpoint groups for custom routing accelerators in AWS Global Accelerator (p. 47)</td>
<td>See CreateCustomRoutingEndpointGroup</td>
</tr>
<tr>
<td>Update a custom routing accelerator</td>
<td>See Custom routing accelerators in AWS Global Accelerator (p. 44)</td>
<td>See UpdateCustomRoutingAccelerator</td>
</tr>
<tr>
<td>List your custom routing accelerators</td>
<td>See Viewing your custom routing accelerators (p. 45)</td>
<td>See ListCustomRoutingAccelerator</td>
</tr>
<tr>
<td>Get all information about a custom routing accelerator</td>
<td>See Viewing your custom routing accelerators (p. 45)</td>
<td>See DescribeCustomRoutingAccelerator</td>
</tr>
<tr>
<td>Delete a custom routing accelerator</td>
<td>See Creating or updating a custom routing accelerator (p. 44)</td>
<td>See DeleteCustomRoutingAccelerator</td>
</tr>
<tr>
<td>Get the static port mapping for a custom routing accelerator</td>
<td>N/A</td>
<td>See ListCustomRoutingPortMappings</td>
</tr>
<tr>
<td>Allow all destination traffic for a subnet in a custom routing accelerator</td>
<td>See Adding, editing, or removing a VPC subnet endpoint (p. 49)</td>
<td>See AllowCustomRoutingTraffic</td>
</tr>
<tr>
<td>Deny all destination traffic for a subnet in a custom routing accelerator</td>
<td>See Adding, editing, or removing a VPC subnet endpoint (p. 49)</td>
<td>See DenyCustomRoutingTraffic</td>
</tr>
<tr>
<td>Allow traffic to specific destinations in a custom routing accelerator</td>
<td>See Adding, editing, or removing a VPC subnet endpoint (p. 49)</td>
<td>See AllowCustomRoutingTraffic</td>
</tr>
<tr>
<td>Deny traffic to specific destinations in a custom routing accelerator</td>
<td>See Adding, editing, or removing a VPC subnet endpoint (p. 49)</td>
<td>See DenyCustomRoutingTraffic</td>
</tr>
</tbody>
</table>
Work with standard accelerators in AWS Global Accelerator

This chapter includes procedures and recommendations for creating standard accelerators in AWS Global Accelerator. 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 Work with custom routing accelerators in AWS Global Accelerator (p. 39).

To set up a standard accelerator, do the following:

1. Create an accelerator, and choose the standard accelerator option.
2. Add a listener with a specific set of ports or port range, and choose the protocol to accept: TCP, UDP, or both.
3. Add one or more endpoint groups, one for each AWS Region in which you have endpoint resources.
4. Add one or more endpoints to endpoint groups. This isn't required, but traffic won't be routed if you don't have any endpoints. Endpoints can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses.

The following sections step through working with standard accelerators, listeners, endpoint groups, and endpoints.

Topics
- Standard accelerators in AWS Global Accelerator (p. 22)
- Listeners for standard accelerators in AWS Global Accelerator (p. 26)
- Endpoint groups for standard accelerators in AWS Global Accelerator (p. 28)
- Endpoints for standard accelerators in AWS Global Accelerator (p. 32)

Standard accelerators in AWS Global Accelerator

A standard accelerator in AWS Global Accelerator directs traffic to optimal endpoints over the AWS global network to improve the availability and performance of your internet applications that have a global audience. 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.

When you create an accelerator, by default, Global Accelerator provides you with a set of two static IP addresses. If you bring your own IP address range to AWS (BYOIP), 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 AWS Global Accelerator (p. 53).

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 with Global Accelerator, for example, tag-based permissions, to limit the users who have permissions to delete an accelerator. For more information, see Tag-based policies (p. 88).

This section explains how to create, edit, or delete 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.

**Topics**
- Creating or updating a standard accelerator (p. 23)
- Deleting an accelerator (p. 24)
- Viewing your accelerators (p. 24)
- Add an accelerator when you create a load balancer (p. 25)
- Using global static IP addresses instead of regional static IP addresses (p. 25)

## Creating or updating a standard accelerator

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

**Important**
Make sure that you're in the US West (Oregon) Region. You must be in this Region to create or update accelerators.

### To create a standard accelerator

2. Choose Create accelerator.
3. Provide a name for your accelerator.
4. For Accelerator type, select Standard.
5. 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.
6. Optionally, add one or more tags to help you identify your accelerator resources.
7. Choose Next to add listeners, endpoint groups, and endpoints.

### To edit a standard accelerator

2. In the list of 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 it no longer accepts or routes traffic, or so that you can delete it. Or, if the accelerator is disabled, you can enable it.

4. Choose Save changes.

Deleting an 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.

Important
Global Accelerator is a global service that can front application endpoints in multiple AWS Regions but you must be in the US West (Oregon) Region to delete accelerators by using the AWS Management Console or AWS CLI.

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

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

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.

Viewing 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

2. To see details about an accelerator, in the list, choose an accelerator, and then choose View.
Add an accelerator when you create a load balancer

When you create an Application 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.

**Important**
To create an accelerator, you must have the correct permissions in place. For more information, see Permissions required for console access, authentication management, and access control (p. 83).

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 (p. 52).

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 Viewing your accelerators (p. 24) and Creating or updating a standard accelerator (p. 23).

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 Adding, editing, or removing a standard endpoint (p. 33).
3. Delete the accelerator. For more information, see Deleting an accelerator (p. 24).

Using global static IP addresses instead of 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 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 two global static IP addresses that are announced from AWS edge locations around the world. If you already have
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, UDP, or both 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.

Topics

- Adding, editing, or removing a standard listener (p. 26)
- Client affinity (p. 27)

Adding, editing, or removing a standard listener

This section explains how to work with listeners on the AWS Global Accelerator console. To complete these tasks by using an API operation instead of the console, see CreateListener, UpdateListener, and DeleteListener in the AWS Global Accelerator API Reference.

To add a listener

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 Client affinity (p. 27).

7. Choose Add listener.

**To edit a standard listener**

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 Client affinity (p. 27).

6. Choose Save.

**To remove a listener**

2. On the accelerators page, choose an accelerator.
3. Choose a listener, and then choose Remove.
4. In the confirmation dialog box, choose Remove.

**Client affinity**

If you have stateful applications that you use with a standard accelerator, you can choose to have Global Accelerator direct all requests from a user at a specific source (client) IP address to the same endpoint resource, to maintain client affinity.

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. Global Accelerator honors client affinity after the endpoint group that you select.

**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. 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 AWS Global Accelerator, see the [AWS Global Accelerator API Reference](https://docs.aws.amazon.com/ga/). 

**Topics**

- Adding, editing, or removing a standard endpoint group (p. 28)
- Adjusting traffic flow with traffic dials (p. 29)
- Port overrides (p. 30)
- Health check options (p. 31)

**Adding, editing, or removing 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 work with 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](https://docs.aws.amazon.com/ga/).

**To add a standard endpoint group**

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 Port overrides (p. 30).
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 Health check options (p. 31).
9. Optionally, choose Add endpoint group to add additional endpoint groups for this listener or other listeners.
10. Choose Add endpoint group.

To edit an endpoint group

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.

To remove a standard endpoint group

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.

Adjusting traffic flow with traffic dials

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

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.

---

**Port overrides**

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.

Port overrides can 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, port overrides lets 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 that accepts traffic from users and the endpoint port that Global Accelerator will route that traffic to. For more information, see Adding, editing, or removing a standard endpoint group (p. 28).

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 port overrides, 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.

### Health check options

AWS Global Accelerator regularly sends requests to standard endpoints to test their status. These health checks run automatically. The guidance for determining the health of each endpoint and the timing for the health checks depend on the type of endpoint resource.

**Important**

Global Accelerator requires your router and firewall rules to allow inbound traffic from the IP addresses associated with 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 Amazon Route 53 health checkers in Health Checks for Your Target Groups in the Amazon Route 53 Developer Guide.

You can configure the following health check options for an endpoint group. If you specify health check options, Global Accelerator uses the settings for EC2 instance or Elastic IP address health checks but not for Network Load Balancers or Application Load Balancers.

• For Application Load Balancer or Network Load Balancer endpoints, you configure health checks for the resources by using Elastic Load Balancing configuration options. For more information, see Health Checks for Your Target Groups. Health check options that you choose in Global Accelerator do not affect Application Load Balancers or Network Load Balancers that you've added as endpoints.

  **Note**

  When you have an Application Load Balancer or Network Load Balancer that includes multiple target groups, Global Accelerator considers the load balancer endpoint to be healthy only if each target group behind the load balancer has at least one healthy target. If any single target group for the load balancer has only unhealthy targets, Global Accelerator considers the endpoint to be unhealthy.

• For EC2 instance or Elastic IP address endpoints that are added 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.

• For EC2 instance or Elastic IP address endpoints with UDP listeners, Global Accelerator uses the listener port and the TCP protocol for health checks, so you must have a TCP server on your endpoint.

  **Note**

  Be 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.

**Health check port**

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

*Note*

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.

Each listener routes requests only to healthy endpoints. After you add an endpoint, it must pass a health check to be considered healthy. After each health check is completed, the listener closes the connection that was established for the health check.

---

**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. With standard accelerators, a static IP address serves as a single point of contact for clients, and Global Accelerator then 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.

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.

Be aware of the following for specific types of Global Accelerator standard endpoints:

**Load balancer endpoints**

- An Application Load Balancer endpoint can be internet-facing or internal. A Network Load Balancer endpoint must be internet-facing.

**Amazon EC2 instance endpoints**

- An EC2 instance endpoint (for both standard and custom routing accelerators) 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 only some AWS Regions. For a list of supported Regions, see Supported AWS Regions for client IP address preservation (p. 63).

• 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.

**Topics**

- Adding, editing, or removing a standard endpoint (p. 33)
- Endpoint weights (p. 34)
- Adding endpoints with client IP address preservation (p. 35)
- Transitioning endpoints to use client IP address preservation (p. 36)

---

**Adding, editing, or removing 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.

Endpoints in Global Accelerator can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses. You must create one of those resources 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.

You can add or remove endpoints from endpoint groups based on usage. For example, if demand on your application increases, you can create more resources and then 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.

If you're adding an endpoint with client IP address preservation, first review the information in Supported AWS Regions for client IP address preservation (p. 63) and Preserve client IP addresses in AWS Global Accelerator (p. 59).

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.

This section explains how to work with 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**

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. In the **Endpoints** section, choose **Add endpoint**.

6. 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.

7. 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 **Endpoint weights (p. 34)**.

8. Optionally, enable client IP address preservation for an internet-facing Application Load Balancer endpoint. Under **Preserve client IP address**, select **Preserve address**.

   This option is always selected for internal Application Load Balancer and EC2 instance endpoints, and never selected for Network Load Balancer and Elastic IP address endpoints. For more information, see **Preserve client IP addresses in AWS Global Accelerator (p. 59)**.

   **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.

9. Choose **Add endpoint**.

**To edit a standard endpoint**

You can edit an endpoint configuration to change the weight. For more information, see **Endpoint weights (p. 34)**.


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**.

**To remove an endpoint**


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**.

**Endpoint weights**

A weight is a value 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.
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.

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

How endpoint weights work

To use weights, you assign each endpoint in an endpoint group a relative weight that corresponds with how much traffic 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. 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 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.

Failover for unhealthy endpoints

If there are no healthy endpoints in an endpoint group that have a weight greater than zero, Global Accelerator tries to failover to a healthy endpoint with a weight greater than zero in another endpoint group. 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 three additional endpoint groups (that is, three 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.

Adding endpoints with client IP address preservation

A feature that you can use with some endpoint types—in some Regions—is client IP address preservation. With this feature, you preserve the source IP address of the original client for packets that arrive at the endpoint. You can use this feature with Application Load Balancer and Amazon EC2 instance endpoints. Endpoints on custom routing accelerators always have the client IP address preserved. For more information, see Preserve client IP addresses in AWS Global Accelerator (p. 59).
If you intend to use the client IP address preservation feature, be aware of the following when you add endpoints to Global Accelerator:

**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 client IP address preservation](#) (p. 61).

**Endpoints in private subnets**

You can target an Application Load Balancer or an EC2 instance in a private subnet using AWS 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](#) (p. 105).

**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).

**Note**

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

**Transitioning endpoints to use client IP address preservation**

Follow the guidance in this section to transition one or more endpoints in your accelerator to endpoints that preserve the user’s client IP address. You can optionally choose to transition an Application Load Balancer endpoint or an Elastic IP address endpoint to a corresponding endpoint—an Application Load Balancer or an EC2 instance—that has client IP address preservation. For more information, see [Preserve client IP addresses in AWS Global Accelerator](#) (p. 59).

We recommend that you transition to using client IP address preservation slowly. First, add new Application Load Balancer or EC2 instance endpoints that you enable to preserve the client IP address.
Then slowly move traffic from existing endpoints to the new endpoints by configuring weights on the endpoints.

**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.

Client IP address preservation is available only in specific AWS Regions. For more information, see Supported AWS Regions for client IP address preservation (p. 63).

This section explains how to work with 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.

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.

To transition to endpoints that preserve client IP addresses, start by following the steps here to add a new endpoint and, for internet-facing Application Load Balancer endpoints, enable client IP address preservation. (The client IP address preservation option is always selected for internal Application Load Balancers and EC2 instances.)

**To add an endpoint with client IP address preservation**

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 dropdown list, choose an Application Load Balancer endpoint or an EC2 instance endpoint.
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 Endpoint weights (p. 34).
8. For a new external-facing Application Load Balancer endpoint, under Preserve client IP address, select Preserve address. (This option is always selected for internal Application Load Balancers and EC2 instances.)
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—Application 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.
Work with custom routing accelerators in AWS Global Accelerator

This chapter includes procedures and recommendations for creating custom routing accelerators in AWS Global 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 virtual private cloud (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 AWS Global Accelerator (p. 40).

If instead you want Global Accelerator to automatically choose the closest healthy endpoint to your clients, create a standard accelerator. For more information, see Work with standard accelerators in AWS Global Accelerator (p. 22).

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 (p. 42).
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, and select the option for a custom routing accelerator.
4. Add a listener and 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 (p. 42).
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 step through working with custom routing accelerators, listeners, endpoint groups, and endpoints.

Topics

- How custom routing accelerators work in AWS Global Accelerator (p. 40)
- Guidelines and restrictions for custom routing accelerators (p. 42)
- Custom routing accelerators in AWS Global Accelerator (p. 44)
How custom routing accelerators work in AWS 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 virtual private cloud (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 to 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 you might 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 Adding, editing, or removing a VPC subnet endpoint (p. 49). 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.

<table>
<thead>
<tr>
<th>Global Accelerator listener port</th>
<th>VPC subnet</th>
<th>EC2 instance port</th>
</tr>
</thead>
<tbody>
<tr>
<td>10001</td>
<td>192.0.2.4</td>
<td>81</td>
</tr>
<tr>
<td>10002</td>
<td>192.0.2.4</td>
<td>82</td>
</tr>
<tr>
<td>10003</td>
<td>192.0.2.4</td>
<td>83</td>
</tr>
<tr>
<td>10004</td>
<td>192.0.2.4</td>
<td>84</td>
</tr>
<tr>
<td>10005</td>
<td>192.0.2.4</td>
<td>85</td>
</tr>
<tr>
<td>10006</td>
<td>192.0.2.4</td>
<td>86</td>
</tr>
<tr>
<td>10007</td>
<td>192.0.2.4</td>
<td>87</td>
</tr>
<tr>
<td>10008</td>
<td>192.0.2.4</td>
<td>88</td>
</tr>
<tr>
<td>10009</td>
<td>192.0.2.4</td>
<td>89</td>
</tr>
<tr>
<td>10010</td>
<td>192.0.2.4</td>
<td>90</td>
</tr>
<tr>
<td>10011</td>
<td>192.0.2.5</td>
<td>81</td>
</tr>
<tr>
<td>10012</td>
<td>192.0.2.5</td>
<td>82</td>
</tr>
<tr>
<td>10013</td>
<td>192.0.2.5</td>
<td>83</td>
</tr>
<tr>
<td>10014</td>
<td>192.0.2.5</td>
<td>84</td>
</tr>
<tr>
<td>10015</td>
<td>192.0.2.5</td>
<td>85</td>
</tr>
<tr>
<td>10016</td>
<td>192.0.2.5</td>
<td>86</td>
</tr>
<tr>
<td>10017</td>
<td>192.0.2.5</td>
<td>87</td>
</tr>
<tr>
<td>10018</td>
<td>192.0.2.5</td>
<td>88</td>
</tr>
<tr>
<td>10019</td>
<td>192.0.2.5</td>
<td>89</td>
</tr>
</tbody>
</table>
### 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.

#### Amazon EC2 instance 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 accelerator.

The types of EC2 instances that are supported with Global Accelerator are listed in Endpoints for standard accelerators in AWS Global Accelerator (p. 32).

<table>
<thead>
<tr>
<th>Global Accelerator listener port</th>
<th>VPC subnet</th>
<th>EC2 instance port</th>
</tr>
</thead>
<tbody>
<tr>
<td>10020</td>
<td>192.0.2.5</td>
<td>90</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>12501</td>
<td>192.0.2.244</td>
<td>81</td>
</tr>
<tr>
<td>12502</td>
<td>192.0.2.244</td>
<td>82</td>
</tr>
<tr>
<td>12503</td>
<td>192.0.2.244</td>
<td>83</td>
</tr>
<tr>
<td>12504</td>
<td>192.0.2.244</td>
<td>84</td>
</tr>
<tr>
<td>12505</td>
<td>192.0.2.244</td>
<td>85</td>
</tr>
<tr>
<td>12506</td>
<td>192.0.2.244</td>
<td>86</td>
</tr>
<tr>
<td>12507</td>
<td>192.0.2.244</td>
<td>87</td>
</tr>
<tr>
<td>12508</td>
<td>192.0.2.244</td>
<td>88</td>
</tr>
<tr>
<td>12509</td>
<td>192.0.2.244</td>
<td>89</td>
</tr>
<tr>
<td>12510</td>
<td>192.0.2.244</td>
<td>90</td>
</tr>
<tr>
<td>12511</td>
<td>192.0.3.4</td>
<td>81</td>
</tr>
<tr>
<td>12512</td>
<td>192.0.3.4</td>
<td>82</td>
</tr>
<tr>
<td>12513</td>
<td>192.0.3.4</td>
<td>83</td>
</tr>
<tr>
<td>12514</td>
<td>192.0.3.4</td>
<td>84</td>
</tr>
<tr>
<td>12515</td>
<td>192.0.3.4</td>
<td>85</td>
</tr>
<tr>
<td>12516</td>
<td>192.0.3.4</td>
<td>86</td>
</tr>
<tr>
<td>12517</td>
<td>192.0.3.4</td>
<td>87</td>
</tr>
<tr>
<td>12518</td>
<td>192.0.3.4</td>
<td>88</td>
</tr>
<tr>
<td>12519</td>
<td>192.0.3.4</td>
<td>89</td>
</tr>
<tr>
<td>12520</td>
<td>192.0.3.4</td>
<td>90</td>
</tr>
</tbody>
</table>
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.

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 AWS Global Accelerator (p. 40).

When you create an accelerator, by default, Global Accelerator provides you with a set of two static IP addresses. If you bring your own IP address range to AWS (BYOIP), 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 AWS Global Accelerator (p. 53).

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 Tag-based policies (p. 88).

This section explains how to create, edit, or delete 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.

Topics

- Creating or updating a custom routing accelerator (p. 44)
- Viewing your custom routing accelerators (p. 45)
- Deleting a custom routing accelerator (p. 45)

Creating or updating a custom routing accelerator

Important

Global Accelerator is a global service that can front application endpoints in multiple AWS Regions, but you must be in the US West (Oregon) Region to create or update accelerators by using the AWS Management Console or AWS CLI.

To create a custom routing accelerator

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.

**To edit a custom routing accelerator**

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**.

**Viewing your custom routing accelerators**

You can 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 your custom routing accelerators**

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

**Deleting a custom routing 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.

**Important**
Make sure that you're in the US West (Oregon) Region. You must be in this Region to update or delete accelerators.

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

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

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 AWS 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 AWS Global Accelerator (p. 40).

Adding, editing, or removing a custom routing listener

This section explains how to work with custom routing listeners 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.

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.

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.

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.

To remove a listener

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 AWS 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 AWS Global Accelerator (p. 40).

Adding, editing, or removing an endpoint group for a custom routing 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 work with 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

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.

To edit an endpoint group for a custom routing accelerator

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**.

### To remove a custom routing accelerator


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**.

---

### VPC subnet endpoints for custom routing accelerators in AWS Global Accelerator

Endpoints for custom routing accelerators are 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](https://docs.aws.amazon.com/accelerator/latest/APIReference/API_ListCustomRoutingPortMappings.html).

You can only direct traffic to EC2 instances in the subnets, not other resources like load balancers (in contrast to standard accelerators). The EC2 instance types that are supported are listed in [Endpoints for standard accelerators in AWS Global Accelerator](https://docs.aws.amazon.com/accelerator/latest/userguide/accelerator-endpoints-standard.html) (p. 32).

To learn more, see [How custom routing accelerators work in AWS Global Accelerator](https://docs.aws.amazon.com/accelerator/latest/userguide/accelerator-endpoints-standard.html) (p. 40).

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

- 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 call the [DescribeCustomRoutingAccelerator](https://docs.aws.amazon.com/accelerator/latest/APIReference/API_DescribeCustomRoutingAccelerator.html) API action to check the accelerator status. For more information, see [DescribeCustomRoutingAccelerator](https://docs.aws.amazon.com/accelerator/latest/APIReference/API_DescribeCustomRoutingAccelerator.html).

- 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 [Adding endpoints with client IP address preservation](https://docs.aws.amazon.com/accelerator/latest/userguide/accelerator-endpoints-standard.html#accelerator-endpoints-standard-clientip) (p. 35).

### Adding, editing, or removing a VPC subnet endpoint

You add 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, edit, or remove 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](https://docs.aws.amazon.com/ga2012/api/index.html).

### To add a VPC subnet endpoint

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**.

### To allow or deny traffic to specific destinations

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

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.
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.

To remove an endpoint

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.
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.

Topics
- Support for DNS addressing in Global Accelerator (p. 52)
- Route custom domain traffic to your accelerator (p. 52)
- Bring your own IP addresses (BYOIP) in AWS Global Accelerator (p. 53)

Support for DNS addressing in Global Accelerator

When you create a custom routing or standard accelerator, Global Accelerator provisions two static IP 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. The static IP addresses are advertised globally using anycast from the AWS edge network to your endpoints. You can use your accelerator’s static IP addresses or DNS name to route traffic to your accelerator. DNS servers and DNS resolvers use a round robin 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
Global Accelerator creates two Pointer (PTR) records that map an accelerator’s static IP addresses 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.
Bring your own IP addresses (BYOIP) in AWS Global Accelerator

AWS 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 bring part or all of your public IPv4 address ranges from your on-premises network to your AWS account to use with Global Accelerator. You continue to own the address ranges, but AWS advertises them on the internet.

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.

Topics
- Requirements (p. 53)
- Prepare to bring your IP address range to your AWS account: Authorization (p. 54)
- Provision the address range for use with AWS Global Accelerator (p. 56)
- Advertise the address range through AWS (p. 57)
- Deprovision the address range (p. 58)
- Create an accelerator with your IP addresses (p. 58)

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
The address range must be registered to a business or institutional entity. It can't be registered to an individual.

- The most specific address range that you can bring is /24. The first 24 bits of the IP address specify the network number. For example, 198.51.100.0 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 (p. 55)
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 by doing the following:

      **Important**
      Make sure to include the -----BEGIN CERTIFICATE----- and -----END
      CERTIFICATE----- from the certificate.

      - For ARIN, add the certificate in the Public Comments section for your IP address range.
      - For RIPE, add the certificate as a new descr field for your IP address range.
      - For APNIC, send the public key in email to helpdesk@apnic.net, the APNIC authorized
        contact for the IP addresses, to request that they manually add it to the remarks field.
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 AWS 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 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 provision-byoip-cidr
   --cidr 203.0.113.25/24
   --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 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:

```json
{
    "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 advertise-byoip-cidr --cidr address-range
```

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

```
aws globalaccelerator 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 list-byoip-cidrs
```

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

```json
{
    "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 Deleting an accelerator (p. 24).

```
aws globalaccelerator withdraw-byoip-cidr --cidr address-range
```

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

```
aws globalaccelerator withdraw-byoip-cidr
   --cidr 203.0.113.25/24
```

## 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 Deleting an accelerator (p. 24).

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

```
aws globalaccelerator withdraw-byoip-cidr --cidr address-range
```

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

```
aws globalaccelerator deprovision-byoip-cidr --cidr address-range
```

## Create an accelerator with your IP addresses

Now you can create an accelerator with your IP addresses. 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 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 Creating or updating a standard accelerator (p. 23) and Creating or updating a custom routing accelerator (p. 44).
- **Use the Global Accelerator API to create an accelerator.** For more information, including examples of using the CLI, see CreateAccelerator and CreateCustomRoutingAccelerator in the AWS Global Accelerator API Reference.
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. There are two types of endpoints that can preserve the source IP address of the client in incoming packets: Application Load Balancers and Amazon EC2 instances.

- When you use an internet-facing Application Load Balancer as an endpoint with Global Accelerator, client IP address preservation is enabled by default for new accelerators. This means that the source IP address of the original client is preserved for packets that arrive at the load balancer. 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.

**Note**
Global Accelerator does not support client IP address preservation for Network Load Balancer and Elastic IP address endpoints.

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

- 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.
- Client IP address preservation is supported only in specific AWS Regions. For more information, see Supported AWS Regions for client IP address preservation (p. 63).

**Topics**
- How to enable client IP address preservation (p. 59)
- Benefits of client IP address preservation (p. 60)
- How the client IP address is preserved in AWS Global Accelerator (p. 61)
- Best practices for client IP address preservation (p. 61)
- Supported AWS Regions for client IP address preservation (p. 63)

**How to enable client IP address preservation**

When you create a new accelerator, client IP address preservation is enabled, by default, for supported endpoints.

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. You can disable the option at any time if you don't want client IP address preservation for an internet-facing Application Load Balancer endpoint.

• 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, internet-facing Application Load Balancer endpoints have client IP address preservation enabled by default.

• Global Accelerator does not support client IP address preservation for Network Load Balancer and Elastic IP address endpoints.

For existing accelerators, you can transition endpoints without client IP address preservation to endpoints that do preserve the client IP address. Existing Application Load Balancer endpoints can be transitioned to new Application Load Balancer endpoints, and existing Elastic IP address endpoints can be transitioned to EC2 instance endpoints. (Network Load Balancer endpoints don't support client IP address preservation.) 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 endpoints, first add to Global Accelerator a duplicate Application Load Balancer endpoint that targets the same backends and, if it's an internet-facing Application Load Balancer, enable client IP address preservation 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 (p. 36).

Benefits of client IP address preservation

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.

However, for other applications 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. For more information, see How the client IP address is preserved in AWS Global Accelerator (p. 61). For EC2 instance endpoints, the original client IP address is preserved.

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
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 and Application Load Balancers:

- For an EC2 instance endpoint, the client's IP address is preserved for all traffic.
- 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
```

Best practices for 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 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. The supported endpoints for routing traffic this way are Application Load Balancers (internal and internet-facing) 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 (p. 105).

**How Global Accelerator uses elastic network interfaces**

When you have an Application Load Balancer 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 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 subnet A and subnet B 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 subnet A and one in subnet B.

- **Example 3:** If you add an ALB1 that has elastic network interfaces in subnet A and subnet B to Accelerator1, and then add an ALB2 with elastic network interfaces in subnet A and subnet C to Accelerator2, Global Accelerator creates three elastic network interfaces: one in subnet A, one in subnet B, and one in subnet C. The elastic network interface in subnet A 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, 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 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.

- 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.

- Global Accelerator doesn't delete security groups that it creates. However, Global Accelerator does delete an elastic network interface if it isn't being used by any of the endpoints in accelerators in your account.

- 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.

Supported AWS Regions for client IP address preservation

You can enable client IP address preservation for AWS Global Accelerator in the following AWS Regions.

<table>
<thead>
<tr>
<th>Region Name</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (Ohio)</td>
<td>us-east-2</td>
</tr>
<tr>
<td>US East (N. Virginia)</td>
<td>us-east-1</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>us-west-1 (except AZ usw1-az2)</td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td>us-west-2</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>af-south-1</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>ap-east-1</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>ap-south-1</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>ap-northeast-3</td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>ap-southeast-1</td>
</tr>
<tr>
<td>Asia Pacific (Sydney)</td>
<td>ap-southeast-2</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>ap-northeast-1 (except AZ apne1-az3)</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>ap-northeast-2</td>
</tr>
<tr>
<td>Canada (Central)</td>
<td>ca-central-1 (except AZ cac1-az3)</td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>eu-central-1</td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>eu-west-1</td>
</tr>
<tr>
<td>Europe (London)</td>
<td>eu-west-2</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>eu-south-1</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>eu-west-3</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>eu-north-1</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>me-south-1</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>sa-east-1</td>
</tr>
</tbody>
</table>
Logging and monitoring in AWS Global Accelerator

You can use flow logs and AWS CloudTrail to monitor your accelerator in AWS Global Accelerator, analyze traffic patterns, and troubleshoot issues with your listeners and endpoints.

Topics
- Flow logs in AWS Global Accelerator (p. 64)
- Using Amazon CloudWatch with AWS Global Accelerator (p. 70)
- Using AWS CloudTrail to log AWS Global Accelerator API calls (p. 74)

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.

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 approximately 10 seconds, but can take up to 1 minute.

CloudWatch Logs charges apply when using flow logs, even when logs are published directly to Amazon S3. For more information, see Deliver Logs to S3 at Amazon CloudWatch Pricing.

Topics
- Publishing flow logs to Amazon S3 (p. 64)
- Timing of log file delivery (p. 68)
- Flow log record syntax (p. 68)

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 a Bucket in the Amazon Simple Storage Service Getting Started 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. 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_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:

```
```

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 user, must have sufficient permissions to publish flow logs to the Amazon S3 bucket. The IAM policy must include the following permissions:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DeliverLogs",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowGlobalAcceleratorService",
            "Effect": "Allow",
            "Action": [
                "globalaccelerator:*"
            ],
            "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:

```
{
    "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 How Do I Add an S3 Bucket Policy? in the Amazon Simple Storage Service Getting Started 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`:

```
{
    "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/123123123123/",
            "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"
        }
    ]
}
```
"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 customer master key (CMK), you must add the following to the key policy for your CMK so that flow logs can write log files to the bucket:

```json
{
  "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 Getting Started Guide.

### Enable publishing flow logs to Amazon S3

To enable flow logs in AWS Global Accelerator, follow the steps in this procedure.

**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 (p. 65).
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
   --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.

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:

```plaintext
<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>
```

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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>The flow logs version.</td>
</tr>
<tr>
<td>aws_account_id</td>
<td>The AWS account ID for the flow log.</td>
</tr>
<tr>
<td>accelerator_id</td>
<td>The ID of the accelerator for which the traffic is recorded.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>client_ip</td>
<td>The source IPv4 address.</td>
</tr>
<tr>
<td>client_port</td>
<td>The source port.</td>
</tr>
<tr>
<td>accelerator_ip</td>
<td>The accelerator's IP address.</td>
</tr>
<tr>
<td>accelerator_port</td>
<td>The accelerator's port.</td>
</tr>
<tr>
<td>endpoint_ip</td>
<td>The destination IP address of the traffic.</td>
</tr>
<tr>
<td>endpoint_port</td>
<td>The destination port of the traffic.</td>
</tr>
<tr>
<td>protocol</td>
<td>The IANA protocol number of the traffic. For more information, see <a href="#">Assigned Internet Protocol Numbers</a>.</td>
</tr>
<tr>
<td>ip_address_type</td>
<td>IPv4.</td>
</tr>
<tr>
<td>packets</td>
<td>The number of packets transferred during the capture window.</td>
</tr>
<tr>
<td>bytes</td>
<td>The number of bytes transferred during the capture window.</td>
</tr>
<tr>
<td>start_time</td>
<td>The time, in Unix seconds, of the start of the capture window.</td>
</tr>
<tr>
<td>end_time</td>
<td>The time, in Unix seconds, of the end of the capture window.</td>
</tr>
<tr>
<td>action</td>
<td>The action associated with the traffic:</td>
</tr>
<tr>
<td></td>
<td>• ACCEPT: The recorded traffic was permitted by the security groups or network ACLs. The value is currently always ACCEPT.</td>
</tr>
<tr>
<td>log-status</td>
<td>The logging status of the flow log:</td>
</tr>
<tr>
<td></td>
<td>• OK: Data is logging normally to the chosen destinations.</td>
</tr>
<tr>
<td></td>
<td>• NODATA: There was no network traffic to or from the network interface during the capture window.</td>
</tr>
<tr>
<td></td>
<td>• SKIPDATA: Some flow log records were skipped during the capture window. This can be because of an internal capacity constraint, or an internal error.</td>
</tr>
<tr>
<td>globalaccelerator_source_ip</td>
<td>The IP address used by the Global Accelerator network interface.</td>
</tr>
<tr>
<td>globalaccelerator_source_port</td>
<td>The port used by the Global Accelerator network interface.</td>
</tr>
<tr>
<td>endpoint_region</td>
<td>The AWS Region where the endpoint is located.</td>
</tr>
<tr>
<td>globalaccelerator_region</td>
<td>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.)</td>
</tr>
<tr>
<td>direction</td>
<td>The direction of the traffic. Denotes traffic coming into the Global Accelerator network (INGRESS) or returning to the client (EGRESS).</td>
</tr>
<tr>
<td>vpc_id</td>
<td>The VPC identifier. Included with Version 2.0 flow logs when Global Accelerator sends traffic to an endpoint with client IP address preservation.</td>
</tr>
</tbody>
</table>

If a field does not apply for a specific record, the record displays a '-' symbol for that entry.
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.

You can use metrics to verify that your system is performing as expected. For example, you can create a CloudWatch alarm to monitor a specified metric and initiate an action (such as sending a notification to an email address) if the metric goes outside 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 (p. 70)
- Metric dimensions for accelerators (p. 71)
- Statistics for Global Accelerator metrics (p. 72)
- View CloudWatch metrics for your accelerators (p. 73)

Global Accelerator metrics

The AWS/GlobalAccelerator namespace includes the following metrics.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewFlowCount</td>
<td>The total number of new TCP and UDP flows (or connections) established from clients to endpoints in the time period.</td>
</tr>
<tr>
<td></td>
<td><strong>Reporting criteria</strong>: There is a nonzero value.</td>
</tr>
<tr>
<td></td>
<td><strong>Statistics</strong>: The only useful statistic is Sum.</td>
</tr>
<tr>
<td></td>
<td><strong>Dimensions</strong></td>
</tr>
<tr>
<td></td>
<td>• Accelerator</td>
</tr>
<tr>
<td></td>
<td>• Accelerator, Listener</td>
</tr>
<tr>
<td></td>
<td>• Accelerator, Listener, EndpointGroup</td>
</tr>
<tr>
<td></td>
<td>• Accelerator, SourceRegion</td>
</tr>
<tr>
<td></td>
<td>• Accelerator, DestinationEdge</td>
</tr>
<tr>
<td></td>
<td>• Accelerator, TransportProtocol</td>
</tr>
<tr>
<td></td>
<td>• Accelerator, AcceleratorIPAddress</td>
</tr>
<tr>
<td>ProcessedBytesIn</td>
<td>The total number of incoming bytes processed by the accelerator, including TCP/IP headers. This count includes all traffic to endpoints.</td>
</tr>
<tr>
<td></td>
<td><strong>Reporting criteria</strong>: There is a nonzero value.</td>
</tr>
</tbody>
</table>
Metric dimensions for accelerators

To filter the metrics for your accelerator, use the following dimensions.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerator</td>
<td>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 <code>arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcddefgh</code>, you specify the following: <code>1234abcd-abcd-1234-abcd-1234abcddefgh</code>.</td>
</tr>
<tr>
<td>Listener</td>
<td>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 <code>arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcddefgh/listener/0123wxyz</code>, you specify the following: <code>0123wxyz</code>.</td>
</tr>
<tr>
<td>EndpointGroup</td>
<td>Filters the metric data by endpoint group. Specify the endpoint group by the AWS Region, for example, <code>us-east-1</code> (all lowercase).</td>
</tr>
</tbody>
</table>
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 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

2. In the navigation pane, choose Metrics.
3. Select the GlobalAccelerator namespace.
4. (Optional) To view a metric across all dimensions, type its name in the search field.

To view metrics using the AWS CLI

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

```bash
aws cloudwatch list-metrics --namespace AWS/GlobalAccelerator --region us-west-2
```

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.

```bash
aws cloudwatch get-metric-statistics --namespace AWS/GlobalAccelerator --metrics-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:

```json
{
    "Label": "ProcessedBytesIn",
    "Datapoints": [
        {
            "Timestamp": "2019-12-18T20:45:00Z",
            "Sum": 2410870.0
        }
    ]
}
```
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

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 IAM user credentials
- Whether the request was made with temporary security credentials for a role or federated user
- Whether the request was made by another AWS service

For more information, see the CloudTrail userIdentity Element.

Understanding 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 can contain 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. The log entries are not guaranteed to be in any particular order; they are not an ordered stack trace of API calls.

The following example shows a CloudTrail log entry that includes these 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.

```json
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smith",
        "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": "2018-11-17T21:02:52Z",
  "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": [
        {
          "ipFamily": "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": "11f9a762-8c00-4f3c-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
}
]
Understanding Global Accelerator log file entries

```json
{  
  "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/0339bf6d-13bc-4d45-a114-5d7fexample/listener/abcde1234"  
  },  
  "responseElements": null,  
  "requestID": "9980e368-82fa-40da-95a3-4b0example",  
  "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/0339bf6d-13bc-4d45-a114-5d7fexample/listener/abcde1234"  
  },  
  "responseElements": null,  
  "requestID": "9980e368-82fa-40da-95a3-4b0example",  
  "eventType": "AwsApiCall",  
  "recipientAccountId": "111122223333"  
}
```
"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-b1b0-2dd7example",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }
}
AWS Global Accelerator security

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is 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. The effectiveness of our security is regularly tested and verified by third-party auditors as part of the AWS compliance programs. To learn about the compliance programs that apply to 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 organization's requirements, and applicable laws and regulations.

This documentation will help 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 objectives.

**Topics**
- Identity and access management for AWS Global Accelerator (p. 81)
- Secure VPC connections in AWS Global Accelerator (p. 105)
- Logging and monitoring in AWS Global Accelerator (p. 105)
- Compliance validation for AWS Global Accelerator (p. 106)
- Resilience in AWS Global Accelerator (p. 106)
- Infrastructure security in AWS Global Accelerator (p. 107)

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, including AWS Global Accelerator resources. Administrators use IAM to control who is authenticated (signed in) and authorized (has permissions) to use Global Accelerator resources. IAM is a feature included with your AWS account at no additional charge.

**Important**
If you're not familiar with IAM, review the introductory information on this page, and then see Getting started with IAM (p. 98). Optionally, you can learn more about authentication and access control by viewing What is authentication? (p. 91), What is access control? (p. 92), and What are policies? (p. 95).

**Topics**
- Concepts and terms (p. 82)
- Permissions required for console access, authentication management, and access control (p. 83)
- Understanding how Global Accelerator works with IAM (p. 86)
Concepts and terms

**Authentication** – To sign in to AWS, you must use one of the following: root user credentials (not recommended), IAM user credentials, or temporary credentials using IAM roles. To learn more about these entities, see What is authentication? (p. 91).

**Access control** – AWS administrators use policies to control access to AWS resources, such as accelerators in Global Accelerator. To learn more, see What is access control? (p. 92) and What are policies? (p. 95).

**Important**

All resources in an account are owned by the account, regardless of who created those resources. You must be granted access to create a resource. However, just because you created a resource doesn't mean that you automatically have full access to that resource. An administrator must explicitly grant permissions for each action that you want to perform. That administrator can also revoke your permissions at any time.

To help you understand the basics of how IAM works, review the following terms:

**Resources**

AWS services, such as Global Accelerator and IAM, typically include objects called resources. In most cases, you can create, manage, and delete these resources from the service. IAM resources include users, groups, roles, and policies:

**Users**

An IAM user represents the person or application who uses its credentials to interact with AWS. A user consists of a name, a password to sign in to the AWS Management Console, and up to two access keys that can be used with the AWS CLI or AWS API.

**Groups**

An IAM group is a collection of IAM users. Administrators can use groups to specify permissions for member users. This makes it easier for an administrator to manage permissions for multiple users.

**Roles**

An IAM role does not have any long-term credentials (password or access keys) associated with it. A role can be assumed by anyone who needs it and has permissions. An IAM user can assume a role to temporarily take on different permissions for a specific task. Federated users can assume a role by using an external identity provider that is mapped to the role. Some AWS services can assume a service role to access AWS resources on your behalf.

**Policies**

Policies are JSON documents that define the permissions for the object to which they are attached. AWS supports identity-based policies that you attach to identities (users, groups, or roles). Some AWS services allow you to attach resource-based policies to resources to control what a principal (person or application) can do to that resource. Global Accelerator does not support resource-based policies.

**Identities**

Identities are IAM resources for which you can define permissions. These include users, groups, and roles.

**Entities**

Entities are IAM resources that you use for authentication. These include users and roles.
Principals

In AWS, a principal is a person or application that uses an entity to sign in and make requests to AWS. As a principal, you can use the AWS Management Console, the AWS CLI, or the AWS API to perform an operation (such as deleting an accelerator). This creates a request for that operation. Your request specifies the action, resource, principal, principal account, and any additional information about your request. All of this information provides AWS with context for your request. AWS checks all the policies that apply to the context of your request. AWS authorizes the request only if each part of your request is allowed by the policies.

To view a diagram of the authentication and access control process, see Understanding How IAM Works in the IAM User Guide. For details about how AWS determines whether a request is allowed, see Policy Evaluation Logic in the IAM User Guide.

Permissions required for console access, authentication management, and access control

To use Global Accelerator or to manage authorization and access control for yourself or others, you must have the correct permissions.

Permissions required to create a Global Accelerator accelerator

To create a 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, users with that policy won't be able to create an accelerator.

```json
{
   "Effect": "Allow",
   "Action": [ "iam:CreateServiceLinkedRole", "iam:DeleteServiceLinkedRole", "iam:GetServiceLinkedRoleDeletionStatus" ],
   "Resource": "arn:aws:iam::*:role/aws-service-role/globalaccelerator.amazonaws.com/AWSServiceRoleForGlobalAccelerator"
}
```

Permissions required to use the Global Accelerator console

To access the AWS Global Accelerator console, you must have a minimum set of permissions that allows you to list and view details about the Global Accelerator resources in your AWS account. If you create an identity-based permissions policy that is more restrictive than the minimum required permissions, the console won’t function as intended for entities with that policy.
To ensure that those entities can still use the Global Accelerator console or API actions, also attach one of the following AWS managed policies to the user, as described in Creating Policies on the JSON Tab:

- GlobalAcceleratorReadOnlyAccess
- GlobalAcceleratorFullAccess

Attach the first policy, GlobalAcceleratorReadOnlyAccess, if users only need to view information in the console or make calls to the AWS CLI 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.

The following is the full access policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "globalaccelerator:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeInstances",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeSubnets",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteSecurityGroup",
        "ec2:DescribeSecurityGroups"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/AWSServiceName": "GlobalAccelerator"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateSecurityGroup",
        "ec2:DescribeSecurityGroups"
      ],
      "Resource": "*"
    }
  ]
}
```
Permissions required for console access, authentication management, and access control

Permissions required for authentication management

To manage your own credentials, such as your password, access keys, and multi-factor authentication (MFA) devices, your administrator must grant you the required permissions. To view the policy that includes these permissions, see Allow users to self-manage their credentials (p. 101).

As an AWS administrator, you need full access to IAM so that you can create and manage users, groups, roles, and policies in IAM. You should use the AdministratorAccess AWS managed policy that includes full access to all of AWS. This policy doesn't provide access to the AWS Billing and Cost Management console or allow tasks that require AWS account root user credentials. For more information, see AWS Tasks That Require AWS account root user Credentials in the AWS General Reference.

Warning

Only an administrator user should have full access to AWS. Anyone with this policy has permission to fully manage authentication and access control, in addition to modifying every resource in AWS. To learn how to create this user, see Create your IAM admin user (p. 98).

Permissions required for access control

If your administrator provided you with IAM user credentials, they attached policies to your IAM user to control what resources you can access. To view the policies that are attached to your user identity in the AWS Management Console, you must have the following permissions:

```json
{
    "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": "ListUsersViewGroupsAndPolicies",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
```
If you need additional permissions, ask your administrator to update your policies to allow you to access the actions that you require.

**Understanding how Global Accelerator works with IAM**

Services can work with IAM in several ways:

**Actions**

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, to allow an entity to call the `GetPolicy` AWS API operation to view a policy, an administrator must attach a policy that allows the `iam:GetPolicy` action.

The following example policy allows a user to perform the `CreateAccelerator` operation to programmatically create an accelerator for your AWS account:

```json
{
  "Version": "2018-08-08",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "globalaccelerator:CreateAccelerator"
      ],
      "Resource": "*"
    }
  ]
}
```

**Resource-level permissions**

Global Accelerator supports resource-level permissions. Resource-level permissions allow you to use ARNs to specify individual resources in the policy.

**Resource-based policies**

Global Accelerator does not support resource-based policies. With resource-based policies, you can attach a policy to a resource within the service. Resource-based policies include a `Principal` element to specify which IAM identities can access that resource.

**Authorization based on tags**

Global Accelerator supports authorization-based tags. This feature allows you to use resource tags in the condition of a policy.

**Temporary credentials**

Global Accelerator supports temporary credentials. With temporary credentials, you can sign in with federation, assume an IAM role, or assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as `AssumeRole` or `GetFederationToken`. 

---

86
Service-linked roles

Global Accelerator supports service-linked roles. This feature allows a service to assume a service-linked role on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account, and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

Service roles

Global Accelerator does not support service roles. This feature allows a service to assume a service role on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in your IAM account and are owned by the account. This means that an IAM administrator can change the permissions for this role. However, this might break the functionality of the service.

Troubleshooting authentication and access control

Use the following information to help you diagnose and fix common issues that you might encounter when working with IAM.

Topics

• I am not authorized to perform an action in Global Accelerator (p. 87)
• I’m an administrator and want to allow others to access Global Accelerator (p. 87)
• I want to understand IAM without becoming an expert (p. 87)

I am not authorized to perform an action in Global Accelerator

If the AWS Management Console tells you that you’re not authorized to perform an action, you must contact the administrator who provided you with your user name and password.

The following example occurs when an IAM user named my-user-name tries to use the console to perform the globalaccelerator:CreateAccelerator action but does not have permissions:

```
User: arn:aws:iam::123456789012:user/my-user-name is not authorized to perform: aws-globalaccelerator:CreateAccelerator on resource: my-example-accelerator
```

In this case, ask your administrator to update your policies to allow you to access the my-example-accelerator resource using the aws-globalaccelerator:CreateAccelerator action.

I’m an administrator and want to allow others to access Global Accelerator

To allow others to access Global Accelerator, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in Global Accelerator.

To get started right away, see Getting started with IAM (p. 98).

I want to understand IAM without becoming an expert

To learn more about IAM terms, concepts, and procedures, see the following topics:

• What is authentication? (p. 91)
Tag-based policies

When you design IAM policies, you might set granular permissions by granting access to specific resources. As the number of resources that you manage grows, this task becomes more difficult. Tagging accelerators and using tags in policy statement conditions can make this task easier. You grant access in bulk to any accelerator with a certain tag. Then you repeatedly apply this tag to relevant accelerators, when you create the accelerator or by updating the accelerator later.

Note
Using tags in conditions is one way to control access to resources and requests. For information about tagging in Global Accelerator, see Tagging in AWS Global Accelerator (p. 10).

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. 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 the complete syntax and semantics of tag condition keys, see Control access using IAM tags in the IAM User Guide.

For example, the Global Accelerator 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"
            }
        }
    }
]
}`
Service-linked role for 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 a service. Service-linked roles are predefined by the service and include all of the permissions that the service requires to call other AWS services on your behalf.

Global Accelerator uses the following IAM service-linked role:

- **AWSServiceRoleForGlobalAccelerator**—Global Accelerator uses this role to allow Global Accelerator to create and manage resources required 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. The AWSServiceRoleForGlobalAccelerator role allows Global Accelerator create and manage resources necessary for client IP address preservation. 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
```

A service-linked role makes setting up and using 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 only Global Accelerator can assume the roles. The defined permissions include the trust policy and the permissions policy. The permissions policy cannot be attached to any other IAM entity.

You must remove any associated Global Accelerator resources before you can delete a service-linked role. This helps protect your Global Accelerator resources by making sure that you don't remove a service-linked role that is still required to access active 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.

Service-linked role permissions for Global Accelerator

Global Accelerator uses a service-linked role named **AWSServiceRoleForGlobalAccelerator**. The following sections describe the permissions for the role.

**Service-linked role permissions**

This service-linked role allows Global Accelerator to manage EC2 Elastic Network Interfaces and security groups.

The AWSServiceRoleForGlobalAccelerator service-linked role trusts the following service to assume the role:

- **globalaccelerator.amazonaws.com**

The role permissions policy allows Global Accelerator to complete the following actions on the specified resources:

- Action: ec2:CreateNetworkInterface on arn:aws:lambda:*:*:function:*  
- Action: ec2:DescribeNetworkInterfaces on arn:aws:lambda:*:*:function:*  
- Action: ec2:ModifyNetworkInterfaceAttribute on arn:aws:lambda:*:*:function:*  
- Action: ec2:DeleteNetworkInterface on arn:aws:lambda:*:*:function:*
• Action: ec2:DeleteSecurityGroup on arn:aws:lambda:*:*:function:* when ec2:ResourceTag/AWSServiceName is GlobalAccelerator
• Action: ec2:CreateSecurityGroup on arn:aws:lambda:*:*:function:*
• Action: ec2:DescribeSecurityGroups on arn:aws:lambda:*:*:function:*
• Action: elasticloadbalancing:DescribeLoadBalancers on arn:aws:lambda:*:*:function:*
• Action: ec2:CreateTags on arn:aws:ec2:*:*:security-group/*
• Action: ec2:CreateTags on arn:aws:ec2:*:*:network-interface/*

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 Creating or updating a standard accelerator (p. 23).

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.
Overview of access and authentication

If you're new to IAM, read the following topics to get started with authorization and access in AWS.

Topics

- What is authentication? (p. 91)
- What is access control? (p. 92)
- What are policies? (p. 95)
- Getting started with IAM (p. 98)

What is authentication?

Authentication is how you sign in to AWS using your credentials.

Note
To get started quickly, you can ignore this section. First, review the introductory information on Identity and access management for AWS Global Accelerator (p. 81), and then see Getting started with IAM (p. 98).

As a principal, you must be authenticated (signed in to AWS) using an entity (root user, IAM user, or IAM role) to send a request to AWS. An IAM user can have long-term credentials such as a user name and password or a set of access keys. When you assume an IAM role, you are given temporary security credentials.

To get authenticated from the AWS Management Console as a user, you must sign in with your user name and password. To get authenticated from the AWS CLI or AWS API, you must provide your access key and secret key or temporary credentials. AWS provides SDK and CLI tools to cryptographically sign your request using your credentials. If you don’t use AWS tools, you must sign the request yourself. Regardless of the authentication method that you use, you might also 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.

As a principal, you can sign in to AWS using the following entities (users or roles):

AWS account root user

When you first create an AWS account, you begin with a single 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 do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM user

An IAM user is an entity within your AWS account that has specific permissions. Global Accelerator supports Signature Version 4, a protocol for authenticating inbound API requests. For more
information about authenticating requests, see Signature Version 4 Signing Process in the AWS General Reference.

IAM role

An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity with permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. IAM roles with temporary credentials are useful in the following situations:

Federated user access

Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as federated users. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated users and roles in the IAM User Guide.

Temporary user permissions

An IAM user can assume a role temporarily to take on different permissions for a specific task.

Cross-account access

You can use an IAM role to allow 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). Global Accelerator does not support these resource-based policies. For more information about choosing whether to use a role or a resource-based policy to allow cross-account access, see Controlling access to Principals in a different account (p. 94).

AWS service access

A service role is an IAM role that a service assumes to perform actions on your behalf. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the IAM User Guide.

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 Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the IAM User Guide.

What is access control?

After you sign in (are authenticated) to AWS, your access to AWS resources and operations is governed by policies. Access control is also known as authorization.

Note
To get started quickly, you can ignore this page. First, review the introductory information on Identity and access management for AWS Global Accelerator (p. 81), and then see Getting started with IAM (p. 98).

During authorization, AWS uses values from the request context to check for policies that apply. It then uses the policies to determine whether to allow or deny the request. Most policies are stored
in AWS as JSON documents and specify the permissions that are allowed or denied for principals. For more information about the structure and contents of JSON policy documents, see What are policies? (p. 95).

Policies let an administrator specify who has access to AWS resources and what actions they can perform on those resources. Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even view their own access keys. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or they can add the user to a group that has the intended permissions. When an administrator then gives permissions to a group, all users in that group get those permissions.

You might have valid credentials to authenticate your requests, but unless an administrator grants you permissions you cannot create or access AWS Global Accelerator resources. For example, you must have explicit permissions to create an AWS Global Accelerator accelerator.

As an administrator, you can write a policy to control access to the following:

- **Principals (p. 93)** – Control what the person or application making the request (the principal) is allowed to do.
- **IAM identities (p. 93)** – Control which IAM identities (groups, users, and roles) can be accessed and how.
- **IAM policies (p. 93)** – Control who can create, edit, and delete customer managed policies, and who can attach and detach all managed policies.
- **AWS resources (p. 94)** – Control who has access to resources using an identity-based policy or a resource-based policy.
- **AWS accounts (p. 94)** – Control whether a request is allowed only for members of a specific account.

**Controlling access for principals**

Permissions policies control what you, as a principal, are allowed to do. An administrator must attach an identity-based permissions policy to the identity (user, group, or role) that provides your permissions. Permissions policies allow or deny access to AWS. Administrators can also set a permissions boundary for an IAM entity (user or role) to define the maximum permissions that the entity can have. Permissions boundaries are an advanced IAM feature. For more information about permissions boundaries, see Permissions boundaries for IAM identities in the IAM User Guide.

For more information and an example of how to control AWS access for principals, see Controlling access for principals in the IAM User Guide.

**Controlling access to identities**

Administrators control what you can do to an IAM identity (user, group, or role) by creating a policy that limits what can be done to an identity or who can access it. Then they attach that policy to the identity that provides your permissions.

For example, an administrator might allow you to reset the password for three specific users. To do this, they attach a policy to your IAM user that allows you to reset the password for only yourself and users with the ARN of the three specified users. This allows you to reset the password of your team members but not other IAM users.

For more information and an example of using a policy to control AWS access to identities, see Controlling access to identities in the IAM User Guide.

**Controlling access to policies**

Administrators can control who can create, edit, and delete customer managed policies, and who can attach and detach all managed policies. When you review a policy, you can view the policy summary
that includes a summary of the access level for each service within that policy. AWS categorizes each service action into one of four access levels based on what each action does: List, Read, Write, or Permissions management. You can use these access levels to determine which actions to include in your policies. For more information, see Understanding access level summaries within policy summaries in the IAM User Guide.

**Warning**
You should limit Permissions Management access-level permissions in your account. Otherwise, your account members can create policies for themselves with more permissions than they should have. Or they can create separate users with full access to AWS.

For more information and an example for how to control AWS access to policies, see Controlling access to policies in the IAM User Guide.

**Controlling access to resources**

Administrators can control access to resources using an identity-based policy or a resource-based policy. In an identity-based policy, you attach the policy to an identity and specify what resources that identity can access. In a resource-based policy, you attach a policy to the resource that you want to control. In the policy, you specify which principals can access that resource.

For more information, see Controlling Access to Resources in the IAM User Guide.

**Resource creators do not automatically have permissions**

All resources in an account are owned by the account, regardless of who created those resources. The AWS account root user is the account owner, and therefore has permission to perform any action on any resource in the account.

**Important**
We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, follow the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks. To view the tasks that require you to sign in as the root user, see AWS Tasks That Require Root User.

Entities (users or roles) in the AWS account must be granted access to create a resource. But just because they create a resource doesn't mean they automatically have full access to that resource. Administrators must explicitly grant permissions for each action. Additionally, administrators can revoke those permissions at any time, as long as they have access to manage user and role permissions.

**Controlling access to Principals in a different account**

Administrators can use AWS resource-based policies, IAM cross-account roles, or the AWS Organizations service to allow principals in another account to access resources in your account.

For some AWS services, administrators can grant cross-account access to your resources. To do this, an administrator attaches a policy directly to the resource that they want to share, instead of using a role as a proxy. If the service supports this policy type, then the resource that the administrator shares must also support resource-based policies. Unlike a user-based policy, a resource-based policy specifies who (in the form of a list of AWS account ID numbers) can access that resource. Global Accelerator does not support resource-based policies.

Cross-account access with a resource-based policy has some advantages over a role. With a resource that is accessed through a resource-based policy, the principal (person or application) still works in the trusted account and does not have to give up their user permissions in place of the role permissions. In other words, the principal has access to resources in the trusted account and in the trusting account at the same time. This is useful for tasks such as copying information from one account to another. For more information about using cross-account roles, see Providing Access to an IAM User in Another AWS Account That You Own in the IAM User Guide.
AWS Organizations offers policy-based management for multiple AWS accounts that you own. With Organizations, you can create groups of accounts, automate account creation, and apply and manage policies for those groups. Organizations enables you to centrally manage policies across multiple accounts, without requiring custom scripts and manual processes. Using AWS Organizations, you can create Service Control Policies (SCPs) that centrally control AWS service use across AWS accounts. For more information, see What Is AWS Organizations? in the AWS Organizations User Guide.

What are policies?

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources.

**Note**

To get started quickly, you can ignore this page. First, review the introductory information on Identity and access management for AWS Global Accelerator (p. 81), and then see Getting started with IAM (p. 98).

A policy is an object in AWS that, when associated with an entity or resource, defines their permissions. AWS evaluates these policies when a principal, such as a user, makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, if a policy allows the GetUser action, then a user with that policy can get user information from the AWS Management Console, the AWS CLI, or the AWS API. When you create an IAM user, you can set up the user to allow console or programmatic access. The IAM user can sign in to the console using a user name and password. Or they can use access keys to work with the CLI or API.

The following policy types, listed in order of frequency, can affect whether a request is authorized. For more details, see Policy Types in the IAM User Guide.

**Identity-based policies**

You can attach managed and inline policies to IAM identities (users, groups to which users belong, and roles).

**Resource-based policies**

You can attach inline policies to resources in some AWS services. The most common examples of resource-based policies are Amazon S3 bucket policies and IAM role trust policies. Global Accelerator does not support resource-based policies.

**Organizations SCPs**

You can use an AWS Organizations service control policy (SCP) to apply a permissions boundary to an AWS Organizations organization or organizational unit (OU). Those permissions are applied to all entities within the member accounts.

**Access control lists (ACLs)**

You can use ACLs to control what principals can access a resource. ACLs are similar to resource-based policies, although they are the only policy type that does not use the JSON policy document structure. Global Accelerator supports OR does not support ACLs.

These policies types can be categorized as permissions policies or permissions boundaries.

**Permissions policies**

You can attach permissions policies to a resource in AWS to define the permissions for that object. Within a single account, AWS evaluates all permissions policies together. Permissions policies are the most common policies. You can use the following policy types as permissions policies:
Identity-based policies

When you attach a managed or inline policy to an IAM user, group, or role, the policy defines the permissions for that entity.

Resource-based policies

When you attach a JSON policy document to a resource, you define the permissions for that resource. The service must support resource-based policies.

Access control lists (ACLs)

When you attach an ACL to a resource, you define a list of principals with permission to access that resource. The resource must support ACLs.

Permissions boundaries

You can use policies to define the permissions boundary for an entity (user or role). A permissions boundary controls the maximum permissions that an entity can have. Permissions boundaries are an advanced AWS feature. When more than one permissions boundary applies to a request, AWS evaluates each permissions boundary separately. You can apply a permissions boundary in the following situations:

Organizations

You can use an AWS Organizations service control policy (SCP) to apply a permissions boundary to an AWS Organizations organization or organizational unit (OU).

IAM users or roles

You can use a managed policy for a user's or role's permissions boundary. For more information, see Permissions Boundaries for IAM Entities in the IAM User Guide.

Topics

- Identity-based policies (p. 96)
- Resource-based policies (p. 97)
- Policy access level classifications (p. 97)

Identity-based policies

You can attach policies to IAM identities. For example, you can do the following:

Attach a permissions policy to a user or a group in your account

To grant a user permissions to create an AWS Global Accelerator resource, such as an accelerator, you can attach a permissions policy to a user or a group to which the user belongs.

Attach a permissions policy to a role (grant cross-account permissions)

You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, the administrator in account A can create a role to grant cross-account permissions to another AWS account (for example, account B) or an AWS service as follows:

1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in account A.
2. Account A administrator attaches a trust policy to the role identifying account B as the principal who can assume the role.
3. Account B administrator can then delegate permissions to assume the role to any users in account B. Doing this allows users in account B to create or access resources in account A. The principal in the trust policy can also be an AWS service principal if you want to grant an AWS service permissions to assume the role.
For more information about using IAM to delegate permissions, see Access Management in the IAM User Guide.

For more information about users, groups, roles, and permissions, see Identities (Users, Groups, and Roles) in the IAM User Guide.

The following are two examples of policies that you could use with Global Accelerator. The first example policy grants a user programmatic access to all List and Describe actions for accelerators in your AWS account:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "globalaccelerator:List*",
                "globalaccelerator:Describe*"
            ],
            "Resource": "*"
        }
    ]
}
```

The following example grants programmatic access to the ListAccelerators operation:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "globalaccelerator:ListAccelerators"
            ],
            "Resource": "*"
        }
    ]
}
```

**Resource-based policies**

Resource-based policies are JSON policy documents that you attach to a resource. These policies allow you to specify what actions a specified principal can perform on that resource and under what conditions. The most common resource-based policy is for an Amazon S3 bucket. Resource-based policies are inline policies that exist only on the resource. There are no managed resource-based policies.

Granting permissions to members of other AWS accounts using a resource-based policy has some advantages over an IAM role. For more information, see How IAM Roles Differ from Resource-based Policies in the IAM User Guide.

**Policy access level classifications**

In the IAM console, actions are grouped using the following access-level classifications:

**List**

Provides permission to list resources within the service to determine whether an object exists. Actions with this level of access can list objects but cannot see the contents of a resource. Most actions with the List access level cannot be performed on a specific resource. When you create a policy statement with these actions, you must specify All resources ("*").
Read

Provides permission to read but not edit the contents and attributes of resources in the service. For example, the Amazon S3 operations GetObject and GetBucketLocation have the Read access level.

Write

Provides permission to create, delete, or modify resources in the service. For example, the Amazon S3 operations CreateBucket, DeleteBucket, and PutObject have the Write access level.

Permissions management

Provides permission to grant or modify resource permissions in the service. For example, most IAM and AWS Organizations policy actions have the Permissions management access level.

Tip
To improve the security of your AWS account, restrict or regularly monitor policies that include the Permissions management access-level classification.

Tagging

Provides permission to create, delete, or modify tags that are attached to a resource in the service. For example, the Amazon EC2 CreateTags and DeleteTags operations have the Tagging access level.

Getting started with IAM

AWS Identity and Access Management (IAM) is an AWS service that allows you manage access to services and resources securely. IAM is a feature of your AWS account offered at no additional charge.

Note
Before you start with IAM, review the introductory information on Identity and access management for AWS Global Accelerator (p. 81).

When you first create an AWS account, you begin with a single 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 do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

Create your IAM admin user

To create an administrator user for yourself and add the user to an administrators group (console)

1. Sign in to the IAM console as the account owner by choosing Root user and entering your AWS account email address. On the next page, enter your password.
   
   Note
   We strongly recommend that you adhere to the best practice of using the Administrator IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.

2. In the navigation pane, choose Users and then choose Add user.
3. For User name, enter Administrator.
4. Select the check box next to AWS Management Console access. Then select Custom password, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to User must create a new password at next sign-in to allow the new user to reset their password after they sign in.

6. Choose Next: Permissions.
7. Under Set permissions, choose Add user to group.
8. Choose Create group.
9. In the Create group dialog box, for Group name enter Administrators.
10. Choose Filter policies, and then select AWS managed -job function to filter the table contents.
11. In the policy list, select the check box for AdministratorAccess. Then choose Create group.

Note
You must activate IAM user and role access to Billing before you can use the AdministratorAccess permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.

12. Back in the list of groups, select the check box for your new group. Choose Refresh if necessary to see the group in the list.
13. Choose Next: Tags.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM entities in the IAM User Guide.
15. Choose Next: Review to see the list of group memberships to be added to the new user. When you are ready to proceed, choose Create user.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access management and Example policies.

Create delegated users for Global Accelerator

To support multiple users in your AWS account, you must delegate permission to allow other people to perform only the actions that you want to allow. To do this, create an IAM group with the permissions those people need and then add IAM users to the necessary groups as you create them. You can use this process to set up the groups, users, and permissions for your entire AWS account. This solution is best used by small and medium organizations where an AWS administrator can manually manage the users and groups. For large organizations, you can use custom IAM roles, federation, or single sign-on.

In the following procedure, you create three users named arnav, carlos, and martha and attach a policy that grants permission to create an accelerator named my-example-accelerator, but only within the next 30 days. You can use the steps provided here to add users with different permissions.

To create a delegated user for someone else (console)

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, choose Users, and then choose Add user.
3. For User name, enter arnav.
4. Choose Add another user and enter carlos for the second user. Then choose Add another user and enter martha for the third user.
5. Select the check box next to AWS Management Console access, and then select Autogenerated password.
6. Clear the check box next to User must create a new password at next sign-in to allow the new user to reset their password after they sign in.
7. Choose **Next: Permissions**.
8. Choose **Attach existing policies directly**. You will create a new managed policy for the users.
9. Choose **Create policy**.

The **Create policy** wizard opens in a new tab or browser window.

10. On the **Visual editor** tab, choose **Choose a service**. Then choose Global Accelerator. You can use the search box at the top to limit the results in the list of services.

The **Service** section closes, and the **Actions** section opens automatically.

11. Choose the Global Accelerator actions that you want to allow. For example, to grants permission to create an accelerator, enter `globalaccelerator:CreateAccelerator` in the **Filter actions** text box. When the list of Global Accelerator actions is filtered, select the check box next to `globalaccelerator:CreateAccelerator`.

The Global Accelerator actions are grouped by access-level classification to make it easy for you to quickly determine the level of access that each action provides. For more information, see **Policy access level classifications** (p. 97).

12. If the actions that you selected in the preceding steps do not support choosing specific resources, then **All resources** is selected for you. In that case, you cannot edit this section.

If you chose one or more actions that support resource-level permissions, then the visual editor lists those resource types in the **Resources** section. Choose **You chose actions that require the accelerator resource type** to choose whether you want to enter a specific accelerator for your policy.

13. If you want to allow the `globalaccelerator:CreateAccelerator` action for all resources, choose **All resources**.

If you want to specify a resource, choose **Add ARN**. Specify the region and account ID (or account ID) (or choose **Any**), and then enter `my-example-accelerator` for the resource. Then choose **Add**.

14. Choose **Specify request conditions (optional)**.

15. Choose **Add condition** to grants permission to create an accelerator within the next 7 days. Assume that today's date is January 1, 2019.

16. For **Condition Key**, choose `aws:CurrentTime`. This condition key checks the date and time that the user makes the request. It returns true (and therefore allows the `globalaccelerator:CreateAccelerator` action only if the date and time are within the specified range.

17. For **Qualifier**, keep the default value.

18. To specify the start of the allowed date and time range, for **Operator**, choose **DateGreaterThan**. Then for **Value**, enter `2019-01-01T00:00:00Z`.

19. Choose **Add** to save your condition.

20. Choose **Add another condition** to specify the end date.

21. Follow similar steps to specify the end of the allowed date and time range. For **Condition Key**, choose `aws:CurrentTime`. For **Operator**, choose **DateLessThan**. For **Value**, enter `2019-01-06T23:59:59Z`, seven days after the first date. Then choose **Add** to save your condition.

22. (Optional) To see the JSON policy document for the policy that you are creating, choose the **JSON** tab. You can switch between the **Visual editor** and **JSON** tabs any time. However, if you make changes or choose **Review policy** in the **Visual editor** tab, IAM might restructure your policy to optimize it for the visual editor. For more information, see **Policy Restructuring** in the **IAM User Guide**.

23. When you are finished, choose **Review policy**.

24. On the **Review policy** page, for **Name**, enter `globalaccelerator:CreateAcceleratorPolicy`. For **Description**, enter **Policy to grants permission to create an accelerator**. Review the policy summary to make sure that you have granted the intended permissions, and then choose **Create policy** to save your new policy.
25. Return to the original tab or window, and refresh your list of policies.
26. In the search box, enter `globalaccelerator:CreateAcceleratorPolicy`. Select the check box next to your new policy. Then choose Next Step.
27. Choose Next: Review to preview your new users. When you are ready to proceed, choose Create users.
28. Download or copy the passwords for your new users and deliver them to the users securely.
   Separately, provide your users with a link to your IAM user console page and the user names that you just created.

**Allow users to self-manage their credentials**

You must have physical access to the hardware that will host the user's virtual MFA device in order to configure MFA. For example, you might configure MFA for a user who will use a virtual MFA device running on a smartphone. In that case, you must have the smartphone available in order to finish the wizard. Because of this, you might want to let users configure and manage their own virtual MFA devices. In that case, you must grant users the permissions to perform the necessary IAM actions.

**To create a policy to allow credential self-management (console)**

1. Sign in to the AWS Management Console and open the IAM console at [https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. In the navigation pane, choose Policies, and then choose Create policy.
3. Choose the JSON tab and copy the text from the following JSON policy document. Paste this text into the JSON text box.

   **Important**
   
   This example policy does not allow users to reset their password while signing in. New users and users with an expired password might try to do so. You can allow this by adding `iam:ChangePassword` and `iam:CreateLoginProfile` to the statement `BlockMostAccessUnlessSignedInWithMFA`. However, IAM does not recommend this.

   ```json
   {
   "Version": "2012-10-17",
   "Statement": [
   {
   "Sid": "AllowAllUsersToListAccounts",
   "Effect": "Allow",
   "Action": [
   "iam:ListAccountAliases",
   "iam:ListUsers",
   "iam:ListVirtualMFADevices",
   "iam:GetAccountPasswordPolicy",
   "iam:GetAccountSummary"
   ],
   "Resource": "*"
   },
   {
   "Sid": "AllowIndividualUserToSeeAndManageOnlyTheirOwnAccountInformation",
   "Effect": "Allow",
   "Action": [
   "iam:ChangePassword",
   "iam:CreateAccessKey",
   "iam:CreateLoginProfile",
   "iam:DeleteAccessKey",
   "iam:DeleteLoginProfile",
   "iam:GetLoginProfile",
   "iam:ListAccessKeys",
   "iam:UpdateAccessKey",
   "iam:UpdateLoginProfile",
   "iam:DescribeAccount"
   ],
   "Resource": "*"
   }
   ]
   }"
"iam:ListSigningCertificates",
"iam:DeleteSigningCertificate",
"iam:UpdateSigningCertificate",
"iam:UploadSigningCertificate",
"iam:ListSSHPublicKeys",
"iam:GetSSHPublicKey",
"iam:DeleteSSHPublicKey",
"iam:UpdateSSHPublicKey",
"iam:UploadSSHPublicKey"
],
"Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
"Sid": "AllowIndividualUserToViewAndManageTheirOwnMFA",
"Effect": "Allow",
"Action": [
"iam:CreateVirtualMFADevice",
"iam:DeleteVirtualMFADevice",
"iam:EnableMFADevice",
"iam:ListMFADevices",
"iam:ResyncMFADevice"
],
"Resource": [
"arn:aws:iam::*:mfa/${aws:username}"
,"arn:aws:iam::*:user/${aws:username}"
]
},
{
"Sid": "AllowIndividualUserToDeactivateOnlyTheirOwnMFAOnlyWhenUsingMFA",
"Effect": "Allow",
"Action": [
"iam:DeactivateMFADevice"
],
"Resource": [
"arn:aws:iam::*:mfa/${aws:username}"
,"arn:aws:iam::*:user/${aws:username}"
]
,"Condition": {
"Bool": {
"aws:MultiFactorAuthPresent": "true"
}
}
},
{
"Sid": "BlockMostAccessUnlessSignedInWithMFA",
"Effect": "Deny",
"NotAction": [
"iam:CreateVirtualMFADevice",
"iam:DeleteVirtualMFADevice",
"iam:ListVirtualMFADevices",
"iam:EnableMFADevice",
"iam:ResyncMFADevice",
"iam:ListAccountAliases",
"iam:ListUsers",
"iam:ListSSHPublicKeys",
"iam:ListSSHAccessKeys",
"iam:ListServiceSpecificCredentials",
"iam:ListMFADevices",
"iam:GetAccountSummary",
"sts:GetSessionToken"
],
"Resource": "*",
"Condition": {
"BoolIfExists": {
"aws:MultiFactorAuthPresent": "false"
}
}
What does this policy do?

- The `AllowAllUsersToListAccounts` statement enables the user to see basic information about the account and its users in the IAM console. These permissions must be in their own statement because they do not support or do not need to specify a specific resource ARN, and instead specify "Resource" : "*".

- The `AllowIndividualUserToSeeAndManageOnlyTheirOwnAccountInformation` statement enables the user to manage his or her own user, password, access keys, signing certificates, SSH public keys, and MFA information in the IAM console. It also allows users to sign in for the first time in an administrator requires them to set a first-time password. The resource ARN limits the use of these permissions to only the user's own IAM user entity.

- The `AllowIndividualUserToViewAndManageTheirOwnMFA` statement enables the user to view or manage his or her own MFA device. Notice that the resource ARNs in this statement allow access to only an MFA device or user that has the same name as the currently signed-in user. Users can't create or alter any MFA device other than their own.

- The `AllowIndividualUserToDeactivateOnlyTheirOwnMFAOnlyWhenUsingMFA` statement allows the user to deactivate only his or her own MFA device, and only if the user signed in using MFA. This prevents others with only the access keys (and not the MFA device) from deactivating the MFA device and accessing the account.

- The `BlockMostAccessUnlessSignedInWithMFA` statement uses a combination of "Deny" and "NotAction" to deny access to all but a few actions in IAM and other AWS services if the user is not signed-in with MFA. For more information about the logic for this statement, see `NotAction with Deny` in the IAM User Guide. If the user is signed-in with MFA, then the "Condition" test fails and the final "deny" statement has no effect and other policies or statements for the user determine the user's permissions. This statement ensures that when the user is not signed-in with MFA, they can perform only the listed actions and only if another statement or policy allows access to those actions.

The ...IfExists version of the Bool operator ensures that if the `aws:MultiFactorAuthPresent` key is missing, the condition returns true. This means that a user accessing an API with long-term credentials, such as an access key, is denied access to the non-IAM API operations.

4. When you are finished, choose Review policy.
5. On the Review page, enter Force_MFA for the policy name. For the policy description, enter This policy allows users to manage their own passwords and MFA devices but nothing else unless they authenticate with MFA. Review the policy Summary to see the permissions granted by your policy, and then choose Create policy to save your work.

The new policy appears in the list of managed policies and is ready to attach.

**To attach the policy to a user (console)**

1. In the navigation pane, choose Users.
2. Choose the name (not the check box) of the user you want to edit.
3. On the Permissions tab, choose Add permissions.
4. Choose Attach existing policies directly.
5. In the search box, enter Force, and then select the check box next to Force_MFA in the list. Then choose Next: Review.
6. Review your changes and choose Add permissions.
Enable MFA for your IAM user

For increased security, we recommend that all IAM users configure multi-factor authentication (MFA) to help protect your Global Accelerator resources. MFA adds extra security because it requires users to provide unique authentication from an AWS-supported MFA device in addition to their regular sign-in credentials. The most secure AWS MFA device is the U2F security key. If your company already has U2F devices, then we recommend that you enable those devices for AWS. Otherwise, you must purchase a device for each of your users and wait for the hardware to arrive. For more information, see Enabling a U2F Security Key in the IAM User Guide.

If you don’t already have a U2F device, you can get started quickly and at a low cost by enabling a virtual MFA device. This requires that you install a software app on an existing phone or other mobile device. The device generates a six-digit numeric code based upon a time-synchronized one-time password algorithm. When the user signs in to AWS, they are prompted to enter a code from the device. Each virtual MFA device assigned to a user must be unique. A user cannot enter a code from another user's virtual MFA device to authenticate. For a list of a few supported apps that you can use as virtual MFA devices, see Multi-Factor Authentication.

Note
You must have physical access to the mobile device that will host the user's virtual MFA device in order to configure MFA for an IAM user.

To enable a virtual MFA device for an IAM user (console)

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, choose Users.
3. In the User Name list, choose the name of the intended MFA user.
5. In the Manage MFA Device wizard, choose Virtual MFA device, and then choose Continue.

IAM generates and displays configuration information for the virtual MFA device, including a QR code graphic. The graphic is a representation of the "secret configuration key" that is available for manual entry on devices that do not support QR codes.

6. Open your virtual MFA app.

For a list of apps that you can use for hosting virtual MFA devices, see Multi-Factor Authentication. If the virtual MFA app supports multiple accounts (multiple virtual MFA devices), choose the option to create a new account (a new virtual MFA device).

7. Determine whether the MFA app supports QR codes, and then do one of the following:
   - From the wizard, choose Show QR code, and then use the app to scan the QR code. For example, you might choose the camera icon or choose an option similar to Scan code, and then use the device’s camera to scan the code.
   - In the Manage MFA Device wizard, choose Show secret key, and then enter the secret key into your MFA app.

When you are finished, the virtual MFA device starts generating one-time passwords.

8. In the Manage MFA Device wizard, in the MFA code 1 box, enter the one-time password that currently appears in the virtual MFA device. Wait up to 30 seconds for the device to generate a new one-time password. Then enter the second one-time password into the MFA code 2 box. Choose Assign MFA.

Important
Submit your request immediately after generating the codes. If you generate the codes and then wait too long to submit the request, the MFA device successfully associates with
the user but the MFA device is out of sync. This happens because time-based one-time passwords (TOTP) expire after a short period of time. If this happens, you can resync the device. For more information, see Resynchronizing Virtual and Hardware MFA Devices in the IAM User Guide.

The virtual MFA device is now ready for use with AWS.

Secure VPC connections in AWS Global Accelerator

When you add 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 for internet traffic, 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.

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:

AWS Global Accelerator flow logs

Server flow logs provide detailed records about traffic that flows through an accelerator to an endpoint. Server flow logs are useful for many applications. For example, flow log information can be useful in security and access audits. For more information, see Flow logs in AWS Global Accelerator (p. 64).

Amazon CloudWatch metrics and alarms

Using CloudWatch, you can monitor, in real time, your AWS resources and the applications that you run on AWS. CloudWatch collects and tracks metrics, which are variables that you measure over time. You can 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 (p. 70).
Compliance validation for AWS Global Accelerator

Third-party auditors assess the security and compliance of AWS Global Accelerator as part of multiple AWS compliance programs. These include SOC, PCI, HIPAA, GDPR, ISO, and ENS High.

For a list of AWS services, including Global Accelerator, in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. 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 Global Accelerator 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 and Compliance Quick Start Guides** – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- **Architecting for HIPAA Security and Compliance Whitepaper** – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- **AWS Compliance Resources** – This collection of workbooks and guides might apply to your industry and location.
- **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 that helps you check your compliance with security industry standards and best practices.

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:

- A network zone services the static IP addresses for your accelerator from a unique IP subnet. Similar to an AWS Availability Zone, a network zone is an isolated unit with its own set of physical infrastructure. When you configure an accelerator, Global Accelerator allocates two IPv4 addresses for it. If one IP address from a network zone becomes unavailable due to IP address blocking by certain client
networks, or due to 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 the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access Global Accelerator through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). 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. Note that you must be in US East (N. Virginia) when you request quota increases for Global Accelerator.

Topics
- General quotas (p. 108)
- Quotas for endpoints per endpoint group (p. 108)
- Related quotas (p. 109)

General quotas

The following are overall quotas for Global Accelerator.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerators per AWS account</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>You can request a quota increase.</td>
</tr>
<tr>
<td>Listeners per accelerator</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>You can request a quota increase.</td>
</tr>
<tr>
<td>Port ranges per listener</td>
<td>10</td>
</tr>
<tr>
<td>Port overrides per endpoint group</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>You can request a quota increase.</td>
</tr>
</tbody>
</table>

Quotas for endpoints per endpoint group

The following are Global Accelerator quotas that apply to the number of endpoints in endpoint groups.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Description</th>
<th>Quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint groups with more than one endpoint type</td>
<td>Number of endpoints in an endpoint group containing more than one endpoint type.</td>
<td>10</td>
</tr>
<tr>
<td>Endpoint groups with just Application Load Balancers</td>
<td>Number of Application Load Balancers in an endpoint group containing only Application Load Balancer endpoints.</td>
<td>10</td>
</tr>
<tr>
<td>Entity</td>
<td>Description</td>
<td>Quota</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Endpoint groups with just Network Load Balancers</td>
<td>Number of Network Load Balancers in an endpoint group containing only Network Load Balancer endpoints.</td>
<td>10</td>
</tr>
<tr>
<td>Endpoint groups with just Amazon EC2 instances</td>
<td>Number of EC2 instances in an endpoint group containing only EC2 instance endpoints.</td>
<td>10</td>
</tr>
<tr>
<td>You can  request a quota increase.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Endpoint groups with just Elastic IP addresses</td>
<td>Number of Elastic IP addresses in an endpoint group containing only Elastic IP address endpoints.</td>
<td>10</td>
</tr>
<tr>
<td>You can  request a quota increase.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Endpoint groups with just Amazon Virtual Private Cloud subnets</td>
<td>Number of Amazon VPC subnets in an endpoint group containing only subnet endpoints.</td>
<td>10</td>
</tr>
<tr>
<td>You can  request a quota increase.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 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
- Additional AWS Global Accelerator documentation (p. 110)
- Getting support (p. 110)
- Tips from the Amazon Web Services Blog (p. 110)

Additional AWS Global Accelerator documentation

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.
- 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.
- AWS 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 Amazon Web Services Blog

The AWS Blog has a number of posts to help you use AWS services. For example, see the following blog posts about Global Accelerator:

- AWS Global Accelerator for Availability and Performance
- Traffic management with AWS Global Accelerator
• Analyzing and visualizing AWS Global Accelerator flow logs using Amazon Athena and Amazon 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: December 9, 2020

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added custom routing accelerators</td>
<td>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 <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/work-with-custom-routing-accelerators.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/work-with-custom-routing-accelerators.html</a>.</td>
<td>December 9, 2020</td>
</tr>
<tr>
<td>Added port overrides support</td>
<td>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 <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/about-endpoint-groups-port-override.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/about-endpoint-groups-port-override.html</a>.</td>
<td>October 21, 2020</td>
</tr>
<tr>
<td>Added two new Regions</td>
<td>Global Accelerator now supports Africa (Cape Town) and Europe (Milan). For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/preserve-client-ip-address.regions.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/preserve-client-ip-address.regions.html</a>.</td>
<td>May 20, 2020</td>
</tr>
<tr>
<td>Tagging and BYOIP</td>
<td>This release adds support for adding tags to accelerators and bringing your own IP address to AWS Global Accelerator (BYOIP). For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/tagging-">https://docs.aws.amazon.com/global-accelerator/latest/dg/tagging-</a>.</td>
<td>February 27, 2020</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>Updated Security chapter</td>
<td>Added content for compliance, resilience, and infrastructure security. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/security.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/security.html</a>.</td>
<td>December 20, 2019</td>
</tr>
<tr>
<td>Support for EC2 instances and default DNS name</td>
<td>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 <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/introduction-how-it-works-client-ip.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/introduction-how-it-works-client-ip.html</a> and <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/about-accelerators.html#about-accelerators.dns-addressing">https://docs.aws.amazon.com/global-accelerator/latest/dg/about-accelerators.html#about-accelerators.dns-addressing</a>.</td>
<td>October 29, 2019</td>
</tr>
<tr>
<td>Client IP address preservation for Application Load Balancers</td>
<td>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 <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/introduction-how-it-works-client-ip.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/introduction-how-it-works-client-ip.html</a>.</td>
<td>August 28, 2019</td>
</tr>
<tr>
<td>Release of AWS Global Accelerator service</td>
<td>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.</td>
<td>November 26, 2018</td>
</tr>
</tbody>
</table>
AWS glossary

For the latest AWS terminology, see the AWS glossary in the AWS General Reference.