AWS Global Accelerator

Developer Guide

AWS Global Accelerator: Developer Guide

Copyright © 2019 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What Is AWS Global Accelerator?

AWS Global Accelerator is a network layer service that you use to create *accelerators* that direct traffic to optimal endpoints over the AWS global network. This improves the availability and performance of your internet applications that are used by a global audience.

Global Accelerator provides you with static IP addresses that you associate with your accelerator. These IP addresses are anycast from the AWS edge network. They distribute incoming application traffic across multiple endpoints in one or more AWS Regions, which increases the availability of your applications. Endpoints can be Network Load Balancers, Application Load Balancers, EC2 instances, and Elastic IP addresses.

> **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 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 to limit the users who have permissions to delete an accelerator. For more information, see Authentication and Access Control for AWS Global Accelerator (p. 31).

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. The service reacts instantly to changes in health or configuration to ensure that internet traffic from clients is always directed to healthy endpoints.

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

Global Accelerator includes components that work together to help you improve the availability and performance of your applications:

Static IP addresses

AWS Global Accelerator provides you with a set of two static IP addresses that are anycast from the AWS edge network. The IP addresses serve as single fixed entry points for your clients. If you already have Elastic Load Balancing load balancers, EC2 instances, or Elastic IP address resources set up for your applications, you can easily add those to Global Accelerator. This allows Global Accelerator to use static IP addresses to access the resources.

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 to limit the users who have permissions to delete an accelerator. For more information, see Authentication and Access Control for AWS Global Accelerator (p. 31).

Accelerator

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

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. 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, 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.

Listener

A listener processes inbound connections from clients to Global Accelerator, based on the protocol and port (or port range) that you configure. 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. 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. 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 new releases across different AWS Regions, for example.

Endpoint

Endpoints 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 is routed to endpoints based on 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.

How AWS Global Accelerator Works

AWS Global Accelerator provides you with a set of static IP addresses that are anycast from the AWS edge network. The IP addresses serve as single fixed entry points for your clients. When you set up your accelerator with Global Accelerator, you associate your static IP addresses to regional endpoints—such as Network Load Balancers, Application Load Balancers, EC2 instances, and Elastic IP addresses—in one or more AWS Regions. The static IP addresses accept incoming traffic onto the AWS global network

from the edge location that is closest to your users. (Note that the idle timeout is 90 seconds for TCP connections and 30 seconds for UDP connections.)

From the edge location, traffic for your application 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. Traffic travels over the well-monitored, congestion-free, redundant AWS global network to the endpoint. By maximizing the time that traffic is on the AWS network, Global Accelerator ensures that traffic is always routed over the optimum network path.

Global Accelerator 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.

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. 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.

Global Accelerator supports both TCP and UDP protocols.

> **Note**
> 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.

Topics

Static IP Addresses in AWS Global Accelerator

You use the static IP addresses that Global Accelerator assigns to your accelerator to route internet traffic to the AWS global network close to where your users are, regardless of their location. You associate the addresses with Elastic Load Balancing resources, EC2 instances, or Elastic IP addresses that run in a single AWS Region or multiple 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. 14).

Global Accelerator provides the static IP addresses for you. 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 To see the steps for creating an accelerator, see Creating, Editing, or Deleting an Accelerator (p. 12).

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:

- 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 an 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. 18).

How Weights Work

For each endpoint, 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. 23).

Traffic dials and weights affect how the 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.

Health Checks for AWS Global Accelerator

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 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. 19).

When you add an endpoint, 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, it routes requests to all endpoints.

IP Address Ranges of Global Accelerator Edge Servers

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:

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

Global Accelerator entries that include `"region": "GLOBAL"` refer to the static IP addresses that are allocated to customer 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 EC2 instances, to two static IP addresses. You whitelist these addresses 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.

Origin cloaking

When you set up your Application Load Balancers as internet-facing to serve your end users, you also increase your exposure to attacks from the internet. Global Accelerator allows you to add an internal Application Load Balancer or an EC2 instance as an endpoint. Then, by using Global Accelerator as the single internet-facing access point, you help protect your applications running on AWS from distributed denial of service (DDoS) attacks and control how your end users reach your applications. Global Accelerator creates a peering connection between your accelerator and a VPC that you created with Amazon Virtual Private Cloud (Amazon VPC). The traffic between Global Accelerator and your VPC uses private IP addresses.

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.

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

1. **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.
2. **Configure regional endpoint groups for your accelerator:** You can select one or more regional endpoint groups to add to your listener by specifying the Regions that you want to distribute traffic to. The listener routes requests to the endpoints that you've added to an endpoint group. 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, 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.
3. **Add endpoints to endpoint groups:** 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 detailed steps about how to create an accelerator using the AWS Global Accelerator console, see Getting Started with AWS Global Accelerator (p. 7). To work with API operations, see Common Actions That You Can Use with AWS Global Accelerator (p. 11) and the AWS Global Accelerator API Reference.

Pricing for AWS Global Accelerator

With AWS Global Accelerator, you pay only for what you use. For more information, see AWS Global Accelerator Pricing.

Getting Started with AWS Global Accelerator

This tutorial provides 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 accelerator. At each step in this tutorial, there's a link to the corresponding API operation for completing the task programmatically. For more information about working with AWS Global Accelerator API operations, see the AWS Global Accelerator API Reference.

Tasks

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. 66).
- 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 Application Load Balancer, 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*.

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

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global Accelerator:.
2. Choose **Create accelerator**.
3. Provide a name for your accelerator.
4. 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 for the ports that you entered.
3. Optionally, choose to enable client affinity. Client affinity for a listener means that Global Accelerator ensures that connections from a specific source (client) IP address are always routed to the same endpoint. To enable this behavior, in the dropdown list, choose **Source IP**.

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

   For more information, see Client Affinity (p. 16).
4. Optionally, choose **Add listener** to add an additional listener.
5. When you're finished adding listeners, choose **Next**.

Step 3: Add Endpoint Groups

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

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. 19).

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 from the dropdown list.
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. 23).
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. 26).
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: 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 to limit the users who have permissions to delete an accelerator. For more information, see Authentication and Access Control for AWS Global Accelerator (p. 31).
- If you terminate an EC2 instance before you remove it from an endpoint group in Global Accelerator, and then you create another instance with the same private IP address, and health checks pass, Global Accelerator will route traffic to the new endpoint. If you don't want this to happen, remove the EC2 instance from the endpoint group before you terminate the instance.

To delete an accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global 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

The following table lists common AWS Global Accelerator actions that you can use with Global Accelerator resources. The table also provides links to relevant documentation.

Action	Using the Global Accelerator Console	Using the Global Accelerator API
Create an accelerator	See Getting Started with AWS Global Accelerator (p. 7)	See `CreateAccelerator`
Create a listener	See Adding, Editing, or Removing a Listener (p. 15)	See `CreateListener`
Create an endpoint group	See Adding, Editing, or Removing an Endpoint Group (p. 17)	See `CreateEndpointGroup`
List your accelerators	See Viewing Your Accelerators (p. 13)	See `ListAccelerator`
Get all information about an accelerator	See Viewing Your Accelerators (p. 13)	See `DescribeAccelerator`
Update an accelerator	See Creating, Editing, or Deleting an Accelerator (p. 12)	See `UpdateAccelerator`
Delete an accelerator	See Creating, Editing, or Deleting an Accelerator (p. 12)	See `DeleteAccelerator`

Accelerators in AWS Global Accelerator

An *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 and port (or port range) that you configure.

> **Important**
> 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 to limit the users who have permissions to delete an accelerator. For more information, see Authentication and Access Control for AWS Global Accelerator (p. 31).

This section explains how to create, edit, or delete an 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, Editing, or Deleting an Accelerator

This section explains how to work with 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 an accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global Accelerator:.
2. Choose **Create accelerator**.
3. Provide a name for your accelerator.
4. Choose **Next** to add listeners, endpoint groups, and endpoints.

To edit an accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global 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 you can delete it.
4. Choose **Save**.

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

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

To disable an accelerator

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global 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

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global 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

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global Accelerator:.
2. To see details about an accelerator, in the list, choose an accelerator, and then choose **View**.

Support for DNS Addressing in Global Accelerator

When you create an 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 such as Network Load Balancers, Application Load Balancers, EC2 instances, or Elastic IP addresses. 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.

Route Custom Domain Traffic to Your Accelerator

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

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

To set up Route 53 with an alias record for an accelerator, follow the guidance included in the following topic: Alias Target in the Amazon Route 53 Developer Guide. Scroll down to see the information for Global Accelerator.

Listeners 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. You define a listener when you create your 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 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

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global Accelerator:.
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. 16).
7. Choose **Add listener**.

To edit a listener

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global Accelerator:.
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. 16).

6. Choose **Save**.

To remove a listener

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global Accelerator:.
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, 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 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.

Endpoint Groups in AWS Global Accelerator

An endpoint group routes requests to one or more registered endpoints in AWS Global Accelerator. When you add a listener, 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 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 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 an 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 on the AWS Global Accelerator console. If you want to use API operations with AWS Global Accelerator, see the AWS Global Accelerator API Reference.

Topics

Adding, Editing, or Removing an 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 endpoint groups on the AWS Global Accelerator console. If you want to use API operations with Global Accelerator, see the AWS Global Accelerator API Reference.

To add an endpoint group

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global Accelerator:.
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 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. 19).
8. Optionally, choose **Add endpoint group** to add additional endpoint groups for this listener or other listeners.
9. Choose **Add endpoint group**.

To edit an endpoint group

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global Accelerator:.
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 an endpoint group

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global 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**.

Adjusting Traffic Flow With Traffic Dials

For each 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.

Health Check Options

AWS Global Accelerator regularly sends requests to 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 Application Load Balancer, 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.
- 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.

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 in AWS Global Accelerator

Endpoints in AWS Global Accelerator can be Network Load Balancers, Application Load Balancers, EC2 instances, or Elastic IP addresses. 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.

Global Accelerator continually monitors the health of all endpoints that are included in an 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 endpoints:

Application Load Balancer endpoints

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

EC2 instance endpoints

- An EC2 instance endpoint can't be one of the following types: C1, CC1, CC2, CG1, CG2, CR1, CS1, G1, G2, HI1, HS1, M1, M2, M3, or T1.
- EC2 instances are supported as endpoints in only some AWS Regions. For a list of supported Regions, see Supported Regions for Client IP Address Preservation (p. 29).
- 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.

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 EC2 instance endpoints. For more information, see Preserve Client IP Addresses in AWS Global Accelerator (p. 26).

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. 28).

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. 66).

Topics

- Endpoint Weights (p. 23)
- Transitioning Endpoints to Use Client IP Address Preservation (p. 23)

Adding, Editing, or Removing an Endpoint

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

You can add or remove endpoints from endpoint groups based on usage. For example, if demand on your application increases, you can add more endpoints to one or more endpoint groups to handle the increased traffic. Global Accelerator starts routing requests to an endpoint as soon as you add it and the endpoint passes the initial health checks. You can manage traffic to endpoints by adjusting the weights on an endpoint, to send proportionally more or less traffic to the endpoint.

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 an endpoint

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global Accelerator:.
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 an endpoint from the dropdown 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. 23).
8. Optionally, enable client IP address preservation for an internet-facing Application Load Balancer endpoint. Under **Preserve client IP address**, select **Preserve address**.

   Note
   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. 26).
9. Choose **Add endpoint**.

To edit an endpoint

You can edit an endpoint configuration to change the weight. For more information, see Endpoint Weights (p. 23).

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global Accelerator:.
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

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global Accelerator:.
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. 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.

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

$$\frac{\text{Weight for a specified record}}{\text{Sum of the weights for all records}}$$

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.

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. 26).

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 whitelisted Global Accelerator client IP addresses are updated to whitelist the user client IP address instead.

Client IP address preservation is available only in specific AWS Regions. For more information, see Supported Regions for Client IP Address Preservation (p. 29).

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

1. Open the Global Accelerator console at https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Global Accelerator:.
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. 23).
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.

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 EC2 instances.

- When you use an internet-facing Application Load Balancer as an endpoint with Global Accelerator, you can choose to preserve the source IP address of the original client for packets that arrive at the load balancer by enabling client IP address preservation.
- When you use an internal Application Load Balancer or an EC2 instance with Global Accelerator, the endpoint always has client IP address preservation enabled.

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 that client IP address preservation is supported only in specific AWS Regions. For more information, see Supported Regions for Client IP Address Preservation (p. 29).

Topics

How To Enable Client IP Address Preservation

When you create a new accelerator with internet-facing Application Load Balancer endpoints, you can choose to enable or disable client IP address preservation for each Application Load Balancer endpoint. When you create a new accelerator with internal Application Load Balancer endpoints or EC2 instance endpoints, the endpoints always have client IP address preservation.

> **Note**
> When you use 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. Internal Application Load Balancers and EC2 instances always have client IP address preservation enabled. 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. 23).

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. 27). 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 IP Address Ranges of Global Accelerator Edge Servers (p. 5) and Flow Logs in AWS Global Accelerator (p. 54).

How the Client IP Address is Preserved in AWS Global Accelerator

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

- For an EC2 instance endpoint, the client's IP address is preserved in the TCP packet header.
- 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 TPC 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 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, AWS 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 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. 66).

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 subnetA and subnetB, and then you add the load balancer as an accelerator endpoint, Global Accelerator creates two elastic network interfaces, one in each subnet.
- **Example 2:** If you add, for example, an ALB1 that has elastic network interfaces in subnetA and subnetB to Accelerator1, and then add an ALB2 with elastic network interfaces in subnetA and

subnetB to Accelerator2, Global Accelerator creates only two elastic network interfaces: one in subnetA and one in subnetB.

- **Example 3:** If you add an ALB1 that has elastic network interfaces in subnetA and subnetB to Accelerator1, and then add an ALB2 with elastic network adaptors in subnetA and subnetC to Accelerator2, Global Accelerator creates three elastic network interfaces: one in subnetA, one in subnetB, and one in subnetC. The elastic network interface in subnetA delivers traffic on for both Accelerator1 and Accelerator2.

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

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

Evaluate the number of subnets that are used by endpoints in your accelerators to determine the number of elastic network interfaces that Global Accelerator will create. Before you create an accelerator, make sure that you have enough IP address space capacity for the required elastic network interfaces, 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.
- 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 Regions for Client IP Address Preservation

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

Region Name	Region
US East (N. Virginia)	`us-east-1`
US East (Ohio)	`us-east-2`

Region Name	Region
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Asia Pacific (Seoul)	ap-northeast-2
EU (Frankfurt)	eu-central-1
EU (Ireland)	eu-west-1
EU (London)	eu-west-2
EU (Paris)	eu-west-3
EU (Stockholm)	eu-north-1
South America (São Paulo)	sa-east-1

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

Authentication and Access Control 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. 47). Optionally, you can learn more about authentication and access control by viewing What is Authentication? (p. 40), What is Access Control? (p. 41), and What are Policies? (p. 44).

Topics

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. 40).

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. 41) and What are Policies? (p. 44).

> **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.

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

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:

```
{
    "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",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/AWSServiceName": "GlobalAccelerator"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSecurityGroup",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
```

```
            "Action": "elasticloadbalancing:DescribeLoadBalancers",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": [
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:network-interface/*"
            ]
        }
    ]
}
```

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. 50).

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. 47).

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:

```
{
    "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",
```

```
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

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:

```
{
   "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 does not support 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`.

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

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. 47).

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. 40)
- What is Access Control? (p. 41)

- What are Policies? (p. 44)

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.

When you first create an accelerator in Global Accelerator and add an endpoint group, a role named AWSServiceRoleForGlobalAccelerator is automatically created to allow 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 ??? (p. 12).

> **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*.

Supported Regions for Global Accelerator Service-Linked Roles

Global Accelerator supports using service-linked roles in AWS Regions where Global Accelerator is supported.

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

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?

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 Authentication and Access Control for AWS Global Accelerator (p. 31), and then see Getting Started with IAM (p. 47).

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. 43).

AWS service access

A service role is an IAM role that a service assumes to perform actions in your account on your behalf. When you set up some AWS service environments, you must define a role for the service to assume. This service role must include all the permissions that are required for the service to access the AWS resources that it needs. Service roles vary from service to service, but many allow you to choose your permissions as long as you meet the documented requirements for that service. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. You can create, modify, and delete a service role from within IAM. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. 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 Authentication and Access Control for AWS Global Accelerator (p. 31), and then see Getting Started with IAM (p. 47).

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. 44).

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. 42)** – Control what the person or application making the request (the *principal*) is allowed to do.
- **IAM Identities (p. 42)** – Control which IAM identities (groups, users, and roles) can be accessed and how.
- **IAM Policies (p. 43)** – Control who can create, edit, and delete customer managed policies, and who can attach and detach all managed policies.
- **AWS Resources (p. 43)** – Control who has access to resources using an identity-based policy or a resource-based policy.
- **AWS Accounts (p. 43)** – 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, 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. 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 Authentication and Access Control for AWS Global Accelerator (p. 31), and then see Getting Started with IAM (p. 47).

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

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:

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

The following example grants programmatic access to the `ListAccelerators` operation:

```
{
    "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 Authentication and Access Control for AWS Global Accelerator (p. 31).

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. Use your AWS account email address and password to sign in as the *AWS account root user* to the IAM console at https://console.aws.amazon.com/iam/.

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. 47).
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/.
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.

```
{
    "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: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:ListAccessKeys",
            "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.
4. Choose the **Security credentials** tab. Next to **Assigned MFA device**, choose **Manage**.
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.

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

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

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_log_id_timestamp_hash.log.gz
```

Note the following about the folder and file name structure for log files:

- The timestamp uses the `YYYYMMDDTHHmmZ` format.
- If you specify slash (/) for the S3 bucket prefix, the log file bucket folder structure will include a double slash (//), like the following:

  ```
  s3-bucket_name//AWSLogs/aws_account_id
  ```

The following example shows the folder structure and file name of a log file for a flow log created by AWS account `123456789012` for an accelerator with an ID of `1234abcd-abcd-1234-abcd-1234abcdefgh`, on November 23, 2018 at 00:05 UTC:

```
my-s3-bucket/prefix1/AWSLogs/123456789012/globalaccelerator/us-
west-2/2018/11/23/123456789012_globalaccelerator_1234abcd-abcd-1234-
abcd-1234abcdefgh_20181123T0005Z_1fb1234.log.gz
```

A single flow log file contains interleaved entries with multiple 5-tuple records; that is, `client_ip`, `client_port`, `accelerator_ip`, `accelerator_port`, `protocol`. To see all the flow log files for your accelerator, look for entries aggregated by the `accelerator_id` and your `account_id`.

IAM Roles for Publishing Flow Logs to Amazon S3

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

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DeliverLogs",
            "Effect": "Allow",
```

```
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowGlobalAcceleratorService",
            "Effect": "Allow",
            "Action": [
                "globalaccelerator:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "s3Perms",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketPolicy",
                "s3:PutBucketPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon S3 Bucket Permissions for Flow Logs

By default, Amazon S3 buckets and the objects that they contain are private. Only the bucket owner can access the bucket and the objects stored in it. The bucket owner, however, can grant access to other resources and users by writing an access policy.

If the user creating the flow log owns the bucket, the service automatically attaches the following policy to the bucket to give the flow log permission to publish logs to it:

```
{
    "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/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:

```
{
    "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. 55).
3. Run the following AWS CLI command, with the Amazon S3 bucket name and prefix that you want to use for your log files:

```
aws globalaccelerator update-accelerator-attributes
       --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh
       --region us-west-2
       --flow-logs-enabled
       --flow-logs-s3-bucket s3-bucket-name
       --flow-logs-s3-prefix s3-bucket-prefix
```

Processing Flow Log Records in Amazon S3

The log files are compressed. If you open the log files using the Amazon S3 console, they are decompressed and the flow log records are displayed. If you download the files, you must decompress them to view the flow log records.

Timing of Log File Delivery

AWS Global Accelerator delivers log files for your configured accelerator up to several times an hour. In general, a log file contains information about the requests that your accelerator received during a given time period. Global Accelerator usually delivers the log file for that time period to your Amazon S3 bucket within an hour of the events that appear in the log. Some or all log file entries for a time period can sometimes be delayed by up to 24 hours. When log entries are delayed, Global Accelerator saves them in a log file for which the file name includes the date and time of the period in which the requests occurred, not the date and time when the file was delivered.

When creating a log file, Global Accelerator consolidates information for your accelerator from all the edge locations that received requests during the time period that the log file covers.

Global Accelerator begins to reliably deliver log files about four hours after you enable logging. You might get a few log files before that time.

> **Note**
> If no users connect to your accelerator during the time period, you don't receive any log files for that period.

Flow Log Record Syntax

A flow log record is a space-separated string that has the following format:

```
<version> <aws_account_id> <accelerator_id> <client_ip> <client_port>
<accelerator_ip> <accelerator_port> <endpoint_ip> <endpoint_port> <protocol>
<ip_address_type> <packets> <bytes> <start_time> <end_time> <action> <log-
status> <globalaccelerator_source_ip> <globalaccelerator_source_port>
<endpoint_region> <globalaccelerator_region> <direction> <vpc_id>
```

The Version 1.0 format does not include the VPC identifier, `vpc_id`. The Version 2.0 format, which includes `vpc_id`, is generated when Global Accelerator sends traffic to an endpoint with client IP address preservation.

The following table describes the fields of a flow log record.

Field	Description
`version`	The flow logs version.
`aws_account_id`	The AWS account ID for the flow log.
`accelerator_id`	The ID of the accelerator for which the traffic is recorded.
`client_ip`	The source IPv4 address.
`client_port`	The source port.
`accelerator_ip`	The accelerator's IP address.
`accelerator_port`	The accelerator's port.
`endpoint_ip`	The destination IP address of the traffic.
`endpoint_port`	The destination port of the traffic.
`protocol`	The IANA protocol number of the traffic. For more information, see Assigned Internet Protocol Numbers.
`ip_address_type`	IPv4.
`packets`	The number of packets transferred during the capture window.
`bytes`	The number of bytes transferred during the capture window.
`start_time`	The time, in Unix seconds, of the start of the capture window.
`end_time`	The time, in Unix seconds, of the end of the capture window.
`action`	The action associated with the traffic: • `ACCEPT`: The recorded traffic was permitted by the security groups or network ACLs.
`log-status`	The logging status of the flow log: • `OK`: Data is logging normally to the chosen destinations. • `NODATA`: There was no network traffic to or from the network interface during the capture window. • `SKIPDATA`: Some flow log records were skipped during the capture window. This can be because of an internal capacity constraint, or an internal error.
`globalaccelerator_interface_ip`	The IP address used by the Global Accelerator network interface.
`globalaccelerator_interface_port`	The port used by the Global Accelerator network interface.
`endpoint_region`	The AWS Region where the endpoint is located.
`globalaccelerator_region`	The edge location (point of presence) that served the request. Each edge location has a three-letter code and an arbitrarily assigned number, for example, DFW3. The three-letter code typically corresponds with the International Air Transport Association airport code for an airport near the edge location. (These abbreviations might change in the future.)

Field	Description
`direction`	The direction of the traffic. Denotes traffic coming into the Global Accelerator network (`INGRESS`) or returning to the client (`EGRESS`).
`vpc_id`	The VPC identifier. Included with Version 2.0 flow logs when Global Accelerator sends traffic to an endpoint with client IP address preservation.

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

Logging AWS Global Accelerator API Calls with AWS CloudTrail

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**. Using the information collected by CloudTrail, you can determine the request that was made to Global Accelerator, the IP address from which the request was made, who made the request, when it was made, and additional details.

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

Global Accelerator Information in CloudTrail

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

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

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All Global Accelerator actions are logged by CloudTrail and are documented in the AWS Global Accelerator API Reference. For example, calls to the `CreateAccelerator`, `ListAccelerators` and `UpdateAccelerator` operations generate entries in the CloudTrail log files.

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

- Whether the request was made with root or 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`.


```
v{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:03:14Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "ListAccelerators",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": null,
      "responseElements": null,
      "requestID": "083cae81-28ab-4a66-862f-096e1example",
      "eventID": "fe8b1c13-8757-4c73-b842-fe2a3example",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
```

```
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:04:49Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "CreateListener",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": {
        "acceleratorArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-a114-5d7fexample",
        "portRanges": [
          {
            "fromPort": 80,
            "toPort": 80
          }
        ],
        "protocol": "TCP"
      },
      "responseElements": {
        "listener": {
          "listenerArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-a114-5d7fexample/
listener/abcde1234",
          "portRanges": [
            {
              "fromPort": 80,
              "toPort": 80
            }
          ],
          "protocol": "TCP",
          "clientAffinity": "NONE"
        }
      },
      "requestID": "6090509a-5a97-4be6-8e6a-7d73example",
      "eventID": "9cab44ef-0777-41e6-838f-f249example",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:03:52Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "CreateAccelerator",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": {
        "name": "cloudTrailTest"
      },
      "responseElements": {
        "accelerator": {
          "acceleratorArn":
"arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-a114-5d7fexample",
          "name": "cloudTrailTest",
          "ipAddressType": "IPV4",
          "enabled": true,
          "ipSets": [
            {
              "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": "11f9a762-8c00-4fcc-80f9-848a29example",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
```

```
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:05:27Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "UpdateListener",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": {
        "listenerArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-a114-5d7fexample/
listener/abcde1234",
        "portRanges": [
          {
            "fromPort": 80,
            "toPort": 80
          },
          {
            "fromPort": 81,
            "toPort": 81
          }
        ]
      },
      "responseElements": {
        "listener": {
          "listenerArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-a114-5d7fexample/
listener/abcde1234",
          "portRanges": [
            {
              "fromPort": 80,
              "toPort": 80
            },
            {
              "fromPort": 81,
              "toPort": 81
            }
          ],
          "protocol": "TCP",
          "clientAffinity": "NONE"
        }
      },
      "requestID": "008ef93c-b3a3-44b4-afb3-768example",
      "eventID": "85958f0d-63ff-4a2c-99e3-6ffbexample",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
```

```
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:06:05Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "DescribeListener",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": {
        "listenerArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-a114-5d7fexample/
listener/abcde1234"
      },
      "responseElements": null,
      "requestID": "9980e368-82fa-40da-95a3-4b0example",
      "eventID": "885a02e9-2a60-4626-b1ba-57285example",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:05:47Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "ListListeners",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": {
        "acceleratorArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-a114-5d7fexample"
      },
      "responseElements": null,
      "requestID": "08e4b0f7-689b-4c84-af2d-47619example",
      "eventID": "f4fb8e41-ed21-404d-af9d-037c4example",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
```

```
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        }
      },
      "eventTime": "2018-11-17T21:06:24Z",
      "eventSource": "globalaccelerator.amazonaws.com",
      "eventName": "DeleteListener",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
      "requestParameters": {
        "listenerArn":
 "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-a114-5d7fexample/
listener/abcde1234"
      },
      "responseElements": null,
      "requestID": "04d37bf9-3e50-41d9-9932-6112example",
      "eventID": "afedb874-2e21-4ada-b1b0-2ddb2example",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    }
  ]
}
```


Secure VPC Connections in AWS Global Accelerator

When you add an internal Application Load Balancer or an 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.

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.

Limits for AWS Global Accelerator

Your AWS account has the following limits related to AWS Global Accelerator.

AWS Global Accelerator Limits

- Number of accelerators for each AWS account – 20
- Number of listeners for each accelerator – 10
- Number of port ranges for each listener – 10
- Number of endpoints for each endpoint group – 10

In addition, there are limits for Network Load Balancers, Application Load Balancers, EC2 instances, or Elastic IP addresses that are used as endpoints for an accelerator. For more information, see the following:

- Elastic IP Address Limit in the *Amazon EC2 User Guide*.
- Amazon EC2 Service Limits in the *Amazon EC2 User Guide*.
- Limits for Your Network Load Balancers in the *User Guide for Network Load Balancers*.
- Limits for Your Application Load Balancers in the *User Guide for Application Load Balancers*.

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

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

Document History

The following table describes the documentation for this release of AWS Global Accelerator.

- **API version: latest**
- **Latest documentation update:** October 29, 2019

Change	Description	Date
Support for EC2 instances and default DNS name	AWS Global Accelerator now supports adding EC2 instances in supported AWS Regions. In addition, Global Accelerator creates a default DNS name that is mapped to the static IP addresses for your accelerator. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/introduction-how-it-works-client-ip.html and https://docs.aws.amazon.com/global-accelerator/latest/dg/about-accelerators.html#about-accelerators.dns-addressing.	October 29, 2019
Client IP address preservation for Application Load Balancers	You can now choose to have AWS Global Accelerator preserve the client IP address for Application Load Balancers in supported AWS Regions. For more information, see https://docs.aws.amazon.com/global-accelerator/latest/dg/introduction-how-it-works-client-ip.html.	August 28, 2019
Release of AWS Global Accelerator service	The AWS Global Accelerator Developer Guide provides information about setting up and using accelerators—network layer traffic managers—that improve availability and performance for your internet applications that have a global audience.	November 26, 2018

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.