# Table of Contents

What is AWS Global Accelerator? ................................................................. 1  
Components ................................................................................................. 2  
AWS Regions ................................................................................................. 3  
How it works .................................................................................................. 5  
  Idle timeout ............................................................................................... 6  
  Static IP addresses .................................................................................... 6  
  Traffic dials and endpoint weights ............................................................... 7  
  Health checks ............................................................................................ 8  
Types of accelerators .................................................................................... 8  
Location and IP address ranges of Edge servers ........................................ 9  
Use cases ...................................................................................................... 9  
Speed Comparison Tool ............................................................................... 10  
How to get started ....................................................................................... 11  
Tagging ......................................................................................................... 11  
  Tagging support in Global Accelerator ..................................................... 12  
  Adding, editing, and deleting tags in Global Accelerator ......................... 12  
Pricing .......................................................................................................... 13  
Getting started ............................................................................................ 14  
Create a standard accelerator .................................................................... 14  
  Before you begin ..................................................................................... 14  
  Step 1: Create a standard accelerator ................................................... 15  
  Step 2: Add listeners ............................................................................... 15  
  Step 3: Add endpoint groups ................................................................. 16  
  Step 4: Add endpoints ............................................................................ 16  
  Step 5: Test your accelerator ................................................................. 16  
  Step 6 (optional): Delete your accelerator ............................................. 17  
Create a custom routing accelerator ............................................................ 17  
  Before you begin ..................................................................................... 18  
  Step 1: Create a custom routing accelerator .......................................... 18  
  Step 2: Add listeners ............................................................................... 18  
  Step 3: Add endpoint groups ................................................................. 19  
  Step 4: Add VPC subnet endpoints ....................................................... 19  
  Step 5 (optional): Delete your accelerator ............................................. 20  
Actions ........................................................................................................ 21  
Work with standard accelerators .................................................................. 23  
  Standard accelerators ............................................................................ 23  
    Global static IP addresses for your accelerator .................................. 23  
    Creating or updating a standard accelerator ........................................ 24  
    Deleting an accelerator ....................................................................... 25  
    Viewing your accelerators .................................................................... 26  
    Add an accelerator when you create a load balancer ......................... 26  
    Using global static IP addresses instead of regional static IP addresses .. 27  
Listeners for standard accelerators ............................................................. 28  
  Adding, editing, or removing a standard listener .................................... 28  
  Client affinity .......................................................................................... 29  
Endpoint groups for standard accelerators .................................................. 29  
  Adding, editing, or removing a standard endpoint group ....................... 30  
  Using traffic dials ................................................................................... 31  
  Overriding listener ports ....................................................................... 32  
  Changing health check options ............................................................. 33  
Endpoints for standard accelerators ............................................................ 34  
  Adding, editing, or removing a standard endpoint ................................... 36  
  Endpoint weights ................................................................................... 37  
  Failover for unhealthy endpoints ............................................................ 38
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avoiding TCP connection time delays</td>
<td>38</td>
</tr>
<tr>
<td>Adding endpoints with client IP address preservation</td>
<td>40</td>
</tr>
<tr>
<td><strong>Work with custom routing accelerators</strong></td>
<td>43</td>
</tr>
<tr>
<td>How custom routing accelerators work</td>
<td>44</td>
</tr>
<tr>
<td>Example of how custom routing works in Global Accelerator</td>
<td>44</td>
</tr>
<tr>
<td>Guidelines and restrictions for custom routing accelerators</td>
<td>46</td>
</tr>
<tr>
<td>Custom routing accelerators</td>
<td>48</td>
</tr>
<tr>
<td>Creating or updating a custom routing accelerator</td>
<td>49</td>
</tr>
<tr>
<td>Viewing your custom routing accelerators</td>
<td>49</td>
</tr>
<tr>
<td>Deleting a custom routing accelerator</td>
<td>49</td>
</tr>
<tr>
<td>Listeners for custom routing accelerators</td>
<td>50</td>
</tr>
<tr>
<td>Adding, editing, or removing a custom routing listener</td>
<td>50</td>
</tr>
<tr>
<td>Endpoint groups for custom routing accelerators</td>
<td>51</td>
</tr>
<tr>
<td>Adding, editing, or removing an endpoint group</td>
<td>52</td>
</tr>
<tr>
<td>VPC subnet endpoints for custom routing accelerators</td>
<td>53</td>
</tr>
<tr>
<td>Adding, editing, or removing a VPC subnet endpoint</td>
<td>53</td>
</tr>
<tr>
<td><strong>DNS addressing and custom domains</strong></td>
<td>56</td>
</tr>
<tr>
<td>Support for DNS addressing</td>
<td>56</td>
</tr>
<tr>
<td>Route custom domain traffic to your accelerator</td>
<td>56</td>
</tr>
<tr>
<td>Bring your own IP addresses</td>
<td>57</td>
</tr>
<tr>
<td>Requirements</td>
<td>58</td>
</tr>
<tr>
<td>IP address range authorization</td>
<td>58</td>
</tr>
<tr>
<td>Provision the address range for use with AWS Global Accelerator</td>
<td>60</td>
</tr>
<tr>
<td>Advertise the address range through AWS</td>
<td>61</td>
</tr>
<tr>
<td>Deprovision the address range</td>
<td>62</td>
</tr>
<tr>
<td>Create an accelerator</td>
<td>62</td>
</tr>
<tr>
<td><strong>Preserve client IP addresses</strong></td>
<td>64</td>
</tr>
<tr>
<td>How to enable client IP address preservation</td>
<td>65</td>
</tr>
<tr>
<td>Benefits of client IP address preservation</td>
<td>66</td>
</tr>
<tr>
<td>How the client IP address is preserved</td>
<td>66</td>
</tr>
<tr>
<td>Best practices for client IP address preservation</td>
<td>67</td>
</tr>
<tr>
<td><strong>Logging and monitoring</strong></td>
<td>69</td>
</tr>
<tr>
<td>CloudWatch monitoring</td>
<td>69</td>
</tr>
<tr>
<td>Global Accelerator metrics</td>
<td>70</td>
</tr>
<tr>
<td>Metric dimensions for accelerators</td>
<td>72</td>
</tr>
<tr>
<td>Statistics for Global Accelerator metrics</td>
<td>73</td>
</tr>
<tr>
<td>View CloudWatch metrics for your accelerators</td>
<td>74</td>
</tr>
<tr>
<td><strong>Flow logs</strong></td>
<td>75</td>
</tr>
<tr>
<td>Publishing to Amazon S3</td>
<td>76</td>
</tr>
<tr>
<td>Timing of log file delivery</td>
<td>79</td>
</tr>
<tr>
<td>Flow log record syntax</td>
<td>80</td>
</tr>
<tr>
<td><strong>CloudTrail logging</strong></td>
<td>81</td>
</tr>
<tr>
<td>Global Accelerator information in CloudTrail</td>
<td>82</td>
</tr>
<tr>
<td>Viewing Global Accelerator events in event history</td>
<td>82</td>
</tr>
<tr>
<td>Understanding Global Accelerator log file entries</td>
<td>82</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>89</td>
</tr>
<tr>
<td>Identity and Access Management</td>
<td>89</td>
</tr>
<tr>
<td>Audience</td>
<td>90</td>
</tr>
<tr>
<td>Authenticating with identities</td>
<td>90</td>
</tr>
<tr>
<td>Managing access using policies</td>
<td>92</td>
</tr>
<tr>
<td>How Global Accelerator works with IAM</td>
<td>94</td>
</tr>
<tr>
<td>Identity-based policy examples</td>
<td>99</td>
</tr>
<tr>
<td>Service-linked role</td>
<td>102</td>
</tr>
<tr>
<td>AWS managed policies</td>
<td>104</td>
</tr>
<tr>
<td>Tag-based policies</td>
<td>107</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>108</td>
</tr>
<tr>
<td>Secure VPC connections</td>
<td>110</td>
</tr>
</tbody>
</table>
What is AWS Global Accelerator?

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

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

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

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

**Important**
The static IP addresses remain assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you delete an accelerator, you lose the static IP addresses that are assigned to it, so you can no longer route traffic by using them. You can use IAM policies, like tag-based permissions with Global Accelerator, to limit the users who have permissions to delete an accelerator. For more information, see ABAC with Global Accelerator (p. 97).

For standard accelerators, Global Accelerator uses the AWS global network to route traffic to the optimal regional endpoint based on health, client location, and policies that you configure, which increases the availability of your applications. Endpoints for standard accelerators can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses that are located in one AWS Region or multiple Regions. The service reacts instantly to changes in health or configuration to ensure that internet traffic from clients is always directed to healthy endpoints.

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

**Topics**
- AWS Global Accelerator components (p. 2)
- AWS Region availability for AWS Global Accelerator (p. 3)
- How AWS Global Accelerator works (p. 5)
- Types of accelerators (p. 8)
- Location and IP address ranges of Global Accelerator Edge servers (p. 9)
- AWS Global Accelerator use cases (p. 9)
- AWS Global Accelerator Speed Comparison Tool (p. 10)
AWS Global Accelerator components

AWS Global Accelerator includes the following components:

**Static IP addresses**

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

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

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

**Accelerator**

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

There are two types of accelerators:

- A *standard* accelerator directs traffic to the optimal AWS endpoint based on several factors, including the user's location, the health of the endpoint, and the endpoint weights that you configure. This improves the availability and performance of your applications. Endpoints can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses.

- A *custom routing* accelerator lets you deterministically route multiple users to a specific EC2 destination behind your accelerator, as is required for some use cases. You do this by directing users to a unique IP address and port on your accelerator, which Global Accelerator has mapped to the destination. Note that custom routing accelerators do not support dual-stack for IP addresses.

For more information, see Types of accelerators (p. 8).

**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

---

AWS Global Accelerator Developer Guide
Components

- How to get started with AWS Global Accelerator (p. 11)
- Tagging in AWS Global Accelerator (p. 11)
- Pricing for AWS Global Accelerator (p. 13)
that Global Accelerator assigns to you or that you choose from your own IP address range. If you have a dual-stack accelerator, Global Accelerator also assigns you a dual-stack DNS name, similar to a1234567890abcdef.dualstack.awsglobalaccelerator.com that points to the four static IP addresses for your dual-stack accelerator.

Depending on the use case, you can use your accelerator's static IP addresses or DNS name to route traffic to your accelerator, or set up DNS records to route traffic using your own custom domain name. For more information, see Support for DNS addressing in AWS Global Accelerator (p. 56).

**Network zone**

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

**Listener**

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

**Endpoint group**

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

**Endpoint**

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

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

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

**AWS Region availability for AWS Global Accelerator**

For detailed information about Regional support and service endpoints for AWS Global Accelerator, see AWS Global Accelerator endpoints and quotas in the Amazon Web Services General Reference.
Note
AWS Global Accelerator is a global service. However, you must specify the US West (Oregon) Region (that is, specify the parameter `--region us-west-2`) in Regional Global Accelerator AWS CLI commands. That is, when you create resources, such as accelerators.

Global Accelerator is currently available in the following AWS Regions. Availability Zone (AZ) exceptions are noted.

<table>
<thead>
<tr>
<th>Region Name</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (Ohio)</td>
<td>us-east-2</td>
</tr>
<tr>
<td>US East (N. Virginia)</td>
<td>us-east-1</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>us-west-1 (except AZ usw1-az2)</td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td>us-west-2</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>af-south-1</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>ap-east-1</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>ap-south-1</td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>ap-south-2</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>ap-southeast-3</td>
</tr>
<tr>
<td>Asia Pacific (Melbourne)</td>
<td>ap-southeast-4</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>ap-northeast-3</td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>ap-southeast-1</td>
</tr>
<tr>
<td>Asia Pacific (Sydney)</td>
<td>ap-southeast-2</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>ap-northeast-1 (except AZ apne1-az3)</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>ap-northeast-2</td>
</tr>
<tr>
<td>Canada (Central)</td>
<td>ca-central-1 (except AZ cac1-az3)</td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>eu-central-1</td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>eu-west-1</td>
</tr>
<tr>
<td>Europe (London)</td>
<td>eu-west-2</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>eu-south-1</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>eu-west-3</td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>eu-south-2</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>eu-north-1</td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>eu-central-2</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>il-central-1</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>me-south-1</td>
</tr>
</tbody>
</table>
### How AWS Global Accelerator works

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

**Note**

If you bring your own IP address range to AWS (BYOIP) to use with Global Accelerator, you can instead assign static IP addresses from your own pool to use with your accelerator. For more information, see Bring your own IP addresses (BYOIP) in AWS Global Accelerator (p. 57).

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

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

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

With some endpoint types, you have the option to preserve and access the client IP address. The following types of endpoints can preserve the source IP address of the client in incoming packets: Application Load Balancers, Amazon EC2 instances, and Network Load Balancers with security groups. Be aware of the following for client IP address preservation support in Global Accelerator:

- Client IP address preservation for Network Load Balancers is supported only for Network Load Balancers with security groups that do not have TLS listeners attached and do not perform IPv4 to IPv6 NAT translation to their EC2 targets.
- Client IP address preservation is not supported for Network Load Balancers without security groups or for Elastic IP addresses.
- Global Accelerator always preserves client IP addresses for endpoints on custom routing accelerators.

Global Accelerator terminates TCP connections from clients at AWS edge locations and, almost concurrently, establishes a new TCP connection with your endpoints. This gives clients faster response times (lower latency) and increased throughput.

In standard accelerators, Global Accelerator continuously monitors the health of all endpoints, and instantly begins directing traffic for all new connections to another available endpoint when it determines that an active endpoint is unhealthy. This allows you to create a high-availability architecture for your applications on AWS. Health checks aren't used with custom routing accelerators and there is no failover, because you specify the destination to route traffic to.
When you add an accelerator, security groups and AWS WAF rules that you have already configured continue to work as they did before you added the accelerator.

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

Be aware of the following when you use Global Accelerator:

- **IP fragmentation**: IP packets that are too large to fit into a standard Ethernet frame (1500+ bytes) when transmitted across the internet or other large networks are fragmented by intermediate routers and sent individually. The TCP protocol does not require IP fragmentation because clients and endpoints automatically negotiate a smaller Maximum Segment Size (MSS). However, the UDP protocol requires IP fragmentation. When packets are fragmented, Global Accelerator forwards UDP fragments to the configured endpoint, which reassembles the original IP packet. Global Accelerator drops TCP fragments at the edge, because they are not supported by the AWS network.

- **Cross-account resources**: When you add a resource as an endpoint in Global Accelerator, the resource cannot belong to another AWS account.

### Topics

- [Idle timeout in AWS Global Accelerator](#)
- [Static IP addresses in AWS Global Accelerator](#)
- [Traffic flow management with traffic dials and endpoint weights](#)
- [Health checks for AWS Global Accelerator](#)

#### Idle timeout in AWS Global Accelerator

AWS Global Accelerator sets an idle timeout period that applies to its connections. If no data has been sent or received by the time that the idle timeout period elapses, Global Accelerator closes the connection. To ensure that the connection stays alive, the client or the endpoint must send at least 1 byte of data before the idle timeout period elapses.

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

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

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

#### Static IP addresses in AWS Global Accelerator

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

If you like, you can associate your own custom domain name with the static IP addresses for your accelerator. For more information, see Route custom domain traffic to your accelerator (p. 56).

Global Accelerator provides the static IP addresses for you from the Amazon pool of IP addresses, unless you bring your own IP address range to AWS, and then specify the static IP addresses from that pool. (For more information, see Bring your own IP addresses (BYOIP) in AWS Global Accelerator (p. 57).) To create an accelerator on the console, the first step is to prompt Global Accelerator to provision the static IP addresses by entering a name for your accelerator or choose your own static IP addresses. To see the steps for creating an accelerator, see Getting started with AWS Global Accelerator (p. 14).

The static IP addresses remain assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you delete an accelerator, you lose the static IP addresses that are assigned to it, so you can no longer route traffic by using them. You can use IAM policies like tag-based permissions with Global Accelerator to limit the users who have permissions to delete an accelerator. For more information, see ABAC with Global Accelerator (p. 97).

Traffic flow management with traffic dials and endpoint weights

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

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

How traffic dials work

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

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

For more information, see Adjusting traffic flow with traffic dials (p. 31).

How weights work

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

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

The accelerator calculates the sum of the weights for the endpoints in an endpoint group, and then directs traffic to the endpoints based on the ratio of each endpoint’s weight to the total. For an example of how weights work, see Endpoint weights (p. 37).
Traffic dials and weights affect how the standard accelerator serves traffic in different ways:

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

**Note**
For more information about how traffic dials and weights affect failover, see [Failover for unhealthy endpoints](p. 38).

### Health checks for AWS Global Accelerator

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

Global Accelerator includes default health checks that are run automatically, but you can configure the timing for the checks and other options. If you've configured custom health check settings, Global Accelerator uses those settings in specific ways, depending on your configuration. You configure those settings in Global Accelerator for Amazon EC2 instance or Elastic IP address endpoints or by configuring settings on the Elastic Load Balancing console for Network Load Balancers or Application Load Balancers. For more information, see [Changing health check options](p. 33).

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

### Types of accelerators

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

**Standard accelerator**

By using a standard accelerator, you can improve the availability and performance of your applications running on Application Load Balancers, Network Load Balancers, or Amazon EC2 instances. With a standard accelerator, Global Accelerator routes client traffic across regional endpoints based on geo-proximity and endpoint health. It also allows customers to shift client traffic across endpoints based on controls such as traffic dials and endpoint weights. This works for a wide variety of use cases, including blue/green deployment, A/B testing, and multi-Region deployment. To see more use cases, see [AWS Global Accelerator use cases](p. 9).

To learn more, see [Work with standard accelerators in AWS Global Accelerator](p. 23).

**Custom routing accelerator**

Custom routing accelerators work well for scenarios where you want to use custom application logic to direct one or more users to a specific destination and port among many, while still gaining the performance benefits of Global Accelerator. One example is VoIP applications that assign multiple callers to a specific media server to start voice, video, and messaging sessions. Another example is online real-time gaming applications where you want to assign multiple players to a single session on a game server based on factors such as geographic location, player skill, and game mode.
Note
Custom routing accelerators support only the IPv4 IP address type.

To learn more, see Work with custom routing accelerators in AWS Global Accelerator (p. 43).

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

Location and IP address ranges of Global Accelerator Edge servers

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

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

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

- To find the IP address ranges that are associated with AWS Global Accelerator Edge servers, search ip-ranges.json for the following string:
  
  "service": "GLOBALACCELERATOR"
  
- Global Accelerator entries that include "region": "GLOBAL" refer to the static IP addresses that are allocated to accelerators. If you want to filter for traffic through your accelerator that comes from points of presence (POPs) in one area, filter for entries that include a specific geographical area, such as us-* or eu-*. So, for example, if you filter for us-*, you will see only traffic coming through POPs in the United States (U.S.).

- Global Accelerator supports two ways of routing traffic: using client IP preservation or using network address translation (NAT). The way that traffic is routed determines the client IP address that AWS WAF can apply rules to. When you use IP preservation, AWS WAF rules target the client IP address—that is, the IP address of the clients who access your service. When you use NAT, AWS WAF rules are applied to the global IP addresses that Global Accelerator uses to route traffic.

AWS Global Accelerator use cases

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

Scale for increased application utilization

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

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

Disaster recovery and multi-Region resiliency

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

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

Protect your applications

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

Improve performance for VoIP or online gaming applications

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

AWS Global Accelerator Speed Comparison Tool

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

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

https://speedtest.globalaccelerator.aws

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

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

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

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

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

   For a custom routing accelerators, traffic is deterministically routed to a specific destination in a VPC subnet, based on the listener port that the traffic is received on.

5. **Add endpoints to endpoint groups**: The endpoints that you add depend on the type of accelerator.
   - For a standard accelerator, you can add one or more regional resources, such as load balancers or EC2 instances endpoints, to each endpoint group. Next, you can decide how much traffic you want to route to each endpoint by setting endpoint weights.
   - For a custom routing accelerator, you add one or more virtual private cloud (VPC) subnets with up to thousands of Amazon EC2 instance destinations.

For detailed steps about how to create a standard accelerator or a custom routing accelerator using the AWS Global Accelerator console, see Getting started with AWS Global Accelerator (p. 14). To work with API operations, see Common actions that you can use with AWS Global Accelerator (p. 21) and the AWS Global Accelerator API Reference.

Tagging in AWS Global Accelerator

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

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

- Use tags to track billing information in different categories. To do this, apply tags to accelerators or other AWS resources (such as Network Load Balancers, Application Load Balancers, or Amazon EC2 instances) and activate the tags. Then AWS generates a cost allocation report as a comma-separated
Tagging support in Global Accelerator

AWS Global Accelerator supports tagging for accelerators.

Global Accelerator supports the tag-based access control feature of AWS Identity and Access Management (IAM). For more information, see ABAC with Global Accelerator (p. 97).

Adding, editing, and deleting tags in Global Accelerator

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

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

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

2. Choose the accelerator that you want to add or update tags for.
3. In the Tags section, you can do the following:

   **Add a tag**

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

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

**Delete a tag**

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

4. Choose **Save changes**.

---

**Pricing for AWS Global Accelerator**

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

For details about pricing, information about pricing by source and destination Regions, and a pricing example, see [AWS Global Accelerator pricing](https://aws.amazon.com/global-accelerator/pricing/).
Getting started with AWS Global Accelerator

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

To learn more about the two types of accelerators you can create in Global Accelerator, see Work with standard accelerators in AWS Global Accelerator (p. 23) and Work with custom routing accelerators in AWS Global Accelerator (p. 43).

The tutorials provide steps that primarily use AWS Management Console. (When you set up a custom routing accelerator, you must use the API for certain configuration steps.) You can also use Global Accelerator API operations with the AWS Command Line Interface (AWS CLI) or AWS SDKs to create and customize your accelerators. For a list of API operations, see Common actions that you can use with AWS Global Accelerator (p. 21). For detailed information about working with AWS Global Accelerator API operations, see the AWS Global Accelerator API Reference.

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

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

Topics
• Getting started with a standard accelerator (p. 14)
• Getting started with a custom routing accelerator (p. 17)

Getting started with a standard accelerator

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

Tasks
• Before you begin (p. 14)
• Step 1: Create a standard accelerator (p. 15)
• Step 2: Add listeners (p. 15)
• Step 3: Add endpoint groups (p. 16)
• Step 4: Add endpoints (p. 16)
• Step 5: Test your accelerator (p. 16)
• Step 6 (optional): Delete your accelerator (p. 17)

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 include EC2 instances. For more information, see Create a Network Load Balancer in the User Guide for Network Load Balancers.

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

- When you add an internal Application Load Balancer or an EC2 instance endpoint in Global Accelerator, you enable internet traffic to flow directly to and from the endpoint in virtual private clouds (VPCs) by targeting it in a private subnet. The VPC that contains the load balancer or EC2 instance must have an internet gateway attached to it, to indicate that the VPC accepts internet traffic. For more information, see Secure VPC connections in AWS Global Accelerator (p. 110).
- Global Accelerator requires your router and firewall rules to allow inbound traffic from the IP addresses associated with Amazon Route 53 health checkers to complete health checks for EC2 instance or Elastic IP address endpoints. You can find information about the IP address ranges associated with Route 53 health checkers in IP address ranges of Amazon Route 53 servers in the Amazon Route 53 Developer Guide.

### Step 1: Create a standard accelerator

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

**To create an accelerator**

2. Choose Create accelerator.
3. Provide a name for your accelerator.
4. For Accelerator type, select Standard.
5. For IP address type, select IPv4 or Dual-stack.
6. Optionally, add one or more tags to help you identify your Global Accelerator resources.
7. Choose Next.

### Step 2: Add listeners

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

**To create a listener**

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

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

   For more information, see Client affinity (p. 29).
Step 3: Add endpoint groups

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

To add an endpoint group

1. On the Add endpoint groups page, in the section for a listener, choose a Region from the dropdown list.
2. Optionally, for Traffic dial, enter a number from 0 to 100 to set a percentage of traffic for this endpoint group. The percentage is applied only to the traffic already directed to this endpoint group, not all listener traffic. By default, the traffic dial for an endpoint group is set to 100 (that is, 100%).
3. Optionally, for custom health check values, choose Configure health checks. When you configure health check settings, Global Accelerator uses the settings for health checks for EC2 instance and Elastic IP address endpoints. For Network Load Balancer and Application Load Balancer endpoints, Global Accelerator uses the health check settings that you've already configured for the load balancers themselves. For more information, see Changing health check options (p. 33).
4. Optionally, choose Add endpoint group to add additional endpoint groups for this listener or other listeners.
5. Choose Next.

Step 4: Add endpoints

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

To add endpoints

1. On the Create endpoints page, in the section for an endpoint, choose an Endpoint.
2. Optionally, for Weight, enter a number from 0 to 255 to set a weight for routing traffic to this endpoint. When you add weights to endpoints, you configure Global Accelerator to route traffic based on proportions that you specify. By default, all endpoints have a weight of 128. For more information, see Endpoint weights (p. 37).
3. Optionally, under Preserve client IP address, select Preserve address. (For some endpoint types, this option is selected and can't be cleared.) For more information, see Preserve client IP addresses in AWS Global Accelerator (p. 64).
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.
Step 6 (optional): Delete your accelerator

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

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

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

- When you create an accelerator, Global Accelerator provides you with a set of two static IP addresses. The IP addresses are assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you delete an accelerator, you lose the static IP addresses that are assigned to the accelerator, so you can no longer route traffic by using them. As a best practice, ensure that you have permissions in place to avoid inadvertently deleting accelerators. You can use IAM policies with Global Accelerator, for example, tag-based permissions, to limit the users who have permissions to delete an accelerator. For more information, see ABAC with Global Accelerator (p. 97).

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

To delete an accelerator

2. Choose the accelerator that you want to delete.
3. Choose Edit.
4. Choose Disable accelerator, and then choose Save.
5. Choose the accelerator that you want to delete.
6. Choose Delete accelerator.
7. In the confirmation dialog box, choose Delete.

Getting started with a custom routing accelerator

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

Tasks

- Before you begin (p. 18)
Before you create a custom routing accelerator, create a resource that you can add as an endpoint to direct traffic to. A custom routing accelerator endpoint must be a virtual private cloud (VPC) subnet, which can include multiple Amazon EC2 instances. For instructions for creating the resources see the following:

- Create a VPC subnet. For more information, see [Create and Configure Your VPC](https://docs.aws.amazon.com/DirectoryService/latest/dg/vpc-configuration.html) in the AWS Directory Service Administration Guide.
- Optionally, launch one or more Amazon EC2 instances in your VPC. For more information, see [Create your EC2 resources and launch your EC2 instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-launch.html) in the Amazon EC2 User Guide for Linux Instances.

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

- When you add an EC2 instance endpoint in Global Accelerator, you enable internet traffic to flow directly to and from the endpoint in VPCs by targeting it in a private subnet. The VPC that contains the EC2 instance must have an internet gateway attached to it, to indicate that the VPC accepts internet traffic. For more information, see [Secure VPC connections in AWS Global Accelerator](https://docs.aws.amazon.com/GlobalAccel/latest/GlobalAccelForeColornsis/0110-secure-vpc-connections.html).

Before you create a custom routing accelerator, make sure that you review the best practices described in [Guidelines and restrictions for custom routing accelerators](https://docs.aws.amazon.com/GlobalAccel/latest/GlobalAccelForeColornsis/0150-guidelines-for-accelerators.html) (p. 46).

### Step 1: Create a custom routing accelerator

**To create an accelerator**

2. Provide a name for your accelerator.
3. For **Accelerator type**, select **Custom routing**.
4. Optionally, add one or more tags to help you identify your accelerator resources.
5. Choose Next to add listeners, endpoint groups, and VPC subnet endpoints.

### Step 2: Add listeners

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

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

**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.
Step 3: Add endpoint groups

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

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

To add an endpoint group

1. On the Add endpoint groups page, in the section for a listener, choose a Region.
2. For Ports and protocols sets, enter port ranges and protocols for your Amazon EC2 instances.
   - Enter a From port and a To port to specify a range of ports.
   - For each port range, specify the protocol or protocols for that range.

   The port range doesn’t have to be a subset of your listener port range, but there must be enough total ports in the listener port range to support the total number of ports that you specify.
3. Choose Save.
4. Optionally, choose Add endpoint group to add additional endpoint groups for this listener or other listeners.
5. Choose Next.

Step 4: Add VPC subnet endpoints

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

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

To add endpoints

1. On the Add endpoints page, in the section for the endpoint group that you want to add the endpoint to, choose a subnet ID for Endpoint.
2. Optionally, do one of the following to enable traffic to EC2 instance destinations in the subnet:
   - To allow traffic to be directed to all EC2 endpoints and ports on the subnet, select Allow all traffic
   - To allow traffic to specific EC2 endpoints and ports on the subnet, select Allow traffic to specific destination socket addresses. Then specify the IP addresses and ports or port ranges to allow. Finally, choose Allow these destinations.

By default, no traffic is allowed to subnet endpoints. If you don’t select an option to allow traffic, traffic is denied to all destinations in the subnet.
Step 5 (optional): Delete your accelerator

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

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

Be aware of the following when you delete an accelerator:

- When you create an accelerator, Global Accelerator provides you with a set of two static IP addresses. The IP addresses are assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you delete an accelerator, you lose the static IP addresses that are assigned to the accelerator, so you can no longer route traffic by using them. As a best practice, ensure that you have permissions in place to avoid inadvertently deleting accelerators. You can use IAM policies like tag-based permissions with Global Accelerator to limit the users who have permissions to delete an accelerator. For more information, see ABAC with Global Accelerator (p. 97).

To delete an accelerator:

2. Choose the accelerator that you want to delete.
3. Choose Edit.
4. Choose Disable accelerator, and then choose Save.
5. Choose the accelerator that you want to delete.
6. Choose Delete accelerator.
7. In the confirmation dialog box, choose Delete.
Common actions that you can use with AWS Global Accelerator

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

Actions to use with standard accelerators

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

<table>
<thead>
<tr>
<th>Action</th>
<th>Using the Global Accelerator Console</th>
<th>Using the Global Accelerator API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a standard accelerator</td>
<td>See Getting started with a standard accelerator (p. 14)</td>
<td>See CreateAccelerator</td>
</tr>
<tr>
<td>Create a listener for a standard accelerator</td>
<td>See Listeners for standard accelerators in AWS Global Accelerator (p. 28)</td>
<td>See CreateListener</td>
</tr>
<tr>
<td>Create an endpoint group for a standard accelerator</td>
<td>See Endpoint groups for standard accelerators in AWS Global Accelerator (p. 29)</td>
<td>See CreateEndpointGroup</td>
</tr>
<tr>
<td>Update a standard accelerator</td>
<td>See Standard accelerators in AWS Global Accelerator (p. 23)</td>
<td>See UpdateAccelerator</td>
</tr>
<tr>
<td>Update an endpoint group</td>
<td>See Adding, editing, or removing a standard endpoint group (p. 30)</td>
<td>See UpdateEndpointGroup</td>
</tr>
<tr>
<td>Add an endpoint</td>
<td>See Adding, editing, or removing a standard endpoint (p. 36)</td>
<td>See AddEndpoints</td>
</tr>
<tr>
<td>Remove an endpoint</td>
<td>See Adding, editing, or removing a standard endpoint (p. 36)</td>
<td>See RemoveEndpoints</td>
</tr>
<tr>
<td>List standard accelerators</td>
<td>See Viewing your accelerators (p. 26)</td>
<td>See ListAccelerator</td>
</tr>
<tr>
<td>Get all information about an accelerator</td>
<td>See Viewing your accelerators (p. 26)</td>
<td>See DescribeAccelerator</td>
</tr>
<tr>
<td>Delete an accelerator</td>
<td>See Creating or updating a standard accelerator (p. 24)</td>
<td>See DeleteAccelerator</td>
</tr>
</tbody>
</table>

Actions to use with custom routing accelerators

The following table lists common Global Accelerator actions that you can use with custom routing accelerators, with links to relevant documentation.
<table>
<thead>
<tr>
<th>Action</th>
<th>Using the Global Accelerator Console</th>
<th>Using the Global Accelerator API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a custom routing accelerator</td>
<td>See [Getting started with a custom routing accelerator](p. 17)</td>
<td>See [CreateCustomRoutingAccelerator](p. 49)</td>
</tr>
<tr>
<td>Create a listener for a custom routing accelerator</td>
<td>See [Listeners for custom routing accelerators in AWS Global Accelerator](p. 50)</td>
<td>See [CreateCustomRoutingListener](p. 49)</td>
</tr>
<tr>
<td>Create an endpoint group for a custom routing accelerator</td>
<td>See [Endpoint groups for custom routing accelerators in AWS Global Accelerator](p. 51)</td>
<td>See [CreateCustomRoutingEndpointGroup](p. 49)</td>
</tr>
<tr>
<td>Update a custom routing accelerator</td>
<td>See [Custom routing accelerators in AWS Global Accelerator](p. 48)</td>
<td>See [UpdateCustomRoutingAccelerator](p. 49)</td>
</tr>
<tr>
<td>List your custom routing accelerators</td>
<td>See [Viewing your custom routing accelerators](p. 49)</td>
<td>See [ListCustomRoutingAccelerator](p. 49)</td>
</tr>
<tr>
<td>Get all information about a custom routing accelerator</td>
<td>See [Viewing your custom routing accelerators](p. 49)</td>
<td>See [DescribeCustomRoutingAccelerator](p. 49)</td>
</tr>
<tr>
<td>Delete a custom routing accelerator</td>
<td>See [Creating or updating a custom routing accelerator](p. 49)</td>
<td>See [DeleteCustomRoutingAccelerator](p. 49)</td>
</tr>
<tr>
<td>Get the static port mapping for a custom routing accelerator</td>
<td>N/A</td>
<td>See [ListCustomRoutingPortMappings](p. 49)</td>
</tr>
<tr>
<td>Allow all destination traffic for a subnet in a custom routing accelerator</td>
<td>See [Adding, editing, or removing a VPC subnet endpoint](p. 53)</td>
<td>See [AllowCustomRoutingTraffic](p. 49)</td>
</tr>
<tr>
<td>Deny all destination traffic for a subnet in a custom routing accelerator</td>
<td>See [Adding, editing, or removing a VPC subnet endpoint](p. 53)</td>
<td>See [DenyCustomRoutingTraffic](p. 49)</td>
</tr>
<tr>
<td>Allow traffic to specific destinations in a custom routing accelerator</td>
<td>See [Adding, editing, or removing a VPC subnet endpoint](p. 53)</td>
<td>See [AllowCustomRoutingTraffic](p. 49)</td>
</tr>
<tr>
<td>Deny traffic to specific destinations in a custom routing accelerator</td>
<td>See [Adding, editing, or removing a VPC subnet endpoint](p. 53)</td>
<td>See [DenyCustomRoutingTraffic](p. 49)</td>
</tr>
</tbody>
</table>
Work with standard accelerators in AWS Global Accelerator

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

If instead you want to use custom application logic to direct one or more users to a specific endpoint among many endpoints, create a custom routing accelerator. For more information, see Work with custom routing accelerators in AWS Global Accelerator (p. 43).

To set up a standard accelerator, do the following:

1. Create an accelerator, and choose the standard accelerator option.
2. For Address type, select IPv4 or Dual-stack.
3. Add a listener with a specific set of ports or port range, and choose the protocol to accept: TCP or UDP.
4. Add one or more endpoint groups, one for each AWS Region in which you have endpoint resources.
5. Add one or more endpoints to the endpoint groups. This isn't required, but traffic won't be routed if you don't have any endpoints. Endpoints can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses.

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

Topics
- Standard accelerators in AWS Global Accelerator (p. 23)
- Listeners for standard accelerators in AWS Global Accelerator (p. 28)
- Endpoint groups for standard accelerators in AWS Global Accelerator (p. 29)
- Endpoints for standard accelerators in AWS Global Accelerator (p. 34)

Standard accelerators in AWS Global Accelerator

A standard accelerator in AWS Global Accelerator directs traffic to optimal endpoints over the AWS global network to improve the availability and performance of your internet applications that have a global audience. Each accelerator includes one or more listeners. A listener processes inbound connections from clients to Global Accelerator, based on the protocol (or protocols) and port (or port range) that you configure.

Global static IP addresses for your accelerator

By default, Global Accelerator provides you with static IP addresses that are associated with your accelerator. The static IP addresses are anycast from the AWS edge network.

For IPv4, Global Accelerator provides two static IPv4 addresses. For dual-stack, Global Accelerator provides a total of four addresses: two static IPv4 addresses and two static IPv6 addresses. If you bring your own IP address range to AWS (BYOIP) to use with Global Accelerator (IPv4 only), you can instead assign IPv4 addresses from your own pool to use with your accelerator. For more information, see Bring your own IP addresses (BYOIP) in AWS Global Accelerator (p. 57).
For accelerators with dual-stack, Global Accelerator allocates the IPv6 addresses from the same two /64 CIDR prefixes. This can help simplify steps for allow-listing and setting ACL controls.

You can add IPv4-only endpoints to standard accelerators that are configured for IPv4 IP address types, but accelerators that you configure as dual-stack require that you add only endpoints that also support dual-stack. Only Amazon EC2 instances with an attached primary IPv6 elastic network interface (ENI) and Application Load Balancers can be added as dual-stack endpoints.

**Important**
The IP addresses are assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you delete an accelerator, you lose the Global Accelerator static IP addresses that are assigned to the accelerator, so you can no longer route traffic by using them. As a best practice, ensure that you have permissions in place to avoid inadvertently deleting accelerators. You can use IAM policies with Global Accelerator, for example, tag-based permissions, to limit the users who have permissions to delete an accelerator. For more information, see [ABAC with Global Accelerator](p. 97).

This section explains how to create, edit, or delete a standard accelerator on the Global Accelerator console. If you want to use API operations with Global Accelerator, see the [AWS Global Accelerator API Reference](p. 26).

**Topics**
- Creating or updating a standard accelerator (p. 24)
- Deleting an accelerator (p. 25)
- Viewing your accelerators (p. 26)
- Add an accelerator when you create a load balancer (p. 26)
- Using global static IP addresses instead of regional static IP addresses (p. 27)

### Creating or updating a standard accelerator

This section explains how to create or update standard accelerators on the console. To work with Global Accelerator programmatically, see the [AWS Global Accelerator API Reference](p. 26).

**To create a standard accelerator**

2. Choose Create accelerator.
3. Provide a name for your accelerator.
4. For **Accelerator type**, select **Standard**.
5. For **IP address type**, select **IPv4** or **DUAL-STANDARD**.
6. Optionally, if you brought your own IP address ranges to AWS (BYOIP), you can specify a static IP address for your accelerator, one from each address pool. Make this choice for each of the two static IP addresses for your accelerator.
   - For each static IP address, choose the IP address pool to use.
   - **Note**
     You must choose a different IP address pool for each static IP address. This restriction is because Global Accelerator assigns each address range to a different network zone, for high availability.
   - If you chose your own IP address pool, also choose a specific IP address from the pool. If you choose the default Amazon IP address pool, Global Accelerator assigns a specific IP address to your accelerator.
7. Optionally, add one or more tags to help you identify your accelerator resources.
8. Choose **Next** to add listeners, endpoint groups, and endpoints.
To edit a standard accelerator

2. In the list of accelerators, choose one, and then choose Edit.
3. On the Edit accelerator page, make changes, such as the following:
   - Change the name of the accelerator.
   - Disable the accelerator so that it no longer accepts or routes traffic, or so that you can delete it.
   - Enable the accelerator, if it is disabled.
   - Update the IP address type. If it's set to IPv4, change it to dual-stack. Or if it's dual-stack, change it to IPv4.
   - Update tags.
4. Choose Save changes.

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

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

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

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

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

- Adding, editing, or removing a standard listener (p. 28)
- Adding, editing, or removing a standard endpoint group (p. 30)
- Adding, editing, or removing a standard endpoint (p. 36)

Deleting an accelerator

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

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

To disable an accelerator

2. In the list, choose an accelerator that you want to disable.
3. Choose Edit.
4. Choose Disable accelerator, and then choose Save.

To delete an accelerator
2. In the list, choose an accelerator that you want to delete.
3. Choose Delete.

Note
If you haven't disabled the accelerator, Delete is unavailable.
4. In the confirmation dialog box, choose Delete.

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

Viewing your accelerators

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

To view information about your accelerator
2. To see details about an accelerator, in the list, choose an accelerator, and then choose View.

Add an accelerator when you create a load balancer

When you create an Application Load Balancer in the AWS Management Console, you can optionally add an accelerator at the same time. Elastic Load Balancing and Global Accelerator work together to transparently add the accelerator for you. The accelerator is created in your account, with the load balancer as an endpoint. Using an accelerator provides static IP addresses and improves the availability and performance of your applications. (Learn more about accelerators by reading What is AWS Global Accelerator? (p. 1).)

Important
To create an accelerator, you must have the correct permissions in place. For more information, see Identity-based policy examples for AWS Global Accelerator (p. 99).

Configure and view your accelerator

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

After you create your load balancer by choosing the Global Accelerator add-on on the Amazon EC2 console, go to the Integrated services tab to see the static IP addresses and Domain Name System (DNS) name for your accelerator. You use this information to start routing user traffic to the load balancer over the AWS global network. For more information about the DNS name assigned to your accelerator, see DNS addressing and custom domains in AWS Global Accelerator (p. 56).
You can view and configure your accelerator by navigating to Global Accelerator in the AWS Management Console. For example, you can see the accelerators that are associated with your account or add additional load balancers to your accelerator. For more information, see Viewing your accelerators (p. 26) and Creating or updating a standard accelerator (p. 24).

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

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

1. Update your DNS configuration to point your traffic directly to the load balancer.
2. Delete the load balancer from the accelerator. For more information, see To remove an endpoint in Adding, editing, or removing a standard endpoint (p. 36).
3. Delete the accelerator. For more information, see Deleting an accelerator (p. 25).

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

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

Opting to use global static IP addresses provisioned by Global Accelerator can also improve the availability and performance of your applications. With Global Accelerator, static IP addresses accept incoming traffic onto the AWS global network from the edge location that is closest to your users. Maximizing time that traffic is on the AWS network can provide a faster and better customer experience. For more information, see How AWS Global Accelerator works (p. 5).

You can add an accelerator from the AWS Management Console or by using API operations with the AWS CLI or SDKs. For more information, see Creating or updating a standard accelerator (p. 24).

Note the following when you add an accelerator:

- The global static IP addresses provisioned by Global Accelerator remain assigned to you for as long as your accelerator exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, if you delete an accelerator, you lose the static IP addresses that are assigned to it. For more information, see Deleting an accelerator (p. 25).
- With Global Accelerator, you pay only for what you use. You are charged an hourly rate and data transfer costs for each accelerator in your account. For more information, see AWS Global Accelerator Pricing.
Listeners for standard accelerators in AWS Global Accelerator

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

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

Topics
- Adding, editing, or removing a standard listener (p. 28)
- Client affinity (p. 29)

Adding, editing, or removing a standard listener

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

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

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

   For more information, see Client affinity (p. 29).
7. Choose Add listener.

To edit a standard listener
2. On the accelerators page, choose an accelerator.
3. Choose a listener, and then choose Edit listener.
4. On the Edit listener page, change the ports, port ranges, or protocols that you want to associate with the listener.
5. Optionally, choose to enable client affinity. Client affinity for a listener means that Global Accelerator ensures that connections from a specific source (client) IP address are always routed to the same endpoint. To enable this behavior, in the dropdown list, choose Source IP.

   The default is None, which means that client affinity is not enabled and Global Accelerator distributes traffic equally between the endpoints in the endpoint groups for the listener.
For more information, see Client affinity (p. 29).

6. Choose Save.

To remove a listener

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

Client affinity

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

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

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

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

Note

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

Endpoint groups for standard accelerators in AWS Global Accelerator

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

Global Accelerator directs traffic to endpoint groups in standard accelerators based on the location of the client and the health of the endpoint group. If you like, you can also set the percentage of traffic to send to an endpoint group. You do that by using the traffic dial to increase (dial up) or decrease (dial down) traffic to the group. The percentage is applied only to the traffic that Global Accelerator is already directing to the endpoint group, not all traffic coming to a listener.
You can define health check settings for Global Accelerator for each endpoint group. By updating health check settings, you can change your requirements for polling and verifying the health of Amazon EC2 instance and Elastic IP address endpoints. For Network Load Balancer and Application Load Balancer endpoints, configure health check settings on the Elastic Load Balancing console.

Global Accelerator continually monitors the health of all endpoints that are included in a standard endpoint group, and routes requests only to the active endpoints that are healthy. For more information, see Changing health check options (p. 33) if there aren’t any healthy endpoints to route traffic to, Global Accelerator routes requests to all endpoints.

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

**Topics**

- Adding, editing, or removing a standard endpoint group (p. 30)
- Adjusting traffic flow with traffic dials (p. 31)
- Overriding listener ports (p. 32)
- Changing health check options (p. 33)

### Adding, editing, or removing a standard endpoint group

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

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

**To add a standard endpoint group**

2. On the Accelerators page, choose an accelerator.
3. In the Listeners section, for Listener ID, choose the ID of the listener that you want to add an endpoint group to.
4. Choose Add endpoint group.
5. In the section for a listener, specify a Region for the endpoint group by choosing one from the dropdown list.
6. Optionally, for Traffic dial, enter a number from 0 to 100 to set a percentage of traffic for this endpoint group. The percentage is applied only to the traffic that is already directed to this endpoint group, not all listener traffic. By default, the traffic dial is set to 100.
7. Optionally, to override the listener port used for routing traffic to endpoints and reroute traffic to specific ports on your endpoints, choose Configure port overrides. For more information, see Overriding listener ports (p. 32).
8. Optionally, to specify custom health check values to be applied to EC2 instance and Elastic IP address endpoints, choose Configure health checks. For more information, see Changing health check options (p. 33).
9. Optionally, choose Add endpoint group to add additional endpoint groups for this listener or other listeners.
10. Choose Add endpoint group.
To edit an endpoint group

2. On the Accelerators page, choose an accelerator.
3. In the Listeners section, for Listener ID, choose the ID of the listener that the endpoint group is associated with.
4. Choose Edit endpoint group.
5. On the Edit endpoint group page, change the Region, adjust the traffic dial percentage, or choose Configure health checks to modify the health check settings.
6. Choose Save.

To remove a standard endpoint group

2. On the Accelerators page, choose an accelerator.
3. In the Listeners section, choose a listener.
4. In the Endpoint groups section, choose an endpoint group, and then choose Remove.
5. On the confirmation dialog box, choose Remove.

Adjusting traffic flow with traffic dials

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

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

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

Upgrade your application by Region

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

Mix traffic between two Regions

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

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

- The first 25 requests on each coast (50 requests in total) are served from their nearby endpoint group. That is, 25 requests are directed to the endpoint group in us-west-2 and 25 are directed to the endpoint group in us-east-1.
- The next 50 requests are directed to the opposite Regions. That is, the next 25 requests from the East Coast are served by us-west-2, and the next 25 requests from the West Coast are served by us-east-1.
The result in this scenario is that both endpoint groups serve the same amount of traffic. However, each one receives a mix of traffic from both Regions.

**Load sharing multi-Region architectures**

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

### Overriding listener ports

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

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

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

Overriding a port can also help you avoid issues with listening on restricted ports. It's safer to run applications that don't require superuser (root) privileges on your endpoints. However, in Linux and other UNIX-like systems, you must have superuser privileges to listen on restricted ports (TCP or UDP ports below 1024). By mapping a restricted port on a listener to a non-restricted port on an endpoint, you avoid this issue. You can accept traffic on restricted ports while running applications without root access on your endpoints behind Global Accelerator. For example, you can override a listener port 443 to an endpoint port 8443.

For each port override, you specify a listener port that accepts traffic from users and the endpoint port that Global Accelerator will route that traffic to. For more information, see [Adding, editing, or removing a standard endpoint group](#).

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

- **Endpoint ports can't overlap listener port ranges.** The endpoint ports that you specify in a port override cannot be included in any of the listener port ranges that you've configured for the accelerator. For example, say that you have two listeners for an accelerator, and you've defined the port ranges for those listeners as 100-199 and 200-299, respectively. When you create a port override, you can't define one from listener port 100 to endpoint port 210, for example, because the endpoint port (210) is included in a listener port range that you defined (200-299).

- **No duplicate endpoint ports.** If one port override in an accelerator specifies an endpoint port, you can't specify the same endpoint port with port override from a different listener port. For example, you can't specify a port override from listener port 80 to endpoint port 90 together with an override from listener port 81 to endpoint port 90.

- **Health check continues to use the original port.** If you specify a port override for a port that is configured as a health check port, the health check still uses the original port, not the override port. For example, say that you specify health checks on listener port 80, and you also specify a port override from listener port 80 to endpoint port 480. Health checks continue to use endpoint port 80. However, user traffic that comes in through port 80 goes to port 480 on the endpoint.
This behavior maintains consistency between Network Load Balancer, Application Load Balancer, EC2 instance, and Elastic IP address endpoints. Because Network Load Balancers and Application Load Balancers don’t map health check ports to a different endpoint ports when you specify a port override in Global Accelerator, it would be inconsistent for Global Accelerator to map health check ports to different endpoint ports for EC2 instance and Elastic IP address endpoints.

- **Security group settings must allow port access.** Make sure that your security groups allow traffic to arrive at endpoint ports that you've designated in port overrides. For example, if you override listener port 443 to endpoint port 1433, make sure that any port restrictions set in your security group for that Application Load Balancer or Amazon EC2 endpoint allow inbound traffic on port 1433.

### Changing health check options

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

Note that if there aren’t any healthy endpoints to route traffic to, Global Accelerator routes incoming client requests to **all** endpoints in the endpoint group. For more information, see [[236, 256] failover for unhealthy endpoints](#).

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

### Security and access for health checks

Global Accelerator requires that your router and firewall rules allow inbound traffic from the IP addresses associated with Amazon Route 53 health checkers to complete health checks for EC2 instance or Elastic IP address endpoints. To see the list of IP address ranges associated with Route 53 health checkers, see [IP address ranges of Route 53 servers](#) in the [Amazon Route 53 Developer Guide](#).

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

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

### Health check guidance for different endpoint types

Overall, make sure that the health checks that you choose for endpoints with HTTP workloads are representative of the overall health of your application, and that you follow the guidance in the [Security and access for health checks](#) section.

In addition, review the following information about health checks for each endpoint type.

- For Network Load Balancer or Application Load Balancer endpoints, be aware of the following:
  - The [health check options](#) that you choose in Global Accelerator do not affect Network Load Balancers or Application Load Balancers that you’ve added as endpoints. That is, health check
Endpoints for standard accelerators in AWS Global Accelerator

Endpoints for standard accelerators in AWS Global Accelerator can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses. A static IP address serves as a single point of contact for clients, and, with a standard accelerator, Global Accelerator distributes options that you specify in Global Accelerator are used for Amazon EC2 and Elastic IP address health checks but not for health checks on load balancer endpoints.

For load balancer endpoints, configure health checks by using Elastic Load Balancing configuration options. For more information, see Health checks for your target groups.

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

- For EC2 instance or Elastic IP address endpoints, be aware of the following:
  - When you add EC2 instance or Elastic IP address endpoints to a listener configured with TCP, you can specify the port to use for health checks. By default, if you don't specify a port for health checks, Global Accelerator uses the listener port that you specified for your accelerator.
  - When you add these endpoint types with a UDP listener, Global Accelerator uses the listener port and the TCP protocol for health checks, so you must have a TCP server on your endpoint.

Make sure to check that the port that you've configured for the TCP server on each endpoint is the same as the port that you specify for the health check in Global Accelerator. If the port numbers aren't the same, or if you haven't set up a TCP server for the endpoint, Global Accelerator marks the endpoint as unhealthy, regardless of the endpoint's health.

- Make sure to follow the guidance for security and access (p. 33) when you configure ports for health checks for your EC2 instance or Elastic IP address endpoints.

Setting health check options

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

Health check port

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

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

Health check protocol

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

Health check interval

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

Threshold count

The number of consecutive health checks required before considering an unhealthy target healthy or a healthy target unhealthy.
incoming traffic across healthy endpoints. Global Accelerator directs traffic to endpoints by using the port (or port range) that you specify for the listener that the endpoint group for the endpoint belongs to.

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

Important
Accelerators that you configure as dual-stack require that you add only endpoints that also support dual-stack. Only Application Load Balancers and Amazon EC2 instances can be added as dual-stack endpoints.

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

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

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

**Network Load Balancer endpoints**
- A Network Load Balancer endpoint can be internet-facing or internal.
- For Network Load Balancer endpoints, we recommend that you disable cross-zone traffic for the load balancers to avoid connection collisions, which can result in increased TCP connection time. For more information, see [TCP Connection Delays](https://docs.aws.amazon.com/ingress/latest/lb/TCP-Connection-Delays.html) in the [User Guide for Network Load Balancers](https://docs.aws.amazon.com/ingress/latest/lb/).
- Global Accelerator only supports Network Load Balancers running inside an AWS Region. Global Accelerator does not support a Network Load Balancer running as an endpoint in a Local Zone.
- Network Load Balancers cannot be added as dual-stack endpoints.

**Amazon EC2 instance endpoints**
- An EC2 instance endpoint can't be one of the following types: C1, CC1, CC2, CG1, CG2, CR1, CS1, G1, G2, H1, HS1, M1, M2, M3, or T1.
- EC2 instances are supported as endpoints in specific AWS Regions. For more information, see [AWS Region availability for AWS Global Accelerator](https://docs.aws.amazon.com/globalaccelerator/latest/dg/region-availability.html) (p. 3).
- Global Accelerator only supports EC2 instances inside an AWS Region. Global Accelerator does not support routing to an Elastic IP address as an endpoint in a Local Zone.
- We recommend that you remove an EC2 instance from Global Accelerator endpoint groups before you terminate the instance. If you terminate an EC2 instance before you remove it from an endpoint group in Global Accelerator, and then you create another instance in the same VPC with the same private IP address, and health checks pass, Global Accelerator will route traffic to the new endpoint.
- EC2 instances can be added as dual-stack endpoints. However, the instances must have a primary IPv6 elastic network interface (ENI) attached to them. For more information, see [Work with network interfaces](https://docs.aws.amazon.com/elasticloadbalancing/latest/network/generic-ip.html) in the Amazon Elastic Compute Cloud User Guide.

For all endpoints, when you configure resources as endpoints behind Global Accelerator, we recommend that you don't also send traffic directly to the same endpoints over the internet. Sending direct traffic can lead to connection collision issues.

In addition, be aware that the resources that you add as endpoints for an accelerator and the accelerator itself must be owned by the same account. However, the target instances behind a load balancer...
Adding, editing, or removing a standard endpoint

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

Endpoints in Global Accelerator can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses. You must create one of those resources first, and then you can add it as an endpoint in Global Accelerator. A resource must be valid and active when you add it as an endpoint.

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

If you're adding an endpoint with client IP address preservation, first review the information in AWS Region availability for AWS Global Accelerator (p. 3) and Preserve client IP addresses in AWS Global Accelerator (p. 64).

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

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

To add a standard endpoint

2. On the Accelerators page, choose an accelerator.
3. In the Listeners section, for Listener ID, choose the ID of a listener.
4. In the Endpoint groups section, for Endpoint group ID, choose the ID of the endpoint group that you want to add an endpoint to.
5. Choose Edit.
6. In the Endpoints section, choose Add endpoint.
7. On the Add endpoints page, choose a resource from the dropdown list.
If you don't have any AWS resources, there aren't any items in the list. To continue, create AWS resources such as load balancers, Amazon EC2 instances, or Elastic IP addresses. Then come back to the steps here, and choose a resource from the list.

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

8. **Optional**, 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](#).

9. Optionally, enable client IP address preservation for an internet-facing Application Load Balancer endpoint. Under **Preserve client IP address**, select **Preserve address**. This option is always selected for internal Application Load Balancer and EC2 instance endpoints, and never selected for Network Load Balancer and Elastic IP address endpoints. For more information, see [Preserve client IP addresses in AWS Global Accelerator](#). Note
Before you add and begin to route traffic to endpoints that preserve the client IP address, make sure that all your required security configurations, for example, security groups, are updated to include the user client IP address on allow lists.

10. Choose **Add endpoint**.

**To edit a standard endpoint**
You can edit an endpoint configuration to change the weight. For more information, see [Endpoint weights](#).

2. On the **accelerators** page, choose an accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of a listener.
4. In the **Endpoint groups** section, for **Endpoint group ID**, choose the ID of the endpoint group.
5. Choose **Edit endpoint**.
6. On the **Edit endpoint** page, make updates, and then choose **Save**.

**To remove an endpoint**

2. On the **accelerators** page, choose an accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of a listener.
4. In the **Endpoint groups** section, for **Endpoint group ID**, choose the ID of the endpoint group.
5. Choose **Remove endpoint**.
6. In the confirmation dialog box, choose **Remove**.

**Endpoint weights**
A weight is a value that determines the proportion of traffic that Global Accelerator directs to an endpoint in a standard accelerator. Endpoints can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses. Global Accelerator calculates the sum of the weights for the endpoints in an endpoint group, and then directs traffic to the endpoints based on the ratio of each endpoint's weight to the total.
Weighted routing lets you choose how much traffic is routed to a resource in an endpoint group. This can be useful in several ways, including load balancing and testing new versions of an application.

How endpoint weights work

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

\[
\text{Weight for a specified endpoint} \div \text{Sum of the weights for all endpoints}
\]

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

Failover for unhealthy endpoints

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

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

Note the following:

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

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

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

Avoiding connection collisions that result in TCP connection time delays

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

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

**Scenarios that can cause connection collisions**

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

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

For Network Load Balancer endpoints, we recommend that you disable cross-zone traffic for the load balancers to avoid connection collisions. For more information, see [TCP Connection Delays](https://docs.aws.amazon.com/glu/latest/userguide/tcp_connections.html) in the [User Guide for Network Load Balancers](https://docs.aws.amazon.com/glu/latest/userguide/tcp_connections.html).

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

The next section provides specific examples for each of the scenarios of how you can configure port overrides to avoid connection collisions. For more information about configuring port overrides, see [Overriding listener ports](https://docs.aws.amazon.com/glu/latest/userguide/tcp_connections.html#port_overrides) (p. 32).

**How to prevent connection collisions by using port overrides**

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

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

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

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

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

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

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

### Adding or updating endpoints with client IP address preservation

A feature that you can use with some endpoint types is **client IP address preservation**. With this feature, AWS Global Accelerator preserves the source IP address of the original client for packets that arrive at the endpoint.

You can use this feature with endpoints that are Application Load Balancers, Network Load Balancers with security groups, and Amazon EC2 instances. Endpoints on custom routing accelerators always have the client IP address preserved. For more information, see [Preserve client IP addresses in AWS Global Accelerator](p. 64).

### About adding endpoints with client IP address preservation

If you intend to use the client IP address preservation feature, be aware of the following when you add endpoints to Global Accelerator. For more information about best practices with client IP address preservation, see [Best practices for client IP address preservation](p. 67).

#### 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. 67).

#### Endpoints in private subnets

You can target an Application Load Balancer, Network 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. 110).

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

#### Add the client IP address to the allow list

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

**Configure network access control lists (ACLs)**

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

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

Be aware of the following for security groups and WAF:

- Security group and AWS WAF rules are an additional set of capabilities that you can apply to protect your resources. For example, the inbound security group rules associated with your Amazon EC2 instances and Application Load Balancers allow you to control the destination ports that clients can connect to through Global Accelerator, such as port 80 for HTTP or port 443 for HTTPS.
- Amazon EC2 instance security groups apply to any traffic that arrives to your instances, including traffic from Global Accelerator and any public or Elastic IP address that is assigned to your instance.

**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, Network Load Balancer with security groups, or an Elastic IP address endpoint to a corresponding endpoint—a corresponding load balancer endpoint or an EC2 instance endpoint—that has client IP address preservation. For more information, see [Preserve client IP addresses in AWS Global Accelerator](p. 64).

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

**Important**

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

Client IP address preservation is supported in all AWS Regions where Global Accelerator is supported. For a list of supported Regions, see [AWS Region availability for AWS Global Accelerator](p. 3).

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, if needed, enable client IP address preservation. (The client IP address preservation option is always selected for internal Application Load Balancers and EC2 instances.)

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

2. On the Accelerators page, choose an accelerator.
3. In the Listeners section, choose a listener.
4. In the Endpoint group section, choose an endpoint group.
5. In the Endpoints section, choose Add endpoint.
6. On the Add endpoints page, in the Endpoints dropdown list, choose an endpoint that supports client IP address preservation.
7. In the Weight field, choose a low number compared to the weights that are set for your existing endpoints. For example, if the weight for a corresponding Application Load Balancer is 255, you could enter a weight of 5 for the new Application Load Balancer, to start with. For more information, see Endpoint weights (p. 37).
8. If needed, under Preserve client IP address, select Preserve address.
9. Choose Save changes.

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

**To reduce traffic for the existing endpoints**

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

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

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

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

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

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

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

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

Endpoints for custom routing accelerators must be virtual private cloud (VPC) subnets, and a custom routing accelerator can only route traffic to Amazon EC2 instances in those subnets. When you create a custom routing accelerator, you can include thousands of Amazon EC2 instances running in a single or multiple VPC subnets. To learn more, see How custom routing accelerators work in AWS Global Accelerator (p. 44).

If instead you want Global Accelerator to automatically choose the closest healthy endpoint to your clients, create a standard accelerator. For more information, see Work with standard accelerators in AWS Global Accelerator (p. 23).

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

1. Review the guidelines and requirements for creating a custom routing accelerator. See Guidelines and restrictions for custom routing accelerators (p. 46).
2. Create a VPC subnet. You can add EC2 instances to the subnet at any time after adding the subnet to Global Accelerator.
3. Create an accelerator, and select the option for a custom routing accelerator.
4. Add a listener and specify a range of ports for Global Accelerator to listen on. Make sure that you include a large range with enough ports for Global Accelerator to map to all the destinations that you expect to have. These ports are distinct from destination ports, which you specify in the next step. For more information about listener port requirements, see Guidelines and restrictions for custom routing accelerators (p. 46).
5. Add one or more endpoint groups for AWS Regions in which you have VPC subnets. You specify the following for each endpoint group:
   - An endpoint port range, which represents the ports on your destination EC2 instances that will be able to receive traffic.
   - The protocol for each destination port range: UDP, TCP, or both UDP and TCP.
6. For the endpoint subnet, select a subnet ID. You can add multiple subnets in each endpoint group and subnets can be different sizes (up to /17).

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

Topics
- How custom routing accelerators work in AWS Global Accelerator (p. 44)
- Guidelines and restrictions for custom routing accelerators (p. 46)
How custom routing accelerators work in AWS Global Accelerator

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

For example, you can use a custom routing accelerator with an online real-time gaming application in which you assign multiple players to a single session on an Amazon EC2 game server based on factors that you choose, such as geographic location, player skill, and game mode. Or you might have a VoIP or social media application that assigns multiple users to a specific media server to for voice, video, and messaging sessions.

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

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

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

**Note**  
By default, all VPC subnet destinations in a custom routing accelerator aren't allowed to receive traffic. This is to be secure by default, and also to give you granular control over which private EC2 instance destinations in your subnet are allowed to receive traffic. You can allow or deny traffic to the subnet, or to specific IP address and port combinations (destination sockets). For more information, see Adding, editing, or removing a VPC subnet endpoint (p. 53). You can also specify destinations by using the Global Accelerator API. For more information, see AllowCustomRoutingTraffic and DenyCustomRoutingTraffic.

**Example of how custom routing works in Global Accelerator**

As an example, let's say that you want to support 10,000 sessions where groups of users interact, such as gaming sessions or VoIP call sessions, across 1,000 Amazon EC2 instances behind Global Accelerator. In this example, we'll specify a listener port range of 10001–20040 and a destination port range of 81–90. We'll say that we have the four VPC subnets in us-east-1: subnet-1, subnet-2, subnet-3, and subnet-4.
In our example configuration, each VPC subnet has a block size of /24 so it can support 251 Amazon EC2 instances. (Five addresses are reserved and unavailable from each subnet, and these addresses are not mapped.) Each server running on each EC2 instance serves the following 10 ports, that we specified for the destination ports in our endpoint group: 81-90. This means that we have 2510 ports (10 x 251) associated with each subnet. Each port can be associated with a session.

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

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

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

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

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

#### Supported endpoint destinations

The virtual public cloud (VPC) subnet endpoints in a custom routing accelerator can only include EC2 instances. No other resources, such as load balancers, are supported for custom routing accelerators.

<table>
<thead>
<tr>
<th>Global Accelerator listener port</th>
<th>VPC subnet</th>
<th>EC2 instance port</th>
</tr>
</thead>
<tbody>
<tr>
<td>10019</td>
<td>192.0.2.5</td>
<td>89</td>
</tr>
<tr>
<td>10020</td>
<td>192.0.2.5</td>
<td>90</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>12501</td>
<td>192.0.2.244</td>
<td>81</td>
</tr>
<tr>
<td>12502</td>
<td>192.0.2.244</td>
<td>82</td>
</tr>
<tr>
<td>12503</td>
<td>192.0.2.244</td>
<td>83</td>
</tr>
<tr>
<td>12504</td>
<td>192.0.2.244</td>
<td>84</td>
</tr>
<tr>
<td>12505</td>
<td>192.0.2.244</td>
<td>85</td>
</tr>
<tr>
<td>12506</td>
<td>192.0.2.244</td>
<td>86</td>
</tr>
<tr>
<td>12507</td>
<td>192.0.2.244</td>
<td>87</td>
</tr>
<tr>
<td>12508</td>
<td>192.0.2.244</td>
<td>88</td>
</tr>
<tr>
<td>12509</td>
<td>192.0.2.244</td>
<td>89</td>
</tr>
<tr>
<td>12510</td>
<td>192.0.2.244</td>
<td>90</td>
</tr>
<tr>
<td>12511</td>
<td>192.0.3.4</td>
<td>81</td>
</tr>
<tr>
<td>12512</td>
<td>192.0.3.4</td>
<td>82</td>
</tr>
<tr>
<td>12513</td>
<td>192.0.3.4</td>
<td>83</td>
</tr>
<tr>
<td>12514</td>
<td>192.0.3.4</td>
<td>84</td>
</tr>
<tr>
<td>12515</td>
<td>192.0.3.4</td>
<td>85</td>
</tr>
<tr>
<td>12516</td>
<td>192.0.3.4</td>
<td>86</td>
</tr>
<tr>
<td>12517</td>
<td>192.0.3.4</td>
<td>87</td>
</tr>
<tr>
<td>12518</td>
<td>192.0.3.4</td>
<td>88</td>
</tr>
<tr>
<td>12519</td>
<td>192.0.3.4</td>
<td>89</td>
</tr>
<tr>
<td>12520</td>
<td>192.0.3.4</td>
<td>90</td>
</tr>
</tbody>
</table>
The types of EC2 instances that are supported with Global Accelerator are listed in Endpoints for standard accelerators in AWS Global Accelerator (p. 34).

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

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

Port mappings

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

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

VPC subnet size

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

IP address type

Custom routing accelerators support only the IPv4 IP address type.

Listener port ranges

You must specify enough listener ports, by specifying listener port ranges, to accommodate the number of destinations included in the subnets that you plan to add to your custom routing accelerator. The range that you specify when you create a listener determines how many listener port and destination IP address combinations that you can use with your custom routing accelerator. For maximum flexibility and to reduce the possibility of getting an error that you don't have enough listener ports available, we recommend that you specify a large port range.

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

Note

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

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

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

Destination port ranges

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

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

A custom routing accelerator routes traffic only to ports on Amazon EC2 instances that are running in virtual private cloud (VPC) subnets. With a custom routing accelerator, Global Accelerator does not route traffic based on the geoproximity or health of the endpoint. To learn more, see How custom routing accelerators work in AWS Global Accelerator (p. 44).

When you create an accelerator, by default, Global Accelerator provides you with a set of two static IPv4 addresses. Custom routing accelerators support only the IPv4 IP address type. If you bring your own IP address range to AWS (BYOIP), you can assign static IPv4 addresses from your own pool to use with your accelerator. For more information, see Bring your own IP addresses (BYOIP) in AWS Global Accelerator (p. 57).

Important
The IP addresses are assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic. However, when you delete an accelerator, you lose the Global Accelerator static IP addresses that are assigned to the accelerator, so you can no longer route traffic by using them. As a best practice, ensure that you have permissions in place to avoid inadvertently deleting accelerators. You can use IAM policies such as tag-based permissions with Global Accelerator to limit the users who have permissions to delete an accelerator. For more information, see ABAC with Global Accelerator (p. 97).

This section explains how to create, edit, or delete a custom routing accelerator on the Global Accelerator console. To learn about using API operations with Global Accelerator, see the AWS Global Accelerator API Reference.
Creating or updating a custom routing accelerator

To create a custom routing accelerator

2. Choose Create accelerator.
3. Provide a name for your accelerator.
4. For Accelerator type, select Custom routing.
5. Optionally, if you have brought your own IP address range to AWS (BYOIP), you can specify static IP addresses for your accelerator from that address pool. Make this choice for each of the two static IP addresses for your accelerator.
   - For each static IP address, choose the IP address pool to use.
   - If you chose your own IP address pool, also choose a specific IP address from the pool. If you chose the default Amazon IP address pool, Global Accelerator assigns a specific IP address to your accelerator.
6. Optionally, add one or more tags to help you identify your accelerator resources.
7. Choose Next to go to the next pages in the wizard to add listeners, endpoint groups, and VPC subnet endpoints.

To edit a custom routing accelerator

2. In the list of custom routing accelerators, choose one, and then choose Edit.
3. On the Edit accelerator page, make any changes that you like. For example, you can disable the accelerator so that you can delete it.
4. Choose Save.

Viewing your custom routing accelerators

You can view information about your custom routing accelerators on the console. To see descriptions of your custom routing accelerators programmatically, see ListCustomRoutingAccelerator and DescribeCustomRoutingAccelerator in the AWS Global Accelerator API Reference.

To view information your custom routing accelerators

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

Deleting a custom routing accelerator

If you created a custom routing accelerator as a test, or if you're no longer using an accelerator, you can delete it. On the console, disable the accelerator, and then you can delete it. You don't have to remove listeners and endpoint groups from the accelerator.
To delete a custom routing accelerator by using an API operation instead of the console, you must first remove all listeners and endpoint groups that are associated with the accelerator, and then disable it. For more information, see the `DeleteAccelerator` operation in the *AWS Global Accelerator API Reference*.

**To disable a custom routing accelerator**

2. In the list, choose an accelerator that you want to disable.
3. Choose *Edit*.
4. Choose *Disable accelerator*, and then choose *Save*.

**To delete a custom routing accelerator**

2. In the list, choose an accelerator that you want to delete.
3. Choose *Delete*.
   
   **Note**
   
   If you haven't disabled the accelerator, *Delete* is unavailable. To disable the accelerator, see the previous procedure.
4. In the confirmation dialog box, choose *Delete*.

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

Listeners for custom routing accelerators in AWS Global Accelerator

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

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

For more information, see How custom routing accelerators work in AWS Global Accelerator (p. 44).

**Adding, editing, or removing a custom routing listener**

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

50
To add a listener for a custom routing accelerator

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

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

2. On the **Accelerators** page, choose a custom routing accelerator.
3. Choose **Add listener**.
4. On the **Add listener** page, enter the listener port range that you want to associate with the accelerator.

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

To edit a listener for a custom routing accelerator

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

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

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

To remove a listener

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

### Endpoint groups for custom routing accelerators in AWS Global Accelerator

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

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

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

To learn more, see How custom routing accelerators work in AWS Global Accelerator (p. 44).

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

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

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

To add an endpoint group for a custom routing accelerator

2. On the Accelerators page, choose a custom routing accelerator.
3. In the Listeners section, for Listener ID, choose the ID of the listener that you want to add an endpoint group to.
4. Choose Add endpoint group.
5. In the section for a listener, specify a Region for the endpoint group.
6. For Ports and protocols sets, enter port ranges and protocols for your Amazon EC2 instances.
   - Enter a From port and a To port to specify a range of ports.
   - For each port range, specify the protocol or protocols for that range.

The port range doesn't have to be a subset of your listener port range, but there must be enough total ports in the listener port range to support the total number of ports that you specify for the endpoint groups in your custom routing accelerator.
7. Choose Save.
8. Optionally, choose Add endpoint group to add additional endpoint groups for this listener. You can also choose another listener and add endpoint groups.
9. Choose Add endpoint group.

To edit an endpoint group for a custom routing accelerator

2. On the Accelerators page, choose a custom routing accelerator.
3. In the Listeners section, for Listener ID, choose the ID of the listener that the endpoint group is associated with.
4. Choose Edit endpoint group.
5. On the Edit endpoint group page, change the Region, the range of ports, or the protocol for a range of ports.
6. Choose **Save**.

**To remove a custom routing accelerator**

2. On the **Accelerators** page, choose an accelerator.
3. In the **Listeners** section, choose a listener, and then choose **Remove**.
4. In the **Endpoint groups** section, choose an endpoint group, and then choose **Remove**.
5. On the confirmation dialog box, choose **Remove**.

---

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

Endpoints for custom routing accelerators are virtual private cloud (VPC) subnets that can receive traffic through an accelerator. Each subnet can contain one or many Amazon EC2 instance destinations. When you add a subnet endpoint, Global Accelerator generates new port mapping. Then you can use the Global Accelerator API to get a static list of all the port mappings for the subnet, which you can use to route traffic to destination EC2 instance IP addresses in the subnet. For more information, see [ListCustomRoutingPortMappings](https://docs.aws.amazon.com/glu/latest/APIReference/API_ListCustomRoutingPortMappings).

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

- You can only direct traffic to EC2 instances in the subnets, not other resources, like load balancers (in contrast to standard accelerators).
- An EC2 instance destination in a subnet endpoint can’t be one of the following types: C1, CC1, CC2, CG1, CG2, CR1, CS1, G1, G2, HI1, HS1, M1, M2, M3, or T1.
- By default, traffic directed through a custom routing accelerator can’t arrive at any destinations in your subnet. To enable destination instances to receive traffic, you must choose to allow all traffic to the subnet or, alternatively, enable traffic to specific instance IP addresses and ports (destination sockets) in the subnet.

**Important**

Updating a subnet or specific destination to allow or deny traffic takes time to propagate across the internet. To determine if a change has propagated, you can call the [DescribeCustomRoutingAccelerator](https://docs.aws.amazon.com/glu/latest/APIReference/API_DescribeCustomRoutingAccelerator) API action to check the accelerator status. For more information, see [DescribeCustomRoutingAccelerator](https://docs.aws.amazon.com/glu/latest/APIReference/API_DescribeCustomRoutingAccelerator).

- Because VPC subnets preserve the client IP address, you should review the relevant security and configuration information when you add subnets as endpoints for custom routing accelerators. For more information, see [About adding endpoints with client IP address preservation](https://docs.aws.amazon.com/glu/latest/getting_started/about_endpoints.html#about-endpoints-client-ip-preservation).

- When you configure resources as endpoints behind Global Accelerator, we recommend that you don’t also send traffic directly to the same endpoints over the internet. Sending direct traffic can lead to connection collision issues.

To learn more, see [How custom routing accelerators work in AWS Global Accelerator](https://docs.aws.amazon.com/glu/latest/getting_started/how-routing-works.html).

---

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

You add virtual private cloud (VPC) subnet endpoints to endpoint groups in your custom routing accelerators so that you can direct user traffic to destination Amazon EC2 instances in the subnet.
When you add and remove EC2 instances from the subnet, or enable or disable traffic to EC2 destinations, you change whether those destinations can receive traffic. However the Global Accelerator port mapping doesn't change.

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

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

The steps in this section explain how to add, edit, or remove VPC subnet endpoints on the AWS Global Accelerator console. To learn about using API operations with AWS Global Accelerator, see the [AWS Global Accelerator API Reference](https://docs.aws.amazon.com/ga-api/ga-developerguide/ga-api-reference.html).

### To add a VPC subnet endpoint

2. On the **Accelerators** page, choose a custom routing accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of a listener.
4. In the **Endpoint groups** section, for **Endpoint group ID**, choose the ID of the endpoint group (AWS Region) that you want to add the VPC subnet endpoint to.
5. In the **Endpoints** section, choose **Add endpoint**.
6. On the **Add endpoints** page, for **Endpoint**, choose a VPC subnet.
   
   If you don’t have any VPCs, there aren’t any items in the list. To continue, add at least one VPC, then come back to the steps here, and choose a VPC from the list.
7. For VPC subnet endpoint that you add, you can choose to allow or deny traffic to all destinations in the subnet, or you can allow traffic to only specific EC2 instances and ports. The default is to deny traffic to all destinations in the subnet.
8. Choose **Add endpoint**.

### To allow or deny traffic to specific destinations

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

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

### To allow or deny ALL traffic to a subnet

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

2. On the **Accelerators** page, choose a custom routing accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of a listener.
4. In the **Endpoint groups** section, for **Endpoint group ID**, choose the ID of the endpoint group (AWS Region) of the VPC subnet endpoint that you want to update.
5. Choose **Allow/Deny all traffic**.
6. Choose an option, to allow all traffic or deny all traffic, and then choose **Save**.

**To remove an endpoint**

2. On the **Accelerators** page, choose a custom routing accelerator.
3. In the **Listeners** section, for **Listener ID**, choose the ID of a listener.
4. In the **Endpoint groups** section, for **Endpoint group ID**, choose the ID of the endpoint group (AWS Region) of the VPC subnet endpoint that you want to remove.
5. Choose **Remove endpoint**.
6. In the confirmation dialog box, choose **Remove**.
DNS addressing and custom domains in AWS Global Accelerator

This chapter explains how AWS Global Accelerator does DNS routing and includes information about using a custom domain with Global Accelerator.

Topics
- Support for DNS addressing in AWS Global Accelerator (p. 56)
- Route custom domain traffic to your accelerator (p. 56)
- Bring your own IP addresses (BYOIP) in AWS Global Accelerator (p. 57)

Support for DNS addressing in AWS Global Accelerator

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

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

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

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

Route custom domain traffic to your accelerator

In most scenarios, you can configure DNS to use your custom domain name (such as `www.example.com`) with your accelerator, instead of using the assigned static IP addresses or the default DNS name. First, using Amazon Route 53 or another DNS provider, create a domain name, and then add or update DNS records with your Global Accelerator IP addresses. Or you can associate your custom domain name
Bring your own IP addresses 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. To see the information for Global Accelerator, scroll down on the Alias Target page.

**Bring your own IP addresses (BYOIP) in AWS Global Accelerator**

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

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

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

**Important**

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

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

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

**Topics**

- Requirements (p. 58)
- Prepare to bring your IP address range to your AWS account: Authorization (p. 58)
- Provision the address range for use with AWS Global Accelerator (p. 60)
- Advertise the address range through AWS (p. 61)
- Deprovision the address range (p. 62)
Requirements

You can bring up to two qualifying IP address ranges to AWS Global Accelerator per AWS account. To qualify, your IP address range must meet the following requirements:

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

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

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

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

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

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

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

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

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

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

Steps to provide authorization

- **Step 1: Create a ROA object (p. 59)**
- **Step 2: Create a self-signed X.509 certificate (p. 59)**
- **Step 3: Create a signed authorization message (p. 60)**

**Step 1: Create a ROA object**

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

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

- ARIN: [ROA Requests](#)
- RIPE: [Managing ROAs](#)
- APNIC: [Route Management](#)

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

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

*Note*

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

**To create and add an X.509 certificate**

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

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

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

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

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

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

   1. View your certificate using the following command.

      ```bash
      cat publickey.cer
      ```
2. Add the certificate by doing the following:

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

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

Step 3: Create a signed authorization message

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

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

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

**To create the signed authorization message**

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

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

2. Sign the authorization message in `text_message` using the key pair that you created in the previous section.

3. Store the message in a variable named `signed_message`, as the following example shows.

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

Provision the address range for use with AWS Global Accelerator

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

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

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

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

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

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

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

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

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

### Advertise the address range through AWS

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

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

**Important**

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

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

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

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

```
aws globalaccelerator advertise-byoip-cidr --cidr 203.0.113.0/24
```

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

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

When your IP address range is advertised, the State returned by `list-byoip-cidrs` is ADVERTISING. For example:
To stop advertising the address range, use the following `withdraw-byoip-cidr` command.

**Important**
To stop advertising your address range, you first must remove any accelerators that have static IP addresses that are allocated from the address pool. To delete an accelerator using the console or using API operations, see [Deleting an accelerator](p. 25).

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

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

```bash
/aws globalaccelerator withdraw-byoip-cidr --cidr 203.0.113.0/24
```

**Deprovision the address range**

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

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

**Step 1: Delete any associated accelerators.** To delete an accelerator using the console or using API operations, see [Deleting an accelerator](p. 25).

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

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

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

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

**Create an accelerator with your IP addresses**

Now you can create an accelerator with your IP addresses. If you brought one address range to AWS, you can assign one IP address to your accelerator. If you brought two address ranges, you can assign one IP address from each address range to your accelerator.

You have several options for creating an accelerator using your own IP addresses for the static IP addresses:
• **Use Global Accelerator console to create an accelerator.** For more information, see [Creating or updating a standard accelerator](p. 24) and [Creating or updating a custom routing accelerator](p. 49).

• **Use the Global Accelerator API to create an accelerator.** For more information, including examples of using the CLI, see [CreateAccelerator](p. 24) and [CreateCustomRoutingAccelerator](p. 49) in the AWS Global Accelerator API Reference.
Preserve client IP addresses in AWS Global Accelerator

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

There are three types of endpoints that can preserve the source IP address of the client in incoming packets: Application Load Balancers, Amazon EC2 instances, and Network Load Balancers with security groups. Be aware that you cannot enable client IP address preservation for Network Load Balancers that have TLS listeners attached and that perform IPv4 to IPv6 NAT translation to their EC2 targets.

**Note**
Global Accelerator does not support client IP address preservation for the following endpoint types:
- Network Load Balancers without security groups
- Elastic IP addresses

The default for client IP address preservation depends on the endpoint type:

- When you use an internet-facing Application Load Balancer as an endpoint with Global Accelerator, client IP address preservation is enabled by default for new accelerators. You can choose to disable the option when you create the accelerator or by editing the accelerator later.
- When you use an internal Application Load Balancer or an EC2 instance with Global Accelerator, the endpoint always has client IP address preservation enabled.
- When you use an Network Load Balancer with security groups with Global Accelerator, client IP address preservation is not enabled by default.

Endpoints on custom routing accelerators always have the client IP address preserved.

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

- Before you add and begin to route traffic to endpoints that preserve the client IP address, make sure that all your required security configurations, for example, security groups, are updated to include the user client IP address on allow lists.
- For Network Load Balancers with security groups, client IP address preservation is supported only when targets are in the same VPC as the Network Load Balancer. Traffic must flow directly from the Network Load Balancer to the target.
- Client IP address preservation for Network Load Balancers is supported only for Network Load Balancers with security groups that do not have TLS listeners attached and do not perform IPv4 to IPv6 NAT translation to their EC2 targets.
- You might see client IP addresses in AWS WAF, instead of Global Accelerator IP addresses. Client IP addresses appear in AWS WAF when you configure Global Accelerator for client IP address preservation and you enable AWS WAF to block connections from your Application Load Balancers that don't come from Global Accelerator.
- Client IP address preservation is supported in all AWS Regions where Global Accelerator is supported. For a list of supported Regions, see [AWS Region availability for AWS Global Accelerator](#).
How to enable client IP address preservation

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

Be aware of the following:

- Internal Application Load Balancers and EC2 instances always have client IP address preservation enabled. You can't disable the option for these endpoints.
- When you use the AWS console to create a new accelerator, the option for client IP address preservation is enabled by default for Application Load Balancer endpoints. The option is not enabled by default for Network Load Balancer with security groups endpoints. You can update the option for client IP address preservation for these endpoints at any time after you add it.
- When you use the AWS CLI or an API action to create a new accelerator and you don't specify the option for client IP address preservation, the following is the default setting for client IP address preservation:
  - Internet-facing Application Load Balancer endpoints have client IP address preservation enabled by default.
  - Network Load Balancer with security group endpoints do not have client IP address preservation enabled by default.
  - Client IP address preservation for Network Load Balancers is supported only for Network Load Balancers with security groups that do not have TLS listeners attached and do not perform IPv4 to IPv6 NAT translation to their EC2 targets.
  - Client IP address preservation is not supported for Network Load Balancers without security groups or for Elastic IP addresses.

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

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

For step-by-step transition guidance, see Transitioning endpoints to use client IP address preservation (p. 41).
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 or Network Load Balancer. For more information, see How the client IP address is preserved in AWS Global Accelerator (p. 66). 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 Location and IP address ranges of Global Accelerator Edge servers (p. 9) and Configuring and using flow logs in AWS Global Accelerator (p. 75).

How the client IP address is preserved in AWS Global Accelerator

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

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

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

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

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

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

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

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

```
```

**Best practices for client IP address preservation**

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

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

**Note**
When you add an internal Application Load Balancer or an EC2 instance endpoint in Global Accelerator, you enable internet traffic to flow directly to and from the endpoint in Virtual Private Clouds (VPCs) by targeting it in a private subnet. For more information, see [Secure VPC connections in AWS Global Accelerator](p. 110).

**How Global Accelerator uses elastic network interfaces**

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

The following examples illustrate how this works:

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

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

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

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

**Security groups created by Global Accelerator**

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

- Global Accelerator doesn't delete security groups that it creates. However, Global Accelerator does delete an elastic network interface if it isn't being used by any of the endpoints in accelerators in your account.
- You can use the security groups created by Global Accelerator as a source group in other security groups that you maintain, but Global Accelerator only forwards traffic to the targets that you specify in your VPC.
- If you modify the security group rules created by Global Accelerator, the endpoint might become unhealthy. If that happens, contact AWS Support for assistance.
- Global Accelerator creates a specific security group for each VPC. Elastic network interfaces that are created for the endpoints within a specific VPC all use the same security group, no matter which subnet an elastic network interface is associated with.

**Important**

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

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

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

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

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

**Topics**
- Using Amazon CloudWatch with AWS Global Accelerator (p. 69)
- Configuring and using flow logs in AWS Global Accelerator (p. 75)
- Using AWS CloudTrail to log AWS Global Accelerator API calls (p. 81)

Using Amazon CloudWatch with AWS Global Accelerator

AWS Global Accelerator publishes data points to Amazon CloudWatch for your accelerators. CloudWatch enables you to retrieve statistics about those data points as an ordered set of time-series data, known as metrics. Think of a metric as a variable to monitor, and the data points as the values of that variable over time. For example, you can monitor traffic through an accelerator over a specified time period. Each data point has an associated time stamp and an optional unit of measurement.

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

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

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

For more information, see the Amazon CloudWatch User Guide.

**Contents**
- Global Accelerator metrics (p. 70)
- Metric dimensions for accelerators (p. 72)
- Statistics for Global Accelerator metrics (p. 73)
- View CloudWatch metrics for your accelerators (p. 74)

## Global Accelerator metrics

The AWS/GlobalAccelerator namespace includes the following metrics.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flows_Dropped_No_Endpoint</td>
<td>The total number of TCP IPv6 packet flows that were dropped because no IPv6 endpoints were available. This could happen, for example, if you had an accelerator with a dual-stack IP address type and you changed the IP address type to IPv4 for an Application Load Balancer endpoint.</td>
</tr>
</tbody>
</table>

*Reporting criteria:* Reported for accelerators with dual-stack IP address types that are receiving IPv6 traffic when one of the following occurs:

- An accelerator with IPv6 endpoints serving traffic reports a 0 metric
- An accelerator with misconfigured endpoints reports the total number of flows dropped

*Statistics:* The only useful statistic is Sum.

*Dimensions*:
- Accelerator
- Accelerator, Listener
- Accelerator, AcceleratorIPAddress

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HealthyEndpointCount</td>
<td>The total number of endpoints that are considered healthy. Global Accelerator regularly checks the status of endpoints on standard accelerators. These health checks run automatically. How and when these health checks run depend on the type of endpoint and the health check options for the endpoint. To learn more, see Changing health check options (p. 33).</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Reporting criteria:</strong></td>
<td>Reported for accelerators that are configured and enabled.</td>
</tr>
<tr>
<td><strong>Statistics:</strong></td>
<td>The most useful statistics are Minimum and Maximum.</td>
</tr>
</tbody>
</table>
| **Dimensions** | • Accelerator  
                • Accelerator, Listener  
                • Accelerator, Listener, EndpointGroup                                                                                                   |

**NewFlowCount**  
The total number of new TCP and UDP flows (or connections) established from clients to endpoints in the time period.

**Reporting criteria:** There is a nonzero value.

**Statistics:** The only useful statistic is Sum.

**Dimensions**  
• Accelerator  
• Accelerator, Listener  
• Accelerator, Listener, EndpointGroup  
• Accelerator, SourceRegion  
• Accelerator, DestinationEdge  
• Accelerator, TransportProtocol  
• Accelerator, AcceleratorIPAddress

**ProcessedBytesIn**  
The total number of incoming bytes processed by the accelerator, including TCP/IP headers. This count includes all traffic to endpoints.

**Reporting criteria:** There is a nonzero value.

**Statistics:** The only useful statistic is Sum.

**Dimensions**  
• Accelerator  
• Accelerator, Listener  
• Accelerator, Listener, EndpointGroup  
• Accelerator, SourceRegion  
• Accelerator, DestinationEdge  
• Accelerator, TransportProtocol  
• Accelerator, AcceleratorIPAddress
### Metric dimensions for accelerators

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

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
</table>
| ProcessedBytesOut       | The total number of outgoing bytes processed by the accelerator, including TCP/IP headers. This count includes traffic from endpoints, minus health check traffic.  
  Reporting criteria: There is a nonzero value.  
  Statistics: The only useful statistic is Sum.  
  Dimensions  
  • Accelerator  
  • Accelerator, Listener  
  • Accelerator, Listener, EndpointGroup  
  • Accelerator, SourceRegion  
  • Accelerator, DestinationEdge  
  • Accelerator, TransportProtocol  
  • Accelerator, AcceleratorIPAddress                                                                                             |
| UnhealthyEndpointCount  | The total number of endpoints that are considered unhealthy. Global Accelerator regularly checks the status of endpoints on standard accelerators. These health checks run automatically. How and when these health checks run depend on the type of endpoint and the health check options for the endpoint. To learn more, see Changing health check options (p. 33).  
  Reporting criteria: Reported for accelerators that are configured and enabled.  
  Statistics: The most useful statistics are Minimum and Maximum.  
  Dimensions  
  • Accelerator  
  • Accelerator, Listener  
  • Accelerator, Listener, EndpointGroup                                                                                           |

### Metric dimensions for accelerators

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

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerator</td>
<td>Filters the metric data by accelerator. Specify the accelerator by the accelerator id (the final portion of the accelerator ARN). For example, if the ARN is arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcd-fgh, you specify the following: 1234abcd-abcd-1234-abcd-1234abcd-fgh.</td>
</tr>
</tbody>
</table>
| Listener    | Filters the metric data by listener. Specify the listener by the listener id (the final portion of the listener ARN). For example, if the ARN is arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EndpointGroup</td>
<td>Filters the metric data by endpoint group. Specify the endpoint group by the AWS Region, for example, <code>us-east-1</code> (all lowercase).</td>
</tr>
<tr>
<td>SourceRegion</td>
<td>Filters the metric data by source region, which is the geographic area of the AWS Regions where your application endpoints are running. Source region is one of the following:</td>
</tr>
</tbody>
</table>
|                 | • NA – United States and Canada  
|                 | • EU – Europe  
|                 | • AP – Asia Pacific*  
|                 | • KR – South Korea  
|                 | • IN – India  
|                 | • AU – Australia  
|                 | • ME – Middle East  
|                 | • SA – South America  
|                 | • ZA – South Africa  
|                 | *Excluding South Korea and India  |
| DestinationEdge | Filters the metric data by destination edge, which is the geographic area of the AWS edge locations that serve your client traffic. Destination edge is one of the following:                                                                 |
|                 | • NA – United States and Canada  
|                 | • EU – Europe  
|                 | • AP – Asia Pacific*  
|                 | • KR – South Korea  
|                 | • IN – India  
|                 | • AU – Australia  
|                 | • ME – Middle East  
|                 | • SA – South America  
|                 | • ZA – South Africa  
|                 | *Excluding South Korea and India  |
| TransportProtocol | Filters the metric data by transport protocol: UDP or TCP.                                                                                                                                               |
| AcceleratorIPAddress | Filters the metric data by the IP address of the accelerator: that is, one of the static IP addresses assigned to an accelerator.                                                                 |

**Statistics for Global Accelerator metrics**

CloudWatch provides statistics based on the metric data points published by Global Accelerator. Statistics are aggregations of metric data over a specified period of time. When you request statistics, the returned data stream is identified by the metric name and dimension. A dimension is a name/value pair that uniquely identifies a metric. For example, you can request the processed bytes out for an accelerator where the bytes are served from AWS edge locations in Europe (destination edge is “EU”).
The following are examples of metric/dimension combinations that you might find useful:

- View the amount of traffic served (such as ProcessedBytesOut) by each of your two accelerator IP addresses to validate that your DNS configuration is correct.
- View the geographical distribution of your user traffic and monitor how much of it is local (for example, North America to North America) or global (for example, Australia or India to North America). To determine this, view the metrics ProcessedBytesIn or ProcessedBytesOut with the dimensions DestinationEdge and SourceRegion set to specific values.
- View the number of unhealthy endpoints across your accelerator, and determine which endpoint groups they belong to. If you have a large number of endpoint groups, this is especially useful to help you quickly find endpoint groups with endpoints that are experiencing issues. To determine this, view the metric UnhealthyEndpointCount with the dimensions Accelerator, Listener, and EndpointGroup.

**View CloudWatch metrics for your accelerators**

You can view the CloudWatch metrics for your accelerators using the CloudWatch console or the AWS CLI. In the console, metrics are displayed as monitoring graphs. The monitoring graphs show data points only if the accelerator is active and receiving requests.

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

**To view metrics using the CloudWatch console**

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

**To view metrics using the AWS CLI**

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

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

**To get the statistics for a metric using the AWS CLI**

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

The following example lists the total processed bytes in, per minute, for your accelerator serving from the North America (NA) destination edge.

```
aws cloudwatch get-metric-statistics --namespace AWS/GlobalAccelerator \
--metric-name ProcessedBytesIn \
--region us-west-2 \
--statistics Sum --period 60 \
--dimensions Name=Accelerator,Value=1234abcd-abcd-1234-abcd-1234abcdefgh \
Name=DestinationEdge,Value=NA \
--start-time 2019-12-18T20:00:00Z --end-time 2019-12-18T21:00:00Z
```

The following is example output from the command:

```
74
```
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

```json
{
    "Label": "ProcessedBytesIn",
    "Datapoints": [
        {
            "Timestamp": "2019-12-18T20:45:00Z",
            "Sum": 2410870.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:47:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:46:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:42:00Z",
            "Sum": 1560.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:48:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:43:00Z",
            "Sum": 1343.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:49:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "Timestamp": "2019-12-18T20:44:00Z",
            "Sum": 35791560.0,
            "Unit": "Bytes"
        }
    ]
}
```
source, destination, and protocol for an IP flow. The capture window is a duration of time during which
the flow logs service aggregates data before publishing flow log records. The capture window is up to 1
minute.

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

**Tip**
Using Amazon Athena and Amazon QuickSight with your Global Accelerator flow log data can
help you troubleshoot reachability issues for your application, identify security vulnerabilities,
and get an overview of how users access your application. To learn more, see the following AWS
blog post: Analyzing and visualizing AWS Global Accelerator flow logs using Amazon Athena
and Amazon QuickSight.

**Topics**
- Publishing flow logs to Amazon S3 (p. 76)
- Timing of log file delivery (p. 79)
- Flow log record syntax (p. 80)

## Publishing flow logs to Amazon S3

Flow logs for AWS Global Accelerator are published to Amazon S3 to an existing S3 bucket that you
specify. Flow log records are published to a series of log file objects that are stored in the bucket.

To create an Amazon S3 bucket for use with flow logs, see Create your first S3 bucket in the Amazon
Simple Storage Service User Guide.

### Flow logs files

Flow logs collect flow log records, consolidate them into log files, and then publish the log files to the
Amazon S3 bucket at 5-minute intervals. 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:
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-1234abcdefg, on November 23, 2018 at 00:05 UTC:

```
```

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

**IAM roles for publishing flow logs to Amazon S3**

An IAM principal, such as an IAM role or 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:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite",
      "Effect": "Allow",
      "Principal": {"Service": "delivery.logs.amazonaws.com"},
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucket_name/optional_folder/AWSLogs/account_id/*",
      "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-control"}}
    },
    {
      "Sid": "AWSLogDeliveryAclCheck",
      "Effect": "Allow",
      "Principal": {"Service": "delivery.logs.amazonaws.com"},
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws:s3:::bucket_name"
    }
  ]
}
```

If the user creating the flow log does not own the bucket, or does not have the GetBucketPolicy and PutBucketPolicy permissions for the bucket, the flow log creation fails. In this case, the bucket owner must manually add the preceding policy to the bucket and specify the flow log creator's AWS account ID. For more information, see Adding a bucket policy by using the Amazon S3 console in the Amazon Simple Storage Service User Guide. If the bucket receives flow logs from multiple accounts, add a Resource element entry to the AWSLogDeliveryWrite policy statement for each account.

For example, the following bucket policy allows AWS accounts 123123123123 and 456456456456 to publish flow logs to a folder named flow-logs in a bucket named log-bucket:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite",
      "Effect": "Allow",
      "Principal": {"Service": "delivery.logs.amazonaws.com"},
      "Action": "s3:PutObject",
      "Resource": [
        "arn:aws:s3:::log-bucket/flow-logs/AWSLogs/123123123123/*",
        "arn:aws:s3:::log-bucket/flow-logs/AWSLogs/456456456456/*"
      ],
      "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-control"}}
    },
    {
      "Sid": "AWSLogDeliveryAclCheck",
      "Effect": "Allow",
      "Principal": {"Service": "delivery.logs.amazonaws.com"},
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws:s3:::log-bucket"
    }
  ]
}
```

**Note**

We recommend that you grant the AWSLogDeliveryAclCheck and AWSLogDeliveryWrite permissions to the log delivery service principal instead of individual AWS account ARNs.
Required CMK key policy for use with SSE-KMS buckets

If you enabled server-side encryption for your Amazon S3 bucket using AWS KMS-managed keys (SSE-KMS) with a customer-managed CMK, you must add the following to the key policy for your CMK so that flow logs can write log files to the bucket:

```json
{
    "Sid": "Allow AWS Global Accelerator Flow Logs to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "delivery.logs.amazonaws.com"
        ]
    },
    "Action": "kms:GenerateDataKey*",
    "Resource": "*"
}
```

Amazon S3 log file permissions

In addition to the required bucket policies, Amazon S3 uses access control lists (ACLs) to manage access to the log files created by a flow log. By default, the bucket owner has FULL_CONTROL permissions on each log file. The log delivery owner, if different from the bucket owner, has no permissions. The log delivery account has READ and WRITE permissions. For more information, see Access control list (ACL) overview in the Amazon Simple Storage Service User 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. 77).
3. Run the following AWS CLI command, with the Amazon S3 bucket name and prefix that you want to use for your log files:

```bash
aws globalaccelerator update-accelerator-attributes
   --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234abcddefgh
   --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
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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>The flow logs version.</td>
</tr>
<tr>
<td>aws_account_id</td>
<td>The AWS account ID for the flow log.</td>
</tr>
<tr>
<td>accelerator_id</td>
<td>The ID of the accelerator for which the traffic is recorded.</td>
</tr>
<tr>
<td>client_ip</td>
<td>The source IPv4 or IPv6 address.</td>
</tr>
<tr>
<td>client_port</td>
<td>The source port.</td>
</tr>
<tr>
<td>accelerator_ip</td>
<td>The accelerator's IP address.</td>
</tr>
<tr>
<td>accelerator_port</td>
<td>The accelerator's port.</td>
</tr>
<tr>
<td>endpoint_ip</td>
<td>The destination IP address of the traffic.</td>
</tr>
<tr>
<td>endpoint_port</td>
<td>The destination port of the traffic.</td>
</tr>
<tr>
<td>protocol</td>
<td>The IANA protocol number of the traffic. For more information, see <a href="#">Assigned Internet Protocol Numbers</a>.</td>
</tr>
<tr>
<td>ip_address_type</td>
<td>IPv4 or IPv6.</td>
</tr>
<tr>
<td>packets</td>
<td>The number of packets transferred during the capture window.</td>
</tr>
</tbody>
</table>
### AWS Global Accelerator Developer Guide

**CloudTrail logging**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bytes</td>
<td>The number of bytes transferred during the capture window.</td>
</tr>
<tr>
<td>start_time</td>
<td>The time, in Unix seconds, of the start of the capture window.</td>
</tr>
<tr>
<td>end_time</td>
<td>The time, in Unix seconds, of the end of the capture window.</td>
</tr>
</tbody>
</table>
| action            | The action associated with the traffic:  
  - ACCEPT: The recorded traffic was permitted by the security groups or network ACLs. The value is currently always ACCEPT. |
| 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_source_ip | The IP address used by the Global Accelerator network interface. If client IP address preservation is enabled, this value is set to - (hyphen).  
For more information, see [Preserve client IP addresses in AWS Global Accelerator (p. 64)]. |
| globalaccelerator_source_port | The port used by the Global Accelerator network interface. If client IP address preservation is enabled, this value is set to 0 (zero).  
For more information, see [Preserve client IP addresses in AWS Global Accelerator (p. 64)]. |
| endpoint_region   | The AWS Region where the endpoint is located. |
| globalaccelerator_region | The edge location (point of presence) that served the request. Each edge location has a three-letter code and an arbitrarily assigned number, for example, DFW3. The three-letter code typically corresponds with the International Air Transport Association airport code for an airport near the edge location. (These abbreviations might change in the future.) |
| direction         | The direction of the traffic. Denotes traffic coming into the Global Accelerator network (INGRESS) or returning to the client (EGRESS). |
| vpc_id            | The VPC identifier. Included with Version 2.0 flow logs when Global Accelerator sends traffic to an endpoint with client IP address preservation. |

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

## Using AWS CloudTrail to log AWS Global Accelerator API calls

AWS Global Accelerator is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Global Accelerator. CloudTrail captures all API calls for Global Accelerator as events, including calls from the Global Accelerator console and from code calls to the
Global Accelerator API. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Global Accelerator. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in Event history.

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

Global Accelerator information in CloudTrail

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

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

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

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

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

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

For more information, see the CloudTrail userIdentity Element.

Viewing Global Accelerator events in event history

CloudTrail lets you view recent events in Event history. To view events for Global Accelerator API requests, you must choose US West (Oregon) in the Region selector at the top of the console. For more information, see Viewing events with CloudTrail event history in the AWS CloudTrail User Guide.

Understanding Global Accelerator log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. Each JSON-formatted CloudTrail log file contains one or more log entries. A log entry represents a single request from any source and includes information about the requested action, including any parameters, the date and time of the action, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.
The following example shows a CloudTrail log entry that includes the following Global Accelerator actions:

- Listing the accelerators for an account: `eventName` is `ListAccelerators`.
- Creating a listener: `eventName` is `CreateListener`.
- Updating a listener: `eventName` is `UpdateListener`.
- Describing a listener: `eventName` is `DescribeListener`.
- Listing the listeners for an account: `eventName` is `ListListeners`.
- Deleting a listener: `eventName` is `DeleteListener`.

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/smithj",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-11-17T21:02:36Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "userName": "smithj"
          }
        },
        "eventTime": "2018-11-17T21:03:14Z",
        "eventSource": "globalaccelerator.amazonaws.com",
        "eventName": "ListAccelerators",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "192.0.2.50",
        "userAgent": "aws-cli/1.16.34 Python/2.7.10 Darwin/16.7.0 botocore/1.12.24",
        "requestParameters": null,
        "responseElements": null,
        "requestID": "083cae81-28ab-4a66-862f-096e1example",
        "eventID": "fe8b1c13-8757-4c73-b042-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": "2018-11-17T21:03:14Z",
        "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-b042-fe2a3example",
        "eventType": "AwsApiCall",
        "recipientAccountId": "111122223333"
      }
    }
  ]
}
```
Understanding Global Accelerator log file entries

```
{
  "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"
    }
  }
}
```

```json
{
  "type": "Role",
  "principalId": "A1B2C3D4E5F6G7EXAMPLE",
  "arn": "arn:aws:iam::111122223333:user/smithj",
  "accountId": "111122223333",
  "userName": "smithj"
}"
```

```json
"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-7d7fexample",
"eventID": "9cab44ef-0777-41e6-838f-f249example",
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}
```
"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",}
Understanding Global Accelerator log file entries

```
"requestParameters": {
  "listenerArn":
  "arn:aws:globalaccelerator::111122223333:accelerator/0339bfd6-13bc-4d45-a114-5d7fexample/
  listener/abcde1234"
},
"responseElements": null,
"requestID": "9980e568-82fa-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"
      }
    }
  }
}
```

```
"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"
      }
    }
  }
}
```

87
"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": "afedb874-2e21-44da-b1b0-2ddb2example",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
]}
}
Security in AWS Global Accelerator

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

• **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to AWS Global Accelerator, see AWS Services in Scope by Compliance Program.

• **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company’s requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Global Accelerator. The following topics show you how to configure Global Accelerator to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Global Accelerator resources.

**Topics**

• Identity and Access Management for AWS Global Accelerator (p. 89)
• Secure VPC connections in AWS Global Accelerator (p. 110)
• Logging and monitoring in AWS Global Accelerator (p. 110)
• Compliance validation for AWS Global Accelerator (p. 111)
• Resilience in AWS Global Accelerator (p. 111)
• Infrastructure security in AWS Global Accelerator (p. 112)

Identity and Access Management for AWS Global Accelerator

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be authenticated (signed in) and authorized (have permissions) to use Global Accelerator resources. IAM is an AWS service that you can use with no additional charge.

**Contents**

• Audience (p. 90)
• Authenticating with identities (p. 90)
Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Global Accelerator.

**Service user** – If you use the Global Accelerator service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Global Accelerator features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Global Accelerator, see Troubleshooting AWS Global Accelerator identity and access (p. 108).

**Service administrator** – If you’re in charge of Global Accelerator resources at your company, you probably have full access to Global Accelerator. It’s your job to determine which Global Accelerator features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Global Accelerator, see How AWS Global Accelerator works with IAM (p. 94).

**IAM administrator** – If you’re an IAM administrator, you might want to learn details about how you can write policies to manage access to Global Accelerator. To view example Global Accelerator identity-based policies that you can use in IAM, see Identity-based policy examples for AWS Global Accelerator (p. 99).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company’s single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the AWS Sign-In User Guide.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don’t use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signing AWS API requests in the IAM User Guide.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA)
to increase the security of your account. To learn more, see Multi-factor authentication in the AWS IAM Identity Center User Guide and Using multi-factor authentication (MFA) in AWS in the IAM User Guide.

**AWS account root user**

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account **root user** and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the IAM User Guide.

**Federated identity**

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials. A **federated identity** is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the AWS IAM Identity Center User Guide.

**IAM users and groups**

An **IAM user** is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the IAM User Guide.

An **IAM group** is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named **IAMAdmins** and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the IAM User Guide.

**IAM roles**

An **IAM role** is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the IAM User Guide.
IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see [Creating a role for a third-party Identity Provider](https://docs.aws.amazon.com/IAM/latest/userguide/iam-create-role-for-third-party-idp.html) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see [Permission sets](https://docs.aws.amazon.com/IAM/latest/userguide/idc-permission-sets.html) in the *AWS IAM Identity Center User Guide*.

- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.

- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](https://docs.aws.amazon.com/IAM/latest/userguide/iam-resource-based-policies.html) in the *IAM User Guide*.

- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it’s common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal’s permissions, using a service role, or using a service-linked role.

- **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see [Actions, resources, and condition keys for AWS Global Accelerator](https://docs.aws.amazon.com/IAM/latest/userguide/sigv4-action-keys.html) in the *Service Authorization Reference*.

- **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Creating a role to delegate permissions to an AWS service](https://docs.aws.amazon.com/IAM/latest/userguide/iam-create-service-role.html) in the *IAM User Guide*.

- **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using an IAM role to grant permissions to applications running on Amazon EC2 instances](https://docs.aws.amazon.com/IAM/latest/userguide/iam-role-to-instance.html) in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see [When to create an IAM role (instead of a user)](https://docs.aws.amazon.com/IAM/latest/userguide/idc-when-to-use-iam-role.html) in the *IAM User Guide*.

### Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see [Overview of JSON policies](https://docs.aws.amazon.com/IAM/latest/userguide/creating-policies-json.html) in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which principal can perform actions on what resources, and under what conditions.
By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the IAM User Guide.

Identity-based policies can be further categorized as inline policies or managed policies. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the IAM User Guide.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM role trust policies and Amazon S3 bucket policies. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can’t use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the Amazon Simple Storage Service Developer Guide.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity’s identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the Principal field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the IAM User Guide.

- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all
features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the AWS Organizations User Guide.

- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the IAM User Guide.

### Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the IAM User Guide.

### How AWS Global Accelerator works with IAM

Before you use IAM to manage access to Global Accelerator, learn what IAM features are available to use with Global Accelerator.

To see tables showing a similar high-level view of how AWS services work with most IAM features, see AWS services that work with IAM in the IAM User Guide.

### IAM features you can use with AWS Global Accelerator

<table>
<thead>
<tr>
<th>IAM feature</th>
<th>Global Accelerator support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity-based policies (p. 94)</td>
<td>Yes</td>
</tr>
<tr>
<td>Resource-based policies (p. 95)</td>
<td>No</td>
</tr>
<tr>
<td>Policy actions (p. 95)</td>
<td>Yes</td>
</tr>
<tr>
<td>Policy resources (p. 96)</td>
<td>Yes</td>
</tr>
<tr>
<td>Policy condition keys (service-specific) (p. 96)</td>
<td>Yes</td>
</tr>
<tr>
<td>ACLs (p. 97)</td>
<td>Yes</td>
</tr>
<tr>
<td>ABAC (tags in policies) (p. 97)</td>
<td>Partial</td>
</tr>
<tr>
<td>Temporary credentials (p. 98)</td>
<td>Yes</td>
</tr>
<tr>
<td>Principal permissions (p. 98)</td>
<td>Yes</td>
</tr>
<tr>
<td>Service roles (p. 98)</td>
<td>No</td>
</tr>
<tr>
<td>Service-linked roles (p. 98)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Identity-based policies for Global Accelerator

| Supports identity-based policies                  | Yes                        |
Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the IAM User Guide.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the IAM User Guide.

To view examples of Global Accelerator identity-based policies, see Identity-based policy examples for AWS Global Accelerator (p. 99).

Resource-based policies within Global Accelerator

| Supports resource-based policies | No |

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM role trust policies and Amazon S3 bucket policies. In services that support resource-based policies, service administrators can use them to control access to a specific resource.

Policy actions for Global Accelerator

| Supports policy actions | Yes |

Administrators can use AWS JSON policies to specify who has access to what. That is, which principal can perform actions on what resources, and under what conditions.

The Action element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as permission-only actions that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called dependent actions.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Global Accelerator actions, see Actions defined by AWS Global Accelerator in the Service Authorization Reference.

Policy actions in Global Accelerator use the following prefix before the action:

aws-globalaccelerator

To specify multiple actions in a single statement, separate them with commas.

"Action": [
   "aws-globalaccelerator:action1",
   "aws-globalaccelerator:action2"
]
You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word Describe, include the following action:

```
"Action": "aws-globalaccelerator:Describe*"
```

To view examples of Global Accelerator identity-based policies, see [Identity-based policy examples for AWS Global Accelerator](p. 99).

### Policy resources for Global Accelerator

<table>
<thead>
<tr>
<th>Supports policy resources</th>
<th>Yes</th>
</tr>
</thead>
</table>

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The **Resource** JSON policy element specifies the object or objects to which the action applies. Statements must include either a Resource or a NotResource element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as **resource-level permissions**.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

In the *Service Authorization Reference*, you can see the following information related to Global Accelerator:

- To see a list of Global Accelerator resource types and their ARNs, see [Resources defined by AWS Global Accelerator](p. 99).
- To learn the actions that you can specify with the ARN of each resource, see [Actions defined by AWS Global Accelerator](p. 99).

To view examples of Global Accelerator identity-based policies, see [Identity-based policy examples for AWS Global Accelerator](p. 99).

### Policy condition keys for Global Accelerator

<table>
<thead>
<tr>
<th>Supports service-specific policy condition keys</th>
<th>Yes</th>
</tr>
</thead>
</table>

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The **Condition** element (or **Condition block**) lets you specify conditions in which a statement is in effect. The **Condition** element is optional. You can create conditional expressions that use **condition operators**, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple **Condition** elements in a statement, or multiple keys in a single **Condition** element, AWS evaluates them using a logical **AND** operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical **OR** operation. All of the conditions must be met before the statement's permissions are granted.
You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the IAM User Guide.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the IAM User Guide.

To see a list of Global Accelerator condition keys, see Condition keys for AWS Global Accelerator in the Service Authorization Reference. To learn with which actions and resources you can use a condition key, see Actions defined by AWS Global Accelerator.

To view examples of Global Accelerator identity-based policies, see Identity-based policy examples for AWS Global Accelerator (p. 99).

**ACLs in Global Accelerator**

<table>
<thead>
<tr>
<th>Supports ACLs</th>
<th>Yes</th>
</tr>
</thead>
</table>

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

**ABAC with Global Accelerator**

<table>
<thead>
<tr>
<th>Supports ABAC (tags in policies)</th>
<th>Partial</th>
</tr>
</thead>
</table>

Global Accelerator has partial support for tags in policies. It supports tagging for one resource, accelerators. For more information about using tags in policy statement conditions, and to view an example policy for limiting access to a resource based on tags on the resource, see Using tag-based policies with AWS Global Accelerator (p. 107).

For more information about tagging Global Accelerator resources, see Tagging in AWS Global Accelerator (p. 11).

To learn more about using tags in policies, review the following information.

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called tags. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal’s tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the aws:ResourceTag/key-name, aws:RequestTag/key-name, or aws:TagKeys condition keys.

If a service supports all three condition keys for every resource type, then the value is Yes for the service. If a service supports all three condition keys for only some resource types, then the value is Partial.

For more information about ABAC, see What is ABAC? in the IAM User Guide. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the IAM User Guide.
Using temporary credentials with Global Accelerator

| Supports temporary credentials | Yes |

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the IAM User Guide.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switching to a role (console) in the IAM User Guide.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Cross-service principal permissions for Global Accelerator

| Supports principal permissions | Yes |

When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, resources, and condition keys for AWS Global Accelerator in the Service Authorization Reference.

Service roles for Global Accelerator

| Supports service roles | No |

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the IAM User Guide.

Service-linked role for Global Accelerator

| Supports service-linked roles | Yes |

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For more information about the service-linked role for Global Accelerator, see Service-linked role for AWS Global Accelerator (p. 102).

For details about creating or managing service-linked roles in general in AWS, see AWS services that work with IAM. Find a service in the table that includes a Yes in the Service-linked role column. Choose the Yes link to view the service-linked role documentation for that service.
Identity-based policy examples for AWS Global Accelerator

By default, users and roles don’t have permission to create or modify Global Accelerator resources. They also can’t perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Creating IAM policies in the IAM User Guide.

For details about actions and resource types defined by Global Accelerator, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS Global Accelerator in the Service Authorization Reference.

Topics
- Policy best practices (p. 99)
- Creating a Global Accelerator accelerator (p. 100)
- Using the Global Accelerator console (p. 100)
- Using a Global Accelerator API action (p. 101)
- Allow users to view their own permissions (p. 101)

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Global Accelerator resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the AWS managed policies that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the IAM User Guide.

- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as least-privilege permissions. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the IAM User Guide.

- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the IAM User Guide.

- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation in the IAM User Guide.

- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are
called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the IAM User Guide.

For more information about best practices in IAM, see Security best practices in IAM in the IAM User Guide.

Creating a Global Accelerator accelerator

To create an AWS Global Accelerator accelerator, users must have permission to create service-linked roles that are associated with Global Accelerator.

To ensure that users have the correct permissions to create accelerators in Global Accelerator, attach a policy to the user such as the following.

**Note**

If you create an identity-based permissions policy that is more restrictive than the following policy, users with the more restrictive policy won't be able to create an accelerator.

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

Using the Global Accelerator console

To access the AWS Global Accelerator console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Global Accelerator resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the Global Accelerator console, also attach the Global Accelerator GlobalAcceleratorReadOnlyAccess or GlobalAcceleratorFullAccess AWS managed policy to the entities.

Attach the first policy, GlobalAcceleratorReadOnlyAccess, if users only need to view information in the console or make calls to the AWS Command Line Interface or the API that use List* or Describe* operations.

Attach the second policy, GlobalAcceleratorFullAccess, to users who need to create or make updates to accelerators. The full access policy includes full permissions for Global Accelerator as well as describe permissions for Amazon EC2 and Elastic Load Balancing.
Note
If you create an identity-based permissions policy that does not include the required permissions for Amazon EC2 and Elastic Load Balancing, users with that policy will not be able to add Amazon EC2 and Elastic Load Balancing resources to accelerators.

For more information, see the Global Accelerator AWS managed policies page (p. 104) or Adding permissions to a user in the IAM User Guide.

Using a Global Accelerator API action
AWS Global Accelerator supports using actions in a policy. This allows an administrator to control whether an entity can complete an operation in Global Accelerator.

For example, the following policy allows a user to perform the CreateAccelerator operation to programmatically create an accelerator in an AWS account:

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

Allow users to view their own permissions
This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "ViewOwnUserInfo",
         "Effect": "Allow",
         "Action": [
            "iam:GetUserPolicy",
            "iam:ListGroupsForUser",
            "iam:ListAttachedUserPolicies",
            "iam:ListUserPolicies",
            "iam:GetUser"
         ],
         "Resource": ["arn:aws:iam::*:user/${aws:username}"],
         "Resource": [
            "arn:aws:iam::*:user/${aws:username}"
         ],
      },
      {
         "Sid": "NavigateInConsole",
         "Effect": "Allow",
         "Action": [
            "iam:GetGroupPolicy",
            "iam:GetPolicyVersion",
            "iam:GetPolicy",
            "iam:ListAttachedGroupPolicies",
            "iam:ListGroupPolicies",
            "iam:ListGroupVersions",
            "iam:ListPolicies",
            "iam:ListUsers"
         ]
      }
   ]
}
```
Service-linked role for AWS Global Accelerator

AWS Global Accelerator uses an AWS Identity and Access Management (IAM) service-linked role. A service-linked role is a unique type of IAM role that is linked directly to Global Accelerator. The service-linked role is predefined by Global Accelerator and includes all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Global Accelerator easier because you don't have to manually add the necessary permissions. Global Accelerator defines the permissions of its service-linked role, and unless defined otherwise, only Global Accelerator can assume its role. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting its related resources. This protects your Global Accelerator resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS services that work with IAM and look for the services that have Yes in the Service-linked role column. Choose a Yes with a link to view the service-linked role documentation for that service.

Service-linked role permissions for Global Accelerator

AWS Global Accelerator uses a service-linked role named AWSServiceRoleForGlobalAccelerator. This role allows Global Accelerator to access resources in your account, such as load balancers and other endpoints, to help make sure, for example, that you can add only resources that are configured to work with Global Accelerator. The AWSServiceRoleForGlobalAccelerator role also allows Global Accelerator to create and manage resources necessary for client IP address preservation.

Global Accelerator automatically creates a role named AWSServiceRoleForGlobalAccelerator when the role is first required to support a Global Accelerator API operation. This role is required for using accelerators in Global Accelerator. The ARN for the AWSServiceRoleForGlobalAccelerator role looks like this:

```
arn:aws:iam::123456789012:role/aws-service-role/
globalaccelerator.amazonaws.com/AWSServiceRoleForGlobalAccelerator
```

Service-linked role permissions

Global Accelerator uses the service-linked role named AWSServiceRoleForGlobalAccelerator to access resources and configurations to check readiness. This service-linked role uses the managed policy AWSGlobalAcceleratorSLRPolicy.

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

* globalaccelerator.amazonaws.com

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

```
{
  "Version": "2012-10-17",
  "Statement": [
```
You must configure permissions to allow an IAM entity (such as a user, group, or role) to delete the Global Accelerator service-linked role. For more information, see Service-Linked Role Permissions in the IAM User Guide.

Creating the service-linked role for Global Accelerator

You don't manually create the service-linked role for Global Accelerator. The service creates the role for you automatically the first time that you create an accelerator. If you remove your Global Accelerator...
resources and delete the service-linked role, the service creates the role again automatically when you create a new accelerator.

**Editing the Global Accelerator service-linked role**

Global Accelerator does not allow you to edit the AWSServiceRoleForGlobalAccelerator service-linked role. After the service has created a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of a role by using IAM. For more information, see [Editing a Service-Linked Role](#) in the IAM User Guide.

**Deleting the Global Accelerator service-linked role**

If you no longer need to use Global Accelerator, we recommend that you delete the service-linked role. That way you don’t have unused entities that are not actively monitored or maintained. However, you must clean up the Global Accelerator resources in your account before you can manually delete the roles.

After you have disabled and deleted your accelerators, then you can delete the service-linked role. For more information about deleting accelerators, see [Creating or updating a standard accelerator (p. 24)](#).

**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/](https://console.aws.amazon.com/iam/).
2. In the navigation pane of the IAM console, choose Roles. Then select the check box next to the role name that you want to delete, not the name or row itself.
3. For Role actions at the top of the page, choose Delete role.
4. In the confirmation dialog box, review the service last accessed data, which shows when each of the selected roles last accessed an AWS service. This helps you to confirm whether the role is currently active. If you want to proceed, choose Yes, Delete to submit the service-linked role for deletion.
5. Watch the IAM console notifications to monitor the progress of the service-linked role deletion. Because the IAM service-linked role deletion is asynchronous, after you submit the role for deletion, the deletion task can succeed or fail. For more information, see [Deleting a service-linked role](#) in the IAM User Guide.

**Updates to the policy for the Global Accelerator service-linked role**

For updates to AWSSGlobalAcceleratorSLRPolicy, the AWS managed policy for the Global Accelerator service-linked role, see the AWS managed policies updates table (p. 106). You can also subscribe to automatic RSS alerts on the AWS Global Accelerator Document history (p. 117) page.

**AWS managed policies for AWS Global Accelerator**

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they’re available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.
You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the IAM User Guide.

**AWS managed policy: AWSServiceRoleForGlobalAccelerator**

You can't attach AWSServiceRoleForGlobalAccelerator to your IAM entities. This policy is attached to a service-linked role that allows AWS Global Accelerator to access AWS services and resources that are used or managed by Global Accelerator. For more information, see Service-linked role for AWS Global Accelerator (p. 102).

**AWS managed policy: GlobalAcceleratorFullAccess**

You can attach GlobalAcceleratorFullAccess to your IAM entities. This policy grants full access to actions for working with accelerators in Global Accelerator. Attach it to IAM users and other principals who need full access to Global Accelerator actions.

**Note**

If you create an identity-based permissions policy that does not include the required permissions for Amazon EC2 and Elastic Load Balancing, users with that policy will not be able to add Amazon EC2 and Elastic Load Balancing resources to accelerators.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "globalaccelerator:*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "elasticloadbalancing:DescribeLoadBalancers",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeInstances",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeSubnets",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:DeleteNetworkInterface"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteSecurityGroup",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/AWSServiceName": "GlobalAccelerator"
                }
            }
        }
    ]
}
```
AWS managed policy: GlobalAcceleratorReadOnlyAccess

You can attach GlobalAcceleratorReadOnlyAccess to your IAM entities. This policy grants read-only access to actions for working with accelerators in Global Accelerator. It's useful for users who only need to view information in the console or make calls to the AWS Command Line Interface or the API that use List* or Describe* operations.

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

Global Accelerator updates to AWS managed policies

View details about updates to AWS managed policies for Global Accelerator since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Global Accelerator Document history page (p. 117).

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSGlobalAcceleratorSLRPolicy</td>
<td>Global Accelerator added a new permission to describe target groups on load balancers.</td>
<td>October 20, 2023</td>
</tr>
<tr>
<td>– Updated policy</td>
<td>Global Accelerator uses elasticloadbalancing:DescribeTargetGroups to identify load balancers with target type ip, which is not a supported target type for dual-</td>
<td></td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>stack load balancer endpoints in Global Accelerator.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AWSGlobalAcceleratorSLRPolicy</strong> - Updated policy</td>
<td>Global Accelerator added new permissions to describe listeners on load balancers and describe addresses on EC2 instances.</td>
<td>May 23, 2023</td>
</tr>
<tr>
<td></td>
<td>Global Accelerator uses <code>elasticloadbalancing:DescribeListeners</code> to support making listener management decisions for load balancers, based on listener configurations.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Global Accelerator uses <code>ec2:DescribeAddresses</code> to add Elastic IP address endpoints to accelerators.</td>
<td></td>
</tr>
<tr>
<td><strong>AWSGlobalAcceleratorSLRPolicy</strong> - Updated policy</td>
<td>Global Accelerator added new permissions to support IPv6 addresses.</td>
<td>November 15, 2021</td>
</tr>
<tr>
<td></td>
<td>Global Accelerator uses <code>ec2:AssignIpv6Addresses</code> to update the Global Accelerator ENI on a customer subnet with an IPv6 address for sending and receiving IPv6 traffic, and uses <code>UnassignIpv6Addresses</code> to remove the IPv6 address when it's no longer needed.</td>
<td></td>
</tr>
<tr>
<td><strong>AWSGlobalAcceleratorSLRPolicy</strong> - Updated policy</td>
<td>Global Accelerator added a new permission to help Global Accelerator to diagnose errors.</td>
<td>May 18, 2021</td>
</tr>
<tr>
<td></td>
<td>Global Accelerator uses <code>ec2:DescribeRegions</code> to determine the AWS Region that a customer is in, which can help Global Accelerator to troubleshoot errors.</td>
<td></td>
</tr>
<tr>
<td>Global Accelerator started tracking changes</td>
<td>Global Accelerator started tracking changes for its AWS managed policies.</td>
<td>May 18, 2021</td>
</tr>
</tbody>
</table>

### Using tag-based policies with AWS Global Accelerator

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

Using tags in conditions is one way to control access to resources and requests. Tags can be attached to a resource or passed in the request to services that support tagging. In Global Accelerator, only accelerators can include tags. For more information about tagging in Global Accelerator, see Tagging in AWS Global Accelerator (p. 11).

When you create an IAM policy, you can use tag condition keys to control:

- Which users can perform actions on an accelerator, based on tags that it already has.
- What tags can be passed in an action's request.
- Whether specific tag keys can be used in a request.

For example, the AWS GlobalAcceleratorFullAccess managed user policy gives users unlimited permission to perform any Global Accelerator action on any resource. The following policy limits this power and denies unauthorized users permission to perform any Global Accelerator action on any production accelerators. A customer's administrator must attach this IAM policy to unauthorized IAM users, in addition to the managed user policy.

```json
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:RequestTag/stage":"prod"
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:ResourceTag/stage":"prod"
        }
      }
    }
  ]
}
```

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

**Troubleshooting AWS Global Accelerator identity and access**

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

**Topics**

- I am not authorized to perform an action in Global Accelerator (p. 109)
• **I am not authorized to perform iam:PassRole** *(p. 109)*

• **I want to allow people outside of my AWS account to access my Global Accelerator resources** *(p. 109)*

---

### I am not authorized to perform an action in Global Accelerator

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the mateojackson IAM user tries to use the console to view details about a fictional `my-example-widget` resource but doesn't have the fictional `aws-globalaccelerator:GetWidget` permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: aws-globalaccelerator:GetWidget on resource: my-example-widget
```

In this case, the policy for the mateojackson user must be updated to allow access to the `my-example-widget` resource by using the `aws-globalaccelerator:GetWidget` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

### I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Global Accelerator.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named marymajor tries to use the console to perform an action in Global Accelerator. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

### I want to allow people outside of my AWS account to access my Global Accelerator resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

• To learn whether Global Accelerator supports these features, see [How AWS Global Accelerator works with IAM](p. 94).

• To learn how to provide access to your resources across AWS accounts that you own, see [Providing access to an IAM user in another AWS account that you own](in the `IAM User Guide`).

---

109
Secure VPC connections in AWS Global Accelerator

When you add an internal Application Load Balancer or an Amazon EC2 instance endpoint in AWS Global Accelerator, you enable internet traffic to flow directly to and from the endpoint in Virtual Private Clouds (VPCs) by targeting it in a private subnet. The VPC that contains the load balancer or EC2 instance must have an internet gateway attached to it, to indicate that the VPC accepts internet traffic. However, you don't need public IP addresses on the load balancer or EC2 instance. You also don't need an associated internet gateway route for the subnet.

This is different from the typical internet gateway use case in which both public IP addresses and internet gateway routes are required for internet traffic to flow to instances or load balancers in a VPC. Even if the elastic network interfaces of your targets are present in a public subnet (that is, a subnet with an internet gateway route), when you use Global Accelerator for internet traffic, Global Accelerator overrides the typical internet route and all logical connections that arrive through the Global Accelerator also return through Global Accelerator rather than through the internet gateway.

Note
Using public IP addresses and using a public subnet for your Amazon EC2 instances are not typical, though it's possible to set up your configuration with them. Security groups apply to any traffic that arrives to your instances, including traffic from Global Accelerator and any public or Elastic IP address that is assigned to your instance ENI. Use private subnets to ensure that traffic is delivered only by Global Accelerator.

Keep this information in mind when considering network perimeter issues and configuring IAM privileges related to internet access management. For more information about controlling internet access to your VPC, see this service control policy example.

Logging and monitoring in AWS Global Accelerator

Monitoring is an important part of maintaining the availability and performance of Global Accelerator and your AWS solutions. You should collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your Global Accelerator resources and activity, and responding to potential incidents:

Global Accelerator provides the following three main avenues for logging and tracking:

Amazon CloudWatch metrics and alarms

Using CloudWatch, you can monitor, in real time, your AWS resources and the applications that you run on AWS. As soon as your accelerator is deployed, CloudWatch begins collect and tracking metrics for Global Accelerator. Metrics are variables that you can view for confirmation that traffic is flowing, or that you can measure over time.

You can use metrics, for example, to verify that traffic is flowing through Global Accelerator to your endpoints, and back out to clients, and to help troubleshoot issues. You can also create alarms that watch specific metrics, and then send notifications or automatically make changes to the resources you are monitoring when the metric exceeds a certain threshold for a period of time.
Compliance validation for AWS Global Accelerator

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- **Security and Compliance Quick Start Guides** – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.

- **Architecting for HIPAA Security and Compliance on Amazon Web Services** – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

  **Note**  
  Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.

- **AWS Compliance Resources** – This collection of workbooks and guides might apply to your industry and location.

- **Evaluating Resources with Rules** in the AWS Config Developer Guide – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.

- **AWS Security Hub** – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.

- **AWS Audit Manager** – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in AWS Global Accelerator

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate
Infrastructure security in AWS Global Accelerator

As a managed service, AWS Global Accelerator is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

You use AWS published API calls to access Global Accelerator through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.
Quotas for AWS Global Accelerator

Your AWS account has specific quotas, also known as limits, related to AWS Global Accelerator.

The Service Quotas console provides information about Global Accelerator quotas. Along with viewing the default quotas, you can use the Service Quotas console to request quota increases for adjustable quotas. Note that you must be in US East (N. Virginia) when you request quota increases for Global Accelerator.

Topics
- General quotas (p. 113)
- Quotas for endpoints per endpoint group (p. 114)
- Related quotas (p. 114)

General quotas

The following are overall quotas for Global Accelerator.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard accelerators per AWS account</td>
<td>20</td>
</tr>
<tr>
<td>You can request a quota increase.</td>
<td></td>
</tr>
<tr>
<td>Custom routing accelerators per AWS account</td>
<td>10</td>
</tr>
<tr>
<td>You can request a quota increase.</td>
<td></td>
</tr>
<tr>
<td>Listeners per accelerator</td>
<td>10</td>
</tr>
<tr>
<td>You can request a quota increase.</td>
<td></td>
</tr>
<tr>
<td>Endpoint groups per accelerator, across all listeners</td>
<td>42</td>
</tr>
<tr>
<td>AWS Regions that Global Accelerator can point to, across all listeners and endpoint groups</td>
<td>42</td>
</tr>
<tr>
<td>If your accelerator has one listener, you can point to all Global Accelerator supported Regions with your accelerator's endpoint group configuration. Note that the maximum number of Regions that you can reference in an accelerator using endpoint groups decreases proportionally as you increase the number of listeners. Your (total # of listeners) x (# of endpoint groups) must not exceed 42.</td>
<td></td>
</tr>
<tr>
<td>Port ranges per listener</td>
<td>10</td>
</tr>
<tr>
<td>Port overrides per endpoint group</td>
<td>10</td>
</tr>
<tr>
<td>You can request a quota increase.</td>
<td></td>
</tr>
</tbody>
</table>
Quotas for endpoints per endpoint group

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

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

Related quotas

In addition to quotas in Global Accelerator, there are quotas that apply to the resources that you use as endpoints for an accelerator. For more information, see the following:

- **Elastic IP address quotas** in the Amazon EC2 User Guide.
- **Amazon EC2 service quotas** in the Amazon EC2 User Guide.
- **Quotas for your Network Load Balancers** in the User Guide for Network Load Balancers.
- **Quotas for your Application Load Balancers** in the User Guide for Application Load Balancers.
- **Amazon VPC quotas** in the Amazon VPC User Guide.
AWS Global Accelerator Related information

The information and resources listed here can help you learn more about Global Accelerator.

Topics
- Additional AWS Global Accelerator documentation (p. 115)
- Getting support (p. 115)
- Tips from the Amazon Web Services Blog (p. 115)

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:

- Accessing an Amazon API Gateway via static IP addresses provided by AWS Global Accelerator
- AWS Global Accelerator Custom Routing with Amazon Elastic Kubernetes Service
- Deploying multi-Region applications in AWS using AWS Global Accelerator
- Maximising application resiliency with AWS Global Accelerator
- Starting Small with AWS Global Accelerator
- Traffic management with AWS Global Accelerator
- Analyzing and visualizing AWS Global Accelerator flow logs using Amazon Athena and Amazon QuickSight

For a complete list of AWS Global Accelerator blogs, see [AWS Global Accelerator](https://aws.amazon.com) in the Networking & Content Delivery category of AWS blog posts.
## Document history

The following entries describe important changes made to the AWS Global Accelerator documentation.

- **API version:** latest
- **Latest documentation update:** September 26, 2023

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adds support in four AWS Regions</td>
<td>Support in the following AWS Regions has been added for Global Accelerator: Asia Pacific (Melbourne), Europe (Spain), Europe (Zurich), and Israel (Tel Aviv). For more information, see <a href="https://docs.aws.amazon.com/glu/latest/userguide/global-accelerator-region-availability.html">AWS Region availability for AWS Global Accelerator</a>.</td>
<td>September 26, 2023</td>
</tr>
<tr>
<td>Updates service linked role</td>
<td>Adds a new permission, <code>elasticloadbalancing:DescribeTargetGroups</code>, to the service. Global Accelerator uses the permission to identify load balancers with target type <code>ip</code>, which is not a supported target type for dual-stack load balancer endpoints in Global Accelerator. For more information, see <a href="https://docs.aws.amazon.com/glu/latest/userguide/service-linked-roles.html">Service-linked role for AWS Global Accelerator</a>.</td>
<td>September 12, 2023</td>
</tr>
<tr>
<td>Adds support for client IP address preservation for Network Load Balancers</td>
<td>Global Accelerator now supports enabling client IP address preservation for Network Load Balancers with security groups. For more information, see <a href="https://docs.aws.amazon.com/glu/latest/userguide/endpoint-address-preservation.html">Adding or updating endpoints with client IP address preservation</a>.</td>
<td>August 22, 2023</td>
</tr>
<tr>
<td>Adds IPv6 support for EC2 instances</td>
<td>Global Accelerator now supports adding Amazon EC2 instances to dual-stack accelerators, to enable both IPv4 and IPv6 traffic to EC2 endpoints. For a full list of supported endpoint types and more information, see <a href="https://docs.aws.amazon.com/glu/latest/userguide/global-accelerator-endpoints.html">Endpoints for standard accelerators in AWS Global Accelerator</a>.</td>
<td>August 8, 2023</td>
</tr>
<tr>
<td>Added a new Region</td>
<td>Global Accelerator now supports Asia Pacific (Jakarta). For a full list of supported Regions, see</td>
<td>June 15, 2023</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Added two new Regions</td>
<td>Global Accelerator now supports Asia Pacific (Hyderabad) and Middle East (UAE). For a full list of supported Regions, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/regions.html">AWS Region availability for AWS Global Accelerator</a>.</td>
<td>May 23, 2023</td>
</tr>
<tr>
<td>Updates service linked role</td>
<td>Adds new permissions, elasticloadbalancing:DescribeListeners and ec2:DescribeAddresses, to the service linked role for Global Accelerator, to support making listener management decisions for load balancers, based on listener configurations, and to add Elastic IP address endpoints to accelerators. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/service-linked-role.html">Service-linked role for AWS Global Accelerator</a>.</td>
<td>May 23, 2023</td>
</tr>
<tr>
<td>Adds custom routing accelerator quotas</td>
<td>Adds custom routing accelerator quotas. Global Accelerator also has quotas for standard accelerators. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/quotas.html">Quotas for AWS Global Accelerator</a>.</td>
<td>February 13, 2023</td>
</tr>
<tr>
<td>Updates the IAM guidance in the guide</td>
<td>Updated guide to align with the IAM best practices. For more information, see [Security best practices in IAM](<a href="https://docs.aws.amazon.com">https://docs.aws.amazon.com</a> IAM/latest/UserGuide/security_best_practices.html).</td>
<td>February 10, 2023</td>
</tr>
<tr>
<td>Updates for AddEndpoints and RemoveEndpoints</td>
<td>Global Accelerator now supports adding and removing endpoints separately from using the UpdateEndpointGroup API operation, by using the new AddEndpoints and RemoveEndpoints API operations. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/global-accelerator-actions.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/global-accelerator-actions.html</a>.</td>
<td>October 20, 2022</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>Updates for dual-stack accelerators</td>
<td>Global Accelerator now supports dual-stack accelerators. For IPv4, Global Accelerator provides two static IPv4 addresses. For dual-stack, Global Accelerator provides a total of four addresses: two static IPv4 addresses and two static IPv6 addresses. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/what-is-global-accelerator.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/what-is-global-accelerator.html</a>.</td>
<td>July 27, 2022</td>
</tr>
<tr>
<td>Update to the Global Accelerator existing service-linked role</td>
<td>Global Accelerator added new permissions, ec2:AssignIpv6Addresses and ec2:UnassignIpv6Addresses, to support IPv6 addresses. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/security-iam-awsmanpol-updates.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/security-iam-awsmanpol-updates.html</a>.</td>
<td>November 2, 2021</td>
</tr>
<tr>
<td>Added new CloudWatch metrics</td>
<td>Global Accelerator has added two new CloudWatch metrics. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/cloudwatch-monitoring.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/cloudwatch-monitoring.html</a>.</td>
<td>October 28, 2021</td>
</tr>
<tr>
<td>Update to the flow logs capture window</td>
<td>Global Accelerator has expanded the flow log capture window from 10 seconds to 60 seconds. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/monitoring-global-accelerator.flow-logs.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/monitoring-global-accelerator.flow-logs.html</a>.</td>
<td>July 30, 2021</td>
</tr>
<tr>
<td>Update to the Global Accelerator existing service-linked role</td>
<td>Global Accelerator added a new permission, ec2:DescribeRegions, to allow Global Accelerator to get AWS Region information to help diagnose errors. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/security-iam-awsmanpol-updates.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/security-iam-awsmanpol-updates.html</a>.</td>
<td>May 7, 2021</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Added custom routing accelerators</td>
<td>Global Accelerator introduced a new type of accelerator custom routing accelerators. Custom routing accelerators work well for scenarios where you want to use custom application logic to direct one or more users to a specific destination and port among many, while still gaining the performance benefits of Global Accelerator. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/work-with-custom-routing-accelerators.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/work-with-custom-routing-accelerators.html</a>.</td>
<td>December 9, 2020</td>
</tr>
<tr>
<td>Added port overrides support</td>
<td>Global Accelerator now supports overriding the listener port used for routing traffic to endpoints so you can reroute traffic to specific ports on your endpoints. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/about-endpoint-groups-port-override.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/about-endpoint-groups-port-override.html</a>.</td>
<td>October 21, 2020</td>
</tr>
<tr>
<td>Added two new Regions</td>
<td>Global Accelerator now supports Africa (Cape Town) and Europe (Milan). For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/preserve-client-ip-address.regions.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/preserve-client-ip-address.regions.html</a>.</td>
<td>May 20, 2020</td>
</tr>
<tr>
<td>Tagging and BYOIP</td>
<td>This release adds support for adding tags to accelerators and bringing your own IP address to AWS Global Accelerator (BYOIP). For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/tagging-in-global-accelerator.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/tagging-in-global-accelerator.html</a> and <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/using-byoip.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/using-byoip.html</a>.</td>
<td>February 27, 2020</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Updated Security chapter</td>
<td>Added content for compliance, resilience, and infrastructure security. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/security.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/security.html</a>.</td>
<td>December 20, 2019</td>
</tr>
<tr>
<td>Support for EC2 instances and default DNS name</td>
<td>AWS Global Accelerator now supports adding EC2 instances in supported AWS Regions. In addition, Global Accelerator creates a default DNS name that is mapped to the static IP addresses for your accelerator. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/introduction-how-it-works-client-ip.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/introduction-how-it-works-client-ip.html</a> and <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/about-accelerators.html#about-accelerators.dns-addressing">https://docs.aws.amazon.com/global-accelerator/latest/dg/about-accelerators.html#about-accelerators.dns-addressing</a>.</td>
<td>October 29, 2019</td>
</tr>
<tr>
<td>Client IP address preservation for Application Load Balancers</td>
<td>You can now choose to have AWS Global Accelerator preserve the client IP address for Application Load Balancers in supported AWS Regions. For more information, see <a href="https://docs.aws.amazon.com/global-accelerator/latest/dg/introduction-how-it-works-client-ip.html">https://docs.aws.amazon.com/global-accelerator/latest/dg/introduction-how-it-works-client-ip.html</a>.</td>
<td>August 28, 2019</td>
</tr>
<tr>
<td>Release of AWS Global Accelerator service</td>
<td>The AWS Global Accelerator Developer Guide provides information about setting up and using accelerators—network layer traffic managers—that improve availability and performance for your internet applications that have a global audience.</td>
<td>November 26, 2018</td>
</tr>
</tbody>
</table>
AWS glossary

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