



AWS Transit Gateway

Amazon VPC



Amazon VPC: AWS Transit Gateway

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

What is AWS Transit Gateway?	1
Transit gateway concepts	1
How to get started with transit gateways	2
Work with transit gateways	2
Pricing	3
How transit gateways work	4
Example architecture diagram	4
Resource attachments	5
Equal Cost Multipath routing	6
Availability Zones	7
Routing	8
Route tables	8
Route table association	9
Route propagation	9
Routes for peering attachments	10
Route evaluation order	10
Network function attachments	12
AWS Network Firewall integration	13
Example transit gateway scenarios	14
Get started with transit gateways	36
Create a transit gateway using the console	36
Prerequisites	36
Step 1: Create the transit gateway	37
Step 2: Attach your VPCs to your transit gateway	38
Step 3: Add routes between the transit gateway and your VPCs	39
Step 4: Test the transit gateway	40
Step 5: Delete the transit gateway	40
Create a transit gateway using the command line	40
Prerequisites	40
Step 1: Create the transit gateway	41
Step 2: Verify the transit gateway availability state	42
Step 3: Attach your VPCs to your transit gateway	44
Step 4: Verify that the transit gateway attachments are available	45
Step 5: Add routes between your transit gateway and VPCs	47

Step 6: Test the transit gateway	48
Step 7: Delete the transit gateway attachments and transit gateway	48
Conclusion	51
Design best practices	52
Work with transit gateways	53
Shared transit gateways	53
Share your transit gateways	53
Unshare a transit gateway	55
Shared subnets	55
Transit gateways	55
Create a transit gateway	57
View a transit gateway	59
Manage transit gateway tags	59
Modify a transit gateway	60
Accept a resource share	61
Accept a shared attachment	61
Delete a transit gateway	62
Encryption Support	62
VPC attachments	64
Route table requirements for VPC attachments	65
VPC attachment lifecycle	66
Appliance mode	69
Security group referencing	71
IPv6 support	72
Create a VPC attachment	73
Modify a VPC attachment	74
Modify VPC attachment tags	75
View a VPC attachment	75
Delete a VPC attachment	76
Update security group inbound rules	76
Identify referenced security groups	77
Remove stale security group rules	77
Troubleshoot VPC attachments	78
Network function attachments	79
Accept or reject a transit gateway network function attachment	79
View network function attachments	80

Route traffic through a transit gateway network function attachment	81
VPN attachments	83
Create a transit gateway attachment to a VPN	84
View a VPN attachment	84
Delete a VPN attachment	85
VPN Concentrator attachments	85
How VPN Concentrator works	85
Benefits of VPN Concentrator	86
Create a VPN Concentrator attachment	87
View a VPN Concentrator attachment	89
Delete a VPN Concentrator attachment	89
Client VPN attachments	90
Create a Client VPN attachment	91
View a Client VPN attachment	92
Delete a Client VPN attachment	92
Accept or reject a Client VPN attachment	93
Transit gateway attachments to a Direct Connect gateway	94
Peering attachments	95
Opt-in AWS Region considerations	95
Create a peering attachment	96
Accept or reject a peering request	97
Add a route to a transit gateway route table	98
Delete a peering attachment	99
Connect attachments and Connect peers	99
Connect peers	100
Requirements and considerations	103
Create a Connect attachment	104
Create a Connect peer	105
View Connect attachments and Connect peers	106
Modify Connect attachment and Connect peer tags	107
Delete a Connect peer	107
Delete a Connect attachment	108
Transit gateway route tables	108
Create a transit gateway route table	109
View transit gateway route tables	110
Associate a transit gateway route table	110

Disassociate a transit gateway route table	111
Enable route propagation	111
Disable route propagation	112
Create a static route	112
Delete a static route	113
Replace a static route	114
Export route tables to Amazon S3	114
Delete a transit gateway route table	116
Create a prefix list reference	116
Modify a prefix list reference	117
Delete a prefix list reference	118
Transit gateway policy tables	118
Policy table concepts	119
Prerequisites	123
Create a transit gateway policy table	124
View transit gateway policy tables	125
Create a policy table entry	128
Modify a policy table entry	130
Delete a policy table entry	132
Associate a transit gateway policy table	134
Disassociate a transit gateway policy table	137
Delete a transit gateway policy table	138
Example: Steering traffic to a security appliance	139
Limitations	141
Troubleshooting	141
Related resources	142
Multicast on transit gateways	143
Multicast concepts	1
Considerations	144
Multicast routing	146
Multicast domains	147
Shared multicast domains	153
Register sources with a multicast group	158
Register members with a multicast group	159
Deregister sources from a multicast group	159
Deregister members from a multicast group	160

View multicast groups	160
Set up multicast for Windows Server	161
Example: Manage IGMP configurations	162
Example: Manage static source configurations	163
Example: Manage static group member configurations	164
Flexible cost allocation	165
Metering policies	165
Create a metering policy	170
Manage metering policies	172
Create a metering policy entry	177
Delete a metering policy entry	180
Manage metering policy middlebox attachments	167
Transit Gateway Flow Logs	187
Limitations	188
Transit Gateway Flow Log records	188
Default format	189
Custom format	189
Available fields	189
Control the use of flow logs	195
Transit Gateway Flow Logs pricing	196
Create or update a Flow Logs IAM role	196
CloudWatch Logs Flow Logs	197
IAM roles for publishing flow logs to CloudWatch Logs	198
Permissions for IAM users to pass a role	199
Create a Flow Log that publishes to CloudWatch Logs	200
View Flow Logs records	201
Process Flow Log records	202
Amazon S3 Flow Logs	203
Flow log files	204
IAM policy for IAM principals that publish flow logs to Amazon S3	206
Amazon S3 bucket permissions for flow logs	206
Required key policy for use with SSE-KMS	208
Amazon S3 log file permissions	209
Create the source account role	210
Create a Flow Log that publishes to Amazon S3	211
View Flow Logs records	212

Processed AWS Transit Gateway Flow Logs records in Amazon S3	213
Amazon Data Firehose Flow Logs	213
IAM roles for cross account delivery	213
Create the source account role	216
Create the destination account role	217
Create a Flow Log that publishes to Firehose	218
Create and manage Flow Logs using the APIs or CLI	220
View Flow Logs	221
Manage Flow Logs tags	221
Search Flow Logs records	222
Delete a Flow Logs record	223
Metrics and events	225
CloudWatch metrics	226
Transit gateway metrics	226
Attachment-level and availability zone metrics	228
Transit gateway metric dimensions	230
CloudTrail logs	230
Management events	232
Event examples	232
Identity and access management	235
Example policies to manage transit gateways	235
Service-linked roles	238
Transit gateway	238
AWS managed policies	239
AWSVPCTransitGatewayServiceRolePolicy	240
Policy updates	240
Network ACLs	240
Same subnet for EC2 instances and transit gateway association	241
Different subnets for EC2 instances and transit gateway association	241
Best Practices	242
Quotas	243
General	243
Routing	243
Transit gateway attachments	244
Bandwidth	245
Direct Connect gateways	247

Maximum transmission unit (MTU)	247
Multicast	248
Network Manager	249
Additional quota resources	250
Document history	251

What is AWS Transit Gateway for Amazon VPC?

AWS Transit Gateway is a network transit hub used to interconnect virtual private clouds (VPCs) and on-premises networks. As your cloud infrastructure expands globally, inter-Region peering connects transit gateways together using the AWS Global Infrastructure. All network traffic between AWS data centers is automatically encrypted at the physical layer.

For more information, see the [AWS Transit Gateway](#) website.

Transit gateway concepts

The following are the key concepts for transit gateways:

- **Attachments** — You can attach the following:
 - One or more VPCs
 - A Connect SD-WAN/third-party network appliance
 - An AWS Direct Connect gateway
 - A peering connection with another transit gateway
 - A VPN connection to a transit gateway
 - A VPN Concentrator to a transit gateway
 - A Client VPN endpoint to a transit gateway
 - A network function attachment. For more information, see [the section called “Network function attachments”](#).
- **Transit gateway Maximum Transmission Unit (MTU)** — The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The larger the MTU of a connection, the more data that can be passed in a single packet. A transit gateway supports an MTU of 8500 bytes for traffic between VPCs, Direct Connect, Transit Gateway Connect, and peering attachments (intra-Region, inter-Region, and Cloud WAN peering attachments). Traffic over VPN connections can have an MTU of 1500 bytes.
- **Encryption control** — A transit gateway can be configured to support Encryption control, which enforces encryption-in-transit for all traffic on VPCs attached to the transit gateway. When Encryption control is enabled, the transit gateway can be attached to VPCs with Encryption control enforced. This feature ensures that all traffic flowing through the transit gateway is encrypted, providing enhanced security for your network communications.

- **Transit gateway route table** — A transit gateway has a default route table and can optionally have additional route tables. A route table includes dynamic and static routes that decide the next hop based on the destination IP address of the packet. The target of these routes could be any transit gateway attachment. By default, transit gateway attachments are associated with the default transit gateway route table.
- **Associations** — Each attachment is associated with exactly one route table. Each route table can be associated with zero to many attachments.
- **Route propagation** — A VPC, VPN connection, or Direct Connect gateway can dynamically propagate routes to a transit gateway route table. With a Connect attachment, the routes are propagated to a transit gateway route table by default. With a VPC, you must create static routes to send traffic to the transit gateway. With a VPN connection, routes are propagated from the transit gateway to your on-premises router using Border Gateway Protocol (BGP). With a Direct Connect gateway, allowed prefixes are originated to your on-premises router using BGP. With a peering attachment, you must create a static route in the transit gateway route table to point to the peering attachment.

How to get started with transit gateways

Use the following resources to help you create and use a transit gateway.

- [How transit gateways work](#)
- [Get started with transit gateways](#)
- [Design best practices](#)

Work with transit gateways

You can create, access, and manage your transit gateways using any of the following interfaces:

- **AWS Management Console** — Provides a web interface that you can use to access your transit gateways.
- **AWS Command Line Interface (AWS CLI)** — Provides commands for a broad set of AWS services, including Amazon VPC, and is supported on Windows, macOS, and Linux. For more information, see [AWS Command Line Interface](#).

- **AWS SDKs** — Provides language-specific API operations and takes care of many of the connection details, such as calculating signatures, handling request retries, and handling errors. For more information, see [AWS SDKs](#).
- **Query API** — Provides low-level API actions that you call using HTTPS requests. Using the Query API is the most direct way to access Amazon VPC, but it requires that your application handle low-level details such as generating the hash to sign the request, and handling errors. For more information, see the [Amazon EC2 API Reference](#).

Pricing

You are charged hourly for each attachment on a transit gateway, and you are charged for the amount of traffic processed on the transit gateway. By default, data processing charges are allocated to the account that owns the source attachment. You can use flexible cost allocation to customize how these charges are allocated based on your organizational needs. For more information, see [AWS Transit Gateway pricing](#) and [Flexible cost allocation](#).

How AWS Transit Gateway works

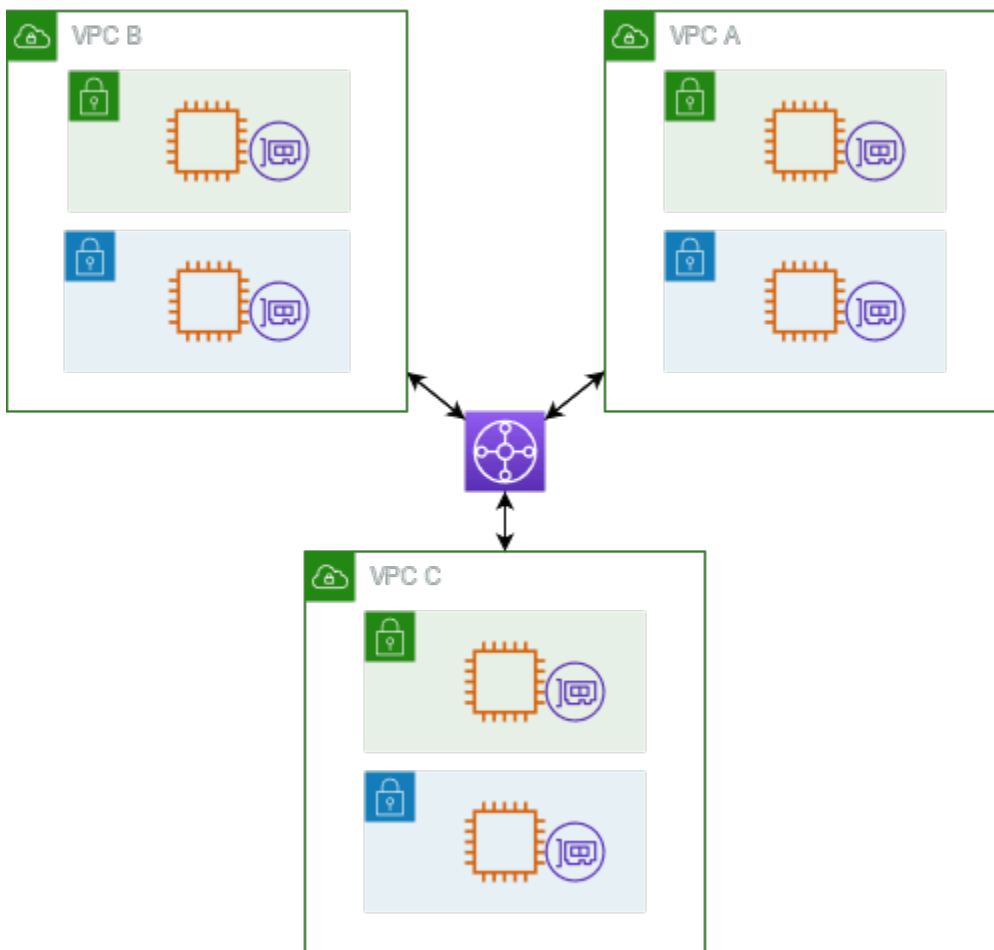
In AWS Transit Gateway, a transit gateway acts as a Regional virtual router for traffic flowing between your virtual private clouds (VPCs) and on-premises networks. A transit gateway scales elastically based on the volume of network traffic. Routing through a transit gateway operates at layer 3, where the packets are sent to a specific next-hop attachment, based on the destination IP addresses.

Topics

- [Example architecture diagram](#)
- [Resource attachments](#)
- [Equal Cost Multipath routing](#)
- [Availability Zones](#)
- [Routing](#)
- [Network function attachments](#)
- [Example transit gateway scenarios](#)

Example architecture diagram

The following diagram shows a transit gateway with three VPC attachments. The route table for each of these VPCs includes the local route and routes that send traffic destined for the other two VPCs to the transit gateway.



The following is an example of a default transit gateway route table for the attachments shown in the previous diagram. The CIDR blocks for each VPC propagate to the route table. Therefore, each attachment can route packets to the other two attachments.

Destination	Target	Route type
<i>VPC A CIDR</i>	<i>Attachment for VPC A</i>	propagated
<i>VPC B CIDR</i>	<i>Attachment for VPC B</i>	propagated
<i>VPC C CIDR</i>	<i>Attachment for VPC C</i>	propagated

Resource attachments

A transit gateway attachment is both a source and a destination of packets. You can attach the following resources to your transit gateway:

- One or more VPCs. AWS Transit Gateway deploys an elastic network interface within VPC subnets, which is then used by the transit gateway to route traffic to and from the chosen subnets. You must have at least one subnet for each Availability Zone, which then enables traffic to reach resources in every subnet of that zone. During attachment creation, resources within a particular Availability Zone can reach a transit gateway only if a subnet is enabled within the same zone. If a subnet route table includes a route to the transit gateway, traffic is only forwarded to the transit gateway if the transit gateway has an attachment in the subnet of the same Availability Zone.
- One or more VPN connections
- One or more VPN Concentrators
- One or more AWS Direct Connect gateways
- One or more Transit Gateway Connect attachments
- One or more transit gateway peering connections

Equal Cost Multipath routing

AWS Transit Gateway supports Equal Cost Multipath (ECMP) routing for most attachments. For a VPN attachment, you can enable or disable ECMP support using the console when creating or modifying a transit gateway. For all other attachment types, the following ECMP restrictions apply:

- VPC - VPC does not support ECMP since CIDR blocks cannot overlap. For example, you can't attach a VPC with a CIDR 10.1.0.0/16 with a second VPC using the same CIDR to a transit gateway, and then set up routing to load balance the traffic between them.
- VPN - When the **VPN ECMP support** option is disabled, a transit gateway uses internal metrics to determine the preferred path in the event of equal prefixes across multiple paths. For more information on enabling or disabling ECMP for a VPN attachment, see [the section called "Transit gateways"](#).
- AWS Transit Gateway Connect - AWS Transit Gateway Connect attachments automatically support ECMP.
- AWS Direct Connect Gateway - AWS Direct Connect Gateway attachments automatically support ECMP across multiple Direct Connect Gateway attachments when the network prefix, prefix length, and AS_PATH are exactly the same.
- Transit gateway peering - Transit gateway peering does not support ECMP since it neither supports dynamic routing nor can you configure the same static route against two different targets.

- VPN Concentrator - VPN Concentrator does not support ECMP.

Note

- BGP Multipath AS-Path Relax is not supported, so you can't use ECMP over different Autonomous System Numbers (ASNs).
- ECMP is not supported between different attachment types. For example, you can't enable ECMP between a VPN and a VPC attachment. Instead, transit gateway routes are evaluated and traffic routed accordingly to the evaluated route. For more information, see [the section called "Route evaluation order"](#).
- A single Direct Connect gateway supports ECMP across multiple transit virtual interfaces. Therefore, we recommended that you set up and use only a single Direct Connect gateway and to not set up and use multiple gateways to take advantage of ECMP. For more information about Direct Connect gateways and public virtual interfaces, see [How do I set up an Active/Active or Active/Passive Direct Connect connection to AWS from a public virtual interface?](#).

Availability Zones

When you attach a VPC to a transit gateway, you must enable one or more Availability Zones to be used by the transit gateway to route traffic to resources in the VPC subnets. To enable each Availability Zone, you specify exactly one subnet. The transit gateway places a network interface in that subnet using one IP address from the subnet. After you enable an Availability Zone by specifying a subnet, traffic can be routed to all subnets in that Availability Zone, not just the one you specified. However, only resources that reside in Availability Zones where there is a transit gateway attachment can reach the transit gateway.

If traffic is sourced from an Availability Zone that the destination attachment is not present in, AWS Transit Gateway will internally route that traffic to a random Availability Zone where the attachment is present. There is no additional transit gateway charge for this type of cross-Availability Zone traffic.

We recommend that you enable multiple Availability Zones to ensure availability.

Using appliance mode support

If you plan to configure a stateful network appliance in your VPC, you can enable appliance mode support for the VPC attachment in which the appliance is located. This ensures that the transit gateway uses the same Availability Zone for that VPC attachment for the lifetime of a flow of traffic between source and destination. It also allows the transit gateway to send traffic to any Availability Zone in the VPC, as long as there is a subnet association in that zone. For more information, see [Example: Appliance in a shared services VPC](#).

Routing

Your transit gateway routes IPv4 and IPv6 packets between attachments using transit gateway route tables. You can configure these route tables to propagate routes from the route tables for the attached VPCs, VPN connections, and Direct Connect gateways. You can also add static routes to the transit gateway route tables. When a packet comes from one attachment, it is routed to another attachment using the route that matches the destination IP address.

For transit gateway peering attachments, only static routes are supported.

Routing topics

- [Route tables](#)
- [Route table association](#)
- [Route propagation](#)
- [Routes for peering attachments](#)
- [Route evaluation order](#)

Route tables

Your transit gateway automatically comes with a default route table. By default, this route table is the default association route table and the default propagation route table. If you disable both route propagation and route table association, AWS does not create a default route table for the transit gateway. However, if either route propagation or route table association is enabled, AWS then creates a default route table.

You can create additional route tables for your transit gateway. This enables you to isolate subsets of attachments. Each attachment can be associated with one route table. An attachment can propagate its routes to one or more route tables.

You can create a blackhole route in your transit gateway route table that drops traffic that matches the route.

When you attach a VPC to a transit gateway, you must add a route to your subnet route table in order for traffic to route through the transit gateway. For more information, see [Routing for a Transit Gateway](#) in the *Amazon VPC User Guide*.

Route table association

You can associate a transit gateway attachment with a single route table. Each route table can be associated with zero to many attachments and can forward packets to other attachments.

Route propagation

Each attachment comes with routes that can be installed in one or more transit gateway route tables. When an attachment is propagated to a transit gateway route table, these routes are installed in the route table. You can't filter on advertised routes.

For a VPC attachment, the CIDR blocks of the VPC are propagated to the transit gateway route table.

When dynamic routing is used with a VPN attachment, VPN Concentrator attachment or a Direct Connect gateway attachment, you can propagate the routes learned from the on-premises router through BGP to any of the transit gateway route tables.

When dynamic routing is used with a VPN attachment or a VPN Concentrator attachment, the routes in the route table associated with the VPN attachment or VPN Concentrator attachment are advertised to the customer gateway through BGP.

For a Connect attachment, routes in the route table associated with the Connect attachment are advertised to the third-party virtual appliances, such as SD-WAN appliances, running in a VPC through BGP.

For a Direct Connect gateway attachment, [allowed prefixes interactions](#) control which routes are advertised to the customer network from AWS.

When a static route and a propagated route have the same destination, the static route has the higher priority, so the propagated route is not included in the route table. If you remove the static route, the overlapping propagated route is included in the route table.

Routes for peering attachments

You can peer two transit gateways, and route traffic between them. To do this, you create a peering attachment on your transit gateway, and specify the peer transit gateway with which to create the peering connection. You then create a static route in your transit gateway route table to route traffic to the transit gateway peering attachment. Traffic that's routed to the peer transit gateway can then be routed to the VPC and VPN attachments for the peer transit gateway.

For more information, see [Example: Peered transit gateways](#).

Route evaluation order

Transit gateway routes are evaluated in the following order:

- The most specific route for the destination address.
- For routes with the same CIDR, but from different attachment types, the route priority is as follows:
 - Static routes (for example, Site-to-Site VPN static routes)
 - Prefix list referenced routes
 - VPC-propagated routes
 - Direct Connect gateway-propagated routes
 - Transit Gateway Connect-propagated routes
 - Site-to-Site VPN over private Direct Connect-propagated routes
 - Site-to-Site VPN-propagated routes
 - Site-to-Site VPN-Concentrator propagated routes
 - Client VPN propagated routes
 - Transit Gateway peering-propagated routes (Cloud WAN)

Some attachments support route advertisement over BGP. For routes with the same CIDR, and from the same attachment type, the route priority is controlled by BGP attributes:

- Shorter AS Path length
- Lower MED value
- eBGP over iBGP routes are preferred, if the attachment supports it

⚠ Important

- AWS can't guarantee a consistent route prioritization order for BGP routes with the same CIDR, attachment type, and BGP attributes as listed above.
- For routes advertised to a transit gateway without MED, AWS Transit Gateway will assign the following default values:
 - 0 for inbound routes advertised on Direct Connect attachments.
 - 100 for inbound routes advertised on VPN and Connect attachments.

AWS Transit Gateway only shows a preferred route. A backup route will only appear in the transit gateway route table if the previously active route is no longer advertised — for example, if you are advertising the same routes over the Direct Connect gateway and over Site-to-Site VPN. AWS Transit Gateway will only show the routes received from the Direct Connect gateway route, which is the preferred route. The Site-to-Site VPN, which is the backup route, will only display when the Direct Connect gateway is no longer advertised.

VPC and transit gateway route table differences

Route table evaluation differs between whether you're using a VPC route table or a transit gateway route table.

The following example shows a VPC route table. The VPC local route has the highest priority, followed by the routes that are the most specific. When a static route and a propagated route have the same destination, the static route has a higher priority.

Destination	Target	Priority
10.0.0.0/16	local	1
192.168.0.0/16	pcx-12345	2
172.31.0.0/16	vgw-12345 (static) or tgw-12345 (static)	2
172.31.0.0/16	vgw-12345 (propagated)	3

Destination	Target	Priority
0.0.0.0/0	igw-12345	4

The following example shows a transit gateway route table. If you prefer the Direct Connect gateway attachment to the VPN attachment, use a BGP VPN connection and propagate the routes in the transit gateway route table.

Destination	Attachment (Target)	Resource type	Route type	Priority
10.0.0.0/16	tgw-attach-123 vpc-1234	VPC	Static or propagated	1
192.168.0.0/16	tgw-attach-789 vpn-5678	VPN	Static	2
172.31.0.0/16	tgw-attach-456 dxgw_id	Direct Connect gateway	Propagated	3
172.31.0.0/16	tgw-attach-789 tgw-connect-peer-123	Connect	Propagated	4
172.31.0.0/16	tgw-attach-789 vpn-5678	VPN	Propagated	5

Network function attachments

A network function attachment is a resource that connects a network security function — for example, an AWS Network Firewall attachment — directly to your transit gateway. It eliminates the need to manually create and manage inspection VPCs.

With a network function attachment:

- AWS automatically creates and manages the underlying infrastructure
- Traffic can be inspected as it flows through your transit gateway

- Security policies are applied consistently across your network
- You can direct traffic through the firewall using simple routing rules
- The attachment works across multiple Availability Zones for high availability

This integration simplifies network security by allowing you to attach firewalls directly to your transit gateway rather than creating complex routing configurations and managing separate endpoints through separate VPCs.

AWS Network Firewall integration

AWS Network Firewall integration allows you to connect a firewall in the form of a group of Gateway Load Balancer Endpoints, one per Availability Zone, in a service-managed buffer VPC. A Network Firewall attachment is created with appliance mode automatically enabled. This eliminates the need to explicitly manage inspection VPCs.

With Network Firewall integration, you no longer need to create and manage inspection VPCs for your Network Firewall deployments. Instead of selecting a VPC and subnets when creating your firewall, you directly select the Transit Gateway, and AWS automatically provisions and manages all the necessary resources behind the scenes. You'll see a new transit gateway network function attachment rather than an individual firewall endpoint.

For cross-account scenarios, the Transit Gateway can be RAM-shared from the Transit Gateway owner to the Network Firewall owner account, allowing either account to manage the firewall attachment. Once your firewall and attachment are ready, you can simply modify your Transit Gateway route tables to send traffic to the attachment for inspection.

Note

- Transit Gateway supports only static routing on Network Firewall attachments.
- Third-party firewalls are not supported.

For more information about firewalls and attachments see [Transit gateway network function attachments](#).

Example transit gateway scenarios

The following are common use cases for transit gateways. Your transit gateways are not limited to these use cases.

Example: Centralized router

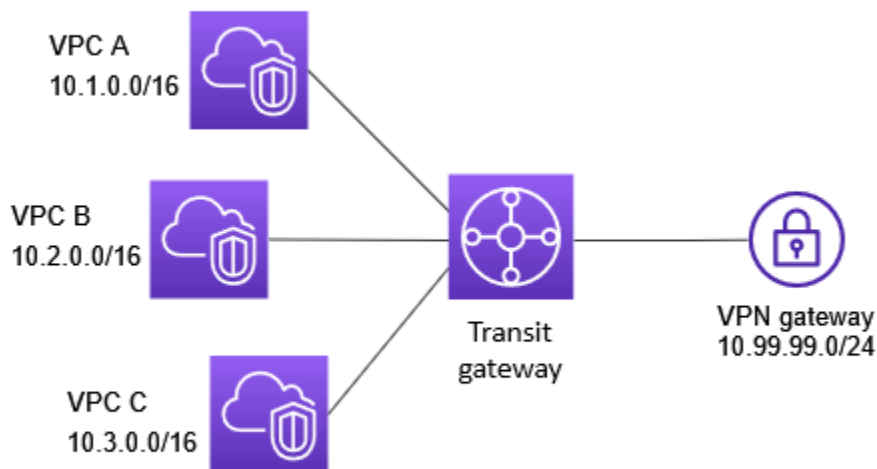
You can configure your transit gateway as a centralized router that connects all of your VPCs, AWS Direct Connect, and Site-to-Site VPN connections. In this scenario, all attachments are associated with the transit gateway default route table and propagate to the transit gateway default route table. Therefore, all attachments can route packets to each other, with the transit gateway serving as a simple layer 3 IP router.

Contents

- [Overview](#)
- [Resources](#)
- [Routing](#)

Overview

The following diagram shows the key components of the configuration for this scenario. In this scenario, there are three VPC attachments and one Site-to-Site VPN attachment to the transit gateway. Packets from the subnets in VPC A, VPC B, and VPC C that are destined for a subnet in another VPC or for the VPN connection first route through the transit gateway.



Resources

Create the following resources for this scenario:

- Three VPCs. For more information, see [Create a VPC](#) in the *Amazon VPC User Guide*.
- A transit gateway. For more information, see [the section called “Create a transit gateway”](#).
- Three VPC attachments on the transit gateway. For more information, see [the section called “Create a VPC attachment”](#).
- A Site-to-Site VPN attachment on the transit gateway. The CIDR blocks for each VPC propagate to the transit gateway route table. When the VPN connection is up, the BGP session is established and the Site-to-Site VPN CIDR propagates to the transit gateway route table and the VPC CIDRs are added to the customer gateway BGP table. For more information, see [the section called “Create a transit gateway attachment to a VPN”](#).

Ensure that you review the [requirements for your customer gateway device](#) in the *AWS Site-to-Site VPN User Guide*.

Routing

Each VPC has a route table and there is a route table for the transit gateway.

VPC route tables

Each VPC has a route table with 2 entries. The first entry is the default entry for local IPv4 routing in the VPC; this entry enables the instances in this VPC to communicate with each other. The second entry routes all other IPv4 subnet traffic to the transit gateway. The following table shows the VPC A routes.

Destination	Target
10.1.0.0/16	local
0.0.0.0/0	<i>tgw-id</i>

Transit gateway route table

The following is an example of a default route table for the attachments shown in the previous diagram, with route propagation enabled.

Destination	Target	Route type
10.1.0.0/16	<i>Attachment for VPC A</i>	propagated
10.2.0.0/16	<i>Attachment for VPC B</i>	propagated
10.3.0.0/16	<i>Attachment for VPC C</i>	propagated
10.99.99.0/24	<i>Attachment for VPN connection</i>	propagated

Customer gateway BGP table

The customer gateway BGP table contains the following VPC CIDRs.

- 10.1.0.0/16
- 10.2.0.0/16

- 10.3.0.0/16

Example: Isolated VPCs

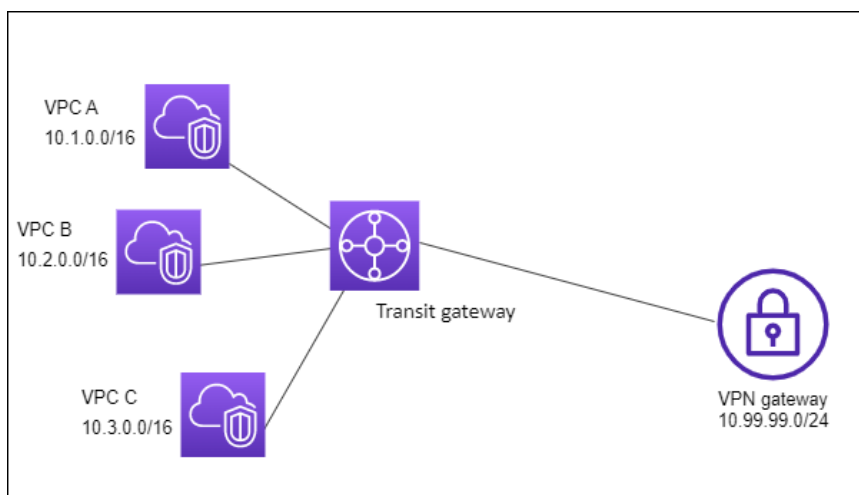
You can configure your transit gateway as multiple isolated routers. This is similar to using multiple transit gateways, but provides more flexibility in cases where the routes and attachments might change. In this scenario, each isolated router has a single route table. All attachments associated with an isolated router propagate and associate with its route table. Attachments associated with one isolated router can route packets to each other, but cannot route packets to or receive packets from the attachments for another isolated router.

Contents

- [Overview](#)
- [Resources](#)
- [Routing](#)

Overview

The following diagram shows the key components of the configuration for this scenario. Packets from VPC A, VPC B, and VPC C route to the transit gateway. Packets from the subnets in VPC A, VPC B, and VPC C that have the internet as a destination first route through the transit gateway and then route to the Site-to-Site VPN connection (if the destination is within that network). Packets from one VPC that have a destination of a subnet in another VPC, for example from 10.1.0.0 to 10.2.0.0, route through the transit gateway, where they are blocked because there is no route for them in the transit gateway route table.



Resources

Create the following resources for this scenario:

- Three VPCs. For more information, see [Create a VPC](#) in the *Amazon VPC User Guide*.
- A transit gateway. For more information, see [the section called “Create a transit gateway”](#).
- Three attachments on the transit gateway for the three VPCs. For more information, see [the section called “Create a VPC attachment”](#).
- A Site-to-Site VPN attachment on the transit gateway. For more information, see [the section called “Create a transit gateway attachment to a VPN”](#). Ensure that you review the [requirements for your customer gateway device](#) in the *AWS Site-to-Site VPN User Guide*.

When the VPN connection is up, the BGP session is established and the VPN CIDR propagates to the transit gateway route table and the VPC CIDRs are added to the customer gateway BGP table.

Routing

Each VPC has a route table, and the transit gateway has two route tables—one for the VPCs and one for the VPN connection.

VPC A, VPC B, and VPC C route tables

Each VPC has a route table with 2 entries. The first entry is the default entry for local IPv4 routing in the VPC. This entry enables the instances in this VPC to communicate with each other. The second entry routes all other IPv4 subnet traffic to the transit gateway. The following table shows the VPC A routes.

Destination	Target
10.1.0.0/16	local
0.0.0.0/0	<i>tgw-id</i>

Transit gateway route tables

This scenario uses one route table for the VPCs and one route table for the VPN connection.

The VPC attachments are associated with the following route table, which has a propagated route for the VPN attachment.

Destination	Target	Route type
10.99.99.0/24	<i>Attachment for VPN connection</i>	propagated

The VPN attachment is associated with the following route table, which has propagated routes for each of the VPC attachments.

Destination	Target	Route type
10.1.0.0/16	<i>Attachment for VPC A</i>	propagated
10.2.0.0/16	<i>Attachment for VPC B</i>	propagated
10.3.0.0/16	<i>Attachment for VPC C</i>	propagated

For more information about propagating routes in a transit gateway route table, see [Enable route propagation to a transit gateway route table in AWS Transit Gateway](#).

Customer gateway BGP table

The customer gateway BGP table contains the following VPC CIDRs.

- 10.1.0.0/16
- 10.2.0.0/16
- 10.3.0.0/16

Example: Isolated VPCs with shared services

You can configure your transit gateway as multiple isolated routers that use a shared service. This is similar to using multiple transit gateways, but provides more flexibility in cases where the routes and attachments might change. In this scenario, each isolated router has a single route table.

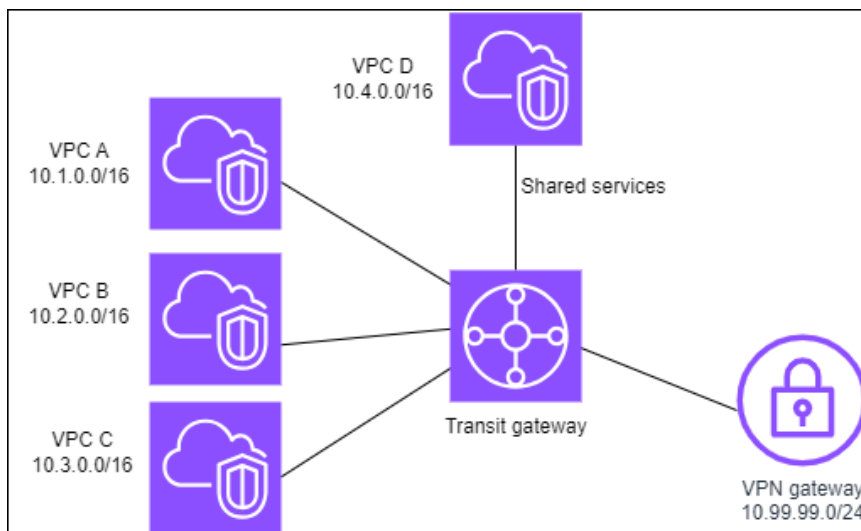
All attachments associated with an isolated router propagate and associate with its route table. Attachments associated with one isolated router can route packets to each other, but cannot route packets to or receive packets from the attachments for another isolated router. Attachments can route packets to or receive packets from the shared services. You can use this scenario when you have groups that need to be isolated, but use a shared service, for example a production system.

Contents

- [Overview](#)
- [Resources](#)
- [Routing](#)

Overview

The following diagram shows the key components of the configuration for this scenario. Packets from the subnets in VPC A, VPC B, and VPC C that have the internet as a destination, first route through the transit gateway and then route to the customer gateway for Site-to-Site VPN. Packets from subnets in VPC A, VPC B, or VPC C that have a destination of a subnet in VPC A, VPC B, or VPC C route through the transit gateway, where they are blocked because there is no route for them in the transit gateway route table. Packets from VPC A, VPC B, and VPC C that have VPC D as the destination route through the transit gateway and then to VPC D.



Resources

Create the following resources for this scenario:

- Four VPCs. For more information, see [Create a VPC](#) in the *Amazon VPC User Guide*.

- A transit gateway. For more information, see [Create a transit gateway](#).
- Four attachments on the transit gateway, one per VPC. For more information, see [the section called “Create a VPC attachment”](#).
- A Site-to-Site VPN attachment on the transit gateway. For more information, see [the section called “Create a transit gateway attachment to a VPN”](#).

Ensure that you review the [requirements for your customer gateway device](#) in the *AWS Site-to-Site VPN User Guide*.

When the VPN connection is up, the BGP session is established and the VPN CIDR propagates to the transit gateway route table and the VPC CIDRs are added to the customer gateway BGP table.

- Each isolated VPC is associated with the isolated route table and propagated to the shared route table.
- Each shared services VPC is associated with the shared route table and propagated to both route tables.

Routing

Each VPC has a route table, and the transit gateway has two route tables—one for the VPCs and one for the VPN connection and shared services VPC.

VPC A, VPC B, VPC C, and VPC D route tables

Each VPC has a route table with two entries. The first entry is the default entry for local routing in the VPC; this entry enables the instances in this VPC to communicate with each other. The second entry routes all other IPv4 subnet traffic to the transit gateway.

Destination	Target
10.1.0.0/16	local
0.0.0.0/0	<i>transit gateway ID</i>

Transit gateway route tables

This scenario uses one route table for the VPCs and one route table for the VPN connection.

The VPC A, B, and C attachments are associated with the following route table, which has a propagated route for the VPN attachment and a propagated route for the attachment for VPC D.

Destination	Target	Route type
10.99.99.0/24	<i>Attachment for VPN connection</i>	propagated
10.4.0.0/16	<i>Attachment for VPC D</i>	propagated

The VPN attachment and shared services VPC (VPC D) attachments are associated with the following route table, which has entries that point to each of the VPC attachments. This enables communication to the VPCs from the VPN connection and the shared services VPC.

Destination	Target	Route type
10.1.0.0/16	<i>Attachment for VPC A</i>	propagated
10.2.0.0/16	<i>Attachment for VPC B</i>	propagated
10.3.0.0/16	<i>Attachment for VPC C</i>	propagated

For more information, see [Enable route propagation to a transit gateway route table in AWS Transit Gateway](#).

Customer gateway BGP table

The customer gateway BGP table contains the CIDRs for all four VPCs.

Example: Peered transit gateways

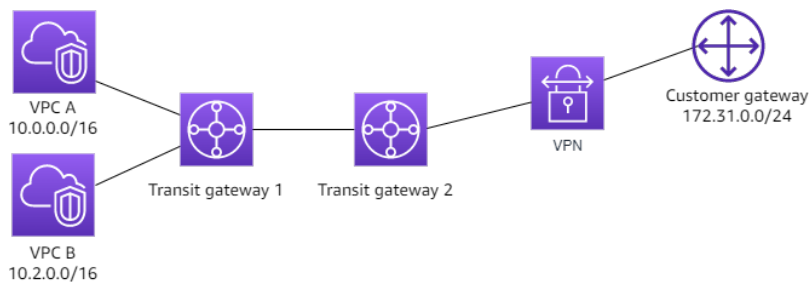
You can create a transit gateway peering connection between transit gateways. You can then route traffic between the attachments for each of the transit gateways. In this scenario, VPC and VPN attachments are associated with the transit gateway default route tables, and they propagate to the transit gateway default route tables. Each transit gateway route table has a static route that points to the transit gateway peering attachment.

Contents

- [Overview](#)
- [Resources](#)
- [Routing](#)

Overview

The following diagram shows the key components of the configuration for this scenario. Transit gateway 1 has two VPC attachments, and transit gateway 2 has one Site-to-Site VPN attachment. Packets from the subnets in VPC A and VPC B that have the internet as a destination first route through transit gateway 1, then transit gateway 2, and then route to the VPN connection.



Resources

Create the following resources for this scenario:

- Two VPCs. For more information, see [Create a VPC](#) in the *Amazon VPC User Guide*.
- Two transit gateways. They can be in the same Region or in different Regions. For more information, see [the section called "Create a transit gateway"](#).
- Two VPC attachments on the first transit gateway. For more information, see [the section called "Create a VPC attachment"](#).
- A Site-to-Site VPN attachment on the second transit gateway. For more information, see [the section called "Create a transit gateway attachment to a VPN"](#). Ensure that you review the [requirements for your customer gateway device](#) in the *AWS Site-to-Site VPN User Guide*.
- A transit gateway peering attachment between the two transit gateways. For more information, see [Transit gateway peering attachments in AWS Transit Gateway](#).

When you create the VPC attachments, the CIDRs for each VPC propagate to the route table for transit gateway 1. When the VPN connection is up, the following actions occur:

- The BGP session is established

- The Site-to-Site VPN CIDR propagates to the route table for transit gateway 2
- The VPC CIDRs are added to the customer gateway BGP table

Routing

Each VPC has a route table and each transit gateway has a route table.

VPC A and VPC B route tables

Each VPC has a route table with 2 entries. The first entry is the default entry for local IPv4 routing in the VPC. This default entry enables the resources in this VPC to communicate with each other. The second entry routes all other IPv4 subnet traffic to the transit gateway. The following table shows the VPC A routes.

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	<i>tgw-1-id</i>

Transit gateway route tables

The following is an example of the default route table for transit gateway 1, with route propagation enabled.

Destination	Target	Route type
10.0.0.0/16	<i>Attachment ID for VPC A</i>	propagated
10.2.0.0/16	<i>Attachment ID for VPC B</i>	propagated
0.0.0.0/0	<i>Attachment ID for peering connection</i>	static

The following is an example of the default route table for transit gateway 2, with route propagation enabled.

Destination	Target	Route type
172.31.0.0/24	<i>Attachment ID for VPN connection</i>	propagated
10.0.0.0/16	<i>Attachment ID for peering connection</i>	static
10.2.0.0/16	<i>Attachment ID for peering connection</i>	static

Customer gateway BGP table

The customer gateway BGP table contains the following VPC CIDRs.

- 10.0.0.0/16
- 10.2.0.0/16

Example: Centralized outbound routing to the internet

You can configure a transit gateway to route outbound internet traffic from a VPC without an internet gateway to a VPC that contains a NAT gateway and an internet gateway.

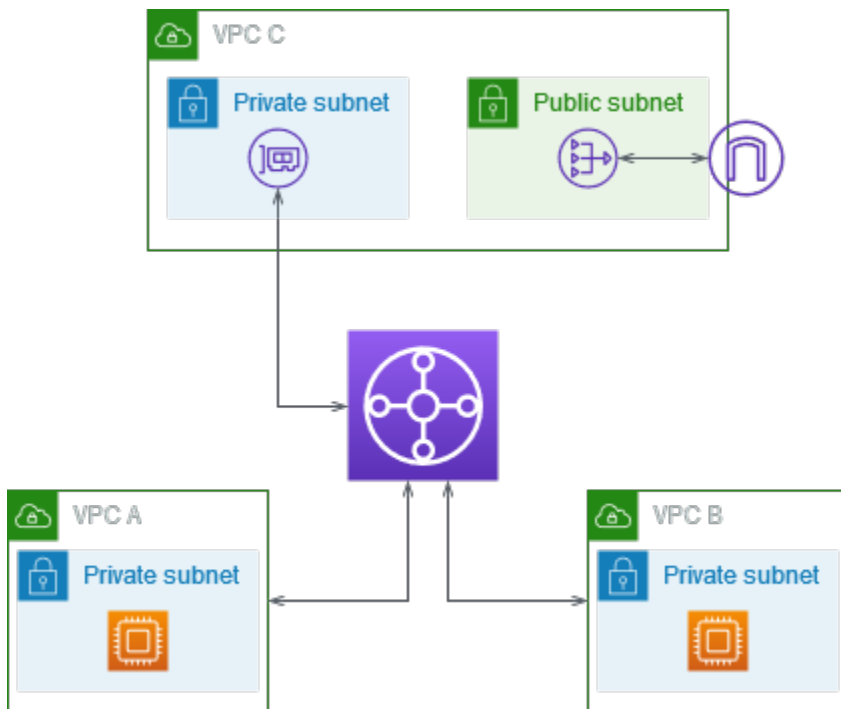
Contents

- [Overview](#)
- [Resources](#)
- [Routing](#)

Overview

The following diagram shows the key components of the configuration for this scenario. You have applications in VPC A and VPC B that need outbound only internet access. You configure VPC C with a public NAT gateway and an internet gateway, and a private subnet for the VPC attachment.

Connect all VPCs to a transit gateway. Configure routing so that outbound internet traffic from VPC A and VPC B traverses the transit gateway to VPC C. The NAT gateway in VPC C routes the traffic to the internet gateway.



Resources

Create the following resources for this scenario:

- Three VPCs with IP address ranges that are neither identical nor overlap. For more information, see [Create a VPC](#) in the *Amazon VPC User Guide*.
- VPC A and VPC B each have private subnets with EC2 instances.
- VPC C has the following:
 - An internet gateway attached to the VPC. For more information, see [Create and attach an internet gateway](#) in the *Amazon VPC User Guide*.
 - A public subnet with a NAT gateway. For more information, see [Create a NAT gateway](#) in the *Amazon VPC User Guide*.
 - A private subnet for the transit gateway attachment. The private subnet should be in the same Availability Zone as the public subnet.
- One transit gateway. For more information, see [the section called “Create a transit gateway”](#).
- Three VPC attachments on the transit gateway. The CIDR blocks for each VPC propagate to the transit gateway route table. For more information, see [the section called “Create a VPC](#)

[attachment](#)". For VPC C, you must create the attachment using the private subnet. If you create the attachment using the public subnet, the instance traffic is routed to the internet gateway, but the internet gateway drops the traffic because the instances don't have public IP addresses. By placing the attachment in the private subnet, the traffic is routed to the NAT gateway, and the NAT gateway sends the traffic to the internet gateway using its Elastic IP address as the source IP address.

Routing

There are route tables for each VPC and a route table for the transit gateway.

Route tables

- [Route table for VPC A](#)
- [Route table for VPC B](#)
- [Route tables for VPC C](#)
- [Transit gateway route table](#)

Route table for VPC A

The following is an example route table. The first entry enables instances in the VPC to communicate with each other. The second entry routes all other IPv4 subnet traffic to the transit gateway.

Destination	Target
<i>VPC A CIDR</i>	local
0.0.0.0/0	<i>transit-gateway-id</i>

Route table for VPC B

The following is an example route table. The first entry enables the instances in the VPC to communicate with each other. The second entry routes all other IPv4 subnet traffic to the transit gateway.

Destination	Target
<i>VPC B CIDR</i>	local
0.0.0.0/0	<i>transit-gateway-id</i>

Route tables for VPC C

Configure the subnet with the NAT gateway as a public subnet by adding a route to the internet gateway. Leave the other subnet as a private subnet.

The following is an example route table for the public subnet. The first entry enables instances in the VPC to communicate with each other. The second and third entries route traffic for VPC A and VPC B to the transit gateway. The remaining entry routes all other IPv4 subnet traffic to the internet gateway.

Destination	Target
<i>VPC C CIDR</i>	local
<i>VPC A CIDR</i>	<i>transit-gateway-id</i>
<i>VPC B CIDR</i>	<i>transit-gateway-id</i>
0.0.0.0/0	<i>internet-gateway-id</i>

The following is an example route table for the private subnet. The first entry enables instances in the VPC to communicate with each other. The second entry routes all other IPv4 subnet traffic to the NAT gateway.

Destination	Target
<i>VPC C CIDR</i>	local
0.0.0.0/0	<i>nat-gateway-id</i>

Transit gateway route table

The following is an example of the transit gateway route table. The CIDR blocks for each VPC propagate to the transit gateway route table. The static route sends outbound internet traffic to VPC C. You can optionally prevent inter-VPC communication by adding a blackhole route for each VPC CIDR.

CIDR	Attachment	Route type
<i>VPC A CIDR</i>	<i>Attachment for VPC A</i>	propagated
<i>VPC B CIDR</i>	<i>Attachment for VPC B</i>	propagated
<i>VPC C CIDR</i>	<i>Attachment for VPC C</i>	propagated
0.0.0.0/0	<i>Attachment for VPC C</i>	static

Example: Appliance in a shared services VPC

You can configure an appliance (such as a security appliance) in a shared services VPC. All traffic that's routed between transit gateway attachments is first inspected by the appliance in the shared services VPC. When appliance mode is enabled, a transit gateway selects a single network interface in the appliance VPC, using a flow hash algorithm, to send traffic to for the life of the flow. The transit gateway uses the same network interface for the return traffic. This ensures that bidirectional traffic is routed symmetrically—it's routed through the same Availability Zone in the VPC attachment for the life of the flow. If you have multiple transit gateways in your architecture, each transit gateway maintains its own session affinity, and each transit gateway can select a different network interface.

You must connect exactly one transit gateway to the appliance VPC to guarantee flow stickiness. Connecting multiple transit gateways to a single appliance VPC does not guarantee flow stickiness because the transit gateways do not share flow state information with each other.

Important

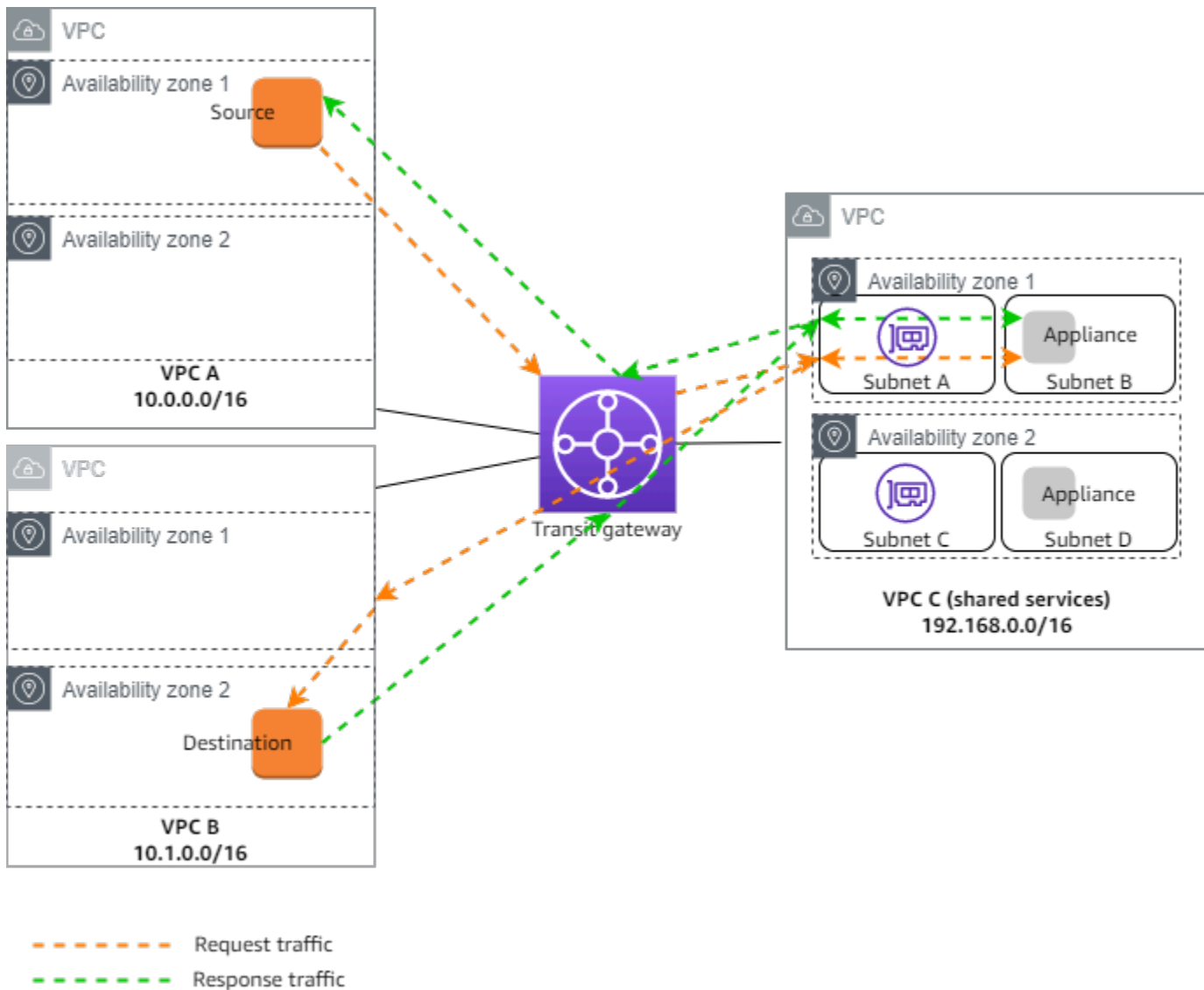
- Traffic in appliance mode is routed correctly as long as the source and destination traffic are coming to a centralized VPC (Inspection VPC) from the same transit gateway attachment. Traffic can drop if the source and destination are on two different transit gateway attachments. Traffic can drop if the centralized VPC receives the traffic from a different gateway — for example, an Internet gateway — and then sends that traffic to the transit gateway attachment after inspection.
- Enabling appliance mode on an existing attachment might affect that attachment's current route as the attachment can flow through any Availability Zone. When appliance mode is not enabled, traffic is kept to the originating Availability Zone.

Contents

- [Overview](#)
- [Stateful appliances and appliance mode](#)
- [Routing](#)

Overview

The following diagram shows the key components of the configuration for this scenario. The transit gateway has three VPC attachments. VPC C is a shared services VPC. Traffic between VPC A and VPC B is routed to the transit gateway, then routed to a security appliance in VPC C for inspection before it's routed to the final destination. The appliance is a stateful appliance, therefore both the request and response traffic is inspected. For high availability, there is an appliance in each Availability Zone in VPC C.



You create the following resources for this scenario:

- Three VPCs. For more information, see [Create a VPC](#) in the *Amazon VPC User Guide*.
- A transit gateway. For more information, see [the section called “Create a transit gateway”](#).
- Three VPC attachments - one for each of the VPCs. For more information, see [the section called “Create a VPC attachment”](#).

For each VPC attachment, specify a subnet in each Availability Zone. For the shared services VPC, these are the subnets where traffic is routed to the VPC from the transit gateway. In the preceding example, these are subnets A and C.

For the VPC attachment for VPC C, enable appliance mode support so that response traffic is routed to the same Availability Zone in VPC C as the source traffic.

The Amazon VPC console supports appliance mode. You can also use the Amazon VPC API, an AWS SDK, the AWS CLI to enable appliance mode, or CloudFormation. For example, add `--options ApplianceModeSupport=enable` to the [create-transit-gateway-vpc-attachment](#) or [modify-transit-gateway-vpc-attachment](#) command.

Note

Flow stickiness in appliance mode is guaranteed only for source and destination traffic that originate towards the Inspection VPC.

Stateful appliances and appliance mode

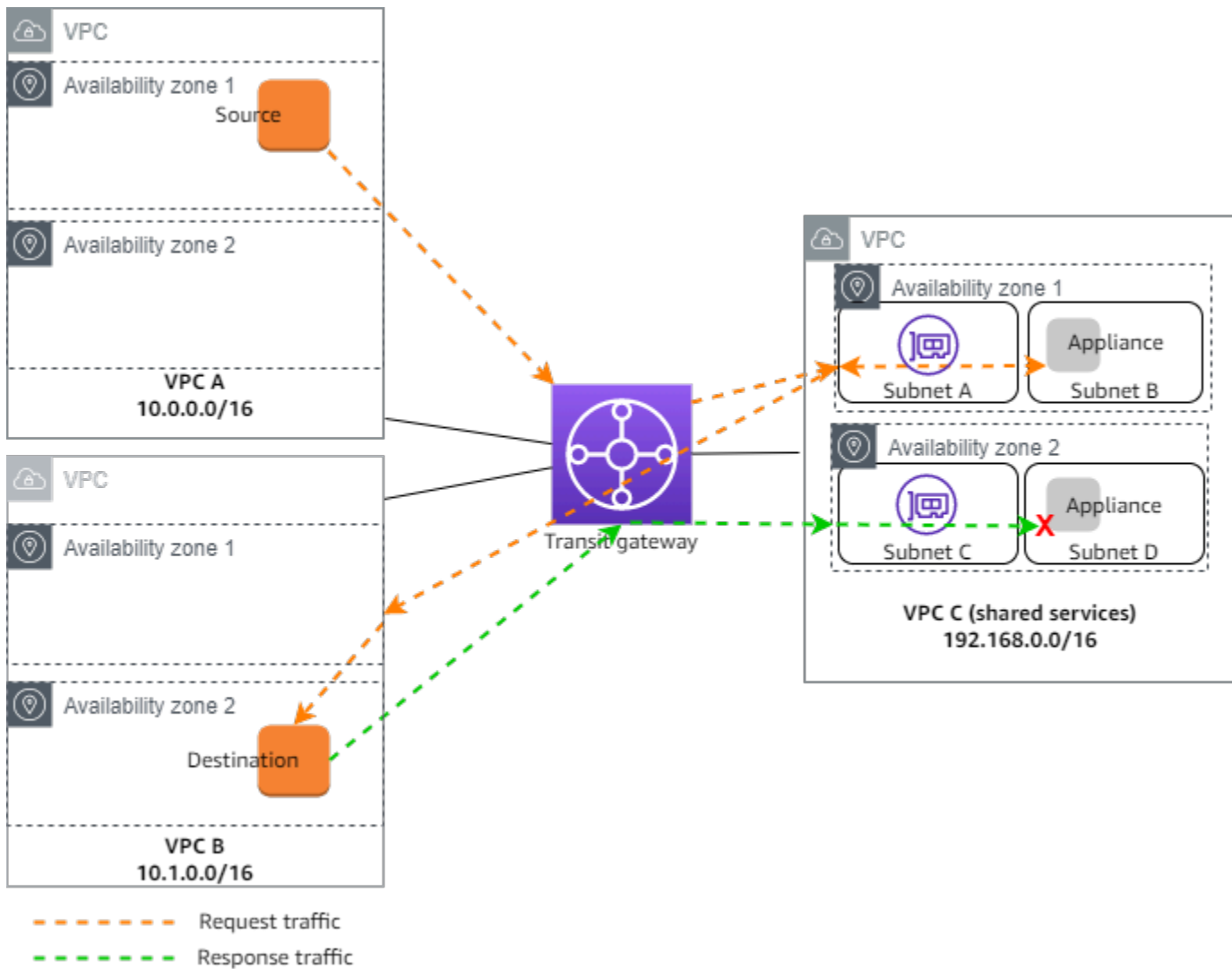
If your VPC attachments span multiple Availability Zones and you require traffic between source and destination hosts to be routed through the same appliance for stateful inspection, enable appliance mode support for the VPC attachment in which the appliance is located.

For more information, see [Centralized inspection architecture](#) in the AWS blog.

Behavior when appliance mode is not enabled

When appliance mode is not enabled, a transit gateway attempts to keep traffic routed between VPC attachments in the originating Availability Zone until it reaches its destination. Traffic crosses Availability Zones between attachments only if there is an Availability Zone failure or if there are no subnets associated with a VPC attachment in that Availability Zone.

The following diagram shows a traffic flow when appliance mode support is not enabled. The response traffic that originates from Availability Zone 2 in VPC B is routed by the transit gateway to the same Availability Zone in VPC C. The traffic is therefore dropped, because the appliance in Availability Zone 2 is not aware of the original request from the source in VPC A.



Routing

Each VPC has one or more route tables and the transit gateway has two route tables.

VPC route tables

VPC A and VPC B

VPCs A and B have route tables with 2 entries. The first entry is the default entry for local IPv4 routing in the VPC. This default entry enables the resources in this VPC to communicate with each other. The second entry routes all other IPv4 subnet traffic to the transit gateway. The following is the route table for VPC A.

Destination	Target
-------------	--------

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	<i>tgw-id</i>

VPC C

The shared services VPC (VPC C) has different route tables for each subnet. Subnet A is used by the transit gateway (you specify this subnet when you create the VPC attachment). The route table for subnet A routes all traffic to the appliance in subnet B.

Destination	Target
192.168.0.0/16	local
0.0.0.0/0	<i>appliance-eni-id</i>

The route table for subnet B (which contains the appliance) routes the traffic back to the transit gateway.

Destination	Target
192.168.0.0/16	local
0.0.0.0/0	<i>tgw-id</i>

Transit gateway route tables

This transit gateway uses one route table for VPC A and VPC B, and one route table for the shared services VPC (VPC C).

The VPC A and VPC B attachments are associated with the following route table. The route table routes all traffic to VPC C.

Destination	Target	Route type
0.0.0.0/0	<i>Attachment ID for VPC C</i>	static

The VPC C attachment is associated with the following route table. It routes traffic to VPC A and VPC B.

Destination	Target	Route type
10.0.0.0/16	<i>Attachment ID for VPC A</i>	propagated
10.1.0.0/16	<i>Attachment ID for VPC B</i>	propagated

Tutorials: Get started with AWS Transit Gateway

The following tutorials help you become familiar with transit gateways in AWS Transit Gateway. The tasks in the following tutorials guide you through creating a transit gateway and then connecting two of your VPCs using that transit gateway. You can create a transit gateway using either the Amazon VPC console or using the AWS CLI.

Tasks

- [Tutorial: Create an AWS Transit Gateway using the Amazon VPC Console](#)
- [Tutorial: Create an AWS Transit Gateway using the AWS command line](#)

Tutorial: Create an AWS Transit Gateway using the Amazon VPC Console

In this tutorial, you'll learn how to use the Amazon VPC Console to create a transit gateway and connect two VPCs to it. You'll create the transit gateway, attach both VPCs, and then configure the necessary routes to enable communication between the transit gateway and your VPCs.

Prerequisites

- To demonstrate a simple example of using a transit gateway, create two VPCs in the same Region. The VPCs can neither have identical nor overlapping CIDRs. Launch one Amazon EC2 instance in each VPC. For more information, see [Create a VPC](#) in the *Amazon VPC User Guide* and [Launch an instance](#) in the *Amazon EC2 User Guide*.
- You can't have identical routes pointing to two different VPCs. A transit gateway does not propagate the CIDRs of a newly attached VPC if an identical route exists in the transit gateway route tables.
- Verify that you have the permissions required to work with transit gateways. For more information, see [Identity and access management in AWS Transit Gateway](#).
- You can't ping between hosts if you haven't added an ICMP rule to each of the host security groups. For more information, see [Configure security group rules](#) in the *Amazon VPC User Guide*.

Steps

- [Step 1: Create the transit gateway](#)

- [Step 2: Attach your VPCs to your transit gateway](#)
- [Step 3: Add routes between the transit gateway and your VPCs](#)
- [Step 4: Test the transit gateway](#)
- [Step 5: Delete the transit gateway](#)

Step 1: Create the transit gateway

When you create a transit gateway, we create a default transit gateway route table and use it as the default association route table and the default propagation route table.

To create a transit gateway

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the Region selector, choose the Region that you used when you created the VPCs.
3. On the navigation pane, choose **Transit Gateways**.
4. Choose **Create transit gateway**.
5. (Optional) For **Name tag**, enter a name for the transit gateway. This creates a tag with "Name" as the key and the name that you specified as the value.
6. (Optional) For **Description**, enter a description for the transit gateway.
7. In **Configure the transit gateway** section, do the following:
 1. For **Amazon side Autonomous System Number (ASN)**, enter the private ASN for your transit gateway. This should be the ASN for the AWS side of a Border Gateway Protocol (BGP) session.

The range is from 64512 to 65534 for 16-bit ASNs.

The range is from 4200000000 to 4294967294 for 32-bit ASNs.

If you have a multi-Region deployment, we recommend that you use a unique ASN for each of your transit gateways.

2. (Optional) Choose whether to enable any of the following:
 - **DNS support** for VPCs attached to this transit gateway.
 - **VPN ECMP** support for VPN connections attached to the transit gateway.
 - **Default route table association**, which automatically associates transit gateway attachments with this transit gateway's default route table.

- **Default route table propagation**, which automatically propagates route table attachments to this transit gateway's default route table.
 - **Multicast support**, which allows you to create multicast domains in this transit gateway.
8. (Optional) In the **Configure-cross-account sharing options** section, choose whether to **Auto accept shared attachments**. If enabled, attachments are automatically accepted. Otherwise, you must accept or reject attachment requests.
 9. (Optional) In the **Transit gateway CIDR blocks section**, add a size /24 CIDR block or larger for IPv4 addresses or /64 block or larger CIDR block for IPv6 addresses. You can associate any public or private IP address range, except for addresses in the 169.254.0.0/16 range, and ranges that overlap with the addresses for your VPC attachments and on-premises networks.
-  **Note**

Transit gateway CIDR blocks are used if you are configuring Connect (GRE) attachments or PrivateIP VPNs. Transit Gateway assigns IPs for the Tunnel endpoints (GRE/PrivateIP VPN) from this range.
10. (Optional) Add key-value tags to this transit gateway to further help identify it.
 1. Choose **Add new tag**.
 2. Enter a **Key** name and associated **Value**.
 3. Choose **Add new tag** to add additional tags, or skip to the next step.
 11. Choose **Create transit gateway**. When the gateway is created, the initial state of the transit gateway is pending.

Step 2: Attach your VPCs to your transit gateway

Wait until the transit gateway you created in the previous section shows as available before proceeding with creating an attachment. Create an attachment for each VPC.

Confirm that you have created two VPCs and launched an EC2 instance in each, as described in [Prerequisites](#).

Create a transit gateway attachment to a VPC

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.

3. Choose **Create transit gateway attachment**.
4. (Optional) For **Name tag**, enter a name for the attachment.
5. For **Transit gateway ID**, choose the transit gateway to use for the attachment.
6. For **Attachment type**, choose **VPC**.
7. Choose whether to enable **DNS support**. For this exercise, do not enable **IPv6 support**.
8. For **VPC ID**, choose the VPC to attach to the transit gateway.
9. For **Subnet IDs**, select one subnet for each Availability Zone to be used by the transit gateway to route traffic. You must select at least one subnet. You can select only one subnet per Availability Zone.
10. Choose **Create transit gateway attachment**.

Each attachment is always associated with exactly one route table. Route tables can be associated with zero to many attachments. To determine the routes to configure, decide on the use case for your transit gateway, and then configure the routes. For more information, see [the section called "Example transit gateway scenarios"](#).

Step 3: Add routes between the transit gateway and your VPCs

A route table includes dynamic and static routes that determine the next hop for associated VPCs based on the destination IP address of the packet. Configure a route that has a destination for non-local routes and the target of the transit gateway attachment ID. For more information, see [Routing for a transit gateway](#) in the *Amazon VPC User Guide*.

To add a route to a VPC route table

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Route Tables**.
3. Choose the route table associated with your VPC.
4. Choose the **Routes** tab, then choose **Edit routes**.
5. Choose **Add route**.
6. In the **Destination** column, enter the destination IP address range. For **Target**, choose **Transit Gateway**, and then choose the transit gateway ID.
7. Choose **Save changes**.

Step 4: Test the transit gateway

You can confirm that the transit gateway was successfully created by connecting to an Amazon EC2 instance in each VPC, and then sending data between them, such as a ping command. For more information, see [Connect to your EC2 instance](#) in the *Amazon EC2 User Guide*.

Step 5: Delete the transit gateway

When you no longer need a transit gateway, you can delete it.

You cannot delete a transit gateway that has resource attachments. If you try to delete a transit gateway with attachments, you'll be prompted to first delete those attachments before you can delete the transit gateway. As soon as the transit gateway is deleted, you stop incurring charges for it.

To delete your transit gateway

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateways**.
3. Select the transit gateway, and then choose **Actions, Delete transit gateway**.
4. Enter **delete** and choose **Delete**.

The **State** of the transit gateway on the **Transit gateways** page is **Deleting**. Once deleted the transit gateway is removed from the page.

Tutorial: Create an AWS Transit Gateway using the AWS command line

In this tutorial, you'll learn how to use the AWS CLI to create a transit gateway and connect two VPCs to it. You'll create the transit gateway, attach both VPCs, and then configure the necessary routes to enable communication between the transit gateway and your VPCs.

Prerequisites

Before you begin, make sure you have:

- AWS CLI installed and configured with appropriate permissions. If you don't have the AWS CLI installed, see the *AWS Command Line Interface Documentation*.

- The VPCs can neither have identical nor overlapping CIDRs. For more information, see [Create a VPC](#) in the *Amazon VPC User Guide*.
- One EC2 instance in each VPC. For the steps to launch an EC2 instance into a VPC, see [Launch an instance](#) in the *Amazon EC2 User Guide*.
- Security groups configured to allow ICMP traffic between the instances. For the steps to control traffic using security groups, see [Control traffic to your AWS resources using security groups](#) in the *Amazon VPC User Guide*.
- Appropriate IAM permissions to work with transit gateways. To check transit gateway IAM permissions, see [Identity and access management in AWS Transit Gateways](#) in the *AWS Transit Gateway Guide*.

Steps

- [Step 1: Create the transit gateway](#)
- [Step 2: Verify the transit gateway availability state](#)
- [Step 3: Attach your VPCs to your transit gateway](#)
- [Step 4: Verify that the transit gateway attachments are available](#)
- [Step 5: Add routes between your transit gateway and VPCs](#)
- [Step 6: Test the transit gateway](#)
- [Step 7: Delete the transit gateway attachments and transit gateway](#)
- [Conclusion](#)

Step 1: Create the transit gateway

When you create a transit gateway, AWS creates a default transit gateway route table and uses it as the default association route table and the default propagation route table. The following shows an example create-transit-gateway request in the us-west-2 Region. Additional options were passed in the request. For more information about the create-transit-gateway command, including a list of the options you can pass in the request, see [create-transit-gateway](#).

```
aws ec2 create-transit-gateway \  
  --description "My Transit Gateway" \  
  --region us-west-2
```

The response then shows that the transit gateway was created. In the response, the Options that are returned are all default values.

```
{
  "TransitGateway": {
    "TransitGatewayId": "tgw-1234567890abcdef0",
    "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/tgw-1234567890abcdef0",
    "State": "pending",
    "OwnerId": "123456789012",
    "Description": "My Transit Gateway",
    "CreationTime": "2025-06-23T17:39:33+00:00",
    "Options": {
      "AmazonSideAsn": 64512,
      "AutoAcceptSharedAttachments": "disable",
      "DefaultRouteTableAssociation": "enable",
      "AssociationDefaultRouteTableId": "tgw-rtb-abcdef1234567890a",
      "DefaultRouteTablePropagation": "enable",
      "PropagationDefaultRouteTableId": "tgw-rtb-abcdef1234567890a",
      "VpnEcmpSupport": "enable",
      "DnsSupport": "enable",
      "SecurityGroupReferencingSupport": "disable",
      "MulticastSupport": "disable"
    }
  }
}
```

Note

This command returns information about your new transit gateway, including its ID. Make note of the transit gateway ID (tgw-1234567890abcdef0) as you'll need it in subsequent steps.

Step 2: Verify the transit gateway availability state

When you create a transit gateway, it's placed in a pending state. The state will change from pending to available automatically, but until it does you can't attach any VPCs until the state changes. To verify the state, run the `describe-transit-gateways` command using the newly created transit gateway ID along with the `filters` option. The `filters` option uses `Name=state` and `Values=available` pairs. The command then searches to verify if the state of your transit

gateway is in an available state. If it is, the response shows "State": "available". If it's in any other state then it is not yet available for use. Wait several minutes before running the command.

For more information about the `describe-transit-gateways` command, see [describe-transit-gateways](#).

```
aws ec2 describe-transit-gateways \
  --transit-gateway-ids tgw-1234567890abcdef0 \
  --filters Name=state,Values=available
```

Wait until the transit gateway state changes from pending to available before proceeding. In the following response, the State has changed to available.

```
{
  "TransitGateways": [
    {
      "TransitGatewayId": "tgw-1234567890abcdef0",
      "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/tgw-1234567890abcdef0",
      "State": "available",
      "OwnerId": "123456789012",
      "Description": "My Transit Gateway",
      "CreationTime": "2022-04-20T19:58:25+00:00",
      "Options": {
        "AmazonSideAsn": 64512,
        "AutoAcceptSharedAttachments": "disable",
        "DefaultRouteTableAssociation": "enable",
        "AssociationDefaultRouteTableId": "tgw-rtb-abcdef1234567890a",
        "DefaultRouteTablePropagation": "enable",
        "PropagationDefaultRouteTableId": "tgw-rtb-abcdef1234567890a",
        "VpnEcmpSupport": "enable",
        "DnsSupport": "enable",
        "SecurityGroupReferencingSupport": "disable",
        "MulticastSupport": "disable"
      },
      "Tags": [
        {
          "Key": "Name",
          "Value": "example-transit-gateway"
        }
      ]
    }
  ]
}
```

```
}
```

Step 3: Attach your VPCs to your transit gateway

Once your transit gateway is available, create an attachment for each VPC using the `create-transit-gateway-vpc-attachment`. You'll need to include the `transit-gateway-id`, the `vpc-id`, and the `subnet-ids`.

For more information about the `create-transit-vpc attachment` command, see [create-transit-gateway-vpc-attachment](#).

In the following example, the command is run twice, once for each VPC.

For the first VPC run the following using the first `vpc_id` and `subnet-ids`:

```
aws ec2 create-transit-gateway-vpc-attachment \
  --transit-gateway-id tgw-1234567890abcdef0 \
  --vpc-id vpc-1234567890abcdef0 \
  --subnet-ids subnet-1234567890abcdef0
```

The response shows the successful attachment. The attachment is created in a `pending` state. There's no need to change this state as it changes to an `available` state automatically. This might take several minutes.

```
{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-1234567890abcdef0",
    "TransitGatewayId": "tgw-1234567890abcdef0",
    "VpcId": "vpc-1234567890abcdef0",
    "VpcOwnerId": "123456789012",
    "State": "pending",
    "SubnetIds": [
      "subnet-1234567890abcdef0",
      "subnet-abcdef1234567890"
    ],
    "CreationTime": "2025-06-23T18:35:11+00:00",
    "Options": {
      "DnsSupport": "enable",
      "SecurityGroupReferencingSupport": "enable",
      "Ipv6Support": "disable",
      "ApplianceModeSupport": "disable"
    }
  }
}
```

```
    }
  }
}
```

For the second VPC, run the same command as above using the second `vpc_id` and `subnet-ids`:

```
aws ec2 create-transit-gateway-vpc-attachment \
  --transit-gateway-id tgw-1234567890abcdef0 \
  --vpc-id vpc-abcdef1234567890 \
  --subnet-ids subnet-abcdef01234567890
```

The response for this command also shows a successful attachment, with the attachment currently in a pending state.

```
{
  {
    "TransitGatewayVpcAttachment": {
      "TransitGatewayAttachmentId": "tgw-attach-abcdef1234567890",
      "TransitGatewayId": "tgw-1234567890abcdef0",
      "VpcId": "vpc-abcdef1234567890",
      "VpcOwnerId": "123456789012",
      "State": "pending",
      "SubnetIds": [
        "subnet-fedcba0987654321",
        "subnet-0987654321fedcba"
      ],
      "CreationTime": "2025-06-23T18:42:56+00:00",
      "Options": {
        "DnsSupport": "enable",
        "SecurityGroupReferencingSupport": "enable",
        "Ipv6Support": "disable",
        "ApplianceModeSupport": "disable"
      }
    }
  }
}
```

Step 4: Verify that the transit gateway attachments are available

Transit gateway attachments are created in an initial pending state. You won't be able to use these attachments in your routes until the state changes to available. This happens automatically. Use the `describe-transit-gateways` command, along with the `transit-gateway-id`, to

check the State. For more information about the `describe-transit-gateways` command, see [describe-transit-gateways](#).

Run the following command to check the status. In this example, optional Name and Values filters fields are passed in the request:

```
aws ec2 describe-transit-gateway-vpc-attachments \
  --filters Name=transit-gateway-id,Values=tgw-1234567890abcdef0
```

The following response shows that both attachments in an available state:

```
{
  "TransitGatewayVpcAttachments": [
    {
      "TransitGatewayAttachmentId": "tgw-attach-1234567890abcdef0",
      "TransitGatewayId": "tgw-1234567890abcdef0",
      "VpcId": "vpc-1234567890abcdef0",
      "VpcOwnerId": "123456789012",
      "State": "available",
      "SubnetIds": [
        "subnet-1234567890abcdef0",
        "subnet-abcdef1234567890"
      ],
      "CreationTime": "2025-06-23T18:35:11+00:00",
      "Options": {
        "DnsSupport": "enable",
        "SecurityGroupReferencingSupport": "enable",
        "Ipv6Support": "disable",
        "ApplianceModeSupport": "disable"
      },
      "Tags": []
    },
    {
      "TransitGatewayAttachmentId": "tgw-attach-abcdef1234567890",
      "TransitGatewayId": "tgw-1234567890abcdef0",
      "VpcId": "vpc-abcdef1234567890",
      "VpcOwnerId": "123456789012",
      "State": "available",
      "SubnetIds": [
        "subnet-fedcba0987654321",
        "subnet-0987654321fedcba"
      ],
      "CreationTime": "2025-06-23T18:42:56+00:00",
```

```

        "Options": {
            "DnsSupport": "enable",
            "SecurityGroupReferencingSupport": "enable",
            "Ipv6Support": "disable",
            "ApplianceModeSupport": "disable"
        },
        "Tags": []
    }
]
}

```

Step 5: Add routes between your transit gateway and VPCs

Configure routes in each VPC's route table to direct traffic to the other VPC through the transit gateway using the `create-route` command along with the `transit-gateway-id` for each VPC route table. In the following example, the command is run twice, once for each route table. The request includes the `route-table-id`, the `destination-cidr-block`, and `transit-gateway-id` for each VPC route you're creating.

For more information about `create-route` command, see [create-route](#).

For the first VPC's route table run the following command:

```

aws ec2 create-route \
  --route-table-id rtb-1234567890abcdef0 \
  --destination-cidr-block 10.2.0.0/16 \
  --transit-gateway-id tgw-1234567890abcdef0

```

For the second VPC's route table run the following command. This route uses a `route-table-id` and `destination-cidr-block` different from the first VPC. However, since you're only using a single transit gateway, the same `transit-gateway-id` is used.

```

aws ec2 create-route \
  --route-table-id rtb-abcdef1234567890 \
  --destination-cidr-block 10.1.0.0/16 \
  --transit-gateway-id tgw-1234567890abcdef0

```

The response returns `true` for each route, indicating the routes were created.

```
{
```

```
"Return": true  
}
```

Note

Replace the destination CIDR blocks with the actual CIDR blocks of your VPCs.

Step 6: Test the transit gateway

You can confirm that the transit gateway was successfully created by connecting to an EC2 instance in one VPC and pinging an instance in the other VPC, and then running the ping command.

1. Connect to your EC2 instance in the first VPC using SSH or EC2 Instance Connect
2. Ping the private IP address of the EC2 instance in the second VPC:

```
ping 10.2.0.50
```

Note

Replace `10.2.0.50` with the actual private IP address of your EC2 instance in the second VPC.

If the ping is successful, your transit gateway is correctly configured and routing traffic between your VPCs.

Step 7: Delete the transit gateway attachments and transit gateway

When you no longer need the transit gateway, you can delete it. First, you must delete all attachments. Run the `delete-transit-gateway-vpc-attachment` command, using the `transit-gateway-attachment-id` for each attachment. After running the command, use `delete-transit-gateway` to delete the transit gateway. For the following, delete the two VPC attachments and the single transit gateway that were created in the previous steps.

Important

You'll stop incurring charges once you delete all of the transit gateway attachments.

1. Delete the VPC attachments using the `delete-transit-gateway-vpc-attachment` command. For more information about `delete-transit-gateway-vpc-attachment` command, see [delete-transit-gateway-vpc-attachment](#).

For the first attachment, run the following command:

```
aws ec2 delete-transit-gateway-vpc-attachment \  
  --transit-gateway-attachment-id tgw-attach-1234567890abcdef0
```

The delete response for the first VPC attachment returns the following:

```
{  
  "TransitGatewayVpcAttachment": {  
    "TransitGatewayAttachmentId": "tgw-attach-1234567890abcdef0",  
    "TransitGatewayId": "tgw-1234567890abcdef0",  
    "VpcId": "vpc-abcdef1234567890",  
    "VpcOwnerId": "123456789012",  
    "State": "deleting",  
    "CreationTime": "2025-06-23T18:42:56+00:00"  
  }  
}
```

Run the `delete-transit-gateway-vpc-attachment` command for the second attachment:

```
aws ec2 delete-transit-gateway-vpc-attachment \  
  --transit-gateway-attachment-id tgw-attach-abcdef1234567890
```

The delete response for the second VPC attachment returns the following:

```
The response returns:  
{  
  "TransitGatewayVpcAttachment": {  
    "TransitGatewayAttachmentId": "tgw-attach-abcdef1234567890",  
    "TransitGatewayId": "tgw-1234567890abcdef0",  
    "VpcId": "vpc-abcdef1234567890",  
    "VpcOwnerId": "123456789012",  
    "State": "deleting",  
    "CreationTime": "2025-06-23T18:42:56+00:00"  
  }  
}
```

```
}
```

2. Attachments are in a deleting state until they're deleted. Once deleted, you can then delete the transit gateway. Use the `delete-transit-gateway` command along with the `transit-gateway-id`. For more information about `delete-transit-gateway` command, see [delete-transit-gateway](#).

The following example deletes My Transit Gateway which you created in the first step above:

```
aws ec2 delete-transit-gateway \  
  --transit-gateway-id tgw-1234567890abcdef0
```

The following shows the response to the request, which includes the deleted transit gateway ID and name, along with the original options set for the transit gateway when it was created.

```
{  
  "TransitGateway": {  
    "TransitGatewayId": "tgw-1234567890abcdef0",  
    "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/  
tgw-1234567890abcdef0",  
    "State": "deleting",  
    "OwnerId": "123456789012",  
    "Description": "My Transit Gateway",  
    "CreationTime": "2025-06-23T17:39:33+00:00",  
    "Options": {  
      "AmazonSideAsn": 64512,  
      "AutoAcceptSharedAttachments": "disable",  
      "DefaultRouteTableAssociation": "enable",  
      "AssociationDefaultRouteTableId": "tgw-rtb-abcdef1234567890a",  
      "DefaultRouteTablePropagation": "enable",  
      "PropagationDefaultRouteTableId": "tgw-rtb-abcdef1234567890a",  
      "VpnEcmpSupport": "enable",  
      "DnsSupport": "enable",  
      "SecurityGroupReferencingSupport": "disable",  
      "MulticastSupport": "disable"  
    },  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "example-transit-gateway"  
      }  
    ]  
  }  
}
```

```
}  
  }  
}
```

Conclusion

You've successfully created a transit gateway, attached two VPCs to it, configured routing between them, and verified connectivity. This simple example demonstrates the basic functionality of AWS Transit Gateways. For more complex scenarios, such as connecting to on-premises networks or implementing more advanced routing configurations, see the [AWS Transit Gateways Guide](#).

AWS Transit Gateway design best practices

The following are best practices for your transit gateway design:

- Use a separate subnet for each transit gateway VPC attachment. For each subnet, use a small CIDR, for example /28, so that you have more addresses for EC2 resources. When you use a separate subnet, you can configure the following:
 - Keep the inbound and outbound network ACLs associated with the transit gateway subnets open.
 - Depending on your traffic flow, you can apply network ACLs to your workload subnets.
- Create one network ACL and associate it with all of the subnets that are associated with the transit gateway. Keep the network ACL open in both the inbound and outbound directions.
- Associate the same VPC route table with all of the subnets that are associated with the transit gateway, unless your network design requires multiple VPC route tables (for example, a middle-box VPC that routes traffic through multiple NAT gateways).
- Use Border Gateway Protocol (BGP) Site-to-Site VPN connections. If your customer gateway device or firewall for the connection supports multipath, enable the feature.
- Enable route propagation for Direct Connect gateway attachments and BGP Site-to-Site VPN attachments.
- When migrating from VPC peering to use a transit gateway. An MTU size mismatch between VPC peering and the transit gateway might result in some packets dropping for asymmetric traffic. Update both VPCs at the same time to avoid jumbo packets dropping due to size mismatches.
- You do not need additional transit gateways for high availability, because transit gateways are highly available by design.
- Limit the number of transit gateway route tables unless your design requires multiple transit gateway route tables.
- For redundancy, use a single transit gateway in each Region for disaster recovery.
- For deployments with multiple transit gateways, we recommend that you use a unique Autonomous System Number (ASN) for each of your transit gateways. You can also use inter-Region peering. For more information, see [Building a global network using AWS Transit Gateway Inter-Region peering](#).

Work with AWS Transit Gateway

You can work with transit gateways using the Amazon VPC console or the AWS CLI. For information about enabling and managing Encryption support for your transit gateway, see [the section called “Encryption Support”](#).

Topics

- [Shared transit gateways](#)
- [Transit gateways in AWS Transit Gateway](#)
- [Amazon VPC attachments in AWS Transit Gateway](#)
- [AWS Transit Gateway network function attachments](#)
- [AWS Site-to-Site VPN attachments in AWS Transit Gateway](#)
- [VPN Concentrator attachments in AWS Transit Gateway](#)
- [Client VPN attachments in AWS Transit Gateway](#)
- [Transit gateway attachments to a Direct Connect gateway in AWS Transit Gateway](#)
- [Transit gateway peering attachments in AWS Transit Gateway](#)
- [Connect attachments and Connect peers in AWS Transit Gateway](#)
- [Transit gateway route tables in AWS Transit Gateway](#)
- [Transit gateway policy tables in AWS Transit Gateway](#)
- [Multicast in AWS Transit Gateway](#)
- [Flexible cost allocation](#)

Shared transit gateways

You can use AWS Resource Access Manager (RAM) to share a transit gateway for VPC attachments across accounts or across your organization in AWS Organizations. RAM must be enabled and resources shared with an organization. For more information, see [Enable resource sharing with AWS Organizations](#) in the *AWS RAM User Guide*.

Considerations

Take the following into account when you want to share a transit gateway.

- An AWS Site-to-Site VPN attachment must be created in the same AWS account that owns the transit gateway.
- An attachment to a Direct Connect gateway uses a transit gateway association and can be in the same AWS account as the Direct Connect gateway, or a different one from the Direct Connect gateway.

By default, users do not have permission to create or modify AWS RAM resources. To allow users to create or modify resources and perform tasks, you must create IAM policies that grant permission to use specific resources and API actions. You then attach those policies to the IAM users or groups that require those permissions.

Only the resource owner can perform the following operations:

- Create a resource share.
- Update a resource share.
- View a resource share.
- View the resources that are shared by your account, across all resource shares.
- View the principals with whom you are sharing your resources, across all resource shares. Viewing the principals with whom you are sharing enables you to determine who has access to your shared resources.
- Delete a resource share.
- Run all transit gateway, transit gateway attachment, and transit gateway route tables APIs.

You can perform the following operations on resources that are shared with you:

- Accept, or reject a resource share invitation.
- View a resource share.
- View the shared resources that you can access.
- View a list of all the principals that are sharing resources with you. You can see which resources and resource shares they have shared with you.
- Can run the `DescribeTransitGateways` API.
- Run the APIs that create and describe attachments, for example `CreateTransitGatewayVpcAttachment` and `DescribeTransitGatewayVpcAttachments`, in their VPCs.

- Leave a resource share.

When a transit gateway is shared with you, you cannot create, modify, or delete its transit gateway route tables, or its transit gateway route table propagations and associations.

When you create a transit gateway, the transit gateway, is created in the Availability Zone that is mapped to your account and is independent from other accounts. When the transit gateway and the attachment entities are in different accounts, use the Availability Zone ID to uniquely and consistently identify the Availability Zone. For example, use `us-east-1-az1` is an AZ ID for the us-east-1 Region and maps to the same location in every AWS account.

Unshare a transit gateway

When the share owner unshares the transit gateway, the following rules apply:

- The transit gateway attachment remains functional.
- The shared account can not describe the transit gateway.
- The transit gateway owner, and the share owner can delete the transit gateway attachment.

When a transit gateway is unshared with another AWS account, or if the AWS account that the transit gateway is shared with is removed from the organization, the transit gateway itself won't be impacted.

Shared subnets

A VPC owner can attach a transit gateway to a shared VPC subnet. Participants cannot. The traffic from participant's resources can use the attachments depending on the routes set up on the shared VPC subnet by the VPC owner.

For more information, see [Share your VPC with other accounts](#) in the *Amazon VPC User Guide*.

Transit gateways in AWS Transit Gateway

A transit gateway enables you to attach VPCs and VPN connections and route traffic between them. A transit gateway works across AWS accounts, and you can use AWS RAM to share your transit gateway with other accounts. After you share a transit gateway with another AWS account,

the account owner can attach their VPCs to your transit gateway. A user from either account can delete the attachment at any time.

You can enable multicast on a transit gateway, and then create a transit gateway multicast domain that allows multicast traffic to be sent from your multicast source to multicast group members over VPC attachments that you associate with the domain.

Each VPC or VPN attachment is associated with a single route table. That route table decides the next hop for the traffic coming from that resource attachment. A route table inside the transit gateway allows for both IPv4 or IPv6 CIDRs and targets. The targets are VPCs and VPN connections. When you attach a VPC or create a VPN connection on a transit gateway, the attachment is associated with the default route table of the transit gateway.

You can create additional route tables inside the transit gateway, and change the VPC or VPN association to these route tables. This enables you to segment your network. For example, you can associate development VPCs with one route table and production VPCs with a different route table. This enables you to create isolated networks inside a transit gateway similar to virtual routing and forwarding (VRFs) in traditional networks.

Transit gateways support dynamic and static routing between attached VPCs and VPN connections. You can enable or disable route propagation for each attachment. VPN Concentrator attachments support BGP (dynamic) routing only. Transit gateway peering attachments support static routing only. You can point routes in transit gateway route tables to the peering attachment for routing traffic between the peered transit gateways.

You can optionally associate one or more IPv4 or IPv6 CIDR blocks with your transit gateway. You specify an IP address from the CIDR block when you establish a Transit Gateway Connect peer for a [Transit Gateway Connect attachment](#). You can associate any public or private IP address range, except for addresses in the 169.254.0.0/16 range, and ranges that overlap with addresses for your VPC attachments and on-premises networks. For more information about IPv4 and IPv6 CIDR blocks, see [IP addressing](#) in the *Amazon VPC User Guide*.

Tasks

- [Create a transit gateway in AWS Transit Gateway](#)
- [View transit gateway information in AWS Transit Gateway](#)
- [Manage transit gateway tags in AWS Transit Gateway](#)
- [Modify a transit gateway in AWS Transit Gateway](#)

- [Accept an AWS Transit Gateway resource share using the AWS Resource Access Manager console](#)
- [Accept a shared attachment in AWS Transit Gateway](#)
- [Delete a transit gateway in AWS Transit Gateway](#)
- [Encryption Support for AWS Transit Gateway](#)

Create a transit gateway in AWS Transit Gateway

When you create a transit gateway, we create a default transit gateway route table and use it as the default association route table and the default propagation route table. If you choose not to create the default transit gateway route table, you can create one later on. For more information about routes and route tables, see [???](#).

Note

If you want to enable Encryption support on a transit gateway, you can't enable it while creating the gateway. After you create the transit gateway, and it's in the available state, you can then modify it to enable Encryption support. For more information, see [the section called "Encryption Support"](#).

To create a transit gateway using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateways**.
3. Choose **Create transit gateway**.
4. For **Name tag**, optionally enter a name for the transit gateway. A name tag can make it easier to identify a specific gateway from the list of gateways. When you add a **Name tag**, a tag is created with a key of **Name** and with a value equal to the value you enter.
5. For **Description**, optionally enter a description for the transit gateway.
6. For **Amazon side Autonomous System Number (ASN)**, either leave the default value to use the default ASN or enter the private ASN for your transit gateway. This should be the ASN for the AWS side of a Border Gateway Protocol (BGP) session.

The range is 64512 to 65534 for 16-bit ASNs.

The range is 4200000000 to 4294967294 for 32-bit ASNs.

If you have a multi-Region deployment, we recommend that you use a unique ASN for each of your transit gateways.

7. For **DNS support**, select this option if you need the VPC to resolve public IPv4 DNS host names to private IPv4 addresses when queried from instances in another VPC attached to the transit gateway.
8. For **Security Group Referencing support**, enable this feature to reference a security group across VPCs attached to a transit gateway. For more information about security group referencing see [the section called "Security group referencing"](#).
9. For **VPN ECMP support**, select this option if you need Equal Cost Multipath (ECMP) routing support between VPN tunnels. If connections advertise the same CIDRs, the traffic is distributed equally between them.

When you select this option, the advertised BGP ASN, then the BGP attributes such as the AS-path, must be the same.

 Note

To use ECMP, you must create a VPN connection that uses dynamic routing. VPN connections that use static routing do not support ECMP.

10. For **Default route table association**, select this option to automatically associate transit gateway attachments with the default route table for the transit gateway.
11. For **Default route table propagation**, select this option to automatically propagate transit gateway attachments to the default route table for the transit gateway.
12. (Optional) To use the transit gateway as a router for multicast traffic, select **Multicast support**.
13. (Optional) In the **Configure-cross-account sharing options** section, choose whether to **Auto accept shared attachments**. If enabled, attachments are automatically accepted. Otherwise, you must accept or reject attachment requests.

For **Auto accept shared attachments**, select this option to automatically accept cross-account attachments.

14. (Optional) For **Transit gateway CIDR blocks**, specify one or more IPv4 or IPv6 CIDR blocks for your transit gateway.

You can specify a size /24 CIDR block or larger (for example, /23 or /22) for IPv4, or a size /64 CIDR block or larger (for example, /63 or /62) for IPv6. You can associate any public or private

IP address range, except for addresses in the 169.254.0.0/16 range, and ranges that overlap with the addresses for your VPC attachments and on-premises networks.

 Note

Transit gateway CIDR blocks are used if you are configuring Connect (GRE) attachments, PrivateIP VPNs, or Client VPN attachments. Transit Gateway assigns IPs for the Tunnel endpoints (GRE/PrivateIP VPN) and Client VPN attachments from this range.

15. Choose **Create transit gateway**.

To create a transit gateway using the AWS CLI

Use the [create-transit-gateway](#) command.

View transit gateway information in AWS Transit Gateway

View any of your transit gateways.

To view a transit gateway using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateways**. Details for the transit gateway are displayed below the list of gateways on the page.

To view a transit gateway using the AWS CLI

Use the [describe-transit-gateways](#) command.

Manage transit gateway tags in AWS Transit Gateway

Add tags to your resources to help organize and identify them, such as by purpose, owner, or environment. You can add multiple tags to each transit gateway. Tag keys must be unique for each transit gateway. If you add a tag with a key that is already associated with the transit gateway, it updates the value of that tag. For more information, see [Tagging your Amazon EC2 Resources](#).

Add tags to a transit gateway using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

2. On the navigation pane, choose **Transit Gateways**.
3. Choose the transit gateway that you want to add or edit tags for.
4. Choose the **Tags** tab in the lower part of the page.
5. Choose **Manage tags**.
6. Choose **Add new tag**.
7. Enter a **Key** and **Value** for the tag.
8. Choose **Save**.

Modify a transit gateway in AWS Transit Gateway

You can modify the configuration options for a transit gateway. When you modify a transit gateway, any existing transit gateway attachments don't experience any service interruptions.

You cannot modify a transit gateway that has been shared with you.

You cannot remove a CIDR block for the transit gateway if any of the IP addresses are currently used for a [Connect peer](#).

Note

Transit gateways that have Encryption Support enabled can be attached to VPCs with Encryption Controls in monitor or Enforce mode, or to VPCs that don't have Encryption Controls enabled. VPCs that have Encryption Controls in Enforce mode can ONLY be attached to Transit Gateways that have Encryption Support enabled. For more detailed information, see [the section called "Encryption Support"](#).

To modify a transit gateway

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateways**.
3. Choose the transit gateway to modify.
4. Choose **Actions, Modify transit gateway**.
5. Modify the options as needed, and choose **Modify transit gateway**.

To modify your transit gateway using the AWS CLI

Use the [modify-transit-gateway](#) command.

Accept an AWS Transit Gateway resource share using the AWS Resource Access Manager console

If you were added to a resource share, you receive an invitation to join the resource share. You must accept the resource share through the AWS Resource Access Manager (AWS RAM) console before you can access the shared resources.

To accept a resource share

1. Open the AWS RAM console at <https://console.aws.amazon.com/ram/>.
2. In the navigation pane, choose **Shared with me, Resource shares**.
3. Select the resource share.
4. Choose **Accept resource share**.
5. To view the shared transit gateway, open the **Transit Gateways** page in the Amazon VPC console.

Accept a shared attachment in AWS Transit Gateway

If you didn't enable the **Auto accept shared attachments** functionality when you created your transit gateway, you must manually accept cross-account (shared) attachment using either the Amazon VPC Console or the AWS CLI.

To manually accept a shared attachment

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Select the transit gateway attachment that's pending acceptance.
4. Choose **Actions, Accept transit gateway attachment**.

To accept a shared attachment using the AWS CLI

Use the [accept-transit-gateway-vpc-attachment](#) command.

Delete a transit gateway in AWS Transit Gateway

You can't delete a transit gateway with existing attachments. You need to delete all attachments before you can delete a transit gateway.

To delete a transit gateway using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. Choose the transit gateway to delete.
3. Choose **Actions, Delete transit gateway**. Enter **delete** and choose **Delete** to confirm the deletion.

To delete a transit gateway using the AWS CLI

Use the [delete-transit-gateway](#) command.

Encryption Support for AWS Transit Gateway

Encryption Controls allows you to audit the encryption status of the traffic flows in your VPC and then enforce encryption-in-transit for all traffic within the VPC. When VPC Encryption Control is in enforce mode, all Elastic Network Interfaces (ENI) in that VPC will be restricted to attach only to AWS Nitro encryption capable instances; and only AWS services that encrypt data in transit will be allowed to attach to Encryption Controls enforced VPC. For more information on VPC Encryption Controls, please refer to this [documentation](#).

Transit Gateway Encryption Support and VPC Encryption Control

Encryption Support on Transit Gateway allows you to enforce encryption-in-transit for traffic between VPCs attached to a Transit Gateway. You will need to manually activate Encryption Support on the Transit Gateway using the [modify-transit-gateway](#) command to encrypt traffic between the VPCs. Once enabled, all traffic will traverse 100% encrypted links between VPCs that are in Enforce mode (without exclusions) through the Transit Gateway. You can also connect VPCs that don't have Encryption Controls turned on, or are in Monitor mode through a Transit Gateway that has Encryption Support enabled. In this scenario Transit Gateway is guaranteed to encrypt traffic up to the Transit Gateway attachment in the VPC not running in enforce mode. Beyond that, it depends on the instance the traffic is being sent to in the VPC not running in enforce mode.

You can only add encryption support to an existing transit gateway and not while you're creating one. As the Transit Gateway transitions to Encryption Support Enabled state, there will be no

downtime on the Transit Gateway or the attachments. The migration is seamless and transparent with no traffic being dropped. For the steps to modify a transit gateway to add Encryption Support, see [Modify a transit gateway](#).

Requirements

Before enabling encryption support on a transit gateway, ensure that:

- The transit gateway doesn't have Connect attachments
- The transit gateway doesn't have Peering attachments
- The transit gateway doesn't have Network Firewall attachments
- The transit gateway doesn't have VPN Concentrator attachments
- The transit gateway doesn't have Client VPN attachments
- The transit gateway doesn't have security group references enabled
- The transit gateway doesn't have Multicast features enabled

Encryption Support states

A transit gateway can have one of the following encryption states:

- **enabling** - The transit gateway is in the process of enabling encryption support. This process can take up to 14 days to complete.
- **enabled** - Encryption support is enabled on the transit gateway. You can create VPC attachments with Encryption Control enforced.
- **disabling** - The transit gateway is in the process of disabling Encryption support.
- **disabled** - Encryption support is disabled on the transit gateway.

Transit Gateway attachment rules

When a transit gateway has Encryption support enabled, the following attachment rules apply:

- When the transit gateway encryption state is **enabling** or **disabling**, you can create Direct Connect attachments, VPN attachments, and VPC attachment not in Encryption Control enforced or enforcing mode.
- When the transit gateway encryption state is **enabled**, you can create VPC, Direct Connect attachments, VPN attachments, and VPC attachments in any Encryption Control mode.

- When the transit gateway encryption state is **disabling**, you cannot create new VPC attachments with Encryption control enforced.
- Connect attachments, Peering attachments, Network Firewall attachments, VPN Concentrator attachments, Client VPN attachments, security group references, and multicast features are not supported with Encryption Support.
- Encryption Support is not supported for VPC attachments in the Availability Zone `use1-az3`. When Encryption Support is enabled on a transit gateway, you cannot create a VPC attachment that uses a subnet in this Availability Zone. To attach a subnet in `use1-az3`, you must first disable Encryption Support on the transit gateway.

Attempting to create incompatible attachments will fail with an API error.

Amazon VPC attachments in AWS Transit Gateway

An Amazon Virtual Private Cloud (VPC) attachment to a transit gateway allows you to route traffic to and from one or more VPC subnets. When you attach a VPC to a transit gateway, you must specify one subnet from each Availability Zone to be used by the transit gateway to route traffic. The specified subnets serve as the entry and exit points for transit gateway traffic. Traffic can only reach resources in other subnets within the same Availability Zone if the transit gateway attachment subnets have appropriate routes configured in their route tables pointing to the target subnets.

Limits

- When you attach a VPC to a transit gateway, any resources in Availability Zones where there is no transit gateway attachment cannot reach the transit gateway.

Note

Within Availability Zones that do have transit gateway attachments, traffic is only forwarded to the transit gateway from the specific subnets that are associated with the attachment. If there is a route to the transit gateway in a subnet route table, traffic is forwarded to the transit gateway only when the transit gateway has an attachment in a subnet in the same Availability Zone and the attachment subnet's route table contains appropriate routes to the traffic's intended destination within the VPC.

- A transit gateway does not support DNS resolution for custom DNS names of attached VPCs set up using private hosted zones in Amazon Route 53. To configure name resolution for private hosted zones for all VPCs attached to a transit gateway, see [Centralized DNS management of hybrid cloud with Amazon Route 53 and AWS Transit Gateway](#).
- A transit gateway doesn't support routing between VPCs with identical CIDRs, or if a CIDR in a range overlaps a CIDR in an attached VPC. If you attach a VPC to a transit gateway and its CIDR is identical to, or overlaps with, the CIDR of another VPC that's already attached to the transit gateway, the routes for the newly attached VPC aren't propagated to the transit gateway route table.
- You can't create an attachment for a VPC subnet that resides in a Local Zone. However, you can configure your network so that subnets in the Local Zone can connect to a transit gateway through the parent Availability Zone. For more information, see [Connect Local Zone subnets to a transit gateway](#).
- You can't create a transit gateway attachment using IPv6-only subnets. Transit gateway attachment subnets must also support IPv4 addresses.
- A transit gateway must have at least one VPC attachment before that transit gateway can be added to a route table.

Route table requirements for VPC attachments

Transit gateway VPC attachments require specific route table configurations to function properly:

- **Attachment subnet route tables:** The subnets associated with the transit gateway attachment must have route table entries for any destinations within the VPC that need to be reachable via the transit gateway. This includes routes to other subnets, internet gateways, NAT gateways, and VPC endpoints.
- **Target subnet route tables:** Subnets containing resources that need to communicate through the transit gateway must have routes pointing back to the transit gateway for return traffic to external destinations.
- **Local VPC traffic:** Transit gateway attachment does not automatically enable communication between subnets within the same VPC. Standard VPC routing rules apply, and the local route (VPC CIDR) must be present in route tables for intra-VPC communication.

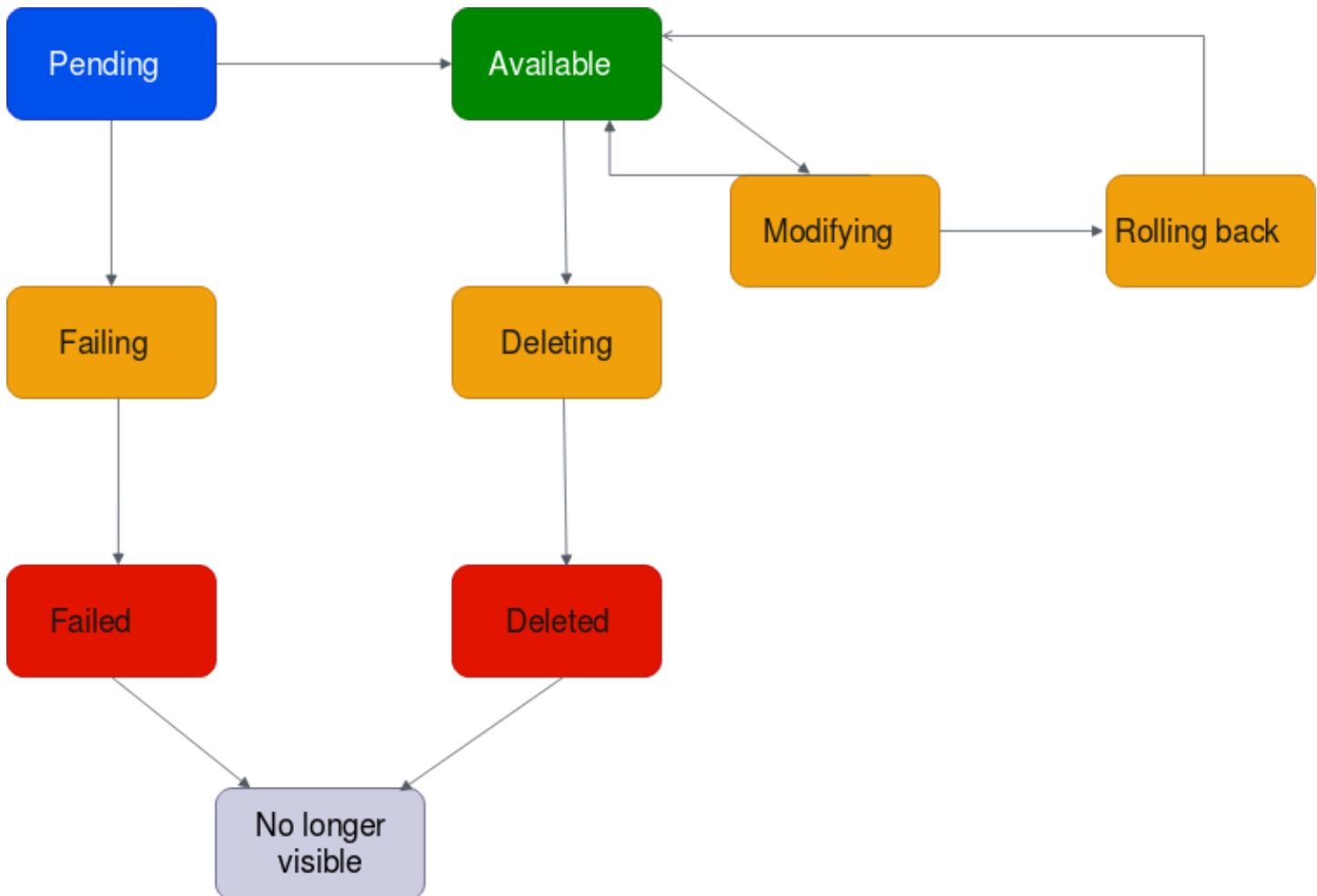
Note

Having routes configured in non-attachment subnets within the same Availability Zone does not enable traffic flow. Only the specific subnets associated with the transit gateway attachment can serve as entry/exit points for transit gateway traffic.

VPC attachment lifecycle

A VPC attachment goes through various stages, starting when the request is initiated. At each stage, there may be actions that you can take, and at the end of its lifecycle, the VPC attachment remains visible in the Amazon Virtual Private Cloud Console and in API or command line output, for a period of time.

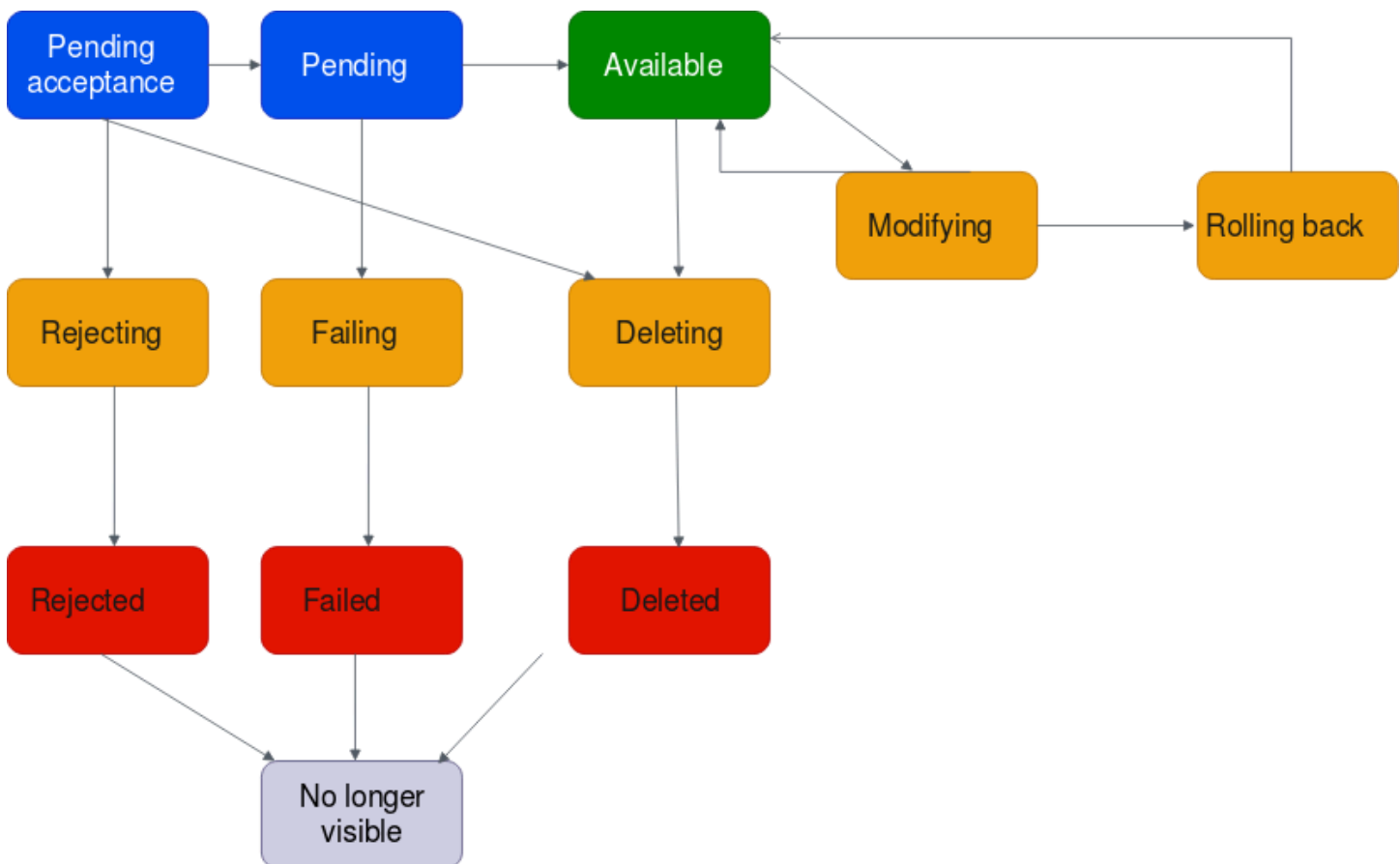
The following diagram shows the states an attachment can go through in a single account configuration, or a cross-account configuration that has **Auto accept shared attachments** turned on.



- **Pending:** A request for a VPC attachment has been initiated and is in the provisioning process. At this stage, the attachment can fail, or can go to available.
- **Failing:** A request for a VPC attachment is failing. At this stage, the VPC attachment goes to failed.
- **Failed:** The request for the VPC attachment has failed. While in this state, it cannot be deleted. The failed VPC attachment remains visible for 2 hours, and then is no longer visible.
- **Available:** The VPC attachment is available, and traffic can flow between the VPC and the transit gateway. At this stage, the attachment can go to modifying, or go to deleting.
- **Deleting:** A VPC attachment that is in the process of being deleted. At this stage, the attachment can go to deleted.
- **Deleted:** An available VPC attachment has been deleted. While in this state, the VPC attachment cannot be modified. The VPC attachment remains visible for 2 hours, and then is no longer visible.

- **Modifying:** A request has been made to modify the properties of the VPC attachment. At this stage, the attachment can go to available, or go to rolling back.
- **Rolling back:** The VPC attachment modification request cannot be completed, and the system is undoing any changes that were made. At this stage, the attachment can go to available.

The following diagram shows the states an attachment can go through in a cross-account configuration that has **Auto accept shared attachments** turned off.



- **Pending-acceptance:** The VPC attachment request is awaiting acceptance. At this stage, the attachment can go to pending, to rejecting, or to deleting.
- **Rejecting:** A VPC attachment that is in the process of being rejected. At this stage, the attachment can go to rejected.
- **Rejected:** A pending acceptance VPC attachment has been rejected. While in this state, the VPC attachment cannot be modified. The VPC attachment remains visible for 2 hours, and then is no longer visible.
- **Pending:** The VPC attachment has been accepted and is in the provisioning process. At this stage, the attachment can fail, or can go to available.

- **Failing:** A request for a VPC attachment is failing. At this stage, the VPC attachment goes to failed.
- **Failed:** The request for the VPC attachment has failed. While in this state, it cannot be deleted. The failed VPC attachment remains visible for 2 hours, and then is no longer visible.
- **Available:** The VPC attachment is available, and traffic can flow between the VPC and the transit gateway. At this stage, the attachment can go to modifying, or go to deleting.
- **Deleting:** A VPC attachment that is in the process of being deleted. At this stage, the attachment can go to deleted.
- **Deleted:** An available or pending acceptance VPC attachment has been deleted. While in this state, the VPC attachment cannot be modified. The VPC attachment remains visible 2 hours, and then is no longer visible.
- **Modifying:** A request has been made to modify the properties of the VPC attachment. At this stage, the attachment can go to available, or go to rolling back.
- **Rolling back:** The VPC attachment modification request cannot be completed, and the system is undoing any changes that were made. At this stage, the attachment can go to available.

Appliance mode

If you plan to configure a stateful network appliance in your VPC, you can enable appliance mode support for the VPC attachment in which the appliance is located when you create an attachment. This ensures that AWS Transit Gateway uses the same Availability Zone for that VPC attachment for the lifetime of the flow of traffic between a source and destination. It also allows a transit gateway to send traffic to any Availability Zone in the VPC as long as there is a subnet association in that zone. While appliance mode is only supported on VPC attachments, the network flow can come from any other transit gateway attachment type, including VPC, VPN, and Connect attachments. Appliance mode also works for network flows that have sources and destinations across different AWS Regions. Network flows can potentially be rebalanced across different Availability Zones if you don't initially enable appliance mode but later edit the attachment configuration to enable it. You can enable or disable appliance mode using either the console or the command line or API.

Important

Appliance mode is only supported for VPC attachments.

Prerequisites for AZ-aware routing: Route propagation must be enabled for the transit gateway route table associated with the appliance-mode VPC attachment. Without

propagation, the transit gateway cannot determine the source and destination Availability Zones. All traffic—including the same-Availability-Zone flows described in Scenario 1—falls back to flow-hash-based Availability Zone selection. This means traffic within the same Availability Zone might be routed to a different Availability Zone in the appliance VPC, breaking Availability Zone isolation.

Appliance mode in AWS Transit Gateway optimizes traffic routing by considering the source and destination Availability Zones when determining the path through an appliance mode VPC. This approach enhances efficiency and reduces latency. The behavior varies depending on the specific configuration and traffic patterns.

The following scenarios assume that route propagation is enabled for the transit gateway route table associated with the appliance-mode VPC attachment. Without propagation, the transit gateway cannot determine the source and destination Availability Zones, and all scenarios default to flow-hash-based Availability Zone selection (the behavior described in Scenario 2).

Scenario 1: Intra-Availability Zone Traffic Routing via Appliance VPC

When traffic flows from source Availability Zone us-east-1a to destination Availability Zone us-east-1a, with Appliance Mode VPC attachments in both us-east-1a and us-east-1b, Transit Gateway selects a network interface from us-east-1a within the appliance VPC. This Availability Zone is maintained for the entire duration of the traffic flow between source and destination.

Scenario 2: Inter-Availability Zone Traffic Routing via Appliance VPC

For traffic flowing from source Availability Zone us-east-1a to destination Availability Zone us-east-1b, with Appliance Mode VPC attachments in both us-east-1a and us-east-1b, Transit Gateway uses a flow hash algorithm to select either us-east-1a or us-east-1b in the appliance VPC. The chosen Availability Zone is used consistently for the lifetime of the flow.

Scenario 3: Routing traffic through an appliance VPC without Availability Zone data

When traffic originates from source Availability Zone us-east-1a to a destination without Availability Zone information (for example, internet-bound traffic), with Appliance Mode VPC attachments in both us-east-1a and us-east-1b, Transit Gateway selects a network interface from us-east-1a within the appliance VPC.

Scenario 4: Routing traffic through an appliance VPC in an Availability Zone distinct from either the source or destination

When traffic flows from source Availability Zone us-east-1a to destination Availability Zone us-east-1b, with Appliance Mode VPC attachments in different Availability Zone example us-east-1c and us-east-1d, Transit Gateway uses a flow hash algorithm to select either us-east-1c or us-east-1d in the appliance VPC. The chosen Availability Zone is used consistently for the lifetime of the flow.

Security group referencing

You can use this feature to simplify security group management and control of instance-to-instance traffic across VPCs that are attached to the same transit gateway. You can cross-reference security groups in inbound rules only. Outbound security rules do not support security group referencing. There are no additional costs associated with enabling or using security group referencing.

Security group referencing support can be configured for both transit gateways and transit gateway VPC attachments and will only work if it has been enabled for both a transit gateway and its VPC attachments.

Limitations

The following limitations apply when using security group referencing with a VPC attachment.

- Security group referencing is not supported across transit gateway peering connections. Both VPCs must be attached to the same transit gateway.
- Security group referencing is not supported for VPC attachments in the availability zone use1-az3.
- Security group referencing is not supported for PrivateLink endpoints. We recommend using IP CIDR-based security rules as an alternative.
- Security group referencing works for Elastic File System (EFS) as long as an allow all egress security group rule is configured for the EFS interfaces in the VPC.
- AWS Outposts, AWS Wavelength Zones, and some AWS Local Zones do not support security group referencing. To avoid service disruption, disable security group referencing at the VPC attachment level for VPCs with subnets in these locations.

Unsupported AWS Local Zones

The following AWS Local Zones do not support this feature: us-east-1-bos-1a, us-east-1-bue-1a, us-east-1-lim-1a, us-east-1-mci-1a, us-east-1-msp-1a, us-east-1-phl-1a, us-east-1-scl-1a, us-west-2-den-1a, us-west-2-hnl-1a, us-west-2-las-1a, us-west-2-sea-1a, ap-south-1-ccu-1a, ap-south-1-del-1a, ap-southeast-1-mnl-1a, ap-southeast-2-per-1a, eu-north-1-cph-1a, eu-north-1-hel-1a, eu-central-1-ham-1a, eu-central-1-waw-1a, af-south-1-los-1a, me-south-1-mct-1a.

- If you have an inspection VPC, then security group referencing through the transit gateway does not work across AWS Gateway Load Balancer or an AWS Network Firewall.

IPv6 support

When you create or modify a VPC attachment, you can enable or disable IPv6 support. The default value is `disable`.

Enabling IPv6 support does the following:

- Assigns an IPv6 address to the transit gateway network interface in the attachment subnet. The transit gateway uses this address to send and receive IPv6 traffic through the attachment.
- Propagates IPv6 VPC CIDRs to the transit gateway route table when route propagation is configured.

When you disable IPv6 support, the following behaviors still apply:

- You can still create static IPv6 routes that target the attachment.
- IPv6 traffic can still enter the transit gateway from the VPC if the VPC route table has an IPv6 route that points to the transit gateway, and network ACLs and security groups permit the traffic.

This setting controls only network interface addressing and route propagation—it does not filter packets.

Tasks

- [Create a VPC attachment in AWS Transit Gateway](#)

- [Modify a VPC attachment in AWS Transit Gateway](#)
- [Modify VPC attachment tags in AWS Transit Gateway](#)
- [View a VPC attachment in AWS Transit Gateway](#)
- [Delete a VPC attachment in AWS Transit Gateway](#)
- [Update AWS Transit Gateway security group inbound rules](#)
- [Identify AWS Transit Gateway referenced security groups](#)
- [Remove stale AWS Transit Gateway security group rules](#)
- [Troubleshoot AWS Transit Gateway VPC attachment creation](#)

Create a VPC attachment in AWS Transit Gateway

To create a VPC attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Choose **Create transit gateway attachment**.
4. For **Name tag**, optionally enter a name for the transit gateway attachment.
5. For **Transit gateway ID**, choose the transit gateway for the attachment. You can choose a transit gateway that you own or a transit gateway that was shared with you.
6. For **Attachment type**, choose **VPC**.
7. Choose whether to enable **DNS Support**, **IPv6 Support** and **Appliance mode support**.

If appliance mode is chosen, traffic flow between a source and destination uses the same Availability Zone for the VPC attachment for the lifetime of that flow.

8. Choose whether to enable **Security Group Referencing support**. Enable this feature to reference a security group across VPCs attached to a transit gateway. For more information about security group referencing, see [the section called "Security group referencing"](#).
9. Choose whether to enable **IPv6 Support**. When enabled, the transit gateway network interface receives an IPv6 address. IPv6 VPC CIDRs propagate to transit gateway route tables when route propagation is configured. When disabled, the network interface does not receive an IPv6 address, and IPv6 CIDRs do not propagate. For more information, see [the section called "IPv6 support"](#).
10. For **VPC ID**, choose the VPC to attach to the transit gateway.

This VPC must have at least one subnet associated with it.

11. For **Subnet IDs**, select one subnet for each Availability Zone to be used by the transit gateway to route traffic. You must select at least one subnet. You can select only one subnet per Availability Zone.
12. Choose **Create transit gateway attachment**.

To create a VPC attachment using the AWS CLI

Use the [create-transit-gateway-vpc-attachment](#) command.

Modify a VPC attachment in AWS Transit Gateway

To modify your VPC attachments using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Select the VPC attachment, and then choose **Actions, Modify transit gateway attachment**.
4. Enable or disable any of the following:
 - **DNS support**
 - **IPv6 support**
 - **Appliance mode support**
5. To add or remove a subnet from the attachment, choose or clear the checkbox by the **Subnet ID** you want to add or remove.

Note

Adding or modifying a VPC attachment subnet might impact data traffic while the attachment is in a modifying state.

6. To be able to reference a security group across VPCs attached to a transit gateway, select **Security Group Referencing support**. For more information about security group referencing, see [the section called "Security group referencing"](#).

Note

If you disable security group referencing for an existing transit gateway, it will be disabled on all VPC attachments.

7. Choose **Modify transit gateway attachment**.

To modify your VPC attachments using the AWS CLI

Use the [modify-transit-gateway-vpc-attachment](#) command.

Modify VPC attachment tags in AWS Transit Gateway**To modify your VPC attachment tags using the console**

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Select the VPC attachment, and then choose **Actions, Manage tags**.
4. [Add a tag] Choose **Add new tag** and do the following:
 - For **Key**, enter the key name.
 - For **Value**, enter the key value.
5. [Remove a tag] Next to the tag, choose **Remove**.
6. Choose **Save**.

VPC attachment tags can only be modified using the console.

View a VPC attachment in AWS Transit Gateway**To view your VPC attachments using the console**

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. In the **Resource type** column, look for **VPC**. These are the VPC attachments.
4. Choose an attachment to view its details.

To view your VPC attachments using the AWS CLI

Use the [describe-transit-gateway-vpc-attachments](#) command.

Delete a VPC attachment in AWS Transit Gateway

To delete a VPC attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Select the VPC attachment.
4. Choose **Actions, Delete transit gateway attachment**.
5. When prompted, enter **delete** and choose **Delete**.

To delete a VPC attachment using the AWS CLI

Use the [delete-transit-gateway-vpc-attachment](#) command.

Update AWS Transit Gateway security group inbound rules

You can update any of the inbound security group rules associated with a transit gateway. You can update security group rules using either the Amazon VPC Console console or by using the command-line or API. For more information about security group referencing, see [the section called "Security group referencing"](#).

To update your security group rules using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Security groups**.
3. Select the security group, and choose **Actions, Edit inbound rules** to modify the inbound rules.
4. To add a rule, choose **Add rule** and specify the type, protocol, and port range. For **Source** (inbound rule), enter the ID of the security group in the VPC connected to the transit gateway.

Note

Security groups in a VPC connected to the transit gateway are not automatically displayed.

5. To edit an existing rule, change its values (for example, the source or the description).
6. To delete a rule, choose **Delete** next to the rule.
7. Choose **Save rules**.

To update inbound rules using the command line

- [authorize-security-group-ingress](#) (AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)
- [Revoke-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)
- [revoke-security-group-ingress](#) (AWS CLI)

Identify AWS Transit Gateway referenced security groups

To determine if your security group is being referenced in the rules of a security group in a VPC attached to the same transit gateway, use one of the following commands.

- [describe-security-group-references](#) (AWS CLI)
- [Get-EC2SecurityGroupReference](#) (AWS Tools for Windows PowerShell)

Remove stale AWS Transit Gateway security group rules

A stale security group rule is a rule that references a deleted security group in the same VPC or in VPC attached to the same transit gateway. When a security group rule becomes stale, it's not automatically removed from your security group—you must manually remove it.

You can view and delete the stale security group rules for a VPC using the Amazon VPC console.

To view and delete stale security group rules

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Security groups**.

3. Choose **Actions, Manage stale rules**.
4. For **VPC**, choose the VPC with the stale rules.
5. Choose **Edit**.
6. Choose the **Delete** button next to the rule that you want to delete. Choose **Preview changes, Save rules**.

To describe your stale security group rules using the command line

- [describe-stale-security-groups](#) (AWS CLI)
- [Get-EC2StaleSecurityGroup](#) (AWS Tools for Windows PowerShell)

After you've identified the stale security group rules, you can delete them using the [revoke-security-group-ingress](#) or [revoke-security-group-egress](#) commands.

Troubleshoot AWS Transit Gateway VPC attachment creation

The following topic can help you troubleshoot problems that you might have when you create a VPC attachment.

Problem

The VPC attachment failed.

Cause

The cause might be one of the following:

1. The user that is creating the VPC attachment does not have correct permissions to create service-linked role.
2. There is a throttling issue because of too many IAM requests, for example you are using CloudFormation to create permissions and roles.
3. The account has the service-linked role, and the service-linked role has been modified.
4. The transit gateway is not in the available state.

Solution

Depending on the cause, try the following:

1. Verify that the user has the correct permissions to create service-linked roles. For more information, see [Service-linked role permissions](#) in the *IAM User Guide*. After the user has the permissions, create the VPC attachment.
2. Create the VPC attachment manually. For more information, see [the section called "Create a VPC attachment"](#).
3. Verify that the service-linked role has the correct permissions. For more information, see [the section called "Transit gateway"](#).
4. Verify that the transit gateway is in the available state. For more information, see [the section called "View a transit gateway"](#).

AWS Transit Gateway network function attachments

You can create a network function attachment to connect your transit gateway directly to AWS Network Firewall. This eliminates the need to create and manage inspection VPCs.

With a firewall attachment, AWS automatically provisions and manages all the necessary resources behind the scenes. You'll see a new transit gateway attachment rather than individual firewall endpoints. This simplifies the process of implementing centralized network traffic inspection.

Before you can use a firewall attachment, you must first create the attachment in AWS Network Firewall. For the steps to create the attachment, see [Getting Started with AWS Network Firewall Management](#) in the *AWS Network Firewall Developer Guide*. After the firewall is created, you can view the attachment in Transit Gateway console under the **Attachments** section. The attachment will be listed with a type of **Network function**.

Topics

- [Accept or reject an AWS Transit Gateway network function attachment](#)
- [View AWS Transit Gateway network function attachments](#)
- [Route traffic through an AWS Transit Gateway network function attachment](#)

Accept or reject an AWS Transit Gateway network function attachment

You can use either the Amazon VPC console or the AWS Network Firewall CLI or API to accept or reject a transit gateway network function attachment, including Network Firewall attachments. If you are the owner of a transit gateway and someone has created a firewall attachment to your transit gateway from another account, you need to accept or reject the attachment request.

To accept or reject a network function attachment using the Network Firewall CLI, see the `AcceptNetworkFirewallTransitGatewayAttachment` or `RejectNetworkFirewallTransitGatewayAttachment` APIs in the [AWS Network Firewall API Reference](#).

Accept or reject a network function attachment using the console

Use the Amazon VPC console to accept or reject a transit gateway network function attachment.

To accept or reject a network function attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit Gateways**.
3. Choose **Transit gateway attachments**.
4. Select the attachment with a state of **Pending acceptance** and a type of **Network function**.
5. Choose **Actions**, and then choose either **Accept attachment** or **Reject attachment**.
6. In the confirmation dialog box, choose **Accept** or **Reject**.

If you accept the attachment, it becomes active and the firewall can inspect traffic. If you reject the attachment, it enters a rejected state and will eventually be deleted.

View AWS Transit Gateway network function attachments

You can view your network function attachments, including your AWS Network Firewall attachments, using either Amazon VPC Console or the Network Manager console to get a visual representation of your network topology.

View a network function attachment using the Network Manager console

You can view a network function attachments using the Network Manager console.

To view firewall attachments in Network Manager

1. Open the Network Manager console at <https://console.aws.amazon.com/networkmanager/home/>.
2. Create a global network in Network Manager if you don't already have one.
3. Register your transit gateway with Network Manager.

4. Under **Global Networks**, choose the global network where the attachment is located.
5. In the navigation pane, choose **Transit gateways**.
6. Choose the transit gateway that you want to view attachments for.
7. Choose **Topology tree** view. Network Firewall attachments appear with a network function icon.
8. To view details about a specific firewall attachment, select the transit gateway in the topology view, then select the **Network function** tab.

The Network Manager console provides detailed information about your firewall attachments, including their status, associated transit gateway, and Availability Zones.

View a network function attachment using the Amazon VPC Console console

Use the VPC console to see a list of your transit gateway attachment types.

To view transit gateway attachment types using the VPC console

- See [View a VPC attachment](#).

Route traffic through an AWS Transit Gateway network function attachment

After creating a network function attachment, you need to update your transit gateway route tables to send traffic through the firewall for inspection using either the Amazon VPC Console or by using the CLI. For the steps to update a transit gateway route table association, see [Associate a transit gateway route table](#).

Route traffic through a firewall attachment using the console

Use the Amazon VPC Console console to route traffic through a transit gateway network function attachment.

To route traffic through a network function attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit Gateways**.
3. Choose **Transit gateway route tables**.

4. Select the route table you want to modify.
5. Choose **Actions**, and then choose **Create static route**.
6. For **CIDR**, enter the destination CIDR block for the route.
7. For **Attachment**, select the network function attachment. For example, this might be an AWS Network Firewall attachment.
8. Choose **Create static route**.

 Note

Only static routes are supported.

Traffic matching the CIDR block in your route table will now be sent to the firewall attachment for inspection before being forwarded to its final destination.

Route traffic through a network function attachment using the CLI or API

Use the command line or API to route a transit gateway network function attachment.

To route traffic through a network function attachment using the command line or API

- Use [create-transit-gateway-route](#).

For example, the request might be to route a network firewall attachment:

```
aws ec2 create-transit-gateway-route \
  --transit-gateway-route-table-id tgw-rtb-0123456789abcdef0 \
  --destination-cidr-block 0.0.0.0/0 \
  --transit-gateway-attachment-id tgw-attach-0123456789abcdef0
```

The output then returns:

```
{
  "Route": {
    "DestinationCidrBlock": "0.0.0.0/0",
    "TransitGatewayAttachments": [
      {
        "ResourceId": "network-firewall",
        "TransitGatewayAttachmentId": "tgw-attach-0123456789abcdef0",
        "ResourceType": "network-function"
      }
    ]
  }
}
```

```
    }  
  ],  
  "Type": "static",  
  "State": "active"  
}  
}
```

Traffic matching the CIDR block in your route table will now be sent to the firewall attachment for inspection before being forwarded to its final destination.

AWS Site-to-Site VPN attachments in AWS Transit Gateway

You can connect a Site-to-Site VPN attachment to a transit gateway in AWS Transit Gateway, allowing you to connect your VPCs and on-premises networks. Both dynamic and static routes are supported, as well as IPv4 and IPv6.

Requirements

- Attaching a VPN connection to your transit gateway requires that you specify the VPN customer gateway, which have specific device requirements. Before creating a Site-to-Site VPN attachment, review the customer gateway requirements to ensure that your gateway is set up correctly. For more information about these requirements, including example gateway configuration files, see [Requirements for your Site-to-Site VPN customer gateway device](#) in the *AWS Site-to-Site VPN User Guide*.
- For static VPNs, you'll also need to first add the static routes to the transit gateway route table. Static routes in a transit gateway route table that target a VPN attachment are not filtered by the Site-to-Site VPN as this might allow unintended outbound traffic flow when using a BGP-based VPN. For the steps to add a static route to a transit gateway route table, see [Create a static route](#).

You can create, view, or delete a transit gateway Site-to-Site VPN attachment using either the Amazon VPC console or using the AWS CLI.

Tasks

- [Create a transit gateway attachment to a VPN in AWS Transit Gateway](#)
- [View a VPN attachment in AWS Transit Gateway](#)
- [Delete a VPN attachment in AWS Transit Gateway](#)

Create a transit gateway attachment to a VPN in AWS Transit Gateway

To create a VPN attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Choose **Create transit gateway attachment**.
4. For **Transit gateway ID**, choose the transit gateway for the attachment. You can choose a transit gateway that you own.
5. For **Attachment type**, choose **VPN**.
6. For **Customer Gateway**, do one of the following:
 - To use an existing customer gateway, choose **Existing**, and then select the gateway to use.

If your customer gateway is behind a network address translation (NAT) device that's enabled for NAT traversal (NAT-T), use the public IP address of your NAT device, and adjust your firewall rules to unblock UDP port 4500.
 - To create a customer gateway, choose **New**, then for **IP Address**, type a static public IP address and **BGP ASN**.

For **Routing options**, choose whether to use **Dynamic** or **Static**. For more information, see [Site-to-Site VPN Routing Options](#) in the *AWS Site-to-Site VPN User Guide*.
7. For **Tunnel Options**, enter the CIDR ranges and pre-shared keys for your tunnel. For more information, see [Site-to-Site VPN architectures](#).
8. Choose **Create transit gateway attachment**.

To create a VPN attachment using the AWS CLI

Use the [create-vpn-connection](#) command.

View a VPN attachment in AWS Transit Gateway

To view your VPN attachments using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. In the **Resource type** column, look for **VPN**. These are the VPN attachments.

4. Choose an attachment to view its details or to add tags.

To view your VPN attachments using the AWS CLI

Use the [describe-transit-gateway-attachments](#) command.

Delete a VPN attachment in AWS Transit Gateway

To delete a VPN attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Select the VPN attachment.
4. Choose the resource ID of the VPN connection to navigate to the **VPN Connections** page.
5. Choose **Actions, Delete**.
6. When prompted for confirmation, choose **Delete**.

To delete a VPN attachment using the AWS CLI

Use the [delete-vpn-connection](#) command.

VPN Concentrator attachments in AWS Transit Gateway

AWS Site-to-Site VPN Concentrator is a new feature that simplifies multi-site connectivity for distributed enterprises. VPN Concentrator is suitable for customers who need to connect 25+ remote sites to AWS, with each site needing low bandwidth (under 100 Mbps).

How VPN Concentrator works

A VPN Concentrator appears as a single attachment on your transit gateway, but can host multiple Site-to-Site VPN connections.

Traffic from all VPN connections on the Concentrator is routed through the same transit gateway attachment, allowing you to apply consistent routing policies and security rules across all connected sites. The Concentrator integrates seamlessly with transit gateway route tables, enabling you to control traffic flow between your remote sites and other attachments such as VPCs, other VPN connections, and peering connections.

Benefits of VPN Concentrator

- **Cost optimization:** Reduce costs by consolidating multiple low-bandwidth VPN connections onto a single transit gateway attachment, especially beneficial when individual sites don't require full VPN attachment capacity.
- **Simplified management:** Manage multiple remote site connections through a unified attachment while maintaining individual VPN connection control and monitoring.
- **Consistent routing:** Apply unified routing policies across all connected sites through a single transit gateway route table association.
- **Scalable architecture:** Connect up to 100 remote sites using a single Concentrator, with support for up to 5 Concentrators per transit gateway.
- **Standard VPN features:** Each VPN connection supports the same security, monitoring, and routing capabilities as standard Site-to-Site VPN connections.

Requirements and limitations

- **BGP routing only:** VPN Concentrator supports BGP (dynamic) routing only. Static routing is not supported at launch.
- **Customer gateway requirements:** Each remote site requires a customer gateway that supports BGP routing. Before creating VPN connections on a Concentrator, review the customer gateway requirements in [Requirements for your Site-to-Site VPN customer gateway device](#) in the *AWS Site-to-Site VPN User Guide*.
- **Performance considerations:** Each VPN connection on a Concentrator is designed for a maximum of 100 Mbps bandwidth. For higher bandwidth requirements, consider using standard transit gateway VPN attachments.

You can create, view, or delete a VPN Concentrator attachment using either the AWS VPC console or the AWS CLI. Individual VPN connections on the Concentrator are managed through the standard VPN connection APIs and console interfaces.

Tasks

- [Create a VPN Concentrator attachment in AWS Transit Gateway](#)
- [View a VPN Concentrator attachment in AWS Transit Gateway](#)
- [Delete a VPN Concentrator attachment in AWS Transit Gateway](#)

Create a VPN Concentrator attachment in AWS Transit Gateway

Prerequisites

- You must have an existing transit gateway in your account.

To create a VPN Concentrator attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Site-to-Site VPN Concentrators**.
3. Choose **Create Site-to-Site VPN Concentrator**.
4. (Optional) For **Name tag**, enter a name for your Site-to-Site VPN Concentrator.
5. For **Transit gateway**, select an existing transit gateway.
6. (Optional) To add additional tags, choose **Add new tag** and specify the key and value for each tag.
7. Choose **Create Site-to-Site VPN Concentrator**.

After you create the VPN Concentrator attachment, it appears in the list of attachments with a resource type of **VPN Concentrator** and an initial state of **Pending**. When the attachment is ready, the state changes to **Available**. You can then create Site-to-Site VPN connections on this Concentrator.

To create a VPN Concentrator attachment using the AWS CLI

Use the [create-vpn-concentrator](#) command.

To create a VPN connection on a VPN Concentrator using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Site-to-Site VPN Connections**.
3. Choose **Create VPN connection**.
4. For **Target Gateway Type**, choose **Site-to-Site VPN Concentrator**.
5. For **Site-to-Site VPN Concentrator**, choose the VPN Concentrator where you want to create the VPN connection.
6. For **Customer Gateway**, do one of the following:

- To use an existing customer gateway, choose **Existing**, and then select the gateway to use. Ensure that the customer gateway supports BGP routing.
- To create a customer gateway, choose **New**. For **IP Address**, enter the static public IP address for your customer gateway device. For **BGP ASN**, enter the Border Gateway Protocol (BGP) Autonomous System Number (ASN) for your customer gateway.

If your customer gateway is behind a network address translation (NAT) device that's enabled for NAT traversal (NAT-T), use the public IP address of your NAT device, and adjust your firewall rules to unblock UDP port 4500.

7. For **Routing options**, **Dynamic (requires BGP)** is automatically selected. VPN Concentrator only supports dynamic routing with BGP.
8. For **Pre-shared key storage**, select either **Standard** or **Secrets Manager**.
9. For **Tunnel bandwidth**, **Standard** is automatically selected. VPN Concentrator only supports standard tunnel bandwidth.
10. For **Tunnel inside IP version**, select either **IPv4** or **IPv6**.
11. (Optional) Select **Enable acceleration** to improve performance of VPN tunnels.
12. (Optional) For **Local IPv4 network CIDR**, provide an IPv4 CIDR range.
13. (Optional) For **Remote IPv4 network CIDR**, provide an IPv4 CIDR range.
14. For **Outside IP Address Type**, you can select either **Public IPv4** or **IPv6** address.
15. (Optional) For **Tunnel Options**, you can configure tunnel settings such as inside tunnel IP addresses and pre-shared keys. For more information, see [Site-to-Site VPN architectures](#) in the *AWS Site-to-Site VPN User Guide*.
16. (Optional) To add additional tags, choose **Add new tag** and specify the key and value for each tag.
17. Choose **Create VPN connection**.

The VPN connection appears in the list of VPN connections with the VPN Concentrator ID in the **Transit Gateway ID** column and an initial state of **Pending**. When the VPN connection is ready, the state changes to **Available**.

To create a VPN connection on a VPN Concentrator using the AWS CLI

Use the [create-vpn-connection](#) command and specify the VPN Concentrator ID using the `--vpn-concentrator-id` parameter.

View a VPN Concentrator attachment in AWS Transit Gateway

To view your VPN Concentrator attachments using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. In the **Resource type** column, look for **VPN Concentrator**. These are the VPN Concentrator attachments.
4. Choose an attachment to view its details.

To view VPN connections on a VPN Concentrator using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Site-to-Site VPN Connections**.
3. In the list of VPN connections, identify connections that show a VPN Concentrator ID in the **Transit Gateway ID** column. These are the VPN connections hosted on VPN Concentrators.
4. Choose a VPN connection to view its details.

To view your VPN Concentrator attachments using the AWS CLI

Use the [describe-vpn-concentrator](#) command to view VPN Concentrator details, or use the [describe-transit-gateway-attachments](#) command with a filter for resource type `vpn-concentrator`.

To view VPN connections on a VPN Concentrator using the AWS CLI

Use the [describe-vpn-connections](#) command with a filter for `vpn-concentrator-id` to view VPN connections associated with a specific Concentrator.

Delete a VPN Concentrator attachment in AWS Transit Gateway

Prerequisites

- All VPN connections on the VPN Concentrator must be deleted before you can delete the Concentrator attachment.
- Ensure that you have updated your routing configurations to account for the removal of the VPN Concentrator and its associated VPN connections.

To delete VPN connections on a VPN Concentrator using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Site-to-Site VPN Connections**.
3. Identify the VPN connections associated with your VPN Concentrator by looking for the VPN Concentrator ID in the **Transit Gateway ID** column.
4. Select a VPN connection that you want to delete.
5. Choose **Actions, Delete**.
6. When prompted for confirmation, choose **Delete**.
7. Repeat steps 4-6 for each VPN connection associated with the VPN Concentrator.

To delete a VPN Concentrator attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Select the VPN Concentrator attachment that you want to delete. Verify that no VPN connections are associated with this Concentrator.
4. Choose **Actions, Delete attachment**.
5. When prompted for confirmation, choose **Delete**.

The VPN Concentrator attachment enters the **Deleting** state and will be removed from your account. This process may take a few minutes to complete.

To delete VPN connections on a VPN Concentrator using the AWS CLI

Use the [delete-vpn-connection](#) command for each VPN connection associated with the VPN Concentrator.

To delete a VPN Concentrator attachment using the AWS CLI

Use the [delete-vpn-concentrator](#) command after all VPN connections have been deleted.

Client VPN attachments in AWS Transit Gateway

When you associate a Client VPN endpoint with a transit gateway, a Client VPN attachment is automatically created, allowing you to route traffic between your VPCs, on-premises networks,

and Client VPN endpoints. AWS Transit Gateway supports cross-account Client VPN attachments, allowing accounts that the transit gateway is shared with to create their own Client VPN attachments.

After the Client VPN endpoint is associated with a transit gateway, you can view the attachment in the Transit Gateway console under **Transit gateway attachments**. The attachment will be listed with a type of **Client VPN**.

Requirements and limitations

- Your transit gateway must have an assigned IPv4 or IPv6 CIDR block before you can create a Client VPN attachment.
- Route table propagation must be enabled for Client VPN attachments to allow traffic between your Client VPN endpoint and transit gateway. See [Enable route propagation](#).

Tasks

- [Create a Client VPN attachment in AWS Transit Gateway](#)
- [View a Client VPN attachment in AWS Transit Gateway](#)
- [Delete a Client VPN attachment in AWS Transit Gateway](#)
- [Accept or reject a Client VPN attachment in AWS Transit Gateway](#)

Create a Client VPN attachment in AWS Transit Gateway

Prerequisites

- You must have an existing transit gateway in your account.
- Your transit gateway must have an assigned IPv4 or IPv6 CIDR block.

A Client VPN attachment is automatically created when you associate a Client VPN endpoint with a transit gateway.

To create a Client VPN attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Client VPN endpoints**.
3. Choose **Create Client VPN endpoint**.

4. Select **Transit Gateway** as the association type and enter the Transit Gateway ID to use.
5. Choose **Create Client VPN endpoint**.

After you create the Client VPN attachment, it appears in the list of attachments with a resource type of **Client VPN** and an initial state of **Pending**. When the attachment is ready, the state changes to **Available**. If the transit gateway is in a different account, the attachment state is **Pending acceptance** until the transit gateway owner accepts it.

For more information about creating Client VPN endpoints, see [Getting Started with AWS Client VPN](#).

To create a Client VPN attachment using the AWS CLI

Use the [create-client-vpn-endpoint](#) command.

View a Client VPN attachment in AWS Transit Gateway

To view your Client VPN attachments using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit gateways**.
3. Choose **Transit gateway attachments**.
4. In the **Resource type** column, look for **Client VPN**.
5. Choose an attachment to view its details.

To view your Client VPN attachments using the AWS CLI

Use the [describe-transit-gateway-attachments](#) command with a filter for resource type `client-vpn`.

Delete a Client VPN attachment in AWS Transit Gateway

To delete a Client VPN attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit gateways**.
3. Choose **Transit gateway attachments**.

4. Select the Client VPN attachment that you want to delete.
5. Choose **Actions, Delete transit gateway attachment**.
6. When prompted for confirmation, enter **delete** and choose **Delete**.

The Client VPN attachment enters the **Deleting** state and will be removed from your account. This process may take some time to complete.

To delete a Client VPN attachment using the AWS CLI

Use the [delete-transit-gateway-client-vpn-attachment](#) command.

Accept or reject a Client VPN attachment in AWS Transit Gateway

If a Client VPN endpoint in another account creates an attachment to your transit gateway, you must accept or reject the attachment request before traffic can flow.

To accept or reject a Client VPN attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit gateways**.
3. Choose **Transit gateway attachments**.
4. Select the attachment with a state of **Pending acceptance** and a type of **Client VPN**.
5. Choose **Actions**, and then choose either **Accept attachment** or **Reject attachment**.
6. In the confirmation dialog box, choose **Accept** or **Reject**.

If you accept the attachment, it becomes active and AWS Transit Gateway will begin processing traffic to and from the Client VPN endpoint. If you reject the attachment, it enters a rejected state and will eventually be deleted.

To accept a Client VPN attachment using the AWS CLI

Use the [accept-transit-gateway-client-vpn-attachment](#) command.

To reject a Client VPN attachment using the AWS CLI

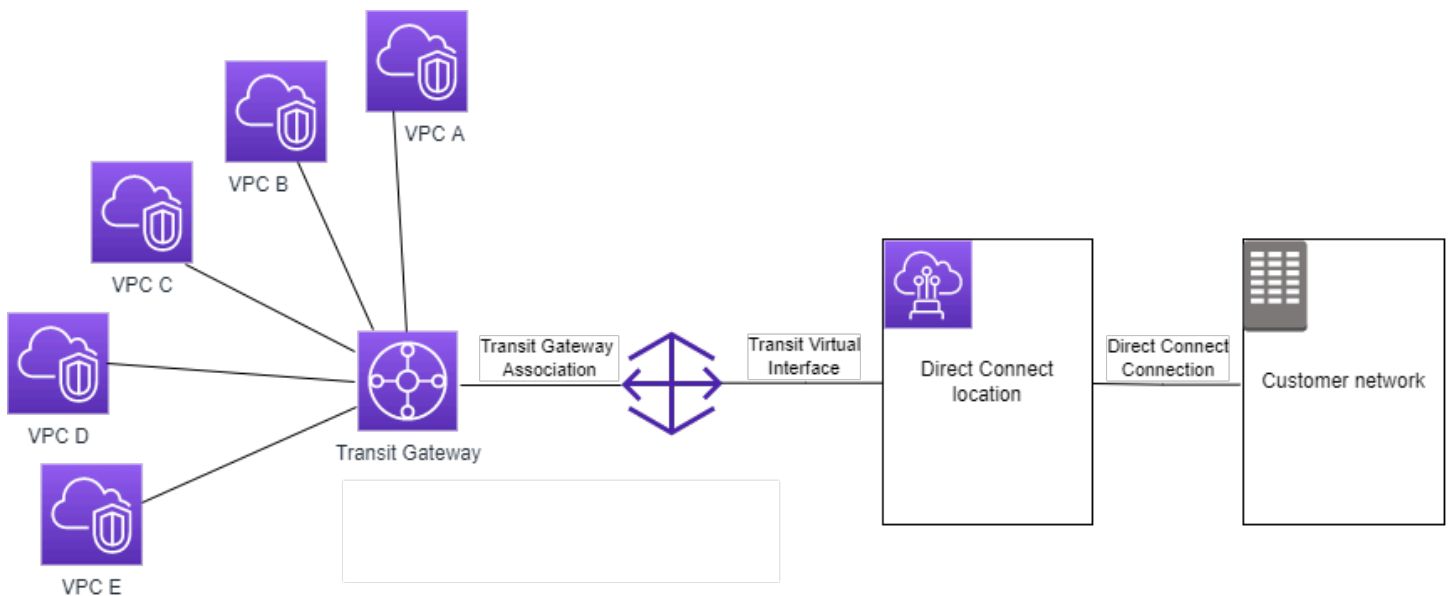
Use the [reject-transit-gateway-client-vpn-attachment](#) command.

Transit gateway attachments to a Direct Connect gateway in AWS Transit Gateway

Attach a transit gateway to a Direct Connect gateway using a transit virtual interface. This configuration offers the following benefits. You can:

- Manage a single connection for multiple VPCs or VPNs that are in the same Region.
- Advertise prefixes from on-premises to AWS and from AWS to on-premises.

The following diagram illustrates how the Direct Connect gateway enables you to create a single connection to your Direct Connect connection that all of your VPCs can use.



The solution involves the following components:

- A transit gateway.
- A Direct Connect gateway.
- An association between the Direct Connect gateway and the transit gateway.
- A transit virtual interface that is attached to the Direct Connect gateway.

For information about configuring Direct Connect gateways with transit gateways, see [Transit gateway associations](#) in the *AWS Direct Connect User Guide*.

Transit gateway peering attachments in AWS Transit Gateway

You can peer both intra-Region and inter-Region transit gateways, and route traffic between them, which includes IPv4 and IPv6 traffic. To do this, create a peering attachment on your transit gateway, and specify a transit gateway. The peer transit gateway can either be in your account or can be from another account. You can also request a peering attachment from your own account to a transit gateway in another account.

After you create a peering attachment request, the owner of the peer transit gateway (also referred to as the *accepter transit gateway*) must accept the request. To route traffic between the transit gateways, add a static route to the transit gateway route table that points to the transit gateway peering attachment.

We recommend using unique ASNs for each peered transit gateway to take advantage of future route propagation capabilities.

Transit gateway peering does not support resolving public or private IPv4 DNS host names to private IPv4 addresses across VPCs on either side of the transit gateway peering attachment using the Amazon Route 53 Resolver in another Region. For more information about the Route 53 Resolver, see [What is Route 53 Resolver?](#) in the *Amazon Route 53 Developer Guide*.

Inter-Region gateway peering uses the same network infrastructure as VPC peering. Therefore traffic is encrypted using AES-256 encryption at the virtual network layer as it travels between Regions. Traffic is also encrypted using AES-256 encryption at the physical layer when it traverses network links that are outside of the physical control of AWS. As a result, traffic is double encrypted on network links outside the physical control of AWS. Within the same Region, traffic is encrypted at the physical layer only when it traverses network links that are outside of the physical control of AWS.

For information about which Regions support transit gateway peering attachments, see [AWS Transit Gateways FAQs](#).

Opt-in AWS Region considerations

You can peer transit gateways across opt-in Region boundaries. For information about these Regions, and how to opt in, see [Managing AWS Regions](#). Take the following into consideration when you use transit gateway peering in these Regions:

- You can peer into an opt-in Region as long as the account that accepts the peering attachment has opted into that Region.

- Regardless of the Region opt-in status, AWS shares the following account data with the account that accepts the peering attachment:
 - AWS account ID
 - Transit gateway ID
 - Region code
- When you delete the transit gateway attachment, the above account data is deleted.
- We recommend that you delete the transit gateway peering attachment before you opt out of the Region. If you do not delete the peering attachment, traffic might continue to go over the attachment and you continue to incur charges. If you do not delete the attachment, you can opt back in, and then delete the attachment.
- In general, the transit gateway has a sender pays model. By using a transit gateway peering attachment across an opt in boundary, you might incur charges in a Region accepting the attachment, including those Regions you have not opted into. For more information, see [AWS Transit Gateway Pricing](#).

Tasks

- [Create a peering attachment in AWS Transit Gateway](#)
- [Accept or reject a peering attachment request in AWS Transit Gateway](#)
- [Add a route to a transit gateway route table using AWS Transit Gateway](#)
- [Delete a peering attachment in AWS Transit Gateway](#)

Create a peering attachment in AWS Transit Gateway

Before you begin, ensure that you have the ID of the transit gateway that you want to attach. If the transit gateway is in another AWS account, ensure that you have the AWS account ID of the owner of the transit gateway. After you create the peering attachment, the owner of the acceptor transit gateway must accept or reject the attachment request.

To create a peering attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Choose **Create transit gateway attachment**.

4. For **Transit gateway ID**, choose the transit gateway for the attachment. You can choose a transit gateway that you own. Transit gateways that are shared with you are not available for peering.
5. For **Attachment type**, choose **Peering Connection**.
6. Optionally enter a name tag for the attachment.
7. For **Account**, do one of the following:
 - If the transit gateway is in your account, choose **My account**.
 - If the transit gateway is in different AWS account, choose **Other account**. For **Account ID**, enter the AWS account ID.
8. For **Region**, choose the Region that the transit gateway is located in.
9. For **Transit gateway (accepter)**, enter the ID of the transit gateway that you want to attach.
10. Choose **Create transit gateway attachment**.

To create a peering attachment using the AWS CLI

Use the [create-transit-gateway-peering-attachment](#) command.

Accept or reject a peering attachment request in AWS Transit Gateway

When created, a transit gateway peering attachment is automatically created in a pendingAcceptance state and remains in this state indefinitely until it's either accepted or rejected. To activate the peering attachment, the owner of the accepter transit gateway must accept the peering attachment request, even if both transit gateways are in the same account. Accept the peering attachment request from the Region that the accepter transit gateway is located in. Alternatively, if you reject the peering attachment, you must reject the request from the Region that the accepter transit gateway is located in.

To accept a peering attachment request using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Select the transit gateway peering attachment that's pending acceptance.
4. Choose **Actions, Accept transit gateway attachment**.
5. Add the static route to the transit gateway route table. For more information, see [the section called "Create a static route"](#).

To reject a peering attachment request using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Select the transit gateway peering attachment that's pending acceptance.
4. Choose **Actions, Reject transit gateway attachment**.

To accept or reject a peering attachment using the AWS CLI

Use the [accept-transit-gateway-peering-attachment](#) and [reject-transit-gateway-peering-attachment](#) commands.

Add a route to a transit gateway route table using AWS Transit Gateway

To route traffic between the peered transit gateways, you must add a static route to the transit gateway route table that points to the transit gateway peering attachment. The owner of the acceptor transit gateway must also add a static route to their transit gateway's route table.

To create a static route using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Select the route table for which to create a route.
4. Choose **Actions, Create static route**.
5. On the **Create static route** page, enter the CIDR block for which to create the route. For example, specify the CIDR block of a VPC that's attached to the peer transit gateway.
6. Choose the peering attachment for the route.
7. Choose **Create static route**.

To create a static route using the AWS CLI

Use the [create-transit-gateway-route](#) command.

⚠ Important

After you create the route, the transit gateway peering attachment must already be associated with the transit gateway route table. For more information, see [the section called "Associate a transit gateway route table"](#).

Delete a peering attachment in AWS Transit Gateway

You can delete a transit gateway peering attachment. The owner of either of the transit gateways can delete the attachment.

To delete a peering attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Attachments**.
3. Select the transit gateway peering attachment.
4. Choose **Actions, Delete transit gateway attachment**.
5. Enter **delete** and choose **Delete**.

To delete a peering attachment using the AWS CLI

Use the [delete-transit-gateway-peering-attachment](#) command.

Connect attachments and Connect peers in AWS Transit Gateway

You can create a *Transit Gateway Connect attachment* to establish a connection between a transit gateway and third-party virtual appliances (such as SD-WAN appliances) running in a VPC. A Connect attachment supports the Generic Routing Encapsulation (GRE) tunnel protocol for high performance, and Border Gateway Protocol (BGP) for dynamic routing. After you create a Connect attachment, you can create one or more GRE tunnels (also referred to as *Transit Gateway Connect peers*) on the Connect attachment to connect the transit gateway and the third-party appliance. You establish two BGP sessions over the GRE tunnel to exchange routing information.

Important

A Transit Gateway Connect peer consists of two BGP peering sessions terminating on AWS-managed infrastructure. The two BGP peering sessions provide routing plane redundancy, ensuring that losing one BGP peering session does not impact your routing operation. The routing information received from both BGP sessions is accumulated for the given Connect peer. The two BGP peering sessions also protect against any AWS infrastructure operations such as routine maintenance, patching, hardware upgrades, and replacements. If your Connect peer is operating without the recommended dual BGP peering session configured for redundancy, it might experience a momentary loss of connectivity during AWS infrastructure operations. We strongly recommend that you configure both the BGP peering sessions on your Connect peer. If you have configured multiple Connect peers to support high availability on the appliance side, we recommend that you configure both the BGP peering sessions on each of your Connect peers.

A Connect attachment uses an existing VPC or Direct Connect attachment as the underlying transport mechanism. This is referred to as the *transport attachment*. The transit gateway identifies matched GRE packets from the third-party appliance as traffic from the Connect attachment. It treats any other packets, including GRE packets with incorrect source or destination information, as traffic from the transport attachment.

Note

To use a Direct Connect attachment as a transport mechanism, you'll first need to integrate Direct Connect with AWS Transit Gateway. For the steps to create this integration, see [Integrate SD-WAN devices with AWS Transit Gateway and Direct Connect](#).

Connect peers

A Connect peer (GRE tunnel) consists of the following components.

Inside CIDR blocks (BGP addresses)

The inside IP addresses that are used for BGP peering. You must specify a /29 CIDR block from the 169.254.0.0/16 range for IPv4. You can optionally specify a /125 CIDR block from the fd00::/8 range for IPv6. The following CIDR blocks are reserved and cannot be used:

- 169.254.0.0/29
- 169.254.1.0/29
- 169.254.2.0/29
- 169.254.3.0/29
- 169.254.4.0/29
- 169.254.5.0/29
- 169.254.169.248/29

You must configure the first address from the IPv4 range on the appliance as the BGP IP address. When you use IPv6, if your inside CIDR block is fd00::/125, then you must configure the first address in this range (fd00::1) on the tunnel interface of the appliance.

The BGP addresses must be unique across all tunnels on a transit gateway.

Peer IP address

The peer IP address (GRE outer IP address) on the appliance side of the Connect peer. This can be any IP address. The IP address can be an IPv4 or IPv6 address, but it must be the same IP address family as the transit gateway address.

Transit gateway address

The peer IP address (GRE outer IP address) on the transit gateway side of the Connect peer. Specify this address from a transit gateway CIDR block. The address must be unique across the entire transit gateway. transit gateway CIDR blocks form a shared address pool used by Private-IP VPN tunnel outside IPs, Client VPN endpoints, and Connect peers. Only one resource can use each address at a time. For details, see "Transit gateway CIDR blocks" on the [Create a transit gateway](#) page. If you do not specify an address, Amazon VPC uses the first available address from the transit gateway CIDR block.

You can add a transit gateway CIDR block when you [create](#) or [modify](#) a transit gateway.

The IP address can be an IPv4 or IPv6 address, but it must be the same IP address family as the peer IP address.

The peer IP address and transit gateway address are used to uniquely identify the GRE tunnel. You can reuse either address across multiple tunnels, but not both in the same tunnel.

Transit Gateway Connect for the BGP peering only supports Multiprotocol BGP (MP-BGP), where IPv4 Unicast addressing is required to also establish a BGP session for IPv6 Unicast. You can use both IPv4 and IPv6 addresses for the GRE outer IP addresses.

The following example shows a Connect attachment between a transit gateway and an appliance in a VPC.

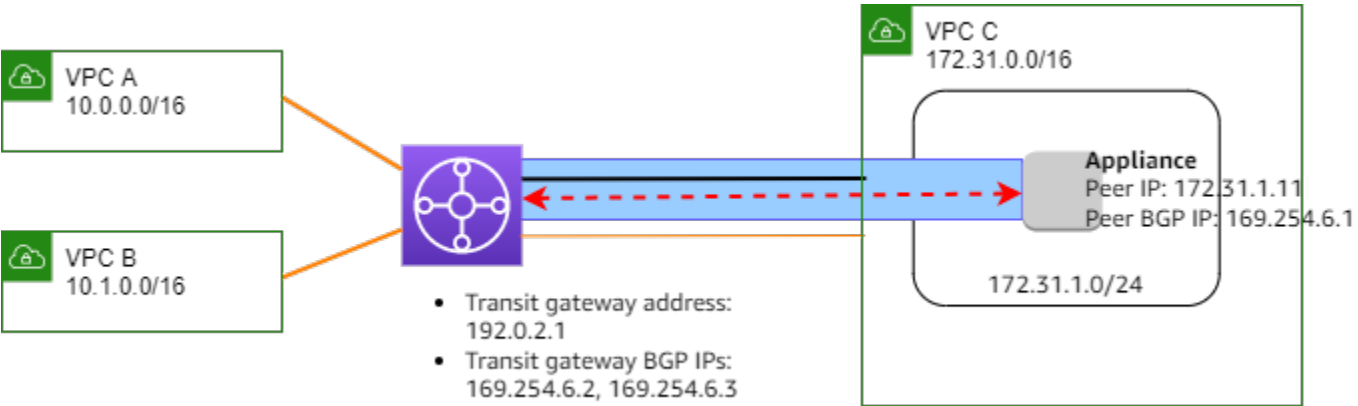


Diagram component	Description
	VPC attachment
	Connect attachment
	GRE tunnel (Connect peer)
	BGP peering session

In the preceding example, a Connect attachment is created on an existing VPC attachment (the transport attachment). A Connect peer is created on the Connect attachment to establish a connection to an appliance in the VPC. The transit gateway address is 192.0.2.1, and the range of BGP addresses is 169.254.6.0/29. The first IP address in the range (169.254.6.1) is configured on the appliance as the peer BGP IP address.

The subnet route table for VPC C has a route that points traffic destined for the transit gateway CIDR block to the transit gateway.

Destination	Target
172.31.0.0/16	Local
192.0.2.0/24	<i>tgw-id</i>

Requirements and considerations

The following are the requirements and considerations for a Connect attachment.

- For information about what Regions support Connect attachments, see the [AWS Transit Gateways FAQ](#).
- The third-party appliance must be configured to send and receive traffic over a GRE tunnel to and from the transit gateway using the Connect attachment.
- The third-party appliance must be configured to use BGP for dynamic route updates and health checks.
- The following types of BGP are supported:
 - Exterior BGP (eBGP): Used for connecting to routers that are in a different autonomous system than the transit gateway. If you use eBGP, you must configure `ebgp-multihop` with a time-to-live (TTL) value of 2.
 - Interior BGP (iBGP): Used for connecting to routers that are in the same autonomous system as the transit gateway. The transit gateway will not install routes from an iBGP peer (third-party appliance), unless the routes are originated from an eBGP peer and should have `next-hop-self` configured. The routes advertised by third-party appliance over the iBGP peering must have an ASN.
 - MP-BGP (multiprotocol extensions for BGP): Used for supporting multiple protocol types, such as IPv4 and IPv6 address families.
- The default BGP keep-alive timeout is 10 seconds and the default hold timer is 30 seconds.
- IPv6 BGP peering is not supported; only IPv4-based BGP peering is supported. IPv6 prefixes are exchanged over IPv4 BGP peering using MP-BGP.
- Bidirectional Forwarding Detection (BFD) is not supported.
- BGP graceful restart is not supported.

- When you create a transit gateway peer, if you do not specify a peer ASN number, we pick the transit gateway ASN number. This means that your appliance and transit gateway will be in the same autonomous system doing iBGP.
- A Connect peer using the BGP AS-PATH attribute is the preferred route when you have two Connect peers.

To use equal-cost multi-path (ECMP) routing between multiple appliances, you must configure the appliance to advertise the same prefixes to the transit gateway with the same BGP AS-PATH attribute. For the transit gateway to choose all of the available ECMP paths, the AS-PATH and Autonomous System Number (ASN) must match. The transit gateway can use ECMP between Connect peers for the same Connect attachment or between Connect attachments on the same transit gateway. The transit gateway cannot use ECMP between both of the redundant BGP peerings a single peer establishes to it.

- With a Connect attachment, the routes are propagated to a transit gateway route table by default.
- Static routes are not supported.
- Configure the GRE tunnel MTU to be smaller than the external interface MTU by subtracting the GRE header (4 bytes) and outer IP header (20 bytes) overhead. For example, if your external interface MTU is 1500 bytes, set the GRE tunnel MTU to 1476 bytes ($1500 - 4 - 20 = 1476$) to prevent packet fragmentation.

Tasks

- [Create a Connect attachment in AWS Transit Gateway](#)
- [Create a Connect peer in AWS Transit Gateway](#)
- [View Connect attachments and Connect peers in AWS Transit Gateway](#)
- [Modify Connect attachment and Connect peer tags in AWS Transit Gateway](#)
- [Delete a Connect peer in AWS Transit Gateway](#)
- [Delete a Connect attachment in AWS Transit Gateway](#)

Create a Connect attachment in AWS Transit Gateway

To create a Connect attachment, you must specify an existing attachment as the transport attachment. You can specify a VPC attachment or a Direct Connect attachment as the transport attachment.

To create a Connect attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit gateway attachments**.
3. Choose **Create transit gateway attachment**.
4. (Optional) For **Name tag**, specify a name tag for the attachment.
5. For **Transit gateway ID**, choose the transit gateway for the attachment.
6. For **Attachment type**, choose **Connect**.
7. For **Transport attachment ID**, choose the ID of an existing attachment (the transport attachment).
8. Choose **Create transit gateway attachment**.

To create a Connect attachment using the AWS CLI

Use the [create-transit-gateway-connect](#) command.

Create a Connect peer in AWS Transit Gateway

You can create a Connect peer (GRE tunnel) for an existing Connect attachment. Before you begin, ensure that you have configured a transit gateway CIDR block. You can configure a transit gateway CIDR block when you [create](#) or [modify](#) a transit gateway.

When you create the Connect peer, you must specify the GRE outer IP address on the appliance side of the Connect peer.

To create a Connect peer using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit gateway attachments**.
3. Select the Connect attachment, and choose **Actions, Create connect peer**.
4. (Optional) For **Name tag**, specify a name tag for the Connect peer.
5. (Optional) For **Transit gateway GRE Address**, specify the GRE outer IP address for the transit gateway. By default, the first available address from the transit gateway CIDR block is used.
6. For **Peer GRE address**, specify the GRE outer IP address for the appliance side of the Connect peer.

7. For **BGP Inside CIDR blocks IPv4**, specify the range of inside IPv4 addresses that are used for BGP peering. Specify a /29 CIDR block from the 169.254.0.0/16 range.
8. (Optional) For **BGP Inside CIDR blocks IPv6**, specify the range of inside IPv6 addresses that are used for BGP peering. Specify a /125 CIDR block from the fd00::/8 range.
9. (Optional) For **Peer ASN**, specify the Border Gateway Protocol (BGP) Autonomous System Number (ASN) for the appliance. You can use an existing ASN assigned to your network. If you do not have one, you can use a private ASN in the 64512–65534 (16-bit ASN) or 4200000000–4294967294 (32-bit ASN) range.

The default is the same ASN as the transit gateway. If you configure the **Peer ASN** to be different than the transit gateway ASN (eBGP), you must configure ebgp-multihop with a time-to-live (TTL) value of 2.

10. Choose **Create connect peer**.

To create a Connect peer using the AWS CLI

Use the [create-transit-gateway-connect-peer](#) command.

View Connect attachments and Connect peers in AWS Transit Gateway

View your Connect attachments and Connect peers.

To view your Connect attachments and Connect peers using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit gateway attachments**.
3. Select the Connect attachment.
4. To view the Connect peers for the attachment, choose the **Connect Peers** tab.

To view your Connect attachments and Connect peers using the AWS CLI

Use the [describe-transit-gateway-connects](#) and [describe-transit-gateway-connect-peers](#) commands.

Modify Connect attachment and Connect peer tags in AWS Transit Gateway

You can modify the tags for your Connect attachment.

To modify your Connect attachment tags using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit Gateway Attachments**.
3. Select the Connect attachment, and then choose **Actions, Manage tags**.
4. To add a tag, choose **Add new tag** and specify the key name and key value.
5. To remove a tag, choose **Remove**.
6. Choose **Save**.

You can modify the tags for your Connect peer.

To modify your Connect peer tags using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit Gateway Attachments**.
3. Select the Connect attachment, and then choose **Connect peers**.
4. Select the Connect peer and then choose **Actions, Manage tags**.
5. To add a tag, choose **Add new tag** and specify the key name and key value.
6. To remove a tag, choose **Remove**.
7. Choose **Save**.

To modify your Connect attachment and Connect peer tags using the AWS CLI

Use the [create-tags](#) and [delete-tags](#) commands.

Delete a Connect peer in AWS Transit Gateway

If you no longer need a Connect peer, you can delete it.

To delete a Connect peer using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit gateway attachments**.
3. Select the Connect attachment.
4. In the **Connect Peers** tab, select the Connect peer and choose **Actions, Delete connect peer**.

To delete a Connect peer using the AWS CLI

Use the [delete-transit-gateway-connect-peer](#) command.

Delete a Connect attachment in AWS Transit Gateway

If you no longer need a Connect attachment, you can delete it. You must first delete any Connect peers for the attachment.

To delete a Connect attachment using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit gateway attachments**.
3. Select the Connect attachment, and choose **Actions, Delete transit gateway attachment**.
4. Enter **delete** and choose **Delete**.

To delete a Connect attachment using the AWS CLI

Use the [delete-transit-gateway-connect](#) command.

Transit gateway route tables in AWS Transit Gateway

Use transit gateway route tables to configure routing for your transit gateway attachments. A route table is a table that contains rules that direct how your network traffic is routed between your VPCs and VPNs. Each route in the table contains the range of IP addresses for the destinations that you want to send traffic to.

Transit gateway route tables allows you to associate a table with a transit gateway attachment. VPC, VPN, VPN Concentrator, Client VPN, Direct Connect gateway, Peering, and Connect attachments are all supported. When associated, routes for these attachments are propagated

from the attachment to the target transit gateway route table. An attachment can be propagated to multiple route tables.

Additionally you can create and manage static routes with a route table. For example, you might have a static route that's used as a backup route in the event of a network disruption that affects any dynamic routes.

Tasks

- [Create a transit gateway route table in AWS Transit Gateway](#)
- [View transit gateway route tables using AWS Transit Gateway](#)
- [Associate a transit gateway route table in AWS Transit Gateway](#)
- [Delete an association for a transit gateway route table in AWS Transit Gateway](#)
- [Enable route propagation to a transit gateway route table in AWS Transit Gateway](#)
- [Disable route propagation in AWS Transit Gateway](#)
- [Create a static route in AWS Transit Gateway](#)
- [Delete a static route in AWS Transit Gateway](#)
- [Replace a static route in AWS Transit Gateway](#)
- [Export route tables to Amazon S3 in AWS Transit Gateway](#)
- [Delete a transit gateway route table in AWS Transit Gateway](#)
- [Create a route table prefix list reference in AWS Transit Gateway](#)
- [Modify a prefix list reference in AWS Transit Gateway](#)
- [Delete a prefix list reference in AWS Transit Gateway](#)

Create a transit gateway route table in AWS Transit Gateway

To create a transit gateway route table using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Choose **Create transit gateway route table**.
4. (Optional) For **Name tag**, type a name for the transit gateway route table. This creates a tag with the tag key "Name", where the tag value is the name that you specify.
5. For **Transit gateway ID**, select the transit gateway for the route table.

6. Choose **Create transit gateway route table**.

To create a transit gateway route table using the AWS CLI

Use the [create-transit-gateway-route-table](#) command.

View transit gateway route tables using AWS Transit Gateway

To view your transit gateway route tables using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. (Optional) To find a specific route table or set of tables, enter all or part of the name, keyword, or attribute in the filter field.
4. Select the checkbox for a route table, or choose its ID, to display information about its associations, propagations, routes, and tags.

To view your transit gateway route tables using the AWS CLI

Use the [describe-transit-gateway-route-tables](#) command.

To view the routes for a transit gateway route table using the AWS CLI

Use the [search-transit-gateway-routes](#) command.

To view the route propagations for a transit gateway route table using the AWS CLI

Use the [get-transit-gateway-route-table-propagations](#) command.

To view the associations for a transit gateway route table using the AWS CLI

Use the [get-transit-gateway-route-table-associations](#) command.

Associate a transit gateway route table in AWS Transit Gateway

You can associate a transit gateway route table with a transit gateway attachment.

To associate a transit gateway route table using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Select the route table.
4. In the lower part of the page, choose the **Associations** tab.
5. Choose **Create association**.
6. Choose the attachment to associate and then choose **Create association**.

To associate a transit gateway route table using the AWS CLI

Use the [associate-transit-gateway-route-table](#) command.

Delete an association for a transit gateway route table in AWS Transit Gateway

You can disassociate a transit gateway route table from a transit gateway attachment.

To disassociate a transit gateway route table using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Select the route table.
4. In the lower part of the page, choose the **Associations** tab.
5. Choose the attachment to disassociate and then choose **Delete association**.
6. When prompted for confirmation, choose **Delete association**.

To disassociate a transit gateway route table using the AWS CLI

Use the [disassociate-transit-gateway-route-table](#) command.

Enable route propagation to a transit gateway route table in AWS Transit Gateway

Use route propagation to add a route from an attachment to a route table.

To propagate a route to a transit gateway attachment route table

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Select the route table for which to create a propagation.
4. Choose **Actions, Create propagation**.
5. On the **Create propagation** page, choose the attachment.
6. Choose **Create propagation**.

To enable route propagation using the AWS CLI

Use the [enable-transit-gateway-route-table-propagation](#) command.

Disable route propagation in AWS Transit Gateway

Remove a propagated route from a route table attachment.

To disable route propagation using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Select the route table to delete the propagation from.
4. On the lower part of the page, choose the **Propagations** tab.
5. Select the attachment and then choose **Delete propagation**.
6. When prompted for confirmation, choose **Delete propagation**.

To disable route propagation using the AWS CLI

Use the [disable-transit-gateway-route-table-propagation](#) command.

Create a static route in AWS Transit Gateway

Create a static route for a VPC, VPN, or transit gateway peering attachment, or you can create a blackhole route that drops traffic that matches the route.

Static routes in a transit gateway route table that target a VPN attachment are not filtered by the Site-to-Site VPN. This might allow unintended outbound traffic flow when using a BGP-based VPN.

To create a static route using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Select the route table for which to create a route.
4. Choose **Actions, Create static route**.
5. On the **Create static route** page, enter the CIDR block for which to create the route, and then choose **Active**.
6. Choose the attachment for the route.
7. Choose **Create static route**.

To create a blackhole route using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Select the route table for which to create a route.
4. Choose **Actions, Create static route**.
5. On the **Create static route** page, enter the CIDR block for which to create the route, and then choose **Blackhole**.
6. Choose **Create static route**.

To create a static route or blackhole route using the AWS CLI

Use the [create-transit-gateway-route](#) command.

Delete a static route in AWS Transit Gateway

Delete static routes from a transit gateway route table.

To delete a static route using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Select the route table for which to delete the route, and choose **Routes**.
4. Choose the route to delete.
5. Choose **Delete static route**.
6. In the confirmation box, choose **Delete static route**.

To delete a static route using the AWS CLI

Use the [delete-transit-gateway-route](#) command.

Replace a static route in AWS Transit Gateway

Replace a static route in a transit gateway route table with a different static route.

To replace a static route using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Choose the route that you want to replace in the route table.
4. In the details section, choose the **Routes** tab.
5. Choose **Actions, Replace static route**.
6. For the **Type**, choose either **Active** or **Blackhole**.
7. From the **Choose attachment** drop-down, choose the transit gateway that will replace the current one in the route table.
8. Choose **Replace static route**.

To replace a static route using the AWS CLI

Use the [replace-transit-gateway-route](#) command.

Export route tables to Amazon S3 in AWS Transit Gateway

You can export the routes in your transit gateway route tables to an Amazon S3 bucket. The routes are saved to the specified Amazon S3 bucket in a JSON file.

To export transit gateway route tables using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Choose the route table that includes the routes to export.
4. Choose **Actions, Export routes**.
5. On the **Export routes** page, for **S3 bucket name**, type the name of the S3 bucket.

6. To filter the routes exported, specify filter parameters in the **Filters** section of the page.
7. Choose **Export routes**.

To access the exported routes, open the Amazon S3 console at <https://console.aws.amazon.com/s3/>, and navigate to the bucket that you specified. The file name includes the AWS account ID, AWS Region, route table ID, and a timestamp. Select the file and choose **Download**. The following is an example of a JSON file that contains information about two propagated routes for VPC attachments.

```
{
  "filter": [
    {
      "name": "route-search.subnet-of-match",
      "values": [
        "0.0.0.0/0",
        "::/0"
      ]
    }
  ],
  "routes": [
    {
      "destinationCidrBlock": "10.0.0.0/16",
      "transitGatewayAttachments": [
        {
          "resourceId": "vpc-0123456abcd123456",
          "transitGatewayAttachmentId": "tgw-attach-1122334455aabbcc1",
          "resourceType": "vpc"
        }
      ],
      "type": "propagated",
      "state": "active"
    },
    {
      "destinationCidrBlock": "10.2.0.0/16",
      "transitGatewayAttachments": [
        {
          "resourceId": "vpc-abcabc123123abca",
          "transitGatewayAttachmentId": "tgw-attach-6677889900aabbcc7",
          "resourceType": "vpc"
        }
      ],
      "type": "propagated",
```

```
    "state": "active"
  }
]
}
```

Delete a transit gateway route table in AWS Transit Gateway

To delete a transit gateway route table using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Route Tables**.
3. Select the route table to delete.
4. Choose **Actions, Delete transit gateway route table**.
5. Enter **delete** and choose **Delete** to confirm the deletion.

To delete a transit gateway route table using the AWS CLI

Use the [delete-transit-gateway-route-table](#) command.

Create a route table prefix list reference in AWS Transit Gateway

You can reference a prefix list in your transit gateway route table. A prefix list is a set of one or more CIDR block entries that you define and manage. You can use a prefix list to simplify the management of the IP addresses that you reference in your resources to route network traffic. For example, if you frequently specify the same destination CIDRs across multiple transit gateway route tables, you can manage those CIDRs in a single prefix list, instead of repeatedly referencing the same CIDRs in each route table. If you need to remove a destination CIDR block, you can remove its entry from the prefix list instead of removing the route from every affected route table.

When you create a prefix list reference in your transit gateway route table, each entry in the prefix list is represented as a route in your transit gateway route table.

For more information about prefix lists, see [Prefix lists](#) in the *Amazon VPC User Guide*.

To create a prefix list reference using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit Gateway Route Tables**.

3. Select the transit gateway route table.
4. Choose **Actions, Create prefix list reference**.
5. For **Prefix list ID**, choose the ID of the prefix list.
6. For **Type**, choose if traffic to this prefix list should be allowed (**Active**) or dropped (**Blackhole**).
7. For **Transit gateway attachment ID**, choose the ID of the attachment to which to route traffic.
8. Choose **Create prefix list reference**.

To create a prefix list reference using the AWS CLI

Use the [create-transit-gateway-prefix-list-reference](#) command.

Modify a prefix list reference in AWS Transit Gateway

You can modify a prefix list reference by changing the attachment that the traffic is routed to, or indicating whether to drop traffic that matches the route.

You cannot modify the individual routes for a prefix list in the **Routes** tab. To modify the entries in the prefix list, use the **Managed Prefix Lists** screen. For more information, see [Modifying a prefix list](#) in the *Amazon VPC User Guide*.

To modify a prefix list reference using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit Gateway Route Tables**.
3. Select the transit gateway route table.
4. In the lower pane, choose **Prefix list references**.
5. Choose the prefix list reference, and choose **Modify references**.
6. For **Type**, choose if traffic to this prefix list should be allowed (**Active**) or dropped (**Blackhole**).
7. For **Transit gateway attachment ID**, choose the ID of the attachment to which to route traffic.
8. Choose **Modify prefix list reference**.

To modify a prefix list reference using the AWS CLI

Use the [modify-transit-gateway-prefix-list-reference](#) command.

Delete a prefix list reference in AWS Transit Gateway

If you no longer need a prefix list reference, you can delete it from your transit gateway route table. Deleting the reference does not delete the prefix list.

To delete a prefix list reference using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit Gateway Route Tables**.
3. Select the transit gateway route table.
4. Choose the prefix list reference, and choose **Delete references**.
5. Choose **Delete references**.

To modify a prefix list reference using the AWS CLI

Use the [delete-transit-gateway-prefix-list-reference](#) command.

Transit gateway policy tables in AWS Transit Gateway

Policy-Based Routing (PBR) for AWS Transit Gateway gives network administrators rule-based control over how traffic is forwarded across a transit gateway. With PBR, forwarding decisions can be based on a combination of packet attributes including source and destination IP addresses, source and destination ports, and protocol rather than destination IP address alone. This enables you to apply different forwarding behaviors to different traffic flows even when those flows share the same destination.

With PBR, you can:

- Steer traffic selectively to security appliances or inspection services based on source, port, or protocol.
- Route traffic from different applications or user populations over different network paths.
- Isolate network traffic into separate routing domains for security segmentation.

PBR is configured through policy tables, which are ordered lists of rules that classify traffic and direct matching packets to a specified transit gateway route table. You associate a policy table with a transit gateway attachment. A policy table replaces the standard route table on an attachment. An attachment can be associated with either a policy table or a route table, but not both.

PBR is available in all AWS Regions where transit gateway is available at no additional charge beyond standard transit gateway fees.

Tasks

- [Policy table concepts in AWS Transit Gateway](#)
- [Prerequisites for transit gateway policy tables in AWS Transit Gateway](#)
- [Create a transit gateway policy table in AWS Transit Gateway](#)
- [View transit gateway policy tables in AWS Transit Gateway](#)
- [Create a policy table entry in AWS Transit Gateway](#)
- [Modify a policy table entry in AWS Transit Gateway](#)
- [Delete a policy table entry in AWS Transit Gateway](#)
- [Associate a transit gateway policy table in AWS Transit Gateway](#)
- [Disassociate a transit gateway policy table in AWS Transit Gateway](#)
- [Delete a transit gateway policy table in AWS Transit Gateway](#)
- [Example: Steering traffic to a security appliance in AWS Transit Gateway](#)
- [Transit gateway policy table limitations in AWS Transit Gateway](#)
- [Troubleshooting transit gateway policy tables in AWS Transit Gateway](#)
- [Related resources for transit gateway policy tables](#)

Policy table concepts in AWS Transit Gateway

This topic describes the key concepts for transit gateway policy tables and Policy-Based Routing (PBR).

Policy tables and policy rules

A **policy table** contains an ordered set of rules. Each rule specifies:

- **Match criteria** — the packet attributes used to classify traffic (source IP CIDR, destination IP CIDR, source port, destination port, and protocol).
- **Target route table** — the transit gateway route table used to forward traffic that matches the rule criteria.

When traffic arrives on an attachment associated with a policy table, the transit gateway evaluates each rule in order and applies the first matching rule's target route table. If no rule matches, the packet is dropped (implicit deny).

A transit gateway attachment can be associated with either a **policy table** or a **route table**, but not both. By default, all attachments are associated with the default route table.

Rule evaluation order

Rules in a policy table are evaluated in **ascending numeric order**, starting at rule number 1. The first rule that matches the incoming traffic is applied. No subsequent rules are evaluated after a match. Because evaluation stops at the first match, rule order matters:

- Place more specific rules (narrow IP ranges, specific ports) at lower rule numbers.
- Place broader or catch-all rules at higher rule numbers.

We recommend that you leave gaps between rule numbers (for example, 100, 110, 120) rather than using consecutive values, so that you can insert rules later without renumbering.

System and customer policy table entries

Policy tables support two types of entries: **customer-managed** and **system-managed**. Understanding both types is important for predicting how your traffic is evaluated and routed.

Customer-managed entries

Customer-managed entries are rules that you define to route traffic based on packet attributes. You create these entries using the `CreateTransitGatewayPolicyTableEntry` API or the AWS Management Console.

Each customer-managed entry specifies:

- **Rule number** — determines evaluation order. Rules are evaluated in ascending order. The lowest-numbered matching rule is applied.
- **Match conditions** — a combination of source CIDR, destination CIDR, protocol, source port, and destination port. All fields are optional. Omitted fields default to Any (*).
- **Target route table** — the transit gateway route table to which matching traffic is forwarded.

If no customer-managed rule matches and no system-managed rule matches, the traffic is dropped (implicit deny).

Example — you have two VPCs connected to a transit gateway: a production VPC and a development VPC. You want to route HTTP traffic destined for `10.0.0.0/16` through a security inspection route table, while all other traffic uses a default route table.

Example customer-managed entries

Rule number	Source CIDR	Destination CIDR	Protocol	Source port	Dest. port	Target route table
10	<code>0.0.0.0/0</code>	<code>10.0.0.0/16</code>	TCP	1024-65535	80	tgw-rtb-inspection
20	<code>0.0.0.0/0</code>	<code>0.0.0.0/0</code>	All	All	All	tgw-rtb-default

In this configuration, HTTP traffic to `10.0.0.0/16` matches rule 10 and is routed through the inspection route table. All other traffic matches rule 20 and uses the default route table.

System-managed entries

System-managed entries are created and maintained automatically by AWS to support AWS-internal routing functions, such as AWS Cloud WAN dynamic routing and network segment isolation. You cannot create, modify, or delete system-managed entries. System-managed entries appear in your policy table when you use features like AWS Cloud WAN that require segment-level traffic isolation on transit gateway-to-Cloud WAN peering attachments.

How system-managed entries affect your traffic:

- **Precedence** — system-managed entries are always evaluated before customer-managed entries. If a system-managed entry matches incoming traffic, it is applied regardless of any customer-managed rules you have configured.
- **Visibility** — system-managed entries are visible in the `GetTransitGatewayPolicyTableEntries` API response and in the console, where their rule number displays as `*`.
- **No action required** — these entries are managed entirely by AWS and do not require configuration on your part.

Example — you use AWS Cloud WAN with two routing segments, production and development. AWS automatically creates system-managed entries on the policy table associated with your transit gateway-to-Cloud WAN peering attachment to ensure traffic stays within its assigned segment. If you also add customer-managed rules to the same policy table, the system-managed segmentation rules take effect first. Your customer-managed rules apply only to traffic that does not match a system-managed entry.

Comparison of entry types

Created by	Rule number	Evaluated	Can be modified
AWS (for example, transit gateway-to-Cloud WAN peering attachments)	*	First, before all customer entries	No
You	1–50,000	After system entries, in ascending rule number order	Yes

Both entry types are returned together by `GetTransitGatewayPolicyTableEntries` and displayed together in the AWS Management Console.

How policy table evaluation works

When traffic enters a transit gateway attachment that is associated with a policy table, evaluation proceeds as follows:

1. System-managed entries are evaluated first. If a system-managed entry matches the traffic, it is applied and evaluation stops.
2. Customer-managed entries are evaluated next, in ascending rule number order. The first matching rule is applied and evaluation stops.
3. Implicit deny — if no entry matches, the traffic is dropped.

Best practices

- **Leave gaps between rule numbers.** Use increments of 10 or 100 (for example, 10, 20, 30 or 100, 200, 300) so you can insert new rules later without renumbering existing entries.

- **Put the most specific rules first.** Place narrower match conditions at lower rule numbers so they are evaluated before broader catch-all rules. A broad rule at a low number will shadow more specific rules at higher numbers.
- **Always include a catch-all rule.** Because traffic is dropped if no rule matches, add a default rule at a high rule number if you want unmatched traffic to reach a route table rather than be silently dropped.
- **Configure protocol before port ranges.** Protocol selection determines whether port range fields are active. Port ranges are only supported for TCP (6) and UDP (17). For ICMPv4 (1), GRE (47), or Any (*), port ranges are automatically set to Any (*).
- **Account for system-managed entries.** If your policy table includes system-managed entries (for example, from AWS Cloud WAN), your customer-managed rules apply only to traffic that does not match a system-managed entry. Review all entries using `GetTransitGatewayPolicyTableEntries` to confirm your expected evaluation order.
- **Verify your configuration with the API.** After making changes, use `GetTransitGatewayPolicyTableEntries` to view all entries in both entry types and confirm that rule numbers and match conditions are correct before routing live traffic.

Prerequisites for transit gateway policy tables in AWS Transit Gateway

Before configuring Policy-Based Routing (PBR) on a transit gateway attachment, confirm the following:

- You have an existing transit gateway. PBR is enabled on all transit gateways by default. No additional configuration is required to activate the feature.
- You have the IAM permissions required to call the PBR APIs. At minimum, the following actions must be allowed on the relevant resources:
 - `ec2:CreateTransitGatewayPolicyTable`
 - `ec2:DescribeTransitGatewayPolicyTables`
 - `ec2>DeleteTransitGatewayPolicyTable`
 - `ec2:AssociateTransitGatewayPolicyTable`
 - `ec2:DisassociateTransitGatewayPolicyTable`
 - `ec2:GetTransitGatewayPolicyTableAssociations`
 - `ec2:GetTransitGatewayPolicyTableEntries`
 - `ec2>CreateTransitGatewayPolicyTableEntry`

- `ec2:ModifyTransitGatewayPolicyTableEntry`
- `ec2:DeleteTransitGatewayPolicyTableEntry`
- The transit gateway route tables you intend to reference as targets already exist.
- If an attachment is currently associated with a route table, you must disassociate it before associating a policy table.

For information about resource-level IAM conditions for PBR, see the example policy in [Associate a transit gateway policy table](#).

Create a transit gateway policy table in AWS Transit Gateway

Create a policy table and associate it with your transit gateway.

To create a transit gateway policy table using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, under **Transit gateways**, choose **Transit Gateway Policy Tables**.
3. Choose **Create transit gateway policy table**.
4. For **Transit gateway ID**, select your transit gateway from the dropdown list.
5. (Optional) Under **Tags**, add tags to help identify the policy table.
6. Choose **Create transit gateway policy table**.

The policy table enters a **Pending** state. Wait for the state to change to **Available** before proceeding.

To create a transit gateway policy table using the AWS CLI

Use the [create-transit-gateway-policy-table](#) command. Replace `tgw-0bc994abffEXAMPLE` with your transit gateway ID.

```
aws ec2 create-transit-gateway-policy-table \  
    --transit-gateway-id tgw-0bc994abffEXAMPLE
```

Example response:

```
{  
    "TransitGatewayPolicyTable": {
```

```
"TransitGatewayPolicyTableId": "tgw-ptb-0ca78a549EXAMPLE",
"TransitGatewayId": "tgw-0bc994abffEXAMPLE",
"State": "pending"
}
}
```

View transit gateway policy tables in AWS Transit Gateway

You can view your transit gateway policy tables and their entries.

View policy tables

To view transit gateway policy tables using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, under **Transit Gateways**, choose **Transit Gateway Policy Tables**.
3. To filter by transit gateway, use the search bar and filter on **Transit gateway ID**.
4. The list shows each policy table's **Transit gateway policy table ID**, **Transit gateway ID**, and **State**.
5. Select a policy table to view its details, associations, and entries on the detail page.

To track your entry usage against the per-transit gateway limit, use the Service Quotas console. See [Quotas](#) for details.

To view transit gateway policy tables using the AWS CLI

To list all policy tables on your account, use the [describe-transit-gateway-policy-tables](#) command.

```
aws ec2 describe-transit-gateway-policy-tables \
  --filters "Name=transit-gateway-id,Values=tgw-0bc994abffEXAMPLE"
```

Example response:

```
{
  "TransitGatewayPolicyTables": [
    {
      "TransitGatewayPolicyTableId": "tgw-ptb-0ca78a549EXAMPLE",
      "TransitGatewayId": "tgw-0bc994abffEXAMPLE",
      "State": "available",
      "CreationTime": "2026-01-01T00:00:00.000Z",

```

```

        "Tags": []
      }
    ]
  }

```

View policy table entries

To view policy table entries using the console

1. In **Transit Gateway Policy Tables**, select the policy table.
2. Choose the **Entries** tab.
3. The **Entries** tab displays all entries with the following columns: **Rule number**, **State**, **Target route table**, **Source CIDR block**, **Source port**, **Destination CIDR block**, **Destination port**, and **Protocol**. System-managed entries display * as their rule number.
4. To filter entries, use the search bar. You can filter by rule number, target route table, source CIDR, destination CIDR, source port, destination port, or protocol.

To check total usage against the per-transit gateway limit, use the Service Quotas console. See [Quotas](#) for details.

To view policy table entries using the AWS CLI

Use the [get-transit-gateway-policy-table-entries](#) command.

```

aws ec2 get-transit-gateway-policy-table-entries \
  --transit-gateway-policy-table-id tgw-ptb-0ca78a549EXAMPLE

```

Example response:

```

{
  "TransitGatewayPolicyTableEntries": [
    {
      "PolicyRuleNumber": "100",
      "PolicyRule": {
        "SourceCidrBlock": "10.100.0.0/16",
        "SourcePortRange": "*",
        "DestinationCidrBlock": "*",
        "DestinationPortRange": "443",
        "Protocol": "6"
      }
    },

```

```
        "TargetRouteTableId": "tgw-rtb-0ca78a549EXAMPLE",
        "State": "active"
    },
    {
        "PolicyRuleNumber": "*",
        "PolicyRule": {
            "SourceCidrBlock": "*",
            "SourcePortRange": "*",
            "DestinationCidrBlock": "*",
            "DestinationPortRange": "*",
            "Protocol": "*",
            "MetaData": {
                "MetaDataKey": "TgwRouteTable",
                "MetaDataValue": "tgw-rtb-0ca62a892EXAMPLE"
            }
        },
        "TargetRouteTableId": "tgw-rtb-0ca62a892EXAMPLE",
        "State": "active"
    }
]
}
```

Supported filters

The following filters are supported for GetTransitGatewayPolicyTableEntries.

Supported filters

Filter	Description
policy-rule-number	Filter by rule number
target-route-table-id	Filter by target route table ID
policy-rule.source-ip	Filter by source CIDR block
policy-rule.destination-ip	Filter by destination CIDR block
policy-rule.source-port	Filter by source port range
policy-rule.destination-port	Filter by destination port range

Filter	Description
policy-rule.protocol	Filter by protocol
policy-rule.meta-data.key	Filter by metadata key
policy-rule.meta-data.value	Filter by metadata value

Create a policy table entry in AWS Transit Gateway

Add one or more rules to the policy table. Each rule specifies match criteria and a target route table. All match attributes are optional; omitting an attribute defaults it to Any (*).

To create a policy table entry using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In **Transit Gateway Policy Tables**, select the policy table you created.
3. Choose the **Entries** tab, and then choose **Create entry**.
4. Under **Entry details**, configure the required fields.
 - a. **Rule number** — enter an integer from 1 to 50,000. Rules are evaluated lowest-number-first. Leave gaps between numbers (for example, 100, 110, 120) so you can insert rules later without renumbering.
 - b. **Target transit gateway route table** — select the transit gateway route table to use for matching traffic.
5. Under **Rule conditions**, configure the optional match criteria. Leave any field empty to match all values. All specified conditions must match for the rule to apply.
 - a. **Source CIDR block** — IPv4 or IPv6 CIDR (for example, 10.0.0.0/16). Leave blank for all.
 - b. **Destination CIDR block** — IPv4 or IPv6 CIDR. Leave blank for all.
 - c. **Protocol** — select a supported protocol. Supported values: 1 (ICMPv4), 6 (TCP), 17 (UDP), 47 (GRE), or Any to match all protocols (*). Port range fields are only available when Protocol is TCP (6) or UDP (17). For all other protocols — ICMPv4, GRE, or Any — port ranges are automatically set to Any (*) and cannot be configured.

- d. **Source port range** — single port (443) or range (1024-65535). Leave blank for all. Available only when Protocol is TCP (6) or UDP (17).
 - e. **Destination port range** — single port (443) or range (1024-65535). Leave blank for all. Available only when Protocol is TCP (6) or UDP (17).
6. Choose **Create entry**.

To create a policy table entry using the AWS CLI

Use the [create-transit-gateway-policy-table-entry](#) command. Replace the policy table ID, rule number, policy rule values, and route table ID with your values.

```
aws ec2 create-transit-gateway-policy-table-entry \
  --transit-gateway-policy-table-id tgw-ptb-0ca78a549EXAMPLE \
  --policy-rule-number 100 \
  --policy-rule '{"SourceCidrBlock": "10.100.0.0/16", "Protocol": "6",
  "DestinationPortRange": "443"}' \
  --target-route-table-id tgw-rtb-0a823eddbdeEXAMPLE
```

Policy rule match attributes

The following table describes the policy rule match attributes.

Policy rule match attributes

Attribute	Description	Valid values	Default
SourceCidrBlock	Source IP address or CIDR block	IPv4 or IPv6 CIDR (for example, 10.0.0.0/16), or omit for any	* (any)
SourcePortRange	Source port or range. Only evaluated when Protocol is TCP (6) or UDP (17). Automatically set to * for all other protocols.	Single port (443) or range (1024-65535); valid range 0–65535, start ≤ end	* (any)
DestinationCidrBlock	Destination IP address or CIDR block	IPv4 or IPv6 CIDR, or omit for any	* (any)

Attribute	Description	Valid values	Default
DestinationPortRange	Destination port or range. Only evaluated when Protocol is TCP (6) or UDP (17). Automatically set to * for all other protocols.	Single port or range; valid range 0–65535, start ≤ end	* (any)
Protocol	IANA protocol number. Determines whether port range fields are evaluated.	1 (ICMPv4), 6 (TCP), 17 (UDP), 47 (GRE), or * for any. Port ranges are only supported for TCP and UDP.	* (any)

Both IPv4 and IPv6 CIDR formats are supported.

Modify a policy table entry in AWS Transit Gateway

ModifyTransitGatewayPolicyTableEntry is a **PATCH-style update**. Only the fields you explicitly include in the request are modified. Fields you omit are left unchanged. They are **not** reset to Any (*).

Important

To change a field back to match any value (Any), you must explicitly pass * for that field. Omitting the field has no effect on its current value. For example, if a rule has "SourceCidrBlock": "10.0.0.0/16" and you want it to match any source IP, you must pass "SourceCidrBlock": "*". Omitting SourceCidrBlock leaves the 10.0.0.0/16 restriction in place.

At least one of --target-route-table-id or --policy-rule must be included in the request.

To modify a policy table entry using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, under **Transit Gateways**, choose **Transit Gateway Policy Tables**.
3. In **Transit Gateway Policy Tables**, select the policy table.
4. Choose the **Entries** tab.

5. Select the entry you want to modify. System entries (rule number *) are read-only and cannot be selected for editing.
6. Choose **Modify entry**.
7. The **Rule number** is displayed as read-only — it identifies the entry and cannot be changed. To change a rule number, delete the entry and recreate it with the new number.
8. Update only the fields you want to change. All other fields are pre-populated with current values. CIDR fields provide an autosuggest dropdown with common prefixes.
 - To reset a field to Any (match any), clear the field — the console substitutes * for any empty field. Leaving it as-is retains the current value.
9. Choose **Save changes**.

The console returns the fully merged entry, reflecting both updated and retained field values.

To modify a policy table entry using the AWS CLI

Use the [modify-transit-gateway-policy-table-entry](#) command.

```
aws ec2 modify-transit-gateway-policy-table-entry \
  --transit-gateway-policy-table-id tgw-ptb-0ca78a549EXAMPLE \
  --policy-rule-number 100 \
  --policy-rule '{"SourceCidrBlock":"11.0.0.0/16","Protocol":"17"}' \
  --target-route-table-id tgw-rtb-0a823edbdeEXAMPLE
```

The response returns the fully merged entry. Only `SourceCidrBlock`, `Protocol`, and `TargetRouteTableId` were updated — `SourcePortRange`, `DestinationCidrBlock`, and `DestinationPortRange` retain their previous values unchanged:

```
{
  "TransitGatewayPolicyTableEntry": {
    "PolicyRuleNumber": "100",
    "PolicyRule": {
      "SourceCidrBlock": "11.0.0.0/16",
      "SourcePortRange": "1024-65535",
      "DestinationCidrBlock": "192.168.0.0/16",
      "DestinationPortRange": "443",
      "Protocol": "17"
    },
    "TargetRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
```

```

    "State": "active"
  }
}

```

Example: Reset a field to Any (match any)

To remove the destination port restriction set in the example above and allow any destination port, explicitly pass `*` for `DestinationPortRange`. Omitting the field would leave 443 in place.

```

aws ec2 modify-transit-gateway-policy-table-entry \
  --transit-gateway-policy-table-id tgw-ptb-0ca78a549EXAMPLE \
  --policy-rule-number 100 \
  --policy-rule '{"DestinationPortRange": "*'}'

```

```

{
  "TransitGatewayPolicyTableEntry": {
    "PolicyRuleNumber": "100",
    "PolicyRule": {
      "SourceCidrBlock": "11.0.0.0/16",
      "SourcePortRange": "1024-65535",
      "DestinationCidrBlock": "192.168.0.0/16",
      "DestinationPortRange": "*",
      "Protocol": "17"
    },
    "TargetRouteTableId": "tgw-rtb-0a823eddbdeEXAMPLE",
    "State": "active"
  }
}

```

Delete a policy table entry in AWS Transit Gateway

Delete a customer-managed entry from a transit gateway policy table. System-managed entries cannot be deleted.

Important

If you delete all customer-managed entries and there are no applicable system-managed entries, the transit gateway drops all packets arriving on attachments associated with that table. Add at least one rule or re-associate the attachment with a route table to restore traffic forwarding.

To delete a policy table entry using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, under **Transit Gateways**, choose **Transit Gateway Policy Tables**.
3. In **Transit Gateway Policy Tables**, select the policy table.
4. Choose the **Entries** tab.
5. Select the customer-managed entry you want to delete. System entries cannot be deleted.
6. Choose **Delete**.
7. In the confirmation dialog, review the entry details (rule number, target route table, source CIDR block, source port, destination CIDR block, destination port, protocol). Acknowledge the message *Once deleted, you'll need to create a new entry with the same attributes to restore it* and choose **Delete**.

A success banner appears: *Policy table entry with rule number {ruleNumber} was successfully deleted.*

To delete a policy table entry using the AWS CLI

Use the [delete-transit-gateway-policy-table-entry](#) command. The deleted entry is returned in the response with "State": "deleted".

```
aws ec2 delete-transit-gateway-policy-table-entry \  
  --transit-gateway-policy-table-id tgw-ptb-0ca78a549EXAMPLE \  
  --policy-rule-number 100
```

```
{  
  "TransitGatewayPolicyTableEntry": {  
    "PolicyRuleNumber": "100",  
    "PolicyRule": {  
      "SourceCidrBlock": "10.100.0.0/16",  
      "SourcePortRange": "*",  
      "DestinationCidrBlock": "*",  
      "DestinationPortRange": "443",  
      "Protocol": "6"  
    },  
    "TargetRouteTableId": "tgw-rtb-0a823eddbdeEXAMPLE",  
    "State": "deleted"  
  }  
}
```

Note

You cannot delete a transit gateway route table that is referenced as a target by any policy table entry. Use `GetTransitGatewayPolicyTableEntries` with a `target-route-table-id` filter to identify referencing entries, then update or delete them before retrying the route table deletion. Unlike route entries (which are black-holed when a target attachment is deleted), policy table entries are not automatically cleaned up. They must be explicitly removed.

Associate a transit gateway policy table in AWS Transit Gateway

Associate a policy table with one or more transit gateway attachments. An attachment can be associated with either a policy table or a route table, but not both. If the attachment currently has a route table associated, you must disassociate it first.

Customer-managed policy based routing is compatible with all transit gateway attachment types: VPC, Direct Connect, VPN (Site-to-Site), Client VPN, VPN Concentrator, Network Functions, transit gateway Connect, and Peering. The exception is transit gateway-to-Cloud WAN (CWAN) peering attachments. For details, see [Limitations](#).

You can associate the same policy table with multiple attachments. All traffic arriving on those attachments is evaluated against the same rule set.

Disassociate an existing route table

If the attachment is currently associated with a route table, disassociate it first.

To disassociate a route table using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit Gateway Route Tables** and select the route table.
3. Choose the **Associations** tab.
4. Select the attachment row and choose **Delete association**.
5. In the confirmation dialog, choose **Delete association**.

To disassociate a route table using the AWS CLI

Use the [disassociate-transit-gateway-route-table](#) command.

```
# Disassociate existing route table (if applicable)
aws ec2 disassociate-transit-gateway-route-table \
  --transit-gateway-attachment-id tgw-attach-0def6EXAMPLE \
  --transit-gateway-route-table-id tgw-rtb-CURRENTEXAMPLE
```

Example response. The association transitions to the disassociating state:

```
{
  "Association": {
    "TransitGatewayRouteTableId": "tgw-rtb-CURRENTEXAMPLE",
    "TransitGatewayAttachmentId": "tgw-attach-0def6EXAMPLE",
    "ResourceId": "vpc-0abcd1234EXAMPLE",
    "ResourceType": "vpc",
    "State": "disassociating"
  }
}
```

Associate the policy table

To associate a policy table using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, under **Transit Gateways**, choose **Transit Gateway Policy Tables** and select your policy table.
3. Choose the **Associations** tab.
4. Choose **Create association**.
5. On the **Create association** modal, the **Transit gateway policy table** and **Transit gateway ID** fields are pre-filled.
6. For **Transit gateway attachment**, select the attachment from the dropdown. The dropdown lists all eligible attachments on the same transit gateway (VPC, Direct Connect, VPN, transit gateway Connect, Peering).
7. Choose **Create**.

The association **State** transitions from **Associating** to **Associated**. You can repeat these steps to associate additional attachments with the same policy table.

To associate a policy table using the AWS CLI

Use the [associate-transit-gateway-policy-table](#) command.

```
# Associate the policy table
aws ec2 associate-transit-gateway-policy-table \
    --transit-gateway-policy-table-id tgw-ptb-0ca78a549EXAMPLE \
    --transit-gateway-attachment-id tgw-attach-0def6EXAMPLE
```

Example response. The association starts in the associating state and transitions to associated:

```
{
  "Association": {
    "TransitGatewayPolicyTableId": "tgw-ptb-0ca78a549EXAMPLE",
    "TransitGatewayAttachmentId": "tgw-attach-0def6EXAMPLE",
    "ResourceType": "vpc",
    "State": "associating"
  }
}
```

IAM condition example

You can use resource-level conditions to restrict which attachments or policy tables a principal can act on. For example, the following policy allows a user to associate policy tables only with attachments tagged network: Prod Network:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": ["ec2:AssociateTransitGatewayPolicyTable"],
    "Resource": "arn:aws:ec2:*:*:transit-gateway-attachment/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/network": "Prod Network"
      }
    }
  }]
}
```

Disassociate a transit gateway policy table in AWS Transit Gateway

Disassociate a policy table from a transit gateway attachment.

Important

After the association is removed, the attachment has no route table or policy table associated with it. All ingressing traffic is dropped until you associate the attachment with a route table or a new policy table.

To disassociate a policy table using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, under **Transit Gateways**, choose **Transit Gateway Policy Tables**, and select the policy table.
3. Choose the **Associations** tab.
4. Select the association you want to remove.
5. Choose **Delete association**.
6. In the confirmation dialog, review the policy table ID and attachment ID. Acknowledge the warning: *Removing this association will disable policy-based routing for traffic from this attachment. All ingress traffic from this attachment will be dropped until you associate it with a route table or a new policy table.* Choose **Delete**.

A success banner appears: *Successfully deleted association from {tgw-attach-id}*.

To disassociate a policy table using the AWS CLI

Use the [disassociate-transit-gateway-policy-table](#) command.

```
aws ec2 disassociate-transit-gateway-policy-table \
  --transit-gateway-policy-table-id tgw-ptb-0ca78a549EXAMPLE \
  --transit-gateway-attachment-id tgw-attach-0def6EXAMPLE
```

Example response. The association transitions to the disassociating state:

```
{
  "Association": {
```

```
"TransitGatewayPolicyTableId": "tgw-ptb-0ca78a549EXAMPLE",  
"TransitGatewayAttachmentId": "tgw-attach-0def6EXAMPLE",  
"State": "disassociating"  
}  
}
```

Delete a transit gateway policy table in AWS Transit Gateway

Delete a transit gateway policy table. When a table is deleted, all policy rules within that table are deleted.

Important

Before deleting a policy table, disassociate all attachments from it. Any attachment that loses its policy table association without being reassociated with a route table will drop all ingressing traffic.

To delete a transit gateway policy table using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, under **Transit gateways**, choose **Transit Gateway Policy Tables**.
3. Select the transit gateway policy table to delete.
4. Choose **Actions**, and then choose **Delete policy table**.
5. In the confirmation dialog, confirm that you want to delete the table.

To delete a transit gateway policy table using the AWS CLI

Use the [delete-transit-gateway-policy-table](#) command.

```
aws ec2 delete-transit-gateway-policy-table \  
--transit-gateway-policy-table-id tgw-ptb-0ca78a549EXAMPLE
```

Example response. The table transitions to the deleting state:

```
{  
  "TransitGatewayPolicyTable": {  
    "TransitGatewayPolicyTableId": "tgw-ptb-0ca78a549EXAMPLE",  
    "TransitGatewayId": "tgw-0bc994abffEXAMPLE",
```

```
    "State": "deleting",
    "CreationTime": "2026-01-01T00:00:00.000Z"
  }
}
```

Example: Steering traffic to a security appliance in AWS Transit Gateway

The following example steers traffic from a specific internal subnet to a firewall VPC for inspection, while routing all other traffic directly.

Scenario

Consider the following requirements.

- Traffic from 10.1.10.0/24 (sensitive workload subnet) must pass through a firewall in a VPC before forwarding.
- All other traffic forwards using the standard route table.

Policy table configuration

Configure the policy table with the following entries.

Policy table configuration

Rule #	Entry type	Source CIDR block	Destination CIDR block	Protocol	Source port range	Destination port range	Target route table
100	customer-managed	10.1.10.0/24	Any	Any	Any	Any	tgw-rtb-firewall
200	customer-managed	Any	Any	Any	Any	Any	tgw-rtb-default

How it works

Traffic is evaluated as follows.

- Traffic from `10.1.10.0/24` arrives on the transit gateway attachment associated with the policy table.
- The transit gateway evaluates rule 100 first. The source CIDR matches `10.1.10.0/24`, so the traffic is forwarded using route table `tgw-rtb-firewall`.
- Route table `tgw-rtb-firewall` contains routes that send traffic to the firewall VPC attachment.
- The firewall inspects the traffic and forwards it back to the transit gateway.
- On return, the transit gateway re-evaluates the traffic for onward routing using destination-based lookup on the route table associated with the firewall VPC attachment.

All other traffic (not from `10.1.10.0/24`) does not match rule 100 and falls through to rule 200, which forwards it using the default route table without inspection.

AWS CLI commands to create this configuration

Use the following commands to create the policy table, add the rules, and associate the table with the source VPC attachment.

```
# Create the policy table
aws ec2 create-transit-gateway-policy-table \
    --transit-gateway-id tgw-0bc994abffEXAMPLE

# Add rule 100: steer sensitive traffic to firewall route table
aws ec2 create-transit-gateway-policy-table-entry \
    --transit-gateway-policy-table-id tgw-ptb-0ca78a549EXAMPLE \
    --policy-rule-number 100 \
    --policy-rule '{"SourceCidrBlock": "10.1.10.0/24"}' \
    --target-route-table-id tgw-rtb-firewall

# Add rule 200: catch-all for remaining traffic
aws ec2 create-transit-gateway-policy-table-entry \
    --transit-gateway-policy-table-id tgw-ptb-0ca78a549EXAMPLE \
    --policy-rule-number 200 \
    --policy-rule '{}' \
    --target-route-table-id tgw-rtb-default

# Associate policy table with the source VPC attachment
aws ec2 associate-transit-gateway-policy-table \
    --transit-gateway-policy-table-id tgw-ptb-0ca78a549EXAMPLE \
    --transit-gateway-attachment-id tgw-attach-0def6EXAMPLE
```

Transit gateway policy table limitations in AWS Transit Gateway

The following limitations apply to transit gateway policy tables at launch.

- **TGW-to-Cloud WAN (CWAN) peering attachments** — PBR is not supported for customer-managed entries on these attachments. Traffic on CWAN peering attachments is controlled exclusively by system-managed entries, which have higher priority and are read-only. You can add customer entries to the same policy table for use with other attachment types.
- **BGP route advertisement** — AWS Transit Gateway does not advertise any routes to BGP-speaking attachments (Site-to-Site VPN and Connect appliances) that are associated with a policy table. Associating a policy table with one of these attachments stops all BGP route advertisements to the peer.

Direct Connect behaves differently: the prefixes advertised to the customer gateway are governed by the allowed-prefixes list on the Direct Connect gateway association, so those advertisements continue regardless of policy-table association.

If your design relies on TGW-advertised BGP routes, do not associate a policy table with the attachment until you have verified the impact on your routing topology.

- **Attachment exclusivity** — an attachment can be associated with either a policy table or a route table, but not both. If you attempt to associate a policy table with an attachment that already has a route table associated, the request fails. Disassociate the route table first.
- **System entry modification** — system-managed entries cannot be modified or deleted by customers at this time. These entries are managed entirely by AWS and are visible in the `GetTransitGatewayPolicyTableEntries` API response and in the console with a rule number of *.

Troubleshooting transit gateway policy tables in AWS Transit Gateway

Use the following information to diagnose and resolve issues with transit gateway policy tables.

Traffic is being dropped unexpectedly

Check the `PacketDropCountNoPolicy` CloudWatch metric. If it is incrementing, traffic is arriving on an attachment associated with a policy table but not matching any rule. Review your rule match criteria and ensure a catch-all rule (all attributes set to Any) is present at a high rule number if you want unmatched traffic to have a default forwarding path.

A rule is not matching traffic I expect it to match

Verify rule evaluation order. Rules are evaluated in ascending numeric order and the first match wins. A broader rule at a lower number may be matching your traffic before the more specific rule you intended. Review your rule numbers and reorder as needed.

Use `GetTransitGatewayPolicyTableEntries` to view all entries, including system-managed entries. System-managed entries are evaluated before customer-managed entries and may be matching your traffic.

Port ranges are not being applied to my rule

Port ranges are only evaluated when Protocol is TCP (6) or UDP (17). If your rule specifies ICMPv4 (1), GRE (47), or Any (*) as the protocol, any port range values are automatically set to Any (*) and are not evaluated. To use port-based matching, set Protocol to TCP or UDP.

If you submitted port ranges with a non-TCP/non-UDP protocol via the API, the request is rejected with a validation error. Confirm the protocol value in your request.

I cannot delete a route table

A transit gateway route table that is referenced as a target by any policy table entry cannot be deleted. Use `GetTransitGatewayPolicyTableEntries` with a filter on `target-route-table-id` to identify which rules reference the table, then update or delete those rules before retrying the deletion.

I cannot associate a policy table to my attachment

Confirm that the attachment does not already have a route table associated with it. If it does, disassociate the route table first. See [Associate a transit gateway policy table](#) for instructions. Also confirm that the attachment type is supported — TGW-to-CWAN peering attachments do not support customer-managed policy based routing. See [Limitations](#).

My policy table is empty and traffic is being dropped

An attachment associated with an empty policy table drops all ingressing packets. This is the same behavior as an empty route table. Add at least one rule, or reassociate the attachment with a route table, to restore traffic forwarding.

Related resources for transit gateway policy tables

For more information, see the following resources:

- [the section called “Transit gateway route tables”](#)
- [VPC attachments](#)
- [AWS Network Firewall with transit gateway](#)
- [CloudWatch metrics in AWS Transit Gateway](#)

Multicast in AWS Transit Gateway

Multicast is a communication protocol used for delivering a single stream of data to multiple receiving computers simultaneously. Transit Gateway supports routing multicast traffic between subnets of attached VPCs, and it serves as a multicast router for instances sending traffic destined for multiple receiving instances.

Topics

- [Multicast concepts](#)
- [Considerations](#)
- [Multicast routing](#)
- [Multicast domains in AWS Transit Gateway](#)
- [Shared multicast domains in AWS Transit Gateway](#)
- [Register sources with a multicast group in AWS Transit Gateway](#)
- [Register members with a multicast group in AWS Transit Gateway](#)
- [Deregister sources from a multicast group in AWS Transit Gateway](#)
- [Deregister members from a multicast group in AWS Transit Gateway](#)
- [View multicast groups in AWS Transit Gateway](#)
- [Set up multicast for Windows Server in AWS Transit Gateway](#)
- [Example: Manage IGMP configurations using AWS Transit Gateway](#)
- [Example: Manage static source configurations in AWS Transit Gateway](#)
- [Example: Manage static group member configurations in AWS Transit Gateway](#)

Multicast concepts

The following are the key concepts for multicast:

- **Multicast domain** — Allows segmentation of a multicast network into different domains, and makes the transit gateway act as multiple multicast routers. You define multicast domain membership at the subnet level.
- **Multicast group** — Identifies a set of hosts that will send and receive the same multicast traffic. A multicast group is identified by a group IP address. Multicast group membership is defined by individual elastic network interfaces attached to EC2 instances.
- **Internet Group Management Protocol (IGMP)** — An internet protocol that allows hosts and routers to dynamically manage multicast group membership. An IGMP multicast domain contains hosts that use the IGMP protocol to join, leave, and send messages. AWS supports the IGMPv2 protocol and both IGMP and static (API-based) group membership multicast domains.
- **Multicast source** — An elastic network interface associated with a supported EC2 instance that is statically configured to send multicast traffic. A multicast source only applies to static source configurations.

A static source multicast domain contains hosts that do not use the IGMP protocol to join, leave, and send messages. You use the AWS CLI to add a source and group members. The statically-added source sends multicast traffic and the members receive multicast traffic.

- **Multicast group member** — An elastic network interface associated with a supported EC2 instance that receives multicast traffic. A multicast group has multiple group members. In a static source group membership configuration, multicast group members can only receive traffic. In an IGMP group configuration, members can both send and receive traffic.

Considerations

- Transit gateway multicast may not be suitable for high-frequency trading or performance-sensitive applications. We strongly recommend that you review the [Multicast quotas](#) for the limits. Contact your account or Solution Architect team for a detailed review of your performance requirements.
- For information about supported Regions, see [AWS Transit Gateway FAQs](#).
- You must create a new transit gateway to support multicast.
- Multicast group membership is managed using the Amazon Virtual Private Cloud Console or the AWS CLI, or IGMP.
- A subnet can only be in one multicast domain.

- If you use a non-Nitro instance, you must disable the **Source/Dest** checkbox. For information about disabling the check, see [Changing the source or destination checking](#) in the *Amazon EC2 User Guide*.
- A non-Nitro instance cannot be a multicast sender.
- Multicast routing is not supported over Direct Connect, Site-to-Site VPN, peering attachments, or transit gateway Connect attachments.
- A transit gateway does not support fragmentation of multicast packets. Fragmented multicast packets are dropped. For more information, see [Maximum transmission unit \(MTU\)](#).
- At startup, an IGMP host sends multiple IGMP JOIN messages to join a multicast group (typically 2 to 3 retries). In the unlikely event that all the IGMP JOIN messages get lost, the host will not become part of transit gateway multicast group. In such a scenario you will need to re-trigger the IGMP JOIN message from the host using application specific methods.
- A group membership starts with the receipt of IGMPv2 JOIN message by the transit gateway and ends with the receipt of the IGMPv2 LEAVE message. The transit gateway keeps track of hosts that successfully joined the group. As a cloud multicast router, transit gateway issues an IGMPv2 QUERY message to all members every two minutes. Each member sends an IGMPv2 JOIN message in response, which is how the members renew their membership. If a member fails to reply to three consecutive queries, the transit gateway removes this membership from all joined groups. However, it continues sending queries to this member for 12 hours before permanently removing the member from its to-be-queried list. An explicit IGMPv2 LEAVE message immediately and permanently removes the host from any further multicast processing.
- The transit gateway keeps track of hosts that successfully joined the group. In the event of a transit gateway outage, the transit gateway continues to send multicast data to the host for seven minutes (420 seconds) after the last successful IGMP JOIN message. The transit gateway continues to send membership queries to the host for up to 12 hours or until it receives a IGMP LEAVE message from the host.
- The transit gateway sends membership query packets to all the IGMP members so that it can track multicast group membership. The source IP of these IGMP query packets is 0.0.0.0/32, and the destination IP is 224.0.0.1/32 and the protocol is 2. Your security group configuration on the IGMP hosts (instances), and any ACLs configuration on the host subnets must allow these IGMP protocol messages.
- When the multicast source and destination are in the same VPC, you cannot use security group referencing to set the destination security group to accept traffic from the source's security group.

- For static multicast groups and sources, AWS Transit Gateway automatically remove static groups and sources for ENIs that no longer exist. This is performed by periodically assuming the [Transit Gateway service-linked role](#) to describe ENIs in the account.
- Only static multicast supports IPv6. Dynamic multicast does not.

Multicast routing

When you enable multicast on a transit gateway, it acts as a multicast router. When you add a subnet to a multicast domain, we send all multicast traffic to the transit gateway that is associated with that multicast domain.

Network ACLs

Network ACL rules operate at the subnet level. They apply to multicast traffic, because transit gateways reside outside of the subnet. For more information, see [Network ACLs](#) in the *Amazon VPC User Guide*.

For Internet Group Management Protocol (IGMP) multicast traffic, the following are the minimum inbound rules. The remote host is the host sending the multicast traffic.

Type	Protocol	Source	Description
Custom Protocol	IGMP(2)	0.0.0.0/32	IGMP query
Custom UDP Protocol	UDP	Remote host IP address	Inbound multicast traffic

The following are the minimum outbound rules for IGMP.

Type	Protocol	Destination	Description
Custom Protocol	IGMP(2)	224.0.0.2/32	IGMP leave
Custom Protocol	IGMP(2)	Multicast group IP address	IGMP join
Custom UDP Protocol	UDP	Multicast group IP address	Outbound multicast traffic

Security groups

Security group rules operate at the instance level. They can be applied to both inbound and outbound multicast traffic. The behavior is the same as with unicast traffic. For all group member instances, you must allow inbound traffic from the group source. For more information, see [Security groups](#) in the *Amazon VPC User Guide*.

For IGMP multicast traffic, you must have the following inbound rules at a minimum. The remote host is the host sending the multicast traffic. You can't specify a security group as the source of the UDP inbound rule.

Type	Protocol	Source	Description
Custom Protocol	2	0.0.0.0/32	IGMP query
Custom UDP Protocol	UDP	Remote host IP address	Inbound multicast traffic

For IGMP multicast traffic, you must have the following outbound rules at a minimum.

Type	Protocol	Destination	Description
Custom Protocol	2	224.0.0.2/32	IGMP leave
Custom Protocol	2	Multicast group IP address	IGMP join
Custom UDP Protocol	UDP	Multicast group IP address	Outbound multicast traffic

Multicast domains in AWS Transit Gateway

A multicast domain allows segmentation of a multicast network into different domains. To begin using multicast with a transit gateway, create a multicast domain, and then associate subnets with the domain.

Multicast domain attributes

The following table details the multicast domain attributes. You cannot enable both attributes at the same time.

Attribute	Description
IgmPv2Support (AWS CLI) IGMPv2 support (console)	This attribute determines how group members join or leave a multicast group. When this attribute is disabled, you must add the group members to the domain manually. Enable this attribute if at least one member uses the IGMP protocol. Members join the multicast group in one of the following ways: - Members that support IGMP use the JOIN and LEAVE messages. - Members that do not support IGMP must be added or removed from the group using the Amazon VPC console or the AWS CLI. If you register multicast group members, you must deregister them, too. The transit gateway ignores an IGMP LEAVE message sent by a manually added group member.
StaticSourcesSupport (AWS CLI) Static sources support (console)	This attribute determines whether there are static multicast sources for the group. When this attribute is enabled, you must add sources for a multicast domain using register-transit-gateway-multicast-group-sources. Only multicast sources can send multicast traffic. When this attribute is disabled, there are no designated multicast sources. Any instances that are in subnets associate

Attribute	Description
	d with the multicast domain can send multicast traffic, and the group members receive the multicast traffic.

Create an IGMP multicast domain in AWS Transit Gateway

If you have not already done so, review the available multicast domain attributes. For more information, see [the section called “Multicast domains”](#).

To create an IGMP multicast domain using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Multicast**.
3. Choose **Create transit gateway multicast domain**.
4. For **Name tag**, enter a name for the domain.
5. For **Transit gateway ID**, choose the transit gateway that processes the multicast traffic.
6. For **IGMPv2 support**, select the checkbox.
7. For **Static sources support**, clear the checkbox.
8. To automatically accept cross-account subnet associations for this multicast domain, select **Auto accept shared associations**.
9. Choose **Create transit gateway multicast domain**.

To create an IGMP multicast domain using the AWS CLI

Use the [create-transit-gateway-multicast-domain](#) command.

```
aws ec2 create-transit-gateway-multicast-domain --transit-gateway-id tgw-0xexampleid12345 --options StaticSourcesSupport=disable,Igmpv2Support=enable
```

Create a static source multicast domain in AWS Transit Gateway

If you have not already done so, review the available multicast domain attributes. For more information, see [the section called “Multicast domains”](#).

To create a static multicast domain using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Multicast**.
3. Choose **Create transit gateway multicast domain**.
4. For **Name tag**, enter a name to identify the domain.
5. For **Transit gateway ID**, choose the transit gateway that processes the multicast traffic.
6. For **IGMPv2 support**, clear the checkbox.
7. For **Static sources support**, select the checkbox.
8. To automatically accept cross-account subnet associations for this multicast domain, select **Auto accept shared associations**.
9. Choose **Create transit gateway multicast domain**.

To create a static multicast domain using the AWS CLI

Use the [create-transit-gateway-multicast-domain](#) command.

```
aws ec2 create-transit-gateway-multicast-domain --transit-gateway-id tgw-0xexampleid12345 --options StaticSourcesSupport=enable,Igmpv2Support=disable
```

Associating VPC attachments and subnets with a multicast domain in AWS Transit Gateway

Use the following procedure to associate a VPC attachment with a multicast domain. When you create an association, you can then select the subnets to include in the multicast domain.

Before you begin, you must create a VPC attachment on your transit gateway. For more information, see [Amazon VPC attachments in AWS Transit Gateway](#).

To associate VPC attachments with a multicast domain using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Multicast**.
3. Select the multicast domain, and then choose **Actions, Create association**.
4. For **Choose attachment to associate**, select the transit gateway attachment.

5. For **Choose subnets to associate**, select the subnets to include in the multicast domain.
6. Choose **Create association**.

To associate VPC attachments with a multicast domain using the AWS CLI

Use the [associate-transit-gateway-multicast-domain](#) command.

Disassociate a subnet from a multicast domain in AWS Transit Gateway

Use the following procedure to disassociate subnets from a multicast domain.

To disassociate subnets using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Multicast**.
3. Select the multicast domain.
4. Choose the **Associations** tab.
5. Select the subnet, and then choose **Actions, Delete association**.

To disassociate subnets using the AWS CLI

Use the [disassociate-transit-gateway-multicast-domain](#) command.

View multicast domain associations in AWS Transit Gateway

View your multicast domains to verify that they are available, and that they contain the appropriate subnets and attachments.

To view a multicast domain using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Multicast**.
3. Select the multicast domain.
4. Choose the **Associations** tab.

To view a multicast domain using the AWS CLI

Use the [describe-transit-gateway-multicast-domains](#) command.

Add tags to a multicast domain in AWS Transit Gateway

Add tags to your resources to help organize and identify them, such as by purpose, owner, or environment. You can add multiple tags to each multicast domain. Tag keys must be unique for each multicast domain. If you add a tag with a key that is already associated with the multicast domain, it updates the value of that tag. For more information, see [Tagging your Amazon EC2 Resources](#).

To add tags to a multicast domain using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Multicast**.
3. Select the multicast domain.
4. Choose **Actions, Manage tags**.
5. For each tag, choose **Add new tag** and enter a **Key** and **Value** for the tag.
6. Choose **Save**.

To add tags to a multicast domain using the AWS CLI

Use the [create-tags](#) command.

Delete a multicast domain in AWS Transit Gateway

Use the following procedure to delete a multicast domain.

To delete a multicast domain using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Multicast**.
3. Select the multicast domain, and then choose **Actions, Delete multicast domain**.
4. When prompted for confirmation, enter **delete** and then choose **Delete**.

To delete a multicast domain using the AWS CLI

Use the [delete-transit-gateway-multicast-domain](#) command.

Shared multicast domains in AWS Transit Gateway

With multicast domain sharing, multicast domain owners can share the domain with other AWS accounts inside its organization or across organizations in AWS Organizations. As the multicast domain owner, you can create and manage the multicast domain centrally. Once shared, those users can perform the following operations on a shared multicast domain:

- Register and deregister group members or group sources in the multicast domain
- Associate a subnet with the multicast domain, and disassociate subnets from the multicast domain

A multicast domain owner can share a multicast domain with:

- AWS accounts inside its organization or across organizations in AWS Organizations
- An organizational unit inside its organization in AWS Organizations
- Its entire organization in AWS Organizations
- AWS accounts outside of AWS Organizations.

To share a multicast domain with an AWS account outside of your Organization, you must create a resource share using AWS Resource Access Manager, and then choose **Allow sharing with anyone** when selecting the Principals to share the multicast domain with. For more information on creating a resource share, see [Creating a resource share in AWS RAM](#) in the *AWS RAM User Guide*

Contents

- [Prerequisites for sharing a multicast domain](#)
- [Related services](#)
- [Shared multicast domain permissions](#)
- [Billing and metering](#)
- [Quotas](#)
- [Share resources across Availability Zones in AWS Transit Gateway](#)
- [Share a multicast domain in AWS Transit Gateway](#)
- [Unshare a shared multicast domain in AWS Transit Gateway](#)
- [Identify a shared multicast domain in AWS Transit Gateway](#)

Prerequisites for sharing a multicast domain

- To share a multicast domain, you must own it in your AWS account. You cannot share a multicast domain that has been shared with you.
- To share a multicast domain with your organization or an organizational unit in AWS Organizations, you must enable sharing with AWS Organizations. For more information, see [Enable Sharing with AWS Organizations](#) in the *AWS RAM User Guide*.

Related services

Multicast domain sharing integrates with AWS Resource Access Manager (AWS RAM). AWS RAM is a service that enables you to share your AWS resources with any AWS account or through AWS Organizations. With AWS RAM, you share resources that you own by creating a *resource share*. A resource share specifies the resources to share, and the users with whom to share them. Consumers can be individual AWS accounts, or organizational units or an entire organization in AWS Organizations.

For more information about AWS RAM, see the [AWS RAM User Guide](#).

Shared multicast domain permissions

Permissions for owners

Owners are responsible for managing the multicast domain and the members and attachments that they register or associate with the domain. Owners can change or revoke shared access at any time. They can use AWS Organizations to view, modify, and delete resources that consumers create on shared multicast domains.

Permissions for consumers

Users of the shared multicast domain can perform the following operations on shared multicast domains in the same way that they would on multicast domains that they created:

- Register and deregister group members or group sources in the multicast domain
- Associate a subnet with the multicast domain, and disassociate subnets from the multicast domain

Consumers are responsible for managing the resources that they create on the shared multicast domain.

Customers cannot view or modify resources owned by other consumers or by the multicast domain owner, and they cannot modify multicast domains that are shared with them.

Billing and metering

There are no additional charges for sharing multicast domains for either the owner, or consumers.

Quotas

A shared multicast domain counts toward the owner's and shared user's multicast domain quotas.

Share resources across Availability Zones in AWS Transit Gateway

To ensure that resources are distributed across the Availability Zones for a Region, AWS Transit Gateway independently maps Availability Zones to names for each account. This could lead to Availability Zone naming differences across accounts. For example, the Availability Zone `us-east-1a` for your AWS account might not have the same location as `us-east-1a` for another AWS account.

To identify the location of your multicast domain relative to your accounts, you must use the *Availability Zone ID* (AZ ID). The AZ ID is a unique and consistent identifier for an Availability Zone across all AWS accounts. For example, `use1-az1` is an AZ ID for the `us-east-1` Region and it is the same location in every AWS account.

To view the AZ IDs for the Availability Zones in your account

1. Open the AWS RAM console at <https://console.aws.amazon.com/ram/home>.
2. The AZ IDs for the current Region are displayed in the **Your AZ ID** panel on the right-hand side of the screen.

Share a multicast domain in AWS Transit Gateway

When an owner shares a multicast domain with you, you can do the following:

- Register and deregister group members or group sources
- Associate and disassociate subnets

Note

To share a multicast domain, you must add it to a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. A resource share specifies the resources to share, and the consumers with whom they are shared. When you share a multicast domain using the Amazon Virtual Private Cloud Console, you add it to an existing resource share. To add the multicast domain to a new resource share, you must first create the resource share using the [AWS RAM console](#).

If you are part of an organization in AWS Organizations and sharing within your organization is enabled, consumers in your organization are automatically granted access to the shared multicast domain. Otherwise, consumers receive an invitation to join the resource share and are granted access to the shared multicast domain after accepting the invitation.

You can share a multicast domain that you own using the Amazon Virtual Private Cloud console, AWS RAM console, or the AWS CLI.

To share a multicast domain that you own using the *Amazon Virtual Private Cloud Console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Multicast Domains**.
3. Select your multicast domain, and then choose **Actions, Share multicast domain**.
4. Select your resource share and choose **Share multicast domain**.

To share a multicast domain that you own using the AWS RAM console

See [Creating a Resource Share](#) in the *AWS RAM User Guide*.

To share a multicast domain that you own using the AWS CLI

Use the [create-resource-share](#) command.

Unshare a shared multicast domain in AWS Transit Gateway

When a shared multicast domain is unshared, the following happens to consumer multicast domain resources:

- Consumer subnets are disassociated from the multicast domain. The subnets remain in the consumer account.
- Consumer group sources and group members are disassociated from the multicast domain, and then deleted from the consumer account.

To unshare a multicast domain, you must remove it from the resource share. You can do this from the AWS RAM console or the AWS CLI.

To unshare a shared multicast domain that you own, you must remove it from the resource share. You can do this using the Amazon Virtual Private Cloud, AWS RAM console, or the AWS CLI.

To unshare a shared multicast domain that you own using the *Amazon Virtual Private Cloud Console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Multicast Domains**.
3. Select your multicast domain, and then choose **Actions, Stop sharing**.

To unshare a shared multicast domain that you own using the AWS RAM console

See [Updating a Resource Share](#) in the *AWS RAM User Guide*.

To unshare a shared multicast domain that you own using the AWS CLI

Use the [disassociate-resource-share](#) command.

Identify a shared multicast domain in AWS Transit Gateway

Owners and consumers can identify shared multicast domains using the Amazon Virtual Private Cloud and AWS CLI

To identify a shared multicast domain using the *Amazon Virtual Private Cloud Console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Multicast Domains**.
3. Select your multicast domain.
4. On the **Transit Multicast Domain Details** page, view the **Owner ID** to identify the AWS account ID of the multicast domain.

To identify a shared multicast domain using the AWS CLI

Use the [describe-transit-gateway-multicast-domains](#) command. The command returns the multicast domains that you own and multicast domains that are shared with you. `OwnerId` shows the AWS account ID of the multicast domain owner.

Register sources with a multicast group in AWS Transit Gateway

Note

This procedure is only required when you have set the **Static sources support** attribute to **enable**.

Use the following procedure to register sources with a multicast group. The source is the network interface that sends multicast traffic.

You need the following information before you add a source:

- The ID of the multicast domain
- The IDs of the sources' network interfaces
- The multicast group IP address

To register sources using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Multicast**.
3. Select the multicast domain, and then choose **Actions, Add group sources**.
4. For **Group IP address**, enter either the IPv4 CIDR block or IPv6 CIDR block to assign to the multicast domain.
5. Under **Choose network interfaces**, select the multicast senders' network interfaces.
6. Choose **Add sources**.

To register sources using the AWS CLI

Use the [register-transit-gateway-multicast-group-sources](#) command.

Register members with a multicast group in AWS Transit Gateway

Use the following procedure to register group members with a multicast group.

You need the following information before you add members:

- The ID of the multicast domain
- The IDs of the group members' network interfaces
- The multicast group IP address

To register members using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Multicast**.
3. Select the multicast domain, and then choose **Actions, Add group members**.
4. For **Group IP address**, enter either the IPv4 CIDR block or IPv6 CIDR block to assign to the multicast domain.
5. Under **Choose network interfaces**, select the multicast receivers' network interfaces.
6. Choose **Add members**.

To register members using the AWS CLI

Use the [register-transit-gateway-multicast-group-members](#) command.

Deregister sources from a multicast group in AWS Transit Gateway

You don't need to follow this procedure unless you manually added a source to the multicast group.

To remove a source using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Multicast**.
3. Select the multicast domain.
4. Choose the **Groups** tab.
5. Select the sources, and then choose **Remove source**.

To remove a source using the AWS CLI

Use the [deregister-transit-gateway-multicast-group-sources](#) command.

Deregister members from a multicast group in AWS Transit Gateway

You don't need to follow this procedure unless you manually added a member to the multicast group.

To deregister members using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Multicast**.
3. Select the multicast domain.
4. Choose the **Groups** tab.
5. Select the members, and then choose **Remove member**.

To deregister members using the AWS CLI

Use the [deregister-transit-gateway-multicast-group-members](#) command.

View multicast groups in AWS Transit Gateway

You can view information about your multicast groups to verify that members were discovered using the IGMPv2 protocol. **Member type** (in the console), or `MemberType` (in the AWS CLI) displays IGMP when AWS discovered members with the protocol.

To view multicast groups using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. On the navigation pane, choose **Transit Gateway Multicast**.
3. Select the multicast domain.
4. Choose the **Groups** tab.

To view multicast groups using the AWS CLI

Use the [search-transit-gateway-multicast-groups](#) command.

The following example shows that the IGMP protocol discovered multicast group members.

```
aws ec2 search-transit-gateway-multicast-groups --transit-gateway-multicast-domain tgw-mcast-domain-000fb24d04EXAMPLE
{
  "MulticastGroups": [
    {
      "GroupIpAddress": "224.0.1.0",
      "TransitGatewayAttachmentId": "tgw-attach-0372e72386EXAMPLE",
      "SubnetId": "subnet-0187aff814EXAMPLE",
      "ResourceId": "vpc-0065accde4EXAMPLE",
      "ResourceType": "vpc",
      "NetworkInterfaceId": "eni-03847706f6EXAMPLE",
      "MemberType": "igmp"
    }
  ]
}
```

Set up multicast for Windows Server in AWS Transit Gateway

You'll need to perform additional steps when setting up multicast to work with transit gateways on Windows Server 2019 or 2022. To set this up you'll need to use PowerShell, and run the following commands:

To set up multicast for Windows Server using PowerShell

1. Change Windows Server to use IGMPv2 instead of IGMPv3 for the TCP/IP stack:

```
PS C:\> New-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters -Name IGMPVersion -PropertyType DWord -Value 3
```

Note

New-ItemProperty is a property index that specifies the IGMP version. Because IGMP v2 is the supported version for multicast, the property Value must be 3. Instead of editing the Windows registry you can run the following command to set the IGMP version to 2.:

```
Set-NetIPv4Protocol -IGMPVersion Version2
```

2. Windows Firewall drops most UDP traffic by default. You'll first need to check which connection profile is being used for multicast:

```
PS C:\> Get-NetConnectionProfile | Select-Object NetworkCategory

NetworkCategory
-----
Public
```

3. Update the connection profile from the previous step to allow access to the required UDP port(s):

```
PS C:\> Set-NetFirewallProfile -Profile Public -Enabled False
```

4. Reboot the EC2 instance.
5. Test your multicast application to ensure traffic is flowing as expected.

Example: Manage IGMP configurations using AWS Transit Gateway

This example shows at least one host that uses the IGMP protocol for multicast traffic. AWS automatically creates the multicast group when it receives an IGMP JOIN message from an instance, and then adds the instance as a member in this group. You can also statically add non-IGMP hosts as members to a group using the AWS CLI. Any instances that are in subnets associated with the multicast domain can send traffic, and the group members receive the multicast traffic.

Use the following steps to complete the configuration:

1. Create a VPC. For more information, see [Create a VPC](#) in the *Amazon VPC User Guide*.
2. Create a subnet in the VPC. For more information, see [Create a subnet](#) in the *Amazon VPC User Guide*.
3. Create a transit gateway configured for multicast traffic. For more information, see [the section called "Create a transit gateway"](#).
4. Create a VPC attachment. For more information, see [the section called "Create a VPC attachment"](#).
5. Create a multicast domain configured for IGMP support. For more information, see [the section called "Create an IGMP multicast domain"](#).

Use the following settings:

- Enable **IGMPv2 support**.
- Disable **Static sources support**.

6. Create an association between subnets in the transit gateway VPC attachment and the multicast domain. For more information see [the section called “Associating VPC attachments and subnets with a multicast domain”](#).
7. The default IGMP version for EC2 is IGMPv3. You need to change the version for all IGMP group members. You can run the following command:

```
sudo sysctl net.ipv4.conf.eth0.force_igmp_version=2
```

8. Add the members that do not use the IGMP protocol to the multicast group. For more information, see [the section called “Register members with a multicast group”](#).

Example: Manage static source configurations in AWS Transit Gateway

This example statically adds multicast sources to a group. Hosts do not use the IGMP protocol to join or leave multicast groups. You need to statically add the group members that receive the multicast traffic.

Use the following steps to complete the configuration:

1. Create a VPC. For more information, see [Create a VPC](#) in the *Amazon VPC User Guide*.
2. Create a subnet in the VPC. For more information, see [Create a subnet](#) in the *Amazon VPC User Guide*.
3. Create a transit gateway configured for multicast traffic. For more information, see [the section called “Create a transit gateway”](#).
4. Create a VPC attachment. For more information, see [the section called “Create a VPC attachment”](#).
5. Create a multicast domain configured for no IGMP support, and support for statically adding sources. For more information, see [the section called “Create a static source multicast domain”](#).

Use the following settings:

- Disable **IGMPv2 support**.
- To manually add sources, enable **Static sources support**.

The sources are the only resources that can send multicast traffic when the attribute is enabled. Otherwise, any instances that are in subnets associated with the multicast domain can send multicast traffic, and the group members receive the multicast traffic.

6. Create an association between subnets in the transit gateway VPC attachment and the multicast domain. For more information see [the section called "Associating VPC attachments and subnets with a multicast domain"](#).
7. If you enable **Static sources support**, add the source to the multicast group. For more information, see [the section called "Register sources with a multicast group"](#).
8. Add the members to the multicast group. For more information, see [the section called "Register members with a multicast group"](#).

Example: Manage static group member configurations in AWS Transit Gateway

This example shows statically adding multicast members to a group. Hosts cannot use the IGMP protocol to join or leave multicast groups. Any instances that are in subnets associated with the multicast domain can send multicast traffic, and the group members receive the multicast traffic.

Use the following steps to complete the configuration:

1. Create a VPC. For more information, see [Create a VPC](#) in the *Amazon VPC User Guide*.
2. Create a subnet in the VPC. For more information, see [Create a subnet](#) in the *Amazon VPC User Guide*.
3. Create a transit gateway configured for multicast traffic. For more information, see [the section called "Create a transit gateway"](#).
4. Create a VPC attachment. For more information, see [the section called "Create a VPC attachment"](#).
5. Create a multicast domain configured for no IGMP support, and support for statically adding sources. For more information, see [the section called "Create a static source multicast domain"](#).

Use the following settings:

- Disable **IGMPv2 support**.
 - Disable **Static sources support**.
6. Create an association between subnets in the transit gateway VPC attachment and the multicast domain. For more information see [the section called "Associating VPC attachments and subnets with a multicast domain"](#).

7. Add the members to the multicast group. For more information, see [the section called "Register members with a multicast group"](#).

Flexible cost allocation

By default, transit gateway uses a sender-based cost allocation model where data processing charges are allocated to the account that owns the source attachment. You can create custom metering policies that define which accounts should be charged based on traffic flow properties such as attachment types, specific attachment IDs, or network addresses.

Metering policies consist of ordered rules that are evaluated from lowest to highest rule number. When traffic matches a rule, the specified account is charged according to the rule's configuration. You can specify the account owner for allocating costs from the following options:

- **Source attachment owner** - Charges are allocated to the account that owns the source attachment (default behavior)
- **Destination attachment owner** - Charges are allocated to the account that owns the destination attachment
- **Transit Gateway owner** - Charges are allocated to the account that owns the transit gateway

Flexible Cost Allocation enables better cost management for organizations using centralized network architectures, allowing costs to be allocated to the appropriate business units or application owners regardless of network topology.

Note

Flexible Cost Allocation enables flexible allocation of metering usage and in turn costs to account owners of your choice. However, tax implications for AWS accounts can vary significantly based on geographic location, usage patterns and other factors. Please review the billing, tax and cost management implications for accounts in your AWS Organization prior to enabling this feature. Reference: [What is AWS Billing and Cost Management?](#)

Metering policies

Metering policies allow you to configure cost allocation rules for your transit gateway to control which accounts are charged for data processing and transfer costs based on traffic flow properties.

This feature enables better cost management and chargeback capabilities for organizations using centralized network architectures.

A metering policy is composed of the following:

- **Metering policy** - The overall configuration container that contains the Metering Policy Rules. When created, it contains a single default metering policy entry that is configured to charge all traffic to the source attachment owner. Each transit gateway can have only one metering policy.
- **Metering policy entry** - Individual rules within a metering policy that define specific matching criteria and the account to meter usage. Each entry includes a rule number for evaluation order, traffic matching conditions (such as source and destination attachment types, attachment IDs, CIDR blocks, ports, and protocols), and which account owner to charge for matching traffic. A policy can contain up to 50 entries, evaluated in order from lowest to highest rule number.

You can allocate metering usage to any of the following:

- **Source attachment owner**: Allocates metering usage to the account that owns the attachment where traffic originates (default behavior)
- **Destination attachment owner**: Allocates metering usage to the account that owns the attachment where traffic terminates, and
- **transit gateway owner**: Allocates metering usage to the account that owns the transit gateway.
- **Middlebox attachments** - (Optional) Designated transit gateway attachments that route traffic through network appliances for security inspection, load balancing, or other network functions. Data usage for the traffic traversing middlebox attachments is metered to the account owner specified in the metering policy. You can specify a maximum of 10 middlebox attachments. Supported middlebox attachment types are Network Function (AWS Network Firewall), VPC and VPN attachments.

How metering policies work

By default, transit gateway uses a sender-based cost allocation model where data processing charges are metered to the account that owns the source attachment. With metering policies, you can create custom rules to flexibly meter usage based on the following traffic flow properties:

- Source and destination attachment types (VPC, VPN, Client VPN, Direct Connect Gateway, Peering, Network Function, and VPN Concentrator)
- Source and destination attachment IDs

- Source and destination IP addresses, Port ranges and protocols

Metering policies consist of ordered rules that are evaluated from lowest to highest rule number. When traffic matches a rule, the specified account is charged according to the rule's metered account setting. Metering policies address several common organizational scenarios:

- **Hybrid environment cost allocation:** Allocate costs for data entering AWS from on-premises through Direct Connect Gateway to the destination VPC account owner rather than the central IT admin account owner.
- **Centralized inspection architecture:** Allocate costs to individual application or VPC account owners rather than the central security team for traffic traversing via inspection VPCs.
- **Application-based chargeback:** Allocate all data usage costs for a workload to the VPC owner regardless of traffic direction.
- **Client cost allocation:** Allocate data costs to client accounts when they create attachments to your transit gateway.

Middlebox attachments

Transit gateway metering policies support Middlebox attachments allowing you to flexibly allocate data processing charges for network traffic routed via middlebox appliances such as network firewalls and load balancers. Examples of middlebox attachments are Network Function attachment to AWS Network Firewall or VPC attachments that route traffic to third-party security appliances in a VPC. Traffic between source and destination transit gateway attachments traverses via these middlebox attachments for typical security inspection use-cases. You can define metering policies to flexibly allocated data processing usage on middlebox attachments to the original source attachment, final destination attachment or transit gateway account owner. For Network Function attachments, the AWS Network Firewall data processing charges are also allocated to the metered account.

Flexible Cost Allocation - Metering usage types

Flexible cost allocation via metering policies applies to following data usage types:

- Transit gateway Data Processing Usage on VPC, VPN, Client VPN, VPN Concentrator, and Direct Connect attachments
- Client VPN Data Transfer Out usage on Client VPN attachments

- Site-to-site VPN Data Transfer Out usage on VPN attachments
- Direct Connect Data Transfer Out usage on Direct Connect attachments.
- Data transfer usage on TGW peering attachments
- Transit gateway Data processing usage on Network Function attachments
- AWS network firewall (NFW) data processing usage on Network Function attachments.

Flexible cost allocation does not apply to attachments hourly usage and multicast data processing usage. For Transit Gateway Connect attachments, metering policy can be defined for the underlying transport VPC or Direct Connect attachment. For Private IP VPN attachments, metering policy can be defined for the underlying transport Direct Connect attachment.

Considerations and limitations

Consider the following when implementing metering policies for your transit gateway.

Permissions

- Only the transit gateway owner can create, modify, or delete metering policies.
- Cost allocation settings apply at the transit gateway level.
- Attachment owners cannot override cost allocation settings configured by the transit gateway owner.

Transit Gateway peering

When traffic traverses transit gateway peering connections:

- Each transit gateway applies its own metering policy independently.
- Data charges are allocated separately by each transit gateway based on its local policy.
- Traffic can be thought of as two separate flows: source attachment to peering, and peering to destination attachment.

Cloud WAN integration

When a transit gateway is attached to a Cloud WAN core network:

- Transit gateway data transfer charges on peering connections are allocated according to the transit gateway metering policy.

- Metering policies are not supported on Cloud WAN core networks.

Performance impact

- Metering policies do not introduce any additional data-path latency.
- Metering policies have no impact on maximum bandwidth per attachment.
- There are no changes to transit gateway resource sharing capabilities.

Billing integration

- Cost allocation tags continue to work with metering policies for organizing costs by business unit.
- Metering policies define which accounts incur costs, while cost allocation tags help categorize those costs.
- Changes to metering policies take effect at the end of the next billing hour.

IPv6 support

Metering policies are supported for both IPv4 and IPv6 traffic. CIDR block matching in policy entries works with both address families.

Middlebox attachment support

- Middlebox metering policy assumes traffic between the original source and destination attachment is hair-pinned via the specified middle-box attachment (example east-west inspection for VPC-to-VPC traffic). Hence the network 5-tuple (source/destination IPs, source/destination ports and protocol) for flows ingressing and egressing out of middle-box attachments must match. Flows with 5-tuple mis-matches on middle-box attachments (e.g. NAT transformation in inspection VPC) are treated as regular source-destination attachment flows (as opposed to middle-box attachment flows).
- All egress-only flows on the middlebox attachment (for example north-south traffic to internet via IGW in an inspection VPC) are treated as regular source-destination flows (as opposed to middle-box attachment flows).
- For Network Function attachments when AWS Network firewall drops packets, all data processing usage is charged back to the sender account regardless of metering policy configuration.

Create an AWS Transit Gateway metering policy

To enable metering policies, you must create a metering policy for your transit gateway and configure policy entries that define how metering usage is allocated. The metering policy establishes the framework and default settings, while policy entries contain the specific rules that determine which accounts are metered based on traffic characteristics.

Metering policy entries function as ordered rules that are applied sequentially from lowest to highest rule number for traffic flowing through your transit gateway. Each entry defines matching criteria such as source and destination attachment types, CIDR blocks, protocols, and port ranges, along with the account that should be metered for matching traffic. When a traffic flow matches multiple entries, the entry with the lowest rule number takes precedence. If no entries match a particular flow, the default metered account specified in the policy is charged.

After creating a policy, you'll need to add policy entries to implement your cost allocation logic. For the steps to create a metering policy entry, see [Create a metering policy entry](#).

Create a metering policy using the console

Create a policy to define flexible cost allocation rules for transit gateway data usage. By default, all flows are metered to the source attachment owner. Create entries to bill specific network flows to different accounts.

To create a metering policy

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Metering policies**.
3. Choose **Create metering policy**.
4. For **Transit gateway ID** choose the transit gateway you'd like to create metering policy for.
5. (Optional) For **Middlebox attachment IDs**, choose one or more middlebox attachment. By default, data usage is metered to the middlebox owner. Middlebox attachment support enables metering policy to be applied for traffic traversing middlebox attachments. Additional attachments can be added later.
6. (Optional) In the **Tags** section, add tags to help you identify and organize your metering policy:
 - a. Choose **Add new tag**.
 - b. Enter a tag **Key** and optionally a tag **Value**.

- c. Choose **Add new tag** to add additional tags, or skip to the next step. You can add up to 50 tags.
7. Choose **Create transit gateway metering policy**.

Note

The default metered account is the source attachment owner, and after creating a metering policy, you can add entries that define which account gets charged based on traffic flow properties, noting that the default policy entry (which is the last entry) cannot be modified or deleted like other policy entries.

Create a metering policy using the AWS CLI

A metering policy defines the default cost allocation behavior and global settings for your transit gateway. Use the [create-transit-gateway-metering-policy](#).

Required parameters:

- `--transit-gateway-id` - The ID of the transit gateway to create the policy for

Optional parameters:

- `--middle-box-attachment-ids` - Supported transit gateway attachment IDs to add to the policy as middlebox
- `--tag-specifications` - tags for metering policy

To create a metering policy using the AWS CLI

1. Run the **create-transit-gateway-metering-policy** command to create a new metering policy with optional middlebox attachments.

```
aws ec2 create-transit-gateway-metering-policy \  
  --transit-gateway-id tgw-07a5946195a67dc47 \  
  --middle-box-attachment-ids \  
    tgw-attach-0123456789abcdef0 \  
    tgw-attach-0abc123def456789a \  
  --tag-specifications tag:Name=tag1,value=value1,tag:Name=tag2,value=value2
```

```
--tag-specifications \  
'[{"ResourceType": "transit-gateway-metering-policy", \  
  "Tags": [ {"Key": "Env", "Value": "Prod" } ] }]'
```

This command creates a metering policy for the specified transit gateway with provided middlebox attachments and tags.

2. The command returns the following output when the policy is successfully created:

```
{  
  "TransitGatewayMeteringPolicy": {  
    "TransitGatewayMeteringPolicyId": "tgw-mp-042d444564d4b2da7",  
    "TransitGatewayId": "tgw-07a5946195a67dc47",  
    "MiddleboxAttachmentIds": ["tgw-attach-0123456789abcdef0",  
    "tgw-attach-0abc123def456789a"],  
    "State": "pending",  
    "UpdateEffectiveAt": "2025-11-05T21:00:00.000Z",  
    "Tags": [{"Key": "Env", "Value": "Prod"}]  
  }  
}
```

Note the metering policy ID returned in the response for use in subsequent commands.

describe-transit-gateway-metering-policies command can be used to get metering policy associated with transit gateway.

Manage AWS Transit Gateway metering policies

After creating a metering policy, you can manage it by viewing current settings, modifying configuration options, or deleting the policy when no longer needed. Management operations allow you to add or remove middlebox attachments as your network requirements change. You can only create or delete a policy entry. If you need to modify an existing rule, you can delete the entry and create a new one with the modified configuration. All management operations require transit gateway owner permissions and take effect after two billing hour.

Effective metering policy management is crucial for maintaining accurate cost allocation as your network architecture evolves. Organizations often need to adjust their policies when business units change, new applications are deployed, or network topologies are modified. For example middlebox metering support settings may require updates when firewall security architectures change or when new inspection services are introduced into the traffic path.

Policy modifications support various operational scenarios including seasonal traffic pattern changes, merger and acquisition activities, and compliance requirement updates. When managing policies, consider the impact on existing billing arrangements and communicate changes to affected stakeholders before implementation.

Regular policy reviews help ensure that cost allocation remains aligned with business objectives and organizational structures. Best practices include documenting policy changes, testing modifications in non-production environments when possible, and coordinating with finance teams to understand billing implications. Additionally, consider the timing of policy changes to minimize disruption to monthly billing cycles and financial reporting processes.

Topics

- [Edit an AWS Transit Gateway metering policy](#)
- [Delete an AWS Transit Gateway metering policy](#)

Edit an AWS Transit Gateway metering policy

Edit existing metering policies to modify middlebox attachment configurations. Policy modifications take effect at the next billing hour and apply to all future traffic flows through your transit gateway.

Edit a metering policy using the console

Use the console to modify existing metering policy settings for your transit gateway.

To edit an existing metering policy using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Metering policies**.
3. Select the metering policy you want to modify by choosing its policy ID
4. Modify the available policy settings under **Actions**. Console only allow add and remove of Middle box attachments.
 - **Middlebox attachments** - Add or remove transit gateway attachments that should be treated as middleboxes for specialized billing.

Edit a metering policy using the AWS CLI

Use the **modify-transit-gateway-metering-policy** command to view and modify metering policies.

Required parameters for modify operations:

- **--transit-gateway-metering-policy-id** - The ID of the metering policy to modify
- **--add-middle-box-attachment-ids** or **--remove-middle-box-attachment-ids** - Supported transit gateway attachment IDs to add or remove from the policy as middlebox

To view and edit metering policies using the AWS CLI

1. (Optional) View existing metering policies using the **describe-transit-gateway-metering-policies** command to see current configuration settings:

```
aws ec2 describe-transit-gateway-metering-policies
```

This command returns all metering policies in your account, showing their current state, and attachments enabled as middlebox for each of the metering policy.

2. Modify a metering policy using the **modify-transit-gateway-metering-policy** command to update configuration options:

```
aws ec2 modify-transit-gateway-metering-policy \
  --transit-gateway-metering-policy-id tgw-mp-042d444564d4b2da7 \
  --add-middle-box-attachment-ids tgw-attach-0123456789abcdef1 \
  --remove-middle-box-attachment-ids tgw-attach-0abc123def456789a
```

This command modifies a metering policy by adding and/or removing middlebox attachments.

3. The command returns the following output when the policy is successfully modified:

```
{
  "TransitGatewayMeteringPolicy": {
    "TransitGatewayMeteringPolicyId": "tgw-mp-042d444564d4b2da7",
    "TransitGatewayId": "tgw-07a5946195a67dc47",
    "MiddleboxAttachmentIds": ["tgw-attach-0123456789abcdef0",
    "tgw-attach-0123456789abcdef1"],
    "State": "modifying",
    "UpdateEffectiveAt": "2025-11-05T21:00:00.000Z"
  }
}
```

```
}
```

The changes can take up to two billing hours to take effect.

Delete an AWS Transit Gateway metering policy

Delete metering policies when they are no longer required for your transit gateway cost allocation strategy. Deleting a policy reverts cost allocation to the default sender-based model where data processing and data transfer charges are allocated to the account that owns the source attachment. All policy entries associated with the deleted metering policy are also removed.

Delete a metering policy using the console

Use the console to remove metering policies that are no longer needed.

To delete a metering policy using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Metering policies**.
3. Select the policy you want to delete by choosing its policy ID.
4. Choose **Actions**, and then **Delete**.
5. Confirm the deletion by typing **delete** in the confirmation dialog.
6. Choose **Delete**.

Important

Deleting a metering policy is irreversible. All policy entries and configuration settings will be permanently removed, and cost allocation will revert to the default sender-based model.

Delete a metering policy using the AWS CLI

Use the **delete-transit-gateway-metering-policy** command to delete metering policies programmatically.

Requirements:

- Transit gateway owner permissions

Required parameters:

- `--transit-gateway-metering-policy-id` - The ID of the metering policy to delete

To view and delete metering policies using the AWS CLI

1. (Optional) View existing metering policies using the **describe-transit-gateway-metering-policies** command to see current configuration settings:

```
aws ec2 describe-transit-gateway-metering-policies
```

This command returns all metering policies in your account, showing their current state and configuration.

2. Delete a metering policy using the **delete-transit-gateway-metering-policy** command to permanently remove the policy:

```
aws ec2 delete-transit-gateway-metering-policy \  
  --transit-gateway-metering-policy-id tgw-mp-042d444564d4b2da7
```

This command permanently removes the specified metering policy and all associated entries. Cost allocation will revert to the default sender-based model for all future traffic flows. This change also takes 2 billing hours to take effect.

3. The command returns the following output when the policy is successfully deleted:

```
{  
  "TransitGatewayMeteringPolicy": {  
    "TransitGatewayMeteringPolicyId": "tgw-mp-042d444564d4b2da7",  
    "TransitGatewayId": "tgw-07a5946195a67dc47",  
    "MiddleboxAttachmentIds": ["tgw-attach-0123456789abcdef0",  
    "tgw-attach-0123456789abcdef1"],  
    "State": "deleting",  
    "UpdateEffectiveAt": "2025-11-05T21:00:00.000Z"  
  }  
}
```

The response confirms the policy is being deleted with a deleting state while the removal is processed across the transit gateway infrastructure.

Create an AWS Transit Gateway metering policy entry

By default, all flows are metered to the source attachment owner. To meter specific flows to different accounts, create individual policy entries that define which account gets charged based on traffic flow properties.

Metering policy entries function as conditional rules that are evaluated in sequential order based on their rule numbers when traffic flows through your transit gateway. Each entry acts as an "if-then" statement: if the traffic matches the specified criteria (such as source attachment type, destination CIDR block, or protocol), then charge the designated account. The system evaluates entries from lowest to highest rule number, and the first matching entry determines the billing account for that traffic flow.

Entries support a wide range of matching criteria including attachment types (VPC, VPN, Client VPN, Direct Connect Gateway, Peering, Network Function, and VPN Concentrator), specific attachment IDs, source and destination CIDR blocks, protocol types, and port ranges. You can combine multiple criteria within a single entry to create precise targeting rules. For example, you might create an entry that matches all HTTPS traffic (port 443) from VPC attachments to a specific destination CIDR range and charges those flows to a security team's account. If no entries match a particular traffic flow, the default metered account specified in the parent metering policy is charged, ensuring all traffic is properly billed. Creating an entry takes 2 billing hours to take effect.

Important

- Plan rule numbers carefully - Leave gaps (e.g., 10, 20, 30) to allow for future insertions
- Test entries with less specific conditions first before adding more restrictive rules
- Use specific matching conditions to avoid unintended billing

Create a metering policy entry using the console

A metering policy defines the default cost allocation behavior and global settings for your transit gateway.

To create a metering policy entry using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Metering policies**.
3. Select the metering policy ID link to view its details.
4. Choose the **Metering policy entries** tab.
5. Choose **Create metering policy entry**.
6. **Policy rule number** - This should be a unique number (1- 32,766) that determines evaluation order. Lower numbers have higher priority.
7. **Metered account** - Choose one of the following account types to be charged for matching traffic flows:
 - a. **Source Attachment Owner**
 - b. **Destination Attachment Owner**
 - c. **Transit Gateway Attachment Owner**
8. (Optional) Choose **Rule conditions** - These optional conditions define criteria to match specific traffic:
 - **Source attachment type or ID** - Filter by attachment type (VPC, VPN, Client VPN, Direct Connect Gateway, Peering, Network Function, and VPN Concentrator) or ID.
 - **Destination attachment type or ID** - Filter by destination attachment type or ID
 - **Source CIDR block** - Match traffic from specific IP ranges
 - **Destination CIDR block** - Match traffic to specific IP ranges
 - **Source port range** - Match specific source ports
 - **Destination port range** - Match specific destination ports
 - **Protocol** - Filter by protocol for the rule (1, 6, 17, etc.)
9. Choose **Create metering policy entry** to save the configuration.

Create a metering policy entry using the AWS CLI

Policy entries define specific rules for cost allocation based on traffic characteristics. Rules are evaluated in order from lowest to highest rule number.

Required parameters:

- `--transit-gateway-metering-policy-id` - The ID of the metering policy to add the entry to
- `--policy-rule-number` - A unique number (1-32,766) that determines evaluation order
- `--metered-account` - payer type (source-attachment-owner/ destination-attachment-owner / transit-gateway-owner)

Optional parameters:

These optional parameters that define criteria to match specific traffic:

- `--source-transit-gateway-attachment-id` - The ID of the source transit gateway attachment.
- `--source-transit-gateway-attachment-type` - The type of the source transit gateway attachment.
- `--source-cidr-block` - The source CIDR block for the rule.
- `--source-port-range` - The source port range for the rule.
- `--destination-transit-gateway-attachment-id` - The ID of the destination transit gateway attachment.
- `--destination-transit-gateway-attachment-type` - The type of the destination transit gateway attachment.
- `--destination-cidr-block` - The destination CIDR block for the rule.
- `--destination-port-range` - The destination port range for the rule.
- `--protocol` - The protocol number for the rule

To create a metering policy entry using the AWS CLI

1. Use the **`create-transit-gateway-metering-policy-entry`** command to create a new policy entry that routes VPC traffic to a specific metered account:

```
aws ec2 create-transit-gateway-metering-policy-entry \
  --transit-gateway-metering-policy-id tgw-mp-042d444564d4b2da7 \
  --policy-rule-number 100 \
  --destination-transit-gateway-attachment-type vpc \
  --metered-account destination-attachment-owner
```

This command creates a policy entry with rule number 100 that matches traffic destined for VPC attachments and charges the destination attachment owner for those flows.

2. The command returns the following output when the entry is successfully created:

```
{
  "TransitGatewayMeteringPolicyEntry": {
    "MeteredAccount": "destination-attachment-owner",
    "MeteringPolicyRule": {
      "DestinationTransitGatewayAttachmentType": "vpc"
    },
    "PolicyRuleNumber": 100,
    "State": "available",
    "UpdateEffectiveAt": "2025-11-06T02:00:00.000Z"
  }
}
```

The response confirms the entry was created with a "available" state while it's being activated across the transit gateway infrastructure.

Delete an AWS Transit Gateway metering policy entry

Delete metering policy entries when specific cost allocation rules are no longer required for your network traffic flows. Entry deletion helps simplify policy management by removing outdated or unnecessary rules while maintaining the overall policy structure. When you delete an entry, traffic that previously matched the deleted rule will be evaluated against remaining entries in rule number order, or fall back to the default policy behavior if no other entries match.

Before deleting entries, consider the impact on current billing arrangements and traffic flows. Once deleted, the change takes up to 2 billing hours to get effective and cannot be undone, so coordinate changes with affected account owners and finance teams. Review remaining entries to ensure proper traffic coverage and billing allocation after the deletion. The rule evaluation order for remaining entries stays unchanged, maintaining predictable cost allocation behavior for continuing traffic flows.

Important

- Deletion is irreversible

- Traffic previously matching this entry will be re-evaluated against remaining entries
- Review remaining entries to ensure proper traffic coverage

Delete a metering policy entry using the console

Use the console to remove policy entries through an intuitive interface that provides confirmation dialogs to prevent accidental deletions.

To delete a policy entry using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Metering policies**.
3. Select the metering policy containing the entry you want to delete.
4. Select the entry you want to remove and choose **Delete**.
5. In the confirmation dialog, review the entry details and type **delete** to confirm the removal.
6. Choose **Delete** to permanently remove the entry.

Delete a metering policy entry using the AWS CLI

Use the **delete-transit-gateway-metering-policy-entry** command to remove policy entries programmatically.

Requirements:

- Transit gateway owner permissions
- Valid metering policy ID and entry rule number

Required parameters:

- **--transit-gateway-metering-policy-id** - The ID of the metering policy
- **--policy-rule-number** - The rule number of the entry to delete

To view and delete policy entries using the AWS CLI

1. (Optional) View existing policy entries using the **get-transit-gateway-metering-policy-entries** command to see current configuration settings:

```
aws ec2 get-transit-gateway-metering-policy-entries \
  --transit-gateway-metering-policy-id tgw-mp-0123456789abcdefg
```

This command returns all entries for the specified policy, showing their rule numbers, matching criteria, and metered accounts.

2. Delete a policy entry using the **delete-transit-gateway-metering-policy-entry** command to permanently remove the entry:

```
aws ec2 delete-transit-gateway-metering-policy-entry \
  --transit-gateway-metering-policy-id tgw-mp-0123456789abcdefg \
  --policy-rule-number 100
```

This command permanently removes the specified entry from the policy. Traffic that previously matched this entry will be immediately re-evaluated against remaining entries or fall back to the default policy behavior.

3. The command returns the following output when the entry is successfully deleted:

```
{
  "TransitGatewayMeteringPolicyEntry": [
    {
      "PolicyRuleNumber": 100,
      "MeteredAccount": "destination-attachment-owner",
      "UpdateEffectiveAt": "2024-01-01T01:00:00+00:00",
      "state": "deleted",
      "MeteringPolicyRule": {
        "DestinationTransitGatewayAttachmentType": "vpc"
      }
    }
  ]
}
```

The response confirms the entry is being deleted with a "deleted" state while the removal is processed across the transit gateway infrastructure.

Manage AWS Transit Gateway metering policy middlebox attachments

transit gateway metering policies support Middlebox attachments allowing you to flexibly allocate data processing charges for network traffic routed via middlebox appliances such as network firewalls and load balancers. Examples of middlebox attachments are Network Function attachment to AWS Network Firewall or VPC attachments that route traffic to third-party security appliances in a VPC. Traffic between source and destination transit gateway attachments traverses via these middlebox attachments for typical security inspection use-cases. You can define metering policies to flexibly allocated data processing usage on middlebox attachments to the original source attachment, final destination attachment or transit gateway account owner. For Network Function attachments, the AWS Network Firewall data processing charges are also allocated to the metered account.

Designated transit gateway attachments that route traffic through network appliances for security inspection, load balancing, or other network functions. Data usage for the traffic traversing middlebox attachments is metered to the account owner specified in the metering policy. You can specify a maximum of 10 middlebox attachments. Supported middlebox attachment types are Network Function (AWS Network Firewall), VPC and VPN attachments.

Topics

- [Add AWS Transit Gateway metering policy middlebox attachments](#)
- [Remove AWS Transit Gateway metering policy middlebox attachments](#)

Add AWS Transit Gateway metering policy middlebox attachments

You can add middlebox attachments to integrate network appliances into your Transit Gateway metering policy. This allows you to route specific traffic through security appliances, load balancers, or other network functions while maintaining granular cost allocation control.

Important

- Ensure middlebox appliances are properly configured and accessible
- Test traffic routing before applying to production workloads
- Monitor middlebox performance to avoid introducing latency
- Configure appropriate failover behavior for high availability

Add middlebox attachments using the console

To add a middlebox attachment entry

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Metering policies**.
3. Select the metering policy ID link to view its details.
4. Choose the **Middlebox attachments** tab.
5. Choose **Add**.
6. When prompted, Select the middlebox attachment IDs that should be treated as middleboxes for specialized billing. You can select up to 10 middlebox attachments.
7. Choose **Add middlebox attachments** to save the configuration.

Add middlebox attachments using the AWS CLI

Use the **modify-transit-gateway-metering-policy** command to add attachments.

Before you begin, ensure you have the following required parameters:

- **--transit-gateway-metering-policy-id** - The ID of the existing metering policy
- **--add-middle-box-attachment-ids** - One or more attachment IDs to add to the policy (for adding attachments)

To add middlebox attachments to an existing policy using the AWS CLI

1. In the following example, **modify-transit-gateway-metering-policy** is used to add four middlebox attachments to an existing metering policy. The command adds the specified attachment IDs to the existing list without removing current attachments:

```
aws ec2 modify-transit-gateway-metering-policy \
  --transit-gateway-metering-policy-id tgw-mp-0123456789abcdefg \
  --add-middle-box-attachment-ids tgw-attach-0bdc681c211bf71f3 tgw-
  attach-0987654321fedcba0 tgw-attach-0456789012345abcd tgw-attach-0fedcba0987654321
```

2. In the following example response, the JSON output shows the updated policy configuration with all four middlebox attachments now included:

```
{
```

```
"TransitGatewayMeteringPolicy": {
  "TransitGatewayMeteringPolicyId": "tgw-mp-0123456789abcdefg",
  "TransitGatewayId": "tgw-0ecec6433f4bfe55a",
  "MiddleBoxAttachmentIds": [
    "tgw-attach-0bdc681c211bf71f3",
    "tgw-attach-0987654321fedcba0",
    "tgw-attach-0456789012345abcd",
    "tgw-attach-0fedcba0987654321"
  ],
  "State": "available",
  "UpdateEffectiveAt": "2024-09-05T16:00:00.000Z"
}
```

Remove AWS Transit Gateway metering policy middlebox attachments

By default, metering costs are attributed to the middlebox attachment owner. However, you can modify these assignments to ensure costs are properly allocated to the actual source or destination of the traffic. You can add or remove up to 10 total middlebox attachment for a metering policy.

Remove middlebox attachments using the console

Use the Amazon VPC console to remove middlebox attachments from your metering policy configuration.

To remove middlebox attachments

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit gateways, Metering policies**.
3. Select the metering policy that you want to modify.
4. Choose the **Middlebox attachments** tab.
5. Select up to 10 middlebox attachments to remove from the metering policy.
6. Choose **Remove**.
7. When prompted, you can update your chosen middlebox attachments to remove. Traffic through removed attachments will be metered to the middlebox attachment owner.
8. Choose **Remove middlebox attachments**.

Remove middlebox attachments using the AWS CLI

Use the **modify-transit-gateway-metering-policy** command to remove attachments.

Before you begin, ensure you have the following required parameters:

- **--transit-gateway-metering-policy-id** - The ID of the existing metering policy
- **--remove-middle-box-attachment-ids** - One or more attachment IDs to remove from the policy (for removing attachments)

To remove middlebox attachments from an existing policy using the AWS CLI

1. In the following example, **modify-transit-gateway-metering-policy** is used to remove two specific middlebox attachments from an existing metering policy. The command removes only the specified attachment IDs while preserving the remaining attachments:

```
aws ec2 modify-transit-gateway-metering-policy \
  --transit-gateway-metering-policy-id tgw-mp-0123456789abcdefg \
  --remove-middle-box-attachment-ids tgw-attach-0456789012345abcd tgw-
attach-0fedcba0987654321
```

2. In the following example response, the JSON output shows the updated policy configuration with the specified attachments removed and the remaining attachments still active:

```
{
  "TransitGatewayMeteringPolicy": {
    "TransitGatewayMeteringPolicyId": "tgw-mp-0123456789abcdefg",
    "TransitGatewayId": "tgw-0ecec6433f4bfe55a",
    "MiddleBoxAttachmentIds": [
      "tgw-attach-0bdc681c211bf71f3",
      "tgw-attach-0987654321fedcba0"
    ],
    "State": "available",
    "UpdateEffectiveAt": "2024-09-05T16:00:00.000Z"
  }
}
```

AWS Transit Gateway Flow Logs

Transit Gateway Flow Logs is a feature of AWS Transit Gateway that enables you to capture information about the IP traffic going to and from your transit gateways. Flow log data can be published to Amazon CloudWatch Logs, Amazon S3, or Firehose. After you create a flow log, you can retrieve and view its data in the chosen destination. Flow log data is collected outside of the path of your network traffic, and therefore does not affect network throughput or latency. You can create or delete flow logs without any risk of impact to network performance. Transit Gateway Flow Logs capture information related only to transit gateways, described in [the section called “Transit Gateway Flow Log records”](#). If you want to capture information about IP traffic going to and from network interfaces in your VPCs, use VPC Flow Logs. See [Logging IP traffic using VPC Flow Logs](#) in the *Amazon VPC User Guide* for more information.

Note

To create a transit gateway flow log, you must be the owner of the transit gateway. If you are not the owner, the transit gateway owner must give you permission.

Flow log data for a monitored transit gateway is recorded as *flow log records*, which are log events consisting of fields that describe the traffic flow. For more information, see [Transit Gateway Flow Log records](#).

To create a flow log, you specify:

- The resource for which to create the flow log
- The destinations to which you want to publish the flow log data

After you create a flow log, it can take several minutes to begin collecting and publishing data to the chosen destinations. Flow logs do not capture real-time log streams for your transit gateways.

You can apply tags to your flow logs. Each tag consists of a key and an optional value, both of which you define. Tags can help you organize your flow logs, for example by purpose or owner.

If you no longer require a flow log, you can delete it. Deleting a flow log disables the flow log service for the resource, and no new flow log records are created or published to CloudWatch Logs or Amazon S3. Deleting the flow log does not delete any existing flow log records or log streams (for CloudWatch Logs) or log file objects (for Amazon S3) for a transit gateway. To delete

an existing log stream, use the CloudWatch Logs console. To delete existing log file objects, use the Amazon S3 console. After you've deleted a flow log, it can take several minutes to stop collecting data. For more information, see [Delete an AWS Transit Gateway Flow Logs record](#).

You can create flow logs for your transit gateways that can publish data to CloudWatch Logs, Amazon S3, or Amazon Data Firehose. For more information, see the following:

- [Create a Flow Log that publishes to CloudWatch Logs](#)
- [Create a Flow Log that publishes to Amazon S3](#)
- [Create a Flow Log that publishes to Firehose](#)

Limitations

The following limitations apply to Transit Gateway Flow Logs:

- Multicast traffic is not supported.
- Connect attachments are not supported. All Connect flow logs appear under the transport attachment and must therefore be enabled on the transit gateway or the Connect transport attachment.
- Transit Gateway Flow Logs supports a maximum of 250 subscriptions per resource per account. To create additional subscriptions on a resource that has reached this limit, you must first delete existing subscriptions.

Transit Gateway Flow Log records

A flow log record represents a network flow in your transit gateway. Each record is a string with fields separated by spaces. A record includes values for the different components of the traffic flow, for example, the source, destination, and protocol.

When you create a flow log, you can use the default format for the flow log record, or you can specify a custom format.

Contents

- [Default format](#)
- [Custom format](#)
- [Available fields](#)

Default format

With the default format, the flow log records includes all version 2 to version 6 fields, in the order shown in the [available fields](#) table. You cannot customize or change the default format. To capture additional fields or a different subset of fields, specify a custom format instead.

Custom format

With a custom format, you specify which fields are included in the flow log records and in which order. This enables you to create flow logs that are specific to your needs, and to omit fields that are not relevant. Using a custom format can reduce the need for separate processes to extract specific information from the published flow logs. You can specify any number of the available flow log fields, but you must specify at least one.

Available fields

The following table describes all of the available fields for a transit gateway flow log record. The **Version** column indicates which version the field was introduced in.

When publishing flow log data to Amazon S3, the data type for the fields depends on the flow log format. If the format is plain text, all fields are of type STRING. If the format is Parquet, see the table for the field data types.

If a field is not applicable or could not be computed for a specific record, the record displays a '-' symbol for that entry. Metadata fields that do not come directly from the packet header are best effort approximations, and their values might be missing or inaccurate.

Field	Description	Version
version	Indicates the version in which the field was introduced. The default format includes all version 2 fields, in the same order that they appear in the table. Parquet data type: INT_32	2
resource-type	The type of resource on which the subscription is created. For Transit Gateway Flow Logs, this will be TransitGateway. Parquet data type: STRING	6

Field	Description	Version
account-id	The AWS account ID of the owner of the source transit gateway. Parquet data type: STRING	2
tgw-id	The ID of the transit gateway for which traffic is being recorded. Parquet data type: STRING	6
tgw-attachment-id	The ID of the transit gateway attachment for which traffic is being recorded. Parquet data type: STRING	6
tgw-src-vpc-account-id	The AWS account ID for the source VPC traffic. Parquet data type: STRING	6
tgw-dst-vpc-account-id	The AWS account ID for the destination VPC traffic. Parquet data type: STRING	6
tgw-src-vpc-id	The ID of the source VPC for the transit gateway Parquet data type: STRING	6
tgw-dst-vpc-id	The ID of the destination VPC for the transit gateway. Parquet data type: STRING	6
tgw-src-subnet-id	The ID of the subnet for the transit gateway source traffic. Parquet data type: STRING	6
tgw-dst-subnet-id	The ID of the subnet for the transit gateway destination traffic. Parquet data type: STRING	6
tgw-src-eni	The ID of the source transit gateway attachment ENI for the flow. Parquet data type: STRING	6

Field	Description	Version
tgw-dst-eni	The ID of the destination transit gateway attachment ENI for the flow. Parquet data type: STRING	6
tgw-src-az-id	The ID of the Availability Zone that contains the source transit gateway for which traffic is recorded. If the traffic is from a sublocation, the record displays a '-' symbol for this field. Parquet data type: STRING	6
tgw-dst-az-id	The ID of the Availability Zone that contains the destination transit gateway for which traffic is recorded. Parquet data type: STRING	6
tgw-pair-attachment-id	Depending on the flow direction, this is either the egress or ingress attachment ID of the flow. Parquet data type: STRING	6
srcaddr	The source address for incoming traffic. Parquet data type: STRING	2
dstaddr	The destination address for outgoing traffic. Parquet data type: STRING	2
srcport	The source port of the traffic. Parquet data type: INT_32	2
dstport	The destination port of the traffic. Parquet data type: INT_32	2

Field	Description	Version
protocol	The IANA protocol number of the traffic. For more information, see Assigned Internet Protocol Numbers. Parquet data type: INT_32	2
packets	The number of packets transferred during the flow. Parquet data type: INT_64	2
bytes	The number of bytes transferred during the flow. Parquet data type: INT_64	2
start	The time, in Unix seconds, when the first packet of the flow was received within the aggregation interval. This might be up to 60 seconds after the packet was transmitted or received on the transit gateway. Parquet data type: INT_64	2
end	The time, in Unix seconds, when the last packet of the flow was received within the aggregation interval. This might be up to 60 seconds after the packet was transmitted or received on the transit gateway. Parquet data type: INT_64	2
log-status	The 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 aggregation interval. • SKIPDATA — Some flow log records were skipped during the aggregation interval. This might be because of an internal capacity constraint, or an internal error. Parquet data type: STRING	2

Field	Description	Version
type	The type of traffic. Possible values are IPv4 IPv6 EFA. For more information, see Elastic Fabric Adapter in the <i>Amazon EC2 User Guide</i>. Parquet data type: STRING	3
packets-lost-no-route	The packets lost due to no route being specified. Parquet data type: INT_64	6
packets-lost-blackhole	The packets lost due to a black hole. Parquet data type: INT_64	6
packets-lost-mtu-exceeded	The packets lost due to the size exceeding the MTU. Parquet data type: INT_64	6
packets-lost-ttl-expired	The packets lost due to the expiration of time-to-live. Parquet data type: INT_64	6

Field	Description	Version
tcp-flags	The bitmask value for the following TCP flags: • FIN — 1 • SYN — 2 • RST — 4 • PSH — 8 • ACK — 16 • SYN-ACK — 18 • URG — 32  Important When a flow log entry consists of only ACK packets, the flag value is 0, not 16. For general information about TCP flags (such as the meaning of flags like FIN, SYN, and ACK), see TCP segment structure on Wikipedia. TCP flags can be OR-ed during the aggregation interval. For short connections, the flags might be set on the same line in the flow log record, for example, 19 for SYN-ACK and FIN, and 3 for SYN and FIN. Parquet data type: INT_32	3
region	The Region that contains the transit gateway where traffic is recorded. Parquet data type: STRING	4

Field	Description	Version
flow-direction	The direction of the flow with respect to the transit gateway. The possible values are: ingress egress. Parquet data type: STRING	5
pkt-src-aws-service	The name of the subset of IP address ranges for the srcaddr if the source IP address is for an AWS service. The possible values are: AMAZON AMAZON_APPFLOW AMAZON_CONNECT API_GATEWAY CHIME_MEETINGS CHIME_VOICECONNECTOR CLOUD9 CLOUDFRONT CODEBUILD DYNAMODB EBS EC2 EC2_INSTANCE_CONNECT GLOBALACCELERATOR KINESIS_VIDEO_STREAMS ROUTE53 ROUTE53_HEALTHCHECKS ROUTE53_HEALTHCHECKS_PUBLISHING ROUTE53_RESOLVER S3 WORKSPACES_GATEWAYS. Parquet data type: STRING	5
pkt-dst-aws-service	The name of the subset of IP address ranges for the dstaddr field, if the destination IP address is for an AWS service. For a list of possible values, see the pkt-src-aws-service field. Parquet data type: STRING	5

Control the use of flow logs

By default, users do not have permission to work with flow logs. You can create a user policy that grants users the permissions to create, describe, and delete flow logs. For more information, see [Granting IAM Users Required Permissions for Amazon EC2 Resources](#) in the *Amazon EC2 API Reference*.

The following is an example policy that grants users full permissions to create, describe, and delete flow logs.

JSON

```
{
  "Version": "2012-10-17",
```

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ec2:DeleteFlowLogs",
      "ec2:CreateFlowLogs",
      "ec2:DescribeFlowLogs"
    ],
    "Resource": "*"
  }
]
```

Some additional IAM role and permission configuration is required, depending on whether you're publishing to CloudWatch Logs or Amazon S3. For more information, see [AWS Transit Gateway Flow Logs records in Amazon CloudWatch Logs](#) and [AWS Transit Gateway Flow Logs records in Amazon S3](#).

Transit Gateway Flow Logs pricing

Data ingestion and storage charges for vended logs apply when you publish transit gateway flow logs. For more information about pricing when publishing vended logs, open [Amazon CloudWatch Pricing](#), and then under **Paid tier**, select **Logs** and find **Vended Logs**.

Create or update an IAM role for AWS Transit Gateway Flow Logs

You can update an existing role or use the following procedure to create a new role for use with flow logs using the AWS Identity and Access Management console.

To create an IAM role for flow logs

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**, **Create role**.
3. For **Select type of trusted entity**, choose **AWS service**. For **Use case**, choose **EC2**. Choose **Next**.
4. On the **Add permissions** page, choose **Next: Tags** and optionally add tags. Choose **Next**.

5. On the **Name, review, and create** page enter a name for your role and optionally provide a **Description**. Choose **Create role**.
6. Choose the name of your role. For **Add permissions**, choose **Create inline policy**, and then choose the **JSON** tab.
7. Copy the first policy from [IAM roles for publishing flow logs to CloudWatch Logs](#) and paste it in the window. Choose **Review policy**.
8. Enter a name for your policy, and choose **Create policy**.
9. Select the name of your role. For **Trust relationships**, choose **Edit trust relationship**. In the existing policy document, change the service from `ec2.amazonaws.com` to `vpc-flow-logs.amazonaws.com`. Choose **Update Trust Policy**.
10. On the **Summary** page, note the ARN for your role. You need this ARN when you create your flow log.

AWS Transit Gateway Flow Logs records in Amazon CloudWatch Logs

Flow logs can publish flow log data directly to Amazon CloudWatch.

When published to CloudWatch Logs, the flow log data is published to a log group, and each transit gateway has a unique log stream in the log group. Log streams contain flow log records. You can create multiple flow logs that publish data to the same log group. If the same transit gateway is present in one or more flow logs in the same log group, it has one combined log stream. If you've specified that one flow log should capture rejected traffic, and the other flow log should capture accepted traffic, then the combined log stream captures all traffic.

Data ingestion and archival charges for vended logs apply when you publish flow logs to CloudWatch Logs. For more information, see [Amazon CloudWatch Pricing](#).

In CloudWatch Logs, the **timestamp** field corresponds to the start time that's captured in the flow log record. The **ingestionTime** field provides the date and time when the flow log record was received by CloudWatch Logs. The timestamp is later than the end time that's captured in the flow log record.

For more information about CloudWatch Logs, see [Logs sent to CloudWatch Logs](#) in the *Amazon CloudWatch Logs User Guide*.

Contents

- [IAM roles for publishing flow logs to CloudWatch Logs](#)
- [Permissions for IAM users to pass a role](#)
- [Create an AWS Transit Gateway Flow Logs record that publishes to Amazon CloudWatch Logs](#)
- [View AWS Transit Gateway Flow Logs records in Amazon CloudWatch](#)
- [Process AWS Transit Gateway Flow Logs records in Amazon CloudWatch Logs](#)

IAM roles for publishing flow logs to CloudWatch Logs

The IAM role that's associated with your flow log must have sufficient permissions to publish flow logs to the specified log group in CloudWatch Logs. The IAM role must belong to your AWS account.

The IAM policy that's attached to your IAM role must include at least the following permissions.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Resource": "*"
    }
  ]
}
```

Also ensure that your role has a trust relationship that allows the flow logs service to assume the role.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vpc-flow-logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

We recommend that you use the `aws:SourceAccount` and `aws:SourceArn` condition keys to protect yourself against [the confused deputy problem](#). For example, you could add the following condition block to the previous trust policy. The source account is the owner of the flow log and the source ARN is the flow log ARN. If you don't know the flow log ID, you can replace that portion of the ARN with a wildcard (*) and then update the policy after you create the flow log.

```
"Condition": {
  "StringEquals": {
    "aws:SourceAccount": "account_id"
  },
  "ArnLike": {
    "aws:SourceArn": "arn:aws:ec2:region:account_id:vpc-flow-log/flow-log-id"
  }
}
```

Permissions for IAM users to pass a role

Users must also have permissions to use the `iam:PassRole` action for the IAM role that's associated with the flow log.

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::111122223333:role/flow-log-role-name"
  }
]
```

Create an AWS Transit Gateway Flow Logs record that publishes to Amazon CloudWatch Logs

You can create flow logs for transit gateways. If you perform these steps as an IAM user, ensure that you have permissions to use the `iam:PassRole` action. For more information, see [Permissions for IAM users to pass a role](#).

You can create an Amazon CloudWatch flow log using either the Amazon VPC Console or the AWS CLI.

To create a transit gateway flow log using the console

1. Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit gateways**.
3. Choose the checkboxes for one or more transit gateways and choose **Actions, Create flow log**.
4. For **Destination**, choose **Send to CloudWatch Logs**.
5. For **Destination log group**, choose the name of a current destination log group.

Note

If the destination log group does not yet exist, entering a new name in this field will create a new destination log group.

6. For **IAM role**, specify the name of the role that has permissions to publish logs to CloudWatch Logs.

7. For **Log record format**, select the format for the flow log record.
 - To use the default format, choose **AWS default format**.
 - To use a custom format, choose **Custom format** and then select fields from **Log format**.
8. (Optional) Choose **Add new tag** to apply tags to the flow log.
9. Choose **Create flow log**.

To create a flow log using the command line

Use one of the following commands.

- [create-flow-logs](#) (AWS CLI)
- [New-EC2FlowLog](#) (AWS Tools for Windows PowerShell)

The following AWS CLI example creates a flow log that captures transit gateway information. The flow logs are delivered to a log group in CloudWatch Logs called `my-flow-logs`, in account 123456789101, using the IAM role `publishFlowLogs`.

```
aws ec2 create-flow-logs --resource-type TransitGateway --resource-ids
tgw-1a2b3c4d --log-group-name my-flow-logs --deliver-logs-permission-arn
arn:aws:iam::123456789101:role/publishFlowLogs
```

View AWS Transit Gateway Flow Logs records in Amazon CloudWatch

You can view your flow log records using the CloudWatch Logs console or Amazon S3 console, depending on the chosen destination type. It might take a few minutes after you've created your flow log for it to be visible in the console.

To view flow log records published to CloudWatch Logs

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Logs**, and select the log group that contains your flow log. A list of log streams for each transit gateway is displayed.
3. Select the log stream that contains the ID of the transit gateway that you want to view the flow log records for. For more information, see [Transit Gateway Flow Log records](#).

Process AWS Transit Gateway Flow Logs records in Amazon CloudWatch Logs

You can work with flow log records as you would with any other log events collected by CloudWatch Logs. For more information about monitoring log data and metric filters, see [Creating metrics from log events using filters](#) in the *Amazon CloudWatch User Guide*.

Example: Create a CloudWatch metric filter and alarm for a flow log

In this example, you have a flow log for `tgw-123abc456bca`. You want to create an alarm that alerts you if there have been 10 or more rejected attempts to connect to your instance over TCP port 22 (SSH) within a 1-hour time period. First, you must create a metric filter that matches the pattern of the traffic for which to create the alarm. Then, you can create an alarm for the metric filter.

To create a metric filter for rejected SSH traffic and create an alarm for the filter

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Logs, Log groups**.
3. Select the checkbox for the log group, and then choose **Actions, Create metric filter**.
4. For **Filter Pattern**, enter the following.

```
[version, resource_type, account_id, tgw_id="tgw-123abc456bca", tgw_attachment_id, tgw_src_vpc_account_id, tgw_dst_vpc_account_id, tgw_src_vpc_id, tgw_dst_vpc_id, tgw_src_subnet_id, tgw_dst_subnet_id, tgw_src_eni, tgw_dst_eni, tgw_src_az_id, tgw_dst_az_id, tgw_pair_attachment_id, srcaddr="10.0.0.1", dstaddr, srcport="80", dstport, protocol="6", packets, bytes, start, end, log_status, type, packets_lost_no_route, packets_lost_blackhole, packets_lost_mtu_exceeded, packets_lost_ttl_expired, tcp_flags, region, flow_direction, pkt_src_aws_service, pkt_dst_aws_service]
```

5. For **Select log data to test**, select the log stream for your transit gateway. (Optional) To view the lines of log data that match the filter pattern, choose **Test pattern**. When you're ready, choose **Next**.
6. Enter a filter name, metric namespace, and metric name. Set the metric value to **1**. When you're done, choose **Next** and then choose **Create metric filter**.
7. In the navigation pane, choose **Alarms, All alarms**.
8. Choose **Create alarm**.

9. Choose the namespace for the metric filter that you created.

It can take a few minutes for a new metric to display in the console.

10. Select the metric name that you created, and then choose **Select metric**.

11. Configure the alarm as follows, and then choose **Next**:

- For **Statistic**, choose **Sum**. This ensure that you capture the total number of data points for the specified time period.
- For **Period**, choose **1 hour**.
- For **Whenever**, choose **Greater/Equal** and enter **10** for the threshold.
- For **Additional configuration, Datapoints to alarm**, leave the default of **1**.

12. For **Notification**, select an existing SNS topic, or choose **Create new topic** to create a new one. Choose **Next**.

13. Enter a name and description for the alarm and choose **Next**.

14. When you are done configuring the alarm, choose **Create alarm**.

AWS Transit Gateway Flow Logs records in Amazon S3

Flow logs can publish flow log data to Amazon S3.

When publishing to Amazon S3, flow log data is published to an existing Amazon S3 bucket that you specify. Flow log records for all of the monitored transit gateways are published to a series of log file objects that are stored in the bucket.

Data ingestion and archival charges are applied by Amazon CloudWatch for vended logs when you publish flow logs to Amazon S3. For more information on CloudWatch pricing for vended logs, open [Amazon CloudWatch Pricing](#), choose **Logs**, and then find **Vended Logs**.

To create an Amazon S3 bucket for use with flow logs, see [Create a bucket](#) in the *Amazon S3 User Guide*.

For more information about multiple account logging, see [Central Logging](#) in the AWS Solutions Library.

For more information about CloudWatch Logs, see [Logs sent to Amazon S3](#) in the *Amazon CloudWatch Logs User Guide*.

Contents

- [Flow log files](#)
- [IAM policy for IAM principals that publish flow logs to Amazon S3](#)
- [Amazon S3 bucket permissions for flow logs](#)
- [Required key policy for use with SSE-KMS](#)
- [Amazon S3 log file permissions](#)
- [Create the AWS Transit Gateway Flow Logs source account role for Amazon S3](#)
- [Create an AWS Transit Gateway Flow Logs record that publishes to Amazon S3](#)
- [View AWS Transit Gateway Flow Logs records in Amazon S3](#)
- [Processed AWS Transit Gateway Flow Logs records in Amazon S3](#)

Flow log files

VPC Flow Logs is a feature that collects flow log records, consolidates them into log files, and then publishes the log files to the Amazon S3 bucket at 5-minute intervals. Each log file contains flow log records for the IP 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. Then it publishes the flow log to the Amazon S3 bucket, and creates a new log file.

In Amazon S3, the **Last modified** field for the flow log file indicates the date and time when the file was uploaded to the Amazon S3 bucket. This is later than the timestamp in the file name, and differs by the amount of time taken to upload the file to the Amazon S3 bucket.

Log file format

You can specify one of the following formats for the log files. Each file is compressed into a single Gzip file.

- **Text** – Plain text. This is the default format.
- **Parquet** – Apache Parquet is a columnar data format. Queries on data in Parquet format are 10 to 100 times faster compared to queries on data in plain text. Data in Parquet format with Gzip compression takes 20 percent less storage space than plain text with Gzip compression.

Log file options

You can optionally specify the following options.

- **Hive-compatible S3 prefixes** – Enable Hive-compatible prefixes instead of importing partitions into your Hive-compatible tools. Before you run queries, use the **MSCK REPAIR TABLE** command.
- **Hourly partitions** – If you have a large volume of logs and typically target queries to a specific hour, you can get faster results and save on query costs by partitioning logs on an hourly basis.

Log file S3 bucket structure

Log files are saved to the specified Amazon S3 bucket using a folder structure that is based on the flow log's ID, Region, creation date, and destination options.

By default, the files are delivered to the following location.

```
bucket-and-optional-prefix/AWSLogs/account_id/vpcflowlogs/region/year/month/day/
```

If you enable Hive-compatible S3 prefixes, the files are delivered to the following location.

```
bucket-and-optional-prefix/AWSLogs/aws-account-id=account_id/service=vpcflowlogs/aws-region=region/year=year/month=month/day=day/
```

If you enable hourly partitions, the files are delivered to the following location.

```
bucket-and-optional-prefix/AWSLogs/account_id/vpcflowlogs/region/year/month/day/hour/
```

If you enable Hive-compatible partitions and partition the flow log per hour, the files are delivered to the following location.

```
bucket-and-optional-prefix/AWSLogs/aws-account-id=account_id/service=vpcflowlogs/aws-region=region/year=year/month=month/day=day/hour=hour/
```

Log file names

The file name of a log file is based on the flow log ID, Region, and creation date and time. File names use the following format.

```
aws_account_id_vpcflowlogs_region_flow_log_id_YYYYMMDDTHHmmZ_hash.log.gz
```

The following is an example of a log file for a flow log created by AWS account 123456789012, for a resource in the us-east-1 Region, on June 20, 2018 at 16:20 UTC. The file contains the flow log records with an end time between 16:20:00 and 16:24:59.

```
123456789012_vpcflowlogs_us-east-1_fl-1234abcd_20180620T1620Z_fe123456.log.gz
```

IAM policy for IAM principals that publish flow logs to Amazon S3

The IAM principal that creates the flow log must have the following permissions, which are required to publish flow logs to the destination Amazon S3 bucket.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs>DeleteLogDelivery"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon S3 bucket permissions for flow logs

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

If the user creating the flow log owns the bucket and has `PutBucketPolicy` and `GetBucketPolicy` permissions for the bucket, we automatically attach the following policy to the bucket. This new auto-generated policy is appended to the original policy.

Otherwise, the bucket owner must add this policy to the bucket, specifying the AWS account ID of the flow log creator, or flow log creation fails. For more information, see [Bucket policies](#) in the *Amazon Simple Storage Service User Guide*.

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/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl": "bucket-owner-full-control",
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:logs:us-east-1:123456789012:*"
        }
      }
    },
    {
      "Sid": "AWSLogDeliveryCheck",
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": [
        "s3:GetBucketAcl"
      ],
      "Resource": "arn:aws:s3:::bucket_name",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:logs:us-east-1:123456789012:*"
        }
      }
    }
  ]
}
```

```
}
```

The ARN that you specify for *my-s3-arn* depends on whether you use Hive-compatible S3 prefixes.

- Default prefixes

```
arn:aws:s3:::bucket_name/optional_folder/AWSLogs/account_id/*
```

- Hive-compatible S3 prefixes

```
arn:aws:s3:::bucket_name/optional_folder/AWSLogs/aws-account-id=account_id/*
```

As a best practice, we recommend that you grant these permissions to the log delivery service principal instead of individual AWS account ARNs. It is also a best practice to use the `aws:SourceAccount` and `aws:SourceArn` condition keys to protect against [the confused deputy problem](#). The source account is the owner of the flow log and the source ARN is the wildcard (*) ARN of the logs service.

Required key policy for use with SSE-KMS

You can protect the data in your Amazon S3 bucket by enabling either Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3) or Server-Side Encryption with KMS Keys (SSE-KMS). For more information, see [Protecting data using server-side encryption](#) in the *Amazon S3 User Guide*.

With SSE-KMS, you can use either an AWS managed key or a customer managed key. With an AWS managed key, you can't use cross-account delivery. Flow logs are delivered from the log delivery account, so you must grant access for cross-account delivery. To grant cross-account access to your S3 bucket, use a customer managed key and specify the Amazon Resource Name (ARN) of the customer managed key when you enable bucket encryption. For more information, see [Specifying server-side encryption with AWS KMS](#) in the *Amazon S3 User Guide*.

When you use SSE-KMS with a customer managed key, you must add the following to the key policy for your key (not the bucket policy for your S3 bucket), so that VPC Flow Logs can write to your S3 bucket.

Note

Using S3 Bucket Keys allows you to save on AWS Key Management Service (AWS KMS) request costs by decreasing your requests to AWS KMS for Encrypt, GenerateDataKey, and Decrypt operations through the use of a bucket-level key. By design, subsequent requests that take advantage of this bucket-level key do not result in AWS KMS API requests or validate access against the AWS KMS key policy.

```
{
  "Sid": "Allow Transit Gateway Flow Logs to use the key",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "delivery.logs.amazonaws.com"
    ]
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "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*.

Create the AWS Transit Gateway Flow Logs source account role for Amazon S3

From the source account, create the source role in the AWS Identity and Access Management console.

To create the source account role

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**.
3. Choose **Create policy**.
4. On the Create policy page, do the following:
 1. Choose **JSON**.
 2. Replace the contents of this window with the permissions policy at the start of this section.
 3. Choose **Next: Tags** and **Next: Review**.
 4. Enter a name for your policy and an optional description, and then choose **Create policy**.
5. In the navigation pane, choose **Roles**.
6. Choose **Create role**.
7. For the **Trusted entity type**, choose **Custom trust policy**. For **Custom trust policy**, replace "Principal": {}, with the following, which specifies the log delivery service. Choose **Next**.

```
"Principal": {  
  "Service": "delivery.logs.amazonaws.com"  
},
```
8. On the **Add permissions** page, select the checkbox for the policy that you created earlier in this procedure, and then choose **Next**.
9. Enter a name for your role and optionally provide a description.
10. Choose **Create role**.

Create an AWS Transit Gateway Flow Logs record that publishes to Amazon S3

After you have created and configured your Amazon S3 bucket, you can create flow logs for transit gateways. You can create an Amazon S3 flow log using either the Amazon VPC Console or the AWS CLI.

To create a transit gateway flow log that publishes to Amazon S3 using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit gateways** or **Transit gateway attachments**.
3. Select the checkboxes for one or more transit gateways or transit gateway attachments.
4. Choose **Actions, Create flow log**.
5. Configure the flow log settings. For more information, see [To configure flow log settings](#).

To configure flow log settings using the console

1. For **Destination**, choose **Send to an S3 bucket**.
2. For **S3 bucket ARN**, specify the Amazon Resource Name (ARN) of an existing Amazon S3 bucket. You can optionally include a subfolder. For example, to specify a subfolder named my-logs in a bucket named my-bucket, use the following ARN:

```
arn:aws:s3::my-bucket/my-logs/
```

The bucket cannot use AWSLogs as a subfolder name, as this is a reserved term.

If you own the bucket, we automatically create a resource policy and attach it to the bucket. For more information, see [Amazon S3 bucket permissions for flow logs](#).

3. For **Log record format**, specify the format for the flow log record.
 - To use the default flow log record format, choose **AWS default format**.
 - To create a custom format, choose **Custom format**. For **Log format**, choose the fields to include in the flow log record.
4. For **Log file format**, specify the format for the log file.
 - **Text** – Plain text. This is the default format.

- **Parquet** – Apache Parquet is a columnar data format. Queries on data in Parquet format are 10 to 100 times faster compared to queries on data in plain text. Data in Parquet format with Gzip compression takes 20 percent less storage space than plain text with Gzip compression.
5. (Optional) To use Hive-compatible S3 prefixes, choose **Hive-compatible S3 prefix, Enable**.
 6. (Optional) To partition your flow logs per hour, choose **Every 1 hour (60 mins)**.
 7. (Optional) To add a tag to the flow log, choose **Add new tag** and specify the tag key and value.
 8. Choose **Create flow log**.

To create a flow log that publishes to Amazon S3 using a command line tool

Use one of the following commands.

- [create-flow-logs](#) (AWS CLI)
- [New-EC2FlowLog](#) (AWS Tools for Windows PowerShell)

The following AWS CLI example creates a flow log that captures all transit gateway traffic for VPC `tgw-00112233344556677` and delivers the flow logs to an Amazon S3 bucket called `flow-log-bucket`. The `--log-format` parameter specifies a custom format for the flow log records.

```
aws ec2 create-flow-logs --resource-type TransitGateway --resource-ids
tgw-00112233344556677 --log-destination-type s3 --log-destination arn:aws:s3:::flow-
log-bucket/my-custom-flow-logs/
```

View AWS Transit Gateway Flow Logs records in Amazon S3

To view flow log records published to Amazon S3

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. For **Bucket name**, select the bucket to which the flow logs are published.
3. For **Name**, select the checkbox next to the log file. On the object overview panel, choose **Download**.

Processed AWS Transit Gateway Flow Logs 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.

AWS Transit Gateway, Flow Logs records in Amazon Data Firehose

Topics

- [IAM roles for cross account delivery](#)
- [Create the AWS Transit Gateway Flow Logs source account role for Amazon Data Firehose](#)
- [Create the AWS Transit Gateway Flow Logs destination account role for Amazon Data Firehose](#)
- [Create an AWS Transit Gateway Flow Logs record that publishes to Amazon Data Firehose](#)

Flow logs can publish flow log data directly to Firehose. You can choose to publish flow logs to the same account as the resource monitor or to a different account.

Prerequisites

When publishing to Firehose, flow log data is published to a Firehose delivery stream, in plain text format. You must first have created a Firehose delivery stream. For the steps to create a delivery stream, see [Creating an Amazon Data Firehose Delivery Stream](#) in the *Amazon Data Firehose Developer Guide*.

Pricing

Standard ingestion and delivery charges apply. For more information, open [Amazon CloudWatch Pricing](#), select **Logs** and find **Vended Logs**.

IAM roles for cross account delivery

When you publish to Kinesis Data Firehose, you can choose a delivery stream that's in the same account as the resource to monitor (the source account), or in a different account (the destination account). To enable cross account delivery of flow logs to Firehose, you must create an IAM role in the source account and an IAM role in the destination account.

Roles

- [Source account role](#)
- [Destination account role](#)

Source account role

In the source account, create a role that grants the following permissions. In this example, the name of the role is `mySourceRole`, but you can choose a different name for this role. The last statement allows the role in the destination account to assume this role. The condition statements ensure that this role is passed only to the log delivery service, and only when monitoring the specified resource. When you create your policy, specify the VPCs, network interfaces, or subnets that you're monitoring with the condition key `iam:AssociatedResourceARN`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/mySourceRole",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "delivery.logs.amazonaws.com"
        },
        "StringLike": {
          "iam:AssociatedResourceARN": [
            "arn:aws:ec2:us-east-1:source-account:transit-gateway/
            tgw-0fb8421e2da853bf"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs>DeleteLogDelivery",
        "logs:ListLogDeliveries",
```

```

        "logs:GetLogDelivery"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::111122223333:role/
AWSLogDeliveryFirehoseCrossAccountRole"
    }
  ]
}

```

Ensure that this role has the following trust policy, which allows the log delivery service to assume the role.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Destination account role

In the destination account, create a role with a name that starts with **AWSLogDeliveryFirehoseCrossAccountRole**. This role must grant the following permissions.

JSON

```

{

```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "iam:CreateServiceLinkedRole",
      "firehose:TagDeliveryStream"
    ],
    "Resource": "*"
  }
]
```

Ensure that this role has the following trust policy, which allows the role that you created in the source account to assume this role.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/mySourceRole"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Create the AWS Transit Gateway Flow Logs source account role for Amazon Data Firehose

From the source account, create the source role in the AWS Identity and Access Management console.

To create the source account role

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**.
3. Choose **Create policy**.
4. On the Create policy page, do the following:
 1. Choose **JSON**.
 2. Replace the contents of this window with the permissions policy at the start of this section.
 3. Choose **Next: Tags** and **Next: Review**.
 4. Enter a name for your policy and an optional description, and then choose **Create policy**.
5. In the navigation pane, choose **Roles**.
6. Choose **Create role**.
7. For the **Trusted entity type**, choose **Custom trust policy**. For **Custom trust policy**, replace "Principal": {}, with the following, which specifies the log delivery service. Choose **Next**.

```
"Principal": {  
  "Service": "delivery.logs.amazonaws.com"  
},
```
8. On the **Add permissions** page, select the checkbox for the policy that you created earlier in this procedure, and then choose **Next**.
9. Enter a name for your role and optionally provide a description.
10. Choose **Create role**.

Create the AWS Transit Gateway Flow Logs destination account role for Amazon Data Firehose

From the destination account, create the destination role in the AWS Identity and Access Management console.

To create the destination account role

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.

2. In the navigation pane, choose **Policies**.
3. Choose **Create policy**.
4. On the Create policy page, do the following:
 1. Choose **JSON**.
 2. Replace the contents of this window with the permissions policy at the start of this section.
 3. Choose **Next: Tags** and **Next: Review**.
 4. Enter a name for your policy that starts with **AWSLogDeliveryFirehoseCrossAccountRole**, and then choose **Create policy**.
5. In the navigation pane, choose **Roles**.
6. Choose **Create role**.
7. For the **Trusted entity type**, choose **Custom trust policy**. For **Custom trust policy**, replace "Principal": {}, with the following, which specifies the log delivery service. Choose **Next**.

```
"Principal": {  
  "AWS": "arn:aws:iam::source-account:role/mySourceRole"  
},
```
8. On the **Add permissions** page, select the checkbox for the policy that you created earlier in this procedure, and then choose **Next**.
9. Enter a name for your role and optionally provide a description.
10. Choose **Create role**.

Create an AWS Transit Gateway Flow Logs record that publishes to Amazon Data Firehose

Create a Transit Gateway Flow Log that publishes to Amazon Data Firehose. Before you can create the flow log, ensure that you've set up the source and destination IAM account roles for cross-account delivery and that you've created the Firehose delivery stream. See [Amazon Data Firehose Flow Logs](#) for more information. You can create a Firehose flow log using either the Amazon VPC Console or the AWS CLI.

To create a transit gateway flow log that publishes to Firehose using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

2. In the navigation pane, choose **Transit gateways** or **Transit gateway attachments**.
3. Select the checkboxes for one or more transit gateways or transit gateway attachments.
4. Choose **Actions, Create flow log**.
5. For **Destination** choose Send to a **Firehose Delivery System**.
6. For the **Firehose Delivery Stream ARN**, choose the ARN of a delivery stream you created where the flow log is to be published.
7. For **Log record format**, specify the format for the flow log record.
 - To use the default flow log record format, choose **AWS default format**.
 - To create a custom format, choose **Custom format**. For **Log format**, choose the fields to include in the flow log record.
8. (Optional) To add a tag to the flow log, choose **Add new tag** and specify the tag key and value.
9. Choose **Create flow log**.

To create a flow log that publishes to Firehose using the command line tool

Use one of the following commands:

- [create-flow-logs](#) (AWS CLI)
- [New-EC2FlowLog](#) (AWS Tools for Windows PowerShell)

The following AWS CLI example creates a flow log that captures transit gateway information and delivers the flow log to the specified Firehose delivery stream.

```
aws ec2 create-flow-logs \
    --resource-type TransitGateway \
    --resource-ids tgw-1a2b3c4d \
    --log-destination-type kinesis-data-firehose \
    --log-destination arn:aws:firehose:us-
east-1:123456789012:deliverystream:flowlogs_stream
```

The following AWS CLI example creates a flow log that captures transit gateway information and delivers the flow log to a different Firehose delivery stream from the source account.

```
aws ec2 create-flow-logs \
```

```
--resource-type TransitGateway \  
--resource-ids gw-1a2b3c4d \  
--log-destination-type kinesis-data-firehose \  
--log-destination arn:aws:firehose:us-  
east-1:123456789012:deliverystream:flowlogs_stream \  
--deliver-logs-permission-arn arn:aws:iam::source-account:role/mySourceRole \  
--deliver-cross-account-role arn:aws:iam::destination-account:role/  
AWSLogDeliveryFirehoseCrossAccountRole
```

Create and manage AWS Transit Gateway Flow Logs using APIs or the CLI

You can perform the tasks described on this page using the command line.

The following limitations apply when using the [create-flow-logs](#) command:

- `--resource-ids` has a maximum constraint of 25 TransitGateway or TransitGatewayAttachment resource types.
- `--traffic-type` is not a required field by default. An error is returned if you provide this for transit gateway resource types. This limit applies only to transit gateway resource types.
- `--max-aggregation-interval` has a default value of 60, and is the only accepted value for transit gateway resource types. An error is returned if you try to pass any other value. This limit applies only to transit gateway resource types.
- `--resource-type` supports two new resource types, TransitGateway and TransitGatewayAttachment.
- `--log-format` includes all log fields for transit gateway resource types if you do not set which fields you want to include. This applies only to transit gateway resource types.

Create a flow log

- [create-flow-logs](#) (AWS CLI)
- [New-EC2FlowLog](#) (AWS Tools for Windows PowerShell)

Describe your flow logs

- [describe-flow-logs](#) (AWS CLI)

- [Get-EC2FlowLog](#) (AWS Tools for Windows PowerShell)

View your flow log records (log events)

- [get-log-events](#) (AWS CLI)
- [Get-CWLLogEvent](#) (AWS Tools for Windows PowerShell)

Delete a flow log

- [delete-flow-logs](#) (AWS CLI)
- [Remove-EC2FlowLog](#) (AWS Tools for Windows PowerShell)

View AWS Transit Gateway Flow Logs records

View information about your transit gateway flow logs through the Amazon VPC. When you choose a resource, all of the flow logs for that resource are listed. The information displayed includes the ID of the flow log, the flow log configuration, and information about the status of the flow log.

To view information about flow logs for transit gateways

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit gateways** or **Transit gateway attachments**.
3. Select a transit gateway or transit gateway attachment and choose **Flow Logs**. Information about the flow logs is displayed on the tab. The **Destination type** column indicates the destination to which the flow logs are published.

Manage AWS Transit Gateway Flow Logs tags

You can add or remove tags for a flow log in the Amazon EC2 and Amazon VPC consoles.

To add or remove tags for a transit gateway flow log

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit gateways** or **Transit gateway attachments**.
3. Select a transit gateway or transit gateway attachment
4. Choose **Manage tags** for the required flow log.

5. To add a new tag, choose **Create Tag**. To remove a tag, choose the delete button (x).
6. Choose **Save**.

Search AWS Transit Gateway Flow Logs records

You can search your flow log records that are published to CloudWatch Logs by using the CloudWatch Logs console. You can use [metric filters](#) to filter flow log records. Flow log records are space delimited.

To search flow log records using the CloudWatch Logs console

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Logs**, and then choose **Log groups**.
3. Select the log group that contains your flow log. A list of log streams for each transit gateway is displayed.
4. Select the individual log stream if you know the transit gateway that you are searching for. Alternatively, choose **Search Log Group** to search the entire log group. This might take some time if there are many transit gateways in your log group, or depending on the time range that you select.
5. For **Filter events**, enter the following string. This assumes that the flow log record uses the [default format](#).

```
[version, resource_type, account_id,tgw_id, tgw_attachment_id,
tgw_src_vpc_account_id, tgw_dst_vpc_account_id, tgw_src_vpc_id, tgw_dst_vpc_id,
tgw_src_subnet_id, tgw_dst_subnet_id, tgw_src_eni, tgw_dst_eni, tgw_src_az_id,
tgw_dst_az_id, tgw_pair_attachment_id, srcaddr, dstaddr, srcport, dstport,
protocol, packets, bytes,start,end, log_status, type,packets_lost_no_route,
packets_lost_blackhole, packets_lost_mtu_exceeded, packets_lost_ttl_expired,
tcp_flags,region, flow_direction, pkt_src_aws_service, pkt_dst_aws_service]
```

6. Modify the filter as needed by specifying values for the fields. The following examples filter by specific source IP addresses.

```
[version, resource_type, account_id,tgw_id, tgw_attachment_id,
tgw_src_vpc_account_id, tgw_dst_vpc_account_id, tgw_src_vpc_id, tgw_dst_vpc_id,
tgw_src_subnet_id, tgw_dst_subnet_id, tgw_src_eni, tgw_dst_eni, tgw_src_az_id,
tgw_dst_az_id, tgw_pair_attachment_id, srcaddr= 10.0.0.1, dstaddr,
srcport, dstport, protocol, packets, bytes,start,end, log_status,
```

```

type,packets_lost_no_route, packets_lost_blackhole, packets_lost_mtu_exceeded,
packets_lost_ttl_expired, tcp_flags,region, flow_direction, pkt_src_aws_service,
pkt_dst_aws_service]
[version, resource_type, account_id,tgw_id, tgw_attachment_id,
tgw_src_vpc_account_id, tgw_dst_vpc_account_id, tgw_src_vpc_id, tgw_dst_vpc_id,
tgw_src_subnet_id, tgw_dst_subnet_id, tgw_src_eni, tgw_dst_eni, tgw_src_az_id,
tgw_dst_az_id, tgw_pair_attachment_id, srcaddr= 10.0.2.*, dstaddr,
srcport, dstport, protocol, packets, bytes,start,end, log_status,
type,packets_lost_no_route, packets_lost_blackhole, packets_lost_mtu_exceeded,
packets_lost_ttl_expired, tcp_flags,region, flow_direction, pkt_src_aws_service,
pkt_dst_aws_service]

```

The following example filters by transit gateway ID tgw-123abc456bca, destination port, and number of bytes.

```

[version, resource_type, account_id,tgw_id=tgw-123abc456bca, tgw_attachment_id,
tgw_src_vpc_account_id, tgw_dst_vpc_account_id, tgw_src_vpc_id, tgw_dst_vpc_id,
tgw_src_subnet_id, tgw_dst_subnet_id, tgw_src_eni, tgw_dst_eni, tgw_src_az_id,
tgw_dst_az_id, tgw_pair_attachment_id, srcaddr, dstaddr, srcport, dstport =
80 || dstport = 8080, protocol, packets, bytes >= 500,start,end, log_status,
type,packets_lost_no_route, packets_lost_blackhole, packets_lost_mtu_exceeded,
packets_lost_ttl_expired, tcp_flags,region, flow_direction, pkt_src_aws_service,
pkt_dst_aws_service]

```

Delete an AWS Transit Gateway Flow Logs record

You can delete a transit gateway flow log using the Amazon VPC console.

These procedures disable the flow log service for a resource. Deleting a flow log does not delete the existing log streams from CloudWatch Logs or log files from Amazon S3. Existing flow log data must be deleted using the respective service's console. In addition, deleting a flow log that publishes to Amazon S3 does not remove the bucket policies and log file access control lists (ACLs).

To delete a transit gateway flow log

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Transit gateways**.
3. Choose a **Transit gateway ID**.
4. In the Flow logs section, choose the flow logs that you want to delete.

5. Choose **Actions**, and then choose **Delete flow logs**.
6. Confirm that you want to delete the flow by choosing **Delete**.

Metrics and events in AWS Transit Gateway

You can use the following features to monitor your transit gateways, analyze traffic patterns, and troubleshoot issues with your transit gateways.

CloudWatch metrics

You can use Amazon CloudWatch to retrieve statistics about data points for your transit gateways as an ordered set of time series data, known as *metrics*. You can use these metrics to verify that your system is performing as expected. For more information, see [CloudWatch metrics in AWS Transit Gateway](#).

Transit Gateway Flow Logs

You can use Transit Gateway Flow Logs to capture detailed information about the network traffic on your transit gateways. For more information, see [Transit Gateway Flow Logs](#).

VPC Flow Logs

You can use VPC Flow Logs to capture detailed information about the traffic going to and from the VPCs that are attached to your transit gateways. For more information, see [VPC Flow Logs](#) in the *Amazon VPC User Guide*.

CloudTrail logs

You can use AWS CloudTrail to capture detailed information about the calls made to the transit gateway API and store them as log files in Amazon S3. You can use these CloudTrail logs to determine which calls were made, the source IP address where the call came from, who made the call, when the call was made, and so on. For more information, see [CloudTrail logs](#).

CloudWatch Events using Network Manager

You can use AWS Network Manager to forward events to CloudWatch, and then route those events to target functions or streams. Network Manager generates events for topology changes, routing updates, and status updates, all of which can be used to alert you to changes in your transit gateways. For more information, see [Monitoring your global network with CloudWatch Events](#) in the *AWS Global Networks for Transit Gateways User Guide*.

CloudWatch metrics in AWS Transit Gateway

Amazon VPC publishes data points to Amazon CloudWatch for your transit gateways and transit gateway attachments. 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. Each data point has an associated timestamp and an optional unit of measurement.

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

Amazon VPC measures and sends its metrics to CloudWatch in 60-second intervals.

For more information, see the [Amazon CloudWatch User Guide](#).

Contents

- [Transit gateway metrics](#)
- [Attachment-level and availability zone metrics](#)
- [Transit gateway metric dimensions](#)

Transit gateway metrics

The AWS/TransitGateway namespace includes the following metrics.

All metrics are always reported. Their values are dependent on the traffic through the transit gateway. See [Transit gateway metric dimensions](#) for the supported dimensions.

Metric	Description
BytesDropCountBlackhole	The number of bytes dropped because they matched a blackhole route. Statistics: The only meaningful statistic is Sum.
BytesDropCountNoRoute	The number of bytes dropped because they did not match a route. Statistics: The only meaningful statistic is Sum.

Metric	Description
BytesDropCountNoPolicy	The number of bytes dropped because no policy rule matched. Equivalent to BytesDropCountNoRoute for standard route tables. A sustained non-zero value indicates missing or misconfigured rules. Statistics: The only meaningful statistic is Sum.
BytesIn	The number of bytes received by the transit gateway. Statistics: The only meaningful statistic is Sum.
BytesOut	The number of bytes sent from the transit gateway. Statistics: The only meaningful statistic is Sum.
PacketsIn	The number of packets received by the transit gateway. Statistics: The only meaningful statistic is Sum.
PacketsOut	The number of packets sent by the transit gateway. Statistics: The only meaningful statistic is Sum.
PacketDropCountBlackhole	The number of packets dropped because they matched a blackhole route. Statistics: The only meaningful statistic is Sum.
PacketDropCountNoRoute	The number of packets dropped because they did not match a route. Statistics: The only meaningful statistic is Sum.
PacketDropCountNoPolicy	The number of packets dropped because no policy rule matched. Equivalent to PacketDropCountNoRoute for standard route tables. A sustained non-zero value indicates missing or misconfigured rules. Statistics: The only meaningful statistic is Sum.

Metric	Description
PacketDropCountTTLExpired	The number of packets dropped because the TTL expired. Statistics: The only meaningful statistic is Sum.

Attachment-level and availability zone metrics

The following metrics are available for transit gateway attachments. All attachment metrics are published to the transit gateway owner's account. Individual attachment metrics are also published to the attachment owner's account. The attachment owner can view only the metrics for their own attachment. For more information on the supported attachment types, see [the section called "Resource attachments"](#).

Availability zone metrics are available for enabled for availability zones (AZs) on transit gateway attachments. Only VPC attachments support per-AZ metrics. All AZ-level metrics are published to the transit gateway owner's account. Individual AZ metrics for an attachment are also published to the attachment owner's account. The attachment owner can view only the per-AZ metrics for their own attachment.

All metrics are always reported. Their values are dependent on the traffic in and/or out of the transit gateway attachment. See [Transit gateway metric dimensions](#) for the supported dimensions.

Metric	Description
BytesDropCountBlackhole	The number of bytes dropped because they matched a blackhole route on the transit gateway attachment. Statistics: The only meaningful statistic is Sum.
BytesDropCountNoRoute	The number of bytes dropped because they did not match a route on the transit gateway attachment. Statistics: The only meaningful statistic is Sum.
BytesDropCountNoPolicy	The number of bytes dropped because no policy rule matched on the transit gateway attachment. Equivalent to BytesDropCountNoRoute for standard route tables. A sustained non-zero value indicates missing or misconfigured rules.

Metric	Description
	Statistics: The only meaningful statistic is Sum.
BytesIn	The number of bytes received by the transit gateway from the attachment. Statistics: The only meaningful statistic is Sum.
BytesOut	The number of bytes sent from the transit gateway to the attachment. Statistics: The only meaningful statistic is Sum.
PacketsIn	The number of packets received by the transit gateway from the attachment. Statistics: The only meaningful statistic is Sum.
PacketsOut	The number of packets sent by the transit gateway to the attachment. Statistics: The only meaningful statistic is Sum.
PacketDropCountBlackhole	The number of packets dropped because they matched a blackhole route on the transit gateway attachment. Statistics: The only meaningful statistic is Sum.
PacketDropCountNoRoute	The number of packets dropped because they did not match a route. Statistics: The only meaningful statistic is Sum.
PacketDropCountNoPolicy	The number of packets dropped because no policy rule matched on the transit gateway attachment. Equivalent to PacketDropCountNoRoute for standard route tables. A sustained non-zero value indicates missing or misconfigured rules. Statistics: The only meaningful statistic is Sum.

Metric	Description
PacketDropCountTTLExpired	The number of packets dropped because the TTL expired. Statistics: The only meaningful statistic is Sum.

Transit gateway metric dimensions

Filter transit gateway metric data using the following dimensions:

Dimension	Description
TransitGateway	Filters the metric data by transit gateway.
TransitGatewayAttachment	Filters the metric data by transit gateway attachment.
TransitGateway, AvailabilityZone	Filters the metric data by both transit gateway and availability zone.
TransitGatewayAttachment, AvailabilityZone	Filters the metric data by both transit gateway attachment and availability zone.

Log AWS Transit Gateway API calls using AWS CloudTrail

AWS Transit Gateway; is integrated with [AWS CloudTrail](#), a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all API calls for Transit Gateway as events. The calls captured include calls from the Transit Gateway console and code calls to the Transit Gateway API operations. Using the information collected by CloudTrail, you can determine the request that was made to Transit Gateway, the IP address from which the request was made, when it was made, and additional details.

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 user or user credentials.
- Whether the request was made on behalf of an IAM Identity Center user.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

CloudTrail is active in your AWS account when you create the account and you automatically have access to the CloudTrail **Event history**. The CloudTrail **Event history** provides a viewable, searchable, downloadable, and immutable record of the past 90 days of recorded management events in an AWS Region. For more information, see [Working with CloudTrail Event history](#) in the *AWS CloudTrail User Guide*. There are no CloudTrail charges for viewing the **Event history**.

For an ongoing record of events in your AWS account past 90 days, create a trail or a [CloudTrail Lake](#) event data store.

CloudTrail trails

A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. All trails created using the AWS Management Console are multi-Region. You can create a single-Region or a multi-Region trail by using the AWS CLI. Creating a multi-Region trail is recommended because you capture activity in all AWS Regions in your account. If you create a single-Region trail, you can view only the events logged in the trail's AWS Region. For more information about trails, see [Creating a trail for your AWS account](#) and [Creating a trail for an organization](#) in the *AWS CloudTrail User Guide*.

You can deliver one copy of your ongoing management events to your Amazon S3 bucket at no charge from CloudTrail by creating a trail, however, there are Amazon S3 storage charges. For more information about CloudTrail pricing, see [AWS CloudTrail Pricing](#). For information about Amazon S3 pricing, see [Amazon S3 Pricing](#).

CloudTrail Lake event data stores

CloudTrail Lake lets you run SQL-based queries on your events. CloudTrail Lake converts existing events in row-based JSON format to [Apache ORC](#) format. ORC is a columnar storage format that is optimized for fast retrieval of data. Events are aggregated into *event data stores*, which are immutable collections of events based on criteria that you select by applying [advanced event selectors](#). The selectors that you apply to an event data store control which events persist

and are available for you to query. For more information about CloudTrail Lake, see [Working with AWS CloudTrail Lake](#) in the *AWS CloudTrail User Guide*.

CloudTrail Lake event data stores and queries incur costs. When you create an event data store, you choose the [pricing option](#) you want to use for the event data store. The pricing option determines the cost for ingesting and storing events, and the default and maximum retention period for the event data store. For more information about CloudTrail pricing, see [AWS CloudTrail Pricing](#).

Transit Gateway management events

[Management events](#) provide information about management operations that are performed on resources in your AWS account. These are also known as control plane operations. By default, CloudTrail logs management events.

AWS Transit Gateway logs all Transit Gateway control plane operations as management events. For a list of the AWS Transit Gateway control plane operations that Transit Gateway logs to CloudTrail, see [AWS Transit Gateway actions](#) in the *Amazon EC2 API Reference*.

Transit Gateway event examples

An event represents a single request from any source and includes information about the requested API operation, the date and time of the operation, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so events don't appear in any specific order.

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

The log files include events for all API calls for your AWS account, not just transit gateway API calls. You can locate calls to the transit gateway API by checking for `eventSource` elements with the value `ec2.amazonaws.com`. To view a record for a specific action, such as `CreateTransitGateway`, check for `eventName` elements with the action name.

The following is an example CloudTrail log record for the transit gateway API for a user who created a transit gateway using the console. You can identify the console using the `userAgent`

element. You can identify the requested API call using the eventName elements. Information about the user (Alice) can be found in the userIdentity element.

Example: CreateTransitGateway

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:user/Alice",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2018-11-15T05:25:50Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CreateTransitGateway",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "198.51.100.1",
  "userAgent": "console.ec2.amazonaws.com",
  "requestParameters": {
    "CreateTransitGatewayRequest": {
      "Options": {
        "DefaultRouteTablePropagation": "enable",
        "AutoAcceptSharedAttachments": "disable",
        "DefaultRouteTableAssociation": "enable",
        "VpnEcmpSupport": "enable",
        "DnsSupport": "enable"
      },
      "TagSpecification": {
        "ResourceType": "transit-gateway",
        "tag": 1,
        "Tag": {
          "Value": "my-tgw",
          "tag": 1,
          "Key": "Name"
        }
      }
    }
  },
  "responseElements": {
    "CreateTransitGatewayResponse": {
      "xmlns": "http://ec2.amazonaws.com/doc/2016-11-15/",

```

```

    "requestId": "a07c1edf-c201-4e44-bfffb-3ce90EXAMPLE",
    "transitGateway": {
      "tagSet": {
        "item": {
          "value": "my-tgw",
          "key": "Name"
        }
      },
      "creationTime": "2018-11-15T05:25:50.000Z",
      "transitGatewayId": "tgw-0a13743bd6c1f5fcb",
      "options": {
        "propagationDefaultRouteTableId": "tgw-rtb-0123cd602be10b00a",
        "amazonSideAsn": 64512,
        "defaultRouteTablePropagation": "enable",
        "vpnEcmpSupport": "enable",
        "autoAcceptSharedAttachments": "disable",
        "defaultRouteTableAssociation": "enable",
        "dnsSupport": "enable",
        "associationDefaultRouteTableId": "tgw-rtb-0123cd602be10b00a"
      },
      "state": "pending",
      "ownerId": 123456789012
    }
  },
  "requestID": "a07c1edf-c201-4e44-bfffb-3ce90EXAMPLE",
  "eventID": "e8fa575f-4964-4ab9-8ca4-6b5b4EXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}

```

Identity and access management in AWS Transit Gateway

AWS uses security credentials to identify you and to grant you access to your AWS resources. You can use features of AWS Identity and Access Management (IAM) to allow other users, services, and applications to use your AWS resources fully or in a limited way, without sharing your security credentials.

By default, IAM users don't have permission to create, view, or modify AWS resources. To allow a user to access resources such as a transit gateway, and to perform tasks, you must create an IAM policy that grants the user permission to use the specific resources and API actions they'll need, then attach the policy to the group to which that user belongs. When you attach a policy to a user or group of users, it allows or denies the users permission to perform the specified tasks on the specified resources.

To work with a transit gateway, one of the following AWS managed policies might meet your needs:

- [AmazonEC2FullAccess](#)
- [AmazonEC2ReadOnlyAccess](#)
- [PowerUserAccess](#)
- [ReadOnlyAccess](#)

Example policies to manage transit gateways

The following are example IAM policies for working with transit gateways.

Create a transit gateway with required tags

The following example enables users to create transit gateway. The `aws:RequestTag` condition key requires users to tag the transit gateway with the tag `stack=prod`. The `aws:TagKeys` condition key uses the `ForAllValues` modifier to indicate that only the key `stack` is allowed in the request (no other tags can be specified). If users don't pass this specific tag when they create the transit gateway, or if they don't specify tags at all, the request fails.

The second statement uses the `ec2:CreateAction` condition key to allow users to create tags only in the context of `CreateTransitGateway`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateTaggedTGWs",
      "Effect": "Allow",
      "Action": "ec2:CreateTransitGateway",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:transit-gateway/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/stack": "prod"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": [
            "stack"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:123456789012:transit-gateway/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateTransitGateway"
        }
      }
    }
  ]
}
```

Working with transit gateway route tables

The following example enables users to create and delete transit gateway route tables for a specific transit gateway only (tgw-11223344556677889). Users can also create and replace routes

in any transit gateway route table, but only for attachments that have the tag `network=new-york-office`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteTransitGatewayRouteTable",
        "ec2:CreateTransitGatewayRouteTable"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:transit-gateway/tgw-11223344556677889",
        "arn:aws:ec2:*:*:transit-gateway-route-table/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTransitGatewayRoute",
        "ec2:ReplaceTransitGatewayRoute"
      ],
      "Resource": "arn:aws:ec2:*:*:transit-gateway-attachment/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/network": "new-york-office"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTransitGatewayRoute",
        "ec2:ReplaceTransitGatewayRoute"
      ],
      "Resource": "arn:aws:ec2:*:*:transit-gateway-route-table/*"
    }
  ]
}
```

```
}
```

Use service-linked roles for transit gateways in AWS Transit Gateway

Amazon VPC uses service-linked roles for the permissions that it requires to call other AWS services on your behalf. For more information, see [Service-linked roles](#) in the *IAM User Guide*.

Transit gateway service-linked role

Amazon VPC uses service-linked roles for the permissions that it requires to call other AWS services on your behalf when you work with a transit gateway.

Permissions granted by the service-linked role

Amazon VPC uses the service-linked role named **AWSServiceRoleForVPCTransitGateway** to call the following actions on your behalf when you work with a transit gateway:

- `ec2:CreateNetworkInterface`
- `ec2:DescribeNetworkInterfaces`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2>DeleteNetworkInterface`
- `ec2:CreateNetworkInterfacePermission`
- `ec2:AssignIpv6Addresses`
- `ec2:UnAssignIpv6Addresses`

The **AWSServiceRoleForVPCTransitGateway** role trusts the following services to assume the role:

- `transitgateway.amazonaws.com`

AWSServiceRoleForVPCTransitGateway uses the managed policy [AWSVPCTransitGatewayServiceRolePolicy](#).

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see [Service-linked role permissions](#) in the *IAM User Guide*.

Create the service-linked role

You don't need to manually create the **AWSServiceRoleForVPCTransitGateway** role. Amazon VPC creates this role for you when you attach a VPC in your account to a transit gateway.

Edit the service-linked role

You can edit the description of **AWSServiceRoleForVPCTransitGateway** using IAM. For more information, see [Edit a service-linked role description](#) in the *IAM User Guide*.

Delete the service-linked role

If you no longer need to use transit gateways, we recommend that you delete **AWSServiceRoleForVPCTransitGateway**.

You can delete this service-linked role only after you delete all transit gateway VPC attachments in your AWS account. This ensures that you can't inadvertently remove permission to access your VPC attachments.

You can use the IAM console, the IAM CLI, or the IAM API to delete service-linked roles. For more information, see [Delete a service-linked role](#) in the *IAM User Guide*.

After you delete **AWSServiceRoleForVPCTransitGateway**, Amazon VPC creates the role again if you attach a VPC in your account to a transit gateway.

AWS managed policies for transit gateways in AWS Transit Gateway

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*.

To work with a transit gateway, one of the following AWS managed policies might meet your needs:

- [AmazonEC2FullAccess](#)
- [AmazonEC2ReadOnlyAccess](#)
- [PowerUserAccess](#)
- [ReadOnlyAccess](#)

AWS managed policy: AWSVPCTransitGatewayServiceRolePolicy

This policy is attached to the role [AWSServiceRoleForVPCTransitGateway](#). This allows Amazon VPC to create and manage resources for your transit gateway attachments.

To view the permissions for this policy, see [AWSVPCTransitGatewayServiceRolePolicy](#) in the *AWS Managed Policy Reference*.

Transit gateway updates to AWS managed policies

View details about updates to AWS managed policies for transit gateways since Amazon VPC began tracking these changes in March 2021.

Change	Description	Date
Amazon VPC started tracking changes	Amazon VPC started tracking changes to its AWS managed policies.	March 1, 2021

Network ACLs for transit gateways in AWS Transit Gateway

A network access control list (NACL) is an optional layer of security.

Network access control list (NACL) rules are applied differently, depending on the scenario:

- [the section called "Same subnet for EC2 instances and transit gateway association"](#)
- [the section called "Different subnets for EC2 instances and transit gateway association"](#)

Same subnet for EC2 instances and transit gateway association

Consider a configuration where you have EC2 instances and a transit gateway association in the same subnet. The same network ACL is used for both the traffic from the EC2 instances to the transit gateway and traffic from the transit gateway to the instances.

NACL rules are applied as follows for traffic from instances to the transit gateway:

- Outbound rules use the destination IP address for evaluation.
- Inbound rules use the source IP address for evaluation.

NACL rules are applied as follows for traffic from the transit gateway to the instances:

- Outbound rules are not evaluated.
- Inbound rules are not evaluated.

Different subnets for EC2 instances and transit gateway association

Consider a configuration where you have EC2 instances in one subnet and a transit gateway association in a different subnet, and each subnet is associated with a different network ACL.

Network ACL rules are applied as follows for the EC2 instance subnet:

- Outbound rules use the destination IP address to evaluate traffic from the instances to the transit gateway.
- Inbound rules use the source IP address to evaluate traffic from the transit gateway to the instances.

NACL rules are applied as follows for the transit gateway subnet:

- Outbound rules use the destination IP address to evaluate traffic from the transit gateway to the instances.

- Outbound rules are not used to evaluate traffic from the instances to the transit gateway.
- Inbound rules use the source IP address to evaluate traffic from the instances to the transit gateway.
- Inbound rules are not used to evaluate traffic from the transit gateway to the instances.

Best Practices

Use a separate subnet for each transit gateway VPC attachment. For each subnet, use a small CIDR, for example /28, so that you have more addresses for EC2 resources. When you use a separate subnet, you can configure the following:

- Keep the inbound and outbound NACL that is associated with the transit gateway subnets open.
- Depending on your traffic flow, you can apply NACLs to your workload subnets.

For more information about how VPC attachments work, see [the section called “Resource attachments”](#).

AWS Transit Gateway Quotas

Your AWS account has the following quotas (previously referred to as *limits*) related to transit gateways. Unless otherwise noted, each quota is Region-specific.

The Service Quotas console provides information about the quotas for your account. You can use the Service Quotas console to view default quotas and [request quota increases](#) for adjustable quotas. For more information, see [Requesting a quota increase](#) in the *Service Quotas User Guide*.

If an adjustable quota is not yet available in Service Quotas, you can open a support case.

General

Name	Default	Adjustable
Transit gateways per account	5	Yes
CIDR blocks per transit gateway	5	No

The CIDR blocks are used in the [the section called “Connect attachments and Connect peers”](#) feature.

Routing

Name	Default	Adjustable
Transit gateway route tables per transit gateway	20	Yes
Transit gateway policy tables per transit gateway	20	No
Policy table entries per transit gateway (customer-managed entries, across all policy tables on a single transit gateway)	200	Yes

Name	Default	Adjustable
Total combined routes (dynamic and static) across all route tables for a single transit gateway	10,000	Contact your Solutions Architect (SA) or Technical Account Manager (TAM) for further assistance.
Dynamic routes advertised from a virtual router appliance to a Connect peer	1,000	Contact your Solutions Architect (SA) or Technical Account Manager (TAM) for further assistance.
Routes advertised from a Connect peer on a transit gateway to a virtual router appliance	5,000	No
Static routes for a prefix to a single attachment	1	No

Advertised routes come from the route table that's associated with the Connect attachment.

Transit gateway attachments

A transit gateway cannot have more than one VPC attachment to the same VPC.

Name	Default	Adjustable
Attachments per transit gateway	5,000	Yes
Transit gateways per VPC	5	No
Peering attachments per transit gateway	50	Yes
Pending peering attachments per transit gateway	10	Yes

Name	Default	Adjustable
Peering attachments between two transit gateways or between one transit gateway and a Cloud WAN core network edge (CNE)	1	No
Connect peers (GRE tunnels) per Connect attachment	4	No
VPN Concentrators per transit gateway	5	No
VPN connections per VPN Concentrator	100	No

Bandwidth

There are many factors that can affect realized bandwidth through a Site-to-Site VPN connection, including but not limited to: packet size, traffic mix (TCP/UDP), shaping or throttling policies on intermediate networks, internet weather, and specific application requirements. For VPC attachments, Direct Connect gateways, or peered transit gateway attachments, we will attempt to provide additional bandwidth beyond the default value.

Name	Default	Adjustable
Bandwidth per VPC attachment per Availability Zone	Up to 100 Gbps each direction (i.e., 100 Gbps ingress and 100 Gbps egress)	Contact your Solutions Architect (SA) or Technical Account Manager (TAM) for further assistance.
Packets per second per transit gateway VPC attachment per Availability Zone	Up to 7,500,000	Contact your Solutions Architect (SA) or Technical Account Manager (TAM) for further assistance.

Name	Default	Adjustable
Bandwidth for Direct Connect gateway or peered transit gateway connection per available Availability Zone in the Region	Up to 100 Gbps each direction (i.e., 100 Gbps ingress and 100 Gbps egress)	Contact your Solutions Architect (SA) or Technical Account Manager (TAM) for further assistance.
Packets per second per transit gateway attachment (Direct Connect and peering attachments) per available Availability Zone in the Region	Up to 7,500,000	Contact your Solutions Architect (SA) or Technical Account Manager (TAM) for further assistance.
Maximum bandwidth per Connect peer (GRE tunnel) per Connect attachment	Up to 5 Gbps	No
Maximum packets per second per Connect peer	Up to 300,000	No

You can use equal-cost multipath routing (ECMP) to get higher VPN bandwidth by aggregating multiple VPN tunnels. To use ECMP, the VPN connection must be configured for dynamic routing. ECMP is not supported on VPN connections that use static routing.

You can create up to 4 Connect peers per Connect attachment (up to 20 Gbps in total bandwidth per Connect attachment), as long as the underlying transport (VPC or Direct Connect) attachment supports the required bandwidth. You can use ECMP to get higher bandwidth by scaling horizontally across multiple Connect peers of the same Connect attachment or across multiple Connect attachments on the same transit gateway. The transit gateway cannot use ECMP between the BGP peerings of the same Connect peer.

For bandwidth and packet limits with VPN tunnel, please refer to [VPN bandwidth and throughput](#).

Direct Connect gateways

Name	Default	Adjustable
Direct Connect gateways per transit gateway	20	No
Transit gateways per Direct Connect gateway	6	No

Maximum transmission unit (MTU)

- The MTU of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The larger the MTU of a connection, the more data that can be passed in a single packet. A transit gateway supports an MTU of 8500 bytes for traffic between VPCs, Direct Connect, Transit Gateway Connect, and peering attachments (intra-Region, inter-Region, and Cloud WAN peering attachments). Traffic over VPN connections can have an MTU of 1500 bytes.
- When migrating from VPC peering to use a transit gateway, an MTU size mismatch between VPC peering and the transit gateway might result in some asymmetric traffic packets dropping. Update both VPCs at the same time to avoid jumbo packets dropping due to a size mismatch.
- The transit gateway enforces Maximum Segment Size (MSS) clamping for all packets. For more information, see [RFC879](#).
- For details about Site-to-Site VPN quotas for MTU, see [Maximum transmission unit \(MTU\)](#) in the *AWS Site-to-Site VPN User Guide*.
- Transit gateways support Path MTU Discovery (PMTUD) for traffic ingressing on VPC and Connect attachments. Transit gateway generates the FRAG_NEEDED for ICMPv4 packets and Packet Too Big (PTB) for ICMPv6 packets. Transit gateways does not support PMTUD on Site-to-site VPN, Direct Connect, and Peering attachments. For more information about Path MTU Discovery, see [Path MTU Discovery](#) in the *Amazon VPC User Guide*

Multicast

Note

Transit gateway multicast may not be suitable for high-frequency trading or performance-sensitive applications. We strongly recommend that you review the following multicast limits. Contact your account or Solution Architect team for a detailed review of your performance requirements.

Name	Default	Adjustable
Multicast domains per transit gateway	20	Contact your Solutions Architect (SA) or Technical Account Manager (TAM) for further assistance.
Multicast network interfaces per transit gateway	10,000	Contact your Solutions Architect (SA) or Technical Account Manager (TAM) for further assistance.
Multicast domain associations per VPC	20	Contact your Solutions Architect (SA) or Technical Account Manager (TAM) for further assistance.
Static and IGMPv2 multicast group members and sources per transit gateway	10,000	No

Name	Default	Adjustable
Static and IGMPv2 multicast group members per transit gateway multicast group	100	No
Maximum multicast throughput per flow	1 Gbps	No
Maximum aggregate multicast throughput per Availability Zone	20 Gbps	No
Maximum packets per second per flow (less than 10 receivers)	75,000	No
Maximum packets per second per flow (greater than 10 receivers)	15,000	No
Maximum aggregate packets per second (less than 10 receivers)	2,500,000	No
Maximum aggregate packets per second (greater than 10 receivers)	500,000	No

AWS Network Manager

Name	Default	Adjustable
Global networks per AWS account	5	Yes
Devices per global network	200	Yes
Links per global network	200	Yes
Sites per global network	200	Yes
Connections per global network	500	No

Additional quota resources

For more information, see the following:

- [Site-to-Site VPN quotas](#) in the *AWS Site-to-Site VPN User Guide*
- [Amazon VPC quotas](#) in the *Amazon VPC User Guide*
- [Direct Connect quotas](#) in the *AWS Direct Connect User Guide*

Document history for transit gateways

The following table describes the releases for transit gateways.

Change	Description	Date
Policy-Based Routing for transit gateways	Use Policy-Based Routing (PBR) and policy tables to forward traffic across a transit gateway based on packet attributes such as source and destination IP addresses, ports, and protocol.	July 28, 2026
Client VPN attachments	Create a Client VPN attachment to connect a transit gateway to a Client VPN endpoint.	April 20, 2026
Flexible Cost Allocation	Configure flexible cost allocation policies to control how data processing and transfer costs are allocated across your organization.	November 20, 2025
Encryption Support for transit gateways	Managing Encryption Support on transit gateways to enforce encryption-in-transit for all traffic.	November 20, 2025
Network function attachments	Create a network function attachment to connect a transit gateway directly to AWS Network Firewall.	June 16, 2025

Security group referencing support	You can now reference a security group across VPCs attached to a transit gateway.	September 25, 2024
AWS Transit Gateway Quotas	Bandwidth limits were added.	August 14, 2023
AWS Transit Gateway Flow Logs	Transit Gateways now support Transit Gateway Flow Logs, allowing you to monitor and log network traffic between transit gateways.	July 14, 2022
Transit gateway policy tables	Use policy tables to set up dynamic routing for transit gateways for automatically exchanging routing and reachability information with peered transit gateway types.	July 13, 2022
Network Manager User Guide	Network Manager was created as a standalone guide, and is no longer included as part of the <i>AWS Transit Gateway User Guide</i> .	December 2, 2021
Peering attachments	You can create a peering connection with a transit gateway in the same Region.	December 1, 2021
Transit Gateway Connect	You can establish a connection between a transit gateway and third-party virtual appliances running in a VPC.	December 10, 2020

Appliance mode	You can enable appliance mode on a VPC attachment to ensure that bidirectional traffic flows through the same Availability Zone for the attachment.	October 29, 2020
Prefix list references	You can reference a prefix list in your transit gateway route table.	August 24, 2020
Modify transit gateway	You can modify the configuration options for your transit gateway.	August 24, 2020
CloudWatch metrics for transit gateway attachments	You can view CloudWatch metrics for individual transit gateway attachments.	July 6, 2020
Network Manager Route Analyzer	You can analyze the routes in your transit gateway route tables in your global network.	May 4, 2020
Peering attachments	You can create a peering connection with a transit gateway in another Region.	December 3, 2019
Multicast support	Transit Gateway supports routing multicast traffic between subnets of attached VPCs and serves as a multicast router for instances sending traffic destined for multiple receiving instances.	December 3, 2019
AWS Network Manager	You can visualize and monitor your global networks that are built around transit gateways.	December 3, 2019

[AWS Direct Connect support](#)

You can use an Direct Connect gateway to connect your Direct Connect connection over a transit virtual interface to the VPCs or VPNs attached to your transit gateway.

[Initial release](#)

This release introduces transit gateways.