



User Guide

AWS Site-to-Site VPN



AWS Site-to-Site VPN: User Guide

Copyright © 2024 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 Site-to-Site VPN	1
Concepts	1
Site-to-Site VPN features	2
Site-to-Site VPN limitations	2
Working with Site-to-Site VPN	3
Pricing	3
How AWS Site-to-Site VPN works	4
Virtual private gateway	4
Transit gateway	5
Customer gateway device	6
Customer gateway	6
VPN tunnel options	7
VPN tunnel authentication options	13
Pre-shared keys	13
Private certificate from AWS Private Certificate Authority	13
VPN tunnel initiation options	14
VPN tunnel IKE initiation options	14
Rules and limitations	14
Working with VPN tunnel initiation options	15
Endpoint replacements	15
Customer initiated endpoint replacements	16
AWS managed endpoint replacements	16
Tunnel endpoint lifecycle	17
Customer gateway options	22
Accelerated VPN connections	24
Enabling acceleration	25
Rules and restrictions	25
Site-to-Site VPN routing options	26
Static and dynamic routing	26
Route tables and VPN route priority	27
Routing during VPN tunnel endpoint updates	29
IPv4 and IPv6 traffic	30
Getting started tutorial	31
Prerequisites	31

Create a customer gateway	33
Create a target gateway	34
Create a virtual private gateway	34
Create a transit gateway	35
Configure routing	35
(Virtual private gateway) Enable route propagation in your route table	35
(Transit gateway) Add a route to your route table	37
Update your security group	37
Create a VPN connection	38
Download the configuration file	39
Configure the customer gateway device	40
Architectures	41
Single and multiple VPN connections	41
Single Site-to-Site VPN connection	41
Single Site-to-Site VPN connection with a transit gateway	42
Multiple Site-to-Site VPN connections	43
Multiple Site-to-Site VPN connections with a transit gateway	43
Site-to-Site VPN connection with AWS Direct Connect	44
Private IP Site-to-Site VPN connection with AWS Direct Connect	45
AWS VPN CloudHub	46
Overview	46
Pricing	47
Redundant VPN connections	48
Your customer gateway device	50
Example configuration files	51
Requirements for your customer gateway device	53
Best practices for your customer gateway device	56
Firewall Rules	58
Multiple VPN connection scenarios	60
Routing for your customer gateway device	61
Example configurations for static routing	62
Example configuration files	62
User interface procedures for static routing	63
Additional information for Cisco devices	75
Testing	75
Example configurations for dynamic routing (BGP)	76

Example configuration files	76
User interface procedures for dynamic routing	78
Additional information for Cisco devices	87
Additional information for Juniper devices	87
Testing	88
Windows Server as a customer gateway device	88
Configuring your Windows instance	88
Step 1: Create a VPN connection and configure your VPC	89
Step 2: Download the configuration file for the VPN connection	90
Step 3: Configure the Windows Server	93
Step 4: Set up the VPN tunnel	94
Step 5: Enable dead gateway detection	101
Step 6: Test the VPN connection	101
Troubleshooting	102
Device with BGP	103
Device without BGP	106
Cisco ASA	109
Cisco IOS	113
Cisco IOS without BGP	119
Juniper JunOS	125
Juniper ScreenOS	129
Yamaha	133
Work with Site-to-Site VPN	138
Create a VPN attachment for AWS Cloud WAN	138
Create a transit gateway VPN attachment	140
Test a VPN connection	141
Delete a VPN connection	143
Delete a VPN connection	144
Delete a customer gateway	144
Detach and delete a virtual private gateway	145
Modify the target gateway of a VPN connection	146
Step 1: Create the new target gateway	146
Step 2: Delete your static routes (conditional)	147
Step 3: Migrate to a new gateway	147
Step 4: Update VPC route tables	148
Step 5: Update the target gateway routing (conditional)	149

Step 6: Update the customer gateway ASN (conditional)	150
Modify VPN connection options	150
Modify VPN tunnel options	151
Edit static routes for a VPN connection	151
Change the customer gateway for a VPN connection	152
Replace compromised credentials	153
Rotate VPN tunnel endpoint certificates	154
Private IP VPN with AWS Direct Connect	154
Benefits of private IP VPN	155
How private IP VPN works	156
Prerequisites	156
Create the customer gateway	157
Prepare the transit gateway	157
Create the AWS Direct Connect gateway	158
Create the transit gateway association	158
Create the VPN connection	159
Security	161
Data protection	161
Internetwork traffic privacy	162
Identity and access management	163
Audience	164
Authenticating with identities	164
Managing access using policies	168
How AWS Site-to-Site VPN works with IAM	170
Identity-based policy examples	177
Troubleshooting	180
Using service-linked roles	182
Resilience	184
Two tunnels per VPN connection	185
Redundancy	185
Infrastructure security	185
Monitoring your Site-to-Site VPN connection	187
Monitoring tools	188
Automated monitoring tools	188
Manual monitoring tools	188
AWS Site-to-Site VPN logs	189

Benefits of Site-to-Site VPN logs	190
Amazon CloudWatch Logs resource policy size restrictions	190
Contents of Site-to-Site VPN logs	191
IAM requirements to publish to CloudWatch Logs	194
View Site-to-Site VPN logs configuration	195
Enable Site-to-Site VPN logs	196
Disable Site-to-Site VPN logs	197
Monitoring VPN tunnels using Amazon CloudWatch	197
VPN metrics and dimensions	198
Viewing VPN CloudWatch metrics	199
Creating CloudWatch alarms to monitor VPN tunnels	200
Monitoring VPN connections using AWS Health events	203
Tunnel endpoint replacement notifications	203
Single tunnel VPN notifications	203
Quotas	204
Site-to-Site VPN resources	204
Routes	205
Bandwidth and throughput	206
Maximum transmission unit (MTU)	206
Additional quota resources	206
Document history	207

What is AWS Site-to-Site VPN?

By default, instances that you launch into an Amazon VPC can't communicate with your own (remote) network. You can enable access to your remote network from your VPC by creating an AWS Site-to-Site VPN (Site-to-Site VPN) connection, and configuring routing to pass traffic through the connection.

Although the term *VPN connection* is a general term, in this documentation, a VPN connection refers to the connection between your VPC and your own on-premises network. Site-to-Site VPN supports Internet Protocol security (IPsec) VPN connections.

Contents

- [Concepts](#)
- [Site-to-Site VPN features](#)
- [Site-to-Site VPN limitations](#)
- [Working with Site-to-Site VPN](#)
- [Pricing](#)

Concepts

The following are the key concepts for Site-to-Site VPN:

- **VPN connection:** A secure connection between your on-premises equipment and your VPCs.
- **VPN tunnel:** An encrypted link where data can pass from the customer network to or from AWS.

Each VPN connection includes two VPN tunnels which you can simultaneously use for high availability.

- **Customer gateway:** An AWS resource which provides information to AWS about your customer gateway device.
- **Customer gateway device:** A physical device or software application on your side of the Site-to-Site VPN connection.
- **Target gateway:** A generic term for the VPN endpoint on the Amazon side of the Site-to-Site VPN connection.
- **Virtual private gateway:** A virtual private gateway is the VPN endpoint on the Amazon side of your Site-to-Site VPN connection that can be attached to a single VPC.

- **Transit gateway:** A transit hub that can be used to interconnect multiple VPCs and on-premises networks, and as a VPN endpoint for the Amazon side of the Site-to-Site VPN connection.

Site-to-Site VPN features

The following features are supported on AWS Site-to-Site VPN connections:

- Internet Key Exchange version 2 (IKEv2)
- NAT traversal
- 4-byte ASN in the range of 1 – 2147483647 for Virtual Private Gateway (VGW) configuration. See [Customer gateway options for your Site-to-Site VPN connection](#) for more information.
- 2-byte ASN for Customer Gateway (CGW) in the range of 1 – 65535. See [Customer gateway options for your Site-to-Site VPN connection](#) for more information.
- CloudWatch metrics
- Reusable IP addresses for your customer gateways
- Additional encryption options; including AES 256-bit encryption, SHA-2 hashing, and additional Diffie-Hellman groups
- Configurable tunnel options
- Custom private ASN for the Amazon side of a BGP session
- Private Certificate from a subordinate CA from AWS Private Certificate Authority
- Support for IPv6 traffic for VPN connections on a transit gateway

Site-to-Site VPN limitations

A Site-to-Site VPN connection has the following limitations.

- IPv6 traffic is not supported for VPN connections on a virtual private gateway.
- An AWS VPN connection does not support Path MTU Discovery.

In addition, take the following into consideration when you use Site-to-Site VPN.

- When connecting your VPCs to a common on-premises network, we recommend that you use non-overlapping CIDR blocks for your networks.

Working with Site-to-Site VPN

You can create, access, and manage your Site-to-Site VPN resources using any of the following interfaces:

- **AWS Management Console**— Provides a web interface that you can use to access your Site-to-Site VPN resources.
- **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** — Provide language-specific APIs and takes care of many of the connection details, such as calculating signatures, handling request retries, and error handling. 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 error handling. For more information, see the [Amazon EC2 API Reference](#).

Pricing

You are charged for each VPN connection hour that your VPN connection is provisioned and available. For more information, see [AWS Site-to-Site VPN and Accelerated Site-to-Site VPN Connection pricing](#).

You are charged for data transfer out from Amazon EC2 to the internet. For more information, see [Data Transfer](#) on the Amazon EC2 On-Demand Pricing page.

When you create an accelerated VPN connection, we create and manage two accelerators on your behalf. You are charged an hourly rate and data transfer costs for each accelerator. For more information, see [AWS Global Accelerator pricing](#).

How AWS Site-to-Site VPN works

A Site-to-Site VPN connection consists of the following components:

- A [virtual private gateway](#) or a [transit gateway](#)
- A [customer gateway device](#)
- A [customer gateway](#)

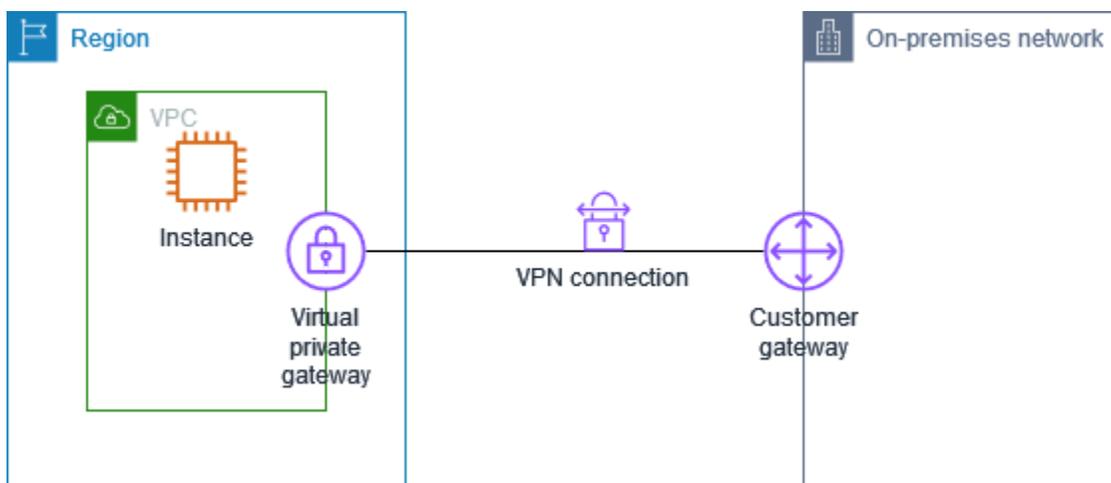
The VPN connection offers two VPN tunnels between a virtual private gateway or transit gateway on the AWS side, and a customer gateway on the on-premises side.

For more information about Site-to-Site VPN quotas, see [Site-to-Site VPN quotas](#).

Virtual private gateway

A *virtual private gateway* is the VPN concentrator on the Amazon side of the Site-to-Site VPN connection. You create a virtual private gateway and attach it to a virtual private cloud (VPC) with resources that must access the Site-to-Site VPN connection.

The following diagram shows a VPN connection between a VPC and your on-premises network using a virtual private gateway.



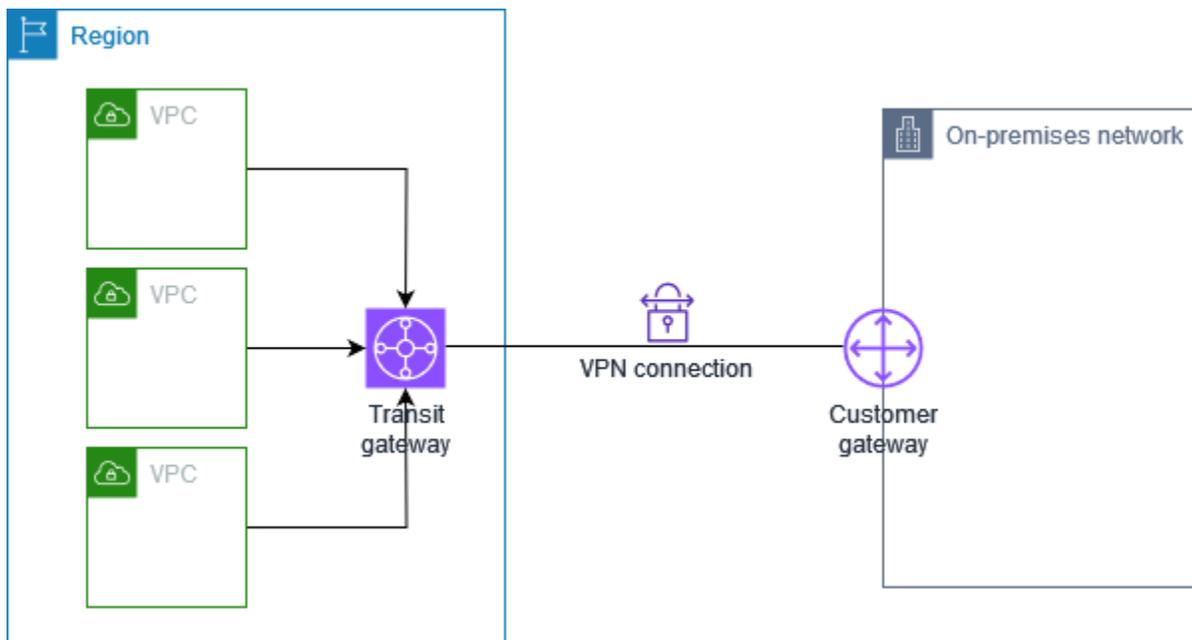
When you create a virtual private gateway, you can specify the private Autonomous System Number (ASN) for the Amazon side of the gateway. If you don't specify an ASN, the virtual private gateway is created with the default ASN (64512). You cannot change the ASN after you've created

the virtual private gateway. To check the ASN for your virtual private gateway, view its details in the **Virtual private gateways** page in the Amazon VPC console, or use the [describe-vpn-gateways](#) AWS CLI command.

Transit gateway

A transit gateway is a transit hub that you can use to interconnect your VPCs and your on-premises networks. For more information, see [Amazon VPC Transit Gateways](#). You can create a Site-to-Site VPN connection as an attachment on a transit gateway.

The following diagram shows a VPN connection between multiple VPCs and your on-premises network using a transit gateway. The transit gateway has three VPC attachments and a VPN attachment.



Your Site-to-Site VPN connection on a transit gateway can support either IPv4 traffic or IPv6 traffic inside the VPN tunnels. For more information, see [IPv4 and IPv6 traffic](#).

You can modify the target gateway of a Site-to-Site VPN connection from a virtual private gateway to a transit gateway. For more information, see [the section called “Modify the target gateway of a VPN connection”](#).

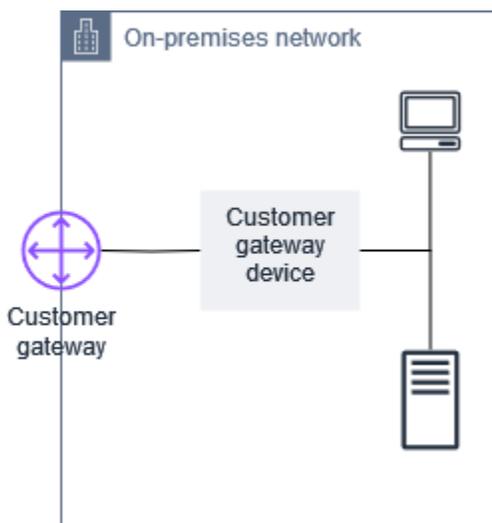
Customer gateway device

A *customer gateway device* is a physical device or software application on your side of the Site-to-Site VPN connection. You configure the device to work with the Site-to-Site VPN connection. For more information, see [Your customer gateway device](#).

By default, your customer gateway device must bring up the tunnels for your Site-to-Site VPN connection by generating traffic and initiating the Internet Key Exchange (IKE) negotiation process. You can configure your Site-to-Site VPN connection to specify that AWS must initiate the IKE negotiation process instead. For more information, see [Site-to-Site VPN tunnel initiation options](#).

Customer gateway

A *customer gateway* is a resource that you create in AWS that represents the customer gateway device in your on-premises network. When you create a customer gateway, you provide information about your device to AWS. For more information, see [the section called "Customer gateway options"](#).

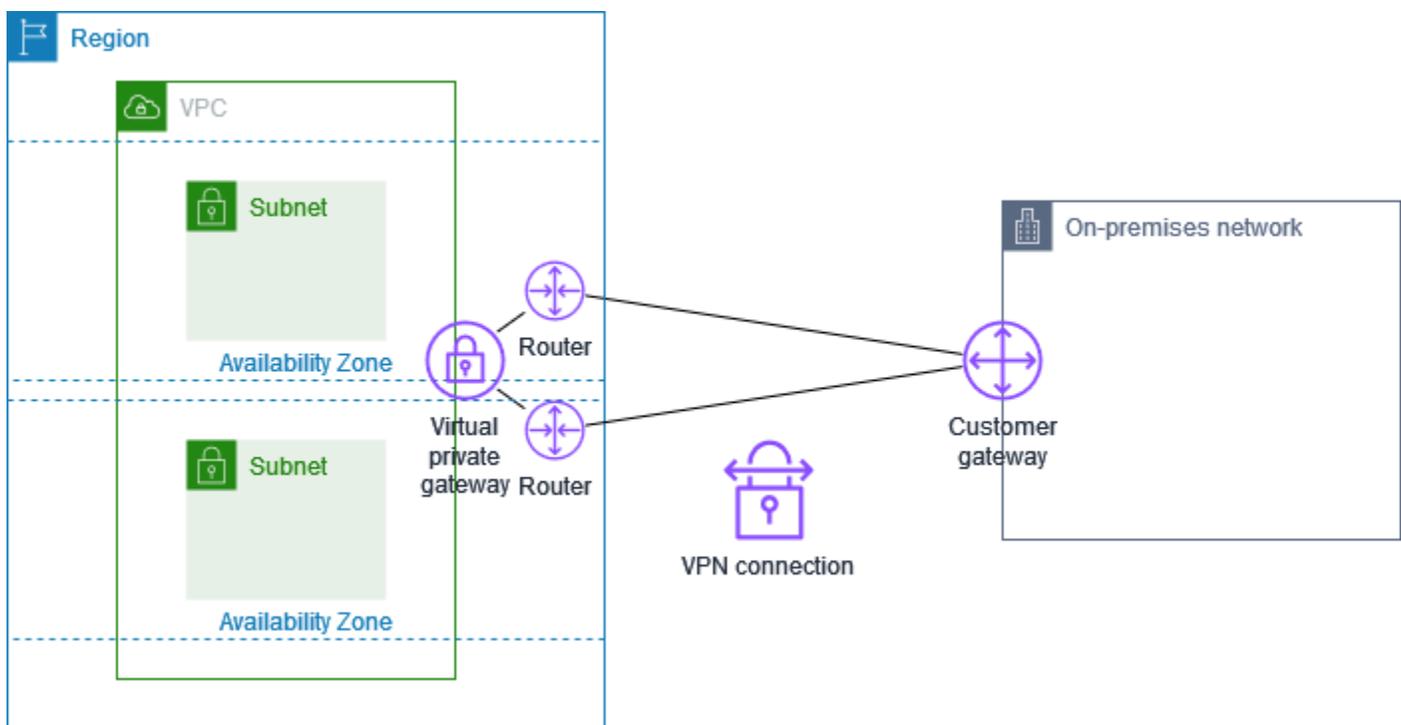


To use Amazon VPC with a Site-to-Site VPN connection, you or your network administrator must also configure the customer gateway device or application in your remote network. When you create the Site-to-Site VPN connection, we provide you with the required configuration information and your network administrator typically performs this configuration. For information about the customer gateway requirements and configuration, see [Your customer gateway device](#).

Tunnel options for your Site-to-Site VPN connection

You use a Site-to-Site VPN connection to connect your remote network to a VPC. Each Site-to-Site VPN connection has two tunnels, with each tunnel using a unique public IP address. It is important to configure both tunnels for redundancy. When one tunnel becomes unavailable (for example, down for maintenance), network traffic is automatically routed to the available tunnel for that specific Site-to-Site VPN connection.

The following diagram shows the two tunnels of a VPN connection. Each tunnel terminates in a different Availability Zone to provide increased availability. Traffic from the on-premises network to AWS uses both tunnels. Traffic from AWS to the on-premises network prefers one of the tunnels, but can automatically fail over to the other tunnel if there is a failure on the AWS side.



When you create a Site-to-Site VPN connection, you download a configuration file specific to your customer gateway device that contains information for configuring the device, including information for configuring each tunnel. You can optionally specify some of the tunnel options yourself when you create the Site-to-Site VPN connection. Otherwise, AWS provides default values.

Note

Site-to-Site VPN tunnel endpoints evaluate proposals from your customer gateway starting with the lowest configured value from the list below, regardless of the proposal order from

the customer gateway. You can use the `modify-vpn-connection-options` command to restrict the list of options AWS endpoints will accept. For more information, see [modify-vpn-connection-options](#) in *Amazon EC2 Command Line Reference*.

The following are the tunnel options that you can configure.

Dead peer detection (DPD) timeout

The number of seconds after which a DPD timeout occurs. A DPD timeout of 40 seconds means that the VPN endpoint will consider the peer dead 30 seconds after the first failed keep-alive. You can specify 30 or higher.

Default: 40

DPD timeout action

The action to take after dead peer detection (DPD) timeout occurs. You can specify the following:

- `Clear`: End the IKE session when DPD timeout occurs (stop the tunnel and clear the routes)
- `None`: Take no action when DPD timeout occurs
- `Restart`: Restart the IKE session when DPD timeout occurs

For more information, see [Site-to-Site VPN tunnel initiation options](#).

Default: `Clear`

VPN logging options

With Site-to-Site VPN logs, you can gain access to details on IP Security (IPsec) tunnel establishment, Internet Key Exchange (IKE) negotiations, and dead peer detection (DPD) protocol messages.

For more information, see [AWS Site-to-Site VPN logs](#).

Available log formats: `json`, `text`

IKE versions

The IKE versions that are permitted for the VPN tunnel. You can specify one or more of the default values.

Default: `ikev1`, `ikev2`

Inside tunnel IPv4 CIDR

The range of inside (internal) IPv4 addresses for the VPN tunnel. You can specify a size /30 CIDR block from the 169.254.0.0/16 range. The CIDR block must be unique across all Site-to-Site VPN connections that use the same virtual private gateway.

Note

The CIDR block does not need to be unique across all connections on a transit gateway. However, if they are not unique, it can create a conflict on your customer gateway. Proceed carefully when re-using the same CIDR block on multiple Site-to-Site VPN connections on a transit gateway.

The following CIDR blocks are reserved and cannot be used:

- 169.254.0.0/30
- 169.254.1.0/30
- 169.254.2.0/30
- 169.254.3.0/30
- 169.254.4.0/30
- 169.254.5.0/30
- 169.254.169.252/30

Default: A size /30 IPv4 CIDR block from the 169.254.0.0/16 range.

Inside tunnel IPv6 CIDR

(IPv6 VPN connections only) The range of inside (internal) IPv6 addresses for the VPN tunnel. You can specify a size /126 CIDR block from the local fd00::/8 range. The CIDR block must be unique across all Site-to-Site VPN connections that use the same transit gateway.

Default: A size /126 IPv6 CIDR block from the local fd00::/8 range.

Local IPv4 Network CIDR

(IPv4 VPN connection only) The IPv4 CIDR range on the customer gateway (on-premises) side that is allowed to communicate over the VPN tunnels.

Default: 0.0.0.0/0

Remote IPv4 Network CIDR

(IPv4 VPN connection only) The IPv4 CIDR range on the AWS side that is allowed to communicate over the VPN tunnels.

Default: 0.0.0.0/0

Local IPv6 Network CIDR

(IPv6 VPN connection only) The IPv6 CIDR range on the customer gateway (on-premises) side that is allowed to communicate over the VPN tunnels.

Default: ::/0

Remote IPv6 Network CIDR

(IPv6 VPN connection only) The IPv6 CIDR range on the AWS side that is allowed to communicate over the VPN tunnels.

Default: ::/0

Phase 1 Diffie-Hellman (DH) group numbers

The DH group numbers that are permitted for the VPN tunnel for phase 1 of the IKE negotiations. You can specify one or more of the default values.

Default: 2, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24

Phase 2 Diffie-Hellman (DH) group numbers

The DH group numbers that are permitted for the VPN tunnel for phase 2 of the IKE negotiations. You can specify one or more of the default values.

Default: 2, 5, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24

Phase 1 encryption algorithms

The encryption algorithms that are permitted for the VPN tunnel for phase 1 of the IKE negotiations. You can specify one or more of the default values.

Default: AES128, AES256, AES128-GCM-16, AES256-GCM-16

Phase 2 encryption algorithms

The encryption algorithms that are permitted for the VPN tunnel for phase 2 IKE negotiations. You can specify one or more of the default values.

Default: AES128, AES256, AES128-GCM-16, AES256-GCM-16

Phase 1 integrity algorithms

The integrity algorithms that are permitted for the VPN tunnel for phase 1 of the IKE negotiations. You can specify one or more of the default values.

Default: SHA1, SHA2-256, SHA2-384, SHA2-512

Phase 2 integrity algorithms

The integrity algorithms that are permitted for the VPN tunnel for phase 2 of the IKE negotiations. You can specify one or more of the default values.

Default: SHA1, SHA2-256, SHA2-384, SHA2-512

Phase 1 lifetime

Note

AWS initiate re-keys with the timing values set in the Phase 1 lifetime and Phase 2 lifetime fields. If such lifetimes are different than the negotiated handshake values, this may interrupt tunnel connectivity.

The lifetime in seconds for phase 1 of the IKE negotiations. You can specify a number between 900 and 28,800.

Default: 28,800 (8 hours)

Phase 2 lifetime

Note

AWS initiate re-keys with the timing values set in the Phase 1 lifetime and Phase 2 lifetime fields. If such lifetimes are different than the negotiated handshake values, this may interrupt tunnel connectivity.

The lifetime in seconds for phase 2 of the IKE negotiations. You can specify a number between 900 and 3,600. The number that you specify must be less than the number of seconds for the phase 1 lifetime.

Default: 3,600 (1 hour)

Pre-shared key (PSK)

The pre-shared key (PSK) to establish the initial internet key exchange (IKE) security association between the target gateway and customer gateway.

The PSK must be between 8 and 64 characters in length and cannot start with zero (0). Allowed characters are alphanumeric characters, periods (.), and underscores (_).

Default: A 32-character alphanumeric string.

Rekey fuzz

The percentage of the rekey window (determined by the rekey margin time) within which the rekey time is randomly selected.

You can specify a percentage value between 0 and 100.

Default: 100

Rekey margin time

The margin time in seconds before the phase 1 and phase 2 lifetime expires, during which the AWS side of the VPN connection performs an IKE rekey.

You can specify a number between 60 and half of the value of the phase 2 lifetime.

The exact time of the rekey is randomly selected based on the value for rekey fuzz.

Default: 270 (4.5 minutes)

Replay window size packets

The number of packets in an IKE replay window.

You can specify a value between 64 and 2048.

Default: 1024

Startup action

The action to take when establishing the tunnel for a VPN connection. You can specify the following:

- **Start:** AWS initiates the IKE negotiation to bring the tunnel up. Only supported if your customer gateway is configured with an IP address.
- **Add:** Your customer gateway device must initiate the IKE negotiation to bring the tunnel up.

For more information, see [Site-to-Site VPN tunnel initiation options](#).

Default: Add

Tunnel endpoint lifecycle control

Tunnel endpoint lifecycle control provides control over the schedule of endpoint replacements.

For more information, see [Tunnel endpoint lifecycle control](#).

Default: Off

You can specify the tunnel options when you create a Site-to-Site VPN connection, or you can modify the tunnel options for an existing VPN connection. For more information, see the following topics:

- [Step 5: Create a VPN connection](#)
- [Modify Site-to-Site VPN tunnel options](#)

Site-to-Site VPN tunnel authentication options

You can use pre-shared keys, or certificates to authenticate your Site-to-Site VPN tunnel endpoints.

Pre-shared keys

A pre-shared key is the default authentication option.

A pre-shared key is a Site-to-Site VPN tunnel option that you can specify when you create a Site-to-Site VPN tunnel.

A pre-shared key is a string that you enter when you configure your customer gateway device. If you do not specify a string, we auto-generate one for you. For more information, see [Your customer gateway device](#).

Private certificate from AWS Private Certificate Authority

If you do not want to use pre-shared keys, you can use a private certificate from AWS Private Certificate Authority to authenticate your VPN.

You must create a private certificate from a subordinate CA using AWS Private Certificate Authority (AWS Private CA). To sign the ACM subordinate CA, you can use an ACM Root CA or an external CA.

For more information about creating a private certificate, see [Creating and Managing a Private CA](#) in the *AWS Private Certificate Authority User Guide*.

You must create a service-linked role to generate and use the certificate for the AWS side of the Site-to-Site VPN tunnel endpoint. For more information, see [the section called "Service-linked roles"](#).

After you generate the private certificate, you specify the certificate when you create the customer gateway, and then apply it to your customer gateway device.

If you do not specify the IP address of your customer gateway device, we do not check the IP address. This operation allows you to move the customer gateway device to a different IP address without having to re-configure the VPN connection.

Site-to-Site VPN tunnel initiation options

By default, your customer gateway device must bring up the tunnels for your Site-to-Site VPN connection by generating traffic and initiating the Internet Key Exchange (IKE) negotiation process. You can configure your VPN tunnels to specify that AWS must initiate or restart the IKE negotiation process instead.

VPN tunnel IKE initiation options

The following IKE initiation options are available. You can implement either or both options, for one or both of the tunnels in your Site-to-Site VPN connection. See [VPN tunnel options](#) for more details on these and other tunnel option settings.

- **Startup action:** The action to take when establishing the VPN tunnel for a new or modified VPN connection. By default, your customer gateway device initiates the IKE negotiation process to bring the tunnel up. You can specify that AWS must initiate the IKE negotiation process instead.
- **DPD timeout action:** The action to take after dead peer detection (DPD) timeout occurs. By default, the IKE session is stopped, the tunnel goes down, and the routes are removed. You can specify that AWS must restart the IKE session when DPD timeout occurs, or you can specify that AWS must take no action when DPD timeout occurs.

Rules and limitations

The following rules and limitations apply:

- To initiate IKE negotiation, AWS requires the public IP address of your customer gateway device. If you configured certificate-based authentication for your VPN connection and you did not specify an IP address when you created the customer gateway resource in AWS, you must create a new customer gateway and specify the IP address. Then, modify the VPN connection and specify the new customer gateway. For more information, see [Change the customer gateway for a Site-to-Site VPN connection](#).
- IKE initiation (startup action) from the AWS side of the VPN connection is supported for IKEv2 only.
- If using IKE initiation from the AWS side of the VPN connection, it does not include a timeout setting. It will continuously try to establish a connection until one is made. Additionally, the AWS side of VPN connection will re-initiate IKE negotiation when it receives a delete SA message from your customer gateway.
- If your customer gateway device is behind a firewall or other device using Network Address Translation (NAT), it must have an identity (IDr) configured. For more information about IDr, see [RFC 7296](#).

If you do not configure IKE initiation from the AWS side for your VPN tunnel and the VPN connection experiences a period of idle time (usually 10 seconds, depending on your configuration), the tunnel might go down. To prevent this, you can use a network monitoring tool to generate keepalive pings.

Working with VPN tunnel initiation options

For more information about working with VPN tunnel initiation options, see the following topics:

- To create a new VPN connection and specify the VPN tunnel initiation options: [Step 5: Create a VPN connection](#)
- To modify the VPN tunnel initiation options for an existing VPN connection: [Modify Site-to-Site VPN tunnel options](#)

Site-to-Site VPN tunnel endpoint replacements

Your Site-to-Site VPN connection consists of two VPN tunnels for redundancy. Sometimes, one or both of the VPN tunnel endpoints is replaced when AWS performs tunnel updates, or when you modify your VPN connection. During a tunnel endpoint replacement, connectivity over the tunnel might be interrupted while the new tunnel endpoint is provisioned.

Topics

- [Customer initiated endpoint replacements](#)
- [AWS managed endpoint replacements](#)
- [Tunnel endpoint lifecycle control](#)

Customer initiated endpoint replacements

When you modify the following components of your VPN connection, one or both of your tunnel endpoints is replaced.

Modification	API action	Tunnel impact
Modify the target gateway for the VPN connection	ModifyVpnConnection	Both tunnels are unavailable while new tunnel endpoints are provisioned.
Change the customer gateway for the VPN connection	ModifyVpnConnection	Both tunnels are unavailable while new tunnel endpoints are provisioned.
Modify the VPN connection options	ModifyVpnConnectionOptions	Both tunnels are unavailable while new tunnel endpoints are provisioned.
Modify the VPN tunnel options	ModifyVpnTunnelOptions	The modified tunnel is unavailable during the update.

AWS managed endpoint replacements

AWS Site-to-Site VPN is a managed service, and periodically applies updates to your VPN tunnel endpoints. These updates happen for a variety of reasons, including the following:

- To apply general upgrades, such as patches, resiliency improvements, and other enhancements
- To retire underlying hardware
- When automated monitoring determines that a VPN tunnel endpoint is unhealthy

AWS applies tunnel endpoint updates to one tunnel of your VPN connection at a time. During a tunnel endpoint update, your VPN connection might experience a brief loss of redundancy. It's therefore important to configure both tunnels in your VPN connection for high availability.

Tunnel endpoint lifecycle control

Tunnel endpoint lifecycle control provides control over the schedule of endpoint replacements, and can help minimize connectivity disruptions during AWS managed tunnel endpoint replacements. With this feature, you can choose to accept AWS managed updates to tunnel endpoints at a time that works best for your business. Use this feature if you have short-term business needs or can only support a single tunnel per VPN connection.

Note

In rare circumstances, AWS might apply critical updates to tunnel endpoints immediately, even if the tunnel endpoint lifecycle control feature is enabled.

Topics

- [How tunnel endpoint lifecycle control works](#)
- [Enable tunnel endpoint lifecycle control](#)
- [Verify if tunnel endpoint lifecycle control is enabled](#)
- [Check for available updates](#)
- [Accept a maintenance update](#)
- [Turn tunnel endpoint lifecycle control off](#)

How tunnel endpoint lifecycle control works

Turn on the tunnel endpoint lifecycle control feature for individual tunnels within a VPN connection. It can be enabled at the time of VPN creation or by modifying tunnel options for an existing VPN connection.

After tunnel endpoint lifecycle control is enabled, you will gain additional visibility into upcoming tunnel maintenance events in two ways:

- You will receive AWS Health notifications for upcoming tunnel endpoint replacements.

- The status of pending maintenance, along with the **Maintenance auto applied after** and **Last maintenance applied** timestamps, can be seen in the AWS Management Console or by using the [get-vpn-tunnel-replacement-status](#) AWS CLI command.

When a tunnel endpoint maintenance is available, you will have the opportunity to accept the update at a time that is convenient for you, before the given **Maintenance auto applied after** timestamp.

If you do not apply updates before the **Maintenance auto applied after** date, AWS will automatically perform the tunnel endpoint replacement soon after, as part of the regular maintenance update cycle.

Enable tunnel endpoint lifecycle control

You can enable this feature using the AWS Management Console or AWS CLI.

Note

By default when you turn on the feature for an existing VPN connection, a tunnel endpoint replacement will be initiated at the same time. If you want to turn the feature on, but not initiate a tunnel endpoint replacement immediately, you can use the **skip tunnel replacement** option.

Existing VPN connection

The following steps demonstrate how to enable tunnel endpoint lifecycle control on an existing VPN connection.

To enable tunnel endpoint lifecycle control using the AWS Management Console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the left-side navigation pane, choose **Site-to-Site VPN Connections**.
3. Select the appropriate connection under **VPN connections**.
4. Choose **Actions**, then **Modify VPN tunnel options**.
5. Select the specific tunnel that you want to modify by choosing the appropriate **VPN tunnel outside IP address**.
6. Under **Tunnel Endpoint Lifecycle Control**, select the **Enable** check box.

7. (Optional) Select **Skip tunnel replacement**.
8. Choose **Save changes**.

To enable tunnel endpoint lifecycle control using the AWS CLI

Use the [modify-vpn-tunnel-options](#) command to turn on tunnel endpoint lifecycle control.

New VPN connection

The following steps demonstrate how to enable tunnel endpoint lifecycle control during creation of a new VPN connection.

To enable tunnel endpoint lifecycle control during creation of a new VPN connection using the AWS Management Console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN Connections**.
3. Choose **Create VPN connection**.
4. In the sections for **Tunnel 1 options** and **Tunnel 2 options**, under **Tunnel Endpoint Lifecycle Control**, select **Enable**.
5. Choose **Create VPN Connection**.

To enable tunnel endpoint lifecycle control during creation of a new VPN connection using the AWS CLI

Use the [create-vpn-connection](#) command to turn on tunnel endpoint lifecycle control.

Verify if tunnel endpoint lifecycle control is enabled

You can verify whether tunnel endpoint lifecycle control is enabled on an existing VPN tunnel by using the AWS Management Console or CLI.

To verify if tunnel endpoint lifecycle control is enabled using the AWS Management Console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the left-side navigation pane, choose **Site-to-Site VPN Connections**.
3. Select the appropriate connection under **VPN connections**.

4. Select the **Tunnel details** tab.
5. In the tunnel details, look for **Tunnel Endpoint Lifecycle Control**, which will report whether the feature is **Enabled** or **Disabled**.

To verify if tunnel endpoint lifecycle control is enabled using the AWS CLI

Use the [describe-vpn-connections](#) command to verify if tunnel endpoint lifecycle control is enabled.

Check for available updates

After you enable the tunnel endpoint lifecycle control feature, you can view whether a maintenance update is available for your VPN connection by using the AWS Management Console or CLI.

To check for available updates using the AWS Management Console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the left-side navigation pane, choose **Site-to-Site VPN Connections**.
3. Select the appropriate connection under **VPN connections**.
4. Select the **Tunnel details** tab.
5. Check the **Pending maintenance** column. The status will be either **Available** or **None**.

To check for available updates using the AWS CLI

Use the [get-vpn-tunnel-replacement-status](#) command to check for available updates.

Accept a maintenance update

When a maintenance update is available, you can accept it using the AWS Management Console or CLI.

To accept an available maintenance update using the AWS Management Console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the left-side navigation pane, choose **Site-to-Site VPN Connections**.
3. Select the appropriate connection under **VPN connections**.

4. Choose **Actions**, then **Replace VPN Tunnel**.
5. Select the specific tunnel that you want to replace by choosing the appropriate **VPN tunnel outside IP address**.
6. Choose **Replace**.

To accept an available maintenance update using the AWS CLI

Use the [replace-vpn-tunnel](#) command to accept an available maintenance update.

Turn tunnel endpoint lifecycle control off

If you no longer want to use the tunnel endpoint lifecycle control feature, you can turn it off using the AWS Management Console or the AWS CLI. When you turn off this feature, AWS will automatically deploy maintenance updates periodically, and these updates might happen during your business hours. To avoid any business impact, we highly recommend that you configure both tunnels in your VPN connection for high availability.

Note

While there is an available pending maintenance, you cannot specify the **skip tunnel replacement** option while turning the feature off. You can always turn the feature off without using the **skip tunnel replacement** option, but AWS will automatically deploy the available pending maintenance updates by initiating a tunnel endpoint replacement immediately.

To turn off tunnel endpoint lifecycle control using the AWS Management Console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the left-side navigation pane, choose **Site-to-Site VPN Connections**.
3. Select the appropriate connection under **VPN connections**.
4. Choose **Actions**, then **Modify VPN tunnel options**.
5. Select the specific tunnel that you want to modify by choosing the appropriate **VPN tunnel outside IP address**.
6. To turn off tunnel endpoint lifecycle control, under **Tunnel Endpoint Lifecycle Control**, clear the **Enable** check box.

7. (Optional) Select **Skip tunnel replacement**.
8. Choose **Save changes**.

To turn off tunnel endpoint lifecycle control using the AWS CLI

Use the [modify-vpn-tunnel-options](#) command to turn off tunnel endpoint lifecycle control.

Customer gateway options for your Site-to-Site VPN connection

The following table describes the information you'll need to create a customer gateway resource in AWS.

Item	Description
(Optional) Name tag.	Creates a tag with a key of 'Name' and a value that you specify.
(Dynamic routing only) Border Gateway Protocol (BGP) Autonomous System Number (ASN) of the customer gateway.	<p>ASN in the range of 1 – 2,147,483,647 is supported. You can use an existing public ASN assigned to your network, with the exception of the following:</p> <ul style="list-style-type: none"> • 7224 - Reserved in all Regions • 9059 - Reserved in the eu-west-1 Region • 10124 - Reserved in the ap-northeast-1 Region • 17943 - Reserved in the ap-southeast-1 Region <p>If you don't have a public ASN, you can use a private ASN in the range of 64,512–65,534. The default ASN is 65000. Customer gateways do not support private ASNs in the range 4,200,000,000 to 4,294,967,294. See Site-to-Site VPN routing options for more details on routing.</p>

Item	Description
(Optional) The IP address of the customer gateway device's external interface.	<p>The IP address must be static.</p> <p>If your customer gateway device is behind a network address translation (NAT) device, use the IP address of your NAT device. Also, ensure that UDP packets on port 500 (and port 4500, if NAT-traversal is being used) are allowed to pass between your network and the AWS Site-to-Site VPN endpoints. See Firewall Rules for more info.</p> <p>An IP address is not required when you are using a private certificate from AWS Private Certificate Authority and a public VPN.</p>

Item	Description
(Optional) Private certificate from a subordinate CA using AWS Certificate Manager (ACM).	<p>If you want to use certificate based authentication, provide the ARN of an ACM private certificate that will be used on your customer gateway device.</p> <p>When you create a customer gateway, you can configure the customer gateway to use AWS Private Certificate Authority private certificates to authenticate the Site-to-Site VPN.</p> <p>When you choose to use this option, you create an entirely AWS-hosted private certificate authority (CA) for internal use by your organization. Both the root CA certificate and subordinate CA certificates are stored and managed by AWS Private CA.</p> <p>Before you create the customer gateway, you create a private certificate from a subordinate CA using AWS Private Certificate Authority, and then specify the certificate when you configure the customer gateway. For information about creating a private certificate, see Creating and managing a private CA in the <i>AWS Private Certificate Authority User Guide</i>.</p>
(Optional) Device.	A name for the customer gateway device associated with this customer gateway.

Accelerated Site-to-Site VPN connections

You can optionally enable acceleration for your Site-to-Site VPN connection. An accelerated Site-to-Site VPN connection (accelerated VPN connection) uses AWS Global Accelerator to route traffic from your on-premises network to an AWS edge location that is closest to your customer gateway device. AWS Global Accelerator optimizes the network path, using the congestion-free AWS global

network to route traffic to the endpoint that provides the best application performance (for more information, see [AWS Global Accelerator](#)). You can use an accelerated VPN connection to avoid network disruptions that might occur when traffic is routed over the public internet.

When you create an accelerated VPN connection, we create and manage two accelerators on your behalf, one for each VPN tunnel. You cannot view or manage these accelerators yourself by using the AWS Global Accelerator console or APIs.

For information about the AWS Regions that support Accelerated VPN connections, see the [AWS Accelerated Site-to-Site VPN FAQs](#).

Enabling acceleration

By default, when you create a Site-to-Site VPN connection, acceleration is disabled. You can optionally enable acceleration when you create a new Site-to-Site VPN attachment on a transit gateway. For more information and steps, see [Create a transit gateway VPN attachment](#).

Accelerated VPN connections use a separate pool of IP addresses for the tunnel endpoint IP addresses. The IP addresses for the two VPN tunnels are selected from two separate [network zones](#).

Rules and restrictions

To use an accelerated VPN connection, the following rules apply:

- Acceleration is only supported for Site-to-Site VPN connections that are attached to a transit gateway. Virtual private gateways do not support accelerated VPN connections.
- An Accelerated Site-to-Site VPN connection cannot be used with an AWS Direct Connect public virtual interface.
- You cannot turn on or turn off acceleration for an existing Site-to-Site VPN connection. Instead, you can create a new Site-to-Site VPN connection with acceleration on or off as needed. Then, configure your customer gateway device to use the new Site-to-Site VPN connection and delete the old Site-to-Site VPN connection.
- NAT-traversal (NAT-T) is required for an accelerated VPN connection and is enabled by default. If you downloaded a [configuration file](#) from the Amazon VPC console, check the NAT-T setting and adjust it if necessary.
- IKE negotiation for accelerated VPN tunnels must be initiated from the customer gateway device. The two tunnel options that effect this behavior are `Startup Action` and `DPD Timeout Action`. See [VPN tunnel options](#) and [VPN tunnel initiation options](#) for more information.

- Site-to-Site VPN connections that use certificate-based authentication might not be compatible with AWS Global Accelerator, due to limited support for packet fragmentation in Global Accelerator. For more information, see [How AWS Global Accelerator works](#). If you require an accelerated VPN connection that uses certificate-based authentication, then your customer gateway device must support IKE fragmentation. Otherwise, do not enable your VPN for acceleration.

Site-to-Site VPN routing options

When you create a Site-to-Site VPN connection, you must do the following:

- Specify the type of routing that you plan to use (static or dynamic)
- Update the [route table](#) for your subnet

There are quotas on the number of routes that you can add to a route table. For more information, see the Route Tables section in [Amazon VPC quotas](#) in the *Amazon VPC User Guide*.

Topics

- [Static and dynamic routing](#)
- [Route tables and VPN route priority](#)
- [Routing during VPN tunnel endpoint updates](#)
- [IPv4 and IPv6 traffic](#)

Static and dynamic routing

The type of routing that you select can depend on the make and model of your customer gateway device. If your customer gateway device supports Border Gateway Protocol (BGP), specify dynamic routing when you configure your Site-to-Site VPN connection. If your customer gateway device does not support BGP, specify static routing.

If you use a device that supports BGP advertising, you don't specify static routes to the Site-to-Site VPN connection because the device uses BGP to advertise its routes to the virtual private gateway. If you use a device that doesn't support BGP advertising, you must select static routing and enter the routes (IP prefixes) for your network that should be communicated to the virtual private gateway.

We recommend that you use BGP-capable devices, when available, because the BGP protocol offers robust liveness detection checks that can assist failover to the second VPN tunnel if the first tunnel goes down. Devices that don't support BGP may also perform health checks to assist failover to the second tunnel when needed.

You must configure your customer gateway device to route traffic from your on-premises network to the Site-to-Site VPN connection. The configuration depends on the make and model of your device. For more information, see [Your customer gateway device](#).

Route tables and VPN route priority

[Route tables](#) determine where network traffic from your VPC is directed. In your VPC route table, you must add a route for your remote network and specify the virtual private gateway as the target. This enables traffic from your VPC that's destined for your remote network to route via the virtual private gateway and over one of the VPN tunnels. You can enable route propagation for your route table to automatically propagate your network routes to the table for you.

We use the most specific route in your route table that matches the traffic to determine how to route the traffic (longest prefix match). If your route table has overlapping or matching routes, the following rules apply:

- If propagated routes from a Site-to-Site VPN connection or AWS Direct Connect connection overlap with the local route for your VPC, the local route is most preferred even if the propagated routes are more specific.
- If propagated routes from a Site-to-Site VPN connection or AWS Direct Connect connection have the same destination CIDR block as other existing static routes (longest prefix match cannot be applied), we prioritize the static routes whose targets are an internet gateway, a virtual private gateway, a network interface, an instance ID, a VPC peering connection, a NAT gateway, a transit gateway, or a gateway VPC endpoint.

For example, the following route table has a static route to an internet gateway, and a propagated route to a virtual private gateway. Both routes have a destination of `172.31.0.0/24`. In this case, all traffic destined for `172.31.0.0/24` is routed to the internet gateway — it is a static route and therefore takes priority over the propagated route.

Destination	Target
10.0.0.0/16	Local

Destination	Target
172.31.0.0/24	vgw-11223344556677889 (propagated)
172.31.0.0/24	igw-12345678901234567 (static)

Only IP prefixes that are known to the virtual private gateway, whether through BGP advertisements or a static route entry, can receive traffic from your VPC. The virtual private gateway does not route any other traffic destined outside of received BGP advertisements, static route entries, or its attached VPC CIDR. Virtual private gateways do not support IPv6 traffic.

When a virtual private gateway receives routing information, it uses path selection to determine how to route traffic. Longest prefix match applies, if all endpoints are healthy. The health of a tunnel endpoint takes precedence over other routing attributes. This precedence applies to VPNs on virtual private gateways and Transit Gateways. If the prefixes are the same, then the virtual private gateway prioritizes routes as follows, from most preferred to least preferred:

- BGP propagated routes from an AWS Direct Connect connection
- Manually added static routes for a Site-to-Site VPN connection
- BGP propagated routes from a Site-to-Site VPN connection
- For matching prefixes where each Site-to-Site VPN connection uses BGP, the AS PATH is compared and the prefix with the shortest AS PATH is preferred.

Note

AWS strongly recommends using customer gateway devices that support asymmetric routing.

For customer gateway devices that support asymmetric routing, we *do not* recommend using AS PATH prepending, to ensure that both tunnels have equal AS PATH. This helps to ensure that the multi-exit discriminator (MED) value that we set on a tunnel during [VPN tunnel endpoint updates](#) is used to determine tunnel priority.

For customer gateway devices that do not support asymmetric routing, you can use AS PATH prepending and Local Preference to prefer one tunnel over the other. However, when the egress path changes, this may cause traffic to drop.

- When the AS PATHs are the same length and if the first AS in the AS_SEQUENCE is the same across multiple paths, multi-exit discriminators (MEDs) are compared. The path with the lowest MED value is preferred.

Route priority is affected during [VPN tunnel endpoint updates](#).

On a Site-to-Site VPN connection, AWS selects one of the two redundant tunnels as the primary egress path. This selection may change at times, and we strongly recommend that you configure both tunnels for high availability, and allow asymmetric routing. The health of a tunnel endpoint takes precedence over other routing attributes. This precedence applies to VPNs on virtual private gateways and Transit Gateways.

For a virtual private gateway, one tunnel across all Site-to-Site VPN connections on the gateway will be selected. To use more than one tunnel, we recommend exploring Equal Cost Multipath (ECMP), which is supported for Site-to-Site VPN connections on a transit gateway. For more information, see [Transit gateways](#) in *Amazon VPC Transit Gateways*. ECMP is not supported for Site-to-Site VPN connections on a virtual private gateway.

For Site-to-Site VPN connections that use BGP, the primary tunnel can be identified by the multi-exit discriminator (MED) value. We recommend advertising more specific BGP routes to influence routing decisions.

For Site-to-Site VPN connections that use static routing, the primary tunnel can be identified by traffic statistics or metrics.

Routing during VPN tunnel endpoint updates

A Site-to-Site VPN connection consists of two VPN tunnels between a customer gateway device and a virtual private gateway or a transit gateway. We recommend that you configure both tunnels for redundancy. From time to time, AWS also performs routine maintenance on your VPN connection, which might briefly disable one of the two tunnels of your VPN connection. For more information, see [Tunnel endpoint replacement notifications](#).

When we perform updates on one VPN tunnel, we set a lower outbound multi-exit discriminator (MED) value on the other tunnel. If you have configured your customer gateway device to use both tunnels, your VPN connection uses the other (up) tunnel during the tunnel endpoint update process.

Note

To ensure that the up tunnel with the lower MED is preferred, ensure that your customer gateway device uses the same Weight and Local Preference values for both tunnels (Weight and Local Preference have higher priority than MED).

IPv4 and IPv6 traffic

Your Site-to-Site VPN connection on a transit gateway can support either IPv4 traffic or IPv6 traffic inside the VPN tunnels. By default, a Site-to-Site VPN connection supports IPv4 traffic inside the VPN tunnels. You can configure a new Site-to-Site VPN connection to support IPv6 traffic inside the VPN tunnels. Then, if your VPC and your on-premises network are configured for IPv6 addressing, you can send IPv6 traffic over the VPN connection.

If you enable IPv6 for the VPN tunnels for your Site-to-Site VPN connection, each tunnel has two CIDR blocks. One is a size /30 IPv4 CIDR block, and the other is a size /126 IPv6 CIDR block.

The following rules apply:

- IPv6 addresses are only supported for the inside IP addresses of the VPN tunnels. The outside tunnel IP addresses for the AWS endpoints are IPv4 addresses, and the public IP address of your customer gateway must be an IPv4 address.
- Site-to-Site VPN connections on a virtual private gateway do not support IPv6.
- You cannot enable IPv6 support for an existing Site-to-Site VPN connection.
- A Site-to-Site VPN connection cannot support both IPv4 and IPv6 traffic.

For more information about creating a VPN connection, see [Step 5: Create a VPN connection](#).

Getting started with AWS Site-to-Site VPN

Use the following procedure to set up an AWS Site-to-Site VPN connection. During creation, you will specify a virtual private gateway, a transit gateway, or "Not associated" as the target gateway type. If you specify "Not associated", you can choose the target gateway type at a later time, or you can use it as a VPN attachment for AWS Cloud WAN. This tutorial helps you create a VPN connection using a virtual private gateway. It assumes that you have an existing VPC with one or more subnets.

To set up a VPN connection using a virtual private gateway, complete the following steps:

Tasks

- [Prerequisites](#)
- [Step 1: Create a customer gateway](#)
- [Step 2: Create a target gateway](#)
- [Step 3: Configure routing](#)
- [Step 4: Update your security group](#)
- [Step 5: Create a VPN connection](#)
- [Step 6: Download the configuration file](#)
- [Step 7: Configure the customer gateway device](#)

Related tasks

- To create a VPN connection for AWS Cloud WAN, see [Create a VPN attachment for AWS Cloud WAN](#).
- To create a VPN connection on a transit gateway, see [Create a transit gateway VPN attachment](#).

Prerequisites

You need the following information to set up and configure the components of a VPN connection.

Item	Information
Customer gateway device	The physical or software device on your side of the VPN connection. You need the vendor

Item	Information
	<p>(for example, Cisco), platform (for example, ISR Series Routers), and software version (for example, IOS 12.4).</p>
Customer gateway	<p>To create the customer gateway resource in AWS, you need the following information:</p> <ul style="list-style-type: none"> • The internet-routable IP address for the device's external interface • The type of routing: static or dynamic • For dynamic routing, the Border Gateway Protocol (BGP) Autonomous System Number (ASN) • (Optional) Private certificate from AWS Private Certificate Authority to authenticate your VPN <p>For more information, see Customer gateway options.</p>
(Optional) The ASN for the AWS side of the BGP session	<p>You specify this when you create a virtual private gateway or transit gateway. If you do not specify a value, the default ASN applies. For more information, see Virtual private gateway.</p>
VPN connection	<p>To create the VPN connection, you need the following information:</p> <ul style="list-style-type: none"> • For static routing, the IP prefixes for your private network. • (Optional) Tunnel options for each VPN tunnel. For more information, see Tunnel options for your Site-to-Site VPN connection.

Step 1: Create a customer gateway

A customer gateway provides information to AWS about your customer gateway device or software application. For more information, see [Customer gateway](#).

If you plan to use a private certificate to authenticate your VPN, create a private certificate from a subordinate CA using AWS Private Certificate Authority. For information about creating a private certificate, see [Creating and managing a private CA](#) in the *AWS Private Certificate Authority User Guide*.

Note

You must specify either an IP address, or the Amazon Resource Name of the private certificate.

To create a customer gateway using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Customer gateways**.
3. Choose **Create customer gateway**.
4. (Optional) For **Name tag**, enter a name for your customer gateway. Doing so creates a tag with a key of Name and the value that you specify.
5. For **BGP ASN**, enter a Border Gateway Protocol (BGP) Autonomous System Number (ASN) for your customer gateway.
6. (Optional) For **IP address**, enter the static, internet-routable IP address for your customer gateway device. If your customer gateway device is behind a NAT device that's enabled for NAT-T, use the public IP address of the NAT device.
7. (Optional) If you want to use a private certificate, for **Certificate ARN**, choose the Amazon Resource Name of the private certificate.
8. (Optional) For **Device**, enter a name for the customer gateway device associated with this customer gateway.
9. Choose **Create customer gateway**.

To create a customer gateway using the command line or API

- [CreateCustomerGateway](#) (Amazon EC2 Query API)
- [create-customer-gateway](#) (AWS CLI)
- [New-EC2CustomerGateway](#) (AWS Tools for Windows PowerShell)

Step 2: Create a target gateway

To establish a VPN connection between your VPC and your on-premises network, you must create a target gateway on the AWS side of the connection. The target gateway can be a virtual private gateway or a transit gateway.

Create a virtual private gateway

When you create a virtual private gateway, you can specify a custom private Autonomous System Number (ASN) for the Amazon side of the gateway, or use the Amazon default ASN. This ASN must be different from the ASN that you specified for the customer gateway.

After you create a virtual private gateway, you must attach it to your VPC.

To create a virtual private gateway and attach it to your VPC

1. In the navigation pane, choose **Virtual private gateways**.
2. Choose **Create virtual private gateway**.
3. (Optional) For **Name tag**, enter a name for your virtual private gateway. Doing so creates a tag with a key of Name and the value that you specify.
4. For **Autonomous System Number (ASN)**, keep the default selection, **Amazon default ASN**, to use the default Amazon ASN. Otherwise, choose **Custom ASN** and enter a value. For a 16-bit ASN, the value must be in the 64512 to 65534 range. For a 32-bit ASN, the value must be in the 4200000000 to 4294967294 range.
5. Choose **Create virtual private gateway**.
6. Select the virtual private gateway you created, then choose **Actions, Attach to VPC**.
7. For **Available VPCs**, choose your VPC and then choose **Attach to VPC**.

To create a virtual private gateway using the command line or API

- [CreateVpnGateway](#) (Amazon EC2 Query API)

- [create-vpn-gateway](#) (AWS CLI)
- [New-EC2VpnGateway](#) (AWS Tools for Windows PowerShell)

To attach a virtual private gateway to a VPC using the command line or API

- [AttachVpnGateway](#) (Amazon EC2 Query API)
- [attach-vpn-gateway](#) (AWS CLI)
- [Add-EC2VpnGateway](#) (AWS Tools for Windows PowerShell)

Create a transit gateway

For more information about creating a transit gateway, see [Transit gateways](#) in *Amazon VPC Transit Gateways*.

Step 3: Configure routing

To enable instances in your VPC to reach your customer gateway, you must configure your route table to include the routes used by your VPN connection and point them to your virtual private gateway or transit gateway.

(Virtual private gateway) Enable route propagation in your route table

You can enable route propagation for your route table to automatically propagate Site-to-Site VPN routes.

For static routing, the static IP prefixes that you specify for your VPN configuration are propagated to the route table when the status of the VPN connection is UP. Similarly, for dynamic routing, the BGP-advertised routes from your customer gateway are propagated to the route table when the status of the VPN connection is UP.

Note

If your connection is interrupted but the VPN connection remains UP, any propagated routes that are in your route table are not automatically removed. Keep this in mind if, for example, you want traffic to fail over to a static route. In that case, you might have to disable route propagation to remove the propagated routes.

To enable route propagation using the console

1. In the navigation pane, choose **Route tables**.
2. Select the route table that's associated with the subnet.
3. On the **Route propagation** tab, choose **Edit route propagation**. Select the virtual private gateway that you created in the previous procedure, and then choose **Save**.

Note

If you do not enable route propagation, you must manually enter the static routes used by your VPN connection. To do this, select your route table, choose **Routes**, **Edit**. For **Destination**, add the static route used by your Site-to-Site VPN connection. For **Target**, select the virtual private gateway ID, and choose **Save**.

To disable route propagation using the console

1. In the navigation pane, choose **Route tables**.
2. Select the route table that's associated with the subnet.
3. On the **Route propagation** tab, choose **Edit route propagation**. Clear the **Propagate** check box for the virtual private gateway.
4. Choose **Save**.

To enable route propagation using the command line or API

- [EnableVgwRoutePropagation](#) (Amazon EC2 Query API)
- [enable-vgw-route-propagation](#) (AWS CLI)
- [Enable-EC2VgwRoutePropagation](#) (AWS Tools for Windows PowerShell)

To disable route propagation using the command line or API

- [DisableVgwRoutePropagation](#) (Amazon EC2 Query API)
- [disable-vgw-route-propagation](#) (AWS CLI)
- [Disable-EC2VgwRoutePropagation](#) (AWS Tools for Windows PowerShell)

(Transit gateway) Add a route to your route table

If you enabled route table propagation for your transit gateway, the routes for the VPN attachment are propagated to the transit gateway route table. For more information, see [Routing](#) in *Amazon VPC Transit Gateways*.

If you attach a VPC to your transit gateway and you want to enable resources in the VPC to reach your customer gateway, you must add a route to your subnet route table to point to the transit gateway.

To add a route to a VPC route table

1. On the navigation pane, choose **Route tables**.
2. Choose the route table that is associated with your VPC.
3. On the **Routes** tab, choose **Edit routes**.
4. Choose **Add route**.
5. For **Destination**, enter the destination IP address range. For **Target**, choose the transit gateway.
6. Choose **Save changes**.

Step 4: Update your security group

To allow access to instances in your VPC from your network, you must update your security group rules to enable inbound SSH, RDP, and ICMP access.

To add rules to your security group to enable access

1. In the navigation pane, choose **Security groups**.
2. Select the default security group for the VPC.
3. On the **Inbound rules** tab, choose **Edit inbound rules**.
4. Add rules that allow inbound SSH, RDP, and ICMP access from your network, and then choose **Save rules**. For more information, see [Work with security group rules](#) in the *Amazon VPC User Guide*.

Step 5: Create a VPN connection

Create the VPN connection using the customer gateway in combination with the virtual private gateway or transit gateway that you created earlier.

To create a VPN connection

1. In the navigation pane, choose **Site-to-Site VPN connections**.
2. Choose **Create VPN connection**.
3. (Optional) For **Name tag**, enter a name for your VPN connection. Doing so creates a tag with a key of Name and the value that you specify.
4. For **Target gateway type**, choose either **Virtual private gateway** or **Transit gateway**. Then, choose the virtual private gateway or transit gateway that you created earlier.
5. For **Customer gateway**, select **Existing**, then choose the customer gateway that you created earlier from **Customer gateway ID**.
6. Select one of the routing options based on whether your customer gateway device supports Border Gateway Protocol (BGP):
 - If your customer gateway device supports BGP, choose **Dynamic (requires BGP)**.
 - If your customer gateway device does not support BGP, choose **Static**. For **Static IP Prefixes**, specify each IP prefix for the private network of your VPN connection.
7. If your target gateway type is transit gateway, for **Tunnel inside IP version**, specify whether the VPN tunnels support IPv4 or IPv6 traffic. IPv6 traffic is only supported for VPN connections on a transit gateway.
8. If you specified **IPv4** for **Tunnel inside IP version**, you can optionally specify the IPv4 CIDR ranges for the customer gateway and AWS sides that are allowed to communicate over the VPN tunnels. The default is `0.0.0.0/0`.

If you specified **IPv6** for **Tunnel inside IP version**, you can optionally specify the IPv6 CIDR ranges for the customer gateway and AWS sides that are allowed to communicate over the VPN tunnels. The default for both ranges is `::/0`.
9. For **Outside IP address type**, keep the default option, **PublicIPv4**.
10. (Optional) For **Tunnel options**, you can specify the following information for each tunnel:
 - A size /30 IPv4 CIDR block from the `169.254.0.0/16` range for the inside tunnel IPv4 addresses.

- If you specified **IPv6** for **Tunnel inside IP version**, a /126 IPv6 CIDR block from the fd00::/8 range for the inside tunnel IPv6 addresses.
- The IKE pre-shared key (PSK). The following versions are supported: IKEv1 or IKEv2.
- To edit the advanced options for your tunnel, choose **Edit tunnel options**. For more information, see [VPN tunnel options](#).

11. Choose **Create VPN connection**. It might take a few minutes to create the VPN connection.

To create a VPN connection using the command line or API

- [CreateVpnConnection](#) (Amazon EC2 Query API)
- [create-vpn-connection](#) (AWS CLI)
- [New-EC2VpnConnection](#) (AWS Tools for Windows PowerShell)

Step 6: Download the configuration file

After you create the VPN connection, you can download a sample configuration file to use for configuring the customer gateway device.

Important

The configuration file is an example only and might not match your intended VPN connection settings entirely. It specifies the minimum requirements for a VPN connection of AES128, SHA1, and Diffie-Hellman group 2 in most AWS Regions, and AES128, SHA2, and Diffie-Hellman group 14 in the AWS GovCloud Regions. It also specifies pre-shared keys for authentication. You must modify the example configuration file to take advantage of additional security algorithms, Diffie-Hellman groups, private certificates, and IPv6 traffic. We have introduced IKEv2 support in the configuration files for many popular customer gateway devices and will continue to add additional files over time. For a list of configuration files with IKEv2 support, see [Your customer gateway device](#).

Permissions

To properly load the download configuration screen from the AWS Management Console, you must ensure that your IAM role or user has permission for the following Amazon EC2 APIs: `GetVpnConnectionDeviceTypes` and `GetVpnConnectionDeviceSampleConfiguration`.

To download the configuration file using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN connections**.
3. Select your VPN connection and choose **Download configuration**.
4. Select the **Vendor, Platform, Software**, and **IKE version** that correspond to your customer gateway device. If your device is not listed, choose **Generic**.
5. Choose **Download**.

To download a sample configuration file using the command line or API

- [GetVpnConnectionDeviceTypes](#) (Amazon EC2 Query API)
- [GetVpnConnectionDeviceSampleConfiguration](#) (Amazon EC2 Query API)
- [get-vpn-connection-device-types](#) (AWS CLI)
- [get-vpn-connection-device-sample-configuration](#) (AWS CLI)

Step 7: Configure the customer gateway device

Use the sample configuration file to configure your customer gateway device. The customer gateway device is the physical or software appliance on your side of the VPN connection. For more information, see [Your customer gateway device](#).

Site-to-Site VPN architectures

The following are common Site-to-Site VPN architectures:

- [the section called “Single and multiple VPN connections”](#)
- [the section called “Redundant VPN connections”](#)
- [the section called “AWS VPN CloudHub”](#)

Site-to-Site VPN single and multiple VPN connection examples

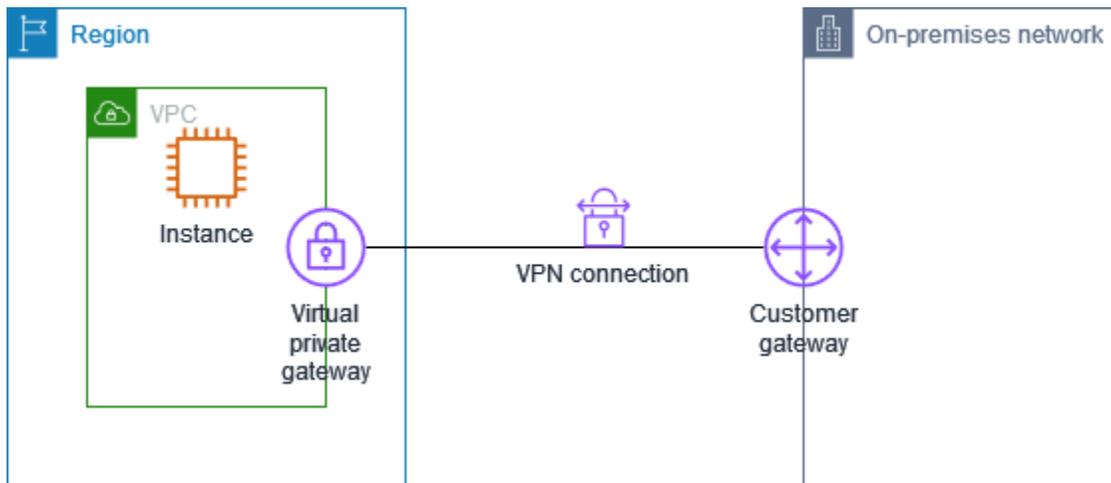
The following diagrams illustrate single and multiple Site-to-Site VPN connections.

Examples

- [Single Site-to-Site VPN connection](#)
- [Single Site-to-Site VPN connection with a transit gateway](#)
- [Multiple Site-to-Site VPN connections](#)
- [Multiple Site-to-Site VPN connections with a transit gateway](#)
- [Site-to-Site VPN connection with AWS Direct Connect](#)
- [Private IP Site-to-Site VPN connection with AWS Direct Connect](#)

Single Site-to-Site VPN connection

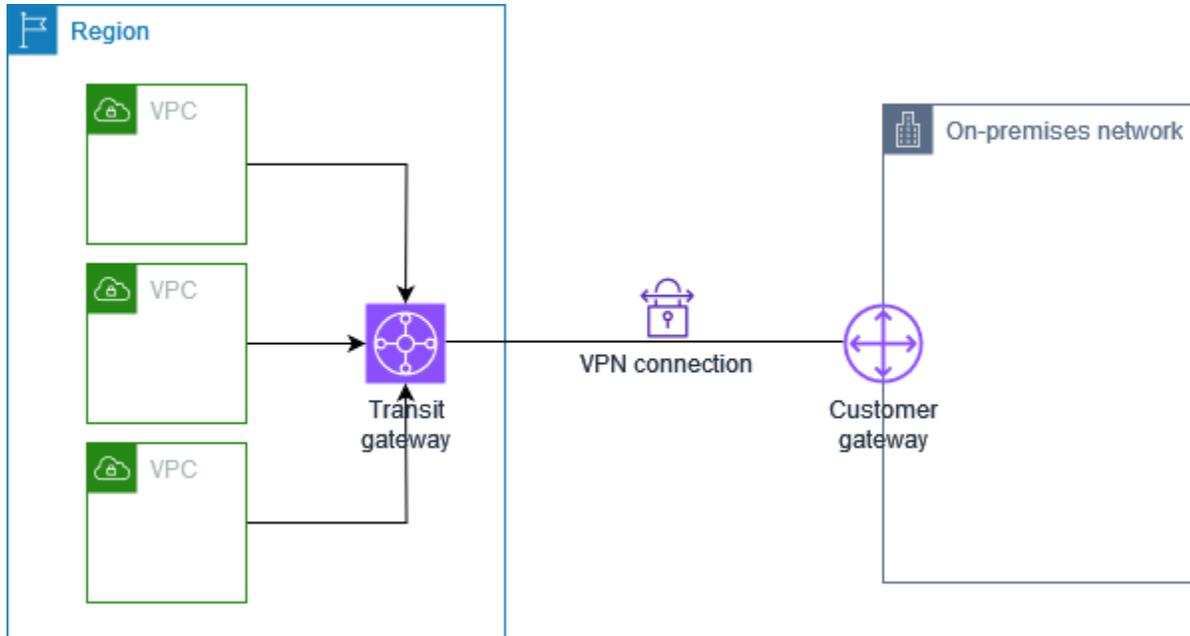
The VPC has an attached virtual private gateway, and your on-premises (remote) network includes a customer gateway device, which you must configure to enable the VPN connection. You must update the VPC route tables so that any traffic from the VPC bound for your network goes to the virtual private gateway.



For steps to set up this scenario, see [Getting started with AWS Site-to-Site VPN](#).

Single Site-to-Site VPN connection with a transit gateway

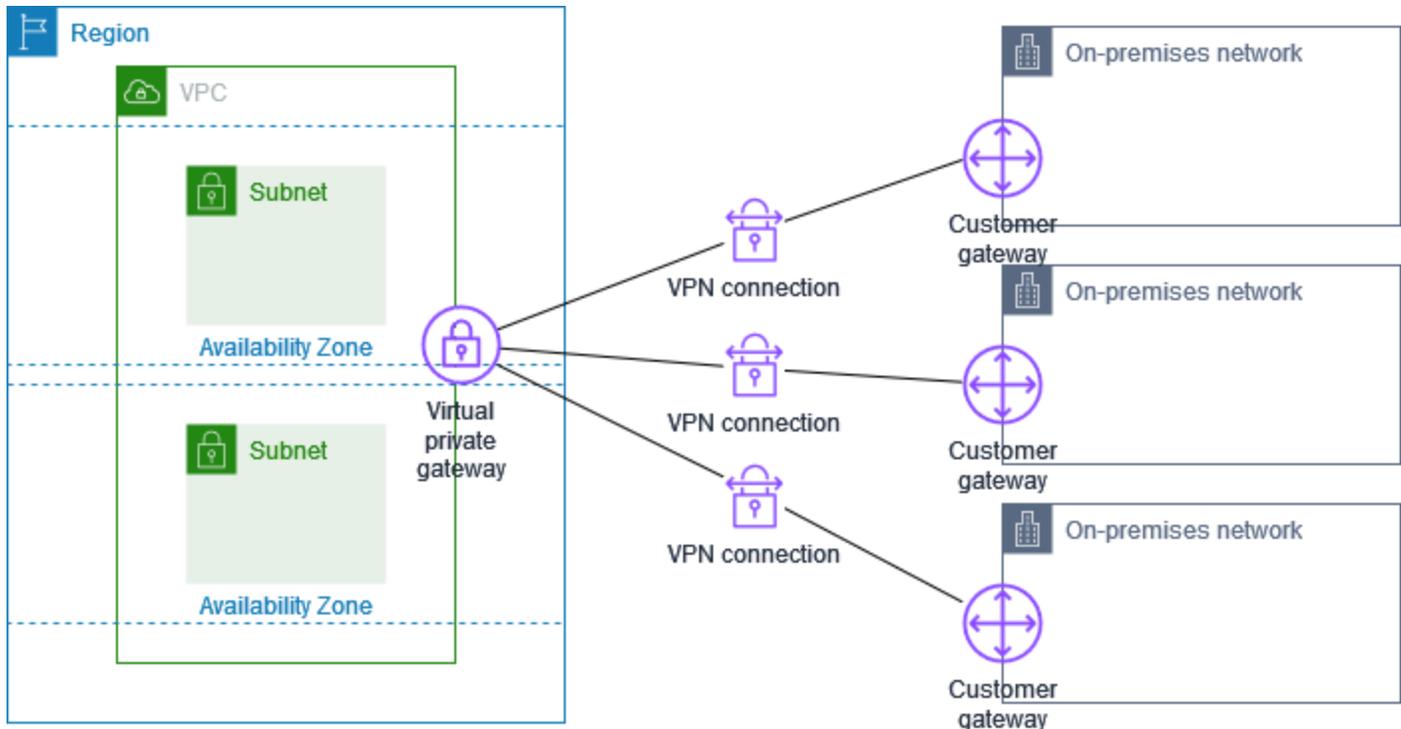
The VPC has an attached transit gateway, and your on-premises (remote) network includes a customer gateway device, which you must configure to enable the VPN connection. You must update the VPC route tables so that any traffic from the VPC bound for your network goes to the transit gateway.



For steps to set up this scenario, see [Getting started with AWS Site-to-Site VPN](#).

Multiple Site-to-Site VPN connections

The VPC has an attached virtual private gateway, and you have multiple Site-to-Site VPN connections to multiple on-premises locations. You set up the routing so that any traffic from the VPC bound for your networks is routed to the virtual private gateway.

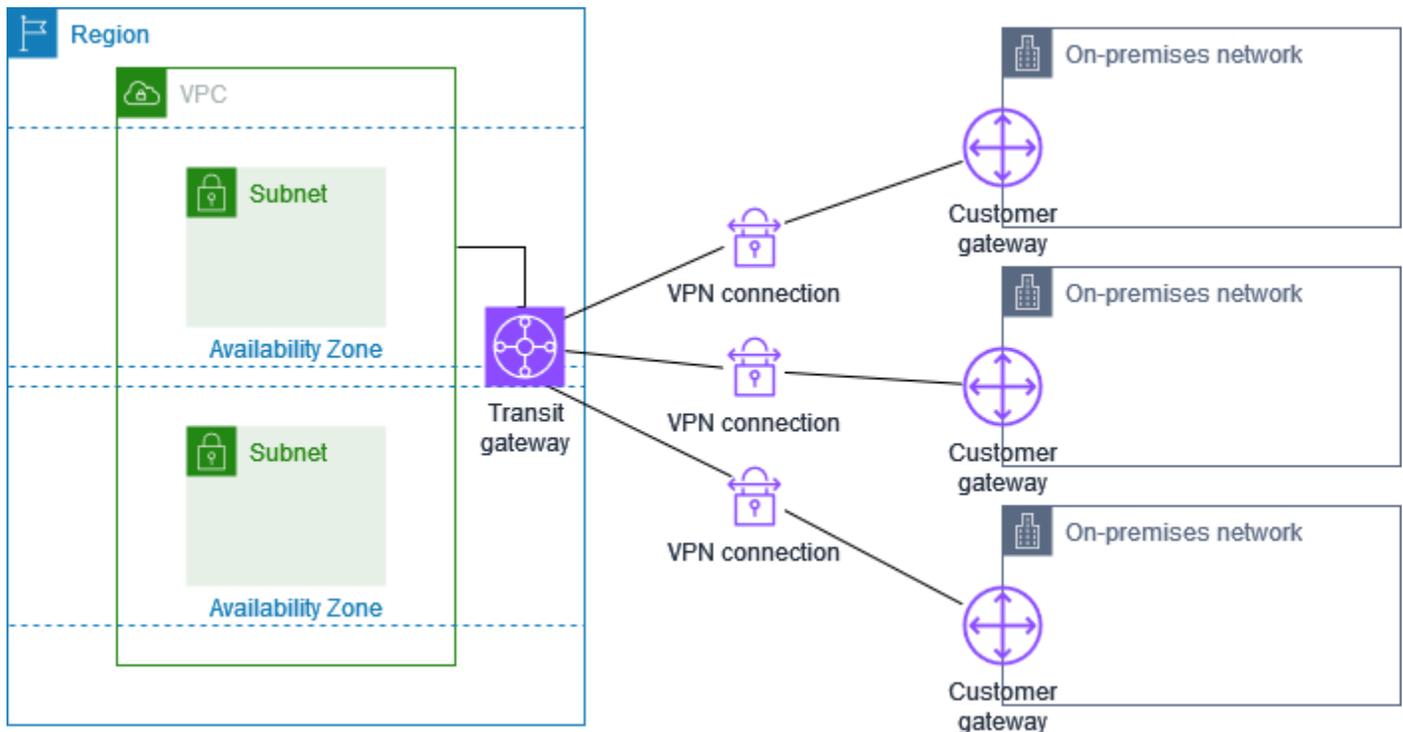


When you create multiple Site-to-Site VPN connections to a single VPC, you can configure a second customer gateway to create a redundant connection to the same external location. For more information, see [Using redundant Site-to-Site VPN connections to provide failover](#).

You can also use this scenario to create Site-to-Site VPN connections to multiple geographic locations and provide secure communication between sites. For more information, see [Providing secure communication between sites using VPN CloudHub](#).

Multiple Site-to-Site VPN connections with a transit gateway

The VPC has an attached transit gateway, and you have multiple Site-to-Site VPN connections to multiple on-premises locations. You set up the routing so that any traffic from the VPC bound for your networks is routed to the transit gateway.

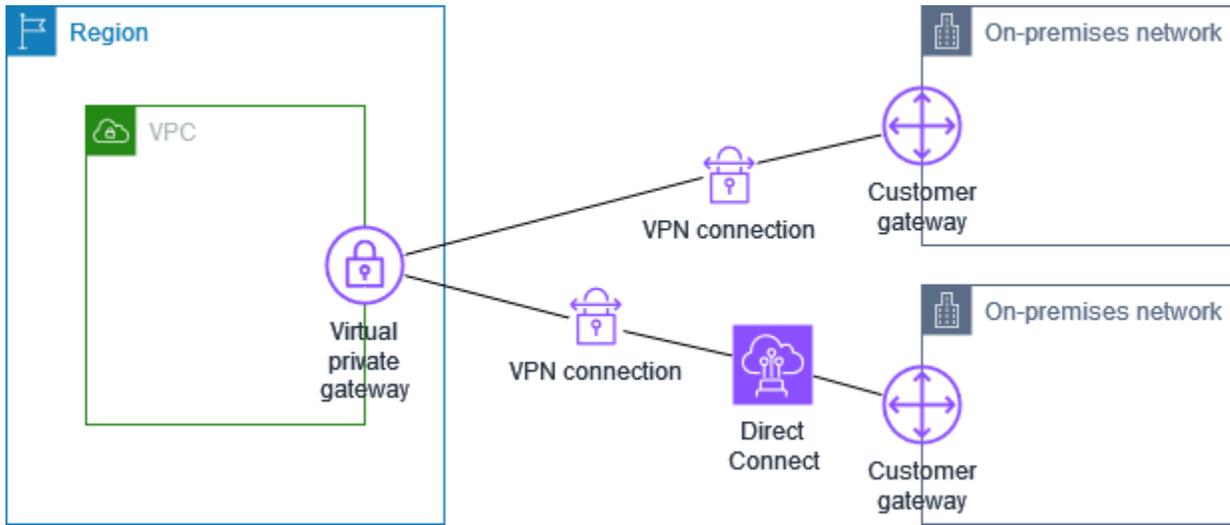


When you create multiple Site-to-Site VPN connections to a single transit gateway, you can configure a second customer gateway to create a redundant connection to the same external location.

You can also use this scenario to create Site-to-Site VPN connections to multiple geographic locations and provide secure communication between sites.

Site-to-Site VPN connection with AWS Direct Connect

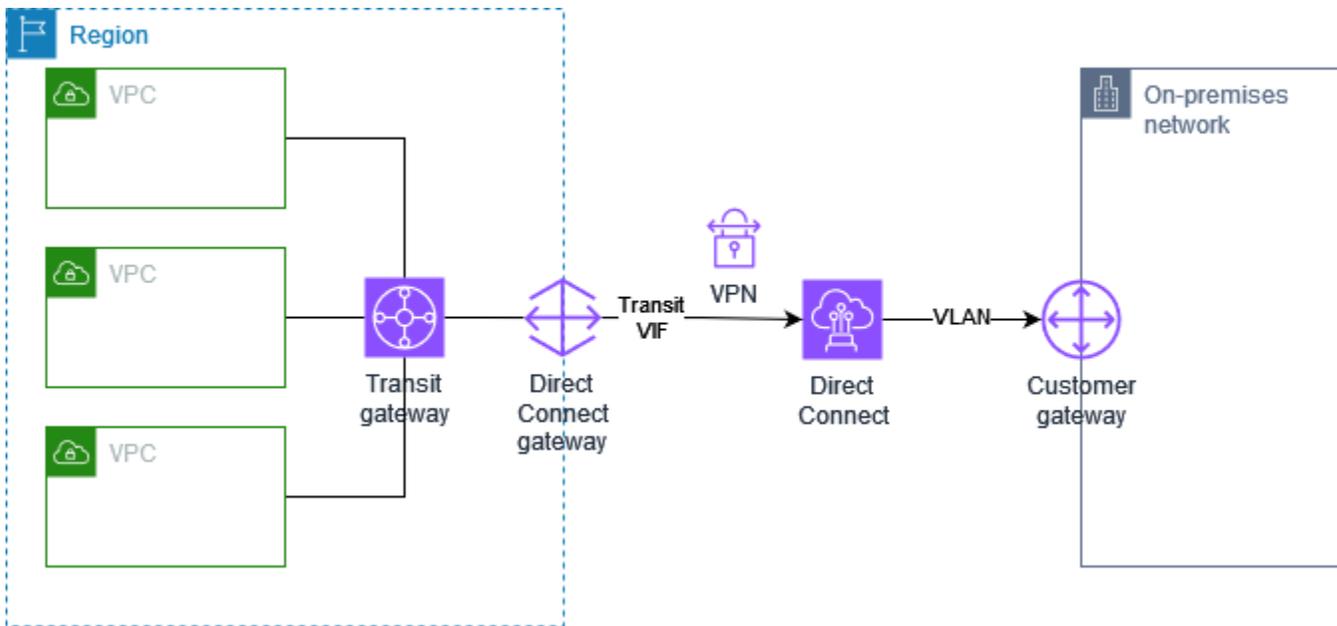
The VPC has an attached virtual private gateway, and connects to your on-premises (remote) network through AWS Direct Connect. You can configure an AWS Direct Connect public virtual interface to establish a dedicated network connection between your network to public AWS resources through a virtual private gateway. You set up the routing so that any traffic from the VPC bound for your network routes to the virtual private gateway and the AWS Direct Connect connection.



When both AWS Direct Connect and the VPN connection are set up on the same virtual private gateway, adding or removing objects might cause the virtual private gateway to enter the 'attaching' state. This indicates a change is being made to internal routing that will switch between AWS Direct Connect and the VPN connection to minimize interruptions and packet loss. When this is complete, the virtual private gateway returns to the 'attached' state.

Private IP Site-to-Site VPN connection with AWS Direct Connect

With a private IP Site-to-Site VPN you can encrypt AWS Direct Connect traffic between your on-premises network and AWS without the use of public IP addresses. Private IP VPN over AWS Direct Connect ensures that traffic between AWS and on-premises networks is both secure and private, allowing customers to comply with regulatory and security mandates.



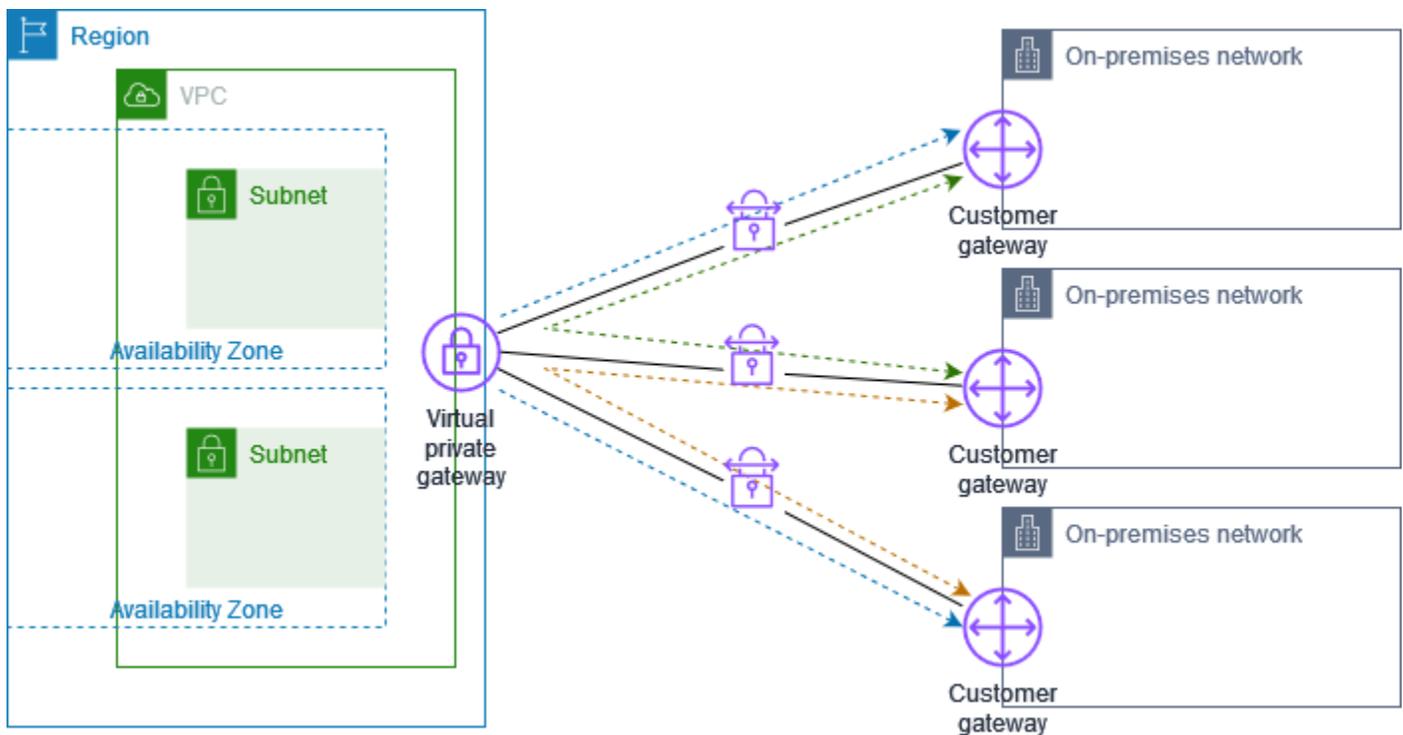
For more information, see the following blog post: [Introducing AWS Site-to-Site VPN Private IP VPNs](#).

Providing secure communication between sites using VPN CloudHub

If you have multiple AWS Site-to-Site VPN connections, you can provide secure communication between sites using the AWS VPN CloudHub. This enables your sites to communicate with each other, and not just with the resources in your VPC. The VPN CloudHub operates on a simple hub-and-spoke model that you can use with or without a VPC. This design is suitable if you have multiple branch offices and existing internet connections and would like to implement a convenient, potentially low-cost hub-and-spoke model for primary or backup connectivity between these sites.

Overview

The following diagram shows the VPN CloudHub architecture. The dashed lines show network traffic between remote sites being routed over the VPN connections. The sites must not have overlapping IP ranges.



For this scenario, do the following:

1. Create a single virtual private gateway.
2. Create multiple customer gateways, each with the public IP address of the gateway. You must use a unique Border Gateway Protocol (BGP) Autonomous System Number (ASN) for each customer gateway.
3. Create a dynamically routed Site-to-Site VPN connection from each customer gateway to the common virtual private gateway.
4. Configure the customer gateway devices to advertise a site-specific prefix (such as 10.0.0.0/24, 10.0.1.0/24) to the virtual private gateway. These routing advertisements are received and re-advertised to each BGP peer, enabling each site to send data to and receive data from the other sites. This is done using the network statements in the VPN configuration files for the Site-to-Site VPN connection. The network statements differ slightly depending on the type of router you use.
5. Configure the routes in your subnet route tables to enable instances in your VPC to communicate with your sites. For more information, see [\(Virtual private gateway\) Enable route propagation in your route table](#). You can configure an aggregate route in your route table (for example, 10.0.0.0/16). Use more specific prefixes between customer gateways devices and the virtual private gateway.

Sites that use AWS Direct Connect connections to the virtual private gateway can also be part of the AWS VPN CloudHub. For example, your corporate headquarters in New York can have an AWS Direct Connect connection to the VPC and your branch offices can use Site-to-Site VPN connections to the VPC. The branch offices in Los Angeles and Miami can send and receive data with each other and with your corporate headquarters, all using the AWS VPN CloudHub.

Pricing

To use AWS VPN CloudHub, you pay typical Amazon VPC Site-to-Site VPN connection rates. You are billed the connection rate for each hour that each VPN is connected to the virtual private gateway. When you send data from one site to another using the AWS VPN CloudHub, there is no cost to send data from your site to the virtual private gateway. You only pay standard AWS data transfer rates for data that is relayed from the virtual private gateway to your endpoint.

For example, if you have a site in Los Angeles and a second site in New York and both sites have a Site-to-Site VPN connection to the virtual private gateway, you pay the per hour rate for each Site-to-Site VPN connection (so if the rate was \$.05 per hour, it would be a total of \$.10 per hour). You also pay the standard AWS data transfer rates for all data that you send from Los Angeles to New

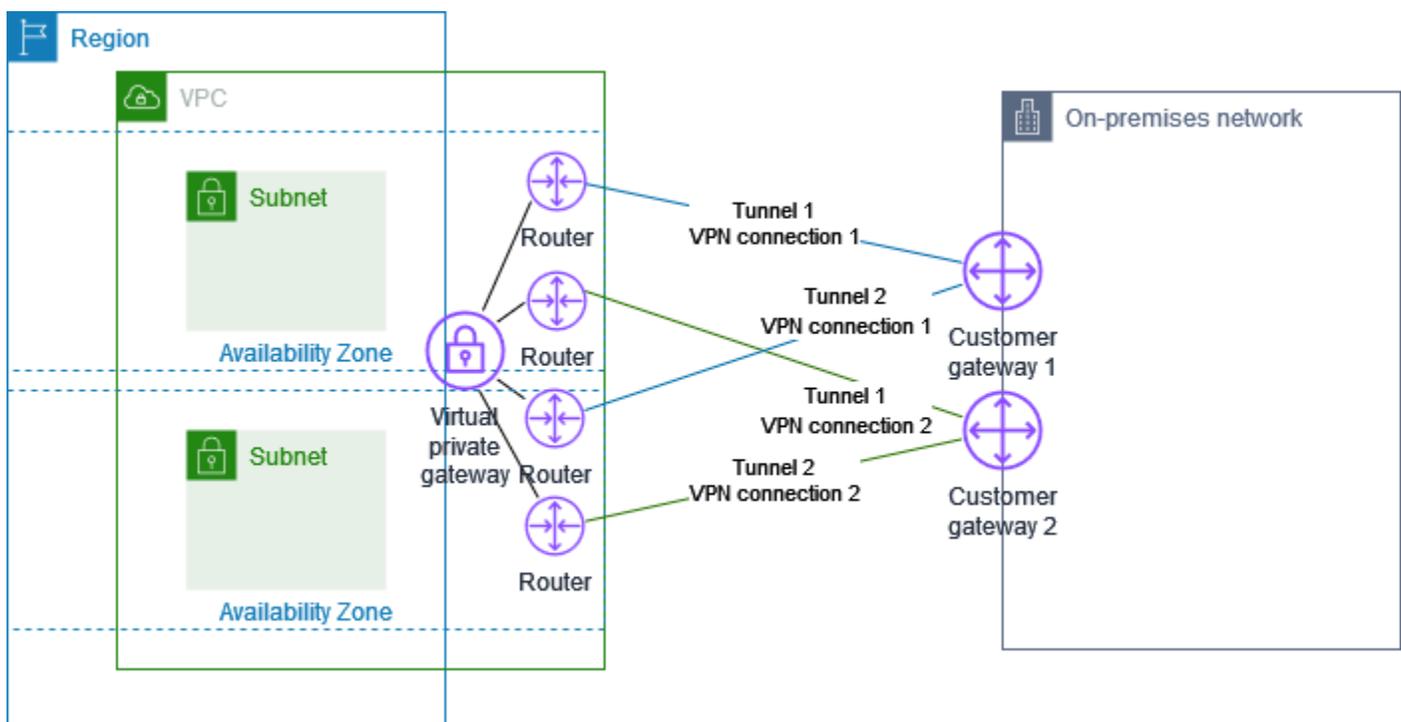
York (and vice versa) that traverses each Site-to-Site VPN connection. Network traffic sent over the Site-to-Site VPN connection to the virtual private gateway is free but network traffic sent over the Site-to-Site VPN connection from the virtual private gateway to the endpoint is billed at the standard AWS data transfer rate.

For more information, see [Site-to-Site VPN Connection Pricing](#).

Using redundant Site-to-Site VPN connections to provide failover

To protect against a loss of connectivity in case your customer gateway device becomes unavailable, you can set up a second Site-to-Site VPN connection to your VPC and virtual private gateway by adding a second customer gateway device. By using redundant VPN connections and customer gateway devices, you can perform maintenance on one of your devices while traffic continues to flow over the second VPN connection.

The following diagram shows two VPN connections. Each VPN connection has its own tunnels and its own customer gateway.



For this scenario, do the following:

- Set up a second Site-to-Site VPN connection by using the same virtual private gateway and creating a new customer gateway. The customer gateway IP address for the second Site-to-Site VPN connection must be publicly accessible.
- Configure a second customer gateway device. Both devices should advertise the same IP ranges to the virtual private gateway. We use BGP routing to determine the path for traffic. If one customer gateway device fails, the virtual private gateway directs all traffic to the working customer gateway device.

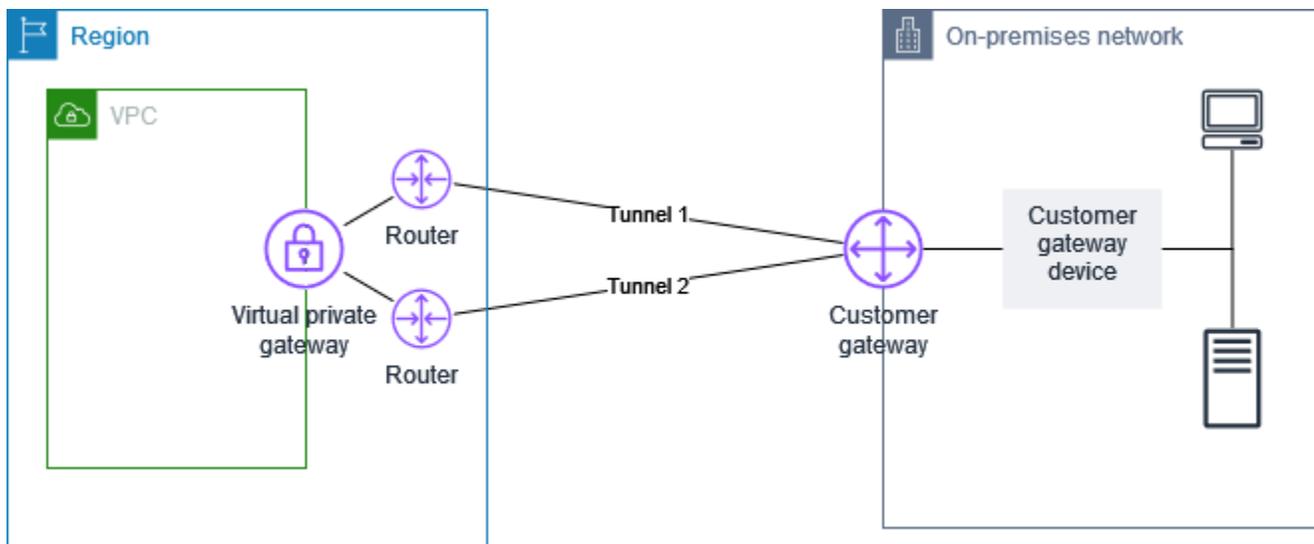
Dynamically routed Site-to-Site VPN connections use the Border Gateway Protocol (BGP) to exchange routing information between your customer gateways and the virtual private gateways. Statically routed Site-to-Site VPN connections require you to enter static routes for the remote network on your side of the customer gateway. BGP-advertised and statically entered route information allow gateways on both sides to determine which tunnels are available and reroute traffic if a failure occurs. We recommend that you configure your network to use the routing information provided by BGP (if available) to select an available path. The exact configuration depends on the architecture of your network.

For more information about creating and configuring a customer gateway and a Site-to-Site VPN connection, see [Getting started with AWS Site-to-Site VPN](#).

Your customer gateway device

A *customer gateway device* is a physical or software appliance that you own or manage in your on-premises network (on your side of a Site-to-Site VPN connection). You or your network administrator must configure the device to work with the Site-to-Site VPN connection.

The following diagram shows your network, the customer gateway device, and the VPN connection that goes to the virtual private gateway that is attached to your VPC. The two lines between the customer gateway and virtual private gateway represent the tunnels for the VPN connection. If there's a device failure within AWS, your VPN connection automatically fails over to the second tunnel so that your access isn't interrupted. From time to time, AWS also performs routine maintenance on the VPN connection, which might briefly disable one of the two tunnels of your VPN connection. For more information, see [Site-to-Site VPN tunnel endpoint replacements](#). When you configure your customer gateway device, it's therefore important that you configure it to use both tunnels.



For the steps to set up a VPN connection, see [Getting started with AWS Site-to-Site VPN](#). During this process, you create a customer gateway resource in AWS, which provides information to AWS about your device, for example, its public-facing IP address. For more information, see [Customer gateway options for your Site-to-Site VPN connection](#). The customer gateway resource in AWS does not configure or create the customer gateway device. You must configure the device yourself.

You can also find software VPN appliances on the [AWS Marketplace](#).

Topics

- [Example configuration files](#)
- [Requirements for your customer gateway device](#)
- [Best practices for your customer gateway device](#)
- [Configuring a firewall between the internet and your customer gateway device](#)
- [Multiple VPN connection scenarios](#)
- [Routing for your customer gateway device](#)
- [Example customer gateway device configurations for static routing](#)
- [Example customer gateway device configurations for dynamic routing \(BGP\)](#)
- [Configuring Windows Server as a customer gateway device](#)
- [Troubleshooting your customer gateway device](#)

Example configuration files

After you create the VPN connection, you additionally have the option to download an AWS-provided sample configuration file from the Amazon VPC console, or by using the EC2 API. See [Step 6: Download the configuration file](#) for more information. You can also download .zip files of sample configurations specifically for static vs. dynamic routing:

Download .zip files

- Static configuration: [the section called “Example configuration files”](#)
- Dynamic configuration: [the section called “Example configuration files”](#)

The AWS-provided sample configuration file contains information specific to your VPN connection which you can use to configure your customer gateway device. These device-specific configuration files are only available for devices that AWS has tested. If your specific customer gateway device is not listed, you can download a generic configuration file to begin with.

Important

The configuration file is an example only and might not match your intended Site-to-Site VPN connection settings entirely. It specifies the minimum requirements for a Site-to-Site VPN connection of AES128, SHA1, and Diffie-Hellman group 2 in most AWS Regions, and AES128, SHA2, and Diffie-Hellman group 14 in the AWS GovCloud Regions. It also specifies pre-shared keys for authentication. You must modify the example configuration file to take

advantage of additional security algorithms, Diffie-Hellman groups, private certificates, and IPv6 traffic.

Note

These device-specific configuration files are provided by AWS on a best-effort basis. While they have been tested by AWS, this testing is limited. If you are experiencing an issue with the configuration files, you might need to contact the specific vendor for additional support.

The following table contains a list of devices which have an example configuration file available for download that has been updated to support IKEv2. We have introduced IKEv2 support in the configuration files for many popular customer gateway devices and will continue to add additional files over time. This list will be updated as more example configuration files are added.

Vendor	Platform	Software
Checkpoint	Gaia	R80.10+
Cisco Meraki	MX Series	15.12+ (WebUI)
Cisco Systems, Inc.	ASA 5500 Series	ASA 9.7+ VTI
Cisco Systems, Inc.	CSRv AMI	IOS 12.4+
Fortinet	Fortigate 40+ Series	FortiOS 6.4.4+ (GUI)
Juniper Networks, Inc.	J-Series Routers	JunOS 9.5+
Juniper Networks, Inc.	SRX Routers	JunOS 11.0+
Mikrotik	RouterOS	6.44.3
Palo Alto Networks	PA Series	PANOS 7.0+
SonicWall	NSA, TZ	OS 6.5
Sophos	Sophos Firewall	v19+

Vendor	Platform	Software
Strongswan	Ubuntu 16.04	Strongswan 5.5.1+
Yamaha	RTX Routers	Rev.10.01.16+

Requirements for your customer gateway device

If you have a device that isn't in the preceding list of examples, this section describes the requirements that the device must meet for you to use it to establish a Site-to-Site VPN connection.

There are four main parts to the configuration of your customer gateway device. The following symbols represent each part of the configuration.

IKE	Internet key exchange (IKE) security association. This is required to exchange keys used to establish the IPsec security association.
IPsec	IPsec security association. This handles the tunnel's encryption, authentication, and so on.
Tunnel	Tunnel interface. This receives traffic going to and from the tunnel.
BGP	(Optional) Border Gateway Protocol (BGP) peering. For devices that use BGP, this exchanges routes between the customer gateway device and the virtual private gateway.

The following table lists the requirements for the customer gateway device, the related RFC (for reference), and comments about the requirements.

Each VPN connection consists of two separate tunnels. Each tunnel contains an IKE security association, an IPsec security association, and a BGP peering. You are limited to one unique security association (SA) pair per tunnel (one inbound and one outbound), and therefore two unique SA pairs in total for two tunnels (four SAs). Some devices use a policy-based VPN and create as many SAs as ACL entries. Therefore, you might need to consolidate your rules and then filter so that you don't permit unwanted traffic.

By default, the VPN tunnel comes up when traffic is generated and the IKE negotiation is initiated from your side of the VPN connection. You can configure the VPN connection to initiate the IKE negotiation from the AWS side of the connection instead. For more information, see [Site-to-Site VPN tunnel initiation options](#).

VPN endpoints support rekey and can start renegotiations when phase 1 is about to expire if the customer gateway device hasn't sent any renegotiation traffic.

Requirement	RFC	Comments
Establish IKE security association <div style="background-color: #FFD700; padding: 2px; display: inline-block; margin-left: 10px;">IKE</div>	RFC 2409 RFC 7296	<p>The IKE security association is established first between the virtual private gateway and the customer gateway device using a pre-shared key or a private certificate that uses AWS Private Certificate Authority as the authenticator. When established, IKE negotiates an ephemeral key to secure future IKE messages. There must be complete agreement among the parameters, including encryption and authentication parameters.</p> <p>When you create a VPN connection in AWS, you can specify your own pre-shared key for each tunnel, or you can let AWS generate one for you. Alternatively, you can specify the private certificate using AWS Private Certificate Authority to use for your customer gateway device. For more information, about configuring VPN tunnels see Tunnel options for your Site-to-Site VPN connection.</p> <p>The following versions are supported: IKEv1 and IKEv2.</p> <p>We support Main mode only with IKEv1.</p> <p>The Site-to-Site VPN service is a route-based solution. If you are using a policy-based configuration, you must limit your configuration to a single security association (SA).</p>

Requirement	RFC	Comments
Establish IPsec security associations in Tunnel mode 	RFC 4301	Using the IKE ephemeral key, keys are established between the virtual private gateway and the customer gateway device to form an IPsec security association (SA). Traffic between gateways is encrypted and decrypted using this SA. The ephemeral keys used to encrypt traffic within the IPsec SA are automatically rotated by IKE on a regular basis to ensure confidentiality of communications.
Use the AES 128-bit encryption or AES 256-bit encryption function	RFC 3602	The encryption function is used to ensure privacy for both IKE and IPsec security associations.
Use the SHA-1 or SHA-2 (256) hashing function	RFC 2404	This hashing function is used to authenticate both IKE and IPsec security associations.
Use Diffie-Hellman Perfect Forward Secrecy.	RFC 2409	<p>IKE uses Diffie-Hellman to establish ephemeral keys to secure all communication between customer gateway devices and virtual private gateways.</p> <p>The following groups are supported:</p> <ul style="list-style-type: none"> • Phase 1 groups: 2, 14-24 • Phase 2 groups: 2, 5, 14-24
(Dynamically-routed VPN connections) Use IPsec Dead Peer Detection	RFC 3706	Dead Peer Detection enables the VPN devices to rapidly identify when a network condition prevents delivery of packets across the internet. When this occurs, the gateways delete the security associations and attempt to create new associations. During this process, the alternate IPsec tunnel is used if possible.

Requirement	RFC	Comments
(Dynamically-routed VPN connections) Bind tunnel to logical interface (route-based VPN)	None	Your device must be able to bind the IPsec tunnel to a logical interface. The logical interface contains an IP address that is used to establish BGP peering to the virtual private gateway. This logical interface should perform no additional encapsulation (for example, GRE or IP in IP). Your interface should be set to a 1399 byte Maximum Transmission Unit (MTU).
(Dynamically-routed VPN connections) Establish BGP peerings	RFC 4271	BGP is used to exchange routes between the customer gateway device and the virtual private gateway for devices that use BGP. All BGP traffic is encrypted and transmitted via the IPsec Security Association. BGP is required for both gateways to exchange the IP prefixes that are reachable through the IPsec SA.

An AWS VPN connection does not support Path MTU Discovery ([RFC 1191](#)).

If you have a firewall between your customer gateway device and the internet, see [Configuring a firewall between the internet and your customer gateway device](#).

Best practices for your customer gateway device

Reset the "Don't Fragment (DF)" flag on packets

Some packets carry a flag, known as the Don't Fragment (DF) flag, which indicates that the packet should not be fragmented. If the packets carry the flag, the gateways generate an ICMP Path MTU Exceeded message. In some cases, applications do not contain adequate mechanisms for processing these ICMP messages and for reducing the amount of data transmitted in each packet. Some VPN devices can override the DF flag and fragment packets unconditionally as required. If your customer gateway device has this ability, we recommend that you use it as appropriate. See [RFC 791](#) for more details.

Fragment IP packets before encryption

If packets being sent to over your Site-to-Site VPN connection exceed the MTU size, they must be fragmented. To avoid decreased performance, we recommend that you configure your customer gateway device to fragment the packets *before* they are encrypted. Site-to-Site VPN will then reassemble any fragmented packets before forwarding them to the next destination, in order to achieve higher packet-per-second flows through the AWS network. See [RFC 4459](#) for more details.

Ensure packet size does not exceed MTU for destination networks

Since Site-to-Site VPN will reassemble any fragmented packets received from your customer gateway device before forwarding to the next destination, keep in mind, there may be packet size/MTU considerations for destination networks where these packets get forwarded next, such as over AWS Direct Connect.

Adjust MTU and MSS sizes according to the algorithms in use

TCP packets are often the most common type of packet across IPsec tunnels. Site-to-Site VPN supports a maximum transmission unit (MTU) of 1446 bytes and a corresponding maximum segment size (MSS) of 1406 bytes. However, encryption algorithms have varying header sizes and can prevent the ability to achieve these maximum values. To obtain optimal performance by avoiding fragmentation, we recommend that you set the MTU and MSS based specifically on the algorithms being used.

Use the following table to set your MTU/MSS to avoid fragmentation and achieve optimal performance:

Encryption Algorithm	Hashing Algorithm	NAT-Traversal	MTU	MSS (IPv4)	MSS (IPv6-in-IPv4)
AES-GCM-16	N/A	disabled	1446	1406	1386
AES-GCM-16	N/A	enabled	1438	1398	1378
AES-CBC	SHA1/SHA2-256	disabled	1438	1398	1378
AES-CBC	SHA1/SHA2-256	enabled	1422	1382	1362
AES-CBC	SHA2-384	disabled	1422	1382	1362

Encryption Algorithm	Hashing Algorithm	NAT-Traversal	MTU	MSS (IPv4)	MSS (IPv6-in-IPv4)
AES-CBC	SHA2-384	enabled	1422	1382	1362
AES-CBC	SHA2-512	disabled	1422	1382	1362
AES-CBC	SHA2-512	enabled	1406	1366	1346

 **Note**

The AES-GCM algorithms cover both encryption and authentication, so there is no distinct authentication algorithm choice which would affect MTU.

Configuring a firewall between the internet and your customer gateway device

You must have a static IP address to use as the endpoint for the IPsec tunnels that connect your customer gateway device to AWS Site-to-Site VPN endpoints. If a firewall is in place between AWS and your customer gateway device, the rules in the following tables must be in place to establish the IPsec tunnels. The IP addresses for the AWS-side will be in the configuration file.

Inbound (from the internet)

Input rule I1

Source IP	Tunnel1 Outside IP
Dest IP	Customer Gateway
Protocol	UDP
Source port	500
Destination	500

Input rule I2

Source IP	Tunnel2 Outside IP
Dest IP	Customer Gateway
Protocol	UDP
Source port	500
Destination port	500

Input rule I3

Source IP	Tunnel1 Outside IP
Dest IP	Customer Gateway
Protocol	IP 50 (ESP)

Input rule I4

Source IP	Tunnel2 Outside IP
Dest IP	Customer Gateway
Protocol	IP 50 (ESP)

Outbound (to the internet)

Output rule O1

Source IP	Customer Gateway
Dest IP	Tunnel1 Outside IP
Protocol	UDP
Source port	500
Destination port	500

Output rule O2

Source IP	Customer Gateway
Dest IP	Tunnel2 Outside IP
Protocol	UDP
Source port	500
Destination port	500

Output rule O3

Source IP	Customer Gateway
Dest IP	Tunnel1 Outside IP
Protocol	IP 50 (ESP)

Output rule O4

Source IP	Customer Gateway
Dest IP	Tunnel2 Outside IP
Protocol	IP 50 (ESP)

Rules I1, I2, O1, and O2 enable the transmission of IKE packets. Rules I3, I4, O3, and O4 enable the transmission of IPsec packets that contain the encrypted network traffic.

Note

If you are using NAT traversal (NAT-T) on your device, ensure that UDP traffic on port 4500 is also allowed to pass between your network and the AWS Site-to-Site VPN endpoints. Check if your device is advertising NAT-T.

Multiple VPN connection scenarios

The following are scenarios in which you might create multiple VPN connections with one or more customer gateway devices.

Multiple VPN connections using the same customer gateway device

You can create additional VPN connections from your on-premises location to other VPCs using the same customer gateway device. You can reuse the same customer gateway IP address for each of those VPN connections.

Redundant VPN connection using a second customer gateway device

To protect against a loss of connectivity if your customer gateway device becomes unavailable, you can set up a second VPN connection using a second customer gateway device. For more information, see [Using redundant Site-to-Site VPN connections to provide failover](#). When you establish redundant customer gateway devices at a single location, both devices should advertise the same IP ranges.

Multiple customer gateway devices to a single virtual private gateway (AWS VPN CloudHub)

You can establish multiple VPN connections to a single virtual private gateway from multiple customer gateway devices. This enables you to have multiple locations connected to the AWS VPN CloudHub. For more information, see [Providing secure communication between sites using VPN CloudHub](#). When you have customer gateway devices at multiple geographic locations, each device should advertise a unique set of IP ranges specific to the location.

Routing for your customer gateway device

AWS recommends advertising specific BGP routes to influence routing decisions in the virtual private gateway. Check your vendor documentation for the commands that are specific to your device.

When you create multiple VPN connections, the virtual private gateway sends network traffic to the appropriate VPN connection using statically assigned routes or BGP route advertisements. Which route depends on how the VPN connection was configured. Statically assigned routes are preferred over BGP advertised routes in cases where identical routes exist in the virtual private gateway. If you select the option to use BGP advertisement, then you cannot specify static routes.

For more information about route priority, see [Route tables and VPN route priority](#).

Example customer gateway device configurations for static routing

Topics

- [Example configuration files](#)
- [User interface procedures for static routing](#)
- [Additional information for Cisco devices](#)
- [Testing](#)

Example configuration files

To download a sample configuration file with values specific to your Site-to-Site VPN connection configuration, use the Amazon VPC console, the AWS command line or the Amazon EC2 API. For more information, see [Step 6: Download the configuration file](#).

You can also download generic example configuration files for static routing that do not include values specific to your Site-to-Site VPN connection configuration: [static-routing-examples.zip](#)

The files use placeholder values for some components. For example, they use:

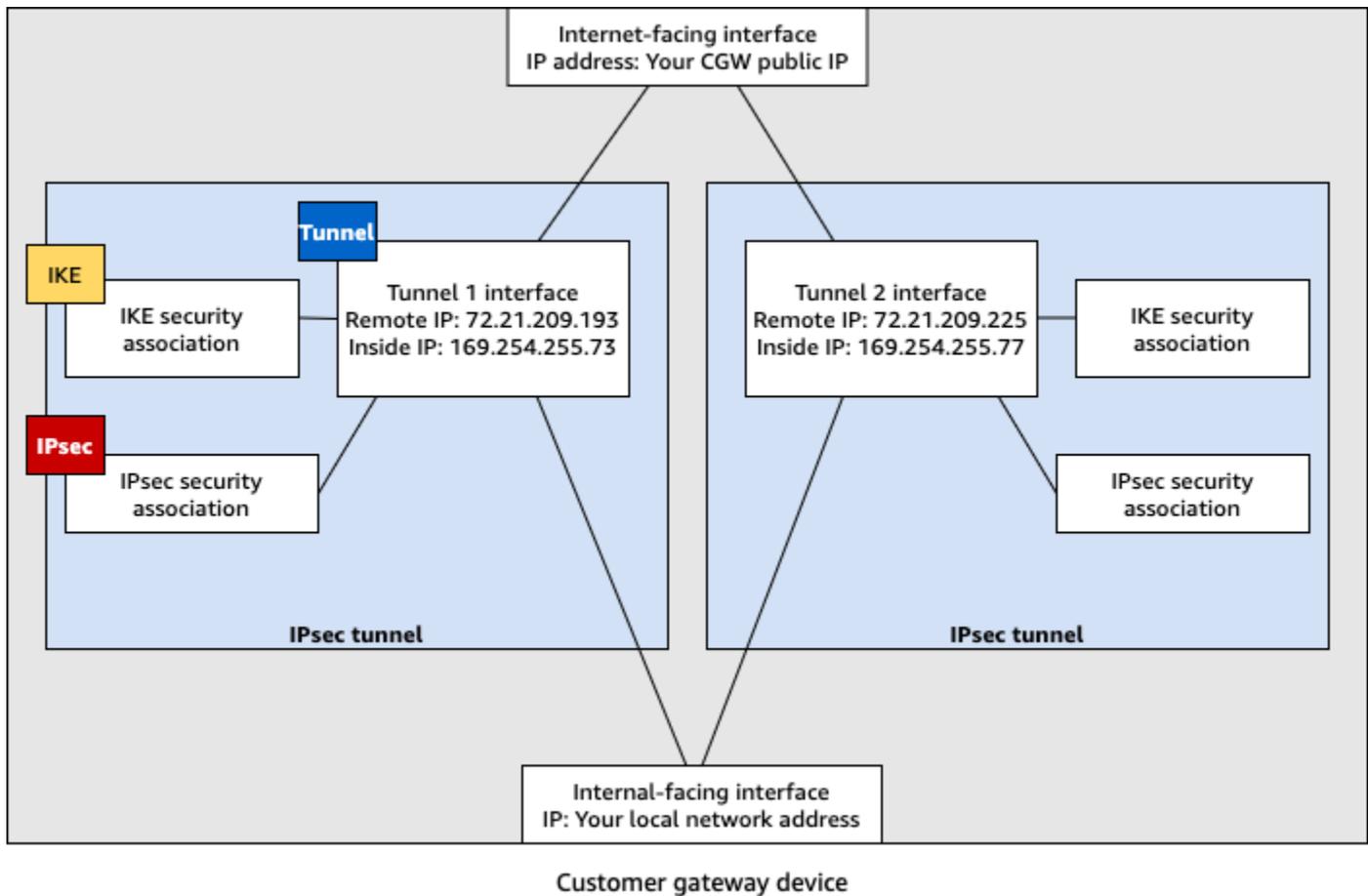
- Example values for the VPN connection ID, customer gateway ID and virtual private gateway ID
- Placeholders for the remote (outside) IP address AWS endpoints (*AWS_ENDPOINT_1* and *AWS_ENDPOINT_2*)
- A placeholder for the IP address for the internet-routable external interface on the customer gateway device (*your-cgw-ip-address*)
- A placeholder for the pre-shared key value (pre-shared-key)
- Example values for the tunnel inside IP addresses.
- Example values for MTU setting.

Note

MTU settings provided in the sample configuration files are examples only. Please refer to [Best practices for your customer gateway device](#) for information on setting the optimal MTU value for your situation.

In addition to providing placeholder values, the files specify the minimum requirements for a Site-to-Site VPN connection of AES128, SHA1, and Diffie-Hellman group 2 in most AWS Regions, and AES128, SHA2, and Diffie-Hellman group 14 in the AWS GovCloud Regions. They also specify pre-shared keys for [authentication](#). You must modify the example configuration file to take advantage of additional security algorithms, Diffie-Hellman groups, private certificates, and IPv6 traffic.

The following diagram provides an overview of the different components that are configured on the customer gateway device. It includes example values for the tunnel interface IP addresses.



User interface procedures for static routing

The following are some example procedures for configuring a customer gateway device using its user interface (if available).

Check Point

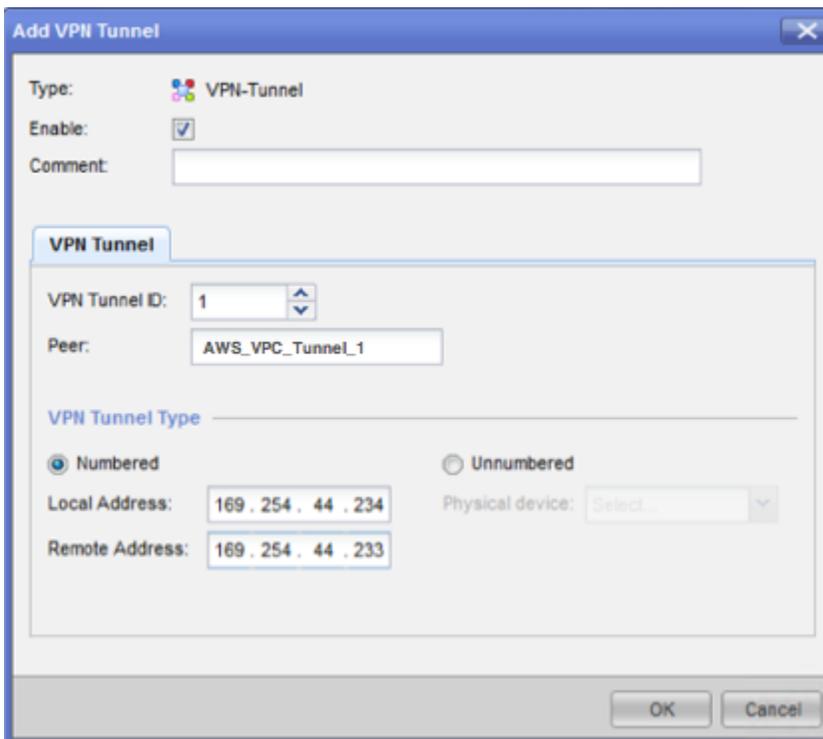
The following are steps for configuring your customer gateway device if your device is a Check Point Security Gateway device running R77.10 or above, using the Gaia operating system and

Check Point SmartDashboard. You can also refer to the [Check Point Security Gateway IPsec VPN to Amazon Web Services VPC](#) article on the Check Point Support Center.

To configure the tunnel interface

The first step is to create the VPN tunnels and provide the private (inside) IP addresses of the customer gateway and virtual private gateway for each tunnel. To create the first tunnel, use the information provided under the IPsec Tunnel #1 section of the configuration file. To create the second tunnel, use the values provided in the IPsec Tunnel #2 section of the configuration file.

1. Open the Gaia portal of your Check Point Security Gateway device.
2. Choose **Network Interfaces, Add, VPN tunnel**.
3. In the dialog box, configure the settings as follows, and choose **OK** when you are done:
 - For **VPN Tunnel ID**, enter any unique value, such as 1.
 - For **Peer**, enter a unique name for your tunnel, such as AWS_VPC_Tunnel_1 or AWS_VPC_Tunnel_2.
 - Ensure that **Numbered** is selected, and for **Local Address**, enter the IP address specified for CGW Tunnel IP in the configuration file, for example, 169.254.44.234.
 - For **Remote Address**, enter the IP address specified for VGW Tunnel IP in the configuration file, for example, 169.254.44.233.



4. Connect to your security gateway over SSH. If you're using the non-default shell, change to `clish` by running the following command: `clish`
5. For tunnel 1, run the following command.

```
set interface vpnt1 mtu 1436
```

For tunnel 2, run the following command.

```
set interface vpnt2 mtu 1436
```

6. Repeat these steps to create a second tunnel, using the information under the IPsec Tunnel1 #2 section of the configuration file.

To configure the static routes

In this step, specify the static route to the subnet in the VPC for each tunnel to enable you to send traffic over the tunnel interfaces. The second tunnel enables failover in case there is an issue with the first tunnel. If an issue is detected, the policy-based static route is removed

from the routing table, and the second route is activated. You must also enable the Check Point gateway to ping the other end of the tunnel to check if the tunnel is up.

1. In the Gaia portal, choose **IPv4 Static Routes, Add**.
2. Specify the CIDR of your subnet, for example, `10.28.13.0/24`.
3. Choose **Add Gateway, IP Address**.
4. Enter the IP address specified for VGW Tunnel IP in the configuration file (for example, `169.254.44.233`), and specify a priority of 1.
5. Select **Ping**.
6. Repeat steps 3 and 4 for the second tunnel, using the VGW Tunnel IP value under the IPsec Tunnel #2 section of the configuration file. Specify a priority of 2.

Destination: 10.28.13.0/24

Next Hop Type: Normal

Normal: Accept and forward packets.
Reject: Drop packets, and send *unreachable* messages.
Black Hole: Drop packets, but don't send *unreachable* messages.

Rank: Default: 60

Local Scope:

Comment:

Add Gateway

Ping:

Gateway	Priority
169.254.44.233	1
169.254.44.5	2

Save Cancel

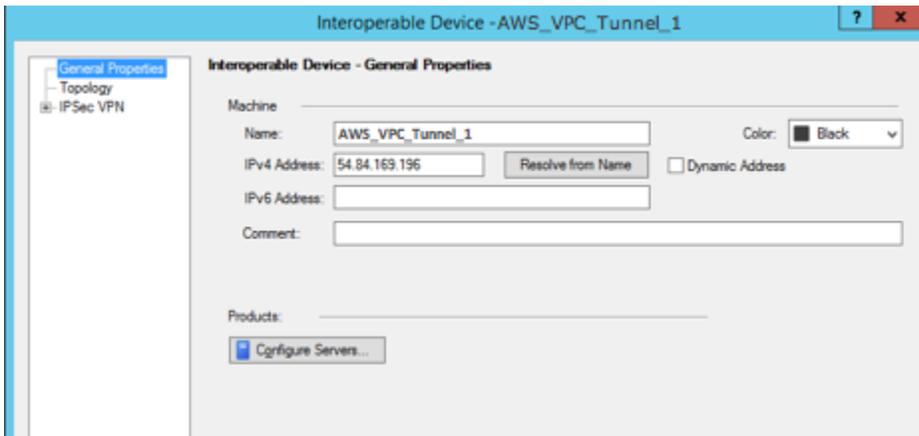
7. Choose **Save**.

If you're using a cluster, repeat the preceding steps for the other members of the cluster.

To define a new network object

In this step, you create a network object for each VPN tunnel, specifying the public (outside) IP addresses for the virtual private gateway. You later add these network objects as satellite gateways for your VPN community. You also need to create an empty group to act as a placeholder for the VPN domain.

1. Open the Check Point SmartDashboard.
2. For **Groups**, open the context menu and choose **Groups, Simple Group**. You can use the same group for each network object.
3. For **Network Objects**, open the context (right-click) menu and choose **New, Interoperable Device**.
4. For **Name**, enter the name that you provided for your tunnel, for example, `AWS_VPC_Tunnel_1` or `AWS_VPC_Tunnel_2`.
5. For **IPv4 Address**, enter the outside IP address of the virtual private gateway provided in the configuration file, for example, `54.84.169.196`. Save your settings and close the dialog box.



6. In the SmartDashboard, open your gateway properties and in the category pane, choose **Topology**.
7. To retrieve the interface configuration, choose **Get Topology**.
8. In the **VPN Domain** section, choose **Manually defined**, and then browse to and select the empty simple group that you created in step 2. Choose **OK**.

Note

You can keep any existing VPN domain that you've configured. However, ensure that the hosts and networks that are used or served by the new VPN connection

are not declared in that VPN domain, especially if the VPN domain is automatically derived.

9. Repeat these steps to create a second network object, using the information under the IPsec Tunnel #2 section of the configuration file.

Note

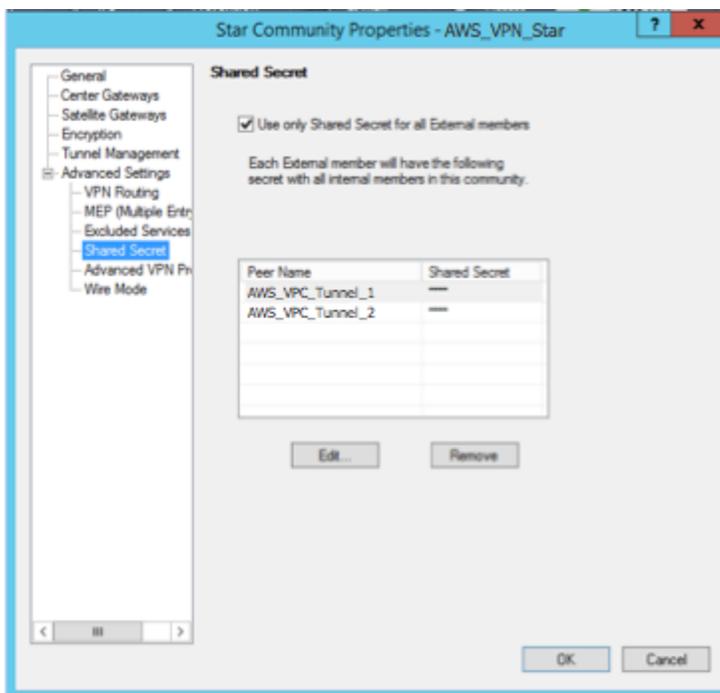
If you're using clusters, edit the topology and define the interfaces as cluster interfaces. Use the IP addresses that are specified in the configuration file.

To create and configure the VPN community, IKE, and IPsec settings

In this step, you create a VPN community on your Check Point gateway, to which you add the network objects (interoperable devices) for each tunnel. You also configure the Internet Key Exchange (IKE) and IPsec settings.

1. From your gateway properties, choose **IPsec VPN** in the category pane.
2. Choose **Communities, New, Star Community**.
3. Provide a name for your community (for example, `AWS_VPN_Star`), and then choose **Center Gateways** in the category pane.
4. Choose **Add**, and add your gateway or cluster to the list of participant gateways.
5. In the category pane, choose **Satellite Gateways, Add**, and then add the interoperable devices that you created earlier (`AWS_VPC_Tunnel_1` and `AWS_VPC_Tunnel_2`) to the list of participant gateways.
6. In the category pane, choose **Encryption**. In the **Encryption Method** section, choose **IKEv1 only**. In the **Encryption Suite** section, choose **Custom, Custom Encryption**.
7. In the dialog box, configure the encryption properties as follows, and choose **OK** when you're done:
 - IKE Security Association (Phase 1) Properties:
 - **Perform key exchange encryption with: AES-128**
 - **Perform data integrity with: SHA-1**
 - IPsec Security Association (Phase 2) Properties:

- **Perform IPsec data encryption with: AES-128**
 - **Perform data integrity with: SHA-1**
8. In the category pane, choose **Tunnel Management**. Choose **Set Permanent Tunnels, On all tunnels in the community**. In the **VPN Tunnel Sharing** section, choose **One VPN tunnel per Gateway pair**.
 9. In the category pane, expand **Advanced Settings**, and choose **Shared Secret**.
 10. Select the peer name for the first tunnel, choose **Edit**, and then enter the pre-shared key as specified in the configuration file in the IPsec Tunnel #1 section.
 11. Select the peer name for the second tunnel, choose **Edit**, and then enter the pre-shared key as specified in the configuration file in the IPsec Tunnel #2 section.



12. Still in the **Advanced Settings** category, choose **Advanced VPN Properties**, configure the properties as follows, and then choose **OK** when you're done:
 - IKE (Phase 1):
 - **Use Diffie-Hellman group: Group 2**
 - **Renegotiate IKE security associations every 480 minutes**
 - IPsec (Phase 2):
 - **Choose Use Perfect Forward Secrecy**
 - **Use Diffie-Hellman group: Group 2**

- **Renegotiate IPsec security associations every 3600 seconds**

To create firewall rules

In this step, you configure a policy with firewall rules and directional match rules that allow communication between the VPC and the local network. You then install the policy on your gateway.

1. In the SmartDashboard, choose **Global Properties** for your gateway. In the category pane, expand **VPN**, and choose **Advanced**.
2. Choose **Enable VPN Directional Match in VPN Column**, and save your changes.
3. In the SmartDashboard, choose **Firewall**, and create a policy with the following rules:
 - Allow the VPC subnet to communicate with the local network over the required protocols.
 - Allow the local network to communicate with the VPC subnet over the required protocols.
4. Open the context menu for the cell in the VPN column, and choose **Edit Cell**.
5. In the **VPN Match Conditions** dialog box, choose **Match traffic in this direction only**. Create the following directional match rules by choosing **Add** for each, and choose **OK** when you're done:
 - `internal_clear > VPN community` (The VPN star community that you created earlier, for example, `AWS_VPN_Star`)
 - `VPN community > VPN community`
 - `VPN community > internal_clear`
6. In the SmartDashboard, choose **Policy, Install**.
7. In the dialog box, choose your gateway and choose **OK** to install the policy.

To modify the `tunnel_keepalive_method` property

Your Check Point gateway can use Dead Peer Detection (DPD) to identify when an IKE association is down. To configure DPD for a permanent tunnel, the permanent tunnel must be configured in the AWS VPN community (refer to Step 8).

By default, the `tunnel_keepalive_method` property for a VPN gateway is set to `tunnel_test`. You must change the value to `dpd`. Each VPN gateway in the VPN community that requires DPD monitoring must be configured with the `tunnel_keepalive_method` property, including any 3rd party VPN gateway. You cannot configure different monitoring mechanisms for the same gateway.

You can update the `tunnel_keepalive_method` property using the GuiDBedit tool.

1. Open the Check Point SmartDashboard, and choose **Security Management Server, Domain Management Server**.
2. Choose **File, Database Revision Control...** and create a revision snapshot.
3. Close all SmartConsole windows, such as the SmartDashboard, SmartView Tracker, and SmartView Monitor.
4. Start the GuiDBedit tool. For more information, see the [Check Point Database Tool](#) article on the Check Point Support Center.
5. Choose **Security Management Server, Domain Management Server**.
6. In the upper left pane, choose **Table, Network Objects, network_objects**.
7. In the upper right pane, select the relevant **Security Gateway, Cluster** object.
8. Press CTRL+F, or use the **Search** menu to search for the following:
`tunnel_keepalive_method`.
9. In the lower pane, open the context menu for `tunnel_keepalive_method`, and choose **Edit...** Choose **dpd** and then choose **OK**.
10. Repeat steps 7 through 9 for each gateway that's part of the AWS VPN Community.
11. Choose **File, Save All**.
12. Close the GuiDBedit tool.
13. Open the Check Point SmartDashboard, and choose **Security Management Server, Domain Management Server**.
14. Install the policy on the relevant **Security Gateway, Cluster** object.

For more information, see the [New VPN features in R77.10](#) article on the Check Point Support Center.

To enable TCP MSS clamping

TCP MSS clamping reduces the maximum segment size of TCP packets to prevent packet fragmentation.

1. Navigate to the following directory: C:\Program Files (x86)\CheckPoint\SmartConsole\R77.10\PROGRAM\.
2. Open the Check Point Database Tool by running the GuiDBEdit.exe file.
3. Choose **Table, Global Properties, properties**.
4. For fw_clamp_tcp_mss, choose **Edit**. Change the value to true and choose **OK**.

To verify the tunnel status

You can verify the tunnel status by running the following command from the command line tool in expert mode.

```
vpn tunnelutil
```

In the options that display, choose **1** to verify the IKE associations and **2** to verify the IPsec associations.

You can also use the Check Point Smart Tracker Log to verify that packets over the connection are being encrypted. For example, the following log indicates that a packet to the VPC was sent over tunnel 1 and was encrypted.

Log Info		Rule	
Product	Security Gateway/Management	Action	Encrypt
Date	4Nov2015	Rule	4
Time	9:42:01	Current Rule Number	4-Standard
Number	21254	Rule Name	---
Type	Log	User	---
Origin	cpgw-997695	More	
Traffic		Rule UID	{0AA18015-FF7B-4650-B0CE-3989E658CF04}
Source	Management_PC (192.168.1.116)	Community	AWS_VPN_Star
Destination	10.28.13.28	Encryption Scheme	IKE
Service	---	Data Encryption Methods	ESP: AES-128 + SHA1 + PFS (group 2)
Protocol	icmp	VPN Peer Gateway	AWS_VPC_Tunnel_1 (54.84.169.196)
Interface	eth0	Subproduct	VPN
Source Port	---	VPN Feature	VPN
Policy		Product Family	Network
Policy Name	Standard	Information	service_id: icmp-proto ICMP: Echo Request ICMP Type: 8 ICMP Code: 0
Policy Date	Tue Nov 03 11:33:45 2015		
Policy Management	cpgw-997695		

SonicWALL

The following procedure demonstrates how to configure the VPN tunnels on the SonicWALL device using the SonicOS management interface.

To configure the tunnels

1. Open the SonicWALL SonicOS management interface.
2. In the left pane, choose **VPN, Settings**. Under **VPN Policies**, choose **Add...**
3. In the VPN policy window on the **General** tab, complete the following information:
 - **Policy Type:** Choose **Tunnel Interface**.
 - **Authentication Method:** Choose **IKE using Preshared Secret**.
 - **Name:** Enter a name for the VPN policy. We recommend that you use the name of the VPN ID, as provided in the configuration file.
 - **IPsec Primary Gateway Name or Address:** Enter the IP address of the virtual private gateway as provided in the configuration file (for example, 72.21.209.193).
 - **IPsec Secondary Gateway Name or Address:** Leave the default value.

- **Shared Secret:** Enter the pre-shared key as provided in the configuration file, and enter it again in **Confirm Shared Secret**.
 - **Local IKE ID:** Enter the IPv4 address of the customer gateway (the SonicWALL device).
 - **Peer IKE ID:** Enter the IPv4 address of the virtual private gateway.
4. On the **Network** tab, complete the following information:
- Under **Local Networks**, choose **Any address**. We recommend this option to prevent connectivity issues from your local network.
 - Under **Remote Networks**, choose **Choose a destination network from list**. Create an address object with the CIDR of your VPC in AWS.
5. On the **Proposals** tab, complete the following information:
- Under **IKE (Phase 1) Proposal**, do the following:
 - **Exchange:** Choose **Main Mode**.
 - **DH Group:** Enter a value for the Diffie-Hellman group (for example, 2).
 - **Encryption:** Choose **AES-128** or **AES-256**.
 - **Authentication:** Choose **SHA1** or **SHA256**.
 - **Life Time:** Enter 28800.
 - Under **IKE (Phase 2) Proposal**, do the following:
 - **Protocol:** Choose **ESP**.
 - **Encryption:** Choose **AES-128** or **AES-256**.
 - **Authentication:** Choose **SHA1** or **SHA256**.
 - Select the **Enable Perfect Forward Secrecy** check box, and choose the Diffie-Hellman group.
 - **Life Time:** Enter 3600.

 **Important**

If you created your virtual private gateway before October 2015, you must specify Diffie-Hellman group 2, AES-128, and SHA1 for both phases.

6. On the **Advanced** tab, complete the following information:
- **Select Enable Keep Alive**.

- Select **Enable Phase2 Dead Peer Detection** and enter the following:
 - For **Dead Peer Detection Interval**, enter 60 (this is the minimum that the SonicWALL device accepts).
 - For **Failure Trigger Level**, enter 3.
 - For **VPN Policy bound to**, select **Interface X1**. This is the interface that's typically designated for public IP addresses.
- 7. Choose **OK**. On the **Settings** page, the **Enable** check box for the tunnel should be selected by default. A green dot indicates that the tunnel is up.

Additional information for Cisco devices

Some Cisco ASAs only support Active/Standby mode. When you use these Cisco ASAs, you can have only one active tunnel at a time. The other standby tunnel becomes active if the first tunnel becomes unavailable. With this redundancy, you should always have connectivity to your VPC through one of the tunnels.

Cisco ASAs from version 9.7.1 and later support Active/Active mode. When you use these Cisco ASAs, you can have both tunnels active at the same time. With this redundancy, you should always have connectivity to your VPC through one of the tunnels.

For Cisco devices, you must do the following:

- Configure the outside interface.
- Ensure that the Crypto ISAKMP Policy Sequence number is unique.
- Ensure that the Crypto List Policy Sequence number is unique.
- Ensure that the Crypto IPsec Transform Set and the Crypto ISAKMP Policy Sequence are harmonious with any other IPsec tunnels that are configured on the device.
- Ensure that the SLA monitoring number is unique.
- Configure all internal routing that moves traffic between the customer gateway device and your local network.

Testing

For more information about testing your Site-to-Site VPN connection, see [Test a Site-to-Site VPN connection](#).

Example customer gateway device configurations for dynamic routing (BGP)

Topics

- [Example configuration files](#)
- [User interface procedures for dynamic routing](#)
- [Additional information for Cisco devices](#)
- [Additional information for Juniper devices](#)
- [Testing](#)

Example configuration files

To download a sample configuration file with values specific to your Site-to-Site VPN connection configuration, use the Amazon VPC console, the AWS command line or the Amazon EC2 API. For more information, see [Step 6: Download the configuration file](#).

You can also download generic example configuration files for dynamic routing that do not include values specific to your Site-to-Site VPN connection configuration: [dynamic-routing-examples.zip](#)

The files use placeholder values for some components. For example, they use:

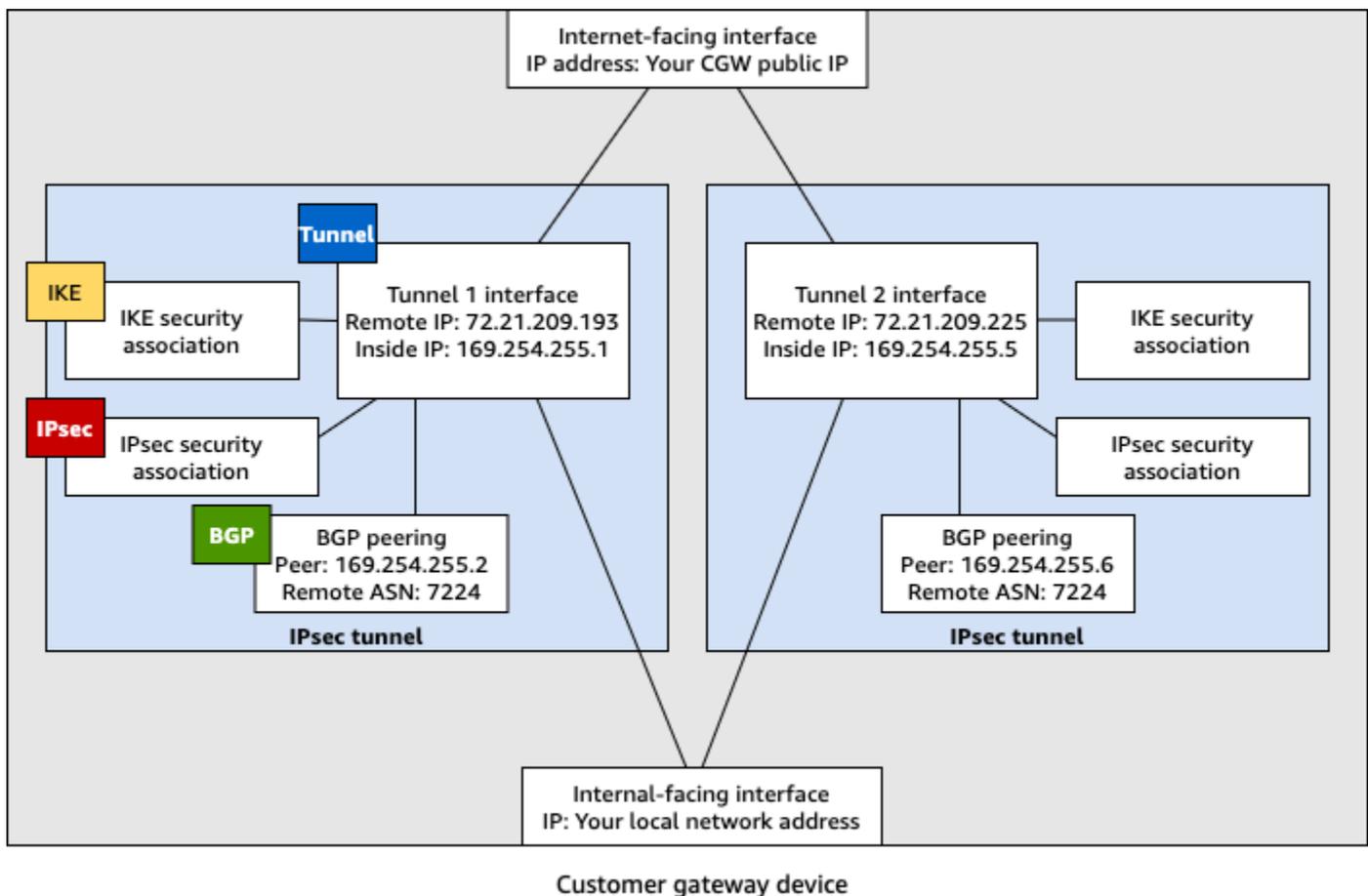
- Example values for the VPN connection ID, customer gateway ID and virtual private gateway ID
- Placeholders for the remote (outside) IP address AWS endpoints (*AWS_ENDPOINT_1* and *AWS_ENDPOINT_2*)
- A placeholder for the IP address for the internet-routable external interface on the customer gateway device (*your-cgw-ip-address*)
- A placeholder for the pre-shared key value (pre-shared-key)
- Example values for the tunnel inside IP addresses.
- Example values for MTU setting.

Note

MTU settings provided in the sample configuration files are examples only. Please refer to [Best practices for your customer gateway device](#) for information on setting the optimal MTU value for your situation.

In addition to providing placeholder values, the files specify the minimum requirements for a Site-to-Site VPN connection of AES128, SHA1, and Diffie-Hellman group 2 in most AWS Regions, and AES128, SHA2, and Diffie-Hellman group 14 in the AWS GovCloud Regions. They also specify pre-shared keys for [authentication](#). You must modify the example configuration file to take advantage of additional security algorithms, Diffie-Hellman groups, private certificates, and IPv6 traffic.

The following diagram provides an overview of the different components that are configured on the customer gateway device. It includes example values for the tunnel interface IP addresses.



User interface procedures for dynamic routing

The following are some example procedures for configuring a customer gateway device using its user interface (if available).

Check Point

The following are steps for configuring a Check Point Security Gateway device running R77.10 or above, using the Gaia web portal and Check Point SmartDashboard. You can also refer to the [Amazon Web Services \(AWS\) VPN BGP](#) article on the Check Point Support Center.

To configure the tunnel interface

The first step is to create the VPN tunnels and provide the private (inside) IP addresses of the customer gateway and virtual private gateway for each tunnel. To create the first tunnel, use the information provided under the IPsec Tunnel #1 section of the configuration file. To create the second tunnel, use the values provided in the IPsec Tunnel #2 section of the configuration file.

1. Connect to your security gateway over SSH. If you're using the non-default shell, change to clish by running the following command: `clish`
2. Set the customer gateway ASN (the ASN that was provided when the customer gateway was created in AWS) by running the following command.

```
set as 65000
```

3. Create the tunnel interface for the first tunnel, using the information provided under the IPsec Tunnel #1 section of the configuration file. Provide a unique name for your tunnel, such as `AWS_VPC_Tunnel_1`.

```
add vpn tunnel 1 type numbered local 169.254.44.234 remote 169.254.44.233
peer AWS_VPC_Tunnel_1
set interface vpnt1 state on
set interface vpnt1 mtu 1436
```

4. Repeat these commands to create the second tunnel, using the information provided under the IPsec Tunnel #2 section of the configuration file. Provide a unique name for your tunnel, such as `AWS_VPC_Tunnel_2`.

```
add vpn tunnel 1 type numbered local 169.254.44.38 remote 169.254.44.37
peer AWS_VPC_Tunnel_2
set interface vpnt2 state on
set interface vpnt2 mtu 1436
```

5. Set the virtual private gateway ASN.

```
set bgp external remote-as 7224 on
```

6. Configure the BGP for the first tunnel, using the information provided IPsec Tunnel #1 section of the configuration file.

```
set bgp external remote-as 7224 peer 169.254.44.233 on
set bgp external remote-as 7224 peer 169.254.44.233 holdtime 30
set bgp external remote-as 7224 peer 169.254.44.233 keepalive 10
```

7. Configure the BGP for the second tunnel, using the information provided IPsec Tunnel #2 section of the configuration file.

```
set bgp external remote-as 7224 peer 169.254.44.37 on
set bgp external remote-as 7224 peer 169.254.44.37 holdtime 30
set bgp external remote-as 7224 peer 169.254.44.37 keepalive 10
```

8. Save the configuration.

```
save config
```

To create a BGP policy

Next, create a BGP policy that allows the import of routes that are advertised by AWS. Then, configure your customer gateway to advertise its local routes to AWS.

1. In the Gaia WebUI, choose **Advanced Routing, Inbound Route Filters**. Choose **Add**, and select **Add BGP Policy (Based on AS)**.
2. For **Add BGP Policy**, select a value between 512 and 1024 in the first field, and enter the virtual private gateway ASN in the second field (for example, 7224).
3. Choose **Save**.

To advertise local routes

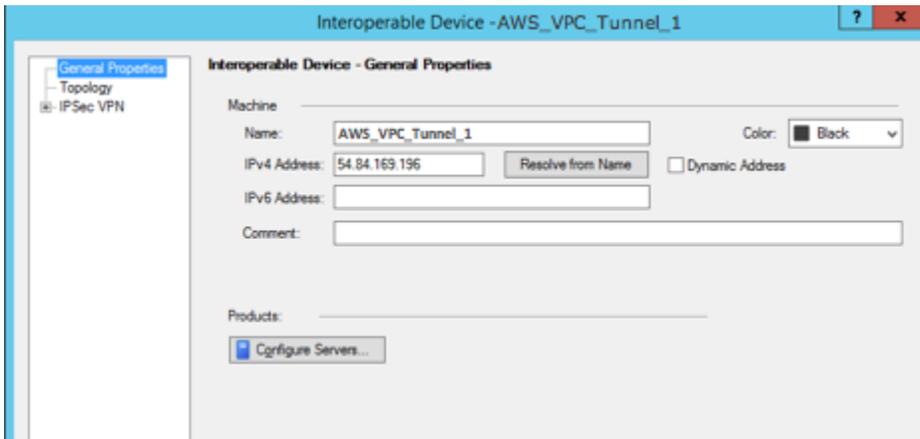
The following steps are for distributing local interface routes. You can also redistribute routes from different sources (for example, static routes, or routes obtained through dynamic routing protocols). For more information, see the [Gaia Advanced Routing R77 Versions Administration Guide](#).

1. In the Gaia WebUI, choose **Advanced Routing, Routing Redistribution**. Choose **Add Redistribution From** and then select **Interface**.
2. For **To Protocol**, select the virtual private gateway ASN (for example, 7224).
3. For **Interface**, select an internal interface. Choose **Save**.

To define a new network object

Next, create a network object for each VPN tunnel, specifying the public (outside) IP addresses for the virtual private gateway. You later add these network objects as satellite gateways for your VPN community. You also need to create an empty group to act as a placeholder for the VPN domain.

1. Open the Check Point SmartDashboard.
2. For **Groups**, open the context menu and choose **Groups, Simple Group**. You can use the same group for each network object.
3. For **Network Objects**, open the context (right-click) menu and choose **New, Interoperable Device**.
4. For **Name**, enter the name that you provided for your tunnel in step 1, for example, `AWS_VPC_Tunnel_1` or `AWS_VPC_Tunnel_2`.
5. For **IPv4 Address**, enter the outside IP address of the virtual private gateway provided in the configuration file, for example, `54.84.169.196`. Save your settings and close the dialog box.



6. In the left category pane, choose **Topology**.
7. In the **VPN Domain** section, choose **Manually defined**, and then browse to and select the empty simple group that you created in step 2. Choose **OK**.
8. Repeat these steps to create a second network object, using the information under the IPsec Tunnel #2 section of the configuration file.
9. Go to your gateway network object, open your gateway or cluster object, and choose **Topology**.
10. In the **VPN Domain** section, choose **Manually defined**, and then browse to and select the empty simple group that you created in step 2. Choose **OK**.

Note

You can keep any existing VPN domain that you've configured. However, ensure that the hosts and networks that are used or served by the new VPN connection are not declared in that VPN domain, especially if the VPN domain is automatically derived.

Note

If you're using clusters, edit the topology and define the interfaces as cluster interfaces. Use the IP addresses that are specified in the configuration file.

To create and configure the VPN community, IKE, and IPsec settings

Next, create a VPN community on your Check Point gateway, to which you add the network objects (interoperable devices) for each tunnel. You also configure the Internet Key Exchange (IKE) and IPsec settings.

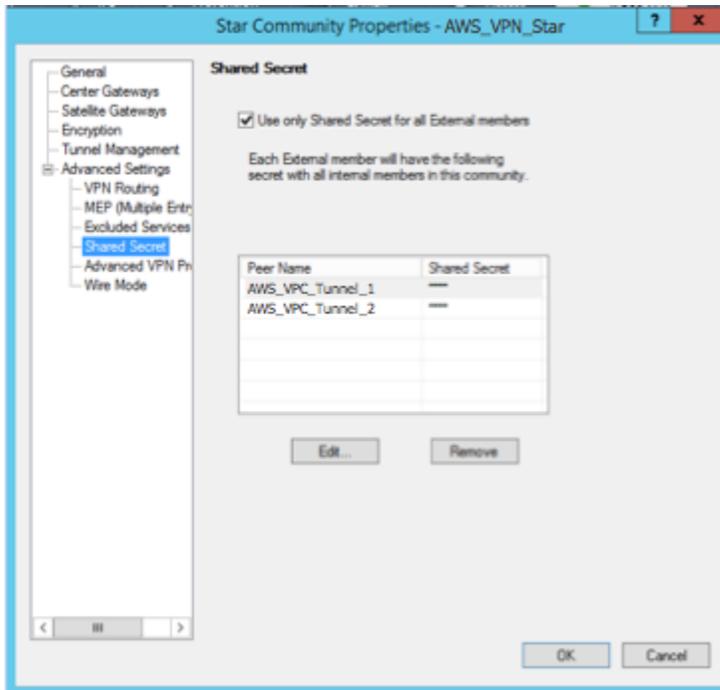
1. From your gateway properties, choose **IPSec VPN** in the category pane.
2. Choose **Communities, New, Star Community**.
3. Provide a name for your community (for example, `AWS_VPN_Star`), and then choose **Center Gateways** in the category pane.
4. Choose **Add**, and add your gateway or cluster to the list of participant gateways.
5. In the category pane, choose **Satellite Gateways, Add**, and add the interoperable devices that you created earlier (`AWS_VPC_Tunnel_1` and `AWS_VPC_Tunnel_2`) to the list of participant gateways.
6. In the category pane, choose **Encryption**. In the **Encryption Method** section, choose **IKEv1 for IPv4 and IKEv2 for IPv6**. In the **Encryption Suite** section, choose **Custom, Custom Encryption**.

Note

You must select the **IKEv1 for IPv4 and IKEv2 for IPv6** option for IKEv1 functionality.

7. In the dialog box, configure the encryption properties as follows, and then choose **OK** when you're done:
 - IKE Security Association (Phase 1) Properties:
 - **Perform key exchange encryption with: AES-128**
 - **Perform data integrity with: SHA-1**
 - IPsec Security Association (Phase 2) Properties:
 - **Perform IPsec data encryption with: AES-128**
 - **Perform data integrity with: SHA-1**
8. In the category pane, choose **Tunnel Management**. Choose **Set Permanent Tunnels, On all tunnels in the community**. In the **VPN Tunnel Sharing** section, choose **One VPN tunnel per Gateway pair**.
9. In the category pane, expand **Advanced Settings**, and choose **Shared Secret**.

10. Select the peer name for the first tunnel, choose **Edit**, and then enter the pre-shared key as specified in the configuration file in the IPsec Tunnel #1 section.
11. Select the peer name for the second tunnel, choose **Edit**, and then enter the pre-shared key as specified in the configuration file in the IPsec Tunnel #2 section.



12. Still in the **Advanced Settings** category, choose **Advanced VPN Properties**, configure the properties as follows, and then choose **OK** when you're done:
 - IKE (Phase 1):
 - **Use Diffie-Hellman group: Group 2 (1024 bit)**
 - **Renegotiate IKE security associations every 480 minutes**
 - IPsec (Phase 2):
 - **Choose Use Perfect Forward Secrecy**
 - **Use Diffie-Hellman group: Group 2 (1024 bit)**
 - **Renegotiate IPsec security associations every 3600 seconds**

To create firewall rules

Next, configure a policy with firewall rules and directional match rules that allow communication between the VPC and the local network. You then install the policy on your gateway.

1. In the SmartDashboard, choose **Global Properties** for your gateway. In the category pane, expand **VPN**, and choose **Advanced**.
2. Choose **Enable VPN Directional Match in VPN Column**, and choose **OK**.
3. In the SmartDashboard, choose **Firewall**, and create a policy with the following rules:
 - Allow the VPC subnet to communicate with the local network over the required protocols.
 - Allow the local network to communicate with the VPC subnet over the required protocols.
4. Open the context menu for the cell in the VPN column, and choose **Edit Cell**.
5. In the **VPN Match Conditions** dialog box, choose **Match traffic in this direction only**. Create the following directional match rules by choosing **Add** for each, and then choose **OK** when you're done:
 - `internal_clear > VPN community` (The VPN star community that you created earlier, for example, `AWS_VPN_Star`)
 - `VPN community > VPN community`
 - `VPN community > internal_clear`
6. In the SmartDashboard, choose **Policy, Install**.
7. In the dialog box, choose your gateway and choose **OK** to install the policy.

To modify the `tunnel_keepalive_method` property

Your Check Point gateway can use Dead Peer Detection (DPD) to identify when an IKE association is down. To configure DPD for a permanent tunnel, the permanent tunnel must be configured in the AWS VPN community.

By default, the `tunnel_keepalive_method` property for a VPN gateway is set to `tunnel_test`. You must change the value to `dpd`. Each VPN gateway in the VPN community that requires DPD monitoring must be configured with the `tunnel_keepalive_method` property, including any 3rd party VPN gateway. You cannot configure different monitoring mechanisms for the same gateway.

You can update the `tunnel_keepalive_method` property using the `GuiDBedit` tool.

1. Open the Check Point SmartDashboard, and choose **Security Management Server, Domain Management Server**.

2. Choose **File, Database Revision Control...** and create a revision snapshot.
3. Close all SmartConsole windows, such as the SmartDashboard, SmartView Tracker, and SmartView Monitor.
4. Start the GuiDBedit tool. For more information, see the [Check Point Database Tool](#) article on the Check Point Support Center.
5. Choose **Security Management Server, Domain Management Server**.
6. In the upper left pane, choose **Table, Network Objects, network_objects**.
7. In the upper right pane, select the relevant **Security Gateway, Cluster** object.
8. Press CTRL+F, or use the **Search** menu to search for the following:
tunnel_keepalive_method.
9. In the lower pane, open the context menu for tunnel_keepalive_method, and select **Edit...** Choose **dpd, OK**.
10. Repeat steps 7 through 9 for each gateway that's part of the AWS VPN Community.
11. Choose **File, Save All**.
12. Close the GuiDBedit tool.
13. Open the Check Point SmartDashboard, and choose **Security Management Server, Domain Management Server**.
14. Install the policy on the relevant **Security Gateway, Cluster** object.

For more information, see the [New VPN features in R77.10](#) article on the Check Point Support Center.

To enable TCP MSS clamping

TCP MSS clamping reduces the maximum segment size of TCP packets to prevent packet fragmentation.

1. Navigate to the following directory: C:\Program Files (x86)\CheckPoint\SmartConsole\R77.10\PROGRAM\.
2. Open the Check Point Database Tool by running the GuiDBEdit.exe file.
3. Choose **Table, Global Properties, properties**.
4. For fw_clamp_tcp_mss, choose **Edit**. Change the value to true and then choose **OK**.

To verify the tunnel status

You can verify the tunnel status by running the following command from the command line tool in expert mode.

```
vpn tunnelutil
```

In the options that display, choose **1** to verify the IKE associations and **2** to verify the IPsec associations.

You can also use the Check Point Smart Tracker Log to verify that packets over the connection are being encrypted. For example, the following log indicates that a packet to the VPC was sent over tunnel 1 and was encrypted.

Log Info		Rule	
Product	Security Gateway/Management	Action	Encrypt
Date	4Nov2015	Rule	4
Time	9:42:01	Current Rule Number	4-Standard
Number	21254	Rule Name	---
Type	Log	User	---
Origin	cpgw-997695	More	
Traffic		Rule UID	{0AA18015-FF7B-4650-B0CE-3989E658CF04}
Source	Management_PC (192.168.1.116)	Community	AWS_VPN_Star
Destination	10.28.13.28	Encryption Scheme	IKE
Service	---	Data Encryption Methods	ESP: AES-128 + SHA1 + PFS (group 2)
Protocol	icmp	VPN Peer Gateway	AWS_VPC_Tunnel_1 (54.84.169.196)
Interface	eth0	Subproduct	VPN
Source Port	---	VPN Feature	VPN
Policy		Product Family	Network
Policy Name	Standard	Information	service_jd: icmp-proto ICMP: Echo Request ICMP Type: 8 ICMP Code: 0
Policy Date	Tue Nov 03 11:33:45 2015		
Policy Management	cpgw-997695		

SonicWALL

You can configure a SonicWALL device using the SonicOS management interface. For more information about configuring the tunnels, see [User interface procedures for static routing](#).

You cannot configure BGP for the device using the management interface. Instead, use the command line instructions provided in the example configuration file, under the section named **BGP**.

Additional information for Cisco devices

Some Cisco ASAs only support Active/Standby mode. When you use these Cisco ASAs, you can have only one active tunnel at a time. The other standby tunnel becomes active if the first tunnel becomes unavailable. With this redundancy, you should always have connectivity to your VPC through one of the tunnels.

Cisco ASAs from version 9.7.1 and later support Active/Active mode. When you use these Cisco ASAs, you can have both tunnels active at the same time. With this redundancy, you should always have connectivity to your VPC through one of the tunnels.

For Cisco devices, you must do the following:

- Configure the outside interface.
- Ensure that the Crypto ISAKMP Policy Sequence number is unique.
- Ensure that the Crypto List Policy Sequence number is unique.
- Ensure that the Crypto IPsec Transform Set and the Crypto ISAKMP Policy Sequence are harmonious with any other IPsec tunnels that are configured on the device.
- Ensure that the SLA monitoring number is unique.
- Configure all internal routing that moves traffic between the customer gateway device and your local network.

Additional information for Juniper devices

The following information applies to the example configuration files for Juniper J-Series and SRX customer gateway devices.

- The outside interface is referred to as *ge-0/0/0.0*.
- The tunnel interface IDs are referred to as *st0.1* and *st0.2*.
- Ensure that you identify the security zone for the uplink interface (the configuration information uses the default 'untrust' zone).
- Ensure that you identify the security zone for the inside interface (the configuration information uses the default 'trust' zone).

Testing

For more information about testing your Site-to-Site VPN connection, see [Test a Site-to-Site VPN connection](#).

Configuring Windows Server as a customer gateway device

You can configure a server running Windows Server as a customer gateway device for your VPC. Use the following process whether you are running Windows Server on an EC2 instance in a VPC, or on your own server. The following procedures apply to Windows Server 2012 R2 and later.

Contents

- [Configuring your Windows instance](#)
- [Step 1: Create a VPN connection and configure your VPC](#)
- [Step 2: Download the configuration file for the VPN connection](#)
- [Step 3: Configure the Windows Server](#)
- [Step 4: Set up the VPN tunnel](#)
- [Step 5: Enable dead gateway detection](#)
- [Step 6: Test the VPN connection](#)

Configuring your Windows instance

If you are configuring Windows Server on an EC2 instance that you launched from a Windows AMI, do the following:

- Disable source/destination checking for the instance:
 1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 2. Select your Windows instance, and choose **Actions, Networking, Change source/destination check**. Choose **Stop**, and then choose **Save**.
- Update your adapter settings so that you can route traffic from other instances:
 1. Connect to your Windows instance. For more information, see [Connecting to your Windows instance](#).
 2. Open the Control Panel, and start the Device Manager.
 3. Expand the **Network adapters** node.

4. Select the network adapter (depending on the instance type, this might be Amazon Elastic Network Adapter or Intel 82599 Virtual Function), and choose **Action, Properties**.
 5. On the **Advanced** tab, disable the **IPv4 Checksum Offload**, **TCP Checksum Offload (IPv4)**, and **UDP Checksum Offload (IPv4)** properties, and then choose **OK**.
- Allocate an Elastic IP address to your account and associate it with the instance. For more information, see [Working with Elastic IP addresses](#). Take note of this address — you need it when you create the customer gateway in your VPC.
 - Ensure that the instance's security group rules allow outbound IPsec traffic. By default, a security group allows all outbound traffic. However, if the security group's outbound rules have been modified from their original state, you must create the following outbound custom protocol rules for IPsec traffic: IP protocol 50, IP protocol 51, and UDP 500.

Take note of the CIDR range of the network in which your Windows instance is located, for example, `172.31.0.0/16`.

Step 1: Create a VPN connection and configure your VPC

To create a VPN connection from your VPC, do the following:

1. Create a virtual private gateway and attach it to your VPC. For more information, see [Create a virtual private gateway](#).
2. Create a VPN connection and new customer gateway. For the customer gateway, specify the public IP address of your Windows Server. For the VPN connection, choose static routing, and then enter the CIDR range for your network in which the Windows Server is located, for example, `172.31.0.0/16`. For more information, see [Step 5: Create a VPN connection](#).

After you create the VPN connection, configure the VPC to enable communication over the VPN connection.

To configure your VPC

- Create a private subnet in your VPC (if you don't have one already) for launching instances to communicate with the Windows Server. For more information, see [Creating a subnet in your VPC](#).

Note

A private subnet is a subnet that does not have a route to an internet gateway. The routing for this subnet is described in the next item.

- Update your route tables for the VPN connection:
 - Add a route to your private subnet's route table with the virtual private gateway as the target, and the Windows Server's network (CIDR range) as the destination. For more information, see [Adding and removing routes from a route table](#) in the *Amazon VPC User Guide*.
 - Enable route propagation for the virtual private gateway. For more information, see [\(Virtual private gateway\) Enable route propagation in your route table](#).
- Create a security group for your instances that allows communication between your VPC and network:
 - Add rules that allow inbound RDP or SSH access from your network. This enables you to connect to instances in your VPC from your network. For example, to allow computers in your network to access Linux instances in your VPC, create an inbound rule with a type of SSH, and the source set to the CIDR range of your network (for example, 172.31.0.0/16). For more information, see [Security groups for your VPC](#) in the *Amazon VPC User Guide*.
 - Add a rule that allows inbound ICMP access from your network. This enables you to test your VPN connection by pinging an instance in your VPC from your Windows Server.

Step 2: Download the configuration file for the VPN connection

You can use the Amazon VPC console to download a Windows Server configuration file for your VPN connection.

To download the configuration file

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN Connections**.
3. Select your VPN connection and choose **Download Configuration**.
4. Select **Microsoft** as the vendor, **Windows Server** as the platform, and **2012 R2** as the software. Choose **Download**. You can open the file or save it.

The configuration file contains a section of information similar to the following example. You see this information presented twice, one time for each tunnel.

```
vgw-1a2b3c4d Tunnel1
-----
Local Tunnel Endpoint:      203.0.113.1
Remote Tunnel Endpoint:    203.83.222.237
Endpoint 1:                 [Your_Static_Route_IP_Prefix]
Endpoint 2:                 [Your_VPC_CIDR_Block]
Preshared key:              xCjNLSLoCmKsawkdoR9yX6GsEXAMPLE
```

Local Tunnel Endpoint

The IP address that you specified for the customer gateway when you created the VPN connection.

Remote Tunnel Endpoint

One of two IP addresses for the virtual private gateway that terminates the VPN connection on the AWS side of the connection.

Endpoint 1

The IP prefix that you specified as a static route when you created the VPN connection. These are the IP addresses in your network that are allowed to use the VPN connection to access your VPC.

Endpoint 2

The IP address range (CIDR block) of the VPC that is attached to the virtual private gateway (for example 10.0.0.0/16).

Preshared key

The pre-shared key that is used to establish the IPsec VPN connection between Local Tunnel Endpoint and Remote Tunnel Endpoint.

We suggest that you configure both tunnels as part of the VPN connection. Each tunnel connects to a separate VPN concentrator on the Amazon side of the VPN connection. Although only one tunnel at a time is up, the second tunnel automatically establishes itself if the first tunnel goes down. Having redundant tunnels ensure continuous availability in the case of a device failure. Because only one tunnel is available at a time, the Amazon VPC console indicates that one tunnel is down. This is expected behavior, so there's no action required from you.

With two tunnels configured, if a device failure occurs within AWS, your VPN connection automatically fails over to the second tunnel of the virtual private gateway within a matter of minutes. When you configure your customer gateway device, it's important that you configure both tunnels.

 **Note**

From time to time, AWS performs routine maintenance on the virtual private gateway. This maintenance might disable one of the two tunnels of your VPN connection for a brief period of time. Your VPN connection automatically fails over to the second tunnel while we perform this maintenance.

Additional information regarding the Internet Key Exchange (IKE) and IPsec Security Associations (SA) is presented in the downloaded configuration file.

```
MainModeSecMethods:    DHGroup2-AES128-SHA1
MainModeKeyLifetime:   480min,0sess
QuickModeSecMethods:  ESP:SHA1-AES128+60min+100000kb
QuickModePFS:         DHGroup2
```

MainModeSecMethods

The encryption and authentication algorithms for the IKE SA. These are the suggested settings for the VPN connection, and are the default settings for Windows Server IPsec VPN connections.

MainModeKeyLifetime

The IKE SA key lifetime. This is the suggested setting for the VPN connection, and is the default setting for Windows Server IPsec VPN connections.

QuickModeSecMethods

The encryption and authentication algorithms for the IPsec SA. These are the suggested settings for the VPN connection, and are the default settings for Windows Server IPsec VPN connections.

QuickModePFS

We suggest that you use master key perfect forward secrecy (PFS) for your IPsec sessions.

Step 3: Configure the Windows Server

Before you set up the VPN tunnel, you must install and configure Routing and Remote Access Services on Windows Server. That allows remote users to access resources on your network.

To install Routing and Remote Access Services

1. Log on to your Windows Server.
2. Go to the **Start** menu, and choose **Server Manager**.
3. Install Routing and Remote Access Services:
 - a. From the **Manage** menu, choose **Add Roles and Features**.
 - b. On the **Before You Begin** page, verify that your server meets the prerequisites, and then choose **Next**.
 - c. Choose **Role-based or feature-based installation**, and then choose **Next**.
 - d. Choose **Select a server from the server pool**, select your Windows Server, and then choose **Next**.
 - e. Select **Network Policy and Access Services** in the list. In the dialog box that displays, choose **Add Features** to confirm the features that are required for this role.
 - f. In the same list, choose **Remote Access, Next**.
 - g. On the **Select features** page, choose **Next**.
 - h. On the **Network Policy and Access Services** page, choose **Next**.
 - i. On the **Remote Access** page, choose **Next**. On the next page, select **DirectAccess and VPN (RAS)**. In the dialog box that displays, choose **Add Features** to confirm the features that are required for this role service. In the same list, select **Routing**, and then choose **Next**.
 - j. On the **Web Server Role (IIS)** page, choose **Next**. Leave the default selection, and choose **Next**.
 - k. Choose **Install**. When the installation completes, choose **Close**.

To configure and enable Routing and Remote Access Server

1. On the dashboard, choose **Notifications** (the flag icon). There should be a task to complete the post-deployment configuration. Choose the **Open the Getting Started Wizard** link.
2. Choose **Deploy VPN only**.

3. In the **Routing and Remote Access** dialog box, choose the server name, choose **Action**, and then select **Configure and Enable Routing and Remote Access**.
4. In the **Routing and Remote Access Server Setup Wizard**, on the first page, choose **Next**.
5. On the **Configuration** page, choose **Custom Configuration, Next**.
6. Choose **LAN routing, Next, Finish**.
7. When prompted by the **Routing and Remote Access** dialog box, choose **Start service**.

Step 4: Set up the VPN tunnel

You can configure the VPN tunnel by running the netsh scripts included in the downloaded configuration file, or by using the Windows Server user interface.

Important

We suggest that you use master key perfect forward secrecy (PFS) for your IPsec sessions. If you choose to run the netsh script, it includes a parameter to enable PFS (`qmpfs=dhgroup2`). You cannot enable PFS using the Windows user interface — you must enable it using the command line.

Options

- [Option 1: Run the netsh script](#)
- [Option 2: Use the Windows Server user interface](#)

Option 1: Run the netsh script

Copy the netsh script from the downloaded configuration file and replace the variables. The following is an example script.

```
netsh advfirewall consec add rule Name="vgw-1a2b3c4d Tunnel 1" ^
Enable=Yes Profile=any Type=Static Mode=Tunnel ^
LocalTunnelEndpoint=Windows_Server_Private_IP_address ^
RemoteTunnelEndpoint=203.83.222.236 Endpoint1=Your_Static_Route_IP_Prefix ^
Endpoint2=Your_VPC_CIDR_Block Protocol=Any Action=RequireInClearOut ^
Auth1=ComputerPSK Auth1PSK=xCjNLSLoCmKsawkcdor9yX6GsEXAMPLE ^
QMSecMethods=ESP:SHA1-AES128+60min+100000kb ^
```

```
ExemptIPsecProtectedConnections=No ApplyAuthz=No QMPFS=dhgroup2
```

Name: You can replace the suggested name (vgw-1a2b3c4d Tunnel 1) with a name of your choice.

LocalTunnelEndpoint: Enter the private IP address of the Windows Server on your network.

Endpoint1: The CIDR block of your network on which the Windows Server resides, for example, 172.31.0.0/16. Surround this value with double quotes (").

Endpoint2: The CIDR block of your VPC or a subnet in your VPC, for example, 10.0.0.0/16. Surround this value with double quotes (").

Run the updated script in a command prompt window on your Windows Server. (The ^ enables you to cut and paste wrapped text at the command line.) To set up the second VPN tunnel for this VPN connection, repeat the process using the second netsh script in the configuration file.

When you are done, go to [Configure the Windows firewall](#).

For more information about the netsh parameters, see [Netsh AdvFirewall Consec Commands](#) in the *Microsoft TechNet Library*.

Option 2: Use the Windows Server user interface

You can also use the Windows Server user interface to set up the VPN tunnel.

Important

You can't enable master key perfect forward secrecy (PFS) using the Windows Server user interface. You must enable PFS using the command line, as described in [Enable master key perfect forward secrecy](#).

Tasks

- [Configure a security rule for a VPN tunnel](#)
- [Confirm the tunnel configuration](#)
- [Enable master key perfect forward secrecy](#)
- [Configure the Windows firewall](#)

Configure a security rule for a VPN tunnel

In this section, you configure a security rule on your Windows Server to create a VPN tunnel.

To configure a security rule for a VPN tunnel

1. Open Server Manager, choose **Tools**, and then select **Windows Defender Firewall with Advanced Security**.
2. Select **Connection Security Rules**, choose **Action**, and then **New Rule**.
3. In the **New Connection Security Rule** wizard, on the **Rule Type** page, choose **Tunnel**, and then choose **Next**.
4. On the **Tunnel Type** page, under **What type of tunnel would you like to create**, choose **Custom configuration**. Under **Would you like to exempt IPsec-protected connections from this tunnel**, leave the default value checked (**No. Send all network traffic that matches this connection security rule through the tunnel**), and then choose **Next**.
5. On the **Requirements** page, choose **Require authentication for inbound connections. Do not establish tunnels for outbound connections**, and then choose **Next**.
6. On the **Tunnel Endpoints** page, under **Which computers are in Endpoint 1**, choose **Add**. Enter the CIDR range of your network (behind your Windows Server customer gateway device; for example, 172.31.0.0/16), and then choose **OK**. The range can include the IP address of your customer gateway device.
7. Under **What is the local tunnel endpoint (closest to computer in Endpoint 1)**, choose **Edit**. In the **IPv4 address** field, enter the private IP address of your Windows Server, and then choose **OK**.
8. Under **What is the remote tunnel endpoint (closest to computers in Endpoint 2)**, choose **Edit**. In the **IPv4 address** field, enter the IP address of the virtual private gateway for Tunnel 1 from the configuration file (see `Remote Tunnel Endpoint`), and then choose **OK**.

Important

If you are repeating this procedure for Tunnel 2, be sure to select the endpoint for Tunnel 2.

9. Under **Which computers are in Endpoint 2**, choose **Add**. In the **This IP address or subnet field**, enter the CIDR block of your VPC, and then choose **OK**.

⚠ Important

You must scroll in the dialog box until you locate **Which computers are in Endpoint 2**. Do not choose **Next** until you have completed this step, or you won't be able to connect to your server.

The screenshot shows the 'New Connection Security Rule Wizard' dialog box, specifically the 'Tunnel Endpoints' step. The title bar reads 'New Connection Security Rule Wizard'. The main heading is 'Tunnel Endpoints' with the instruction 'Specify the endpoints for the IPsec tunnel defined by this rule.' On the left, a 'Steps' pane shows 'Tunnel Endpoints' as the current step. The main area is divided into sections: 'Which computers are in Endpoint 1?' with a list box containing '172.31.0.0/16' and buttons for 'Add...', 'Edit...', and 'Remove'; 'What is the local tunnel endpoint (closest to computers in Endpoint 1)?' with fields for IPv4 address (172.31.13.36) and IPv6 address, and an 'Edit...' button; a checkbox for 'Apply IPsec tunnel authorization as specified on the IPsec Settings tab of Windows Firewall with Advanced Security Properties'; 'What is the remote tunnel endpoint (closest to computers in Endpoint 2)?' with fields for IPv4 address (54.240.204.89) and IPv6 address, and an 'Edit...' button; and 'Which computers are in Endpoint 2?' with a list box containing '10.0.0.0/16' and an 'Add...' button. At the bottom are '< Back', 'Next >', and 'Cancel' buttons.

10. Confirm that all of the settings you've specified are correct and then choose **Next**.
11. On the **Authentication Method** page, select **Advanced** and choose **Customize**.
12. Under **First authentication methods**, choose **Add**.
13. Select **Preshared key**, enter the pre-shared key value from the configuration file and then choose **OK**.

⚠ Important

If you are repeating this procedure for Tunnel 2, be sure to select the pre-shared key for Tunnel 2.

14. Ensure that **First authentication is optional** is not selected, and choose **OK**.
15. Choose **Next**.
16. On the **Profile** page, select all three check boxes: **Domain**, **Private**, and **Public**. Choose **Next**.
17. On the **Name** page, enter a name for your connection rule; for example, VPN to Tunnel 1, and then choose **Finish**.

Repeat the preceding procedure, specifying the data for Tunnel 2 from your configuration file.

After you've finished, you'll have two tunnels configured for your VPN connection.

Confirm the tunnel configuration

To confirm the tunnel configuration

1. Open Server Manager, choose **Tools**, select **Windows Firewall with Advanced Security**, and then select **Connection Security Rules**.
2. Verify the following for both tunnels:
 - **Enabled** is Yes
 - **Endpoint 1** is the CIDR block for your network
 - **Endpoint 2** is the CIDR block of your VPC
 - **Authentication mode** is Require inbound and clear outbound
 - **Authentication method** is Custom
 - **Endpoint 1 port** is Any
 - **Endpoint 2 port** is Any
 - **Protocol** is Any
3. Select the first rule and choose **Properties**.
4. On the **Authentication** tab, under **Method**, choose **Customize**. Verify that **First authentication methods** contains the correct pre-shared key from your configuration file for the tunnel, and then choose **OK**.

5. On the **Advanced** tab, verify that **Domain**, **Private**, and **Public** are all selected.
6. Under **IPsec tunneling**, choose **Customize**. Verify the following IPsec tunneling settings, and then choose **OK** and **OK** again to close the dialog box.
 - **Use IPsec tunneling** is selected.
 - **Local tunnel endpoint (closest to Endpoint 1)** contains the IP address of your Windows Server. If your customer gateway device is an EC2 instance, this is the instance's private IP address.
 - **Remote tunnel endpoint (closest to Endpoint 2)** contains the IP address of the virtual private gateway for this tunnel.
7. Open the properties for your second tunnel. Repeat steps 4 to 7 for this tunnel.

Enable master key perfect forward secrecy

You can enable master key perfect forward secrecy by using the command line. You cannot enable this feature using the user interface.

To enable master key perfect forward secrecy

1. In your Windows Server, open a new command prompt window.
2. Enter the following command, replacing `rule_name` with the name that you gave the first connection rule.

```
netsh advfirewall consec set rule name="rule_name" new QMPFS=dhgroup2
QMSecMethods=ESP:SHA1-AES128+60min+100000kb
```

3. Repeat step 2 for the second tunnel, this time replacing `rule_name` with the name that you gave the second connection rule.

Configure the Windows firewall

After setting up your security rules on your server, configure some basic IPsec settings to work with the virtual private gateway.

To configure the Windows firewall

1. Open Server Manager, choose **Tools**, select **Windows Defender Firewall with Advanced Security**, and then choose **Properties**.

2. On the **IPsec Settings** tab, under **IPsec exemptions**, verify that **Exempt ICMP from IPsec** is **No (default)**. Verify that **IPsec tunnel authorization** is **None**.
3. Under **IPsec defaults**, choose **Customize**.
4. Under **Key exchange (Main Mode)**, select **Advanced** and then choose **Customize**.
5. In **Customize Advanced Key Exchange Settings**, under **Security methods**, verify that the following default values are used for the first entry:
 - Integrity: SHA-1
 - Encryption: AES-CBC 128
 - Key exchange algorithm: Diffie-Hellman Group 2
 - Under **Key lifetimes**, verify that **Minutes** is 480 and **Sessions** is 0.

These settings correspond to these entries in the configuration file.

```
MainModeSecMethods: DHGroup2-AES128-SHA1,DHGroup2-3DES-SHA1
MainModeKeyLifetime: 480min,0sec
```

6. Under **Key exchange options**, select **Use Diffie-Hellman for enhanced security**, and then choose **OK**.
7. Under **Data protection (Quick Mode)**, select **Advanced**, and then choose **Customize**.
8. Select **Require encryption for all connection security rules that use these settings**.
9. Under **Data integrity and encryption**, leave the default values:
 - Protocol: ESP
 - Integrity: SHA-1
 - Encryption: AES-CBC 128
 - Lifetime: 60 minutes

These values correspond to the following entry from the configuration file.

```
QuickModeSecMethods:
ESP:SHA1-AES128+60min+100000kb
```

10. Choose **OK** to return to the **Customize IPsec Settings** dialog box and choose **OK** again to save the configuration.

Step 5: Enable dead gateway detection

Next, configure TCP to detect when a gateway becomes unavailable. You can do this by modifying this registry key: `HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters`. Do not perform this step until you've completed the preceding sections. After you change the registry key, you must reboot the server.

To enable dead gateway detection

1. From your Windows Server, launch the command prompt or a PowerShell session, and enter **regedit** to start Registry Editor.
2. Expand **HKEY_LOCAL_MACHINE**, expand **SYSTEM**, expand **CurrentControlSet**, expand **Services**, expand **Tcpip**, and then expand **Parameters**.
3. From the **Edit** menu, select **New** and select **DWORD (32-bit) Value**.
4. Enter the name **EnableDeadGWDetect**.
5. Select **EnableDeadGWDetect** and choose **Edit, Modify**.
6. In **Value data**, enter **1**, and then choose **OK**.
7. Close the Registry Editor and reboot the server.

For more information, see [EnableDeadGWDetect](#) in the *Microsoft TechNet Library*.

Step 6: Test the VPN connection

To test that the VPN connection is working correctly, launch an instance into your VPC, and ensure that it does not have an internet connection. After you've launched the instance, ping its private IP address from your Windows Server. The VPN tunnel comes up when traffic is generated from the customer gateway device. Therefore, the ping command also initiates the VPN connection.

For steps to test the VPN connection, see [Test a Site-to-Site VPN connection](#).

If the ping command fails, check the following information:

- Ensure that you have configured your security group rules to allow ICMP to the instance in your VPC. If your Windows Server is an EC2 instance, ensure that its security group's outbound rules allow IPsec traffic. For more information, see [Configuring your Windows instance](#).
- Ensure that the operating system on the instance you are pinging is configured to respond to ICMP. We recommend that you use one of the Amazon Linux AMIs.

- If the instance you are pinging is a Windows instance, connect to the instance and enable inbound ICMPv4 on the Windows firewall.
- Ensure that you have configured the route tables correctly for your VPC or your subnet. For more information, see [Step 1: Create a VPN connection and configure your VPC](#).
- If your customer gateway device is an EC2 instance, ensure that you've disabled source/destination checking for the instance. For more information, see [Configuring your Windows instance](#).

In the Amazon VPC console, on the **VPN Connections** page, select your VPN connection. The first tunnel is in the UP state. The second tunnel should be configured, but it isn't used unless the first tunnel goes down. It may take a few moments to establish the encrypted tunnels.

Troubleshooting your customer gateway device

The following topics can help you troubleshoot connectivity issues on customer gateway devices.

For general testing instructions, see [Test a Site-to-Site VPN connection](#).

In addition to the topics in this section, you can also use [AWS Site-to-Site VPN logs](#) to help troubleshoot and resolve VPN connectivity issues.

Topics

- [Troubleshooting connectivity when using Border Gateway Protocol](#)
- [Troubleshooting connectivity without Border Gateway Protocol](#)
- [Troubleshooting Cisco ASA customer gateway device connectivity](#)
- [Troubleshooting Cisco IOS customer gateway device connectivity](#)
- [Troubleshooting Cisco IOS customer gateway device without Border Gateway Protocol connectivity](#)
- [Troubleshooting Juniper JunOS customer gateway device connectivity](#)
- [Troubleshooting Juniper ScreenOS customer gateway device connectivity](#)
- [Troubleshooting Yamaha customer gateway device connectivity](#)

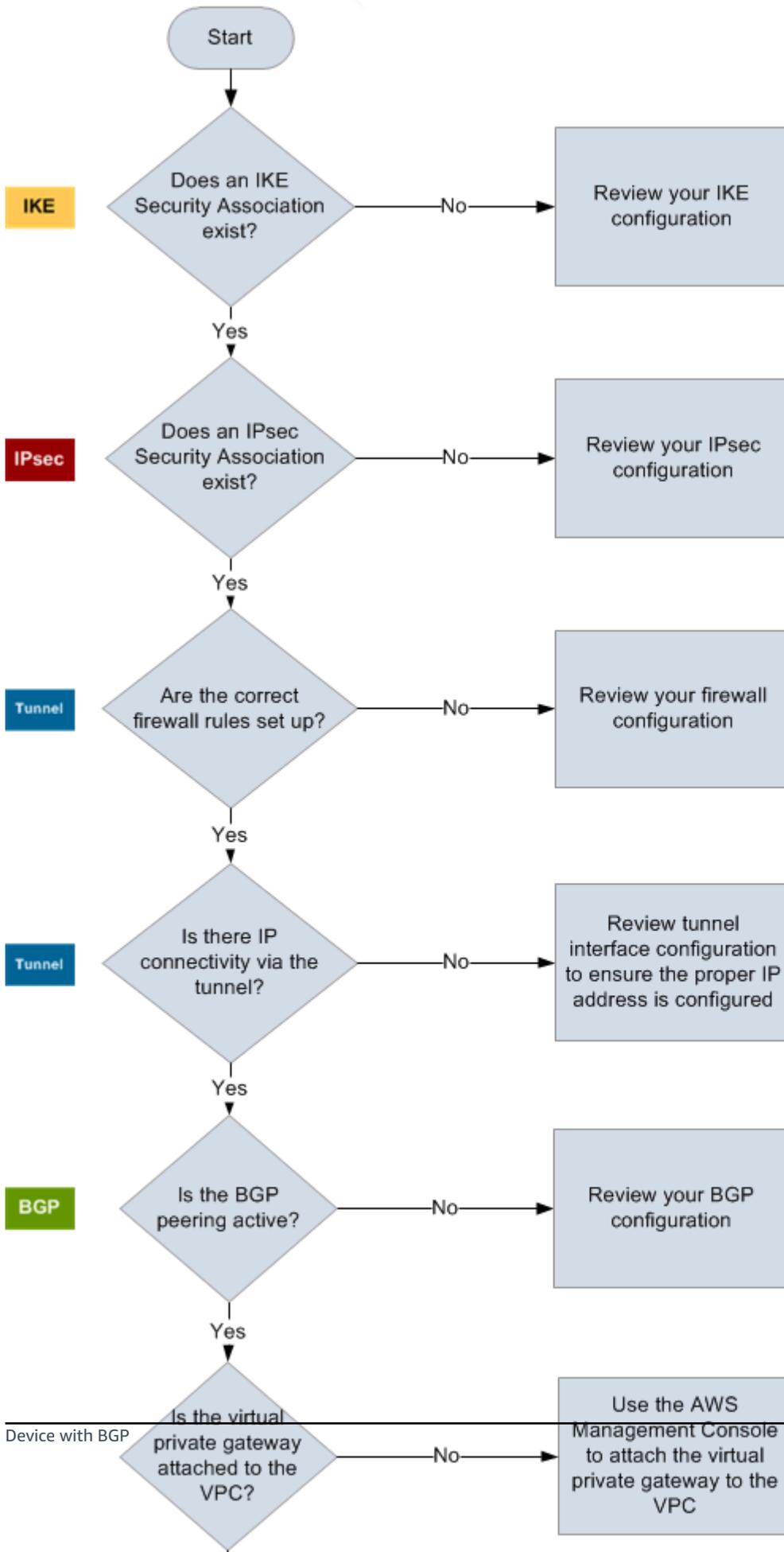
Additional resources

- [Amazon VPC forum](#)

- [How do I troubleshoot VPN tunnel connectivity to my Amazon VPC?](#)

Troubleshooting connectivity when using Border Gateway Protocol

The following diagram and table provide general instructions for troubleshooting a customer gateway device that uses Border Gateway Protocol (BGP). We also recommend that you enable the debug features of your device. Consult your gateway device vendor for details.



IKE	<p>Determine if an IKE security association exists.</p> <p>An IKE security association is required to exchange keys that are used to establish the IPsec security association.</p> <p>If no IKE security association exists, review your IKE configuration settings. You must configure the encryption, authentication, perfect forward secrecy, and mode parameters as listed in the configuration file.</p> <p>If an IKE security association exists, move on to 'IPsec'.</p>
IPsec	<p>Determine if an IPsec security association (SA) exists.</p> <p>An IPsec SA is the tunnel itself. Query your customer gateway device to determine if an IPsec SA is active. Ensure that you configure the encryption, authentication, perfect forward secrecy, and mode parameters as listed in the configuration file.</p> <p>If no IPsec SA exists, review your IPsec configuration.</p> <p>If an IPsec SA exists, move on to 'Tunnel'.</p>
Tunnel	<p>Confirm that the required firewall rules are set up (for a list of the rules, see Configuring a firewall between the internet and your customer gateway device). If they are, move forward.</p> <p>Determine if there is IP connectivity through the tunnel.</p> <p>Each side of the tunnel has an IP address as specified in the configuration file. The virtual private gateway address is the address used as the BGP neighbor address. From your customer gateway device, ping this address to determine if IP traffic is being properly encrypted and decrypted.</p> <p>If the ping isn't successful, review your tunnel interface configuration to make sure that the proper IP address is configured.</p> <p>If the ping is successful, move on to 'BGP'.</p>
BGP	<p>Determine if the BGP peering session is active.</p> <p>For each tunnel, do the following:</p>

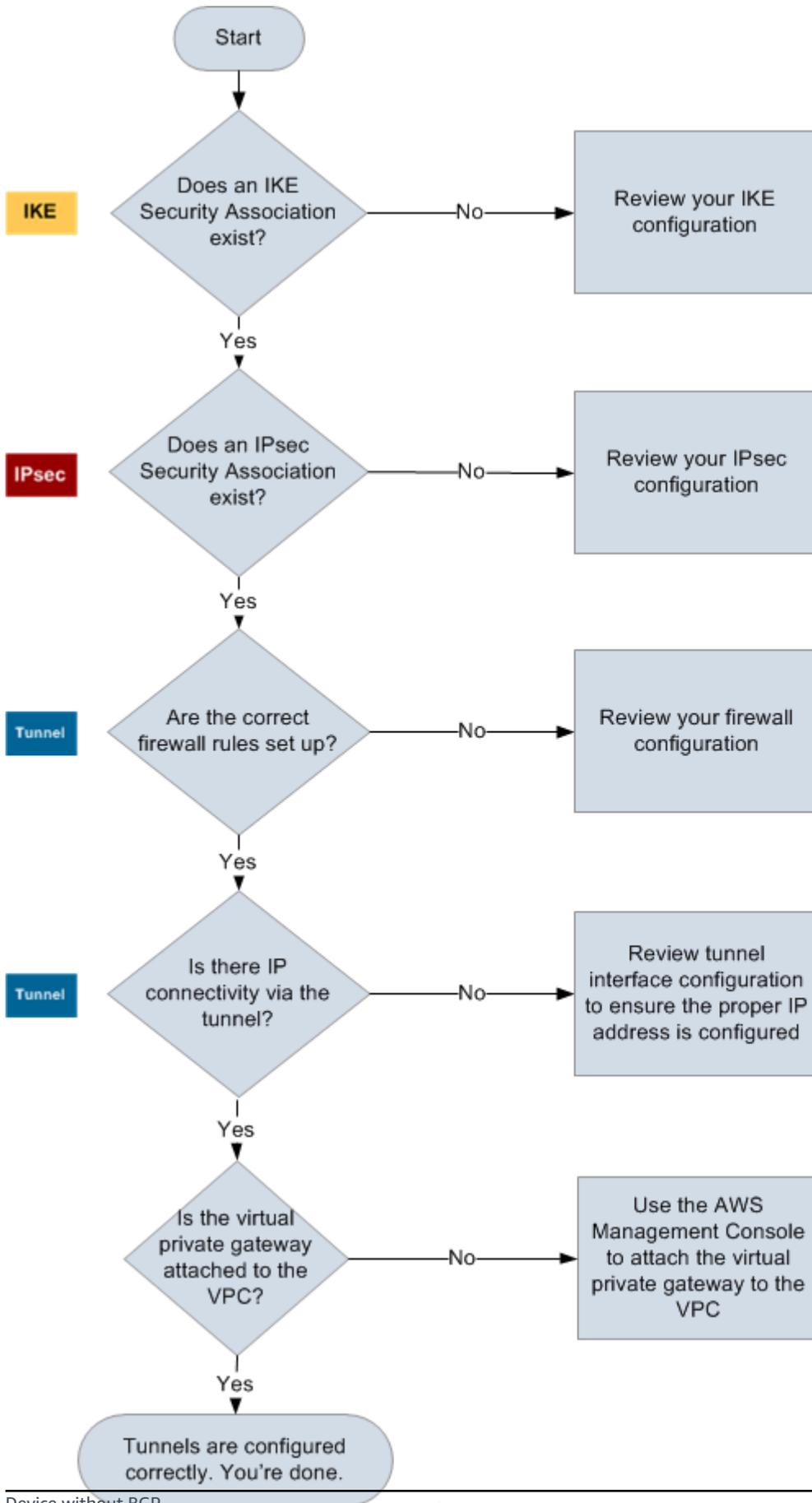
- On your customer gateway device, determine if the BGP status is `Active` or `Established` . It may take approximately 30 seconds for a BGP peering to become active.
- Ensure that the customer gateway device is advertising the default route (`0.0.0.0/0`) to the virtual private gateway.

If the tunnels are not in this state, review your BGP configuration.

If the BGP peering is established, you are receiving a prefix, and you are advertising a prefix, your tunnel is configured correctly. Make sure that both tunnels are in this state.

Troubleshooting connectivity without Border Gateway Protocol

The following diagram and table provide general instructions for troubleshooting a customer gateway device that does not use Border Gateway Protocol (BGP). We also recommend that you enable the debug features of your device. Consult your gateway device vendor for details.



IKE	<p>Determine if an IKE security association exists.</p> <p>An IKE security association is required to exchange keys that are used to establish the IPsec security association.</p> <p>If no IKE security association exists, review your IKE configuration settings. You must configure the encryption, authentication, perfect forward secrecy, and mode parameters as listed in the configuration file.</p> <p>If an IKE security association exists, move on to 'IPsec'.</p>
IPsec	<p>Determine if an IPsec security association (SA) exists.</p> <p>An IPsec SA is the tunnel itself. Query your customer gateway device to determine if an IPsec SA is active. Ensure that you configure the encryption, authentication, perfect forward secrecy, and mode parameters as listed in the configuration file.</p> <p>If no IPsec SA exists, review your IPsec configuration.</p> <p>If an IPsec SA exists, move on to 'Tunnel'.</p>
Tunnel	<p>Confirm that the required firewall rules are set up (for a list of the rules, see Configuring a firewall between the internet and your customer gateway device). If they are, move forward.</p> <p>Determine if there is IP connectivity through the tunnel.</p> <p>Each side of the tunnel has an IP address as specified in the configuration file. The virtual private gateway address is the address used as the BGP neighbor address. From your customer gateway device, ping this address to determine if IP traffic is being properly encrypted and decrypted.</p> <p>If the ping isn't successful, review your tunnel interface configuration to make sure that the proper IP address is configured.</p> <p>If the ping is successful, move on to 'Static routes'.</p>
Static routes	<p>For each tunnel, do the following:</p>

- Verify that you have added a static route to your VPC CIDR with the tunnels as the next hop.
- Verify that you have added a static route on the Amazon VPC console, to tell the virtual private gateway to route traffic back to your internal networks.

If the tunnels are not in this state, review your device configuration.

Make sure that both tunnels are in this state, and you're done.

Troubleshooting Cisco ASA customer gateway device connectivity

When you troubleshoot the connectivity of a Cisco customer gateway device, consider IKE, IPsec, and routing. You can troubleshoot these areas in any order, but we recommend that you start with IKE (at the bottom of the network stack) and move up.

Important

Some Cisco ASAs only support Active/Standby mode. When you use these Cisco ASAs, you can have only one active tunnel at a time. The other standby tunnel becomes active only if the first tunnel becomes unavailable. The standby tunnel might produce the following error in your log files, which can be ignored: Rejecting IPsec tunnel: no matching crypto map entry for remote proxy 0.0.0.0/0.0.0.0/0/0 local proxy 0.0.0.0/0.0.0.0/0/0 on interface outside.

IKE

Use the following command. The response shows a customer gateway device with IKE configured correctly.

```
ciscoasa# show crypto isakmp sa
```

```
Active SA: 2
Rekey SA: 0 (A tunnel will report 1 Active and 1 Rekey SA during rekey)
Total IKE SA: 2
```

```

1  IKE Peer: AWS_ENDPOINT_1
   Type    : L2L           Role    : initiator
   Rekey   : no           State   : MM_ACTIVE

```

You should see one or more lines containing an `src` value for the remote gateway that is specified in the tunnels. The state value should be `MM_ACTIVE` and status should be `ACTIVE`. The absence of an entry, or any entry in another state, indicates that IKE is not configured properly.

For further troubleshooting, run the following commands to enable log messages that provide diagnostic information.

```

router# term mon
router# debug crypto isakmp

```

To disable debugging, use the following command.

```

router# no debug crypto isakmp

```

IPsec

Use the following command. The response shows a customer gateway device with IPsec configured correctly.

```

ciscoasa# show crypto ipsec sa

```

```

interface: outside
  Crypto map tag: VPN_crypto_map_name, seq num: 2, local addr: 172.25.50.101

  access-list integ-ppe-loopback extended permit ip any vpc_subnet subnet_mask
  local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
  remote ident (addr/mask/prot/port): (vpc_subnet/subnet_mask/0/0)
  current_peer: integ-ppel

  #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
  #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts comp failed: 0, #pkts decomp failed: 0
  #pre-frag successes: 0, #pre-frag failures: 0, #fragments created: 0
  #PMTUs sent: 0, #PMTUs rcvd: 0, #decapsulated frgs needing reassembly: 0
  #send errors: 0, #recv errors: 0

```

```
local crypto endpt.: 172.25.50.101, remote crypto endpt.: AWS_ENDPOINT_1

path mtu 1500, ipsec overhead 74, media mtu 1500
current outbound spi: 6D9F8D3B
current inbound spi : 48B456A6

inbound esp sas:
spi: 0x48B456A6 (1219778214)
  transform: esp-aes esp-sha-hmac no compression
  in use settings = {L2L, Tunnel, PFS Group 2, }
  slot: 0, conn_id: 4710400, crypto-map: VPN_cry_map_1
  sa timing: remaining key lifetime (kB/sec): (4374000/3593)
  IV size: 16 bytes
  replay detection support: Y
  Anti replay bitmap:
    0x00000000 0x00000001
outbound esp sas:
spi: 0x6D9F8D3B (1839172923)
  transform: esp-aes esp-sha-hmac no compression
  in use settings = {L2L, Tunnel, PFS Group 2, }
  slot: 0, conn_id: 4710400, crypto-map: VPN_cry_map_1
  sa timing: remaining key lifetime (kB/sec): (4374000/3593)
  IV size: 16 bytes
  replay detection support: Y
  Anti replay bitmap:
    0x00000000 0x00000001
```

For each tunnel interface, you should see both inbound esp sas and outbound esp sas. This assumes that an SA is listed (for example, spi: 0x48B456A6), and that IPsec is configured correctly.

In Cisco ASA, the IPsec only comes up after interesting traffic (traffic that should be encrypted) is sent. To always keep the IPsec active, we recommend configuring an SLA monitor. The SLA monitor continues to send interesting traffic, keeping the IPsec active.

You can also use the following ping command to force your IPsec to start negotiation and go up.

```
ping ec2_instance_ip_address
```

```
Pinging ec2_instance_ip_address with 32 bytes of data:
```

```
Reply from ec2_instance_ip_address: bytes=32 time<1ms TTL=128
```

```
Reply from ec2_instance_ip_address: bytes=32 time<1ms TTL=128
```

```
Reply from ec2_instance_ip_address: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 10.0.0.4:
```

```
Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
```

```
Approximate round trip times in milliseconds:
```

```
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

For further troubleshooting, use the following command to enable debugging.

```
router# debug crypto ipsec
```

To disable debugging, use the following command.

```
router# no debug crypto ipsec
```

Routing

Ping the other end of the tunnel. If this is working, then your IPsec should be established. If this is not working, check your access lists, and refer to the previous IPsec section.

If you are not able to reach your instances, check the following information.

1. Verify that the access list is configured to allow traffic that is associated with the crypto map.

You can do this using the following command.

```
ciscoasa# show run crypto
```

```
crypto ipsec transform-set transform-amzn esp-aes esp-sha-hmac
crypto map VPN_crypto_map_name 1 match address access-list-name
crypto map VPN_crypto_map_name 1 set pfs
crypto map VPN_crypto_map_name 1 set peer AWS_ENDPOINT_1 AWS_ENDPOINT_2
crypto map VPN_crypto_map_name 1 set transform-set transform-amzn
crypto map VPN_crypto_map_name 1 set security-association lifetime seconds 3600
```

2. Check the access list using the following command.

```
ciscoasa# show run access-list access-list-name
```

```
access-list access-list-name extended permit ip any vpc_subnet subnet_mask
```

3. Verify that the access list is correct. The following example access list allows all internal traffic to the VPC subnet 10.0.0.0/16.

```
access-list access-list-name extended permit ip any 10.0.0.0 255.255.0.0
```

4. Run a traceroute from the Cisco ASA device, to see if it reaches the Amazon routers (for example, *AWS_ENDPOINT_1/AWS_ENDPOINT_2*).

If this reaches the Amazon router, then check the static routes that you added in the Amazon VPC console, and also the security groups for the particular instances.

5. For further troubleshooting, review the configuration.

Troubleshooting Cisco IOS customer gateway device connectivity

When you troubleshoot the connectivity of a Cisco customer gateway device, consider four things: IKE, IPsec, the tunnel, and BGP. You can troubleshoot these areas in any order, but we recommend that you start with IKE (at the bottom of the network stack) and move up.

IKE

Use the following command. The response shows a customer gateway device with IKE configured correctly.

```
router# show crypto isakmp sa
```

```
IPv4 Crypto ISAKMP SA
dst          src          state          conn-id slot status
192.168.37.160 72.21.209.193 QM_IDLE        2001    0 ACTIVE
192.168.37.160 72.21.209.225 QM_IDLE        2002    0 ACTIVE
```

You should see one or more lines containing an `src` value for the remote gateway that is specified in the tunnels. The state should be `QM_IDLE` and status should be `ACTIVE`. The absence of an entry, or any entry in another state, indicate that IKE is not configured properly.

For further troubleshooting, run the following commands to enable log messages that provide diagnostic information.

```
router# term mon
router# debug crypto isakmp
```

To disable debugging, use the following command.

```
router# no debug crypto isakmp
```

IPsec

Use the following command. The response shows a customer gateway device with IPsec configured correctly.

```
router# show crypto ipsec sa
```

```
interface: Tunnel1
  Crypto map tag: Tunnel1-head-0, local addr 192.168.37.160

  protected vrf: (none)
  local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
  remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
  current_peer 72.21.209.225 port 500
    PERMIT, flags={origin_is_acl,}
    #pkts encaps: 149, #pkts encrypt: 149, #pkts digest: 149
    #pkts decaps: 146, #pkts decrypt: 146, #pkts verify: 146
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

  local crypto endpt.: 174.78.144.73, remote crypto endpt.: 72.21.209.225
  path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0
  current outbound spi: 0xB8357C22(3090512930)

  inbound esp sas:
    spi: 0x6ADB173(112046451)
      transform: esp-aes esp-sha-hmac ,
      in use settings ={Tunnel, }
      conn id: 1, flow_id: Motorola SEC 2.0:1, crypto map: Tunnel1-head-0
      sa timing: remaining key lifetime (k/sec): (4467148/3189)
      IV size: 16 bytes
      replay detection support: Y  replay window size: 128
```

```
Status: ACTIVE
```

```
inbound ah sas:
```

```
inbound pcp sas:
```

```
outbound esp sas:
```

```
spi: 0xB8357C22(3090512930)
```

```
transform: esp-aes esp-sha-hmac ,
```

```
in use settings ={Tunnel, }
```

```
conn id: 2, flow_id: Motorola SEC 2.0:2, crypto map: Tunnel1-head-0
```

```
sa timing: remaining key lifetime (k/sec): (4467148/3189)
```

```
IV size: 16 bytes
```

```
replay detection support: Y replay window size: 128
```

```
Status: ACTIVE
```

```
outbound ah sas:
```

```
outbound pcp sas:
```

```
interface: Tunnel2
```

```
Crypto map tag: Tunnel2-head-0, local addr 174.78.144.73
```

```
protected vrf: (none)
```

```
local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
```

```
remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
```

```
current_peer 72.21.209.193 port 500
```

```
PERMIT, flags={origin_is_acl,}
```

```
#pkts encaps: 26, #pkts encrypt: 26, #pkts digest: 26
```

```
#pkts decaps: 24, #pkts decrypt: 24, #pkts verify: 24
```

```
#pkts compressed: 0, #pkts decompressed: 0
```

```
#pkts not compressed: 0, #pkts compr. failed: 0
```

```
#pkts not decompressed: 0, #pkts decompress failed: 0
```

```
#send errors 0, #recv errors 0
```

```
local crypto endpt.: 174.78.144.73, remote crypto endpt.: 72.21.209.193
```

```
path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0
```

```
current outbound spi: 0xF59A3FF6(4120526838)
```

```
inbound esp sas:
```

```
spi: 0xB6720137(3060924727)
```

```
transform: esp-aes esp-sha-hmac ,
```

```
in use settings ={Tunnel, }
```

```
conn id: 3, flow_id: Motorola SEC 2.0:3, crypto map: Tunnel2-head-0
```

```
sa timing: remaining key lifetime (k/sec): (4387273/3492)
IV size: 16 bytes
replay detection support: Y  replay window size: 128
Status: ACTIVE

inbound ah sas:

inbound pcp sas:

outbound esp sas:
spi: 0xF59A3FF6(4120526838)
transform: esp-aes esp-sha-hmac ,
in use settings ={Tunnel, }
conn id: 4, flow_id: Motorola SEC 2.0:4, crypto map: Tunnel2-head-0
sa timing: remaining key lifetime (k/sec): (4387273/3492)
IV size: 16 bytes
replay detection support: Y  replay window size: 128
Status: ACTIVE

outbound ah sas:

outbound pcp sas:
```

For each tunnel interface, you should see both inbound esp sas and outbound esp sas. Assuming an SA is listed (spi: 0xF95D2F3C, for example) and the Status is ACTIVE, IPsec is configured correctly.

For further troubleshooting, use the following command to enable debugging.

```
router# debug crypto ipsec
```

Use the following command to disable debugging.

```
router# no debug crypto ipsec
```

Tunnel

First, check that you have the necessary firewall rules in place. For more information, see [Configuring a firewall between the internet and your customer gateway device](#).

If your firewall rules are set up correctly, then continue troubleshooting with the following command.

```
router# show interfaces tun1
```

```
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 169.254.255.2/30
  MTU 17867 bytes, BW 100 Kbit/sec, DLY 50000 usec,
    reliability 255/255, txload 2/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 174.78.144.73, destination 72.21.209.225
  Tunnel protocol/transport IPSEC/IP
  Tunnel TTL 255
  Tunnel transport MTU 1427 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Tunnel protection via IPSec (profile "ipsec-vpn-92df3bfb-0")
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 1 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
    407 packets input, 30010 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
```

Make sure that the `line protocol` is up. Check that the tunnel source IP address, source interface, and destination respectively match the tunnel configuration for the customer gateway device outside IP address, interface, and virtual private gateway outside IP address. Make sure that `Tunnel protection via IPSec` is present. Run the command on both tunnel interfaces. To resolve any problems, review the configuration and check the physical connections to your customer gateway device.

Also use the following command, replacing `169.254.255.1` with the inside IP address of your virtual private gateway.

```
router# ping 169.254.255.1 df-bit size 1410
```

```
Type escape sequence to abort.
Sending 5, 1410-byte ICMP Echos to 169.254.255.1, timeout is 2 seconds:
```

```
Packet sent with the DF bit set
!!!!!
```

You should see five exclamation points.

For further troubleshooting, review the configuration.

BGP

Use the following command.

```
router# show ip bgp summary
```

```
BGP router identifier 192.168.37.160, local AS number 65000
BGP table version is 8, main routing table version 8
2 network entries using 312 bytes of memory
2 path entries using 136 bytes of memory
3/1 BGP path/bestpath attribute entries using 444 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 1 (at peak 2) using 32 bytes of memory
BGP using 948 total bytes of memory
BGP activity 4/1 prefixes, 4/1 paths, scan interval 15 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
169.254.255.1	4	7224	363	323	8	0	0	00:54:21	1
169.254.255.5	4	7224	364	323	8	0	0	00:00:24	1

Both neighbors should be listed. For each, you should see a State/PfxRcd value of 1.

If the BGP peering is up, verify that your customer gateway device is advertising the default route (0.0.0.0/0) to the VPC.

```
router# show bgp all neighbors 169.254.255.1 advertised-routes
```

```
For address family: IPv4 Unicast
BGP table version is 3, local router ID is 174.78.144.73
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
Originating default network 0.0.0.0
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.120.0.0/16	169.254.255.1	100	0	7224	i

```
Total number of prefixes 1
```

Additionally, ensure that you're receiving the prefix corresponding to your VPC from the virtual private gateway.

```
router# show ip route bgp
```

```
10.0.0.0/16 is subnetted, 1 subnets
B      10.255.0.0 [20/0] via 169.254.255.1, 00:00:20
```

For further troubleshooting, review the configuration.

Troubleshooting Cisco IOS customer gateway device without Border Gateway Protocol connectivity

When you troubleshoot the connectivity of a Cisco customer gateway device, consider three things: IKE, IPsec, and tunnel. You can troubleshoot these areas in any order, but we recommend that you start with IKE (at the bottom of the network stack) and move up.

IKE

Use the following command. The response shows a customer gateway device with IKE configured correctly.

```
router# show crypto isakmp sa
```

IPv4 Crypto ISAKMP SA					
dst	src	state	conn-id	slot	status
174.78.144.73	205.251.233.121	QM_IDLE	2001	0	ACTIVE
174.78.144.73	205.251.233.122	QM_IDLE	2002	0	ACTIVE

You should see one or more lines containing an `src` value for the remote gateway that is specified in the tunnels. The `state` should be `QM_IDLE` and `status` should be `ACTIVE`. The absence of an entry, or any entry in another state, indicates that IKE is not configured properly.

For further troubleshooting, run the following commands to enable log messages that provide diagnostic information.

```
router# term mon
router# debug crypto isakmp
```

To disable debugging, use the following command.

```
router# no debug crypto isakmp
```

IPsec

Use the following command. The response shows a customer gateway device with IPsec configured correctly.

```
router# show crypto ipsec sa
```

```
interface: Tunnel1
  Crypto map tag: Tunnel1-head-0, local addr 174.78.144.73

protected vrf: (none)
local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
current_peer 72.21.209.225 port 500
  PERMIT, flags={origin_is_acl,}
  #pkts encaps: 149, #pkts encrypt: 149, #pkts digest: 149
  #pkts decaps: 146, #pkts decrypt: 146, #pkts verify: 146
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts compr. failed: 0
  #pkts not decompressed: 0, #pkts decompress failed: 0
  #send errors 0, #recv errors 0

local crypto endpt.: 174.78.144.73, remote crypto endpt.:205.251.233.121
path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0
current outbound spi: 0xB8357C22(3090512930)

inbound esp sas:
  spi: 0x6ADB173(112046451)
  transform: esp-aes esp-sha-hmac ,
  in use settings ={Tunnel, }
  conn id: 1, flow_id: Motorola SEC 2.0:1, crypto map: Tunnel1-head-0
  sa timing: remaining key lifetime (k/sec): (4467148/3189)
```

```
    IV size: 16 bytes
    replay detection support: Y  replay window size: 128
    Status: ACTIVE

inbound ah sas:

inbound pcp sas:

outbound esp sas:
spi: 0xB8357C22(3090512930)
transform: esp-aes esp-sha-hmac ,
in use settings ={Tunnel, }
conn id: 2, flow_id: Motorola SEC 2.0:2, crypto map: Tunnel1-head-0
sa timing: remaining key lifetime (k/sec): (4467148/3189)
IV size: 16 bytes
replay detection support: Y  replay window size: 128
Status: ACTIVE

outbound ah sas:

outbound pcp sas:

interface: Tunnel2
Crypto map tag: Tunnel2-head-0, local addr 205.251.233.122

protected vrf: (none)
local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
current_peer 72.21.209.193 port 500
PERMIT, flags={origin_is_acl,}
#pkts encaps: 26, #pkts encrypt: 26, #pkts digest: 26
#pkts decaps: 24, #pkts decrypt: 24, #pkts verify: 24
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0

local crypto endpt.: 174.78.144.73, remote crypto endpt.:205.251.233.122
path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0
current outbound spi: 0xF59A3FF6(4120526838)

inbound esp sas:
spi: 0xB6720137(3060924727)
transform: esp-aes esp-sha-hmac ,
```

```
in use settings ={Tunnel, }
conn id: 3, flow_id: Motorola SEC 2.0:3, crypto map: Tunnel2-head-0
sa timing: remaining key lifetime (k/sec): (4387273/3492)
IV size: 16 bytes
replay detection support: Y  replay window size: 128
Status: ACTIVE
```

inbound ah sas:

inbound pcp sas:

outbound esp sas:

```
spi: 0xF59A3FF6(4120526838)
transform: esp-aes esp-sha-hmac ,
in use settings ={Tunnel, }
conn id: 4, flow_id: Motorola SEC 2.0:4, crypto map: Tunnel2-head-0
sa timing: remaining key lifetime (k/sec): (4387273/3492)
IV size: 16 bytes
replay detection support: Y  replay window size: 128
Status: ACTIVE
```

outbound ah sas:

outbound pcp sas:

For each tunnel interface, you should see both an inbound esp sas and outbound esp sas. This assumes that an SA is listed (for example, spi: 0x48B456A6), that the status is ACTIVE, and that IPsec is configured correctly.

For further troubleshooting, use the following command to enable debugging.

```
router# debug crypto ipsec
```

To disable debugging, use the following command.

```
router# no debug crypto ipsec
```

Tunnel

First, check that you have the necessary firewall rules in place. For more information, see [Configuring a firewall between the internet and your customer gateway device](#).

If your firewall rules are set up correctly, then continue troubleshooting with the following command.

```
router# show interfaces tun1
```

```
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 169.254.249.18/30
  MTU 17867 bytes, BW 100 Kbit/sec, DLY 50000 usec,
    reliability 255/255, txload 2/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 174.78.144.73, destination 205.251.233.121
  Tunnel protocol/transport IPSEC/IP
  Tunnel TTL 255
  Tunnel transport MTU 1427 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Tunnel protection via IPSec (profile "ipsec-vpn-92df3bfb-0")
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 1 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
    407 packets input, 30010 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
```

Make sure that the line protocol is up. Check that the tunnel source IP address, source interface, and destination respectively match the tunnel configuration for the customer gateway device outside IP address, interface, and virtual private gateway outside IP address. Make sure that Tunnel protection through IPSec is present. Run the command on both tunnel interfaces. To resolve any problems, review the configuration and check the physical connections to your customer gateway device.

You can also use the following command, replacing 169.254.249.18 with the inside IP address of your virtual private gateway.

```
router# ping 169.254.249.18 df-bit size 1410
```

```
Type escape sequence to abort.  
Sending 5, 1410-byte ICMP Echos to 169.254.249.18, timeout is 2 seconds:  
Packet sent with the DF bit set  
!!!!!
```

You should see five exclamation points.

Routing

To see your static route table, use the following command.

```
router# sh ip route static
```

```
1.0.0.0/8 is variably subnetted  
S      10.0.0.0/16 is directly connected, Tunnel1  
is directly connected, Tunnel2
```

You should see that the static route for the VPC CIDR through both tunnels exists. If it does not exist, add the static routes as follows.

```
router# ip route 10.0.0.0 255.255.0.0 Tunnel1 track 100  
router# ip route 10.0.0.0 255.255.0.0 Tunnel2 track 200
```

Checking the SLA monitor

```
router# show ip sla statistics 100
```

```
IPSLAs Latest Operation Statistics  
  
IPSLA operation id: 100  
    Latest RTT: 128 milliseconds  
Latest operation start time: *18:08:02.155 UTC Wed Jul 15 2012  
Latest operation return code: OK  
Number of successes: 3  
Number of failures: 0  
Operation time to live: Forever
```

```
router# show ip sla statistics 200
```

IPSLAs Latest Operation Statistics

```
IPSLA operation id: 200
    Latest RTT: 128 milliseconds
Latest operation start time: *18:08:02.155 UTC Wed Jul 15 2012
Latest operation return code: OK
Number of successes: 3
Number of failures: 0
Operation time to live: Forever
```

The value for `Number of successes` indicates whether the SLA monitor has been set up successfully.

For further troubleshooting, review the configuration.

Troubleshooting Juniper JunOS customer gateway device connectivity

When you troubleshoot the connectivity of a Juniper customer gateway device, consider four things: IKE, IPsec, tunnel, and BGP. You can troubleshoot these areas in any order, but we recommend that you start with IKE (at the bottom of the network stack) and move up.

IKE

Use the following command. The response shows a customer gateway device with IKE configured correctly.

```
user@router> show security ike security-associations
```

Index	Remote Address	State	Initiator cookie	Responder cookie	Mode
4	72.21.209.225	UP	c4cd953602568b74	0d6d194993328b02	Main
3	72.21.209.193	UP	b8c8fb7dc68d9173	ca7cb0abaedeb4bb	Main

You should see one or more lines containing a remote address of the remote gateway specified in the tunnels. The `State` should be `UP`. The absence of an entry, or any entry in another state (such as `DOWN`), is an indication that IKE is not configured properly.

For further troubleshooting, enable the IKE trace options as recommended in the example configuration file. Then run the following command to print a variety of debugging messages to the screen.

```
user@router> monitor start kmd
```

From an external host, you can retrieve the entire log file with the following command.

```
scp username@router.hostname:/var/log/kmd
```

IPsec

Use the following command. The response shows a customer gateway device with IPsec configured correctly.

```
user@router> show security ipsec security-associations
```

```
Total active tunnels: 2
ID      Gateway      Port  Algorithm      SPI      Life:sec/kb Mon vsys
<131073 72.21.209.225 500   ESP:aes-128/sha1 df27aae4 326/ unlim - 0
>131073 72.21.209.225 500   ESP:aes-128/sha1 5de29aa1 326/ unlim - 0
<131074 72.21.209.193 500   ESP:aes-128/sha1 dd16c453 300/ unlim - 0
>131074 72.21.209.193 500   ESP:aes-128/sha1 c1e0eb29 300/ unlim - 0
```

Specifically, you should see at least two lines per gateway address (corresponding to the remote gateway). The carets at the beginning of each line (< >) indicate the direction of traffic for the particular entry. The output has separate lines for inbound traffic ("<", traffic from the virtual private gateway to this customer gateway device) and outbound traffic (">").

For further troubleshooting, enable the IKE traceoptions (for more information, see the preceding section about IKE).

Tunnel

First, double-check that you have the necessary firewall rules in place. For a list of rules, see [Configuring a firewall between the internet and your customer gateway device](#).

If your firewall rules are set up correctly, then continue troubleshooting with the following command.

```
user@router> show interfaces st0.1
```

```
Logical interface st0.1 (Index 70) (SNMP ifIndex 126)
  Flags: Point-To-Point SNMP-Traps Encapsulation: Secure-Tunnel
```

```

Input packets : 8719
Output packets: 41841
Security: Zone: Trust
Allowed host-inbound traffic : bgp ping ssh traceroute
Protocol inet, MTU: 9192
  Flags: None
  Addresses, Flags: Is-Preferred Is-Primary
  Destination: 169.254.255.0/30, Local: 169.254.255.2

```

Make sure that the `Security: Zone` is correct, and that the `Local` address matches the customer gateway device tunnel inside address.

Next, use the following command, replacing `169.254.255.1` with the inside IP address of your virtual private gateway. Your results should look like the response shown here.

```
user@router> ping 169.254.255.1 size 1382 do-not-fragment
```

```

PING 169.254.255.1 (169.254.255.1): 1410 data bytes
64 bytes from 169.254.255.1: icmp_seq=0 ttl=64 time=71.080 ms
64 bytes from 169.254.255.1: icmp_seq=1 ttl=64 time=70.585 ms

```

For further troubleshooting, review the configuration.

BGP

Run the following command.

```
user@router> show bgp summary
```

```

Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed  History  Damp State  Pending
inet.0         2           1           0           0         0         0
Peer           AS        InPkt    OutPkt    OutQ    Flaps Last Up/Dwn State|
#Active/Received/Accepted/Damped...
169.254.255.1  7224      9        10        0        0        1:00 1/1/1/0
    0/0/0/0
169.254.255.5  7224      8         9         0        0        56 0/1/1/0
    0/0/0/0

```

For further troubleshooting, use the following command, replacing `169.254.255.1` with the inside IP address of your virtual private gateway.

```
user@router> show bgp neighbor 169.254.255.1
```

```
Peer: 169.254.255.1+179 AS 7224 Local: 169.254.255.2+57175 AS 65000
  Type: External      State: Established      Flags: <ImportEval Sync>
  Last State: OpenConfirm  Last Event: RecvKeepAlive
  Last Error: None
  Export: [ EXPORT-DEFAULT ]
  Options: <Preference HoldTime PeerAS LocalAS Refresh>
  Holdtime: 30 Preference: 170 Local AS: 65000 Local System AS: 0
  Number of flaps: 0
  Peer ID: 169.254.255.1      Local ID: 10.50.0.10      Active Holdtime: 30
  Keepalive Interval: 10      Peer index: 0
  BFD: disabled, down
  Local Interface: st0.1
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 7224)
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:          1
    Received prefixes:        1
    Accepted prefixes:        1
    Suppressed due to damping: 0
    Advertised prefixes:      1
  Last traffic (seconds): Received 4      Sent 8      Checked 4
  Input messages:  Total 24      Updates 2      Refreshes 0      Octets 505
  Output messages: Total 26      Updates 1      Refreshes 0      Octets 582
  Output Queue[0]: 0
```

Here you should see Received prefixes and Advertised prefixes listed at 1 each. This should be within the Table inet.0 section.

If the State is not Established, check the Last State and Last Error for details of what is required to correct the problem.

If the BGP peering is up, verify that your customer gateway device is advertising the default route (0.0.0.0/0) to the VPC.

```
user@router> show route advertising-protocol bgp 169.254.255.1
```

```
inet.0: 10 destinations, 11 routes (10 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED    Lclpref    AS path
* 0.0.0.0/0             Self              0      0           I
```

Additionally, make sure that you're receiving the prefix that corresponds to your VPC from the virtual private gateway.

```
user@router> show route receive-protocol bgp 169.254.255.1
```

```
inet.0: 10 destinations, 11 routes (10 active, 0 holddown, 0 hidden)
  Prefix                Nexthop          MED    Lclpref    AS path
* 10.110.0.0/16        169.254.255.1   100    0           7224 I
```

Troubleshooting Juniper ScreenOS customer gateway device connectivity

When you troubleshoot the connectivity of a Juniper ScreenOS-based customer gateway device, consider four things: IKE, IPsec, tunnel, and BGP. You can troubleshoot these areas in any order, but we recommend that you start with IKE (at the bottom of the network stack) and move up.

IKE and IPsec

Use the following command. The response shows a customer gateway device with IKE configured correctly.

```
ssg5-serial-> get sa
```

```
total configured sa: 2
HEX ID    Gateway          Port Algorithm    SPI    Life:sec kb Sta    PID vsys
```

```
00000002< 72.21.209.225 500 esp:a128/sha1 80041ca4 3385 unlim A/- -1 0
00000002> 72.21.209.225 500 esp:a128/sha1 8cdd274a 3385 unlim A/- -1 0
00000001< 72.21.209.193 500 esp:a128/sha1 ecf0bec7 3580 unlim A/- -1 0
00000001> 72.21.209.193 500 esp:a128/sha1 14bf7894 3580 unlim A/- -1 0
```

You should see one or more lines containing a remote address of the remote gateway that is specified in the tunnels. The Sta value should be A/- and SPI should be a hexadecimal number other than 00000000. Entries in other states indicate that IKE is not configured properly.

For further troubleshooting, enable the IKE trace options (as recommended in the example configuration file).

Tunnel

First, double-check that you have the necessary firewall rules in place. For a list of rules, see [Configuring a firewall between the internet and your customer gateway device](#).

If your firewall rules are set up correctly, then continue troubleshooting with the following command.

```
ssg5-serial-> get interface tunnel.1
```

```
Interface tunnel.1:
description tunnel.1
number 20, if_info 1768, if_index 1, mode route
link ready
vsys Root, zone Trust, vr trust-vr
admin mtu 1500, operating mtu 1500, default mtu 1500
*ip 169.254.255.2/30
*manage ip 169.254.255.2
route-deny disable
bound vpn:
  IPSEC-1

Next-Hop Tunnel Binding table
Flag Status Next-Hop(IP)   tunnel-id  VPN

pmtu-v4 disabled
ping disabled, telnet disabled, SSH disabled, SNMP disabled
web disabled, ident-reset disabled, SSL disabled
```

```

OSPF disabled  BGP enabled  RIP disabled  RIPng disabled  mtrace disabled
PIM: not configured  IGMP not configured
NHRP disabled
bandwidth: physical 0kbps, configured egress [gbw 0kbps mbw 0kbps]
           configured ingress mbw 0kbps, current bw 0kbps
           total allocated gbw 0kbps

```

Make sure that you see `link:ready`, and that the IP address matches the customer gateway device tunnel inside address.

Next, use the following command, replacing `169.254.255.1` with the inside IP address of your virtual private gateway. Your results should look like the response shown here.

```

ssg5-serial-> ping 169.254.255.1

```

```

Type escape sequence to abort

```

```

Sending 5, 100-byte ICMP Echos to 169.254.255.1, timeout is 1 seconds
!!!!

```

```

Success Rate is 100 percent (5/5), round-trip time min/avg/max=32/32/33 ms

```

For further troubleshooting, review the configuration.

BGP

Run the following command.

```

ssg5-serial-> get vrouter trust-vr protocol bgp neighbor

```

Peer AS	Remote IP	Local IP	Wt	Status	State	ConnID	Up/Down
7224	169.254.255.1	169.254.255.2	100	Enabled	ESTABLISH	10	00:01:01
7224	169.254.255.5	169.254.255.6	100	Enabled	ESTABLISH	11	00:00:59

The state of both BGP peers should be `ESTABLISH`, which means that the BGP connection to the virtual private gateway is active.

For further troubleshooting, use the following command, replacing `169.254.255.1` with the inside IP address of your virtual private gateway.

```
s5g5-serial-> get vr trust-vr prot bgp neigh 169.254.255.1
```

```
peer: 169.254.255.1, remote AS: 7224, admin status: enable
type: EBGp, multihop: 0(disable), MED: node default(0)
connection state: ESTABLISH, connection id: 18 retry interval: node default(120s), cur
  retry time 15s
configured hold time: node default(90s), configured keepalive: node default(30s)
configured adv-interval: default(30s)
designated local IP: n/a
local IP address/port: 169.254.255.2/13946, remote IP address/port: 169.254.255.1/179
router ID of peer: 169.254.255.1, remote AS: 7224
negotiated hold time: 30s, negotiated keepalive interval: 10s
route map in name: , route map out name:
weight: 100 (default)
self as next hop: disable
send default route to peer: disable
ignore default route from peer: disable
send community path attribute: no
reflector client: no
Neighbor Capabilities:
  Route refresh: advertised and received
  Address family IPv4 Unicast: advertised and received
force reconnect is disable
total messages to peer: 106, from peer: 106
update messages to peer: 6, from peer: 4
Tx queue length 0, Tx queue HWM: 1
route-refresh messages to peer: 0, from peer: 0
last reset 00:05:33 ago, due to BGP send Notification(Hold Timer Expired)(code 4 :
  subcode 0)
number of total successful connections: 4
connected: 2 minutes 6 seconds
Elapsed time since last update: 2 minutes 6 seconds
```

If the BGP peering is up, verify that your customer gateway device is advertising the default route (0.0.0.0/0) to the VPC. This command applies to ScreenOS version 6.2.0 and higher.

```
s5g5-serial-> get vr trust-vr protocol bgp rib neighbor 169.254.255.1 advertised
```

```
i: IBGP route, e: EBGp route, >: best route, *: valid route
      Prefix      Nexthop   Wt  Pref  Med Orig   AS-Path
-----
```

```
>i          0.0.0.0/0          0.0.0.0 32768   100    0  IGP
Total IPv4 routes advertised: 1
```

Additionally, ensure that you're receiving the prefix that corresponds to your VPC from the virtual private gateway. This command applies to ScreenOS version 6.2.0 and higher.

```
ssg5-serial-> get vr trust-vr protocol bgp rib neighbor 169.254.255.1 received
```

```
i: IBGP route, e: EBG route, >: best route, *: valid route
      Prefix          Nexthop    Wt  Pref  Med Orig   AS-Path
-----
>e*   10.0.0.0/16    169.254.255.1  100  100  100  IGP   7224
Total IPv4 routes received: 1
```

Troubleshooting Yamaha customer gateway device connectivity

When you troubleshoot the connectivity of a Yamaha customer gateway device, consider four things: IKE, IPsec, tunnel, and BGP. You can troubleshoot these areas in any order, but we recommend that you start with IKE (at the bottom of the network stack) and move up.

Note

The `proxy ID` setting used in phase 2 of IKE is disabled by default on the Yamaha router. This can cause problems connecting to Site-to-Site VPN. If the `proxy ID` is not configured on your router, please see the AWS-provided example configuration file for Yamaha to set properly.

IKE

Run the following command. The response shows a customer gateway device with IKE configured correctly.

```
# show ipsec sa gateway 1
```

```
sgw  flags local-id          remote-id          # of sa
-----
1    U K  YOUR_LOCAL_NETWORK_ADDRESS  72.21.209.225    i:2 s:1 r:1
```

You should see a line containing a `remote-id` value for the remote gateway that is specified in the tunnels. You can list all of the security associations (SAs) by omitting the tunnel number.

For further troubleshooting, run the following commands to enable DEBUG level log messages that provide diagnostic information.

```
# syslog debug on
# ipsec ike log message-info payload-info key-info
```

To cancel the logged items, run the following command.

```
# no ipsec ike log
# no syslog debug on
```

IPsec

Run the following command. The response shows a customer gateway device with IPsec configured correctly.

```
# show ipsec sa gateway 1 detail
```

```
SA[1] Duration: 10675s
Local ID: YOUR_LOCAL_NETWORK_ADDRESS
Remote ID: 72.21.209.225
Protocol: IKE
Algorithm: AES-CBC, SHA-1, MODP 1024bit

SPI: 6b ce fd 8a d5 30 9b 02 0c f3 87 52 4a 87 6e 77
Key: ** ** ** ** (confidential) ** ** ** ** **
-----
SA[2] Duration: 1719s
Local ID: YOUR_LOCAL_NETWORK_ADDRESS
Remote ID: 72.21.209.225
Direction: send
Protocol: ESP (Mode: tunnel)
Algorithm: AES-CBC (for Auth.: HMAC-SHA)
SPI: a6 67 47 47
Key: ** ** ** ** ** (confidential) ** ** ** ** **
-----
SA[3] Duration: 1719s
Local ID: YOUR_LOCAL_NETWORK_ADDRESS
```

```

Remote ID: 72.21.209.225
Direction: receive
Protocol: ESP (Mode: tunnel)
Algorithm: AES-CBC (for Auth.: HMAC-SHA)
SPI: 6b 98 69 2b
Key: ** ** ** ** ** (confidential)  ** ** ** ** **
-----
SA[4] Duration: 10681s
Local ID: YOUR_LOCAL_NETWORK_ADDRESS
Remote ID: 72.21.209.225
Protocol: IKE
Algorithm: AES-CBC, SHA-1, MODP 1024bit
SPI: e8 45 55 38 90 45 3f 67 a8 74 ca 71 ba bb 75 ee
Key: ** ** ** ** ** (confidential)  ** ** ** ** **
-----

```

For each tunnel interface, you should see both `receive sas` and `send sas`.

For further troubleshooting, use the following command to enable debugging.

```

# syslog debug on
# ipsec ike log message-info payload-info key-info

```

Run the following command to disable debugging.

```

# no ipsec ike log
# no syslog debug on

```

Tunnel

First, check that you have the necessary firewall rules in place. For a list of rules, see [Configuring a firewall between the internet and your customer gateway device](#).

If your firewall rules are set up correctly, then continue troubleshooting with the following command.

```

# show status tunnel 1

```

```

TUNNEL[1]:
Description:

```

```
Interface type: IPsec
Current status is Online.
from 2011/08/15 18:19:45.
5 hours 7 minutes 58 seconds connection.
Received: (IPv4) 3933 packets [244941 octets]
          (IPv6) 0 packet [0 octet]
Transmitted: (IPv4) 3933 packets [241407 octets]
            (IPv6) 0 packet [0 octet]
```

Make sure that the `current` status value is online and that `Interface` type is IPsec. Make sure to run the command on both tunnel interfaces. To resolve any problems here, review the configuration.

BGP

Run the following command.

```
# show status bgp neighbor
```

```
BGP neighbor is 169.254.255.1, remote AS 7224, local AS 65000, external link
BGP version 0, remote router ID 0.0.0.0
BGP state = Active
Last read 00:00:00, hold time is 0, keepalive interval is 0 seconds
Received 0 messages, 0 notifications, 0 in queue
Sent 0 messages, 0 notifications, 0 in queue
Connection established 0; dropped 0
Last reset never
Local host: unspecified
Foreign host: 169.254.255.1, Foreign port: 0
```

```
BGP neighbor is 169.254.255.5, remote AS 7224, local AS 65000, external link
BGP version 0, remote router ID 0.0.0.0
BGP state = Active
Last read 00:00:00, hold time is 0, keepalive interval is 0 seconds
Received 0 messages, 0 notifications, 0 in queue
Sent 0 messages, 0 notifications, 0 in queue
Connection established 0; dropped 0
Last reset never
Local host: unspecified
Foreign host: 169.254.255.5, Foreign port:
```

Both neighbors should be listed. For each, you should see a `BGP` state value of `Active`.

If the BGP peering is up, verify that your customer gateway device is advertising the default route (0.0.0.0/0) to the VPC.

```
# show status bgp neighbor 169.254.255.1 advertised-routes
```

```
Total routes: 1
*: valid route
  Network          Next Hop          Metric LocPrf Path
* default          0.0.0.0           0       IGP
```

Additionally, ensure that you're receiving the prefix that corresponds to your VPC from the virtual private gateway.

```
# show ip route
```

Destination	Gateway	Interface	Kind	Additional Info.
default	***.***.***.***	LAN3(DHCP)	static	
10.0.0.0/16	169.254.255.1	TUNNEL[1]	BGP	path=10124

Work with Site-to-Site VPN

You can work with Site-to-Site VPN resources using the Amazon VPC console or the AWS CLI.

Contents

- [Create a Site-to-Site VPN attachment for AWS Cloud WAN](#)
- [Create a transit gateway VPN attachment](#)
- [Test a Site-to-Site VPN connection](#)
- [Delete a Site-to-Site VPN connection](#)
- [Modify the target gateway of a Site-to-Site VPN connection](#)
- [Modify Site-to-Site VPN connection options](#)
- [Modify Site-to-Site VPN tunnel options](#)
- [Edit static routes for a Site-to-Site VPN connection](#)
- [Change the customer gateway for a Site-to-Site VPN connection](#)
- [Replace compromised credentials for your Site-to-Site VPN connection](#)
- [Rotate Site-to-Site VPN tunnel endpoint certificates](#)
- [Private IP VPN with AWS Direct Connect](#)

Create a Site-to-Site VPN attachment for AWS Cloud WAN

Follow the procedure below to create a Site-to-Site VPN attachment for AWS Cloud WAN.

To create a VPN attachment for AWS Cloud WAN using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN connections**.
3. Choose **Create VPN connection**.
4. (Optional) For **Name tag**, enter a name for the connection. Doing so creates a tag with a key of Name and the value that you specify.
5. For **Target gateway type**, choose **Not associated**.
6. For **Customer gateway**, do one of the following:
 - To use an existing customer gateway, choose **Existing**, and then choose the customer gateway.

- To create a customer gateway, choose **New**. For **IP address**, enter a static public IP address. For **Certificate ARN**, choose the ARN of your private certificate (if using certificate-based authentication). For **BGP ASN**, enter the Border Gateway Protocol (BGP) Autonomous System Number (ASN) of your customer gateway. For more information, see [Customer gateway options](#).
7. For **Routing options**, choose **Dynamic** or **Static**.
 8. For **Tunnel inside IP version**, choose **IPv4** or **IPv6**.
 9. (Optional) For **Enable acceleration**, select the check box to enable acceleration. For more information, see [Accelerated VPN connections](#).

If you enable acceleration, we create two accelerators that are used by your VPN connection. Additional charges apply.

10. (Optional) For **Local IPv4 network CIDR**, specify the IPv4 CIDR range on the customer gateway (on-premises) side that is allowed to communicate over the VPN tunnels. The default is `0.0.0.0/0`.

For **Remote IPv4 network CIDR**, specify the IPv4 CIDR range on the AWS side that is allowed to communicate over the VPN tunnels. The default is `0.0.0.0/0`.

If you specified **IPv6** for **Tunnel inside IP version**, then specify the IPv6 CIDR ranges on the customer gateway side and AWS side that are allowed to communicate over the VPN tunnels. The default for both ranges is `::/0`.

11. (Optional) For **Tunnel options**, you can specify the following information for each tunnel:
 - A size /30 IPv4 CIDR block from the `169.254.0.0/16` range for the inside tunnel IPv4 addresses.
 - If you specified **IPv6** for **Tunnel inside IP version**, a /126 IPv6 CIDR block from the `fd00::/8` range for the inside tunnel IPv6 addresses.
 - The IKE pre-shared key (PSK). The following versions are supported: IKEv1 or IKEv2.
 - To edit the advanced options for your tunnel, choose **Edit tunnel options**. For more information, see [VPN tunnel options](#).

12. Choose **Create VPN connection**.

To create a Site-to-Site VPN connection using the command line or API

- [CreateVpnConnection](#) (Amazon EC2 Query API)

- [create-vpn-connection](#) (AWS CLI)

Create a transit gateway VPN attachment

To create a VPN attachment on a transit gateway, you must specify the transit gateway and the customer gateway. The transit gateway will need to be created before following this procedure. For more information about creating a transit gateway, see [Transit gateways](#) in *Amazon VPC Transit Gateways*.

To create a VPN attachment on a transit gateway using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN connections**.
3. Choose **Create VPN connection**.
4. (Optional) For **Name tag**, enter a name for the connection. Doing so creates a tag with a key of Name and the value that you specify.
5. For **Target gateway type**, choose **Transit gateway**, and then choose the transit gateway.
6. For **Customer gateway**, do one of the following:
 - To use an existing customer gateway, choose **Existing**, and then choose the 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.
 - To create a customer gateway, choose **New**. For **IP Address**, enter a static public IP address. For **Certificate ARN**, choose the ARN of your private certificate (if using certificate-based authentication). For **BGP ASN**, enter the Border Gateway Protocol (BGP) Autonomous System Number (ASN) of your customer gateway. For more information, see [Customer gateway options](#).
7. For **Routing options**, choose **Dynamic** or **Static**.
8. For **Tunnel inside IP version**, specify whether the VPN tunnels support IPv4 or IPv6 traffic. IPv6 traffic is only supported for VPN connections on a transit gateway.
9. (Optional) For **Enable acceleration**, select the check box to enable acceleration. For more information, see [Accelerated VPN connections](#).

If you enable acceleration, we create two accelerators that are used by your VPN connection. Additional charges apply.

10. (Optional) For **Local IPv4 network CIDR**, specify the IPv4 CIDR range on the customer gateway (on-premises) side that is allowed to communicate over the VPN tunnels. The default is `0.0.0.0/0`.

For **Remote IPv4 network CIDR**, specify the IPv4 CIDR range on the AWS side that is allowed to communicate over the VPN tunnels. The default is `0.0.0.0/0`.

If you specified **IPv6** for **Tunnel inside IP version**, then specify the IPv6 CIDR ranges on the customer gateway side and AWS side that are allowed to communicate over the VPN tunnels. The default for both ranges is `::/0`.

11. (Optional) For **Tunnel options**, you can specify the following information for each tunnel:

- A size /30 IPv4 CIDR block from the `169.254.0.0/16` range for the inside tunnel IPv4 addresses.
- If you specified **IPv6** for **Tunnel inside IP version**, a /126 IPv6 CIDR block from the `fd00::/8` range for the inside tunnel IPv6 addresses.
- The IKE pre-shared key (PSK). The following versions are supported: IKEv1 or IKEv2.
- To edit the advanced options for your tunnel, choose **Edit tunnel options**. For more information, see [VPN tunnel options](#).

12. Choose **Create VPN connection**.

To create a VPN attachment using the AWS CLI

Use the [create-vpn-connection](#) command and specify the transit gateway ID for the `--transit-gateway-id` option.

Test a Site-to-Site VPN connection

After you create the AWS Site-to-Site VPN connection and configure the customer gateway, you can launch an instance and test the connection by pinging the instance.

Before you begin, make sure of the following:

- Use an AMI that responds to ping requests. We recommend that you use one of the Amazon Linux AMIs.

- Configure any security group or network ACL in your VPC that filters traffic to the instance to allow inbound and outbound ICMP traffic. This enables the instance to receive ping requests.
- If you are using instances running Windows Server, connect to the instance and enable inbound ICMPv4 on the Windows firewall in order to ping the instance.
- (Static routing) Ensure that the customer gateway device has a static route to your VPC, and that your VPN connection has a static route so that traffic can get back to your customer gateway device.
- (Dynamic routing) Ensure that the BGP status on your customer gateway device is established. It takes approximately 30 seconds for a BGP peering session to be established. Ensure that routes are advertised with BGP correctly and showing in the subnet route table, so that traffic can get back to your customer gateway. Make sure that both tunnels are configured with BGP routing.
- Ensure that you have configured routing in your subnet route tables for the VPN connection.

To test connectivity

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch instance**.
3. (Optional) For **Name**, enter a descriptive name for your instance.
4. For **Application and OS Images (Amazon Machine Image)**, choose **Quick Start**, and then choose the operating system for your instance.
5. For **Key pair name**, choose an existing key pair or create a new one.
6. For **Network settings**, choose **Select existing security group**, and then choose the security group that you configured.
7. In the **Summary** panel, choose **Launch instance**.
8. After the instance is running, get its private IP address (for example, 10.0.0.4). The Amazon EC2 console displays the address as part of the instance's details.
9. From a computer in your network that is behind the customer gateway device, use the **ping** command with the instance's private IP address.

```
ping 10.0.0.4
```

A successful response is similar to the following.

```
Pinging 10.0.0.4 with 32 bytes of data:
```

```
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.4:
Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),

Approximate round trip times in milliseconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway device and then repeat this step. You can't disable a tunnel on the AWS side of the VPN connection.

10. To test the connection from AWS to your on-premises network, you can use SSH or RDP to connect to your instance from your network. You can then run the `ping` command with the private IP address of another computer in your network, to verify that both sides of the connection can initiate and receive requests.

For more information about how to connect to a Linux instance, see [Connect to your Linux instance](#) in the *Amazon EC2 User Guide for Linux Instances*. For more information about how to connect to a Windows instance, see [Connect to your Windows instance](#) in the *Amazon EC2 User Guide for Windows Instances*.

Delete a Site-to-Site VPN connection

If you no longer need an AWS Site-to-Site VPN connection, you can delete it. When you delete a Site-to-Site VPN connection, we do not delete the customer gateway or virtual private gateway that was associated with the Site-to-Site VPN connection. If you no longer need the customer gateway and virtual private gateway, you can delete them.

Warning

If you delete your Site-to-Site VPN connection and then create a new one, you must download a new configuration file and reconfigure the customer gateway device.

Tasks

- [Delete a VPN connection](#)
- [Delete a customer gateway](#)
- [Detach and delete a virtual private gateway](#)

Delete a VPN connection

After you delete your Site-to-Site VPN connection, it remains visible for a short while with a state of `deleted`, and then the entry is automatically removed.

To delete a VPN connection using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN connections**.
3. Select the VPN connection and choose **Actions, Delete VPN connection**.
4. When prompted for confirmation, enter **delete** and then choose **Delete**.

To delete a VPN connection using the command line or API

- [DeleteVpnConnection](#) (Amazon EC2 Query API)
- [delete-vpn-connection](#) (AWS CLI)
- [Remove-EC2VpnConnection](#) (AWS Tools for Windows PowerShell)

Delete a customer gateway

If you no longer need a customer gateway, you can delete it. You can't delete a customer gateway that's being used in a Site-to-Site VPN connection.

To delete a customer gateway using the console

1. In the navigation pane, choose **Customer gateways**.
2. Select the customer gateway and choose **Actions, Delete customer gateway**.
3. When prompted for confirmation, enter **delete** and then choose **Delete**.

To delete a customer gateway using the command line or API

- [DeleteCustomerGateway](#) (Amazon EC2 Query API)

- [delete-customer-gateway](#) (AWS CLI)
- [Remove-EC2CustomerGateway](#) (AWS Tools for Windows PowerShell)

Detach and delete a virtual private gateway

If you no longer require a virtual private gateway for your VPC, you can detach it from the VPC.

To detach a virtual private gateway using the console

1. In the navigation pane, choose **Virtual private gateways**.
2. Select the virtual private gateway and choose **Actions, Detach from VPC**.
3. Choose **Detach virtual private gateway**.

If you no longer require a detached virtual private gateway, you can delete it. You can't delete a virtual private gateway that's still attached to a VPC. After you delete your virtual private gateway, it remains visible for a short while with a state of `deleted`, and then the entry is automatically removed.

To delete a virtual private gateway using the console

1. In the navigation pane, choose **Virtual private gateways**.
2. Select the virtual private gateway and choose **Actions, Delete virtual private gateway**.
3. When prompted for confirmation, enter **delete** and then choose **Delete**.

To detach a virtual private gateway using the command line or API

- [DetachVpnGateway](#) (Amazon EC2 Query API)
- [detach-vpn-gateway](#) (AWS CLI)
- [Dismount-EC2VpnGateway](#) (AWS Tools for Windows PowerShell)

To delete a virtual private gateway using the command line or API

- [DeleteVpnGateway](#) (Amazon EC2 Query API)
- [delete-vpn-gateway](#) (AWS CLI)
- [Remove-EC2VpnGateway](#) (AWS Tools for Windows PowerShell)

Modify the target gateway of a Site-to-Site VPN connection

You can modify the target gateway of a AWS Site-to-Site VPN connection. The following migration options are available:

- An existing virtual private gateway to a transit gateway
- An existing virtual private gateway to another virtual private gateway
- An existing transit gateway to another transit gateway
- An existing transit gateway to a virtual private gateway

After you modify the target gateway, your Site-to-Site VPN connection will be temporarily unavailable for a brief period while we provision the new endpoints.

The following tasks help you complete the migration to a new gateway.

Tasks

- [Step 1: Create the new target gateway](#)
- [Step 2: Delete your static routes \(conditional\)](#)
- [Step 3: Migrate to a new gateway](#)
- [Step 4: Update VPC route tables](#)
- [Step 5: Update the target gateway routing \(conditional\)](#)
- [Step 6: Update the customer gateway ASN \(conditional\)](#)

Step 1: Create the new target gateway

Before you perform the migration to the new target gateway, you must first configure the new gateway. For information about adding a virtual private gateway, see [the section called "Create a virtual private gateway"](#). For more information about adding a transit gateway, see [Create a transit gateway](#) in *Amazon VPC Transit Gateways*.

If the new target gateway is a transit gateway, attach the VPCs to the transit gateway. For information about VPC attachments, see [Transit gateway attachments to a VPC](#) in *Amazon VPC Transit Gateways*.

When you modify the target from a virtual private gateway to a transit gateway, you can optionally set the transit gateway ASN to be the same value as the virtual private gateway ASN. If you choose

to have a different ASN, then you must set the ASN on your customer gateway device to the transit gateway ASN. For more information, see [the section called “Step 6: Update the customer gateway ASN \(conditional\)”](#).

Step 2: Delete your static routes (conditional)

This step is required when you migrate from a virtual private gateway with static routes to a transit gateway.

You must delete the static routes before you migrate to the new gateway.

Tip

Keep a copy of the static route before you delete it. You will need to add back these routes to the transit gateway after the VPN connection migration is complete.

To delete a route from a route table

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Route tables**, and then select the route table.
3. On the **Routes** tab, choose **Edit routes**.
4. Choose **Remove** for the static route to the virtual private gateway.
5. Choose **Save changes**.

Step 3: Migrate to a new gateway

To change the target gateway

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN connections**.
3. Select the VPN connection and choose **Actions, Modify VPN connection**.
4. For **Target type**, choose the gateway type.
 - a. If the new target gateway is a virtual private gateway, choose **VPN gateway**.
 - b. If the new target gateway is transit gateway, choose **Transit gateway**.

5. Choose **Save changes**.

To modify a Site-to-Site VPN connection using the command line or API

- [ModifyVpnConnection](#) (Amazon EC2 Query API)
- [modify-vpn-connection](#) (AWS CLI)

Step 4: Update VPC route tables

After you migrate to the new gateway, you might need to modify your VPC route table. For more information, see [Route tables](#) in the *Amazon VPC User Guide*.

The following table provides information about the VPC route table updates to make after you modify the VPN gateway target.

Existing gateway	New gateway	VPC route table change
Virtual private gateway with propagated routes	Transit gateway	Add a route that contains the ID of the transit gateway.
Virtual private gateway with propagated routes	Virtual private gateway with propagated routes	There is no action required.
Virtual private gateway with propagated routes	Virtual private gateway with static route	Add a route that contains the ID of the new virtual private gateway.
Virtual private gateway with static routes	Transit gateway	Update the route that contains the ID of the virtual private gateway to the ID of the transit gateway.
Virtual private gateway with static routes	Virtual private gateway with static routes	Update the route that contains the ID of the virtual private gateway to the ID of the new virtual private gateway.

Existing gateway	New gateway	VPC route table change
Virtual private gateway with static routes	Virtual private gateway with propagated routes	Delete the route that contains the ID of the virtual private gateway.
Transit gateway	Virtual private gateway with static routes	Update the route that contains the ID of the transit gateway to the ID of the virtual private gateway.
Transit gateway	Virtual private gateway with propagated routes	Delete the route that contains the ID of the transit gateway.
Transit gateway	Transit gateway	Update the route that contains the ID of the transit gateway to the ID of the new transit gateway.

Step 5: Update the target gateway routing (conditional)

When the new gateway is a transit gateway, modify the transit gateway route table to allow traffic between the VPC and the Site-to-Site VPN. For more information, see [Transit gateway route tables](#) in *Amazon VPC Transit Gateways*.

If you deleted VPN static routes, you must add the static routes to the transit gateway route table.

Unlike a virtual private gateway, a transit gateway sets the same value for the multi-exit discriminator (MED) across all the tunnels on a VPN attachment. If you are migrating from a virtual private gateway to a transit gateway and relied on the MED value for tunnel selection, we recommend that you make routing changes to avoid connection issues. For example, you can advertise more specific routes on your transit gateway. For more information, see [Route tables and VPN route priority](#).

Step 6: Update the customer gateway ASN (conditional)

When the new gateway has a different ASN from the old gateway, you must update the ASN on your customer gateway device to point to the new ASN. See [Customer gateway options for your Site-to-Site VPN connection](#) for more information.

Modify Site-to-Site VPN connection options

You can modify the connection options for your Site-to-Site VPN connection. You can modify the following options:

- The IPv4 CIDR ranges on the local (customer gateway) side and the remote (AWS) side of the VPN connection that can communicate over the VPN tunnels. The default is `0.0.0.0/0` for both ranges.
- The IPv6 CIDR ranges on the local (customer gateway) and the remote (AWS) side of the VPN connection that can communicate over the VPN tunnels. The default is `::/0` for both ranges.

When you modify the VPN connection options, the VPN endpoint IP addresses on the AWS side do not change, and the tunnel options do not change. Your VPN connection will be temporarily unavailable for a brief period while the VPN connection is updated.

To modify the VPN connection options using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN connections**.
3. Select your VPN connection, and choose **Actions, Modify VPN connection options**.
4. Enter new CIDR ranges as needed.
5. Choose **Save changes**.

To modify the VPN connection options using the command line or API

- [modify-vpn-connection-options](#) (AWS CLI)
- [ModifyVpnConnectionOptions](#) (Amazon EC2 Query API)

Modify Site-to-Site VPN tunnel options

You can modify the tunnel options for the VPN tunnels in your Site-to-Site VPN connection. You can modify one VPN tunnel at a time.

Important

When you modify a VPN tunnel, connectivity over the tunnel is interrupted for up to several minutes. Ensure that you plan for the expected downtime.

To modify the VPN tunnel options using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN connections**.
3. Select the Site-to-Site VPN connection, and choose **Actions, Modify VPN tunnel options**.
4. For **VPN tunnel outside IP address**, choose the tunnel endpoint IP of the VPN tunnel.
5. Choose or enter new values for the tunnel options as needed. For more information, see [VPN tunnel options](#).
6. Choose **Save changes**.

To modify the VPN tunnel options using the command line or API

- (AWS CLI) Use [describe-vpn-connections](#) to view the current tunnel options, and [modify-vpn-tunnel-options](#) to modify the tunnel options.
- (Amazon EC2 Query API) Use [DescribeVpnConnections](#) to view the current tunnel options, and [ModifyVpnTunnelOptions](#) to modify the tunnel options.

Edit static routes for a Site-to-Site VPN connection

For a Site-to-Site VPN connection on a virtual private gateway that's configured for static routing, you can add or remove static routes from your VPN configuration.

To add or remove a static route using the console

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

2. In the navigation pane, choose **Site-to-Site VPN connections**.
3. Select the VPN connection.
4. Choose **Edit static routes**.
5. Add or remove routes as needed.
6. Choose **Save changes**.
7. If you have not enabled route propagation for your route table, you must manually update the routes in your route table to reflect the updated static IP prefixes in your VPN connection. For more information, see [\(Virtual private gateway\) Enable route propagation in your route table](#).
8. For a VPN connection on a transit gateway, you add, modify, or remove the static routes in the transit gateway route table. For more information, see [Transit gateway route tables](#) in *Amazon VPC Transit Gateways*.

To add a static route using the command line or API

- [CreateVpnConnectionRoute](#) (Amazon EC2 Query API)
- [create-vpn-connection-route](#) (AWS CLI)
- [New-EC2VpnConnectionRoute](#) (AWS Tools for Windows PowerShell)

To delete a static route using the command line or API

- [DeleteVpnConnectionRoute](#) (Amazon EC2 Query API)
- [delete-vpn-connection-route](#) (AWS CLI)
- [Remove-EC2VpnConnectionRoute](#) (AWS Tools for Windows PowerShell)

Change the customer gateway for a Site-to-Site VPN connection

You can change the customer gateway of your Site-to-Site VPN connection by using the Amazon VPC console or a command line tool.

After you change the customer gateway, your VPN connection will be temporarily unavailable for a brief period while we provision the new endpoints.

To change the customer gateway using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN connections**.
3. Select the VPN connection.
4. Choose **Actions, Modify VPN connection**.
5. For **Target type**, choose **Customer gateway**.
6. For **Target customer gateway**, choose the new customer gateway.
7. Choose **Save changes**.

To change the customer gateway using the command line or API

- [ModifyVpnConnection](#) (Amazon EC2 Query API)
- [modify-vpn-connection](#) (AWS CLI)

Replace compromised credentials for your Site-to-Site VPN connection

If you believe that the tunnel credentials for your Site-to-Site VPN connection have been compromised, you can change the IKE pre-shared key or change the ACM certificate. The method you use depends on the authentication option you used for your VPN tunnels. For more information, see [Site-to-Site VPN tunnel authentication options](#).

To change the IKE pre-shared key

You can modify the tunnel options for the VPN connection and specify a new IKE pre-shared key for each tunnel. For more information, see [Modify Site-to-Site VPN tunnel options](#).

Alternatively, you can delete the VPN connection. For more information, see [Delete a VPN connection](#). You don't need to delete the VPC or the virtual private gateway. Then, create a new VPN connection using the same virtual private gateway, and configure the new keys on your customer gateway device. You can specify your own pre-shared keys for the tunnels or let AWS generate new pre-shared keys for you. For more information, see [Create a VPN connection](#). The tunnel's inside and outside addresses might change when you recreate the VPN connection.

To change the certificate for the AWS side of the tunnel endpoint

Rotate the certificate. For more information, see [Rotate VPN tunnel endpoint certificates](#).

To change the certificate on the customer gateway device

1. Create a new certificate. For information, see [Issuing and managing certificates](#) in the *AWS Certificate Manager User Guide*.
2. Add the certificate to the customer gateway device.

Rotate Site-to-Site VPN tunnel endpoint certificates

You can rotate the certificates on the tunnel endpoints on the AWS side by using the Amazon VPC console. When a tunnel endpoint's certificate is close to expiration, AWS automatically rotates the certificate using the service-linked role. For more information, see [the section called "Service-linked roles"](#).

To rotate the Site-to-Site VPN tunnel endpoint certificate using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN connections**.
3. Select the Site-to-Site VPN connection, and then choose **Actions, Modify VPN tunnel certificate**.
4. Select the tunnel endpoint.
5. Choose **Save**.

To rotate the Site-to-Site VPN tunnel endpoint certificate using the AWS CLI

Use the [modify-vpn-tunnel-certificate](#) command.

Private IP VPN with AWS Direct Connect

With private IP VPN, you can deploy IPsec VPN over AWS Direct Connect, encrypting traffic between your on-premises network and AWS, without the use of public IP addresses or additional third-party VPN equipment.

One of the main use cases for private IP VPN over AWS Direct Connect is helping customers in the financial, healthcare, and federal industries meet regulatory and compliance goals. Private IP

VPN over AWS Direct Connect ensures that traffic between AWS and on-premises networks is both secure and private, allowing customers to comply with their regulatory and security mandates.

Contents

- [Benefits of private IP VPN](#)
- [How private IP VPN works](#)
- [Prerequisites](#)
- [Create the customer gateway](#)
- [Prepare the transit gateway](#)
- [Create the AWS Direct Connect gateway](#)
- [Create the transit gateway association](#)
- [Create the VPN connection](#)

Benefits of private IP VPN

- **Simplified network management and operations:** Without private IP VPN, customers have to deploy third-party VPN and routers to implement private VPNs over AWS Direct Connect networks. With private IP VPN capability, customers don't have to deploy and manage their own VPN infrastructure. This leads to simplified network operations and reduced costs.
- **Improved security posture:** Previously, customers had to use a public AWS Direct Connect virtual interface (VIF) for encrypting traffic over AWS Direct Connect, which requires public IP addresses for VPN endpoints. Using public IPs increases the probability of external (DOS) attacks, which in turn compels customers to deploy additional security gear for network protection. Also, a public VIF opens access between all AWS public services and customer on-premises networks, increasing the severity of the risk. The private IP VPN feature allows encryption over AWS Direct Connect transit VIFs (instead of public VIFs), coupled with the ability to configure private IPs. This provides end-to-end private connectivity in addition to encryption, improving the overall security posture.
- **Higher route scale:** Private IP VPN connections offer higher route limits (5000 outbound routes and 1000 inbound routes) as compared to AWS Direct Connect alone, which currently has a limit of 200 outbound and 100 inbound routes.

How private IP VPN works

Private IP Site-to-Site VPN works over an AWS Direct Connect transit virtual interface (VIF). It uses an AWS Direct Connect gateway and a transit gateway to interconnect your on-premises networks with AWS VPCs. A private IP VPN connection has termination points at the transit gateway on the AWS side, and at your customer gateway device on the on-premises side. You must assign private IP addresses (RFC1918) to both the transit gateway and the customer gateway device ends of the IPsec tunnels.

You attach a private IP VPN connection to a transit gateway. You then route traffic between the VPN attachment and any VPCs (or other networks) that are also attached to the transit gateway. You do that by associating a route table with the VPN attachment. In the reverse direction, you can route traffic from your VPCs to the private IP VPN attachment by using route tables that are associated with the VPCs.

The route table that's associated with the VPN attachment can be the same or different from the one associated with the underlying AWS Direct Connect attachment. This gives you the ability to route both encrypted and unencrypted traffic simultaneously between your VPCs and your on-premises networks.

For more details on the traffic path leaving the VPN, see [Private virtual interface and transit virtual interface routing policies](#) in the *AWS Direct Connect User Guide*.

Prerequisites

The following resources are needed to complete the setup of a private IP VPN over AWS Direct Connect:

- An AWS Direct Connect connection between your on-premises network and AWS
- An AWS Direct Connect gateway with an association with the appropriate transit gateway
- A transit gateway with an available private IP CIDR block
- A customer gateway device in your on-premises network and a corresponding AWS customer gateway

Create the customer gateway

A *customer gateway* is a resource that you create in AWS. It represents the customer gateway device in your on-premises network. When you create a customer gateway, you provide information about your device to AWS. For more details, see [Customer gateway](#).

To create a customer gateway using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Customer gateways**.
3. Choose **Create customer gateway**.
4. (Optional) For **Name tag**, enter a name for your customer gateway. Doing so creates a tag with a key of Name and the value that you specify.
5. For **BGP ASN**, enter a Border Gateway Protocol (BGP) Autonomous System Number (ASN) for your customer gateway.
6. For **IP address**, enter the private IP address for your customer gateway device.
7. (Optional) For **Device**, enter a name for the device that hosts this customer gateway.
8. Choose **Create customer gateway**.

To create a customer gateway using the command line or API

- [CreateCustomerGateway](#) (Amazon EC2 Query API)
- [create-customer-gateway](#) (AWS CLI)

Prepare the transit gateway

A transit gateway is a network transit hub that you can use to interconnect your VPCs and on-premises networks. You can create a new transit gateway or use an existing one for the private IP VPN connection. When you create the transit gateway, or modify an existing transit gateway, you specify a private IP CIDR block for the connection.

Note

When specifying the transit gateway CIDR block to be associated with your Private IP VPN, ensure the CIDR block does not overlap with any IP addresses for any other network

attachments on the transit gateway. If any IP CIDR blocks do overlap, it may cause configuration issues with your customer gateway device.

For specific AWS console steps to create or modify a transit gateway to use for the private IP VPN, see [Transit gateways](#) in the *Amazon VPC Transit Gateways Guide*.

To create a transit gateway using the command line or API

- [CreateTransitGateway](#) (Amazon EC2 Query API)
- [create-transit-gateway](#) (AWS CLI)

Create the AWS Direct Connect gateway

Create an AWS Direct Connect gateway by following the [Creating a Direct Connect gateway](#) procedure in the *AWS Direct Connect User Guide*.

To create an AWS Direct Connect gateway using the command line or API

- [CreateDirectConnectGateway](#) (AWS Direct Connect Query API)
- [create-direct-connect-gateway](#) (AWS CLI)

Create the transit gateway association

After creating the AWS Direct Connect gateway, create a transit gateway association for the AWS Direct Connect gateway. Specify the private IP CIDR for the transit gateway that was identified earlier in the allowed prefixes list.

For more information, see [Transit Gateway associations](#) in the *AWS Direct Connect User Guide*.

To create an AWS Direct Connect gateway association using the command line or API

- [CreateDirectConnectGatewayAssociation](#) (AWS Direct Connect Query API)
- [create-direct-connect-gateway-association](#) (AWS CLI)

Create the VPN connection

To create a VPN connection using private IP addresses

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN connections**.
3. Choose **Create VPN connection**.
4. (Optional) For **Name tag**, enter a name for your Site-to-Site VPN connection. Doing so creates a tag with a key of Name and the value that you specify.
5. For **Target gateway type**, choose **Transit gateway**. Then, choose the transit gateway that you identified earlier.
6. For **Customer gateway**, select **Existing**. Then, choose the customer gateway that you created earlier.
7. Select one of the routing options based on whether your customer gateway device supports Border Gateway Protocol (BGP):
 - If your customer gateway device supports BGP, choose **Dynamic (requires BGP)**.
 - If your customer gateway device does not support BGP, choose **Static**.
8. For **Tunnel inside IP version**, specify whether the VPN tunnels support IPv4 or IPv6 traffic.
9. (Optional) If you specified **IPv4** for **Tunnel inside IP Version**, you can optionally specify the IPv4 CIDR ranges for the customer gateway and AWS sides that are allowed to communicate over the VPN tunnels. The default is `0.0.0.0/0`.

If you specified **IPv6** for **Tunnel inside IP version**, you can optionally specify the IPv6 CIDR ranges for the customer gateway and AWS sides that are allowed to communicate over the VPN tunnels. The default for both ranges is `::/0`.
10. For **Outside IP address type**, choose **PrivateIpv4**.
11. For **Transport attachment ID**, choose the transit gateway attachment for the appropriate AWS Direct Connect gateway.
12. Choose **Create VPN connection**.

Note

The **Enable acceleration** option is not applicable for VPN connections over AWS Direct Connect.

Security in AWS Site-to-Site VPN

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

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the [AWS Compliance Programs](#). To learn about the compliance programs that apply to AWS Site-to-Site VPN, see [AWS Services in Scope by Compliance Program](#).
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

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

Contents

- [Data protection in AWS Site-to-Site VPN](#)
- [Identity and access management for AWS Site-to-Site VPN](#)
- [Resilience in AWS Site-to-Site VPN](#)
- [Infrastructure security in AWS Site-to-Site VPN](#)

Data protection in AWS Site-to-Site VPN

The AWS [shared responsibility model](#) applies to data protection in AWS Site-to-Site VPN. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks

for the AWS services that you use. For more information about data privacy, see the [Data Privacy FAQ](#). For information about data protection in Europe, see the [AWS Shared Responsibility Model and GDPR](#) blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-2](#).

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Site-to-Site VPN or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Internetwork traffic privacy

A Site-to-Site VPN connection privately connects your VPC to your on-premises network. Data that's transferred between your VPC and your network routes over an encrypted VPN connection to help maintain the confidentiality and integrity of the data in transit. Amazon supports Internet Protocol security (IPsec) VPN connections. IPsec is a protocol suite for securing IP communications by authenticating and encrypting each IP packet in a data stream.

Each Site-to-Site VPN connection consists of two encrypted IPsec VPN tunnels that link AWS and your network. Traffic in each tunnel can be encrypted with AES128 or AES256 and use Diffie-

Hellman groups for key exchange, providing Perfect Forward Secrecy. AWS authenticates with SHA1 or SHA2 hashing functions.

Instances in your VPC do not require a public IP address to connect to resources on the other side of your Site-to-Site VPN connection. Instances can route their internet traffic through the Site-to-Site VPN connection to your on-premises network. They can then access the internet through your existing outbound traffic points and your network security and monitoring devices.

See the following topics for more information:

- [Tunnel options for your Site-to-Site VPN connection](#): Provides information about the IPsec and Internet Key Exchange (IKE) options that are available for each tunnel.
- [Site-to-Site VPN tunnel authentication options](#): Provides information about the authentication options for your VPN tunnel endpoints.
- [Requirements for your customer gateway device](#): Provides information about the requirements for the customer gateway device on your side of the VPN connection.
- [Providing secure communication between sites using VPN CloudHub](#): If you have multiple Site-to-Site VPN connections, you can provide secure communication between your on-premises sites by using the AWS VPN CloudHub.

Identity and access management for AWS Site-to-Site VPN

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

Topics

- [Audience](#)
- [Authenticating with identities](#)
- [Managing access using policies](#)
- [How AWS Site-to-Site VPN works with IAM](#)
- [Identity-based policy examples for AWS Site-to-Site VPN](#)
- [Troubleshooting AWS Site-to-Site VPN identity and access](#)
- [Using service-linked roles for Site-to-Site VPN](#)

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Site-to-Site VPN.

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

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

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Site-to-Site VPN. To view example Site-to-Site VPN identity-based policies that you can use in IAM, see [Identity-based policy examples for AWS Site-to-Site VPN](#).

Authenticating with identities

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

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

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

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

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

AWS account root user

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

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

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

IAM users and groups

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

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

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

IAM roles

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

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see [Creating a role for a third-party Identity Provider](#) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see [Permission sets](#) in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.

- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
 - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see [Forward access sessions](#).
 - **Service role** – A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.
 - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using an IAM role to grant permissions to applications running on Amazon EC2 instances](#) in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see [When to create an IAM role \(instead of a user\)](#) in the *IAM User Guide*.

Managing access using policies

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

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

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

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

Identity-based policies

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

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

Resource-based policies

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

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

Access control lists (ACLs)

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

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

Other policy types

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

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see [Permissions boundaries for IAM entities](#) in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If

you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see [How SCPs work](#) in the *AWS Organizations User Guide*.

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

Multiple policy types

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

How AWS Site-to-Site VPN works with IAM

Before you use IAM to manage access to Site-to-Site VPN, learn what IAM features are available to use with Site-to-Site VPN.

IAM features you can use with AWS Site-to-Site VPN

IAM feature	Site-to-Site VPN support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	No
ABAC (tags in policies)	No

IAM feature	Site-to-Site VPN support
Temporary credentials	Yes
Principal permissions	Yes
Service roles	Yes
Service-linked roles	Yes

To get a high-level view of how Site-to-Site VPN and other AWS services work with most IAM features, see [AWS services that work with IAM](#) in the *IAM User Guide*.

Identity-based policies for Site-to-Site VPN

Supports identity-based policies	Yes
----------------------------------	-----

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

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

Identity-based policy examples for Site-to-Site VPN

To view examples of Site-to-Site VPN identity-based policies, see [Identity-based policy examples for AWS Site-to-Site VPN](#).

Resource-based policies within Site-to-Site VPN

Supports resource-based policies	No
----------------------------------	----

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

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.

Policy actions for Site-to-Site VPN

Supports policy actions	Yes
-------------------------	-----

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

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

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

To see a list of Site-to-Site VPN actions, see [Actions defined by AWS Site-to-Site VPN](#) in the *Service Authorization Reference*.

Policy actions in Site-to-Site VPN use the following prefix before the action:

```
ec2
```

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

```
"Action": [  
    "ec2:action1",  
    "ec2:action2"  
]
```

To view examples of Site-to-Site VPN identity-based policies, see [Identity-based policy examples for AWS Site-to-Site VPN](#).

Policy resources for Site-to-Site VPN

Supports policy resources	Yes
---------------------------	-----

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

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

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

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

To see a list of Site-to-Site VPN resource types and their ARNs, see [Resources defined by AWS Site-to-Site VPN](#) in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see [Actions defined by AWS Site-to-Site VPN](#).

To view examples of Site-to-Site VPN identity-based policies, see [Identity-based policy examples for AWS Site-to-Site VPN](#).

Policy condition keys for Site-to-Site VPN

Supports service-specific policy condition keys	Yes
---	-----

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

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

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical AND operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical OR operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see [IAM policy elements: variables and tags](#) in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see [AWS global condition context keys](#) in the *IAM User Guide*.

To see a list of Site-to-Site VPN condition keys, see [Condition keys for AWS Site-to-Site VPN](#) in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see [Actions defined by AWS Site-to-Site VPN](#).

To view examples of Site-to-Site VPN identity-based policies, see [Identity-based policy examples for AWS Site-to-Site VPN](#).

ACLs in Site-to-Site VPN

Supports ACLs	No
---------------	----

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

ABAC with Site-to-Site VPN

Supports ABAC (tags in policies)	No
----------------------------------	----

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

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

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

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

For more information about ABAC, see [What is ABAC?](#) in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see [Use attribute-based access control \(ABAC\)](#) in the *IAM User Guide*.

Using temporary credentials with Site-to-Site VPN

Supports temporary credentials	Yes
--------------------------------	-----

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

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

switch roles. For more information about switching roles, see [Switching to a role \(console\)](#) in the *IAM User Guide*.

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

Cross-service principal permissions for Site-to-Site VPN

Supports forward access sessions (FAS)	Yes
--	-----

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see [Forward access sessions](#).

Service roles for Site-to-Site VPN

Supports service roles	Yes
------------------------	-----

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

Warning

Changing the permissions for a service role might break Site-to-Site VPN functionality. Edit service roles only when Site-to-Site VPN provides guidance to do so.

Service-linked roles for Site-to-Site VPN

Supports service-linked roles Yes

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

For details about creating or managing service-linked roles, see [AWS services that work with IAM](#). Find a service in the table that includes a Yes in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for AWS Site-to-Site VPN

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

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

For details about actions and resource types defined by Site-to-Site VPN, including the format of the ARNs for each of the resource types, see [Actions, resources, and condition keys for AWS Site-to-Site VPN](#) in the *Service Authorization Reference*.

Topics

- [Policy best practices](#)
- [Using the Site-to-Site VPN console](#)
- [Describe specific Site-to-Site VPN connections](#)
- [Create and describe resources needed for a AWS Site-to-Site VPN connection](#)

Policy best practices

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

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see [AWS managed policies](#) or [AWS managed policies for job functions](#) in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see [Policies and permissions in IAM](#) in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see [IAM JSON policy elements: Condition](#) in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see [IAM Access Analyzer policy validation](#) in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see [Configuring MFA-protected API access](#) in the *IAM User Guide*.

For more information about best practices in IAM, see [Security best practices in IAM](#) in the *IAM User Guide*.

Using the Site-to-Site VPN console

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

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

To ensure that users and roles can still use the Site-to-Site VPN console, also attach the Site-to-Site VPN `AmazonVPCFullAccess` or `AmazonVPCReadOnlyAccess` AWS managed policy to the entities. For more information, see [Adding permissions to a user](#) in the *IAM User Guide*.

Describe specific Site-to-Site VPN connections

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpnConnections"
      ],
      "Resource": [
        "arn:aws:ec2:us-west-2:123456789012:vpn-connection/vpn-04d5cc9b88example",
        "arn:aws:ec2:us-west-2:123456789012:vpn-connection/vpn-903004f88example"
      ]
    }
  ]
}
```

Create and describe resources needed for a AWS Site-to-Site VPN connection

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

```

        "ec2:DescribeVpnConnections",
        "ec2:DescribeVpnGateways",
        "ec2:DescribeCustomerGateways",
        "ec2:CreateCustomerGateway",
        "ec2:CreateVpnGateway",
        "ec2:CreateVpnConnection"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "arn:aws:iam::*:role/aws-service-role/s2svpn.amazonaws.com/
AWSServiceRoleForVPCS2SVPNInternal",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "s2svpn.amazonaws.com"
        }
    }
}
]
}

```

Troubleshooting AWS Site-to-Site VPN identity and access

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

Topics

- [I am not authorized to perform an action in Site-to-Site VPN](#)
- [I am not authorized to perform iam:PassRole](#)
- [I want to allow people outside of my AWS account to access my Site-to-Site VPN resources](#)

I am not authorized to perform an action in Site-to-Site VPN

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

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional `my-example-widget` resource but doesn't have the fictional `ec2:GetWidget` permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
ec2:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the `my-example-widget` resource by using the `ec2:GetWidget` action.

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

I am not authorized to perform iam:PassRole

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

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

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

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

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

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

I want to allow people outside of my AWS account to access my Site-to-Site VPN resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support

resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Site-to-Site VPN supports these features, see [How AWS Site-to-Site VPN works with IAM](#).
- To learn how to provide access to your resources across AWS accounts that you own, see [Providing access to an IAM user in another AWS account that you own](#) in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see [Providing access to AWS accounts owned by third parties](#) in the *IAM User Guide*.
- To learn how to provide access through identity federation, see [Providing access to externally authenticated users \(identity federation\)](#) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.

Using service-linked roles for Site-to-Site VPN

AWS Site-to-Site VPN uses AWS Identity and Access Management (IAM) [service-linked roles](#). A service-linked role is a unique type of IAM role that is linked directly to Site-to-Site VPN. Service-linked roles are predefined by Site-to-Site VPN and include all the permissions that the service requires to call other AWS services on your behalf.

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

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

For information about other services that support service-linked roles, see [AWS Services That Work with IAM](#) and look for the services that have **Yes** in the **Service-linked roles** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for Site-to-Site VPN

Site-to-Site VPN uses the service-linked role named **AWSServiceRoleForVPCS2SVPN** – Allow Site-to-Site VPN to create and manage resources related to your VPN connections.

The AWSServiceRoleForVPCS2SVPN service-linked role trusts the following services to assume the role:

- AWS Certificate Manager
- AWS Private Certificate Authority

The role permissions policy named AWSVPCS2SVpnServiceRolePolicy allows Site-to-Site VPN to complete the following actions on the specified resources:

- Action: `acm:ExportCertificate` on Resource: `"*"`
- Action: `acm:DescribeCertificate` on Resource: `"*"`
- Action: `acm:ListCertificates` on Resource: `"*"`
- Action: `acm-pca:DescribeCertificateAuthority` on Resource: `"*"`

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*.

Creating a service-linked role for Site-to-Site VPN

You don't need to manually create a service-linked role. When you create a customer gateway with an associated ACM private certificate in the AWS Management Console, the AWS CLI, or the AWS API, Site-to-Site VPN creates the service-linked role for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you create a customer gateway with an associated ACM private certificate, Site-to-Site VPN creates the service-linked role for you again.

Editing a service-linked role for Site-to-Site VPN

Site-to-Site VPN does not allow you to edit the AWSServiceRoleForVPCS2SVPN service-linked role. After you create a service-linked role, you cannot change the name of the role because various

entities might reference the role. However, you can edit the description of the role using IAM. For more information, see [Editing a service-linked role](#) in the *IAM User Guide*.

Deleting a service-linked role for Site-to-Site VPN

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

Note

If the Site-to-Site VPN service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete Site-to-Site VPN resources used by the `AWSServiceRoleForVPCS2SVPN`

You can delete this service-linked role only after you delete all customer gateways that have an associated ACM private certificate. This ensures that you cannot inadvertently remove permission to access your ACM certificates in use by Site-to-Site VPN connections.

To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the `AWSServiceRoleForVPCS2SVPN` service-linked role. For more information, see [Deleting a service-linked role](#) in the *IAM User Guide*.

Resilience in AWS Site-to-Site VPN

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see [AWS Global Infrastructure](#).

In addition to the AWS global infrastructure, Site-to-Site VPN offers features to help support your data resiliency and backup needs.

Two tunnels per VPN connection

A Site-to-Site VPN connection consists of two tunnels, each terminating in a different Availability Zone, to provide increased availability to your VPC. If there's a device failure within AWS, your VPN connection automatically fails over to the second tunnel so that your access isn't interrupted. From time to time, AWS also performs routine maintenance on your VPN connection, which may briefly disable one of the two tunnels of your VPN connection. For more information, see [Site-to-Site VPN tunnel endpoint replacements](#). When you configure your customer gateway, it's therefore important that you configure both tunnels.

Redundancy

To protect against a loss of connectivity in case your customer gateway becomes unavailable, you can set up a second Site-to-Site VPN connection. For more information, see the following documentation:

- [Using redundant Site-to-Site VPN connections to provide failover](#)
- [Amazon Virtual Private Cloud Connectivity Options](#)
- [Building a Scalable and Secure Multi-VPC AWS Network Infrastructure](#)

Infrastructure security in AWS Site-to-Site VPN

As a managed service, AWS Site-to-Site VPN is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see [AWS Cloud Security](#). To design your AWS environment using the best practices for infrastructure security, see [Infrastructure Protection](#) in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access Site-to-Site VPN through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the [AWS Security Token Service](#) (AWS STS) to generate temporary security credentials to sign requests.

Monitoring your Site-to-Site VPN connection

Monitoring is an important part of maintaining the reliability, availability, and performance of your AWS Site-to-Site VPN connection. You should collect monitoring data from all of the parts of your solution so that you can more easily debug a multi-point failure if one occurs. Before you start monitoring your Site-to-Site VPN connection; however, you should create a monitoring plan that includes answers to the following questions:

- What are your monitoring goals?
- What resources will you monitor?
- How often will you monitor these resources?
- What monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

The next step is to establish a baseline for normal VPN performance in your environment, by measuring performance at various times and under different load conditions. As you monitor your VPN, store historical monitoring data so that you can compare it with current performance data, identify normal performance patterns and performance anomalies, and devise methods to address issues.

To establish a baseline, you should monitor the following items:

- The state of your VPN tunnels
- Data into the tunnel
- Data out of the tunnel

Contents

- [Monitoring tools](#)
- [AWS Site-to-Site VPN logs](#)
- [Monitoring VPN tunnels using Amazon CloudWatch](#)
- [Monitoring VPN connections using AWS Health events](#)

Monitoring tools

AWS provides various tools that you can use to monitor a Site-to-Site VPN connection. You can configure some of these tools to do the monitoring for you, while some of the tools require manual intervention. We recommend that you automate monitoring tasks as much as possible.

Automated monitoring tools

You can use the following automated monitoring tools to watch a Site-to-Site VPN connection and report when something is wrong:

- **Amazon CloudWatch Alarms** – Watch a single metric over a time period that you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon SNS topic. CloudWatch alarms do not invoke actions simply because they are in a particular state; the state must have changed and been maintained for a specified number of periods. For more information, see [Monitoring VPN tunnels using Amazon CloudWatch](#).
- **AWS CloudTrail Log Monitoring** – Share log files between accounts, monitor CloudTrail log files in real time by sending them to CloudWatch Logs, write log processing applications in Java, and validate that your log files have not changed after delivery by CloudTrail. For more information, see [Logging API Calls Using AWS CloudTrail](#) in the *Amazon EC2 API Reference* and [Working with CloudTrail log files](#) in the *AWS CloudTrail User Guide*.
- **AWS Health events** – Receive alerts and notifications related to changes in the health of your Site-to-Site VPN tunnels, best practice configuration recommendations, or when approaching scaling limits. Use events on the [Personal Health Dashboard](#) to trigger automated failovers, reduce troubleshooting time, or optimize connections for high availability. For more information, see [Monitoring VPN connections using AWS Health events](#).

Manual monitoring tools

Another important part of monitoring a Site-to-Site VPN connection involves manually monitoring those items that the CloudWatch alarms don't cover. The Amazon VPC and CloudWatch console dashboards provide an at-a-glance view of the state of your AWS environment.

Note

In the Amazon VPC console, Site-to-Site VPN tunnel state parameters such as "Status" and "Last status change", may not reflect transient state changes or momentary tunnel flaps. It is recommended to use CloudWatch metrics and logs for granular tunnel state change updates.

- The Amazon VPC dashboard shows:
 - Service health by Region
 - Site-to-Site VPN connections
 - VPN tunnel status (In the navigation pane, choose **Site-to-Site VPN Connections**, select a Site-to-Site VPN connection, and then choose **Tunnel Details**)
- The CloudWatch home page shows:
 - Current alarms and status
 - Graphs of alarms and resources
 - Service health status

In addition, you can use CloudWatch to do the following:

- Create [customized dashboards](#) to monitor the services you care about
- Graph metric data to troubleshoot issues and discover trends
- Search and browse all your AWS resource metrics
- Create and edit alarms to be notified of problems

AWS Site-to-Site VPN logs

AWS Site-to-Site VPN logs provide you with deeper visibility into your Site-to-Site VPN deployments. With this feature, you have access to Site-to-Site VPN connection logs that provide details on IP Security (IPsec) tunnel establishment, Internet Key Exchange (IKE) negotiations, and dead peer detection (DPD) protocol messages.

Site-to-Site VPN logs can be published to Amazon CloudWatch Logs. This feature provides customers with a single consistent way to access and analyze detailed logs for all of their Site-to-Site VPN connections.

Contents

- [Benefits of Site-to-Site VPN logs](#)
- [Amazon CloudWatch Logs resource policy size restrictions](#)
- [Contents of Site-to-Site VPN logs](#)
- [IAM requirements to publish to CloudWatch Logs](#)
- [View Site-to-Site VPN logs configuration](#)
- [Enable Site-to-Site VPN logs](#)
- [Disable Site-to-Site VPN logs](#)

Benefits of Site-to-Site VPN logs

- **Simplified VPN troubleshooting:** Site-to-Site VPN logs help you to pinpoint configuration mismatches between AWS and your customer gateway device, and address initial VPN connectivity issues. VPN connections can intermittently flap over time due to misconfigured settings (such as poorly tuned timeouts), there can be issues in the underlying transport networks (like internet weather), or routing changes or path failures can cause disruption of connectivity over VPN. This feature allows you to accurately diagnose the cause of intermittent connection failures and fine-tune low-level tunnel configuration for reliable operation.
- **Centralized AWS Site-to-Site VPN visibility:** Site-to-Site VPN logs can provide tunnel activity logs for all of the different ways that Site-to-Site VPN is connected: Virtual Gateway, Transit Gateway, and CloudHub, using both internet and AWS Direct Connect as transport. This feature provides customers with a single consistent way to access and analyze detailed logs for all of their Site-to-Site VPN connections.
- **Security and compliance:** Site-to-Site VPN logs can be sent to Amazon CloudWatch Logs for retrospective analysis of VPN connection status and activity over time. This can help you meet compliance and regulatory requirements.

Amazon CloudWatch Logs resource policy size restrictions

CloudWatch Logs resource policies are limited to 5120 characters. When CloudWatch Logs detects that a policy approaches this size limit, it automatically enables log groups that start with `/aws/vendedlogs/`. When you enable logging, Site-to-Site VPN must update your CloudWatch Logs resource policy with the log group you specify. To avoid reaching the CloudWatch Logs resource policy size limit, prefix your log group names with `/aws/vendedlogs/`.

Contents of Site-to-Site VPN logs

The following information is included in the Site-to-Site VPN tunnel activity log.

Field	Description
VpnLogCreationTimestamp	Log creation timestamp in human readable format.
VpnConnectionId	The VPN connection identifier.
TunnelOutsideIPAddress	The external IP of the VPN tunnel that generated the log entry.
TunnelDPDEnabled	Dead Peer Detection Protocol Enabled Status (True/False).
TunnelCGWNATTDetectionStatus	NAT-T detected on customer gateway device (True/False).
TunnelIKEPhase1State	IKE Phase 1 Protocol State (Established Rekeying Negotiating Down).
TunnelIKEPhase2State	IKE Phase 2 Protocol State (Established Rekeying Negotiating Down).
VpnLogDetail	Verbose messages for IPsec, IKE and DPD protocols.

Contents

- [IKEv1 Error Messages](#)
- [IKEv2 Error Messages](#)
- [IKEv2 Negotiation Messages](#)

IKEv1 Error Messages

Message	Explanation
Peer is not responsive - Declaring peer dead	Peer has not responded to DPD Messages, enforcing DPD time-out action.
AWS tunnel payload decryption was unsuccessful due to invalid Pre-shared Key	Same Pre-Shared key needs to be configured on both IKE Peers.
No Proposal Match Found by AWS	Proposed Attributes for Phase 1 (Encryption, Hashing and DH Group) are not supported by AWS VPN Endpoint. e.g 3DES
No Proposal Match Found. Notifying with "No proposal chosen"	No Proposal Chosen error message is exchanged between Peers to inform that correct Proposals/Policies must be configured for phase 2 on IKE Peers.
AWS tunnel received DELETE for Phase 2 SA with SPI: xxxx	CGW has sent the Delete_SA message for Phase 2
AWS tunnel received DELETE for IKE_SA from CGW	CGW has sent the Delete_SA message for Phase 1

IKEv2 Error Messages

Message	Explanation
AWS tunnel DPD timed out after {retry_count} retransmits	Peer has not responded to DPD Messages, enforcing DPD time-out action.
AWS tunnel received DELETE for IKE_SA from CGW	Peer has sent the Delete_SA message for Parent/IKE_SA
AWS tunnel received DELETE for Phase 2 SA with SPI: xxxx	Peer has sent the Delete_SA message for CHILD_SA

Message	Explanation
AWS tunnel detected a (CHILD_REKEY) collision as CHILD_DELETE	CGW has sent the Delete_SA message for the Active SA, which is being rekeyed.
AWS tunnel (CHILD_SA) redundant SA is being deleted due to detected collision	Due to Collision, If redundant SAs are generated, Peers will close redundant SA after matching the nonce values as per RFC
AWS tunnel Phase 2 was unable to establish while keeping Phase 1	Peer was unable to establish CHILD_SA due to negotiation error e.g incorrect proposal.
AWS: Traffic Selector: TS_UNACCEPTABLE: received from responder	Peer has proposed Incorrect Traffic Selectors/ Encryption Domain. Peers should be configured with identical and correct CIDRs.
AWS tunnel is sending AUTHENTICATION_FAILED as the response	Peer is unable to Authenticate the Peer by verifying IKE_AUTH message's contents
AWS tunnel detected a pre-shared key mismatch with cgw: xxxx	Same Pre-Shared key needs to be configured on both IKE Peers.
AWS tunnel Timeout: deleting un-established Phase 1 IKE_SA with cgw: xxxx	Deleting the half-opened IKE_SA as peer has not proceeded with negotiations
No Proposal Match Found. Notifying with "No proposal chosen"	No Proposal Chosen error message is exchanged between Peers to inform that correct Proposals must be configured on IKE Peers.
No Proposal Match Found by AWS	Proposed Attributes for Phase 1 (Encryption, Hashing and DH Group) are not supported by AWS VPN Endpoint. e.g 3DES

IKEv2 Negotiation Messages

Message	Explanation
AWS tunnel processed request (id=xxx) for CREATE_CHILD_SA	AWS has received the CREATE_CHILD_SA request from CGW
AWS tunnel is sending response (id=xxx) for CREATE_CHILD_SA	AWS is sending CREATE_CHILD_SA response to CGW
AWS tunnel is sending request (id=xxx) for CREATE_CHILD_SA	AWS is sending CREATE_CHILD_SA request to CGW
AWS tunnel processed response (id=xxx) for CREATE_CHILD_SA	AWS has received CREATE_CHILD_SA response from CGW

IAM requirements to publish to CloudWatch Logs

For the logging feature to work properly, the IAM policy attached to the IAM principal being used to configure the feature, must include the following permissions at minimum. More details can also be found in the [Enabling logging from certain AWS services](#) section of the *Amazon CloudWatch Logs User Guide*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs>DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "S2SVPNLogging"
    }
  ]
}
```

```
    },
    {
      "Sid": "S2SVPNLoggingCWL",
      "Action": [
        "logs:PutResourcePolicy",
        "logs:DescribeResourcePolicies",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

View Site-to-Site VPN logs configuration

To view current tunnel logging settings

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN Connections**.
3. Select the VPN connection that you want to view from the **VPN connections** list.
4. Choose the **Tunnel details** tab.
5. Expand the **Tunnel 1 options** and **Tunnel 2 options** sections to view all tunnel configuration details.
6. You can view the current status of the logging feature under **Tunnel VPN log**, and the currently configured CloudWatch log group (if any) under **CloudWatch log group**.

To view current tunnel logging settings on a Site-to-Site VPN connection using the AWS command line or API

- [DescribeVpnConnections](#) (Amazon EC2 Query API)
- [describe-vpn-connections](#) (AWS CLI)

Enable Site-to-Site VPN logs

Note

When you enable Site-to-Site VPN logs for an existing VPN connection tunnel, your connectivity over that tunnel can be interrupted for several minutes. However, each VPN connection offers two tunnels for high availability, so you can enable logging on one tunnel at a time while maintaining connectivity over the tunnel not being modified. For more information, see [Site-to-Site VPN tunnel endpoint replacements](#).

To enable VPN logging during creation of a new Site-to-Site VPN connection

Follow the procedure [Step 5: Create a VPN connection](#). During Step 9 **Tunnel Options**, you can specify all the options you want to use for both tunnels, including **VPN logging** options. For more information about these options, see [Tunnel options for your Site-to-Site VPN connection](#).

To enable tunnel logging on a new Site-to-Site VPN connection using the AWS command line or API

- [CreateVpnConnection](#) (Amazon EC2 Query API)
- [create-vpn-connection](#) (AWS CLI)

To enable tunnel logging on an existing Site-to-Site VPN connection

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN Connections**.
3. Select the VPN connection that you want to modify from the **VPN connections** list.
4. Select **Actions, Modify VPN tunnel options**.
5. Select the tunnel that you want to modify by choosing the appropriate IP address from the **VPN tunnel outside IP address** list.
6. Under **Tunnel activity log**, select **Enable**.
7. Under **Amazon CloudWatch log group**, select the Amazon CloudWatch log group where you want the logs to be sent.
8. (Optional) Under **Output format**, choose the desired format for the log output, either **json** or **text**.

9. Select **Save changes**.
10. (Optional) Repeat steps 4 through 9 for the other tunnel if desired.

To enable tunnel logging on an existing Site-to-Site VPN connection using the AWS command line or API

- [ModifyVpnTunnelOptions](#) (Amazon EC2 Query API)
- [modify-vpn-tunnel-options](#) (AWS CLI)

Disable Site-to-Site VPN logs

To disable tunnel logging on a Site-to-Site VPN connection

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Site-to-Site VPN Connections**.
3. Select the VPN connection that you want to modify from the **VPN connections** list.
4. Select **Actions, Modify VPN tunnel options**.
5. Select the tunnel that you want to modify by choosing the appropriate IP address from the **VPN tunnel outside IP address** list.
6. Under **Tunnel activity log**, clear **Enable**.
7. Select **Save changes**.
8. (Optional) Repeat steps 4 through 7 for the other tunnel if desired.

To disable tunnel logging on a Site-to-Site VPN connection using the AWS command line or API

- [ModifyVpnTunnelOptions](#) (Amazon EC2 Query API)
- [modify-vpn-tunnel-options](#) (AWS CLI)

Monitoring VPN tunnels using Amazon CloudWatch

You can monitor VPN tunnels using CloudWatch, which collects and processes raw data from the VPN service into readable, near real-time metrics. These statistics are recorded for a period of 15 months, so that you can access historical information and gain a better perspective on how your

web application or service is performing. VPN metric data is automatically sent to CloudWatch as it becomes available.

For more information, see the [Amazon CloudWatch User Guide](#).

Contents

- [VPN metrics and dimensions](#)
- [Viewing VPN CloudWatch metrics](#)
- [Creating CloudWatch alarms to monitor VPN tunnels](#)

VPN metrics and dimensions

The following CloudWatch metrics are available for your Site-to-Site VPN connections.

Metric	Description
TunnelState	<p>The state of the tunnels. For static VPNs, 0 indicates DOWN and 1 indicates UP. For BGP VPNs, 1 indicates ESTABLISHED and 0 is used for all other states. For both types of VPNs, values between 0 and 1 indicate at least one tunnel is not UP.</p> <p>Units: Fractional value between 0 and 1</p>
TunnelDataIn †	<p>The bytes received on the AWS side of the connection through the VPN tunnel from a customer gateway. Each metric data point represents the number of bytes received after the previous data point. Use the Sum statistic to show the total number of bytes received during the period.</p> <p>This metric counts the data after decryption.</p> <p>Units: Bytes</p>

Metric	Description
TunnelDataOut †	<p>The bytes sent from the AWS side of the connection through the VPN tunnel to the customer gateway. Each metric data point represents the number of bytes sent after the previous data point. Use the Sum statistic to show the total number of bytes sent during the period.</p> <p>This metric counts the data before encryption.</p> <p>Units: Bytes</p>

† These metrics can report network usage even when the tunnel is down. This is due to periodic status checks performed on the tunnel, and background ARP and BGP requests.

To filter the metric data, use the following dimensions.

Dimension	Description
VpnId	Filters the metric data by the Site-to-Site VPN connection ID.
TunnelIpAddress	Filters the metric data by the IP address of the tunnel for the virtual private gateway.

Viewing VPN CloudWatch metrics

When you create a Site-to-Site VPN connection, the VPN service sends metrics about your VPN connection to CloudWatch, as they become available. You can view the metrics for your VPN connection as follows.

To view metrics using the CloudWatch console

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.

2. In the navigation pane, choose **Metrics**.
3. Under **All metrics**, choose the **VPN** metric namespace.
4. Select the metric dimension to view the metrics (for example, **VPN Tunnel Metrics**).

 **Note**

The VPN namespace will not appear in the CloudWatch console until after a Site-to-Site VPN connection has been created in the AWS region you are viewing.

To view metrics using the AWS CLI

At a command prompt, use the following command:

```
aws cloudwatch list-metrics --namespace "AWS/VPN"
```

Creating CloudWatch alarms to monitor VPN tunnels

You can create a CloudWatch alarm that sends an Amazon SNS message when the alarm changes state. An alarm watches a single metric over a time period you specify, and sends a notification to an Amazon SNS topic based on the value of the metric relative to a given threshold over a number of time periods.

For example, you can create an alarm that monitors the state of a single VPN tunnel, and sends a notification when the tunnel state is DOWN for 3 datapoints within 15 minutes.

To create an alarm for a single tunnel state

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, expand **Alarms**, then choose **All alarms**.
3. Choose **Create alarm**, then choose **Select metric**.
4. Choose **VPN**, then **VPN Tunnel Metrics**.
5. Select the IP address of the desired tunnel, on the same line with the **TunnelState** metric. Choose **Select metric**.
6. For **Whenever TunnelState is...**, select **Lower**, and then enter "1" in the input field under **than....**

7. Under **Additional configuration**, set the inputs to "3 out of 3" for **Datapoints to alarm**.
8. Choose **Next**.
9. Under **Send a notification to the following SNS topic**, select an existing notification list or create a new one.
10. Choose **Next**.
11. Enter a name for your alarm. Choose **Next**.
12. Check the settings for your alarm, and then choose **Create alarm**.

You can create an alarm that monitors the state of the Site-to-Site VPN connection. For example, you can create an alarm that sends a notification when the status of one or both tunnels is DOWN for one 5-minute period.

To create an alarm for Site-to-Site VPN connection state

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, expand **Alarms**, then choose **All alarms**.
3. Choose **Create alarm**, then choose **Select metric**.
4. Choose **VPN**, then choose **VPN Connection Metrics**.
5. Select your Site-to-Site VPN connection and the **TunnelState** metric. Choose **Select metric**.
6. For **Statistic**, specify **Maximum**.

Alternatively, if you've configured your Site-to-Site VPN connection so that both tunnels are up, you can specify a statistic of **Minimum** to send a notification when at least one tunnel is down.

7. For **Whenever**, choose **Lower/Equal (<=)** and enter **0** (or **0.5** for when at least one tunnel is down). Choose **Next**.
8. Under **Select an SNS topic**, select an existing notification list or choose **New list** to create a new one. Choose **Next**.
9. Enter a name and description for your alarm. Choose **Next**.
10. Check the settings for your alarm, and then choose **Create alarm**.

You can also create alarms that monitor the amount of traffic coming in or leaving the VPN tunnel. For example, the following alarm monitors the amount of traffic coming into the VPN tunnel from

your network, and sends a notification when the number of bytes reaches a threshold of 5,000,000 during a 15 minute period.

To create an alarm for incoming network traffic

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, expand **Alarms**, then choose **All alarms**.
3. Choose **Create alarm**, then choose **Select metric**.
4. Choose **VPN**, then choose **VPN Tunnel Metrics**.
5. Select the IP address of the VPN tunnel and the **TunnelDataIn** metric. Choose **Select metric**.
6. For **Statistic**, specify **Sum**.
7. For **Period**, select **15 minutes**.
8. For **Whenever**, choose **Greater/Equal(>=)** and enter **5000000**. Choose **Next**.
9. Under **Select an SNS topic**, select an existing notification list or choose **New list** to create a new one. Choose **Next**.
10. Enter a name and description for your alarm. Choose **Next**.
11. Check the settings for your alarm, and then choose **Create alarm**.

The following alarm monitors the amount of traffic leaving the VPN tunnel to your network, and sends a notification when the number of bytes is less than 1,000,000 during a 15 minute period.

To create an alarm for outgoing network traffic

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, expand **Alarms**, then choose **All alarms**.
3. Choose **Create alarm**, then choose **Select metric**.
4. Choose **VPN**, then choose **VPN Tunnel Metrics**.
5. Select the IP address of the VPN tunnel and the **TunnelDataOut** metric. Choose **Select metric**.
6. For **Statistic**, specify **Sum**.
7. For **Period**, select **15 minutes**.
8. For **Whenever**, choose **Lower/Equal (<=)** and enter **1000000**. Choose **Next**.
9. Under **Select an SNS topic**, select an existing notification list or choose **New list** to create a new one. Choose **Next**.
10. Enter a name and description for your alarm. Choose **Next**.

11. Check the settings for your alarm, and then choose **Create alarm**.

For more examples of creating alarms, see [Creating Amazon CloudWatch alarms](#) in the *Amazon CloudWatch User Guide*.

Monitoring VPN connections using AWS Health events

AWS Site-to-Site VPN automatically sends notifications to the AWS [AWS Health Dashboard](#) (PHD), which is powered by the AWS Health API. This dashboard requires no setup, and is ready to use for authenticated AWS users. You can configure multiple actions in response to event notifications through the AWS Health Dashboard.

The AWS Health Dashboard provides the following types of notifications for your VPN connections:

- [Tunnel endpoint replacement notifications](#)
- [Single tunnel VPN notifications](#)

Tunnel endpoint replacement notifications

You receive a **Tunnel endpoint replacement notification** in the AWS Health Dashboard when one or both of the VPN tunnel endpoints in your VPN connection is replaced. A tunnel endpoint is replaced when AWS performs tunnel updates, or when you modify your VPN connection. For more information, see [Site-to-Site VPN tunnel endpoint replacements](#).

When a tunnel endpoint replacement is complete, AWS sends the **Tunnel endpoint replacement notification** through a AWS Health Dashboard event.

Single tunnel VPN notifications

A Site-to-Site VPN connection consists of two tunnels for redundancy. We strongly recommend that you configure both tunnels for high availability. If your VPN connection has one tunnel up but the other is down for more than one hour in a day, you receive a **monthly VPN single tunnel notification** through an AWS Health Dashboard event. This event will be updated daily with any new VPN connections detected as single tunnel, with notifications sent weekly. A new event will be created each month, which will clear any VPN connections no longer detected as single tunnel.

Site-to-Site VPN quotas

Your AWS account has the following quotas, formerly referred to as limits, related to Site-to-Site VPN. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, and other quotas cannot be increased.

To request a quota increase for an adjustable quota, choose **Yes** in the **Adjustable** column. For more information, see [Requesting a quota increase](#) in the *Service Quotas User Guide*.

Site-to-Site VPN resources

Name	Default	Adjustable
Customer gateways per Region	50	Yes
Virtual private gateways per Region	5	Yes
Site-to-Site VPN connections per Region	50	Yes
Site-to-Site VPN connections per virtual private gateway	10	Yes
Accelerated Site-to-Site VPN connections per Region	10	Yes
Unassociated Site-to-Site VPN connections per Region	10	Yes

Note

Both Accelerated and Unassociated connections count towards the total Site-to-Site VPN connections per Region quota.

You can attach one virtual private gateway to a VPC at a time. To connect the same Site-to-Site VPN connection to multiple VPCs, we recommend that you explore using a transit gateway instead. For more information, see [Transit gateways](#) in *Amazon VPC Transit Gateways*.

Site-to-Site VPN connections on a transit gateway are subject to the total transit gateway attachments limit. For more information, see [Transit gateway quotas](#).

Routes

Advertised route sources include VPC routes, other VPN routes, and routes from AWS Direct Connect virtual interfaces. Advertised routes come from the route table that's associated with the VPN attachment.

Note

If you are using a virtual private gateway and route propagation is enabled on your VPC route table, both dynamic and static routes will automatically be added for your VPN connection, up to the limit of the VPC's route table. See [Amazon VPC quotas](#) in the *Amazon VPC User Guide* for further details.

Name	Default	Adjustable
Dynamic routes advertised from a customer gateway device to a Site-to-Site VPN connection on a virtual private gateway	100	No
Routes advertised from a Site-to-Site VPN connection on a virtual private gateway to a customer gateway device	1,000	No
Dynamic routes advertised from a customer gateway device to a Site-to-Site VPN connection on a transit gateway	1,000	No
Routes advertised from a Site-to-Site VPN connection on a transit gateway to a customer gateway device	5,000	No
Static routes from a customer gateway device to a Site-to-Site VPN connection on a virtual private gateway	100	No

Bandwidth and throughput

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.

Name	Default	Adjustable
Maximum bandwidth per VPN tunnel	Up to 1.25 Gbps	No
Maximum packets per second (PPS) per VPN tunnel	Up to 140,000	No

For Site-to-Site VPN connections on a transit gateway, you can use 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. For more information, see [Transit gateways](#).

Maximum transmission unit (MTU)

Site-to-Site VPN supports a maximum transmission unit (MTU) of 1446 bytes and a corresponding maximum segment size (MSS) of 1406 bytes. However, certain algorithms that use larger TCP headers can effectively reduce that maximum value. To avoid fragmentation, we recommend that you set the MTU and MSS based on the algorithms selected. For more details on MTU, MSS, and the optimal values, see [Best practices for your customer gateway device](#).

Jumbo frames are not supported. For more information, see [Jumbo frames](#) in the *Amazon EC2 User Guide for Linux Instances*.

A Site-to-Site VPN connection does not support Path MTU Discovery.

Additional quota resources

For quotas related to transit gateways, including the number of attachments on a transit gateway, see [Quotas for your transit gateways](#) in the *Amazon VPC Transit Gateways Guide*.

For additional VPC quotas, see [Amazon VPC quotas](#) in the *Amazon VPC User Guide*.

Document history for the Site-to-Site VPN User Guide

The following table describes the AWS Site-to-Site VPN User Guide updates.

Change	Description	Date
Classic VPN info removed	Removed info about classic VPN from guide.	January 19, 2023
VPN log example messages	Sample logs added for Site-to-Site VPN connections.	December 9, 2022
Updated Download Configuration utility	Site-to-Site VPN customers can generate configuration templates for compatible Customer Gateway (CGW) devices, making it easier to create VPN connections to AWS. This update adds support for Internet Key Exchange version 2 (IKEv2) parameters for many popular CGW devices and includes two new APIs — <code>GetVpnConnectionDeviceTypes</code> and <code>GetVpnConnectionDeviceSampleConfiguration</code> .	September 21, 2021
VPN connection notifications	Site-to-Site VPN automatically sends notifications about your VPN connection to the AWS Health Dashboard.	October 29, 2020
VPN tunnel initiation	You can configure your VPN tunnels so that AWS brings up the tunnels.	August 27, 2020

Modify VPN connection options	You can modify the connection options for your Site-to-Site VPN connection.	August 27, 2020
Additional security algorithms	You can apply additional security algorithms to your VPN tunnels.	August 14, 2020
IPv6 support	Your VPN tunnels can support IPv6 traffic inside the tunnels.	August 12, 2020
Merge AWS Site-to-Site VPN guides	This release merges the contents of the AWS Site-to-Site VPN Network Administrator Guide into this guide.	March 31, 2020
Accelerated AWS Site-to-Site VPN connections	You can enable acceleration for your AWS Site-to-Site VPN connection.	December 3, 2019
Modify AWS Site-to-Site VPN tunnel options	You can modify the options for a VPN tunnel in a AWS Site-to-Site VPN connection. You can also configure additional tunnel options.	August 29, 2019
AWS Private Certificate Authority private certificate support	You can use a private certificate from AWS Private Certificate Authority to authenticate your VPN.	August 15, 2019
New Site-to-Site VPN User Guide	This release separates the AWS Site-to-Site VPN (previously known as AWS Managed VPN) content from the <i>Amazon VPC User Guide</i> .	December 18, 2018

Modify the target gateway	You can modify the target gateway of AWS Site-to-Site VPN connection.	December 18, 2018
Custom ASN	When you create a virtual private gateway, you can specify the private Autonomous System Number (ASN) for the Amazon side of the gateway.	October 10, 2017
VPN tunnel options	You can specify inside tunnel CIDR blocks and custom pre-shared keys for your VPN tunnels.	October 3, 2017
VPN metrics	You can view CloudWatch metrics for your VPN connections.	May 15, 2017
VPN enhancements	A VPN connection now supports the AES 256-bit encryption function, SHA-256 hashing function, NAT traversal, and additional Diffie-Hellman groups during Phase 1 and Phase 2 of a connection. In addition, you can now use the same customer gateway IP address for each VPN connection that uses the same customer gateway device.	October 28, 2015

[VPN connections using static routing configuration](#)

You can create IPsec VPN connections to Amazon VPC using static routing configurations. Previously, VPN connections required the use of the Border Gateway Protocol (BGP). We now support both types of connections and you can now establish connectivity from devices that do not support BGP, including Cisco ASA and Microsoft Windows Server 2008 R2.

September 13, 2012

[Automatic route propagation](#)

You can now configure automatic propagation of routes from your VPN and AWS Direct Connect links to your VPC routing tables.

September 13, 2012

[AWS VPN CloudHub and redundant VPN connections](#)

You can securely communicate from one site to another with or without a VPC. You can use redundant VPN connections to provide a fault-tolerant connection to your VPC.

September 29, 2011