AWS Site-to-Site VPN

User Guide

AWS Site-to-Site VPN: User Guide

Table of Contents

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.

Your Site-to-Site VPN connection is either an AWS Classic VPN or an AWS VPN. For more information, see Site-to-Site VPN categories (p. 4).

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.
- **Virtual private gateway**: The VPN concentrator on the Amazon side of the Site-to-Site VPN connection. You use a virtual private gateway or a transit gateway as the gateway for the Amazon side of the Site-to-Site VPN connection.
- **Transit gateway**: A transit hub that can be used to interconnect your VPCs and on-premises networks. You use a transit gateway or virtual private gateway as the gateway for the Amazon side of the Site-to-Site VPN connection.

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.

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.

Pricing

For information about pricing, see VPN pricing.

How AWS Site-to-Site VPN works

Site-to-Site VPN Components

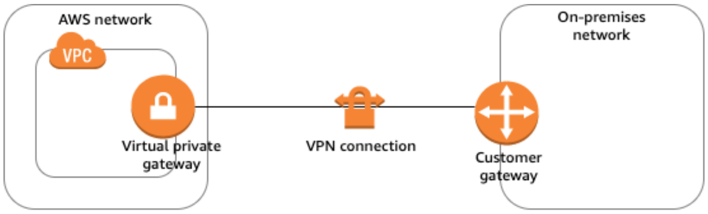
A Site-to-Site VPN connection offers two VPN tunnels between a virtual private gateway or a transit gateway on the AWS side, and a customer gateway (which represents a VPN device) on the remote (on-premises) side.

A Site-to-Site VPN connection consists of the following components. For more information about Site-to-Site VPN quotas, see Site-to-Site VPN quotas (p. 128).

Contents

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 the VPC from which you want to create the Site-to-Site VPN connection.

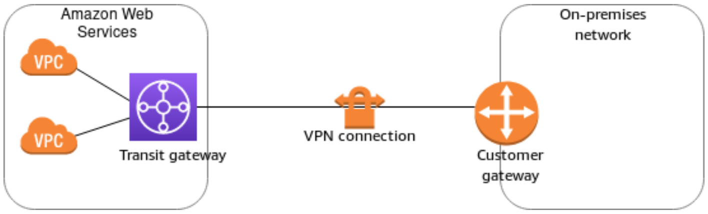


When you create a virtual private gateway, you can specify the private Autonomous System Number (ASN) for the Amazon side of the gateway. If you don't specify an ASN, the virtual private gateway is created with the default ASN (64512). You cannot change the ASN after you've created the virtual private gateway. To check the ASN for your virtual private gateway, view its details in the **Virtual Private Gateways** screen in the Amazon VPC console, or use the describe-vpn-gateways AWS CLI command.

> **Note**
> If you created your virtual private gateway before 2018-06-30, the default ASN is 17493 in the Asia Pacific (Singapore) region, 10124 in the Asia Pacific (Tokyo) region, 9059 in the Europe (Ireland) region, and 7224 in all other regions.

Transit gateway

A transit gateway is a transit hub that you can use to interconnect your virtual private clouds (VPC) and 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.



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 “Modifying a Site-to-Site VPN connection's target gateway” (p. 107).

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 (p. 32).

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 (p. 10).

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” (p. 12).

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 (p. 32).

IPv4 and IPv6 support

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 (p. 16).

Site-to-Site VPN categories

Your Site-to-Site VPN connection is either an AWS Classic VPN connection or an AWS VPN connection. Any new Site-to-Site VPN connection that you create is an AWS VPN connection. The following features are supported on AWS VPN connections only:

- Internet Key Exchange version 2 (IKEv2)
- NAT traversal
- 4-byte ASN (in addition to 2-byte ASN)
- 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

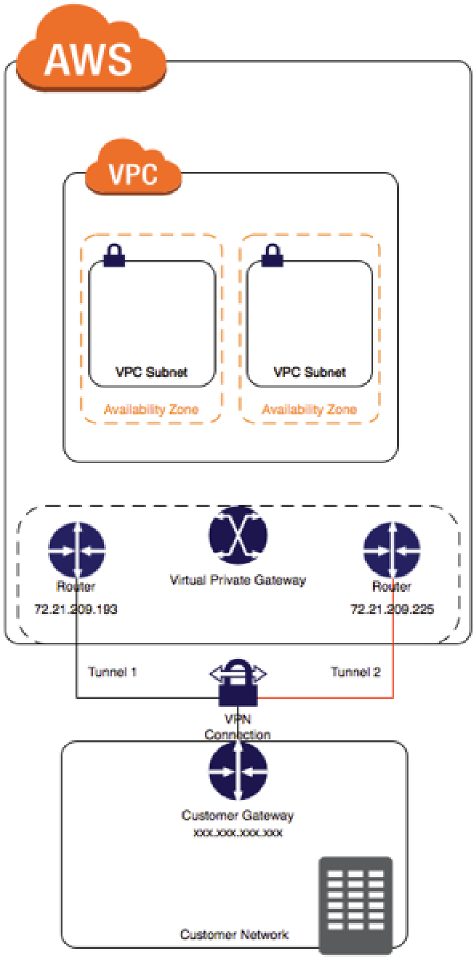
- Configurable tunnel options
- Custom private ASN for the Amazon side of a BGP session
- Private Certificate from a subordinate CA from AWS Certificate Manager Private Certificate Authority
- Support for IPv6 traffic for VPN connections on a transit gateway

For information about identifying and migrating your connection, see the section called "Identifying a Site-to-Site VPN connection" (p. 98) and the section called "Migrating from AWS Classic VPN to AWS VPN" (p. 99).

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 virtual private gateway 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 the Site-to-Site VPN connection.



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 duration, in seconds, after which DPD timeout occurs. You can specify 30 or higher.

Default: 30

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 (p. 10).

Default: `Clear`

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.

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: SHA-1, 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: SHA-1, 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 virtual private 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 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 seconds.

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

Default: 540 (9 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 (p. 10).

Default: `Add`

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. You cannot configure tunnel options for an AWS Classic VPN connection. For more information, see the following topics:

- Create a Site-to-Site VPN connection (p. 22)
- Modifying Site-to-Site VPN tunnel options (p. 110)

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* (p. 32).

Private certificate from AWS Certificate Manager Private Certificate Authority

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

You must create a private certificate from a subordinate CA using AWS Certificate Manager Private Certificate Authority (ACM 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 Certificate Manager Private Certificate Authority User Guide*.

You must create a service-link 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 "Permissions granted by the service-linked role" (p. 119).

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 your VPN tunnels.

- **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.

You can configure the IKE initiation options for one or both of the VPN tunnels in your Site-to-Site VPN connection.

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 Changing the customer gateway for a Site-to-Site VPN connection (p. 112).
- You cannot configure IKE initiation options for an AWS Classic VPN connection.
- IKE initiation (startup action) from the AWS side of the VPN connection is supported for IKEv2 only.
- 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: Create a Site-to-Site VPN connection (p. 22)
- To modify the VPN tunnel initiation options for an existing VPN connection: Modifying Site-to-Site VPN tunnel options (p. 110)

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.

If your tunnel endpoint has been replaced, AWS sends a notification through a Personal Health Dashboard event. For more information, see Monitoring VPN connections using AWS Health events (p. 127).

Endpoint replacements during VPN tunnel updates

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 a 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 which time 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.

Endpoint replacements during VPN connection modifications

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 (p. 107)	ModifyVpnConnection	Both tunnels are unavailable while new tunnel endpoints are provisioned.
Change the customer gateway for the VPN connection (p. 112)	ModifyVpnConnection	Both tunnels are unavailable while new tunnel endpoints are provisioned.
Modify the VPN connection options (p. 110)	ModifyVpnConnectionOptions	Both tunnels are unavailable while new tunnel endpoints are provisioned.

Modification	API action	Tunnel impact
Modify the VPN tunnel options (p. 110)	ModifyVpnTunnelOptions	The modified tunnel is unavailable during the update.

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) Internet-routable IP address (static) of the customer gateway device's external interface.	The public IP address value must be static. 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. This is not required when you are using a private certificate from AWS Certificate Manager Private Certificate Authority.
The type of routing—static or dynamic.	For more information, see Site-to-Site VPN routing options (p. 14).
(Dynamic routing only) Border Gateway Protocol (BGP) Autonomous System Number (ASN) of the customer gateway.	You can use an existing public ASN assigned to your network. If you don't have one, you can use a private ASN in the 64512–65534 range. The default ASN is 65000. Amazon EC2 supports all 4-byte ASN numbers in the range of 1 - 2147483647, with the exception of the following: • 7224 - reserved in the `us-east-1` Region • 9059 - reserved in the `eu-west-1` Region • 17943 - reserved in the `ap-southeast-1` Region • 10124 - reserved in the `ap-northeast-1` Region
(Optional) Private certificate from a subordinate CA using AWS Certificate Manager (ACM)	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. When you create a customer gateway, you can configure the customer gateway to use AWS Certificate Manager Private Certificate Authority private certificates to authenticate the Site-to-Site VPN.

Item	Description
	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 ACM Private CA. Before you create the customer gateway, you create a private certificate from a subordinate CA using AWS Certificate Manager 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 *AWS Certificate Manager Private Certificate Authority User Guide*.

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 Creating a transit gateway VPN attachment (p. 103).

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 (p. 23) from the Amazon VPC console, check the NAT-T setting and adjust it if necessary.
- IKE rekeys for accelerated VPN tunnels must be initiated from the customer gateway device to keep the tunnels up.
- 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.

Pricing

Hourly charges apply for a Site-to-Site VPN connection. For more information, see AWS VPN pricing. 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.

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

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 (p. 32).

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
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 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**
> 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 (p. 16) is used to determine tunnel priority.

- 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 (p. 16).

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.

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 (p. 127).

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 Create a Site-to-Site VPN connection (p. 22).

Getting started

Use the following procedures to manually set up the AWS Site-to-Site VPN connection. You can create a Site-to-Site VPN connection with either a virtual private gateway or a transit gateway as the target gateway.

To set up a Site-to-Site VPN connection, complete the following steps:

- Prerequisites (p. 18)
- Step 1: Create a customer gateway (p. 19)
- Step 2: Create a target gateway (p. 20)
- Step 3: Configure routing (p. 20)
- Step 4: Update your security group (p. 22)
- Step 5: Create a Site-to-Site VPN connection (p. 22)
- Step 6: Download the configuration file (p. 23)
- Step 7: Configure the customer gateway device (p. 24)

These procedures assume that you have a VPC with one or more subnets.

For steps to create a Site-to-Site VPN connection on a transit gateway, see Creating a transit gateway VPN attachment (p. 103).

Prerequisites

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

Item	Information
Customer gateway device	The physical or software device on your side of the VPN connection. You need the vendor (for example, Cisco), platform (for example, ISR Series Routers), and software version (for example, IOS 12.4).
Customer gateway	To create the customer gateway resource in AWS, you need the following information: • The internet-routable IP address for the device's external interface • The type of routing: static or dynamic (p. 14) • For dynamic routing, the Border Gateway Protocol (BGP) Autonomous System Number (ASN) • (Optional) Private certificate from AWS Certificate Manager Private Certificate Authority to authenticate your VPN

Item	Information
	For more information, see Customer gateway options for your Site-to-Site VPN connection (p. 12).
(Optional) The ASN for the AWS side of the BGP session	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. 3).
VPN connection	To create the VPN connection, you need the following information: • 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 (p. 5).

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 (p. 4).

If you plan to use a private certificate to authenticate your VPN, create a private certificate from a subordinate CA using AWS Certificate Manager Private Certificate Authority. For information about creating a private certificate, see Creating and managing a private CA in the *AWS Certificate Manager 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**, and then **Create Customer Gateway**.
3. Complete the following and then choose **Create Customer Gateway**:

   - (Optional) For **Name**, enter a name for your customer gateway. Doing so creates a tag with a key of `Name` and the value that you specify.
   - For **Routing**, select the routing type.
   - For dynamic routing, for **BGP ASN**, enter the Border Gateway Protocol (BGP) Autonomous System Number (ASN).
   - (Optional) For **IP Address**, enter the static, internet-routable IP address for your customer gateway device. If your customer gateway is behind a NAT device that's enabled for NAT-T, use the public IP address of the NAT device.
   - (Optional) If you want to use a private certificate, for **Certificate ARN**, choose the Amazon Resource Name of the private certificate.

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)

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 optionally specify the private Autonomous System Number (ASN) for the Amazon side of the gateway. This ASN must be different from the BGP 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**, **Create Virtual Private Gateway**.
2. (Optional) Enter a name for your virtual private gateway. Doing so creates a tag with a key of `Name` and the value that you specify.
3. For **ASN**, leave the default selection 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.
4. Choose **Create Virtual Private Gateway**.
5. Select the virtual private gateway that you created, and then choose **Actions**, **Attach to VPC**.
6. Select your VPC from the list and choose **Yes, Attach**.

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*.

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 Site-to-Site 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 Site-to-Site 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 Site-to-Site 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**, and then select the route table that's associated with the subnet. By default, this is the main route table for the VPC.
2. On the **Route Propagation** tab in the details pane, choose **Edit**, select the virtual private gateway that you created in the previous procedure, and then choose **Save**.

> **Note**
> For static routing, if you do not enable route propagation, you must manually enter the static routes used by your Site-to-Site 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**, and then select the route table that's associated with the subnet.
2. Choose **Route Propagation**, **Edit**. Clear the **Propagate** check box for the virtual private gateway, and 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. Choose the **Routes** tab, then choose **Edit routes**.
4. Choose **Add route**.
5. In the **Destination** column, enter the destination IP address range. For **Target**, choose the transit gateway.
6. Choose **Save routes**, and then choose **Close**.

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 inbound SSH, RDP and ICMP access

1. In the navigation pane, choose **Security Groups**, and then select the default security group for the VPC.
2. On the **Inbound** tab in the details pane, add rules that allow inbound SSH, RDP, and ICMP access from your network, and then choose **Save**. For more information about adding inbound rules, see Adding, removing, and updating rules in the *Amazon VPC User Guide.*

For more information about working with security groups using the AWS CLI, see Security groups for your VPC in the *Amazon VPC User Guide*.

Create a Site-to-Site VPN connection

Create the Site-to-Site VPN connection using the customer gateway and the virtual private gateway or transit gateway that you created earlier.

To create a Site-to-Site VPN connection

1. In the navigation pane, choose **Site-to-Site VPN Connections**, **Create VPN Connection**.
2. (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.
3. 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.
4. For **Customer Gateway ID**, select the customer gateway that you created earlier.
5. 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 Site-to-Site VPN connection.

6. (Optional) 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.
7. (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`.
8. (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.
   - Advanced tunnel information, which includes the following:
     - Encryption algorithms for phases 1 and 2 of the IKE negotiations
     - Integrity algorithms for phases 1 and 2 of the IKE negotiations
     - Diffie-Hellman groups for phases 1 and 2 of the IKE negotiations
     - IKE version
     - Phase 1 and 2 lifetimes
     - Rekey margin time
     - Rekey fuzz
     - Replay window size
     - Dead peer detection interval
     - Dead peer detection timeout action
     - Startup action

   For more information about these options, see Tunnel options for your Site-to-Site VPN connection (p. 5).
9. Choose **Create VPN Connection**. It might take a few minutes to create the Site-to-Site 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)
- New-EC2VpnConnection (AWS Tools for Windows PowerShell)

Download the configuration file

After you create the Site-to-Site VPN connection, download the configuration information and use it to configure the customer gateway device or software application.

Important

The configuration file is an example only and might not match your intended VPN connection settings. For example, it specifies the minimum requirements of IKE version 1, AES128, SHA1, and DH group 2 in most AWS Regions, and IKE version 1, AES128, SHA2, and DH group 14 in the AWS GovCloud Regions. It also specifies pre-shared keys for authentication (p. 9). You must modify the example configuration file to take advantage of IKE version 2, additional security algorithms and DH groups, and private certificates.

If you specified custom tunnel options when creating or modifying your Site-to-Site VPN connection, modify the example configuration file to match the custom settings for your tunnels.

The file also contains the value for the outside IP address for the virtual private gateway. This value is static unless you recreate the VPN connection in AWS.

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 the vendor, platform, and software that corresponds to your customer gateway device or software. If your device is not listed, choose **Generic**. Choose **Download**.

Configure the customer gateway device

Use the configuration file to configure your customer gateway device. The customer gateway device is the physical or software appliance on your side of the Site-to-Site VPN connection. For more information, see Your customer gateway device (p. 32).

Site-to-Site VPN architectures

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

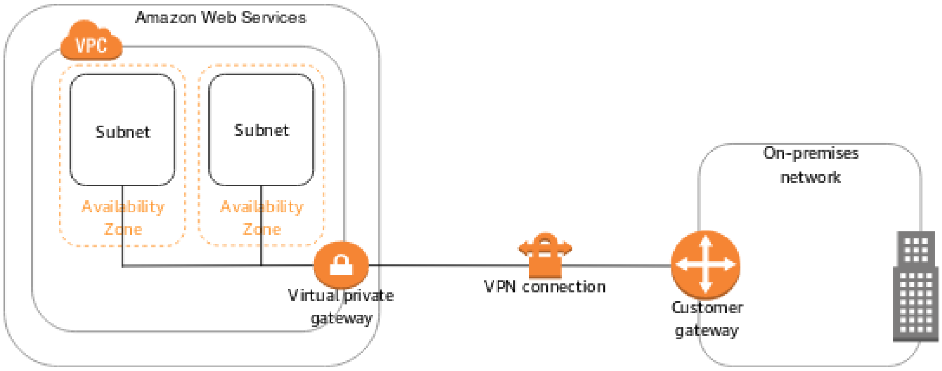
- the section called “Single and multiple connection examples” (p. 25)
- the section called “Using redundant Site-to-Site VPN connections to provide failover” (p. 29)
- the section called “AWS VPN CloudHub” (p. 27)

Site-to-Site VPN single and multiple connection examples

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

Single Site-to-Site VPN connection

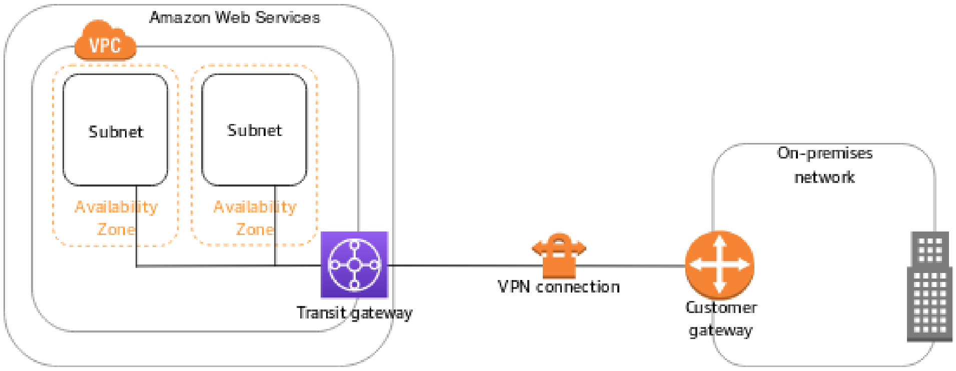
The VPC has an attached virtual private gateway, and your on-premises (remote) network includes a customer gateway device, which you must configure to enable the Site-to-Site VPN connection. You set up the routing so that any traffic from the VPC bound for your network is routed to the virtual private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



For steps to set up this scenario, see Getting started (p. 18).

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 Site-to-Site VPN connection. You set up the routing so that any traffic from the VPC bound for your network is routed to the transit gateway.

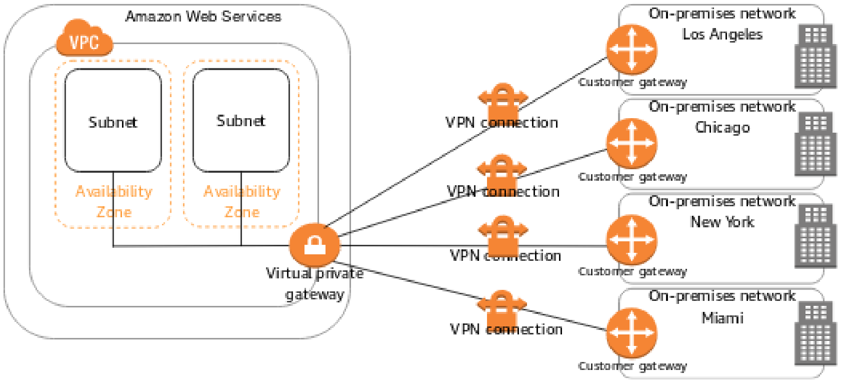


For steps to set up this scenario, see Getting started (p. 18).

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.

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 (p. 27).

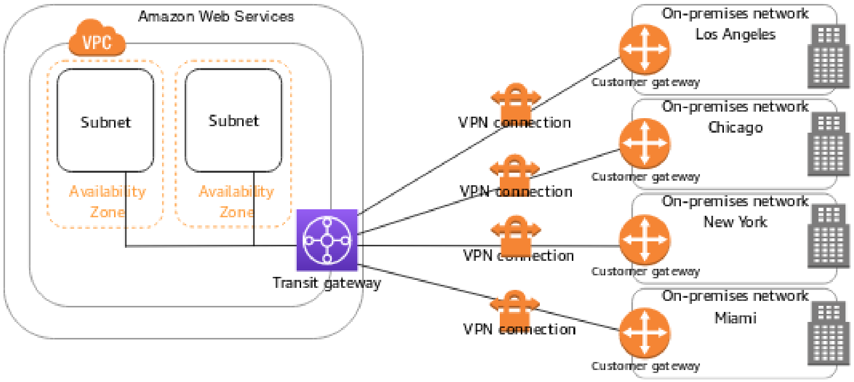


When you create multiple Site-to-Site VPN connections to a single VPC, you can configure a second customer gateway to create a redundant connection to the same external location. For more information, see Using redundant Site-to-Site VPN connections to provide failover (p. 29).

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.

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



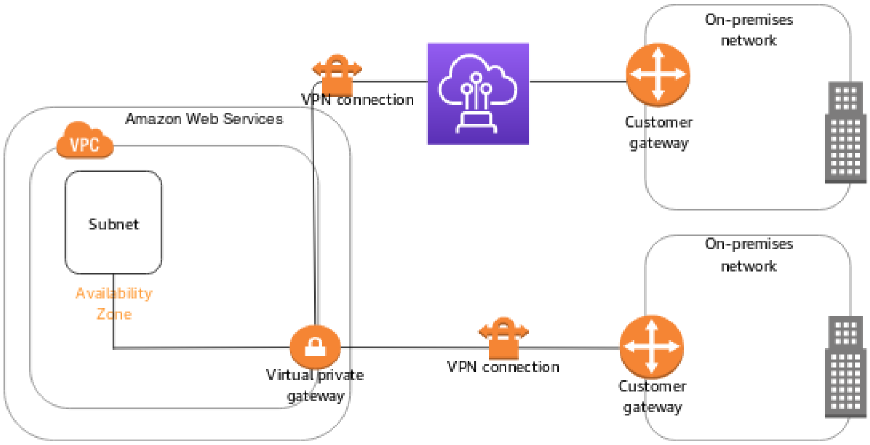
When you create multiple Site-to-Site VPN connections to a single transit gateway, you can configure a second customer gateway to create a redundant connection to the same external location.

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.

Note

When both AWS Direct Connect and the VPN connection are set up on the same virtul private gateway, adding or removing objects might cause the virtul 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 virtul private gateway returns to the ‘attached’ state.



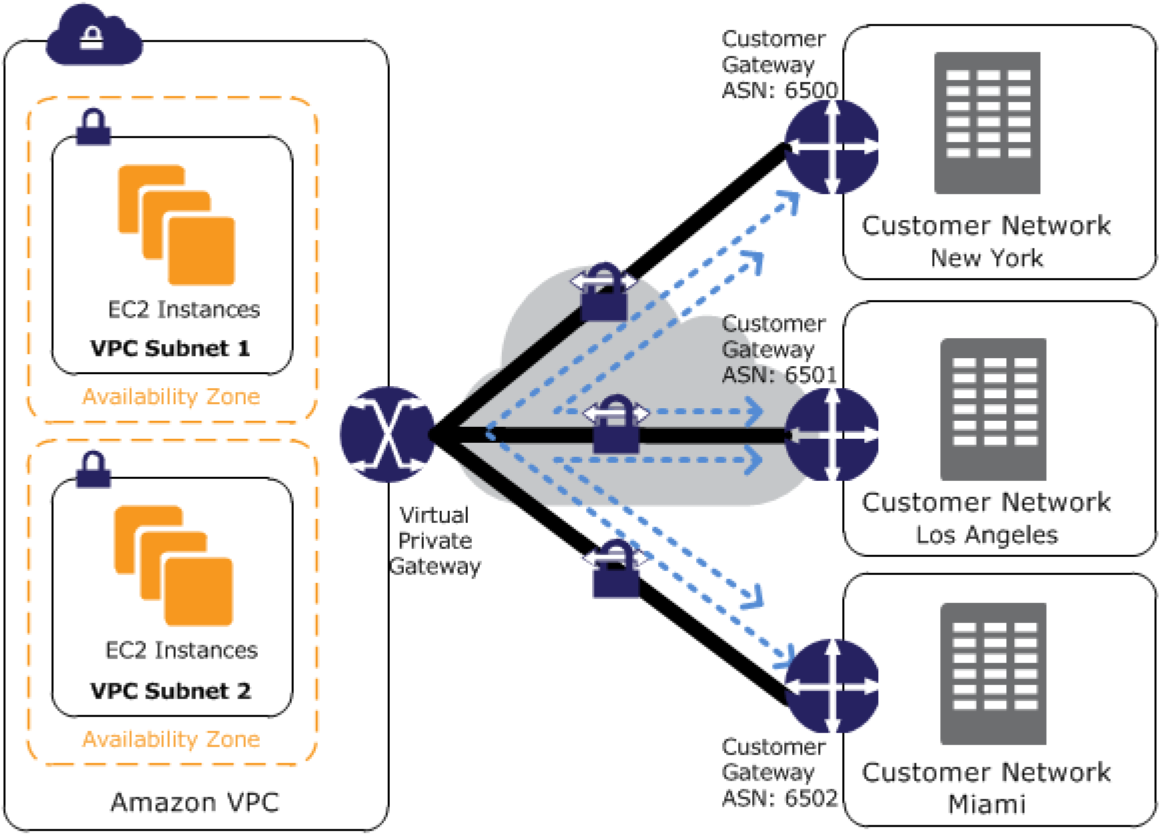
Providing secure communication between sites using VPN CloudHub

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 remote sites to communicate with each other, and not just with the 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 remote offices.

The sites must not have overlapping IP ranges.

Overview

The following diagram shows the VPN CloudHub architecture, with blue dashed lines indicating network traffic between remote sites being routed over their Site-to-Site VPN connections.



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 (p. 21). 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.

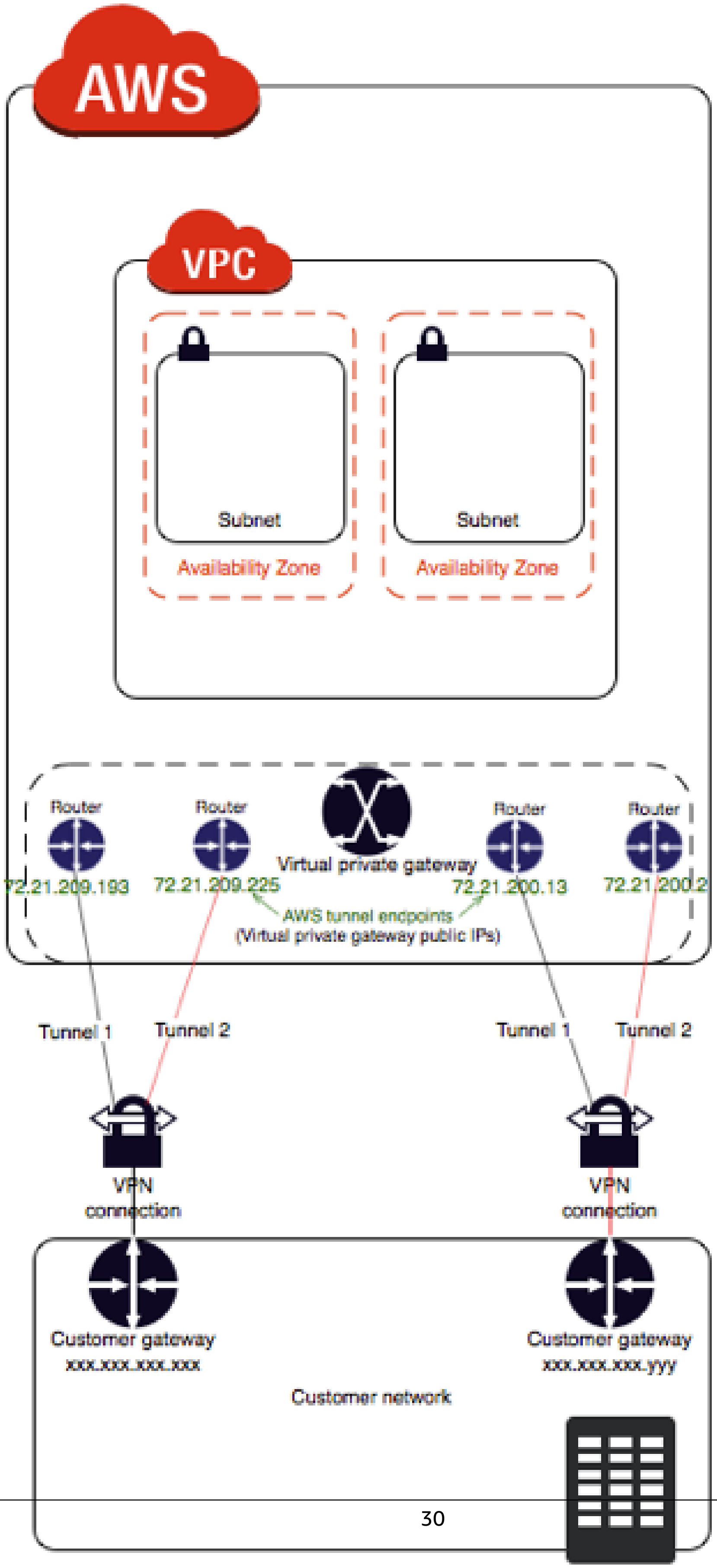
For example, if you have a site in Los Angeles and a second site in New York and both sites have a Site-to-Site VPN connection to the virtual private gateway, you pay the per hour rate for each Site-to-Site VPN connection (so if the rate was $.05 per hour, it would be a total of $.10 per hour). You also 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 using a second customer gateway device. By using redundant Site-to-Site VPN connections and customer gateway devices, you can perform maintenance on one of your devices while traffic continues to flow over the second customer gateway's Site-to-Site VPN connection.

The following diagram shows the two tunnels of each Site-to-Site VPN connection and two customer gateways.


AWS
VPC
Subnet
Availability Zone
Subnet
Availability Zone
Router
Router
Router
Router
Virtual private gateway
72.21.209.193
72.21.209.225
72.21.200.13
72.21.200.2
AWS tunnel endpoints
(Virtual private gateway public IPs)
Tunnel 1
Tunnel 2
Tunnel 1
Tunnel 2
VPN connection
VPN connection
Customer gateway
xxx.xxx.xxx.xxx
Customer gateway
xxx.xxx.xxx.yyy
Customer network

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.

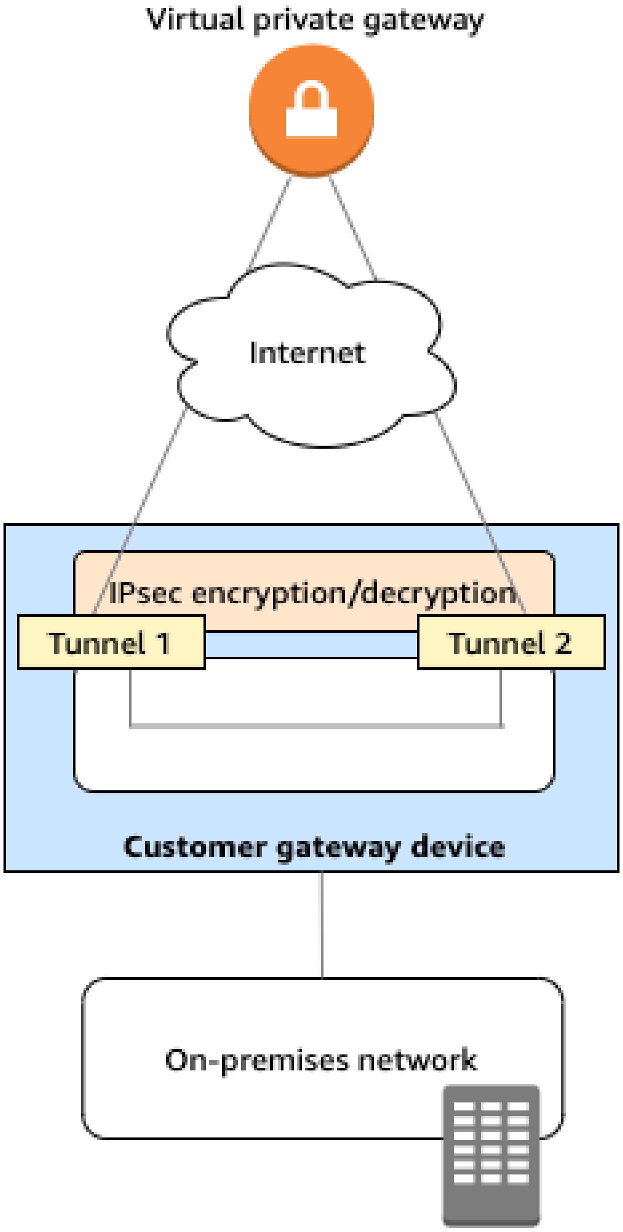
Dynamically routed Site-to-Site VPN connections use the Border Gateway Protocol (BGP) to exchange routing information between your customer gateways and the virtual private gateways. Statically routed Site-to-Site VPN connections require you to enter static routes for the remote network on your side 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 (p. 18).

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, the VPN connection that goes to a virtual private gateway (which is attached to your VPC). The two lines between the customer gateway device 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 (p. 11). When you configure your customer gateway device, it's therefore important that you configure both tunnels.



For the steps to set up a VPN connection, see Getting started (p. 18). 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 (p. 12). The customer gateway resource in AWS does not configure or create the customer gateway device. You must configure the device yourself.

After you create the VPN connection, download the configuration file (p. 23) from the Amazon VPC console, which contains information specific to your VPN connection. Use this information to configure your customer gateway device. In some cases, device-specific configuration files are available for devices that we've tested. Otherwise, you can download the generic configuration file.

Example configuration files

Your customer gateway device can be a physical or software appliance. The following are examples of some devices that have device-specific configuration files in the Amazon VPC console.

- Check Point Security Gateway running R77.10 (or later) software
- Cisco ASA running Cisco ASA 8.2 (or later) software
- Cisco IOS running Cisco IOS 12.4 (or later) software
- SonicWALL running SonicOS 5.9 (or later) software
- Fortinet Fortigate 40+ Series running FortiOS 4.0 (or later) software
- Juniper J-Series running JunOS 9.5 (or later) software
- Juniper SRX running JunOS 11.0 (or later) software
- Juniper SSG running ScreenOS 6.1, or 6.2 (or later) software
- Juniper ISG running ScreenOS 6.1, or 6.2 (or later) software
- Netgate pfSense running OS 2.2.5 (or later) software.
- Palo Alto Networks PANOS 4.1.2 (or later) software
- Yamaha RT107e, RTX1200, RTX1210, RTX1500, RTX3000 and SRT100 routers
- Microsoft Windows Server 2008 R2 (or later) software
- Microsoft Windows Server 2012 R2 (or later) software
- Zyxel Zywall Series 4.20 (or later) software for statically routed VPN connections, or 4.30 (or later) software for dynamically routed VPN connections

If you have one of these devices, but configure it for IPsec in a different way than presented in the configuration file, you can change our suggested configuration to match your needs. You can get sample configuration files from either of the following:

- Static configuration: the section called “Example configuration files” (p. 40)
- Dynamic configuration: the section called “Example configuration files” (p. 51)

You can also find software VPN appliances on the AWS Marketplace.

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 (p. 10).

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 IKE	RFC 2409 RFC 7296	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 Certificate Manager 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. 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 Certificate Manager 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. 5). The following versions are supported: IKEv1 and IKEv2. We support Main mode only with IKEv1. 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).
Establish IPsec security associations in Tunnel mode IPsec	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

Requirement	RFC	Comments
		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	IKE uses Diffie-Hellman to establish ephemeral keys to secure all communication between customer gateway devices and virtual private gateways. The following groups are supported: • Phase 1 groups: 2, 14-24 • Phase 2 groups: 2, 5, 14-24
Fragment IP packets before encryption	RFC 4459	When packets are too large to be transmitted, they must be fragmented. We do not reassemble fragmented encrypted packets. Therefore, your VPN device must fragment packets *before* encapsulating with the VPN headers. The fragments are individually transmitted to the remote host, which reassembles them.
(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.
(Dynamically-routed VPN connections) Bind tunnel to logical interface (route-based VPN) Tunnel	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 BGP	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.

Because the connection encapsulates packets with additional network headers (including IPsec), the amount of data that can be transmitted in a single packet is reduced. We recommend that you use the techniques listed in the following table to help you to minimize problems related to the amount of data that can be transmitted through the IPsec tunnel.

Technique	RFC	Comments
Adjust the maximum segment size (MSS) of TCP packets entering the VPN tunnel	RFC 4459	TCP packets are often the most common type of packet across IPsec tunnels. Some gateways can change the TCP maximum segment size (MSS) parameter. This causes the TCP endpoints (clients, servers) to reduce the amount of data sent with each packet. This is an ideal approach, as the packets arriving at the VPN devices are small enough to be encapsulated and transmitted. We recommend setting the MSS on your customer gateway device to 1359 when using the SHA2-384 or SHA2-512 hashing algorithms. This is necessary to accommodate for the larger header.
Reset the "Don't Fragment" flag on packets	RFC 791	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.

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 (p. 37).

Configuring a firewall between the internet and your customer gateway device

You must have an internet-routable IP address to use as the endpoint for the IPsec tunnels that connect your customer gateway device to the virtual private gateway. If a firewall is in place between the internet and your gateway, the rules in the following tables must be in place to establish the IPsec tunnels. The virtual private gateway addresses are in the configuration file.

Inbound (from the internet)

Input rule I1	
Source IP	Virtual Private Gateway 1
Dest IP	Customer Gateway
Protocol	UDP
Source port	500
Destination	500
Input rule I2	

Source IP	Virtual Private Gateway 2
Dest IP	Customer Gateway
Protocol	UDP
Source port	500
Destination port	500
Input rule I3	
Source IP	Virtual Private Gateway 1
Dest IP	Customer Gateway
Protocol	IP 50 (ESP)
Input rule I4	
Source IP	Virtual Private Gateway 2
Dest IP	Customer Gateway
Protocol	IP 50 (ESP)

Outbound (to the internet)

Output rule O1	
Source IP	Customer Gateway
Dest IP	Virtual Private Gateway 1
Protocol	UDP
Source port	500
Destination port	500
Output rule O2	
Source IP	Customer Gateway
Dest IP	Virtual Private Gateway 2
Protocol	UDP
Source port	500
Destination port	500
Output rule O3	
Source IP	Customer Gateway
Dest IP	Virtual Private Gateway 1
Protocol	IP 50 (ESP)
Output rule O4	
Source IP	Customer Gateway

Dest IP	Virtual Private Gateway 2
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.

If you are using NAT traversal (NAT-T) on your device, you must include rules that allow UDP access over port 4500. 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 (p. 29). 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 (p. 27). 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 (p. 15).

Example customer gateway device configurations for static routing

Topics

Example configuration files

You can download static-routing-examples.zip to view example configuration files for the following customer gateway devices:

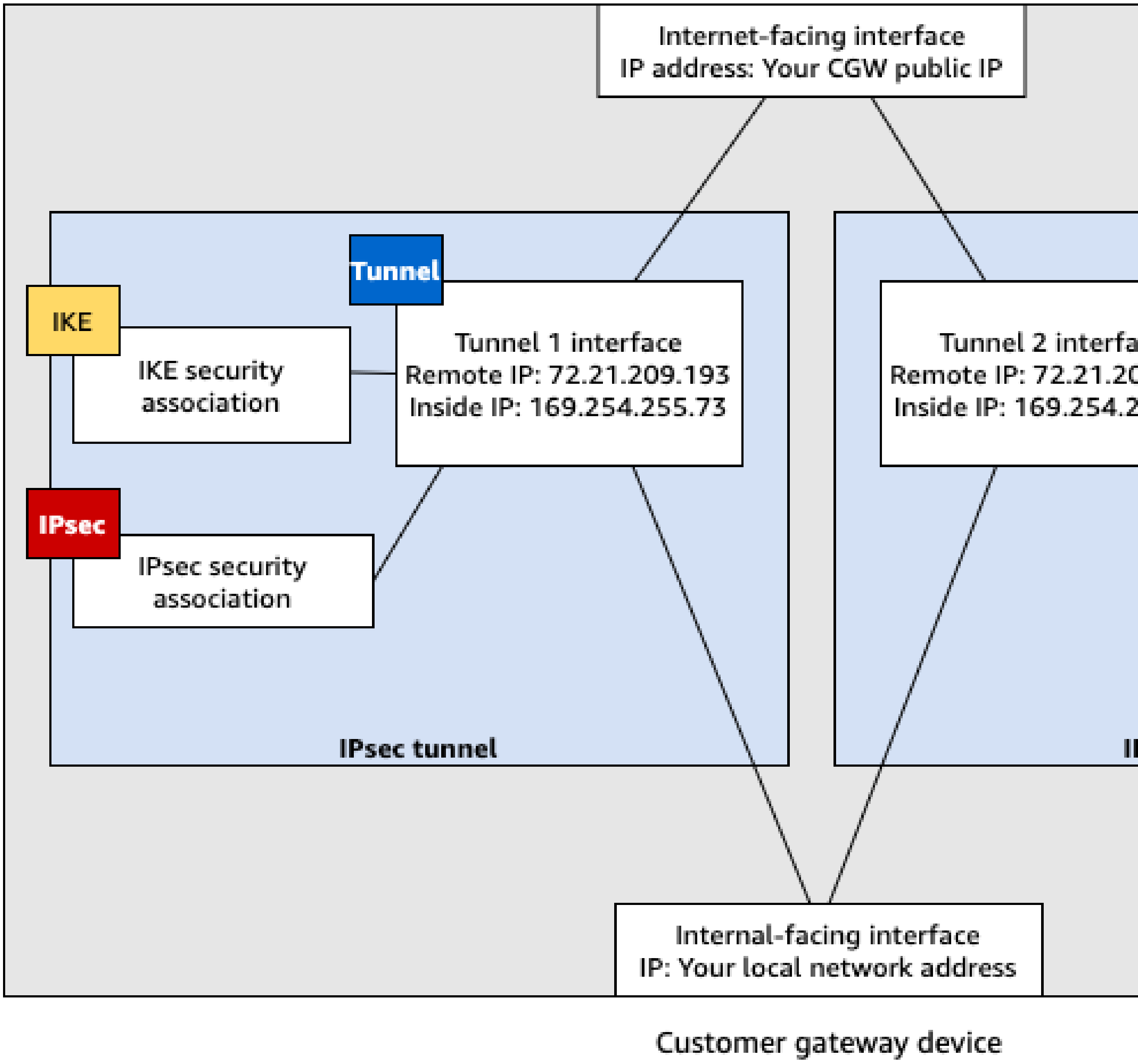
- Cisco ASA running Cisco ASA 8.2+
- Cisco ASA running Cisco ASA 9.7.1+
- Cisco IOS running Cisco IOS
- Cisco Meraki MX Series running 9.0+
- Citrix Netscaler CloudBridge running NS 11+
- Cyberoam CR15iNG running V 10.6.5 MR-1
- F5 Networks BIG-IP running v12.0.0+
- Fortinet Fortigate 40+ Series running FortiOS 4.0+
- Generic configuration for static routing
- H3C MSR800 running version 5.20
- IIJ SEIL/B1 running SEIL/B1 3.70+
- Mikrotik RouterOS running 6.36
- Openswan running 2.6.38+
- pfSense running OS 2.2.5+
- SonicWALLrunning SonicOS 5.9 or 6.2
- Strongswan Ubuntu 16.04 running Strongswan 5.5.1+
- WatchGuard XTM, Firebox running Fireware OS 11.11.4
- Zyxel Zywall running Zywall 4.20+

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

- Example values for the VPN connection 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*)
- Example values for the tunnel inside IP addresses.

In addition to providing placeholder values, the files specify the minimum requirements of IKE version 1, AES128, SHA1, and DH Group 2 in most AWS Regions. They also specify pre-shared keys for authentication (p. 9). You must modify the example configuration files to take advantage of IKE version 2, AES256, SHA256, other DH groups such as 2, 14-18, 22, 23, and 24, and private certificates.

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.



To download a configuration file with values that are specific to your VPN connection configuration, use the Amazon VPC console. For more information, see Download the configuration file (p. 23).

User interface procedures for static routing

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

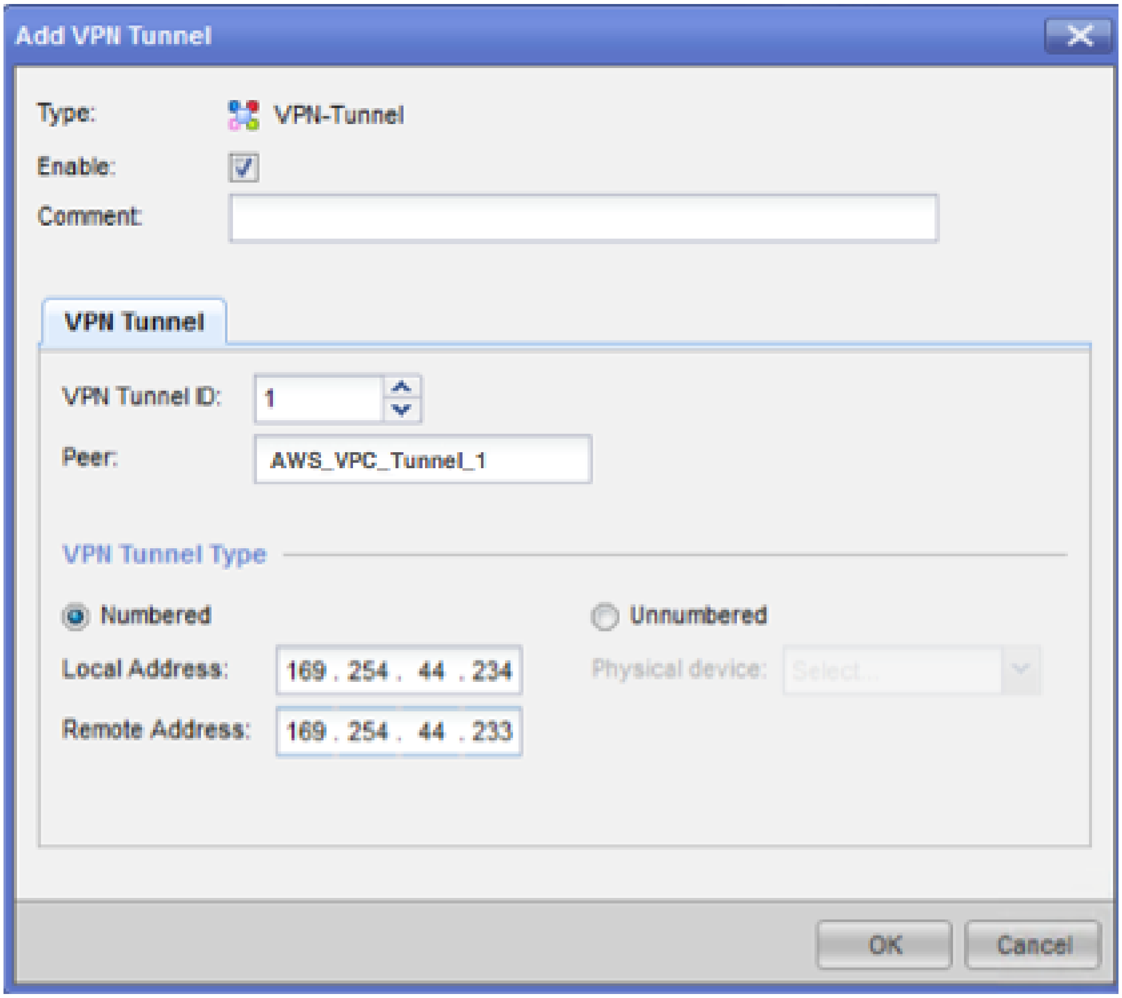
Check Point

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`.



Type: VPN-Tunnel
Enable: ☑
Comment:
VPN Tunnel ID: 1
Peer: AWS_VPC_Tunnel_1
● Numbered ○ Unnumbered
Local Address: 169 . 254 . 44 . 234
Physical device: Select...
Remote Address: 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.

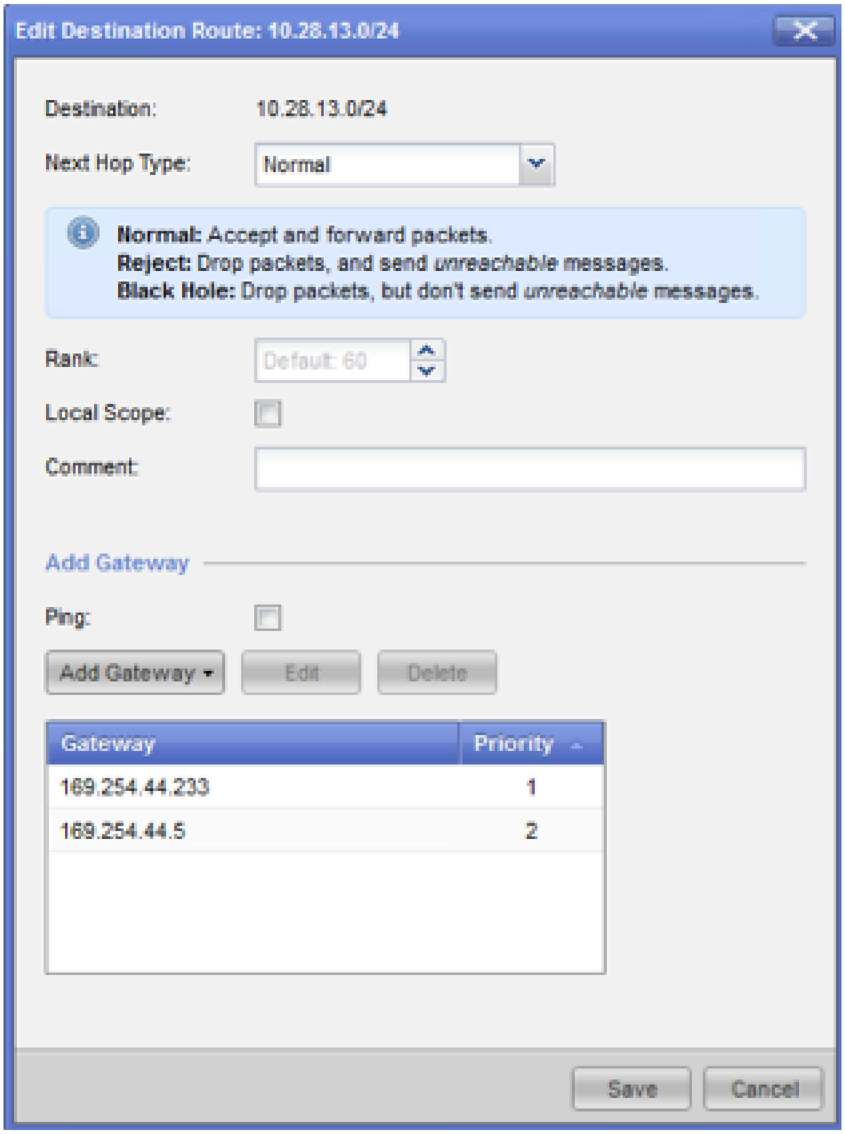
   ```
   set interface vpnt2 mtu 1436
   ```

6. Repeat these steps to create a second tunnel, using the information under the `IPSec Tunnel #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.



Edit Destination Route: 10.28.13.0/24

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

7. Choose **Save**.

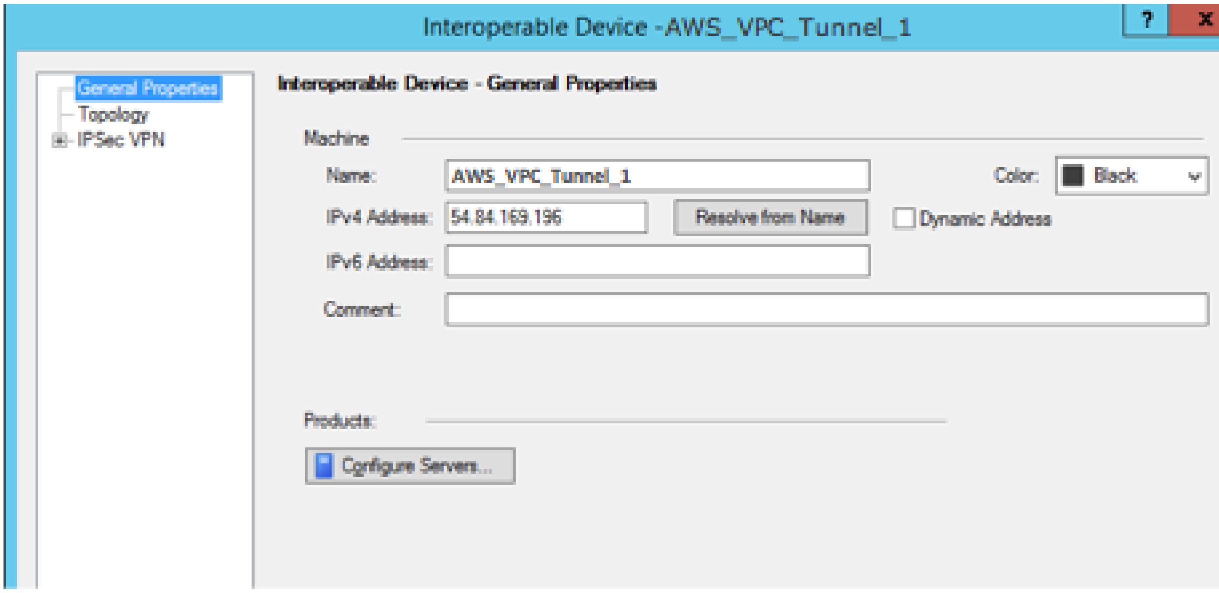
If you're using a cluster, repeat the preceding steps for the other members of the cluster.

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.



Interoperable Device -AWS_VPC_Tunnel_1

Interoperable Device - General Properties

Machine

Name: AWS_VPC_Tunnel_1

Color: Black

IPv4 Address: 54.84.169.196

□ Dynamic Address

IPv6 Address:

Comment:

Products:

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.

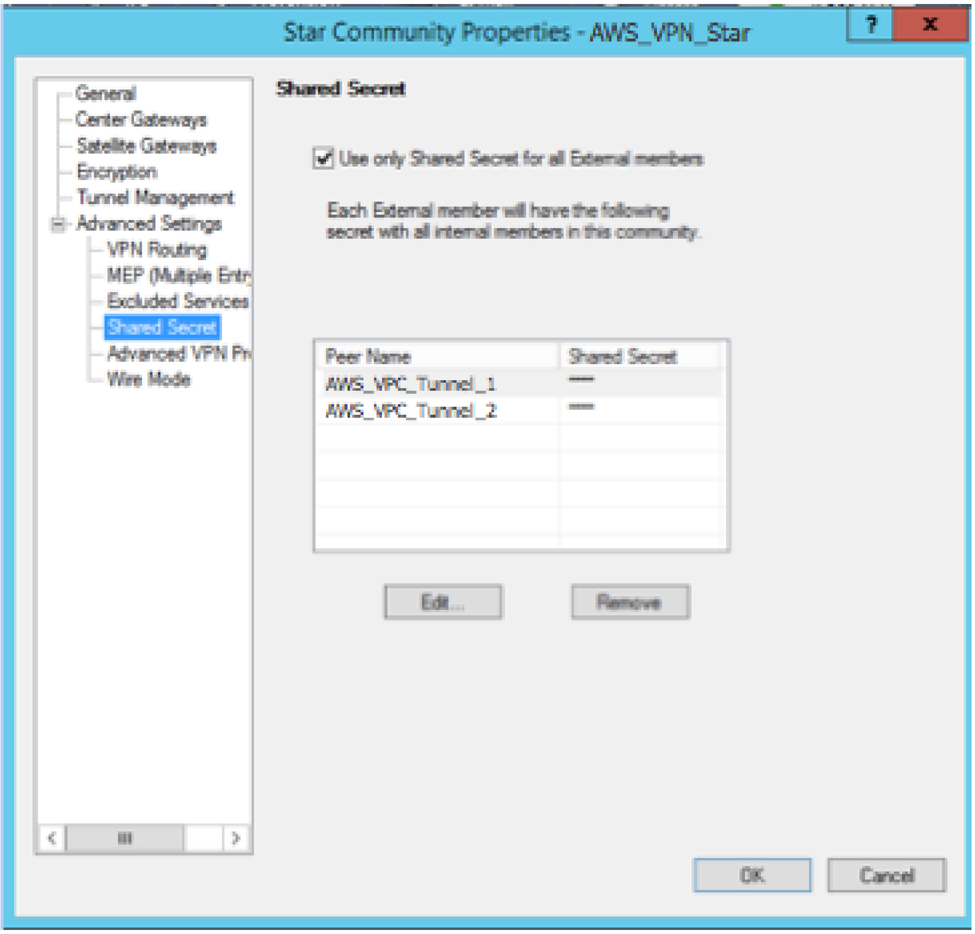
To create and configure the VPN community, IKE, and IPsec settings

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.



Shared Secret

☑ Use only Shared Secret for all External members

Each External member will have the following secret with all internal members in this community.

Peer Name	Shared Secret
AWS_VPC_Tunnel_1	—
AWS_VPC_Tunnel_2	—

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 GuiBDedit 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**.

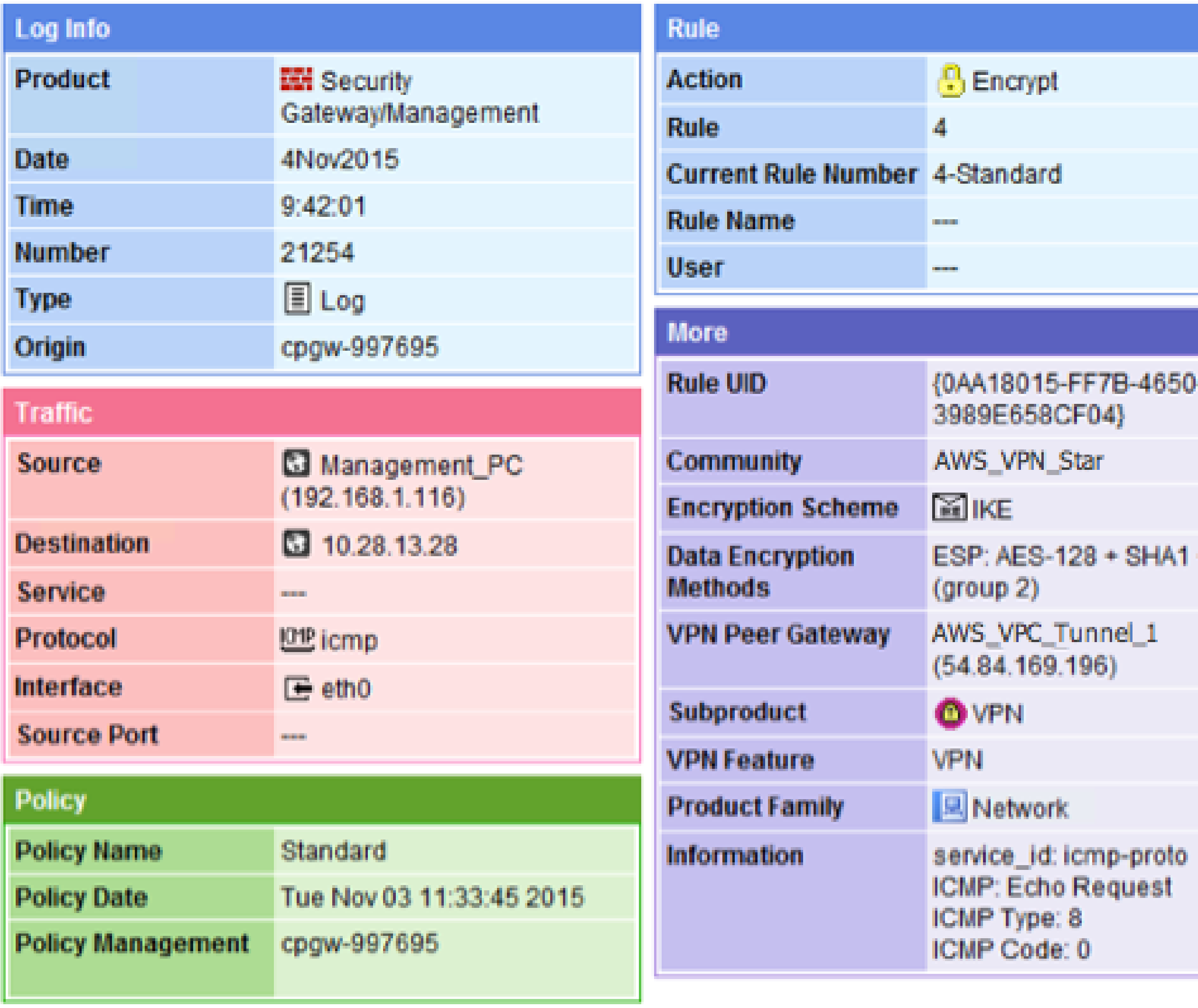
To verify the tunnel status

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	
Product	Security Gateway/Management
Date	4Nov2015
Time	9:42:01
Number	21254
Type	Log
Origin	cpgw-997695

Traffic	
Source	Management_PC (192.168.1.116)
Destination	10.28.13.28
Service	----
Protocol	icmp
Interface	eth0
Source Port	----

Policy	
Policy Name	Standard
Policy Date	Tue Nov 03 11:33:45 2015
Policy Management	cpgw-997695

Rule	
Action	Encrypt
Rule	4
Current Rule Number	4-Standard
Rule Name	----
User	----

More	
Rule UID	{0AA18015-FF7B-4650 3989E658CF04}
Community	AWS_VPN_Star
Encryption Scheme	IKE
Data Encryption Methods	ESP: AES-128 + SHA1 (group 2)
VPN Peer Gateway	AWS_VPC_Tunnel_1 (54.84.169.196)
Subproduct	VPN
VPN Feature	VPN
Product Family	Network
Information	service_id: icmp-proto ICMP: Echo Request ICMP Type: 8 ICMP Code: 0

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 (AWS endpoint) 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 (AWS endpoint).

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 Testing the Site-to-Site VPN connection (p. 104).

Example customer gateway device configurations for dynamic routing (BGP)

Topics

Example configuration files

You can download dynamic-routing-examples.zip to view example configuration files for the following customer gateway devices:

- Barracuda NextGen Firewall F-series running 6.2+
- Cisco ASA running Cisco ASA 9.7.1+
- Cisco IOS running Cisco IOS 12.4+
- F5 Networks BIG-IP running v12.0.0+
- Fortinet Fortigate 40+
- Generic configuration for dynamic routing
- H3C MSR800 running version 5.20
- IIJ SEIL/B1 running SEIL/B1 3.70+
- Juniper J-Series running JunOS 9.5+

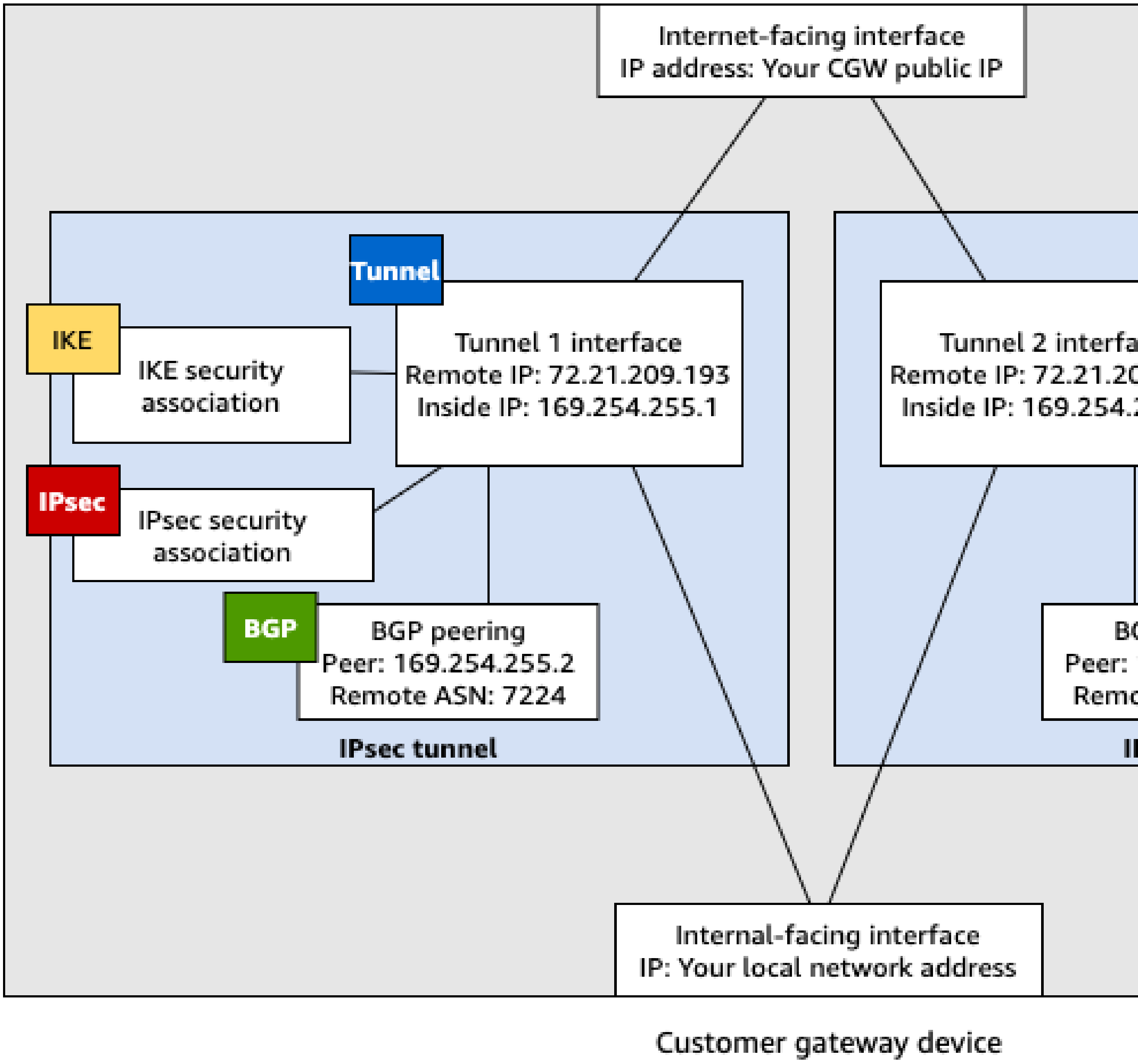
- Juniper SRX running JunOS 11.0+
- Juniper SSG or Netscreen series running Juniper ScreenOS 6.1+
- Mikrotik RouterOS running 6.36
- Palo Alto Networks running PANOS 4.1.2+
- SonicWALL running SonicOS 5.9 or 6.2
- Sophos ASG running V8.300+
- Vyatta running Network OS 6.5+
- WatchGuard XTM, Firebox running Fireware OS 11.12.2+
- Yamaha RT107e, RTX1200, RTX1210, RTX1500, RTX3000, or SRT100
- Zyxel ZyWALL running ZLD 4.3+

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

- Example values for the VPN connection ID and virtual private gateway ID
- Placeholders for the remote (outside) IP address AWS endpoints (*`AWS_ENDPOINT_1`* and *`AWS_ENDPOINT_2`*)
- Placeholders for the IP address for the internet-routable external interface on the customer gateway device (*`your-cgw-ip-address`*), and the BGP ASN.
- Example values for the tunnel inside IP addresses.

In addition to providing placeholder values, the files specify the minimum requirements of IKE version 1, AES128, SHA1, and DH Group 2 in most AWS Regions. They also specify pre-shared keys for authentication (p. 9). You must modify the example configuration files to take advantage of IKE version 2, AES256, SHA256, other DH groups such as 2, 14-18, 22, 23, and 24, and private certificates.

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.



To download a configuration file with values that are specific to your VPN connection configuration, you must use the Amazon VPC console. For more information, see Download the configuration file (p. 23).

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

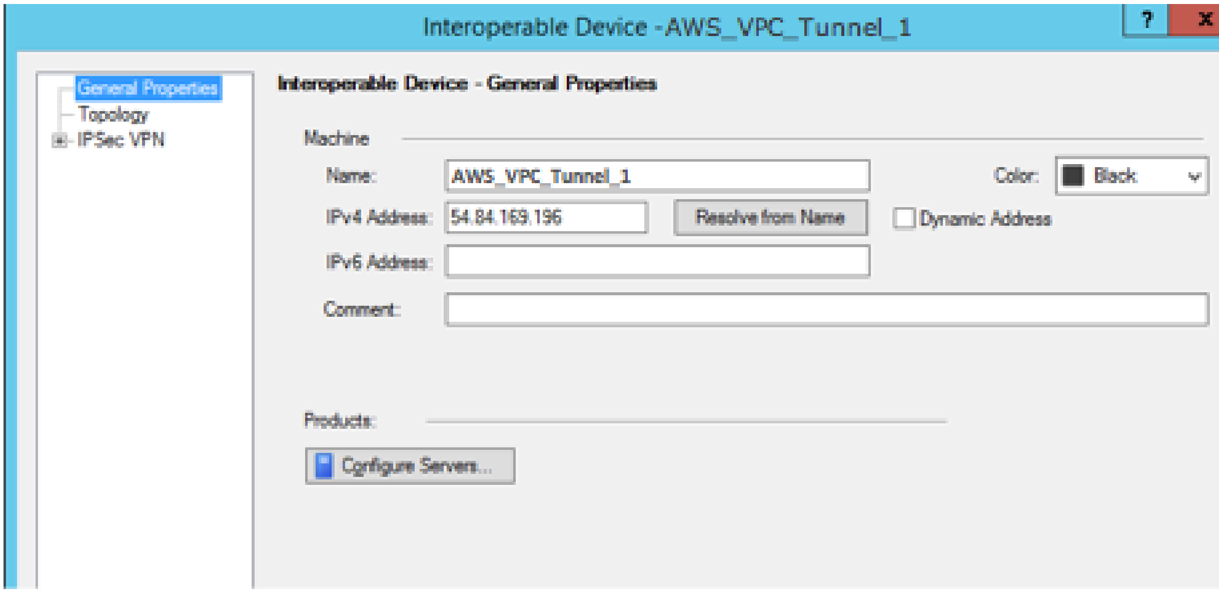
The following steps are for distributing local interface routes. You can also redistribute routes from different sources (for example, static routes, or routes obtained through dynamic routing protocols). For more information, see the Gaia Advanced Routing R77 Versions Administration Guide.

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.



Interoperable Device - AWS_VPC_Tunnel_1

Interoperable Device - General Properties

Name: AWS_VPC_Tunnel_1

Color: Black

IPv4 Address: 54.84.169.196

□ Dynamic Address

IPv6 Address:

Comment:

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.

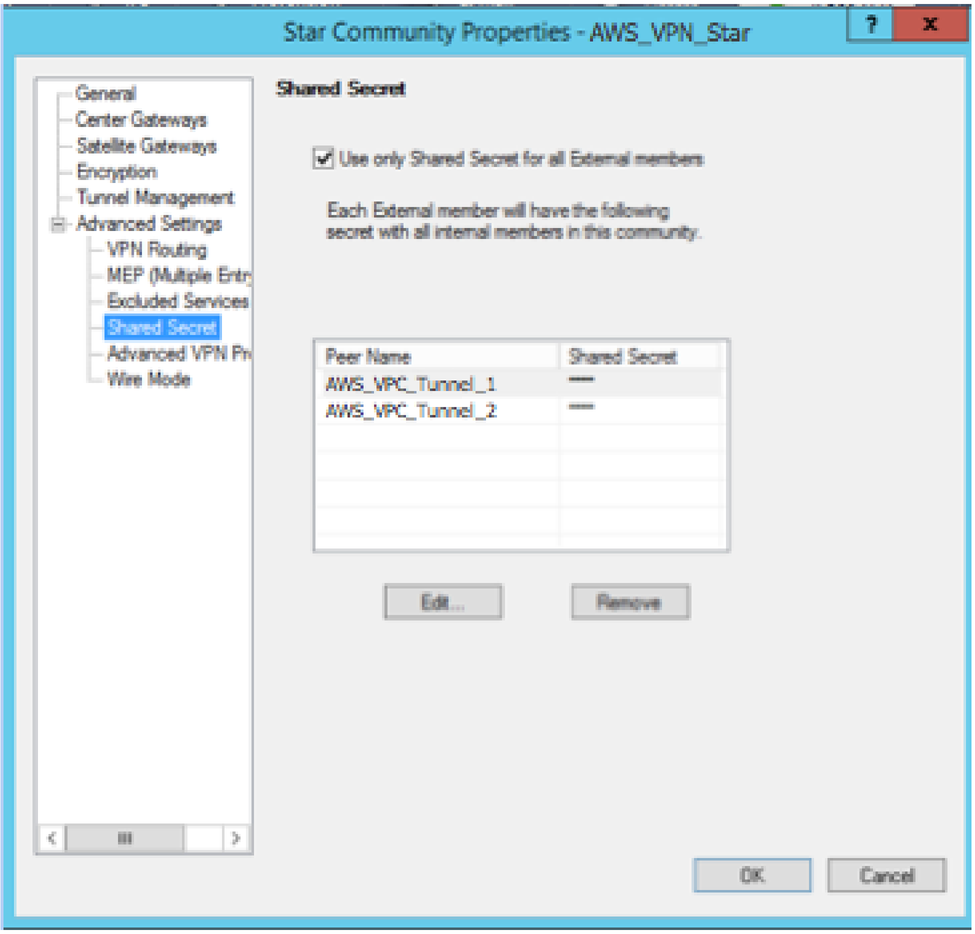
To create and configure the VPN community, IKE, and IPsec settings

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.



Shared Secret

☑ Use only Shared Secret for all External members

Each External member will have the following secret with all internal members in this community.

Peer Name	Shared Secret
AWS_VPC_Tunnel_1	****
AWS_VPC_Tunnel_2	****

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 GuiBDedit 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**.

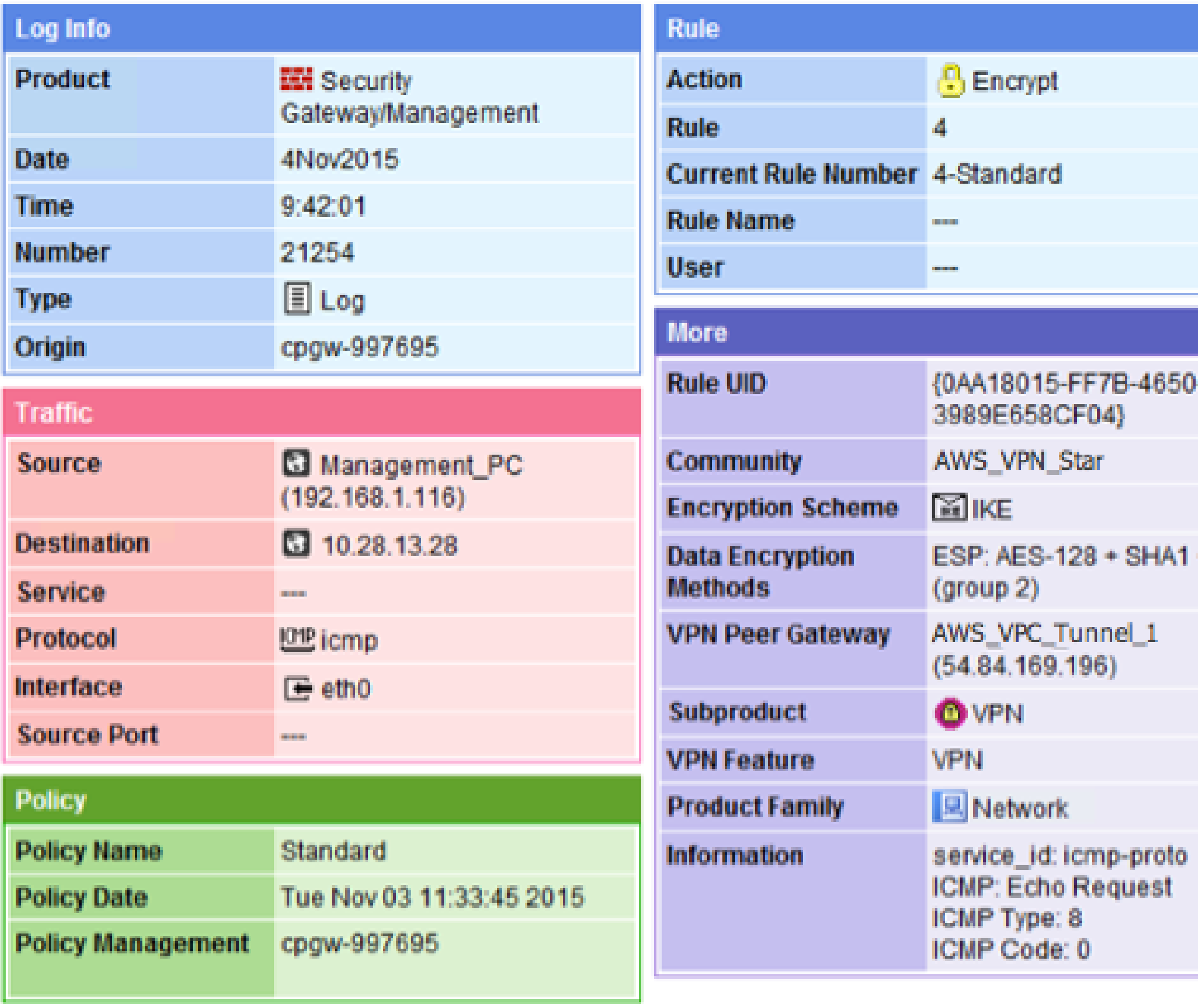
To verify the tunnel status

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	
Product	Security Gateway/Management
Date	4Nov2015
Time	9:42:01
Number	21254
Type	Log
Origin	cpgw-997695

Traffic	
Source	Management_PC (192.168.1.116)
Destination	10.28.13.28
Service	---
Protocol	icmp
Interface	eth0
Source Port	---

Policy	
Policy Name	Standard
Policy Date	Tue Nov 03 11:33:45 2015
Policy Management	cpgw-997695

Rule	
Action	Encrypt
Rule	4
Current Rule Number	4-Standard
Rule Name	---
User	---

More	
Rule UID	{0AA18015-FF7B-4650 3989E658CF04}
Community	AWS_VPN_Star
Encryption Scheme	IKE
Data Encryption Methods	ESP: AES-128 + SHA1 (group 2)
VPN Peer Gateway	AWS_VPC_Tunnel_1 (54.84.169.196)
Subproduct	VPN
VPN Feature	VPN
Product Family	Network
Information	service_id: icmp-proto ICMP: Echo Request ICMP Type: 8 ICMP Code: 0

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 (p. 41).

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 Testing the Site-to-Site VPN connection (p. 104).

Configuring Windows Server as a customer gateway device

You can configure your 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.

Topics

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 (p. 20).
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 Create a Site-to-Site VPN connection (p. 22).

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 (p. 21).
- 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:               xCjNLsLoCmKsakwcdoR9yX6GsEXAMPLE
```

`Local Tunnel Endpoint`

The IP address for the customer gateway—in this case, your Windows Server—that terminates the VPN connection on your network's side. If your customer gateway device is an EC2 instance, this is the instance's private IP address.

`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 (p. 66)
- Option 2: Use the Windows Server user interface (p. 66)

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=xCjNLsLoCmKsakwcdoR9yX6GsEXAMPLE ^
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 (p. 70).

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. This section guides you through the steps.

> **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 (p. 69).

Topics

- Configure a security rule for a VPN tunnel (p. 66)
- Confirm the tunnel configuration (p. 69)
- Enable master key perfect forward secrecy (p. 69)
- Configure the Windows firewall (p. 70)

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.

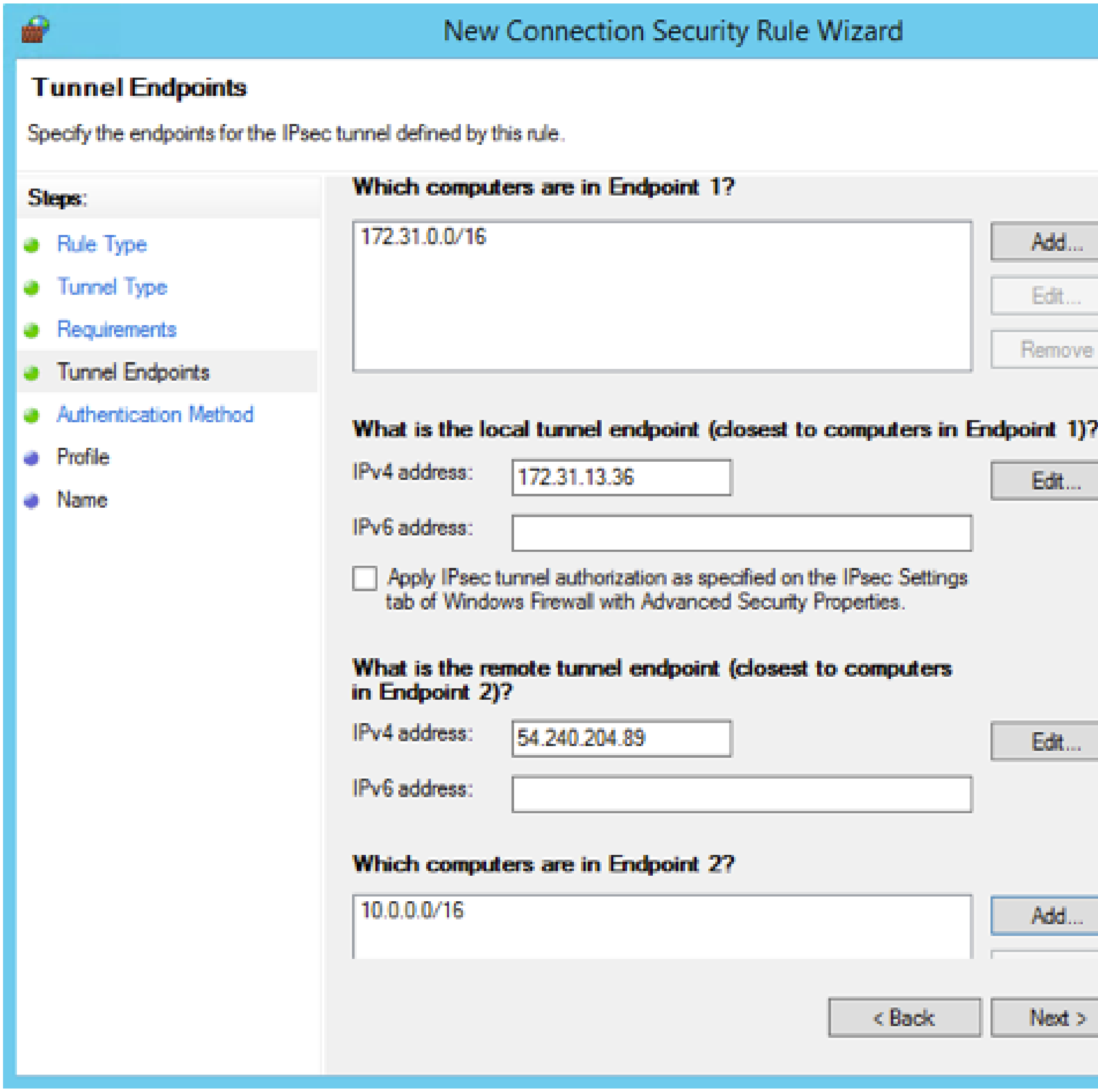
To configure a security rule for a VPN tunnel

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.



Tunnel Endpoints

Specify the endpoints for the IPsec tunnel defined by this rule.

Which computers are in Endpoint 1?

172.31.0.0/16

What is the local tunnel endpoint (closest to computers in Endpoint 1)?

IPv4 address: 172.31.13.36

IPv6 address:

□ 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)?

IPv4 address: 54.240.204.89

IPv6 address:

Which computers are in Endpoint 2?

10.0.0.0/16

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 AWS 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 Testing the Site-to-Site VPN connection (p. 104).

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 (p. 62).
- 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 (p. 62).
- 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 (p. 62).

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 steps can help you troubleshoot connectivity issues on customer gateway devices.

For general testing instructions, see Testing the Site-to-Site VPN connection (p. 104).

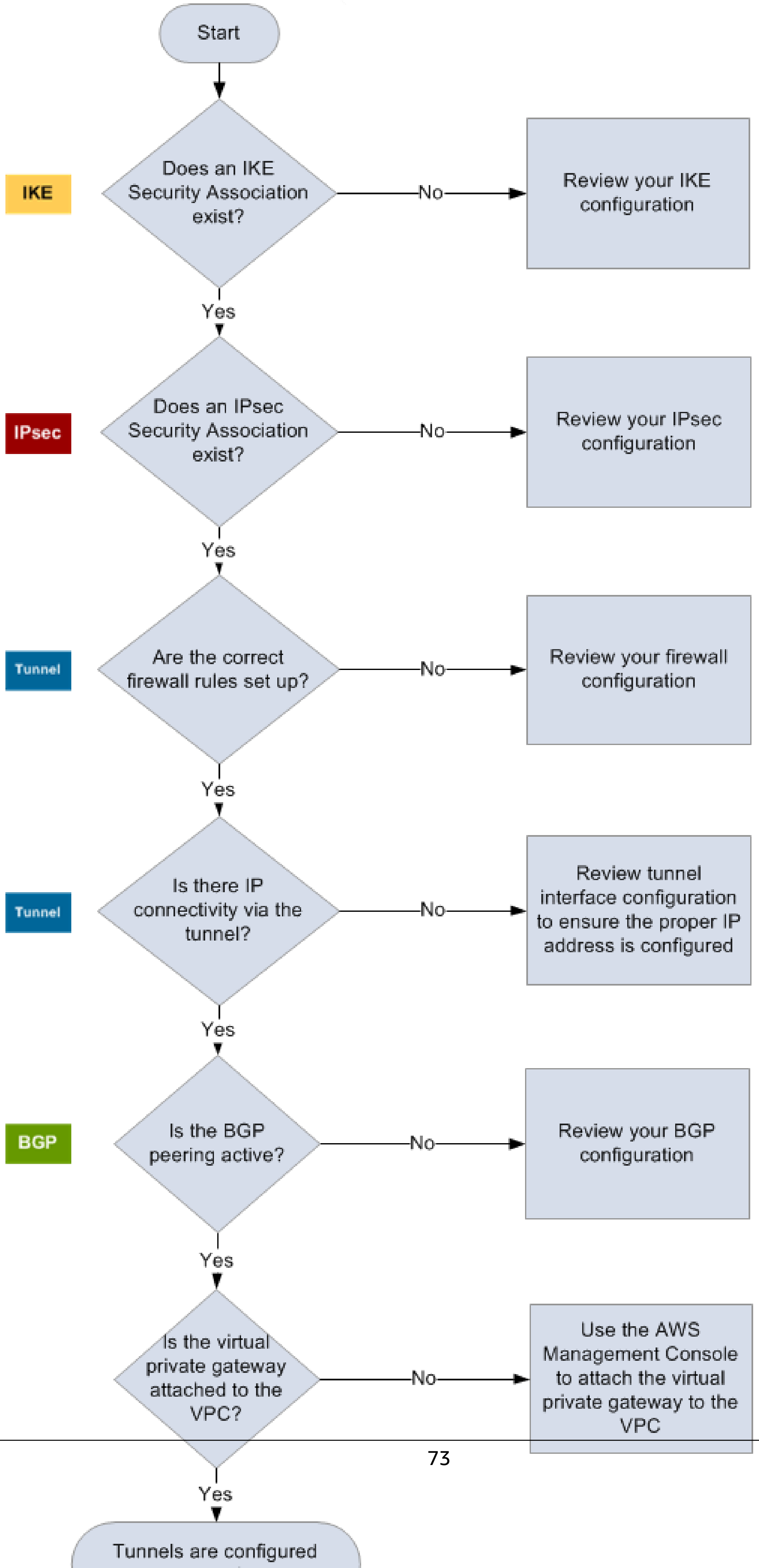
Topics

Additional resources

- Amazon VPC forum
- How do I troubleshoot VPN tunnel connectivity to my Amazon VPC?

Troubleshooting connectivity when using Border Gateway Protocol

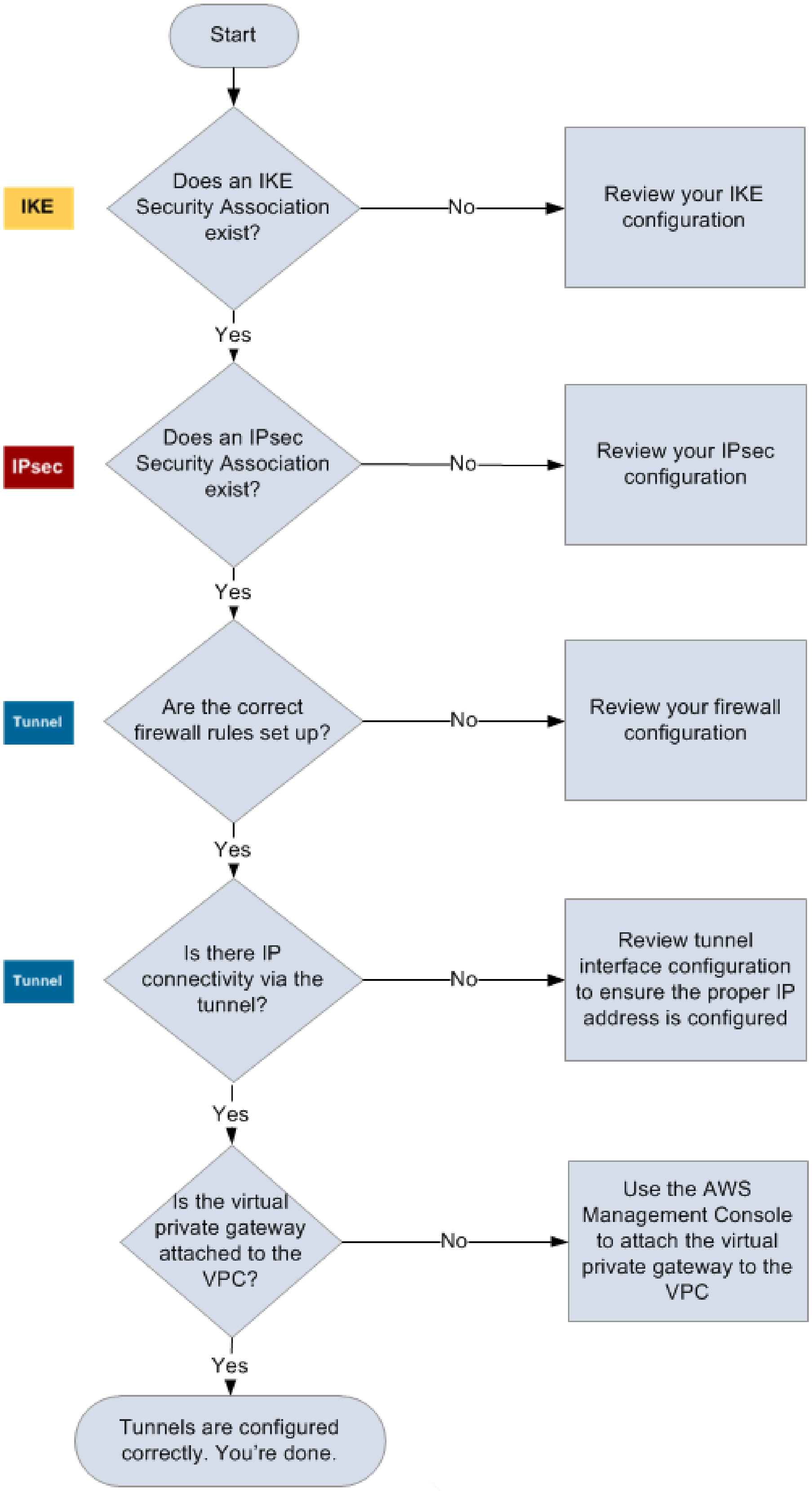
The following diagram and table provide general instructions for troubleshooting a customer gateway device that uses Border Gateway Protocol (BGP). We also recommend that you enable the debug features of your device. Consult your gateway device vendor for details.


Start
IKE
Does an IKE Security Association exist?
No
Review your IKE configuration
Yes
IPsec
Does an IPsec Security Association exist?
No
Review your IPsec configuration
Yes
Tunnel
Are the correct firewall rules set up?
No
Review your firewall configuration
Yes
Tunnel
Is there IP connectivity via the tunnel?
No
Review tunnel interface configuration to ensure the proper IP address is configured
Yes
BGP
Is the BGP peering active?
No
Review your BGP configuration
Yes
Is the virtual private gateway attached to the VPC?
No
Use the AWS Management Console to attach the virtual private gateway to the VPC
Yes
Tunnels are configured

IKE	Determine if an IKE security association exists. An IKE security association is required to exchange keys that are used to establish the IPsec security association. 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. If an IKE security association exists, move on to 'IPsec'.
IPsec	Determine if an IPsec security association (SA) exists. 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. If no IPsec SA exists, review your IPsec configuration. If an IPsec SA exists, move on to 'Tunnel'.
Tunnel	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 (p. 37)). If they are, move forward. Determine if there is IP connectivity through the tunnel. 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. If the ping isn't successful, review your tunnel interface configuration to make sure that the proper IP address is configured. If the ping is successful, move on to 'BGP'.
BGP	Determine if the BGP peering session is active. For each tunnel, do the following: • 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.


Start
IKE
Does an IKE Security Association exist?
No
Review your IKE configuration
Yes
IPsec
Does an IPsec Security Association exist?
No
Review your IPsec configuration
Yes
Tunnel
Are the correct firewall rules set up?
No
Review your firewall configuration
Yes
Tunnel
Is there IP connectivity via the tunnel?
No
Review tunnel interface configuration to ensure the proper IP address is configured
Yes
Is the virtual private gateway attached to the VPC?
No
Use the AWS Management Console to attach the virtual private gateway to the VPC
Yes
Tunnels are configured correctly. You're done.

IKE	Determine if an IKE security association exists. An IKE security association is required to exchange keys that are used to establish the IPsec security association. 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. If an IKE security association exists, move on to 'IPsec'.
IPsec	Determine if an IPsec security association (SA) exists. 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. If no IPsec SA exists, review your IPsec configuration. If an IPsec SA exists, move on to 'Tunnel'.
Tunnel	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 (p. 37)). If they are, move forward. Determine if there is IP connectivity through the tunnel. 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. If the ping isn't successful, review your tunnel interface configuration to make sure that the proper IP address is configured. If the ping is successful, move on to 'Static routes'.
Static routes	For each tunnel, do the following: • 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-ppe1

      #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 (p. 37).

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 (p. 37).

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 (p. 37).

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                                      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                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 (p. 37).

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.

```
ssg5-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.

```
ssg5-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: EBGP 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.

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 (p. 37).

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
```

Working with Site-to-Site VPN

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

Contents

Identifying a Site-to-Site VPN connection

You can find out the category of your Site-to-Site VPN connection by using the Amazon VPC console or a command line tool.

To identify the Site-to-Site VPN category 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 check the value for **Category** in the details pane. A value of `VPN` indicates an AWS VPN connection. A value of `VPN-Classic` indicates an AWS Classic VPN connection.

To identify the Site-to-Site VPN category using a command line tool

- You can use the describe-vpn-connections AWS CLI command. In the output that's returned, take note of the `Category` value. A value of `VPN` indicates an AWS VPN connection. A value of `VPN-Classic` indicates an AWS Classic VPN connection.

  In the following example, the Site-to-Site VPN connection is an AWS VPN connection.

```
aws ec2 describe-vpn-connections --vpn-connection-ids vpn-1a2b3c4d
```

```
{
    "VpnConnections": [
        {
            "VpnConnectionId": "vpn-1a2b3c4d",

            ...
            "State": "available",
            "VpnGatewayId": "vgw-11aa22bb",
            "CustomerGatewayId": "cgw-ab12cd34",
            "Type": "ipsec.1",
            "Category": "VPN"
        }
    ]
}
```

Alternatively, use one of the following commands:

- DescribeVpnConnections (Amazon EC2 Query API)
- Get-EC2VpnConnection (Tools for Windows PowerShell)

Migrating from AWS Classic VPN to AWS VPN

If your existing Site-to-Site VPN connection is an AWS Classic VPN connection, you can migrate to an AWS VPN connection. You can migrate directly to a new virtual private gateway (option 1), or you can migrate using a transit gateway (option 2). During the procedure for option 1, your Site-to-Site VPN connection is temporarily interrupted when you detach the old virtual private gateway from your VPC. During the procedure for option 2, your Site-to-Site VPN connection is not interrupted, however you will incur additional transit gateway costs.

If you use an AWS Classic VPN connection as a backup for your AWS Direct Connect connection, you can delete and recreate the Site-to-Site VPN connection (option 3). During the procedure for option 3, there is zero downtime on the AWS Direct Connect private virtual interface.

If your existing virtual private gateway is associated with multiple Site-to-Site VPN connections, you must recreate each Site-to-Site VPN connection for the new virtual private gateway. If there are multiple AWS Direct Connect private virtual interfaces attached to your virtual private gateway, you must recreate each private virtual interface for the new virtual private gateway. For more information, see Creating a virtual interface in the *AWS Direct Connect User Guide*.

If your existing Site-to-Site VPN connection is an AWS VPN connection, you cannot migrate to an AWS Classic VPN connection.

Topics

Option 1: Migrate directly to a new virtual private gateway

In this option, you create a new virtual private gateway and Site-to-Site VPN connection, detach the old virtual private gateway from your VPC, and attach the new virtual private gateway to your VPC.

Note
During this procedure, connectivity over the current Site-to-Site VPN connection is interrupted when you disable route propagation and detach the old virtual private gateway from your VPC. Connectivity is restored when the new virtual private gateway is attached to your VPC and the new Site-to-Site VPN connection is active. Ensure that you plan for the expected downtime.

To migrate to an AWS VPN connection

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Virtual Private Gateways**, **Create Virtual Private Gateway** and create a virtual private gateway.
3. In the navigation pane, choose **Site-to-Site VPN Connections**, **Create VPN Connection**. Specify the following information, and choose **Yes, Create**.

   - **Virtual Private Gateway**: Select the virtual private gateway that you created in the previous step.
   - **Customer Gateway**: Choose **Existing**, and select the existing customer gateway for your current AWS Classic VPN connection.
   - Specify the routing options as required.
4. Select the new Site-to-Site VPN connection and choose **Download Configuration**. Download the appropriate configuration file for your customer gateway device.
5. Use the configuration file to configure VPN tunnels on your customer gateway device. For more information, see Your customer gateway device (p. 32). Do not enable the tunnels yet. Contact your vendor if you need guidance on keeping the newly configured tunnels disabled.
6. (Optional) Create test VPC and attach the virtual private gateway to the test VPC. Change the encryption domain/source destination addresses as required, and test connectivity from a host in your local network to a test instance in the test VPC.
7. If you are using route propagation for your route table, choose **Route Tables** in the navigation pane. Select the route table for your VPC, and choose **Route Propagation**, **Edit**. Clear the check box for the old virtual private gateway and choose **Save**.

   > **Note**
   > From this step onwards, connectivity is interrupted until the new virtual private gateway is attached and the new Site-to-Site VPN connection is active.
8. In the navigation pane, choose **Virtual Private Gateways**. Select the old virtual private gateway and choose **Actions**, **Detach from VPC**, **Yes, Detach**. Select the new virtual private gateway, and choose **Actions**, **Attach to VPC**. Specify the VPC for your Site-to-Site VPN connection, and choose **Yes, Attach**.
9. In the navigation pane, choose **Route Tables**. Select the route table for your VPC and do one of the following:

   - If you are using route propagation, choose **Route Propagation**, **Edit**. Select the new virtual private gateway that's attached to the VPC and choose **Save**.
   - If you are using static routes, choose **Routes**, **Edit**. Modify the route to point to the new virtual private gateway, and choose **Save**.
10. Enable the new tunnels on your customer gateway device and disable the old tunnels. To bring the tunnel up, you must initiate the connection from your local network.

    If applicable, check your route table to ensure that the routes are being propagated. The routes propagate to the route table when the status of the VPN tunnel is `UP`.

    > **Note**
    > If you need to revert to your previous configuration, detach the new virtual private gateway and follow steps 8 and 9 to re-attach the old virtual private gateway and update your routes.
11. If you no longer need your AWS Classic VPN connection and do not want to continue incurring charges for it, remove the previous tunnel configurations from your customer gateway device, and delete the Site-to-Site VPN connection. To do this, go to **Site-to-Site VPN Connections**, select the Site-to-Site VPN connection, and choose **Delete**.

    > **Important**
    > After you've deleted the AWS Classic VPN connection, you cannot revert or migrate your new AWS VPN connection back to an AWS Classic VPN connection.

Option 2: Migrate using a transit gateway

In this option, you create a transit gateway, attach it to the VPC in which your Site-to-Site VPN connection resides, and create a temporary Site-to-Site VPN connection on the transit gateway using your existing customer gateway. You then route the traffic through the transit gateway VPN connection while you configure a new Site-to-Site VPN connection on a new virtual private gateway.

Alternatively, you can use this option to migrate your Site-to-Site VPN connection directly to a transit gateway. In this case, you create your new VPN connection on the transit gateway instead of creating it on a new virtual private gateway.

Step 1: Create a transit gateway and VPN connection

To create a transit gateway and VPN connection

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Transit Gateways**, **Create Transit Gateway** and create a transit gateway using the default options.
3. In the navigation pane, choose **Transit Gateway Attachments**, **Create Transit Gateway Attachment**. Specify the following information, and choose **Create attachment**.
   - For **Transit Gateway ID**, choose the transit gateway you created.
   - For **VPC ID**, choose the VPC to attach to the transit gateway.
4. Choose **Create Transit Gateway Attachment** again, specify the following information, and choose **Create attachment**.
   - For **Transit Gateway ID**, choose the transit gateway you created.
   - For **Attachment type**, choose **VPN**.
   - For **Customer Gateway ID**, choose the customer gateway for your existing Site-to-Site VPN connection, and choose the required routing option.

Step 2: Create a new virtual private gateway

Create a new virtual private gateway and new Site-to-Site VPN connection. This step is only required if you want to migrate to a new virtual private gateway. If you want to migrate your VPN connection to a transit gateway, you can skip these steps and go directly to Step 3 (p. 101).

To create a new Site-to-Site VPN connection

1. In the navigation pane, choose **Virtual Private Gateways**, **Create Virtual Private Gateway** and create a new virtual private gateway.
2. In the navigation pane, choose **Site-to-Site VPN Connections**, **Create VPN Connection**.
3. For **Virtual Private Gateway**, choose the virtual private gateway you created.
4. For **Customer Gateway ID**, choose the existing customer gateway for your existing Site-to-Site VPN connection, and specify the type of routing. Choose **Create VPN Connection**.
5. Select your new Site-to-Site VPN connection and choose **Download Configuration** to download the example configuration file. Configure the VPN connection on your customer gateway device, but do not route any traffic yet (do not create any static routes or filter out BGP announcements).

Step 3: Switch to the new VPN connection

During this procedure, you'll temporarily enable asymmetric routing for your VPN traffic when you switch traffic to the transit gateway and then to the new Site-to-Site VPN connection.

To switch to the new Site-to-Site VPN connection

1. Configure your customer gateway device to use the VPN connection on the transit gateway (specify a static route or allow BGP announcements, as needed). This starts asymmetric traffic routing.
2. In the navigation pane, choose **Route Tables**, select the route table for your VPC, and choose **Actions**, **Edit routes**.
3. Add routes that point to your on-premises network and choose the transit gateway as the target. For the destination routes, enter more specific routes, for example, if your on-premises network is `10.0.0.0/16`, create a route that points to `10.0.0.0/17` and another route that points to `10.0.128.0/17`. Asymmetric traffic routing stops and all traffic is routed through the transit gateway.

   > **Note**
   > If you're migrating your VPN connection to a transit gateway instead of a new virtual private gateway, you can stop here.
4. In the navigation pane, choose **Virtual Private Gateways**.
5. Select the old virtual private gateway that's attached to your VPC, and choose **Actions**, **Detach from VPC**. Choose **Yes, Detach**.
6. Select the new virtual private gateway that you created earlier, and choose **Actions**, **Attach to VPC**. Choose your VPC, and choose **Yes, Attach**.
7. In the navigation pane, choose **Route Tables**. Select the route table for your VPC, and choose **Route Propagation**, **Edit route propagation**. Choose the check box for the new virtual private gateway and choose **Save**. Verify that the route is propagated to your VPC route table.
8. Configure your customer gateway device to use the new virtual private gateway and route traffic from your on-premise network to your VPC, using static routes or BGP. This starts asymmetric routing.
9. In the navigation pane, choose **Route Tables**. Select the route table for your VPC, and choose **Actions**, **Edit routes**. Delete the more specific routes to your transit gateway. This stops the asymmetric traffic flow and all traffic is routed through your new Site-to-Site VPN connection.

Step 4: Clean up

If you no longer need your AWS Classic VPN connection, you can delete it. If you migrated to a new virtual private gateway, you can also delete the transit gateway VPN connection and the transit gateway that you created for the migration.

To clean up your resources

1. On your customer gateway device, remove the configuration for the temporary VPN connection on the transit gateway, and the configuration for the old VPN connection.
2. In the navigation pane, choose **Site-to-Site VPN Connections**, select your old Site-to-Site VPN connection, and choose **Actions**, **Delete**.
3. In the navigation pane, choose **Virtual Private Gateways**, select your old virtual private gateway, and choose **Actions**, **Delete Virtual Private Gateway**. If you migrated your VPN connection to a transit gateway, you can stop here.
4. In the navigation pane, choose **Site-to-Site VPN Connections** and select the transit gateway VPN connection. Choose **Actions**, **Delete**.
5. In the navigation pane, choose **Transit Gateway Attachments**, and select the VPC attachment. Choose **Actions**, **Delete**.
6. In the navigation pane, choose **Transit Gateways** and select your transit gateway. Choose **Actions**, **Delete**.

Option 3: (Backup VPN connections for AWS Direct Connect) Delete and recreate the VPN connection

Use this option if you have an AWS Direct Connect connection and an AWS Classic VPN connection on the same virtual private gateway, and you use the VPN connection as a backup for the AWS Direct Connect connection. In this option, you delete the existing AWS Classic VPN connections on your virtual private gateway. When the AWS Classic VPN connections are in the `deleted` state, you can then migrate to an AWS VPN connection by creating a new VPN connection on the same virtual private gateway. You do not have to make any changes to your existing AWS Direct Connect private virtual interface.

> **Important**
> During this procedure, connectivity over your AWS Direct Connect private virtual interface is not interrupted, but you will not have any connectivity over the Site-to-Site VPN connection (zero downtime with loss of redundancy). Connectivity through the VPN connection is restored when VPN connections are recreated on the virtual private gateway. Ensure that you plan for this loss of redundancy.
> After you've deleted the AWS Classic VPN connection, you cannot revert or migrate your new AWS VPN connection back to an AWS Classic VPN connection.

To migrate to an AWS 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**, and choose the AWS Classic VPN connection. Choose **Actions**, **Delete**.
3. Remove the previous tunnel configurations from your customer gateway device.
4. Repeat the previous two steps until you've deleted all the existing AWS Classic VPN connections for the virtual private gateway. Wait for the VPN connections to enter the `deleted` state.
5. Choose **Create VPN Connection**. Specify the following information, and choose **Create VPN Connection**.
   - **Virtual Private Gateway**: Choose the virtual private gateway that you used for the AWS Classic VPN connection.
   - **Customer Gateway**: Choose **Existing**, and select the existing customer gateway for your current AWS Classic VPN connection.
   - Specify the routing options as required.
6. Select the new Site-to-Site VPN connection and choose **Download Configuration**. Download the appropriate configuration file for your customer gateway device.
7. Use the configuration file to configure VPN tunnels on your customer gateway device. For more information, see Your customer gateway device (p. 32).
8. Enable the new tunnels on your customer gateway device. To bring the tunnels up, you must initiate the connection from your local network.

If applicable, check your route tables to ensure that the routes are being propagated. The routes propagate to the route table when the status of the VPN tunnel is `UP`.

Creating a transit gateway VPN attachment

To create a VPN attachment on a transit gateway, you must specify the customer gateway. For more information about creating a transit gateway, see Transit gateways in *Amazon VPC Transit Gateways*.

To create a VPN attachment 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. For **Target Gateway Type**, choose **Transit Gateway**, and choose the transit gateway on which to create the attachment.
5. For **Customer Gateway**, do one of the following:
   - To use an existing customer gateway, choose **Existing**, and then select the gateway to use.

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

     For **IP Address**, enter a static public IP address. For **BGP ASN**, enter the Border Gateway Protocol (BGP) Autonomous System Number (ASN) of your customer gateway. For **Certificate ARN**, choose the ARN of your private certificate (if using certificate-based authentication).

     For **Routing options**, choose whether to use **Dynamic** or **Static**.
6. (Optional) For **Enable Acceleration**, select the check box to enable acceleration. For more information, see Accelerated Site-to-Site VPN connections (p. 13).

   If you enable acceleration, we create two accelerators that are used by your VPN connection. Additional charges apply.
7. For **Tunnel Options**, see Tunnel options for your Site-to-Site VPN connection (p. 5).
8. 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.

Testing the 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. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI, and then choose **Select**.
4. Choose an instance type, and then choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, for **Network**, select your VPC. For **Subnet**, select your subnet. Choose **Next** until you reach the **Configure Security Group** page.
6. Select the **Select an existing security group** option, and then select the group that you configured earlier. Choose **Review and Launch**.
7. Review the settings that you've chosen. Make any changes that you need, and then choose **Launch** to select a key pair and launch the 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. A successful response is similar to the following:

   ```
   ping 10.0.0.4
   ```

   ```
   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 repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.

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*.

Deleting 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.

> **Important**
> If you delete your Site-to-Site VPN connection and then create a new one, you have to download a new configuration file and reconfigure the customer gateway device.

Topics

Deleting a Site-to-Site 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 Site-to-Site 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 Site-to-Site VPN connection and choose **Actions**, **Delete**.
4. Choose **Delete**.

To delete a Site-to-Site 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)

Deleting 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 to delete and choose **Actions**, **Delete Customer Gateway**.
3. Choose **Yes, 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)

Detaching and deleting 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 **Yes, Detach**.

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.

To delete a virtual private gateway using the console

1. In the navigation pane, choose **Virtual Private Gateways**.
2. Select the virtual private gateway to delete and choose **Actions**, **Delete Virtual Private Gateway**.
3. Choose **Yes, 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)

Modifying a Site-to-Site VPN connection's target gateway

You can modify the target gateway of 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 transit gateway

Before you perform the migration to the new gateway, you must configure the new gateway. For information about adding a virtual private gateway, see the section called "Create a virtual private gateway" (p. 20). 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 (required when the new gateway has a different ASN from the old gateway)" (p. 110).

Step 2: Delete your static routes (required for a static VPN connection migrating to a transit gateway)

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. In the **Routes** tab, choose **Edit**, and then choose **Remove** for the static route to the virtual private gateway.
4. Choose **Save** when you are done.

Step 3: Migrate to a new 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 Site-to-Site VPN connection and choose **Actions**, **Modify VPN Connection**.
4. Under **Change Target**, do the following:

a. For **Target Type**, choose the gateway type.

b. Configure the connection target:

[Virtual private gateway] For **Target VPN Gateway ID**, choose the virtual private gateway ID.

[Transit Gateway] For **Target transit gateway ID**, choose the transit gateway ID.

5. Choose **Save**.

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. The following table provides information about the actions you need to take. For information about updating VPC route tables, see Route tables in the *Amazon VPC User Guide*.

VPN gateway target modification required VPC route table updates

Existing gateway	New gateway	VPC route table change
Virtual private gateway with propagated routes	Transit gateway	Add a route that points to the transit gateway ID.
Virtual private gateway with propagated routes	Virtual private gateway with propagated routes	There is no action required.
Virtual gateway with propagated routes	Virtual private gateway with static route	Add an entry that contains the new virtual private gateway ID.
Virtual gateway with static routes	Transit gateway	Update the VPC route table and change the entry that contains to the virtual private gateway ID to the transit gateway ID.
Virtual gateway with static routes	Virtual private gateway with static routes	Update the entry that points to the virtual private gateway ID to be the new virtual private gateway ID.
Virtual gateway with static routes	Virtual private gateway with propagated routes	Delete the entry that contains the virtual private gateway ID.
Transit Gateway	Virtual private gateway with static routes	Update the entry that contains the transit gateway to the virtual private gateway ID.
Transit Gateway	Virtual private gateway with propagated routes	Delete the entry that contains the transit gateway ID.
Transit Gateway	Transit Gateway	Update the entry that contains the transit gateway ID to the new transit gateway ID.

Step 5: Update the transit gateway routing (required when the new gateway is a transit gateway)

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 information about transit gateway routing, see Transit gateway route tables in the *Amazon VPC Transit Gateways*.

> **Important**
> If you deleted VPN static routes, you must add the static routes to the transit gateway route table.

Step 6: Update the customer gateway ASN (required when the new gateway has a different ASN from the old gateway)

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.

Modifying 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 values for the options that you want to modify.
5. Choose **Save**.

To modify the VPN connection options using the command line or API

- modify-vpn-connection-options (AWS CLI)
- ModifyVpnConnectionOptions (Amazon EC2 Query API)

Modifying 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 that you're modifying options for.
5. Choose or enter new values for the tunnel options. For more information, see Tunnel options for your Site-to-Site VPN connection (p. 5).
6. Choose **Save**.

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.

Editing 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, modify, or remove the static routes for your VPN configuration.

To add, modify, or remove a static route

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 **Static Routes**, **Edit**.
4. Modify your existing static IP prefixes, or choose **Remove** to delete them. Choose **Add Another Rule** to add a new IP prefix to your configuration. When you are done, choose **Save**.

> **Note**
> 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 Site-to-Site VPN connection. For more information, see (Virtual private gateway) Enable route propagation in your route table (p. 21).

For a Site-to-Site 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.

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)

Changing 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 Site-to-Site 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 Site-to-Site VPN connection, and then choose **Actions**, **Modify VPN Connection**.
4. For **Target Type**, choose **Customer Gateway**.
5. For **Target Customer Gateway ID**, choose the ID for the customer gateway that you want to use for the connection.

To change the customer gateway using the command line or API

- ModifyVpnTunnelC (Amazon EC2 Query API)
- modify-vpn-tunnel (AWS CLI)

Replacing compromised credentials

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 (p. 9).

To change the IKE pre-shared key

You can modify the tunnel options for the Site-to-Site VPN connection and specify a new IKE pre-shared key for each tunnel. For more information, see Modifying Site-to-Site VPN tunnel options (p. 110).

Alternatively, you can delete the Site-to-Site VPN connection. For more information, see Deleting a Site-to-Site VPN connection (p. 105). You don't need to delete the VPC or the virtual private gateway. Then, create a new Site-to-Site 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 Site-to-Site VPN connection (p. 22). The tunnel's inside and outside addresses might change when you recreate the Site-to-Site VPN connection.

To change the certificate for the AWS side of the tunnel endpoint

Rotate the certificate. For more information, see the section called “Rotating Site-to-Site VPN tunnel endpoint certificates” (p. 113).

To change the certificate on the customer gateway device

1. Create a new certificate. For information about creating an ACM certificate, see Getting started in the *AWS Certificate Manager User Guide*.
2. Add the certificate to the customer gateway device.

Rotating 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 “Permissions granted by the service-linked role” (p. 119).

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, Rotate Tunnel Certificates**.
4. Select the tunnel endpoint whose certificate you want to rotate.
5. Choose **Save**.

To rotate the Site-to-Site VPN tunnel endpoint certificate using the AWS CLI

Use the modify-vpn-tunnel-certificate command.

Security in AWS Site-to-Site VPN

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

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

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. 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 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.

Site-to-Site VPN is part of the Amazon VPC service. For more information about security in Amazon VPC, see Security in the *Amazon VPC User Guide*.

The following topics show you how to configure specific components of Site-to-Site VPN to meet your security and compliance objectives.

Contents

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. This content includes 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 user accounts with 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 recommend TLS 1.2 or later.

- 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 personal 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 sensitive identifying information, such as your customers' account numbers, into free-form 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 Site-to-Site VPN or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

Site-to-Site VPN configuration information

When you create a Site-to-Site VPN connection, we generate configuration information that you use to set up your customer gateway device, including the pre-shared keys (if applicable). To prevent unauthorized access to the configuration information, ensure that you grant IAM users only the permissions that they need. If you download the configuration file from the Amazon VPC console, only distribute it to the users who will configure the customer gateway device. For more information, see the following topics:

- Site-to-Site VPN tunnel authentication options (p. 9)
- Identity and access management for AWS Site-to-Site VPN (p. 116)

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 (p. 5): Provides information about the IPsec and Internet Key Exchange (IKE) options that are available for each tunnel.
- Site-to-Site VPN tunnel authentication options (p. 9): Provides information about the authentication options for your VPN tunnel endpoints.
- Requirements for your customer gateway device (p. 34): 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 (p. 27): 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 uses security credentials to identify you and to grant you access to your AWS resources. You can use features of AWS Identity and Access Management (IAM) to allow other users, services, and applications to use your AWS resources fully or in a limited way, without sharing your security credentials.

By default, IAM users do not have permission to create, view, or modify AWS resources. To allow an IAM user to access resources, such as Site-to-Site VPN connections, virtual private gateways, and customer gateways, and to perform tasks, you must:

- Create an IAM policy that grants the IAM user permission to use the specific resources and API actions that they need
- Attach the policy to the IAM user or the group to which the IAM user belongs

When you attach a policy to a user or group of users, it allows or denies the users permission to perform the specified tasks on the specified resources.

Site-to-Site VPN is part of Amazon VPC, which shares its API namespace with Amazon EC2. To work with Site-to-Site VPN connections, virtual private gateways, and customer gateways, one of the following AWS managed policies might meet your needs:

- **PowerUserAccess**
- **ReadOnlyAccess**
- **AmazonEC2FullAccess**
- **AmazonEC2ReadOnlyAccess**

Take care when granting users permission to use the `ec2:DescribeVpnConnections` action. This action enables users to view customer gateway configuration information for Site-to-Site VPN connections in your account.

For more examples, see Identity and Access Management for Amazon VPC in the *Amazon VPC User Guide* and IAM Policies for Amazon EC2 in the *Amazon EC2 User Guide*.

IAM policies for your Site-to-Site VPN connection

You can use resource-level permissions to restrict what resources users can use when they invoke APIs. You can specify the Amazon Resource Name (ARN) for the VPN connection in the `Resource` element of IAM permission policy statements (for example, `arn:aws:ec2:us-west-2:123456789012:vpn-connection/vpn-0d4e855ab7d3536fb`).

The following actions support resource-level permissions for the VPN connection resource:

- `ec2:CreateVpnConnection`
- `ec2:ModifyVpnConnection`
- `ec2:ModifyVpnTunnelOptions`

The following are the supported condition keys:

Condition key	Description	Valid values	Type
ec2:AuthenticationType	The authentication type for the VPN tunnel endpoints.	Pre Shared Keys, Certificates	String
ec2:DPDTimeoutSeconds	The duration after which DPD timeout occurs.	An integer between 0 and 30	Numeric
ec2:GatewayType	The gateway type for the VPN endpoint on the AWS side of the VPN connection.	VGW, TGW	String
ec2:IKEVersions	The internet key exchange (IKE) versions that are permitted for the VPN tunnel.	ikev1, ikev2	String
ec2:InsideTunnelCidr	The range of inside IP addresses for the VPN tunnel.	See Tunnel options for your Site-to-Site VPN connection (p. 5)	String
ec2:Phase1DHGroupNumbers	The Diffie-Hellman groups that are permitted for the VPN tunnel for the phase 1 IKE negotiations.	2, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	Numeric
ec2:Phase2DHGroupNumbers	The Diffie-Hellman groups that are permitted for the VPN tunnel for the phase 2 IKE negotiations.	2, 5, 14, 15, 16, 17, 18, 22, 23, 24	Numeric
ec2:Phase1EncryptionAlgorithms	The encryption algorithms that are permitted for the VPN tunnel for the phase 1 IKE negotiations.	AES128, AES256, AES128-GCM-16, AES256-GCM-16	String
ec2:Phase2EncryptionAlgorithms	The encryption algorithms that are permitted for the VPN tunnel for the phase 2 IKE negotiations.	AES128, AES256	String
ec2:Phase1IntegrityAlgorithms	The integrity algorithms that are permitted for the VPN tunnel for the phase 1 IKE negotiations.	SHA1, SHA2-256	String
ec2:Phase2IntegrityAlgorithms	The integrity algorithms that are permitted for the VPN	SHA1, SHA2-384, SHA2-256, SHA2-512	String

Condition key	Description	Valid values	Type
	tunnel for the phase 2 IKE negotiations.		
ec2:Phase1LifetimeSeconds	The lifetime in seconds for phase 1 of the IKE negotiation.	An integer between 900 and 28,800	Numeric
ec2:Phase2LifetimeSeconds	The lifetime in seconds for phase 2 of the IKE negotiation.	An integer between 900 and 3,600	Numeric
ec2:PresharedKeys	The pre-shared key (PSK) to establish the initial IKE security association between the virtual private gateway and customer gateway.	See Tunnel options for your Site-to-Site VPN connection (p. 5)	String
ec2:RekeyFuzzPercentage	The percentage of the rekey window (determined by the rekey margin time) within which the rekey time is randomly selected.	An integer between 0 and100	Numeric
ec2:RekeyMarginTimeSeconds	The margin time before the phase 2 lifetime expires, during which AWS performs an IKE rekey.	An integer from 60 and above	Numeric
ec2:RoutingType	The routing type for the VPN connection.	Static, BGP	String

You can allow or deny specific values for each supported condition key using IAM condition operators. For more information, see IAM JSON policy elements: condition in the *IAM User Guide*.

The following example policy enables users to create VPN connections, but only VPN connections with static routing types.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpnConnection"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:RoutingType": [
                        "static"
                    ]
                }
            }
        }
    ]
}
```

AWS Site-to-Site VPN service-linked role

AWS Site-to-Site VPN uses service-linked roles for the permissions that it requires to call other AWS services on your behalf. For more information, see Using service-linked roles in the *IAM User Guide*.

Permissions granted by the service-linked role

When you work with a Site-to-Site VPN connection that uses certificate-based authentication, Site-to-Site VPN uses the service-linked role named **AWSServiceRoleForVPCS2SVPN** to call the following AWS Certificate Manager (ACM) actions on your behalf:

- `acm:ExportCertificate`
- `acm:DescribeCertificatee`
- `acm:ListCertificates`
- `acm-pca:DescribeCertificateAuthority`

Create the service-linked role

You don't need to manually create the **AWSServiceRoleForVPCS2SVPN** role. Site-to-Site VPN creates this role for you when you create a customer gateway with an associated ACM private certificate.

For a Site-to-Site VPN user to create a service-linked role on your behalf, you must have the required permissions. For information about service-linked roles, see Service-linked role permissions in the *IAM User Guide*.

Edit the service-linked role

You can edit the description of **AWSServiceRoleForVPCS2SVPN** using IAM. For information about editing service-linked roles, see Editing a service-linked role in the *IAM User Guide*.

Delete the service-linked role

If you no longer need to use the Site-to-Site VPN connections with certificate-based authentication, we recommend that you delete **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.

You can use the IAM console, the IAM CLI, or the IAM API to delete service-linked roles. For information about deleting service-linked roles, see Deleting a service-linked role in the *IAM User Guide*.

After you delete **AWSServiceRoleForVPCS2SVPN**, Amazon VPC creates the role again for a customer gateway with an associated ACM private certificate.

Logging and monitoring

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 AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your resources and responding to potential incidents.

Amazon CloudWatch

Amazon CloudWatch monitors your AWS resources and the applications that you run on AWS in real time. You can collect and track metrics for your Site-to-Site VPN tunnels, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For more information, see Monitoring your Site-to-Site VPN connection (p. 122).

AWS CloudTrail

AWS CloudTrail captures Amazon EC2 API calls and related events made by or on behalf of your AWS account. It then delivers the log files to an Amazon S3 bucket that you specify. For more information, see Logging Amazon EC2, Amazon EBS, and Amazon VPC API calls with AWS CloudTrail in the *Amazon EC2 API Reference*.

AWS Trusted Advisor

AWS Trusted Advisor draws upon best practices learned from serving hundreds of thousands of AWS customers. Trusted Advisor inspects your AWS environment, and then makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps.

Trusted Advisor has a check for VPN tunnel redundancy, which checks the number of tunnels that are active for each of your VPN connections.

For more information, see AWS Trusted Advisor in the *AWS Support 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 has two tunnels 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 (p. 11). 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 topics:

- Using redundant Site-to-Site VPN connections to provide failover (p. 29)
- Single data center HA network connectivity
- Multiple data center HA network connectivity

Infrastructure security in AWS Site-to-Site VPN

As a managed service, AWS Site-to-Site VPN is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access Site-to-Site VPN through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

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 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 (p. 123).

- **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 (p. 127).

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.

- 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

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.

> **Important**
> CloudWatch metrics are not supported for AWS Classic VPN connections. For more information, see Site-to-Site VPN categories (p. 4).

For more information, see the Amazon CloudWatch User Guide.

VPN tunnel metrics and dimensions

The following metrics are available for your VPN tunnels.

Metric	Description
`TunnelState`	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. Units: Fractional value between 0 and 1
`TunnelDataIn`	The bytes received on the AWS side of the connection through the VPN tunnel. 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. This metric counts the data after decryption. Units: Bytes
`TunnelDataOut`	The bytes sent from the AWS side of the connection through the VPN tunnel. 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. This metric counts the data before encryption. Units: Bytes

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 tunnel CloudWatch metrics

When you create a new Site-to-Site VPN connection, the VPN service sends the following metrics about your VPN tunnels to CloudWatch as it becomes available. You can view the metrics for VPN tunnels 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, for the Site-to-Site VPN connection).

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 VPN tunnel and sends a notification when the tunnel state is DOWN for 3 datapoints within 15 minutes.

To create an alarm for tunnel state

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Alarms**, **Create alarm**.
3. Choose **Select metric**.
4. Choose **VPN**, then choose **VPN Tunnel Metrics**.
5. Choose the IP address of the VPN tunnel and the **TunnelState** metric. Choose **Select metric**.
6. Choose **Next**.
7. Do one of the following, and then under **Additional configuration**, enter **3** for the datapoints to alarm. Choose **Next**.

   - To monitor when both tunnels are down, for **Whenever**, choose **Lower/Equal** (**<=**) , and then enter **0.5**.
   - To monitor the DOWN state for either tunnel, for **Whenever**, choose **Lower (<)**, and then enter **1**.
8. Under **Select an SNS topic**, select an existing notification list or 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 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, choose **Alarms**, **Create alarm**.
3. 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, choose **Alarms**, **Create alarm**.
3. 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, choose **Alarms**, **Create alarm**.
3. 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 Personal 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 Personal Health Dashboard.

The Personal Health Dashboard provides the following types of notifications for your VPN connections:

- Tunnel endpoint replacement notifications (p. 127)
- Single tunnel VPN notifications (p. 127)

Tunnel endpoint replacement notifications

You receive a **Tunnel endpoint replacement notification** in the Personal 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 (p. 11).

When a tunnel endpoint replacement is complete, AWS sends the **Tunnel endpoint replacement notification** through a Personal 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 weekly **VPN single tunnel notification** through a Personal Health Dashboard event.

Site-to-Site VPN quotas

Your AWS account has the following quotas, formerly referred to as limits, related to Site-to-Site VPN. To request an increase, use the limits form.

Site-to-Site VPN resources

- Customer gateways per Region: 50
- Virtual private gateways per Region: 5

  You can attach only 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 per Region: 50
- Site-to-Site VPN connections per virtual private gateway: 10

Routes

- Dynamic routes advertised from a customer gateway device to a Site-to-Site VPN connection on a virtual private gateway: 100

  This quota cannot be increased.
- Routes advertised from a Site-to-Site VPN connection on a virtual private gateway to a customer gateway device: 1,000

  Advertised route sources include VPC routes, other VPN routes, and routes from AWS Direct Connect virtual interfaces.

  This quota cannot be increased.
- Dynamic routes advertised from a customer gateway device to a Site-to-Site VPN connection on a transit gateway: 1,000
- Routes advertised from a Site-to-Site VPN connection on a transit gateway to a customer gateway device: 5,000

  Advertised routes come from the route table that's associated with the VPN attachment.

Bandwidth and throughput

- Maximum bandwidth per VPN tunnel: 1.25 Gbps

  This quota cannot be increased. 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 packets per second (PPS) per VPN tunnel: 140,000

Maximum transmission unit (MTU)

- You must set the MTU of the logical interface for your customer gateway device to 1399 bytes. For more information, see Requirements for your customer gateway device (p. 34).

  Jumbo frames are not supported. For more information, see Jumbo frames in the *Amazon EC2 User Guide for Linux Instances*.
- We recommend that you set the maximum segment size (MSS) on your customer gateway device to 1359 when using the SHA2-384 or SHA2-512 hashing algorithms.

Note
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

The following table describes the AWS Site-to-Site VPN User Guide updates.

update-history-change	update-history-description	update-history-date
VPN connection notifications	Site-to-Site VPN automatically sends notifications about your VPN connection to the Personal 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 Certificate Manager Private Certificate Authority private certificate support	You can use a private certificate from AWS Certificate Manager Private Certificate Authority to authenticate your VPN.	August 15, 2019
New Site-to-Site VPN User Guide (p. 130)	This release separates the AWS Site-to-Site VPN (previously known as AWS Managed VPN) content from the *Amazon VPC User Guide*.	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 categories	You can view the category of your VPN connection.	October 3, 2017
VPN metrics	You can view CloudWatch metrics for your VPN connections.	May 15, 2017
VPN enhancements (p. 130)	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 (p. 130)	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 (p. 130)	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 (p. 130)	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