

---

# Amazon Virtual Private Cloud

## Network Administrator Guide



# **Amazon Virtual Private Cloud: Network Administrator Guide**

Copyright © 2017 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

## Table of Contents

Welcome .....	1
Your Customer Gateway .....	2
What Is a Customer Gateway? .....	2
Your Role .....	3
Overview of Setting Up a VPN Connection .....	4
Network Information .....	4
AWS VPN CloudHub and Redundant Customer Gateways .....	5
Configuring Multiple VPN Connections to Your VPC .....	5
Customer Gateway Devices We've Tested .....	6
Requirements for Your Customer Gateway .....	7
Configuring a Firewall Between the Internet and Your Customer Gateway .....	10
Example: Check Point Device using BGP .....	12
High-Level View of the Customer Gateway .....	12
Configuration File .....	13
Configuring the Check Point Device .....	14
Step 1: Configure the Tunnel Interfaces .....	14
Step 2: Configure BGP .....	15
Step 3: Create Network Objects .....	16
Step 4: Create a VPN Community and Configure IKE and IPsec .....	17
Step 5: Configure the Firewall .....	18
Step 6: Enable Dead Peer Detection and TCP MSS Clamping .....	19
How to Test the Customer Gateway Configuration .....	20
Example: Check Point Device (without BGP) .....	23
High-Level View of the Customer Gateway .....	23
Configuration File .....	24
Configuring the Check Point Device .....	25
Step 1: Configure Tunnel Interface .....	25
Step 2: Configure the Static Route .....	26
Step 3: Create Network Objects .....	28
Step 4: Create a VPN Community and Configure IKE and IPsec .....	29
Step 5: Configure the Firewall .....	30
Step 6: Enable Dead Peer Detection and TCP MSS Clamping .....	31
How to Test the Customer Gateway Configuration .....	32
Example: Cisco ASA Device .....	35
A High-Level View of the Customer Gateway .....	35
An Example Configuration .....	36
How to Test the Customer Gateway Configuration .....	40
Example: Cisco IOS Device .....	42
A High-Level View of the Customer Gateway .....	43
A Detailed View of the Customer Gateway and an Example Configuration .....	44
How to Test the Customer Gateway Configuration .....	50
Example: Cisco IOS Device (without BGP) .....	53
A High-Level View of the Customer Gateway .....	53
A Detailed View of the Customer Gateway and an Example Configuration .....	54
How to Test the Customer Gateway Configuration .....	60
Example: Dell SonicWALL Device .....	62
A High-Level View of the Customer Gateway .....	62
Example Configuration File .....	63
Configuring the SonicWALL Device Using the Management Interface .....	67
How to Test the Customer Gateway Configuration .....	67
Example: Dell SonicWALL Device (without BGP) .....	69
A High-Level View of the Customer Gateway .....	69
Example Configuration File .....	70
Configuring the SonicWALL Device Using the Management Interface .....	73

How to Test the Customer Gateway Configuration .....	75
Example: Fortinet Fortigate Device .....	77
A High-Level View of the Customer Gateway .....	78
A Detailed View of the Customer Gateway and an Example Configuration .....	78
How to Test the Customer Gateway Configuration .....	86
Example: Juniper J-Series JunOS Device .....	88
A High-Level View of the Customer Gateway .....	89
A Detailed View of the Customer Gateway and an Example Configuration .....	90
How to Test the Customer Gateway Configuration .....	96
Example: Juniper SRX JunOS Device .....	98
A High-Level View of the Customer Gateway .....	99
A Detailed View of the Customer Gateway and an Example Configuration .....	100
How to Test the Customer Gateway Configuration .....	106
Example: Juniper ScreenOS Device .....	108
A High-Level View of the Customer Gateway .....	109
A Detailed View of the Customer Gateway and an Example Configuration .....	110
How to Test the Customer Gateway Configuration .....	115
Example: Netgate PfSense Device (without BGP) .....	118
A High-Level View of the Customer Gateway .....	118
Example Configuration .....	119
How to Test the Customer Gateway Configuration .....	122
Example: Palo Alto Networks Device .....	124
A High-Level View of the Customer Gateway .....	125
A Detailed View of the Customer Gateway and an Example Configuration .....	125
How to Test the Customer Gateway Configuration .....	132
Example: Yamaha Device .....	134
A High-Level View of the Customer Gateway .....	135
A Detailed View of the Customer Gateway and an Example Configuration .....	135
How to Test the Customer Gateway Configuration .....	141
Example: Generic Customer Gateway Using BGP .....	143
A High-Level View of the Customer Gateway .....	144
A Detailed View of the Customer Gateway and an Example Configuration .....	144
How to Test the Customer Gateway Configuration .....	149
Example: Generic Customer Gateway (without BGP) .....	151
A High-Level View of the Customer Gateway .....	152
A Detailed View of the Customer Gateway and an Example Configuration .....	152
How to Test the Customer Gateway Configuration .....	157
Troubleshooting .....	159
Cisco ASA Customer Gateway Connectivity .....	159
IKE .....	159
IPsec .....	160
Routing .....	161
Cisco IOS Customer Gateway Connectivity .....	162
IKE .....	162
IPsec .....	163
Tunnel .....	164
BGP .....	165
Virtual Private Gateway Attachment .....	166
Cisco IOS Customer Gateway Connectivity (without BGP) .....	166
IKE .....	167
IPsec .....	167
Tunnel .....	169
Virtual Private Gateway Attachment .....	171
Juniper JunOS Customer Gateway Connectivity .....	171
IKE .....	171
IPsec .....	171
Tunnel .....	172

BGP .....	172
Virtual Private Gateway Attachment .....	174
Juniper ScreenOS Customer Gateway Connectivity .....	174
IKE and IPsec .....	174
Tunnel .....	174
BGP .....	175
Virtual Private Gateway Attachment .....	176
Yamaha Customer Gateway Connectivity .....	177
IKE .....	177
IPsec .....	177
Tunnel .....	178
BGP .....	178
Virtual Private Gateway Attachment .....	179
Generic Device Customer Gateway Connectivity .....	180
Generic Device Customer Gateway Connectivity (without BGP) .....	183
Configuring Windows Server 2008 R2 as a Customer Gateway .....	186
Configuring Your Windows Server .....	186
Step 1: Create a VPN Connection and Configure Your VPC .....	187
Step 2: Download the Configuration File for the VPN Connection .....	188
Step 3: Configure the Windows Server .....	190
Step 4: Set Up the VPN Tunnel .....	192
Option 1: Run netsh Script .....	192
Option 2: Use the Windows Server User Interface .....	192
Step 5: Enable Dead Gateway Detection .....	198
Step 6: Test the VPN Connection .....	198
Configuring Windows Server 2012 R2 as a Customer Gateway .....	200
Configuring Your Windows Server .....	200
Step 1: Create a VPN Connection and Configure Your VPC .....	201
Step 2: Download the Configuration File for the VPN Connection .....	202
Step 3: Configure the Windows Server .....	204
Step 4: Set Up the VPN Tunnel .....	205
Option 1: Run netsh Script .....	205
Option 2: Use the Windows Server User Interface .....	205
2.4: Configure the Windows Firewall .....	209
Step 5: Enable Dead Gateway Detection .....	210
Step 6: Test the VPN Connection .....	211
Document History .....	213

# Welcome

Welcome to the *Amazon VPC Network Administrator Guide*. This guide is for customers who plan to use an AWS managed IPsec VPN connection with their virtual private cloud (VPC). The topics in this guide help you configure your customer gateway, which is the device on your side of the VPN connection.

The VPN connection lets you bridge your VPC and IT infrastructure, and extend your existing security and management policies to EC2 instances in your VPC as if they were running within your own infrastructure.

For more information, see the following topics:

- [Your Customer Gateway \(p. 2\)](#)
- [Example: Check Point Device with Border Gateway Protocol \(p. 12\)](#)
- [Example: Check Point Device without Border Gateway Protocol \(p. 23\)](#)
- [Example: Cisco ASA Device \(p. 35\)](#)
- [Example: Cisco IOS Device \(p. 42\)](#)
- [Example: Cisco IOS Device without Border Gateway Protocol \(p. 53\)](#)
- [Example: Dell SonicWALL SonicOS Device Without Border Gateway Protocol \(p. 69\)](#)
- [Example: Dell SonicWALL Device \(p. 62\)](#)
- [Example: Juniper J-Series JunOS Device \(p. 88\)](#)
- [Example: Juniper SRX JunOS Device \(p. 98\)](#)
- [Example: Juniper ScreenOS Device \(p. 108\)](#)
- [Example: Netgate PfSense Device without Border Gateway Protocol \(p. 118\)](#)
- [Example: Palo Alto Networks Device \(p. 124\)](#)
- [Example: Yamaha Device \(p. 134\)](#)
- [Example: Generic Customer Gateway Using Border Gateway Protocol \(p. 143\)](#)
- [Example: Generic Customer Gateway without Border Gateway Protocol \(p. 151\)](#)
- [Configuring Windows Server 2008 R2 as a Customer Gateway \(p. 186\)](#)
- [Configuring Windows Server 2012 R2 as a Customer Gateway \(p. 200\)](#)

# Your Customer Gateway

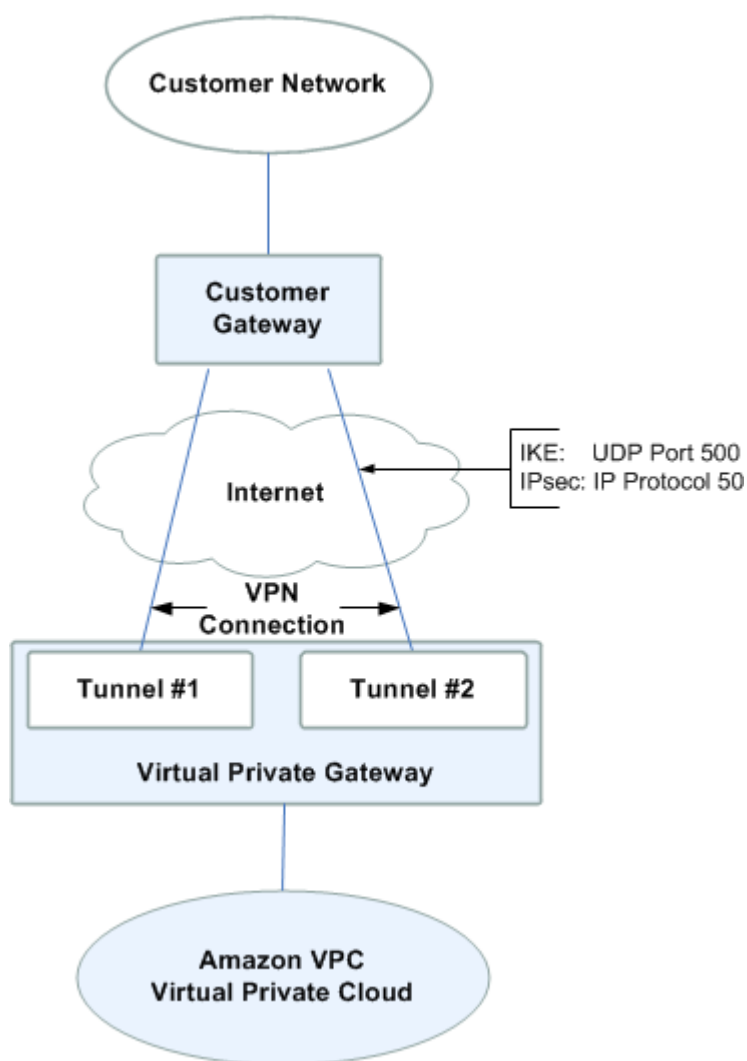
## Topics

- [What Is a Customer Gateway? \(p. 2\)](#)
- [Overview of Setting Up a VPN Connection \(p. 4\)](#)
- [AWS VPN CloudHub and Redundant Customer Gateways \(p. 5\)](#)
- [Configuring Multiple VPN Connections to Your VPC \(p. 5\)](#)
- [Customer Gateway Devices We've Tested \(p. 6\)](#)
- [Requirements for Your Customer Gateway \(p. 7\)](#)
- [Configuring a Firewall Between the Internet and Your Customer Gateway \(p. 10\)](#)

## What Is a Customer Gateway?

An Amazon VPC VPN connection links your data center (or network) to your Amazon VPC virtual private cloud (VPC). A *customer gateway* is the anchor on your side of that connection. It can be a physical or software appliance. The anchor on the AWS side of the VPN connection is called a *virtual private gateway*.

The following diagram shows your network, the customer gateway, the VPN connection that goes to the virtual private gateway, and the VPC. There are two lines between the customer gateway and virtual private gateway because the VPN connection consists of two tunnels to provide increased availability for the Amazon VPC service. 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 virtual private gateway, which may briefly disable one of the two tunnels of your VPN connection. Your VPN connection automatically fails over to the second tunnel while this maintenance is performed. When you configure your customer gateway, it's therefore important that you configure both tunnels.



You can create additional VPN connections to other VPCs using the same customer gateway device. You can reuse the same customer gateway IP address for each of those VPN connections.

When you create a VPN connection, the VPN tunnel comes up when traffic is generated from your side of the VPN connection. The virtual private gateway is not the initiator; your customer gateway must initiate the tunnels.

For more information about the components of a VPN connection, see [VPN Connections](#) in the *Amazon VPC User Guide*.

To protect against a loss of connectivity if your customer gateway becomes unavailable, you can set up a second VPN connection. For more information, see [Using Redundant VPN Connections to Provide Failover](#).

## Your Role

Throughout this guide, we refer to your company's *integration team*, which is the person (or persons) at your company working to integrate your infrastructure with Amazon VPC. This team (which may or



may not consist of you) must use the [AWS Management Console](#) to create a VPN connection and get the information that you need to configure your customer gateway. Your company might have a separate team for each task (an integration team that uses the AWS Management Console, and a separate network engineering group that has access to network devices and configures the customer gateway). This guide assumes that you're someone in the network engineering group who receives information from your company's integration team so you can then configure the customer gateway device.

## Overview of Setting Up a VPN Connection

The process of setting up the VPN connection in AWS is covered in the *Amazon VPC User Guide*. One task in the overall process is to configure the customer gateway. To create the VPN connection, AWS needs information about the customer gateway, and you must configure the customer gateway device itself.

To set up a VPN connection, follow these general steps:

1. Designate an appliance to act as your customer gateway. For more information, see [Customer Gateway Devices We've Tested \(p. 6\)](#) and [Requirements for Your Customer Gateway \(p. 7\)](#).
2. Get the necessary [Network Information \(p. 4\)](#), and provide this information to the team that will create the VPN connection in AWS.
3. Create the VPN connection in AWS and get the configuration file for your customer gateway. For more information, see [Setting Up an AWS VPN Connection](#) in the *Amazon VPC User Guide*.
4. Configure your customer gateway using the information from the configuration file. Examples are provided in this guide.
5. Generate traffic from your side of the VPN connection to bring up the VPN tunnel.

## Network Information

To create a VPN connection in AWS, you need the following information.

Item	Comments
Customer gateway vendor (for example, Cisco), platform (for example, ISR Series Routers), and software version (for example, IOS 12.4)	This information is used to generate a configuration file for the customer gateway.
The internet-routable IP address for the customer gateway device's external interface.	The value must be static. Your customer gateway may reside behind a device performing network address translation (NAT).
(Optional) Border Gateway Protocol (BGP) Autonomous System Number (ASN) of the customer gateway.	You can use an existing ASN assigned to your network. If you don't have one, you can use a private ASN in the 64512–65534 range. Otherwise, we assume that the BGP ASN for the customer gateway is 65000.
(Optional) The ASN for the Amazon side of the BGP session.	Specified when creating a virtual private gateway. If you do not specify a value, the default ASN applies. For more information, see <a href="#">Virtual Private Gateway</a> .
(Optional) Tunnel information for each VPN tunnel	You can specify the following tunnel information for the VPN connection: <ul style="list-style-type: none"><li>• Inside tunnel CIDR</li></ul>

Item	Comments
	<ul style="list-style-type: none"><li>• Pre-shared key to establish the initial IKE Security Association between the virtual private gateway and customer gateway.</li></ul> <p>For more information, see <a href="#">Configuring the VPN Tunnels for Your VPN Connection</a>.</p>

## AWS VPN CloudHub and Redundant Customer Gateways

You can establish multiple VPN connections to a single virtual private gateway from multiple customer gateways. This configuration can be used in different ways; you can have redundant customer gateways between your data center and your VPC, or you can have multiple locations connected to the AWS VPN CloudHub.

If you have redundant customer gateways, each customer gateway advertises the same prefix (for example, 0.0.0.0/0) to the virtual private gateway. We use BGP routing to determine the path for traffic. If one customer gateway fails, the virtual private gateway directs all traffic to the working customer gateway.

If you use the AWS VPN CloudHub configuration, multiple sites can access your VPC or securely access each other using a simple hub-and-spoke model. You configure each customer gateway to advertise a site-specific prefix (such as 10.0.0.0/24, 10.0.1.0/24) to the virtual private gateway. The virtual private gateway routes traffic to the appropriate site and advertises the reachability of one site to all other sites.

To configure the AWS VPN CloudHub, use the Amazon VPC console to create multiple customer gateways, each with the public IP address of the gateway. You can use the same Border Gateway Protocol (BGP) Autonomous System Number (ASN) for each, or if you prefer, you can use a unique ASN for each. Then create a VPN connection from each customer gateway to a common VPN gateway. Use the instructions that follow to configure each customer gateway to connect to the virtual private gateway.

To enable instances in your VPC to reach the virtual private gateway (and then your customer gateways), you must configure routes in your VPC routing tables. For complete instructions, see the [Amazon VPC User Guide](#). For AWS VPN CloudHub, you can configure an aggregate route in your VPC routing table (for example, 10.0.0.0/16), and use more specific prefixes between customer gateways and the virtual private gateway.

## Configuring Multiple VPN Connections to Your VPC

You can create up to ten VPN connections for your VPC. You can use multiple VPN connections to link your remote offices to the same VPC. For example, if you have offices in Los Angeles, Chicago, New York, and Miami, you can link each of these offices to your VPC. You can also use multiple VPN connections to establish redundant customer gateways from a single location.

### Note

If you need more than ten VPN connections, complete the [Request to Increase Amazon VPC Limits](#) form to request an increased limit.

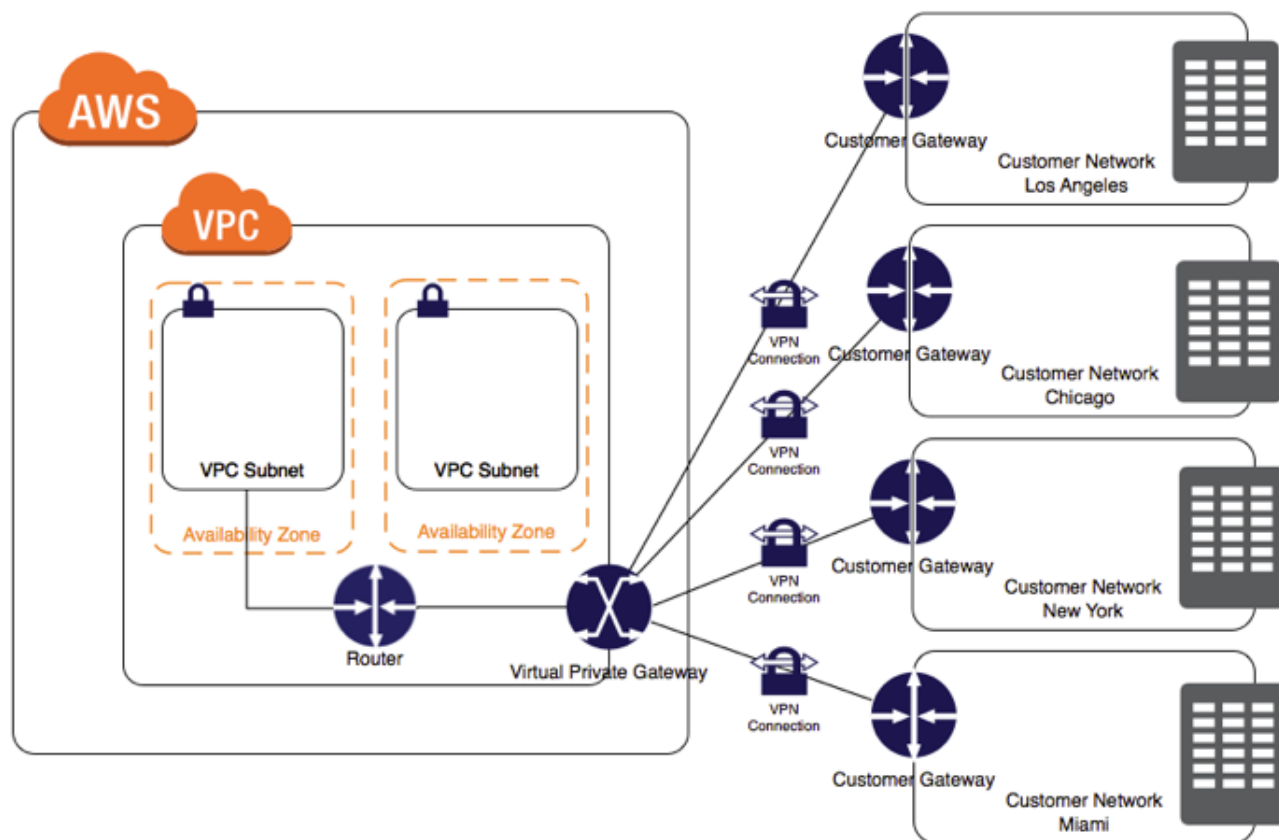
When you create multiple VPN connections, the virtual private gateway sends network traffic to the appropriate VPN connection using statically assigned routes or BGP route advertisements, depending upon how the VPN connection was configured. Statically assigned routes are preferred over BGP advertised routes in cases where identical routes exist in the virtual private gateway.

When you have customer gateways at multiple geographic locations, each customer gateway should advertise a unique set of IP ranges specific to the location. When you establish redundant customer gateways at a single location, both gateways should advertise the same IP ranges.

The virtual private gateway receives routing information from all customer gateways and calculates the set of preferred paths using the BGP best path selection algorithm. The rules of that algorithm, as it applies to VPC, are:

1. The most specific IP prefix is preferred (for example, 10.0.0.0/24 is preferable to 10.0.0.0/16). For more information, see [Route Priority](#) in the *Amazon VPC User Guide*.
2. When the prefixes are the same, statically configured VPN connections, if they exist, are preferred. For matching prefixes where each VPN connection uses BGP, the AS PATH is compared and the prefix with the shortest AS PATH is preferred. Alternatively, you can prepend AS\_PATH, so that the path is less preferred.
3. When the AS PATHs are the same length, the path origin is compared. Prefixes with an Interior Gateway Protocol (IGP) origin are preferred to Exterior Gateway Protocol (EGP) origins, which are preferred to unknown origins.

The following diagram shows the configuration of multiple VPNs.



## Customer Gateway Devices We've Tested

Your customer gateway can be a physical or software appliance.

For information about the specific routers that we've tested, see [What customer gateway devices are known to work with Amazon VPC?](#) in the Amazon VPC FAQ.





This guide presents information about how to configure the following devices:

- 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
- Dell 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

If you have one of these devices, but configure it for IPsec in a different way than presented in this guide, feel free to alter our suggested configuration to match your particular needs.

## Requirements for Your Customer Gateway

There are four main parts to the configuration of your customer gateway. Throughout this guide, we use a symbol for each of these parts to help you understand what you need to do. The following table shows the four parts and the corresponding symbols.



	IKE Security Association (required to exchange keys used to establish the IPsec security association)
	IPsec Security Association (handles the tunnel's encryption, authentication, and so on.)
	Tunnel interface (receives traffic going to and from the tunnel)
Optional 	BGP peering (exchanges routes between the customer gateway and the virtual private gateway) for devices that use BGP

If you have a device that isn't in the preceding list of tested devices, this section describes the requirements the device must meet for you to use it with Amazon VPC. The following table lists the requirement the customer gateway must adhere to, the related RFC (for reference), and comments about the requirement. For an example of the configuration information if your device isn't one of the tested Cisco or Juniper devices, see [Example: Generic Customer Gateway Using Border Gateway Protocol](#) (p. 143).

Each VPN connection consists of 2 separate tunnels. Each tunnel contains an IKE Security Association, an IPsec Security Association, and a BGP Peering. You are limited to 1 unique Security Association (SA) pair

per tunnel (1 inbound and 1 outbound), and therefore 2 unique SA pairs in total for 2 tunnels (4 SAs). Some devices use policy-based VPN and will create as many SAs as ACL entries. Therefore, you may need to consolidate your rules and then filter so you don't permit unwanted traffic.

The VPN tunnel comes up when traffic is generated from your side of the VPN connection. The AWS endpoint is not the initiator; your customer gateway must initiate the tunnels.

Requirement	RFC	Comments
Establish IKE Security Association using pre-shared keys  	<a href="#">RFC 2409</a>	<p>The IKE Security Association is established first between the virtual private gateway and customer gateway using the pre-shared key as the authenticator. Upon establishment, IKE negotiates an ephemeral key to secure future IKE messages. Proper establishment of an IKE Security Association requires complete agreement among the parameters, including encryption and authentication parameters.</p> <p>When you create a VPN connection in AWS, you can specify your own pre-shared key for each tunnel, or you can let AWS generate one for you. For more information, see <a href="#">Configuring the VPN Tunnels for Your VPN Connection</a>.</p>
Establish IPsec Security Associations in Tunnel mode  	<a href="#">RFC 4301</a>	<p>Using the IKE ephemeral key, keys are established between the virtual private gateway and customer gateway to form an IPsec Security Association (SA). Traffic between gateways is encrypted and decrypted using this SA. The ephemeral keys used to encrypt traffic within the IPsec SA are automatically rotated by IKE on a regular basis to ensure confidentiality of communications.</p>
Utilize the AES 128-bit encryption or AES 256-bit encryption function	<a href="#">RFC 3602</a>	The encryption function is used to ensure privacy among both IKE and IPsec Security Associations.
Utilize the SHA-1 or SHA-256 hashing function	<a href="#">RFC 2404</a>	This hashing function is used to authenticate both IKE and IPsec Security Associations.
Utilize Diffie-Hellman Perfect Forward Secrecy. The following groups are supported: <ul style="list-style-type: none"> <li>Phase 1 groups: 2, 14-18, 22, 23, and 24</li> <li>Phase 2 groups: 2, 5, 14-18, 22, 23, and 24</li> </ul>	<a href="#">RFC 2409</a>	IKE uses Diffie-Hellman to establish ephemeral keys to secure all communication between customer gateways and VPN gateways.
Utilize IPsec Dead Peer Detection	<a href="#">RFC 3706</a>	The use of 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 utilized if possible.
Bind tunnel to logical interface (route-based VPN)	None	Your gateway must support the ability to bind the IPsec tunnel to a logical interface. The logical interface

Requirement	RFC	Comments
<b>Tunnel</b>		contains an IP address used to establish BGP peering to the virtual private gateway. This logical interface should perform no additional encapsulation (for example, GRE, IP in IP). Your interface should be set to a 1399 byte Maximum Transmission Unit (MTU).
Fragment IP packets before encryption	<a href="#">RFC 4459</a>	When packets are too large to be transmitted, they must be fragmented. We will not reassemble fragmented encrypted packets. Therefore, your VPN device must fragment packets <i>before</i> encapsulating with the VPN headers. The fragments are individually transmitted to the remote host, which reassembles them. For more information about fragmentation, see the <a href="#">IP fragmentation</a> Wikipedia article.
(Optional) Establish BGP peerings <b>BGP</b>	<a href="#">RFC 4271</a>	BGP is used to exchange routes between the customer gateway and 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 reachable through the IPsec SA.

We recommend you use the techniques listed in the following table to minimize problems related to the amount of data that can be transmitted through the IPsec tunnel. 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.

Technique	RFC	Comments
Adjust the maximum segment size of TCP packets entering the VPN tunnel	<a href="#">RFC 4459</a>	TCP packets are often the most prevalent type of packet across IPsec tunnels. Some gateways have the ability to change the TCP Maximum Segment Size 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.
Reset the "Don't Fragment" flag on packets	<a href="#">RFC 791</a>	Some packets carry a flag, known as the Don't Fragment (DF) flag, that 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 reducing the amount of data transmitted in each packet. Some VPN devices have the ability to override the DF flag and fragment packets unconditionally as required. If your customer gateway has this ability, we recommend that you use it as appropriate.

If you have a firewall between your customer gateway and the Internet, see [Configuring a Firewall Between the Internet and Your Customer Gateway](#) (p. 10).

## Configuring a Firewall Between the Internet and Your Customer Gateway

To use this service, you must have an internet-routable IP address to use as the endpoint for the IPsec tunnels connecting your customer gateway 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 information that you'll get from the integration team.

### 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 containing the encrypted network traffic.

If you are using NAT traversal (NAT-T) on your device, then you must include rules that allow UDP access over port 4500. Check if your device is advertising NAT-T.



# Example: Check Point Device with Border Gateway Protocol

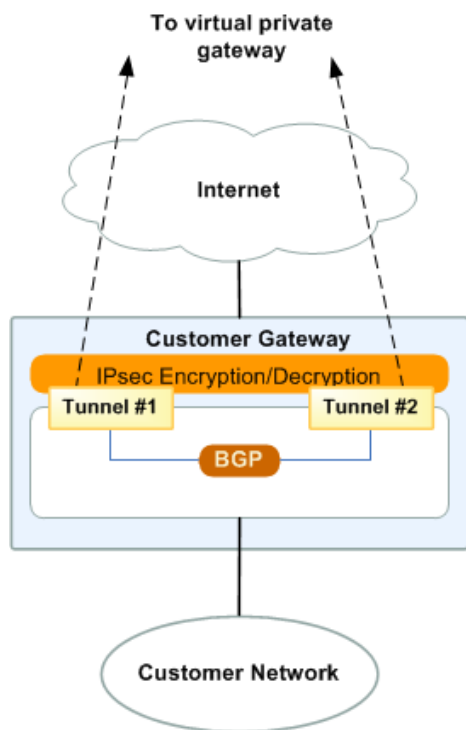
This section has example configuration information provided by your integration team if your customer gateway is a Check Point Security Gateway device running R77.10 or above, and using the Gaia operating system.

## Topics

- [High-Level View of the Customer Gateway \(p. 12\)](#)
- [Configuration File \(p. 13\)](#)
- [Configuring the Check Point Device \(p. 14\)](#)
- [How to Test the Customer Gateway Configuration \(p. 20\)](#)

## High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels. Using redundant tunnels ensures continuous availability in the case that a device fails.



## Configuration File

Your integration team will provide you with a configuration file with the values you need in order to configure each tunnel and the IKE and IPsec settings for your VPN device. The configuration file includes instructions on how to use the Gaia web portal and Check Point SmartDashboard to configure your device. The same steps are provided in the next section.

The following is an extract of an example configuration file. The file contains two sections: IPsec Tunnel #1 and IPsec Tunnel #2. You must use the values provided in each section to configure each tunnel.

```
! Amazon Web Services
! Virtual Private Cloud

! AWS uses unique identifiers to manipulate the configuration of
! a VPN connection. Each VPN connection is assigned an identifier and is
! associated with two other identifiers, namely the
! customer gateway identifier and virtual private gateway identifier.
!
! Your VPN connection ID      : vpn-12345678
! Your virtual private gateway ID : vgw-12345678
! Your customer gateway ID    : cgw-12345678
!
!
! This configuration consists of two tunnels. Both tunnels must be
! configured on your customer gateway.
!
! -----
! IPsec Tunnel #1
! -----
```

```
! #1: Tunnel Interface Configuration
...
! -----
! -----
! IPsec Tunnel #2
! -----
! #1: Tunnel Interface Configuration
...

```

## Configuring the Check Point Device

The following procedures demonstrate how to configure the VPN tunnels, network objects, and security for your VPN connection. You must replace the example values in the procedures with the values that are provided in the configuration file.

### Note

For more information, go to the [Amazon Web Services \(AWS\) VPN BGP](#) article on the Check Point Support Center.

### Topics

- [Step 1: Configure the Tunnel Interfaces \(p. 14\)](#)
- [Step 2: Configure BGP \(p. 15\)](#)
- [Step 3: Create Network Objects \(p. 16\)](#)
- [Step 4: Create a VPN Community and Configure IKE and IPsec \(p. 17\)](#)
- [Step 5: Configure the Firewall \(p. 18\)](#)
- [Step 6: Enable Dead Peer Detection and TCP MSS Clamping \(p. 19\)](#)

## Step 1: Configure the Tunnel Interfaces

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

### To configure the tunnel interface

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
```

## Step 2: Configure BGP

In this step, you create a BGP policy that allows the import of routes that are advertised by AWS, and then configure your customer gateway to advertise its local routes to AWS.

### To create a BGP policy

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**.

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, go to the [Gaia Advanced Routing R77 Versions Administration Guide](#).

### To advertise local routes

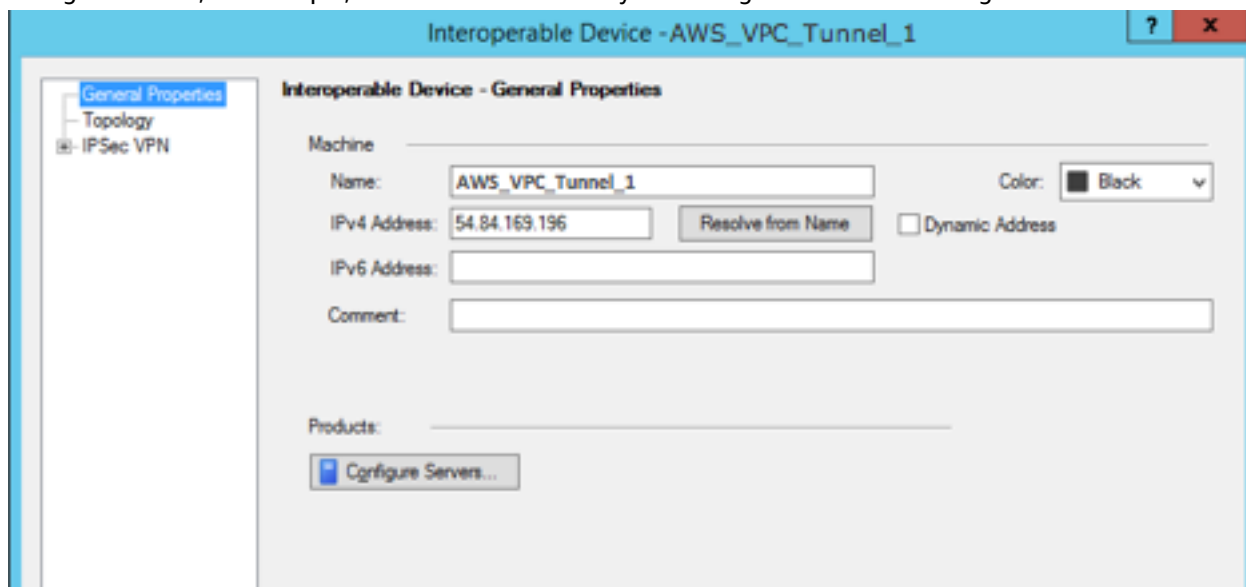
1. In the Gaia WebUI, choose **Advanced Routing, Routing Redistribution**. Choose **Add Redistribution From** and select **Interface**.
2. For **To Protocol**, select the virtual private gateway ASN; for example, 7224.
3. For **Interface**, select an internal interface. Choose **Save**.

## Step 3: Create Network Objects

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.

### To define a new network object

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 you provided for your tunnel in step 1, for example, `AWS_VPC_Tunnel_1` or `AWS_VPC_Tunnel_2`.
5. For **IPv4 Address**, enter the outside IP address of the virtual private gateway provided in the configuration file, for example, `54.84.169.196`. Save your settings and close the dialog box.



6. In the left category pane, choose **Topology**.
7. In the **VPN Domain** section, choose **Manually defined**, and 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 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, then edit the topology and define the interfaces as cluster interfaces. Use the IP addresses specified in the configuration file.

## Step 4: Create a VPN Community and Configure IKE and IPsec

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.

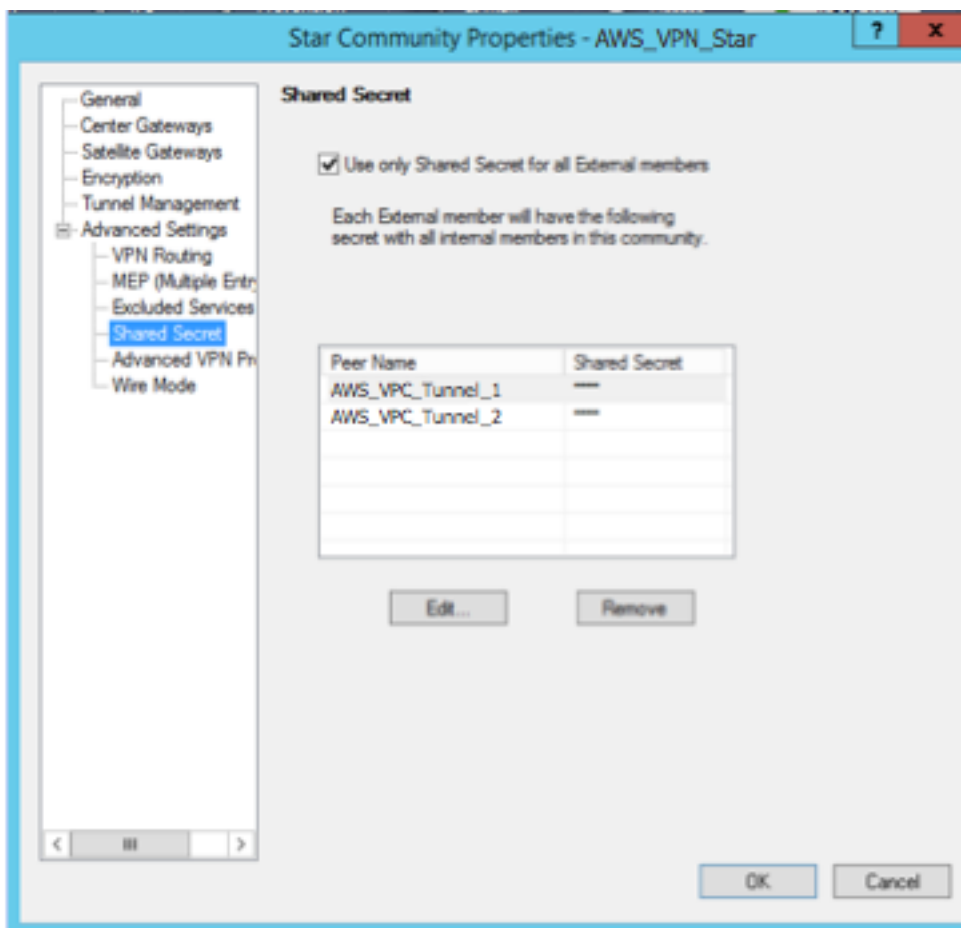
### To create and configure the VPN community, IKE, and IPsec settings

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 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; however, IKEv2 and IPv6 are currently not supported.

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:** SHA1
  - 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 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 enter the pre-shared key as specified in the configuration file in the `IPSec Tunnel #2` section.



12. Still in the **Advanced Settings** category, choose **Advanced VPN Properties**, configure the properties as follows, and 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

## Step 5: Configure the Firewall

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.

### To create firewall rules

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 choose **OK** when you're done:
    - `internal_clear` > VPN community (The VPN star community 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.

## Step 6: Enable Dead Peer Detection and TCP MSS Clamping

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 in [Step 4: Create a VPN Community and Configure IKE and IPsec \(p. 17\)](#)).

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.

### To modify the `tunnel_keepalive_method` property

1. Open the Check Point SmartDashboard, and choose **Security Management Server, Domain Management Server**.
2. Choose **File, Database Revision Control...** and create a revision snapshot.
3. Close all SmartConsole windows, such as the SmartDashboard, SmartView Tracker, and SmartView Monitor.
4. Start the GuiDBedit tool. For more information, see the [Check Point Database Tool](#) article on the Check Point Support Center.
5. Choose **Security Management Server, Domain Management Server**.
6. In the upper left pane, choose **Table, Network Objects, network\_objects**.
7. In the upper right pane, select the relevant **Security Gateway, Cluster** object.
8. Press CTRL+F, or use the **Search** menu to search for the following: `tunnel_keepalive_method`.
9. In the lower pane, open the context menu for `tunnel_keepalive_method`, and select **Edit...**. Choose **dpd** and choose **OK**.
10. Repeat steps 7 - 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.

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

#### To enable TCP MSS clamping

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**.

## How to Test the Customer Gateway Configuration

You can test the gateway configuration for each tunnel.

#### To test the customer gateway configuration for each tunnel

1. On your customer gateway, determine whether the BGP status is `Active`.  
  
It takes approximately 30 seconds for a BGP peering to become active.
2. Ensure that the customer gateway is advertising a route to the virtual private gateway. The route may be the default route (`0.0.0.0/0`) or a more specific route you prefer.

When properly established, your BGP peering should be receiving one route from the virtual private gateway corresponding to the prefix that your VPC integration team specified for the VPC (for example, `10.0.0.0/24`). 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.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection: your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

#### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the Amazon EC2 Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, `10.0.0.4`). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

**Note**

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMLs don't respond to ping messages from tunnel IP addresses.



4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.





On the Check Point gateway side, 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-B003989E658CF04}
Community	AWS_VPN_Star
Encryption Scheme	 IKE
Data Encryption Methods	ESP: AES-128 + SHA1 + PF (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

# Example: Check Point Device without Border Gateway Protocol

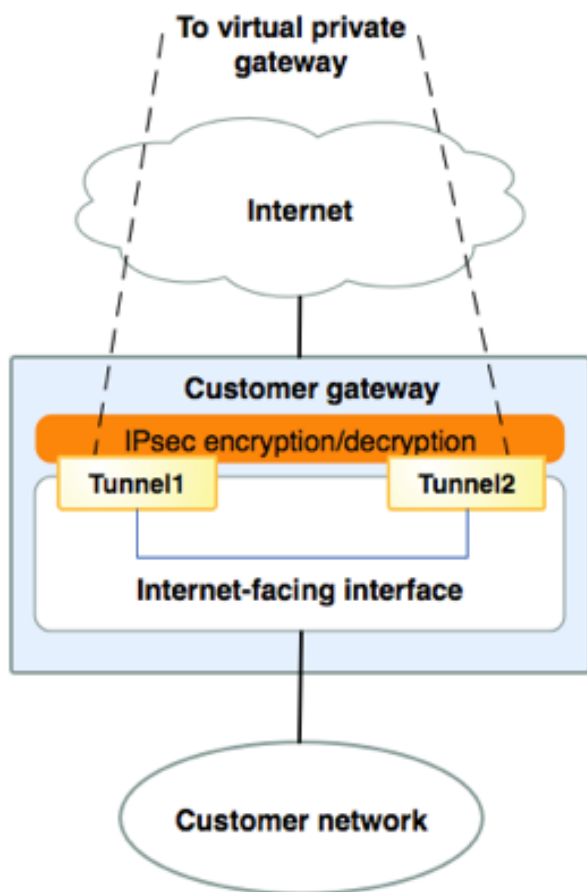
This section has example configuration information provided by your integration team if your customer gateway is a Check Point Security Gateway device running R77.10 or above, and using the Gaia operating system.

## Topics

- [High-Level View of the Customer Gateway \(p. 23\)](#)
- [Configuration File \(p. 24\)](#)
- [Configuring the Check Point Device \(p. 25\)](#)
- [How to Test the Customer Gateway Configuration \(p. 32\)](#)

## High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels. Using redundant tunnels ensures continuous availability in the case that a device fails.



## Configuration File

Your integration team will provide you with a configuration file with the values you need in order to configure each tunnel and the IKE and IPsec settings for your VPN device. The configuration file includes instructions on how to use the Gaia web portal and Check Point SmartDashboard to configure your device. The same steps are provided in the next section.

The following is an extract of an example configuration file. The file contains two sections: IPsec Tunnel #1 and IPsec Tunnel #2. You must use the values provided in each section to configure each tunnel.

```
! Amazon Web Services
! Virtual Private Cloud

! AWS uses unique identifiers to manipulate the configuration of
! a VPN connection. Each VPN connection is assigned an identifier and is
! associated with two other identifiers, namely the
! customer gateway identifier and virtual private gateway identifier.
!
! Your VPN connection ID      : vpn-12345678
! Your virtual private gateway ID : vgw-12345678
! Your customer gateway ID    : cgw-12345678
!
!
! This configuration consists of two tunnels. Both tunnels must be
```

```
! configured on your customer gateway.
!

! -----
! IPsec Tunnel #1
! -----
! #1: Tunnel Interface Configuration
...

! -----
! -----
! IPsec Tunnel #2
! -----
! #1: Tunnel Interface Configuration
...

```

## Configuring the Check Point Device

The following procedures demonstrate how to configure the VPN tunnels, network objects, and security for your VPN connection. You must replace the example values in the procedures with the values that are provided in the configuration file.

### Note

For more information, go to the [Check Point Security Gateway IPsec VPN to Amazon Web Services VPC](#) article on the Check Point Support Center.

### Topics

- [Step 1: Configure Tunnel Interface \(p. 25\)](#)
- [Step 2: Configure the Static Route \(p. 26\)](#)
- [Step 3: Create Network Objects \(p. 28\)](#)
- [Step 4: Create a VPN Community and Configure IKE and IPsec \(p. 29\)](#)
- [Step 5: Configure the Firewall \(p. 30\)](#)
- [Step 6: Enable Dead Peer Detection and TCP MSS Clamping \(p. 31\)](#)

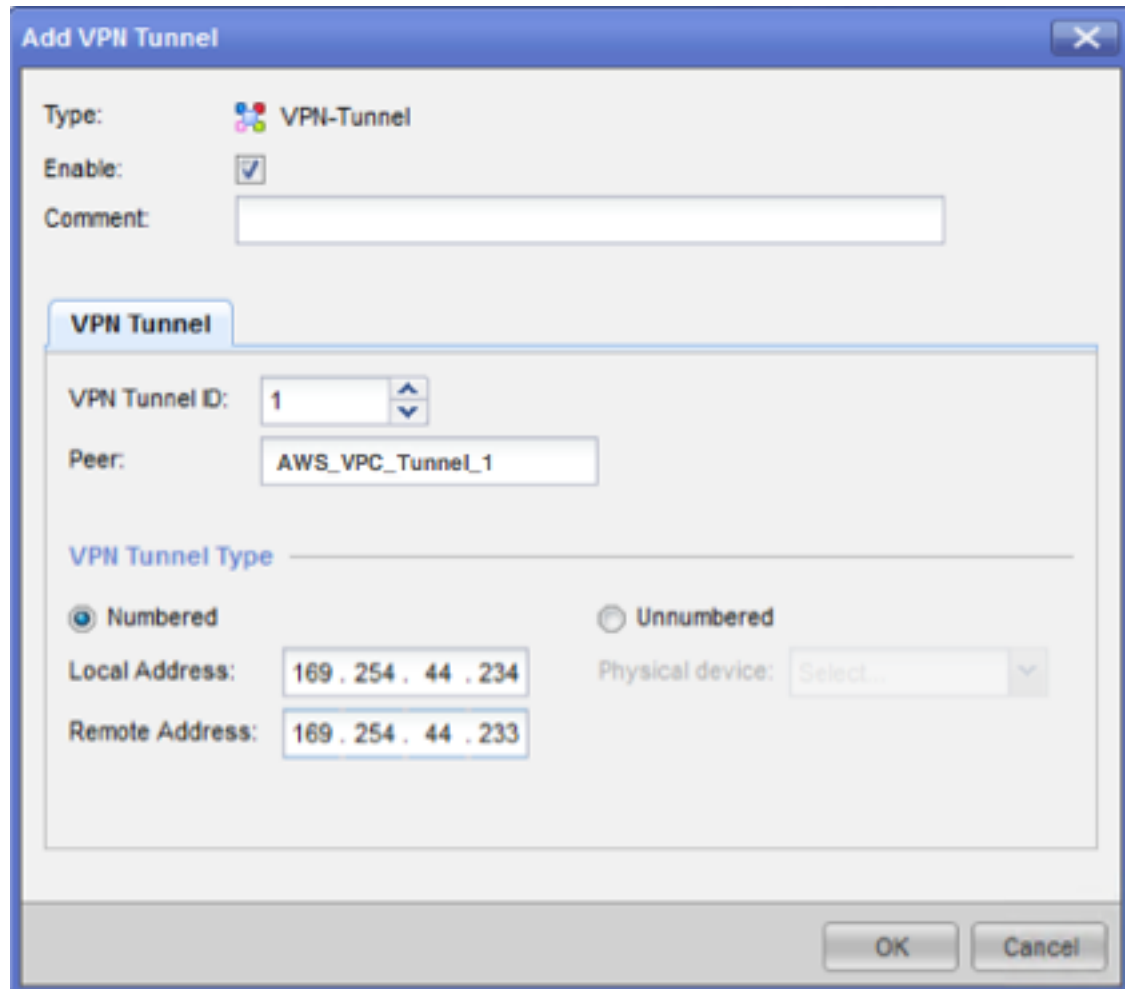
## Step 1: Configure Tunnel Interface

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.

### To configure the tunnel interface

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

- For **Remote Address**, enter the IP address specified for VGW Tunnel IP in the configuration file, for example, 169.254.44.233.



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

```
set interface vpnt1 mtu 1436
```

For tunnel 2, run the following command:

```
set interface vpnt2 mtu 1436
```

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

## Step 2: Configure the Static Route

In this step, you'll 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.

### To configure the static routes

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: ☒

Add Gateway Edit Delete

Gateway	Priority
169.254.44.233	1
169.254.44.5	2

Save Cancel

7. Choose **Save**.



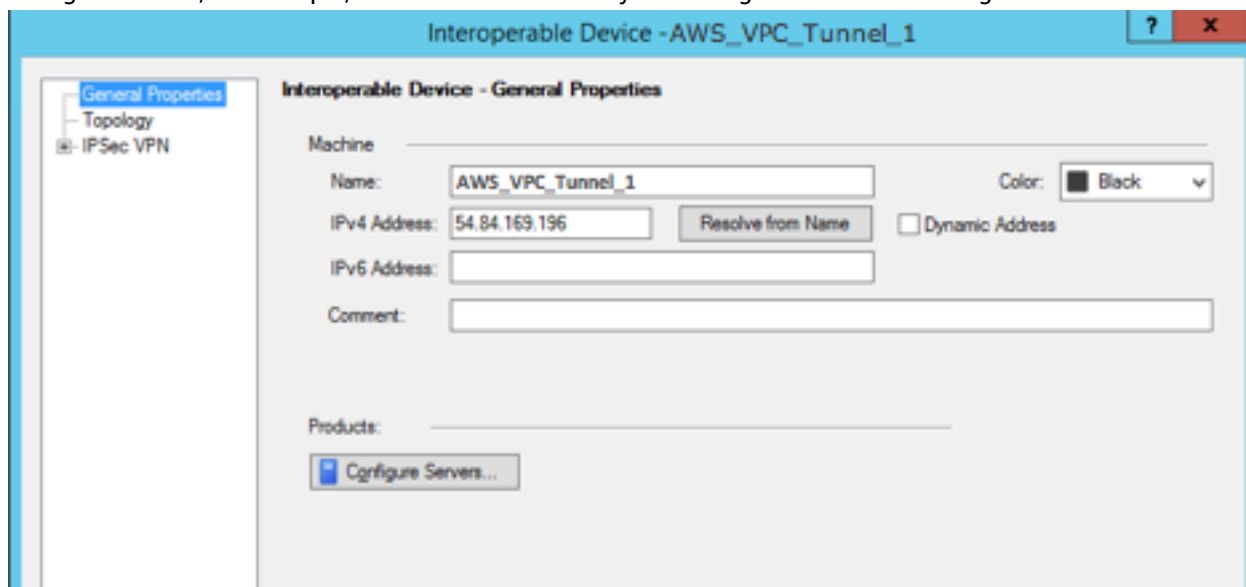
If you're using a cluster, repeat the steps above for the other members of the cluster.

## Step 3: Create Network Objects

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.

### To define a new network object

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 you provided for your tunnel, for example, `AWS_VPC_Tunnel_1` or `AWS_VPC_Tunnel_2`.
5. For **IPv4 Address**, enter the outside IP address of the virtual private gateway provided in the configuration file, for example, `54.84.169.196`. Save your settings and close the dialog box.



6. In the SmartDashboard, open your gateway properties and in the category pane, choose **Topology**.
7. To retrieve the interface configuration, choose **Get Topology**.
8. In the **VPN Domain** section, choose **Manually defined**, and 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

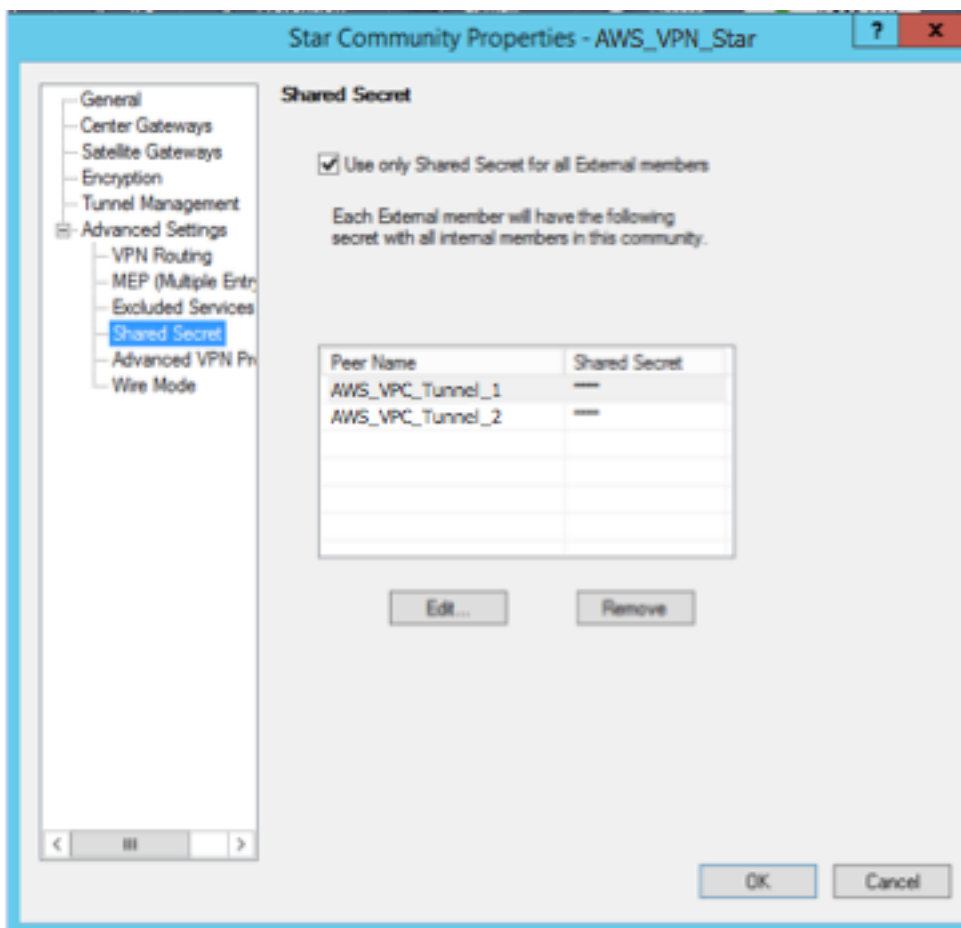
If you're using clusters, then edit the topology and define the interfaces as cluster interfaces. Use the IP addresses specified in the configuration file.

## Step 4: Create a VPN Community and Configure IKE and IPsec

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.

### To create and configure the VPN community, 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 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:** SHA1
  - 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 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 enter the pre-shared key as specified in the configuration file in the `IPSec Tunnel #2` section.



12. Still in the **Advanced Settings** category, choose **Advanced VPN Properties**, configure the properties as follows, and 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

## Step 5: Configure the Firewall

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.

### To create firewall rules

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 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.

## Step 6: Enable Dead Peer Detection and TCP MSS Clamping

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 in [Step 4: Create a VPN Community and Configure IKE and IPsec \(p. 29\)](#)).

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.

### To modify the `tunnel_keepalive_method` property

1. Open the Check Point SmartDashboard, and choose **Security Management Server, Domain Management Server**.
2. Choose **File, Database Revision Control...** and create a revision snapshot.
3. Close all SmartConsole windows, such as the SmartDashboard, SmartView Tracker, and SmartView Monitor.
4. Start the GuiDBedit tool. For more information, see the [Check Point Database Tool](#) article on the Check Point Support Center.
5. Choose **Security Management Server, Domain Management Server**.
6. In the upper left pane, choose **Table, Network Objects, network\_objects**.
7. In the upper right pane, select the relevant **Security Gateway, Cluster** object.
8. Press CTRL+F, or use the **Search** menu to search for the following: `tunnel_keepalive_method`.
9. In the lower pane, open the context menu for `tunnel_keepalive_method`, and select **Edit....** Choose **dpd** and choose **OK**.
10. Repeat steps 7 - 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.

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

#### To enable TCP MSS clamping

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**.

## How to Test the Customer Gateway Configuration

You can test the gateway configuration for each tunnel.

#### To test the customer gateway configuration for each tunnel

1. Ensure that the customer gateway has a static route to your VPC, as suggested in the configuration templates provided by AWS.
2. Ensure that a static route has been added to the VPN connection so that traffic can get back to your customer gateway. For example, if your local subnet prefix is `198.10.0.0/16`, you need to add a static route with that CIDR range to your VPN connection. Make sure that both tunnels have a static route to your VPC.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection - your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

#### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the AWS Management Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, `10.0.0.4`). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

**Note**

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.



4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.





On the Check Point gateway side, 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-B003989E658CF04}
Community	AWS_VPN_Star
Encryption Scheme	 IKE
Data Encryption Methods	ESP: AES-128 + SHA1 + PF (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

## Example: Cisco ASA Device

### Topics

- [A High-Level View of the Customer Gateway \(p. 35\)](#)
- [An Example Configuration \(p. 36\)](#)
- [How to Test the Customer Gateway Configuration \(p. 40\)](#)

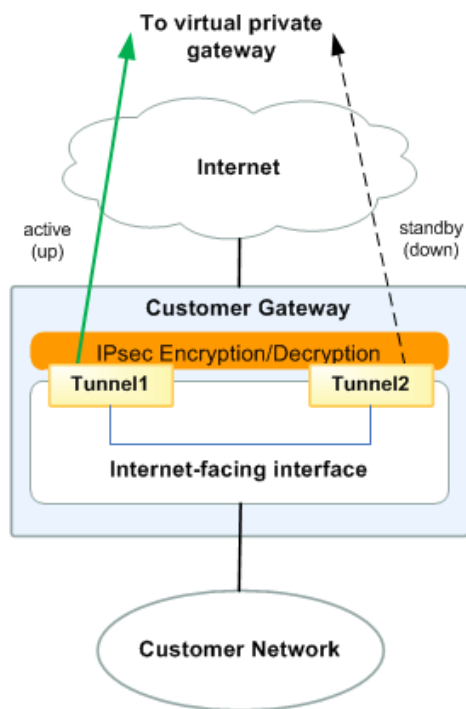
In this section we walk you through an example of the configuration information provided by your integration team if your customer gateway is a Cisco ASA device running Cisco ASA 8.2+ software.

The diagram shows the high-level layout of the customer gateway. You should use the real configuration information that you receive from your integration team and apply it to your customer gateway.

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels. Using redundant tunnels ensures continuous availability in the case that a device fails.





Please note that 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.

## An Example Configuration

The configuration in this section is an example of the configuration information your integration team should provide. The example configuration contains a set of information for each of the tunnels that you must configure.

The example configuration includes example values to help you understand how configuration works. For example, we provide example values for the VPN connection ID (vpn-12345678) and virtual private gateway ID (vgw-12345678), and placeholders for the AWS endpoints ([\*AWS\\_ENDPOINT\\_1\*](#) and [\*AWS\\_ENDPOINT\\_2\*](#)). You'll replace these example values with the actual values from the configuration information that you receive.

In addition, you must:

- 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 configured on the device.
- Ensure that the SLA monitoring number is unique.

- Configure all internal routing that moves traffic between the customer gateway and your local network.

### Important

The following configuration information is an example of what you can expect your integration team to provide. Many of the values in the following example will be different from the actual configuration information that you receive. You must use the actual values and not the example values shown here, or your implementation will fail.

```
! Amazon Web Services
! Virtual Private Cloud
!
! AWS utilizes unique identifiers to manipulate the configuration of
! a VPN Connection. Each VPN Connection is assigned an identifier and is
! associated with two other identifiers, namely the
! Customer Gateway Identifier and Virtual Private Gateway Identifier.
!
! Your VPN Connection ID           : vpn-12345678
! Your Virtual Private Gateway ID  : vgw-12345678
! Your Customer Gateway ID        : cgw-12345678
!
! This configuration consists of two tunnels. Both tunnels must be
! configured on your Customer Gateway. Only a single tunnel will be up at a
! time to the VGW.
!
! You may need to populate these values throughout the config based on your setup:
! outside_interface - External interface of the ASA
! outside_access_in - Inbound ACL on the external interface
! amzn_vpn_map      - Outside crypto map
! vpc_subnet and vpc_subnet_mask - VPC address range
! local_subnet and local_subnet_mask - Local subnet address range
! sla_monitor_address - Target address that is part of acl-amzn to run SLA monitoring
!
! -----
! IPSec Tunnels
! -----
! #1: Internet Key Exchange (IKE) Configuration
!
! A policy is established for the supported ISAKMP encryption,
! authentication, Diffie-Hellman, lifetime, and key parameters.
!
! Note that there are a global list of ISAKMP policies, each identified by
! sequence number. This policy is defined as #201, which may conflict with
! an existing policy using the same or lower number depending on
! the encryption type. If so, we recommend changing the sequence number to
! avoid conflicts and overlap.
!
! Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
! and DH Group 2.
! You will need to modify these sample configuration files to take advantage of AES256,
! SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
! The address of the external interface for your customer gateway must be a static
! address.
! Your customer gateway may reside behind a device performing network address translation
! (NAT).
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
! unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
crypto isakmp identity address
crypto isakmp enable outside_interface
crypto isakmp policy 201
    encryption aes
    authentication pre-share
```

```
group 2
lifetime 28800
hash sha
exit
!
! The tunnel group sets the Pre Shared Key used to authenticate the
! tunnel endpoints.
!
tunnel-group AWS_ENDPOINT_1 type ipsec-l2l
tunnel-group AWS_ENDPOINT_1 ipsec-attributes
    pre-shared-key password_here
!
! This option enables IPSec Dead Peer Detection, which causes periodic
! messages to be sent to ensure a Security Association remains operational.
!
    isakmp keepalive threshold 10 retry 10
exit
!
tunnel-group AWS_ENDPOINT_2 type ipsec-l2l
tunnel-group AWS_ENDPOINT_2 ipsec-attributes
    pre-shared-key password_here
!
! This option enables IPSec Dead Peer Detection, which causes periodic
! messages to be sent to ensure a Security Association remains operational.
!
    isakmp keepalive threshold 10 retry 10
exit

! -----
! #2: Access List Configuration
!
! Access lists are configured to permit creation of tunnels and to send applicable traffic
! over them.
! This policy may need to be applied to an inbound ACL on the outside interface that is
! used to manage control-plane traffic.
! This is to allow VPN traffic into the device from the Amazon endpoints.
!
access-list outside_access_in extended permit ip host AWS_ENDPOINT_1
    host YOUR_UPLINK_ADDRESS
access-list outside_access_in extended permit ip host AWS_ENDPOINT_2
    host YOUR_UPLINK_ADDRESS
!
! The following access list named acl-amzn specifies all traffic that needs to be routed to
! the VPC. Traffic will
! be encrypted and transmitted through the tunnel to the VPC. Association with the IPSec
! security association
! is done through the "crypto map" command.
!
! This access list should contain a static route corresponding to your VPC CIDR and allow
! traffic from any subnet.
! If you do not wish to use the "any" source, you must use a single access-list entry for
! accessing the VPC range.
! If you specify more than one entry for this ACL without using "any" as the source, the
! VPN will function erratically.
! The any rule is also used so the security association will include the ASA outside
! interface where the SLA monitor
! traffic will be sourced from.
! See section #4 regarding how to restrict the traffic going over the tunnel
!
!
access-list acl-amzn extended permit ip any vpc_subnet vpc_subnet_mask

! -----
! #3: IPSec Configuration
!
! The IPSec transform set defines the encryption, authentication, and IPSec
```

```
! mode parameters.
! Please note, you may use these additionally supported IPSec parameters for encryption
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
!
crypto ipsec ikev1 transform-set transform-amzn esp-aes esp-sha-hmac

! The crypto map references the IPSec transform set and further defines
! the Diffie-Hellman group and security association lifetime. The mapping is created
! as #1, which may conflict with an existing crypto map using the same
! number. If so, we recommend changing the mapping number to avoid conflicts.
!
crypto map amzn_vpn_map 1 match address acl-amzn
crypto map amzn_vpn_map 1 set pfs group2
crypto map amzn_vpn_map 1 set peer AWS_ENDPOINT_1 AWS_ENDPOINT_2
crypto map amzn_vpn_map 1 set transform-set transform-amzn
crypto map amzn_vpn_map 1 set security-association lifetime seconds 3600
!
! Only set this if you do not already have an outside crypto map, and it is not applied:
!
crypto map amzn_vpn_map interface outside_interface
!
! Additional parameters of the IPSec configuration are set here. Note that
! these parameters are global and therefore impact other IPSec
! associations.
!
! This option instructs the firewall to clear the "Don't Fragment"
! bit from packets that carry this bit and yet must be fragmented, enabling
! them to be fragmented.
!
crypto ipsec df-bit clear-df outside_interface
!
! This configures the gateway's window for accepting out of order
! IPSec packets. A larger window can be helpful if too many packets
! are dropped due to reordering while in transit between gateways.
!
crypto ipsec security-association replay window-size 128
!
! This option instructs the firewall to fragment the unencrypted packets
! (prior to encryption).
!
crypto ipsec fragmentation before-encryption outside_interface
!
! This option causes the firewall to reduce the Maximum Segment Size of
! TCP packets to prevent packet fragmentation.
sysopt connection tcpmss 1379
!
! In order to keep the tunnel in an active or always up state, the ASA needs to send
! traffic to the subnet
! defined in acl-amzn. SLA monitoring can be configured to send pings to a destination in
! the subnet and
! will keep the tunnel active. This traffic needs to be sent to a target that will return a
! response.
! This can be manually tested by sending a ping to the target from the ASA sourced from the
! outside interface.
! A possible destination for the ping is an instance within the VPC. For redundancy
! multiple SLA monitors
! can be configured to several instances to protect against a single point of failure.
!
! The monitor is created as #1, which may conflict with an existing monitor using the same
! number. If so, we recommend changing the sequence number to avoid conflicts.
!
sla monitor 1
    type echo protocol ipIcmpEcho sla_monitor_address interface outside_interface
    frequency 5
exit
sla monitor schedule 1 life forever start-time now
```

```
!
! The firewall must allow icmp packets to use "sla monitor"
icmp permit any outside_interface

!-----
! #4: VPN Filter
! The VPN Filter will restrict traffic that is permitted through the tunnels. By default
! all traffic is denied.
! The first entry provides an example to include traffic between your VPC Address space and
! your office.
! You may need to run 'clear crypto isakmp sa', in order for the filter to take effect.
!
! access-list amzn-filter extended permit
! ip vpc_subnet vpc_subnet_mask local_subnet local_subnet_mask
access-list amzn-filter extended deny ip any any
group-policy filter internal
group-policy filter attributes
vpn-filter value amzn-filter
tunnel-group AWS_ENDPOINT_1 general-attributes
default-group-policy filter
exit
tunnel-group AWS_ENDPOINT_2 general-attributes
default-group-policy filter
exit

!-----
! #5: NAT Exemption
! If you are performing NAT on the ASA you will have to add a nat exemption rule.
! This varies depending on how NAT is set up. It should be configured along the lines of:
! object network obj-SrcNet
!   subnet 0.0.0.0 0.0.0.0
! object network obj-amzn
!   subnet vpc_subnet vpc_subnet_mask
! nat (inside,outside) 1 source static obj-SrcNet obj-SrcNet destination static obj-amzn
! obj-amzn
! If using version 8.2 or older, the entry would need to look something like this:
! nat (inside) 0 access-list acl-amzn
! Or, the same rule in acl-amzn should be included in an existing no nat ACL.
```

## How to Test the Customer Gateway Configuration

When using Cisco ASA as a customer gateway, only one tunnel will be in the UP state. The second tunnel should be configured, but will only be used if the first tunnel goes down. The second tunnel cannot be in the UP state when the first tunnel is in the UP state. Your console will display that only one tunnel is up and it will show the second tunnel as down. This is expected behavior for Cisco ASA customer gateway tunnels because ASA as a customer gateway only supports a single tunnel being up at one time.

You can test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

- Ensure that a static route has been added to the VPN connection so that traffic can get back to your customer gateway. For example, if your local subnet prefix is 198.10.0.0/16, you need to add a static route with that CIDR range to your VPN connection. Make sure that both tunnels have a static route to your VPC.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection - your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the AWS Management Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

#### Note

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.

If your tunnels do not test successfully, see [Troubleshooting Cisco ASA Customer Gateway Connectivity \(p. 159\)](#).

# Example: Cisco IOS Device

## Topics

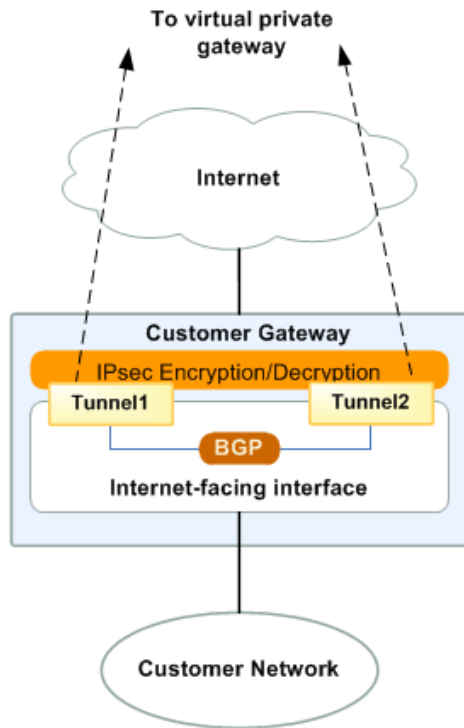
- [A High-Level View of the Customer Gateway \(p. 43\)](#)
- [A Detailed View of the Customer Gateway and an Example Configuration \(p. 44\)](#)
- [How to Test the Customer Gateway Configuration \(p. 50\)](#)

In this section we walk you through an example of the configuration information provided by your integration team if your customer gateway is a Cisco IOS device running Cisco IOS 12.4 (or later) software.

Two diagrams illustrate the example configuration. The first diagram shows the high-level layout of the customer gateway, and the second diagram shows details from the example configuration. You should use the real configuration information that you receive from your integration team and apply it to your customer gateway.

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels. Using redundant tunnels ensures continuous availability in the case that a device fails.





## A Detailed View of the Customer Gateway and an Example Configuration

The diagram in this section illustrates an example Cisco IOS customer gateway. Following the diagram is a corresponding example of the configuration information your integration team should provide. The example configuration contains a set of information for each of the tunnels that you must configure.

In addition, the example configuration refers to these items that you must provide:

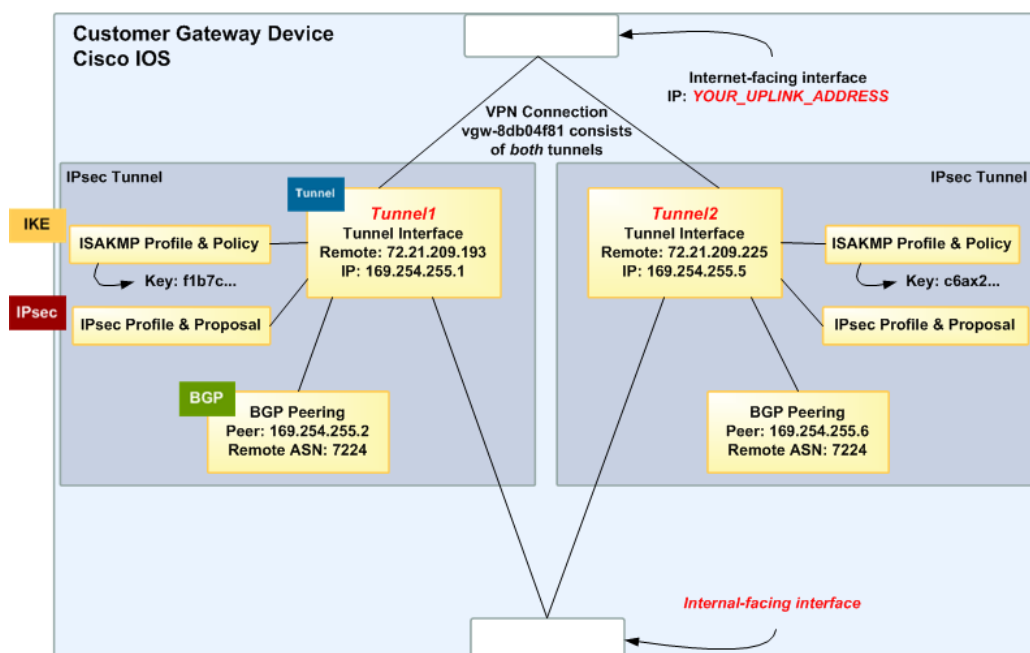
- ***YOUR\_UPLINK\_ADDRESS***—The IP address for the Internet-routable external interface on the customer gateway. The address must be static, and may be behind a device performing network address translation (NAT). To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500.
- ***YOUR\_BGP\_ASN***—The customer gateway's BGP ASN (we use 65000 by default)

The example configuration includes several example values to help you understand how configuration works. For example, we provide example values for the VPN connection ID (vpn-44a8938f), virtual private gateway ID (vgw-8db04f81), the IP addresses (72.21.209.\*, 169.254.255.\*), and the remote ASN (7224). You'll replace these example values with the actual values from the configuration information that you receive.

In addition, you must:

- Configure the outside interface
- Configure the tunnel interface IDs (referred to as ***Tunnel1*** and ***Tunnel2*** in the example configuration).
- Ensure that the Crypto ISAKMP 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 configured on the device.
- Configure all internal routing that moves traffic between the customer gateway and your local network.

In the following diagram and example configuration, you must replace the items in red italics with values that apply to your particular configuration.



### Warning

The following configuration information is an example of what you can expect your integration team to provide. Many of the values in the following example will be different from the actual configuration information that you receive. You must use the actual values and not the example values shown here, or your implementation will fail.

```
! Amazon Web Services
! Virtual Private Cloud

! AWS utilizes unique identifiers to manipulate the configuration of
! a VPN Connection. Each VPN Connection is assigned an identifier
! and is associated with two other identifiers, namely the
! Customer Gateway Identifier and Virtual Private Gateway Identifier.
!
! Your VPN Connection ID           : vpn-44a8938f
! Your Virtual Private Gateway ID  : vgw-8db04f81
! Your Customer Gateway ID         : cgw-b4dc3961
!
!
! This configuration consists of two tunnels. Both tunnels must be
! configured on your Customer Gateway.
!
! -----
! IPsec Tunnel #1
! -----

IKE

! #1: Internet Key Exchange (IKE) Configuration
!
! A policy is established for the supported ISAKMP encryption,
! authentication, Diffie-Hellman, lifetime, and key parameters.
!
! Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
! and DH Group 2.
! You will need to modify these sample configuration files to take advantage of AES256,
! SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
```

```
! The address of the external interface for your customer gateway must be a static
address.
! Your customer gateway may reside behind a device performing network address translation
(NAT).
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
! Note that there are a global list of ISAKMP policies, each identified by
! sequence number. This policy is defined as #200, which may conflict with
! an existing policy using the same number. If so, we recommend changing
! the sequence number to avoid conflicts.
!
crypto isakmp policy 200
  encryption aes 128
  authentication pre-share
  group 2
  lifetime 28800
  hash sha
exit

! The ISAKMP keyring stores the Pre Shared Key used to authenticate the
! tunnel endpoints.
!
crypto keyring keyring-vpn-44a8938f-0
  local-address YOUR_UPLINK_ADDRESS
  pre-shared-key address 72.21.209.225 key plain-text-password1
exit

! An ISAKMP profile is used to associate the keyring with the particular
! endpoint.
!
crypto isakmp profile isakmp-vpn-44a8938f-0
  local-address YOUR_UPLINK_ADDRESS
  match identity address 72.21.209.225
  keyring keyring-vpn-44a8938f-0
exit
```

**IPsec**

```
! #2: IPsec Configuration
!
! The IPsec transform set defines the encryption, authentication, and IPsec
! mode parameters.
! Please note, you may use these additionally supported IPsec parameters for encryption
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
!
crypto ipsec transform-set ipsec-prop-vpn-44a8938f-0 esp-aes 128 esp-sha-hmac
  mode tunnel
exit

! The IPsec profile references the IPsec transform set and further defines
! the Diffie-Hellman group and security association lifetime.
!
crypto ipsec profile ipsec-vpn-44a8938f-0
  set pfs group2
  set security-association lifetime seconds 3600
  set transform-set ipsec-prop-vpn-44a8938f-0
exit

! Additional parameters of the IPsec configuration are set here. Note that
! these parameters are global and therefore impact other IPsec
! associations.
! This option instructs the router to clear the "Don't Fragment"
! bit from packets that carry this bit and yet must be fragmented, enabling
```

```
! them to be fragmented.
!
crypto ipsec df-bit clear

! This option enables IPsec Dead Peer Detection, which causes periodic
! messages to be sent to ensure a Security Association remains operational.
!
crypto isakmp keepalive 10 10 on-demand

! This configures the gateway's window for accepting out of order
! IPsec packets. A larger window can be helpful if too many packets
! are dropped due to reordering while in transit between gateways.
!
crypto ipsec security-association replay window-size 128

! This option instructs the router to fragment the unencrypted packets
! (prior to encryption).
!
crypto ipsec fragmentation before-encryption
```

### Tunnel

```
! #3: Tunnel Interface Configuration
!
! A tunnel interface is configured to be the logical interface associated
! with the tunnel. All traffic routed to the tunnel interface will be
! encrypted and transmitted to the VPC. Similarly, traffic from the VPC
! will be logically received on this interface.
!
! Association with the IPsec security association is done through the
! "tunnel protection" command.
!
! The address of the interface is configured with the setup for your
! Customer Gateway. If the address changes, the Customer Gateway and VPN
! Connection must be recreated with Amazon VPC.
!
interface Tunnel1
  ip address 169.254.255.2 255.255.255.252
  ip virtual-reassembly
  tunnel source YOUR_UPLINK_ADDRESS
  tunnel destination 72.21.209.225
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile ipsec-vpn-44a8938f-0
  ! This option causes the router to reduce the Maximum Segment Size of
  ! TCP packets to prevent packet fragmentation.
  ip tcp adjust-mss 1387
  no shutdown
exit
```

### BGP

```
! #4: Border Gateway Protocol (BGP) Configuration
!
! BGP is used within the tunnel to exchange prefixes between the
! Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway
! will announce the prefix corresponding to your VPC.
!
! Your Customer Gateway may announce a default route (0.0.0.0/0),
! which can be done with the 'network' statement and
! 'default-originate' statements.
!
! The BGP timers are adjusted to provide more rapid detection of outages.
!
```

```

! The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN) is configured
! as part of your Customer Gateway. If the ASN must be changed, the
! Customer Gateway and VPN Connection will need to be recreated with AWS.
!
router bgp YOUR_BGP_ASN
  neighbor 169.254.255.1 remote-as 7224
  neighbor 169.254.255.1 activate
  neighbor 169.254.255.1 timers 10 30 30
  address-family ipv4 unicast
    neighbor 169.254.255.1 remote-as 7224
    neighbor 169.254.255.1 timers 10 30 30
    neighbor 169.254.255.1 default-originate
    neighbor 169.254.255.1 activate
    neighbor 169.254.255.1 soft-reconfiguration inbound
! To advertise additional prefixes to Amazon VPC, copy the 'network' statement
! and identify the prefix you wish to advertise. Make sure the prefix is present
! in the routing table of the device with a valid next-hop.
  network 0.0.0.0
  exit
exit

! -----
! IPsec Tunnel #2
! -----

IKE

! #1: Internet Key Exchange (IKE) Configuration
!
! A policy is established for the supported ISAKMP encryption,
! authentication, Diffie-Hellman, lifetime, and key parameters.
! Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
! and DH Group 2.
! You will need to modify these sample configuration files to take advantage of AES256,
! SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
! The address of the external interface for your customer gateway must be a static
! address.
! Your customer gateway may reside behind a device performing network address translation
! (NAT).
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
! unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
! Note that there are a global list of ISAKMP policies, each identified by
! sequence number. This policy is defined as #201, which may conflict with
! an existing policy using the same number. If so, we recommend changing
! the sequence number to avoid conflicts.
!
crypto isakmp policy 201
  encryption aes 128
  authentication pre-share
  group 2
  lifetime 28800
  hash sha
exit

! The ISAKMP keyring stores the Pre Shared Key used to authenticate the
! tunnel endpoints.
!
crypto keyring keyring-vpn-44a8938f-1
  local-address YOUR_UPLINK_ADDRESS
  pre-shared-key address 72.21.209.193 key plain-text-password2
exit

! An ISAKMP profile is used to associate the keyring with the particular

```

```
! endpoint.  
!  
crypto isakmp profile isakmp-vpn-44a8938f-1  
  local-address YOUR_UPLINK_ADDRESS  
  match identity address 72.21.209.193  
  keyring keyring-vpn-44a8938f-1  
exit
```

## IPsec

```
! #2: IPsec Configuration  
!  
! The IPsec transform set defines the encryption, authentication, and IPsec  
! mode parameters.  
! Please note, you may use these additionally supported IPsec parameters for encryption  
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.  
!  
crypto ipsec transform-set ipsec-prop-vpn-44a8938f-1 esp-aes 128 esp-sha-hmac  
  mode tunnel  
exit  
  
! The IPsec profile references the IPsec transform set and further defines  
! the Diffie-Hellman group and security association lifetime.  
!  
crypto ipsec profile ipsec-vpn-44a8938f-1  
  set pfs group2  
  set security-association lifetime seconds 3600  
  set transform-set ipsec-prop-vpn-44a8938f-1  
exit  
  
! Additional parameters of the IPsec configuration are set here. Note that  
! these parameters are global and therefore impact other IPsec  
! associations.  
! This option instructs the router to clear the "Don't Fragment"  
! bit from packets that carry this bit and yet must be fragmented, enabling  
! them to be fragmented.  
!  
crypto ipsec df-bit clear  
  
! This option enables IPsec Dead Peer Detection, which causes periodic  
! messages to be sent to ensure a Security Association remains operational.  
!  
crypto isakmp keepalive 10 10 on-demand  
  
! This configures the gateway's window for accepting out of order  
! IPsec packets. A larger window can be helpful if too many packets  
! are dropped due to reordering while in transit between gateways.  
!  
crypto ipsec security-association replay window-size 128  
  
! This option instructs the router to fragment the unencrypted packets  
! (prior to encryption).  
!  
crypto ipsec fragmentation before-encryption
```

## Tunnel

```
! #3: Tunnel Interface Configuration  
!  
! A tunnel interface is configured to be the logical interface associated  
! with the tunnel. All traffic routed to the tunnel interface will be  
! encrypted and transmitted to the VPC. Similarly, traffic from the VPC  
! will be logically received on this interface.
```

```
!  
! Association with the IPsec security association is done through the  
! "tunnel protection" command.  
!  
! The address of the interface is configured with the setup for your  
! Customer Gateway. If the address changes, the Customer Gateway and VPN  
! Connection must be recreated with Amazon VPC.  
!  
interface Tunnel2  
  ip address 169.254.255.6 255.255.255.252  
  ip virtual-reassembly  
  tunnel source YOUR_UPLINK_ADDRESS  
  tunnel destination 72.21.209.193  
  tunnel mode ipsec ipv4  
  tunnel protection ipsec profile ipsec-vpn-44a8938f-1  
  ! This option causes the router to reduce the Maximum Segment Size of  
  ! TCP packets to prevent packet fragmentation.  
  ip tcp adjust-mss 1387  
  no shutdown  
exit
```

## BGP

```
! #4: Border Gateway Protocol (BGP) Configuration  
!  
! BGP is used within the tunnel to exchange prefixes between the  
! Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway  
! will announce the prefix corresponding to your Cloud.  
!  
! Your Customer Gateway may announce a default route (0.0.0.0/0),  
! which can be done with the 'network' statement and  
! 'default-originate' statements.  
!  
! The BGP timers are adjusted to provide more rapid detection of outages.  
!  
! The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN) is configured  
! as part of your Customer Gateway. If the ASN must be changed, the  
! Customer Gateway and VPN Connection will need to be recreated with AWS.  
!  
router bgp YOUR_BGP_ASN  
  neighbor 169.254.255.5 remote-as 7224  
  neighbor 169.254.255.5 activate  
  neighbor 169.254.255.5 timers 10 30 30  
  address-family ipv4 unicast  
    neighbor 169.254.255.5 remote-as 7224  
    neighbor 169.254.255.5 timers 10 30 30  
    neighbor 169.254.255.5 default-originate  
    neighbor 169.254.255.5 activate  
    neighbor 169.254.255.5 soft-reconfiguration inbound  
  ! To advertise additional prefixes to Amazon VPC, copy the 'network' statement  
  ! and identify the prefix you wish to advertise. Make sure the prefix is present  
  ! in the routing table of the device with a valid next-hop.  
    network 0.0.0.0  
  exit  
exit
```

## How to Test the Customer Gateway Configuration

You can test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

1. On your customer gateway, determine whether the BGP status is `Active`.  
It takes approximately 30 seconds for a BGP peering to become active.
2. Ensure that the customer gateway is advertising a route to the virtual private gateway. The route may be the default route (0.0.0.0/0) or a more specific route you prefer.

When properly established, your BGP peering should be receiving one route from the virtual private gateway corresponding to the prefix that your VPC integration team specified for the VPC (for example, 10.0.0.0/24). 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.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection: your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the Amazon EC2 Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

#### Note

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.



If your tunnels don't test successfully, see [Troubleshooting Cisco IOS Customer Gateway Connectivity \(p. 162\)](#).

# Example: Cisco IOS Device without Border Gateway Protocol

## Topics

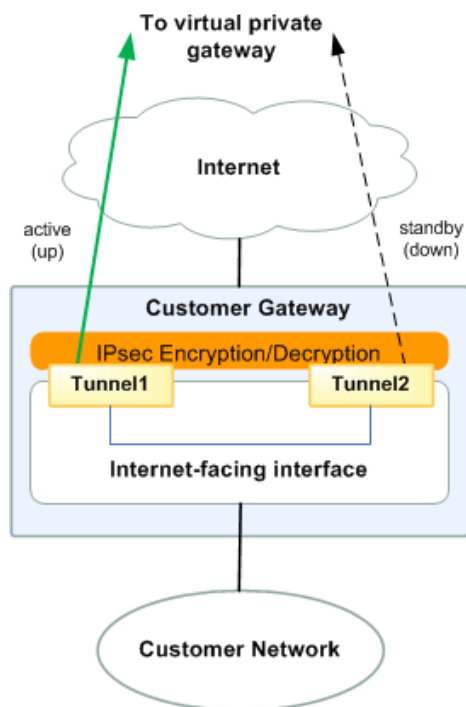
- [A High-Level View of the Customer Gateway \(p. 53\)](#)
- [A Detailed View of the Customer Gateway and an Example Configuration \(p. 54\)](#)
- [How to Test the Customer Gateway Configuration \(p. 60\)](#)

In this section we walk you through an example of the configuration information provided by your integration team if your customer gateway is a Cisco Integrated Services router running Cisco IOS software.

Two diagrams illustrate the example configuration. The first diagram shows the high-level layout of the customer gateway, and the second diagram shows details from the example configuration. You should use the real configuration information that you receive from your integration team, and apply it to your customer gateway.

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels. Using redundant tunnels ensures continuous availability in the case that a device fails.



## A Detailed View of the Customer Gateway and an Example Configuration

The diagram in this section illustrates an example Cisco IOS customer gateway (without BGP). Following the diagram, there is a corresponding example of the configuration information that your integration team should provide. The example configuration contains a set of information for each of the tunnels that you must configure.

In addition, the example configuration refers to this item that you must provide:

- ***YOUR\_UPLINK\_ADDRESS***—The IP address for the Internet-routable external interface on the customer gateway. The address must be static, and may be behind a device performing network address translation (NAT). To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500.

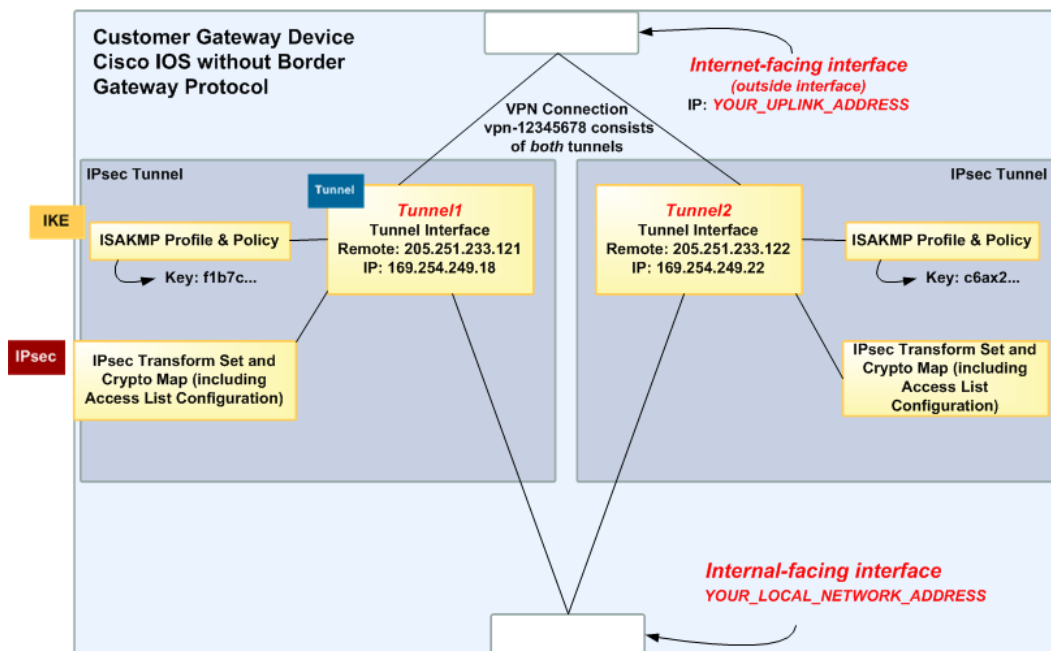
The example configuration includes several example values to help you understand how configuration works. For example, we provide example values for the VPN connection ID (vpn-1a2b3c4d), virtual private gateway ID (vgw-12345678), the IP addresses (205.251.233.\*, 169.254.255.\*). You'll replace these example values with the actual values from the configuration information that you receive.

In addition, you must:

- Configure the outside interface.
- Configure the tunnel interface IDs (referred to as *Tunnel1* and *Tunnel2* in the example configuration).

- Ensure that the Crypto ISAKMP 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 configured on the device.
- Ensure that the SLA monitoring number is unique.
- Configure all internal routing that moves traffic between the customer gateway and your local network.

In the following diagram and example configuration, you must replace the items in red italics with values that apply to your particular configuration.



### Warning

The following configuration information is an example of what you can expect your integration team to provide. Many of the values in the following example will be different from the actual configuration information that you receive. You must use the actual values and not the example values shown here, or your implementation will fail.

```
! -----
! IPsec Tunnel #1
! -----
! #1: Internet Key Exchange (IKE) Configuration
!
! A policy is established for the supported ISAKMP encryption,
! authentication, Diffie-Hellman, lifetime, and key parameters.
! Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
! and DH Group 2.
! You will need to modify these sample configuration files to take advantage of AES256,
! SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
! The address of the external interface for your customer gateway must be a static
! address.
! Your customer gateway may reside behind a device performing network address translation
! (NAT).
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
! unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
! Note that there are a global list of ISAKMP policies, each identified by
```

```
! sequence number. This policy is defined as #200, which may conflict with
! an existing policy using the same number. If so, we recommend changing
! the sequence number to avoid conflicts.
!
crypto isakmp policy 200
  encryption aes 128
  authentication pre-share
  group 2
  lifetime 28800
  hash sha
exit

! The ISAKMP keyring stores the Pre Shared Key used to authenticate the
! tunnel endpoints.
!
crypto keyring keyring-vpn-1a2b3c4d-0
  local-address CUSTOMER_IP
  pre-shared-key address 205.251.233.121 key PASSWORD
exit

! An ISAKMP profile is used to associate the keyring with the particular
! endpoint.
!
crypto isakmp profile isakmp-vpn-1a2b3c4d-0
  local-address CUSTOMER_IP
  match identity address 205.251.233.121
  keyring keyring-vpn-1a2b3c4d-0
exit

! #2: IPSec Configuration
!
! The IPSec transform set defines the encryption, authentication, and IPSec
! mode parameters.
!
! Please note, you may use these additionally supported IPSec parameters for encryption
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
!
crypto ipsec transform-set ipsec-prop-vpn-1a2b3c4d-0 esp-aes 128 esp-sha-hmac
  mode tunnel
exit

! The IPSec profile references the IPSec transform set and further defines
! the Diffie-Hellman group and security association lifetime.
!
crypto ipsec profile ipsec-vpn-1a2b3c4d-0
  set pfs group2
  set security-association lifetime seconds 3600
  set transform-set ipsec-prop-vpn-1a2b3c4d-0
exit

! Additional parameters of the IPSec configuration are set here. Note that
! these parameters are global and therefore impact other IPSec
! associations.
! This option instructs the router to clear the "Don't Fragment"
! bit from packets that carry this bit and yet must be fragmented, enabling
! them to be fragmented.
!
crypto ipsec df-bit clear

! This option enables IPSec Dead Peer Detection, which causes periodic
! messages to be sent to ensure a Security Association remains operational.
!
crypto isakmp keepalive 10 10 on-demand

! This configures the gateway's window for accepting out of order
! IPSec packets. A larger window can be helpful if too many packets
```

```
! are dropped due to reordering while in transit between gateways.
!
crypto ipsec security-association replay window-size 128

! This option instructs the router to fragment the unencrypted packets
! (prior to encryption).
!
crypto ipsec fragmentation before-encryption

! -----
! #3: Tunnel Interface Configuration
!
! A tunnel interface is configured to be the logical interface associated
! with the tunnel. All traffic routed to the tunnel interface will be
! encrypted and transmitted to the VPC. Similarly, traffic from the VPC
! will be logically received on this interface.
!
! Association with the IPSec security association is done through the
! "tunnel protection" command.
!
! The address of the interface is configured with the setup for your
! Customer Gateway. If the address changes, the Customer Gateway and VPN
! Connection must be recreated with Amazon VPC.
!
interface Tunnel1
 ip address 169.254.249.18 255.255.255.252
 ip virtual-reassembly
 tunnel source CUSTOMER_IP
 tunnel destination 205.251.233.121
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec-vpn-1a2b3c4d-0
 ! This option causes the router to reduce the Maximum Segment Size of
 ! TCP packets to prevent packet fragmentation.
 ip tcp adjust-mss 1387
 no shutdown
exit

! -----
! #4 Static Route Configuration
!
! Your Customer Gateway needs to set a static route for the prefix corresponding to your
! VPC to send traffic over the tunnel interface.
! An example for a VPC with the prefix 10.0.0.0/16 is provided below:
! ip route 10.0.0.0 255.255.0.0 Tunnel1 track 100
!
! SLA Monitor is used to provide a failover between the two tunnels. If the primary tunnel
! fails, the redundant tunnel will automatically be used
! This sla is defined as #100, which may conflict with an existing sla using same number.
! If so, we recommend changing the sequence number to avoid conflicts.
!
ip sla 100
 icmp-echo 169.254.249.17 source-interface Tunnel1
 timeout 1000
 frequency 5
exit
ip sla schedule 100 life forever start-time now
track 100 ip sla 100 reachability

! -----
! IPSec Tunnel #2
! -----
! #1: Internet Key Exchange (IKE) Configuration
!
! A policy is established for the supported ISAKMP encryption,
! authentication, Diffie-Hellman, lifetime, and key parameters.
```

```
! Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
and DH Group 2.
! You will need to modify these sample configuration files to take advantage of AES256,
SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
! The address of the external interface for your customer gateway must be a static
address.
! Your customer gateway may reside behind a device performing network address translation
(NAT).
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
! Note that there are a global list of ISAKMP policies, each identified by
! sequence number. This policy is defined as #201, which may conflict with
! an existing policy using the same number. If so, we recommend changing
! the sequence number to avoid conflicts.
!
crypto isakmp policy 201
  encryption aes 128
  authentication pre-share
  group 2
  lifetime 28800
  hash sha
exit

! The ISAKMP keyring stores the Pre Shared Key used to authenticate the
! tunnel endpoints.
!
crypto keyring keyring-vpn-1a2b3c4d-1
  local-address CUSTOMER_IP
  pre-shared-key address 205.251.233.122 key PASSWORD
exit

! An ISAKMP profile is used to associate the keyring with the particular
! endpoint.
!
crypto isakmp profile isakmp-vpn-1a2b3c4d-1
  local-address CUSTOMER_IP
  match identity address 205.251.233.122
  keyring keyring-vpn-1a2b3c4d-1
exit

! #2: IPSec Configuration
!
! The IPSec transform set defines the encryption, authentication, and IPSec
! mode parameters.
! Please note, you may use these additionally supported IPSec parameters for encryption
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
!
crypto ipsec transform-set ipsec-prop-vpn-1a2b3c4d-1 esp-aes 128 esp-sha-hmac
  mode tunnel
exit

! The IPSec profile references the IPSec transform set and further defines
! the Diffie-Hellman group and security association lifetime.
!
crypto ipsec profile ipsec-vpn-1a2b3c4d-1
  set pfs group2
  set security-association lifetime seconds 3600
  set transform-set ipsec-prop-vpn-1a2b3c4d-1
exit

! Additional parameters of the IPSec configuration are set here. Note that
! these parameters are global and therefore impact other IPSec
! associations.
! This option instructs the router to clear the "Don't Fragment"
! bit from packets that carry this bit and yet must be fragmented, enabling
```

```
! them to be fragmented.
!
crypto ipsec df-bit clear

! This option enables IPSec Dead Peer Detection, which causes periodic
! messages to be sent to ensure a Security Association remains operational.
!
crypto isakmp keepalive 10 10 on-demand

! This configures the gateway's window for accepting out of order
! IPSec packets. A larger window can be helpful if too many packets
! are dropped due to reordering while in transit between gateways.
!
crypto ipsec security-association replay window-size 128

! This option instructs the router to fragment the unencrypted packets
! (prior to encryption).
!
crypto ipsec fragmentation before-encryption

! -----
! #3: Tunnel Interface Configuration
!
! A tunnel interface is configured to be the logical interface associated
! with the tunnel. All traffic routed to the tunnel interface will be
! encrypted and transmitted to the VPC. Similarly, traffic from the VPC
! will be logically received on this interface.
!
! Association with the IPSec security association is done through the
! "tunnel protection" command.
!
! The address of the interface is configured with the setup for your
! Customer Gateway. If the address changes, the Customer Gateway and VPN
! Connection must be recreated with Amazon VPC.
!
interface Tunnel2
  ip address 169.254.249.22 255.255.255.252
  ip virtual-reassembly
  tunnel source CUSTOMER_IP
  tunnel destination 205.251.233.122
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile ipsec-vpn-1a2b3c4d-1
  ! This option causes the router to reduce the Maximum Segment Size of
  ! TCP packets to prevent packet fragmentation.
  ip tcp adjust-mss 1387
  no shutdown
exit

! -----
! #4 Static Route Configuration
!
! Your Customer Gateway needs to set a static route for the prefix corresponding to your
! VPC to send traffic over the tunnel interface.
! An example for a VPC with the prefix 10.0.0.0/16 is provided below:
! ip route 10.0.0.0 255.255.0.0 Tunnel2 track 200
!
! SLA Monitor is used to provide a failover between the two tunnels. If the primary tunnel
! fails, the redundant tunnel will automatically be used
! This sla is defined as #200, which may conflict with an existing sla using same number.
! If so, we recommend changing the sequence number to avoid conflicts.
!
ip sla 200
  icmp-echo 169.254.249.21 source-interface Tunnel2
  timeout 1000
  frequency 5
```



```
exit
ip sla schedule 200  life forever start-time now
track 200 ip sla 200 reachability
! -----
```

## How to Test the Customer Gateway Configuration

You can test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

1. Ensure that the customer gateway has a static route to your VPC, as suggested in the configuration templates provided by AWS.
2. Ensure that a static route has been added to the VPN connection so that traffic can get back to your customer gateway. For example, if your local subnet prefix is 198.10.0.0/16, you need to add a static route with that CIDR range to your VPN connection. Make sure that both tunnels have a static route to your VPC.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection - your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the AWS Management Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

**Note**

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.

If your tunnels don't test successfully, see [Troubleshooting Cisco IOS Customer Gateway without Border Gateway Protocol Connectivity \(p. 166\)](#).

## Example: Dell SonicWALL Device

This topic provides an example of how to configure your router if your customer gateway is a Dell SonicWALL router.

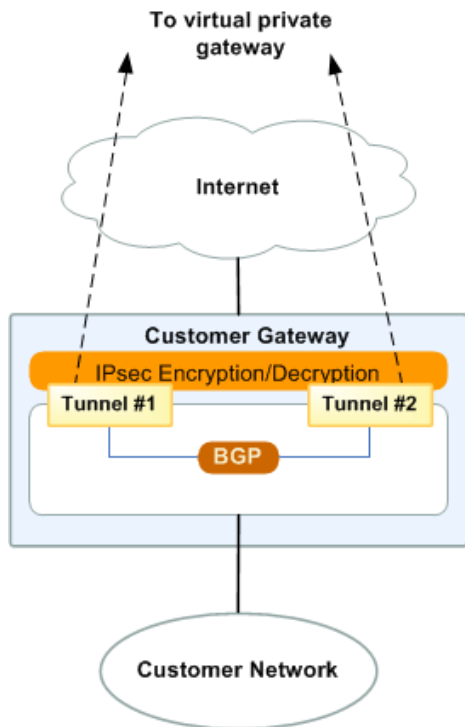
This section assumes that a VPN connection with static routing has been configured in the Amazon VPC console. For more information, see [Adding a Virtual Private Gateway to Your VPC](#) in the *Amazon VPC User Guide*.

### Topics

- [A High-Level View of the Customer Gateway](#) (p. 62)
- [Example Configuration File](#) (p. 63)
- [Configuring the SonicWALL Device Using the Management Interface](#) (p. 67)
- [How to Test the Customer Gateway Configuration](#) (p. 67)

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels: *Tunnel 1* and *Tunnel 2*. Using redundant tunnels ensures continuous availability in the case that a device fails.



## Example Configuration File

The configuration file that you download from the Amazon VPC console includes the values that you need in order to use the command line tools on OS 6.2 to configure each tunnel and the IKE and IPsec settings for your SonicWALL device.

### Important

The following configuration information uses example values — you must use the actual values and not the example values shown here, or your implementation will fail.

```
! Amazon Web Services
! Virtual Private Cloud
!
! VPN Connection Configuration
! =====
! AWS utilizes unique identifiers to manipulate the configuration of
! a VPN Connection. Each VPN Connection is assigned a VPN Connection Identifier
! and is associated with two other identifiers, namely the
! Customer Gateway Identifier and the Virtual Private Gateway Identifier.
!
! Your VPN Connection ID           : vpn-44a8938f
! Your Virtual Private Gateway ID  : vgw-8db04f81
! Your Customer Gateway ID         : cgw-ff628496
!
! This configuration consists of two tunnels. Both tunnels must be
! configured on your Customer Gateway.
!
! This configuration was tested on a SonicWALL TZ 600 running OS 6.2.5.1-26n
!
! You may need to populate these values throughout the config based on your setup:
! <vpc_subnet> - VPC address range
!
```

```
! IPsec Tunnel !1
! =====
```

## IKE

```
! #1: Internet Key Exchange (IKE) Configuration
!
! Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
! and DH Group 2.
! You can modify these sample configuration files to use AES128, SHA1, AES256, SHA256, or
! other DH groups like 2, 14-18, 22, 23, and 24.
! The address of the external interface for your customer gateway must be a static
! address.
! Your customer gateway may reside behind a device performing network address translation
! (NAT).
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
! unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
config
address-object ipv4 AWSVPC network 172.30.0.0/16
vpn policy tunnel-interface vpn-44a8938f-1
gateway primary 72.21.209.193
bound-to interface X1
auth-method shared-secret
shared-secret PRE-SHARED-KEY-IN-PLAIN-TEXT
ike-id local ip your_customer_gateway_IP_address
ike-id peer ip 72.21.209.193
end
!
```

## IPsec

```
! #2: IPsec Configuration
!
! The IPsec (Phase 2) proposal defines the protocol, authentication,
! encryption, and lifetime parameters for our IPsec security association.
! Please note, you may use these additionally supported IPsec parameters for encryption
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
!
config
proposal ipsec lifetime 3600
proposal ipsec authentication sha1
proposal ipsec encryption aes128
proposal ipsec perfect-forward-secrecy dh-group 2
proposal ipsec protocol ESP
keep-alive
enable
commit
end
!
!
! You can use other supported IPsec parameters for encryption such as AES256, and other DH
! groups such as 2, 5, 14-18, 22, 23, and 24.
! IPsec Dead Peer Detection (DPD) will be enabled on the AWS Endpoint. We
! recommend configuring DPD on your endpoint as follows:
!   - DPD Interval      : 120
!   - DPD Retries       : 3
! To configure Dead Peer Detection for the SonicWall device, use the SonicOS management
! interface.
!
```

## Tunnel

```
! #3: Tunnel Interface Configuration
!
```

```
! The tunnel interface is configured with the internal IP address.
!
! To establish connectivity between your internal network and the VPC, you
! must have an interface facing your internal network in the "Trust" zone.
!
```




```
config
tunnel-interface vpn T1
ip-assignment VPN static
ip 169.254.44.242 netmask 255.255.255.252
!
!
```

## BGP

```
! #4: Border Gateway Protocol (BGP) Configuration:
!
! BGP is used within the tunnel to exchange prefixes between the
! Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway
! will announce the prefix corresponding to your VPC.
!!
! The local BGP Autonomous System Number (ASN) (65000)
! is configured as part of your Customer Gateway. If the ASN must
! be changed, the Customer Gateway and VPN Connection will need to be recreated with AWS.
!
routing
bgp
configure terminal
router bgp YOUR_BGP_ASN
network <Local_subnet>/24
neighbor 169.254.44.242 remote-as 7224
neighbor 169.254.44.242 timers 10 30
neighbor 169.254.44.242 soft-reconfiguration inbound
end
exit
commit
end
!
! IPsec Tunnel #2
! -----
```

## IKE

```
! #1: Internet Key Exchange (IKE) Configuration
!
! Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
! and DH Group 2.
! You can modify these sample configuration files to use AES128, SHA1, AES256, SHA256, or
! other DH groups like 2, 14-18, 22, 23, and 24.
! The address of the external interface for your customer gateway must be a static
! address.
! Your customer gateway may reside behind a device performing network address translation
! (NAT).
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
! unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
config
address-object ipv4 AWSVPC network 172.30.0.0/16
vpn policy tunnel-interface vpn-44a8938f-1
gateway primary 72.21.209.225
bound-to interface X1
auth-method shared-secret
shared-secret PRE-SHARED-KEY-IN-PLAIN-TEXT
ike-id local ip your_customer_gateway_IP_address
ike-id peer ip 72.21.209.225
end
```

```
!  
  
! #2: IPsec Configuration  
!  
! The IPsec (Phase 2) proposal defines the protocol, authentication,  
! encryption, and lifetime parameters for our IPsec security association.  
! Please note, you may use these additionally supported IPsec parameters for encryption  
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.  
!  
config  
proposal ipsec lifetime 3600  
proposal ipsec authentication sha1  
proposal ipsec encryption aes128  
proposal ipsec perfect-forward-secrecy dh-group 2  
proposal ipsec protocol ESP  
keep-alive  
enable  
commit  
end  
!  
!  
! You can use other supported IPsec parameters for encryption such as AES256, and other DH  
! groups such as 2, 5, 14-18, 22, 23, and 24.  
! IPsec Dead Peer Detection (DPD) will be enabled on the AWS Endpoint. We  
! recommend configuring DPD on your endpoint as follows:  
! - DPD Interval : 120  
! - DPD Retries : 3  
! To configure Dead Peer Detection for the SonicWall device, use the SonicOS management  
! interface.  
!  
  
! #3: Tunnel Interface Configuration  
!  
! The tunnel interface is configured with the internal IP address.  
!  
! To establish connectivity between your internal network and the VPC, you  
! must have an interface facing your internal network in the "Trust" zone.  
!  
config  
tunnel-interface vpn T2  
ip-assignment VPN static  
ip 169.254.44.114 netmask 255.255.255.252  
!  
  
! #4: Border Gateway Protocol (BGP) Configuration:  
!  
! BGP is used within the tunnel to exchange prefixes between the  
! Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway  
! will announce the prefix corresponding to your VPC.  
!  
! The local BGP Autonomous System Number (ASN) (65000)  
! is configured as part of your Customer Gateway. If the ASN must  
! be changed, the Customer Gateway and VPN Connection will need to be recreated with AWS.  
!  
routing  
bgp  
configure terminal  
router bgp YOUR_BGP_ASN  
network <Local_subnet>/24
```

```
neighbor 169.254.44.114 remote-as 7224
neighbor 169.254.44.114 timers 10 30
neighbor 169.254.44.114 soft-reconfiguration inbound
end
exit
commit
end
```

## Configuring the SonicWALL Device Using the Management Interface

You can also configure the SonicWALL device using the SonicOS management interface. For more information, see [Configuring the SonicWALL Device Using the Management Interface \(p. 73\)](#).

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

## How to Test the Customer Gateway Configuration

You can test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

1. On your customer gateway, determine whether the BGP status is **Active**.  
It takes approximately 30 seconds for a BGP peering to become active.
2. Ensure that the customer gateway is advertising a route to the virtual private gateway. The route may be the default route (0.0.0.0/0) or a more specific route you prefer.

When properly established, your BGP peering should be receiving one route from the virtual private gateway corresponding to the prefix that your VPC integration team specified for the VPC (for example, 10.0.0.0/24). 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.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection: your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the Amazon EC2 Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.



3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

#### Note

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.

If your tunnels don't test successfully, see [Troubleshooting Generic Device Customer Gateway Connectivity Using Border Gateway Protocol \(p. 180\)](#).

# Example: Dell SonicWALL SonicOS Device Without Border Gateway Protocol

This topic provides an example of how to configure your router if your customer gateway is a Dell SonicWALL router running SonicOS 5.9 or 6.2.

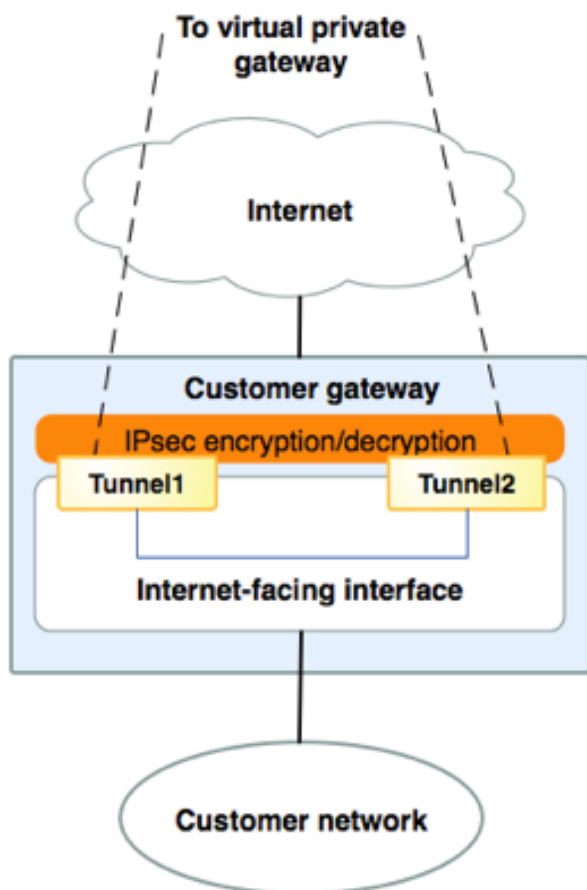
This section assumes that a VPN connection with static routing has been configured in the Amazon VPC console. For more information, see [Adding a Virtual Private Gateway to Your VPC](#) in the *Amazon VPC User Guide*.

## Topics

- [A High-Level View of the Customer Gateway](#) (p. 69)
- [Example Configuration File](#) (p. 70)
- [Configuring the SonicWALL Device Using the Management Interface](#) (p. 73)
- [How to Test the Customer Gateway Configuration](#) (p. 75)

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels: *Tunnel 1* and *Tunnel 2*. Using redundant tunnels ensures continuous availability in the case that a device fails.



## Example Configuration File

The configuration file that you download from the Amazon VPC console includes the values that you need in order to use the command line tools on OS 6.2 to configure each tunnel and the IKE and IPsec settings for your SonicWALL device.

### Important

The following configuration information uses example values — you must use the actual values and not the example values shown here, or your implementation will fail.

```
! Amazon Web Services
! Virtual Private Cloud
!
! VPN Connection Configuration
! =====
! AWS utilizes unique identifiers to manipulate the configuration of
! a VPN Connection. Each VPN Connection is assigned a VPN Connection Identifier
! and is associated with two other identifiers, namely the
! Customer Gateway Identifier and the Virtual Private Gateway Identifier.
!
! Your VPN Connection ID           : vpn-44a8938f
! Your Virtual Private Gateway ID  : vgw-8db04f81
! Your Customer Gateway ID         : cgw-ff628496
!
! This configuration consists of two tunnels. Both tunnels must be
```

```
! configured on your customer gateway.
!
! This configuration was tested on a SonicWALL TZ 600 running OS 6.2.5.1-26n
!
! You may need to populate these values throughout the config based on your setup:
! <vpc_subnet> - VPC IP address range
! =====
```

## IKE

```
! #1: Internet Key Exchange (IKE) Configuration
!
! These sample configurations are for the minimum requirement of AES128, SHA1, and DH Group
! 2.
! You can modify these sample configuration files to use AES128, SHA1, AES256, SHA256, or
! other DH groups like 2, 14-18, 22, 23, and 24.
! The address of the external interface for your customer gateway must be a static
! address.
! Your customer gateway may reside behind a device performing network address translation
! (NAT).
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
! unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
config
address-object ipv4 AWSVPC network 172.30.0.0/16
vpn policy tunnel-interface vpn-44a8938f-1
gateway primary 72.21.209.193
bound-to interface X1
auth-method shared-secret
shared-secret PRE-SHARED-KEY-IN-PLAIN-TEXT
ike-id local ip your_customer_gateway_IP_address
ike-id peer ip 72.21.209.193
end
```

## IPsec

```
! #2: IPsec Configuration
!
! The IPsec (Phase 2) proposal defines the protocol, authentication,
! encryption, and lifetime parameters for our IPsec security association.
! Please note, you may use these additionally supported IPsec parameters for encryption
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
!
config
proposal ipsec lifetime 3600
proposal ipsec authentication sha1
proposal ipsec encryption aes128
proposal ipsec perfect-forward-secrecy dh-group 2
proposal ipsec protocol ESP
keep-alive
enable
commit
end
!
! You can use other supported IPsec parameters for encryption such as AES256, and other DH
! groups such as 1,2, 5, 14-18, 22, 23, and 24.

! IPsec Dead Peer Detection (DPD) will be enabled on the AWS Endpoint. We
! recommend configuring DPD on your endpoint as follows:
! - DPD Interval          : 120
! - DPD Retries           : 3
! To configure Dead Peer Detection for the SonicWall device, use the SonicOS management
! interface.
!
```

## Tunnel

```
! #3: Tunnel Interface Configuration
!
! The tunnel interface is configured with the internal IP address.
!
! To establish connectivity between your internal network and the VPC, you
! must have an interface facing your internal network in the "Trust" zone.
!
!
config
tunnel-interface vpn T1
ip-assignment VPN static
ip 169.254.255.6 netmask 255.255.255.252
exit
!
!
! #4 Static Route Configuration
!
! Create a firewall policy permitting traffic from your local subnet to the VPC subnet and
! vice versa
! This example policy permits all traffic from the local subnet to the VPC through the
! tunnel interface.
!
!
policy interface T1 metric 1 source any destination name AWSVPC service any
gateway 169.254.255.5
!
IPSec Tunnel !2
=====
```


## IKE

```
! #1: Internet Key Exchange (IKE) Configuration
!
! These sample configurations are for the minimum requirement of AES128, SHA1, and DH Group
! 2.
! You can modify these sample configuration files to use AES128, SHA1, AES256, SHA256, or
! other DH groups like 2, 14-18, 22, 23, and 24.
! The address of the external interface for your customer gateway must be a static
! address.
! Your customer gateway may reside behind a device performing network address translation
! (NAT).
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
! unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
config
address-object ipv4 AWSVPC network 172.30.0.0/16
vpn policy tunnel-interface vpn-44a8938f-2
gateway primary 72.21.209.225
bound-to interface X1
auth-method shared-secret
shared-secret PRE-SHARED-KEY-IN-PLAIN-TEXT
ike-id local ip your_customer_gateway_IP_address
ike-id peer ip 72.21.209.225
end
!
```

## IPsec

```
! #2: IPSec Configuration
!
! The IPSec (Phase 2) proposal defines the protocol, authentication,
! encryption, and lifetime parameters for our IPSec security association.
```

```
! Please note, you may use these additionally supported IPSec parameters for encryption
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
!
config
proposal ipsec lifetime 3600
proposal ipsec authentication sha1
proposal ipsec encryption aes128
proposal ipsec perfect-forward-secrecy dh-group 2
proposal ipsec protocol ESP
keep-alive
enable
commit
end
!
! You can use other supported IPSec parameters for encryption such as AES256, and other DH
! groups such as 1,2, 5, 14-18, 22, 23, and 24.
!
! IPSec Dead Peer Detection (DPD) will be enabled on the AWS Endpoint. We
! recommend configuring DPD on your endpoint as follows:
! - DPD Interval          : 120
! - DPD Retries           : 3
! To configure Dead Peer Detection for the SonicWall device, use the SonicOS management
! interface.
!
```



```
! #3: Tunnel Interface Configuration
!
! The tunnel interface is configured with the internal IP address.
!
! To establish connectivity between your internal network and the VPC, you
! must have an interface facing your internal network in the "Trust" zone.
!
!
config
tunnel-interface vpn T2
ip-assignment VPN static
ip 169.254.255.2 netmask 255.255.255.252
!
! #4 Static Route Configuration
!
! Create a firewall policy permitting traffic from your local subnet to the VPC subnet and
! vice versa
! This example policy permits all traffic from the local subnet to the VPC through the
! tunnel interface.
!
!
policy interface T2 metric 1 source any destination name AWSVPC service any
gateway 169.254.255.1
```

## Configuring the SonicWALL Device Using the Management Interface

The following procedure demonstrates how to configure the VPN tunnels on the SonicWALL device using the SonicOS management interface. You must replace the example values in the procedures with the values that are provided in the configuration file.

## 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 **Site to Site**.
    - **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 120 (this is the minimum that the SonicWALL device accepts).
      - For **Failure Trigger Level**, enter 2.
    - 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.

## How to Test the Customer Gateway Configuration

You must first test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

- On your customer gateway, verify that you have added a static route to the VPC CIDR IP space to use the tunnel interface.

Next, you must test the connectivity for each tunnel by launching an instance into your VPC and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection; your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are available in the **Quick Start** menu when you use the Launch Instances wizard in the AWS Management Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

### Note

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.



If your tunnels don't test successfully, see [Troubleshooting Generic Device Customer Gateway Connectivity Using Border Gateway Protocol \(p. 180\)](#).

# Example: Fortinet Fortigate Device

## Topics

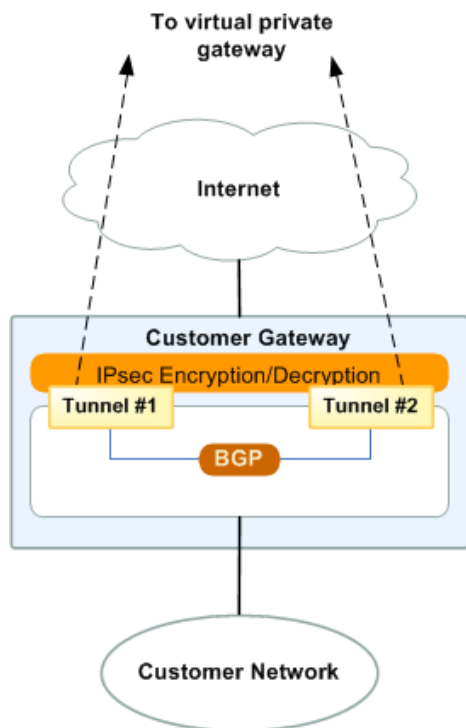
- [A High-Level View of the Customer Gateway \(p. 78\)](#)
- [A Detailed View of the Customer Gateway and an Example Configuration \(p. 78\)](#)
- [How to Test the Customer Gateway Configuration \(p. 86\)](#)

The following topic provides example configuration information provided by your integration team if your customer gateway is a Fortinet Fortigate 40+ device.

Two diagrams illustrate the example configuration. The first diagram shows the high-level layout of the customer gateway, and the second diagram shows the details of the example configuration. You should use the real configuration information that you receive from your integration team and apply it to your customer gateway.

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels. Using redundant tunnels ensures continuous availability in the case that a device fails.



## A Detailed View of the Customer Gateway and an Example Configuration

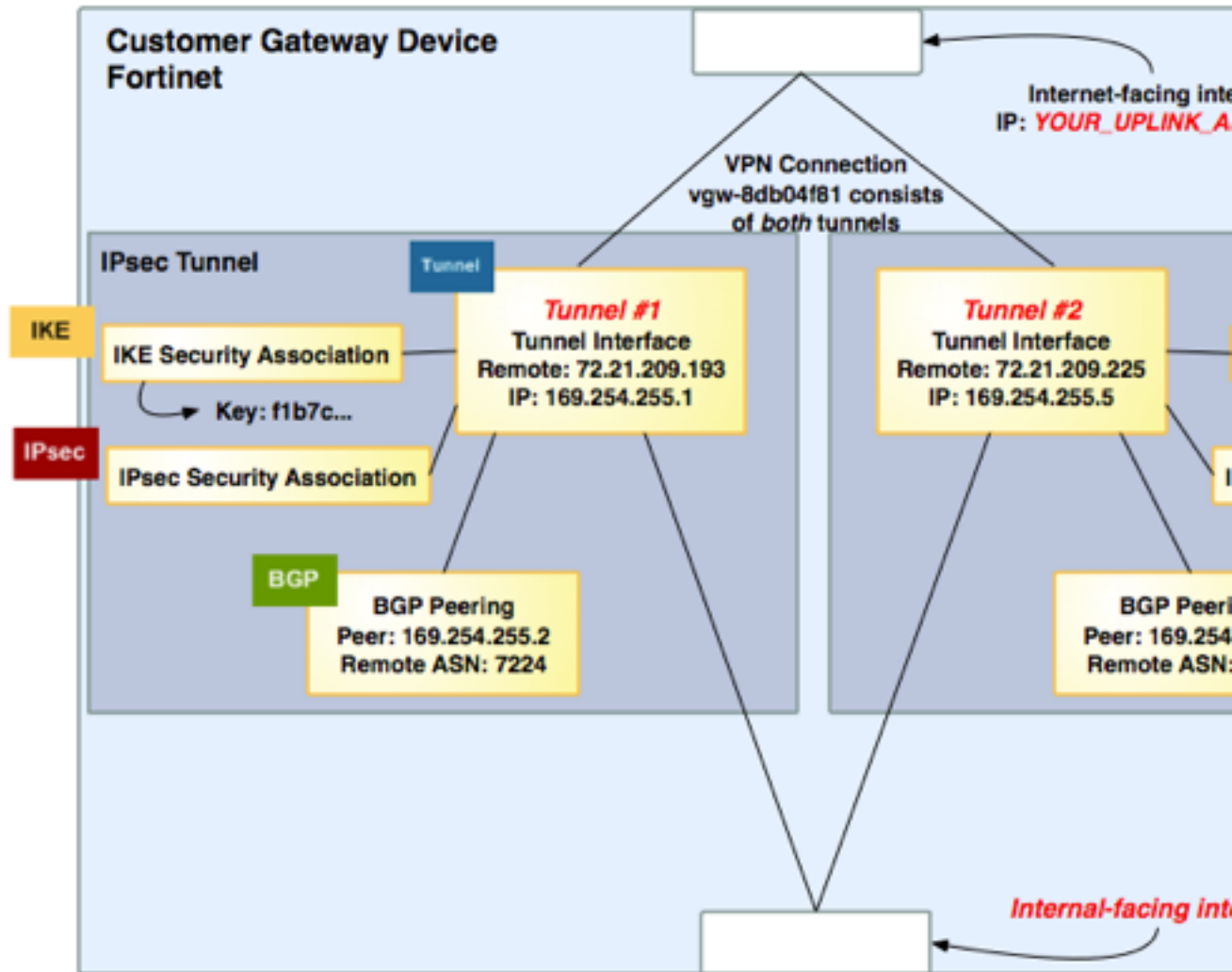
The diagram in this section illustrates an example Fortinet customer gateway. Following the diagram, there is a corresponding example of the configuration information your integration team should provide. The example configuration contains a set of information for each of the tunnels that you must configure.

In addition, the example configuration refers to these items that you must provide:

- ***YOUR\_UPLINK\_ADDRESS***—The IP address for the Internet-routable external interface on the customer gateway (which must be static, and may be behind a device performing network address translation (NAT)).
- ***YOUR\_BGP\_ASN***—The customer gateway's BGP ASN (we use 65000 by default)

The example configuration includes several example values to help you understand how configuration works. For example, we provide example values for the VPN connection ID (vpn-44a8938f), virtual private gateway ID (vgw-8db04f81), the IP addresses (72.21.209.\*, 169.254.255.\*), and the remote ASN (7224). You'll replace these example values with the actual values from the configuration information that you receive.

In the following diagram and example configuration, you must replace the items in red italics with values that apply to your particular configuration.



### Warning

The following configuration information is an example of what you can expect your integration team to provide. Many of the values in the following example will be different from the actual configuration information that you receive. You must use the actual values and not the example values shown here, or your implementation will fail.

```
! Amazon Web Services
! Virtual Private Cloud

! AWS utilizes unique identifiers to manipulate the configuration of
! a VPN Connection. Each VPN Connection is assigned an identifier and is
! associated with two other identifiers, namely the
! Customer Gateway Identifier and Virtual Private Gateway Identifier.
!
! Your VPN Connection ID      : vpn-44a8938f
! Your Virtual Private Gateway ID : vgw-8db04f81
! Your Customer Gateway ID    : cgw-b4dc3961
!
```

```
!  
! This configuration consists of two tunnels. Both tunnels must be  
! configured on your Customer Gateway.  
!
```

```
! -----  
! IPsec Tunnel #1  
! -----
```

## IKE

```
! #1: Internet Key Exchange (IKE) Configuration  
!  
! A policy is established for the supported ISAKMP encryption,  
! authentication, Diffie-Hellman, lifetime, and key parameters.  
! Please note, these sample configurations are for the minimum requirement of AES128, SHA1,  
! and DH Group 2.  
! You will need to modify these sample configuration files to take advantage of AES256,  
! SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.  
!  
! The address of the external interface for your customer gateway must be a static  
! address.  
! Your customer gateway may reside behind a device performing network address translation  
! (NAT).  
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to  
! unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.  
!  
! Configuration begins in root VDOM.  
config vpn ipsec phase1-interface  
edit vpn-44a8938f-0 ! Name must be shorter than 15 chars, best if shorter than 12  
  set interface "wan1"  
  
! The IPsec Dead Peer Detection causes periodic messages to be  
! sent to ensure a Security Association remains operational  
  
  set dpd enable  
  set local-gw YOUR_UPLINK_ADDRESS  
  set dhgrp 2  
  set proposal aes128-sha1  
  set keylife 28800  
  set remote-gw 72.21.209.193  
  set psksecret plain-text-password1  
  set dpd-retryinterval 10  
next  
end
```

## IPsec

```
! #2: IPsec Configuration  
!  
! The IPsec transform set defines the encryption, authentication, and IPsec  
! mode parameters.  
!  
! Please note, you may use these additionally supported IPsec parameters for encryption  
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.  
  
config vpn ipsec phase2-interface  
edit "vpn-44a8938f-0"  
  set phasename "vpn-44a8938f-0"  
  set proposal aes128-sha1  
  set dhgrp 2  
  set keylifeseconds 3600
```

next

## Tunnel

```
! -----
! #3: Tunnel Interface Configuration
!
! A tunnel interface is configured to be the logical interface associated
! with the tunnel. All traffic routed to the tunnel interface will be
! encrypted and transmitted to the VPC. Similarly, traffic from the VPC
! will be logically received on this interface.
!
!
! The address of the interface is configured with the setup for your
! Customer Gateway. If the address changes, the Customer Gateway and VPN
! Connection must be recreated with Amazon VPC.
!

config system interface
edit "vpn-44a8938f-0"
set vdom "root"
set ip 169.254.255.2 255.255.255.255
set allowaccess ping
set type tunnel

! This option causes the router to reduce the Maximum Segment Size of
! TCP packets to prevent packet fragmentation.
!
set tcp-mss 1387
set remote-ip 169.254.255.1
set mtu 1427
set interface "wan1"
next
```

## BGP

```
! -----
! #4: Border Gateway Protocol (BGP) Configuration
!
! BGP is used within the tunnel to exchange prefixes between the
! Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway
! will announce the prefix corresponding to your VPC.
!
!
! The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN)
! is configured as part of your Customer Gateway. If the ASN must
! be changed, the Customer Gateway and VPN Connection will need to be recreated with AWS.
!

config router bgp
set as YOUR_BGP_ASN
config neighbor
edit 169.254.255.1
set remote-as 7224
end
```

```
! Your Customer Gateway may announce a default route (0.0.0.0/0) to us.
! This is done using prefix list and route-map in Fortigate.

config router bgp
config neighbor
edit 169.254.255.1
set capability-default-originate enable
end
end

config router prefix-list
edit "default_route"
config rule
edit 1
set prefix 0.0.0.0 0.0.0.0
next
end
set router-id YOUR_UPLINK_ADDRESS
end

config router route-map
edit "routemap1"
config rule
edit 1
set match-ip-address "default_route"
next
end
next
end

! To advertise additional prefixes to Amazon VPC, add these prefixes to the 'network'
! statement and identify the prefix you wish to advertise. Make sure the prefix is present
! in the routing table of the device with a valid next-hop. If you want to advertise
! 192.168.0.0/16 to Amazon, this can be done using the following:

config router bgp
config network
edit 1
set prefix 192.168.0.0 255.255.0.0
next
end
set router-id YOUR_UPLINK_ADDRESS
end

! -----
! #5 Firewall Policy Configuration
!
! Create a firewall policy permitting traffic from your local subnet to the VPC subnet and
! vice versa
!
! This example policy permits all traffic from the local subnet to the VPC
! First, find the policies that exist

show firewall policy

! Next, create a new firewall policy starting with the next available policy ID. If
! policies 1, 2, 3, and 4 were shown, then in this example the policy created starts 5

config firewall policy
edit 5
set srcintf "vpn-44a8938f-0"
```

```
set dstintf internal
set srcaddr all
set dstaddr all
set action accept
set schedule always
set service ANY
next
end

config firewall policy
edit 5
set srcintf internal
set dstintf "vpn-44a8938f-0"
set srcaddr all
set dstaddr all
set action accept
set schedule always
set service ANY
next
end
```

## IKE

```
! -----
! IPsec Tunnel #2
! -----
! #1: Internet Key Exchange (IKE) Configuration
!
! A policy is established for the supported ISAKMP encryption,
! authentication, Diffie-Hellman, lifetime, and key parameters.
!
! The address of the external interface for your customer gateway must be a static
! address.
! Your customer gateway may reside behind a device performing network address translation
! (NAT).
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
! unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
! Configuration begins in root VDOM.
config vpn ipsec phase1-interface
edit vpn-44a8938f-1 ! Name must be shorter than 15 chars, best if shorter than 12
set interface "wan1"

! The IPsec Dead Peer Detection causes periodic messages to be
! sent to ensure a Security Association remains operational

set dpd enable
set local-gw YOUR_UPLINK_ADDRESS
set dhgrp 2
set proposal aes128-sha1
set keylife 28800
set remote-gw 72.21.209.225
set psksecret plain-text-password2
set dpd-retryinterval 10
next
end
```

## IPsec



```
! #2: IPSec Configuration
!
! The IPSec transform set defines the encryption, authentication, and IPSec
! mode parameters.
!
! Please note, you may use these additionally supported IPSec parameters for encryption
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
```

```
config vpn ipsec phase2-interface
edit "vpn-44a8938f-1"
set phasename "vpn-44a8938f-1"
set proposal aes128-sha1
set dhgrp 2
set keylifeseconds 3600
next
```

### Tunnel

```
! -----
! #3: Tunnel Interface Configuration
!
! A tunnel interface is configured to be the logical interface associated
! with the tunnel. All traffic routed to the tunnel interface will be
! encrypted and transmitted to the VPC. Similarly, traffic from the VPC
! will be logically received on this interface.
!
!
! The address of the interface is configured with the setup for your
! Customer Gateway. If the address changes, the Customer Gateway and VPN
! Connection must be recreated with Amazon VPC.
!

config system interface
edit "vpn-44a8938f-1"
set vdom "root"
set ip 169.254.255.6 255.255.255.255
set allowaccess ping
set type tunnel

! This option causes the router to reduce the Maximum Segment Size of
! TCP packets to prevent packet fragmentation.
!
set tcp-mss 1387
set remote-ip 169.254.255.5
set mtu 1427
set interface "wan1"
next
```

### BGP

```
! -----
! #4: Border Gateway Protocol (BGP) Configuration
!
! BGP is used within the tunnel to exchange prefixes between the
! Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway
! will announce the prefix corresponding to your VPC.
!
!
```

```
!
! The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN)
! is configured as part of your Customer Gateway. If the ASN must
! be changed, the Customer Gateway and VPN Connection will need to be recreated with AWS.
!

config router bgp
  set as YOUR_BGP_ASN
  config neighbor
    edit 169.254.255.5
      set remote-as 7224
    end

! Your Customer Gateway may announce a default route (0.0.0.0/0) to us.
! This is done using prefix list and route-map in Fortigate.

config router bgp
  config neighbor
    edit 169.254.255.5
      set capability-default-originate enable
    end
  end

config router prefix-list
  edit "default_route"
    config rule
      edit 1
        set prefix 0.0.0.0 0.0.0.0
      next
    end
  set router-id YOUR_UPLINK_ADDRESS
end

config router route-map
  edit "routemap1"
    config rule
      edit 1
        set match-ip-address "default_route"
      next
    end
  next
end

! To advertise additional prefixes to Amazon VPC, add these prefixes to the 'network'
! statement and identify the prefix you wish to advertise. Make sure the prefix is present
! in the routing table of the device with a valid next-hop. If you want to advertise
! 192.168.0.0/16 to Amazon, this can be done using the following:

config router bgp
config network
  edit 1
    set prefix 192.168.0.0 255.255.0.0
  next
end
set router-id YOUR_UPLINK_ADDRESS
end

!
! -----
! #5 Firewall Policy Configuration
!
```

```
! Create a firewall policy permitting traffic from your local subnet to the VPC subnet and
vice versa
!
! This example policy permits all traffic from the local subnet to the VPC
! First, find the policies that exist

show firewall policy

! Next, create a new firewall policy starting with the next available policy ID. If
policies 1, 2, 3, and 4 were shown, then in this example the policy created starts 5

config firewall policy
edit 5
set srcintf "vpn-44a8938f-1"
set dstintf internal
    set srcaddr all
    set dstaddr all
set action accept
set schedule always
    set service ANY
next
end

config firewall policy
edit 5
set srcintf internal
set dstintf "vpn-44a8938f-1"
    set srcaddr all
    set dstaddr all
set action accept
set schedule always
    set service ANY
next
end

! -----
```

## How to Test the Customer Gateway Configuration

You can test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

1. On your customer gateway, determine whether the BGP status is Active.  
It takes approximately 30 seconds for a BGP peering to become active.
2. Ensure that the customer gateway is advertising a route to the virtual private gateway. The route may be the default route (0.0.0.0/0) or a more specific route you prefer.

When properly established, your BGP peering should be receiving one route from the virtual private gateway corresponding to the prefix that your VPC integration team specified for the VPC (for example, 10.0.0.0/24). 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.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection: your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the Amazon EC2 Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

#### Note

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.

# Example: Juniper J-Series JunOS Device

## Topics

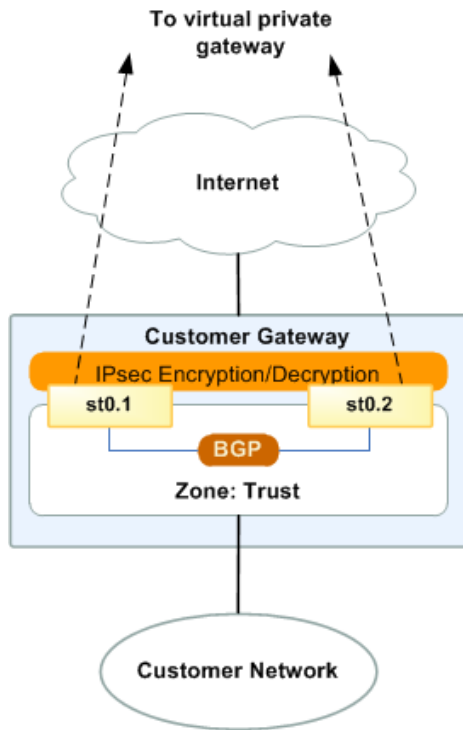
- [A High-Level View of the Customer Gateway \(p. 89\)](#)
- [A Detailed View of the Customer Gateway and an Example Configuration \(p. 90\)](#)
- [How to Test the Customer Gateway Configuration \(p. 96\)](#)

In this section we walk you through an example of the configuration information provided by your integration team if your customer gateway is a Juniper J-Series router running JunOS 9.5 (or later) software.

Two diagrams illustrate the example configuration. The first diagram shows the high-level layout of the customer gateway, and the second diagram shows details from the example configuration. You should use the real configuration information that you receive from your integration team and apply it to your customer gateway.

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels. Using redundant tunnels ensures continuous availability in the case that a device fails.



## A Detailed View of the Customer Gateway and an Example Configuration

The diagram in this section illustrates an example Juniper JunOS customer gateway. Following the diagram, there is a corresponding example of the configuration information your integration team should provide. The example configuration contains a set of information for each of the tunnels that you must configure.

In addition, the example configuration refers to these items that you must provide:

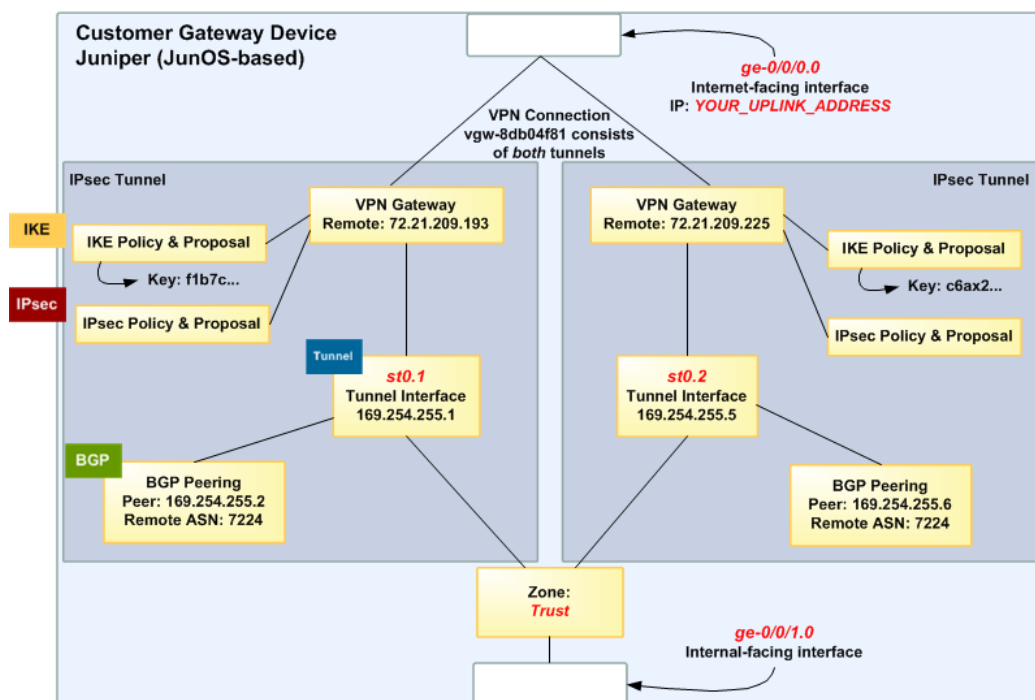
- ***YOUR\_UPLINK\_ADDRESS***—The IP address for the Internet-routable external interface on the customer gateway. The address must be static, and may be behind a device performing network address translation (NAT). To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500.
- ***YOUR\_BGP\_ASN***—The customer gateway's BGP ASN (we use 65000 by default)

The example configuration includes several example values to help you understand how configuration works. For example, we provide example values for the VPN connection ID (vpn-44a8938f), virtual private gateway ID (vgw-8db04f81), the IP addresses (72.21.209.\*, 169.254.255.\*), and the remote ASN (7224). You'll replace these example values with the actual values from the configuration information that you receive.

In addition, you must:

- Configure the outside interface (referred to as *ge-0/0/0.0* in the example configuration).
- Configure the tunnel interface IDs (referred to as *st0.1* and *st0.2* in the example configuration).
- Configure all internal routing that moves traffic between the customer gateway and your local network.
- Identify the security zone for the uplink interface (the following configuration information uses the default "untrust" zone).
- Identify the security zone for the inside interface (the following configuration information uses the default "trust" zone).

In the following diagram and example configuration, you must replace the items in red italics with values that apply to your particular configuration.



## Warning

The following configuration information is an example of what you can expect your integration team to provide. Many of the values in the following example will be different from the actual configuration information that you receive. You must use the actual values and not the example values shown here, or your implementation will fail.

```
# Amazon Web Services
# Virtual Private Cloud
#
# AWS utilizes unique identifiers to manipulate the configuration of
# a VPN Connection. Each VPN Connection is assigned a VPN Connection
# Identifier and is associated with two other identifiers, namely the
# Customer Gateway Identifier and the Virtual Private Gateway Identifier.
#
# Your VPN Connection ID           : vpn-44a8938f
# Your Virtual Private Gateway ID  : vgw-8db04f81
# Your Customer Gateway ID        : cgw-b4dc3961
#
# This configuration consists of two tunnels. Both tunnels must be
# configured on your Customer Gateway.
#
# -----
# IPsec Tunnel #1
# -----
```

**IKE**

```
# #1: Internet Key Exchange (IKE) Configuration
#
# A proposal is established for the supported IKE encryption,
# authentication, Diffie-Hellman, and lifetime parameters.
#
# Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
# and DH Group 2.
```



Amazon Virtual Private Cloud Network Administrator Guide  
A Detailed View of the Customer  
Gateway and an Example Configuration

---

```
# You will need to modify these sample configuration files to take advantage of AES256,
SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
# The address of the external interface for your customer gateway must be a static
address.
# To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
#
set security ike proposal ike-prop-vpn-44a8938f-1 authentication-method pre-shared-keys
set security ike proposal ike-prop-vpn-44a8938f-1 authentication-algorithm sha1
set security ike proposal ike-prop-vpn-44a8938f-1 encryption-algorithm aes-128-cbc
set security ike proposal ike-prop-vpn-44a8938f-1 lifetime-seconds 28800
set security ike proposal ike-prop-vpn-44a8938f-1 dh-group group2

# An IKE policy is established to associate a Pre Shared Key with the
# defined proposal.
#
set security ike policy ike-pol-vpn-44a8938f-1 mode main
set security ike policy ike-pol-vpn-44a8938f-1 proposals ike-prop-vpn-44a8938f-0
set security ike policy ike-pol-vpn-44a8938f-1 pre-shared-key ascii-text plain-text-
password1

# The IKE gateway is defined to be the Virtual Private Gateway. The gateway
# configuration associates a local interface, remote IP address, and
# IKE policy.
#
# This example shows the outside of the tunnel as interface ge-0/0/0.0.
# This should be set to the interface that IP address YOUR_UPLINK_ADDRESS is
# associated with.
# This address is configured with the setup for your Customer Gateway.
#
# If the address changes, the Customer Gateway and VPN Connection must
# be recreated.
set security ike gateway gw-vpn-44a8938f-1 ike-policy ike-pol-vpn-44a8938f-0
set security ike gateway gw-vpn-44a8938f-1 external-interface ge-0/0/0.0
set security ike gateway gw-vpn-44a8938f-1 address 72.21.209.225

# Troubleshooting IKE connectivity can be aided by enabling IKE tracing.
# The configuration below will cause the router to log IKE messages to
# the 'kmd' log. Run 'show messages kmd' to retrieve these logs.
# set security ike traceoptions file kmd
# set security ike traceoptions file size 1024768
# set security ike traceoptions file files 10
# set security ike traceoptions flag all
```

## IPsec

```
# #2: IPsec Configuration
#
# The IPsec proposal defines the protocol, authentication, encryption, and
# lifetime parameters for our IPsec security association.
# Please note, you may use these additionally supported IPsec parameters for encryption
# like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
#
set security ipsec proposal ipsec-prop-vpn-44a8938f-1 protocol esp
set security ipsec proposal ipsec-prop-vpn-44a8938f-1 authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-prop-vpn-44a8938f-1 encryption-algorithm aes-128-cbc
set security ipsec proposal ipsec-prop-vpn-44a8938f-1 lifetime-seconds 3600

# The IPsec policy incorporates the Diffie-Hellman group and the IPsec
# proposal.
#
set security ipsec policy ipsec-pol-vpn-44a8938f-1 perfect-forward-secrecy keys group2
set security ipsec policy ipsec-pol-vpn-44a8938f-1 proposals ipsec-prop-vpn-44a8938f-0
```

```
# A security association is defined here. The IPsec Policy and IKE gateways
# are associated with a tunnel interface (st0.1).
# The tunnel interface ID is assumed; if other tunnels are defined on
# your router, you will need to specify a unique interface name
# (for example, st0.10).
#
set security ipsec vpn vpn-44a8938f-1 bind-interface st0.1
set security ipsec vpn vpn-44a8938f-1 ike gateway gw-vpn-44a8938f-0
set security ipsec vpn vpn-44a8938f-1 ike ipsec-policy ipsec-pol-vpn-44a8938f-0
set security ipsec vpn vpn-44a8938f-1 df-bit clear

# This option enables IPsec Dead Peer Detection, which causes periodic
# messages to be sent to ensure a Security Association remains operational.
#
set security ike gateway gw-vpn-44a8938f-1 dead-peer-detection
```

### Tunnel

```
# #3: Tunnel Interface Configuration
#
# The tunnel interface is configured with the internal IP address.
#
set interfaces st0.1 family inet address 169.254.255.2/30
set interfaces st0.1 family inet mtu 1436
set security zones security-zone trust interfaces st0.1

# The security zone protecting external interfaces of the router must be
# configured to allow IKE traffic inbound.
#
set security zones security-zone untrust host-inbound-traffic system-services ike

# The security zone protecting internal interfaces (including the logical
# tunnel interfaces) must be configured to allow BGP traffic inbound.
#
set security zones security-zone trust host-inbound-traffic protocols bgp

# This option causes the router to reduce the Maximum Segment Size of
# TCP packets to prevent packet fragmentation.
#
set security flow tcp-mss ipsec-vpn mss 1387
```

### BGP

```
# #4: Border Gateway Protocol (BGP) Configuration
#
# BGP is used within the tunnel to exchange prefixes between the
# Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway
# will announce the prefix corresponding to your VPC.
#
# Your Customer Gateway may announce a default route (0.0.0.0/0),
# which can be done with the EXPORT-DEFAULT policy.
#
# To advertise additional prefixes to Amazon VPC, add additional prefixes to the "default"
# term
# EXPORT-DEFAULT policy. Make sure the prefix is present in the routing table of the device
# with
# a valid next-hop.
#
# The BGP timers are adjusted to provide more rapid detection of outages.
#
# The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN) is configured
# as part of your Customer Gateway. If the ASN must be changed, the
```

```
# Customer Gateway and VPN Connection will need to be recreated with AWS.
#
# We establish a basic route policy to export a default route to the
# Virtual Private Gateway.
#
set policy-options policy-statement EXPORT-DEFAULT term default from route-filter 0.0.0.0/0
  exact
set policy-options policy-statement EXPORT-DEFAULT term default then accept
set policy-options policy-statement EXPORT-DEFAULT term reject then reject

set protocols bgp group ebgp type external

set protocols bgp group ebgp neighbor 169.254.255.1 export EXPORT-DEFAULT
set protocols bgp group ebgp neighbor 169.254.255.1 peer-as 7224
set protocols bgp group ebgp neighbor 169.254.255.1 hold-time 30
set protocols bgp group ebgp neighbor 169.254.255.1 local-as YOUR_BGP_ASN

# -----
# IPsec Tunnel #2
# -----

IKE

# #1: Internet Key Exchange (IKE) Configuration
#
# A proposal is established for the supported IKE encryption,
# authentication, Diffie-Hellman, and lifetime parameters.
# Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
# and DH Group 2.
# You will need to modify these sample configuration files to take advantage of AES256,
# SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
# The address of the external interface for your customer gateway must be a static
# address.
# To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
# unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
#
set security ike proposal ike-prop-vpn-44a8938f-2 authentication-method pre-shared-keys
set security ike proposal ike-prop-vpn-44a8938f-2 authentication-algorithm sha1
set security ike proposal ike-prop-vpn-44a8938f-2 encryption-algorithm aes-128-cbc
set security ike proposal ike-prop-vpn-44a8938f-2 lifetime-seconds 28800
set security ike proposal ike-prop-vpn-44a8938f-2 dh-group group2

# An IKE policy is established to associate a Pre Shared Key with the
# defined proposal.
#
set security ike policy ike-pol-vpn-44a8938f-2 mode main
set security ike policy ike-pol-vpn-44a8938f-2 proposals ike-prop-vpn-44a8938f-2
set security ike policy ike-pol-vpn-44a8938f-2 pre-shared-key ascii-text plain-text-
password2

# The IKE gateway is defined to be the Virtual Private Gateway. The gateway
# configuration associates a local interface, remote IP address, and
# IKE policy.
#
# This example shows the outside of the tunnel as interface ge-0/0/0.0.
# This should be set to the interface that IP address YOUR_UPLINK_ADDRESS is
# associated with.
# This address is configured with the setup for your Customer Gateway.
#
# If the address changes, the Customer Gateway and VPN Connection must be recreated.
#
set security ike gateway gw-vpn-44a8938f-2 ike-policy ike-pol-vpn-44a8938f-1
set security ike gateway gw-vpn-44a8938f-2 external-interface ge-0/0/0.0
set security ike gateway gw-vpn-44a8938f-2 address 72.21.209.193
```

```
# Troubleshooting IKE connectivity can be aided by enabling IKE tracing.
# The configuration below will cause the router to log IKE messages to
# the 'kmd' log. Run 'show messages kmd' to retrieve these logs.
# set security ike traceoptions file kmd
# set security ike traceoptions file size 1024768
# set security ike traceoptions file files 10
# set security ike traceoptions flag all
```

## IPsec

```
# #2: IPsec Configuration
#
# The IPsec proposal defines the protocol, authentication, encryption, and
# lifetime parameters for our IPsec security association.
# Please note, you may use these additionally supported IPsec parameters for encryption
# like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
#
set security ipsec proposal ipsec-prop-vpn-44a8938f-2 protocol esp
set security ipsec proposal ipsec-prop-vpn-44a8938f-2 authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-prop-vpn-44a8938f-2 encryption-algorithm aes-128-cbc
set security ipsec proposal ipsec-prop-vpn-44a8938f-2 lifetime-seconds 3600

# The IPsec policy incorporates the Diffie-Hellman group and the IPsec
# proposal.
#
set security ipsec policy ipsec-pol-vpn-44a8938f-2 perfect-forward-secrecy keys group2
set security ipsec policy ipsec-pol-vpn-44a8938f-2 proposals ipsec-prop-vpn-44a8938f-2

# A security association is defined here. The IPsec Policy and IKE gateways
# are associated with a tunnel interface (st0.2).
# The tunnel interface ID is assumed; if other tunnels are defined on
# your router, you will need to specify a unique interface name
# (for example, st0.20).
#
set security ipsec vpn vpn-44a8938f-2 bind-interface st0.2
set security ipsec vpn vpn-44a8938f-2 ike gateway gw-vpn-44a8938f-2
set security ipsec vpn vpn-44a8938f-2 ike ipsec-policy ipsec-pol-vpn-44a8938f-2
set security ipsec vpn vpn-44a8938f-2 df-bit clear

# This option enables IPsec Dead Peer Detection, which causes periodic
# messages to be sent to ensure a Security Association remains operational.
#
set security ike gateway gw-vpn-44a8938f-2 dead-peer-detection
```

## Tunnel

```
# #3: Tunnel Interface Configuration
#
# The tunnel interface is configured with the internal IP address.
#
set interfaces st0.2 family inet address 169.254.255.6/30
set interfaces st0.2 family inet mtu 1436
set security zones security-zone trust interfaces st0.2

# The security zone protecting external interfaces of the router must be
# configured to allow IKE traffic inbound.
#
set security zones security-zone untrust host-inbound-traffic system-services ike

# The security zone protecting internal interfaces (including the logical
# tunnel interfaces) must be configured to allow BGP traffic inbound.
#
```

```
set security zones security-zone trust host-inbound-traffic protocols bgp
```

```
# This option causes the router to reduce the Maximum Segment Size of  
# TCP packets to prevent packet fragmentation.
```

```
#  
set security flow tcp-mss ipsec-vpn mss 1387
```

## BGP

```
# #4: Border Gateway Protocol (BGP) Configuration
```

```
#  
# BGP is used within the tunnel to exchange prefixes between the  
# Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway  
# will announce the prefix corresponding to your VPC.
```

```
#  
# Your Customer Gateway may announce a default route (0.0.0.0/0),  
# which can be done with the EXPORT-DEFAULT policy.
```

```
#  
# To advertise additional prefixes to Amazon VPC, add additional prefixes to the "default"  
term
```

```
# EXPORT-DEFAULT policy. Make sure the prefix is present in the routing table of the device  
with  
# a valid next-hop.
```

```
#  
# The BGP timers are adjusted to provide more rapid detection of outages.
```

```
#  
# The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN) is configured  
# as part of your Customer Gateway. If the ASN must be changed, the  
# Customer Gateway and VPN Connection will need to be recreated with AWS.
```

```
#  
# We establish a basic route policy to export a default route to the  
# Virtual Private Gateway.
```

```
#  
set policy-options policy-statement EXPORT-DEFAULT term default from route-filter 0.0.0.0/0  
exact
```

```
set policy-options policy-statement EXPORT-DEFAULT term default then accept  
set policy-options policy-statement EXPORT-DEFAULT term reject then reject
```

```
set protocols bgp group ebgp type external
```

```
set protocols bgp group ebgp neighbor 169.254.255.5 export EXPORT-DEFAULT
```

```
set protocols bgp group ebgp neighbor 169.254.255.5 peer-as 7224
```

```
set protocols bgp group ebgp neighbor 169.254.255.5 hold-time 30
```

```
set protocols bgp group ebgp neighbor 169.254.255.5 local-as YOUR_BGP_ASN
```

## How to Test the Customer Gateway Configuration

You can test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

1. On your customer gateway, determine whether the BGP status is Active.

It takes approximately 30 seconds for a BGP peering to become active.

2. Ensure that the customer gateway is advertising a route to the virtual private gateway. The route may be the default route (0.0.0.0/0) or a more specific route you prefer.

When properly established, your BGP peering should be receiving one route from the virtual private gateway corresponding to the prefix that your VPC integration team specified for the VPC (for example,

10.0.0.0/24). 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.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection: your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the Amazon EC2 Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

#### Note

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.

If your tunnels don't test successfully, see [Troubleshooting Juniper JunOS Customer Gateway Connectivity \(p. 171\)](#).

# Example: Juniper SRX JunOS Device

## Topics

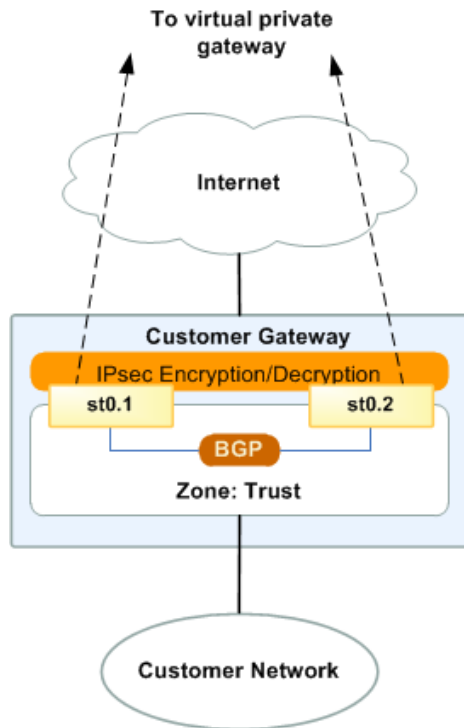
- [A High-Level View of the Customer Gateway \(p. 99\)](#)
- [A Detailed View of the Customer Gateway and an Example Configuration \(p. 100\)](#)
- [How to Test the Customer Gateway Configuration \(p. 106\)](#)

In this section we walk you through an example of the configuration information provided by your integration team if your customer gateway is a Juniper SRX router running JunOS 11.0 (or later) software.

Two diagrams illustrate the example configuration. The first diagram shows the high-level layout of the customer gateway, and the second diagram shows details from the example configuration. You should use the real configuration information that you receive from your integration team and apply it to your customer gateway.

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels. Using redundant tunnels ensures continuous availability in the case that a device fails.





## A Detailed View of the Customer Gateway and an Example Configuration

The diagram in this section illustrates an example Juniper JunOS 11.0+ customer gateway. Following the diagram, there is a corresponding example of the configuration information your integration team should provide. The example configuration contains a set of information for each of the tunnels that you must configure.

In addition, the example configuration refers to these items that you must provide:

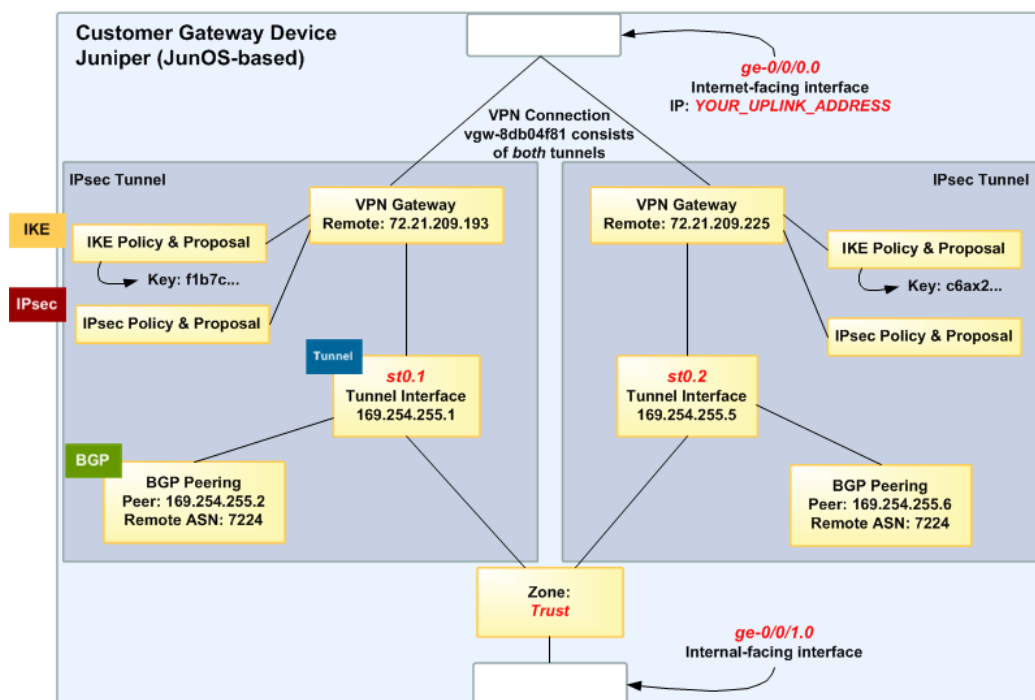
- ***YOUR\_UPLINK\_ADDRESS***—The IP address for the Internet-routable external interface on the customer gateway. The address must be static, and may be behind a device performing network address translation (NAT). To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500.
- ***YOUR\_BGP\_ASN***—The customer gateway's BGP ASN (we use 65000 by default)

The example configuration includes several example values to help you understand how configuration works. For example, we provide example values for the VPN connection ID (vpn-44a8938f), virtual private gateway ID (vgw-8db04f81), the IP addresses (72.21.209.\*, 169.254.255.\*), and the remote ASN (7224). You'll replace these example values with the actual values from the configuration information that you receive.

In addition, you must:

- Configure the outside interface (referred to as *ge-0/0/0.0* in the example configuration).
- Configure the tunnel interface IDs (referred to as *st0.1* and *st0.2* in the example configuration).
- Configure all internal routing that moves traffic between the customer gateway and your local network.
- Identify the security zone for the uplink interface (the following configuration information uses the default "untrust" zone).
- Identify the security zone for the inside interface (the following configuration information uses the default "trust" zone).

In the following diagram and example configuration, you must replace the items in red italics with values that apply to your particular configuration.



## Warning

The following configuration information is an example of what you can expect your integration team to provide. Many of the values in the following example will be different from the actual configuration information that you receive. You must use the actual values and not the example values shown here, or your implementation will fail.

```
# Amazon Web Services
# Virtual Private Cloud
#
# AWS utilizes unique identifiers to manipulate the configuration of
# a VPN Connection. Each VPN Connection is assigned a VPN Connection
# Identifier and is associated with two other identifiers, namely the
# Customer Gateway Identifier and the Virtual Private Gateway Identifier.
#
# Your VPN Connection ID           : vpn-44a8938f
# Your Virtual Private Gateway ID  : vgw-8db04f81
# Your Customer Gateway ID        : cgw-b4dc3961
#
# This configuration consists of two tunnels. Both tunnels must be
# configured on your Customer Gateway.
#
# -----
# IPsec Tunnel #1
# -----
```

**IKE**

```
# #1: Internet Key Exchange (IKE) Configuration
#
# A proposal is established for the supported IKE encryption,
# authentication, Diffie-Hellman, and lifetime parameters.
#
# Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
# and DH Group 2.
```

Amazon Virtual Private Cloud Network Administrator Guide  
A Detailed View of the Customer  
Gateway and an Example Configuration

---

```
# You will need to modify these sample configuration files to take advantage of AES256,
SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
# The address of the external interface for your customer gateway must be a static
address.
# To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
#
set security ike proposal ike-prop-vpn-44a8938f-1 authentication-method pre-shared-keys
set security ike proposal ike-prop-vpn-44a8938f-1 authentication-algorithm sha1
set security ike proposal ike-prop-vpn-44a8938f-1 encryption-algorithm aes-128-cbc
set security ike proposal ike-prop-vpn-44a8938f-1 lifetime-seconds 28800
set security ike proposal ike-prop-vpn-44a8938f-1 dh-group group2

# An IKE policy is established to associate a Pre Shared Key with the
# defined proposal.
#
set security ike policy ike-pol-vpn-44a8938f-1 mode main
set security ike policy ike-pol-vpn-44a8938f-1 proposals ike-prop-vpn-44a8938f-1
set security ike policy ike-pol-vpn-44a8938f-1 pre-shared-key ascii-text plain-text-
password1

# The IKE gateway is defined to be the Virtual Private Gateway. The gateway
# configuration associates a local interface, remote IP address, and
# IKE policy.
#
# This example shows the outside of the tunnel as interface ge-0/0/0.0.
# This should be set to the interface that IP address YOUR_UPLINK_ADDRESS is
# associated with.
# This address is configured with the setup for your Customer Gateway.
#
# If the address changes, the Customer Gateway and VPN Connection must
# be recreated.
set security ike gateway gw-vpn-44a8938f-1 ike-policy ike-pol-vpn-44a8938f-1
set security ike gateway gw-vpn-44a8938f-1 external-interface ge-0/0/0.0
set security ike gateway gw-vpn-44a8938f-1 address 72.21.209.225
set security ike gateway gw-vpn-44a8938f-1 no-nat-traversal

# Troubleshooting IKE connectivity can be aided by enabling IKE tracing.
# The configuration below will cause the router to log IKE messages to
# the 'kmd' log. Run 'show messages kmd' to retrieve these logs.
# set security ike traceoptions file kmd
# set security ike traceoptions file size 1024768
# set security ike traceoptions file files 10
# set security ike traceoptions flag all
```

## IPsec

```
# #2: IPsec Configuration
#
# The IPsec proposal defines the protocol, authentication, encryption, and
# lifetime parameters for our IPsec security association.
# Please note, you may use these additionally supported IPsec parameters for encryption
like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
#
set security ipsec proposal ipsec-prop-vpn-44a8938f-1 protocol esp
set security ipsec proposal ipsec-prop-vpn-44a8938f-1 authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-prop-vpn-44a8938f-1 encryption-algorithm aes-128-cbc
set security ipsec proposal ipsec-prop-vpn-44a8938f-1 lifetime-seconds 3600

# The IPsec policy incorporates the Diffie-Hellman group and the IPsec
# proposal.
#
set security ipsec policy ipsec-pol-vpn-44a8938f-1 perfect-forward-secrecy keys group2
set security ipsec policy ipsec-pol-vpn-44a8938f-1 proposals ipsec-prop-vpn-44a8938f-1
```

```
# A security association is defined here. The IPsec Policy and IKE gateways
# are associated with a tunnel interface (st0.1).
# The tunnel interface ID is assumed; if other tunnels are defined on
# your router, you will need to specify a unique interface name
# (for example, st0.10).
#
set security ipsec vpn vpn-44a8938f-1 bind-interface st0.1
set security ipsec vpn vpn-44a8938f-1 ike gateway gw-vpn-44a8938f-1
set security ipsec vpn vpn-44a8938f-1 ike ipsec-policy ipsec-pol-vpn-44a8938f-1
set security ipsec vpn vpn-44a8938f-1 df-bit clear

# This option enables IPsec Dead Peer Detection, which causes periodic
# messages to be sent to ensure a Security Association remains operational.
#
set security ike gateway gw-vpn-44a8938f-1 dead-peer-detection
```

### Tunnel

```
# #3: Tunnel Interface Configuration
#
# The tunnel interface is configured with the internal IP address.
#
set interfaces st0.1 family inet address 169.254.255.2/30
set interfaces st0.1 family inet mtu 1436
set security zones security-zone trust interfaces st0.1

# The security zone protecting external interfaces of the router must be
# configured to allow IKE traffic inbound.
#
set security zones security-zone untrust host-inbound-traffic system-services ike

# The security zone protecting internal interfaces (including the logical
# tunnel interfaces) must be configured to allow BGP traffic inbound.
#
set security zones security-zone trust host-inbound-traffic protocols bgp

# This option causes the router to reduce the Maximum Segment Size of
# TCP packets to prevent packet fragmentation.
#
set security flow tcp-mss ipsec-vpn mss 1387
```

### BGP

```
# #4: Border Gateway Protocol (BGP) Configuration
#
# BGP is used within the tunnel to exchange prefixes between the
# Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway
# will announce the prefix corresponding to your VPC.
#
# Your Customer Gateway may announce a default route (0.0.0.0/0),
# which can be done with the EXPORT-DEFAULT policy.
#
# To advertise additional prefixes to Amazon VPC, add additional prefixes to the "default"
# term
# EXPORT-DEFAULT policy. Make sure the prefix is present in the routing table of the device
# with
# a valid next-hop.
#
# The BGP timers are adjusted to provide more rapid detection of outages.
#
# The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN) is configured
```

Amazon Virtual Private Cloud Network Administrator Guide  
A Detailed View of the Customer  
Gateway and an Example Configuration

---

```
# as part of your Customer Gateway. If the ASN must be changed, the
# Customer Gateway and VPN Connection will need to be recreated with AWS.
#
# We establish a basic route policy to export a default route to the
# Virtual Private Gateway.
#
set policy-options policy-statement EXPORT-DEFAULT term default from route-filter 0.0.0.0/0
  exact
set policy-options policy-statement EXPORT-DEFAULT term default then accept
set policy-options policy-statement EXPORT-DEFAULT term reject then reject

set protocols bgp group ebgp type external

set protocols bgp group ebgp neighbor 169.254.255.1 export EXPORT-DEFAULT
set protocols bgp group ebgp neighbor 169.254.255.1 peer-as 7224
set protocols bgp group ebgp neighbor 169.254.255.1 hold-time 30
set protocols bgp group ebgp neighbor 169.254.255.1 local-as YOUR_BGP_ASN

# -----
# IPsec Tunnel #2
# -----



## IKE



# #1: Internet Key Exchange (IKE) Configuration
#
# A proposal is established for the supported IKE encryption,
# authentication, Diffie-Hellman, and lifetime parameters.
# Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
# and DH Group 2.
# You will need to modify these sample configuration files to take advantage of AES256,
# SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
# The address of the external interface for your customer gateway must be a static
# address.
# To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
# unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
#
set security ike proposal ike-prop-vpn-44a8938f-2 authentication-method pre-shared-keys
set security ike proposal ike-prop-vpn-44a8938f-2 authentication-algorithm sha1
set security ike proposal ike-prop-vpn-44a8938f-2 encryption-algorithm aes-128-cbc
set security ike proposal ike-prop-vpn-44a8938f-2 lifetime-seconds 28800
set security ike proposal ike-prop-vpn-44a8938f-2 dh-group group2

# An IKE policy is established to associate a Pre Shared Key with the
# defined proposal.
#
set security ike policy ike-pol-vpn-44a8938f-2 mode main
set security ike policy ike-pol-vpn-44a8938f-2 proposals ike-prop-vpn-44a8938f-2
set security ike policy ike-pol-vpn-44a8938f-2 pre-shared-key ascii-text plain-text-
password2

# The IKE gateway is defined to be the Virtual Private Gateway. The gateway
# configuration associates a local interface, remote IP address, and
# IKE policy.
#
# This example shows the outside of the tunnel as interface ge-0/0/0.0.
# This should be set to the interface that IP address YOUR_UPLINK_ADDRESS is
# associated with.
# This address is configured with the setup for your Customer Gateway.
#
# If the address changes, the Customer Gateway and VPN Connection must be recreated.
#
set security ike gateway gw-vpn-44a8938f-2 ike-policy ike-pol-vpn-44a8938f-2
set security ike gateway gw-vpn-44a8938f-2 external-interface ge-0/0/0.0
set security ike gateway gw-vpn-44a8938f-2 address 72.21.209.193
```

```
set security ike gateway gw-vpn-44a8938f-2 no-nat-traversal
```

```
# Troubleshooting IKE connectivity can be aided by enabling IKE tracing.  
# The configuration below will cause the router to log IKE messages to  
# the 'kmd' log. Run 'show messages kmd' to retrieve these logs.  
# set security ike traceoptions file kmd  
# set security ike traceoptions file size 1024768  
# set security ike traceoptions file files 10  
# set security ike traceoptions flag all
```

## IPsec

```
# #2: IPsec Configuration
```

```
#  
# The IPsec proposal defines the protocol, authentication, encryption, and  
# lifetime parameters for our IPsec security association.  
# Please note, you may use these additionally supported IPsec parameters for encryption  
# like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.  
#  
set security ipsec proposal ipsec-prop-vpn-44a8938f-2 protocol esp  
set security ipsec proposal ipsec-prop-vpn-44a8938f-2 authentication-algorithm hmac-shal-96  
set security ipsec proposal ipsec-prop-vpn-44a8938f-2 encryption-algorithm aes-128-cbc  
set security ipsec proposal ipsec-prop-vpn-44a8938f-2 lifetime-seconds 3600
```

```
# The IPsec policy incorporates the Diffie-Hellman group and the IPsec  
# proposal.  
#  
set security ipsec policy ipsec-pol-vpn-44a8938f-2 perfect-forward-secrecy keys group2  
set security ipsec policy ipsec-pol-vpn-44a8938f-2 proposals ipsec-prop-vpn-44a8938f-2
```

```
# A security association is defined here. The IPsec Policy and IKE gateways  
# are associated with a tunnel interface (st0.2).  
# The tunnel interface ID is assumed; if other tunnels are defined on  
# your router, you will need to specify a unique interface name  
# (for example, st0.20).  
#  
set security ipsec vpn vpn-44a8938f-2 bind-interface st0.2  
set security ipsec vpn vpn-44a8938f-2 ike gateway gw-vpn-44a8938f-2  
set security ipsec vpn vpn-44a8938f-2 ike ipsec-policy ipsec-pol-vpn-44a8938f-2  
set security ipsec vpn vpn-44a8938f-2 df-bit clear
```

```
# This option enables IPsec Dead Peer Detection, which causes periodic  
# messages to be sent to ensure a Security Association remains operational.  
#  
set security ike gateway gw-vpn-44a8938f-2 dead-peer-detection
```

## Tunnel

```
# #3: Tunnel Interface Configuration
```

```
#  
# The tunnel interface is configured with the internal IP address.  
#  
set interfaces st0.2 family inet address 169.254.255.6/30  
set interfaces st0.2 family inet mtu 1436  
set security zones security-zone trust interfaces st0.2  
  
# The security zone protecting external interfaces of the router must be  
# configured to allow IKE traffic inbound.  
#  
set security zones security-zone untrust host-inbound-traffic system-services ike  
  
# The security zone protecting internal interfaces (including the logical
```

```
# tunnel interfaces) must be configured to allow BGP traffic inbound.
#
set security zones security-zone trust host-inbound-traffic protocols bgp

# This option causes the router to reduce the Maximum Segment Size of
# TCP packets to prevent packet fragmentation.
#
set security flow tcp-mss ipsec-vpn mss 1387
```

## BGP

```
# #4: Border Gateway Protocol (BGP) Configuration
#
# BGP is used within the tunnel to exchange prefixes between the
# Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway
# will announce the prefix corresponding to your VPC.
#
# Your Customer Gateway may announce a default route (0.0.0.0/0),
# which can be done with the EXPORT-DEFAULT policy.
#
# To advertise additional prefixes to Amazon VPC, add additional prefixes to the "default"
term
# EXPORT-DEFAULT policy. Make sure the prefix is present in the routing table of the device
with
# a valid next-hop.
#
# The BGP timers are adjusted to provide more rapid detection of outages.
#
# The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN) is configured
# as part of your Customer Gateway. If the ASN must be changed, the
# Customer Gateway and VPN Connection will need to be recreated with AWS.
#
# We establish a basic route policy to export a default route to the
# Virtual Private Gateway.
#
set policy-options policy-statement EXPORT-DEFAULT term default from route-filter 0.0.0.0/0
exact
set policy-options policy-statement EXPORT-DEFAULT term default then accept
set policy-options policy-statement EXPORT-DEFAULT term reject then reject

set protocols bgp group ebgp type external

set protocols bgp group ebgp neighbor 169.254.255.5 export EXPORT-DEFAULT
set protocols bgp group ebgp neighbor 169.254.255.5 peer-as 7224
set protocols bgp group ebgp neighbor 169.254.255.5 hold-time 30
set protocols bgp group ebgp neighbor 169.254.255.5 local-as YOUR_BGP_ASN
```

## How to Test the Customer Gateway Configuration

You can test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

1. On your customer gateway, determine whether the BGP status is Active.

It takes approximately 30 seconds for a BGP peering to become active.

2. Ensure that the customer gateway is advertising a route to the virtual private gateway. The route may be the default route (0.0.0.0/0) or a more specific route you prefer.

When properly established, your BGP peering should be receiving one route from the virtual private gateway corresponding to the prefix that your VPC integration team specified for the VPC (for example, 10.0.0.0/24). 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.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection: your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the Amazon EC2 Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

#### Note

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.

If your tunnels don't test successfully, see [Troubleshooting Juniper JunOS Customer Gateway Connectivity \(p. 171\)](#).



# Example: Juniper ScreenOS Device

## Topics

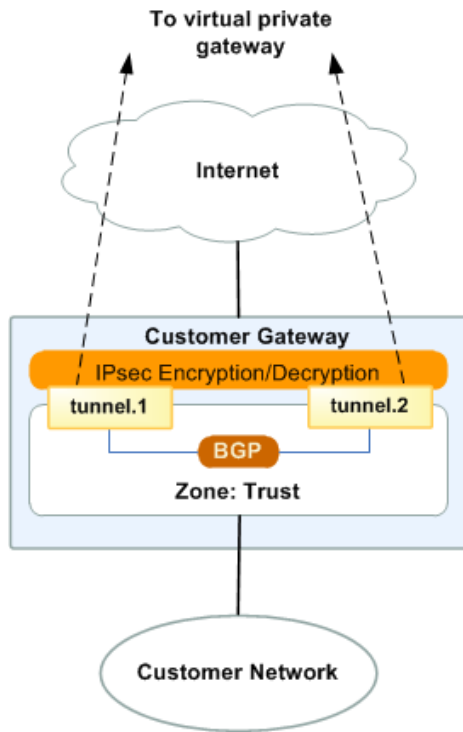
- [A High-Level View of the Customer Gateway \(p. 109\)](#)
- [A Detailed View of the Customer Gateway and an Example Configuration \(p. 110\)](#)
- [How to Test the Customer Gateway Configuration \(p. 115\)](#)

In this section we walk you through an example of the configuration information provided by your integration team if your customer gateway is a Juniper SSG or Netscreen series device running Juniper ScreenOS software.

Two diagrams illustrate the example configuration. The first diagram shows the high-level layout of the customer gateway, and the second diagram shows details from the example configuration. You should use the real configuration information that you receive from your integration team and apply it to your customer gateway.

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels. Using redundant tunnels ensures continuous availability in the case that a device fails.



## A Detailed View of the Customer Gateway and an Example Configuration

The diagram in this section illustrates an example Juniper ScreenOS customer gateway. Following the diagram, there is a corresponding example of the configuration information your integration team should provide. The example configuration contains information for each of the tunnels that you must configure.

In addition, the example configuration refers to these items that you must provide:

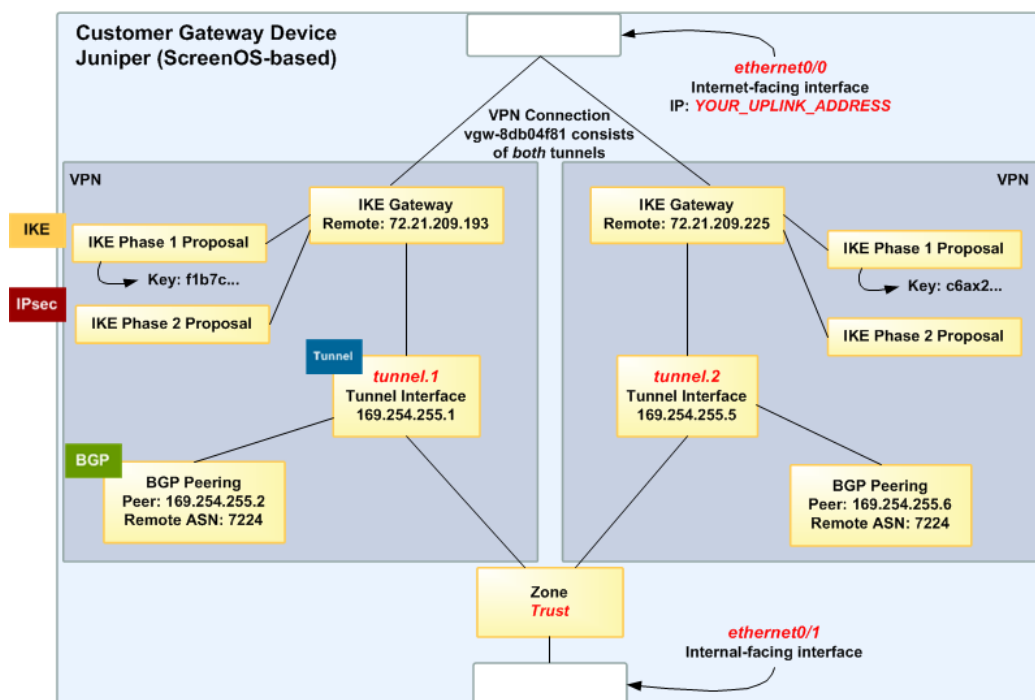
- ***YOUR\_UPLINK\_ADDRESS***—The IP address for the Internet-routable external interface on the customer gateway. The address must be static, and may be behind a device performing network address translation (NAT). To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500.
- ***YOUR\_BGP\_ASN***—The customer gateway's BGP ASN (we use 65000 by default)

The example configuration includes several example values to help you understand how configuration works. For example, we provide example values for the VPN connection ID (vpn-44a8938f), virtual private gateway ID (vgw-8db04f81), the IP addresses (72.21.209.\*, 169.254.255.\*), and the remote ASN (7224). You'll replace these example values with the actual values from the configuration information that you receive.

In addition, you must:

- Configure the outside interface (referred to as ***ethernet0/0*** in the example configuration).
- Configure the tunnel interface IDs (referred to as ***tunnel.1*** and ***tunnel.2*** in the example configuration).
- Configure all internal routing that moves traffic between the customer gateway and your local network.

In the following diagram and example configuration, you must replace the items in red italics with values that apply to your particular configuration.



### Warning

The following configuration information is an example of what you can expect your integration team to provide. Many of the values in the following example will be different from the configuration information that you receive. You must use the actual values and not the example values shown here, or your implementation will fail.

### Important

The configuration below is appropriate for ScreenOS versions 6.2 and later. You can download a configuration that is specific to ScreenOS version 6.1. In the **Download Configuration** dialog box, select **Juniper Networks, Inc.** from the **Vendor** list, **SSG and ISG Series Routers** from the **Platform** list, and **ScreenOS 6.1** from the **Software** list.

```
# Amazon Web Services
# Virtual Private Cloud
#
# AWS utilizes unique identifiers to manipulate the configuration of a VPN
# Connection. Each VPN Connection is assigned a VPN Connection Identifier
# and is associated with two other identifiers, namely the Customer Gateway
# Identifier and the Virtual Private Gateway Identifier.
#
# Your VPN Connection ID       : vpn-44a8938f
# Your Virtual Private Gateway ID : vgw-8db04f81
# Your Customer Gateway ID     : cgw-b4dc3961
#
# This configuration consists of two tunnels. Both tunnels must be configured
# on your Customer Gateway.
#
# This configuration was tested on a Juniper SSG-5 running ScreenOS 6.3R2.
#
# -----
# IPsec Tunnel #1
# -----
```

## IKE

```
# #1: Internet Key Exchange (IKE) Configuration
#
# A proposal is established for the supported IKE encryption, authentication,
# Diffie-Hellman, and lifetime parameters.
#
# Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
# and DH Group 2.
# You will need to modify these sample configuration files to take advantage of AES256,
# SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
# The address of the external interface for your customer gateway must be a static
# address.
# Your customer gateway may reside behind a device performing network address translation
# (NAT).
# To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
# unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
#
set ike p1-proposal ike-prop-vpn-44a8938f-1 preshare group2 esp aes128 sha-1 second 28800

# The IKE gateway is defined to be the Virtual Private Gateway. The gateway configuration
# associates a local interface, remote IP address, and IKE policy.
#
# This example shows the outside of the tunnel as interface ethernet0/0. This
# should be set to the interface that IP address YOUR_UPLINK_ADDRESS is
# associated with.
# This address is configured with the setup for your Customer Gateway.
#
#If the address changes, the Customer Gateway and VPN Connection must be recreated.
#

set ike gateway gw-vpn-44a8938f-1 address 72.21.209.225 id 72.21.209.225 main outgoing-
interface ethernet0/0 preshare "plain-text-password1" proposal ike-prop-vpn-44a8938f-1

# Troubleshooting IKE connectivity can be aided by enabling IKE debugging.
# To do so, run the following commands:
# clear dbuf          -- Clear debug buffer
# debug ike all       -- Enable IKE debugging
# get dbuf stream     -- View debug messages
# undebug all         -- Turn off debugging
```

## IPsec

```
# #2: IPsec Configuration
#
# The IPsec (Phase 2) proposal defines the protocol, authentication,
# encryption, and lifetime parameters for our IPsec security association.
# Please note, you may use these additionally supported IPsec parameters for encryption
# like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
#

set ike p2-proposal ipsec-prop-vpn-44a8938f-1 group2 esp aes128 sha-1 second 3600
set ike gateway gw-vpn-44a8938f-1 dpd-liveness interval 10
set vpn IPSEC-vpn-44a8938f-1 gateway gw-vpn-44a8938f-1 replay tunnel proposal ipsec-prop-
vpn-44a8938f-1
```

## Tunnel

```
# #3: Tunnel Interface Configuration
#
# The tunnel interface is configured with the internal IP address.
#
```

```
# To establish connectivity between your internal network and the VPC, you
# must have an interface facing your internal network in the "Trust" zone.
#

set interface tunnel.1 zone Trust
set interface tunnel.1 ip 169.254.255.2/30
set interface tunnel.1 mtu 1436
set vpn IPSEC-vpn-44a8938f-1 bind interface tunnel.1

# By default, the router will block asymmetric VPN traffic, which may occur
# with this VPN Connection. This occurs, for example, when routing policies
# cause traffic to sent from your router to VPC through one IPsec tunnel
# while traffic returns from VPC through the other.
#
# This command allows this traffic to be received by your device.

set zone Trust asymmetric-vpn

# This option causes the router to reduce the Maximum Segment Size of TCP
# packets to prevent packet fragmentation.
#

set flow vpn-tcp-mss 1387
```

## BGP

```
# #4: Border Gateway Protocol (BGP) Configuration
#
# BGP is used within the tunnel to exchange prefixes between the Virtual Private Gateway
# and your Customer Gateway. The Virtual Private Gateway will announce the prefix
# corresponding to your VPC.
#
# Your Customer Gateway may announce a default route (0.0.0.0/0).
#
# The BGP timers are adjusted to provide more rapid detection of outages.
#
# The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN) is configured
# as part of your Customer Gateway. If the ASN must be changed, the
# Customer Gateway and VPN Connection will need to be recreated with AWS.
#

set vrouter trust-vr
set max-ecmp-routes 2
set protocol bgp YOUR_BGP_ASN
set hold-time 30
set network 0.0.0.0/0
# To advertise additional prefixes to Amazon VPC, copy the 'network' statement and
# identify the prefix you wish to advertise (set ipv4 network X.X.X.X/X). Make sure the
# prefix is present in the routing table of the device with a valid next-hop.

set enable
set neighbor 169.254.255.1 remote-as 7224
set neighbor 169.254.255.1 enable
exit
exit
set interface tunnel.1 protocol bgp

# -----
# IPsec Tunnel #2
# -----
```

## IKE

```
# #1: Internet Key Exchange (IKE) Configuration
#
# A proposal is established for the supported IKE encryption, authentication,
# Diffie-Hellman, and lifetime parameters.
# Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
# and DH Group 2.
# You will need to modify these sample configuration files to take advantage of AES256,
# SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
# The address of the external interface for your customer gateway must be a static
# address.
# Your customer gateway may reside behind a device performing network address translation
# (NAT).
# To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall !rules
# to unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
#

set ike p1-proposal ike-prop-vpn-44a8938f-2 preshare group2 esp aes128 sha-1 second 28800

# The IKE gateway is defined to be the Virtual Private Gateway. The gateway configuration
# associates a local interface, remote IP address, and IKE policy.
#
# This example shows the outside of the tunnel as interface ethernet0/0. This
# should be set to the interface that IP address YOUR_UPLINK_ADDRESS is
# associated with.
#
# This address is configured with the setup for your Customer Gateway. If the
# address changes, the Customer Gateway and VPN Connection must be recreated.
#

set ike gateway gw-vpn-44a8938f-2 address 72.21.209.193 id 72.21.209.193 main outgoing-
interface ethernet0/0 preshare "plain-text-password2" proposal ike-prop-vpn-44a8938f-2

# Troubleshooting IKE connectivity can be aided by enabling IKE debugging.
# To do so, run the following commands:
# clear dbuf          -- Clear debug buffer
# debug ike all       -- Enable IKE debugging
# get dbuf stream     -- View debug messages
# undebug all         -- Turn off debugging
```

## IPsec

```
# #2: IPsec Configuration
#
# The IPsec (Phase 2) proposal defines the protocol, authentication,
# encryption, and lifetime parameters for our IPsec security association.
# Please note, you may use these additionally supported IPsec parameters for encryption
# like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
#

set ike p2-proposal ipsec-prop-vpn-44a8938f-2 group2 esp aes128 sha-1 second 3600
set ike gateway gw-vpn-44a8938f-2 dpd-liveness interval 10
set vpn IPSEC-vpn-44a8938f-2 gateway gw-vpn-44a8938f-2 replay tunnel proposal ipsec-prop-
vpn-44a8938f-2
```

## Tunnel

```
# #3: Tunnel Interface Configuration
#
# The tunnel interface is configured with the internal IP address.
#
# To establish connectivity between your internal network and the VPC, you
```

```
# must have an interface facing your internal network in the "Trust" zone.

set interface tunnel.2 zone Trust
set interface tunnel.2 ip 169.254.255.6/30
set interface tunnel.2 mtu 1436
set vpn IPSEC-vpn-44a8938f-2 bind interface tunnel.2

# By default, the router will block asymmetric VPN traffic, which may occur
# with this VPN Connection. This occurs, for example, when routing policies
# cause traffic to sent from your router to VPC through one IPsec tunnel
# while traffic returns from VPC through the other.
#
# This command allows this traffic to be received by your device.

set zone Trust asymmetric-vpn

# This option causes the router to reduce the Maximum Segment Size of TCP
# packets to prevent packet fragmentation.

set flow vpn-tcp-mss 1387
```

## BGP

```
# #4: Border Gateway Protocol (BGP) Configuration
#
# BGP is used within the tunnel to exchange prefixes between the Virtual Private Gateway
# and your Customer Gateway. The Virtual Private Gateway will announce the prefix
# corresponding to your VPC.
#
# Your Customer Gateway may announce a default route (0.0.0.0/0).
#
# The BGP timers are adjusted to provide more rapid detection of outages.
#
# The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN) is configured
# as part of your Customer Gateway. If the ASN must be changed, the
# Customer Gateway and VPN Connection will need to be recreated with AWS.
#

set vrouter trust-vr
set max-ecmp-routes 2
set protocol bgp YOUR_BGP_ASN
set hold-time 30
set network 0.0.0.0/0
# To advertise additional prefixes to Amazon VPC, copy the 'network' statement and
# identify the prefix you wish to advertise (set ipv4 network X.X.X.X/X). Make sure the
# prefix is present in the routing table of the device with a valid next-hop.
set enable
set neighbor 169.254.255.5 remote-as 7224
set neighbor 169.254.255.5 enable
exit
exit
set interface tunnel.2 protocol bgp
```

## How to Test the Customer Gateway Configuration

You can test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

1. On your customer gateway, determine whether the BGP status is Active.



It takes approximately 30 seconds for a BGP peering to become active.

2. Ensure that the customer gateway is advertising a route to the virtual private gateway. The route may be the default route (0.0.0.0/0) or a more specific route you prefer.

When properly established, your BGP peering should be receiving one route from the virtual private gateway corresponding to the prefix that your VPC integration team specified for the VPC (for example, 10.0.0.0/24). 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.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection: your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the Amazon EC2 Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

#### Note

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.

If your tunnels don't test successfully, see [Troubleshooting Juniper ScreenOS Customer Gateway Connectivity \(p. 174\)](#).

# Example: Netgate PfSense Device without Border Gateway Protocol

## Topics

- [A High-Level View of the Customer Gateway \(p. 118\)](#)
- [Example Configuration \(p. 119\)](#)
- [How to Test the Customer Gateway Configuration \(p. 122\)](#)

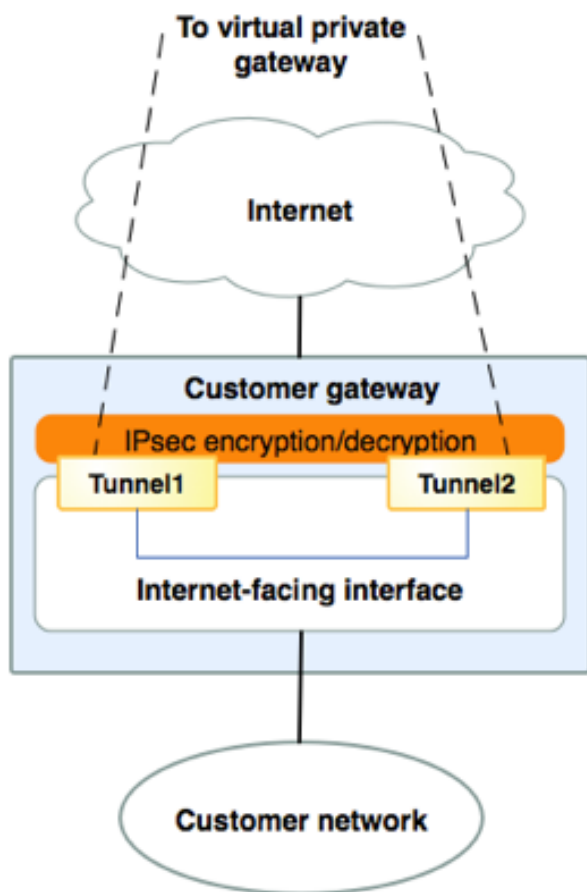
This topic provides an example of how to configure your router if your customer gateway is a Netgate pfSense firewall running OS 2.2.5 or later.

This topic assumes that you've configured a VPN connection with static routing in the Amazon VPC console. For more information, see [Adding a Hardware Virtual Private Gateway to Your VPC](#) in the *Amazon VPC User Guide*.

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels: *Tunnel 1* and *Tunnel 2*. Using redundant tunnels ensures continuous availability in the case that a device fails.

You should use the real configuration information that you receive from your integration team and apply it to your customer gateway.



## Example Configuration

The example configuration includes several example values to help you understand how configuration works. For example, we provide example values for the VPN connection ID (vpn-12345678), virtual private gateway ID (vgw-12345678), and placeholders for the AWS endpoints (AWS\_ENDPOINT\_1 and AWS\_ENDPOINT\_2).

In the following example configuration, you must replace the items in red italics with values that apply to your particular configuration.

### Important

The following configuration information is an example of what you can expect an integration team to provide. Many of the values in the following example will be different from the actual configuration information that you receive. You must use the actual values and not the example values shown here, or your implementation will fail.

```
! Amazon Web Services
! Virtual Private Cloud

! AWS utilizes unique identifiers to manipulate the configuration of
! a VPN Connection. Each VPN Connection is assigned an identifier and is
! associated with two other identifiers, namely the
! Customer Gateway Identifier and Virtual Private Gateway Identifier.
!
```

```
! Your VPN Connection ID      : vpn-12345678
! Your Virtual Private Gateway ID : vgw-12345678
! Your Customer Gateway ID    : cgw-12345678
!
!
! This configuration consists of two tunnels. Both tunnels must be
! configured on your Customer Gateway for redundancy.
!
! -----
! IPsec Tunnel #1
! -----
! #1: Internet Key Exchange (IKE) Configuration
!
! A policy is established for the supported ISAKMP encryption, authentication, Diffie-
! Hellman, lifetime,
! and key parameters. The IKE peer is configured with the supported IKE encryption,
! authentication, Diffie-Hellman, lifetime, and key
! parameters. Please note, these sample configurations are for the minimum requirement of
! AES128, SHA1, and DH Group 2.
! You will need to modify these sample configuration files to take advantage of AES256,
! SHA256, or other DH
! groups like 2, 14-18, 22, 23, and 24. The address of the external interface for your
! customer gateway must be a static address.
! Your customer gateway may reside behind a device performing network address translation
! (NAT). To
! ensure that NAT traversal (NAT-T) can function, you must adjust your firewall
! rules to unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
!
! Go to VPN-->IPsec. Add a new Phase1 entry (click + button )

General information
a. Disabled : uncheck
b. Key Exchange version : V1
c. Internet Protocol : IPv4
d. Interface : WAN
e. Remote Gateway: AWS_ENDPOINT_1
f. Description: Amazon-IKE-vpn-12345678-0

Phase 1 proposal (Authentication)
a. Authentication Method: Mutual PSK
b. Negotiation mode : Main
c. My identifier : My IP address
d. Peer identifier : Peer IP address
e. Pre-Shared Key: plain-text-password1

Phase 1 proposal (Algorithms)
a. Encryption algorithm : aes128
b. Hash algorithm : sha1
c. DH key group : 2
d. Lifetime : 28800 seconds

Advanced Options
a. Disable Rekey : uncheck
b. Responder Only : uncheck
c. NAT Traversal : Auto
d. Dead Peer Detection : Enable DPD
    Delay between requesting peer acknowledgement : 10 seconds
    Number of consecutive failures allowed before disconnect : 3 retries

! #2: IPsec Configuration
!
! The IPsec transform set defines the encryption, authentication, and IPsec
! mode parameters.
```

! Please note, you may use these additionally supported IPSec parameters for encryption like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.

Expand the VPN configuration clicking in "+" and then create a new Phase2 entry as follows:

- a. Disabled : unchecked
- b. Mode : Tunnel
- c. Local Network : Type: LAN subnet  
Address : ! Enter your local network CIDR in the Address tab
- d. Remote Network : Type : Network  
Address : ! Enter your remote network CIDR in the Address tab
- e. Description : Amazon-IPSec-*vpn-12345678-0*

Phase 2 proposal (SA/Key Exchange)

- a. Protocol : ESP
- b. Encryption algorithms : aes128
- c. Hash algorithms : sha1
- d. PFS key group : 2
- e. Lifetime : 3600 seconds

Advanced Options

Automatically ping host : ! Provide the IP address of an EC2 instance in VPC that will respond to ICMP.

! -----

! -----

! IPSec Tunnel #2

! -----

! #1: Internet Key Exchange (IKE) Configuration

!

! A policy is established for the supported ISAKMP encryption, authentication, Diffie-Hellman, lifetime,

! and key parameters. The IKE peer is configured with the supported IKE encryption, authentication, Diffie-Hellman, lifetime, and key

! parameters. Please note, these sample configurations are for the minimum requirement of AES128, SHA1, and DH Group 2.

! You will need to modify these sample configuration files to take advantage of AES256, SHA256, or other DH

! groups like 2, 14-18, 22, 23, and 24. The address of the external interface for your customer gateway must be a static address.

! Your customer gateway may reside behind a device performing network address translation (NAT). To

! ensure that NAT traversal (NAT-T) can function, you must adjust your firewall

! rules to unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.

!

!

Go to VPN-->IPSec. Add a new Phase1 entry (click + button )

General information

- a. Disabled : unchecked
- b. Key Exchange version : V1
- c. Internet Protocol : IPv4
- d. Interface : WAN
- e. Remote Gateway: *AWS\_ENPOINT\_2*
- f. Description: Amazon-IKE-*vpn-12345678-1*

Phase 1 proposal (Authentication)

- a. Authentication Method: Mutual PSK
- b. Negotiation mode : Main
- c. My identifier : *My IP address*
- d. Peer identifier : *Peer IP address*
- e. Pre-Shared Key: *plain-text-password2*

```
Phase 1 proposal (Algorithms)
a. Encryption algorithm : aes128
b. Hash algorithm : sha1
c. DH key group : 2
d. Lifetime : 28800 seconds

Advanced Options
a. Disable Rekey : uncheck
b. Responder Only : uncheck
c. NAT Traversal : Auto
d. Dead Peer Detection : Enable DPD
    Delay between requesting peer acknowledgement : 10 seconds
Number of consecutive failures allowed before disconnect : 3 retries

! #2: IPSec Configuration
!
! The IPSec transform set defines the encryption, authentication, and IPSec
! mode parameters.
! Please note, you may use these additionally supported IPSec parameters for encryption
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.

Expand the VPN configuration clicking in "+" and then create a new Phase2 entry as follows:

a. Disabled :uncheck
b. Mode : Tunnel
c. Local Network : Type: LAN subnet
    Address : ! Enter your local network CIDR in the Address tab
d. Remote Network : Type : Network
    Address : ! Enter your remote network CIDR in the Address tab
e. Description : Amazon-IPSec-vpn-12345678-1

Phase 2 proposal (SA/Key Exchange)
a. Protocol : ESP
b. Encryption algorithms :aes128
c. Hash algorithms : sha1
d. PFS key group : 2
e. Lifetime : 3600 seconds

Advanced Options

Automatically ping host : ! Provide the IP address of an EC2 instance in VPC that will
respond to ICMP.
```

## How to Test the Customer Gateway Configuration

You must first test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

- In the Amazon VPC console, ensure that a static route has been added to the VPN connection so that traffic can get back to your customer gateway. For example, if your local subnet prefix is 198.10.0.0/16, you must add a static route with that CIDR range to your VPN connection. Make sure that both tunnels have a static route to your VPC.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection - your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance from one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are available in the Quick Start menu when you use the Launch Instances Wizard in the Amazon EC2 console. For more information, see [Launching an Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

#### Note

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.



# Example: Palo Alto Networks Device

## Topics

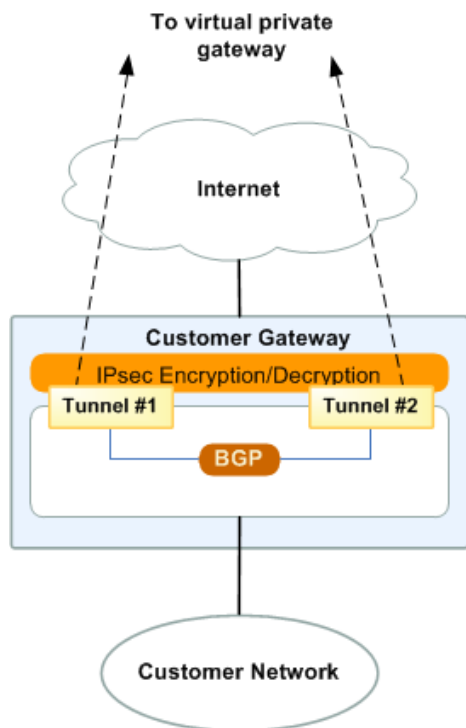
- [A High-Level View of the Customer Gateway \(p. 125\)](#)
- [A Detailed View of the Customer Gateway and an Example Configuration \(p. 125\)](#)
- [How to Test the Customer Gateway Configuration \(p. 132\)](#)

The following topic provides example configuration information provided by your integration team if your customer gateway is a Palo Alto Networks PANOS 4.1.2+ device.

Two diagrams illustrate the example configuration. The first diagram shows the high-level layout of the customer gateway, and the second diagram shows the details of the example configuration. You should use the real configuration information that you receive from your integration team and apply it to your customer gateway.

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels. Using redundant tunnels ensures continuous availability in the case that a device fails.



## A Detailed View of the Customer Gateway and an Example Configuration

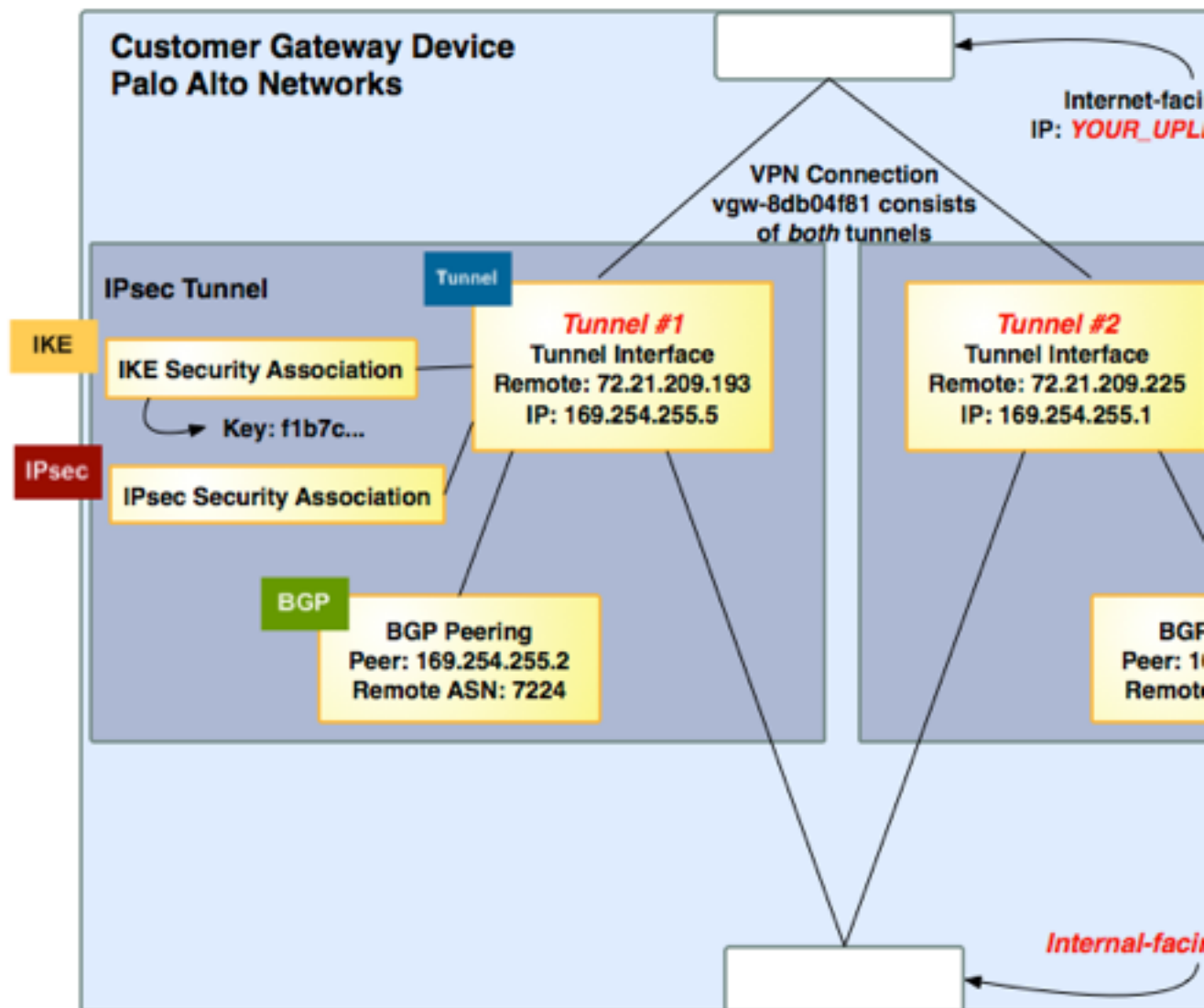
The diagram in this section illustrates an example Palo Alto customer gateway. Following the diagram, there is a corresponding example of the configuration information your integration team should provide. The example configuration contains a set of information for each of the tunnels that you must configure.

In addition, the example configuration refers to these items that you must provide:

- ***YOUR\_UPLINK\_ADDRESS***—The IP address for the Internet-routable external interface on the customer gateway (which must be static, and may be behind a device performing network address translation (NAT); however, NAT traversal (NAT-T) is not supported).
- ***YOUR\_BGP\_ASN***—The customer gateway's BGP ASN (we use 65000 by default)

The example configuration includes several example values to help you understand how configuration works. For example, we provide example values for the VPN connection ID (vpn-44a8938f), virtual private gateway ID (vgw-8db04f81), the IP addresses (72.21.209.\*, 169.254.255.\*), and the remote ASN (7224). You'll replace these example values with the actual values from the configuration information that you receive.

In the following diagram and example configuration, you must replace the items in red italics with values that apply to your particular configuration.



### Warning

The following configuration information is an example of what you can expect your integration team to provide. Many of the values in the following example will be different from the actual configuration information that you receive. You must use the actual values and not the example values shown here, or your implementation will fail.

```
! Amazon Web Services
! Virtual Private Cloud

! AWS utilizes unique identifiers to manipulate the configuration of
! a VPN Connection. Each VPN Connection is assigned an identifier and is
! associated with two other identifiers, namely the
! Customer Gateway Identifier and Virtual Private Gateway Identifier.
!
! Your VPN Connection ID      : vpn-44a8938f
! Your Virtual Private Gateway ID : vgw-8db04f81
! Your Customer Gateway ID    : cgw-b4dc3961
```

```
!  
!  
! This configuration consists of two tunnels. Both tunnels must be  
! configured on your Customer Gateway.  
!  
! -----  
! IPsec Tunnel #1  
! -----
```

## IKE

```
! #1: Internet Key Exchange (IKE) Configuration  
!  
! A policy is established for the supported ISAKMP encryption,  
! authentication, Diffie-Hellman, lifetime, and key parameters.  
! Please note, these sample configurations are for the minimum requirement of AES128, SHA1,  
! and DH Group 2.  
! You will need to modify these sample configuration files to take advantage of AES256,  
! SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.  
! The address of the external interface for your customer gateway must be a static  
! address.  
! Your customer gateway may reside behind a device performing network address translation  
! (NAT).  
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to  
! unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.  
!  
configure  
edit network ike crypto-profiles ike-crypto-profiles ike-crypto-vpn-44a8938f-0  
  set dh-group group2  
  set hash sha1  
  set lifetime seconds 28800  
  set encryption aes128  
top  
  
edit network ike gateway ike-vpn-44a8938f-0  
  set protocol ikev1 dpd interval 10 retry 3 enable yes  
  set protocol ikev1 ike-crypto-profile ike-crypto-vpn-44a8938f-0 exchange-mode main  
  set authentication pre-shared-key key plain-text-password1  
  set local-address ip YOUR_UPLINK_ADDRESS  
  set local-address interface ethernet1/1  
  set peer-address ip 72.21.209.193  
top
```

## IPsec

```
! #2: IPsec Configuration  
!  
! The IPsec transform set defines the encryption, authentication, and IPsec  
! mode parameters.  
!  
! Please note, you may use these additionally supported IPsec parameters for encryption  
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.  
  
edit network ike crypto-profiles ipsec-crypto-profiles ipsec-vpn-44a8938f-0  
  set esp authentication sha1  
  set esp encryption aes128  
  set dh-group group2 lifetime seconds 3600  
top
```

## Tunnel

```
! -----
! #3: Tunnel Interface Configuration
!
! A tunnel interface is configured to be the logical interface associated
! with the tunnel. All traffic routed to the tunnel interface will be
! encrypted and transmitted to the VPC. Similarly, traffic from the VPC
! will be logically received on this interface.
!
! Association with the IPSec security association is done through the
! "tunnel protection" command.
!
! The address of the interface is configured with the setup for your
! Customer Gateway. If the address changes, the Customer Gateway and VPN
! Connection must be recreated with Amazon VPC.
!

edit network interface tunnel
  set ip 169.254.255.5/30
  set units tunnel.1
  set mtu 1427
top

edit network tunnel ipsec ipsec-tunnel-1
  set auto-key ike-gateway ike-vpn-44a8938f-0
  set auto-key ipsec-crypto-profile ipsec-vpn-44a8938f-0
  set tunnel-interface tunnel.1
  set anti-replay yes
```

## BGP

```
! -----
! #4: Border Gateway Protocol (BGP) Configuration
!
! BGP is used within the tunnel to exchange prefixes between the
! Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway
! will announce the prefix corresponding to your VPC.
!
!
! The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN)
! is configured as part of your Customer Gateway. If the ASN must
! be changed, the Customer Gateway and VPN Connection will need to be recreated with AWS.
!

edit network virtual-router default protocol bgp
  set enable yes
  set router-id YOUR_UPLINK_ADDRESS
  set local-as YOUR_BGP_ASN
  edit peer-group AmazonBGP
    edit peer amazon-tunnel-44a8938f-0
      set connection-options keep-alive-interval 10
      set connection-options hold-time 30
    set enable yes
```

```
set local-address ip 169.254.255.5/30
set local-address interface tunnel.1
set peer-as 7224
set peer-address ip 169.254.255.2
top
```

! Your Customer Gateway may announce a default route (0.0.0.0/0) to us.

```
edit network virtual-router default protocol bgp policy
set export rules vr-export action allow
set match address-prefix 0.0.0.0/0 exact yes
set used-by AmazonBGP enable yes
top
```

! To advertise additional prefixes to Amazon VPC, add these prefixes to the 'address-prefix'

! statement and identify the prefix you wish to advertise. Make sure the prefix is present  
! in the routing table of the device with a valid next-hop. If you want to advertise  
! 192.168.0.0/16 to Amazon, this can be done using the following.

```
edit network virtual-router default protocol bgp policy
set export rules vr-export action allow
set match address-prefix 192.168.0.0/16 exact yes
set used-by AmazonBGP enable yes
top
```

!

## IKE

```
! -----
! IPSec Tunnel #2
! -----
! #1: Internet Key Exchange (IKE) Configuration
!
! A policy is established for the supported ISAKMP encryption,
! authentication, Diffie-Hellman, lifetime, and key parameters.
! Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
! and DH Group 2.
! You will need to modify these sample configuration files to take advantage of AES256,
! SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
! The address of the external interface for your customer gateway must be a static
! address.
! Your customer gateway may reside behind a device performing network address translation
! (NAT).
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall !rules
! to unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
```

```
configure
edit network ike crypto-profiles ike-crypto-profiles ike-crypto-vpn-44a8938f-1
set dh-group group2
set hash sha1
set lifetime seconds 28800
set encryption aes128
top
```

```
edit network ike gateway ike-vpn-44a8938f-1
  set protocol ikev1 dpd interval 10 retry 3 enable yes
  set protocol ikev1 ike-crypto-profile ike-crypto-vpn-35a6445c-1 exchange-mode main
  set authentication pre-shared-key key plain-text-password2
  set local-address ip YOUR_UPLINK_ADDRESS
  set local-address interface ethernet1/1
  set peer-address ip 72.21.209.225
top
```

## IPsec

```
! #2: IPsec Configuration
!
! The IPsec transform set defines the encryption, authentication, and IPsec
! mode parameters.
!
! Please note, you may use these additionally supported IPsec parameters for encryption
! like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.

edit network ike crypto-profiles ipsec-crypto-profiles ipsec-vpn-44a8938f-1
  set esp authentication sha1
  set esp encryption aes128
  set dh-group group2 lifetime seconds 3600
top
```

## Tunnel

```
! -----
! #3: Tunnel Interface Configuration
!
! A tunnel interface is configured to be the logical interface associated
! with the tunnel. All traffic routed to the tunnel interface will be
! encrypted and transmitted to the VPC. Similarly, traffic from the VPC
! will be logically received on this interface.
!
! Association with the IPsec security association is done through the
! "tunnel protection" command.
!
! The address of the interface is configured with the setup for your
! Customer Gateway. If the address changes, the Customer Gateway and VPN
! Connection must be recreated with Amazon VPC.
!

edit network interface tunnel
  set ip 169.254.255.1/30
  set units tunnel.2
  set mtu 1427
top

edit network tunnel ipsec ipsec-tunnel-2
  set auto-key ike-gateway ike-vpn-44a8938f-1
  set auto-key ipsec-crypto-profile ipsec-vpn-44a8938f-1
  set tunnel-interface tunnel.2
  set anti-replay yes
```

## BGP

```
! #4: Border Gateway Protocol (BGP) Configuration
!
! BGP is used within the tunnel to exchange prefixes between the
! Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway
! will announce the prefix corresponding to your VPC.
!
!
! The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN)
! is configured as part of your Customer Gateway. If the ASN must
! be changed, the Customer Gateway and VPN Connection will need to be recreated with AWS.
!

edit network virtual-router default protocol bgp
set enable yes
set router-id YOUR_UPLINK_ADDRESS
set local-as YOUR_BGP_ASN
edit peer-group AmazonBGP
edit peer amazon-tunnel-4a8938f-1
set connection-options keep-alive-interval 10
set connection-options hold-time 30
set enable yes
set local-address ip 169.254.255.1/30
set local-address interface tunnel.2
set peer-as 7224
set peer-address ip 169.254.255.6.113
top

! Your Customer Gateway may announce a default route (0.0.0.0/0) to us.

edit network virtual-router default protocol bgp policy
set export rules vr-export action allow
set match address-prefix 0.0.0.0/0 exact yes
set used-by AmazonBGP enable yes
top

! To advertise additional prefixes to Amazon VPC, add these prefixes to the 'address-
prefix'
! statement and identify the prefix you wish to advertise. Make sure the prefix is present
! in the routing table of the device with a valid next-hop. If you want to advertise
! 192.168.0.0/16 to Amazon, this can be done using the following.

edit network virtual-router default protocol bgp policy
set export rules vr-export action allow
set match address-prefix 192.168.0.0/16 exact yes
set used-by AmazonBGP enable yes
top

!
```



# How to Test the Customer Gateway Configuration

You can test the gateway configuration for each tunnel.

## To test the customer gateway configuration for each tunnel

1. On your customer gateway, determine whether the BGP status is **Active**.

It takes approximately 30 seconds for a BGP peering to become active.

2. Ensure that the customer gateway is advertising a route to the virtual private gateway. The route may be the default route (0.0.0.0/0) or a more specific route you prefer.

When properly established, your BGP peering should be receiving one route from the virtual private gateway corresponding to the prefix that your VPC integration team specified for the VPC (for example, 10.0.0.0/24). 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.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection: your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

## To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the Amazon EC2 Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

**Note**

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.

# Example: Yamaha Device

## Topics

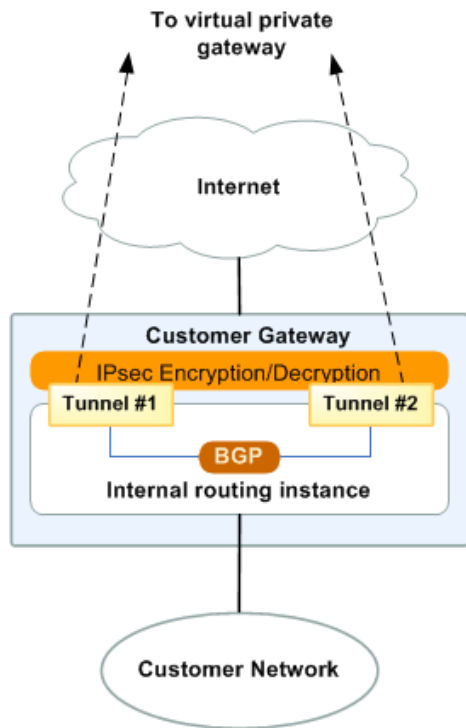
- [A High-Level View of the Customer Gateway \(p. 135\)](#)
- [A Detailed View of the Customer Gateway and an Example Configuration \(p. 135\)](#)
- [How to Test the Customer Gateway Configuration \(p. 141\)](#)

In this section we walk you through an example of the configuration information provided by your integration team if your customer gateway is a Yamaha RT107e, RTX1200, RTX1210, RTX1500, RTX3000, or SRT100 router.

Two diagrams illustrate the example configuration. The first diagram shows the high-level layout of the customer gateway, and the second diagram shows the details of the example configuration. You should use the real configuration information that you receive from your integration team and apply it to your customer gateway.

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels. Using redundant tunnels ensures continuous availability in the case that a device fails.



## A Detailed View of the Customer Gateway and an Example Configuration

The diagram in this section illustrates an example Yamaha customer gateway. Following the diagram, there is a corresponding example of the configuration information your integration team should provide. The example configuration contains a set of information for each of the tunnels that you must configure.

In addition, the example configuration refers to these items that you must provide:

- ***YOUR\_UPLINK\_ADDRESS***—The IP address for the Internet-routable external interface on the customer gateway. The address must be static, and may be behind a device performing network address translation (NAT). To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500.
- ***YOUR\_LOCAL\_NETWORK\_ADDRESS***—The IP address that is assigned to the LAN interface connected to your local network (most likely a private address such as 192.168.0.1)
- ***YOUR\_BGP\_ASN***—The customer gateway's BGP ASN (we use 65000 by default)

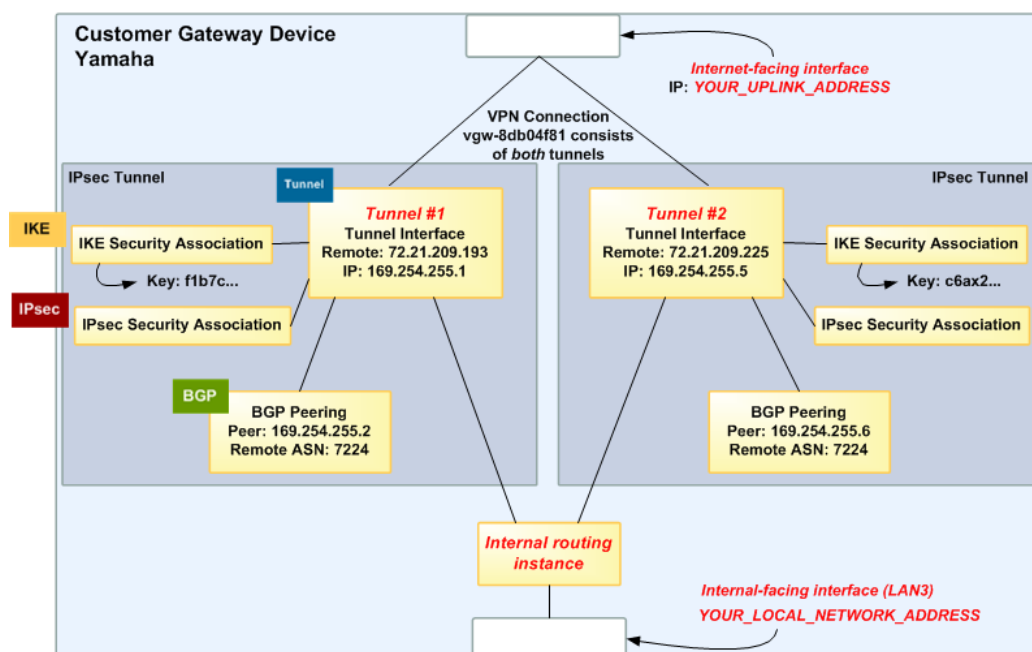
The example configuration includes several example values to help you understand how configuration works. For example, we provide example values for the VPN connection ID (vpn-44a8938f), virtual private gateway ID (vgw-8db04f81), the IP addresses (72.21.209.\*, 169.254.255.\*), and the remote ASN

(7224). You'll replace these example values with the actual values from the configuration information that you receive.

In addition, you must also:

- Configure the outside interface (referred to as *LAN3* in the example configuration).
- Configure the tunnel interface IDs (referred to as *Tunnel #1* and *Tunnel #2* in the example configuration).
- Configure all internal routing that moves traffic between the customer gateway and your local network.

In the following diagram and example configuration, you must replace the items in red italics with values that apply to your particular configuration.



### Warning

The following configuration information is an example of what you can expect your integration team to provide. Many of the values in the following example will be different from the actual configuration information that you receive. You must use the actual values and not the example values shown here, or your implementation will fail.

```
# Amazon Web Services
# Virtual Private Cloud

# AWS utilizes unique identifiers to manage the configuration of
# a VPN Connection. Each VPN Connection is assigned an identifier and is
# associated with two other identifiers, namely the
# Customer Gateway Identifier and Virtual Private Gateway Identifier.
#
# Your VPN Connection ID           : vpn-44a8938f
# Your Virtual Private Gateway ID  : vgw-8db04f81
# Your Customer Gateway ID         : cgw-b4dc3961
#
#
# This configuration consists of two tunnels. Both tunnels must be
# configured on your Customer Gateway.
```

```
#  
# -----  
# IPsec Tunnel #1  
# -----
```

## IKE

```
# #1: Internet Key Exchange (IKE) Configuration  
#  
# A policy is established for the supported ISAKMP encryption,  
# authentication, Diffie-Hellman, lifetime, and key parameters.  
#  
# Please note, these sample configurations are for the minimum requirement of AES128, SHA1,  
# and DH Group 2.  
# You will need to modify these sample configuration files to take advantage of AES256,  
# SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.  
# The address of the external interface for your customer gateway must be a static  
# address.  
# Your customer gateway may reside behind a device performing network address translation  
# (NAT).  
# To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall !rules  
# to unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.  
#  
tunnel select 1  
ipsec ike encryption 1 aes-cbc  
ipsec ike group 1 modp1024  
ipsec ike hash 1 sha  
  
# This line stores the Pre Shared Key used to authenticate the  
# tunnel endpoints.  
#  
ipsec ike pre-shared-key 1 text plain-text-password1
```

## IPsec

```
# #2: IPsec Configuration  
  
# The IPsec policy defines the encryption, authentication, and IPsec  
# mode parameters.  
# Please note, you may use these additionally supported IPSec parameters for encryption  
# like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.  
#  
# Note that there are a global list of IPSec policies, each identified by  
# sequence number. This policy is defined as #201, which may conflict with  
# an existing policy using the same number. If so, we recommend changing  
# the sequence number to avoid conflicts.  
#  
  
ipsec tunnel 201  
ipsec sa policy 201 1 esp aes-cbc sha-hmac  
  
# The IPsec profile references the IPsec policy and further defines  
# the Diffie-Hellman group and security association lifetime.  
  
ipsec ike duration ipsec-sa 1 3600  
ipsec ike pfs 1 on  
  
# Additional parameters of the IPsec configuration are set here. Note that  
# these parameters are global and therefore impact other IPsec  
# associations.  
# This option instructs the router to clear the "Don't Fragment"
```

```
# bit from packets that carry this bit and yet must be fragmented, enabling
# them to be fragmented.
#
ipsec tunnel outer df-bit clear

# This option enables IPsec Dead Peer Detection, which causes periodic
# messages to be sent to ensure a Security Association remains operational.

ipsec ike keepalive use 1 on dpd 10 3
```

### Tunnel

```
# -----
# #3: Tunnel Interface Configuration
#
# A tunnel interface is configured to be the logical interface associated
# with the tunnel. All traffic routed to the tunnel interface will be
# encrypted and transmitted to the VPC. Similarly, traffic from the VPC
# will be logically received on this interface.
#
#
# The address of the interface is configured with the setup for your
# Customer Gateway. If the address changes, the Customer Gateway and VPN
# Connection must be recreated with Amazon VPC.
#
ipsec ike local address 1 YOUR_LOCAL_NETWORK_ADDRESS
ipsec ike remote address 1 72.21.209.225
ip tunnel address 169.254.255.2/30
ip tunnel remote address 169.254.255.1

# This option causes the router to reduce the Maximum Segment Size of
# TCP packets to prevent packet fragmentation

ip tunnel tcp mss limit 1387
tunnel enable 1
tunnel select none
ipsec auto refresh on
```

### BGP

```
# -----
# #4: Border Gateway Protocol (BGP) Configuration
#
# BGP is used within the tunnel to exchange prefixes between the
# Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway
# will announce the prefix corresponding to your VPC.
#
# Your Customer Gateway may announce a default route (0.0.0.0/0),
# which can be done with the 'network' and 'default-originate' statements.
#
# The BGP timers are adjusted to provide more rapid detection of outages.
#
# The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN) is configured
# as part of your Customer Gateway. If the ASN must be changed, the
# Customer Gateway and VPN Connection will need to be recreated with AWS.
#
bgp use on
bgp autonomous-system YOUR_BGP_ASN
bgp neighbor 1 7224 169.254.255.1 hold-time=30 local-address=169.254.255.2

# To advertise additional prefixes to Amazon VPC, copy the 'network' statement and
```

```
# identify the prefix you wish to advertise. Make sure the
# prefix is present in the routing table of the device with a valid next-hop.
# For example, the following two lines will advertise 192.168.0.0/16 and 10.0.0.0/16 to
  Amazon VPC
#
# bgp import filter 1 equal 10.0.0.0/16
# bgp import filter 1 equal 192.168.0.0/16
#
bgp import filter 1 equal 0.0.0.0/0
bgp import 7224 static filter 1
```

## IKE

```
# -----
# IPsec Tunnel #2
# -----

# #1: Internet Key Exchange (IKE) Configuration
#
# A policy is established for the supported ISAKMP encryption,
# authentication, Diffie-Hellman, lifetime, and key parameters.
#
# Please note, these sample configurations are for the minimum requirement of AES128, SHA1,
# and DH Group 2.
# You will need to modify these sample configuration files to take advantage of AES256,
# SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
# The address of the external interface for your customer gateway must be a static
# address.
# Your customer gateway may reside behind a device performing network address translation
# (NAT).
# To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to
# unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
#
tunnel select 2
ipsec ike encryption 2 aes-cbc
ipsec ike group 2 modp1024
ipsec ike hash 2 sha

# This line stores the Pre Shared Key used to authenticate the
# tunnel endpoints.
#
ipsec ike pre-shared-key 2 text plain-text-password2
```

## IPsec

```
# #2: IPsec Configuration

# The IPsec policy defines the encryption, authentication, and IPsec
# mode parameters.
# Please note, you may use these additionally supported IPsec parameters for encryption
# like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.
#
# Note that there are a global list of IPsec policies, each identified by
# sequence number. This policy is defined as #202, which may conflict with
# an existing policy using the same number. If so, we recommend changing
# the sequence number to avoid conflicts.
#
ipsec tunnel 202
```



```
ipsec sa policy 202 2 esp aes-cbc sha-hmac

# The IPsec profile references the IPsec policy and further defines
# the Diffie-Hellman group and security association lifetime.

ipsec ike duration ipsec-sa 2 3600
ipsec ike pfs 2 on

# Additional parameters of the IPsec configuration are set here. Note that
# these parameters are global and therefore impact other IPsec
# associations.
# This option instructs the router to clear the "Don't Fragment"
# bit from packets that carry this bit and yet must be fragmented, enabling
# them to be fragmented.
#
ipsec tunnel outer df-bit clear

# This option enables IPsec Dead Peer Detection, which causes periodic
# messages to be sent to ensure a Security Association remains operational.

ipsec ike keepalive use 2 on dpd 10 3
```

#### Tunnel

```
# -----
# #3: Tunnel Interface Configuration
#
# A tunnel interface is configured to be the logical interface associated
# with the tunnel. All traffic routed to the tunnel interface will be
# encrypted and transmitted to the VPC. Similarly, traffic from the VPC
# will be logically received on this interface.
#
# Association with the IPsec security association is done through the
# "tunnel protection" command.
#
# The address of the interface is configured with the setup for your
# Customer Gateway. If the address changes, the Customer Gateway and VPN
# Connection must be recreated with Amazon VPC.
#
ipsec ike local address 2 YOUR_LOCAL_NETWORK_ADDRESS
ipsec ike remote address 2 72.21.209.193
ip tunnel address 169.254.255.6/30
ip tunnel remote address 169.254.255.5

# This option causes the router to reduce the Maximum Segment Size of
# TCP packets to prevent packet fragmentation

ip tunnel tcp mss limit 1387
tunnel enable 2
tunnel select none
ipsec auto refresh on
```

#### BGP

```
# -----
# #4: Border Gateway Protocol (BGP) Configuration
#
# BGP is used within the tunnel to exchange prefixes between the
# Virtual Private Gateway and your Customer Gateway. The Virtual Private Gateway
# will announce the prefix corresponding to your VPC.
#
```

```
# Your Customer Gateway may announce a default route (0.0.0.0/0),
# which can be done with the 'network' and 'default-originate' statements.
#
#
# The BGP timers are adjusted to provide more rapid detection of outages.
#
# The local BGP Autonomous System Number (ASN) (YOUR_BGP_ASN) is configured
# as part of your Customer Gateway. If the ASN must be changed, the
# Customer Gateway and VPN Connection will need to be recreated with AWS.
#
bgp use on
bgp autonomous-system YOUR_BGP_ASN
bgp neighbor 2 7224 169.254.255.5 hold-time=30 local-address=169.254.255.6

# To advertise additional prefixes to Amazon VPC, copy the 'network' statement and
# identify the prefix you wish to advertise. Make sure the
# prefix is present in the routing table of the device with a valid next-hop.
# For example, the following two lines will advertise 192.168.0.0/16 and 10.0.0.0/16 to
# Amazon VPC
#
# bgp import filter 1 equal 10.0.0.0/16
# bgp import filter 1 equal 192.168.0.0/16
#

bgp import filter 1 equal 0.0.0.0/0
bgp import 7224 static filter 1

bgp configure refresh
```

## How to Test the Customer Gateway Configuration

You can test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

1. On your customer gateway, determine whether the BGP status is Active.

It takes approximately 30 seconds for a BGP peering to become active.

2. Ensure that the customer gateway is advertising a route to the virtual private gateway. The route may be the default route (0.0.0.0/0) or a more specific route you prefer.

When properly established, your BGP peering should be receiving one route from the virtual private gateway corresponding to the prefix that your VPC integration team specified for the VPC (for example, 10.0.0.0/24). 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.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection: your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the Amazon EC2 Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

#### Note

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.

If your tunnels don't test successfully, see [Troubleshooting Yamaha Customer Gateway Connectivity \(p. 177\)](#).

# Example: Generic Customer Gateway Using Border Gateway Protocol

## Topics

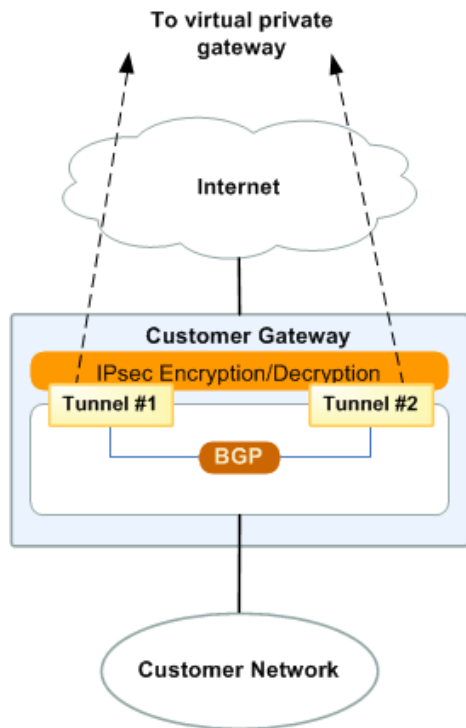
- [A High-Level View of the Customer Gateway \(p. 144\)](#)
- [A Detailed View of the Customer Gateway and an Example Configuration \(p. 144\)](#)
- [How to Test the Customer Gateway Configuration \(p. 149\)](#)

If your customer gateway isn't one of the types discussed earlier in this guide, your integration team will provide you with generic information that you can use to configure your customer gateway. This section contains an example of that information.

Two diagrams illustrate the example configuration. The first diagram shows the high-level layout of the customer gateway, and the second diagram shows details from the example configuration. You should use the real configuration information that you receive from your integration team and apply it to your customer gateway.

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels. Using redundant tunnels ensures continuous availability in the case that a device fails.



## A Detailed View of the Customer Gateway and an Example Configuration

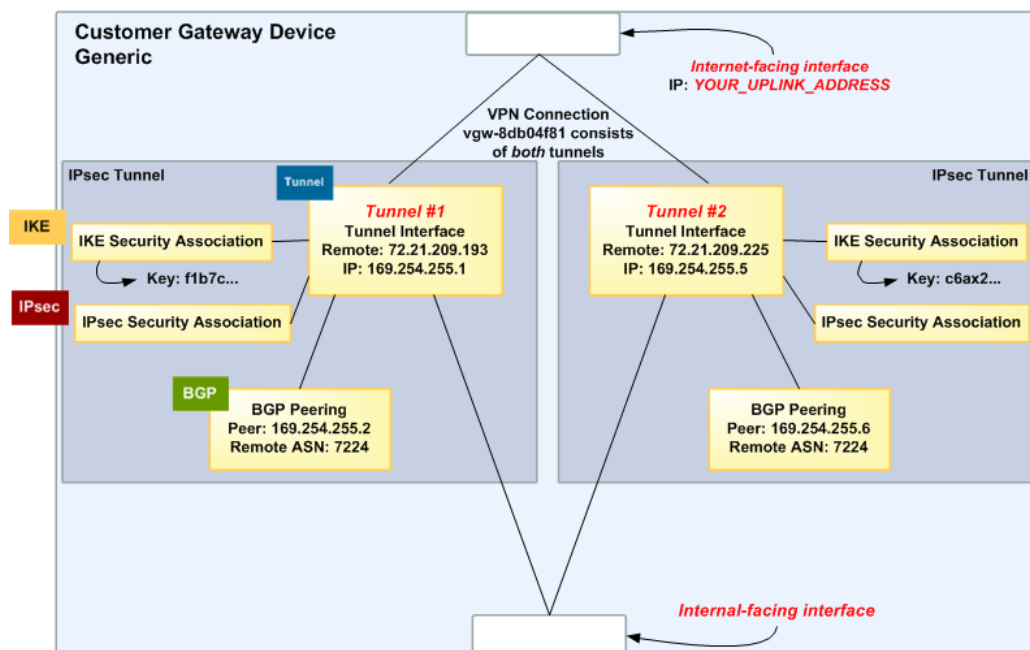
The diagram in this section illustrates an example generic customer gateway. Following the diagram, there is a corresponding example of the configuration information your integration team should provide. The example configuration contains a set of information for each of the tunnels that you must configure.

In addition, the example configuration refers to these items that you must provide:

- ***YOUR\_UPLINK\_ADDRESS***—The IP address for the Internet-routable external interface on the customer gateway. The address must be static, and may be behind a device performing network address translation (NAT). To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500.
- ***YOUR\_BGP\_ASN***—The customer gateway's BGP ASN (we use 65000 by default)

The example configuration includes several example values to help you understand how configuration works. For example, we provide example values for the VPN connection ID (vpn-44a8938f), virtual private gateway ID (vgw-8db04f81), the IP addresses (72.21.209.\*, 169.254.255.\*), and the remote ASN (7224). You'll replace these example values with the actual values from the configuration information that you receive.

In the following diagram and example configuration, you must replace the items in red italics with values that apply to your particular configuration.



Amazon Web Services  
Virtual Private Cloud

#### VPN Connection Configuration

=====

AWS utilizes unique identifiers to manipulate the configuration of a VPN Connection. Each VPN Connection is assigned a VPN identifier and is associated with two other identifiers, namely the Customer Gateway Identifier and the Virtual Private Gateway Identifier.

Your VPN Connection ID : vpn-44a8938f  
Your Virtual Private Gateway ID : vgw-8db04f81  
Your Customer Gateway ID : cgw-b4dc3961

A VPN Connection consists of a pair of IPsec tunnel security associations (SAs). It is important that both tunnel security associations be configured.

#### IPsec Tunnel #1

=====

##### IKE

#### #1: Internet Key Exchange Configuration

Configure the IKE SA as follows:

Please note, these sample configurations are for the minimum requirement of AES128, SHA1, and DH Group 2.

You will need to modify these sample configuration files to take advantage of AES256, SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.

The address of the external interface for your customer gateway must be a static address. Your customer gateway may reside behind a device performing network address translation (NAT).

To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.

```
- IKE version           : IKEv1
- Authentication Method : Pre-Shared Key
- Pre-Shared Key        : plain-text-password1
- Authentication Algorithm : sha1
- Encryption Algorithm  : aes-128-cbc
- Lifetime              : 28800 seconds
- Phase 1 Negotiation Mode : main
- Diffie-Hellman        : Group 2
```

## IPsec

### #2: IPsec Configuration

Configure the IPsec SA as follows:

Please note, you may use these additionally supported IPsec parameters for encryption like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.

```
- Protocol           : esp
- Authentication Algorithm : hmac-sha1-96
- Encryption Algorithm  : aes-128-cbc
- Lifetime           : 3600 seconds
- Mode               : tunnel
- Perfect Forward Secrecy : Diffie-Hellman Group 2
```

IPsec Dead Peer Detection (DPD) will be enabled on the AWS Endpoint. We recommend configuring DPD on your endpoint as follows:

```
- DPD Interval       : 10
- DPD Retries        : 3
```

IPsec ESP (Encapsulating Security Payload) inserts additional headers to transmit packets. These headers require additional space, which reduces the amount of space available to transmit application data. To limit the impact of this behavior, we recommend the following configuration on your Customer Gateway:

```
- TCP MSS Adjustment : 1387 bytes
- Clear Don't Fragment Bit : enabled
- Fragmentation      : Before encryption
```

## Tunnel

### #3: Tunnel Interface Configuration

Your Customer Gateway must be configured with a tunnel interface that is associated with the IPsec tunnel. All traffic transmitted to the tunnel interface is encrypted and transmitted to the Virtual Private Gateway.

The Customer Gateway and Virtual Private Gateway each have two addresses that relate to this IPsec tunnel. Each contains an outside address, upon which encrypted traffic is exchanged. Each also contain an inside address associated with the tunnel interface.

The Customer Gateway outside IP address was provided when the Customer Gateway was created. Changing the IP address requires the creation of a new Customer Gateway.

The Customer Gateway inside IP address should be configured on your tunnel interface.

Outside IP Addresses:

```
- Customer Gateway: YOUR_UPLINK_ADDRESS
- Virtual Private Gateway : 72.21.209.193
```

#### Inside IP Addresses

- Customer Gateway : **169.254.255.2/30**
- Virtual Private Gateway : **169.254.255.1/30**

Configure your tunnel to fragment at the optimal size:

- Tunnel interface MTU : 1436 bytes

### BGP

#### #4: Border Gateway Protocol (BGP) Configuration:

The Border Gateway Protocol (BGPv4) is used within the tunnel, between the inside IP addresses, to exchange routes from the VPC to your home network. Each BGP router has an Autonomous System Number (ASN). Your ASN was provided to AWS when the Customer Gateway was created.

#### BGP Configuration Options:

- Customer Gateway ASN : **YOUR\_BGP\_ASN**
- Virtual Private Gateway ASN : **7224**
- Neighbor IP Address : **169.254.255.1**
- Neighbor Hold Time : 30

Configure BGP to announce routes to the Virtual Private Gateway. The gateway will announce prefixes to your customer gateway based upon the prefix you assigned to the VPC at creation time.

#### IPsec Tunnel #2

=====

### IKE

#### #1: Internet Key Exchange Configuration

Configure the IKE SA as follows:

Please note, these sample configurations are for the minimum requirement of AES128, SHA1, and DH Group 2.

You will need to modify these sample configuration files to take advantage of AES256, SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.

The address of the external interface for your customer gateway must be a static address. Your customer gateway may reside behind a device performing network address translation (NAT).

To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.

- IKE version : IKEv1
- Authentication Method : Pre-Shared Key
- Pre-Shared Key : **plain-text-password2**
- Authentication Algorithm : sha1
- Encryption Algorithm : aes-128-cbc
- Lifetime : 28800 seconds
- Phase 1 Negotiation Mode : main
- Diffie-Hellman : Group 2

### IPsec

#### #2: IPsec Configuration

Configure the IPsec SA as follows:

Please note, you may use these additionally supported IPsec parameters for encryption like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.

- Protocol : esp
- Authentication Algorithm : hmac-sha1-96
- Encryption Algorithm : aes-128-cbc
- Lifetime : 3600 seconds



- Mode : tunnel
- Perfect Forward Secrecy : Diffie-Hellman Group 2

IPsec Dead Peer Detection (DPD) will be enabled on the AWS Endpoint. We recommend configuring DPD on your endpoint as follows:

- DPD Interval : 10
- DPD Retries : 3

IPsec ESP (Encapsulating Security Payload) inserts additional headers to transmit packets. These headers require additional space, which reduces the amount of space available to transmit application data. To limit the impact of this behavior, we recommend the following configuration on your Customer Gateway:

- TCP MSS Adjustment : 1387 bytes
- Clear Don't Fragment Bit : enabled
- Fragmentation : Before encryption



### #3: Tunnel Interface Configuration

Your Customer Gateway must be configured with a tunnel interface that is associated with the IPsec tunnel. All traffic transmitted to the tunnel interface is encrypted and transmitted to the Virtual Private Gateway.

The Customer Gateway and Virtual Private Gateway each have two addresses that relate to this IPsec tunnel. Each contains an outside address, upon which encrypted traffic is exchanged. Each also contain an inside address associated with the tunnel interface.

The Customer Gateway outside IP address was provided when the Customer Gateway was created. Changing the IP address requires the creation of a new Customer Gateway.

The Customer Gateway inside IP address should be configured on your tunnel interface.

Outside IP Addresses:

- Customer Gateway : **YOUR\_UPLINK\_ADDRESS**
- Virtual Private Gateway : **72.21.209.193**

Inside IP Addresses

- Customer Gateway : **169.254.255.6/30**
- Virtual Private Gateway : **169.254.255.5/30**

Configure your tunnel to fragment at the optimal size:

- Tunnel interface MTU : 1436 bytes



### #4: Border Gateway Protocol (BGP) Configuration:

The Border Gateway Protocol (BGPv4) is used within the tunnel, between the inside IP addresses, to exchange routes from the VPC to your home network. Each BGP router has an Autonomous System Number (ASN). Your ASN was provided to AWS when the Customer Gateway was created.

BGP Configuration Options:

- Customer Gateway ASN : **YOUR\_BGP\_ASN**
- Virtual Private Gateway ASN : **7224**
- Neighbor IP Address : **169.254.255.5**
- Neighbor Hold Time : 30

Configure BGP to announce routes to the Virtual Private Gateway. The gateway will announce prefixes to your customer gateway based upon the prefix you assigned to the VPC at creation time.

## How to Test the Customer Gateway Configuration

You can test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

1. On your customer gateway, determine whether the BGP status is **Active**.

It takes approximately 30 seconds for a BGP peering to become active.

2. Ensure that the customer gateway is advertising a route to the virtual private gateway. The route may be the default route (0.0.0.0/0) or a more specific route you prefer.

When properly established, your BGP peering should be receiving one route from the virtual private gateway corresponding to the prefix that your VPC integration team specified for the VPC (for example, 10.0.0.0/24). 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.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection: your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are listed in the launch wizard when you launch an instance from the Amazon EC2 Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be 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
```

**Note**

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

4. (Optional) To test tunnel failover, you can temporarily disable one of the tunnels on your customer gateway, and repeat the above step. You cannot disable a tunnel on the AWS side of the VPN connection.

If your tunnels don't test successfully, see [Troubleshooting Generic Device Customer Gateway Connectivity Using Border Gateway Protocol \(p. 180\)](#).

# Example: Generic Customer Gateway without Border Gateway Protocol

## Topics

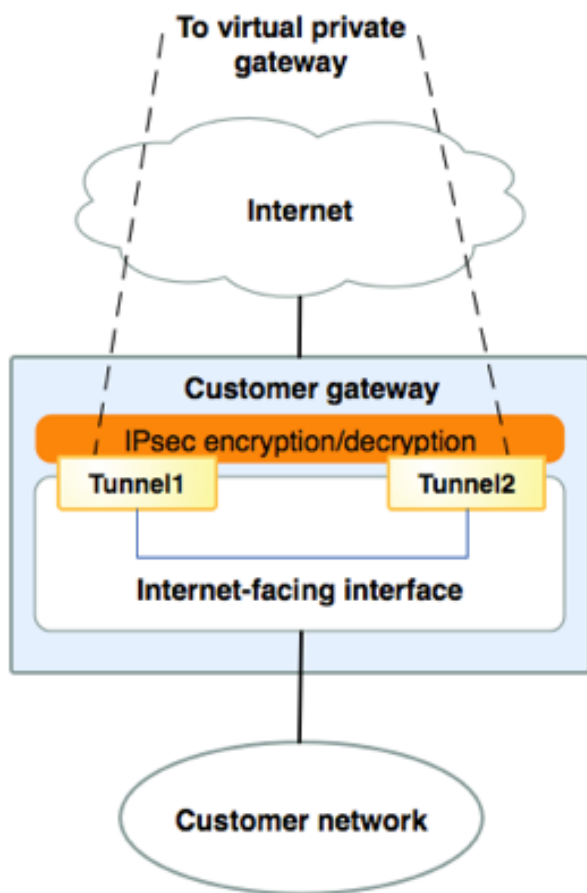
- [A High-Level View of the Customer Gateway \(p. 152\)](#)
- [A Detailed View of the Customer Gateway and an Example Configuration \(p. 152\)](#)
- [How to Test the Customer Gateway Configuration \(p. 157\)](#)

If your customer gateway isn't one of the types discussed earlier in this guide, your integration team will provide you with generic information that you can use to configure your customer gateway. This section contains an example of that information.

Two diagrams illustrate the example configuration. The first diagram shows the high-level layout of the customer gateway, and the second diagram shows details from the example configuration. You should use the real configuration information that you receive from your integration team and apply it to your customer gateway.

## A High-Level View of the Customer Gateway

The following diagram shows the general details of your customer gateway. Note that the VPN connection consists of two separate tunnels: *Tunnel 1* and *Tunnel 2*. Using redundant tunnels ensures continuous availability in the case that a device fails.



## A Detailed View of the Customer Gateway and an Example Configuration

The diagram in this section illustrates an example generic customer gateway (without BGP). Following the diagram, there is a corresponding example of the configuration information your integration team should provide. The example configuration contains a set of information for each of the tunnels that you must configure.

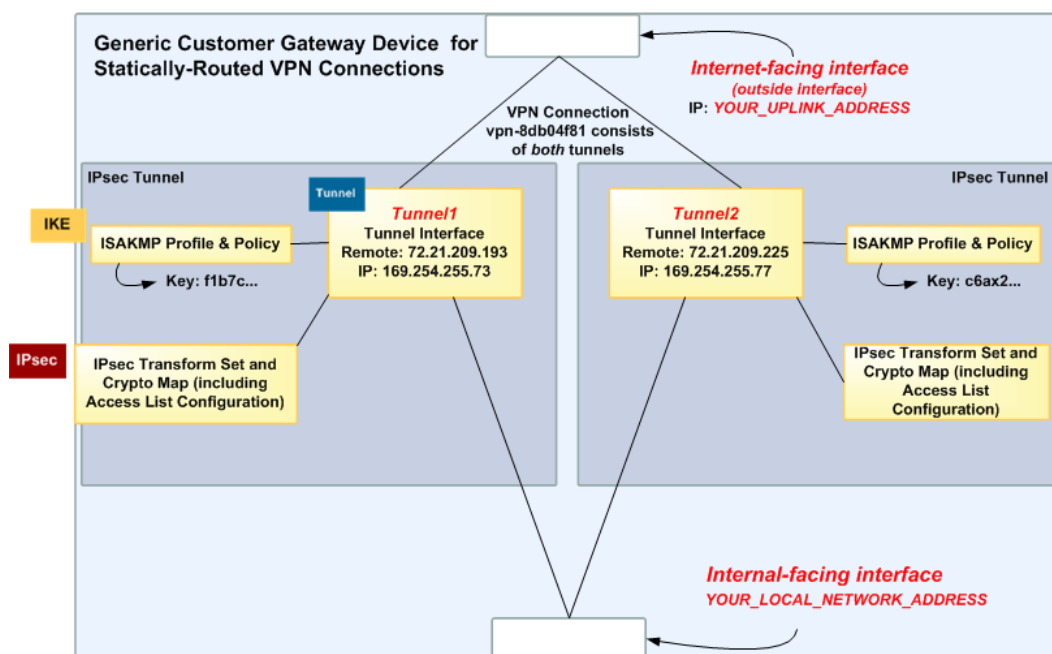
The diagram in this section illustrates a generic customer gateway that uses static routing for its VPN connection (meaning that it does not support dynamic routing, or Border Gateway Protocol (BGP)). Following the diagram, there is a corresponding example of the configuration information your integration team should give you. The example configuration contains a set of information for each of the two tunnels you must configure.

In addition, the example configuration refers to one item that you must provide:

- ***YOUR\_UPLINK\_ADDRESS***—The IP address for the Internet-routable external interface on the customer gateway. The address must be static, and may be behind a device performing network address translation (NAT). To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500.

The example configuration includes several example values to help you understand how configuration works. For example, we provide example values for the VPN connection ID (vpn-44a8938f), virtual private gateway ID (vgw-8db04f81), and the VGW IP addresses (72.21.209.\*, 169.254.255.\*). You'll replace these example values with the actual values from the configuration information that you receive.

In the following diagram and example configuration, you must replace the items in red italics with values that apply to your particular configuration.



### Important

The following configuration information is an example of what you can expect an integration team to provide. Many of the values in the following example will be different from the actual configuration information that you receive. You must use the actual values and not the example values shown here, or your implementation will fail.

Amazon Web Services  
Virtual Private Cloud

#### VPN Connection Configuration

=====

AWS utilizes unique identifiers to manipulate the configuration of a VPN Connection. Each VPN Connection is assigned a VPN Connection Identifier and is associated with two other identifiers, namely the Customer Gateway Identifier and the Virtual Private Gateway Identifier.

Your VPN Connection ID	: vpn-44a8938f
Your Virtual Private Gateway ID	: vgw-8db04f81
Your Customer Gateway ID	: cgw-ff628496

A VPN Connection consists of a pair of IPsec tunnel security associations (SAs). It is important that both tunnel security associations be configured.

#### IPSec Tunnel #1

=====

### IKE

#### #1: Internet Key Exchange Configuration

Configure the IKE SA as follows

Please note, these sample configurations are for the minimum requirement of AES128, SHA1, and DH Group 2.

You will need to modify these sample configuration files to take advantage of AES256, SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.

The address of the external interface for your customer gateway must be a static address. Your customer gateway may reside behind a device performing network address translation (NAT).

To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.

- IKE version : IKEv1
- Authentication Method : Pre-Shared Key
- Pre-Shared Key : **PRE-SHARED-KEY-IN-PLAIN-TEXT**
- Authentication Algorithm : sha1
- Encryption Algorithm : aes-128-cbc
- Lifetime : 28800 seconds
- Phase 1 Negotiation Mode : main
- Diffie-Hellman : Group 2

### IPsec

#### #2: IPSec Configuration

Configure the IPSec SA as follows:

Please note, you may use these additionally supported IPSec parameters for encryption like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.

- Protocol : esp
- Authentication Algorithm : hmac-sha1-96
- Encryption Algorithm : aes-128-cbc
- Lifetime : 3600 seconds
- Mode : tunnel
- Perfect Forward Secrecy : Diffie-Hellman Group 2

IPSec Dead Peer Detection (DPD) will be enabled on the AWS Endpoint. We recommend configuring DPD on your endpoint as follows:

- DPD Interval : 10
- DPD Retries : 3

IPSec ESP (Encapsulating Security Payload) inserts additional headers to transmit packets. These headers require additional space, which reduces the amount of space available to transmit application data. To limit the impact of this behavior, we recommend the following configuration on your Customer Gateway:

- TCP MSS Adjustment : 1387 bytes
- Clear Don't Fragment Bit : enabled
- Fragmentation : Before encryption

### Tunnel

#### #3: Tunnel Interface Configuration

Your Customer Gateway must be configured with a tunnel interface that is associated with the IPSec tunnel. All traffic transmitted to the tunnel interface is encrypted and transmitted to the Virtual Private Gateway.

The Customer Gateway and Virtual Private Gateway each have two addresses that relate to this IPSec tunnel. Each contains an outside address, upon which encrypted traffic is exchanged. Each also contain an inside address associated with the tunnel interface.

The Customer Gateway outside IP address was provided when the Customer Gateway was created. Changing the IP address requires the creation of a new Customer Gateway.

The Customer Gateway inside IP address should be configured on your tunnel interface.

Outside IP Addresses:

- Customer Gateway : **YOUR\_UPLINK\_ADDRESS**
- Virtual Private Gateway : **72.21.209.193**

Inside IP Addresses

- Customer Gateway : **169.254.255.74/30**
- Virtual Private Gateway : **169.254.255.73/30**

Configure your tunnel to fragment at the optimal size:

- Tunnel interface MTU : 1436 bytes

#4: Static Routing Configuration:

To route traffic between your internal network and your VPC, you will need a static route added to your router.

Static Route Configuration Options:

- Next hop : **169.254.255.73**

You should add static routes towards your internal network on the VGW. The VGW will then send traffic towards your internal network over the tunnels.

IPSec Tunnel #2

=====

## IKE

#1: Internet Key Exchange Configuration

Configure the IKE SA as follows:

Please note, these sample configurations are for the minimum requirement of AES128, SHA1, and DH Group 2.

You will need to modify these sample configuration files to take advantage of AES256, SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.

The address of the external interface for your customer gateway must be a static address. Your customer gateway may reside behind a device performing network address translation (NAT).

To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.

- IKE version : IKEv1
- Authentication Method : Pre-Shared Key
- Pre-Shared Key : **PRE-SHARED-KEY-IN-PLAIN-TEXT**
- Authentication Algorithm : sha1
- Encryption Algorithm : aes-128-cbc
- Lifetime : 28800 seconds
- Phase 1 Negotiation Mode : main
- Diffie-Hellman : Group 2



## IPsec

### #2: IPsec Configuration

Configure the IPsec SA as follows:

Please note, you may use these additionally supported IPsec parameters for encryption like AES256 and other DH groups like 2, 5, 14-18, 22, 23, and 24.

- Protocol : esp
- Authentication Algorithm : hmac-sha1-96
- Encryption Algorithm : aes-128-cbc
- Lifetime : 3600 seconds
- Mode : tunnel
- Perfect Forward Secrecy : Diffie-Hellman Group 2

IPsec Dead Peer Detection (DPD) will be enabled on the AWS Endpoint. We recommend configuring DPD on your endpoint as follows:

- DPD Interval : 10
- DPD Retries : 3

IPsec ESP (Encapsulating Security Payload) inserts additional headers to transmit packets. These headers require additional space, which reduces the amount of space available to transmit application data.

To limit the impact of this behavior, we recommend the following configuration on your Customer Gateway:

- TCP MSS Adjustment : 1387 bytes
- Clear Don't Fragment Bit : enabled
- Fragmentation : Before encryption

## Tunnel

### #3: Tunnel Interface Configuration

Your Customer Gateway must be configured with a tunnel interface that is associated with the IPsec tunnel. All traffic transmitted to the tunnel interface is encrypted and transmitted to the Virtual Private Gateway.

The Customer Gateway and Virtual Private Gateway each have two addresses that relate to this IPsec tunnel. Each contains an outside address, upon which encrypted traffic is exchanged. Each also contain an inside address associated with the tunnel interface.

The Customer Gateway outside IP address was provided when the Customer Gateway was created. Changing the IP address requires the creation of a new Customer Gateway.

The Customer Gateway inside IP address should be configured on your tunnel interface.

Outside IP Addresses:

- Customer Gateway : **YOUR\_UPLINK\_ADDRESS**
- Virtual Private Gateway : **72.21.209.225**

Inside IP Addresses

- Customer Gateway : **169.254.255.78/30**
- Virtual Private Gateway : **169.254.255.77/30**

Configure your tunnel to fragment at the optimal size:

- Tunnel interface MTU : 1436 bytes

### #4: Static Routing Configuration:

To route traffic between your internal network and your VPC, you will need a static route added to your router.

Static Route Configuration Options:

- Next hop : **169.254.255.77**

You should add static routes towards your internal network on the VGW. The VGW will then send traffic towards your internal network over the tunnels.

## How to Test the Customer Gateway Configuration

You must first test the gateway configuration for each tunnel.

### To test the customer gateway configuration for each tunnel

- On your customer gateway, verify that you have added a static route to the VPC CIDR IP space to use the tunnel interface.

Next you must test the connectivity for each tunnel by launching an instance into your VPC, and pinging the instance from your home network. 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 your instance's security group and network ACL to enable inbound ICMP traffic.
- Ensure that you have configured routing for your VPN connection - your subnet's route table must contain a route to the virtual private gateway. For more information, see [Enable Route Propagation in Your Route Table](#) in the *Amazon VPC User Guide*.

### To test the end-to-end connectivity of each tunnel

1. Launch an instance of one of the Amazon Linux AMIs into your VPC. The Amazon Linux AMIs are available in the Quick Start menu when you use the Launch Instances Wizard in the AWS Management Console. For more information, see the [Amazon VPC Getting Started Guide](#).
2. After the instance is running, get its private IP address (for example, 10.0.0.4). The console displays the address as part of the instance's details.
3. On a system in your home network, use the **ping** command with the instance's IP address. Make sure that the computer you ping from is behind the customer gateway. A successful response should be similar to the following.

```
PROMPT> 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
```

**Note**

If you ping an instance from your customer gateway router, ensure that you are sourcing ping messages from an internal IP address, not a tunnel IP address. Some AMIs don't respond to ping messages from tunnel IP addresses.

If your tunnels don't test successfully, see [Troubleshooting Generic Device Customer Gateway Connectivity Using Border Gateway Protocol \(p. 180\)](#).

# Troubleshooting

The following topics contain troubleshooting information that you can use if your tunnels aren't in the correct state when you test your customer gateway.

## Topics

- [Troubleshooting Cisco ASA Customer Gateway Connectivity \(p. 159\)](#)
- [Troubleshooting Cisco IOS Customer Gateway Connectivity \(p. 162\)](#)
- [Troubleshooting Cisco IOS Customer Gateway without Border Gateway Protocol Connectivity \(p. 166\)](#)
- [Troubleshooting Juniper JunOS Customer Gateway Connectivity \(p. 171\)](#)
- [Troubleshooting Juniper ScreenOS Customer Gateway Connectivity \(p. 174\)](#)
- [Troubleshooting Yamaha Customer Gateway Connectivity \(p. 177\)](#)
- [Troubleshooting Generic Device Customer Gateway Connectivity Using Border Gateway Protocol \(p. 180\)](#)
- [Troubleshooting Generic Device Customer Gateway without Border Gateway Protocol Connectivity \(p. 183\)](#)

## Troubleshooting Cisco ASA Customer Gateway Connectivity

When you troubleshoot the connectivity of a Cisco customer gateway, you need to consider three things: 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 may 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 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* of the remote gateway specified in the tunnels. The *state* 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 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 an inbound esp sas and outbound esp sas. This assumes that an SA is listed (for example, spi: 0x48B456A6), and IPsec is configured correctly.

In Cisco ASA, the IPsec will only come up after "interesting traffic" is sent. To always keep the IPsec active, we recommend configuring SLA monitor. SLA monitor will continue 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 up and running fine. If this is not working, check your access lists, and refer the previous IPsec section.

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

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. Next, check the access list as follows.

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

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

For example:

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

3. Verify that this access list is correct. The example access list in the previous step allows all internal traffic to the VPC subnet 10.0.0.0/16.
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 you added in the Amazon console, and also the security groups for the particular instances.

5. For further troubleshooting, review the configuration.

## Troubleshooting Cisco IOS Customer Gateway Connectivity

When you troubleshoot the connectivity of a Cisco customer gateway you need to consider four things: IKE, IPsec, tunnel, and BGP. You can troubleshoot these areas 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 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 a *src* of the Remote Gateway 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 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 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 pcsp 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 pcsp 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 an *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 a list of the rules, see [Configuring a Firewall Between the Internet and Your Customer Gateway](#) (p. 10).

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
```

Ensure 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 outside IP address, interface, and virtual private gateway outside IP address. Ensure that *Tunnel protection via IPSec* is present. Make sure to run the command on both tunnel interfaces. To resolve any problems here, review the configuration.

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 5 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

Here, 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 router 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.

## Virtual Private Gateway Attachment

Make sure your virtual private gateway is attached to your VPC. Your integration team does this with the AWS Management Console.

If you have questions or need further assistance, please use the [Amazon VPC forum](#).

## Troubleshooting Cisco IOS Customer Gateway without Border Gateway Protocol Connectivity

When you troubleshoot the connectivity of a Cisco customer gateway, you need to 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 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* of the remote gateway 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 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 pcip 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 pcip 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 pcip 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), the status is `ACTIVE`, and 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 a list of the rules, see [Configuring a Firewall Between the Internet and Your Customer Gateway](#) (p. 10).

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
```

Ensure 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 outside IP address, interface, and virtual private gateway outside IP address. Ensure that Tunnel protection through IPSec is present. Make sure to run the command on both tunnel interfaces. To resolve any problems, review the configuration.

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 shown here.

```
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 of "Number of successes" indicates whether the SLA monitor has been set up successfully.

For further troubleshooting, review the configuration.

## Virtual Private Gateway Attachment

Verify that your virtual private gateway is attached to your VPC. Your integration team does this with the AWS Management Console.

If you have questions or need further assistance, please use the [Amazon VPC forum](#).

## Troubleshooting Juniper JunOS Customer Gateway Connectivity

When you troubleshoot the connectivity of a Juniper customer gateway you need to 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 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	ca7cb0abaedb4bb	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 information (see [Example: Juniper J-Series JunOS Device \(p. 88\)](#)). 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 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). Note the carets at the beginning of each line (< >) which 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) 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 the rules, see [Configuring a Firewall Between the Internet and Your Customer Gateway \(p. 10\)](#).

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 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

Use 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 router 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, ensure that you're receiving the prefix corresponding 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
```

## Virtual Private Gateway Attachment

Make sure your virtual private gateway is attached to your VPC. Your integration team does this with the AWS Management Console.

If you have questions or need further assistance, please use the [Amazon VPC forum](#).

## Troubleshooting Juniper ScreenOS Customer Gateway Connectivity

When you troubleshoot the connectivity of a Juniper ScreenOS-based customer gateway you need to 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 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 specified in the tunnels. The *Sta* should be A/- and the *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 information (see [Example: Juniper ScreenOS Device \(p. 108\)](#))).

### Tunnel

First, double-check that you have the necessary firewall rules in place. For a list of the rules, see [Configuring a Firewall Between the Internet and Your Customer Gateway \(p. 10\)](#).

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 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

Use 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	

Both BGP peers should be listed as State: ESTABLISH, which means 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 router is advertising the default route (0.0.0.0/0) to the VPC. Note that 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 corresponding to your VPC from the virtual private gateway. Note that 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
```

## Virtual Private Gateway Attachment

Make sure your virtual private gateway is attached to your VPC. Your integration team does this with the AWS Management Console.

If you have questions or need further assistance, please use the [Amazon VPC forum](#).

# Troubleshooting Yamaha Customer Gateway Connectivity

When you troubleshoot the connectivity of a Yamaha customer gateway you need to 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 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* of the Remote Gateway specified in the tunnels. You can list all 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, use the following command.

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

## IPsec

Use the following command. The response shows a customer gateway 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
```

Use 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 the rules, see [Configuring a Firewall Between the Internet and Your Customer Gateway \(p. 10\)](#).

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]
```

Ensure the *current status* is online. Also, ensure that *Interface type* is IPsec. Make sure to run the command on both tunnel interfaces. To resolve any problems here, review the configuration.

## BGP

Use 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:
```

Here, 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 router 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 corresponding 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

For further troubleshooting, review the configuration.

## Virtual Private Gateway Attachment

Make sure your virtual private gateway is attached to your VPC. Your integration team does this with the AWS Management Console.

If you have questions or need further assistance, please use the [Amazon VPC forum](#).

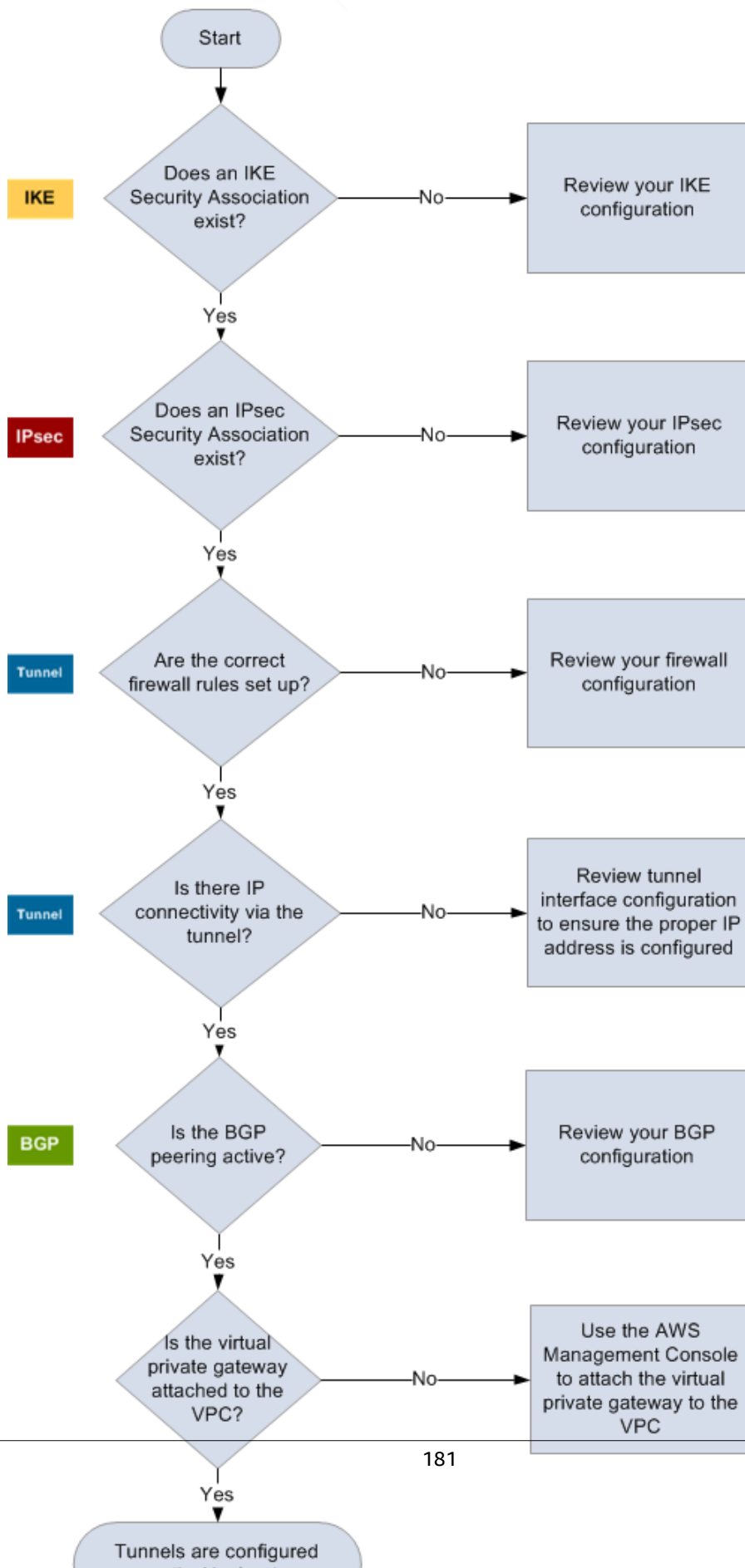


# Troubleshooting Generic Device Customer Gateway Connectivity Using Border Gateway Protocol

The following diagram and table provide general instructions for troubleshooting a customer gateway that uses Border Gateway Protocol for devices other than those listed in this guide.

**Tip**

When troubleshooting problems, you might find it useful to enable the debug features of your gateway device. Consult your gateway device vendor for details.



<b>IKE</b>	<p>Determine if an IKE Security Association exists.</p> <p>An IKE security association is required to exchange keys that are used to establish the IPsec Security Association.</p> <p>If no IKE security association exists, review your IKE configuration settings. You must configure the encryption, authentication, perfect-forward-secrecy, and mode parameters as listed in the customer gateway configuration.</p> <p>If an IKE security association exists, move on to IPsec.</p>
<b>IPsec</b>	<p>Determine if an IPsec Security Association exists.</p> <p>An IPsec security association is the tunnel itself. Query your customer gateway to determine if an IPsec Security Association is active. Proper configuration of the IPsec SA is critical. You must configure the encryption, authentication, perfect-forward-secrecy, and mode parameters as listed in the customer gateway configuration.</p> <p>If no IPsec Security Association exists, review your IPsec configuration.</p> <p>If an IPsec Security Association exists, move on to the tunnel.</p>
<b>Tunnel</b>	<p>Confirm the required firewall rules are set up (for a list of the rules, see <a href="#">Configuring a Firewall Between the Internet and Your Customer Gateway (p. 10)</a>). If they are, move forward.</p> <p>Determine if there is IP connectivity via the tunnel.</p> <p>Each side of the tunnel has an IP address as specified in the customer gateway configuration. The virtual private gateway address is the address used as the BGP neighbor address. From your customer gateway, ping this address to determine if IP traffic is being properly encrypted and decrypted.</p> <p>If the ping isn't successful, review your tunnel interface configuration to ensure the proper IP address is configured.</p> <p>If the ping is successful, move on to BGP.</p>
<b>BGP</b>	<p>Determine if the BGP peering is active.</p> <p>For each tunnel, do the following:</p> <ul style="list-style-type: none"> <li>On your customer gateway, determine if the BGP status is <i>Active</i> or <i>Established</i>. It may take approximately 30 seconds for a BGP peering to become active.</li> <li>Ensure that the customer gateway is advertising the default route (0.0.0.0/0) to the virtual private gateway.</li> </ul> <p>If the tunnels are not in this state, review your BGP configuration.</p> <p>If the BGP peering is established, you are receiving a prefix, and you are advertising a prefix, your tunnel is configured correctly. Ensure both tunnels are in this state, and you're done.</p>
	<p>Make sure your virtual private gateway is attached to your VPC. Your integration team does this with the AWS Management Console.</p>

For general testing instructions applicable to all customer gateways, see [How to Test the Customer Gateway Configuration](#) (p. 149).

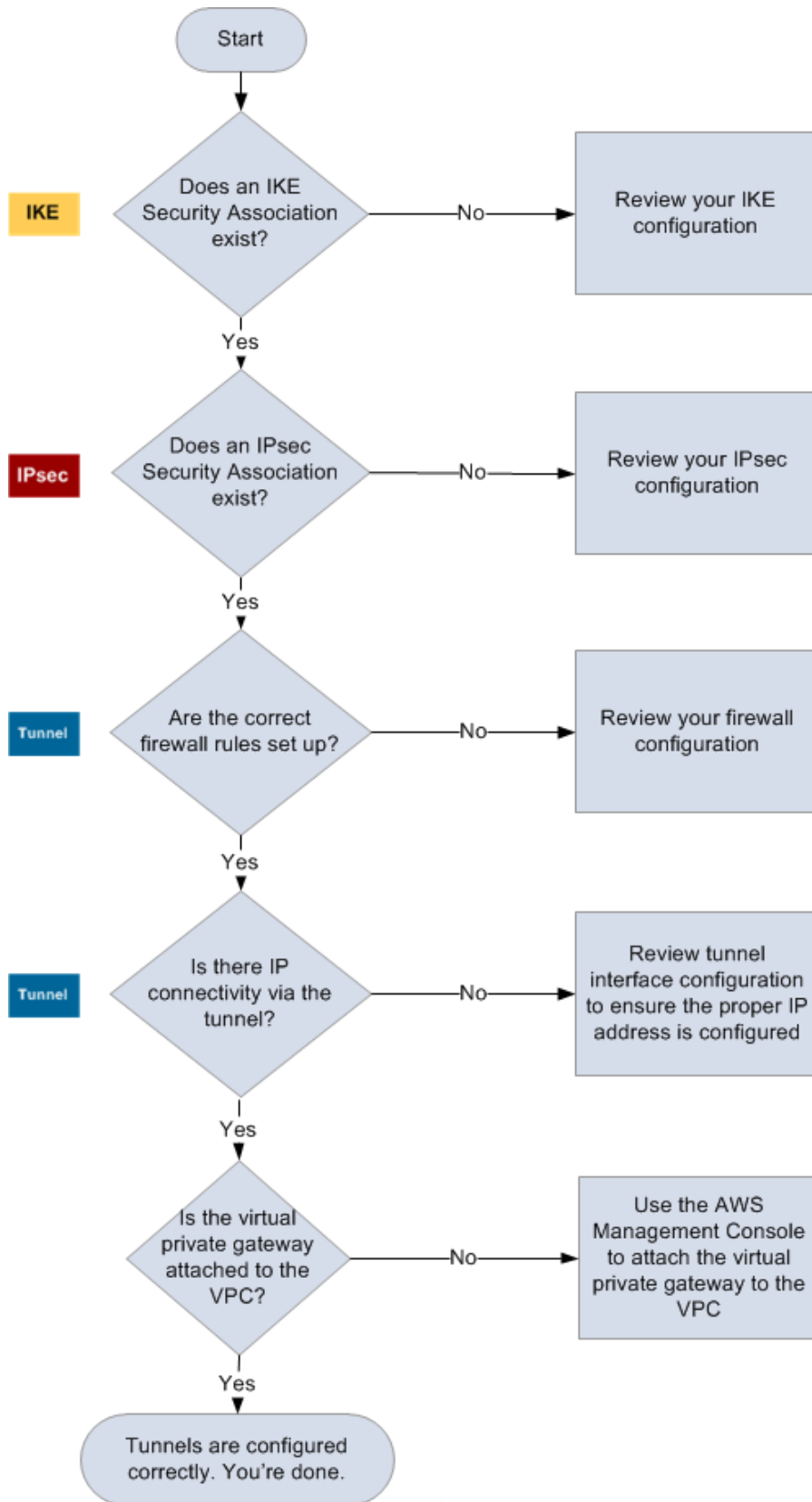
If you have questions or need further assistance, please use the [Amazon VPC forum](#).

## Troubleshooting Generic Device Customer Gateway without Border Gateway Protocol Connectivity

The following diagram and table provide general instructions for troubleshooting a customer gateway device that does not use Border Gateway Protocol.

**Tip**

When troubleshooting problems, you might find it useful to enable the debug features of your gateway device. Consult your gateway device vendor for details.



<b>IKE</b>	<p>Determine if an IKE Security Association exists.</p> <p>An IKE security association is required to exchange keys that are used to establish the IPsec Security Association.</p> <p>If no IKE security association exists, review your IKE configuration settings. You must configure the encryption, authentication, perfect-forward-secrecy, and mode parameters as listed in the customer gateway configuration.</p> <p>If an IKE security association exists, move on to IPsec.</p>
<b>IPsec</b>	<p>Determine if an IPsec Security Association exists.</p> <p>An IPsec security association is the tunnel itself. Query your customer gateway to determine if an IPsec Security Association is active. Proper configuration of the IPsec SA is critical. You must configure the encryption, authentication, perfect-forward-secrecy, and mode parameters as listed in the customer gateway configuration.</p> <p>If no IPsec Security Association exists, review your IPsec configuration.</p> <p>If an IPsec Security Association exists, move on to the tunnel.</p>
<b>Tunnel</b>	<p>Confirm the required firewall rules are set up (for a list of the rules, see <a href="#">Configuring a Firewall Between the Internet and Your Customer Gateway (p. 10)</a>). If they are, move forward.</p> <p>Determine if there is IP connectivity via the tunnel.</p> <p>Each side of the tunnel has an IP address as specified in the customer gateway configuration. The virtual private gateway address is the address used as the BGP neighbor address. From your customer gateway, ping this address to determine if IP traffic is being properly encrypted and decrypted.</p> <p>If the ping isn't successful, review your tunnel interface configuration to ensure the proper IP address is configured.</p> <p>If the ping is successful, move on to Routing.</p>
Static routes	<p>Routing:</p> <p>For each tunnel, do the following:</p> <ul style="list-style-type: none"> <li>• Verify that you have added a static route to your VPC CIDR with the tunnels as the next hop.</li> <li>• Verify that you have added a static route on the AWS console, to tell the VGW to route traffic back to your internal networks.</li> </ul> <p>If the tunnels are not in this state, review your device configuration.</p> <p>Ensure both tunnels are in this state, and you're done.</p>
	<p>Make sure your virtual private gateway is attached to your VPC. Your integration team does this with the AWS Management Console.</p>

If you have questions or need further assistance, please use the [Amazon VPC forum](#).

# Configuring Windows Server 2008 R2 as a Customer Gateway

You can configure Windows Server 2008 R2 as a customer gateway for your VPC. Use the following process whether you are running Windows Server 2008 R2 on an EC2 instance in a VPC, or on your own server.

## Topics

- [Configuring Your Windows Server](#) (p. 186)
- [Step 1: Create a VPN Connection and Configure Your VPC](#) (p. 187)
- [Step 2: Download the Configuration File for the VPN Connection](#) (p. 188)
- [Step 3: Configure the Windows Server](#) (p. 190)
- [Step 4: Set Up the VPN Tunnel](#) (p. 192)
- [Step 5: Enable Dead Gateway Detection](#) (p. 198)
- [Step 6: Test the VPN Connection](#) (p. 198)

## Configuring Your Windows Server

To configure Windows Server as a customer gateway, ensure that you have Windows Server 2008 R2 on your own network, or on an EC2 instance in a VPC. If you use 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 Server instance, and choose **Actions, Networking, Change Source/Dest. Check**. Choose **Yes, Disable**.
- 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. Right-click the Citrix or AWS PV network adapter, and then click **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**.
- Associate an Elastic IP address with the instance:
    1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
    2. In the navigation pane, choose **Elastic IPs**. Choose **Allocate new address**.
    3. Select the Elastic IP address, and choose **Actions, Associate Address**.
    4. For **Instance**, select your Windows Server instance. Choose **Associate**.

Take note of this address — you will need it when you create the customer gateway in your VPC.

- Ensure 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 for your network in which the Windows server 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, you must first create a virtual private gateway and attach it to your VPC. Then you can create a VPN connection and configure your VPC. You must also have the CIDR range for your network in which the Windows server is located, for example, 172.31.0.0/16.

### To create a virtual private gateway

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Virtual Private Gateways**, and then **Create Virtual Private Gateway**.
3. You can optionally enter a name for your virtual private gateway, and then choose **Yes, Create**.
4. Select the virtual private gateway that you created, and then choose **Attach to VPC**.
5. In the **Attach to VPC** dialog box, select your VPC from the list, and then choose **Yes, Attach**.

### To create a VPN connection

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **VPN Connections**, and then **Create VPN Connection**.
3. Select the virtual private gateway from the list.
4. For **Customer Gateway**, choose **New**. For **IP address**, specify the public IP address of your Windows Server.

#### Note

The IP address must be static and may be behind a device performing network address translation (NAT). To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500. If your customer gateway is an EC2 Windows Server instance, use its Elastic IP address.



5. Select the **Static** routing option, enter the **Static IP Prefixes** values for your network in CIDR notation, and then choose **Yes, Create**.

### To configure your VPC

- Create a private subnet in your VPC (if you don't have one already) for launching instances that will communicate with the Windows server. For more information, see [Adding a Subnet to 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.
  - Enable route propagation for the virtual private gateway. For more information, see [Route Tables](#) in the *Amazon VPC User Guide*.
- Create a security group configuration 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, click **VPN Connections**.
3. Select your VPN connection, and then click **Download Configuration**.
4. Select **Microsoft** as the vendor, **Windows Server** as the platform, and **2008 R2** as the software. Click **Yes, Download**. You can open the file or save it.

The configuration file contains a section of information similar to the following example. You'll see this information presented twice, one time for each tunnel. You'll use this information when configuring the Windows Server 2008 R2 server.

```
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:              xCjNLsLoCmKsakwcdR9yX6GsEXAMPLE
```

#### 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 is a Windows server 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.

#### Endpoint 1

The IP prefix that you specified as a static route when you created the VPN connection. These are the IP addresses on 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 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 AWS virtual private gateway within a matter of minutes. When you configure your customer gateway, it's important that you configure both tunnels.

#### Note

From time to time, AWS performs routine maintenance on the virtual private gateway. This maintenance may 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. Because the AWS VPC VPN suggested settings are the same as the Windows Server 2008 R2 default IPsec configuration settings, minimal work is needed on your part.

```
MainModeSecMethods:      DHGroup2-AES128-SHA1,DHGroup2-3DES-SHA1
MainModeKeyLifetime:      480min,0sec
QuickModeSecMethods:      ESP:SHA1-AES128+60min+100000kb,
                           ESP:SHA1-3DES+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 2008 R2 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 2008 R2 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 2008 R2 IPsec VPN connections.

#### QuickModePFS

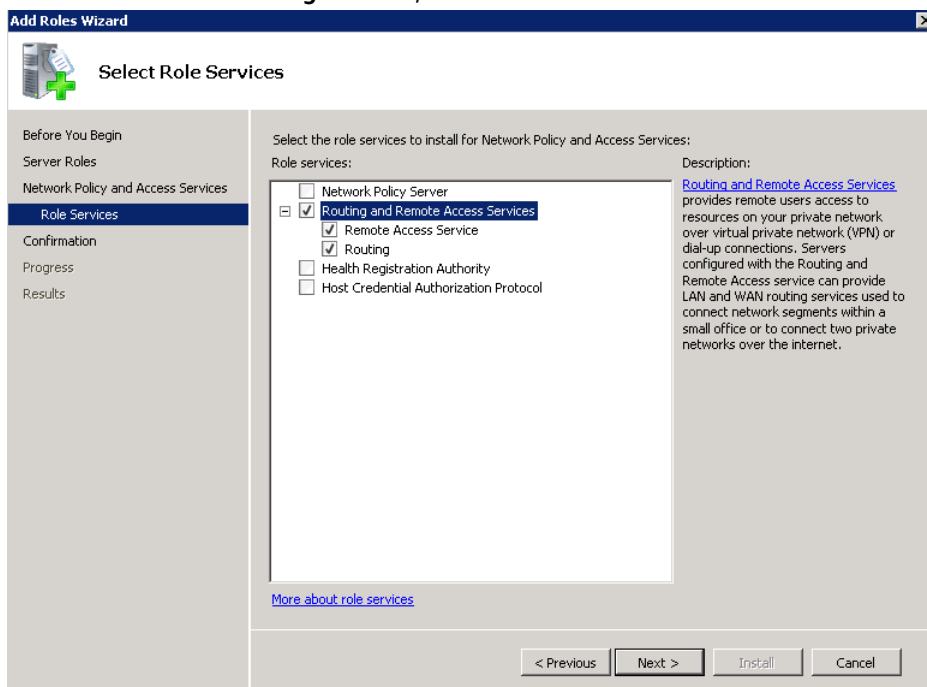
We suggest the use of master key perfect forward secrecy (PFS) for your IPsec sessions.

## Step 3: Configure the Windows Server

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

### To install Routing and Remote Access Services on Windows Server 2008 R2

1. Log on to the Windows Server 2008 R2 server.
2. Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **Server Manager**.
3. Install Routing and Remote Access Services:
  - a. In the Server Manager navigation pane, click **Roles**.
  - b. In the **Roles** pane, click **Add Roles**.
  - c. On the **Before You Begin** page, verify that your server meets the prerequisites and then click **Next**.
  - d. On the **Select Server Roles** page, click **Network Policy and Access Services**, and then click **Next**.
  - e. On the **Network Policy and Access Services** page, click **Next**.
  - f. On the **Select Role Services** page, click **Routing and Remote Access Services**, leave **Remote Access Service** and **Routing** selected, and then click **Next**.



- g. On the **Confirm Installation Selections** page, click **Install**.

- h. When the wizard completes, click **Close**.

**To configure and enable Routing and Remote Access Server**

1. In the Server Manager navigation pane, expand **Roles**, and then expand **Network Policy and Access**.
2. Right-click **Routing and Remote Access Server**, and then click **Configure and Enable Routing and Remote Access**.
3. In the **Routing and Remote Access Setup Wizard**, on the **Welcome** page, click **Next**.
4. On the **Configuration** page, click **Custom Configuration**, and then click **Next**.
5. Click **LAN routing**, and then click **Next**.
6. Click **Finish**.
7. When prompted by the **Routing and Remote Access** dialog box, click **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 New Connection Security Rule Wizard on the Windows server.

### Important

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

However, you can't enable PFS using the Windows Server 2008 R2 user interface; you can only enable this setting by running the netsh script with `qmpfs=dhgroup2`. Therefore, you should consider your requirements before you pick an option. For more information, go to [Key exchange settings](#) in the *Microsoft TechNet Library*.

## Option 1: Run 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=Static_Route_IP_Prefix ^
Endpoint2=VPC_CIDR_Block Protocol=Any Action=RequireInClearOut ^
Auth1=ComputerPSK Auth1PSK=xCjNLsLoCmKsawkcdor9yX6Gsexample ^
QMSecMethods=ESP:SHA1-AES128+60min+100000kb ^
ExemptIPsecProtectedConnections=No ApplyAuthz=No QMPFS=dhgroup2
```

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

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

**Endpoint1:** The CIDR block of your network on which the Windows server resides, for example, 172.31.0.0/16.

**Endpoint2:** The CIDR block of your VPC or a subnet in your VPC, for example, 10.0.0.0/16.

Run the updated script in a command prompt window. (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 [2.4: Configure the Windows Firewall \(p. 196\)](#).

For more information about the netsh parameters, go to [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 2008 R2 user interface. Therefore, if you decide to use PFS, you must use the netsh scripts described in option 1 instead of the user interface described in this option.

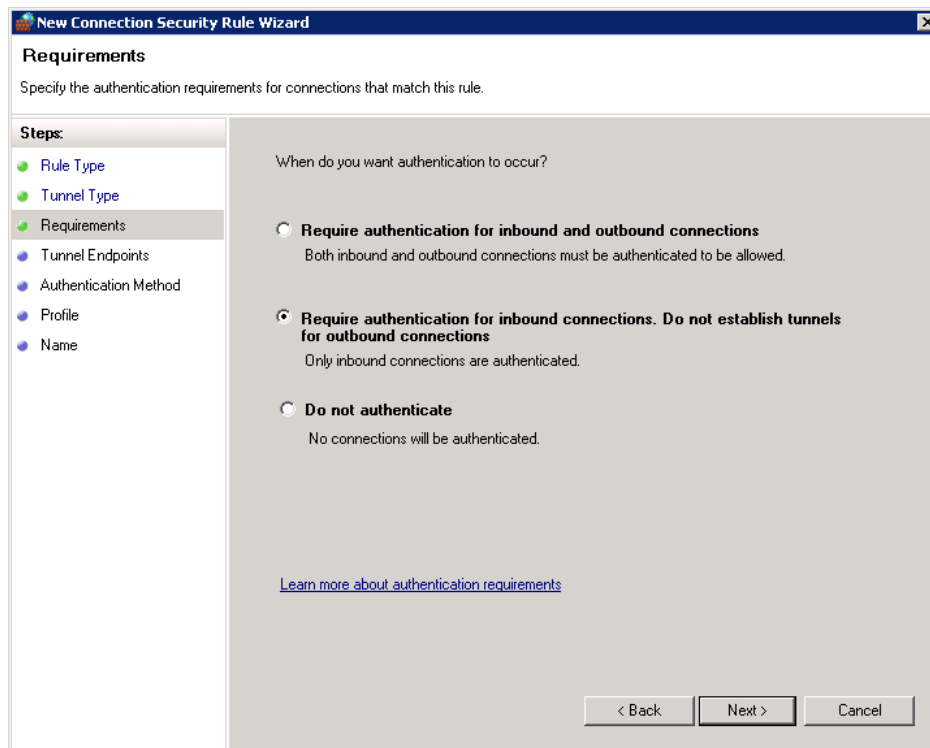
- [2.1: Configure a Security Rule for a VPN Tunnel \(p. 193\)](#)
- [2.3: Confirm the Tunnel Configuration \(p. 196\)](#)
- [2.4: Configure the Windows Firewall \(p. 196\)](#)

## 2.1: 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. In the Server Manager navigation pane, expand **Configuration**, and then expand **Windows Firewall with Advanced Security**.
2. Right-click **Connection Security Rules**, and then click **New Rule**.
3. In the **New Connection Security Rule** wizard, on the **Rule Type** page, click **Tunnel**, and then click **Next**.
4. On the **Tunnel Type** page, under **What type of tunnel would you like to create**, click **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 click **Next**.
5. On the **Requirements** page, click **Require authentication for inbound connections. Do not establish tunnels for outbound connections**, and then click **Next**.



6. On **Tunnel Endpoints** page, under **Which computers are in Endpoint 1**, click **Add**. Enter the CIDR range of your network (behind your Windows server customer gateway), and then click **OK**. (Note that the range can include the IP address of your customer gateway.)
7. Under **What is the local tunnel endpoint (closest to computer in Endpoint 1)**, click **Edit**. Enter the private IP address of your Windows server, and then click **OK**.
8. Under **What is the remote tunnel endpoint (closest to computers in Endpoint 2)**, click **Edit**. Enter the IP address of the virtual private gateway for Tunnel 1 from the configuration file (see Remote Tunnel Endpoint), and then click **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**, click **Add**. Enter the CIDR block of your VPC, and then click **OK**.

### Important

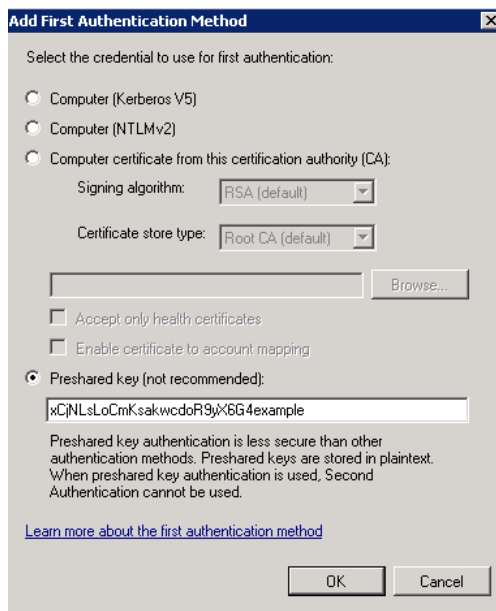
You must scroll in the dialog box until you locate **Which computers are in Endpoint 2**. Do not click **Next** until you have completed this step, or you won't be able to connect to your server.

The screenshot shows the 'New Connection Security Rule Wizard' window, specifically the 'Tunnel Endpoints' step. The title bar reads 'New Connection Security Rule Wizard'. Below the title bar, the section is 'Tunnel Endpoints' with the instruction 'Specify the endpoints for the IPsec tunnel defined by this rule.' On the left, a 'Steps:' pane lists: Rule Type, Tunnel Type, Requirements, Tunnel Endpoints (selected), Authentication Method, Profile, and Name. The main area is divided into sections for Endpoint 1 and Endpoint 2. For Endpoint 1, 'Which computers are in Endpoint 1?', there is a text box containing '172.31.0.0/16' and buttons for 'Add...', 'Edit...', and 'Remove'. Below this, 'What is the local tunnel endpoint (closest to computers in Endpoint 1)?' has fields for 'IPv4 address:' (172.31.12.34) and 'IPv6 address:', with an 'Edit...' button. A checkbox 'Apply IPsec tunnel authorization as specified on the IPsec Settings tab of the Windows Firewall with Advanced Security Properties dialog box.' is present. For Endpoint 2, 'What is the remote tunnel endpoint (closest to computers in Endpoint 2)?' has fields for 'IPv4 address:' (72.11.222.225) and 'IPv6 address:', with an 'Edit...' button. At the bottom, 'Which computers are in Endpoint 2?' has a text box containing '10.0.0.0/16' and an 'Add...' button. Navigation buttons at the bottom right are '< Back', 'Next >', and 'Cancel'.

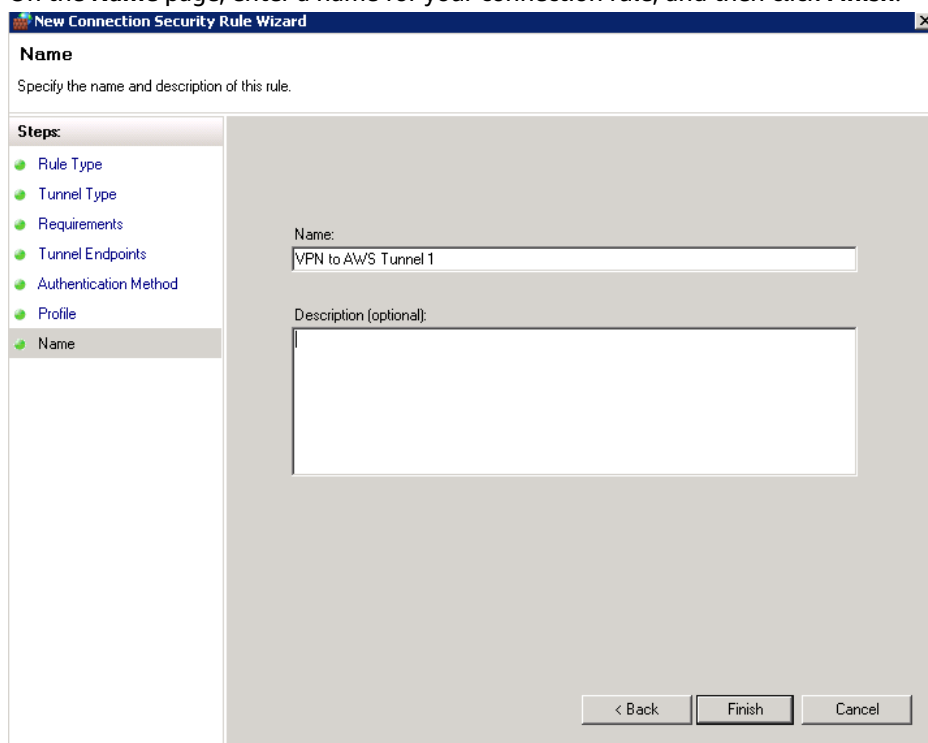
10. Confirm that all the settings you've specified are correct, and then click **Next**.
11. On the **Authentication Method** page, select **Advanced**, and then click **Customize**.
12. Under **First authentication methods**, click **Add**.
13. Select **Pre-Shared key**, enter the pre-shared key value from the configuration file, and click **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 click **OK**.
15. On the **Authentication Method** page, click **Next**.
16. On the **Profile** page, select all three check boxes: **Domain**, **Private**, and **Public**, and then click **Next**.
17. On the **Name** page, enter a name for your connection rule, and then click **Finish**.



Repeat the above 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.



## 2.3: Confirm the Tunnel Configuration

### To confirm the tunnel configuration

1. In the Server Manager navigation pane, expand the **Configuration** node, expand **Windows Firewall with Advanced Security**, and then click **Connection Security Rules**.
2. Verify the following for both tunnels:
  - **Enabled** is Yes
  - **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. Double-click the security rule for your first tunnel.
4. On the **Computers** tab, verify the following:
  - Under **Endpoint 1**, the CIDR block range shown matches the CIDR block range of your network.
  - Under **Endpoint 2**, the CIDR block range shown matches the CIDR block range of your VPC.
5. On the **Authentication** tab, under **Method**, click **Customize**, and verify that **First authentication methods** contains the correct pre-shared key from your configuration file for the tunnel, and then click **OK**.
6. On the **Advanced** tab, verify that **Domain**, **Private**, and **Public** are all selected.
7. Under **IPsec tunneling**, click **Customize**. Verify the following IPsec tunneling settings.
  - **Use IPsec tunneling** is selected.
  - **Local tunnel endpoint (closest to Endpoint 1)** contains the IP address of your server. If your customer gateway is a Windows server 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.
8. Double-click the security rule for your second tunnel. Repeat steps 4 to 7 for this tunnel.

## 2.4: 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.

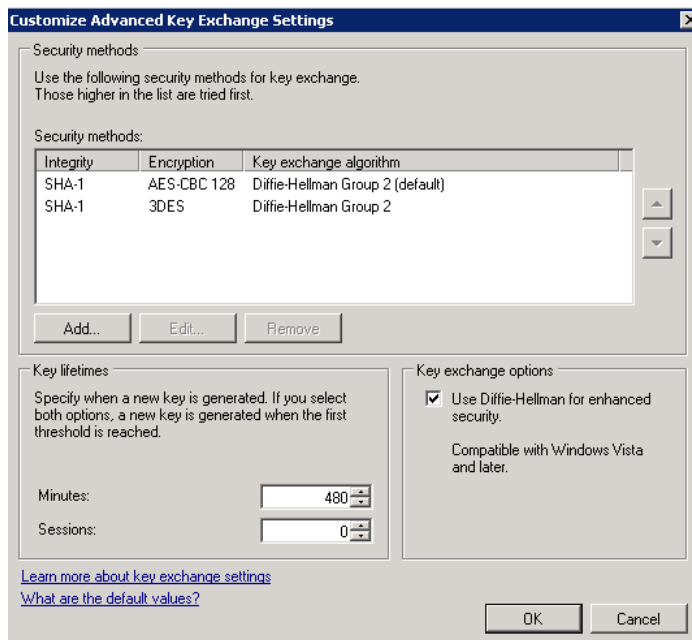
### To configure the Windows firewall

1. In the Server Manager navigation pane, right-click **Windows Firewall with Advanced Security**, and then click **Properties**.
2. Click the **IPsec Settings** tab.
3. Under **IPsec exemptions**, verify that **Exempt ICMP from IPsec** is **No (default)**. Verify that **IPsec tunnel authorization** is **None**.
4. Under **IPsec defaults**, click **Customize**.
5. In the **Customize IPsec Settings** dialog box, under **Key exchange (Main Mode)**, select **Advanced** and then click **Customize**.
6. In **Customize Advanced Key Exchange Settings**, under **Security methods**, verify that these 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
```



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

These value correspond to the following entries from the configuration file.

```
QuickModeSecMethods:
ESP:SHA1-AES128+60min+100000kb,ESP:SHA1-3DES+60min+100000kb
```

11. Click **OK** to return to the **Customize IPsec Settings** dialog box and click **OK** again to save the configuration .

## Step 5: Enable Dead Gateway Detection

Next, you need to 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. On the server, click **Start**, and then type **regedit** to start Registry Editor.
2. Expand **HKEY\_LOCAL\_MACHINE**, expand **SYSTEM**, expand **CurrentControlSet**, expand **Services**, expand **Tcpip**, and then expand **Parameters**.
3. In the other pane, right-click, point to **New**, and select **DWORD (32-bit) Value**.
4. Enter the name **EnableDeadGWDetect**.
5. Right-click **EnableDeadGWDetect**, and click **Modify**.
6. In **Value data**, enter **1**, and then click **OK**.
7. Close Registry Editor and reboot the server.

For more information, go to [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, therefore the ping command also initiates the VPN connection.

### To launch an instance in your VPC and get its private IP address

1. Open the Amazon EC2 console, and click **Launch Instance**.
2. Select an Amazon Linux AMI, and select an instance type.
3. On the **Step3: Configure Instance Details** page, select your VPC from the **Network** list, and select a subnet from the **Subnet** list. Ensure that you select the private subnet that you configured in [Step 1: Create a VPN Connection and Configure Your VPC \(p. 187\)](#).
4. In the **Auto-assign Public IP** list, ensure that the setting is set to **Disable**.
5. Click **Next** until you get to the **Step 6: Configure Security Group** page. You can either select an existing security group that you configured in [Step 1: Create a VPN Connection and Configure Your VPC \(p. 187\)](#), or you can create a new security group and ensure that it has a rule that allows all ICMP traffic from the IP address of your Windows server.
6. Complete the rest of the steps in the wizard, and launch your instance.
7. On the **Instances** page, select your instance. Get the private IP address in the **Private IPs** field on the details pane.

Connect to or log on to your Windows server, open the command prompt, and then use the `ping` command to ping your instance using its private IP address; for example:

```
ping 10.0.0.4
```

```
Pinging 10.0.0.4 with 32 bytes of data:
```

```
Reply from 10.0.0.4: bytes=32 time=2ms TTL=62
Reply from 10.0.0.4: bytes=32 time=2ms TTL=62
Reply from 10.0.0.4: bytes=32 time=2ms TTL=62
Reply from 10.0.0.4: bytes=32 time=2ms TTL=62

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 2ms, Average = 2ms
```

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 Server \(p. 186\)](#).
- Ensure that the operating system on the instance you are pingping is configured to respond to ICMP. We recommend that you use one of the Amazon Linux AMIs.
- If the instance you are pingping is a Windows instance, log in to the instance and enable inbound ICMPv4 on the Windows firewall.
- Ensure that you have configured the route tables for your VPC or your subnet correctly. For more information, see [Step 1: Create a VPN Connection and Configure Your VPC \(p. 187\)](#).
- If your customer gateway is a Windows server instance, ensure that you've disabled source/destination checking for the instance. For more information, see [Configuring Your Windows Server \(p. 186\)](#).

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 won't be used unless the first tunnel goes down. It may take a few moments to establish the encrypted tunnels.

# Configuring Windows Server 2012 R2 as a Customer Gateway

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

## Topics

- [Configuring Your Windows Server](#) (p. 200)
- [Step 1: Create a VPN Connection and Configure Your VPC](#) (p. 201)
- [Step 2: Download the Configuration File for the VPN Connection](#) (p. 202)
- [Step 3: Configure the Windows Server](#) (p. 204)
- [Step 4: Set Up the VPN Tunnel](#) (p. 205)
- [Step 5: Enable Dead Gateway Detection](#) (p. 210)
- [Step 6: Test the VPN Connection](#) (p. 211)

## Configuring Your Windows Server

To configure Windows Server as a customer gateway, ensure that you have Windows Server 2012 R2 on your own network, or on an EC2 instance in a VPC. If you use 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 Server instance, and choose **Actions, Networking, Change Source/Dest. Check**. Choose **Yes, Disable**.
- 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 AWS PV network device, 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**.
- Associate an Elastic IP address with the instance:
    1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
    2. In the navigation pane, choose **Elastic IPs**. Choose **Allocate new address**.
    3. Select the Elastic IP address, and choose **Actions, Associate Address**.
    4. For **Instance**, select your Windows Server instance. Choose **Associate**.

Take note of this address — you will need it when you create the customer gateway in your VPC.

- Ensure 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 for your network in which the Windows server 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, you must first create a virtual private gateway and attach it to your VPC. Then you can create a VPN connection and configure your VPC. You must also have the CIDR range for your network in which the Windows server is located, for example, 172.31.0.0/16.

### To create a virtual private gateway

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Virtual Private Gateways**, and then **Create Virtual Private Gateway**.
3. You can optionally enter a name for your virtual private gateway, and then choose **Yes, Create**.
4. Select the virtual private gateway that you created, and then choose **Attach to VPC**.
5. In the **Attach to VPC** dialog box, select your VPC from the list, and then choose **Yes, Attach**.

### To create a VPN connection

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **VPN Connections**, and then **Create VPN Connection**.
3. Select the virtual private gateway from the list.
4. For **Customer Gateway**, choose **New**. For **IP address**, specify the public IP address of your Windows Server.

#### Note

The IP address must be static and may be behind a device performing network address translation (NAT). To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500. If your customer gateway is an EC2 Windows Server instance, use its Elastic IP address.

5. Select the **Static** routing option, enter the **Static IP Prefixes** values for your network in CIDR notation, and then choose **Yes, Create**.

### To configure your VPC

- Create a private subnet in your VPC (if you don't have one already) for launching instances that will communicate with the Windows server. For more information, see [Adding a Subnet to 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.
  - Enable route propagation for the virtual private gateway. For more information, see [Route Tables](#) in the *Amazon VPC User Guide*.
- Create a security group configuration 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 **VPN Connections**.
3. Select your VPN connection, and then choose **Download Configuration**.
4. Select **Microsoft** as the vendor, **Windows Server** as the platform, and **2012 R2** as the software. Choose **Yes, Download**. You can open the file or save it.

The configuration file contains a section of information similar to the following example. You'll see this information presented twice, one time for each tunnel. You'll use this information when configuring the Windows Server 2012 R2 server.

```
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:             xCjNLSLoCmKsawkcdR9yX6GsEXAMPLE
```

#### 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 is a Windows server 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 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 AWS virtual private gateway within a matter of minutes. When you configure your customer gateway, it's important that you configure both tunnels.

#### Note

From time to time, AWS performs routine maintenance on the virtual private gateway. This maintenance may 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. Because the VPC VPN suggested settings are the same as the Windows Server 2012 R2 default IPsec configuration settings, minimal work is needed on your part.

<code>MainModeSecMethods:</code>	<code>DHGroup2-AES128-SHA1</code>
<code>MainModeKeyLifetime:</code>	<code>480min,0sess</code>
<code>QuickModeSecMethods:</code>	<code>ESP:SHA1-AES128+60min+100000kb</code>
<code>QuickModePFS:</code>	<code>DHGroup2</code>

#### 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 2012 R2 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 2012 R2 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 2012 R2 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 your Windows server to allow remote users to access resources on your network.

### To install Routing and Remote Access Services on Windows Server 2012 R2

1. Log on to the Windows Server 2012 R2 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 2012 R2 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**, and then choose **Next**.
  - g. On the **Select features** page, choose **Next**.
  - h. On the **Network Policy and Access Services** page, choose **Next**. Leave **Network Policy Server** selected, and 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 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** and **Next**.
6. Choose **LAN routing**, **Next**, and **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 New Connection Security Rule wizard on the Windows server.

### Important

We suggest that you use master key perfect forward secrecy (PFS) for your IPsec sessions. For more information, go to [Key exchange settings](#) in the *Microsoft TechNet Library*. If you choose to run the netsh script, it includes a parameter to enable PFS (`qmpfs=dhgroup2`). You cannot enable PFS using the Windows Server 2012 R2 user interface — you must enable it using the command line.

## Option 1: Run 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=xCjNLsLoCmKsawcdor9yX6GsEXAMPLE ^
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`.

**Endpoint2:** The CIDR block of your VPC or a subnet in your VPC, for example, `10.0.0.0/16`.

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 [2.4: Configure the Windows Firewall \(p. 209\)](#).

For more information about the netsh parameters, go to [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 2012 R2 user interface. You must enable PFS using the command line, as described in [Enable Master Key Perfect Forward Secrecy \(p. 208\)](#).

### Topics

- [2.1: Configure a Security Rule for a VPN Tunnel \(p. 206\)](#)

- [2.3: Confirm the Tunnel Configuration \(p. 208\)](#)
- [Enable Master Key Perfect Forward Secrecy \(p. 208\)](#)

## 2.1: 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 select **Windows 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 **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; for example, 172.31.0.0/16 ), and then choose **OK**. (Note that the range can include the IP address of your customer gateway.)
7. Under **What is the local tunnel endpoint (closest to computer in Endpoint 1)**, choose **Edit**. In the **IPv4 address** field, enter the private IP address of your Windows server, and then choose **OK**.
8. Under **What is the remote tunnel endpoint (closest to computers in Endpoint 2)**, choose **Edit**. In the **IPv4 address** field, enter the IP address of the virtual private gateway for Tunnel 1 from the configuration file (see `Remote Tunnel Endpoint`), and then choose **OK**.

#### Important

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

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

#### Important

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

The screenshot shows the 'New Connection Security Rule Wizard' with the 'Tunnel Endpoints' step selected in the left-hand 'Steps' pane. The main area is titled 'Specify the endpoints for the IPsec tunnel defined by this rule.' It contains two sections for configuring endpoints. The first section, 'Which computers are in Endpoint 1?', has a text box containing '172.31.0.0/16' and buttons for 'Add...', 'Edit...', and 'Remove'. Below this, 'What is the local tunnel endpoint (closest to computers in Endpoint 1)?' has input fields for 'IPv4 address' (172.31.13.36) and 'IPv6 address' (empty), with an 'Edit...' button. A checkbox 'Apply IPsec tunnel authorization as specified on the IPsec Settings tab of Windows Firewall with Advanced Security Properties.' is present. The second section, 'What is the remote tunnel endpoint (closest to computers in Endpoint 2)?', has input fields for 'IPv4 address' (54.240.204.89) and 'IPv6 address' (empty), with an 'Edit...' button. Below this, 'Which computers are in Endpoint 2?' has a text box containing '10.0.0.0/16' and an 'Add...' button. At the bottom right are '< Back' and 'Next >' buttons.

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

## 2.3: 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**, and 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 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. Type the following command, replacing `rule_name` with the name you gave the first connection rule.

```
netsh advfirewall consec set rule name="rule_name" new QMPFS=dhgroup2
QMPSecMethods=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.

## 2.4: 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 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 these 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 value 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, you need to 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 type **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 **Modify** from the **Edit** menu.
6. In **Value data**, enter **1**, and then choose **OK**.
7. Close the Registry Editor and reboot the server.

For more information, go to [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, therefore the ping command also initiates the VPN connection.

### To launch an instance in your VPC and get its private IP address

1. Open the Amazon EC2 console, and choose **Launch Instance**.
2. Select an Amazon Linux AMI, and select an instance type.
3. On the **Step 3: Configure Instance Details** page, select your VPC from the **Network** list, and select a subnet from the **Subnet** list. Ensure that you select the private subnet that you configured in [Step 1: Create a VPN Connection and Configure Your VPC \(p. 201\)](#).
4. In the **Auto-assign Public IP** list, ensure that the setting is set to **Disable**.
5. Choose **Next** until you get to the **Step 6: Configure Security Group** page. You can either select an existing security group that you configured in [Step 1: Create a VPN Connection and Configure Your VPC \(p. 201\)](#), or you can create a new security group and ensure that it has a rule that allows all ICMP traffic from the IP address of your Windows server.
6. Complete the rest of the steps in the wizard, and launch your instance.
7. On the **Instances** page, select your instance. Get the private IP address in the **Private IPs** field on the details pane.

Connect to or log on to your Windows server, open the command prompt, and then use the `ping` command to ping your instance using its private IP address; for example:

```
ping 10.0.0.4
```

```
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=2ms TTL=62
Reply from 10.0.0.4: bytes=32 time=2ms TTL=62
Reply from 10.0.0.4: bytes=32 time=2ms TTL=62
Reply from 10.0.0.4: bytes=32 time=2ms TTL=62

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 2ms, Average = 2ms
```

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 Server \(p. 200\)](#).
- Ensure that the operating system on the instance you are ping is configured to respond to ICMP. We recommend that you use one of the Amazon Linux AMIs.
- If the instance you are ping 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 \(p. 201\)](#).
- If your customer gateway is a Windows server instance, ensure that you've disabled source/destination checking for the instance. For more information, see [Configuring Your Windows Server \(p. 200\)](#).



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 won't be used unless the first tunnel goes down. It may take a few moments to establish the encrypted tunnels.

# Document History

The following table describes the important changes in each release of this Amazon VPC guide.

Change	Description	Release Date
VPN tunnel configuration options	When you create a VPN connection, you can specify the inside tunnel CIDR blocks and your own custom pre-shared keys for each tunnel. For more information, see <a href="#">Overview of Setting Up a VPN Connection (p. 4)</a> .	3 October 2017
VPN enhancements	A VPN connection now supports the AES 256-bit encryption function, SHA-256 hashing function, NAT traversal, and additional Diffie-Hellman groups during Phase 1 and Phase 2 of a connection. In addition, you can now use the same customer gateway IP address for each VPN connection that uses the same customer gateway device.	28 October 2015
VPN connections using static routing configuration	With this release, you can now 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.	13 September 2012
Automatic route propagation	You can now configure automatic propagation of routes from your VPN and Direct Connect links to your VPC routing tables. This feature simplifies the effort to create and maintain connectivity to Amazon VPC.	13 September 2012
AWS VPN CloudHub and redundant VPN connections	With this release, the network administrator's guide has been updated with information about AWS VPN CloudHub, which you can use to securely communicate from one site to another with or without a VPC, and updated with information about using redundant VPN connections to provide a fault-tolerant connection to your VPC.	29 September 2011
VPC Everywhere	With this release, the network administrator's guide has been rewritten to reflect the new features available in the 2011-07-15 API version.	03 August 2011

Change	Description	Release Date
Added MTU Support Information	Added information about support for Maximum Transmission Unit (MTU). For more information, see the <i>Bind tunnel to logical interface (route-based VPN)</i> requirement in <a href="#">Requirements for Your Customer Gateway (p. 7)</a> .	04 May 2011
Updates to Configuration Templates	Updated the configuration templates to include information about encrypting packages after fragmentation. Also removed information about VRF from the Cisco configuration and removed information about the routing instance (RI) from the Juniper JunOS configuration.	15 February 2011