AWS Direct Connect

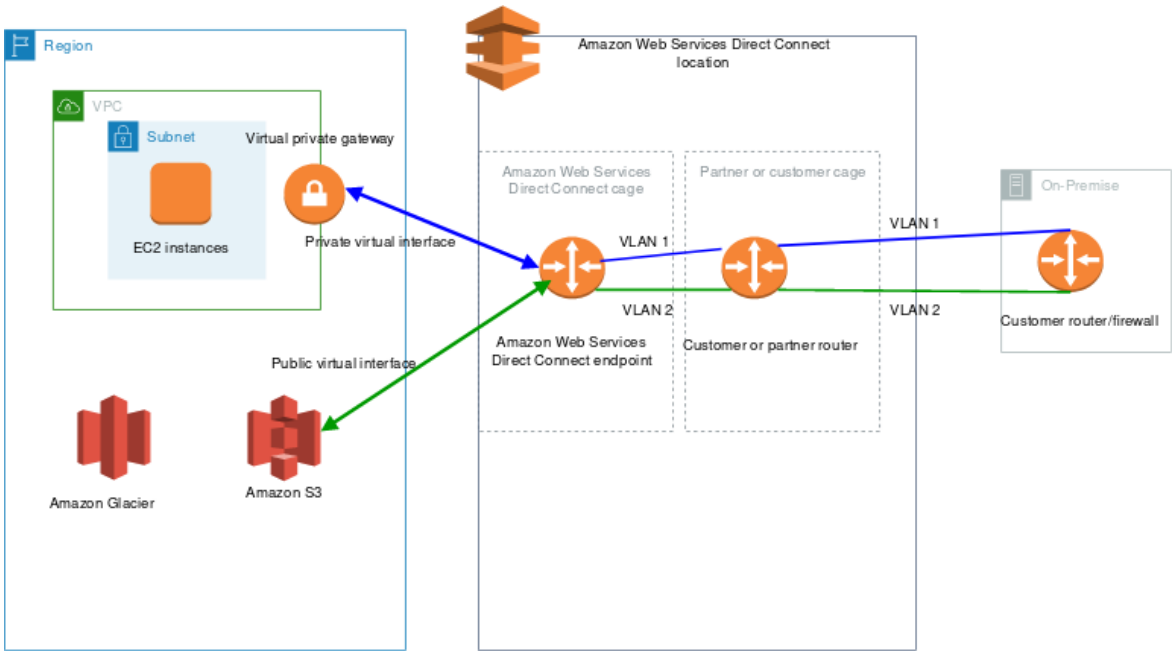
User Guide

Table of Contents

What is AWS Direct Connect?

AWS Direct Connect links your internal network to an AWS Direct Connect location over a standard Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to an AWS Direct Connect router. With this connection, you can create *virtual interfaces* directly to public AWS services (for example, to Amazon S3) or to Amazon VPC, bypassing internet service providers in your network path. An AWS Direct Connect location provides access to AWS in the Region with which it is associated. You can use a single connection in a public Region or AWS GovCloud (US) to access public AWS services in all other public Regions.

The following diagram shows how AWS Direct Connect interfaces with your network.



Contents

AWS Direct Connect components

The following are the key components that you use for AWS Direct Connect:

Connections

Create a *connection* in an AWS Direct Connect location to establish a network connection from your premises to an AWS Region. For more information, see AWS Direct Connect connections (p. 45).

Virtual interfaces

Create a *virtual interface* to enable access to AWS services. A public virtual interface enables access to public services, such as Amazon S3. A private virtual interface enables access to your VPC. For more information, see AWS Direct Connect virtual interfaces (p. 61) and Prerequisites for virtual interfaces (p. 63).

Network requirements

To use AWS Direct Connect in an AWS Direct Connect location, your network must meet one of the following conditions:

- Your network is colocated with an existing AWS Direct Connect location. For more information about available AWS Direct Connect locations, see AWS Direct Connect Product Details.
- You are working with an AWS Direct Connect partner who is a member of the AWS Partner Network (APN). For information, see APN Partners Supporting AWS Direct Connect.
- You are working with an independent service provider to connect to AWS Direct Connect.

In addition, your network must meet the following conditions:

- Your network must use single-mode fiber with a 1000BASE-LX (1310 nm) transceiver for 1 gigabit Ethernet, a 10GBASE-LR (1310 nm) transceiver for 10 gigabit, or a 100GBASE-LR4 for 100 gigabit Ethernet.
- Auto-negotiation for the port must be disabled. Auto-negotiation is supported only if the port speed is 1 Gbps. Port speed and full-duplex mode must be configured manually.
- 802.1Q VLAN encapsulation must be supported across the entire connection, including intermediate devices.
- Your device must support Border Gateway Protocol (BGP) and BGP MD5 authentication.
- (Optional) You can configure Bidirectional Forwarding Detection (BFD) on your network. Asynchronous BFD is automatically enabled for AWS Direct Connect virtual interfaces, but does not take effect until you configure it on your router.

AWS Direct Connect supports both the IPv4 and IPv6 communication protocols. IPv6 addresses provided by public AWS services are accessible through AWS Direct Connect public virtual interfaces.

AWS Direct Connect supports an Ethernet frame size of 1522 or 9023 bytes (14 bytes Ethernet header + 4 bytes VLAN tag + bytes for the IP datagram + 4 bytes FCS) at the link layer. You can set the MTU of your private virtual interfaces. For more information, see Set network MTU for private virtual interfaces or transit virtual interfaces (p. 72).

Pricing for AWS Direct Connect

AWS Direct Connect has two billing elements: port hours and outbound data transfer. Port hour pricing is determined by capacity and connection type (dedicated connection or hosted connection).

Data Transfer Out charges for private interfaces and transit virtual interfaces are allocated to the AWS account responsible for the Data Transfer. There are no additional charges to use a multi-account AWS Direct Connect gateway.

For publicly addressable AWS resources (for example, Amazon S3 buckets, Classic EC2 instances, or EC2 traffic that goes through an internet gateway), if the outbound traffic is destined for public prefixes owned by the same AWS payer account and actively advertised to AWS through an AWS Direct Connect

public virtual Interface, the Data Transfer Out (DTO) usage is metered toward the resource owner at AWS Direct Connect data transfer rate.

For more information, see Amazon Direct Connect Pricing.

Accessing a remote AWS Region

AWS Direct Connect locations in public Regions or AWS GovCloud (US) can access public services in any other public Region (excluding China (Beijing and Ningxia)). In addition, AWS Direct Connect connections in public Regions or AWS GovCloud (US) can be configured to access a VPC in your account in any other public Region (excluding China (Beijing and Ningxia). You can therefore use a single AWS Direct Connect connection to build multi-Region services. All networking traffic remains on the AWS global network backbone, regardless of whether you access public AWS services or a VPC in another Region.

Any data transfer out of a remote Region is billed at the remote Region data transfer rate. For more information about data transfer pricing, see the Pricing section on the AWS Direct Connect detail page.

For more information about the routing policies and supported BGP communities for an AWS Direct Connect connection, see Routing policies and BGP communities (p. 3).

Accessing public services in a remote Region

To access public resources in a remote Region, you must set up a public virtual interface and establish a Border Gateway Protocol (BGP) session. For more information, see AWS Direct Connect virtual interfaces (p. 61).

After you have created a public virtual interface and established a BGP session to it, your router learns the routes of the other public AWS Regions. For more information about prefixes currently advertised by AWS, see AWS IP Address Ranges in the *Amazon Web Services General Reference*.

Accessing VPCs in a remote Region

You can create a *Direct Connect gateway* in any public Region. Use it to connect your AWS Direct Connect connection over a private virtual interface to VPCs in your account that are located in different Regions or to a transit gateway. For more information, see Working with Direct Connect gateways (p. 88).

Alternatively, you can create a public virtual interface for your AWS Direct Connect connection and then establish a VPN connection to your VPC in the remote Region. For more information about configuring VPN connectivity to a VPC, see Scenarios for Using Amazon Virtual Private Cloud in the *Amazon VPC User Guide*.

Network-to-Amazon VPC Connectivity Options

The following configuration can be used to connect remote networks with your Amazon VPC environment. These options are useful for integrating AWS resources with your existing on-site services:

- AWS Direct Connect
- AWS Direct Connect and AWS Virtual Private Network

Routing policies and BGP communities

AWS Direct Connect applies inbound (from your on-premises data center) and outbound (from your AWS Region) routing policies for a public AWS Direct Connect connection. You can also use Border Gateway

Protocol (BGP) community tags on routes advertised by Amazon and apply BGP community tags on the routes you advertise to Amazon.

Public virtual interface routing policies

If you're using AWS Direct Connect to access public AWS services, you must specify the public IPv4 prefixes or IPv6 prefixes to advertise over BGP.

The following inbound routing policies apply:

- You must own the public prefixes and they must be registered as such in the appropriate regional internet registry.
- Traffic must be destined to Amazon public prefixes. Transitive routing between connections is not supported.
- AWS Direct Connect performs inbound packet filtering to validate that the source of the traffic originated from your advertised prefix.

The following outbound routing policies apply:

- AS_PATH and Longest Prefix Match is used to determine the routing path, and AWS Direct Connect is the preferred path for traffic sourced from Amazon.
- AWS Direct Connect advertises all local and remote AWS Region prefixes where available and includes on-net prefixes from other AWS non-Region points of presence (PoP) where available; for example, CloudFront and Route 53.
- AWS Direct Connect advertises prefixes with a minimum path length of 3.
- AWS Direct Connect advertises all public prefixes with the well-known `NO_EXPORT` BGP community.
- If you have multiple AWS Direct Connect connections, you can adjust the load-sharing of inbound traffic by advertising prefixes with similar path attributes.
- The prefixes advertised by AWS Direct Connect must not be advertised beyond the network boundaries of your connection. For example, these prefixes must not be included in any public internet routing table.
- AWS Direct Connect keeps prefixes advertised by customers within the Amazon network. We do not re-advertise customer prefixes learned from a public VIF to any of the following:
  - Other AWS Direct Connect customers
  - Networks that peer with the AWS Global Network
  - Amazon's transit providers

Public virtual interface BGP communities

AWS Direct Connect supports scope BGP community tags and the `NO_EXPORT` BGP community tag to help control the scope (Regional or global) and route preference of traffic on public virtual interfaces.

Scope BGP communities

You can apply BGP community tags on the public prefixes that you advertise to Amazon to indicate how far to propagate your prefixes in the Amazon network, for the local AWS Region only, all Regions within a continent, or all public Regions.

You can use the following BGP communities for your prefixes:

- `7224:9100`—Local AWS Region
- `7224:9200`—All AWS Regions for a continent

  - North America–wide
  - Asia Pacific
  - Europe, the Middle East and Africa
- `7224:9300`—Global (all public AWS Regions)

  Note
  If you do not apply any community tags, prefixes are advertised to all public AWS Regions (global) by default.
  Prefixes that are marked with the same communities, and have identical AS_PATH attributes are candidates for multi-pathing.

The communities `7224:1` – `7224:65535` are reserved by AWS Direct Connect.

AWS Direct Connect applies the following BGP communities to its advertised routes:

- `7224:8100`—Routes that originate from the same AWS Region in which the AWS Direct Connect point of presence is associated.
- `7224:8200`—Routes that originate from the same continent with which the AWS Direct Connect point of presence is associated.
- No tag—Global (all public AWS Regions).

Communities that are not supported for an AWS Direct Connect public connection are removed.

`NO_EXPORT` BGP community

The `NO_EXPORT` BGP community tag is supported for public virtual interfaces.

AWS Direct Connect also provides BGP community tags on advertised Amazon routes. If you use AWS Direct Connect to access public AWS services, you can create filters based on these community tags.

For public virtual interfaces, all routes that AWS Direct Connect advertises to customers are tagged with the NO_EXPORT community tag.

Private virtual interface and transit virtual interface routing policies

The following routing rules apply to traffic on private virtual interfaces and transit virtual interfaces:

- AWS evaluates the longest prefix match first
- By default, AWS uses the distance from the local Region to the AWS Direct Connect to determine the virtual (or transit) interface for routing. You can modify this behavior by assigning local preference communities to virtual interfaces.
- When you have multiple virtual interfaces in a Region, you can set the AS_PATH attribute to prioritize which interface AWS uses to route traffic.

Private virtual interface and transit virtual interface BGP communities

AWS Direct Connect supports local preference BGP community tags to help control the route preference of traffic on private virtual interfaces and transit virtual interfaces.

For an example of a private virtual interface configuration, see the section called “Private virtual interface routing example” (p. 6).

For an example of a transit virtual interface configuration, see the section called “Example: Allowed to prefixes in a transit gateway configuration” (p. 104).

Local preference BGP communities

You can use local preference BGP community tags to achieve load balancing and route preference for incoming traffic to your network. For each prefix that you advertise over a BGP session, you can apply a community tag to indicate the priority of the associated path for returning traffic.

The following local preference BGP community tags are supported:

- `7224:7100`—Low preference
- `7224:7200`—Medium preference
- `7224:7300`—High preference

Local preference BGP community tags are mutually exclusive. To load balance traffic across multiple AWS Direct Connect connections, apply the same community tag across the prefixes for the connections. To support failover across multiple AWS Direct Connect connections, apply a community tag with a higher preference to the prefixes for the primary or active virtual interface. For example set the BGP community tags for your primary or active virtual interfaces to 7224:7300 (high preference).

Local preference BGP community tags are evaluated before any AS_PATH attribute, and are evaluated in order from lowest to highest preference (where highest preference is preferred).

If you do not specify local preference using BGP community tags, the default outbound routing behavior is based on the AWS Direct Connect locations' relative distance to the originating Region.

`NO_EXPORT` BGP community

The `NO_EXPORT` BGP community tag is supported for public virtual interfaces, private virtual interfaces, and transit virtual interfaces.

AWS Direct Connect also provides BGP community tags on advertised Amazon routes. If you use AWS Direct Connect to access public AWS services, you can create filters based on these community tags.

Virtual interface BGP community tags

You can use this traffic engineering technique for private and transit virtual interface BGP advertisements to achieve an active/passive load distribution over redundant virtual interfaces. An active virtual interface has 7224:7300 (High preference) as the tag and a Passive has 7224:7100 (Low preference) as the tag.

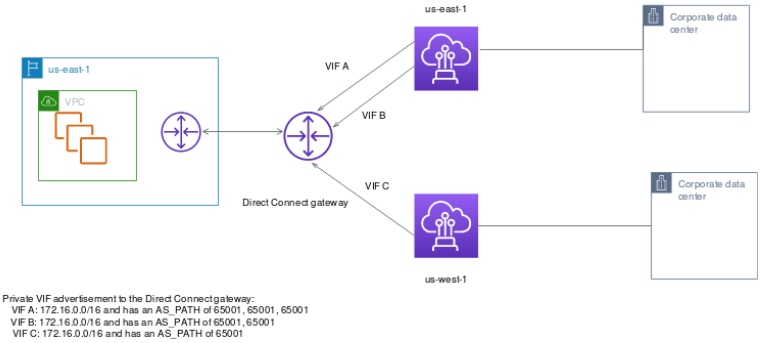
Private virtual interface routing example

Consider the configuration where the AWS Direct Connect location 1 home Region (east) is the same as the VPC home Region. There is a redundant AWS Direct Connect location 2 in a different Region (west). There are two private VIFs from AWS Direct Connect location 1 to the Direct Connect gateway. There is one private VIF from AWS Direct Connect location 2 to the Direct Connect gateway. To have AWS route traffic over VIF B before VIF A, set the AS_PATH attribute of VIF B to be shorter than the VIF A AS_PATH attribute.

The VIFs have the following configurations:

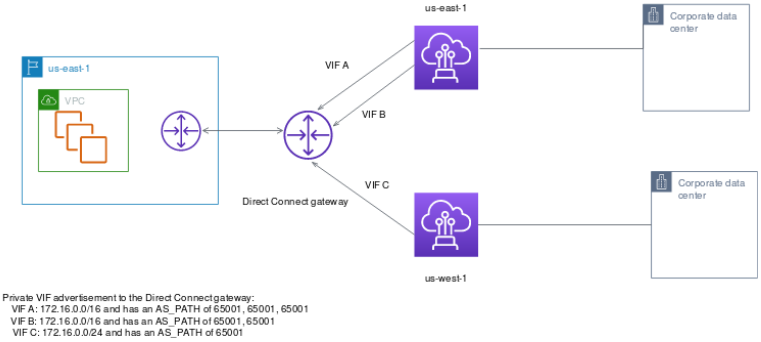
- VIF A (in us-east-1) advertises 172.16.0.0/16 and has an AS_PATH attribute of 65001, 65001, 65001

- VIF B (in us-east-1) advertises 172.16.0.0/16 and has an AS_PATH attribute of 65001, 65001
- VIF C (in us-west-1) advertises 172.16.0.0/16 and has an AS_PATH attribute of 65001



If you change the CIDR range configuration of VIF C, routes that fall in to the VIF C CIDR range use VIF C because it has the longest prefix match.

- VIF C (in us-west-1) advertises 172.16.0.0/24 and has an AS_PATH attribute of 65001



Using the AWS Direct Connect Resiliency Toolkit to get started

AWS offers customers the ability to achieve highly resilient network connections between Amazon Virtual Private Cloud (Amazon VPC) and their on-premises infrastructure. The AWS Direct Connect Resiliency Toolkit provides a connection wizard with multiple resiliency models. These models help you to determine, and then place an order for the number of dedicated connections to achieve your SLA objective. You select a resiliency model, and then the AWS Direct Connect Resiliency Toolkit guides you through the dedicated connection ordering process. The resiliency models are designed to ensure that you have the appropriate number of dedicated connections in multiple locations.

The AWS Direct Connect Resiliency Toolkit has the following benefits:

- Provides guidance on how you determine and then order the appropriate redundant AWS Direct Connect dedicated connections.
- Ensures that the redundant dedicated connections have the same speed.
- Automatically configures the dedicated connection names.
- Automatically approves your dedicated connections when you have an existing AWS account and you select a known AWS Direct Connect Partner. The Letter of Authority (LOA) is available for immediate download.
- Automatically creates a support ticket for the dedicated connection approval when you are a new AWS customer, or you select an unknown (**Other**) partner.
- Provides an order summary for your dedicated connections, with the SLA that you can achieve and the port-hour cost for the ordered dedicated connections.
- Creates link aggregation groups (LAGs), and adds the appropriate number of dedicated connections to the LAGs when you choose a speed other than 1 Gbps, 10 Gbps, or 100 Gbps.
- Provides a LAG summary with the dedicated connection SLA that you can achieve, and the total port-hour cost for each ordered dedicated connection as part of the LAG.
- Prevents you from terminating the dedicated connections on the same AWS Direct Connect device.
- Provides a way for you to test your configuration for resiliency. You work with AWS to bring down the BGP peering session in order to verify that traffic routes to one of your redundant virtual interfaces. For more information, see the section called “AWS Direct Connect Failover Test” (p. 39).
- Provides Amazon CloudWatch metrics for connections and virtual interfaces. For more information, see *Monitoring* (p. 139).

The following resiliency models are available in the AWS Direct Connect Resiliency Toolkit:

- **Maximum Resiliency**: This model provides you a way to order dedicated connections to achieve an SLA of 99.99%. It requires you to meet all of the requirements for achieving the SLA that are specified in the AWS Direct Connect Level Agreement.
- **High Resiliency**: This model provides you a way to order dedicated connections to achieve an SLA of 99.9%. It requires you to meet all of the requirements for achieving the SLA that are specified in the AWS Direct Connect Service Level Agreement.
- **Development and Test**: This model provides you a way to achieve development and test resiliency for non-critical workloads, by using separate connections that terminate on separate devices in one location.
- **Classic**. This model is intended for users that have existing connections and want to add additional connections. This model does not provide an SLA.

The best practice is to use the **Connection wizard** in the AWS Direct Connect Resiliency Toolkit to order the dedicated connections to achieve your SLA objective.

After you select the resiliency model, the AWS Direct Connect Resiliency Toolkit steps you through the following procedures:

- Selecting the number of dedicated connections
- Selecting the connection capacity, and the dedicated connection location
- Ordering the dedicated connections
- Verifying that the dedicated connections are ready to use
- Downloading your Letter of Authority (LOA-CFA) for each dedicated connection
- Verifying that your configuration meets your resiliency requirements

Prerequisites

AWS Direct Connect supports the following port speeds over single-mode fiber: 1000BASE-LX (1310 nm) transceiver for 1 gigabit Ethernet, a 10GBASE-LR (1310 nm) transceiver for 10 gigabit, or a 100GBASE-LR4 for 100 gigabit Ethernet.

You can set up an AWS Direct Connect connection in one of the following ways:

Model	Bandwidth	Method
Dedicated connection	1 Gbps, 10 Gbps, and 100 Gbps	Work with an AWS Direct Connect Partner or a network provider to connect a router from your data center, office, or colocation environment to an AWS Direct Connect location. The network provider does not have to be an AWS Direct Connect Partner to connect you to a dedicated connection. AWS Direct Connect dedicated connections support these port speeds over single-mode fiber: 1 Gbps: 1000BASE-LX (1310 nm), 10 Gbps: 10GBASE-LR (1310 nm), and 100Gbps: 100GBASE-LR4.
Hosted connection	50 Mbps, 100 Mbps, 200 Mbps, 300 Mbps, 400 Mbps, 500 Mbps, 1 Gbps, 2 Gbps, 5 Gbps, and 10 Gbps	Work with a partner in the AWS Direct Connect Partner Program to connect a router from your data center, office, or colocation environment to an AWS Direct Connect location. Only certain partners provide higher capacity connections.

For connections to AWS Direct Connect with bandwidths of 1 Gbps or higher, ensure that your network meets the following requirements:

- Your network must use single-mode fiber with a 1000BASE-LX (1310 nm) transceiver for 1 gigabit Ethernet, a 10GBASE-LR (1310 nm) transceiver for 10 gigabit, or a 100GBASE-LR4 for 100 gigabit Ethernet.
- Auto-negotiation for the port must be disabled. Auto-negotiation is supported only if the port speed is 1 Gbps. Port speed and full-duplex mode must be configured manually.
- 802.1Q VLAN encapsulation must be supported across the entire connection, including intermediate devices.
- Your device must support Border Gateway Protocol (BGP) and BGP MD5 authentication.
- (Optional) You can configure Bidirectional Forwarding Detection (BFD) on your network. Asynchronous BFD is automatically enabled for AWS Direct Connect virtual interfaces, but does not take effect until you configure it on your router.

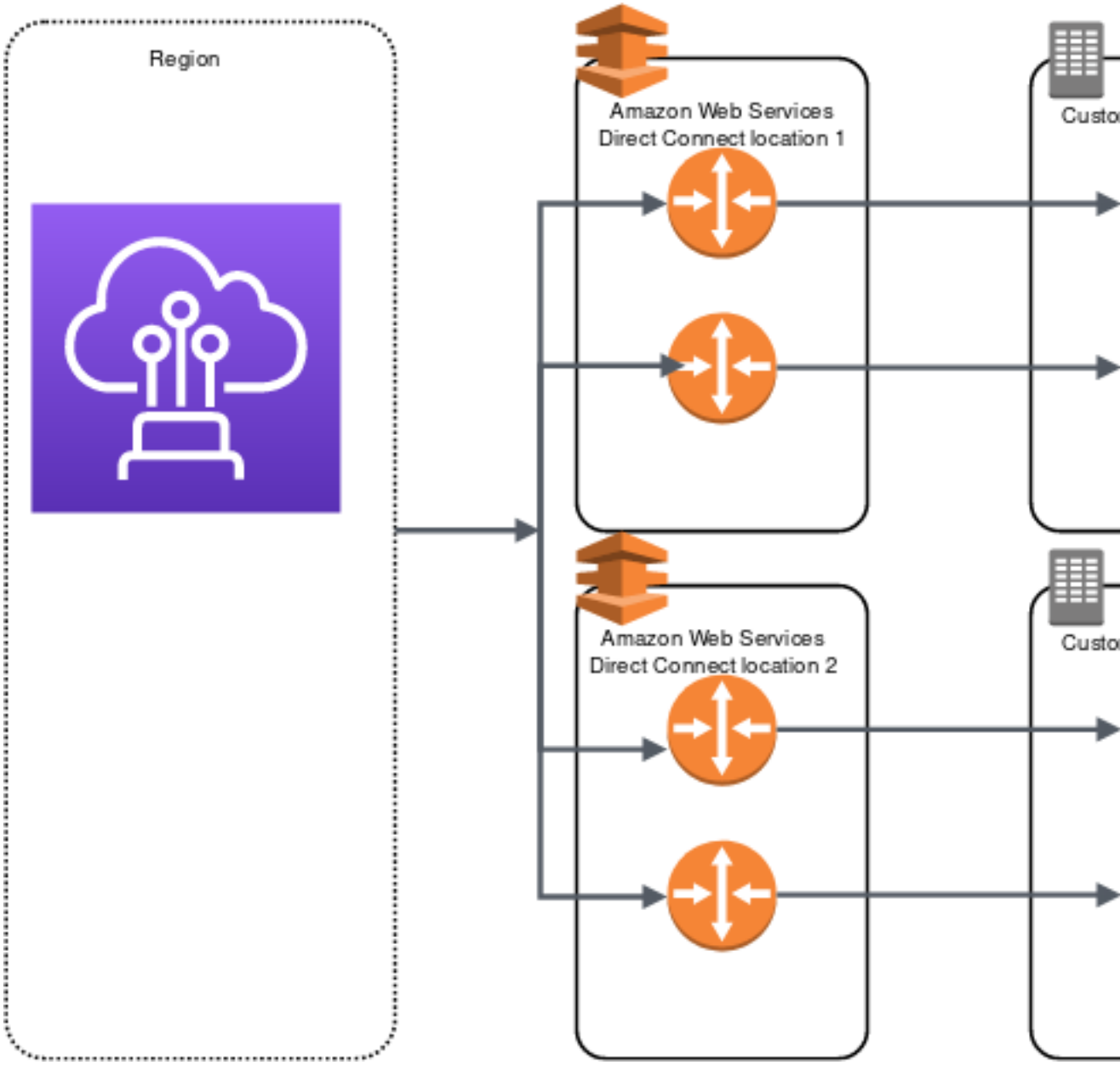
Make sure you have the following information before you begin your configuration:

- The resiliency model that you want to use.
- The speed, location, and partner for all of your connections.

  You only need the speed for one connection.

Maximum resiliency

You can achieve maximum resiliency for critical workloads by using separate connections that terminate on separate devices in more than one location (as shown in the following figure). This model provides resiliency against device, connectivity, and complete location failures. The following figure shows both connections from each customer data center going to the same AWS Direct Connect locations. You can optionally have each connection from a customer data center going to different locations.



The following procedures demonstrate how to use the AWS Direct Connect Resiliency Toolkit to configure a maximum resiliency model.

Contents

Step 1: Sign up for AWS

To use AWS Direct Connect, you need an AWS account if you don't already have one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Step 2: Configure the resiliency model

To configure a maximum resiliency model

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. On the **AWS Direct Connect** screen, under **Get started**, choose **Create a connection**.
3. Under **Connection ordering type**, choose **Connection wizard**.
4. Under **Resiliency level**, choose **Maximum Resiliency**, and then choose **Next**.
5. On the **Configure connections** pane, under **Connection settings,** do the following:

   a. For **Bandwidth**, choose the dedicated connection bandwidth.

      This bandwidth applies to all of the created connections.

   b. For **First location service provider**, select the appropriate AWS Direct Connect location for the dedicated connection.

   c. If applicable, for **First Sub location**, choose the floor closest to you or your network provider. This option is only available if the location has meet-me rooms (MMRs) on multiple floors of the building.

   d. If you selected **Other** for **First location service provider**, for **Name of other provider**, enter the name of the partner that you use.

   e. For **Second location service provider**, select the appropriate AWS Direct Connect location.

   f. If applicable, for **Second Sub location**, choose the floor closest to you or your network provider. This option is only available if the location has meet-me rooms (MMRs) on multiple floors of the building.

   g. If you selected **Other** for **Second location service provider**, for **Name of other provider**, enter the name of the partner that you use.

   h. (Optional) Add or remove a tag.

      [Add a tag] Choose **Add tag** and do the following:

      - For **Key**, enter the key name.
      - For **Value**, enter the key value.

      [Remove a tag] Next to the tag, choose **Remove tag**.

6. Choose **Next**.
7. Review your connections, and then choose **Continue**.

   If your LOAs are ready, you can choose **Download LOA**, and then click **Continue**.

   It can take up to 72 hours for AWS to review your request and provision a port for your connection. During this time, you might receive an email with a request for more information about your use

case or the specified location. The email is sent to the email address that you used when you signed up for AWS. You must respond within 7 days or the connection is deleted.

Step 3: Create your virtual interfaces

You can create a private virtual interface to connect to your VPC. Or, you can create a public virtual interface to connect to public AWS services that aren't in a VPC. When you create a private virtual interface to a VPC, you need a private virtual interface for each VPC that you're connecting to. For example, you need three private virtual interfaces to connect to three VPCs.

Before you begin, ensure that you have the following information:

Resource	Required information
Connection	The AWS Direct Connect connection or link aggregation group (LAG) for which you are creating the virtual interface.
Virtual interface name	A name for the virtual interface.
Virtual interface owner	If you're creating the virtual interface for another account, you need the AWS account ID of the other account.
(Private virtual interface only) **Connection**	For connecting to a VPC in the same AWS Region, you need the virtual private gateway for your VPC. The ASN for the Amazon side of the BGP session is inherited from the virtual private gateway. When you create a virtual private gateway, you can specify your own private ASN. Otherwise, Amazon provides a default ASN. For more information, see Create a Virtual Private Gateway in the *Amazon VPC User Guide*. For connecting to a VPC through a Direct Connect gateway, you need the Direct Connect gateway. For more information, see Direct Connect Gateways.
VLAN	A unique virtual local area network (VLAN) tag that's not already in use on your connection. The value must be between 1 and 4094 and must comply with the Ethernet 802.1Q standard. This tag is required for any traffic traversing the AWS Direct Connect connection. If you have a hosted connection, your AWS Direct Connect Partner provides this value. You can't modify the value after you have created the virtual interface.

Peer IP addresses	A virtual interface can support a BGP peering session for IPv4, IPv6, or one of each (dual-stack). You cannot create multiple BGP sessions for the same IP addressing family on the same virtual interface. The IP address ranges are assigned to each end of the virtual interface for the BGP peering session.
	
	• IPv4:
	  • (Public virtual interface only) You must specify unique public IPv4 addresses that you own. The value can be one of the following:
	  • A customer-owned ASN
	  • An ASN owned by your AWS Direct Connect Partner or ISP
	  • An AWS provided /31 CIDR. Contact contact AWS Support to request a public IPv4 CIDR (and provide a use case in your request)
	  • (Private virtual interface only) Amazon can generate private IPv4 addresses for you. If you specify your own, ensure that you specify private CIDRs for your router interface and the AWS Direct Connect interface only (for example, do not specify other IP addresses from your local network).

Resource	Required information
	• IPv6: Amazon automatically allocates you a /125 IPv6 CIDR. You cannot specify your own peer IPv6 addresses.
Address family	Whether the BGP peering session will be over IPv4 or IPv6.
BGP information	• A public or private Border Gateway Protocol (BGP) Autonomous System Number (ASN) for your side of the BGP session. If you are using a public ASN, you must own it. If you are using a private ASN, it must be in the 1 to 2147483647 range. Autonomous System (AS) prepending does not work if you use a private ASN for a public virtual interface. • AWS enables MD5 by default. You cannot modify this option. • An MD5 BGP authentication key. You can provide your own, or you can let Amazon generate one for you.

(Public virtual interface only) **Prefixes you want to advertise**	Public IPv4 routes or IPv6 routes to advertise over BGP. You must advertise at least one prefix using BGP, up to a maximum of 1,000 prefixes.
	• IPv4: The IPv4 CIDR can overlap with another public IPv4 CIDR announced using AWS Direct Connect when either of the following is true:
	  • The CIDRs are from different AWS Regions. Make sure that you apply BGP community tags on the public prefixes.
	  • You use AS_PATH when you have a public ASN in an active/passive configuration.
	  For more information, see Routing policies and BGP communities.
	• IPv6: Specify a prefix length of /64 or shorter.

(Private virtual interface only) **Jumbo frames**	The maximum transmission unit (MTU) of packets over AWS Direct Connect. The default is 1500. Setting the MTU of a virtual interface to 9001 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your virtual private gateway, then traffic routed through the static routes is sent using 1500 MTU. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo frame capable** on the virtual interface **General configuration** page.
(Transit virtual interface only) **Jumbo frames**	The maximum transmission unit (MTU) of packets over AWS Direct Connect. The default is 1500. Setting the MTU of a virtual interface to 8500 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your transit gateway, then traffic routed through the static routes is sent using 1500 MTU. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo frame capable** on the virtual interface **General configuration** page.

If your public prefixes or ASNs belong to an ISP or network carrier, we request additional information from you. This can be a document using an official company letterhead, or an email from the company's domain name verifying that the network prefix/ASN can be used by you.

When you create a public virtual interface, it can take up to 72 hours for AWS to review and approve your request.

To provision a public virtual interface to non-VPC services

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Public**.
5. Under **Public virtual interface settings**, do the following:

   a. For **Virtual interface name**, enter a name for the virtual interface.
   b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
   c. For **VLAN**, enter the ID number for your virtual local area network (VLAN).
   d. For **BGP ASN**, enter the Border Gateway Protocol (BGP) Autonomous System Number (ASN) of your gateway.

      The valid values are 1-2147483647.
6. Under **Additional settings**, do the following:

   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

      [IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:
      - To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
      - For **Amazon router peer IP**, enter the IPv4 CIDR address to use to send traffic to AWS.

      [IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.
   b. To provide your own BGP key, enter your BGP MD5 key.

      If you do not enter a value, we generate a BGP key.
   c. To advertise prefixes to Amazon, for **Prefixes you want to advertise**, enter the IPv4 CIDR destination addresses (separated by commas) to which traffic should be routed over the virtual interface.
   d. (Optional) Add or remove a tag.

      [Add a tag] Choose **Add tag** and do the following:
      - For **Key**, enter the key name.
      - For **Value**, enter the key value.

      [Remove a tag] Next to the tag, choose **Remove tag**.
7. Choose **Create virtual interface**.

To provision a private virtual interface to a VPC

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Private**.
5. Under **Private virtual interface settings**, do the following:

   a. For **Virtual interface name**, enter a name for the virtual interface.

b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
c. For **Gateway type**, choose **Virtual private gateway**, or **Direct Connect gateway**.
d. For **Virtual interface owner**, choose **Another AWS account**, and then enter the AWS account.
e. For **Virtual private gateway**, choose the virtual private gateway to use for this interface.
f. For **VLAN**, enter the ID number for your virtual local area network (VLAN).
g. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

   The valid values are 1-2147483647.

6. Under **Additional Settings**, do the following:

   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

      [IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:
      - To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
      - For **Amazon router peer ip**, enter the IPv4 CIDR address to use to send traffic to AWS.

      [IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.

   b. To change the maximum transmission unit (MTU) from 1500 (default) to 9001 (jumbo frames), select **Jumbo MTU (MTU size 9001)**.
   c. (Optional) Add or remove a tag.

      [Add a tag] Choose **Add tag** and do the following:
      - For **Key**, enter the key name.
      - For **Value**, enter the key value.

      [Remove a tag] Next to the tag, choose **Remove tag**.

7. Choose **Create virtual interface**.

Step 4: Verify your virtual interface resiliency configuration

After you have established virtual interfaces to the AWS Cloud or to Amazon VPC, perform a virtual interface failover test to verify that your configuration meets your resiliency requirements. For more information, see the section called “AWS Direct Connect Failover Test” (p. 39).

Step 5: Verify your virtual interfaces connectivity

After you have established virtual interfaces to the AWS Cloud or to Amazon VPC, you can verify your AWS Direct Connect connection using the following procedures.

To verify your virtual interface connection to the AWS Cloud

- Run `traceroute` and verify that the AWS Direct Connect identifier is in the network trace.

To verify your virtual interface connection to Amazon VPC

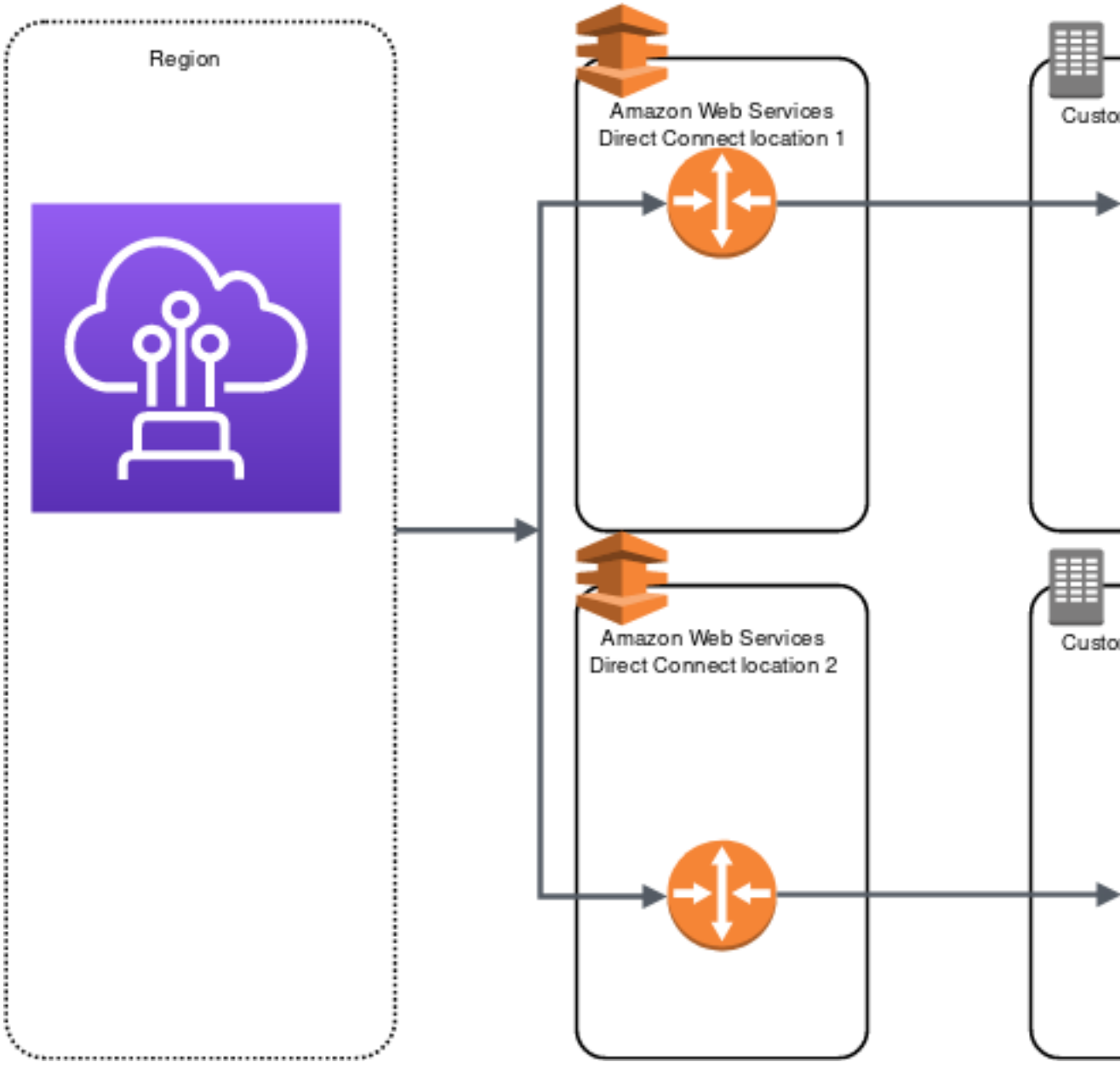
1. Using a pingable AMI, such as an Amazon Linux AMI, launch an EC2 instance into the VPC that is attached to your virtual private gateway. The Amazon Linux AMIs are available in the **Quick Start** tab when you use the instance launch wizard in the Amazon EC2 console. For more information,

see Launch an Instance in the *Amazon EC2 User Guide for Linux Instances.* Ensure that the security group that's associated with the instance includes a rule permitting inbound ICMP traffic (for the ping request).

2. After the instance is running, get its private IPv4 address (for example, 10.0.0.4). The Amazon EC2 console displays the address as part of the instance details.
3. Ping the private IPv4 address and get a response.

High resiliency

You can achieve high resiliency for critical workloads by using two single connections to multiple locations (as shown in the following figure). This model provides resiliency against connectivity failures caused by a fiber cut or a device failure. It also helps prevent a complete location failure.



The following procedures demonstrate how to use the AWS Direct Connect Resiliency Toolkit to configure a high resiliency model.

Contents

Step 1: Sign up for AWS

To use AWS Direct Connect, you need an AWS account if you don't already have one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Step 2: Configure the resiliency model

To configure a high resiliency model

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. On the **AWS Direct Connect** screen, under **Get started**, choose **Create a connection**.
3. Under **Connection ordering type**, choose **Connection wizard**.
4. Under **Resiliency level**, choose **High Resiliency**, and then choose **Next**.
5. On the **Configure connections** pane, under **Connection settings,** do the following:

   a. For **bandwidth**, choose the connection bandwidth.

      This bandwidth applies to all of the created connections.

   b. For **First location service provider**, select the appropriate AWS Direct Connect location.

   c. If applicable, for **First Sub location**, choose the floor closest to you or your network provider. This option is only available if the location has meet-me rooms (MMRs) on multiple floors of the building.

   d. If you selected **Other** for **First location service provider**, for **Name of other provider**, enter the name of the partner that you use.

   e. For **Second location service provider**, select the appropriate AWS Direct Connect location.

   f. If applicable, for **Second Sub location**, choose the floor closest to you or your network provider. This option is only available if the location has meet-me rooms (MMRs) on multiple floors of the building.

   g. If you selected **Other** for **Second location service provider**, for **Name of other provider**, enter the name of the partner that you use.

   h. (Optional) Add or remove a tag.

      [Add a tag] Choose **Add tag** and do the following:

      - For **Key**, enter the key name.

- For **Value**, enter the key value.

[Remove a tag] Next to the tag, choose **Remove tag**.

6. Choose **Next**.
7. Review your connections, and then choose **Continue**.

   If your LOAs are ready, you can choose **Download LOA**, and then click **Continue**.

   It can take up to 72 hours for AWS to review your request and provision a port for your connection. During this time, you might receive an email with a request for more information about your use case or the specified location. The email is sent to the email address that you used when you signed up for AWS. You must respond within 7 days or the connection is deleted.

Step 3: Create your virtual interfaces

You can create a private virtual interface to connect to your VPC. Or, you can create a public virtual interface to connect to public AWS services that aren't in a VPC. When you create a private virtual interface to a VPC, you need a private virtual interface for each VPC that you're connecting to. For example, you need three private virtual interfaces to connect to three VPCs.

Before you begin, ensure that you have the following information:

Resource	Required information
Connection	The AWS Direct Connect connection or link aggregation group (LAG) for which you are creating the virtual interface.
Virtual interface name	A name for the virtual interface.
Virtual interface owner	If you're creating the virtual interface for another account, you need the AWS account ID of the other account.
(Private virtual interface only) **Connection**	For connecting to a VPC in the same AWS Region, you need the virtual private gateway for your VPC. The ASN for the Amazon side of the BGP session is inherited from the virtual private gateway. When you create a virtual private gateway, you can specify your own private ASN. Otherwise, Amazon provides a default ASN. For more information, see Create a Virtual Private Gateway in the *Amazon VPC User Guide*. For connecting to a VPC through a Direct Connect gateway, you need the Direct Connect gateway. For more information, see Direct Connect Gateways.
VLAN	A unique virtual local area network (VLAN) tag that's not already in use on your connection. The value must be between 1 and 4094 and must comply with the Ethernet 802.1Q standard. This tag is required for any traffic traversing the AWS Direct Connect connection. If you have a hosted connection, your AWS Direct Connect Partner provides this value. You can't modify the value after you have created the virtual interface.
Peer IP addresses	A virtual interface can support a BGP peering session for IPv4, IPv6, or one of each (dual-stack). You cannot create multiple BGP sessions for the same IP addressing family on the same virtual interface. The IP address ranges are assigned to each end of the virtual interface for the BGP peering session. • IPv4:

Resource	Required information

	• (Public virtual interface only) You must specify unique public IPv4 addresses that you own. The value can be one of the following:
	  • A customer-owned ASN
	  • An ASN owned by your AWS Direct Connect Partner or ISP
	  • An AWS provided /31 CIDR. Contact contact AWS Support to request a public IPv4 CIDR (and provide a use case in your request)
	• (Private virtual interface only) Amazon can generate private IPv4 addresses for you. If you specify your own, ensure that you specify private CIDRs for your router interface and the AWS Direct Connect interface only (for example, do not specify other IP addresses from your local network).
	• IPv6: Amazon automatically allocates you a /125 IPv6 CIDR. You cannot specify your own peer IPv6 addresses.

Address family	Whether the BGP peering session will be over IPv4 or IPv6.
BGP information	• A public or private Border Gateway Protocol (BGP) Autonomous System Number (ASN) for your side of the BGP session. If you are using a public ASN, you must own it. If you are using a private ASN, it must be in the 1 to 2147483647 range. Autonomous System (AS) prepending does not work if you use a private ASN for a public virtual interface. • AWS enables MD5 by default. You cannot modify this option. • An MD5 BGP authentication key. You can provide your own, or you can let Amazon generate one for you.

(Public virtual interface only) **Prefixes you want to advertise**	Public IPv4 routes or IPv6 routes to advertise over BGP. You must advertise at least one prefix using BGP, up to a maximum of 1,000 prefixes.
	• IPv4: The IPv4 CIDR can overlap with another public IPv4 CIDR announced using AWS Direct Connect when either of the following is true:
	  • The CIDRs are from different AWS Regions. Make sure that you apply BGP community tags on the public prefixes.
	  • You use AS_PATH when you have a public ASN in an active/passive configuration.
	  For more information, see Routing policies and BGP communities.
	• IPv6: Specify a prefix length of /64 or shorter.

(Private virtual interface only) **Jumbo frames**	The maximum transmission unit (MTU) of packets over AWS Direct Connect. The default is 1500. Setting the MTU of a virtual interface to 9001 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your virtual private gateway, then traffic routed through the static routes is sent using 1500 MTU. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo frame capable** on the virtual interface **General configuration** page.

Resource	Required information
(Transit virtual interface only) **Jumbo frames**	The maximum transmission unit (MTU) of packets over AWS Direct Connect. The default is 1500. Setting the MTU of a virtual interface to 8500 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your transit gateway, then traffic routed through the static routes is sent using 1500 MTU. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo frame capable** on the virtual interface **General configuration** page.

If your public prefixes or ASNs belong to an ISP or network carrier, AWS requests additional information from you. This can be a document using an official company letterhead, or an email from the company's domain name verifying that the network prefix/ASN can be used by you.

When you create a public virtual interface, it can take up to 72 hours for AWS to review and approve your request.

To provision a public virtual interface to non-VPC services

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Public**.
5. Under **Public virtual interface settings**, do the following:

   a. For **Virtual interface name**, enter a name for the virtual interface.
   b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
   c. For **VLAN**, enter the ID number for your virtual local area network (VLAN).
   d. For **BGP ASN**, enter the Border Gateway Protocol (BGP) Autonomous System Number (ASN) of your gateway.

      The valid values are 1-2147483647.
6. Under **Additional settings**, do the following:

   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

      [IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:
      - To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
      - For **Amazon router peer IP**, enter the IPv4 CIDR address to use to send traffic to AWS.

      [IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.
   b. To provide your own BGP key, enter your BGP MD5 key.

      If you do not enter a value, we generate a BGP key.
   c. To advertise prefixes to Amazon, for **Prefixes you want to advertise**, enter the IPv4 CIDR destination addresses (separated by commas) to which traffic should be routed over the virtual interface.
   d. (Optional) Add or remove a tag.

[Add a tag] Choose **Add tag** and do the following:

- For **Key**, enter the key name.
- For **Value**, enter the key value.

[Remove a tag] Next to the tag, choose **Remove tag**.

7. Choose **Create virtual interface**.

To provision a private virtual interface to a VPC

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Private**.
5. Under **Private virtual interface settings**, do the following:

   a. For **Virtual interface name**, enter a name for the virtual interface.

   b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.

   c. For **Gateway type**, choose **Virtual private gateway**, or **Direct Connect gateway**.

   d. For **Virtual interface owner**, choose **Another AWS account**, and then enter the AWS account.

   e. For **Virtual private gateway**, choose the virtual private gateway to use for this interface.

   f. For **VLAN**, enter the ID number for your virtual local area network (VLAN).

   g. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

   The valid values are 1-2147483647.

6. Under **Additional Settings**, do the following:

   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

   [IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:

   - To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
   - For **Amazon router peer ip**, enter the IPv4 CIDR address to use to send traffic to AWS.

   [IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.

   b. To change the maximum transmission unit (MTU) from 1500 (default) to 9001 (jumbo frames), select **Jumbo MTU (MTU size 9001)**.

   c. (Optional) Add or remove a tag.

   [Add a tag] Choose **Add tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Next to the tag, choose **Remove tag**.

7. Choose **Create virtual interface**.

Step 4: Verify your virtual interface resiliency configuration

After you have established virtual interfaces to the AWS Cloud or to Amazon VPC, perform a virtual interface failover test to verify that your configuration meets your resiliency requirements. For more information, see the section called "AWS Direct Connect Failover Test" (p. 39).

Step 5: Verify your virtual interfaces connectivity

After you have established virtual interfaces to the AWS Cloud or to Amazon VPC, you can verify your AWS Direct Connect connection using the following procedures.

To verify your virtual interface connection to the AWS Cloud

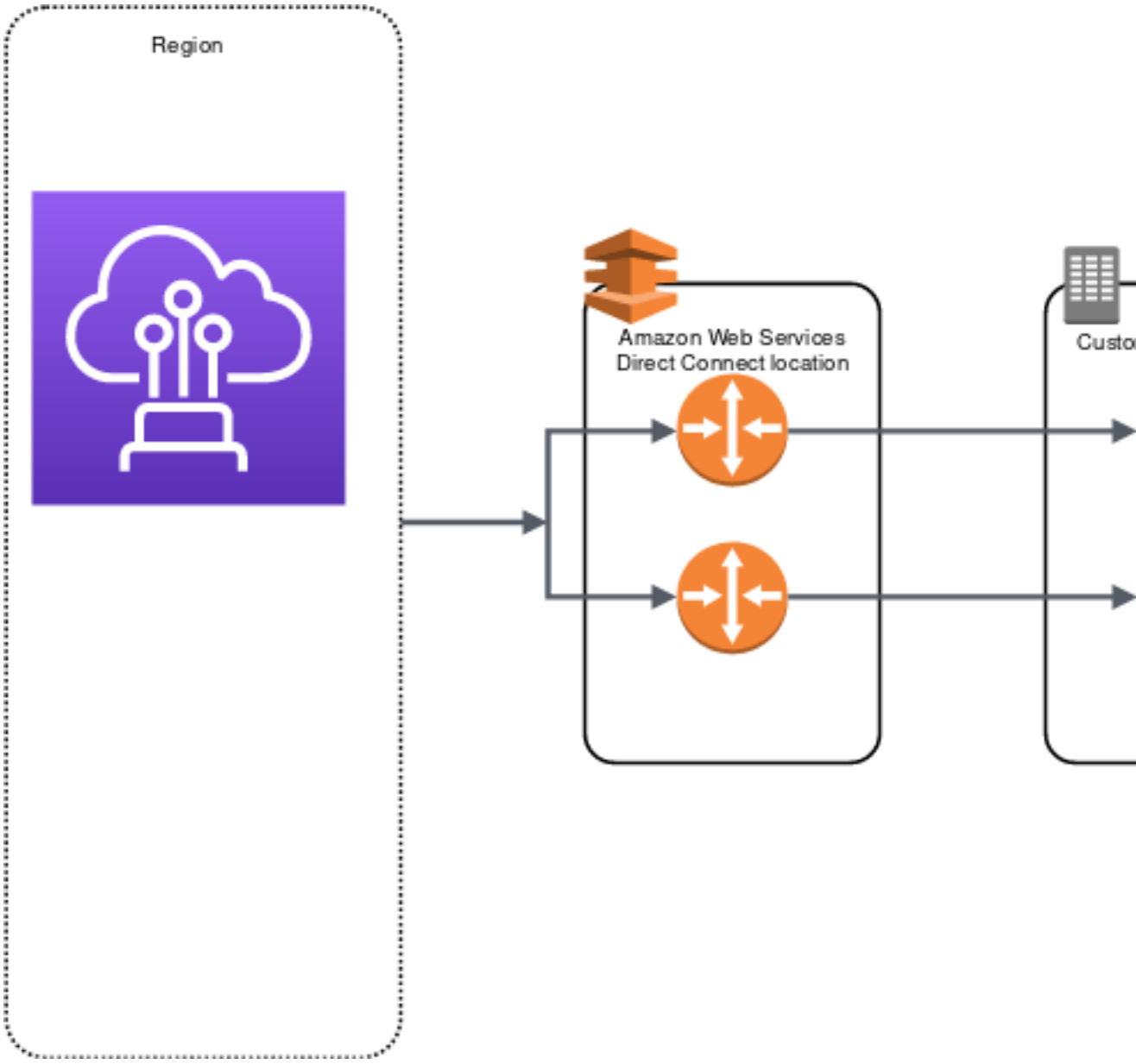
- Run `traceroute` and verify that the AWS Direct Connect identifier is in the network trace.

To verify your virtual interface connection to Amazon VPC

1. Using a pingable AMI, such as an Amazon Linux AMI, launch an EC2 instance into the VPC that is attached to your virtual private gateway. The Amazon Linux AMIs are available in the **Quick Start** tab when you use the instance launch wizard in the Amazon EC2 console. For more information, see Launch an Instance in the *Amazon EC2 User Guide for Linux Instances.* Ensure that the security group that's associated with the instance includes a rule permitting inbound ICMP traffic (for the ping request).
2. After the instance is running, get its private IPv4 address (for example, 10.0.0.4). The Amazon EC2 console displays the address as part of the instance details.
3. Ping the private IPv4 address and get a response.

Development and test

You can achieve development and test resiliency for non-critical workloads by using separate connections that terminate on separate devices in one location (as shown in the following figure). This model provides resiliency against device failure, but does not provide resiliency against location failure.



The following procedures demonstrate how to use the AWS Direct Connect Resiliency Toolkit to configure a development and test resiliency model.

Contents

Step 1: Sign up for AWS

To use AWS Direct Connect, you need an AWS account if you don't already have one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Step 2: Configure the resiliency model

To configure the resiliency model

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. On the **AWS Direct Connect** screen, under **Get started**, choose **Create a connection**.
3. Under **Connection ordering type**, choose **Connection wizard**.
4. Under **Resiliency level**, choose **Development and test**, and then choose **Next**.
5. On the **Configure connections** pane, under **Connection settings,** do the following:

   a. For **bandwidth**, choose the connection bandwidth.

      This bandwidth applies to all of the created connections.

   b. For **First location service provider**, select the appropriate AWS Direct Connect location.

   c. If applicable, for **First Sub location**, choose the floor closest to you or your network provider. This option is only available if the location has meet-me rooms (MMRs) on multiple floors of the building.

   d. If you selected **Other** for **First location service provider**, for **Name of other provider**, enter the name of the partner that you use.

   e. (Optional) Add or remove a tag.

      [Add a tag] Choose **Add tag** and do the following:

      - For **Key**, enter the key name.
      - For **Value**, enter the key value.

      [Remove a tag] Next to the tag, choose **Remove tag**.
6. Choose **Next**.
7. Review your connections, and then choose **Continue**.

   If your LOAs are ready, you can choose **Download LOA**, and then click **Continue**.

   It can take up to 72 hours for AWS to review your request and provision a port for your connection. During this time, you might receive an email with a request for more information about your use case or the specified location. The email is sent to the email address that you used when you signed up for AWS. You must respond within 7 days or the connection is deleted.

Step 3: Create a virtual interface

To begin using your AWS Direct Connect connection, you must create a virtual interface. You can create a private virtual interface to connect to your VPC. Or, you can create a public virtual interface to connect to public AWS services that aren't in a VPC. When you create a private virtual interface to a VPC, you need a private virtual interface for each VPC that you're connecting to. For example, you need three private virtual interfaces to connect to three VPCs.

Before you begin, ensure that you have the following information:

Resource	Required information
Connection	The AWS Direct Connect connection or link aggregation group (LAG) for which you are creating the virtual interface.
Virtual interface name	A name for the virtual interface.
Virtual interface owner	If you're creating the virtual interface for another account, you need the AWS account ID of the other account.
(Private virtual interface only) **Connection**	For connecting to a VPC in the same AWS Region, you need the virtual private gateway for your VPC. The ASN for the Amazon side of the BGP session is inherited from the virtual private gateway. When you create a virtual private gateway, you can specify your own private ASN. Otherwise, Amazon provides a default ASN. For more information, see Create a Virtual Private Gateway in the *Amazon VPC User Guide*. For connecting to a VPC through a Direct Connect gateway, you need the Direct Connect gateway. For more information, see Direct Connect Gateways.
VLAN	A unique virtual local area network (VLAN) tag that's not already in use on your connection. The value must be between 1 and 4094 and must comply with the Ethernet 802.1Q standard. This tag is required for any traffic traversing the AWS Direct Connect connection. If you have a hosted connection, your AWS Direct Connect Partner provides this value. You can't modify the value after you have created the virtual interface.

Peer IP addresses	A virtual interface can support a BGP peering session for IPv4, IPv6, or one of each (dual-stack). You cannot create multiple BGP sessions for the same IP addressing family on the same virtual interface. The IP address ranges are assigned to each end of the virtual interface for the BGP peering session.
	
	• IPv4:
	  • (Public virtual interface only) You must specify unique public IPv4 addresses that you own. The value can be one of the following:
	  • A customer-owned ASN
	  • An ASN owned by your AWS Direct Connect Partner or ISP
	  • An AWS provided /31 CIDR. Contact contact AWS Support to request a public IPv4 CIDR (and provide a use case in your request)
	  • (Private virtual interface only) Amazon can generate private IPv4 addresses for you. If you specify your own, ensure that you specify private CIDRs for your router interface and the AWS Direct Connect interface only (for example, do not specify other IP addresses from your local network).
	• IPv6: Amazon automatically allocates you a /125 IPv6 CIDR. You cannot specify your own peer IPv6 addresses.

Address family	Whether the BGP peering session will be over IPv4 or IPv6.
BGP information	• A public or private Border Gateway Protocol (BGP) Autonomous System Number (ASN) for your side of the BGP session. If you are using a public ASN, you must own it. If you are using a private ASN, it must be in the 1 to 2147483647 range. Autonomous System (AS) prepending does not work if you use a private ASN for a public virtual interface. • AWS enables MD5 by default. You cannot modify this option.

Resource	Required information
	• An MD5 BGP authentication key. You can provide your own, or you can let Amazon generate one for you.

(Public virtual interface only) **Prefixes you want to advertise**	Public IPv4 routes or IPv6 routes to advertise over BGP. You must advertise at least one prefix using BGP, up to a maximum of 1,000 prefixes.
	• IPv4: The IPv4 CIDR can overlap with another public IPv4 CIDR announced using AWS Direct Connect when either of the following is true:
	  • The CIDRs are from different AWS Regions. Make sure that you apply BGP community tags on the public prefixes.
	  • You use AS_PATH when you have a public ASN in an active/passive configuration.
	  For more information, see Routing policies and BGP communities.
	• IPv6: Specify a prefix length of /64 or shorter.

(Private virtual interface only) **Jumbo frames**	The maximum transmission unit (MTU) of packets over AWS Direct Connect. The default is 1500. Setting the MTU of a virtual interface to 9001 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your virtual private gateway, then traffic routed through the static routes is sent using 1500 MTU. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo frame capable** on the virtual interface **General configuration** page.
(Transit virtual interface only) **Jumbo frames**	The maximum transmission unit (MTU) of packets over AWS Direct Connect. The default is 1500. Setting the MTU of a virtual interface to 8500 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your transit gateway, then traffic routed through the static routes is sent using 1500 MTU. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo frame capable** on the virtual interface **General configuration** page.

If your public prefixes or ASNs belong to an ISP or network carrier, we request additional information from you. This can be a document using an official company letterhead, or an email from the company's domain name verifying that the network prefix/ASN can be used by you.

When you create a public virtual interface, it can take up to 72 hours for AWS to review and approve your request.

To provision a public virtual interface to non-VPC services

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Public**.
5. Under **Public virtual interface settings**, do the following:

a. For **Virtual interface name**, enter a name for the virtual interface.
b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
c. For **VLAN**, enter the ID number for your virtual local area network (VLAN).
d. For **BGP ASN**, enter the Border Gateway Protocol (BGP) Autonomous System Number (ASN) of your gateway.

   The valid values are 1-2147483647.

6. Under **Additional settings**, do the following:

   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

      [IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:

      - To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
      - For **Amazon router peer IP**, enter the IPv4 CIDR address to use to send traffic to AWS.

      [IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.

   b. To provide your own BGP key, enter your BGP MD5 key.

      If you do not enter a value, we generate a BGP key.

   c. To advertise prefixes to Amazon, for **Prefixes you want to advertise**, enter the IPv4 CIDR destination addresses (separated by commas) to which traffic should be routed over the virtual interface.

   d. (Optional) Add or remove a tag.

      [Add a tag] Choose **Add tag** and do the following:

      - For **Key**, enter the key name.
      - For **Value**, enter the key value.

      [Remove a tag] Next to the tag, choose **Remove tag**.

7. Choose **Create virtual interface**.

To provision a private virtual interface to a VPC

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Private**.
5. Under **Private virtual interface settings**, do the following:

   a. For **Virtual interface name**, enter a name for the virtual interface.
   b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
   c. For **Gateway type**, choose **Virtual private gateway**, or **Direct Connect gateway**.
   d. For **Virtual interface owner**, choose **Another AWS account**, and then enter the AWS account.
   e. For **Virtual private gateway**, choose the virtual private gateway to use for this interface.
   f. For **VLAN**, enter the ID number for your virtual local area network (VLAN).
   g. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

      The valid values are 1-2147483647.

6. Under **Additional Settings**, do the following:

   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

      [IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:

      - To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
      - For **Amazon router peer ip**, enter the IPv4 CIDR address to use to send traffic to AWS.

      [IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.

   b. To change the maximum transmission unit (MTU) from 1500 (default) to 9001 (jumbo frames), select **Jumbo MTU (MTU size 9001)**.

   c. (Optional) Add or remove a tag.

      [Add a tag] Choose **Add tag** and do the following:

      - For **Key**, enter the key name.
      - For **Value**, enter the key value.

      [Remove a tag] Next to the tag, choose **Remove tag**.

7. Choose **Create virtual interface**.

Step 4: Verify your virtual interface resiliency configuration

After you have established virtual interfaces to the AWS Cloud or to Amazon VPC, perform a virtual interface failover test to verify that your configuration meets your resiliency requirements. For more information, see the section called “AWS Direct Connect Failover Test” (p. 39).

Step 5: Verify your virtual interface

After you have established virtual interfaces to the AWS Cloud or to Amazon VPC, you can verify your AWS Direct Connect connection using the following procedures.

To verify your virtual interface connection to the AWS Cloud

- Run `traceroute` and verify that the AWS Direct Connect identifier is in the network trace.

To verify your virtual interface connection to Amazon VPC

1. Using a pingable AMI, such as an Amazon Linux AMI, launch an EC2 instance into the VPC that is attached to your virtual private gateway. The Amazon Linux AMIs are available in the **Quick Start** tab when you use the instance launch wizard in the Amazon EC2 console. For more information, see Launch an Instance in the *Amazon EC2 User Guide for Linux Instances.* Ensure that the security group that's associated with the instance includes a rule permitting inbound ICMP traffic (for the ping request).
2. After the instance is running, get its private IPv4 address (for example, 10.0.0.4). The Amazon EC2 console displays the address as part of the instance details.
3. Ping the private IPv4 address and get a response.

Classic

Select Classic when you have existing connections.

The following procedures demonstrate the common scenarios to get set up with an AWS Direct Connect connection.

Contents

Prerequisites

For connections to AWS Direct Connect with port speeds of 1 Gbps or higher, ensure that your network meets the following requirements:

- Your network must use single-mode fiber with a 1000BASE-LX (1310 nm) transceiver for 1 gigabit Ethernet, a 10GBASE-LR (1310 nm) transceiver for 10 gigabit, or a 100GBASE-LR4 for 100 gigabit Ethernet.
- Auto-negotiation for the port must be disabled. Auto-negotiation is supported only if the port speed is 1 Gbps. Port speed and full-duplex mode must be configured manually.
- 802.1Q VLAN encapsulation must be supported across the entire connection, including intermediate devices.
- Your device must support Border Gateway Protocol (BGP) and BGP MD5 authentication.
- (Optional) You can configure Bidirectional Forwarding Detection (BFD) on your network. Asynchronous BFD is automatically enabled for AWS Direct Connect virtual interfaces, but does not take effect until you configure it on your router.

Step 1: Sign up for AWS

To use AWS Direct Connect, you need an account if you don't already have one.

To sign up for an account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Step 2: Request an AWS Direct Connect dedicated connection or accept a hosted connection

For dedicated connections, you can submit a connection request using the AWS Direct Connect console. For hosted connections, work with an AWS Direct Connect Partner to request a hosted connection. Ensure that you have the following information:

- The port speed that you require. You cannot change the port speed after you create the connection request.
- The AWS Direct Connect location at which the connection is to be terminated.

You cannot use the AWS Direct Connect console to request a hosted connection. Instead, contact an AWS Direct Connect Partner, who can create a hosted connection for you, which you then accept. Skip the following procedure and go to Accept your hosted connection (p. 32).

To create a new AWS Direct Connect connection

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. On the **AWS Direct Connect** screen, under **Get started**, choose **Create a connection**.
3. Choose **Classic**.
4. On the **Create Connection** pane, under **Connection settings,** do the following:
   a. For **Name**, enter a name for the connection.
   b. For **Location**, select the appropriate AWS Direct Connect location.
   c. If applicable, for **Sub Location**, choose the floor closest to you or your network provider. This option is only available if the location has meet-me rooms (MMRs) in multiple floors of the building.
   d. For **Port Speed**, choose the connection bandwidth.
   e. For **On-premises**, select **Connect through an AWS Direct Connect partner** when you use this connection to connect to your data center.
   f. For **Service provider**, select the AWS Direct Connect Partner. If you use a partner that is not in the list, select **Other**.
   g. If you selected **Other** for **Service provider**, for **Name of other provider**, enter the name of the partner that you use.
   h. (Optional) Add or remove a tag.

      [Add a tag] Choose **Add tag** and do the following:

      - For **Key**, enter the key name.
      - For **Value**, enter the key value.

      [Remove a tag] Next to the tag, choose **Remove tag**.
5. Choose **Create Connection**.

It can take up to 72 hours for AWS to review your request and provision a port for your connection. During this time, you might receive an email with a request for more information about your use case or the specified location. The email is sent to the email address that you used when you signed up for AWS. You must respond within 7 days or the connection is deleted.

For more information, see AWS Direct Connect connections (p. 45).

Accept your hosted connection

You must accept the hosted connection in the AWS Direct Connect console before you can create a virtual interface.

To accept a hosted virtual interface

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Select the virtual interface and then choose **View details**.
4. Choose **Accept**.
5. This applies to private virtual interfaces and transit virtual interfaces.

   (Transit virtual interface) In the **Accept virtual interface** dialog box, select a Direct Connect gateway, and then choose **Accept virtual interface**.

   (Private virtual interface) In the **Accept virtual interface** dialog box, select a virtual private gateway or Direct Connect gateway, and then choose **Accept virtual interface**.
6. After you accept the hosted virtual interface, the owner of the AWS Direct Connect connection can download the router configuration file. The **Download router configuration** option is not available for the account that accepts the hosted virtual interface.
7. Go to Step 4 (p. 33) to continue setting up your AWS Direct Connect connection.

(Dedicated connection) Step 3: Download the LOA-CFA

After you request a connection, we make a Letter of Authorization and Connecting Facility Assignment (LOA-CFA) available to you to download, or emails you with a request for more information. The LOA-CFA is the authorization to connect to AWS, and is required by the colocation provider or your network provider to establish the cross-network connection (cross-connect).

To download the LOA-CFA

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Connections**.
3. Select the connection and choose **View Details**.
4. Choose **Download LOA-CFA**.

   The LOA-CFA is downloaded to your computer as a PDF file.

   Note
   If the link is not enabled, the LOA-CFA is not yet available for you to download. Check your email for a request for more information. If it's still unavailable, or you haven't received an email after 72 hours, contact AWS Support.
5. After you download the LOA-CFA, do one of the following:

   - If you're working with an AWS Direct Connect Partner or network provider, send them the LOA-CFA so that they can order a cross-connect for you at the AWS Direct Connect location. If they cannot order the cross-connect for you, you can contact the colocation provider (p. 53) directly.
   - If you have equipment at the AWS Direct Connect location, contact the colocation provider to request a cross-network connection. You must be a customer of the colocation provider. You must also present them with the LOA-CFA that authorizes the connection to the AWS router, and the necessary information to connect to your network.

AWS Direct Connect locations that are listed as multiple sites (for example, Equinix DC1-DC6 & DC10-DC11) are set up as a campus. If your or your network provider's equipment is located in any of these sites, you can request a cross-connect to your assigned port even if it resides in a different campus building.

Important
A campus is treated as a single AWS Direct Connect location. To achieve high availability, configure connections to different AWS Direct Connect locations.

If you or your network provider experience issues establishing a physical connection, see Troubleshooting layer 1 (physical) issues (p. 148).

Step 4: Create a virtual interface

To begin using your AWS Direct Connect connection, you must create a virtual interface. You can create a private virtual interface to connect to your VPC. Or, you can create a public virtual interface to connect to public AWS services that aren't in a VPC. When you create a private virtual interface to a VPC, you need a private virtual interface for each VPC to which to connect. For example, you need three private virtual interfaces to connect to three VPCs.

Before you begin, ensure that you have the following information:

Resource	Required information
Connection	The AWS Direct Connect connection or link aggregation group (LAG) for which you are creating the virtual interface.
Virtual interface name	A name for the virtual interface.
Virtual interface owner	If you're creating the virtual interface for another account, you need the AWS account ID of the other account.
(Private virtual interface only) **Connection**	For connecting to a VPC in the same AWS Region, you need the virtual private gateway for your VPC. The ASN for the Amazon side of the BGP session is inherited from the virtual private gateway. When you create a virtual private gateway, you can specify your own private ASN. Otherwise, Amazon provides a default ASN. For more information, see Create a Virtual Private Gateway in the *Amazon VPC User Guide*. For connecting to a VPC through a Direct Connect gateway, you need the Direct Connect gateway. For more information, see Direct Connect Gateways.
VLAN	A unique virtual local area network (VLAN) tag that's not already in use on your connection. The value must be between 1 and 4094 and must comply with the Ethernet 802.1Q standard. This tag is required for any traffic traversing the AWS Direct Connect connection. If you have a hosted connection, your AWS Direct Connect Partner provides this value. You can't modify the value after you have created the virtual interface.
Peer IP addresses	A virtual interface can support a BGP peering session for IPv4, IPv6, or one of each (dual-stack). You cannot create multiple BGP sessions for the same IP addressing family on the same virtual interface. The IP address ranges are assigned to each end of the virtual interface for the BGP peering session. • IPv4: • (Public virtual interface only) You must specify unique public IPv4 addresses that you own. The value can be one of the following:

Resource	Required information
	• A customer-owned ASN • An ASN owned by your AWS Direct Connect Partner or ISP • An AWS provided /31 CIDR. Contact contact AWS Support to request a public IPv4 CIDR (and provide a use case in your request) • (Private virtual interface only) Amazon can generate private IPv4 addresses for you. If you specify your own, ensure that you specify private CIDRs for your router interface and the AWS Direct Connect interface only (for example, do not specify other IP addresses from your local network). • IPv6: Amazon automatically allocates you a /125 IPv6 CIDR. You cannot specify your own peer IPv6 addresses.
Address family	Whether the BGP peering session will be over IPv4 or IPv6.
BGP information	• A public or private Border Gateway Protocol (BGP) Autonomous System Number (ASN) for your side of the BGP session. If you are using a public ASN, you must own it. If you are using a private ASN, it must be in the 1 to 2147483647 range. Autonomous System (AS) prepending does not work if you use a private ASN for a public virtual interface. • AWS enables MD5 by default. You cannot modify this option. • An MD5 BGP authentication key. You can provide your own, or you can let Amazon generate one for you.

(Public virtual interface only) **Prefixes you want to advertise**	Public IPv4 routes or IPv6 routes to advertise over BGP. You must advertise at least one prefix using BGP, up to a maximum of 1,000 prefixes.
	• IPv4: The IPv4 CIDR can overlap with another public IPv4 CIDR announced using AWS Direct Connect when either of the following is true:
	  • The CIDRs are from different AWS Regions. Make sure that you apply BGP community tags on the public prefixes.
	  • You use AS_PATH when you have a public ASN in an active/passive configuration.
	  For more information, see Routing policies and BGP communities.
	• IPv6: Specify a prefix length of /64 or shorter.

(Private virtual interface only) **Jumbo frames**	The maximum transmission unit (MTU) of packets over AWS Direct Connect. The default is 1500. Setting the MTU of a virtual interface to 9001 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your virtual private gateway, then traffic routed through the static routes is sent using 1500 MTU. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo frame capable** on the virtual interface **General configuration** page.

Resource	Required information
(Transit virtual interface only) **Jumbo frames**	The maximum transmission unit (MTU) of packets over AWS Direct Connect. The default is 1500. Setting the MTU of a virtual interface to 8500 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your transit gateway, then traffic routed through the static routes is sent using 1500 MTU. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo frame capable** on the virtual interface **General configuration** page.

We request additional information from you if your public prefixes or ASNs belong to an ISP or network carrier. This can be a document using an official company letterhead or an email from the company's domain name verifying that the network prefix/ASN may be used by you.

For private virtual interface and public virtual interfaces, the maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The MTU of a virtual private interface can be either 1500 or 9001 (jumbo frames). The MTU of a transit virtual interface can be either 1500 or 8500 (jumbo frames). You can specify the MTU when you create the interface or update it after you create it. Setting the MTU of a virtual interface to 8500 (jumbo frames) or 9001 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo Frame Capable** on the **Summary** tab.

When you create a public virtual interface, it can take up to 72 hours for AWS to review and approve your request.

To provision a public virtual interface to non-VPC services

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Public**.
5. Under **Public virtual interface settings**, do the following:

   a. For **Virtual interface name**, enter a name for the virtual interface.
   b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
   c. For **VLAN**, enter the ID number for your virtual local area network (VLAN).
   d. For **BGP ASN**, enter the The Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

      The valid values are 1-2147483647.
6. Under **Additional settings**, do the following:

   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

      [IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:
      - To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
      - For **Amazon router peer IP**, enter the IPv4 CIDR address to use to send traffic to AWS.

[IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.

b. To provide your own BGP key, enter your BGP MD5 key.

If you do not enter a value, we generate a BGP key.

c. To advertise prefixes to Amazon, for **Prefixes you want to advertise**, enter the IPv4 CIDR destination addresses (separated by commas) to which traffic should be routed over the virtual interface.

d. (Optional) Add or remove a tag.

[Add a tag] Choose **Add tag** and do the following:

- For **Key**, enter the key name.
- For **Value**, enter the key value.

[Remove a tag] Next to the tag, choose **Remove tag**.

7. Choose **Create virtual interface**.

To provision a private virtual interface to a VPC

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Private**.
5. Under **Private virtual interface settings**, do the following:

a. For **Virtual interface name**, enter a name for the virtual interface.

b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.

c. For **Gateway type**, choose **Virtual private gateway**, or **Direct Connect gateway**.

d. For **Virtual interface owner**, choose **Another AWS account**, and then enter the AWS account.

e. For **Virtual private gateway**, choose the virtual private gateway to use for this interface.

f. For **VLAN**, enter the ID number for your virtual local area network (VLAN).

g. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

The valid values are 1-2147483647.

6. Under **Additional Settings**, do the following:

a. To configure an IPv4 BGP or an IPv6 peer, do the following:

[IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:

- To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
- For **Amazon router peer ip**, enter the IPv4 CIDR address to use to send traffic to AWS.

[IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.

b. To change the maximum transmission unit (MTU) from 1500 (default) to 9001 (jumbo frames), select **Jumbo MTU (MTU size 9001)**.

c. (Optional) Add or remove a tag.

[Add a tag] Choose **Add tag** and do the following:

- For **Key**, enter the key name.
- For **Value**, enter the key value.

[Remove a tag] Next to the tag, choose **Remove tag**.

7. Choose **Create virtual interface**.
8. You need to use your BGP device to advertise the network that you use for the public VIF connection.

Step 5: Download the router configuration

After you have created a virtual interface for your AWS Direct Connect connection, you can download the router configuration file. The file contains the necessary commands to configure your router for use with your private or public virtual interface.

To download a router configuration

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Select the connection and choose **View Details**.
4. Choose **Download router configuration**.
5. For **Download router configuration**, do the following:
   a. For **Vendor**, select the manufacturer of your router.
   b. For **Platform**, select the model of your router.
   c. For **Software**, select the software version for your router.
6. Choose **Download**, and then use the appropriate configuration for your router to ensure that you can connect to AWS Direct Connect.

For example configuration files, see Example Router Configuration Files.

After you configure your router, the status of the virtual interface goes to `UP`. If the virtual interface remains down and you cannot ping the AWS Direct Connect device's peer IP address, see Troubleshooting layer 2 (data link) issues (p. 149). If you can ping the peer IP address, see Troubleshooting layer 3/4 (Network/Transport) issues (p. 151). If the BGP peering session is established but you cannot route traffic, see Troubleshooting routing issues (p. 152).

Step 6: Verify your virtual interface

After you have established virtual interfaces to the AWS Cloud or to Amazon VPC, you can verify your AWS Direct Connect connection using the following procedures.

To verify your virtual interface connection to the AWS Cloud

- Run `traceroute` and verify that the AWS Direct Connect identifier is in the network trace.

To verify your virtual interface connection to Amazon VPC

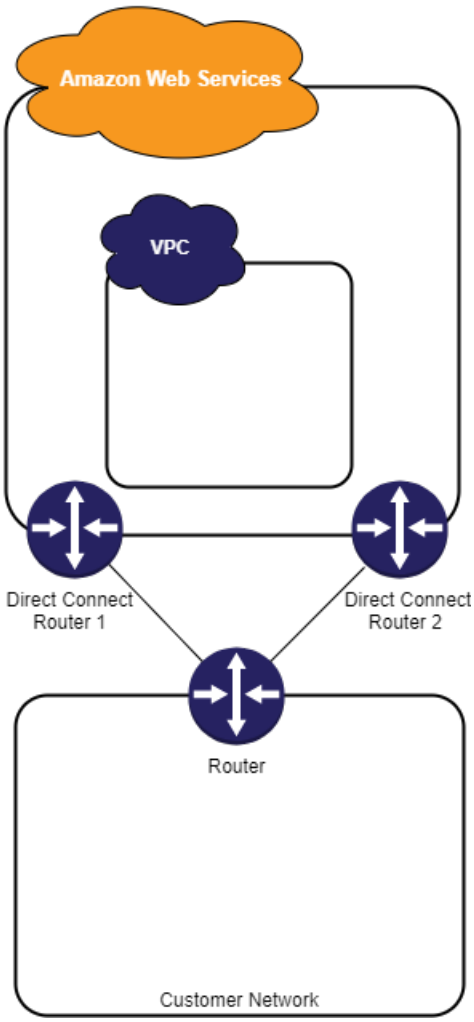
1. Using a pingable AMI, such as an Amazon Linux AMI, launch an EC2 instance into the VPC that is attached to your virtual private gateway. The Amazon Linux AMIs are available in the **Quick Start** tab when you use the instance launch wizard in the Amazon EC2 console. For more information, see Launch an Instance in the *Amazon EC2 User Guide for Linux Instances.* Ensure that the security

group that's associated with the instance includes a rule permitting inbound ICMP traffic (for the ping request).

2. After the instance is running, get its private IPv4 address (for example, 10.0.0.4). The Amazon EC2 console displays the address as part of the instance details.
3. Ping the private IPv4 address and get a response.

(Recommended) Step 7: Configure redundant connections

To provide for failover, we recommend that you request and configure two dedicated connections to AWS, as shown in the following figure. These connections can terminate on one or two routers in your network.



There are different configuration choices available when you provision two dedicated connections:

- Active/Active (BGP multipath). This is the default configuration, where both connections are active. AWS Direct Connect supports multipathing to multiple virtual interfaces within the same location, and traffic is load-shared between interfaces based on flow. If one connection becomes unavailable, all traffic is routed through the other connection.

- Active/Passive (failover). One connection is handling traffic, and the other is on standby. If the active connection becomes unavailable, all traffic is routed through the passive connection. You need to prepend the AS path to the routes on one of your links for that to be the passive link.

How you configure the connections doesn't affect redundancy, but it does affect the policies that determine how your data is routed over both connections. We recommend that you configure both connections as active.

If you use a VPN connection for redundancy, ensure that you implement a health check and failover mechanism. If you use either of the following configurations, then you need to check your route table routing to route to the new network interface.

- You use your own instances for routing, for example the instance is the firewall.
- You use your own instance that terminates a VPN connection.

To achieve high availability, we strongly recommend that you configure connections to different AWS Direct Connect locations.

For more information about AWS Direct Connect resiliency, see AWS Direct Connect Resiliency Recommendations.

AWS Direct Connect Failover Test

The AWS Direct Connect Resiliency Toolkit resiliency models are designed to ensure that you have the appropriate number of virtual interface connections in multiple locations. After you complete the wizard, use the AWS Direct Connect Resiliency Toolkit failover test to bring down the BGP peering session in order to verify that traffic routes to one of your redundant virtual interfaces, and meets your resiliency requirements.

Use the test to make sure that traffic routes over redundant virtual interfaces when a virtual interface is out of service. You start the test by selecting a virtual interface, BGP peering session, and how long to run the test. AWS places the selected virtual interface BGP peering session in the down state. When the interface is in this state, traffic should go over a redundant virtual interface. If your configuration does not contain the appropriate redundant connections, the BGP peering session fails, and traffic does not get routed. When the test completes, or you manually stop the test, AWS restores the BGP session. After the test is complete, you can use the AWS Direct Connect Resiliency Toolkit to adjust your configuration.

Test History

AWS deletes the test history after 365 days. The test history includes the status for tests that were run on all BGP peers. The history includes which BGP peering sessions were tested, the start and end times, and the test status, which can be any of the following values:

- **In progress** - The test is currently running.
- **Completed** - The test ran for the time that you specified.
- **Cancelled** - The test was cancelled before the specified time.
- **Failed** - The test did not run for the time that you specified. This can happen when there is an issue with the router.

For more information, see the section called “Viewing the virtual interface failover test history” (p. 40).

Validation Permissions

The only account that has permission to run the failover test is the account that owns the virtual interface. The account owner receives an indication through AWS CloudTrail that a test ran on a virtual interface.

Starting the virtual interface failover test

You can start the virtual interface failover test using the AWS Direct Connect console, or the AWS CLI.

To start the virtual interface failover test from the AWS Direct Connect console

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. Choose **Virtual interfaces**.
3. Select the virtual interfaces and then choose **Actions**, **Bring down BGP**.

   You can run the test on a public, private, or transit virtual interface.
4. In the **Start failure test** dialog box, do the following:

   a. For **Peerings to bring down to test**, choose which peering sessions to test, for example IPv4.
   b. For **Test maximum time**, enter the number of minutes that the test will last.

      The maximum value is 180 minutes (3 hours).

      The default value is 180 minutes (3 hours).
   c. For **To confirm test**, enter **Confirm**.
   d. Choose **Confirm**.

   The BGP peering session is placed in the DOWN state. You can send traffic to verify that there are no outages. If needed, you can stop the test immediately.

To start the virtual interface failover test using the AWS CLI

Use StartBgpFailoverTest.

Viewing the virtual interface failover test history

You can view the virtual interface failover test history using the AWS Direct Connect console, or the AWS CLI.

To view the virtual interface failover test history from the AWS Direct Connect console

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. Choose **Virtual interfaces**.
3. Select the virtual interface and then choose **View details**.
4. Choose **Test history**.

   The console displays the virtual interface tests that you performed for the virtual interface.
5. To view the details for a specific test, select the test id.

To view the virtual interface failover test history using the AWS CLI

Use ListVirtualInterfaceTestHistory.

Stopping the virtual interface failover test

You can stop the virtual interface failover test using the AWS Direct Connect console, or the AWS CLI.

To stop the virtual interface failover test from the AWS Direct Connect console

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. Choose **Virtual interfaces**.
3. Select the virtual interface, and then choose **Actions**, **Cancel test**.
4. Choose **Confirm**.

AWS restores the BGP peering session. The testing history displays "cancelled" for the test.

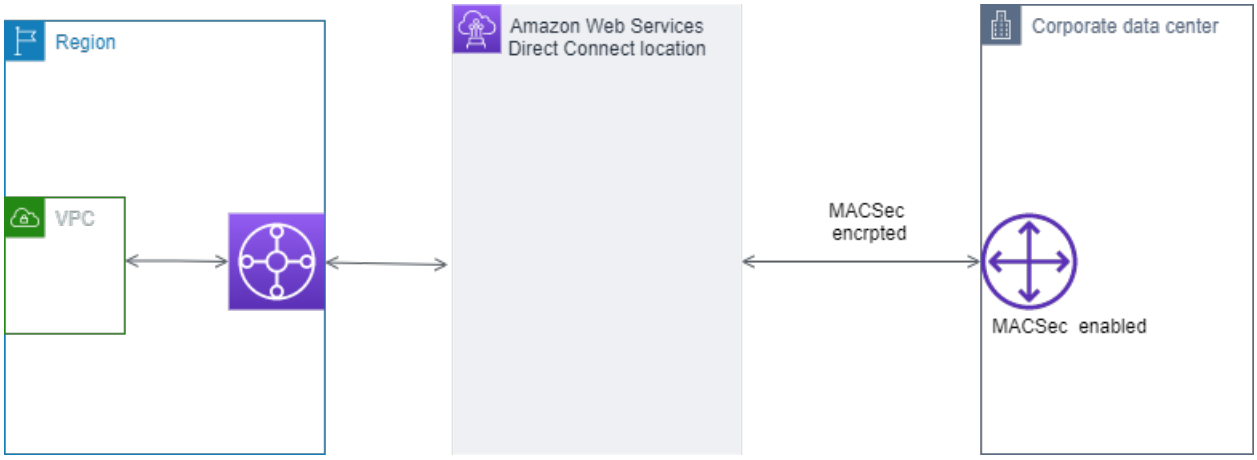
To stop the virtual interface failover test using the AWS CLI

Use StopBgpFailoverTest.

MAC Security

MAC Security (MACsec) is an IEEE standard that provides data confidentiality, data integrity, and data origin authenticity. You can use AWS Direct Connect connections that support MACsec to encrypt your data from your corporate data center to the AWS Direct Connect location.

In the following diagram, both the dedicated connection and your on-premises resources must support MACsec. Layer 2 traffic that travels over the dedicated connection to or from the data center is encrypted.



MACsec concepts

The following are the key concepts for MACsec:

- **MAC Security (MACsec)** — An IEEE 802.1 Layer 2 standard that provides data confidentiality, data integrity, and data origin authenticity. For more information about the protocol, see 802.1AE: MAC Security (MACsec).
- **MACsec secret key** — A pre-shared key that establishes the MACsec connectivity between the customer on-premises router and the connection port at the AWS Direct Connect location. The key is generated by the devices at the ends of the connection using the CKN/CAK pair that you provide to AWS and have also provisioned on your device.
- **Connection Key Name (CKN)** and **Connectivity Association Key (CAK)** — The values in this pair are used to generate the MACsec secret key. You generate the pair values, associate them with an AWS Direct Connect connection, and provision them on your edge device at your end of the AWS Direct Connect connection.

Supported connections

MACsec is available on dedicated connections. For information about how to order connections that support MACsec, see AWS Direct Connect.

Get started with MACsec on dedicated connections

The following tasks help you become familiar with MACsec on AWS Direct Connect dedicated connections.

Follow these steps to create a connection with MACsec support, and then associate a CKN/CAK pair with the connection.

Tasks

MACsec prerequisites

Complete the following tasks before you configure MACsec on a dedicated connection.

- Create a CKN/CAK pair for the MACsec secret key.

  You can create the pair using an open standard tool. The pair must meet the requirements specified in the section called "Step 4: Configure your on-premises router" (p. 44).
- MACsec is available on dedicated connections for certain AWS Direct Connect Partners. For information about which AWS Direct Connect Partners support MACsec, see AWS Direct Connect.
- Make sure that you have a device on your end of the connection that supports MACsec.

Service-Linked roles

AWS Direct Connect uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to AWS Direct Connect. Service-linked roles are predefined by AWS Direct Connect and include all of the permissions that the service requires to call other AWS services on your behalf. A service-linked role makes setting up AWS Direct Connect easier because you don't have to manually add the necessary permissions. AWS Direct Connect defines the permissions of its service-linked roles, and unless defined otherwise, only AWS Direct Connect can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity. For more information, see the section called "Using service-linked roles" (p. 123).

MACsec pre-shared CKN/CAK key considerations

AWS Direct Connect uses AWS managed CMKs for the pre-shared keys that you associate with connections or LAGs. Secrets Manager stores your pre-shared CKN and CAK pairs as a secret that the Secrets Manager's root key encrypts. For more information, see AWS managed CMKs in the *AWS Key Management Service Developer Guide*.

The stored key is read-only by design, but you can schedule a seven- to thirty-day deletion using the AWS Secrets Manager console or API. When you schedule a deletion, the CKN cannot be read, and this might affect your network connectivity. We apply the following rules when this happens:

- If the connection is in a pending state, we disassociate the CKN from the connection.
- If the connection is in an available state, we notify the connection owner by email. If you do not take any action within 30 days, we disassociate the CKN from your connection.

When we disassociate the last CKN from your connection and the connection encryption mode is set to "must encrypt", we set the mode to "should_encrypt" to prevent sudden packet loss.

Step 1: Create a connection

To start using MACsec, you must turn the feature on when you create a dedicated connection. For more information, see the section called “Create a connection” (p. 46).

(Optional) Step 2: Create a link aggregation group (LAG)

If you use multiple connections for redundancy, you can create a LAG that supports MACsec. For more information, see the section called “MACsec considerations” (p. 81) and the section called “Create a LAG” (p. 81).

Step 3: Associate the CKN/CAK with the connection or LAG

After you create the connection or LAG that supports MACsec, you need to associate a CKN/CAK with the connection. For more information, see one of the following:

- the section called “Associate a MACsec CKN/CAK with a connection” (p. 49)
- the section called “Associate a MACsec CKN/CAK with a LAG” (p. 85)

Step 4: Configure your on-premises router

Update your on-premises router with the MACsec secret key. The MACsec secret key on the on-premises router and in the AWS Direct Connect location must match. For more information, see the section called “Download the router configuration file” (p. 69).

Step 5: (Optional) Remove the association between the CKN/CAK and the connection or LAG

If you need to remove the association between the MACsec key and the connection or LAG, see one of the following:

- the section called “Remove the association between a MACsec secret key and a connection” (p. 50)
- the section called “Remove the association between a MACsec secret key and a LAG” (p. 86)

AWS Direct Connect connections

AWS Direct Connect enables you to establish a dedicated network connection between your network and one of the AWS Direct Connect locations.

There are two types of connections:

- Dedicated Connection: A physical Ethernet connection associated with a single customer. Customers can request a dedicated connection through the AWS Direct Connect console, the CLI, or the API.
- Hosted Connection: A physical Ethernet connection that an AWS Direct Connect Partner provisions on behalf of a customer. Customers request a hosted connection by contacting a partner in the AWS Direct Connect Partner Program, who provisions the connection.

Dedicated connections

To create an AWS Direct Connect dedicated connection, you need the following information:

AWS Direct Connect location

Work with a partner in the AWS Direct Connect Partner Program to help you establish network circuits between an AWS Direct Connect location and your data center, office, or colocation environment. They can also help provide colocation space within the same facility as the location. For more information, see APN Partners Supporting AWS Direct Connect.

Port speed

The possible values are 1 Gbps, 10 Gbps, and 100 Gbps.

You cannot change the port speed after you create the connection request. To change the port speed, you must create and configure a new connection.

After you request the connection, we make a Letter of Authorization and Connecting Facility Assignment (LOA-CFA) available to you to download, or emails you with a request for more information. If you receive a request for more information, you must respond within 7 days or the connection is deleted. The LOA-CFA is the authorization to connect to AWS, and is required by your network provider to order a cross connect for you. If you do not have equipment in the AWS Direct Connect location, you cannot order a cross connect for yourself there.

The following operations are available for dedicated connections:

- the section called “Create a connection” (p. 46)
- the section called “View your connection details” (p. 48)
- the section called “Update a connection” (p. 48)
- the section called “Associate a MACsec CKN/CAK with a connection” (p. 49)
- the section called “Remove the association between a MACsec secret key and a connection” (p. 50)
- the section called “Delete connections” (p. 51)

You can add a dedicated connection to a link aggregation group (LAG) allowing you to treat multiple connections as a single one. For information, see Associate a connection with a LAG (p. 84).

After you create a connection, create a virtual interface to connect to public and private AWS resources. For more information, see AWS Direct Connect virtual interfaces (p. 61).

Hosted connections

To create an AWS Direct Connect hosted connection, you need the following information:

AWS Direct Connect location

Work with an AWS Direct Connect Partner in the AWS Direct Connect Partner Program to help you establish network circuits between an AWS Direct Connect location and your data center, office, or colocation environment. They can also help provide colocation space within the same facility as the location. For more information, see APN Partners Supporting AWS Direct Connect.

Port speed

For hosted connections, the possible values are 50 Mbps, 100 Mbps, 200 Mbps, 300 Mbps, 400 Mbps, 500 Mbps, 1 Gbps, 2 Gbps, 5 Gbps, and 10 Gbps. Note that only those AWS Direct Connect partners who have met specific requirements may create a 1 Gbps, 2 Gbps, 5 Gbps or 10 Gbps hosted connection.

You cannot change the port speed after you create the connection request. To change the port speed, you must create and configure a new connection.

AWS uses traffic policing on hosted connections, which means that when the traffic rate reaches the configured maximum rate, excess traffic is dropped. This might result in bursty traffic having a lower throughput than non-bursty traffic.

The following operations are available for hosted connections:

- the section called “Create a connection” (p. 46)

  After the AWS Direct Connect Partner configures the connection, it appears in the **Connections** pane in the AWS Direct Connect console. You must accept the hosted connection before you can use it. For more information, see the section called “Accept a hosted connection” (p. 51).
- the section called “View your connection details” (p. 48)
- the section called “Update a connection” (p. 48)
- the section called “Delete connections” (p. 51)

After you accept a connection, create a virtual interface to connect to public and private AWS resources. For more information, see AWS Direct Connect virtual interfaces (p. 61).

Create a connection

You can create a standalone connection, or you can create a connection to associate with a LAG in your account. If you associate a connection with a LAG, it's created with the same port speed and location that is specified in the LAG.

If you do not have equipment at an AWS Direct Connect location, first contact an AWS Direct Connect Partner at the AWS Direct Connect Partner Program. For more information, see APN Partners Supporting AWS Direct Connect.

If you want to create a connection that uses MAC Security (MACsec), review the prerequisites before you create the connection. For more information, see the section called “MACsec prerequisites ” (p. 43).

Console

To create a new connection

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. On the **AWS Direct Connect** screen, under **Get started**, choose **Create a connection**.
3. On the **Create Connection** pane, under **Connection settings,** do the following:

   a. For **Name**, enter a name for the connection.
   b. For **Location**, select the appropriate AWS Direct Connect location.
   c. If applicable, for **Sub Location**, choose the floor closest to you or your network provider. This option is only available if the location has meet-me rooms (MMRs) in multiple floors of the building.
   d. For **Port Speed**, choose the connection bandwidth.
   e. For **On-premises**, select **Connect through an AWS Direct Connect partner** when you use this connection to connect to your data center.
   f. (Optional) Configure MAC security (MACsec) for the connection. Under **Additional Settings**, select **Request a MACsec capable port**.

      MACsec is only available on dedicated connections.
   g. a. (Optional) Add or remove a tag.

      [Add a tag] Choose **Add tag** and do the following:
      - For **Key**, enter the key name.
      - For **Value**, enter the key value.

      [Remove a tag] Next to the tag, choose **Remove tag**.
4. Choose **Create Connection**.

Command line

Use one of the following commands.

- create-connection (AWS CLI)
- CreateConnection (AWS Direct Connect API)

Download the LOA-CFA

After we have processed your connection request, you can download the LOA-CFA.

If you need to change the LOA-CFA after it has been created (for example, you need to change the ports), contact AWS Support.

The LOA-CFA expires after 90 days. If your connection is not up after 90 days, we send you an email alerting you that the LOA-CFA has expired. To refresh the LOA-CFA with a new issue date, download it again from the AWS Direct Connect console. If you do not take any action, we delete the connection.

Note
Port-hour billing starts 90 days after you created the connection, or after the connection between your router and the AWS Direct Connect endpoint is established, whichever comes first. For more information, see AWS Direct Connect Pricing. If you no longer want the connection after you have reissued the LOA-CFA, you must delete the connection yourself. For more information, see Delete connections (p. 51).

Console

To download the LOA-CFA

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Connections**.
3. Select the connection, and then choose **View details**.
4. Choose **Download LOA-CFA**.

   Note
   If the link is not enabled, the LOA-CFA is not yet available for you to download. Check your email for a request for information. If it's still unavailable, or you haven't received an email after 72 hours, contact AWS Support.
5. Send the LOA-CFA to your network provider or colocation provider so that they can order a cross connect for you. The contact process can vary for each colocation provider. For more information, see Requesting cross connects at AWS Direct Connect locations (p. 53).

Command line

To download the LOA-CFA using the command line or API

- describe-loa (AWS CLI)
- DescribeLoa (AWS Direct Connect API)

View your connection details

You can view the current status of your connection. You can also view your connection ID (for example, `dxcon-12nikabc`) and verify that it matches the connection ID on the LOA-CFA that you received or downloaded.

For information on monitoring connections, see *Monitoring* (p. 139).

Console

To view details about a connection

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the left pane, choose **Connections**.
3. Select a connection, and then choose **View details**.

Command line

To describe a connection using the command line or API

- describe-connections (AWS CLI)
- DescribeConnections (AWS Direct Connect API)

Update a connection

You can update the following connection attributes:

- The name of the connection.
- The connection's MACsec encryption mode.

  MACsec is only available on dedicated connections.

  The valid values are:
  - `should_encrypt`
  - `must_encrypt`

    When you set the encryption mode to this value, the connection goes down when the encryption is down.
  - `no_encrypt`

Console

To update a connection

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Connections**.
3. Select the connection, and then choose **Edit**.
4. Modify the connection:

   [Change the name] For **Name**, enter a new connection name.

   [Add a tag] Choose **Add tag** and do the following:
   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Next to the tag, choose **Remove tag**.
5. Choose **Edit connection**.

Command line

To add a tag or remove a tag using the command line

- tag-resource (AWS CLI)
- untag-resource (AWS CLI)

To update a connection using the command line or API

- update-connection (AWS CLI)
- UpdateConnection (AWS Direct Connect API)

Associate a MACsec CKN/CAK with a connection

After you create the connection that supports MACsec, you can associate a CKN/CAK with the connection.

Note
You cannot modify a MACsec secret key after you associate it with a connection. If you need to modify the key, disassociate the key from the connection, and then associate a new key with the

connection. For information about removing an association, see the section called “Remove the association between a MACsec secret key and a connection” (p. 50).

Console

To associate a MACsec key with a connection

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the left pane, choose **Connections**.
3. Select a connection, and then choose **View details**.
4. Choose **Associate key**.
5. Enter the MACsec key.

   [Use the CAK/CKN pair] Choose **Key Pair**, and then do the following:

   - For **Connectivity Association Key (CAK)**, enter the CAK.
   - For **Connectivity Association Key Name (CKN)**, enter the CKN.

   [Use the secret] Choose **Existing Secret Manager secret**, and then for **Secret**, select the MACsec secret key.
6. Choose **Associate key**.

Command line

To associate a MACsec key with a connection

- associate-mac-sec-key (AWS CLI)
- AssociateMacSecKey (AWS Direct Connect API)

Remove the association between a MACsec secret key and a connection

You can remove the association between the connection and the MACsec key.

Console

To remove an association between a connection and a MACsec key

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2.
3. In the left pane, choose **Connections**.
4. Select a connection, and then choose **View details**.
5. Select the MACsec secret to remove, and then choose **Disassociate key**.
6. In the confirmation dialog box, enter **disassociate**, and then choose **Disassociate**.

Command line

To remove an association between a connection and a MACsec key

- disassociate-mac-sec-key (AWS CLI)
- DisassociateMacSecKey (AWS Direct Connect API)

Delete connections

You can delete a connection as long as there are no virtual interfaces attached to it. Deleting your connection stops all port hour charges for this connection. AWS Direct Connect data transfer charges are associated with virtual interfaces. Any cross connect or network circuit charges are independent of AWS Direct Connect and must be cancelled separately. For more information about how to delete a virtual interface, see Delete virtual interfaces (p. 74).

If the connection is part of a link aggregation group (LAG), you cannot delete the connection if doing so causes the LAG to fall below its setting for the minimum number of operational connections.

Console

To delete a connection

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Connections**.
3. Select the connections and choose **Delete**.
4. In the **Delete confirmation** dialog box, choose **Delete**.

Command line

To delete a connection using the command line or API

- delete-connection (AWS CLI)
- DeleteConnection (AWS Direct Connect API)

Accept a hosted connection

If you are interested in purchasing a hosted connection, you must contact an AWS Direct Connect Partner in the AWS Direct Connect Partner Program. The partner provisions the connection for you. After the connection is configured, it appears in the **Connections** pane in the AWS Direct Connect console.

Before you can begin using a hosted connection, you must accept the connection.

Console

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Connections**.
3. Select the hosted connection and choose **View details**.
4. Select the confirmation check box and choose **Accept connection**.

Command line

To accept a hosted connection using the command line or API

- confirm-connection (AWS CLI)
- ConfirmConnection (AWS Direct Connect API)

Requesting cross connects at AWS Direct Connect locations

After you have downloaded your Letter of Authorization and Connecting Facility Assignment (LOA-CFA), you must complete your cross-network connection, also known as a *cross connect*. If you already have equipment located in an AWS Direct Connect location, contact the appropriate provider to complete the cross connect. For specific instructions for each provider, see the table below. Contact your provider for cross connect pricing. After the cross connect is established, you can create the virtual interfaces using the AWS Direct Connect console.

Some locations are set up as a campus. For more information, see AWS Direct Connect Locations.

If you do not already have equipment located in an AWS Direct Connect location, you can work with one of the partners in the AWS Partner Network (APN). They help you to connect to an AWS Direct Connect location. For more information, see APN Partners supporting AWS Direct Connect. You must share the LOA-CFA with your selected provider to facilitate your cross connect request.

An AWS Direct Connect connection can provide access to resources in other Regions. For more information, see Accessing a remote AWS Region (p. 3).

> **Note**
> If the cross connect is not completed within 90 days, the authority granted by the LOA-CFA expires. To renew a LOA-CFA that has expired, you can download it again from the AWS Direct Connect console. For more information, see Download the LOA-CFA (p. 47).

Contents

- US West (N. California) (p. 60)
- US West (Oregon) (p. 60)

Africa (Cape Town)

Location	How to request a connection
Cape Town Internet Exchange/ Teraco Data Centres	Contact Teraco at support@teraco.co.za for existing Teraco customers or connect@teraco.co.za for new customers.
Teraco JB1, Johannesburg, South Africa	Contact Teraco at support@teraco.co.za for existing Teraco customers or connect@teraco.co.za for new customers.

Asia Pacific (Mumbai)

Location	How to request a connection
GPX, Mumbai	Contact GPX at nkankane@gpxglobal.net.
NetMagic DC2, Bangalore	Contact NetMagic Sales and Marketing toll-free at 18001033130 or at marketing@netmagicsolutions.com.
Sify Rabale, Mumbai	Contact Sify at aws.directconnect@sifycorp.com.
STT Delhi DC2, Delhi	Contact STT at enquiry.AWSDX@sttelemediagdc.in.
STT GDC Pvt. Ltd. VSB, Chennai	Contact STT at enquiry.AWSDX@sttelemediagdc.in.
STT Hyderabad DC1, Hyderabad	Contact STT at enquiry.AWSDX@sttelemediagdc.in.

Asia Pacific (Seoul)

Location	How to request a connection
KINX Gasan Data Center, Seoul	Contact KINX at sales@kinx.net.
LG U+ Pyeong-Chon Mega Center, Seoul	Submit the LOA document to kidcadmin@lguplus.co.kr and center8@kidc.net.

Asia Pacific (Singapore)

Location	How to request a connection
Equinix SG2, Singapore	Contact Equinix at awsdealreg@equinix.com.
Global Switch, Singapore	Contact Global Switch at salessingapore@globalswitch.com.
GPX, Mumbai	Contact GPX at nkankane@gpxglobal.net.

Location	How to request a connection
iAdvantage Mega-i, Hong Kong	Contact iAdvantage at cs@iadvantage.net or place an order using iAdvantage Cabling Order e-Form.
Menara AIMS, Kuala Lumpur	Existing AIMS customers can request an X-Connect order using the Customer Service portal by filling out the Engineering Work Order Request Form. Contacting service.delivery@aims.com.my if there are any problems submitting the request.

Asia Pacific (Sydney)

Location	How to request a connection
Equinix SY3, Sydney	Contact Equinix at awsdealreg@equinix.com.
Global Switch, Sydney	Contact Global Switch at salessydney@globalswitch.com.
NEXTDC C1, Canberra	Contact NEXTDC at nxtops@nextdc.com.
NEXTDC M1, Melbourne	Contact NEXTDC at nxtops@nextdc.com.
NEXTDC P1, Perth	Contact NEXTDC at nxtops@nextdc.com.

Asia Pacific (Tokyo)

Location	How to request a connection
AT Tokyo Chuo Data Center, Tokyo	Contact AT TOKYO at at-sales@attokyo.co.jp.
Chief Telecom LY, Taipei	Contact Chief Telecom at vicky_chan@chief.com.tw.
Chunghwa Telecom, Taipei	Contact CHT Taipei IDC NOC at taipei_idc@cht.com.tw.
Equinix OS1, Osaka	Contact Equinix at awsdealreg@equinix.com.
Equinix TY2, Tokyo	Contact Equinix at awsdealreg@equinix.com.

AWS GovCloud (US-East)

You can't order connections in this Region.

AWS GovCloud (US-West)

Location	How to request a connection
Equinix SV5, San Jose	Contact Equinix at awsdealreg@equinix.com.

Canada (Central)

Location	How to request a connection
Allied 250 Front St W, Toronto	Contact driches@alliedreit.com.
Cologix MTL3, Montreal	Contact Cologix at aws@cologix.com.
Cologix VAN2, Vancouver	Contact Cologix at aws@cologix.com.
eStruxture, Montreal	Contact eStruxture at directconnect@estruxture.com.

China (Beijing)

Location	How to request a connection
CIDS Jiachuang IDC, Beijing	Contact dx-order@sinnet.com.cn.
Sinnet Jiuxianqiao IDC, Beijing	Contact dx-order@sinnet.com.cn.
GDS No. 3 Data Center, Shanghai	Contact dx@nwcdcloud.cn.
GDS No. 3 Data Center, Shenzhen	Contact dx@nwcdcloud.cn.

China (Ningxia)

Location	How to request a connection
Industrial Park IDC, Ningxia	Contact dx@nwcdcloud.cn.
Shapotou IDC, Ningxia	Contact dx@nwcdcloud.cn.

Europe (Frankfurt)

Location	How to request a connection
CE Colo, Prague, Czech Republic	Contact CE Colo at info@cecolo.com.
DigiPlex Ulven, Oslo, Norway	Contact DigiPlex at helpme@digiplex.com.
Equinix AM3, Amsterdam, Netherlands	Contact Equinix at awsdealreg@equinix.com.
Equinix FR5, Frankfurt	Contact Equinix at awsdealreg@equinix.com.
Equinix HE6, Helsinki	Contact Equinix at awsdealreg@equinix.com.
Equinix MU1, Munich	Contact Equinix at awsdealreg@equinix.com.

Location	How to request a connection
Equinix WA1, Warsaw	Contact Equinix at awsdealreg@equinix.com.
Interxion AMS7, Amsterdam	Contact Interxion at customer.services@interxion.com.
Interxion CPH2, Copenhagen	Contact Interxion at customer.services@interxion.com.
Interxion FRA6, Frankfurt	Contact Interxion at customer.services@interxion.com.
Interxion MAD2, Madrid	Contact Interxion at customer.services@interxion.com.
Interxion VIE2, Vienna	Contact Interxion at customer.services@interxion.com.
Interxion ZUR1, Zurich	Contact Interxion at customer.services@interxion.com.
IPB, Berlin	Contact IPB at kontakt@ipb.de.
Equinix ITConic MD2, Madrid	Contact Equinix at awsdealreg@equinix.com.

Europe (Ireland)

Location	How to request a connection
Digital Realty (UK), Docklands	Contact Digital Realty (UK) at amazon.orders@digitalrealty.com.
Eircom Clonshaugh	Contact Eircom at awsorders@eircom.ie.
Equinix DX1, Dubai	Contact Equinix at awsdealreg@equinix.com.
Equinix LD5, London (Slough)	Contact Equinix at awsdealreg@equinix.com.
Interxion DUB2, Dublin	Contact Interxion at customer.services@interxion.com.
Interxion MRS1, Marseille	Contact Interxion at customer.services@interxion.com.

Europe (Italy)

Location	How to request a connection
CDLAN srl Via Caldera 21, Milano	Contact CDLAN at sales@cdlan.it.
Equinix, ML2, Milano, Italy	Contact Equinix at awsdealreg@equinix.com.

Europe (London)

Location	How to request a connection
Digital Realty (UK), Docklands	Contact Digital Realty (UK) at amazon.orders@digitalrealty.com.

Location	How to request a connection
Equinix LD5, London (Slough)	Contact Equinix at awsdealreg@equinix.com.
Equinix MA3, Manchester	Contact Equinix at awsdealreg@equinix.com.
Telehouse West, London	Contact Telehouse UK at sales.support@uk.telehouse.net.

Europe (Paris)

Location	How to request a connection
Equinix PA3, Paris	Contact Equinix at awsdealreg@equinix.com.
Interxion PAR7, Paris	Contact Interxion at customer.services@interxion.com.
Telehouse Voltaire, Paris	Create a request using the Customer Portal. The request type is DFM/SFM Layout/Connectivity/MMR Circuit Commissioning.

Europe (Stockholm)

Location	How to request a connection
Interxion STO1, Stockholm	Contact Interxion at customer.services@interxion.com.

Middle East (Bahrain)

Location	How to request a connection
AWS Bahrain DC53, Manama	To complete the connection, you can work with one of our network provider partners at the location to establish connectivity. You will then provide a Letter of Authorization (LOA) from the network provider to AWS through the AWS Support Center. AWS completes the cross-connect at this location.
AWS Bahrain DC52, Manama	To complete the connection, you can work with one of our network provider partners at the location to establish connectivity. You will then provide a Letter of Authorization (LOA) from the network provider to AWS through the AWS Support Center. AWS completes the cross-connect at this location.

Middle East (Israel)

Location	How to request a connection
MedOne in Haifa, Israel	Contact MedOne at support@Medone.co.il

South America (São Paulo)

Location	How to request a connection
Equinix RJ2, Rio de Janeiro	Contact Equinix at awsdealreg@equinix.com.
Equinix SP4, São Paulo	Contact Equinix at awsdealreg@equinix.com.
Tivit	Contact Tivit at aws@tivit.com.br.

US East (Ohio)

Location	How to request a connection
Cologix COL2, Columbus	Contact Cologix at aws@cologix.com.
Cologix MIN3, Minneapolis	Contact Cologix at aws@cologix.com.
CyrusOne West III, Houston	Submit a request using customer portal.
Equinix CH2, Chicago	Contact Equinix at awsdealreg@equinix.com.
QTS Chicago	Contact QTS at AConnect@qtsdatacenters.com.
Netrality Properties, 1102 Grand, Kansas City	Contact Netrality Properties at support@netrality.com.

US East (N. Virginia)

Location	How to request a connection
165 Halsey Street, Newark	Contact operations@165halsey.com.
CoreSite NY1, New York	Place an order using the CoreSite Customer Portal. After you complete the form, review the order for accuracy, and then approve it using the website.
CoreSite VA1, Reston	Place an order at the CoreSite Customer Portal. After you complete the form, review the order for accuracy, and then approve it using the website.
Digital Realty ATL1, Atlanta	Contact Digital Realty at amazon.orders@digitalrealty.com.
Equinix DC2/DC11, Ashburn	Contact Equinix at awsdealreg@equinix.com.
Equinix DA2, Dallas	Contact Equinix at awsdealreg@equinix.com.
Equinix MI1, Miami	Contact Equinix at awsdealreg@equinix.com.
CrownCastle (formerly Lightower), Philadelphia, PA	Contact CrownCastle at awsorders@crowncastle.com.

Location	How to request a connection
Markley, One Summer Street, Boston	Create a request using the customer portal. For new queries, contact sales@markleygroup.com.

US West (N. California)

Location	How to request a connection
CoreSite LA1, Los Angeles	Place an order using the CoreSite Customer Portal. After you complete the form, review the order for accuracy, and then approve it using the website.
CoreSite SV2, Milpitas	Place an order using the CoreSite Customer Portal. After you complete the form, review the order for accuracy, and then approve it using the website.
CoreSite SV4, Santa Clara	Place an order using the CoreSite Customer Portal. After you complete the form, review the order for accuracy, and then approve it using the MyCoreSite website.
Equinix LA3, El Segundo	Contact Equinix at awsdealreg@equinix.com.
Equinix SV5, San Jose	Contact Equinix at awsdealreg@equinix.com.
PhoenixNAP, Phoenix	Contact phoenixNAP Provisioning at provisioning@phoenixnap.com.

US West (Oregon)

Location	How to request a connection
CoreSite DE1, Denver	Place an order using the CoreSite Customer Portal. After you complete the form, review the order for accuracy, and then approve it using the website.
EdgeConneX, Portland	Place an order using the EdgeOS Customer Portal. After you have submitted the form, EdgeConneX will provide a service order form for approval. You can send questions to cloudaccess@edgeconnex.com.
Equinix SE2, Seattle	Contact Equinix at support@equinix.com.
Pittock Block, Portland	Send requests by email to crossconnect@pittock.com or by phone at +1 503 226 6777.
Switch SUPERNAP 8, Las Vegas	Contact Switch SUPERNAP at orders@supernap.com.
TierPoint Seattle	Contact TierPoint at sales@tierpoint.com.

AWS Direct Connect virtual interfaces

You must create one of the following virtual interfaces to begin using your AWS Direct Connect connection.

- Private virtual interface: A private virtual interface should be used to access an Amazon VPC using private IP addresses.
- Public virtual interface: A public virtual interface can access all AWS public services using public IP addresses.
- Transit virtual interface: A transit virtual interface should be used to access one or more Amazon VPC Transit Gateways associated with Direct Connect gateways. You can use transit virtual interfaces with 1/2/5/10 Gbps AWS Direct Connect connections. For information about Direct Connect gateway configurations, see the section called “Direct Connect gateways” (p. 88).

To connect to other AWS services using IPv6 addresses, check the service documentation to verify that IPv6 addressing is supported.

Public virtual interface prefix advertisement rules

We advertise appropriate Amazon prefixes to you so that you can reach either your VPCs or other AWS services. You can access all AWS prefixes through this connection; for example, Amazon EC2, Amazon S3, and Amazon.com. You do not have access to non-Amazon prefixes. For a current list of prefixes advertised by AWS, see AWS IP Address Ranges in the *Amazon Web Services General Reference*.

Note
We recommend that you use a firewall filter (based on the source/destination address of packets) to control traffic to and from some prefixes. If you're using a prefix filter (route map), ensure that it accepts prefixes with an exact match or longer. Prefixes advertised from AWS Direct Connect may be aggregated and may differ from the prefixes defined in your prefix filter.

Hosted virtual interfaces

To use your AWS Direct Connect connection with another account, you can create a hosted virtual interface for that account. The owner of the other account must accept the hosted virtual interface to begin using it. A hosted virtual interface works the same as a standard virtual interface and can connect to public resources or a VPC.

A connection of less than 1 Gbps supports only one virtual interface.

To create a virtual interface, you need the following information:

Resource	Required information
Connection	The AWS Direct Connect connection or link aggregation group (LAG) for which you are creating the virtual interface.
Virtual interface name	A name for the virtual interface.

Resource	Required information
Virtual interface owner	If you're creating the virtual interface for another account, you need the AWS account ID of the other account.
(Private virtual interface only) **Connection**	For connecting to a VPC in the same AWS Region, you need the virtual private gateway for your VPC. The ASN for the Amazon side of the BGP session is inherited from the virtual private gateway. When you create a virtual private gateway, you can specify your own private ASN. Otherwise, Amazon provides a default ASN. For more information, see Create a Virtual Private Gateway in the *Amazon VPC User Guide*. For connecting to a VPC through a Direct Connect gateway, you need the Direct Connect gateway. For more information, see Direct Connect Gateways.
VLAN	A unique virtual local area network (VLAN) tag that's not already in use on your connection. The value must be between 1 and 4094 and must comply with the Ethernet 802.1Q standard. This tag is required for any traffic traversing the AWS Direct Connect connection. If you have a hosted connection, your AWS Direct Connect Partner provides this value. You can't modify the value after you have created the virtual interface.

Peer IP addresses	A virtual interface can support a BGP peering session for IPv4, IPv6, or one of each (dual-stack). You cannot create multiple BGP sessions for the same IP addressing family on the same virtual interface. The IP address ranges are assigned to each end of the virtual interface for the BGP peering session.
	
	• IPv4:
	  • (Public virtual interface only) You must specify unique public IPv4 addresses that you own. The value can be one of the following:
	  • A customer-owned ASN
	  • An ASN owned by your AWS Direct Connect Partner or ISP
	  • An AWS provided /31 CIDR. Contact contact AWS Support to request a public IPv4 CIDR (and provide a use case in your request)
	  • (Private virtual interface only) Amazon can generate private IPv4 addresses for you. If you specify your own, ensure that you specify private CIDRs for your router interface and the AWS Direct Connect interface only (for example, do not specify other IP addresses from your local network).
	• IPv6: Amazon automatically allocates you a /125 IPv6 CIDR. You cannot specify your own peer IPv6 addresses.

Address family	Whether the BGP peering session will be over IPv4 or IPv6.
BGP information	• A public or private Border Gateway Protocol (BGP) Autonomous System Number (ASN) for your side of the BGP session. If you are using a public ASN, you must own it. If you are using a private ASN, it must be in the 1 to 2147483647 range. Autonomous System (AS) prepending does not work if you use a private ASN for a public virtual interface. • AWS enables MD5 by default. You cannot modify this option. • An MD5 BGP authentication key. You can provide your own, or you can let Amazon generate one for you.

Resource	Required information

(Public virtual interface only) **Prefixes you want to advertise**	Public IPv4 routes or IPv6 routes to advertise over BGP. You must advertise at least one prefix using BGP, up to a maximum of 1,000 prefixes.
	• IPv4: The IPv4 CIDR can overlap with another public IPv4 CIDR announced using AWS Direct Connect when either of the following is true:
	  • The CIDRs are from different AWS Regions. Make sure that you apply BGP community tags on the public prefixes.
	  • You use AS_PATH when you have a public ASN in an active/passive configuration.
	  For more information, see Routing policies and BGP communities.
	• IPv6: Specify a prefix length of /64 or shorter.

(Private virtual interface only) **Jumbo frames**	The maximum transmission unit (MTU) of packets over AWS Direct Connect. The default is 1500. Setting the MTU of a virtual interface to 9001 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your virtual private gateway, then traffic routed through the static routes is sent using 1500 MTU. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo frame capable** on the virtual interface **General configuration** page.
(Transit virtual interface only) **Jumbo frames**	The maximum transmission unit (MTU) of packets over AWS Direct Connect. The default is 1500. Setting the MTU of a virtual interface to 8500 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your transit gateway, then traffic routed through the static routes is sent using 1500 MTU. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo frame capable** on the virtual interface **General configuration** page.

You can create a virtual interface for accounts within your AWS Organizations, or AWS Organizations that are different from yours. You must accept the virtual interface before you can use it. For information about how to create and accept a virtual interface, see the section called “Create a hosted virtual interface” (p. 74) and the section called “Accept a hosted virtual interface” (p. 77).

Prerequisites for virtual interfaces

Before you create a virtual interface, do the following:

- Create a connection. For more information, see the section called “Create a connection” (p. 46).
- Create a link aggregation group (LAG) when you have multiple connections that you want to treat as a single one. For information, see Associate a connection with a LAG (p. 84).

To create a virtual interface, you need the following information:

Resource	Required information
Connection	The AWS Direct Connect connection or link aggregation group (LAG) for which you are creating the virtual interface.
Virtual interface name	A name for the virtual interface.
Virtual interface owner	If you're creating the virtual interface for another account, you need the AWS account ID of the other account.
(Private virtual interface only) **Connection**	For connecting to a VPC in the same AWS Region, you need the virtual private gateway for your VPC. The ASN for the Amazon side of the BGP session is inherited from the virtual private gateway. When you create a virtual private gateway, you can specify your own private ASN. Otherwise, Amazon provides a default ASN. For more information, see Create a Virtual Private Gateway in the *Amazon VPC User Guide*. For connecting to a VPC through a Direct Connect gateway, you need the Direct Connect gateway. For more information, see Direct Connect Gateways.
VLAN	A unique virtual local area network (VLAN) tag that's not already in use on your connection. The value must be between 1 and 4094 and must comply with the Ethernet 802.1Q standard. This tag is required for any traffic traversing the AWS Direct Connect connection. If you have a hosted connection, your AWS Direct Connect Partner provides this value. You can't modify the value after you have created the virtual interface.

Peer IP addresses	A virtual interface can support a BGP peering session for IPv4, IPv6, or one of each (dual-stack). You cannot create multiple BGP sessions for the same IP addressing family on the same virtual interface. The IP address ranges are assigned to each end of the virtual interface for the BGP peering session.
	
	• IPv4:
	  • (Public virtual interface only) You must specify unique public IPv4 addresses that you own. The value can be one of the following:
	  • A customer-owned ASN
	  • An ASN owned by your AWS Direct Connect Partner or ISP
	  • An AWS provided /31 CIDR. Contact contact AWS Support to request a public IPv4 CIDR (and provide a use case in your request)
	  • (Private virtual interface only) Amazon can generate private IPv4 addresses for you. If you specify your own, ensure that you specify private CIDRs for your router interface and the AWS Direct Connect interface only (for example, do not specify other IP addresses from your local network).
	• IPv6: Amazon automatically allocates you a /125 IPv6 CIDR. You cannot specify your own peer IPv6 addresses.

Address family	Whether the BGP peering session will be over IPv4 or IPv6.
BGP information	• A public or private Border Gateway Protocol (BGP) Autonomous System Number (ASN) for your side of the BGP session. If you are using a public ASN, you must own it. If you are using a private ASN, it must be in the 1 to 2147483647 range. Autonomous System (AS) prepending does not work if you use a private ASN for a public virtual interface. • AWS enables MD5 by default. You cannot modify this option.

Resource	Required information
	• An MD5 BGP authentication key. You can provide your own, or you can let Amazon generate one for you.

(Public virtual interface only) **Prefixes you want to advertise**	Public IPv4 routes or IPv6 routes to advertise over BGP. You must advertise at least one prefix using BGP, up to a maximum of 1,000 prefixes.
	• IPv4: The IPv4 CIDR can overlap with another public IPv4 CIDR announced using AWS Direct Connect when either of the following is true:
	  • The CIDRs are from different AWS Regions. Make sure that you apply BGP community tags on the public prefixes.
	  • You use AS_PATH when you have a public ASN in an active/passive configuration.
	  For more information, see Routing policies and BGP communities.
	• IPv6: Specify a prefix length of /64 or shorter.

(Private virtual interface only) **Jumbo frames**	The maximum transmission unit (MTU) of packets over AWS Direct Connect. The default is 1500. Setting the MTU of a virtual interface to 9001 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your virtual private gateway, then traffic routed through the static routes is sent using 1500 MTU. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo frame capable** on the virtual interface **General configuration** page.
(Transit virtual interface only) **Jumbo frames**	The maximum transmission unit (MTU) of packets over AWS Direct Connect. The default is 1500. Setting the MTU of a virtual interface to 8500 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your transit gateway, then traffic routed through the static routes is sent using 1500 MTU. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo frame capable** on the virtual interface **General configuration** page.

When you create a virtual interface, you can specify the account that owns the virtual interface. When you choose an AWS account that is not your account, the following rules apply:

- For private VIFs and transit VIFs, the account applies to the virtual interface and the virtual private gateway/Direct Connect gateway destination.
- For public VIFs, the account is used for virtual interface billing. The Data Transfer Out (DTO) usage is metered toward the resource owner at AWS Direct Connect data transfer rate.

Create a virtual interface

You can create a transit virtual interface to connect to a transit gateway, a public virtual interface to connect to public resources (non-VPC services), or a private virtual interface to connect to a VPC.

To create a virtual interface for accounts within your AWS Organizations, or AWS Organizations that are different from yours, create a hosted virtual interface. For more information, see the section called "Create a hosted virtual interface" (p. 74).

Prerequisites

Before you begin, ensure that you have read the information in Prerequisites for virtual interfaces (p. 63).

Create a public virtual interface

When you create a public virtual interface, it can take up to 72 hours for us to review and approve your request.

To provision a public virtual interface

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Public**.
5. Under **Public virtual interface settings**, do the following:

   a. For **Virtual interface name**, enter a name for the virtual interface.
   b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
   c. For **VLAN**, enter the ID number for your virtual local area network (VLAN).
   d. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

      The valid values are 1-2147483647.
6. Under **Additional settings**, do the following:

   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

      [IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:
      - To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
      - For **Amazon router peer IP**, enter the IPv4 CIDR address to use to send traffic to AWS.

      [IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.
   b. To provide your own BGP key, enter your BGP MD5 key.

      If you do not enter a value, we generate a BGP key.
   c. To advertise prefixes to Amazon, for **Prefixes you want to advertise**, enter the IPv4 CIDR destination addresses (separated by commas) to which traffic should be routed over the virtual interface.
   d. (Optional) Add or remove a tag.

      [Add a tag] Choose **Add tag** and do the following:
      - For **Key**, enter the key name.
      - For **Value**, enter the key value.

      [Remove a tag] Next to the tag, choose **Remove tag**.
7. Choose **Create virtual interface**.

8. Download the router configuration for your device. For more information, see Download the router configuration file (p. 69).

To create a public virtual interface using the command line or API

- create-public-virtual-interface (AWS CLI)
- CreatePublicVirtualInterface (AWS Direct Connect API)

Create a private virtual interface

You can provision a private virtual interface to a virtual private gateway in the same Region as your AWS Direct Connect connection. For more information about provisioning a private virtual interface to an AWS Direct Connect gateway, see Working with Direct Connect gateways (p. 88).

If you use the VPC wizard to create a VPC, route propagation is automatically enabled for you. With route propagation, routes are automatically populated to the route tables in your VPC. If you choose, you can disable route propagation. For more information, see Enable Route Propagation in Your Route Table in the *Amazon VPC User Guide*.

The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The MTU of a virtual private interface can be either 1500 or 9001 (jumbo frames). The MTU of a transit virtual interface can be either 1500 or 8500 (jumbo frames). You can specify the MTU when you create the interface or update it after you create it. Setting the MTU of a virtual interface to 8500 (jumbo frames) or 9001 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo Frame Capable** on the **Summary** tab.

To provision a private virtual interface to a VPC

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Private**.
5. Under **Private virtual interface settings**, do the following:
   a. For **Virtual interface name**, enter a name for the virtual interface.
   b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
   c. For **Gateway type**, choose **Virtual private gateway**, or **Direct Connect gateway**.
   d. For **Virtual interface owner**, choose **Another AWS account**, and then enter the AWS account.
   e. For **Virtual private gateway**, choose the virtual private gateway to use for this interface.
   f. For **VLAN**, enter the ID number for your virtual local area network (VLAN).
   g. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

      The valid values are 1-2147483647.
6. Under **Additional Settings**, do the following:
   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

      [IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:

- To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
- For **Amazon router peer ip**, enter the IPv4 CIDR address to use to send traffic to AWS.

[IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.

b. To change the maximum transmission unit (MTU) from 1500 (default) to 9001 (jumbo frames), select **Jumbo MTU (MTU size 9001)**.

c. (Optional) Add or remove a tag.

[Add a tag] Choose **Add tag** and do the following:

- For **Key**, enter the key name.
- For **Value**, enter the key value.

[Remove a tag] Next to the tag, choose **Remove tag**.

7. Choose **Create virtual interface**.
8. Download the router configuration for your device. For more information, see Download the router configuration file (p. 69).

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

- create-private-virtual-interface (AWS CLI)
- CreatePrivateVirtualInterface (AWS Direct Connect API)

Create a transit virtual interface to the Direct Connect gateway

To connect your AWS Direct Connect connection to the transit gateway, you must create a transit interface for your connection. Specify the Direct Connect gateway to which to connect.

The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The MTU of a virtual private interface can be either 1500 or 9001 (jumbo frames). The MTU of a transit virtual interface can be either 1500 or 8500 (jumbo frames). You can specify the MTU when you create the interface or update it after you create it. Setting the MTU of a virtual interface to 8500 (jumbo frames) or 9001 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo Frame Capable** on the **Summary** tab.

> **Important**
> If you associate your transit gateway with one or more Direct Connect gateways, the Autonomous System Number (ASN) used by the transit gateway and the Direct Connect gateway must be different. For example, if you use the default ASN 64512 for both the transit gateway and the Direct Connect gateway, the association request fails.

To provision a transit virtual interface to a Direct Connect gateway

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Transit**.

5. Under **Transit virtual interface settings**, do the following:

   a. For **Virtual interface name**, enter a name for the virtual interface.
   b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
   c. For **Virtual interface owner**, choose **My AWS account** if the virtual interface is for your AWS account.
   d. For **Direct Connect gateway**, select the Direct Connect gateway.
   e. For **VLAN**, enter the ID number for your virtual local area network (VLAN).
   f. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

      The valid values are 1-2147483647.
6. Under **Additional Settings**, do the following:

   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

      [IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:
      - To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
      - For **Amazon router peer ip**, enter the IPv4 CIDR address to use to send traffic to AWS.

      [IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.
   b. To change the maximum transmission unit (MTU) from 1500 (default) to 8500 (jumbo frames), select **Jumbo MTU (MTU size 8500)**.
   c. (Optional) Add or remove a tag.

      [Add a tag] Choose **Add tag** and do the following:
      - For **Key**, enter the key name.
      - For **Value**, enter the key value.

      [Remove a tag] Next to the tag, choose **Remove tag**.
7. Choose **Create virtual interface**.

After you create the virtual interface, you can download the router configuration for your device. For more information, see Download the router configuration file (p. 69).

To create a transit virtual interface using the command line or API

- create-transit-virtual-interface (AWS CLI)
- CreateTransitVirtualInterface (AWS Direct Connect API)

To view the virtual interfaces that are attached to a Direct Connect gateway using the command line or API

- describe-direct-connect-gateway-attachments (AWS CLI)
- DescribeDirectConnectGatewayAttachments (AWS Direct Connect API)

Download the router configuration file

After you create the virtual interface and the interface state is up, you can download the router configuration file for your router.

If you use any of the following routers for virtual interfaces that have MACsec turned on, we automatically create the configuration file for your router:

- Cisco Nexus 9K+ Series switches running NX-OS 9.3 or later software
- Juniper Networks M/MX Series Routers running JunOS 9.5 or later software

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Select the virtual interface and then choose **View details**.
4. Choose **Download router configuration**.
5. For **Download router configuration**, do the following:
   a. For **Vendor**, select the manufacturer of your router.
   b. For **Platform**, select the model of your router.
   c. For **Software**, select the software version for your router.
6. Choose **Download**, and then use the appropriate configuration for your router to ensure that you can connect to AWS Direct Connect.

MACsec considerations

If you need to manually configure your router for MACsec, use the following table as a guideline.

Parameter	Description
CKN length	This is a 64 hexadecimal character (0–9, A–E) string. Use the full length to maximize cross-platform compatibility.
CAK length	This is a 64 hexadecimal character (0–9, A–E) string. Use the full length to maximize cross-platform compatibility.
Cryptographic algorithm	AES_256_CMAC
SAK Cipher Suite	GCM_AES_XPN_256
Key Cipher Suite	16
Confidentiality Offset	0
ICV Indicator	No
SAK Rekey Time	PN Rollover>
Replay Window (frames)	148809600

View virtual interface details

You can view the current status of your virtual interface. Details include:

- Connection state

- Name
- Location
- VLAN
- BGP details
- Peer IP addresses

To view details about a virtual interface

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the left pane, choose **Virtual Interfaces**.
3. Select the virtual interface and then choose **View details**.

To describe virtual interfaces using the command line or API

- describe-virtual-interfaces (AWS CLI)
- DescribeVirtualInterfaces (AWS Direct Connect API)

Add or delete a BGP peer

Add or delete an IPv4 or IPv6 BGP peering session to your virtual interface.

A virtual interface can support a single IPv4 BGP peering session and a single IPv6 BGP peering session.

You cannot specify your own peer IPv6 addresses for an IPv6 BGP peering session. Amazon automatically allocates you a /125 IPv6 CIDR.

Multi-protocol BGP is not supported. IPv4 and IPv6 operate in dual-stack mode for the virtual interface.

AWS enables MD5 by default. You cannot modify this option.

Add a BGP peer

Use the following procedure to add a BGP peer.

To add a BGP peer

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Select the virtual interface and then choose **View details**.
4. Choose **Add peering**.
5. (Private virtual interface) To add IPv4 BGP peers, do the following:
   - Choose **IPv4**.
   - To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic. For **Amazon router peer ip**, enter the IPv4 CIDR address to use to send traffic to AWS.
6. (Public virtual interface) To add IPv4 BGP peers, do the following:
   - For **Your router peer ip**, enter the IPv4 CIDR destination address where traffic should be sent.

- For **Amazon router peer IP**, enter the IPv4 CIDR address to use to send traffic to AWS.

7. (Private or public virtual interface) To add IPv6 BGP peers, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses; you cannot specify custom IPv6 addresses.
8. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

   For a public virtual interface, the ASN must be private or already on the allow list for the virtual interface.

   The valid values are 1-2147483647.

   Note that if you do not enter a value, we automatically assign one.
9. To provide your own BGP key, for **BGP Authentication Key**, enter your BGP MD5 key.
10. Choose **Add peering**.

To create a BGP peer using the command line or API

- create-bgp-peer (AWS CLI)
- CreateBGPPeer (AWS Direct Connect API)

Delete a BGP peer

If your virtual interface has both an IPv4 and IPv6 BGP peering session, you can delete one of the BGP peering sessions (but not both).

To delete a BGP peer

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Select the virtual interface and then choose **View details**.
4. Under **Peerings,** select the peering that you want to delete and then choose **Delete**.
5. In the **Remove peering from virtual interface** dialog box, choose **Delete**.

To delete a BGP peer using the command line or API

- delete-bgp-peer (AWS CLI)
- DeleteBGPPeer (AWS Direct Connect API)

Set network MTU for private virtual interfaces or transit virtual interfaces

AWS Direct Connect supports an Ethernet frame size of 1522 or 9023 bytes (14 bytes Ethernet header + 4 bytes VLAN tag + bytes for the IP datagram + 4 bytes FCS) at the link layer.

The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The MTU of a virtual private interface can be either 1500 or 9001 (jumbo frames). The MTU of a transit virtual interface can be either 1500 or 8500

(jumbo frames). You can specify the MTU when you create the interface or update it after you create it. Setting the MTU of a virtual interface to 8500 (jumbo frames) or 9001 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. To check whether a connection or virtual interface supports jumbo frames, select it in the AWS Direct Connect console and find **Jumbo Frame Capable** on the **Summary** tab.

After you enable jumbo frames for your private virtual interface, you can only associate it with a connection or LAG that is jumbo frame capable. Jumbo frames are supported on virtual private interfaces attached to a virtual private gateway or a Direct Connect gateway. Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to your virtual private gateway, traffic that is routed through the static route defaults to 1500 MTU. If you have two private virtual interfaces that advertise the same route but use different MTU values, 1500 MTU is used.

> **Important**
> Jumbo frames apply only to propagated routes from AWS Direct Connect. If you add static routes to a route table that point to your virtual private gateway, then traffic routed through the static routes is sent using 1500 MTU.
> If an EC2 instance doesn't support jumbo frames, it drops jumbo frames from AWS Direct Connect. All EC2 instance types support jumbo frames except for C1, CC1, T1, and M1. For more information, see Network Maximum Transmission Unit (MTU) for Your EC2 Instance in the *Amazon EC2 User Guide for Linux Instances*.

To set the MTU of a private virtual interface

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Select the virtual interface and then choose **Edit**.
4. Under **Jumbo MTU (MTU size 9001)** or **Jumbo MTU (MTU size 8500)**, select **Enabled**.
5. Under **Acknowledge**, select **I understand the selected connection(s) will go down for a brief period**. The state of the virtual interface is `pending` until the update is complete.

To set the MTU of a private virtual interface using the command line or API

- update-virtual-interface-attributes (AWS CLI)
- UpdateVirtualInterfaceAttributes (AWS Direct Connect API)

Add or remove virtual interface tags

Tags provide a way to identify the virtual interface. You can add or remove a tag if you are the account owner for the virtual interface.

To add or remove a virtual interface tag

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Select the virtual interface and then choose **Edit**.
4. Add or remove a tag.

   [Add a tag] Choose **Add tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

[Remove a tag] Next to the tag, choose **Remove tag**.

5. Choose **Edit virtual interface**.

To add a tag or remove a tag using the command line

- tag-resource (AWS CLI)
- untag-resource (AWS CLI)

Delete virtual interfaces

Delete one or more virtual interfaces. Before you can delete a connection, you must delete its virtual interface. Deleting a virtual interface stops AWS Direct Connect data transfer charges associated with the virtual interface.

To delete a virtual interface

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the left pane, choose **Virtual Interfaces**.
3. Select the virtual interfaces and then choose **Delete**.
4. In the **Delete** confirmation dialog box, choose **Delete**.

To delete a virtual interface using the command line or API

- delete-virtual-interface (AWS CLI)
- DeleteVirtualInterface (AWS Direct Connect API)

Create a hosted virtual interface

You can create a public, transit, or private hosted virtual interface. Before you begin, ensure that you have read the information in Prerequisites for virtual interfaces (p. 63).

Create a hosted private virtual interface

To create a hosted private virtual interface

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Private**.
5. Under **Private virtual interface settings**, do the following:

   a. For **Virtual interface name**, enter a name for the virtual interface.
   b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
   c. For **Virtual interface owner**, choose **Another AWS account**, and then for **Virtual interface owner**, enter the ID of the account to own this virtual interface.
   d. For **VLAN**, enter the ID number for your virtual local area network (VLAN).

e. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

The valid values are 1-2147483647.

6. Under **Additional Settings**, do the following:

a. To configure an IPv4 BGP or an IPv6 peer, do the following:

[IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:

- To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
- For **Amazon router peer ip**, enter the IPv4 CIDR address to use to send traffic to AWS.

[IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.

b. To change the maximum transmission unit (MTU) from 1500 (default) to 9001 (jumbo frames), select **Jumbo MTU (MTU size 9001)**.

c. (Optional) Add or remove a tag.

[Add a tag] Choose **Add tag** and do the following:

- For **Key**, enter the key name.
- For **Value**, enter the key value.

[Remove a tag] Next to the tag, choose **Remove tag**.

7. After the hosted virtual interface is accepted by the owner of the other AWS account, you can download the router configuration file (p. 69).

To create a hosted private virtual interface using the command line or API

- allocate-private-virtual-interface (AWS CLI)
- AllocatePrivateVirtualInterface (AWS Direct Connect API)

Create a hosted public virtual interface

To create a hosted public virtual interface

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Public**.
5. Under **Public Virtual Interface Settings**, do the following:

a. For **Virtual interface name**, enter a name for the virtual interface.

b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.

c. For **Virtual interface owner**, choose **Another AWS account**, and then for **Virtual interface owner**, enter the ID of the account to own this virtual interface.

d. For **VLAN**, enter the ID number for your virtual local area network (VLAN).

e. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

The valid values are 1-2147483647.

6. To configure an IPv4 BGP or an IPv6 peer, do the following:

[IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:

- To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
- For **Amazon router peer ip**, enter the IPv4 CIDR address to use to send traffic to AWS.

[IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.

7. To advertise prefixes to Amazon, for **Prefixes you want to advertise**, enter the IPv4 CIDR destination addresses (separated by commas) to which traffic should be routed over the virtual interface.
8. To provide your own key to authenticate the BGP session, under **Additional Settings**, for **BGP authentication key**, enter the key.

   If you do not enter a value, then we generate a BGP key.
9. (Optional) Add or remove a tag.

   [Add a tag] Choose **Add tag** and do the following:
   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Next to the tag, choose **Remove tag**.
10. Choose **Create virtual interface**.
11. After the hosted virtual interface is accepted by the owner of the other AWS account, you can download the router configuration file (p. 69).

To create a hosted public virtual interface using the command line or API

- allocate-public-virtual-interface (AWS CLI)
- AllocatePublicVirtualInterface (AWS Direct Connect API)

Create a hosted transit virtual interface

To create a hosted transit virtual interface

Important
If you associate your transit gateway with one or more Direct Connect gateways, the Autonomous System Number (ASN) used by the transit gateway and the Direct Connect gateway must be different. For example, if you use the default ASN 64512 for both the transit gateway and the Direct Connect gateway, the association request fails.

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Transit**.
5. Under **Transit virtual interface settings**, do the following:

   a. For **Virtual interface name**, enter a name for the virtual interface.
   b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
   c. For **Virtual interface owner**, choose **Another AWS account**, and then for **Virtual interface owner**, enter the ID of the account to own this virtual interface.
   d. For **VLAN**, enter the ID number for your virtual local area network (VLAN).

e. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

   The valid values are 1-2147483647.

6. Under **Additional Settings**, do the following:

   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

      [IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:

      - To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
      - For **Amazon router peer ip**, enter the IPv4 CIDR address to use to send traffic to AWS.

      [IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.

   b. To change the maximum transmission unit (MTU) from 1500 (default) to 8500 (jumbo frames), select **Jumbo MTU (MTU size 8500)**.

   c. [Optional] Add a tag. Do the following:

      [Add a tag] Choose **Add tag** and do the following:

      - For **Key**, enter the key name.
      - For **Value**, enter the key value.

      [Remove a tag] Next to the tag, choose **Remove tag**.

7. Choose **Create virtual interface**.
8. After the hosted virtual interface is accepted by the owner of the other AWS account, you can download the router configuration file (p. 69).

To create a hosted transit virtual interface using the command line or API

- allocate-transit-virtual-interface (AWS CLI)
- AllocateTransitVirtualInterface (AWS Direct Connect API)

Accept a hosted virtual interface

Before you can begin using a hosted virtual interface, you must accept the virtual interface. For a private virtual interface, you must also have an existing virtual private gateway or Direct Connect gateway. For a transit virtual interface, you must have an existing transit gateway or Direct Connect gateway.

To accept a hosted virtual interface

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Select the virtual interface and then choose **View details**.
4. Choose **Accept**.
5. This applies to private virtual interfaces and transit virtual interfaces.

   (Transit virtual interface) In the **Accept virtual interface** dialog box, select a Direct Connect gateway, and then choose **Accept virtual interface**.

   (Private virtual interface) In the **Accept virtual interface** dialog box, select a virtual private gateway or Direct Connect gateway, and then choose **Accept virtual interface**.

6. After you accept the hosted virtual interface, the owner of the AWS Direct Connect connection can download the router configuration file. The **Download router configuration** option is not available for the account that accepts the hosted virtual interface.

To accept a hosted private virtual interface using the command line or API

- confirm-private-virtual-interface (AWS CLI)
- ConfirmPrivateVirtualInterface (AWS Direct Connect API)

To accept a hosted public virtual interface using the command line or API

- confirm-public-virtual-interface (AWS CLI)
- ConfirmPublicVirtualInterface (AWS Direct Connect API)

To accept a hosted transit virtual interface using the command line or API

- confirm-transit-virtual-interface (AWS CLI)
- ConfirmTransitVirtualInterface (AWS Direct Connect API)

Migrate a virtual interface

Use this procedure when you want to perform any of the following virtual interface migration operations:

- Migrate an existing virtual interface associated with a connection to another LAG.
- Migrate an existing virtual interface associated with an existing LAG to a new LAG.
- Migrate an existing virtual interface associated with a connection to another connection.

> **Note**
> You can migrate a virtual interface to a new connection within the same Region, but you can't migrate it from one Region to another. When you migrate or associate an existing virtual interface to a new connection, the configuration parameters that are associated with the virtual interfaces are the same. You can pre-stage the configuration on the connection, and then update the BGP configuration.
>
> **Important**
> The virtual interface will go down for a brief period. We recommend you perform this procedure during a maintenance window.

To migrate a virtual interface

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Select the virtual interface, and then choose **Edit**.
4. For **Connection**, select the LAG or connection.
5. Choose **Edit virtual interface**.

To migrate a virtual interface using the command line or API

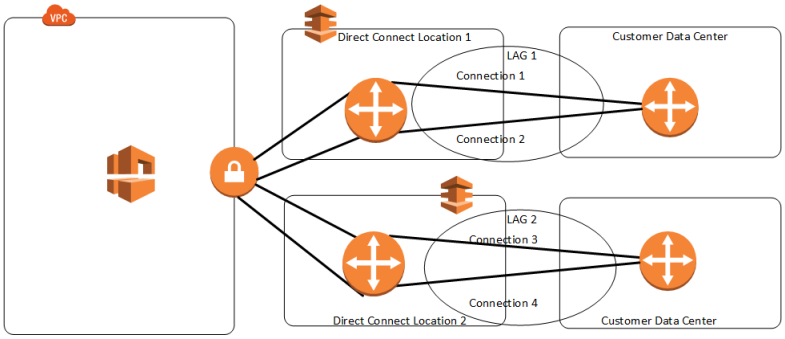
- associate-virtual-interface (AWS CLI)

- AssociateVirtualInterface (AWS Direct Connect API)

Link aggregation groups

You can use multiple connections for redundancy. A link aggregation group (LAG) is a logical interface that uses the Link Aggregation Control Protocol (LACP) to aggregate multiple connections at a single AWS Direct Connect endpoint, allowing you to treat them as a single, managed connection. LAGs streamline configuration because the LAG configuration applies to all connections in the group.

In the following diagram, you have four connections, with two connections to each location. You can create a LAG for the connections that terminate in the same location, and then use the two LAGs instead of the four connections for configuration and management.



You can create a LAG from existing connections, or you can provision new connections. After you've created the LAG, you can associate existing connections (whether standalone or part of another LAG) with the LAG.

The following rules apply:

- All connections must be dedicated connections and have a port speed of 1 Gbps, 10 Gbps, or 100 Gbps.
- All connections in the LAG must use the same bandwidth.
- You can have a maximum of two 100G connections, or four connections with a port speed less than 100G in a LAG. Each connection in the LAG counts towards your overall connection limit for the Region.
- All connections in the LAG must terminate at the same AWS Direct Connect endpoint.

When you create a LAG, you can download the Letter of Authorization and Connecting Facility Assignment (LOA-CFA) for each new physical connection individually from the AWS Direct Connect console. For more information, see Download the LOA-CFA (p. 47).

All LAGs have an attribute that determines the minimum number of connections in the LAG that must be operational for the LAG itself to be operational. By default, new LAGs have this attribute set to 0. You can update your LAG to specify a different value—doing so means that your entire LAG becomes non-operational if the number of operational connections falls below this threshold. This attribute can be used to prevent over-utilization of the remaining connections.

All connections in a LAG operate in Active/Active mode.

> **Note**
> When you create a LAG or associate more connections with the LAG, we may not be able to guarantee enough available ports on a given AWS Direct Connect endpoint.

MACsec considerations

Take the following into consideration when you want to configure MACsec on LAGs:

- When you create a LAG from existing connections, we disassociate all of the MACsec keys from the connections. Then we add the connections to the LAG, and associate the LAG MACsec key with the connections.
- When you associate an existing connection to a LAG, the MACsec keys that are currently associated with the LAG are associated with the connection. Therefore, we disassociate the MACsec keys from the connection, add the connection to the LAG, and then associate the LAG MACsec key with the connection.

Create a LAG

You can create a LAG by provisioning new connections, or aggregating existing connections.

You cannot create a LAG with new connections if this results in you exceeding the overall connections limit for the Region.

To create a LAG from existing connections, the connections must be on the same AWS device (terminate at the same AWS Direct Connect endpoint). They must also use the same bandwidth. You cannot migrate a connection from an existing LAG if removing the connection causes the original LAG to fall below its setting for the minimum number of operational connections.

Important
For existing connections, connectivity to AWS is interrupted during the creation of the LAG.

Create a LAG with new connections using the console

To create a LAG with new connections

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **LAGs**.
3. Choose **Create LAG**.
4. Under **Lag creation type**, choose **Request new connections**, and provide the following information:

   - **LAG name**: A name for the LAG.
   - **Location**: The location for the LAG.
   - **Port speed**: The port speed for the connections.
   - **Number of new connections**: The number of new connections to create.
   - (Optional) Configure MAC security (MACsec) for the connection. Under **Additional Settings**, select **Request a MACsec capable port**.

     MACsec is only available on dedicated connections.
   - (Optional) Add or remove a tag.

     [Add a tag] Choose **Add tag** and do the following:
     - For **Key**, enter the key name.
     - For **Value**, enter the key value.

     [Remove a tag] Next to the tag, choose **Remove tag**.

5. Choose **Create LAG**.

Create a LAG with existing connections using the console

To create a LAG from existing connections

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **LAGs**.
3. Choose **Create LAG**.
4. Under **Lag creation type**, choose **Use existing connections**, and provide the following information:
   - **LAG name**: A name for the LAG.
   - **Connection**: The Direct Connect connection to use for the LAG.
   - **Minimum links**: The minimum number of connections that must be operational for the LAG itself to be operational. If you do not specify a value, we assign a default value of 0.
5. (Optional) Add or remove a tag.

   [Add a tag] Choose **Add tag** and do the following:
   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Next to the tag, choose **Remove tag**.
6. Choose **Create LAG**.

Command line

To create a LAG using the command line or API

- create-lag (AWS CLI)
- CreateLag (AWS Direct Connect API)

To describe your LAGs using the command line or API

- describe-lags (AWS CLI)
- DescribeLags (AWS Direct Connect API)

To download the LOA-CFA using the command line or API

- describe-loa (AWS CLI)
- DescribeLoa (AWS Direct Connect API)

After you create a LAG, you can associate or disassociate connections from it. For more information, see Associate a connection with a LAG (p. 84) and Disassociate a connection from a LAG (p. 85).

View your LAG details

After you create a LAG, you can view its details.

Console

To view information about your LAG

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **LAGs**.
3. Select the LAG and choose **View details**.
4. You can view information about the LAG, including its ID, and the AWS Direct Connect endpoint on which the connections terminate.

Command line

To view information about your LAG using the command line or API

- describe-lags (AWS CLI)
- DescribeLags (AWS Direct Connect API)

Update a LAG

You can update the following link aggregation group (LAG) attributes:

- The name of the LAG.
- The value for the minimum number of connections that must be operational for the LAG itself to be operational.
- The LAG's MACsec encryption mode.

  MACsec is only available on dedicated connections.

  AWS assigns this value to each connection that is part of the LAG.

  The valid values are:

  - `should_encrypt`
  - `must_encrypt`

    When you set the encryption mode to this value, the connections go down when the encryption is down.
  - `no_encrypt`
- The tags.

Note
If you adjust the threshold value for the minimum number of operational connections, ensure that the new value does not cause the LAG to fall below the threshold and become non-operational.

Console

To update a LAG

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **LAGs**.

3. Select the LAG, and then choose **Edit**.
4. Modify the LAG

   [Change the name] For **LAG Name**, enter a new LAG name.

   [Adjust the minimum number of connections] For **Minimum Links**, enter minimum number of operational connections.

   [Add a tag] Choose **Add tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Next to the tag, choose **Remove tag**.
5. Choose **Edit LAG**.

Command line

To update a LAG using the command line or API

- update-lag (AWS CLI)
- UpdateLag (AWS Direct Connect API)

To add a tag or remove a tag using the command line

- tag-resource (AWS CLI)
- untag-resource (AWS CLI)

Associate a connection with a LAG

You can associate an existing connection with a LAG. The connection can be standalone, or it can be part of another LAG. The connection must be on the same AWS device and must use the same bandwidth as the LAG. If the connection is already associated with another LAG, you cannot re-associate it if removing the connection causes the original LAG to fall below its threshold for the minimum number of operational connections.

Associating a connection to a LAG automatically re-associates its virtual interfaces to the LAG.

Important
Connectivity to AWS over the connection is interrupted during association.

Console

To associate a connection with a LAG

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **LAGs**.
3. Select the LAG, and then choose **View details**.
4. Under **Connections**, choose **Associate connection**.
5. For **Connection**, choose the Direct Connect connection to use for the LAG.
6. Choose **Associate Connection**.

Command line

To associate a connection using the command line or API

- associate-connection-with-lag (AWS CLI)
- AssociateConnectionWithLag (AWS Direct Connect API)

Disassociate a connection from a LAG

Convert a connection to standalone by disassociating it from a LAG. You can't disassociate a connection if it causes the LAG to fall below its threshold for the minimum number of operational connections.

Disassociating a connection from a LAG does not automatically disassociate any virtual interfaces.

Important
Your connection to AWS is broken off during disassociation.

Console

To disassociate a connection from a LAG

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the left pane, choose **LAGs**.
3. Select the LAG, and then choose **View details**.
4. Under **Connections**, select the connection from the list of available connections and choose **Disassociate**.
5. In the confirmation dialog box, choose **Disassociate**.

Command line

To disassociate a connection using the command line or API

- disassociate-connection-from-lag (AWS CLI)
- DisassociateConnectionFromLag (AWS Direct Connect API)

Associate a MACsec CKN/CAK with a LAG

After you create the LAG that supports MACsec, you can associate a CKN/CAK with the connection.

Note
You cannot modify a MACsec secret key after you associate it with a LAG. If you need to modify the key, disassociate the key from the connection, and then associate a new key with the connection. For information about removing an association, see the section called “Remove the association between a MACsec secret key and a LAG” (p. 86).

Console

To associate a MACsec key with a LAG

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **LAGs**.

3. Select the LAG and choose **View details**.
4. Choose **Associate key**.
5. Enter the MACsec key.

   [Use the CAK/CKN pair] Choose **Key Pair**, and then do the following:

   - For **Connectivity Association Key (CAK)**, enter the CAK.
   - For **Connectivity Association Key Name (CKN)**, enter the CKN.

   [Use the secret] Choose **Existing Secret Manager secret**, and then for **Secret**, select the MACsec secret key.
6. Choose **Associate key**.

Command line

To associate a MACsec key with a LAG

- associate-mac-sec-key (AWS CLI)
- AssociateMacSecKey (AWS Direct Connect API)

Remove the association between a MACsec secret key and a LAG

You can remove the association between the LAG and the MACsec key.

Console

To remove an association between a LAG and a MACsec key

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **LAGs**.
3. Select the LAG and choose **View details**.
4. Select the MACsec secret to remove, and then choose **Disassociate key**.
5. In the confirmation dialog box, enter **disassociate**, and then choose **Disassociate**.

Command line

To remove an association between a LAG and a MACsec key

- associate-mac-sec-key (AWS CLI)
- DisassociateMacSecKey (AWS Direct Connect API)

Delete LAGs

If you no longer need LAGs, you can delete them. You cannot delete a LAG if it has virtual interfaces associated with it. You must first delete the virtual interfaces, or associate them with a different LAG or connection. Deleting a LAG does not delete the connections in the LAG; you must delete the connections yourself. For more information, see Delete connections (p. 51).

Console

To delete a LAG

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **LAGs**.
3. Select the LAGs, and then choose **Delete**.
4. In the confirmation dialog box, choose **Delete**.

Command line

To delete a LAG using the command line or API

- delete-lag (AWS CLI)
- DeleteLag (AWS Direct Connect API)

Working with Direct Connect gateways

You can work with AWS Direct Connect gateways using the Amazon VPC console or the AWS CLI.

Contents

Direct Connect gateways

Use *AWS Direct Connect gateway* to connect your VPCs. You associate an *AWS Direct Connect gateway* with either of the following gateways:

- A transit gateway when you have multiple VPCs in the same Region
- A virtual private gateway

  You can also use a virtual private gateway to extend your Local Zone. This configuration allows the VPC associated with the Local Zone to connect to a Direct Connect gateway. The Direct Connect gateway connects to an AWS Direct Connect location in a Region. The on-premises data center has an AWS Direct Connect connection to the AWS Direct Connect location. For more information, see Accessing Local Zones using a Direct Connect gateway in the *Amazon VPC User Guide*.

A Direct Connect gateway is a globally available resource. You can create the Direct Connect gateway in any Region and access it from all other Regions. You can use a Direct Connect gateway in the following scenarios.

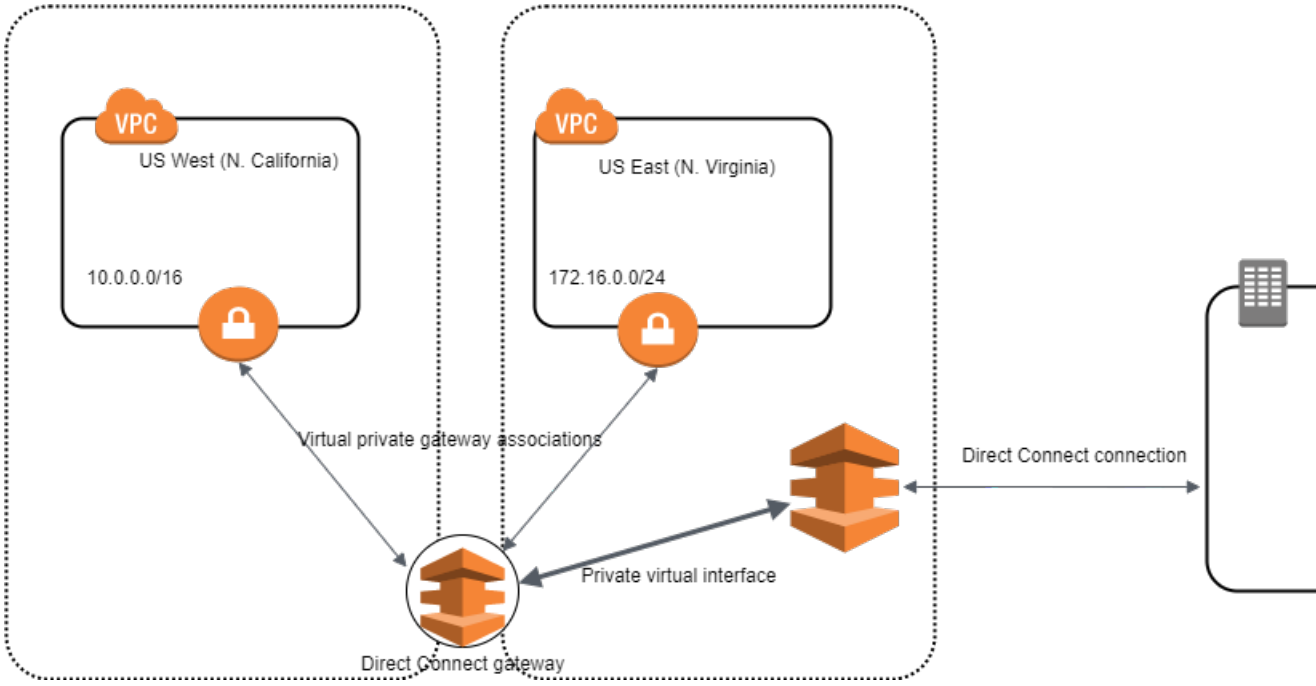
A Direct Connect gateway does not allow gateway associations that are on the same Direct Connect gateway to send traffic to each other (for example, a virtual private gateway to another virtual private gateway). A Direct Connect gateway does not prevent traffic from being sent from one gateway association back to the gateway association itself (for example when you have an on-premises supernet route that contains the prefixes from the gateway association). If you have a configuration with multiple VPCs connected to the same transit gateway, the VPCs could communicate. To prevent the VPCs from communicating, use separate transit gateway attachments, and then associate a route table with the attachments that have the **blackhole** option set.

Virtual private gateway associations

In the following diagram, the Direct Connect gateway enables you to use your AWS Direct Connect connection in the US East (N. Virginia) Region to access VPCs in your account in both the US East (N. Virginia) and US West (N. California) Regions.

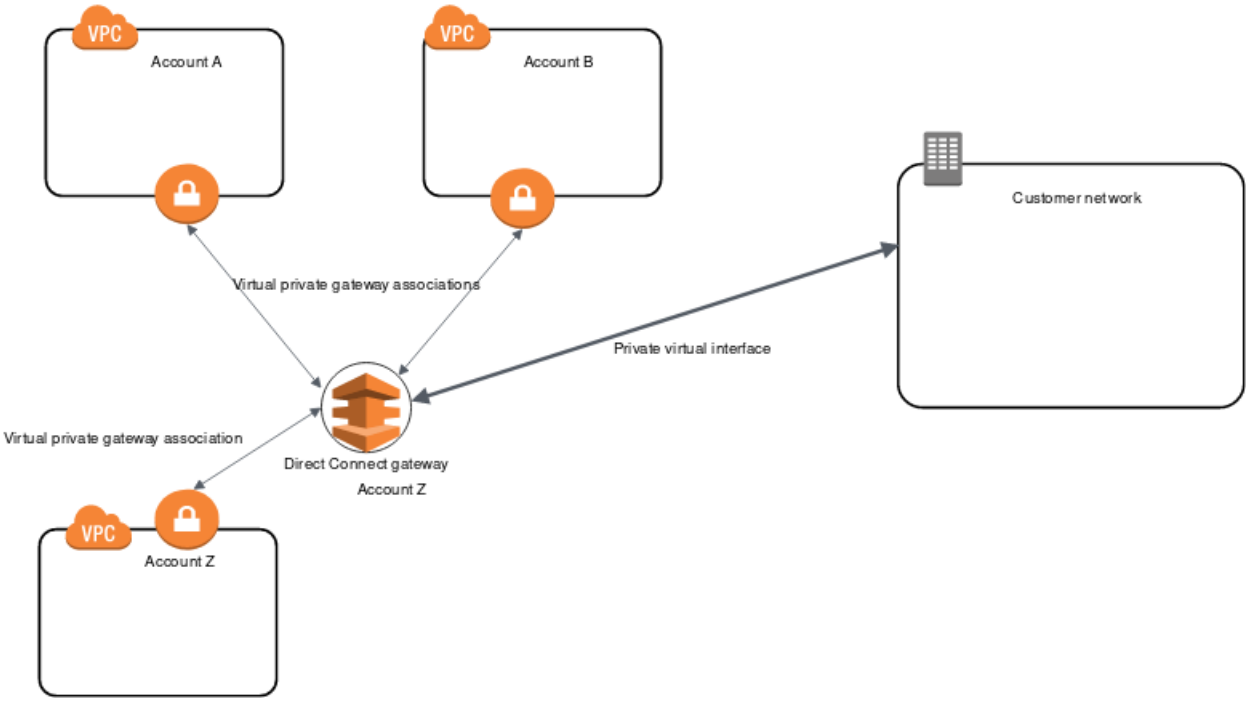
Each VPC has a virtual private gateway that connects to the Direct Connect gateway using a virtual private gateway association. The Direct Connect gateway uses a private virtual interface for the

connection to the AWS Direct Connect location. There is an AWS Direct Connect connection from the location to the customer data center.



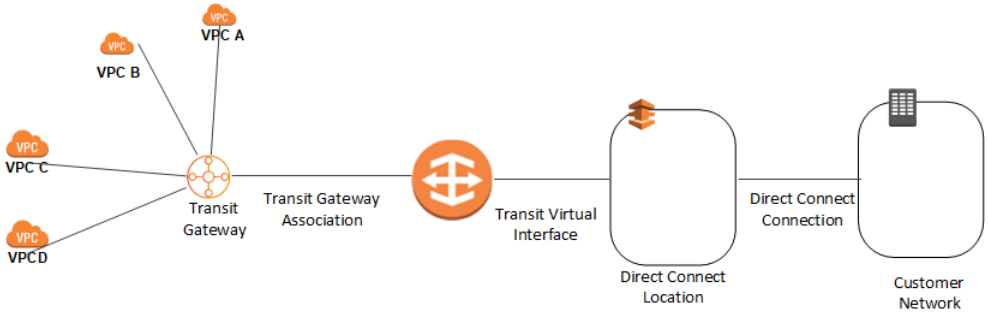
Virtual private gateway associations across accounts

Consider this scenario of a Direct Connect gateway owner (Account Z) who owns the Direct Connect gateway. Account A and Account B want to use the Direct Connect gateway. Account A and Account B each send an association proposal to Account Z. Account Z accepts the association proposals and can optionally update the prefixes that are allowed from Account A's virtual private gateway or Account B's virtual private gateway. After Account Z accepts the proposals, Account A and Account B can route traffic from their virtual private gateway to the Direct Connect gateway. Account Z also owns the routing to the customers because Account Z owns the gateway.



Transit gateway associations

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



The solution involves the following components:

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

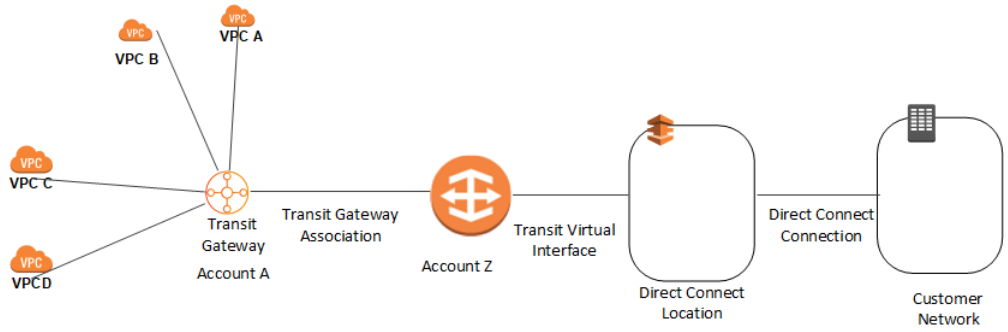
This configuration offers the following benefits. You can:

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

For information about configuring transit gateways, see Working with Transit Gateways in the *Amazon VPC Transit Gateways Guide*.

Transit gateway associations across accounts

Consider this scenario of a Direct Connect gateway owner (Account Z) who owns the Direct Connect gateway. Account A owns the transit gateway and wants to use the Direct Connect gateway. Account Z accepts the association proposals and can optionally update the prefixes that are allowed from Account A's transit gateway. After Account Z accepts the proposals, the VPCs attached to the transit gateway can route traffic from the transit gateway to the Direct Connect gateway. Account Z also owns the routing to the customers because Account Z owns the gateway.



Contents

Creating a Direct Connect gateway

You can create a Direct Connect gateway in any supported Region.

To create a Direct Connect gateway

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Direct Connect Gateways**.
3. Choose **Create Direct Connect gateway**.
4. Specify the following information, and choose **Create Direct Connect gateway**.
   - **Name**: Enter a name to help you identify the Direct Connect gateway.
   - **Amazon side ASN**: Specify the ASN for the Amazon side of the BGP session. The ASN must be in the 64,512 to 65,534 range or 4,200,000,000 to 4,294,967,294 range.
   - **Virtual private gateway**: To associate a virtual private gateway, choose the virtual private gateway.

To create a Direct Connect gateway using the command line or API

- create-direct-connect-gateway (AWS CLI)
- CreateDirectConnectGateway (AWS Direct Connect API)

Deleting Direct Connect gateways

If you no longer require a Direct Connect gateway, you can delete it. You must first disassociate all associated virtual private gateways and delete the attached private virtual interface.

To delete a Direct Connect gateway

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Direct Connect Gateways**.
3. Select the gateways and choose **Delete**.

To delete a Direct Connect gateway using the command line or API

- delete-direct-connect-gateway (AWS CLI)
- DeleteDirectConnectGateway (AWS Direct Connect API)

Migrating from a virtual private gateway to a Direct Connect gateway

If you had a virtual private gateway attached to a virtual interface, and you want to migrate to a Direct Connect gateway, perform the following steps:

To migrate to a Direct Connect gateway

1. Create a Direct Connect gateway. For more information, see the section called “Creating a Direct Connect gateway” (p. 91).
2. Create a virtual interface for the Direct Connect gateway. For more information, see the section called “Create a virtual interface” (p. 65).
3. Associate the virtual private gateway with the Direct Connect gateway. For more information, see the section called “Associating and disassociating virtual private gateways” (p. 94).
4. Delete the virtual interface that was associated with the virtual private gateway. For more information, see the section called “Delete virtual interfaces” (p. 74).

Virtual private gateway associations

You can use an *AWS Direct Connect gateway* to connect your AWS Direct Connect connection over a private virtual interface to one or more VPCs in any account that are located in the same or different Regions. You associate a Direct Connect gateway with the virtual private gateway for the VPC. Then, you create a private virtual interface for your AWS Direct Connect connection to the Direct Connect gateway. You can attach multiple private virtual interfaces to your Direct Connect gateway.

The following rules apply to virtual private gateway associations:

- There are limits for creating and using Direct Connect gateways. For more information, see *Quotas* (p. 146).
- The VPCs to which you connect through a Direct Connect gateway cannot have overlapping CIDR blocks. If you add an IPv4 CIDR block to a VPC that's associated with a Direct Connect gateway, ensure that the CIDR block does not overlap with an existing CIDR block for any other associated VPC. For more information, see Adding IPv4 CIDR Blocks to a VPC in the *Amazon VPC User Guide*.
- You cannot create a public virtual interface to a Direct Connect gateway.
- A Direct Connect gateway supports communication between attached private virtual interfaces and associated virtual private gateways only. The following traffic flows are not supported:
  - Direct communication between the VPCs that are associated with a single Direct Connect gateway. This includes traffic from one VPC to another by using a hairpin through an on-premises network through a single Direct Connect gateway.

  - Direct communication between the virtual interfaces that are attached to a single Direct Connect gateway.
  - Direct communication between the virtual interfaces that are attached to a single Direct Connect gateway and a VPN connection on a virtual private gateway that's associated with the same Direct Connect gateway.
- You cannot associate a virtual private gateway with more than one Direct Connect gateway and you cannot attach a private virtual interface to more than one Direct Connect gateway.
- A virtual private gateway that you associate with a Direct Connect gateway must be attached to a VPC.
- A virtual private gateway association proposal expires 7 days after it is created.
- An accepted virtual private gateway proposal, or a deleted virtual private gateway proposal remains visible for 3 days.
- A virtual private gateway can be associated with a Direct Connect gateway and also attached to a virtual interface.

To connect your AWS Direct Connect connection to a VPC in the same Region only, you can create a Direct Connect gateway. Or, you can create a private virtual interface and attach it to the virtual private gateway for the VPC. For more information, see Create a private virtual interface (p. 67) and VPN CloudHub.

To use your AWS Direct Connect connection with a VPC in another account, you can create a hosted private virtual interface for that account. When the owner of the other account accepts the hosted virtual interface, they can choose to attach it either to a virtual private gateway or to a Direct Connect gateway in their account. For more information, see AWS Direct Connect virtual interfaces (p. 61).

Contents

Creating a virtual private gateway

The virtual private gateway must be attached to the VPC to which you want to connect.

> **Note**
> If you are planning to use the virtual private gateway for a Direct Connect gateway and a dynamic VPN connection, set the ASN on the virtual private gateway to the value that you require for the VPN connection. Otherwise, the ASN on the virtual private gateway can be set to any permitted value. The Direct Connect gateway advertises all connected VPCs over the ASN assigned to it.

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

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

1. In the navigation pane, choose **Virtual Private Gateways**, **Create Virtual Private Gateway**.
2. (Optional) Enter a name for your virtual private gateway. Doing so creates a tag with a key of `Name` and the value that you specify.
3. For **ASN**, leave the default selection to use the default Amazon ASN. Otherwise, choose **Custom ASN** and enter a value. For a 16-bit ASN, the value must be in the 64512 to 65534 range. For a 32-bit ASN, the value must be in the 4200000000 to 4294967294 range.

4. Choose **Create Virtual Private Gateway**.
5. Select the virtual private gateway that you created, and then choose **Actions**, **Attach to VPC**.
6. Select your VPC from the list and choose **Yes, Attach**.

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

- CreateVpnGateway (Amazon EC2 Query API)
- create-vpn-gateway (AWS CLI)
- New-EC2VpnGateway (AWS Tools for Windows PowerShell)

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

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

Associating and disassociating virtual private gateways

You can associate or disassociate a virtual private gateway and Direct Connect gateway. The account owner of the virtual private gateway performs these operations.

To associate a virtual private gateway

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Direct Connect Gateways** and then select the Direct Connect gateway.
3. Choose **View details**.
4. Choose **Gateways associations** and then choose **Associate gateway**.
5. For **Gateways**, choose the virtual private gateways to associate, and then choose **Associate gateway**.

You can view all of the virtual private gateways that are associated with the Direct Connect gateway by choosing **Gateway associations**.

To disassociate a virtual private gateway

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Direct Connect Gateways** and then select the Direct Connect gateway.
3. Choose **View details**.
4. Choose **Gateway associations** and then select the virtual private gateway.
5. Choose **Disassociate**.

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

- create-direct-connect-gateway-association (AWS CLI)
- CreateDirectConnectGatewayAssociation (AWS Direct Connect API)

To view the virtual private gateways associated with a Direct Connect gateway using the command line or API

- describe-direct-connect-gateway-associations (AWS CLI)
- DescribeDirectConnectGatewayAssociations (AWS Direct Connect API)

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

- delete-direct-connect-gateway-association (AWS CLI)
- DeleteDirectConnectGatewayAssociation (AWS Direct Connect API)

Creating a private virtual interface to the Direct Connect gateway

To connect your AWS Direct Connect connection to the remote VPC, you must create a private virtual interface for your connection. Specify the Direct Connect gateway to which to connect.

> **Note**
> If you're accepting a hosted private virtual interface, you can associate it with a Direct Connect gateway in your account. For more information, see Accept a hosted virtual interface (p. 77).

To provision a private virtual interface to a Direct Connect gateway

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, choose **Private**.
5. Under **Private virtual interface settings**, do the following:

   a. For **Virtual interface name**, enter a name for the virtual interface.
   b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
   c. For **Virtual interface owner**, choose **My AWS account** if the virtual interface is for your AWS account.
   d. For **Direct Connect gateway**, select the Direct Connect gateway.
   e. For **VLAN**, enter the ID number for your virtual local area network (VLAN).
   f. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

      The valid values are 1-2147483647.
6. Under **Additional Settings**, do the following:

   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

      [IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:

      - To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
      - For **Amazon router peer ip**, enter the IPv4 CIDR address to use to send traffic to AWS.

      [IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.
   b. To change the maximum transmission unit (MTU) from 1500 (default) to 9001 (jumbo frames), select **Jumbo MTU (MTU size 9001)**.

c. (Optional) Add or remove a tag.

[Add a tag] Choose **Add tag** and do the following:

- For **Key**, enter the key name.
- For **Value**, enter the key value.

[Remove a tag] Next to the tag, choose **Remove tag**.

7. Choose **Create virtual interface**.

After you've created the virtual interface, you can download the router configuration for your device. For more information, see Download the router configuration file (p. 69).

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

- create-private-virtual-interface (AWS CLI)
- CreatePrivateVirtualInterface (AWS Direct Connect API)

To view the virtual interfaces that are attached to a Direct Connect gateway using the command line or API

- describe-direct-connect-gateway-attachments (AWS CLI)
- DescribeDirectConnectGatewayAttachments (AWS Direct Connect API)

Associating a virtual private gateway across accounts

You can associate a Direct Connect gateway with a virtual private gateway that is owned by any AWS account. The Direct Connect gateway can be an existing gateway, or you can create a new gateway. The owner of the virtual private gateway creates an *association proposal* and the owner of the Direct Connect gateway must accept the association proposal.

An association proposal can contain prefixes that will be allowed from the virtual private gateway. The owner of the Direct Connect gateway can optionally override any requested prefixes in the association proposal.

Allowed prefixes

When you associate a virtual private gateway with a Direct Connect gateway, you specify a list of Amazon VPC prefixes to advertise to the Direct Connect gateway. The prefix list acts as a filter that allows the same CIDRs, or smaller CIDRs to be advertised to the Direct Connect gateway. You must set the **Allowed prefixes** to a range that is the same or wider than the VPC CIDR because we provision entire VPC CIDR on the virtual private gateway.

Consider the case where the VPC CIDR is 10.0.0.0/16. You can set the **Allowed prefixes** to 10.0.0.0/16 (the VPC CIDR value), or 10.0.0.0/15 (a value that is wider than the VPC CIDR).

For more information on how allowed prefixes interact with virtual private gateways and transit gateways, see the section called “Allowed prefixes interactions” (p. 104).

Tasks

- Creating an association proposal (p. 97)
- Accepting or rejecting an association proposal (p. 97)
- Updating the allowed prefixes for an association (p. 98)

- Deleting an association proposal (p. 98)

Creating an association proposal

If you own the virtual private gateway, you must create an association proposal. The virtual private gateway must be attached to a VPC in your AWS account. The owner of the Direct Connect gateway must share the ID of the Direct Connect gateway and the ID of its AWS account. After you create the proposal, the owner of the Direct Connect gateway must accept it in order for you to gain access to the on-premises network over AWS Direct Connect.

To create an association proposal

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual private gateways** and select the virtual private gateway.
3. Choose **View details**.
4. Choose **Direct Connect gateway associations** and choose **Associate Direct Connect gateway**.
5. Under **Association account type**, for **Account owner**, choose **Another account**.
6. For **Direct Connect gateway owner**, enter the id of the AWS account that owns the Direct Connect gateway.
7. Under **Association settings**, do the following:
   a. For **Direct Connect gateway ID**, enter the ID of the Direct Connect gateway.
   b. For **Direct Connect gateway owner**, enter the ID of the AWS account that owns the Direct Connect gateway for the association.
   c. (Optional) To specify a list of prefixes to be allowed from the virtual private gateway, add the prefixes to **Allowed prefixes**, separating them using commas, or entering them on separate lines.
8. Choose **Associate Direct Connect gateway**.

To create an association proposal using the command line or API

- create-direct-connect-gateway-association-proposal (AWS CLI)
- CreateDirectConnectGatewayAssociationProposal (AWS Direct Connect API)

Accepting or rejecting an association proposal

If you own the Direct Connect gateway, you must accept the association proposal in order to create the association. Otherwise, you can reject the association proposal.

To accept an association proposal

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Direct Connect gateways**.
3. Select the Direct Connect gateway with pending proposals and choose **View details**.
4. On the **Pending proposals** tab, select the proposal and choose **Accept proposal**.
5. ((Optional) To specify a list of prefixes to be allowed from the virtual private gateway, add the prefixes to **Allowed prefixes**, separating them using commas, or entering them on separate lines.
6. Choose **Accept proposal**.

To reject an association proposal

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.

2. In the navigation pane, choose **Direct Connect gateways**.
3. Select the Direct Connect gateway with pending proposals and choose **View details**.
4. On the **Pending proposals** tab, select the virtual private gateway and choose **Reject proposal**.
5. In the **Reject proposal** dialog box, enter Delete and choose **Reject proposal**.

To view association proposals using the command line or API

- describe-direct-connect-gateway-association-proposals (AWS CLI)
- DescribeDirectConnectGatewayAssociationProposals (AWS Direct Connect API)

To accept an association proposal using the command line or API

- accept-direct-connect-gateway-association-proposal (AWS CLI)
- AcceptDirectConnectGatewayAssociationProposal (AWS Direct Connect API)

To reject an association proposal using the command line or API

- delete-direct-connect-gateway-association-proposal (AWS CLI)
- DeleteDirectConnectGatewayAssociationProposal (AWS Direct Connect API)

Updating the allowed prefixes for an association

You can update the prefixes that are allowed from the virtual private gateway over the Direct Connect gateway.

If you're the owner of the virtual private gateway, create a new association proposal (p. 97) for the same Direct Connect gateway and virtual private gateway, specifying the prefixes to allow.

If you're the owner of the Direct Connect gateway, update the allowed prefixes when you accept the association proposal (p. 97) or update the allowed prefixes for an existing association as follows.

To update the allowed prefixes for an existing association using the command line or API

- update-direct-connect-gateway-association (AWS CLI)
- UpdateDirectConnectGatewayAssociation (AWS Direct Connect API)

Deleting an association proposal

The owner of the virtual private gateway can delete the Direct Connect gateway association proposal if it is still pending acceptance. After an association proposal is accepted, you can't delete it, but you can disassociate the virtual private gateway from the Direct Connect gateway. For more information, see the section called “Associating and disassociating virtual private gateways” (p. 94).

To delete an association proposal

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual private gateways** and select the virtual private gateway.
3. Choose **View details**.
4. Choose **Pending Direct Connect gateway associations**, select the association and choose **Delete association**.

5. In the **Delete association proposal** dialog box, enter Delete and choose **Delete**.

To delete a pending association proposal using the command line or API

- delete-direct-connect-gateway-association-proposal (AWS CLI)
- DeleteDirectConnectGatewayAssociationProposal (AWS Direct Connect API)

Transit gateway associations

You can use an *AWS Direct Connect gateway* to connect your AWS Direct Connect connection over a transit virtual interface to the VPCs or VPNs that are attached to your transit gateway. You associate a Direct Connect gateway with the transit gateway. Then, create a transit virtual interface for your AWS Direct Connect connection to the Direct Connect gateway.

The following rules apply to transit gateway associations:

- You cannot attach a Direct Connect gateway to a transit gateway when the Direct Connect gateway is already associated with a virtual private gateway or is attached to a private virtual interface.
- There are limits for creating and using Direct Connect gateways. For more information, see *Quotas* (p. 146).
- A Direct Connect gateway supports communication between attached transit virtual interfaces and associated transit gateways only.
- If you connect to multiple transit gateways that are in different Regions, use unique ASNs for each transit gateway.

Associating and disassociating transit gateways

To associate a transit gateway

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Direct Connect Gateways** and then select the Direct Connect gateway.
3. Choose **View details**.
4. Choose **Gateway associations** and then choose **Associate gateway**.
5. For **Gateways**, choose the transit gateway to associate.
6. In **Allowed prefixes**, enter the prefixes (separated by a comma, or on a new line) which the Direct Connect gateway advertises to the on-premises data center.
7. Choose **Associate gateway**

You can view all of the gateways that are associated with the Direct Connect gateway by choosing **Gateway associations**.

To disassociate a transit gateway

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Direct Connect gateways** and then select the Direct Connect gateway.
3. Choose **View details**.
4. Choose **Gateway associations** and then select the transit gateway.

5. Choose **Disassociate**.

To associate a transit gateway using the command line or API

- create-direct-connect-gateway-association (AWS CLI)
- CreateDirectConnectGatewayAssociation (AWS Direct Connect API)

To view the transit gateways associated with a Direct Connect gateway using the command line or API

- describe-direct-connect-gateway-associations (AWS CLI)
- DescribeDirectConnectGatewayAssociations (AWS Direct Connect API)

To disassociate a transit gateway using the command line or API

- delete-direct-connect-gateway-association (AWS CLI)
- DeleteDirectConnectGatewayAssociation (AWS Direct Connect API)

Creating a transit virtual interface to the Direct Connect gateway

To connect your AWS Direct Connect connection to the transit gateway, you must create a transit interface for your connection. Specify the Direct Connect gateway to which to connect.

> **Important**
> If you associate your transit gateway with one or more Direct Connect gateways, the Autonomous System Number (ASN) used by the transit gateway and the Direct Connect gateway must be different. For example, if you use the default ASN 64512 for both the transit gateway and the Direct Connect gateway, the association request fails.

To provision a transit virtual interface to a Direct Connect gateway

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Virtual Interfaces**.
3. Choose **Create virtual interface**.
4. Under **Virtual interface type**, for **Type**, choose **Transit**.
5. Under **Transit virtual interface settings**, do the following:

   a. For **Virtual interface name**, enter a name for the virtual interface.
   b. For **Connection**, choose the Direct Connect connection that you want to use for this interface.
   c. For **Virtual interface owner**, choose **My AWS account** if the virtual interface is for your AWS account.
   d. For **Direct Connect gateway**, select the Direct Connect gateway.
   e. For **VLAN**, enter the ID number for your virtual local area network (VLAN).
   f. For **BGP ASN**, enter the Border Gateway Protocol Autonomous System Number of your on-premises peer router for the new virtual interface.

      The valid values are 1-2147483647.
6. Under **Additional Settings**, do the following:

   a. To configure an IPv4 BGP or an IPv6 peer, do the following:

[IPv4] To configure an IPv4 BGP peer, choose **IPv4** and do one of the following:

- To specify these IP addresses yourself, for **Your router peer ip**, enter the destination IPv4 CIDR address to which Amazon should send traffic.
- For **Amazon router peer ip**, enter the IPv4 CIDR address to use to send traffic to AWS.

[IPv6] To configure an IPv6 BGP peer, choose **IPv6**. The peer IPv6 addresses are automatically assigned from Amazon's pool of IPv6 addresses. You cannot specify custom IPv6 addresses.

b. To change the maximum transmission unit (MTU) from 1500 (default) to 8500 (jumbo frames), select **Jumbo MTU (MTU size 8500)**.

c. (Optional) Add or remove a tag.

[Add a tag] Choose **Add tag** and do the following:

- For **Key**, enter the key name.
- For **Value**, enter the key value.

[Remove a tag] Next to the tag, choose **Remove tag**.

7. Choose **Create virtual interface**.

After you've created the virtual interface, you can download the router configuration for your device. For more information, see Download the router configuration file (p. 69).

To create a transit virtual interface using the command line or API

- create-transit-virtual-interface (AWS CLI)
- CreateTransitVirtualInterface (AWS Direct Connect API)

To view the virtual interfaces that are attached to a Direct Connect gateway using the command line or API

- describe-direct-connect-gateway-attachments (AWS CLI)
- DescribeDirectConnectGatewayAttachments (AWS Direct Connect API)

Associating a transit gateway across accounts

You can associate an existing Direct Connect gateway or a new Direct Connect gateway with a transit gateway that is owned by any AWS account. The owner of the transit gateway creates an *association proposal* and the owner of the Direct Connect gateway must accept the association proposal.

An association proposal can contain prefixes that will be allowed from the transit gateway. The owner of the Direct Connect gateway can optionally override any requested prefixes in the association proposal.

Allowed prefixes

For a transit gateway association, you provision the allowed prefixes list on the Direct Connect gateway. The list is used to route traffic from on-premises to AWS into the transit gateway even if the VPCs attached to the transit gateway do not have assigned CIDRs. Prefixes in the Direct Connect gateway allowed prefix list originate on the Direct Connect gateway and are advertised to the on-premises network. For more information on how allowed prefixes interact with transit gateways and virtual private gateways, see the section called “Allowed prefixes interactions” (p. 104).

Tasks

- Creating a transit gateway association proposal (p. 102)

Creating a transit gateway association proposal

If you own the transit gateway, you must create the association proposal. The transit gateway must be attached to a VPC or VPN in your AWS account. The owner of the Direct Connect gateway must share the ID of the Direct Connect gateway and the ID of its AWS account. After you create the proposal, the owner of the Direct Connect gateway must accept it in order for you to gain access to the on-premises network over AWS Direct Connect.

To create an association proposal

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Transit gateways** and then select the transit gateway.
3. Choose **View details**.
4. Choose **Direct Connect gateway associations** and then choose **Associate Direct Connect gateway**.
5. Under **Association account type**, for **Account owner**, choose **Another account**.
6. For **Direct Connect gateway owner**, enter the ID of the account that owns the Direct Connect gateway.
7. Under **Association settings**, do the following:
   a. For **Direct Connect gateway ID**, enter the ID of the Direct Connect gateway.
   b. For **Virtual interface owner**, enter the ID of the account that owns the virtual interface for the association.
   c. (Optional) To specify a list of prefixes to be allowed from the transit gateway, add the prefixes to **Allowed prefixes**, separating them using commas, or entering them on separate lines.
8. Choose **Associate Direct Connect gateway**.

To create an association proposal using the command line or API

- create-direct-connect-gateway-association-proposal (AWS CLI)
- CreateDirectConnectGatewayAssociationProposal (AWS Direct Connect API)

Accepting or rejecting a transit gateway association proposal

If you own the Direct Connect gateway, you must accept the association proposal in order to create the association. You also have the option of rejecting the association proposal.

To accept an association proposal

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Direct Connect gateways**.
3. Select the Direct Connect gateway with pending proposals and then choose **View details**.
4. On the **Pending proposals** tab, select the proposal and then choose **Accept proposal**.
5. ((Optional) To specify a list of prefixes to be allowed from the transit gateway, add the prefixes to **Allowed prefixes**, separating them using commas, or entering them on separate lines.
6. Choose **Accept proposal**.

To reject an association proposal

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Direct Connect gateways**.
3. Select the Direct Connect gateway with pending proposals and then choose **View details**.
4. On the **Pending proposals** tab, select the transit gateway and then choose **Reject proposal**.
5. In the **Reject proposal** dialog box, enter Delete and then choose **Reject proposal**.

To view association proposals using the command line or API

- describe-direct-connect-gateway-association-proposals (AWS CLI)
- DescribeDirectConnectGatewayAssociationProposals (AWS Direct Connect API)

To accept an association proposal using the command line or API

- accept-direct-connect-gateway-association-proposal (AWS CLI)
- AcceptDirectConnectGatewayAssociationProposal (AWS Direct Connect API)

To reject an association proposal using the command line or API

- delete-direct-connect-gateway-association-proposal (AWS CLI)
- DeleteDirectConnectGatewayAssociationProposal (AWS Direct Connect API)

Updating the allowed prefixes for a transit gateway association

You can update the prefixes that are allowed from the transit gateway over the Direct Connect gateway.

If you're the owner of the transit gateway, create a new association proposal (p. 102) for the same Direct Connect gateway and virtual private gateway, specifying the prefixes to allow.

If you're the owner of the Direct Connect gateway, update the allowed prefixes when you accept the association proposal (p. 102) or update the allowed prefixes for an existing association as follows.

To update the allowed prefixes for an existing association using the command line or API

- update-direct-connect-gateway-association (AWS CLI)
- UpdateDirectConnectGatewayAssociation (AWS Direct Connect API)

Deleting a transit gateway association proposal

The owner of the transit gateway can delete the Direct Connect gateway association proposal if it is still pending acceptance. After an association proposal is accepted, you can't delete it, but you can disassociate the transit gateway from the Direct Connect gateway. For more information, see the section called “Creating a transit gateway association proposal” (p. 102).

To delete an association proposal

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/v2/home.
2. In the navigation pane, choose **Transit gateways** and then select the transit gateway.
3. Choose **View details**.
4. Choose **Pending gateway associations**, select the association and then choose **Delete association**.

5. In the **Delete association proposal** dialog box, enter **Delete** and then choose **Delete**.

To delete a pending association proposal using the command line or API

- delete-direct-connect-gateway-association-proposal (AWS CLI)
- DeleteDirectConnectGatewayAssociationProposal (AWS Direct Connect API)

Allowed prefixes interactions

Learn how allowed prefixes interact with transit gateways and virtual private gateways. For more information, see the section called "Routing policies and BGP communities" (p. 3).

Virtual private gateway associations

The prefix list (IPv4 and IPv6) acts as a filter that allows the same CIDRs, or a smaller range of CIDRs to be advertised to the Direct Connect gateway. You must set the prefixes to a range that is the same or wider than the VPC CIDR block.

Consider the scenario where you have a VPC with CIDR 10.0.0.0/16 is attached to a virtual private gateway.

- When the allowed prefixes list is set to 22.0.0.0/24, you do not receive any route because 22.0.0.0/24 is not the same as, or wider than 10.0.0.0/16.
- When the allowed prefixes list is set to 10.0.0.0/24, you do not receive any route because 10.0.0.0/24 is not the same as 10.0.0.0/16.
- When the allowed prefixes list is set to 10.0.0.0/15, you do receive 10.0.0.0/16, because the IP address is wider than 10.0.0.0/16.

Transit gateway associations

For a transit gateway association, you provision the allowed prefixes list on the Direct Connect gateway. The list routes traffic from on-premises to AWS, to the transit gateway even when the VPCs attached to the transit gateway do not have assigned CIDRs. Prefixes in the Direct Connect gateway allowed prefix list originate on the Direct Connect gateway and are advertised to the on-premises network.

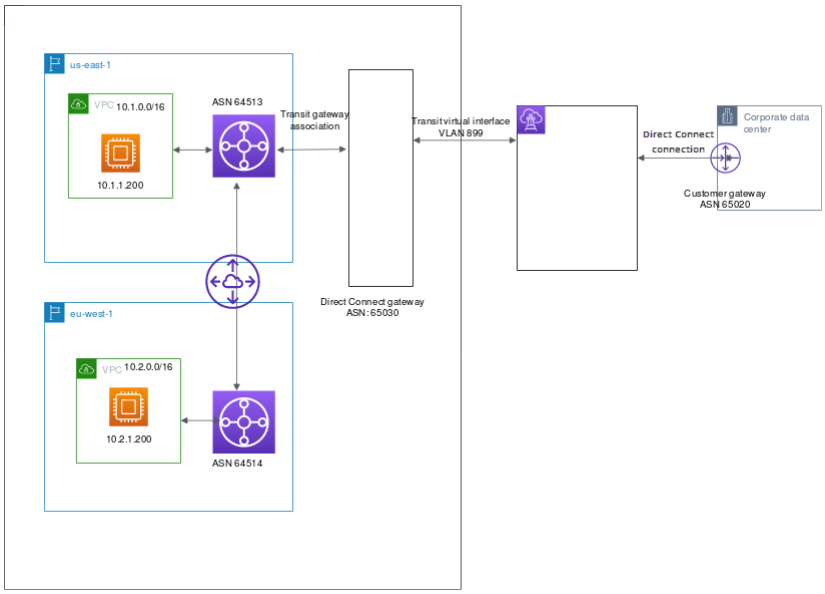
Consider the scenario where you have a VPC with CIDR 10.0.0.0/16 attached to a transit gateway.

- When the allowed prefixes list is set to 22.0.0.0/24, you receive 22.0.0.0/24 through BGP on your transit virtual interface. You do not receive 10.0.0.0/16 because we directly provision the prefixes that are in the allowed prefix list.
- When the allowed prefixes list is set to 10.0.0.0/24, you receive 10.0.0.0/24 through BGP on your transit virtual interface. You do not receive 10.0.0.0/16 because we directly provision the prefixes that are in the allowed prefix list.
- When the allowed prefixes list is set to 10.0.0.0/8, you receive 10.0.0.0/8 through BGP on your transit virtual interface.

Example: Allowed to prefixes in a transit gateway configuration

Consider the configuration where you have instance in two different AWS Regions which need to access the corporate data center. You can use the following resources for this configuration:

- A transit gateway in each Region.
- A transit gateway peering connection.
- A Direct connect gateway.
- A transit gateway association between one of the transit gateways (the one in us-east-1) to the Direct Connect gateway.
- A transit virtual interface from the on-premises location and the AWS Direct Connect location.



Configure the following options for the resources.

- Direct Connect gateway: Set the ASN for to 65030. For more information, see the section called “Creating a Direct Connect gateway” (p. 91).
- Transit virtual interface: Set the VLAN to 899, and the ASN to 65020. For more information, see the section called “Create a transit virtual interface to the Direct Connect gateway” (p. 68).
- Direct Connect gateway association with the transit gateway: Set the allowed to prefixes to 10.0.0.0/8.

  This CIDR block covers both VPC CIDR blocks. For more information, see the section called “Associating and disassociating transit gateways” (p. 99).
- VPC route: To route traffic from the 10.2.0.0 VPC, create a route in the VPC route table which has a Destination of 0.0.0.0/0 and the transit gateway ID as the Target. For more information about routing to a transit gateway, see Routing for a transit gateway in the *Amazon VPC User Guide*.

Tagging AWS Direct Connect resources

A tag is a label that a resource owner assigns to their AWS Direct Connect resources. Each tag consists of a key and an optional value, both of which you define. Tags enable the resource owner to categorize your AWS Direct Connect resources in different ways, for example, by purpose, or environment. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags you've assigned to it.

For example, you have two AWS Direct Connect connections in a Region, each in different locations. Connection `dxcon-11aa22bb` is a connection serving production traffic, and is associated with virtual interface `dxvif-33cc44dd`. Connection `dxcon-abcabcab` is a redundant (backup) connection, and is associated with virtual interface `dxvif-12312312`. You might choose to tag your connections and virtual interfaces as follows, to help distinguish them:

Resource ID	Tag key	Tag value
`dxcon-11aa22bb`	Purpose	Production
	Location	Amsterdam
`dxvif-33cc44dd`	Purpose	Production
`dxcon-abcabcab`	Purpose	Backup
	Location	Frankfurt
`dxvif-12312312`	Purpose	Backup

We recommend that you devise a set of tag keys that meets your needs for each resource type. Using a consistent set of tag keys makes it easier for you to manage your resources. Tags don't have any semantic meaning to AWS Direct Connect and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources. You can edit tag keys and values, and you can remove tags from a resource at any time. You can set the value of a tag to an empty string, but you can't set the value of a tag to null. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. If you delete a resource, any tags for the resource are also deleted.

You can tag the following AWS Direct Connect resources using the AWS Direct Connect console, the AWS Direct Connect API, the AWS CLI, the AWS Tools for Windows PowerShell, or an AWS SDK. When you use these tools to manage tags, you must specify the Amazon Resource Name (ARN) for the resource. For more information about ARNs, see Amazon Resource Names (ARNs) in the *Amazon Web Services General Reference*.

Resource	Supports tags	Supports tags on creation	Supports tags controlling access and resource allocation	Supports cost allocation
Connections	Yes	Yes	Yes	Yes

Resource	Supports tags	Supports tags on creation	Supports tags controlling access and resource allocation	Supports cost allocation
Virtual interfaces	Yes	Yes	Yes	No
Link aggregation groups (LAG)	Yes	Yes	Yes	Yes
Interconnects	Yes	Yes	Yes	Yes
Direct Connect gateways	No	No	No	No

Tag restrictions

The following rules and restrictions apply to tags:

- Maximum number of tags per resource: 50
- Maximum key length: 128 Unicode characters
- Maximum value length: 265 Unicode characters
- Tag keys and values are case-sensitive.
- The `aws:` prefix is reserved for AWS use. You can't edit or delete a tag's key or value when the tag has a tag key with the `aws:` prefix. Tags with a tag key with the `aws:` prefix do not count against your tags per resource limit.
- Allowed characters are letters, spaces, and numbers representable in UTF-8, plus the following special characters: + - = . _ : / @
- Only the resource owner can add or remove tags. For example, if there is a hosted connection, the partner will not be able to add, remove, or view the tags.
- Cost allocation tags are only supported for connections, interconnects, and LAGs. For information about how to use tags with cost management, see Using Cost Allocation Tags in the *AWS Billing and Cost Management User Guide*.

Working with tags using the CLI or API

Use the following to add, update, list, and delete the tags for your resources.

Task	API	CLI
Add or overwrite one or more tags.	TagResource	tag-resource
Delete one or more tags.	UntagResource	untag-resource
Describe one or more tags.	DescribeTags	describe-tags

Examples

Use the tag-resource command to tag the Connection `dxcon-11aa22bb`.

```
aws directconnect tag-resource --resource-arn arn:aws:directconnect:us-
east-1:123456789012:dxcon/dxcon-11aa22bb --tags "key=Purpose,value=Production"
```

Use the describe-tags command to describe the Connection `dxcon-11aa22bb` tags.

```
aws directconnect describe-tags --resource-arn arn:aws:directconnect:us-
east-1:123456789012:dxcon/dxcon-11aa22bb
```

Use the untag-resource command to remove a tag from Connection `dxcon-11aa22bb`.

```
aws directconnect untag-resource --resource-arn arn:aws:directconnect:us-
east-1:123456789012:dxcon/dxcon-11aa22bb --tag-keys Purpose
```

Security in AWS Direct Connect

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

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

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS Direct Connect, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

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

Topics

Data protection in AWS Direct Connect

The AWS shared responsibility model applies to data protection in AWS Direct Connect. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.

- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a **Name** field. This includes when you work with AWS Direct Connect or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

For more information about data protection, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

Topics

Internetwork traffic privacy in AWS Direct Connect

Traffic between service and on-premises clients and applications

You have two connectivity options between your private network and AWS:

- An association to an AWS Site-to-Site VPN. For more information, see the section called “Infrastructure security” (p. 128).
- An association to VPCs. For more information, see the section called “Virtual private gateway associations” (p. 92) and the section called “Transit gateway associations” (p. 99).

Traffic between AWS resources in the same Region

You have two connectivity options:

- An association to an AWS Site-to-Site VPN. For more information, see the section called “Infrastructure security” (p. 128).
- An association to VPCs. For more information, see the section called “Virtual private gateway associations” (p. 92) and the section called “Transit gateway associations” (p. 99).

Encryption in AWS Direct Connect

AWS Direct Connect does not encrypt your traffic that is in transit. To encrypt the data in transit that traverses AWS Direct Connect, you must use the transit encryption options for that service. To learn about EC2 instance traffic encryption, see Encryption in Transit in the Amazon EC2 User Guide for Linux Instances.

With AWS Direct Connect and AWS Site-to-Site VPN, you can combine one or more AWS Direct Connect dedicated network connections with the Amazon VPC VPN. This combination provides an IPsec-encrypted private connection that also reduces network costs, increases bandwidth throughput, and provides a more consistent network experience than internet-based VPN connections. For more information, see Amazon VPC-to-Amazon VPC Connectivity Options.

MAC Security (MACsec) is an IEEE standard that provides data confidentiality, data integrity, and data origin authenticity. You can use AWS Direct Connect connections that support MACsec to encrypt your data from your corporate data center to the AWS Direct Connect location. For more information, see *MAC Security* (p. 42).

Identity and access management for AWS Direct Connect

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

Topics

Audience

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

Service user – If you use the Direct Connect service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Direct Connect features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Direct Connect, see Troubleshooting AWS Direct Connect identity and access (p. 122).

Service administrator – If you're in charge of Direct Connect resources at your company, you probably have full access to Direct Connect. It's your job to determine which Direct Connect features and resources your employees should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Direct Connect, see How AWS Direct Connect works with IAM (p. 114).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Direct Connect. To view example Direct Connect identity-based policies that you can use in IAM, see AWS Direct Connect identity-based policy examples (p. 118).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see Signing in to the AWS Management Console as an IAM user or root user in the *IAM User Guide*.

You must be *authenticated* (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the AWS Management Console, use your password with your root user email address or your IAM user name. You can access AWS programmatically using your root user or IAM users access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. Do this using *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 signing process in the *AWS General Reference*.

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

AWS account root user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see Managing access keys for IAM users in the *IAM User Guide*. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

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

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

IAM roles

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

IAM roles with temporary credentials are useful in the following situations:

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

- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated users and roles in the *IAM User Guide*.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, Resources, and Condition Keys for AWS Direct Connect in the *Service Authorization Reference*.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. You can sign in as the root user or an IAM user, or you can assume an IAM role. When you then make a request, AWS evaluates the related identity-based or resource-based policies. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

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

Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a

group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

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

Identity-based policies

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

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

Resource-based policies

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

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

How AWS Direct Connect works with IAM

Before you use IAM to manage access to Direct Connect, you should understand what IAM features are available to use with Direct Connect. To get a high-level view of how Direct Connect and other AWS services work with IAM, see AWS Services That Work with IAM in the *IAM User Guide*.

Topics

Direct Connect identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. Direct Connect supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see IAM JSON Policy Elements Reference in the *IAM User Guide*.

Actions

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

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

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

Policy actions in Direct Connect use the following prefix before the action: `directconnect:`. For example, to grant someone permission to run an Amazon EC2 instance with the Amazon EC2 `DescribeVpnGateways` API operation, you include the `ec2:DescribeVpnGateways` action in their policy. Policy statements must include either an `Action` or `NotAction` element. Direct Connect defines its own set of actions that describe tasks that you can perform with this service.

The following example policy grants read access to AWS Direct Connect.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "directconnect:Describe*",
                "ec2:DescribeVpnGateways"
            ],
            "Resource": "*"
        }
    ]
}
```

The following example policy grants full access to AWS Direct Connect.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "directconnect:*",
                "ec2:DescribeVpnGateways"
            ],
            "Resource": "*"
        }
    ]
}
```

To see a list of Direct Connect actions, see Actions Defined by AWS Direct Connect in the *IAM User Guidelist_awsdirectconnect.html*.

Resources

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

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

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

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

Direct Connect uses the following ARNs:

Direct connect resource ARNs

Resource Type	ARN
dxcon	`arn:${Partition}:directconnect:${Region}:${Account}:dxcon/${ConnectionId}`
dxlag	`arn:${Partition}:directconnect:${Region}:${Account}:dxlag/${LagId}`
dx-vif	`arn:${Partition}:directconnect:${Region}:${Account}:dxvif/${VirtualInterfaceId}`
dx-gateway	`arn:${Partition}:directconnect::${Account}:dx-gateway/${DirectConnectGatewayId}`

For more information about the format of ARNs, see Amazon Resource Names (ARNs) and AWS Service Namespaces.

For example, to specify the `dxcon-11aa22bb` interface in your statement, use the following ARN:

```
"Resource": "arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-11aa22bb
```

To specify all virtual interfaces that belong to a specific account, use the wildcard (*):

```
"Resource": "arn:aws:directconnect:*:*:dxvif/*"
```

Some Direct Connect actions, such as those for creating resources, cannot be performed on a specific resource. In those cases, you must use the wildcard (*).

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

To see a list of Direct Connect resource types and their ARNs, see Resource Types Defined by AWS Direct Connect in the *IAM User Guide*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by AWS Direct Connect.

Condition keys

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

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

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

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

Direct Connect defines its own set of condition keys and also supports using some global condition keys. To see all AWS global condition keys, see AWS Global Condition Context Keys in the *IAM User Guide*.

You can use condition keys with the tag resource. For more information, see Example: Restricting Access to a Specific Region.

To see a list of Direct Connect condition keys, see Condition Keys for AWS Direct Connect in the *IAM User Guide*. To learn with which actions and resources you can use a condition key, see Actions Defined by AWS Direct Connect.

Examples

To view examples of Direct Connect identity-based policies, see the section called "Identity-based policy examples" (p. 118).

Direct Connect resource-based policies

You can control access to resources and requests by using tag key conditions. You can also use a condition in your IAM policy to control whether specific tag keys can be used on a resource or in a request.

For information about how to use tags with AWS Identity and Access Management policies, see Controlling Access Using Tags in the *IAM User Guide*.

Examples

To view examples of Direct Connect resource-based policies, see the section called "Resource-based policy examples" (p. 120).

Authorization based on Direct Connect tags

You can attach tags to Direct Connect resources or pass tags in a request to Direct Connect. To control access based on tags, you provide tag information in the condition element of a policy using the `directconnect:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys. For more information about tagging Direct Connect resources, see *Tagging resources* (p. 106)..

To view an example identity-based policy for limiting access to a resource based on the tags on that resource, see the section called "Associating Direct Connect virtual interfaces based on tags" (p. 120).

Direct Connect IAM roles

An IAM role is an entity within your AWS account that has specific permissions.

Using temporary credentials with Direct Connect

You can use temporary credentials to sign in with federation, assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as AssumeRole or GetFederationToken.

Direct Connect supports using temporary credentials.

Service-linked roles

Service-linked roles allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

Direct Connect does not support service-linked roles.

Service roles

This feature allows a service to assume a service role on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in your IAM account and are owned by the account. This means that an IAM administrator can change the permissions for this role. However, doing so might break the functionality of the service.

Direct Connect supports service roles.

AWS Direct Connect identity-based policy examples

By default, IAM users and roles don't have permission to create or modify Direct Connect resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

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

Policy best practices

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

- **Get started using AWS managed policies** – To start using Direct Connect quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see Get started using permissions with AWS managed policies in the *IAM User Guide*.
- **Grant least privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see Grant least privilege in the *IAM User Guide*.
- **Enable MFA for sensitive operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.
- **Use policy conditions for extra security** – To the extent that it's practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.

Using the Direct Connect console

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

To ensure that those entities can still use the Direct Connect console, also attach the following AWS managed policy to the entities. For more information, see Adding Permissions to a User in the *IAM User Guide*:

```
directconnect
```

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

Allow users to view their own permissions

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

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

Read-only access to AWS Direct Connect

The following example policy grants read access to AWS Direct Connect.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "directconnect:Describe*",
                "ec2:DescribeVpnGateways"
            ],
            "Resource": "*"
        }
    ]
}
```

Full access to AWS Direct Connect

The following example policy grants full access to AWS Direct Connect.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "directconnect:*",
                "ec2:DescribeVpnGateways"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS Direct Connect resource-based policy examples

You can control access to resources and requests by using tag key conditions. You can also use a condition in your IAM policy to control whether specific tag keys can be used on a resource or in a request.

For information about how to use tags with AWS Identity and Access Management policies, see Controlling Access Using Tags in the *IAM User Guide*.

Associating Direct Connect virtual interfaces based on tags

The following example shows how you might create a policy that allows associating a virtual interface only if the tag contains the environment key and the preprod or production values.

```
      {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": [
           "directconnect:AssociateVirtualInterface"
         ],
         "Resource": "arn:aws:directconnect:*:*:dxvif/*",
         "Condition": {
           "StringEquals": {
             "aws:ResourceTag/environment": [
               "preprod",
               "production"
```

```
            ]
          }
        }
      },
      {
        "Effect": "Allow",
        "Action": "directconnect:DescribeVirtualInterfaces",
        "Resource": "*"
      }
    ]
}
```

Controlling access to requests based on tags

You can use conditions in your IAM policies to control which tag key–value pairs can be passed in a request that tags an AWS resource. The following example shows how you might create a policy that allows using the AWS Direct Connect TagResource action to attach tags to a virtual interface only if the tag contains the environment key and the preprod or production values. As a best practice, use the `ForAllValues` modifier with the `aws:TagKeys` condition key to indicate that only the key environment is allowed in the request.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "directconnect:TagResource",
        "Resource": "arn:aws:directconnect:*:*:dxvif/*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/environment": [
                    "preprod",
                    "production"
                ]
            },
            "ForAllValues:StringEquals": {"aws:TagKeys": "environment"}
        }
    }
}
```

Controlling tag keys

You can use a condition in your IAM policies to control whether specific tag keys can be used on a resource or in a request.

The following example shows how you might create a policy that allows you to tag resources, but only with the tag key environment

```
{
 "Version": "2012-10-17",
 "Statement": {
   "Effect": "Allow",
   "Action": "directconnect:TagResource",
   "Resource": "*",
   "Condition": {
     "ForAllValues:StringEquals": {
       "aws:TagKeys": [
         "environment"
       ]
     }
   }
 }
```

```
    }
```

Troubleshooting AWS Direct Connect identity and access

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

Topics

I am not authorized to perform an action in Direct Connect

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password.

The following example error occurs when the `johndoe` IAM user tries to use the console to view details about a *`connection`* but does not have `directconnect:`*`DeleteConnection`* permissions.

```
User: arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-11aa22bb:user/johndoe is not
 authorized to perform: directconnect:DeleteConnection on resource: MyExampleConnection
```

In this case, John asks his administrator to update his policies to allow him to access the *`MyExampleConnection`* resource using the `directconnect:`*`DeleteConnection`* action.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to Direct Connect.

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

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

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

In this case, Mary asks her administrator to update her policies to allow her to perform the `iam:PassRole` action.

I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

Important
Do not provide your access keys to a third party, even to help find your canonical user ID. By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see Managing access keys in the *IAM User Guide*.

I am an administrator and want to allow others to access Direct Connect

To allow others to access Direct Connect, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in Direct Connect.

To get started right away, see Creating your first IAM delegated user and group in the *IAM User Guide*.

I want to allow people outside of my AWS account to access my Direct Connect resources

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

To learn more, consult the following:

- To learn whether Direct Connect supports these features, see How AWS Direct Connect works with IAM (p. 114).
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

Using service-linked roles for AWS Direct Connect

AWS Direct Connect uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to AWS Direct Connect. Service-linked roles are predefined by AWS Direct Connect and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up AWS Direct Connect easier because you don't have to manually add the necessary permissions. AWS Direct Connect defines the permissions of its service-linked roles, and unless defined otherwise, only AWS Direct Connect can assume its roles. The defined permissions

include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

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

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

Service-linked role permissions for AWS Direct Connect

AWS Direct Connect uses the service-linked role named **`AWSServiceRoleForDirectConnect`** – Allows AWS Direct Connect to retrieve the MACsec secrets that are stored in AWS Secrets Manager on your behalf.

The `AWSServiceRoleForDirectConnect` service-linked role trusts the following services to assume the role:

- `directconnect.amazon.com`

The **`AWSServiceRoleForDirectConnect`** service-linked role uses the managed policy AWSDirectConnectServiceRolePolicy (p. 126).

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For the **`AWSServiceRoleForDirectConnect`** service-linked role to be successfully created, the IAM identity that you use AWS Direct Connect with must have the required permissions. To grant the required permissions, attach the following policy to the IAM identity.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "directconnect.amazonaws.com"
                }
            }
        }
    ]
}
```

For more information, see Service-linked role permissions in the *IAM User Guide*.

Creating a service-linked role for AWS Direct Connect

You don't need to manually create a service-linked role. When you run the `associate-mac-sec-key command`, AWS creates a service linked role that allows AWS Direct Connect team to retrieve AWS Direct Connect to retrieve the MACsec secrets that are stored in AWS Secrets Manager on your behalf in the AWS Management Console, the AWS CLI, or the AWS API, AWS Direct Connect creates the service-linked role for you.

Important
This service-linked role can appear in your account if you completed an action in another service that uses the features supported by this role. To learn more, see A New Role Appeared in My IAM Account.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. AWS Direct Connect creates the service-linked role for you again.

You can also use the IAM console to create a service-linked role with the **AWS Direct Connect** use case. In the AWS CLI or the AWS API, create a service-linked role with the `directconnect.amazon.com` service name. For more information, see Creating a service-linked role in the *IAM User Guide*. If you delete this service-linked role, you can use this same process to create the role again.

Editing a service-linked role for AWS Direct Connect

AWS Direct Connect does not allow you to edit the `AWSServiceRoleForDirectConnect` service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a service-linked role in the *IAM User Guide*.

Deleting a service-linked role for AWS Direct Connect

You don't need to manually delete the `AWSServiceRoleForDirectConnect` role. When you When you delete your service linked role, you must delete all the associated resources that are stored in AWS Secrets Manager web service. in the AWS Management Console, the AWS CLI, or the AWS API, AWS Direct Connect cleans up the resources and deletes the service-linked role for you.

You can also use the IAM console, the AWS CLI or the AWS API to manually delete the service-linked role. To do this, you must first manually clean up the resources for your service-linked role and then you can manually delete it.

> **Note**
> If the AWS Direct Connect service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete AWS Direct Connect resources used by the `AWSServiceRoleForDirectConnect`

1. Remove the association between all MACsec keys and connections. For more information, see the section called “Remove the association between a MACsec secret key and a connection” (p. 50)
2. Remove the association between all MACsec keys and LAGs. For more information, see the section called “Remove the association between a MACsec secret key and a LAG” (p. 86)

To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the `AWSServiceRoleForDirectConnect` service-linked role. For more information, see Deleting a service-linked role in the *IAM User Guide*.

Supported regions for AWS Direct Connect service-linked roles

AWS Direct Connect supports using service-linked roles in all of the Regions where the MAC Security feature is available. For more information, see AWS Direct Connect Locations.

AWS managed policies for AWS Direct Connect

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to

support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ViewOnlyAccess** AWS managed policy provides read-only access to many AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the *IAM User Guide*.

AWS managed policy: AWSDirectConnectFullAccess

You can attach the `AWSDirectConnectFullAccess` policy to your IAM identities. This policy grants permissions that allow full access to AWS Direct Connect.

To view the permissions for this policy, see AWSDirectConnectFullAccess in the AWS Management Console.

AWS managed policy: AWSDirectConnectReadOnlyAccess

You can attach the `AWSDirectConnectReadOnlyAccess` policy to your IAM identities. This policy grants permissions that allow read-only access to AWS Direct Connect.

To view the permissions for this policy, see AWSDirectConnectReadOnlyAccess in the AWS Management Console.

AWS managed policy: AWSDirectConnectServiceRolePolicy

This policy is attached to the service-linked role named **AWSServiceRoleForDirectConnect** to allow AWS Direct Connect to retrieve MAC Security secrets on your behalf. For more information, see the section called “Using service-linked roles” (p. 123).

To view the permissions for this policy, see AWSDirectConnectServiceRolePolicy in the AWS Management Console.

AWS Direct Connect updates to AWS managed policies

View details about updates to AWS managed policies for AWS Direct Connect since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the AWS Direct Connect Document history page.

Change	Description	Date
AWSDirectConnectServiceRolePolicy (p. 126) - New policy	To support MAC Security, the **AWSServiceRoleForDirectConnect** service-linked role was added.	March 31, 2021
AWS Direct Connect started tracking changes	AWS Direct Connect started tracking changes to its AWS managed policies.	March 31, 2021

Logging and monitoring in AWS Direct Connect

You can use the following automated monitoring tools to watch AWS Direct Connect and report when something is wrong:

- **Amazon CloudWatch Alarms** – Watch a single metric over a time period that you specify. Perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon SNS topic. CloudWatch alarms do not invoke actions simply because they are in a particular state; the state must have changed and been maintained for a specified number of periods. For more information, see Monitoring with Amazon CloudWatch (p. 140).
- **AWS CloudTrail Log Monitoring** – Share log files between accounts and monitor CloudTrail log files in real time by sending them to CloudWatch Logs. You can also write log processing applications in Java and validate that your log files have not changed after delivery by CloudTrail. For more information, see Logging AWS Direct Connect API calls using AWS CloudTrail (p. 135) and Working with CloudTrail Log Files in the *AWS CloudTrail User Guide*.

For more information, see *Monitoring* (p. 139).

Compliance validation for AWS Direct Connect

Third-party auditors assess the security and compliance of AWS services as part of multiple AWS compliance programs, such as SOC, PCI, FedRAMP, and HIPAA.

To learn whether AWS Direct Connect or other AWS services are in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- Architecting for HIPAA Security and Compliance Whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.

  Note
  Not all services are compliant with HIPAA.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in AWS Direct Connect

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency,

high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

In addition to the AWS global infrastructure, AWS Direct Connect offers several features to help support your data resiliency and backup needs.

For information about how to use VPN with AWS Direct Connect, see AWS Direct Connect Plus VPN.

Failover

The AWS Direct Connect Resiliency Toolkit provides a connection wizard with multiple resiliency models that helps you order dedicated connections to achieve your SLA objective. You select a resiliency model, and then the AWS Direct Connect Resiliency Toolkit guides you through the dedicated connection ordering process. The resiliency models are designed to ensure that you have the appropriate number of dedicated connections in multiple locations.

- **Maximum Resiliency**: You can achieve maximum resiliency for critical workloads by using separate connections that terminate on separate devices in more than one location. This model provides resiliency against device, connectivity, and complete location failures.
- **High Resiliency**: You can achieve high resiliency for critical workloads by using two single connections to multiple locations. This model provides resiliency against connectivity failures caused by a fiber cut or a device failure. It also helps prevent a complete location failure.
- **Development and Test**: You can achieve development and test resiliency for non-critical workloads by using separate connections that terminate on separate devices in one location. This model provides resiliency against device failure, but does not provide resiliency against location failure.

For more information, see *Using the AWS Direct Connect Resiliency Toolkit to get started* (p. 8).

Logical redundancy

Logical redundancy allows you to create two IPv4 and IPv6 BGP peerings on two different AWS devices over a single connection. Logical redundancy can reduce downtime when a BGP peering session goes down due to a device failure or maintenance activity. This option is available on newly-created 1Gbps or 10Gbps dedicated connections in some locations. For information about the locations that support logical redundancy, see AWS Direct Connect FAQs.

Infrastructure security in AWS Direct Connect

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

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

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

You can call these API operations from any network location, but AWS Direct Connect supports resource-based access policies, which can include restrictions based on the source IP address. You can also use AWS Direct Connect policies to control access from specific Amazon Virtual Private Cloud (Amazon VPC) endpoints or specific VPCs. Effectively, this isolates network access to a given AWS Direct Connect resource from only the specific VPC within the AWS network. For example, see the section called "Resource-based policy examples" (p. 120).

Using the AWS CLI

You can use the AWS CLI to create and work with AWS Direct Connect resources.

The following example uses the AWS CLI commands to create an AWS Direct Connect connection. You can also download the Letter of Authorization and Connecting Facility Assignment (LOA-CFA) or provision a private or public virtual interface.

Before you begin, ensure that you have installed and configured the AWS CLI. For more information, see the AWS Command Line Interface User Guide.

Contents

Step 1: Create a connection

The first step is to submit a connection request. Ensure that you know the port speed that you require and the AWS Direct Connect location. For more information, see AWS Direct Connect connections (p. 45).

To create a connection request

1. Describe the AWS Direct Connect locations for your current Region. In the output that's returned, take note of the location code for the location in which you want to establish the connection.

```
aws directconnect describe-locations
```

```
{
    "locations": [
        {
            "locationName": "City 1, United States",
            "locationCode": "Example Location 1"
        },
        {
            "locationName": "City 2, United States",
            "locationCode": "Example location"
        }
    ]
}
```

2. Create the connection and specify a name, the port speed, and the location code. In the output that's returned, take note of the connection ID. You need the ID to get the LOA-CFA in the next step.

```
aws directconnect create-connection --location Example location --bandwidth 1Gbps --
connection-name "Connection to AWS"
```

```
{
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-EXAMPLE",
    "connectionState": "requested",
    "bandwidth": "1Gbps",
    "location": "Example location",
```

```
    "connectionName": "Connection to AWS",
    "region": "sa-east-1"
}
```

Step 2: Download the LOA-CFA

After you've requested a connection, you can get the LOA-CFA using the `describe-loa` command. The output is base64-encoded. You must extract the relevant LOA content, decode it, and create a PDF file.

To get the LOA-CFA using Linux or macOS

In this example, the final part of the command decodes the content using the base64 utility, and sends the output to a PDF file.

```
aws directconnect describe-loa --connection-id dxcon-fg31dyv6 --output text --query
 loaContent|base64 --decode > myLoaCfa.pdf
```

To get the LOA-CFA using Windows

In this example, the output is extracted to a file called myLoaCfa.base64. The second command uses the `certutil` utility to decode the file and send the output to a PDF file.

```
aws directconneawsct describe-loa --connection-id dxcon-fg31dyv6 --output text --query
 loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

After you've downloaded the LOA-CFA, send it to your network provider or colocation provider.

Step 3: Create a virtual interface and get the router configuration

After you have placed an order for an AWS Direct Connect connection, you must create a virtual interface to begin using it. You can create a private virtual interface to connect to your VPC. Or, you can create a public virtual interface to connect to AWS services that aren't in a VPC. You can create a virtual interface that supports IPv4 or IPv6 traffic.

Before you begin, ensure that you've read the prerequisites in Prerequisites for virtual interfaces (p. 63).

When you create a virtual interface using the AWS CLI, the output includes generic router configuration information. To create a router configuration that's specific to your device, use the AWS Direct Connect console. For more information, see Download the router configuration file (p. 69).

To create a private virtual interface

1. Get the ID of the virtual private gateway (vgw-*xxxxxxxx*) that's attached to your VPC. You need the ID to create the virtual interface in the next step.

   ```
   aws ec2 describe-vpn-gateways
   ```

   ```
   {
   ```

```
    "VpnGateways": [
        {
            "State": "available",
            "Tags": [
                {
                    "Value": "DX_VGW",
                    "Key": "Name"
                }
            ],
            "Type": "ipsec.1",
            "VpnGatewayId": "vgw-ebaa27db",
            "VpcAttachments": [
                {
                    "State": "attached",
                    "VpcId": "vpc-24f33d4d"
                }
            ]
        }
    ]
}
```

2. Create a private virtual interface. You must specify a name, a VLAN ID, and a BGP Autonomous System Number (ASN).

   For IPv4 traffic, you need private IPv4 addresses for each end of the BGP peering session. You can specify your own IPv4 addresses, or you can let Amazon generate the addresses for you. In the following example, the IPv4 addresses are generated for you.

```
aws directconnect create-private-virtual-interface --
connection-id dxcon-fg31dyv6 --new-private-virtual-interface
 virtualInterfaceName=PrivateVirtualInterface,vlan=101,asn=65000,virtualGatewayId=vgw-
ebaa27db,addressFamily=ipv4
```

```
{
    "virtualInterfaceState": "pending",
    "asn": 65000,
    "vlan": 101,
    "customerAddress": "192.168.1.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fg31dyv6",
    "addressFamily": "ipv4",
    "virtualGatewayId": "vgw-ebaa27db",
    "virtualInterfaceId": "dxvif-ffhhk74f",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [],
    "location": "Example location",
    "bgpPeers": [
        {
            "bgpStatus": "down",
            "customerAddress": "192.168.1.2/30",
            "addressFamily": "ipv4",
            "authKey": "asdf34example",
            "bgpPeerState": "pending",
            "amazonAddress": "192.168.1.1/30",
            "asn": 65000
        }
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-ffhhk74f\">\n  <vlan>101</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
 <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>\n
 <connection_type>private</connection_type>\n</logical_connection>\n",
    "amazonAddress": "192.168.1.1/30",
```

```
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "PrivateVirtualInterface"
}
```

To create a private virtual interface that supports IPv6 traffic, use the same command as above and specify `ipv6` for the `addressFamily` parameter. You cannot specify your own IPv6 addresses for the BGP peering session; Amazon allocates you IPv6 addresses.

3. To view the router configuration information in XML format, describe the virtual interface you created. Use the `--query` parameter to extract the `customerRouterConfig` information, and the `--output` parameter to organize the text into tab-delimited lines.

```
aws directconnect describe-virtual-interfaces --virtual-interface-id dxvif-ffhhk74f --
query virtualInterfaces[*].customerRouterConfig --output text
```

```
<?xml version="1.0" encoding="UTF-8"?>
<logical_connection id="dxvif-ffhhk74f">
  <vlan>101</vlan>
  <customer_address>192.168.1.2/30</customer_address>
  <amazon_address>192.168.1.1/30</amazon_address>
  <bgp_asn>65000</bgp_asn>
  <bgp_auth_key>asdf34example</bgp_auth_key>
  <amazon_bgp_asn>7224</amazon_bgp_asn>
  <connection_type>private</connection_type>
</logical_connection>
```

To create a public virtual interface

1. To create a public virtual interface, you must specify a name, a VLAN ID, and a BGP Autonomous System Number (ASN).

   For IPv4 traffic, you must also specify public IPv4 addresses for each end of the BGP peering session, and public IPv4 routes that you will advertise over BGP. The following example creates a public virtual interface for IPv4 traffic.

```
aws directconnect create-public-virtual-interface --
connection-id dxcon-fg31dyv6 --new-public-virtual-interface
 virtualInterfaceName=PublicVirtualInterface,vlan=2000,asn=65000,amazonAddress=203.0.113.1/30,custo
{cidr=203.0.113.4/30}]
```

```
{
    "virtualInterfaceState": "verifying",
    "asn": 65000,
    "vlan": 2000,
    "customerAddress": "203.0.113.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fg31dyv6",
    "addressFamily": "ipv4",
    "virtualGatewayId": "",
    "virtualInterfaceId": "dxvif-fgh0hcrk",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [
        {
            "cidr": "203.0.113.0/30"
        },
        {
            "cidr": "203.0.113.4/30"
        }
    ],
```

```
    "location": "Example location",
    "bgpPeers": [
        {
            "bgpStatus": "down",
            "customerAddress": "203.0.113.2/30",
            "addressFamily": "ipv4",
            "authKey": "asdf34example",
            "bgpPeerState": "verifying",
            "amazonAddress": "203.0.113.1/30",
            "asn": 65000
        }
    ],
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fgh0hcrk\">\n  <vlan>2000</
vlan>\n  <customer_address>203.0.113.2/30</customer_address>\n
 <amazon_address>203.0.113.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>\n
 <connection_type>public</connection_type>\n</logical_connection>\n",
    "amazonAddress": "203.0.113.1/30",
    "virtualInterfaceType": "public",
    "virtualInterfaceName": "PublicVirtualInterface"
}
```

To create a public virtual interface that supports IPv6 traffic, you can specify IPv6 routes that you will advertise over BGP. You cannot specify IPv6 addresses for the peering session; Amazon allocates IPv6 addresses to you. The following example creates a public virtual interface for IPv6 traffic.

```
aws directconnect create-public-virtual-interface --
connection-id dxcon-fg31dyv6 --new-public-virtual-interface
 virtualInterfaceName=PublicVirtualInterface,vlan=2000,asn=65000,addressFamily=ipv6,routeFilterPref
{cidr=2001:db8:64ce:ba01::/64}]
```

2. To view the router configuration information in XML format, describe the virtual interface you created. Use the `--query` parameter to extract the `customerRouterConfig` information, and the `--output` parameter to organize the text into tab-delimited lines.

```
aws directconnect describe-virtual-interfaces --virtual-interface-id dxvif-fgh0hcrk --
query virtualInterfaces[*].customerRouterConfig --output text
```

```
<?xml version="1.0" encoding="UTF-8"?>
<logical_connection id="dxvif-fgh0hcrk">
  <vlan>2000</vlan>
  <customer_address>203.0.113.2/30</customer_address>
  <amazon_address>203.0.113.1/30</amazon_address>
  <bgp_asn>65000</bgp_asn>
  <bgp_auth_key>asdf34example</bgp_auth_key>
  <amazon_bgp_asn>7224</amazon_bgp_asn>
  <connection_type>public</connection_type>
</logical_connection>
```

Logging AWS Direct Connect API calls using AWS CloudTrail

AWS Direct Connect is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Direct Connect. CloudTrail captures all API calls for AWS Direct Connect as events. The calls captured include calls from the AWS Direct Connect console and code calls to the AWS Direct Connect API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Direct Connect. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS Direct Connect, the IP address from which the request was made, who made the request, when it was made, and additional details.

For more information, see the AWS CloudTrail User Guide.

AWS Direct Connect information in CloudTrail

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

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

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

All AWS Direct Connect actions are logged by CloudTrail and are documented in the AWS Direct Connect API Reference. For example, calls to the `CreateConnection` and `CreatePrivateVirtualInterface` actions generate entries in the CloudTrail log files.

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

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

For more information, see the CloudTrail `userIdentity` Element.

Understanding AWS Direct Connect log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following are example CloudTrail log records for AWS Direct Connect.

Example Example: CreateConnection

```
{
    "Records": [
    {
        "eventVersion": "1.0",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/Alice",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice",
            "sessionContext": {
                "attributes": {
                    "mfaAuthenticated": "false",
                    "creationDate": "2014-04-04T12:23:05Z"
                }
            }
        },
        "eventTime": "2014-04-04T17:28:16Z",
        "eventSource": "directconnect.amazonaws.com",
        "eventName": "CreateConnection",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "Coral/Jakarta",
        "requestParameters": {
            "location": "EqSE2",
            "connectionName": "MyExampleConnection",
            "bandwidth": "1Gbps"
        },
        "responseElements": {
            "location": "EqSE2",
            "region": "us-west-2",
            "connectionState": "requested",
            "bandwidth": "1Gbps",
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-fhajolyy",
            "connectionName": "MyExampleConnection"
        }
    },
    ...
  ]
}
```

Example Example: CreatePrivateVirtualInterface

```
{
    "Records": [
    {
        "eventVersion": "1.0",
```

```
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/Alice",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice",
            "sessionContext": {
                "attributes": {
                    "mfaAuthenticated": "false",
                    "creationDate": "2014-04-04T12:23:05Z"
                }
            }
        },
        "eventTime": "2014-04-04T17:39:55Z",
        "eventSource": "directconnect.amazonaws.com",
        "eventName": "CreatePrivateVirtualInterface",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "Coral/Jakarta",
        "requestParameters": {
            "connectionId": "dxcon-fhajolyy",
            "newPrivateVirtualInterface": {
                "virtualInterfaceName": "MyVirtualInterface",
                "customerAddress": "[PROTECTED]",
                "authKey": "[PROTECTED]",
                "asn": -1,
                "virtualGatewayId": "vgw-bb09d4a5",
                "amazonAddress": "[PROTECTED]",
                "vlan": 123
            }
        },
        "responseElements": {
            "virtualInterfaceId": "dxvif-fgq61m6w",
            "authKey": "[PROTECTED]",
            "virtualGatewayId": "vgw-bb09d4a5",
            "customerRouterConfig": "[PROTECTED]",
            "virtualInterfaceType": "private",
            "asn": -1,
            "routeFilterPrefixes": [],
            "virtualInterfaceName": "MyVirtualInterface",
            "virtualInterfaceState": "pending",
            "customerAddress": "[PROTECTED]",
            "vlan": 123,
            "ownerAccount": "123456789012",
            "amazonAddress": "[PROTECTED]",
            "connectionId": "dxcon-fhajolyy",
            "location": "EqSE2"
        }
    },
    ...
  ]
}
```


Example Example: DescribeConnections


```
{
    "Records": [
    {
        "eventVersion": "1.0",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/Alice",
```

```
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice",
            "sessionContext": {
                "attributes": {
                    "mfaAuthenticated": "false",
                    "creationDate": "2014-04-04T12:23:05Z"
                }
            }
        },
        "eventTime": "2014-04-04T17:27:28Z",
        "eventSource": "directconnect.amazonaws.com",
        "eventName": "DescribeConnections",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "Coral/Jakarta",
        "requestParameters": null,
        "responseElements": null
    },
    ...
  ]
}
```

Example Example: DescribeVirtualInterfaces

```
{
    "Records": [
    {
        "eventVersion": "1.0",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/Alice",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice",
            "sessionContext": {
                "attributes": {
                    "mfaAuthenticated": "false",
                    "creationDate": "2014-04-04T12:23:05Z"
                }
            }
        },
        "eventTime": "2014-04-04T17:37:53Z",
        "eventSource": "directconnect.amazonaws.com",
        "eventName": "DescribeVirtualInterfaces",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "Coral/Jakarta",
        "requestParameters": {
            "connectionId": "dxcon-fhajolyy"
        },
        "responseElements": null
    },
    ...
  ]
}
```

Monitoring AWS Direct Connect resources

Monitoring is an important part of maintaining the reliability, availability, and performance of your AWS Direct Connect resources. You should collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. Before you start monitoring AWS Direct Connect; however, you should create a monitoring plan that includes answers to the following questions:

- What are your monitoring goals?
- What resources should be monitored?
- How often should you monitor these resources?
- What monitoring tools can you use?
- Who performs the monitoring tasks?
- Who should be notified when something goes wrong?

The next step is to establish a baseline for normal AWS Direct Connect performance in your environment, by measuring performance at various times and under different load conditions. As you monitor AWS Direct Connect, store historical monitoring data. That way, you can compare it with current performance data, identify normal performance patterns and performance anomalies, and devise methods to address issues.

To establish a baseline, you should monitor the usage, state, and health of your physical AWS Direct Connect connections.

Contents

Monitoring tools

AWS provides various tools that you can use to monitor an AWS Direct Connect connection. You can configure some of these tools to do the monitoring for you, while some of the tools require manual intervention. We recommend that you automate monitoring tasks as much as possible.

Automated monitoring tools

You can use the following automated monitoring tools to watch AWS Direct Connect and report when something is wrong:

- **Amazon CloudWatch Alarms** – Watch a single metric over a time period that you specify. Perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon SNS topic. CloudWatch alarms do not invoke actions simply because they are in a particular state; the state must have changed and been

maintained for a specified number of periods. For information about available metrics and dimensions, see Monitoring with Amazon CloudWatch (p. 140).

- **AWS CloudTrail Log Monitoring** – Share log files between accounts and monitor CloudTrail log files in real time by sending them to CloudWatch Logs. You can also write log processing applications in Java and validate that your log files have not changed after delivery by CloudTrail. For more information, see Logging AWS Direct Connect API calls using AWS CloudTrail (p. 135) and Working with CloudTrail Log Files in the *AWS CloudTrail User Guide*.

Manual monitoring tools

Another important part of monitoring an AWS Direct Connect connection involves manually monitoring those items that the CloudWatch alarms don't cover. The AWS Direct Connect and CloudWatch console dashboards provide an at-a-glance view of the state of your AWS environment.

- The AWS Direct Connect console shows:
  - Connection status (see the **State** column)
  - Virtual interface status (see the **State** column)
- The CloudWatch home page shows:
  - Current alarms and status
  - Graphs of alarms and resources
  - Service health status

  In addition, you can use CloudWatch to do the following:
  - Create customized dashboards to monitor the services you care about.
  - Graph metric data to troubleshoot issues and discover trends.
  - Search and browse all your AWS resource metrics.
  - Create and edit alarms to be notified of problems.

Monitoring with Amazon CloudWatch

You can monitor physical AWS Direct Connect connections, and virtual interfaces, using CloudWatch. CloudWatch collects raw data from AWS Direct Connect, and processes it into readable metrics. By default, CloudWatch provides AWS Direct Connect metric data in 5-minute intervals.

For detailed information about CloudWatch, see the Amazon CloudWatch User Guide. You can also monitor your services CloudWatch to see what ones are using resources. For more information, see AWS Services That Publish CloudWatch Metrics.

Contents

AWS Direct Connect metrics and dimensions

Metrics are available for AWS Direct Connect physical connections, and virtual interfaces.

AWS Direct Connect Connection metrics

The following metrics are available from AWS Direct Connect dedicated connections.

Metric	Description
`ConnectionState`	The state of the connection.1 indicates **up** and 0 indicates **down**. This metric is available for dedicated and hosted connections. Units: Boolean
`ConnectionBpsEgress`	The bitrate for outbound data from the AWS side of the connection. The number reported is the aggregate (average) over the specified time period (5 minutes by default, 1 minute minimum). You can change the default aggregate. This metric might be unavailable for a new connection, or when a device reboots. The metric starts when the connection is used to send or receive traffic. Units: Bits per second
`ConnectionBpsIngress`	The bitrate for inbound data to the AWS side of the connection. This metric might be unavailable for a new connection, or when a device reboots. The metric starts when the connection is used to send or receive traffic. Units: Bits per second
`ConnectionPpsEgress`	The packet rate for outbound data from the AWS side of the connection. The number reported is the aggregate (average) over the specified time period (5 minutes by default, 1 minute minimum). You can change the default aggregate. This metric might be unavailable for a new connection, or when a device reboots. The metric starts when the connection is used to send or receive traffic. Units: Packets per second
`ConnectionPpsIngress`	The packet rate for inbound data to the AWS side of the connection. The number reported is the aggregate (average) over the specified time period (5 minutes by default, 1 minute minimum). You can change the default aggregate. This metric might be unavailable for a new connection, or when a device reboots. The metric starts when the connection is used to send or receive traffic.

Metric	Description
	Units: Packets per second
`ConnectionCRCErrorCount`	This count is no longer in use. Use `ConnectionErrorCount` instead.
`ConnectionErrorCount`	The total error count for all types of MAC level errors on the AWS device. The total includes cyclic redundancy check (CRC) errors. This metric is the error count that occurred since the last reported datapoint. When there are errors on the interface, the metric reports non-zero values. To get the total count of all errors for the selected interval in CloudWatch, for example, 5 minutes, apply the"sum" statistic. For more information about getting the sum statistic, see Getting Statistics for a Metric in the *Amazon CloudWatch User Guide*. The metric value is set to 0 when the errors on the interface stop. **Note** This metric replaces `ConnectionCRCErrorCount`, which is no longer in use. Units: Count
`ConnectionLightLevelTx`	Indicates the health of the fiber connection for outbound (egress) traffic from the AWS side of the connection. There are two dimensions for this metric. For more information, see the section called “AWS Direct Connect available dimensions” (p. 143). Units: dBm
`ConnectionLightLevelRx`	Indicates the health of the fiber connection for inbound (ingress) traffic to the AWS side of the connection. There are two dimensions for this metric. For more information, see the section called “AWS Direct Connect available dimensions” (p. 143). Units: dBm
`ConnectionEncryptionState`	Indicates the connection encryption status. 1 indicates the connection encryption is `up`, and 0 indicates the connection encryption is `down`. When this metric is applied to a LAG, 1 indicates that all connections in the LAG have encrption `up`. 0 indicates at least one LAG connection encrption is `down`. **Note** The MAC Security feature is in Beta release for AWS Direct Connect, and is subject to change.

AWS Direct Connect virtual interface metrics

The following metrics are available from AWS Direct Connect virtual interfaces.

Metric	Description
`VirtualInterfaceBpsEgress`	The bitrate for outbound data from the AWS side of the virtual interface. The number reported is the aggregate (average) over the specified time period (5 minutes by default). Units: Bits per second
`VirtualInterfaceBpsIngress`	The bitrate for inbound data to the AWS side of the virtual interface. The number reported is the aggregate (average) over the specified time period (5 minutes by default). Units: Bits per second
`VirtualInterfacePpsEgress`	The packet rate for outbound data from the AWS side of the virtual interface. The number reported is the aggregate (average) over the specified time period (5 minutes by default). Units: Packets per second
`VirtualInterfacePpsIngress`	The packet rate for inbound data to the AWS side of the virtual interface. The number reported is the aggregate (average) over the specified time period (5 minutes by default). Units: Packets per second

AWS Direct Connect available dimensions

You can filter the AWS Direct Connect data using the following dimensions.

Dimension	Description
`ConnectionId`	This dimension is available on the metrics for AWS Direct Connect connection, and virtual interface. This dimension filters the data by the connection.
`OpticalLaneNumber`	This dimension filters the `ConnectionLightLevelTx` data and the `ConnectionLightLevelRx` data, and filters the data by the optical lane number of the AWS Direct Connect connection.
`VirtualInterfaceId`	This dimension is available on the metrics for AWS Direct Connect virtual interface, and filters the data by the virtual interface.

Viewing AWS Direct Connect CloudWatch metrics

AWS Direct Connect sends the following metrics about your AWS Direct Connect connections at 30-second intervals to Amazon CloudWatch. Amazon CloudWatch then aggregates these data points to 1-minute, or 5-minute intervals. You can use the following procedures to view the metrics for AWS Direct Connect connections.

To view metrics using the CloudWatch console

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. For **All metrics**, choose the **DX** metric namespace.
4. Choose **Connection Metrics**, and select the metric dimension to view the metrics (for example, for the AWS Direct Connect connection).
5. (Optional for Connection metrics) To return data for the selected metric in 1-minute intervals, choose **Graphed metrics**, and select **1 Minute** from the **Period** list.

To view metrics using the AWS Direct Connect console

1. Open the AWS Direct Connect console at https://console.aws.amazon.com/directconnect/.
2. In the navigation pane, choose **Connections**.
3. Select your connection. The **Monitoring** tab displays the metrics for your connection.

To view metrics using the AWS CLI

At a command prompt, use the following command.

```
aws cloudwatch list-metrics --namespace "AWS/DX"
```

Creating CloudWatch alarms to monitor AWS Direct Connect connections

You can create a CloudWatch alarm that sends an Amazon SNS message when the alarm changes state. An alarm watches a single metric over a time period that you specify. It sends a notification to an Amazon SNS topic based on the value of the metric relative to a given threshold over a number of time periods.

For example, you can create an alarm that monitors the state of an AWS Direct Connect connection. It sends a notification when the connection state is **down** for five consecutive 1-minute periods.

To create an alarm for the connection state

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Alarms**.
3. Choose **Create Alarm**.
4. Choose the **DX Metrics** category.
5. Select the AWS Direct Connect connection and choose the **ConnectionState** metric. Choose **Next**.
6. Configure the alarm as follows, and then choose **Create Alarm**:

- For **Alarm Threshold**, enter a name and description for your alarm. For **Whenever**, choose **<** and enter `1`. Enter **5** for the consecutive periods.
- For **Actions**, select an existing notification list or choose **New list** to create a new one.
- For **Alarm Preview**, select a period of 1 minute.

For more examples of creating alarms, see Creating Amazon CloudWatch Alarms in the *Amazon CloudWatch User Guide*.

AWS Direct Connect quotas

The following table lists the quotas related to AWS Direct Connect. Unless indicated otherwise, you can request an increase for any of these limits using the AWS Direct Connect Limits form.

Component	Quota	Comments
Private or public virtual interfaces per AWS Direct Connect dedicated connection	50	This limit cannot be increased.
Transit virtual interfaces per AWS Direct Connect dedicated connection	1	This limit cannot be increased.
Private or public virtual interfaces per AWS Direct Connect dedicated connection and transit virtual interfaces per AWS Direct Connect dedicated connection	50 and 1 transit virtual interface	This limit cannot be increased.
Private, public, or transit virtual interfaces per AWS Direct Connect hosted connection[1]	1	This limit cannot be increased.
Active AWS Direct Connect connections per Region per account	10	
Number of virtual interfaces per Link Aggregation Group (LAG)	50	
Routes per Border Gateway Protocol (BGP) session on a private virtual interface or transit virtual interface If you advertise more than 100 routes over the BGP session, the BGP session will go into an idle state with the BGP session DOWN.	100	This limit cannot be increased.
Routes per Border Gateway Protocol (BGP) session on a public virtual interface	1,000	This limit cannot be increased.
Dedicated connections per link aggregation group (LAG)	4 when the port speed is less than 100G 2 when the port speed is 100G	
Link aggregation groups (LAGs) per Region	10	
AWS Direct Connect gateways per account	200	
Virtual private gateways per AWS Direct Connect gateway	10	This limit cannot be increased.

Component	Quota	Comments
Transit gateways per AWS Direct Connect gateway	3	This limit cannot be increased.
Virtual interfaces (private or transit) per AWS Direct Connect gateway	30	
Number of prefixes from on-premises to AWS on a transit virtual interface	100	This limit cannot be increased.
Number of prefixes per AWS Transit Gateway from AWS to on-premise on a transit virtual interface	20	This limit cannot be increased.
Number of virtual interfaces per virtual private gateway	There is no limit.	

AWS Direct Connect supports these port speeds over single-mode fiber: 1 Gbps: 1000BASE-LX (1310 nm), 10 Gbps: 10GBASE-LR (1310 nm) and 100Gbps: 100GBASE-LR4.

[1]: You cannot create a transit virtual interface on a hosted connection with a capacity less than 1Gbps.

BGP quotas

The following are BGP quotas. The BGP timers negotiate down to the lowest value between the routers. The BFD intervals are defined by the slowest device.

- Default hold timer: 90 seconds
- Minimum hold timer: 3 seconds

  A hold value of 0 is not supported.
- Default keepalive timer: 30 seconds
- Minimum keepalive timer: 1 second
- Graceful restart timer: 120 seconds

  We recommend that you do not configure graceful restart and BFD at the same time.
- BFD liveness detection minimum interval: 300 ms
- BFD minimum multiplier: 3

Load balance considerations

If you want to use load balancing with multiple public VIFs, all the VIFs must be in the same Region.

Troubleshooting AWS Direct Connect

The following troubleshooting information can help you diagnose and fix issues with your AWS Direct Connect connection.

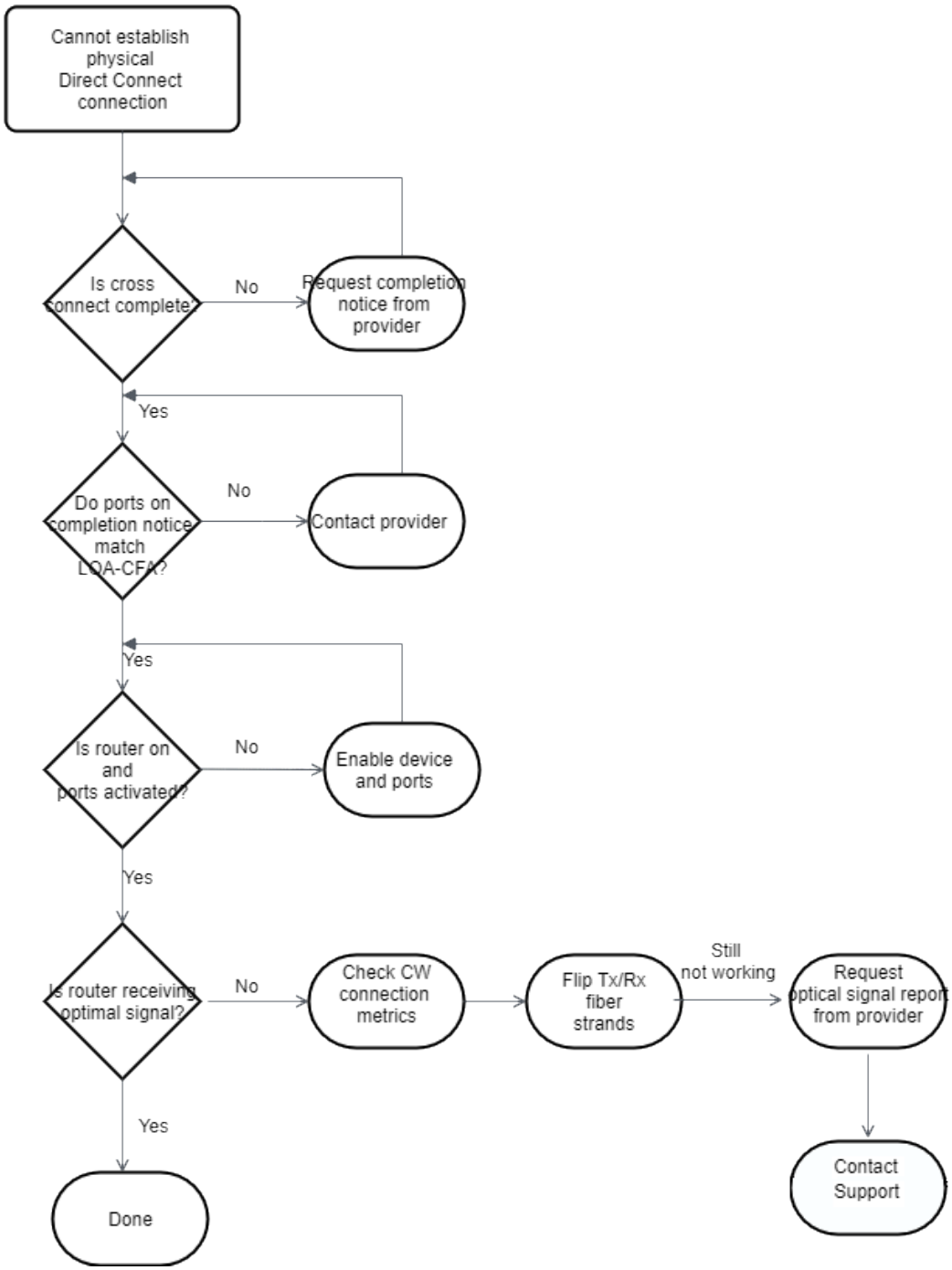
Contents

Troubleshooting layer 1 (physical) issues

If you or your network provider are having difficulty establishing physical connectivity to an AWS Direct Connect device, use the following steps to troubleshoot the issue.

1. Verify with the colocation provider that the cross connect is complete. Ask them or your network provider to provide you with a cross connect completion notice and compare the ports with those listed on your LOA-CFA.
2. Verify that your router or your provider's router is powered on and that the ports are activated.
3. Ensure that the routers are using the correct optical transceiver. Auto-negotiation for the port must be disabled. Auto-negotiation is supported only if the port speed is 1 Gbps. Port speed and full-duplex mode must be configured manually.
4. Verify that the router is receiving an acceptable optical signal over the cross connect.
5. Try flipping (also known as rolling) the Tx/Rx fiber strands.
6. Check the Amazon CloudWatch metrics for AWS Direct Connect. You can verify the AWS Direct Connect device's Tx/Rx optical readings (10-Gbps port speeds only), physical error count, and operational status. For more information, see Monitoring with Amazon CloudWatch.
7. Contact the colocation provider and request a written report for the Tx/Rx optical signal across the cross connect.
8. If the above steps do not resolve physical connectivity issues, contact AWS Support and provide the cross connect completion notice and optical signal report from the colocation provider.

The following flow chart contains the steps to diagnose issues with the physical connection.

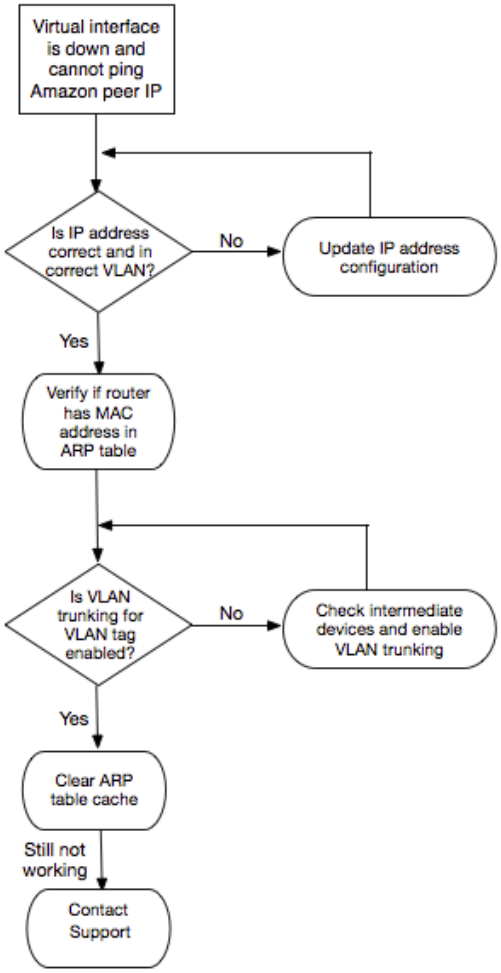


Troubleshooting layer 2 (data link) issues

If your AWS Direct Connect physical connection is up but your virtual interface is down, use the following steps to troubleshoot the issue.

1. If you cannot ping the Amazon peer IP address, verify that your peer IP address is configured correctly and in the correct VLAN. Ensure that the IP address is configured in the VLAN subinterface and not the physical interface (for example, GigabitEthernet0/0.123 instead of GigabitEthernet0/0).
2. Verify if the router has a MAC address entry from the AWS endpoint in your address resolution protocol (ARP) table.
3. Ensure that any intermediate devices between endpoints have VLAN trunking enabled for your 802.1Q VLAN tag. ARP cannot be established on the AWS side until AWS receives tagged traffic.
4. Clear your or your provider's ARP table cache.
5. If the above steps do not establish ARP or you still cannot ping the Amazon peer IP, contact AWS Support.

The following flow chart contains the steps to diagnose issues with the data link.



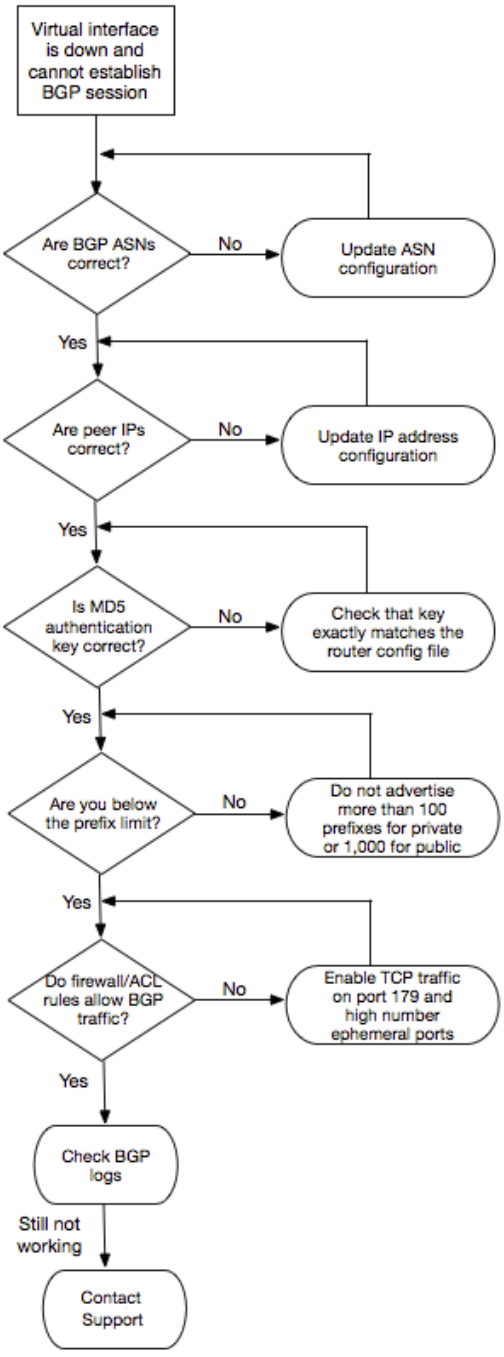
If the BGP session is still not established after verifying these steps, see Troubleshooting layer 3/4 (Network/Transport) issues (p. 151). If the BGP session is established but you are experiencing routing issues, see Troubleshooting routing issues (p. 152).

Troubleshooting layer 3/4 (Network/Transport) issues

Consider a situation where your AWS Direct Connect physical connection is up and you can ping the Amazon peer IP address. If your virtual interface is down and the BGP peering session cannot be established, use the following steps to troubleshoot the issue:

1. Ensure that your BGP local Autonomous System Number (ASN) and Amazon's ASN are configured correctly.
2. Ensure that the peer IPs for both sides of the BGP peering session are configured correctly.
3. Ensure that your MD5 authentication key is configured and exactly matches the key in the downloaded router configuration file. Check that there are no extra spaces or characters.
4. Verify that you or your provider are not advertising more than 100 prefixes for private virtual interfaces or 1,000 prefixes for public virtual interfaces. These are hard limits and cannot be exceeded.
5. Ensure that there are no firewall or ACL rules that are blocking TCP port 179 or any high-numbered ephemeral TCP ports. These ports are necessary for BGP to establish a TCP connection between the peers.
6. Check your BGP logs for any errors or warning messages.
7. If the above steps do not establish the BGP peering session, contact AWS Support.

The following flow chart contains the steps to diagnose issues with the BGP peering session.



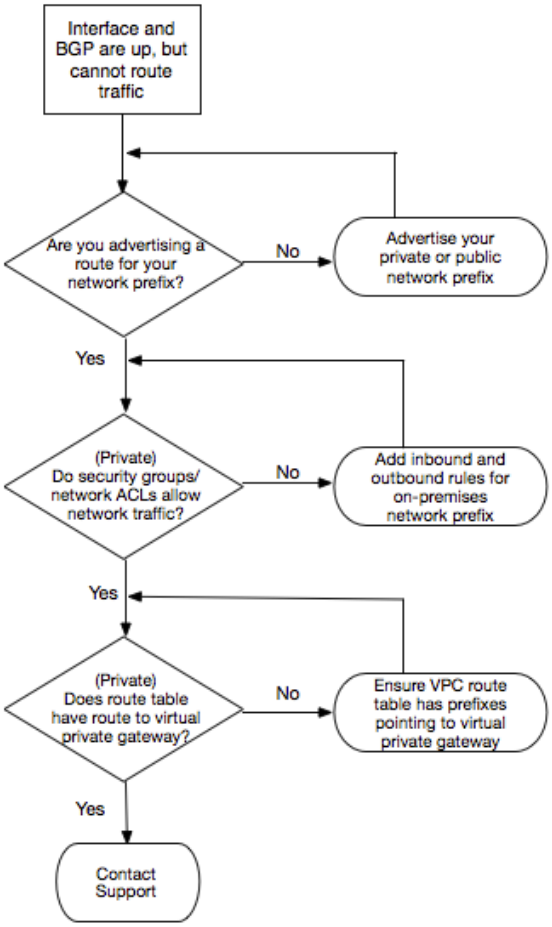
If the BGP peering session is established but you are experiencing routing issues, see Troubleshooting routing issues (p. 152).

Troubleshooting routing issues

Consider a situation where your virtual interface is up and you've established a BGP peering session. If you cannot route traffic over the virtual interface, use the following steps to troubleshoot the issue:

1. Ensure that you are advertising a route for your on-premises network prefix over the BGP session. For a private virtual interface, this can be a private or public network prefix. For a public virtual interface, this must be your publicly routable network prefix.
2. For a private virtual interface, ensure that your VPC security groups and network ACLs allow inbound and outbound traffic for your on-premises network prefix. For more information, see Security Groups and Network ACLs in the *Amazon VPC User Guide*.
3. For a private virtual interface, ensure that your VPC route tables have prefixes pointing to the virtual private gateway to which your private virtual interface is connected. For example, if you prefer to have all your traffic routed towards your on-premises network by default, you can add the default route (0.0.0.0/0 or ::/0) with the virtual private gateway as the target in your VPC route tables.
   - Alternatively, enable route propagation to automatically update routes in your route tables based on your dynamic BGP route advertisement. You can have up to 100 propagated routes per route table. This limit cannot be increased. For more information, see Enabling and Disabling Route Propagation in the *Amazon VPC User Guide*.
4. If the above steps do not resolve your routing issues, contact AWS Support.

The following flow chart contains the steps to diagnose routing issues.



Document history

The following table describes the releases for AWS Direct Connect.

Feature	Description	Date
Support MAC Security	You can use AWS Direct Connect connections that support MACsec to encrypt your data from your corporate data center to the AWS Direct Connect location. For more information, see *MAC Security* (p. 42).	2021-03-31
Support for 100G	Updated topics to include support for 100G dedicated connections.	2021-02-12
New location in Italy	Updated topic to include the addition of the new Israel location. For more information, see the section called “Europe (Italy)” (p. 57).	2021-01-22
New location in Israel	Updated topic to include the addition of the new Israel location. For more information, see the section called “Middle East (Israel)” (p. 58).	2020-07-07
Resiliency Toolkit Failover Testing support	Use the Resiliency Toolkit Failover Testing feature to test the resiliency of your connections.. For more information, see the section called “AWS Direct Connect Failover Test” (p. 39).	2020-06-03
CloudWatch VIF metric support	You can monitor physical AWS Direct Connect connections, and virtual interfaces, using CloudWatch. For more information, see the section called “Monitoring with Amazon CloudWatch” (p. 140).	2020-05-11
AWS Direct Connect Resiliency Toolkit	The AWS Direct Connect Resiliency Toolkit provides a connection wizard with multiple resiliency models that helps you order dedicated connections to achieve your SLA objective. For more information, see *Using the AWS Direct Connect Resiliency Toolkit to get started* (p. 8).	2019-10-07
Additional Region support for Support for AWS Transit Gateway across accounts	For information, see the section called “Transit gateway associations” (p. 99).	2019-09-30
AWS Direct Connect Support for AWS Transit Gateway	You can use an *AWS Direct Connect gateway* to connect your AWS Direct Connect connection over a transit virtual interface to the VPCs or VPNs attached to your transit gateway You associate a Direct Connect gateway with the transit gateway Then, create a transit virtual interface for your AWS Direct Connect connection to the Direct Connect gateway. For information, see the section called “Transit gateway associations” (p. 99).	2019-03-27
Jumbo frames support	You can send jumbo frames (9001 MTU) over AWS Direct Connect. For more information, see Set network MTU for private virtual interfaces or transit virtual interfaces (p. 72).	2018-10-11

Feature	Description	Date
Local preference BGP communities	You can use local preference BGP community tags to achieve load balancing and route preference for incoming traffic to your network. For more information, see Local preference BGP communities (p. 6).	2018-02-06
AWS Direct Connect gateway	You can use a Direct Connect gateway to connect your AWS Direct Connect connection to VPCs in remote Regions. For more information, see Working with Direct Connect gateways (p. 88).	2017-11-01
Amazon CloudWatch metrics	You can view CloudWatch metrics for your AWS Direct Connect connections. For more information, see Monitoring with Amazon CloudWatch (p. 140).	2017-06-29
Link aggregation groups	You can create a link aggregation group (LAG) to aggregate multiple AWS Direct Connect connections. For more information, see Link aggregation groups (p. 80).	2017-02-13
IPv6 support	Your virtual interface can now support an IPv6 BGP peering session. For more information, see Add or delete a BGP peer (p. 71).	2016-12-01
Tagging support	You can now tag your AWS Direct Connect resources. For more information, see Tagging AWS Direct Connect resources (p. 106).	2016-11-04
Self-service LOA-CFA	You can now download your Letter of Authorization and Connecting Facility Assignment (LOA-CFA) using the AWS Direct Connect console or API.	2016-06-22
New location in Silicon Valley	Updated topic to include the addition of the new Silicon Valley location in the US West (N. California) Region.	2016-06-03
New location in Amsterdam	Updated topic to include the addition of the new Amsterdam location in the Europe (Frankfurt) Region.	2016-05-19
New locations in Portland, Oregon, and Singapore	Updated topic to include the addition of the new Portland, Oregon, and Singapore locations in the US West (Oregon) and Asia Pacific (Singapore) Regions.	2016-04-27
New location in Sao Paulo, Brasil	Updated topic to include the addition of the new Sao Paulo location in the South America (São Paulo) Region.	2015-12-09
New locations in Dallas, London, Silicon Valley, and Mumbai	Updated topics to include the addition of the new locations in Dallas (US East (N. Virginia) Region), London (Europe (Ireland) Region), Silicon Valley (AWS GovCloud (US-West) Region), and Mumbai (Asia Pacific (Singapore) Region).	2015-11-27
New location in the China (Beijing) Region	Updated topics to include the addition of the new Beijing location in the China (Beijing) Region.	2015-04-14

Feature	Description	Date
New Las Vegas location in the US West (Oregon) Region	Updated topics to include the addition of the new AWS Direct Connect Las Vegas location in the US West (Oregon) Region.	2014-11-10
New EU (Frankfurt) Region	Updated topics to include the addition of the new AWS Direct Connect locations serving the EU (Frankfurt) Region.	2014-10-23
New locations in the Asia Pacific (Sydney) Region	Updated topics to include the addition of the new AWS Direct Connect locations serving the Asia Pacific (Sydney) Region.	2014-07-14
Support for AWS CloudTrail	Added a new topic to explain how you can use CloudTrail to log activity in AWS Direct Connect. For more information, see Logging AWS Direct Connect API calls using AWS CloudTrail (p. 135).	2014-04-04
Support for accessing remote AWS Regions	Added a new topic to explain how you can access public resources in a remote Region. For more information, see Accessing a remote AWS Region (p. 3).	2013-12-19
Support for hosted connections	Updated topics to include support for hosted connections.	2013-10-22
New location in the EU (Ireland) Region	Updated topics to include the addition of the new AWS Direct Connect location serving the EU (Ireland) Region.	2013-06-24
New Seattle location in the US West (Oregon) Region	Updated topics to include the addition of the new AWS Direct Connect location in Seattle serving the US West (Oregon) Region.	2013-05-08
Support for using IAM with AWS Direct Connect	Added a topic about using AWS Identity and Access Management with AWS Direct Connect. For more information, see the section called “Identity and access management” (p. 111).	2012-12-21
New Asia Pacific (Sydney) Region	Updated topics to include the addition of the new AWS Direct Connect location serving the Asia Pacific (Sydney) Region.	2012-12-14

Feature	Description	Date
New AWS Direct Connect console, and the US East (N. Virginia) and South America (Sao Paulo) Regions	Replaced the AWS Direct Connect Getting Started Guide with the AWS Direct Connect User Guide. Added new topics to cover the new AWS Direct Connect console, added a billing topic, added router configuration information, and updated topics to include the addition of two new AWS Direct Connect locations serving the US East (N. Virginia) and South America (Sao Paulo) Regions.	2012-08-13
Support for the EU (Ireland), Asia Pacific (Singapore), and Asia Pacific (Tokyo) Regions	Added a new troubleshooting section and updated topics to include the addition of four new AWS Direct Connect locations serving the US West (Northern California), EU (Ireland), Asia Pacific (Singapore), and Asia Pacific (Tokyo) Regions.	2012-01-10
Support for the US West (Northern California) Region	Updated topics to include the addition of the US West (Northern California) Region.	2011-09-08
Public release	The first release of AWS Direct Connect.	2011-08-03