

---

# **Direct Connect**

## **API Reference**

### **API Version 2012-10-25**



## Direct Connect: API Reference

Copyright © 2013 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

The following are trademarks of Amazon Web Services, Inc.: Amazon, Amazon Web Services Design, AWS, Amazon CloudFront, Cloudfront, Amazon DevPay, DynamoDB, ElastiCache, Amazon EC2, Amazon Elastic Compute Cloud, Amazon Glacier, Kindle, Kindle Fire, AWS Marketplace Design, Mechanical Turk, Amazon Redshift, Amazon Route 53, Amazon S3, Amazon VPC. In addition, Amazon.com graphics, logos, page headers, button icons, scripts, and service names are trademarks, or trade dress of Amazon in the U.S. and/or other countries. Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon.

All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

---

Welcome .....	1
Actions .....	2
AllocateConnectionOnInterconnect .....	3
AllocatePrivateVirtualInterface .....	7
AllocatePublicVirtualInterface .....	11
ConfirmConnection .....	15
ConfirmPrivateVirtualInterface .....	17
ConfirmPublicVirtualInterface .....	19
CreateConnection .....	21
CreateInterconnect .....	24
CreatePrivateVirtualInterface .....	27
CreatePublicVirtualInterface .....	31
DeleteConnection .....	35
DeleteInterconnect .....	38
DeleteVirtualInterface .....	40
DescribeConnections .....	42
DescribeConnectionsOnInterconnect .....	44
DescribeInterconnects .....	46
DescribeLocations .....	48
DescribeVirtualGateways .....	49
DescribeVirtualInterfaces .....	50
Data Types .....	52
AllocateConnectionOnInterconnectResult .....	53
AllocatePrivateVirtualInterfaceResult .....	54
AllocatePublicVirtualInterfaceResult .....	57
ConfirmConnectionResult .....	60
ConfirmPrivateVirtualInterfaceResult .....	60
ConfirmPublicVirtualInterfaceResult .....	61
Connection .....	62
CreateConnectionResult .....	63
CreateInterconnectResult .....	65
CreatePrivateVirtualInterfaceResult .....	66
CreatePublicVirtualInterfaceResult .....	69
DeleteConnectionResult .....	71
DeleteInterconnectResult .....	73
DeleteVirtualInterfaceResult .....	74
DescribeConnectionsOnInterconnectResult .....	74
DescribeConnectionsResult .....	75
DescribeInterconnectsResult .....	75
DescribeLocationsResult .....	75
DescribeVirtualGatewaysResult .....	76
DescribeVirtualInterfacesResult .....	76
Interconnect .....	76
Location .....	78
NewPrivateVirtualInterface .....	78
NewPrivateVirtualInterfaceAllocation .....	79
NewPublicVirtualInterface .....	80
NewPublicVirtualInterfaceAllocation .....	81
RouteFilterPrefix .....	82
VirtualGateway .....	83
VirtualInterface .....	83
Common Parameters .....	87
Common Errors .....	89

# Welcome

---

AWS Direct Connect makes it easy to establish a dedicated network connection from your premises to Amazon Web Services (AWS). Using AWS Direct Connect, you can establish private connectivity between AWS and your data center, office, or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

The AWS Direct Connect API Reference provides descriptions, syntax, and usage examples for each of the actions and data types for AWS Direct Connect. Use the following links to get started using the *AWS Direct Connect API Reference*:

- [Actions](#): An alphabetical list of all AWS Direct Connect actions.
- [Data Types](#): An alphabetical list of all AWS Direct Connect data types.
- [Common Query Parameters](#): Parameters that all Query actions can use.
- [Common Errors](#): Client and server errors that all actions can return.

This document was last updated on October 28, 2013.

# Actions

---

The following actions are supported:

- [AllocateConnectionOnInterconnect](#) (p. 3)
- [AllocatePrivateVirtualInterface](#) (p. 7)
- [AllocatePublicVirtualInterface](#) (p. 11)
- [ConfirmConnection](#) (p. 15)
- [ConfirmPrivateVirtualInterface](#) (p. 17)
- [ConfirmPublicVirtualInterface](#) (p. 19)
- [CreateConnection](#) (p. 21)
- [CreateInterconnect](#) (p. 24)
- [CreatePrivateVirtualInterface](#) (p. 27)
- [CreatePublicVirtualInterface](#) (p. 31)
- [DeleteConnection](#) (p. 35)
- [DeleteInterconnect](#) (p. 38)
- [DeleteVirtualInterface](#) (p. 40)
- [DescribeConnections](#) (p. 42)
- [DescribeConnectionsOnInterconnect](#) (p. 44)
- [DescribeInterconnects](#) (p. 46)
- [DescribeLocations](#) (p. 48)
- [DescribeVirtualGateways](#) (p. 49)
- [DescribeVirtualInterfaces](#) (p. 50)

# AllocateConnectionOnInterconnect

Creates a hosted connection on an interconnect.

Allocates a VLAN number and a specified amount of bandwidth for use by a hosted connection on the given interconnect.

## Request Syntax

```
{  
  "bandwidth": "string",  
  "connectionName": "string",  
  "interconnectId": "string",  
  "ownerAccount": "string",  
  "vlan": "number"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### bandwidth

Bandwidth of the connection.

Example: "500Mbps"

Default: None

Type: String

Required: Yes

### connectionName

Name of the provisioned connection.

Example: "500M Connection to AWS"

Default: None

Type: String

Required: Yes

### interconnectId

ID of the interconnect on which the connection will be provisioned.

Example: dxcon-456abc78

Default: None

Type: String

Required: Yes

#### ownerAccount

Numeric account Id of the customer for whom the connection will be provisioned.

Example: 123443215678

Default: None

Type: String

Required: Yes

#### vlan

The dedicated VLAN provisioned to the connection.

Example: 101

Default: None

Type: Number

Required: Yes

## Response Syntax

```
{
  "bandwidth": "string",
  "connectionId": "string",
  "connectionName": "string",
  "connectionState": "string",
  "location": "string",
  "ownerAccount": "string",
  "partnerName": "string",
  "region": "string",
  "vlan": "number"
}
```

## Response Elements

The following data is returned in JSON format by the service.

#### bandwidth

Bandwidth of the connection.

Example: 1Gbps (for regular connections), or 500Mbps (for hosted connections)

Default: None

Type: String

#### connectionId

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

#### connectionName

The name of the connection.

Example: "My Connection to AWS"

Default: None

Type: String

#### connectionState

State of the connection.

- **Ordering:** The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
- **Requested:** The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending:** The connection has been approved, and is being initialized.
- **Available:** The network link is up, and the connection is ready for use.
- **Down:** The network link is down.
- **Deleted:** The connection has been deleted.
- **Rejected:** A hosted connection in the 'Ordering' state will enter the 'Rejected' state if it is deleted by the end customer.

Type: String

Valid Values: ordering | requested | pending | available | down | deleting | deleted | rejected

#### location

Where the connection is located.

Example: EqSV5

Default: None

Type: String

#### ownerAccount

Type: String

#### partnerName

Type: String

#### region

The AWS region where the connection is located.

Example: us-east-1

Default: None

Type: String

#### vlan

The VLAN ID.

Example: 101

Type: Number



## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# AllocatePrivateVirtualInterface

Provisions a private virtual interface to be owned by a different customer.

The owner of a connection calls this function to provision a private virtual interface which will be owned by another AWS customer.

Virtual interfaces created using this function must be confirmed by the virtual interface owner by calling `ConfirmPrivateVirtualInterface`. Until this step has been completed, the virtual interface will be in 'Confirming' state, and will not be available for handling traffic.

## Request Syntax

```
{
  "connectionId": "string",
  "newPrivateVirtualInterfaceAllocation": {
    "amazonAddress": "string",
    "asn": "number",
    "authKey": "string",
    "customerAddress": "string",
    "virtualInterfaceName": "string",
    "vlan": "number"
  },
  "ownerAccount": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### **connectionId**

The connection ID on which the private virtual interface is provisioned.

Default: None

Type: String

Required: Yes

### **newPrivateVirtualInterfaceAllocation**

Detailed information for the private virtual interface to be provisioned.

Default: None

Type: [NewPrivateVirtualInterfaceAllocation \(p. 79\)](#) object

Required: Yes

### **ownerAccount**

The AWS account that will own the new private virtual interface.

Default: None

Type: String

Required: Yes

## Response Syntax

```
{
  "amazonAddress": "string",
  "asn": "number",
  "authKey": "string",
  "connectionId": "string",
  "customerAddress": "string",
  "customerRouterConfig": "string",
  "location": "string",
  "ownerAccount": "string",
  "routeFilterPrefixes": [
    {
      "cidr": "string"
    }
  ],
  "virtualGatewayId": "string",
  "virtualInterfaceId": "string",
  "virtualInterfaceName": "string",
  "virtualInterfaceState": "string",
  "virtualInterfaceType": "string",
  "vlan": "number"
}
```

## Response Elements

The following data is returned in JSON format by the service.

### **amazonAddress**

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

### **asn**

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

### **authKey**

Authentication key for BGP configuration.

Example: asdf34example

Type: String

### **connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

**customerAddress**

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

**customerRouterConfig**

Information for generating the customer router configuration.

Type: String

**location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

**ownerAccount**

Type: String

**routeFilterPrefixes**

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

Type: array of [RouteFilterPrefix](#) (p. 82) objects

**virtualGatewayId**

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

Type: String

**virtualInterfaceId**

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Type: String

**virtualInterfaceName**

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

**virtualInterfaceState**

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.

- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: `confirming` | `verifying` | `pending` | `available` | `deleting` | `deleted` | `rejected`

#### **virtualInterfaceType**

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

Type: String

#### **vlan**

The VLAN ID.

Example: 101

Type: Number

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

#### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

#### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# AllocatePublicVirtualInterface

Provisions a public virtual interface to be owned by a different customer.

The owner of a connection calls this function to provision a public virtual interface which will be owned by another AWS customer.

Virtual interfaces created using this function must be confirmed by the virtual interface owner by calling `ConfirmPublicVirtualInterface`. Until this step has been completed, the virtual interface will be in 'Confirming' state, and will not be available for handling traffic.

## Request Syntax

```
{
  "connectionId": "string",
  "newPublicVirtualInterfaceAllocation": {
    "amazonAddress": "string",
    "asn": "number",
    "authKey": "string",
    "customerAddress": "string",
    "routeFilterPrefixes": [
      {
        "cidr": "string"
      }
    ],
    "virtualInterfaceName": "string",
    "vlan": "number"
  },
  "ownerAccount": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### **connectionId**

The connection ID on which the public virtual interface is provisioned.

Default: None

Type: String

Required: Yes

### **newPublicVirtualInterfaceAllocation**

Detailed information for the public virtual interface to be provisioned.

Default: None

Type: [NewPublicVirtualInterfaceAllocation \(p. 81\)](#) object

Required: Yes

#### ownerAccount

The AWS account that will own the new public virtual interface.

Default: None

Type: String

Required: Yes

## Response Syntax

```
{
  "amazonAddress": "string",
  "asn": "number",
  "authKey": "string",
  "connectionId": "string",
  "customerAddress": "string",
  "customerRouterConfig": "string",
  "location": "string",
  "ownerAccount": "string",
  "routeFilterPrefixes": [
    {
      "cidr": "string"
    }
  ],
  "virtualGatewayId": "string",
  "virtualInterfaceId": "string",
  "virtualInterfaceName": "string",
  "virtualInterfaceState": "string",
  "virtualInterfaceType": "string",
  "vlan": "number"
}
```

## Response Elements

The following data is returned in JSON format by the service.

#### amazonAddress

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

#### asn

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

#### authKey

Authentication key for BGP configuration.

Example: asdf34example

Type: String

**connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

**customerAddress**

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

**customerRouterConfig**

Information for generating the customer router configuration.

Type: String

**location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

**ownerAccount**

Type: String

**routeFilterPrefixes**

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

Type: array of [RouteFilterPrefix](#) (p. 82) objects

**virtualGatewayId**

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

Type: String

**virtualInterfaceId**

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Type: String

**virtualInterfaceName**

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

**virtualInterfaceState**

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which



it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.

- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.
- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: `confirming` | `verifying` | `pending` | `available` | `deleting` | `deleted` | `rejected`

#### **virtualInterfaceType**

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

Type: String

#### **vlan**

The VLAN ID.

Example: 101

Type: Number

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

#### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

#### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# ConfirmConnection

Confirm the creation of a hosted connection on an interconnect.

Upon creation, the hosted connection is initially in the 'Ordering' state, and will remain in this state until the owner calls ConfirmConnection to confirm creation of the hosted connection.

## Request Syntax

```
{  
  "connectionId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### connectionId

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: Yes

## Response Syntax

```
{  
  "connectionState": "string"  
}
```

## Response Elements

The following data is returned in JSON format by the service.

### connectionState

State of the connection.

- **Ordering**: The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
- **Requested**: The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending**: The connection has been approved, and is being initialized.

- **Available:** The network link is up, and the connection is ready for use.
- **Down:** The network link is down.
- **Deleted:** The connection has been deleted.
- **Rejected:** A hosted connection in the 'Ordering' state will enter the 'Rejected' state if it is deleted by the end customer.

Type: String

Valid Values: ordering | requested | pending | available | down | deleting | deleted | rejected

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# ConfirmPrivateVirtualInterface

Accept ownership of a private virtual interface created by another customer.

After the virtual interface owner calls this function, the virtual interface will be created and attached to the given virtual private gateway, and will be available for handling traffic.

## Request Syntax

```
{  
  "virtualGatewayId": "string",  
  "virtualInterfaceId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### **virtualGatewayId**

ID of the virtual private gateway that will be attached to the virtual interface.

A virtual private gateway can be managed via the Amazon Virtual Private Cloud (VPC) console or the [EC2 CreateVpnGateway](#) action.

Default: None

Type: String

Required: Yes

### **virtualInterfaceId**

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Type: String

Required: Yes

## Response Syntax

```
{  
  "virtualInterfaceState": "string"  
}
```

## Response Elements

The following data is returned in JSON format by the service.

### **virtualInterfaceState**

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.
- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: `confirming` | `verifying` | `pending` | `available` | `deleting` | `deleted`  
| `rejected`

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# ConfirmPublicVirtualInterface

Accept ownership of a public virtual interface created by another customer.

After the virtual interface owner calls this function, the specified virtual interface will be created and made available for handling traffic.

## Request Syntax

```
{  
  "virtualInterfaceId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### **virtualInterfaceId**

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Type: String

Required: Yes

## Response Syntax

```
{  
  "virtualInterfaceState": "string"  
}
```

## Response Elements

The following data is returned in JSON format by the service.

### **virtualInterfaceState**

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.

- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.
- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: `confirming` | `verifying` | `pending` | `available` | `deleting` | `deleted`  
| `rejected`

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# CreateConnection

Creates a new connection between the customer network and a specific AWS Direct Connect location.

A connection links your internal network to an AWS Direct Connect location over a standard 1 gigabit or 10 gigabit Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to an AWS Direct Connect router. An AWS Direct Connect location provides access to Amazon Web Services in the region it is associated with. You can establish connections with AWS Direct Connect locations in multiple regions, but a connection in one region does not provide connectivity to other regions.

## Request Syntax

```
{
  "bandwidth": "string",
  "connectionName": "string",
  "location": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### bandwidth

Bandwidth of the connection.

Example: 1Gbps

Default: None

Type: String

Required: Yes

### connectionName

The name of the connection.

Example: "My Connection to AWS"

Default: None

Type: String

Required: Yes

### location

Where the connection is located.

Example: EqSV5

Default: None

Type: String



Required: Yes

## Response Syntax

```
{
  "bandwidth": "string",
  "connectionId": "string",
  "connectionName": "string",
  "connectionState": "string",
  "location": "string",
  "ownerAccount": "string",
  "partnerName": "string",
  "region": "string",
  "vlan": "number"
}
```

## Response Elements

The following data is returned in JSON format by the service.

### bandwidth

Bandwidth of the connection.

Example: 1Gbps (for regular connections), or 500Mbps (for hosted connections)

Default: None

Type: String

### connectionId

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

### connectionName

The name of the connection.

Example: "My Connection to AWS"

Default: None

Type: String

### connectionState

State of the connection.

- **Ordering:** The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
- **Requested:** The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending:** The connection has been approved, and is being initialized.
- **Available:** The network link is up, and the connection is ready for use.

- **Down:** The network link is down.
- **Deleted:** The connection has been deleted.
- **Rejected:** A hosted connection in the 'Ordering' state will enter the 'Rejected' state if it is deleted by the end customer.

Type: String

Valid Values: ordering | requested | pending | available | down | deleting | deleted | rejected

#### location

Where the connection is located.

Example: EqSV5

Default: None

Type: String

#### ownerAccount

Type: String

#### partnerName

Type: String

#### region

The AWS region where the connection is located.

Example: us-east-1

Default: None

Type: String

#### vlan

The VLAN ID.

Example: 101

Type: Number

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

#### DirectConnectClientException

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

#### DirectConnectServerException

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# CreateInterconnect

Creates a new interconnect between a AWS Direct Connect partner's network and a specific AWS Direct Connect location.

An interconnect is a connection which is capable of hosting other connections. The AWS Direct Connect partner can use an interconnect to provide sub-1Gbps AWS Direct Connect service to tier 2 customers who do not have their own connections. Like a standard connection, an interconnect links the AWS Direct Connect partner's network to an AWS Direct Connect location over a standard 1 Gbps or 10 Gbps Ethernet fiber-optic cable. One end is connected to the partner's router, the other to an AWS Direct Connect router.

For each end customer, the AWS Direct Connect partner provisions a connection on their interconnect by calling `AllocateConnectionOnInterconnect`. The end customer can then connect to AWS resources by creating a virtual interface on their connection, using the VLAN assigned to them by the AWS Direct Connect partner.

## Request Syntax

```
{
  "bandwidth": "string",
  "interconnectName": "string",
  "location": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### **bandwidth**

The port bandwidth

Example: 1Gbps

Default: None

Available values: 1Gbps,10Gbps

Type: String

Required: Yes

### **interconnectName**

The name of the interconnect.

Example: "1G Interconnect to AWS"

Default: None

Type: String

Required: Yes

### **location**

Where the interconnect is located

Example: EqSV5

Default: None

Type: String

Required: Yes

## Response Syntax

```
{
  "bandwidth": "string",
  "interconnectId": "string",
  "interconnectName": "string",
  "interconnectState": "string",
  "location": "string",
  "region": "string"
}
```

## Response Elements

The following data is returned in JSON format by the service.

### bandwidth

Bandwidth of the connection.

Example: 1Gbps

Default: None

Type: String

### interconnectId

The ID of the interconnect.

Example: dxcon-abc123

Type: String

### interconnectName

The name of the interconnect.

Example: "1G Interconnect to AWS"

Type: String

### interconnectState

State of the interconnect.

- **Requested:** The initial state of an interconnect. The interconnect stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending:** The interconnect has been approved, and is being initialized.
- **Available:** The network link is up, and the interconnect is ready for use.
- **Down:** The network link is down.
- **Deleted:** The interconnect has been deleted.

Type: String

Valid Values: requested | pending | available | down | deleting | deleted

#### location

Where the connection is located.

Example: EqSV5

Default: None

Type: String

#### region

The AWS region where the connection is located.

Example: us-east-1

Default: None

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

#### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

#### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# CreatePrivateVirtualInterface

Creates a new private virtual interface. A virtual interface is the VLAN that transports AWS Direct Connect traffic. A private virtual interface supports sending traffic to a single virtual private cloud (VPC).

## Request Syntax

```
{  
  "connectionId": "string",  
  "newPrivateVirtualInterface": {  
    "amazonAddress": "string",  
    "asn": "number",  
    "authKey": "string",  
    "customerAddress": "string",  
    "virtualGatewayId": "string",  
    "virtualInterfaceName": "string",  
    "vlan": "number"  
  }  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### connectionId

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: Yes

### newPrivateVirtualInterface

Detailed information for the private virtual interface to be created.

Default: None

Type: [NewPrivateVirtualInterface \(p. 78\)](#) object

Required: Yes

## Response Syntax

```
{  
  "amazonAddress": "string",  
  "asn": "number",  
}
```

```
"authKey": "string",
"connectionId": "string",
"customerAddress": "string",
"customerRouterConfig": "string",
"location": "string",
"ownerAccount": "string",
"routeFilterPrefixes": [
  {
    "cidr": "string"
  }
],
"virtualGatewayId": "string",
"virtualInterfaceId": "string",
"virtualInterfaceName": "string",
"virtualInterfaceState": "string",
"virtualInterfaceType": "string",
"vlan": "number"
}
```

## Response Elements

The following data is returned in JSON format by the service.

### amazonAddress

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

### asn

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

### authKey

Authentication key for BGP configuration.

Example: asdf34example

Type: String

### connectionId

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

### customerAddress

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

#### customerRouterConfig

Information for generating the customer router configuration.

Type: String

#### location

Where the connection is located.

Example: EqSV5

Default: None

Type: String

#### ownerAccount

Type: String

#### routeFilterPrefixes

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

Type: array of [RouteFilterPrefix](#) (p. 82) objects

#### virtualGatewayId

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

Type: String

#### virtualInterfaceId

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Type: String

#### virtualInterfaceName

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

#### virtualInterfaceState

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.
- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.



Type: String

Valid Values: confirming | verifying | pending | available | deleting | deleted  
| rejected

**virtualInterfaceType**

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

Type: String

**vlan**

The VLAN ID.

Example: 101

Type: Number

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

**DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# CreatePublicVirtualInterface

Creates a new public virtual interface. A virtual interface is the VLAN that transports AWS Direct Connect traffic. A public virtual interface supports sending traffic to public services of AWS such as Amazon Simple Storage Service (Amazon S3).

## Request Syntax

```
{
  "connectionId": "string",
  "newPublicVirtualInterface": {
    "amazonAddress": "string",
    "asn": "number",
    "authKey": "string",
    "customerAddress": "string",
    "routeFilterPrefixes": [
      {
        "cidr": "string"
      }
    ],
    "virtualInterfaceName": "string",
    "vlan": "number"
  }
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### connectionId

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: Yes

### newPublicVirtualInterface

Detailed information for the public virtual interface to be created.

Default: None

Type: [NewPublicVirtualInterface \(p. 80\)](#) object

Required: Yes

## Response Syntax

```
{
  "amazonAddress": "string",
  "asn": "number",
  "authKey": "string",
  "connectionId": "string",
  "customerAddress": "string",
  "customerRouterConfig": "string",
  "location": "string",
  "ownerAccount": "string",
  "routeFilterPrefixes": [
    {
      "cidr": "string"
    }
  ],
  "virtualGatewayId": "string",
  "virtualInterfaceId": "string",
  "virtualInterfaceName": "string",
  "virtualInterfaceState": "string",
  "virtualInterfaceType": "string",
  "vlan": "number"
}
```

## Response Elements

The following data is returned in JSON format by the service.

### **amazonAddress**

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

### **asn**

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

### **authKey**

Authentication key for BGP configuration.

Example: asdf34example

Type: String

### **connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

**customerAddress**

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

**customerRouterConfig**

Information for generating the customer router configuration.

Type: String

**location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

**ownerAccount**

Type: String

**routeFilterPrefixes**

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

Type: array of [RouteFilterPrefix](#) (p. 82) objects

**virtualGatewayId**

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

Type: String

**virtualInterfaceId**

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Type: String

**virtualInterfaceName**

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

**virtualInterfaceState**

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.

- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: `confirming` | `verifying` | `pending` | `available` | `deleting` | `deleted` | `rejected`

#### **virtualInterfaceType**

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

Type: String

#### **vlan**

The VLAN ID.

Example: 101

Type: Number

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

#### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

#### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# DeleteConnection

Deletes the connection.

Deleting a connection only stops the AWS Direct Connect port hour and data transfer charges. You need to cancel separately with the providers any services or charges for cross-connects or network circuits that connect you to the AWS Direct Connect location.

## Request Syntax

```
{
  "connectionId": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### connectionId

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: Yes

## Response Syntax

```
{
  "bandwidth": "string",
  "connectionId": "string",
  "connectionName": "string",
  "connectionState": "string",
  "location": "string",
  "ownerAccount": "string",
  "partnerName": "string",
  "region": "string",
  "vlan": "number"
}
```

## Response Elements

The following data is returned in JSON format by the service.

**bandwidth**

Bandwidth of the connection.

Example: 1Gbps (for regular connections), or 500Mbps (for hosted connections)

Default: None

Type: String

**connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

**connectionName**

The name of the connection.

Example: "My Connection to AWS"

Default: None

Type: String

**connectionState**

State of the connection.

- **Ordering**: The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
- **Requested**: The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending**: The connection has been approved, and is being initialized.
- **Available**: The network link is up, and the connection is ready for use.
- **Down**: The network link is down.
- **Deleted**: The connection has been deleted.
- **Rejected**: A hosted connection in the 'Ordering' state will enter the 'Rejected' state if it is deleted by the end customer.

Type: String

Valid Values: `ordering` | `requested` | `pending` | `available` | `down` | `deleting` | `deleted` | `rejected`

**location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

**ownerAccount**

Type: String

**partnerName**

Type: String

**region**

The AWS region where the connection is located.

Example: us-east-1

Default: None

Type: String

**vlan**

The VLAN ID.

Example: 101

Type: Number

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400



# DeleteInterconnect

Deletes the specified interconnect.

## Request Syntax

```
{  
  "interconnectId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### interconnectId

The ID of the interconnect.

Example: dxcon-abc123

Type: String

Required: Yes

## Response Syntax

```
{  
  "interconnectState": "string"  
}
```

## Response Elements

The following data is returned in JSON format by the service.

### interconnectState

State of the interconnect.

- **Requested:** The initial state of an interconnect. The interconnect stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending:** The interconnect has been approved, and is being initialized.
- **Available:** The network link is up, and the interconnect is ready for use.
- **Down:** The network link is down.
- **Deleted:** The interconnect has been deleted.

Type: String

Valid Values: requested | pending | available | down | deleting | deleted

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# DeleteVirtualInterface

Deletes a virtual interface.

## Request Syntax

```
{  
  "virtualInterfaceId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### **virtualInterfaceId**

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Type: String

Required: Yes

## Response Syntax

```
{  
  "virtualInterfaceState": "string"  
}
```

## Response Elements

The following data is returned in JSON format by the service.

### **virtualInterfaceState**

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.

- **Available:** A virtual interface that is able to forward traffic.
- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: confirming | verifying | pending | available | deleting | deleted  
| rejected

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# DescribeConnections

Displays all connections in this region.

If a connection ID is provided, the call returns only that particular connection.

## Request Syntax

```
{  
  "connectionId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### connectionId

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: No

## Response Syntax

```
{  
  "connections": [  
    {  
      "bandwidth": "string",  
      "connectionId": "string",  
      "connectionName": "string",  
      "connectionState": "string",  
      "location": "string",  
      "ownerAccount": "string",  
      "partnerName": "string",  
      "region": "string",  
      "vlan": "number"  
    }  
  ]  
}
```

## Response Elements

The following data is returned in JSON format by the service.

### **connections**

A list of connections.

Type: array of [Connection](#) (p. 62) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 89).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# DescribeConnectionsOnInterconnect

Return a list of connections that have been provisioned on the given interconnect.

## Request Syntax

```
{  
  "interconnectId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### **interconnectId**

ID of the interconnect on which a list of connection is provisioned.

Example: dxcon-abc123

Default: None

Type: String

Required: Yes

## Response Syntax

```
{  
  "connections": [  
    {  
      "bandwidth": "string",  
      "connectionId": "string",  
      "connectionName": "string",  
      "connectionState": "string",  
      "location": "string",  
      "ownerAccount": "string",  
      "partnerName": "string",  
      "region": "string",  
      "vlan": "number"  
    }  
  ]  
}
```

## Response Elements

The following data is returned in JSON format by the service.

### connections

A list of connections.

Type: array of [Connection](#) (p. 62) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 89).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400



# DescribeInterconnects

Returns a list of interconnects owned by the AWS account.

If an interconnect ID is provided, it will only return this particular interconnect.

## Request Syntax

```
{  
  "interconnectId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 87\)](#).

The request accepts the following data in JSON format.

### **interconnectId**

The ID of the interconnect.

Example: dxcon-abc123

Type: String

Required: No

## Response Syntax

```
{  
  "interconnects": [  
    {  
      "bandwidth": "string",  
      "interconnectId": "string",  
      "interconnectName": "string",  
      "interconnectState": "string",  
      "location": "string",  
      "region": "string"  
    }  
  ]  
}
```

## Response Elements

The following data is returned in JSON format by the service.

### **interconnects**

A list of interconnects.

Type: array of [Interconnect \(p. 76\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 89\)](#).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# DescribeLocations

Returns the list of AWS Direct Connect locations in the current AWS region. These are the locations that may be selected when calling `CreateConnection` or `CreateInterconnect`.

## Response Syntax

```
{
  "locations": [
    {
      "locationCode": "string",
      "locationName": "string"
    }
  ]
}
```

## Response Elements

The following data is returned in JSON format by the service.

### **locations**

Type: array of [Location](#) (p. 78) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 89).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# DescribeVirtualGateways

Returns a list of virtual private gateways owned by the AWS account.

You can create one or more AWS Direct Connect private virtual interfaces linking to a virtual private gateway. A virtual private gateway can be managed via Amazon Virtual Private Cloud (VPC) console or the [EC2 CreateVpnGateway](#) action.

## Response Syntax

```
{
  "virtualGateways": [
    {
      "virtualGatewayId": "string",
      "virtualGatewayState": "string"
    }
  ]
}
```

## Response Elements

The following data is returned in JSON format by the service.

### **virtualGateways**

A list of virtual private gateways.

Type: array of [VirtualGateway](#) (p. 83) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 89).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400



```
{
  {
    "amazonAddress": "string",
    "asn": "number",
    "authKey": "string",
    "connectionId": "string",
    "customerAddress": "string",
    "customerRouterConfig": "string",
    "location": "string",
    "ownerAccount": "string",
    "routeFilterPrefixes": [
      {
        "cidr": "string"
      }
    ],
    "virtualGatewayId": "string",
    "virtualInterfaceId": "string",
    "virtualInterfaceName": "string",
    "virtualInterfaceState": "string",
    "virtualInterfaceType": "string",
    "vlan": "number"
  }
]
```

## Response Elements

The following data is returned in JSON format by the service.

### **virtualInterfaces**

A list of virtual interfaces.

Type: array of [VirtualInterface](#) (p. 83) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 89).

### **DirectConnectClientException**

The API was called with invalid parameters. The error message will contain additional details about the cause.

HTTP Status Code: 400

### **DirectConnectServerException**

A server-side error occurred during the API call. The error message will contain additional details about the cause.

HTTP Status Code: 400

# Data Types

---

The Direct Connect API contains several data types that various actions use. This section describes each data type in detail.

**Note**

The order of each element in the response is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- [AllocateConnectionOnInterconnectResult](#) (p. 53)
- [AllocatePrivateVirtualInterfaceResult](#) (p. 54)
- [AllocatePublicVirtualInterfaceResult](#) (p. 57)
- [ConfirmConnectionResult](#) (p. 60)
- [ConfirmPrivateVirtualInterfaceResult](#) (p. 60)
- [ConfirmPublicVirtualInterfaceResult](#) (p. 61)
- [Connection](#) (p. 62)
- [CreateConnectionResult](#) (p. 63)
- [CreateInterconnectResult](#) (p. 65)
- [CreatePrivateVirtualInterfaceResult](#) (p. 66)
- [CreatePublicVirtualInterfaceResult](#) (p. 69)
- [DeleteConnectionResult](#) (p. 71)
- [DeleteInterconnectResult](#) (p. 73)
- [DeleteVirtualInterfaceResult](#) (p. 74)
- [DescribeConnectionsOnInterconnectResult](#) (p. 74)
- [DescribeConnectionsResult](#) (p. 75)
- [DescribeInterconnectsResult](#) (p. 75)
- [DescribeLocationsResult](#) (p. 75)
- [DescribeVirtualGatewaysResult](#) (p. 76)
- [DescribeVirtualInterfacesResult](#) (p. 76)
- [Interconnect](#) (p. 76)
- [Location](#) (p. 78)
- [NewPrivateVirtualInterface](#) (p. 78)
- [NewPrivateVirtualInterfaceAllocation](#) (p. 79)

- [NewPublicVirtualInterface](#) (p. 80)
- [NewPublicVirtualInterfaceAllocation](#) (p. 81)
- [RouteFilterPrefix](#) (p. 82)
- [VirtualGateway](#) (p. 83)
- [VirtualInterface](#) (p. 83)

# AllocateConnectionOnInterconnectResult

## Description

A connection represents the physical network connection between the AWS Direct Connect location and the customer.

## Contents

### **bandwidth**

Bandwidth of the connection.

Example: 1Gbps (for regular connections), or 500Mbps (for hosted connections)

Default: None

Type: String

Required: No

### **connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: No

### **connectionName**

The name of the connection.

Example: "My Connection to AWS"

Default: None

Type: String

Required: No

### **connectionState**

State of the connection.

- **Ordering**: The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
- **Requested**: The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending**: The connection has been approved, and is being initialized.



- **Available:** The network link is up, and the connection is ready for use.
- **Down:** The network link is down.
- **Deleted:** The connection has been deleted.
- **Rejected:** A hosted connection in the 'Ordering' state will enter the 'Rejected' state if it is deleted by the end customer.

Type: String

Valid Values: `ordering` | `requested` | `pending` | `available` | `down` | `deleting` | `deleted` | `rejected`

Required: No

**location**

Where the connection is located.

Example: `EqSV5`

Default: None

Type: String

Required: No

**ownerAccount**

Type: String

Required: No

**partnerName**

Type: String

Required: No

**region**

The AWS region where the connection is located.

Example: `us-east-1`

Default: None

Type: String

Required: No

**vlan**

The VLAN ID.

Example: `101`

Type: Number

Required: No

## AllocatePrivateVirtualInterfaceResult

### Description

A virtual interface (VLAN) transmits the traffic between the AWS Direct Connect location and the customer.

## Contents

### **amazonAddress**

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

Required: No

### **asn**

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

Required: No

### **authKey**

Authentication key for BGP configuration.

Example: asdf34example

Type: String

Required: No

### **connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: No

### **customerAddress**

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

Required: No

### **customerRouterConfig**

Information for generating the customer router configuration.

Type: String

Required: No

### **location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

Required: No

**ownerAccount**

Type: String

Required: No

**routeFilterPrefixes**

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

Type: array of [RouteFilterPrefix](#) (p. 82) objects

Required: No

**virtualGatewayId**

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

Type: String

Required: No

**virtualInterfaceId**

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Type: String

Required: No

**virtualInterfaceName**

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

Required: No

**virtualInterfaceState**

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.
- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: confirming | verifying | pending | available | deleting | deleted  
| rejected

Required: No

**virtualInterfaceType**

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

Type: String

Required: No

**vlan**

The VLAN ID.

Example: 101

Type: Number

Required: No

## AllocatePublicVirtualInterfaceResult

### Description

A virtual interface (VLAN) transmits the traffic between the AWS Direct Connect location and the customer.

### Contents

**amazonAddress**

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

Required: No

**asn**

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

Required: No

**authKey**

Authentication key for BGP configuration.

Example: asdf34example

Type: String

Required: No

**connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: No

**customerAddress**

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

Required: No

**customerRouterConfig**

Information for generating the customer router configuration.

Type: String

Required: No

**location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

Required: No

**ownerAccount**

Type: String

Required: No

**routeFilterPrefixes**

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

Type: array of [RouteFilterPrefix \(p. 82\)](#) objects

Required: No

**virtualGatewayId**

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

Type: String

Required: No

**virtualInterfaceId**

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Type: String

Required: No

**virtualInterfaceName**

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

Required: No

**virtualInterfaceState**

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.
- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: `confirming` | `verifying` | `pending` | `available` | `deleting` | `deleted` | `rejected`

Required: No

**virtualInterfaceType**

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

Type: String

Required: No

**vlan**

The VLAN ID.

Example: 101

Type: Number

Required: No

# ConfirmConnectionResult

## Description

The response received when ConfirmConnection is called.

## Contents

### connectionState

State of the connection.

- **Ordering:** The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
- **Requested:** The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending:** The connection has been approved, and is being initialized.
- **Available:** The network link is up, and the connection is ready for use.
- **Down:** The network link is down.
- **Deleted:** The connection has been deleted.
- **Rejected:** A hosted connection in the 'Ordering' state will enter the 'Rejected' state if it is deleted by the end customer.

Type: String

Valid Values: ordering | requested | pending | available | down | deleting | deleted | rejected

Required: No

# ConfirmPrivateVirtualInterfaceResult

## Description

The response received when ConfirmPrivateVirtualInterface is called.

## Contents

### virtualInterfaceState

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.

- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: `confirming` | `verifying` | `pending` | `available` | `deleting` | `deleted` | `rejected`

Required: No

## ConfirmPublicVirtualInterfaceResult

### Description

The response received when `ConfirmPublicVirtualInterface` is called.

### Contents

#### **virtualInterfaceState**

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.
- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: `confirming` | `verifying` | `pending` | `available` | `deleting` | `deleted` | `rejected`

Required: No



# Connection

## Description

A connection represents the physical network connection between the AWS Direct Connect location and the customer.

## Contents

### **bandwidth**

Bandwidth of the connection.

Example: 1Gbps (for regular connections), or 500Mbps (for hosted connections)

Default: None

Type: String

Required: No

### **connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: No

### **connectionName**

The name of the connection.

Example: *"My Connection to AWS"*

Default: None

Type: String

Required: No

### **connectionState**

State of the connection.

- **Ordering**: The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
- **Requested**: The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending**: The connection has been approved, and is being initialized.
- **Available**: The network link is up, and the connection is ready for use.
- **Down**: The network link is down.
- **Deleted**: The connection has been deleted.
- **Rejected**: A hosted connection in the 'Ordering' state will enter the 'Rejected' state if it is deleted by the end customer.

Type: String

Valid Values: ordering | requested | pending | available | down | deleting | deleted | rejected

Required: No

**location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

Required: No

**ownerAccount**

Type: String

Required: No

**partnerName**

Type: String

Required: No

**region**

The AWS region where the connection is located.

Example: us-east-1

Default: None

Type: String

Required: No

**vlan**

The VLAN ID.

Example: 101

Type: Number

Required: No

## CreateConnectionResult

### Description

A connection represents the physical network connection between the AWS Direct Connect location and the customer.

### Contents

**bandwidth**

Bandwidth of the connection.

Example: 1Gbps (for regular connections), or 500Mbps (for hosted connections)

Default: None

Type: String

Required: No

**connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: No

**connectionName**

The name of the connection.

Example: "My Connection to AWS"

Default: None

Type: String

Required: No

**connectionState**

State of the connection.

- **Ordering**: The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
- **Requested**: The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending**: The connection has been approved, and is being initialized.
- **Available**: The network link is up, and the connection is ready for use.
- **Down**: The network link is down.
- **Deleted**: The connection has been deleted.
- **Rejected**: A hosted connection in the 'Ordering' state will enter the 'Rejected' state if it is deleted by the end customer.

Type: String

Valid Values: `ordering | requested | pending | available | down | deleting | deleted | rejected`

Required: No

**location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

Required: No

**ownerAccount**

Type: String

Required: No

**partnerName**

Type: String

Required: No

**region**

The AWS region where the connection is located.

Example: us-east-1

Default: None

Type: String

Required: No

**vlan**

The VLAN ID.

Example: 101

Type: Number

Required: No

## CreateInterconnectResult

### Description

An interconnect is a connection that can host other connections.

Like a standard AWS Direct Connect connection, an interconnect represents the physical connection between an AWS Direct Connect partner's network and a specific Direct Connect location. An AWS Direct Connect partner who owns an interconnect can provision hosted connections on the interconnect for their end customers, thereby providing the end customers with connectivity to AWS services.

The resources of the interconnect, including bandwidth and VLAN numbers, are shared by all of the hosted connections on the interconnect, and the owner of the interconnect determines how these resources are assigned.

### Contents

**bandwidth**

Bandwidth of the connection.

Example: 1Gbps

Default: None

Type: String

Required: No

**interconnectId**

The ID of the interconnect.

Example: dxcon-abc123

Type: String

Required: No

**interconnectName**

The name of the interconnect.

Example: "1G Interconnect to AWS"

Type: String

Required: No

**interconnectState**

State of the interconnect.

- **Requested:** The initial state of an interconnect. The interconnect stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending:** The interconnect has been approved, and is being initialized.
- **Available:** The network link is up, and the interconnect is ready for use.
- **Down:** The network link is down.
- **Deleted:** The interconnect has been deleted.

Type: String

Valid Values: `requested` | `pending` | `available` | `down` | `deleting` | `deleted`

Required: No

**location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

Required: No

**region**

The AWS region where the connection is located.

Example: us-east-1

Default: None

Type: String

Required: No

## CreatePrivateVirtualInterfaceResult

### Description

A virtual interface (VLAN) transmits the traffic between the AWS Direct Connect location and the customer.

### Contents

**amazonAddress**

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

Required: No

**asn**

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

Required: No

**authKey**

Authentication key for BGP configuration.

Example: asdf34example

Type: String

Required: No

**connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: No

**customerAddress**

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

Required: No

**customerRouterConfig**

Information for generating the customer router configuration.

Type: String

Required: No

**location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

Required: No

**ownerAccount**

Type: String

Required: No

**routeFilterPrefixes**

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

Type: array of [RouteFilterPrefix](#) (p. 82) objects

Required: No

**virtualGatewayId**

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

Type: String

Required: No

**virtualInterfaceId**

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Type: String

Required: No

**virtualInterfaceName**

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

Required: No

**virtualInterfaceState**

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.
- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: `confirming` | `verifying` | `pending` | `available` | `deleting` | `deleted` | `rejected`

Required: No

**virtualInterfaceType**

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

Type: String

Required: No

**vlan**

The VLAN ID.

Example: 101

Type: Number

Required: No

## CreatePublicVirtualInterfaceResult

### Description

A virtual interface (VLAN) transmits the traffic between the AWS Direct Connect location and the customer.

### Contents

**amazonAddress**

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

Required: No

**asn**

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

Required: No

**authKey**

Authentication key for BGP configuration.

Example: asdf34example

Type: String

Required: No

**connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None



Type: String

Required: No

**customerAddress**

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

Required: No

**customerRouterConfig**

Information for generating the customer router configuration.

Type: String

Required: No

**location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

Required: No

**ownerAccount**

Type: String

Required: No

**routeFilterPrefixes**

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

Type: array of [RouteFilterPrefix](#) (p. 82) objects

Required: No

**virtualGatewayId**

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

Type: String

Required: No

**virtualInterfaceId**

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Type: String

Required: No

**virtualInterfaceName**

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

Required: No

**virtualInterfaceState**

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.
- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: `confirming` | `verifying` | `pending` | `available` | `deleting` | `deleted` | `rejected`

Required: No

**virtualInterfaceType**

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

Type: String

Required: No

**vlan**

The VLAN ID.

Example: 101

Type: Number

Required: No

## DeleteConnectionResult

### Description

A connection represents the physical network connection between the AWS Direct Connect location and the customer.

## Contents

### **bandwidth**

Bandwidth of the connection.

Example: 1Gbps (for regular connections), or 500Mbps (for hosted connections)

Default: None

Type: String

Required: No

### **connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: No

### **connectionName**

The name of the connection.

Example: *"My Connection to AWS"*

Default: None

Type: String

Required: No

### **connectionState**

State of the connection.

- **Ordering**: The initial state of a hosted connection provisioned on an interconnect. The connection stays in the ordering state until the owner of the hosted connection confirms or declines the connection order.
- **Requested**: The initial state of a standard connection. The connection stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending**: The connection has been approved, and is being initialized.
- **Available**: The network link is up, and the connection is ready for use.
- **Down**: The network link is down.
- **Deleted**: The connection has been deleted.
- **Rejected**: A hosted connection in the 'Ordering' state will enter the 'Rejected' state if it is deleted by the end customer.

Type: String

Valid Values: `ordering` | `requested` | `pending` | `available` | `down` | `deleting` | `deleted` | `rejected`

Required: No

### **location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

Required: No

**ownerAccount**

Type: String

Required: No

**partnerName**

Type: String

Required: No

**region**

The AWS region where the connection is located.

Example: us-east-1

Default: None

Type: String

Required: No

**vlan**

The VLAN ID.

Example: 101

Type: Number

Required: No

## DeleteInterconnectResult

### Description

The response received when DeleteInterconnect is called.

### Contents

**interconnectState**

State of the interconnect.

- **Requested:** The initial state of an interconnect. The interconnect stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending:** The interconnect has been approved, and is being initialized.
- **Available:** The network link is up, and the interconnect is ready for use.
- **Down:** The network link is down.
- **Deleted:** The interconnect has been deleted.

Type: String

Valid Values: `requested` | `pending` | `available` | `down` | `deleting` | `deleted`

Required: No

# DeleteVirtualInterfaceResult

## Description

The response received when DeleteVirtualInterface is called.

## Contents

### virtualInterfaceState

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.
- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String

Valid Values: `confirming` | `verifying` | `pending` | `available` | `deleting` | `deleted` | `rejected`

Required: No

# DescribeConnectionsOnInterconnectResult

## Description

A structure containing a list of connections.

## Contents

### connections

A list of connections.

Type: array of [Connection](#) (p. 62) objects

Required: No

## DescribeConnectionsResult

### Description

A structure containing a list of connections.

### Contents

**connections**

A list of connections.

Type: array of [Connection](#) (p. 62) objects

Required: No

## DescribeInterconnectsResult

### Description

A structure containing a list of interconnects.

### Contents

**interconnects**

A list of interconnects.

Type: array of [Interconnect](#) (p. 76) objects

Required: No

## DescribeLocationsResult

### Description

No action documentation available.

### Contents

**locations**

Type: array of [Location](#) (p. 78) objects

Required: No

# DescribeVirtualGatewaysResult

## Description

A structure containing a list of virtual private gateways.

## Contents

### **virtualGateways**

A list of virtual private gateways.

Type: array of [VirtualGateway](#) (p. 83) objects

Required: No

# DescribeVirtualInterfacesResult

## Description

A structure containing a list of virtual interfaces.

## Contents

### **virtualInterfaces**

A list of virtual interfaces.

Type: array of [VirtualInterface](#) (p. 83) objects

Required: No

# Interconnect

## Description

An interconnect is a connection that can host other connections.

Like a standard AWS Direct Connect connection, an interconnect represents the physical connection between an AWS Direct Connect partner's network and a specific Direct Connect location. An AWS Direct Connect partner who owns an interconnect can provision hosted connections on the interconnect for their end customers, thereby providing the end customers with connectivity to AWS services.

The resources of the interconnect, including bandwidth and VLAN numbers, are shared by all of the hosted connections on the interconnect, and the owner of the interconnect determines how these resources are assigned.

## Contents

### **bandwidth**

Bandwidth of the connection.

Example: 1Gbps

Default: None

Type: String

Required: No

**interconnectId**

The ID of the interconnect.

Example: dxcon-abc123

Type: String

Required: No

**interconnectName**

The name of the interconnect.

Example: "1G Interconnect to AWS"

Type: String

Required: No

**interconnectState**

State of the interconnect.

- **Requested:** The initial state of an interconnect. The interconnect stays in the requested state until the Letter of Authorization (LOA) is sent to the customer.
- **Pending:** The interconnect has been approved, and is being initialized.
- **Available:** The network link is up, and the interconnect is ready for use.
- **Down:** The network link is down.
- **Deleted:** The interconnect has been deleted.

Type: String

Valid Values: requested | pending | available | down | deleting | deleted

Required: No

**location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

Required: No

**region**

The AWS region where the connection is located.

Example: us-east-1

Default: None

Type: String

Required: No



# Location

## Description

An AWS Direct Connect location where connections and interconnects can be requested.

## Contents

### **locationCode**

The code used to indicate the AWS Direct Connect location.

Type: String

Required: No

### **locationName**

The name of the AWS Direct Connect location. The name includes the colocation partner name and the physical site of the lit building.

Type: String

Required: No

# NewPrivateVirtualInterface

## Description

A structure containing information about a new private virtual interface.

## Contents

### **amazonAddress**

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

Required: No

### **asn**

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

Required: Yes

### **authKey**

Authentication key for BGP configuration.

Example: asdf34example

Type: String

Required: No

**customerAddress**

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

Required: No

**virtualGatewayId**

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

Type: String

Required: Yes

**virtualInterfaceName**

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

Required: Yes

**vlan**

The VLAN ID.

Example: 101

Type: Number

Required: Yes

## NewPrivateVirtualInterfaceAllocation

### Description

A structure containing information about a private virtual interface that will be provisioned on a connection.

### Contents

**amazonAddress**

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

Required: No

**asn**

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

Required: Yes

**authKey**

Authentication key for BGP configuration.

Example: asdf34example

Type: String

Required: No

**customerAddress**

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

Required: No

**virtualInterfaceName**

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

Required: Yes

**vlan**

The VLAN ID.

Example: 101

Type: Number

Required: Yes

## NewPublicVirtualInterface

### Description

A structure containing information about a new public virtual interface.

### Contents

**amazonAddress**

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

Required: Yes

**asn**

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

Required: Yes

**authKey**

Authentication key for BGP configuration.

Example: asdf34example

Type: String

Required: No

**customerAddress**

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

Required: Yes

**routeFilterPrefixes**

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

Type: array of [RouteFilterPrefix \(p. 82\)](#) objects

Required: Yes

**virtualInterfaceName**

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

Required: Yes

**vlan**

The VLAN ID.

Example: 101

Type: Number

Required: Yes

## NewPublicVirtualInterfaceAllocation

### Description

A structure containing information about a public virtual interface that will be provisioned on a connection.

### Contents

**amazonAddress**

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

Required: Yes

**asn**

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

Required: Yes

**authKey**

Authentication key for BGP configuration.

Example: asdf34example

Type: String

Required: No

**customerAddress**

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

Required: Yes

**routeFilterPrefixes**

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

Type: array of [RouteFilterPrefix \(p. 82\)](#) objects

Required: Yes

**virtualInterfaceName**

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

Required: Yes

**vlan**

The VLAN ID.

Example: 101

Type: Number

Required: Yes

## RouteFilterPrefix

### Description

A route filter prefix that the customer can advertise through Border Gateway Protocol (BGP) over a public virtual interface.

## Contents

### **cidr**

CIDR notation for the advertised route. Multiple routes are separated by commas.

Example: 10.10.10.0/24,10.10.11.0/24

Type: String

Required: No

## VirtualGateway

### Description

You can create one or more AWS Direct Connect private virtual interfaces linking to your virtual private gateway.

Virtual private gateways can be managed using the Amazon Virtual Private Cloud (Amazon VPC) console or the [Amazon EC2 CreateVpnGateway action](#).

### Contents

#### **virtualGatewayId**

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

Type: String

Required: No

#### **virtualGatewayState**

State of the virtual private gateway.

- **Pending:** This is the initial state after calling *CreateVpnGateway*.
- **Available:** Ready for use by a private virtual interface.
- **Deleting:** This is the initial state after calling *DeleteVpnGateway*.
- **Deleted:** In this state, a private virtual interface is unable to send traffic over this gateway.

Type: String

Required: No

## VirtualInterface

### Description

A virtual interface (VLAN) transmits the traffic between the AWS Direct Connect location and the customer.

## Contents

### **amazonAddress**

IP address assigned to the Amazon interface.

Example: 192.168.1.1/30

Type: String

Required: No

### **asn**

Autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.

Example: 65000

Type: Number

Required: No

### **authKey**

Authentication key for BGP configuration.

Example: asdf34example

Type: String

Required: No

### **connectionId**

ID of the connection.

Example: dxcon-fg5678gh

Default: None

Type: String

Required: No

### **customerAddress**

IP address assigned to the customer interface.

Example: 192.168.1.2/30

Type: String

Required: No

### **customerRouterConfig**

Information for generating the customer router configuration.

Type: String

Required: No

### **location**

Where the connection is located.

Example: EqSV5

Default: None

Type: String

Required: No

**ownerAccount**

Type: String

Required: No

**routeFilterPrefixes**

A list of routes to be advertised to the AWS network in this region (public virtual interface) or your VPC (private virtual interface).

Type: array of [RouteFilterPrefix](#) (p. 82) objects

Required: No

**virtualGatewayId**

The ID of the virtual private gateway to a VPC. This only applies to private virtual interfaces.

Example: vgw-123er56

Type: String

Required: No

**virtualInterfaceId**

ID of the virtual interface.

Example: dxvif-123dfg56

Default: None

Type: String

Required: No

**virtualInterfaceName**

The name of the virtual interface assigned by the customer.

Example: "My VPC"

Type: String

Required: No

**virtualInterfaceState**

State of the virtual interface.

- **Confirming:** The creation of the virtual interface is pending confirmation from the virtual interface owner. If the owner of the virtual interface is different from the owner of the connection on which it is provisioned, then the virtual interface will remain in this state until it is confirmed by the virtual interface owner.
- **Verifying:** This state only applies to public virtual interfaces. Each public virtual interface needs validation before the virtual interface can be created.
- **Pending:** A virtual interface is in this state from the time that it is created until the virtual interface is ready to forward traffic.
- **Available:** A virtual interface that is able to forward traffic.
- **Deleting:** A virtual interface is in this state immediately after calling *DeleteVirtualInterface* until it can no longer forward traffic.
- **Deleted:** A virtual interface that cannot forward traffic.
- **Rejected:** The virtual interface owner has declined creation of the virtual interface. If a virtual interface in the 'Confirming' state is deleted by the virtual interface owner, the virtual interface will enter the 'Rejected' state.

Type: String



Valid Values: confirming | verifying | pending | available | deleting | deleted  
| rejected

Required: No

**virtualInterfaceType**

The type of virtual interface.

Example: private (Amazon VPC) or public (Amazon S3, Amazon DynamoDB, and so on.)

Type: String

Required: No

**vlan**

The VLAN ID.

Example: 101

Type: Number

Required: No

# Common Parameters

---

This section lists the request parameters that all actions use. Any action-specific parameters are listed in the topic for the action.

## Action

The action to be performed.

Default: None

Type: string

Required: Yes

## AuthParams

The parameters that are required to authenticate a Conditional request. Contains:

- AWSAccessKeyID
- SignatureVersion
- Timestamp
- Signature

Default: None

Required: Conditional

## AWSAccessKeyID

The access key ID that corresponds to the secret access key that you used to sign the request.

Default: None

Type: string

Required: Yes

## Expires

The date and time when the request signature expires, expressed in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard.

Condition: Requests must include either *Timestamp* or *Expires*, but not both.

Default: None

Type: string

Required: Conditional

**SecurityToken**

The temporary security token that was obtained through a call to AWS Security Token Service. For a list of services that support AWS Security Token Service, go to [Using Temporary Security Credentials to Access AWS](#) in **Using Temporary Security Credentials**.

Default: None

Type: string

Required: No

**Signature**

The digital signature that you created for the request. For information about generating a signature, go to the service's developer documentation.

Default: None

Type: string

Required: Yes

**SignatureMethod**

The hash algorithm that you used to create the request signature.

Default: None

Type: string

Valid Values: HmacSHA256 | HmacSHA1

Required: Yes

**SignatureVersion**

The signature version you use to sign the request. Set this to the value that is recommended for your service.

Default: None

Type: string

Required: Yes

**Timestamp**

The date and time when the request was signed, expressed in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard.

Condition: Requests must include either *Timestamp* or *Expires*, but not both.

Default: None

Type: string

Required: Conditional

**Version**

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Default: None

Type: string

Required: Yes

# Common Errors

---

This section lists the common errors that all actions return. Any action-specific errors are listed in the topic for the action.

**IncompleteSignature**

The request signature does not conform to AWS standards.

HTTP Status Code: 400

**InternalFailure**

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

**InvalidAction**

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

**InvalidClientTokenId**

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

**InvalidParameterCombination**

Parameters that must not be used together were used together.

HTTP Status Code: 400

**InvalidParameterValue**

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

**InvalidQueryParameter**

AWS query string is malformed, does not adhere to AWS standards.

HTTP Status Code: 400

**MalformedQueryString**

The query string contains a syntax error.

HTTP Status Code: 404

**MissingAction**

The request is missing an action or a required parameter.

HTTP Status Code: 400

**MissingAuthenticationToken**

Request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

**MissingParameter**

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

**OptInRequired**

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

**RequestExpired**

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

**ServiceUnavailable**

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

**Throttling**

Request was denied due to request throttling.

HTTP Status Code: 400