Errors ............................................................................................................................ 102
See Also ......................................................................................................................... 103
CreateTransitVirtualInterface ......................................................................................... 104
Request Syntax ................................................................. 104
Request Parameters ................................................................. 104
Response Syntax ................................................................. 105
Response Elements ................................................................. 106
Errors ............................................................................................................................ 106
See Also ......................................................................................................................... 106
DeleteBGPPeer .............................................................................................................. 107
Request Syntax ................................................................. 107
Request Parameters ................................................................. 107
Response Syntax ................................................................. 107
Response Elements ................................................................. 108
Errors ............................................................................................................................ 109
See Also ......................................................................................................................... 109
DeleteConnection .......................................................................................................... 110
Request Syntax ................................................................. 110
Request Parameters ................................................................. 110
Response Syntax ................................................................. 110
Response Elements ................................................................. 111
Errors ............................................................................................................................ 113
See Also ......................................................................................................................... 114
DeleteDirectConnectGateway ......................................................................................... 115
Request Syntax ................................................................. 115
Request Parameters ................................................................. 115
Response Syntax ................................................................. 115
Response Elements ................................................................. 115
Errors ............................................................................................................................ 115
See Also ......................................................................................................................... 116
DeleteDirectConnectGatewayAssociation .................................................................... 117
Request Syntax ................................................................. 117
Request Parameters ................................................................. 117
Response Syntax ................................................................. 117
Response Elements ................................................................. 118
Errors ............................................................................................................................ 118
See Also ......................................................................................................................... 118
DeleteDirectConnectGatewayAssociationProposal .................................................. 120
Request Syntax ................................................................. 120
Request Parameters ................................................................. 120
Response Syntax ................................................................. 120
Response Elements ................................................................. 120
Errors ............................................................................................................................ 121
See Also ......................................................................................................................... 121
DeleteInterconnect ........................................................................................................ 122
Request Syntax ................................................................. 122
Request Parameters ................................................................. 122
Response Syntax ................................................................. 122
Response Elements ................................................................. 122
Errors ............................................................................................................................ 123
See Also ......................................................................................................................... 123
DeleteLag ....................................................................................................................... 124
Request Syntax ................................................................. 124
Request Parameters ................................................................. 124
Response Syntax ................................................................. 124
Response Elements ................................................................. 125
Errors ............................................................................................................................ 128
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DescribeHostedConnections</td>
<td>151</td>
</tr>
<tr>
<td>Request Syntax</td>
<td>151</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>154</td>
</tr>
<tr>
<td>Response Syntax</td>
<td>154</td>
</tr>
<tr>
<td>Response Elements</td>
<td>154</td>
</tr>
<tr>
<td>Errors</td>
<td>155</td>
</tr>
<tr>
<td>See Also</td>
<td>155</td>
</tr>
<tr>
<td>DescribeInterconnectLoa</td>
<td>154</td>
</tr>
<tr>
<td>Request Syntax</td>
<td>154</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>154</td>
</tr>
<tr>
<td>Response Syntax</td>
<td>154</td>
</tr>
<tr>
<td>Response Elements</td>
<td>155</td>
</tr>
<tr>
<td>Errors</td>
<td>155</td>
</tr>
<tr>
<td>See Also</td>
<td>155</td>
</tr>
<tr>
<td>DescribeInterconnects</td>
<td>156</td>
</tr>
<tr>
<td>Request Syntax</td>
<td>156</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>156</td>
</tr>
<tr>
<td>Response Syntax</td>
<td>156</td>
</tr>
<tr>
<td>Response Elements</td>
<td>156</td>
</tr>
<tr>
<td>Errors</td>
<td>157</td>
</tr>
<tr>
<td>See Also</td>
<td>157</td>
</tr>
<tr>
<td>DescribeLags</td>
<td>158</td>
</tr>
<tr>
<td>Request Syntax</td>
<td>158</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>158</td>
</tr>
<tr>
<td>Response Syntax</td>
<td>158</td>
</tr>
<tr>
<td>Response Elements</td>
<td>159</td>
</tr>
<tr>
<td>Errors</td>
<td>159</td>
</tr>
<tr>
<td>See Also</td>
<td>160</td>
</tr>
<tr>
<td>DescribeLoa</td>
<td>161</td>
</tr>
<tr>
<td>Request Syntax</td>
<td>161</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>161</td>
</tr>
<tr>
<td>Response Syntax</td>
<td>161</td>
</tr>
<tr>
<td>Response Elements</td>
<td>162</td>
</tr>
<tr>
<td>Errors</td>
<td>162</td>
</tr>
<tr>
<td>See Also</td>
<td>162</td>
</tr>
<tr>
<td>DescribeLocations</td>
<td>163</td>
</tr>
<tr>
<td>Request Syntax</td>
<td>163</td>
</tr>
<tr>
<td>Response Syntax</td>
<td>163</td>
</tr>
<tr>
<td>Response Elements</td>
<td>163</td>
</tr>
<tr>
<td>Errors</td>
<td>163</td>
</tr>
<tr>
<td>See Also</td>
<td>163</td>
</tr>
<tr>
<td>DescribeRouterConfiguration</td>
<td>165</td>
</tr>
<tr>
<td>Request Syntax</td>
<td>165</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>165</td>
</tr>
<tr>
<td>Response Syntax</td>
<td>165</td>
</tr>
<tr>
<td>Response Elements</td>
<td>165</td>
</tr>
<tr>
<td>Errors</td>
<td>166</td>
</tr>
<tr>
<td>See Also</td>
<td>166</td>
</tr>
<tr>
<td>DescribeTags</td>
<td>167</td>
</tr>
<tr>
<td>Request Syntax</td>
<td>167</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>167</td>
</tr>
<tr>
<td>Response Syntax</td>
<td>167</td>
</tr>
<tr>
<td>Response Elements</td>
<td>167</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>Response Elements</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Request Syntax</td>
<td>Request Parameters</td>
</tr>
<tr>
<td>Response Syntax</td>
<td>Response Elements</td>
</tr>
<tr>
<td>UpdateConnection</td>
<td>Request Syntax</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>Response Elements</td>
</tr>
<tr>
<td>UpdateDirectConnectGateway</td>
<td>Request Syntax</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>Response Elements</td>
</tr>
<tr>
<td>UpdateDirectConnectGatewayAssociation</td>
<td>Request Syntax</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>Response Elements</td>
</tr>
<tr>
<td>UpdateLag</td>
<td>Request Syntax</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>Response Elements</td>
</tr>
<tr>
<td>UpdateVirtualInterfaceAttributes</td>
<td>Request Syntax</td>
</tr>
<tr>
<td>Request Parameters</td>
<td>Response Elements</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssociatedGateway</td>
</tr>
<tr>
<td>BGPPeer</td>
</tr>
<tr>
<td>Connection</td>
</tr>
<tr>
<td>CustomerAgreement</td>
</tr>
<tr>
<td>DirectConnectGateway</td>
</tr>
<tr>
<td>DirectConnectGatewayAssociation</td>
</tr>
<tr>
<td>Tag</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>RouterType</td>
</tr>
<tr>
<td>NewTransitVirtualInterface</td>
</tr>
<tr>
<td>NewPublicVirtualInterfaceAllocation</td>
</tr>
<tr>
<td>NewPrivateVirtualInterfaceAllocation</td>
</tr>
<tr>
<td>NewPrivateVirtualInterface</td>
</tr>
<tr>
<td>NewPublicVirtualInterface</td>
</tr>
<tr>
<td>NewPublicVirtualInterfaceAllocation</td>
</tr>
<tr>
<td>NewTransitVirtualInterface</td>
</tr>
<tr>
<td>NewTransitVirtualInterfaceAllocation</td>
</tr>
<tr>
<td>ResourceTag</td>
</tr>
<tr>
<td>RouteFilterPrefix</td>
</tr>
<tr>
<td>RouterType</td>
</tr>
<tr>
<td>Tag</td>
</tr>
<tr>
<td>VirtualGateway</td>
</tr>
</tbody>
</table>

---

**AWS Direct Connect API Reference**

API Version 2012-10-25

xii
Welcome

AWS Direct Connect links your internal network to an AWS Direct Connect location over a standard Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to an AWS Direct Connect router. With this connection in place, you can create virtual interfaces directly to the AWS Cloud (for example, to Amazon EC2 and Amazon S3) and to Amazon VPC, bypassing Internet service providers in your network path. A connection provides access to all AWS Regions except the China (Beijing) and (China) Ningxia Regions. AWS resources in the China Regions can only be accessed through locations associated with those Regions.

This document was last published on April 15, 2022.
Actions

The following actions are supported:

- AcceptDirectConnectGatewayAssociationProposal (p. 4)
- AllocateConnectionOnInterconnect (p. 7)
- AllocateHostedConnection (p. 12)
- AllocatePrivateVirtualInterface (p. 18)
- AllocatePublicVirtualInterface (p. 24)
- AllocateTransitVirtualInterface (p. 30)
- AssociateConnectionWithLag (p. 33)
- AssociateHostedConnection (p. 38)
- AssociateMacSecKey (p. 43)
- AssociateVirtualInterface (p. 46)
- ConfirmConnection (p. 52)
- ConfirmCustomerAgreement (p. 54)
- ConfirmPrivateVirtualInterface (p. 56)
- ConfirmPublicVirtualInterface (p. 59)
- ConfirmTransitVirtualInterface (p. 61)
- CreateBGPPeer (p. 63)
- CreateConnection (p. 66)
- CreateDirectConnectGateway (p. 72)
- CreateDirectConnectGatewayAssociation (p. 74)
- CreateDirectConnectGatewayAssociationProposal (p. 77)
- CreateInterconnect (p. 80)
- CreateLag (p. 85)
- CreatePrivateVirtualInterface (p. 92)
- CreatePublicVirtualInterface (p. 98)
- CreateTransitVirtualInterface (p. 104)
- DeleteBGPPeer (p. 107)
- DeleteConnection (p. 110)
- DeleteDirectConnectGateway (p. 115)
- DeleteDirectConnectGatewayAssociation (p. 117)
- DeleteDirectConnectGatewayAssociationProposal (p. 120)
- DeleteInterconnect (p. 122)
- DeleteLag (p. 124)
- DeleteVirtualInterface (p. 129)
- DescribeConnectionLoa (p. 131)
- DescribeConnections (p. 133)
- DescribeConnectionsOnInterconnect (p. 135)
- DescribeCustomerMetadata (p. 138)
- DescribeDirectConnectGatewayAssociationProposals (p. 140)
- DescribeDirectConnectGatewayAssociations (p. 143)
- DescribeDirectConnectGatewayAttachments (p. 146)
• DescribeDirectConnectGateways (p. 149)
• DescribeHostedConnections (p. 151)
• DescribeInterconnectLoa (p. 154)
• DescribeInterconnects (p. 156)
• DescribeLags (p. 158)
• DescribeLoa (p. 161)
• DescribeLocations (p. 163)
• DescribeRouterConfiguration (p. 165)
• DescribeTags (p. 167)
• DescribeVirtualGateways (p. 169)
• DescribeVirtualInterfaces (p. 171)
• DisassociateConnectionFromLag (p. 174)
• DisassociateMacSecKey (p. 179)
• ListVirtualInterfaceTestHistory (p. 181)
• StartBgpFailoverTest (p. 184)
• StopBgpFailoverTest (p. 186)
• TagResource (p. 188)
• UntagResource (p. 190)
• UpdateConnection (p. 192)
• UpdateDirectConnectGateway (p. 197)
• UpdateDirectConnectGatewayAssociation (p. 199)
• UpdateLag (p. 202)
• UpdateVirtualInterfaceAttributes (p. 208)
AcceptDirectConnectGatewayAssociationProposal

Accepts a proposal request to attach a virtual private gateway or transit gateway to a Direct Connect gateway.

Request Syntax

```json
{
  "associatedGatewayOwnerAccount": "string",
  "directConnectGatewayId": "string",
  "overrideAllowedPrefixesToDirectConnectGateway": [ 
    { 
      "cidr": "string"
    }
  ],
  "proposalId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**associatedGatewayOwnerAccount (p. 4)**

The ID of the AWS account that owns the virtual private gateway or transit gateway.

Type: String

Required: Yes

**directConnectGatewayId (p. 4)**

The ID of the Direct Connect gateway.

Type: String

Required: Yes

**overrideAllowedPrefixesToDirectConnectGateway (p. 4)**

Overrides the Amazon VPC prefixes advertised to the Direct Connect gateway.

For information about how to set the prefixes, see Allowed Prefixes in the AWS Direct Connect User Guide.

Type: Array of RouteFilterPrefix (p. 257) objects

Required: No

**proposalId (p. 4)**

The ID of the request proposal.

Type: String

Required: Yes
Response Syntax

```json
{
    "directConnectGatewayAssociation": {
        "allowedPrefixesToDirectConnectGateway": [
            {
                "cidr": "string"
            }
        ],
        "associatedGateway": {
            "id": "string",
            "ownerAccount": "string",
            "region": "string",
            "type": "string"
        },
        "associationId": "string",
        "associationState": "string",
        "directConnectGatewayId": "string",
        "directConnectGatewayOwnerAccount": "string",
        "stateChangeError": "string",
        "virtualGatewayId": "string",
        "virtualGatewayOwnerAccount": "string",
        "virtualGatewayRegion": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**directConnectGatewayAssociation (p. 5)**

Information about an association between a Direct Connect gateway and a virtual private gateway or transit gateway.

Type: `DirectConnectGatewayAssociation (p. 225)` object

Errors

For information about the errors that are common to all actions, see [Common Errors (p. 270)](https://aws.amazon.com/documentation/).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:
• AWS Command Line Interface
• AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java V2
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V3
AllocateConnectionOnInterconnect

Deprecated. Use AllocateHostedConnection (p. 12) instead.

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 specified interconnect.

**Note**
Intended for use by AWS Direct Connect Partners only.

**Request Syntax**

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

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

- **bandwidth (p. 7)**
  
  The bandwidth of the connection. The possible values are 50Mbps, 100Mbps, 200Mbps, 300Mbps, 400Mbps, 500Mbps, 1Gbps, 2Gbps, 5Gbps, and 10Gbps. Note that only those AWS Direct Connect Partners who have met specific requirements are allowed to create a 1Gbps, 2Gbps, 5Gbps or 10Gbps hosted connection.

  Type: String
  
  Required: Yes

- **connectionName (p. 7)**
  
  The name of the provisioned connection.

  Type: String
  
  Required: Yes

- **interconnectId (p. 7)**
  
  The ID of the interconnect on which the connection will be provisioned.

  Type: String
  
  Required: Yes

- **ownerAccount (p. 7)**
  
  The ID of the AWS account of the customer for whom the connection will be provisioned.

  Type: String
Required: Yes

**vlan (p. 7)**

The dedicated VLAN provisioned to the connection.

Type: Integer

Required: Yes

---

**Response Syntax**

```
{
    "awsDevice": "string",
    "awsDeviceV2": "string",
    "awsLogicalDeviceId": "string",
    "bandwidth": "string",
    "connectionId": "string",
    "connectionName": "string",
    "connectionState": "string",
    "encryptionMode": "string",
    "hasLogicalRedundancy": "string",
    "jumboFrameCapable": boolean,
    "lagId": "string",
    "loaIssueTime": number,
    "location": "string",
    "macSecCapable": boolean,
    "macSecKeys": [
        {
            "ckn": "string",
            "secretARN": "string",
            "startOn": "string",
            "state": "string"
        }
    ],
    "ownerAccount": "string",
    "partnerName": "string",
    "portEncryptionStatus": "string",
    "providerName": "string",
    "region": "string",
    "tags": [
        {
            "key": "string",
            "value": "string"
        }
    ],
    "vlan": number
}
```

---

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

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

**awsDevice (p. 8)**

*This parameter has been deprecated.*

The Direct Connect endpoint on which the physical connection terminates.
Type: String

**awsDeviceV2 (p. 8)**

The Direct Connect endpoint that terminates the physical connection.

Type: String

**awsLogicalDeviceId (p. 8)**

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

**bandwidth (p. 8)**

The bandwidth of the connection.

Type: String

**connectionId (p. 8)**

The ID of the connection.

Type: String

**connectionName (p. 8)**

The name of the connection.

Type: String

**connectionState (p. 8)**

The state of the connection. The following are the possible values:

- **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.
- **deleting**: The connection is being deleted.
- **deleted**: The connection has been deleted.
- **rejected**: A hosted connection in the ordering state enters the rejected state if it is deleted by the customer.
- **unknown**: The state of the connection is not available.

Type: String

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

**encryptionMode (p. 8)**

The MAC Security (MACsec) connection encryption mode.

The valid values are `no_encrypt`, `should_encrypt`, and `must_encrypt`.

Type: String
**hasLogicalRedundancy (p. 8)**

Indicates whether the connection supports a secondary BGP peer in the same address family (IPv4/IPv6).

Type: String

Valid Values: unknown | yes | no

**jumboFrameCapable (p. 8)**

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean

**lagId (p. 8)**

The ID of the LAG.

Type: String

**loaIssueTime (p. 8)**

The time of the most recent call to DescribeLoa (p. 161) for this connection.

Type: Timestamp

**location (p. 8)**

The location of the connection.

Type: String

**macSecCapable (p. 8)**

Indicates whether the connection supports MAC Security (MACsec).

Type: Boolean

**macSecKeys (p. 8)**

The MAC Security (MACsec) security keys associated with the connection.

Type: Array of MacSecKey (p. 241) objects

**ownerAccount (p. 8)**

The ID of the AWS account that owns the connection.

Type: String

**partnerName (p. 8)**

The name of the AWS Direct Connect service provider associated with the connection.

Type: String

**portEncryptionStatus (p. 8)**

The MAC Security (MACsec) port link status of the connection.

The valid values are Encryption Up, which means that there is an active Connection Key Name, or Encryption Down.

Type: String

**providerName (p. 8)**

The name of the service provider associated with the connection.
Type: String

**region (p. 8)**

The AWS Region where the connection is located.

Type: String

**tags (p. 8)**

The tags associated with the connection.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

**vlan (p. 8)**

The ID of the VLAN.

Type: Integer

## Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
AllocateHostedConnection

Creates a hosted connection on the specified interconnect or a link aggregation group (LAG) of interconnects.

Allocates a VLAN number and a specified amount of capacity (bandwidth) for use by a hosted connection on the specified interconnect or LAG of interconnects. AWS polices the hosted connection for the specified capacity and the AWS Direct Connect Partner must also police the hosted connection for the specified capacity.

**Note**
Intended for use by AWS Direct Connect Partners only.

**Request Syntax**

```
{
  "bandwidth": "string",
  "connectionId": "string",
  "connectionName": "string",
  "ownerAccount": "string",
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "vlan": number
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**bandwidth (p. 12)**

The bandwidth of the connection. The possible values are 50Mbps, 100Mbps, 200Mbps, 300Mbps, 400Mbps, 500Mbps, 1Gbps, 2Gbps, 5Gbps, and 10Gbps. Note that only those AWS Direct Connect Partners who have met specific requirements are allowed to create a 1Gbps, 2Gbps, 5Gbps or 10Gbps hosted connection.

Type: String

Required: Yes

**connectionId (p. 12)**

The ID of the interconnect or LAG.

Type: String

Required: Yes

**connectionName (p. 12)**

The name of the hosted connection.

Type: String

Type: String

Required: Yes
Required: Yes

**ownerAccount (p. 12)**

The ID of the AWS account ID of the customer for the connection.

Type: String

Required: Yes

**tags (p. 12)**

The tags associated with the connection.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

Required: No

**vlan (p. 12)**

The dedicated VLAN provisioned to the hosted connection.

Type: Integer

Required: Yes

---

## Response Syntax

```
{
    "awsDevice": "string",
    "awsDeviceV2": "string",
    "awsLogicalDeviceId": "string",
    "bandwidth": "string",
    "connectionId": "string",
    "connectionName": "string",
    "connectionState": "string",
    "encryptionMode": "string",
    "hasLogicalRedundancy": "string",
    "jumboFrameCapable": boolean,
    "lagId": "string",
    "loaIssueTime": number,
    "location": "string",
    "macSecCapable": boolean,
    "macSecKeys": [
        {
            "ckn": "string",
            "secretARN": "string",
            "startOn": "string",
            "state": "string"
        }
    ],
    "ownerAccount": "string",
    "partnerName": "string",
    "portEncryptionStatus": "string",
    "providerName": "string",
    "region": "string",
    "tags": [
        {
            "key": "string",
            "value": "string"
        }
    ],
}
```

---

API Version 2012-10-25

13
"vlan": number
}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**awsDevice (p. 13)**

*This parameter has been deprecated.*

The Direct Connect endpoint on which the physical connection terminates.

Type: String

**awsDeviceV2 (p. 13)**

The Direct Connect endpoint that terminates the physical connection.

Type: String

**awsLogicalDeviceId (p. 13)**

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

**bandwidth (p. 13)**

The bandwidth of the connection.

Type: String

**connectionId (p. 13)**

The ID of the connection.

Type: String

**connectionName (p. 13)**

The name of the connection.

Type: String

**connectionState (p. 13)**

The state of the connection. The following are the possible values:

- **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.
- **deleting**: The connection is being deleted.
- **deleted**: The connection has been deleted.
• rejected: A hosted connection in the ordering state enters the rejected state if it is deleted by the customer.
• unknown: The state of the connection is not available.

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

encryptionMode (p. 13)

The MAC Security (MACsec) connection encryption mode.

The valid values are no_encrypt, should_encrypt, and must_encrypt.

Type: String

hasLogicalRedundancy (p. 13)

Indicates whether the connection supports a secondary BGP peer in the same address family (IPv4/IPv6).

Type: String
Valid Values: unknown | yes | no

jumboFrameCapable (p. 13)

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean

lagId (p. 13)

The ID of the LAG.

Type: String

loaIssueTime (p. 13)

The time of the most recent call to DescribeLoa (p. 161) for this connection.

Type: Timestamp

location (p. 13)

The location of the connection.

Type: String

macSecCapable (p. 13)

Indicates whether the connection supports MAC Security (MACsec).

Type: Boolean

macSecKeys (p. 13)

The MAC Security (MACsec) security keys associated with the connection.

Type: Array of MacSecKey (p. 241) objects

ownerAccount (p. 13)

The ID of the AWS account that owns the connection.

Type: String
partnerName (p. 13)

The name of the AWS Direct Connect service provider associated with the connection.

Type: String

portEncryptionStatus (p. 13)

The MAC Security (MACsec) port link status of the connection.

The valid values are Encryption Up, which means that there is an active Connection Key Name, or Encryption Down.

Type: String

providerName (p. 13)

The name of the service provider associated with the connection.

Type: String

region (p. 13)

The AWS Region where the connection is located.

Type: String

tags (p. 13)

The tags associated with the connection.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

vlan (p. 13)

The ID of the VLAN.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

DuplicateTagKeysException

A tag key was specified more than once.

HTTP Status Code: 400

TooManyTagsException

You have reached the limit on the number of tags that can be assigned.
HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
AllocatePrivateVirtualInterface

Provisions a private virtual interface to be owned by the specified AWS account.

Virtual interfaces created using this action must be confirmed by the owner using ConfirmPrivateVirtualInterface (p. 56). Until then, the virtual interface is in the Confirming state and is not available to handle traffic.

Request Syntax

```json
{
  "connectionId": "string",
  "newPrivateVirtualInterfaceAllocation": {
    "addressFamily": "string",
    "amazonAddress": "string",
    "asn": number,
    "authKey": "string",
    "customerAddress": "string",
    "mtu": number,
    "tags": [
      {
        "key": "string",
        "value": "string"
      }
    ],
    "virtualInterfaceName": "string",
    "vlan": number
  },
  "ownerAccount": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 18)**

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

Type: String

Required: Yes

**newPrivateVirtualInterfaceAllocation (p. 18)**

Information about the private virtual interface.

Type: NewPrivateVirtualInterfaceAllocation (p. 246) object

Required: Yes

**ownerAccount (p. 18)**

The ID of the AWS account that owns the virtual private interface.

Type: String

Required: Yes
Response Syntax

```json
{
    "addressFamily": "string",
    "amazonAddress": "string",
    "amazonSideAsn": number,
    "asn": number,
    "authKey": "string",
    "awsDeviceV2": "string",
    "awsLogicalDeviceId": "string",
    "bgpPeers": [
      {
        "addressFamily": "string",
        "amazonAddress": "string",
        "asn": number,
        "authKey": "string",
        "awsDeviceV2": "string",
        "awsLogicalDeviceId": "string",
        "bgpPeerId": "string",
        "bgpPeerState": "string",
        "bgpStatus": "string",
        "customerAddress": "string"
      }
    ],
    "connectionId": "string",
    "customerAddress": "string",
    "customerRouterConfig": "string",
    "directConnectGatewayId": "string",
    "jumboFrameCapable": boolean,
    "location": "string",
    "mtu": number,
    "ownerAccount": "string",
    "region": "string",
    "routeFilterPrefixes": [
      {
        "cidr": "string"
      }
    ],
    "siteLinkEnabled": boolean,
    "tags": [
      {
        "key": "string",
        "value": "string"
      }
    ],
    "virtualGatewayId": "string",
    "virtualInterfaceId": "string",
    "virtualInterfaceName": "string",
    "virtualInterfaceState": "string",
    "virtualInterfaceType": "string",
    "vlan": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**addressFamily (p. 19)**

The address family for the BGP peer.
Type: String

Valid Values: ipv4 | ipv6

**amazonAddress (p. 19)**

The IP address assigned to the Amazon interface.

Type: String

**amazonSideAsn (p. 19)**

The autonomous system number (ASN) for the Amazon side of the connection.

Type: Long

**asn (p. 19)**

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

The valid values are 1-2147483647.

Type: Integer

**authKey (p. 19)**

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

Type: String

**awsDeviceV2 (p. 19)**

The Direct Connect endpoint that terminates the physical connection.

Type: String

**awsLogicalDeviceId (p. 19)**

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

**bgpPeers (p. 19)**

The BGP peers configured on this virtual interface.

Type: Array of BGPPeer (p. 216) objects

**connectionId (p. 19)**

The ID of the connection.

Type: String

**customerAddress (p. 19)**

The IP address assigned to the customer interface.

Type: String

**customerRouterConfig (p. 19)**

The customer router configuration.

Type: String
directConnectGatewayId (p. 19)

The ID of the Direct Connect gateway.

Type: String

jumboFrameCapable (p. 19)

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean

location (p. 19)

The location of the connection.

Type: String

mtu (p. 19)

The maximum transmission unit (MTU), in bytes. The supported values are 1500 and 9001. The default value is 1500.

Type: Integer

ownerAccount (p. 19)

The ID of the AWS account that owns the virtual interface.

Type: String

region (p. 19)

The AWS Region where the virtual interface is located.

Type: String

routeFilterPrefixes (p. 19)

The routes to be advertised to the AWS network in this Region. Applies to public virtual interfaces.

Type: Array of RouteFilterPrefix (p. 257) objects

siteLinkEnabled (p. 19)

Indicates whether SiteLink is enabled.

Type: Boolean

tags (p. 19)

The tags associated with the virtual interface.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

virtualGatewayId (p. 19)

The ID of the virtual private gateway. Applies only to private virtual interfaces.

Type: String

virtualInterfaceId (p. 19)

The ID of the virtual interface.

Type: String
virtualInterfaceName (p. 19)

The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).

Type: String

virtualInterfaceState (p. 19)

The state of the virtual interface. The following are the possible values:

- **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.
- **down**: A virtual interface that is BGP down.
- **deleting**: A virtual interface is in this state immediately after calling DeleteVirtualInterface (p. 129) 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 enters the Rejected state.
- **unknown**: The state of the virtual interface is not available.

Type: String

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

virtualInterfaceType (p. 19)

The type of virtual interface. The possible values are private and public.

Type: String

vlan (p. 19)

The ID of the VLAN.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.
HTTP Status Code: 400

**DuplicateTagKeysException**

A tag key was specified more than once.

HTTP Status Code: 400

**TooManyTagsException**

You have reached the limit on the number of tags that can be assigned.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
AllocatePublicVirtualInterface

Provisions a public virtual interface to be owned by the specified AWS account.

The owner of a connection calls this function to provision a public virtual interface to be owned by the specified AWS account.

Virtual interfaces created using this function must be confirmed by the owner using ConfirmPublicVirtualInterface (p. 59). Until this step has been completed, the virtual interface is in the confirming state and is not available to handle traffic.

When creating an IPv6 public virtual interface, omit the Amazon address and customer address. IPv6 addresses are automatically assigned from the Amazon pool of IPv6 addresses; you cannot specify custom IPv6 addresses.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 24)**

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

Type: String

Required: Yes

**newPublicVirtualInterfaceAllocation (p. 24)**

Information about the public virtual interface.
Type: NewPublicVirtualInterfaceAllocation (p. 250) object

Required: Yes

ownerAccount (p. 24)

The ID of the AWS account that owns the public virtual interface.

Type: String

Required: Yes

Response Syntax

```
{
  "addressFamily": "string",
  "amazonAddress": "string",
  "amazonSideAsn": number,
  "asn": number,
  "authKey": "string",
  "awsDeviceV2": "string",
  "awsLogicalDeviceId": "string",
  "bgpPeers": [
    {
      "addressFamily": "string",
      "amazonAddress": "string",
      "asn": number,
      "authKey": "string",
      "awsDeviceV2": "string",
      "awsLogicalDeviceId": "string",
      "bgpPeerId": "string",
      "bgpPeerState": "string",
      "bgpStatus": "string",
      "customerAddress": "string"
    }
  ],
  "connectionId": "string",
  "customerAddress": "string",
  "customerRouterConfig": "string",
  "directConnectGatewayId": "string",
  "jumboFrameCapable": boolean,
  "location": "string",
  "mtu": number,
  "ownerAccount": "string",
  "region": "string",
  "siteLinkEnabled": boolean,
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "virtualGatewayId": "string",
  "virtualInterfaceId": "string",
  "virtualInterfaceName": "string",
  "virtualInterfaceState": "string",
  "virtualInterfaceType": "string",
  "vlan": number
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

addressFamily (p. 25)

The address family for the BGP peer.

Type: String

Valid Values: ipv4 | ipv6

amazonAddress (p. 25)

The IP address assigned to the Amazon interface.

Type: String

amazonSideAsn (p. 25)

The autonomous system number (ASN) for the Amazon side of the connection.

Type: Long

asn (p. 25)

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

The valid values are 1-2147483647.

Type: Integer

authKey (p. 25)

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

Type: String

awsDeviceV2 (p. 25)

The Direct Connect endpoint that terminates the physical connection.

Type: String

awsLogicalDeviceId (p. 25)

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

bgpPeers (p. 25)

The BGP peers configured on this virtual interface.

Type: Array of BGPPeer (p. 216) objects

connectionId (p. 25)

The ID of the connection.
Type: String

**customerAddress (p. 25)**

The IP address assigned to the customer interface.

Type: String

**customerRouterConfig (p. 25)**

The customer router configuration.

Type: String

**directConnectGatewayId (p. 25)**

The ID of the Direct Connect gateway.

Type: String

**jumboFrameCapable (p. 25)**

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean

**location (p. 25)**

The location of the connection.

Type: String

**mtu (p. 25)**

The maximum transmission unit (MTU), in bytes. The supported values are 1500 and 9001. The default value is 1500.

Type: Integer

**ownerAccount (p. 25)**

The ID of the AWS account that owns the virtual interface.

Type: String

**region (p. 25)**

The AWS Region where the virtual interface is located.

Type: String

**routeFilterPrefixes (p. 25)**

The routes to be advertised to the AWS network in this Region. Applies to public virtual interfaces.

Type: Array of RouteFilterPrefix (p. 257) objects

**siteLinkEnabled (p. 25)**

Indicates whether SiteLink is enabled.

Type: Boolean

**tags (p. 25)**

The tags associated with the virtual interface.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.
virtualGatewayId (p. 25)

The ID of the virtual private gateway. Applies only to private virtual interfaces.

Type: String

virtualInterfaceId (p. 25)

The ID of the virtual interface.

Type: String

virtualInterfaceName (p. 25)

The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).

Type: String

virtualInterfaceState (p. 25)

The state of the virtual interface. The following are the possible values:

- **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.
- **down**: A virtual interface that is BGP down.
- **deleting**: A virtual interface is in this state immediately after calling DeleteVirtualInterface (p. 129) 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 enters the Rejected state.
- **unknown**: The state of the virtual interface is not available.

Type: String

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

virtualInterfaceType (p. 25)

The type of virtual interface. The possible values are private and public.

Type: String

vlan (p. 25)

The ID of the VLAN.

Type: Integer

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 270).
DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

DuplicateTagKeysException

A tag key was specified more than once.

HTTP Status Code: 400

TooManyTagsException

You have reached the limit on the number of tags that can be assigned.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
AllocateTransitVirtualInterface

Provisions a transit virtual interface to be owned by the specified AWS account. Use this type of interface to connect a transit gateway to your Direct Connect gateway.

The owner of a connection provisions a transit virtual interface to be owned by the specified AWS account.

After you create a transit virtual interface, it must be confirmed by the owner using ConfirmTransitVirtualInterface (p. 61). Until this step has been completed, the transit virtual interface is in the requested state and is not available to handle traffic.

Request Syntax

```json
{
    "connectionId": "string",
    "newTransitVirtualInterfaceAllocation": {
        "addressFamily": "string",
        "amazonAddress": "string",
        "asn": number,
        "authKey": "string",
        "customerAddress": "string",
        "mtu": number,
        "tags": [
            {
                "key": "string",
                "value": "string"
            }
        ],
        "virtualInterfaceName": "string",
        "vlan": number
    },
    "ownerAccount": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 30)**

The ID of the connection on which the transit virtual interface is provisioned.

Type: String

Required: Yes

**newTransitVirtualInterfaceAllocation (p. 30)**

Information about the transit virtual interface.

Type: NewTransitVirtualInterfaceAllocation (p. 254) object

Required: Yes
ownerAccount (p. 30)

The ID of the AWS account that owns the transit virtual interface.

Type: String

Required: Yes

Response Syntax

```json
{
   "virtualInterface": {
      "addressFamily": "string",
      "amazonAddress": "string",
      "amazonSideAsn": number,
      "asn": number,
      "authKey": "string",
      "awsDeviceV2": "string",
      "awsLogicalDeviceId": "string",
      "bgpPeers": [
         {
            "addressFamily": "string",
            "amazonAddress": "string",
            "asn": number,
            "authKey": "string",
            "awsDeviceV2": "string",
            "awsLogicalDeviceId": "string",
            "bgpPeerId": "string",
            "bgpPeerState": "string",
            "bgpStatus": "string",
            "customerAddress": "string"
         }
      ],
      "connectionId": "string",
      "customerAddress": "string",
      "customerRouterConfig": "string",
      "directConnectGatewayId": "string",
      "jumboFrameCapable": boolean,
      "location": "string",
      "mtu": number,
      "ownerAccount": "string",
      "region": "string",
      "routeFilterPrefixes": [
         {
            "cidr": "string"
         }
      ],
      "siteLinkEnabled": boolean,
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "virtualGatewayId": "string",
      "virtualInterfaceId": "string",
      "virtualInterfaceName": "string",
      "virtualInterfaceState": "string",
      "virtualInterfaceType": "string",
      "vlan": number
   }
}
```

API Version 2012-10-25
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**virtualInterface (p. 31)**

Information about a virtual interface.

Type: VirtualInterface (p. 262) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

**DuplicateTagKeysException**

A tag key was specified more than once.

HTTP Status Code: 400

**TooManyTagsException**

You have reached the limit on the number of tags that can be assigned.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
AssociateConnectionWithLag

Associates an existing connection with a link aggregation group (LAG). The connection is interrupted and re-established as a member of the LAG (connectivity to AWS is interrupted). The connection must be hosted on the same AWS Direct Connect endpoint as the LAG, and its bandwidth must match the bandwidth for the LAG. You can re-associate a connection that's currently associated with a different LAG; however, if removing the connection would cause the original LAG to fall below its setting for minimum number of operational connections, the request fails.

Any virtual interfaces that are directly associated with the connection are automatically re-associated with the LAG. If the connection was originally associated with a different LAG, the virtual interfaces remain associated with the original LAG.

For interconnects, any hosted connections are automatically re-associated with the LAG. If the interconnect was originally associated with a different LAG, the hosted connections remain associated with the original LAG.

Request Syntax

```
{
  "connectionId": "string",
  "lagId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 33)**

The ID of the connection.

Type: String

Required: Yes

**lagId (p. 33)**

The ID of the LAG with which to associate the connection.

Type: String

Required: Yes

Response Syntax

```
{
  "awsDevice": "string",
  "awsDeviceV2": "string",
  "awsLogicalDeviceId": "string",
  "bandwidth": "string",
  "connectionId": "string",
}
```

API Version 2012-10-25
"connectionName": "string",
"connectionState": "string",
"encryptionMode": "string",
"hasLogicalRedundancy": "string",
"jumboFrameCapable": boolean,
"lagId": "string",
"loaIssueTime": number,
"location": "string",
"macSecCapable": boolean,
"macSecKeys": [
  {
    "ckn": "string",
    "secretARN": "string",
    "startOn": "string",
    "state": "string"
  }
],
"ownerAccount": "string",
"partnerName": "string",
"portEncryptionStatus": "string",
"providerName": "string",
"region": "string",
"tags": [
  {
    "key": "string",
    "value": "string"
  }
],
"vlan": number

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

awsDevice (p. 33)

This parameter has been deprecated.

The Direct Connect endpoint on which the physical connection terminates.

Type: String

awsDeviceV2 (p. 33)

The Direct Connect endpoint that terminates the physical connection.

Type: String

awsLogicalDeviceId (p. 33)

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

bandwidth (p. 33)

The bandwidth of the connection.

Type: String
**connectionId (p. 33)**

The ID of the connection.

Type: String

**connectionName (p. 33)**

The name of the connection.

Type: String

**connectionState (p. 33)**

The state of the connection. The following are the possible values:

- **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.
- **deleting**: The connection is being deleted.
- **deleted**: The connection has been deleted.
- **rejected**: A hosted connection in the ordering state enters the rejected state if it is deleted by the customer.
- **unknown**: The state of the connection is not available.

Type: String

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

**encryptionMode (p. 33)**

The MAC Security (MACsec) connection encryption mode.

The valid values are no_encrypt, should_encrypt, and must_encrypt.

Type: String

**hasLogicalRedundancy (p. 33)**

Indicates whether the connection supports a secondary BGP peer in the same address family (IPv4/IPv6).

Type: String

Valid Values: unknown | yes | no

**jumboFrameCapable (p. 33)**

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean

**lagId (p. 33)**

The ID of the LAG.

Type: String
**loaIssueTime (p. 33)**

The time of the most recent call to *DescribeLoa (p. 161)* for this connection.

Type: Timestamp

**location (p. 33)**

The location of the connection.

Type: String

**macSecCapable (p. 33)**

Indicates whether the connection supports MAC Security (MACsec).

Type: Boolean

**macSecKeys (p. 33)**

The MAC Security (MACsec) security keys associated with the connection.

Type: Array of *MacSecKey (p. 241)* objects

**ownerAccount (p. 33)**

The ID of the AWS account that owns the connection.

Type: String

**partnerName (p. 33)**

The name of the AWS Direct Connect service provider associated with the connection.

Type: String

**portEncryptionStatus (p. 33)**

The MAC Security (MACsec) port link status of the connection.

The valid values are *Encryption Up*, which means that there is an active Connection Key Name, or *Encryption Down*.

Type: String

**providerName (p. 33)**

The name of the service provider associated with the connection.

Type: String

**region (p. 33)**

The AWS Region where the connection is located.

Type: String

**tags (p. 33)**

The tags associated with the connection.

Type: Array of *Tag (p. 260)* objects

Array Members: Minimum number of 1 item.

**vlan (p. 33)**

The ID of the VLAN.
Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
AssociateHostedConnection

Associates a hosted connection and its virtual interfaces with a link aggregation group (LAG) or interconnect. If the target interconnect or LAG has an existing hosted connection with a conflicting VLAN number or IP address, the operation fails. This action temporarily interrupts the hosted connection's connectivity to AWS as it is being migrated.

**Note**
Intended for use by AWS Direct Connect Partners only.

**Request Syntax**

```
{
    "connectionId": "string",
    "parentConnectionId": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 38)**

The ID of the hosted connection.

Type: String

Required: Yes

**parentConnectionId (p. 38)**

The ID of the interconnect or the LAG.

Type: String

Required: Yes

**Response Syntax**

```
{
    "awsDevice": "string",
    "awsDeviceV2": "string",
    "awsLogicalDeviceId": "string",
    "bandwidth": "string",
    "connectionId": "string",
    "connectionName": "string",
    "connectionState": "string",
    "encryptionMode": "string",
    "hasLogicalRedundancy": "string",
    "jumboFrameCapable": boolean,
    "lagId": "string",
    "loaIssueTime": number,
    "location": "string",
    "macSecCapable": boolean,
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**awsDevice (p. 38)**

*This parameter has been deprecated.*

The Direct Connect endpoint on which the physical connection terminates.

Type: String

**awsDeviceV2 (p. 38)**

The Direct Connect endpoint that terminates the physical connection.

Type: String

**awsLogicalDeviceId (p. 38)**

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

**bandwidth (p. 38)**

The bandwidth of the connection.

Type: String

**connectionId (p. 38)**

The ID of the connection.

Type: String

**connectionName (p. 38)**

The name of the connection.

Type: String
connectionState (p. 38)

The state of the connection. The following are the possible values:
- 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.
- deleting: The connection is being deleted.
- deleted: The connection has been deleted.
- rejected: A hosted connection in the ordering state enters the rejected state if it is deleted by the customer.
- unknown: The state of the connection is not available.

Type: String

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

encryptionMode (p. 38)

The MAC Security (MACsec) connection encryption mode.

The valid values are no_encrypt, should_encrypt, and must_encrypt.

Type: String

hasLogicalRedundancy (p. 38)

Indicates whether the connection supports a secondary BGP peer in the same address family (IPv4/IPv6).

Type: String

Valid Values: unknown | yes | no

jumboFrameCapable (p. 38)

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean

lagId (p. 38)

The ID of the LAG.

Type: String

loaiIssueTime (p. 38)

The time of the most recent call to DescribeLoa (p. 161) for this connection.

Type: Timestamp

location (p. 38)

The location of the connection.

Type: String
macSecCapable (p. 38)

Indicates whether the connection supports MAC Security (MACsec).

Type: Boolean

macSecKeys (p. 38)

The MAC Security (MACsec) security keys associated with the connection.

Type: Array of MacSecKey (p. 241) objects

ownerAccount (p. 38)

The ID of the AWS account that owns the connection.

Type: String

partnerName (p. 38)

The name of the AWS Direct Connect service provider associated with the connection.

Type: String

portEncryptionStatus (p. 38)

The MAC Security (MACsec) port link status of the connection.

The valid values are Encryption Up, which means that there is an active Connection Key Name, or Encryption Down.

Type: String

providerName (p. 38)

The name of the service provider associated with the connection.

Type: String

region (p. 38)

The AWS Region where the connection is located.

Type: String

tags (p. 38)

The tags associated with the connection.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

vlan (p. 38)

The ID of the VLAN.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.
HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
AssociateMacSecKey

Associates a MAC Security (MACsec) Connection Key Name (CKN)/ Connectivity Association Key (CAK) pair with an AWS Direct Connect dedicated connection.

You must supply either the `secretARN`, or the CKN/CAK (`ckn` and `cak`) pair in the request.

For information about MAC Security (MACsec) key considerations, see MACsec pre-shared CKN/CAK key considerations in the AWS Direct Connect User Guide.

Request Syntax

```
{
  "cak": "string",
  "ckn": "string",
  "connectionId": "string",
  "secretARN": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**cak (p. 43)**

The MAC Security (MACsec) CAK to associate with the dedicated connection.

You can create the CKN/CAK pair using an industry standard tool.

The valid values are 64 hexadecimal characters (0-9, A-E).

If you use this request parameter, you must use the `ckn` request parameter and not use the `secretARN` request parameter.

Type: String

Required: No

**ckn (p. 43)**

The MAC Security (MACsec) CKN to associate with the dedicated connection.

You can create the CKN/CAK pair using an industry standard tool.

The valid values are 64 hexadecimal characters (0-9, A-E).

If you use this request parameter, you must use the `cak` request parameter and not use the `secretARN` request parameter.

Type: String

Required: No

**connectionId (p. 43)**

The ID of the dedicated connection (dxcon-xxxx), or the ID of the LAG (dxlag-xxxx).
You can use DescribeConnections (p. 133) or DescribeLags (p. 158) to retrieve connection ID.

Type: String
Required: Yes

secretARN (p. 43)

The Amazon Resource Name (ARN) of the MAC Security (MACsec) secret key to associate with the dedicated connection.

You can use DescribeConnections (p. 133) or DescribeLags (p. 158) to retrieve the MAC Security (MACsec) secret key.

If you use this request parameter, you do not use the ckn and cak request parameters.

Type: String
Required: No

Response Syntax

You can use DescribeConnections (p. 133) or DescribeLags (p. 158) to retrieve the MAC Security (MACsec) secret key.

If you use this request parameter, you do not use the ckn and cak request parameters.

Type: String
Required: No

Response Syntax

```
{
    "connectionId": "string",
    "macSecKeys": [
    {
        "ckn": "string",
        "secretARN": "string",
        "startOn": "string",
        "state": "string"
    }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

connectionId (p. 44)

The ID of the dedicated connection (dxcon-xxxx), or the ID of the LAG (dxlag-xxxx).

Type: String

macSecKeys (p. 44)

The MAC Security (MACsec) security keys associated with the dedicated connection.

Type: Array of MacSecKey (p. 241) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.
HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
AssociateVirtualInterface

Associates a virtual interface with a specified link aggregation group (LAG) or connection. Connectivity to AWS is temporarily interrupted as the virtual interface is being migrated. If the target connection or LAG has an associated virtual interface with a conflicting VLAN number or a conflicting IP address, the operation fails.

Virtual interfaces associated with a hosted connection cannot be associated with a LAG; hosted connections must be migrated along with their virtual interfaces using AssociateHostedConnection (p. 38).

To reassociate a virtual interface to a new connection or LAG, the requester must own either the virtual interface itself or the connection to which the virtual interface is currently associated. Additionally, the requester must own the connection or LAG for the association.

Request Syntax

```json
{
  "connectionId": "string",
  "virtualInterfaceId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

connectionId (p. 46)

The ID of the LAG or connection.

Type: String

Required: Yes

virtualInterfaceId (p. 46)

The ID of the virtual interface.

Type: String

Required: Yes

Response Syntax

```json
{
  "addressFamily": "string",
  "amazonAddress": "string",
  "amazonSideAsn": number,
  "asn": number,
  "authKey": "string",
  "awsDeviceV2": "string",
  "awsLogicalDeviceId": "string",
  "bgpPeers": [
  }
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**addressFamily (p. 46)**

The address family for the BGP peer.

Type: String

Valid Values: ipv4 | ipv6

**amazonAddress (p. 46)**

The IP address assigned to the Amazon interface.

Type: String

**amazonSideAsn (p. 46)**

The autonomous system number (ASN) for the Amazon side of the connection.
Type: Long
asn (p. 46)
The autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.
The valid values are 1-2147483647.

Type: Integer
authKey (p. 46)
The authentication key for BGP configuration. This string has a minimum length of 6 characters and
and a maximum length of 80 characters.

Type: String
awsDeviceV2 (p. 46)
The Direct Connect endpoint that terminates the physical connection.

Type: String
awsLogicalDeviceId (p. 46)
The Direct Connect endpoint that terminates the logical connection. This device might be different
than the device that terminates the physical connection.

Type: String
bgpPeers (p. 46)
The BGP peers configured on this virtual interface.

Type: Array of BGPPeer (p. 216) objects
customerAddress (p. 46)
The IP address assigned to the customer interface.

Type: String
customerRouterConfig (p. 46)
The customer router configuration.

Type: String
directConnectGatewayId (p. 46)
The ID of the Direct Connect gateway.

Type: String
jumboFrameCapable (p. 46)
Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean
location (p. 46)
The location of the connection.
Response Elements

Type: String

`mtu` (p. 46)

The maximum transmission unit (MTU), in bytes. The supported values are 1500 and 9001. The default value is 1500.

Type: Integer

`ownerAccount` (p. 46)

The ID of the AWS account that owns the virtual interface.

Type: String

`region` (p. 46)

The AWS Region where the virtual interface is located.

Type: String

`routeFilterPrefixes` (p. 46)

The routes to be advertised to the AWS network in this Region. Applies to public virtual interfaces.

Type: Array of `RouteFilterPrefix` (p. 257) objects

`siteLinkEnabled` (p. 46)

Indicates whether SiteLink is enabled.

Type: Boolean

`tags` (p. 46)

The tags associated with the virtual interface.

Type: Array of `Tag` (p. 260) objects

Array Members: Minimum number of 1 item.

`virtualGatewayId` (p. 46)

The ID of the virtual private gateway. Applies only to private virtual interfaces.

Type: String

`virtualInterfaceId` (p. 46)

The ID of the virtual interface.

Type: String

`virtualInterfaceName` (p. 46)

The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).

Type: String

`virtualInterfaceState` (p. 46)

The state of the virtual interface. The following are the possible values:

- 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.
• down: A virtual interface that is BGP down.
• deleting: A virtual interface is in this state immediately after calling DeleteVirtualInterface (p. 129) 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 enters the Rejected state.
• unknown: The state of the virtual interface is not available.

Type: String
Valid Values: confirming | verifying | pending | available | down | deleting | deleted | rejected | unknown

virtualInterfaceType (p. 46)
The type of virtual interface. The possible values are private and public.

Type: String

vlan (p. 46)
The ID of the VLAN.

Type: Integer

Errors
For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException
One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException
A server-side error occurred.

HTTP Status Code: 400

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

• AWS Command Line Interface
• AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java V2
See Also

- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
ConfirmConnection

Confirms the creation of the specified hosted connection on an interconnect.

Upon creation, the hosted connection is initially in the ordering state, and remains in this state until the owner confirms creation of the hosted connection.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

- **connectionId (p. 52)**
  - The ID of the hosted connection.
  - Type: String
  - Required: Yes

Response Syntax

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

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

- **connectionState (p. 52)**
  - The state of the connection. The following are the possible values:
    - 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.
    - deleting: The connection is being deleted.
• **deleted**: The connection has been deleted.
• **rejected**: A hosted connection in the **ordering** state enters the **rejected** state if it is deleted by the customer.
• **unknown**: The state of the connection is not available.

  **Type**: String
  **Valid Values**: ordering | requested | pending | available | down | deleting | deleted | rejected | unknown

**Errors**

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

**DirectConnectClientException**

  One or more parameters are not valid.

  **HTTP Status Code**: 400

**DirectConnectServerException**

  A server-side error occurred.

  **HTTP Status Code**: 400

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)
ConfirmCustomerAgreement

The confirmation of the terms of agreement when creating the connection/link aggregation group (LAG).

Request Syntax

```json
{
   "agreementName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**agreementName (p. 54)**

The name of the customer agreement.

Type: String

Length Constraints: Maximum length of 100.

Required: No

Response Syntax

```json
{
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**status (p. 54)**

The status of the customer agreement when the connection was created. This will be either signed or unsigned.

Type: String

Length Constraints: Maximum length of 30.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).
DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
ConfirmPrivateVirtualInterface

Accepts ownership of a private virtual interface created by another AWS account.

After the virtual interface owner makes this call, the virtual interface is created and attached to the specified virtual private gateway or Direct Connect gateway, and is made available to handle traffic.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

directConnectGatewayId (p. 56)

  The ID of the Direct Connect gateway.

  Type: String

  Required: No

virtualGatewayId (p. 56)

  The ID of the virtual private gateway.

  Type: String

  Required: No

virtualInterfaceId (p. 56)

  The ID of the virtual interface.

  Type: String

  Required: Yes

Response Syntax

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

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in JSON format by the service.

**virtualInterfaceState** (p. 56)

The state of the virtual interface. The following are the possible values:

- **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.
- **down**: A virtual interface that is BGP down.
- **deleting**: A virtual interface is in this state immediately after calling [DeleteVirtualInterface](p. 129) 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 enters the **Rejected** state.
- **unknown**: The state of the virtual interface is not available.

Type: String

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

**Errors**

For information about the errors that are common to all actions, see [Common Errors](p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2

API Version 2012-10-25
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V3
ConfirmPublicVirtualInterface

Accepts ownership of a public virtual interface created by another AWS account.

After the virtual interface owner makes this call, the specified virtual interface is created and made available to handle traffic.

**Request Syntax**

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

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

`virtualInterfaceId (p. 59)`

The ID of the virtual interface.

Type: String

Required: Yes

**Response Syntax**

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

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

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

`virtualInterfaceState (p. 59)`

The state of the virtual interface. The following are the possible values:

- **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.
• down: A virtual interface that is BGP down.
• deleting: A virtual interface is in this state immediately after calling DeleteVirtualInterface (p. 129) 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 enters the Rejected state.
• unknown: The state of the virtual interface is not available.

Type: String

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

• AWS Command Line Interface
• AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java V2
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V3
ConfirmTransitVirtualInterface

Accepts ownership of a transit virtual interface created by another AWS account.

After the owner of the transit virtual interface makes this call, the specified transit virtual interface is created and made available to handle traffic.

Request Syntax

```json
{
    "directConnectGatewayId": "string",
    "virtualInterfaceId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

directConnectGatewayId (p. 61)

The ID of the Direct Connect gateway.

Type: String

Required: Yes

virtualInterfaceId (p. 61)

The ID of the virtual interface.

Type: String

Required: Yes

Response Syntax

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

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

virtualInterfaceState (p. 61)

The state of the virtual interface. The following are the possible values:

- 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.
- **down**: A virtual interface that is BGP down.
- **deleting**: A virtual interface is in this state immediately after calling
  `DeleteVirtualInterface` (p. 129) 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
  enters the **Rejected** state.
- **unknown**: The state of the virtual interface is not available.

**Type**: String

**Valid Values**: confirming | verifying | pending | available | down | deleting | deleted | rejected | unknown

---

**Errors**

For information about the errors that are common to all actions, see [Common Errors (p. 270)].

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

---

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)
CreateBGPPeer

Creates a BGP peer on the specified virtual interface.

You must create a BGP peer for the corresponding address family (IPv4/IPv6) in order to access AWS resources that also use that address family.

If logical redundancy is not supported by the connection, interconnect, or LAG, the BGP peer cannot be in the same address family as an existing BGP peer on the virtual interface.

When creating a IPv6 BGP peer, omit the Amazon address and customer address. IPv6 addresses are automatically assigned from the Amazon pool of IPv6 addresses; you cannot specify custom IPv6 addresses.

For a public virtual interface, the Autonomous System Number (ASN) must be private or already on the allow list for the virtual interface.

Request Syntax

{  
  "newBGPPeer": {  
    "addressFamily": "string",  
    "amazonAddress": "string",  
    "asn": number,  
    "authKey": "string",  
    "customerAddress": "string"  
  },  
  "virtualInterfaceId": "string"  
}

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

newBGPPeer (p. 63)

Information about the BGP peer.

Type: NewBGPPeer (p. 243) object

Required: No

virtualInterfaceId (p. 63)

The ID of the virtual interface.

Type: String

Required: No

Response Syntax

{  
  
}

API Version 2012-10-25

63
"virtualInterface": {  
  "addressFamily": "string",  
  "amazonAddress": "string",  
  "amazonSideAsn": number,  
  "asn": number,  
  "authKey": "string",  
  "awsDeviceV2": "string",  
  "awsLogicalDeviceId": "string",  
  "bgpPeers": [  
    {  
      "addressFamily": "string",  
      "amazonAddress": "string",  
      "asn": number,  
      "authKey": "string",  
      "awsDeviceV2": "string",  
      "awsLogicalDeviceId": "string",  
      "bgpPeerId": "string",  
      "bgpPeerState": "string",  
      "bgpStatus": "string",  
      "customerAddress": "string"  
    }  
  ],  
  "connectionId": "string",  
  "customerAddress": "string",  
  "customerRouterConfig": "string",  
  "directConnectGatewayId": "string",  
  "jumboFrameCapable": boolean,  
  "location": "string",  
  "mtu": number,  
  "ownerAccount": "string",  
  "region": "string",  
  "routeFilterPrefixes": [  
    {  
      "cidr": "string"  
    }  
  ],  
  "siteLinkEnabled": boolean,  
  "tags": [  
    {  
      "key": "string",  
      "value": "string"  
    }  
  ],  
  "virtualGatewayId": "string",  
  "virtualInterfaceId": "string",  
  "virtualInterfaceName": "string",  
  "virtualInterfaceState": "string",  
  "virtualInterfaceType": "string",  
  "vlan": number  
}  
}  

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**virtualInterface (p. 63)**

The virtual interface.

Type: VirtualInterface (p. 262) object
Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
CreateConnection

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

A connection links your internal network to an AWS Direct Connect location over a standard Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to an AWS Direct Connect router.

To find the locations for your Region, use DescribeLocations (p. 163).

You can automatically add the new connection to a link aggregation group (LAG) by specifying a LAG ID in the request. This ensures that the new connection is allocated on the same AWS Direct Connect endpoint that hosts the specified LAG. If there are no available ports on the endpoint, the request fails and no connection is created.

Request Syntax

```json
{
    "bandwidth": "string",
    "connectionName": "string",
    "lagId": "string",
    "location": "string",
    "providerName": "string",
    "requestMACSec": boolean,
    "tags": [  
        {  
            "key": "string",
            "value": "string"
        }
    ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**bandwidth (p. 66)**

The bandwidth of the connection.

Type: String

Required: Yes

**connectionName (p. 66)**

The name of the connection.

Type: String

Required: Yes

**lagId (p. 66)**

The ID of the LAG.

Type: String
Required: No

**location (p. 66)**

The location of the connection.

Type: String

Required: Yes

**providerName (p. 66)**

The name of the service provider associated with the requested connection.

Type: String

Required: No

**requestMACSec (p. 66)**

Indicates whether you want the connection to support MAC Security (MACsec).

MAC Security (MACsec) is only available on dedicated connections. For information about MAC Security (MACsec) prerequisites, see MACsec prerequisites in the AWS Direct Connect User Guide.

Type: Boolean

Required: No

**tags (p. 66)**

The tags to associate with the lag.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

Required: No

### Response Syntax

```
{
  "awsDevice": "string",
  "awsDeviceV2": "string",
  "awsLogicalDeviceId": "string",
  "bandwidth": "string",
  "connectionId": "string",
  "connectionName": "string",
  "connectionState": "string",
  "encryptionMode": "string",
  "hasLogicalRedundancy": "string",
  "jumboFrameCapable": boolean,
  "lagId": "string",
  "loaIssueTime": number,
  "location": "string",
  "macSecCapable": boolean,
  "macSecKeys": [
    {
      "ckn": "string",
      "secretARN": "string",
      "startOn": "string",
      "state": "string"
    }
  ],
```

API Version 2012-10-25

67
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**awsDevice (p. 67)**

*This parameter has been deprecated.*

The Direct Connect endpoint on which the physical connection terminates.

Type: String

**awsDeviceV2 (p. 67)**

The Direct Connect endpoint that terminates the physical connection.

Type: String

**awsLogicalDeviceId (p. 67)**

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

**bandwidth (p. 67)**

The bandwidth of the connection.

Type: String

**connectionId (p. 67)**

The ID of the connection.

Type: String

**connectionName (p. 67)**

The name of the connection.

Type: String

**connectionState (p. 67)**

The state of the connection. The following are the possible values:

- 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.
• **deleting**: The connection is being deleted.
• **deleted**: The connection has been deleted.
• **rejected**: A hosted connection in the ordering state enters the rejected state if it is deleted by the customer.
• **unknown**: The state of the connection is not available.

Type: String

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

**encryptionMode** (p. 67)

The MAC Security (MACsec) connection encryption mode.

The valid values are no_encrypt, should_encrypt, and must_encrypt.

Type: String

**hasLogicalRedundancy** (p. 67)

Indicates whether the connection supports a secondary BGP peer in the same address family (IPv4/IPv6).

Type: String

Valid Values: unknown | yes | no

**jumboFrameCapable** (p. 67)

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean

**lagId** (p. 67)

The ID of the LAG.

Type: String

**loaIssueTime** (p. 67)

The time of the most recent call to `DescribeLoa` (p. 161) for this connection.

Type: Timestamp

**location** (p. 67)

The location of the connection.

Type: String

**macSecCapable** (p. 67)

Indicates whether the connection supports MAC Security (MACsec).

Type: Boolean
macSecKeys (p. 67)
The MAC Security (MACsec) security keys associated with the connection.
Type: Array of MacSecKey (p. 241) objects

ownerAccount (p. 67)
The ID of the AWS account that owns the connection.
Type: String

partnerName (p. 67)
The name of the AWS Direct Connect service provider associated with the connection.
Type: String

portEncryptionStatus (p. 67)
The MAC Security (MACsec) port link status of the connection.
The valid values are Encryption Up, which means that there is an active Connection Key Name, or Encryption Down.
Type: String

providerName (p. 67)
The name of the service provider associated with the connection.
Type: String

region (p. 67)
The AWS Region where the connection is located.
Type: String

tags (p. 67)
The tags associated with the connection.
Type: Array of Tag (p. 260) objects
Array Members: Minimum number of 1 item.

vlan (p. 67)
The ID of the VLAN.
Type: Integer

Errors
For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException
One or more parameters are not valid.
HTTP Status Code: 400

DirectConnectServerException
A server-side error occurred.
HTTP Status Code: 400

**DuplicateTagKeysException**

A tag key was specified more than once.

HTTP Status Code: 400

**TooManyTagsException**

You have reached the limit on the number of tags that can be assigned.

HTTP Status Code: 400

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
CreateDirectConnectGateway

Creates a Direct Connect gateway, which is an intermediate object that enables you to connect a set of virtual interfaces and virtual private gateways. A Direct Connect gateway is global and visible in any AWS Region after it is created. The virtual interfaces and virtual private gateways that are connected through a Direct Connect gateway can be in different AWS Regions. This enables you to connect to a VPC in any Region, regardless of the Region in which the virtual interfaces are located, and pass traffic between them.

Request Syntax

```json
{
    "amazonSideAsn": number,
    "directConnectGatewayName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**amazonSideAsn** (p. 72)

The autonomous system number (ASN) for Border Gateway Protocol (BGP) to be configured on the Amazon side of the connection. The ASN must be in the private range of 64,512 to 65,534 or 4,200,000,000 to 4,294,967,294. The default is 64512.

Type: Long

Required: No

**directConnectGatewayName** (p. 72)

The name of the Direct Connect gateway.

Type: String

Required: Yes

Response Syntax

```json
{
    "directConnectGateway": {
        "amazonSideAsn": number,
        "directConnectGatewayId": "string",
        "directConnectGatewayName": "string",
        "directConnectGatewayState": "string",
        "ownerAccount": "string",
        "stateChangeError": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in JSON format by the service.

**directConnectGateway (p. 72)**

The Direct Connect gateway.

Type: DirectConnectGateway (p. 223) object

---

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 270).

- **DirectConnectClientException**
  
  One or more parameters are not valid.

  HTTP Status Code: 400

- **DirectConnectServerException**
  
  A server-side error occurred.

  HTTP Status Code: 400

---

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
CreateDirectConnectGatewayAssociation

Creates an association between a Direct Connect gateway and a virtual private gateway. The virtual private gateway must be attached to a VPC and must not be associated with another Direct Connect gateway.

**Request Syntax**

```
{
   "addAllowedPrefixesToDirectConnectGateway": [
      {
         "cidr": "string"
      }
   ],
   "directConnectGatewayId": "string",
   "gatewayId": "string",
   "virtualGatewayId": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

*addAllowedPrefixesToDirectConnectGateway* (p. 74)

The Amazon VPC prefixes to advertise to the Direct Connect gateway

This parameter is required when you create an association to a transit gateway.

For information about how to set the prefixes, see Allowed Prefixes in the AWS Direct Connect User Guide.

Type: Array of RouteFilterPrefix (p. 257) objects

Required: No

*directConnectGatewayId* (p. 74)

The ID of the Direct Connect gateway.

Type: String

Required: Yes

*gatewayId* (p. 74)

The ID of the virtual private gateway or transit gateway.

Type: String

Required: No

*virtualGatewayId* (p. 74)

The ID of the virtual private gateway.

Type: String

Required: No
Response Syntax

```json
{
  "directConnectGatewayAssociation": {
    "allowedPrefixesToDirectConnectGateway": [
      {
        "cidr": "string"
      }
    ],
    "associatedGateway": {
      "id": "string",
      "ownerAccount": "string",
      "region": "string",
      "type": "string"
    },
    "associationId": "string",
    "associationState": "string",
    "directConnectGatewayId": "string",
    "directConnectGatewayOwnerAccount": "string",
    "stateChangeError": "string",
    "virtualGatewayId": "string",
    "virtualGatewayOwnerAccount": "string",
    "virtualGatewayRegion": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**directConnectGatewayAssociation (p. 75)**

The association to be created.

Type: DirectConnectGatewayAssociation (p. 225) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:
• AWS Command Line Interface
• AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java V2
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V3
CreateDirectConnectGatewayAssociationProposal

Creates a proposal to associate the specified virtual private gateway or transit gateway with the specified Direct Connect gateway.

You can associate a Direct Connect gateway and virtual private gateway or transit gateway that is owned by any AWS account.

Request Syntax

```json
{
    "addAllowedPrefixesToDirectConnectGateway": [
    {
        "cidr": "string"
    }
],
"directConnectGatewayId": "string",
"directConnectGatewayOwnerAccount": "string",
"gatewayId": "string",
"removeAllowedPrefixesToDirectConnectGateway": [
    {
        "cidr": "string"
    }
]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

`addAllowedPrefixesToDirectConnectGateway` (p. 77)

The Amazon VPC prefixes to advertise to the Direct Connect gateway.

Type: Array of RouteFilterPrefix (p. 257) objects

Required: No

`directConnectGatewayId` (p. 77)

The ID of the Direct Connect gateway.

Type: String

Required: Yes

`directConnectGatewayOwnerAccount` (p. 77)

The ID of the AWS account that owns the Direct Connect gateway.

Type: String

Required: Yes

`gatewayId` (p. 77)

The ID of the virtual private gateway or transit gateway.
removeAllowedPrefixesToDirectConnectGateway (p. 77)
The Amazon VPC prefixes to no longer advertise to the Direct Connect gateway.

Type: Array of RouteFilterPrefix (p. 257) objects
Required: No

Response Syntax

```
{
    "directConnectGatewayAssociationProposal": {
        "associatedGateway": {
            "id": "string",
            "ownerAccount": "string",
            "region": "string",
            "type": "string"
        },
        "directConnectGatewayId": "string",
        "directConnectGatewayOwnerAccount": "string",
        "existingAllowedPrefixesToDirectConnectGateway": [
            {
                "cidr": "string"
            }
        ],
        "proposalId": "string",
        "proposalState": "string",
        "requestedAllowedPrefixesToDirectConnectGateway": [
            {
                "cidr": "string"
            }
        ]
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in JSON format by the service.

directConnectGatewayAssociationProposal (p. 78)
Information about the Direct Connect gateway proposal.

Type: DirectConnectGatewayAssociationProposal (p. 227) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException
One or more parameters are not valid.
HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
CreateInterconnect

Creates an interconnect between an AWS Direct Connect Partner's network and a specific AWS Direct Connect location.

An interconnect is a connection that is capable of hosting other connections. The AWS Direct Connect Partner can use an interconnect to provide AWS Direct Connect hosted connections to customers through their own network services. Like a standard connection, an interconnect links the partner's network to an AWS Direct Connect location over a standard Ethernet fiber-optic cable. One end is connected to the partner's router, the other to an AWS Direct Connect router.

You can automatically add the new interconnect to a link aggregation group (LAG) by specifying a LAG ID in the request. This ensures that the new interconnect is allocated on the same AWS Direct Connect endpoint that hosts the specified LAG. If there are no available ports on the endpoint, the request fails and no interconnect is created.

For each end customer, the AWS Direct Connect Partner provisions a connection on their interconnect by calling AllocateHostedConnection (p. 12). 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.

**Note**
Intended for use by AWS Direct Connect Partners only.

**Request Syntax**

```json
{
    "bandwidth": "string",
    "interconnectName": "string",
    "lagId": "string",
    "location": "string",
    "providerName": "string",
    "tags": [
        {
            "key": "string",
            "value": "string"
        }
    ]
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**bandwidth (p. 80)**

The port bandwidth, in Gbps. The possible values are 1 and 10.

Type: String

Required: Yes

**interconnectName (p. 80)**

The name of the interconnect.
Type: String
Required: Yes

**lagId (p. 80)**
The ID of the LAG.

Type: String
Required: No

**location (p. 80)**
The location of the interconnect.

Type: String
Required: Yes

**providerName (p. 80)**
The name of the service provider associated with the interconnect.

Type: String
Required: No

**tags (p. 80)**
The tags to associate with the interconnect.

Type: Array of Tag (p. 260) objects
Array Members: Minimum number of 1 item.
Required: No

**Response Syntax**

```
{
   "awsDevice": "string",
   "awsDeviceV2": "string",
   "awsLogicalDeviceId": "string",
   "bandwidth": "string",
   "hasLogicalRedundancy": "string",
   "interconnectId": "string",
   "interconnectName": "string",
   "interconnectState": "string",
   "jumboFrameCapable": boolean,
   "lagId": "string",
   "loaIssueTime": number,
   "location": "string",
   "providerName": "string",
   "region": "string",
   "tags": [
      {
         "key": "string",
         "value": "string"
      }
   ]
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

awsDevice (p. 81)

This parameter has been deprecated.

The Direct Connect endpoint on which the physical connection terminates.

Type: String

awsDeviceV2 (p. 81)

The Direct Connect endpoint that terminates the physical connection.

Type: String

awsLogicalDeviceId (p. 81)

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

bandwidth (p. 81)

The bandwidth of the connection.

Type: String

hasLogicalRedundancy (p. 81)

Indicates whether the interconnect supports a secondary BGP in the same address family (IPv4/IPv6).

Type: String

Valid Values: unknown | yes | no

interconnectId (p. 81)

The ID of the interconnect.

Type: String

interconnectName (p. 81)

The name of the interconnect.

Type: String

interconnectState (p. 81)

The state of the interconnect. The following are the possible values:

- 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 is approved, and is being initialized.
- available: The network link is up, and the interconnect is ready for use.
- down: The network link is down.
- deleting: The interconnect is being deleted.
Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
CreateLag

Creates a link aggregation group (LAG) with the specified number of bundled physical dedicated connections between the customer network and a specific AWS Direct Connect location. A LAG is a logical interface that uses the Link Aggregation Control Protocol (LACP) to aggregate multiple interfaces, enabling you to treat them as a single interface.

All connections in a LAG must use the same bandwidth (either 1Gbps or 10Gbps) and must terminate at the same AWS Direct Connect endpoint.

You can have up to 10 dedicated connections per LAG. Regardless of this limit, if you request more connections for the LAG than AWS Direct Connect can allocate on a single endpoint, no LAG is created.

You can specify an existing physical dedicated connection or interconnect to include in the LAG (which counts towards the total number of connections). Doing so interrupts the current physical dedicated connection, and re-establishes them as a member of the LAG. The LAG will be created on the same AWS Direct Connect endpoint to which the dedicated connection terminates. Any virtual interfaces associated with the dedicated connection are automatically disassociated and re-associated with the LAG. The connection ID does not change.

If the AWS account used to create a LAG is a registered AWS Direct Connect Partner, the LAG is automatically enabled to host sub-connections. For a LAG owned by a partner, any associated virtual interfaces cannot be directly configured.

Request Syntax

```json
{
  "childConnectionTags": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "connectionId": "string",
  "connectionsBandwidth": "string",
  "lagName": "string",
  "location": "string",
  "numberOfConnections": number,
  "providerName": "string",
  "requestMACSec": boolean,
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**childConnectionTags (p. 85)**

The tags to associate with the automatically created LAGs.
Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

Required: No

**connectionId (p. 85)**

The ID of an existing dedicated connection to migrate to the LAG.

Type: String

Required: No

**connectionsBandwidth (p. 85)**

The bandwidth of the individual physical dedicated connections bundled by the LAG. The possible values are 1Gbps and 10Gbps.

Type: String

Required: Yes

**lagName (p. 85)**

The name of the LAG.

Type: String

Required: Yes

**location (p. 85)**

The location for the LAG.

Type: String

Required: Yes

**numberOfConnections (p. 85)**

The number of physical dedicated connections initially provisioned and bundled by the LAG.

Type: Integer

Required: Yes

**providerName (p. 85)**

The name of the service provider associated with the LAG.

Type: String

Required: No

**requestMACSec (p. 85)**

Indicates whether the connection will support MAC Security (MACsec).

**Note**

All connections in the LAG must be capable of supporting MAC Security (MACsec). For information about MAC Security (MACsec) prerequisites, see MACsec prerequisites in the AWS Direct Connect User Guide.

Type: Boolean

Required: No
tags (p. 85)

The tags to associate with the LAG.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

Required: No

Response Syntax

```
{
  "allowsHostedConnections": boolean,
  "awsDevice": "string",
  "awsDeviceV2": "string",
  "awsLogicalDeviceId": "string",
  "connections": [
    {
      "awsDevice": "string",
      "awsDeviceV2": "string",
      "awsLogicalDeviceId": "string",
      "bandwidth": "string",
      "connectionId": "string",
      "connectionName": "string",
      "connectionState": "string",
      "encryptionMode": "string",
      "hasLogicalRedundancy": "string",
      "jumboFrameCapable": boolean,
      "lagId": "string",
      "loaIssueTime": number,
      "location": "string",
      "macSecCapable": boolean,
      "macSecKeys": [
        {
          "ckn": "string",
          "secretARN": "string",
          "startOn": "string",
          "state": "string"
        }
      ],
      "ownerAccount": "string",
      "partnerName": "string",
      "portEncryptionStatus": "string",
      "providerName": "string",
      "region": "string",
      "tags": [
        {
          "key": "string",
          "value": "string"
        }
      ],
      "vlan": number
    }
  ],
  "connectionsBandwidth": "string",
  "encryptionMode": "string",
  "hasLogicalRedundancy": "string",
  "jumboFrameCapable": boolean,
  "lagId": "string",
  "lagName": "string",
  "lagState": "string",
  "location": "string",
```

API Version 2012-10-25

87
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

allowsHostedConnections (p. 87)

Indicates whether the LAG can host other connections.

Type: Boolean

awsDevice (p. 87)

This parameter has been deprecated.

The AWS Direct Connect endpoint that hosts the LAG.

Type: String

awsDeviceV2 (p. 87)

The AWS Direct Connect endpoint that hosts the LAG.

Type: String

awsLogicalDeviceId (p. 87)

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

connections (p. 87)

The connections bundled by the LAG.

Type: Array of Connection (p. 218) objects

connectionsBandwidth (p. 87)

The individual bandwidth of the physical connections bundled by the LAG. The possible values are 1Gbps and 10Gbps.
**Response Elements**

**encryptionMode (p. 87)**

The LAG MAC Security (MACsec) encryption mode.

The valid values are `no_encrypt`, `should_encrypt`, and `must_encrypt`.

**hasLogicalRedundancy (p. 87)**

Indicates whether the LAG supports a secondary BGP peer in the same address family (IPv4/IPv6).

**jumboFrameCapable (p. 87)**

Indicates whether jumbo frames (9001 MTU) are supported.

**lagId (p. 87)**

The ID of the LAG.

**lagName (p. 87)**

The name of the LAG.

**lagState (p. 87)**

The state of the LAG. The following are the possible values:

- `requested`: The initial state of a LAG. The LAG stays in the requested state until the Letter of Authorization (LOA) is available.
- `pending`: The LAG has been approved and is being initialized.
- `available`: The network link is established and the LAG is ready for use.
- `down`: The network link is down.
- `deleting`: The LAG is being deleted.
- `deleted`: The LAG is deleted.
- `unknown`: The state of the LAG is not available.

**location (p. 87)**

The location of the LAG.

**macSecCapable (p. 87)**

Indicates whether the LAG supports MAC Security (MACsec).

**Type**: String
macSecKeys (p. 87)
The MAC Security (MACsec) security keys associated with the LAG.
Type: Array of MacSecKey (p. 241) objects

minimumLinks (p. 87)
The minimum number of physical dedicated connections that must be operational for the LAG itself to be operational.
Type: Integer

numberOfConnections (p. 87)
The number of physical dedicated connections bundled by the LAG, up to a maximum of 10.
Type: Integer

ownerAccount (p. 87)
The ID of the AWS account that owns the LAG.
Type: String

providerName (p. 87)
The name of the service provider associated with the LAG.
Type: String

region (p. 87)
The AWS Region where the connection is located.
Type: String

tags (p. 87)
The tags associated with the LAG.
Type: Array of Tag (p. 260) objects
Array Members: Minimum number of 1 item.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException
One or more parameters are not valid.
HTTP Status Code: 400

DirectConnectServerException
A server-side error occurred.
HTTP Status Code: 400

DuplicateTagKeysException
A tag key was specified more than once.
HTTP Status Code: 400
TooManyTagsException

You have reached the limit on the number of tags that can be assigned.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
CreatePrivateVirtualInterface

Creates a private virtual interface. A virtual interface is the VLAN that transports AWS Direct Connect traffic. A private virtual interface can be connected to either a Direct Connect gateway or a Virtual Private Gateway (VGW). Connecting the private virtual interface to a Direct Connect gateway enables the possibility for connecting to multiple VPCs, including VPCs in different AWS Regions. Connecting the private virtual interface to a VGW only provides access to a single VPC within the same Region.

Setting the MTU of a virtual interface to 9001 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. To check whether your connection supports jumbo frames, call DescribeConnections (p. 133). To check whether your virtual interface supports jumbo frames, call DescribeVirtualInterfaces (p. 171).

Request Syntax

```
{
  "connectionId": "string",
  "newPrivateVirtualInterface": {
    "addressFamily": "string",
    "amazonAddress": "string",
    "asn": number,
    "authKey": "string",
    "customerAddress": "string",
    "directConnectGatewayId": "string",
    "enableSiteLink": boolean,
    "mtu": number,
    "tags": [
      {
        "key": "string",
        "value": "string"
      }
    ],
    "virtualGatewayId": "string",
    "virtualInterfaceName": "string",
    "vlan": number
  }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 92)**

The ID of the connection.

Type: String

Required: Yes

**newPrivateVirtualInterface (p. 92)**

Information about the private virtual interface.

Type: NewPrivateVirtualInterface (p. 244) object

Required: Yes
Response Syntax

```json
{
    "addressFamily": "string",
    "amazonAddress": "string",
    "amazonSideAsn": number,
    "asn": number,
    "authKey": "string",
    "awsDeviceV2": "string",
    "awsLogicalDeviceId": "string",
    "bgpPeers": [
        {
            "addressFamily": "string",
            "amazonAddress": "string",
            "asn": number,
            "authKey": "string",
            "awsDeviceV2": "string",
            "awsLogicalDeviceId": "string",
            "bgpPeerId": "string",
            "bgpPeerState": "string",
            "bgpStatus": "string",
            "customerAddress": "string"
        }
    ],
    "connectionId": "string",
    "customerAddress": "string",
    "customerRouterConfig": "string",
    "directConnectGatewayId": "string",
    "jumboFrameCapable": boolean,
    "location": "string",
    "mtu": number,
    "ownerAccount": "string",
    "region": "string",
    "routeFilterPrefixes": [
        {
            "cidr": "string"
        }
    ],
    "siteLinkEnabled": boolean,
    "tags": [
        {
            "key": "string",
            "value": "string"
        }
    ],
    "virtualGatewayId": "string",
    "virtualInterfaceId": "string",
    "virtualInterfaceName": "string",
    "virtualInterfaceState": "string",
    "virtualInterfaceType": "string",
    "vlan": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**addressFamily (p. 93)**

The address family for the BGP peer.
**amazonAddress (p. 93)**

The IP address assigned to the Amazon interface.

Type: String

**amazonSideAsn (p. 93)**

The autonomous system number (ASN) for the Amazon side of the connection.

Type: Long

**asn (p. 93)**

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

The valid values are 1-2147483647.

Type: Integer

**authKey (p. 93)**

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

Type: String

**awsDeviceV2 (p. 93)**

The Direct Connect endpoint that terminates the physical connection.

Type: String

**awsLogicalDeviceId (p. 93)**

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

**bgpPeers (p. 93)**

The BGP peers configured on this virtual interface.

Type: Array of BGPPeer (p. 216) objects

**connectionId (p. 93)**

The ID of the connection.

Type: String

**customerAddress (p. 93)**

The IP address assigned to the customer interface.

Type: String

**customerRouterConfig (p. 93)**

The customer router configuration.

Type: String
**directConnectGatewayId (p. 93)**

The ID of the Direct Connect gateway.

Type: String

**jumboFrameCapable (p. 93)**

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean

**location (p. 93)**

The location of the connection.

Type: String

**mtu (p. 93)**

The maximum transmission unit (MTU), in bytes. The supported values are 1500 and 9001. The default value is 1500.

Type: Integer

**ownerAccount (p. 93)**

The ID of the AWS account that owns the virtual interface.

Type: String

**region (p. 93)**

The AWS Region where the virtual interface is located.

Type: String

**routeFilterPrefixes (p. 93)**

The routes to be advertised to the AWS network in this Region. Applies to public virtual interfaces.

Type: Array of RouteFilterPrefix (p. 257) objects

**siteLinkEnabled (p. 93)**

Indicates whether SiteLink is enabled.

Type: Boolean

**tags (p. 93)**

The tags associated with the virtual interface.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

**virtualGatewayId (p. 93)**

The ID of the virtual private gateway. Applies only to private virtual interfaces.

Type: String

**virtualInterfaceId (p. 93)**

The ID of the virtual interface.

Type: String
virtualInterfaceName (p. 93)

The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).

Type: String

virtualInterfaceState (p. 93)

The state of the virtual interface. The following are the possible values:

- **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.
- **down**: A virtual interface that is BGP down.
- **deleting**: A virtual interface is in this state immediately after calling DeleteVirtualInterface (p. 129) 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 enters the Rejected state.
- **unknown**: The state of the virtual interface is not available.

Type: String

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

virtualInterfaceType (p. 93)

The type of virtual interface. The possible values are private and public.

Type: String

vlan (p. 93)

The ID of the VLAN.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.
HTTP Status Code: 400

**DuplicateTagKeysException**

A tag key was specified more than once.

HTTP Status Code: 400

**TooManyTagsException**

You have reached the limit on the number of tags that can be assigned.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
CreatePublicVirtualInterface

Creates a 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 S3.

When creating an IPv6 public virtual interface (addressFamily is ipv6), leave the customer and amazon address fields blank to use auto-assigned IPv6 space. Custom IPv6 addresses are not supported.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 98)**

The ID of the connection.

Type: String

Required: Yes

**newPublicVirtualInterface (p. 98)**

Information about the public virtual interface.

Type: NewPublicVirtualInterface (p. 248) object

Required: Yes
Response Syntax

```json
{
    "addressFamily": "string",
    "amazonAddress": "string",
    "amazonSideAsn": number,
    "asn": number,
    "authKey": "string",
    "awsDeviceV2": "string",
    "awsLogicalDeviceId": "string",
    "bgpPeers": [
        {
            "addressFamily": "string",
            "amazonAddress": "string",
            "asn": number,
            "authKey": "string",
            "awsDeviceV2": "string",
            "awsLogicalDeviceId": "string",
            "bgpPeerId": "string",
            "bgpPeerState": "string",
            "bgpStatus": "string",
            "customerAddress": "string"
        }
    ],
    "connectionId": "string",
    "customerAddress": "string",
    "customerRouterConfig": "string",
    "directConnectGatewayId": "string",
    "jumboFrameCapable": boolean,
    "location": "string",
    "mtu": number,
    "ownerAccount": "string",
    "region": "string",
    "routeFilterPrefixes": [
        {
            "cidr": "string"
        }
    ],
    "siteLinkEnabled": boolean,
    "tags": [
        {
            "key": "string",
            "value": "string"
        }
    ],
    "virtualGatewayId": "string",
    "virtualInterfaceId": "string",
    "virtualInterfaceName": "string",
    "virtualInterfaceState": "string",
    "virtualInterfaceType": "string",
    "vlan": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**addressFamily (p. 99)**

The address family for the BGP peer.
**Response Elements**

**Type**: String

**Valid Values**: `ipv4` | `ipv6`

**amazonAddress (p. 99)**

The IP address assigned to the Amazon interface.

**Type**: String

**amazonSideAsn (p. 99)**

The autonomous system number (ASN) for the Amazon side of the connection.

**Type**: Long

**asn (p. 99)**

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

The valid values are 1-2147483647.

**Type**: Integer

**authKey (p. 99)**

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

**Type**: String

**awsDeviceV2 (p. 99)**

The Direct Connect endpoint that terminates the physical connection.

**Type**: String

**awsLogicalDeviceId (p. 99)**

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

**Type**: String

**bgpPeers (p. 99)**

The BGP peers configured on this virtual interface.

**Type**: Array of `BGPPeer (p. 216)` objects

**connectionId (p. 99)**

The ID of the connection.

**Type**: String

**customerAddress (p. 99)**

The IP address assigned to the customer interface.

**Type**: String

**customerRouterConfig (p. 99)**

The customer router configuration.

**Type**: String
directConnectGatewayId (p. 99)

The ID of the Direct Connect gateway.
Type: String

jumboFrameCapable (p. 99)

Indicates whether jumbo frames (9001 MTU) are supported.
Type: Boolean

location (p. 99)

The location of the connection.
Type: String

mtu (p. 99)

The maximum transmission unit (MTU), in bytes. The supported values are 1500 and 9001. The default value is 1500.
Type: Integer

ownerAccount (p. 99)

The ID of the AWS account that owns the virtual interface.
Type: String

region (p. 99)

The AWS Region where the virtual interface is located.
Type: String

routeFilterPrefixes (p. 99)

The routes to be advertised to the AWS network in this Region. Applies to public virtual interfaces.
Type: Array of RouteFilterPrefix (p. 257) objects

siteLinkEnabled (p. 99)

Indicates whether SiteLink is enabled.
Type: Boolean

tags (p. 99)

The tags associated with the virtual interface.
Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

virtualGatewayId (p. 99)

The ID of the virtual private gateway. Applies only to private virtual interfaces.
Type: String

virtualInterfaceId (p. 99)

The ID of the virtual interface.
Type: String
virtualInterfaceName (p. 99)

The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).

Type: String

virtualInterfaceState (p. 99)

The state of the virtual interface. The following are the possible values:

- 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.
- down: A virtual interface that is BGP down.
- deleting: A virtual interface is in this state immediately after calling DeleteVirtualInterface (p. 129) 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 enters the Rejected state.
- unknown: The state of the virtual interface is not available.

Type: String

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

virtualInterfaceType (p. 99)

The type of virtual interface. The possible values are private and public.

Type: String

vlan (p. 99)

The ID of the VLAN.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.
HTTP Status Code: 400

**DuplicateTagKeysException**

A tag key was specified more than once.

HTTP Status Code: 400

**TooManyTagsException**

You have reached the limit on the number of tags that can be assigned.

HTTP Status Code: 400

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
CreateTransitVirtualInterface

Creates a transit virtual interface. A transit virtual interface should be used to access one or more transit gateways associated with Direct Connect gateways. A transit virtual interface enables the connection of multiple VPCs attached to a transit gateway to a Direct Connect gateway.

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

A jumbo MTU value must be either 1500 or 8500. No other values will be accepted. Setting the MTU of a virtual interface to 8500 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. To check whether your connection supports jumbo frames, call DescribeConnections (p. 133). To check whether your virtual interface supports jumbo frames, call DescribeVirtualInterfaces (p. 171).

**Request Syntax**

```json
{
    "connectionId": "string",
    "newTransitVirtualInterface": {
        "addressFamily": "string",
        "amazonAddress": "string",
        "asn": number,
        "authKey": "string",
        "customerAddress": "string",
        "directConnectGatewayId": "string",
        "enableSiteLink": boolean,
        "mtu": number,
        "tags": [
            {
                "key": "string",
                "value": "string"
            }
        ],
        "virtualInterfaceName": "string",
        "vlan": number
    }
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 104)**

The ID of the connection.

- **Type:** String
- **Required:** Yes
newTransitVirtualInterface (p. 104)

Information about the transit virtual interface.

Type: NewTransitVirtualInterface (p. 252) object

Required: Yes

Response Syntax

```
{
    "virtualInterface": {
        "addressFamily": "string",
        "amazonAddress": "string",
        "amazonSideAsn": number,
        "asn": number,
        "authKey": "string",
        "awsDeviceV2": "string",
        "awsLogicalDeviceId": "string",
        "bgpPeers": [
            {
                "addressFamily": "string",
                "amazonAddress": "string",
                "asn": number,
                "authKey": "string",
                "awsDeviceV2": "string",
                "awsLogicalDeviceId": "string",
                "bgpPeerId": "string",
                "bgpPeerState": "string",
                "bgpStatus": "string",
                "customerAddress": "string"
            }
        ],
        "connectionId": "string",
        "customerAddress": "string",
        "customerRouterConfig": "string",
        "directConnectGatewayId": "string",
        "jumboFrameCapable": boolean,
        "location": "string",
        "mtu": number,
        "ownerAccount": "string",
        "region": "string",
        "routeFilterPrefixes": [
            {
                "cidr": "string"
            }
        ],
        "siteLinkEnabled": boolean,
        "tags": [
            {
                "key": "string",
                "value": "string"
            }
        ],
        "virtualGatewayId": "string",
        "virtualInterfaceId": "string",
        "virtualInterfaceName": "string",
        "virtualInterfaceState": "string",
        "virtualInterfaceType": "string",
        "vlan": number
    }
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**virtualInterface (p. 105)**

Information about a virtual interface.

Type: VirtualInterface (p. 262) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

**DuplicateTagKeysException**

A tag key was specified more than once.

HTTP Status Code: 400

**TooManyTagsException**

You have reached the limit on the number of tags that can be assigned.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DeleteBGPPeer

Deletes the specified BGP peer on the specified virtual interface with the specified customer address and ASN.

You cannot delete the last BGP peer from a virtual interface.

Request Syntax

```json
{
    "asn": number,
    "bgpPeerId": "string",
    "customerAddress": "string",
    "virtualInterfaceId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

`asn (p. 107)`

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

Type: Integer

Required: No

`bgpPeerId (p. 107)`

The ID of the BGP peer.

Type: String

Required: No

`customerAddress (p. 107)`

The IP address assigned to the customer interface.

Type: String

Required: No

`virtualInterfaceId (p. 107)`

The ID of the virtual interface.

Type: String

Required: No

Response Syntax

```json
{
}
```
"virtualInterface": {  
  "addressFamily": "string",
  "amazonAddress": "string",
  "amazonSideAsn": number,
  "asn": number,
  "authKey": "string",
  "awsDeviceV2": "string",
  "awsLogicalDeviceId": "string",
  "bgpPeers": [  
    {  
      "addressFamily": "string",
      "amazonAddress": "string",
      "asn": number,
      "authKey": "string",
      "awsDeviceV2": "string",
      "awsLogicalDeviceId": "string",
      "bgpPeerId": "string",
      "bgpPeerState": "string",
      "bgpStatus": "string",
      "customerAddress": "string"
    }
  ],
  "connectionId": "string",
  "customerAddress": "string",
  "customerRouterConfig": "string",
  "directConnectGatewayId": "string",
  "jumboFrameCapable": boolean,
  "location": "string",
  "mtu": number,
  "ownerAccount": "string",
  "region": "string",
  "routeFilterPrefixes": [
    {  
      "cidr": "string"
    }
  ],
  "siteLinkEnabled": boolean,
  "tags": [
    {  
      "key": "string",
      "value": "string"
    }
  ],
  "virtualGatewayId": "string",
  "virtualInterfaceId": "string",
  "virtualInterfaceName": "string",
  "virtualInterfaceState": "string",
  "virtualInterfaceType": "string",
  "vlan": number
}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

virtualInterface (p. 107)

The virtual interface.

Type: VirtualInterface (p. 262) object
Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DeleteConnection

Deletes the specified connection.

Deleting a connection only stops the AWS Direct Connect port hour and data transfer charges. If you are partnering with any third parties to connect with the AWS Direct Connect location, you must cancel your service with them separately.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 110)**

The ID of the connection.

Type: String

Required: Yes

Response Syntax

```json
{
    "awsDevice": "string",
    "awsDeviceV2": "string",
    "awsLogicalDeviceId": "string",
    "bandwidth": "string",
    "connectionId": "string",
    "connectionName": "string",
    "connectionState": "string",
    "encryptionMode": "string",
    "hasLogicalRedundancy": "string",
    "jumboFrameCapable": boolean,
    "lagId": "string",
    "loaIssueTime": number,
    "location": "string",
    "macSecCapable": boolean,
    "macSecKeys": [
        {
            "ckn": "string",
            "secretARN": "string",
            "startOn": "string",
            "state": "string"
        }
    ],
    "ownerAccount": "string",
    "partnerName": "string",
    "portEncryptionStatus": "string",
    "..."
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response. The following data is returned in JSON format by the service.

awsDevice (p. 110)

This parameter has been deprecated.

The Direct Connect endpoint on which the physical connection terminates.

Type: String

awsDeviceV2 (p. 110)

The Direct Connect endpoint that terminates the physical connection.

Type: String

awsLogicalDeviceId (p. 110)

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

bandwidth (p. 110)

The bandwidth of the connection.

Type: String

connectionId (p. 110)

The ID of the connection.

Type: String

connectionName (p. 110)

The name of the connection.

Type: String

connectionState (p. 110)

The state of the connection. The following are the possible values:

- 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.
• **deleting**: The connection is being deleted.
• **deleted**: The connection has been deleted.
• **rejected**: A hosted connection in the ordering state enters the rejected state if it is deleted by the customer.
• **unknown**: The state of the connection is not available.

Type: String

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

**encryptionMode (p. 110)**

The MAC Security (MACsec) connection encryption mode.

The valid values are no_encrypt, should_encrypt, and must_encrypt.

Type: String

**hasLogicalRedundancy (p. 110)**

Indicates whether the connection supports a secondary BGP peer in the same address family (IPv4/IPv6).

Type: String

Valid Values: unknown | yes | no

**jumboFrameCapable (p. 110)**

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean

**lagId (p. 110)**

The ID of the LAG.

Type: String

**loaIssueTime (p. 110)**

The time of the most recent call to DescribeLoa (p. 161) for this connection.

Type: Timestamp

**location (p. 110)**

The location of the connection.

Type: String

**macSecCapable (p. 110)**

Indicates whether the connection supports MAC Security (MACsec).

Type: Boolean

**macSecKeys (p. 110)**

The MAC Security (MACsec) security keys associated with the connection.
The ID of the AWS account that owns the connection.

Type: String

The name of the AWS Direct Connect service provider associated with the connection.

Type: String

The MAC Security (MACsec) port link status of the connection.

The valid values are Encryption Up, which means that there is an active Connection Key Name, or Encryption Down.

Type: String

The name of the service provider associated with the connection.

Type: String

The AWS Region where the connection is located.

Type: String

The tags associated with the connection.

Type: Array of Tag objects

Array Members: Minimum number of 1 item.

The ID of the VLAN.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors.

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DeleteDirectConnectGateway

Deletes the specified Direct Connect gateway. You must first delete all virtual interfaces that are attached to the Direct Connect gateway and disassociate all virtual private gateways associated with the Direct Connect gateway.

Request Syntax

```json
{
   "directConnectGatewayId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

directConnectGatewayId (p. 115)

The ID of the Direct Connect gateway.

Type: String

Required: Yes

Response Syntax

```json
{
   "directConnectGateway": {
      "amazonSideAsn": number,
      "directConnectGatewayId": "string",
      "directConnectGatewayName": "string",
      "directConnectGatewayState": "string",
      "ownerAccount": "string",
      "stateChangeError": "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

directConnectGateway (p. 115)

The Direct Connect gateway.

Type: DirectConnectGateway (p. 223) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).
DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DeleteDirectConnectGatewayAssociation

Deletes the association between the specified Direct Connect gateway and virtual private gateway.

We recommend that you specify the associationID to delete the association. Alternatively, if you own virtual gateway and a Direct Connect gateway association, you can specify the virtualGatewayId and directConnectGatewayId to delete an association.

Request Syntax

```
{
   "associationId": "string",
   "directConnectGatewayId": "string",
   "virtualGatewayId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**associationId** (p. 117)

   The ID of the Direct Connect gateway association.
   
   Type: String
   
   Required: No

**directConnectGatewayId** (p. 117)

   The ID of the Direct Connect gateway.
   
   Type: String
   
   Required: No

**virtualGatewayId** (p. 117)

   The ID of the virtual private gateway.
   
   Type: String
   
   Required: No

Response Syntax

```
{
   "directConnectGatewayAssociation": {
      "allowedPrefixesToDirectConnectGateway": [
         {
            "cidr": "string"
         }
      ],
      "associatedGateway": {
      }
   }
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response. The following data is returned in JSON format by the service.

directConnectGatewayAssociation (p. 117)

Information about the deleted association.

Type: DirectConnectGatewayAssociation (p. 225) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
See Also

- AWS SDK for Ruby V3
DeleteDirectConnectGatewayAssociationProposal

Deletes the association proposal request between the specified Direct Connect gateway and virtual private gateway or transit gateway.

Request Syntax

```
{
    "proposalId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**proposalId** (p. 120)

The ID of the proposal.

Type: String

Required: Yes

Response Syntax

```
{
    "directConnectGatewayAssociationProposal": {
        "associatedGateway": {
            "id": "string",
            "ownerAccount": "string",
            "region": "string",
            "type": "string"
        },
        "directConnectGatewayId": "string",
        "directConnectGatewayOwnerAccount": "string",
        "existingAllowedPrefixesToDirectConnectGateway": [
            {
                "cidr": "string"
            }
        ],
        "proposalId": "string",
        "proposalState": "string",
        "requestedAllowedPrefixesToDirectConnectGateway": [
            {
                "cidr": "string"
            }
        ]
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in JSON format by the service.

directConnectGatewayAssociationProposal (p. 120)

The ID of the associated gateway.

Type: DirectConnectGatewayAssociationProposal (p. 227) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DeleteInterconnect

Deletes the specified interconnect.

**Note**
Intended for use by AWS Direct Connect Partners only.

**Request Syntax**

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

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**interconnectId (p. 122)**

The ID of the interconnect.

Type: String

Required: Yes

**Response Syntax**

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

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

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

**interconnectState (p. 122)**

The state of the interconnect. The following are the possible values:

- 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 is approved, and is being initialized.
- available: The network link is up, and the interconnect is ready for use.
- down: The network link is down.
- deleting: The interconnect is being deleted.
- deleted: The interconnect is deleted.
- unknown: The state of the interconnect is not available.
Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DeleteLag

Deletes the specified link aggregation group (LAG). You cannot delete a LAG if it has active virtual interfaces or hosted connections.

Request Syntax

```json
{
   "lagId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**lagId (p. 124)**

The ID of the LAG.

Type: String

Required: Yes

Response Syntax

```json
{
   "allowsHostedConnections": boolean,
   "awsDevice": "string",
   "awsDeviceV2": "string",
   "awsLogicalDeviceId": "string",
   "connections": [
      {
         "awsDevice": "string",
         "awsDeviceV2": "string",
         "awsLogicalDeviceId": "string",
         "bandwidth": "string",
         "connectionId": "string",
         "connectionName": "string",
         "connectionState": "string",
         "encryptionMode": "string",
         "hasLogicalRedundancy": "string",
         "jumboFrameCapable": boolean,
         "lagId": "string",
         "loaIssueTime": number,
         "location": "string",
         "macSecCapable": boolean,
         "macSecKeys": [
            {
               "ckn": "string",
               "secretARN": "string",
               "startTime": "string",
               "state": "string"
            }
         ],
      }
   ],
   "awsLogicalDeviceId": "string",
   "connectionState": "string",
   "encryptionMode": "string",
   "hasLogicalRedundancy": "string",
   "jumboFrameCapable": boolean,
   "lagId": "string",
   "loaIssueTime": number,
   "location": "string",
   "macSecCapable": boolean,
   "macSecKeys": [
      {
         "ckn": "string",
         "secretARN": "string",
         "startTime": "string",
         "state": "string"
      }
   ]
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

allowsHostedConnections (p. 124)

Indicates whether the LAG can host other connections.

Type: Boolean

awsDevice (p. 124)

This parameter has been deprecated.

The AWS Direct Connect endpoint that hosts the LAG.

Type: String
awsDeviceV2 (p. 124)

The AWS Direct Connect endpoint that hosts the LAG.

Type: String

awsLogicalDeviceId (p. 124)

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

collections (p. 124)

The connections bundled by the LAG.

Type: Array of Connection (p. 218) objects

connectionsBandwidth (p. 124)

The individual bandwidth of the physical connections bundled by the LAG. The possible values are 1Gbps and 10Gbps.

Type: String

encryptionMode (p. 124)

The LAG MAC Security (MACsec) encryption mode.

The valid values are no_encrypt, should_encrypt, and must_encrypt.

Type: String

hasLogicalRedundancy (p. 124)

Indicates whether the LAG supports a secondary BGP peer in the same address family (IPv4/IPv6).

Type: String

Valid Values: unknown | yes | no

jumboFrameCapable (p. 124)

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean

lagId (p. 124)

The ID of the LAG.

Type: String

lagName (p. 124)

The name of the LAG.

Type: String

lagState (p. 124)

The state of the LAG. The following are the possible values:
- requested: The initial state of a LAG. The LAG stays in the requested state until the Letter of Authorization (LOA) is available.
- pending: The LAG has been approved and is being initialized.
- available: The network link is established and the LAG is ready for use.
- **down**: The network link is down.
- **deleting**: The LAG is being deleted.
- **deleted**: The LAG is deleted.
- **unknown**: The state of the LAG is not available.

  Type: String

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

  **location** (p. 124)

  The location of the LAG.

  Type: String

  **macSecCapable** (p. 124)

  Indicates whether the LAG supports MAC Security (MACsec).

  Type: Boolean

  **macSecKeys** (p. 124)

  The MAC Security (MACsec) security keys associated with the LAG.

  Type: Array of `MacSecKey` (p. 241) objects

  **minimumLinks** (p. 124)

  The minimum number of physical dedicated connections that must be operational for the LAG itself to be operational.

  Type: Integer

  **numberOfConnections** (p. 124)

  The number of physical dedicated connections bundled by the LAG, up to a maximum of 10.

  Type: Integer

  **ownerAccount** (p. 124)

  The ID of the AWS account that owns the LAG.

  Type: String

  **providerName** (p. 124)

  The name of the service provider associated with the LAG.

  Type: String

  **region** (p. 124)

  The AWS Region where the connection is located.

  Type: String

  **tags** (p. 124)

  The tags associated with the LAG.

  Type: Array of `Tag` (p. 260) objects

  Array Members: Minimum number of 1 item.
Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DeleteVirtualInterface

Deletes a virtual interface.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

virtualInterfaceId (p. 129)

The ID of the virtual interface.

Type: String

Required: Yes

Response Syntax

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

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

virtualInterfaceState (p. 129)

The state of the virtual interface. The following are the possible values:

- **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.
- **down**: A virtual interface that is BGP down.
- **deleting**: A virtual interface is in this state immediately after calling DeleteVirtualInterface (p. 129) 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 enters the **Rejected** state.
• **unknown**: The state of the virtual interface is not available.

Type: String

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

### Errors

For information about the errors that are common to all actions, see [Common Errors (p. 270)](#).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeConnectionLoa

Deprecated. Use DescribeLoa (p. 161) instead.

Gets the LOA-CFA for a connection.

The Letter of Authorization - Connecting Facility Assignment (LOA-CFA) is a document that your APN partner or service provider uses when establishing your cross connect to AWS at the colocation facility. For more information, see Requesting Cross Connects at AWS Direct Connect Locations in the AWS Direct Connect User Guide.

Request Syntax

```
{
  "connectionId": "string",
  "loaContentType": "string",
  "providerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

connectionId (p. 131)

  The ID of the connection.

  Type: String

  Required: Yes

loaContentType (p. 131)

  The standard media type for the LOA-CFA document. The only supported value is application/pdf.

  Type: String

  Valid Values: application/pdf

  Required: No

providerName (p. 131)

  The name of the APN partner or service provider who establishes connectivity on your behalf. If you specify this parameter, the LOA-CFA lists the provider name alongside your company name as the requester of the cross connect.

  Type: String

  Required: No

Response Syntax

```
{
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response. The following data is returned in JSON format by the service.

**loa** (p. 131)

The Letter of Authorization - Connecting Facility Assignment (LOA-CFA).

Type: Loa (p. 238) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeConnections

Displays the specified connection or all connections in this Region.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId** (p. 133)

The ID of the connection.

Type: String

Required: No

Response Syntax

```json
{
  "connections": [
    {
      "awsDevice": "string",
      "awsDeviceV2": "string",
      "awsLogicalDeviceId": "string",
      "bandwidth": "string",
      "connectionId": "string",
      "connectionName": "string",
      "connectionState": "string",
      "encryptionMode": "string",
      "hasLogicalRedundancy": "string",
      "jumboFrameCapable": boolean,
      "lagId": "string",
      "loaIssueTime": number,
      "location": "string",
      "macSecCapable": boolean,
      "macSecKeys": [
        {
          "ckn": "string",
          "secretARN": "string",
          "startOn": "string",
          "state": "string"
        }
      ],
      "ownerAccount": "string",
      "partnerName": "string",
      "portEncryptionStatus": "string",
      "providerName": "string",
      "region": "string"
    }
  ]
}
```
"tags": [ 
  { 
    "key": "string",
    "value": "string"
  }
],
"vlan": number
]

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

connections (p. 133)

  The connections.

  Type: Array of Connection (p. 218) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

  One or more parameters are not valid.

  HTTP Status Code: 400

DirectConnectServerException

  A server-side error occurred.

  HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeConnectionsOnInterconnect

Deprecated. Use DescribeHostedConnections (p. 151) instead.

Lists the connections that have been provisioned on the specified interconnect.

Note
Intended for use by AWS Direct Connect Partners only.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**interconnectId (p. 135)**

The ID of the interconnect.

- Type: String
- Required: Yes

Response Syntax

```json
{
    "connections": [
        {
            "awsDevice": "string",
            "awsDeviceV2": "string",
            "awsLogicalDeviceId": "string",
            "bandwidth": "string",
            "connectionId": "string",
            "connectionName": "string",
            "connectionState": "string",
            "encryptionMode": "string",
            "hasLogicalRedundancy": "string",
            "jumboFrameCapable": boolean,
            "lagId": "string",
            "loaIssueTime": number,
            "location": "string",
            "macSecCapable": boolean,
            "macSecKeys": [ ]
        }
    ]
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

connections (p. 135)

The connections.

Type: Array of Connection (p. 218) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
• AWS SDK for Ruby V3
DescribeCustomerMetadata

Get and view a list of customer agreements, along with their signed status and whether the customer is an NNIPartner, NNIPartnerV2, or a nonPartner.

Response Syntax

```json
{
    "agreements": [
        {
            "agreementName": "string",
            "status": "string"
        }
    ],
    "nniPartnerType": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

- **agreements (p. 138)**
  
  The list of customer agreements.
  
  Type: Array of CustomerAgreement (p. 222) objects

- **nniPartnerType (p. 138)**
  
  The type of network-to-network interface (NNI) partner. The partner type will be one of the following:
  
  - V1: This partner can only allocate 50Mbps, 100Mbps, 200Mbps, 300Mbps, 400Mbps, or 500Mbps subgigabit connections.
  - V2: This partner can only allocate 1GB, 2GB, 5GB, or 10GB hosted connections.
  - nonPartner: The customer is not a partner.
  
  Type: String
  
  Valid Values: v1 | v2 | nonPartner

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

- **DirectConnectClientException**
  
  One or more parameters are not valid.
  
  HTTP Status Code: 400

- **DirectConnectServerException**
  
  A server-side error occurred.
  
  HTTP Status Code: 400
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeDirectConnectGatewayAssociationProposals

Describes one or more association proposals for connection between a virtual private gateway or transit gateway and a Direct Connect gateway.

Request Syntax

```
{
  "associatedGatewayId": "string",
  "directConnectGatewayId": "string",
  "maxResults": number,
  "nextToken": "string",
  "proposalId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

`associatedGatewayId` (p. 140)

  The ID of the associated gateway.

  Type: String

  Required: No

`directConnectGatewayId` (p. 140)

  The ID of the Direct Connect gateway.

  Type: String

  Required: No

`maxResults` (p. 140)

  The maximum number of results to return with a single call. To retrieve the remaining results, make another call with the returned `nextToken` value.

  If `maxResults` is given a value larger than 100, only 100 results are returned.

  Type: Integer

  Required: No

`nextToken` (p. 140)

  The token for the next page of results.

  Type: String

  Required: No

`proposalId` (p. 140)

  The ID of the proposal.

  Type: String
Response Syntax

```json
{
  "directConnectGatewayAssociationProposals": [
    {
      "associatedGateway": {
        "id": "string",
        "ownerAccount": "string",
        "region": "string",
        "type": "string"
      },
      "directConnectGatewayId": "string",
      "directConnectGatewayOwnerAccount": "string",
      "existingAllowedPrefixesToDirectConnectGateway": [
        { "cidr": "string" }
      ],
      "proposalId": "string",
      "proposalState": "string",
      "requestedAllowedPrefixesToDirectConnectGateway": [
        { "cidr": "string" }
      ]
    }
  ]

  "nextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**directConnectGatewayAssociationProposals** (p. 141)

Describes the Direct Connect gateway association proposals.

Type: Array of **DirectConnectGatewayAssociationProposal** (p. 227) objects

**nextToken** (p. 141)

The token to use to retrieve the next page of results. This value is null when there are no more results to return.

Type: String

Errors

For information about the errors that are common to all actions, see **Common Errors** (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.
HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeDirectConnectGatewayAssociations

Lists the associations between your Direct Connect gateways and virtual private gateways and transit gateways. You must specify one of the following:

- A Direct Connect gateway
  The response contains all virtual private gateways and transit gateways associated with the Direct Connect gateway.
- A virtual private gateway
  The response contains the Direct Connect gateway.
- A transit gateway
  The response contains the Direct Connect gateway.
- A Direct Connect gateway and a virtual private gateway
  The response contains the association between the Direct Connect gateway and virtual private gateway.
- A Direct Connect gateway and a transit gateway
  The response contains the association between the Direct Connect gateway and transit gateway.

Request Syntax

```json
{
  "associatedGatewayId": "string",
  "associationId": "string",
  "directConnectGatewayId": "string",
  "maxResults": number,
  "nextToken": "string",
  "virtualGatewayId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**associatedGatewayId (p. 143)**

The ID of the associated gateway.

Type: String

Required: No

**associationId (p. 143)**

The ID of the Direct Connect gateway association.

Type: String

Required: No
**directConnectGatewayId (p. 143)**

The ID of the Direct Connect gateway.

Type: String

Required: No

**maxResults (p. 143)**

The maximum number of results to return with a single call. To retrieve the remaining results, make another call with the returned nextToken value.

If MaxResults is given a value larger than 100, only 100 results are returned.

Type: Integer

Required: No

**nextToken (p. 143)**

The token provided in the previous call to retrieve the next page.

Type: String

Required: No

**virtualGatewayId (p. 143)**

The ID of the virtual private gateway or transit gateway.

Type: String

Required: No

**Response Syntax**

```json
{
    "directConnectGatewayAssociations": [
        {
            "allowedPrefixesToDirectConnectGateway": [
                {"cidr": "string"}
            ],
            "associatedGateway": {
                "id": "string",
                "ownerAccount": "string",
                "region": "string",
                "type": "string"
            },
            "associationId": "string",
            "associationState": "string",
            "directConnectGatewayId": "string",
            "directConnectGatewayOwnerId": "string",
            "stateChangeError": "string",
            "virtualGatewayId": "string",
            "virtualGatewayOwnerId": "string",
            "virtualGatewayRegion": "string"
        }
    ],
    "nextToken": "string"
}
```

API Version 2012-10-25
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

directConnectGatewayAssociations (p. 144)

Information about the associations.

Type: Array of DirectConnectGatewayAssociation (p. 225) objects

nextToken (p. 144)

The token to retrieve the next page.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeDirectConnectGatewayAttachments

Lists the attachments between your Direct Connect gateways and virtual interfaces. You must specify a Direct Connect gateway, a virtual interface, or both. If you specify a Direct Connect gateway, the response contains all virtual interfaces attached to the Direct Connect gateway. If you specify a virtual interface, the response contains all Direct Connect gateways attached to the virtual interface. If you specify both, the response contains the attachment between the Direct Connect gateway and the virtual interface.

Request Syntax

```
{
  "directConnectGatewayId": "string",
  "maxResults": number,
  "nextToken": "string",
  "virtualInterfaceId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

directConnectGatewayId (p. 146)

  The ID of the Direct Connect gateway.

  Type: String

  Required: No

maxResults (p. 146)

  The maximum number of results to return with a single call. To retrieve the remaining results, make another call with the returned nextToken value.

  If MaxResults is given a value larger than 100, only 100 results are returned.

  Type: Integer

  Required: No

nextToken (p. 146)

  The token provided in the previous call to retrieve the next page.

  Type: String

  Required: No

virtualInterfaceId (p. 146)

  The ID of the virtual interface.

  Type: String

  Required: No
Response Syntax

```json
{
    "directConnectGatewayAttachments": [
        {
            "attachmentState": "string",
            "attachmentType": "string",
            "directConnectGatewayId": "string",
            "stateChangeError": "string",
            "virtualInterfaceId": "string",
            "virtualInterfaceOwnerAccount": "string",
            "virtualInterfaceRegion": "string"
        }
    ],
    "nextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

directConnectGatewayAttachments (p. 147)

The attachments.

Type: Array of DirectConnectGatewayAttachment (p. 229) objects

nextToken (p. 147)

The token to retrieve the next page.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerErrorException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
See Also

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeDirectConnectGateways

Lists all your Direct Connect gateways or only the specified Direct Connect gateway. Deleted Direct Connect gateways are not returned.

Request Syntax

```
{
  "directConnectGatewayId": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

directConnectGatewayId (p. 149)

  The ID of the Direct Connect gateway.

  Type: String

  Required: No

maxResults (p. 149)

  The maximum number of results to return with a single call. To retrieve the remaining results, make another call with the returned nextToken value.

  If MaxResults is given a value larger than 100, only 100 results are returned.

  Type: Integer

  Required: No

nextToken (p. 149)

  The token provided in the previous call to retrieve the next page.

  Type: String

  Required: No

Response Syntax

```
{
  "directConnectGateways": [
    {
      "amazonSideAsn": number,
      "directConnectGatewayId": "string",
      "directConnectGatewayName": "string",
      "directConnectGatewayState": "string",
      "ownerAccount": "string",
    }
  ]
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response. The following data is returned in JSON format by the service.

**directConnectGateways (p. 149)**

The Direct Connect gateways.

Type: Array of DirectConnectGateway (p. 223) objects

**nextToken (p. 149)**

The token to retrieve the next page.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeHostedConnections

Lists the hosted connections that have been provisioned on the specified interconnect or link aggregation group (LAG).

**Note**
Intended for use by AWS Direct Connect Partners only.

**Request Syntax**

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

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

- **connectionId (p. 151)**
  
  The ID of the interconnect or LAG.

  Type: String

  Required: Yes

**Response Syntax**

```
{
    "connections": [
        {
            "awsDevice": "string",
            "awsDeviceV2": "string",
            "awsLogicalDeviceId": "string",
            "bandwidth": "string",
            "connectionId": "string",
            "connectionName": "string",
            "connectionState": "string",
            "encryptionMode": "string",
            "hasLogicalRedundancy": "string",
            "jumboFrameCapable": boolean,
            "lagId": "string",
            "loaIssueTime": number,
            "location": "string",
            "macSecCapable": boolean,
            "macSecKeys": [
                {
                    "ckn": "string",
                    "secretARN": "string",
                    "startOn": "string",
                    "state": "string"
                }
            ],
            "ownerAccount": "string",

...}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

connections (p. 151)

The connections.

Type: Array of Connection (p. 218) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
• AWS SDK for Ruby V3
DescribeInterconnectLoa

Deprecated. Use DescribeLoa (p. 161) instead.

Gets the LOA-CFA for the specified interconnect.

The Letter of Authorization - Connecting Facility Assignment (LOA-CFA) is a document that is used when establishing your cross connect to AWS at the colocation facility. For more information, see Requesting Cross Connects at AWS Direct Connect Locations in the AWS Direct Connect User Guide.

Request Syntax

```json
{
   "interconnectId": "string",
   "loaContentType": "string",
   "providerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

`interconnectId` (p. 154)

The ID of the interconnect.

Type: String

Required: Yes

`loaContentType` (p. 154)

The standard media type for the LOA-CFA document. The only supported value is application/pdf.

Type: String

Valid Values: application/pdf

Required: No

`providerName` (p. 154)

The name of the service provider who establishes connectivity on your behalf. If you supply this parameter, the LOA-CFA lists the provider name alongside your company name as the requester of the cross connect.

Type: String

Required: No

Response Syntax

```json
{
   "loa": {
```

API Version 2012-10-25

154
"loaContent": blob,
"loaContentType": "string"
}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in JSON format by the service.

loa (p. 154)

The Letter of Authorization - Connecting Facility Assignment (LOA-CFA).

Type: Loa (p. 238) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeInterconnects

Lists the interconnects owned by the AWS account or only the specified interconnect.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

interconnectId (p. 156)

The ID of the interconnect.

Type: String

Required: No

Response Syntax

```json
{
    "interconnects": [
        {
            "awsDevice": "string",
            "awsDeviceV2": "string",
            "awsLogicalDeviceId": "string",
            "bandwidth": "string",
            "hasLogicalRedundancy": "string",
            "interconnectId": "string",
            "interconnectName": "string",
            "interconnectState": "string",
            "jumboFrameCapable": boolean,
            "lagId": "string",
            "loaIssueTime": number,
            "location": "string",
            "providerName": "string",
            "region": "string",
            "tags": [
                {
                    "key": "string",
                    "value": "string"
                }
            ]
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in JSON format by the service.

**interconnects (p. 156)**

The interconnects.

Type: Array of **Interconnect (p. 231)** objects

---

**Errors**

For information about the errors that are common to all actions, see [Common Errors (p. 270)](#).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

---

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeLags

Describes all your link aggregation groups (LAG) or the specified LAG.

**Request Syntax**

```json
{
  "lagId": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

`lagId` (p. 158)

- The ID of the LAG.
- Type: String
- Required: No

**Response Syntax**

```json
{
  "lags": [
    {
      "allowsHostedConnections": boolean,
      "awsDevice": "string",
      "awsDeviceV2": "string",
      "awsLogicalDeviceId": "string",
      "connections": [
        {
          "awsDevice": "string",
          "awsDeviceV2": "string",
          "awsLogicalDeviceId": "string",
          "bandwidth": "string",
          "connectionId": "string",
          "connectionName": "string",
          "connectionState": "string",
          "encryptionMode": "string",
          "hasLogicalRedundancy": "string",
          "jumboFrameCapable": boolean,
          "lagId": "string",
          "loaIssueTime": number,
          "location": "string",
          "macSecCapable": boolean,
          "macSecKeys": [
            {
              "ckn": "string",
              "secretARN": "string",
              "startTime": "string",
              "state": "string"
            }
          ]
        }
      ]
    }
  ]
}```

API Version 2012-10-25

158
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

lags (p. 158)

The LAGs.

Type: Array of Lag (p. 234) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).
DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeLoa

Gets the LOA-CFA for a connection, interconnect, or link aggregation group (LAG).

The Letter of Authorization - Connecting Facility Assignment (LOA-CFA) is a document that is used when establishing your cross connect to AWS at the colocation facility. For more information, see Requesting Cross Connects at AWS Direct Connect Locations in the AWS Direct Connect User Guide.

Request Syntax

```
{
  "connectionId": "string",
  "loaContentType": "string",
  "providerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 161)**

The ID of a connection, LAG, or interconnect.

Type: String

Required: Yes

**loaContentType (p. 161)**

The standard media type for the LOA-CFA document. The only supported value is application/pdf.

Type: String

Valid Values: application/pdf

Required: No

**providerName (p. 161)**

The name of the service provider who establishes connectivity on your behalf. If you specify this parameter, the LOA-CFA lists the provider name alongside your company name as the requester of the cross connect.

Type: String

Required: No

Response Syntax

```
{
  "loaContent": blob,
  "loaContentType": "string"
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**loaContent (p. 161)**

The binary contents of the LOA-CFA document.

Type: Base64-encoded binary data object

**loaContentType (p. 161)**

The standard media type for the LOA-CFA document. The only supported value is application/pdf.

Type: String

Valid Values: application/pdf

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeLocations

Lists the AWS Direct Connect locations in the current AWS Region. These are the locations that can be selected when calling CreateConnection (p. 66) or CreateInterconnect (p. 80).

Response Syntax

```json
{
   "locations": [
   {
       "availableMacSecPortSpeeds": [ "string" ],
       "availablePortSpeeds": [ "string" ],
       "availableProviders": [ "string" ],
       "locationCode": "string",
       "locationName": "string",
       "region": "string"
   }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

locations (p. 163)

The locations.

Type: Array of Location (p. 239) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
See Also

- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeRouterConfiguration

Details about the router.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**routerTypeIdentifier (p. 165)**

Identifies the router by a combination of vendor, platform, and software version. For example, CiscoSystemsInc-2900SeriesRouters-IOS124.

Type: String

Required: No

**virtualInterfaceId (p. 165)**

The ID of the virtual interface.

Type: String

Required: Yes

Response Syntax

```
{
  "customerRouterConfig": "string",
  "router": { 
    "platform": "string",
    "routerTypeIdentifier": "string",
    "software": "string",
    "vendor": "string",
    "xsltTemplateName": "string",
    "xsltTemplateNameForMacSec": "string"
  },
  "virtualInterfaceId": "string",
  "virtualInterfaceName": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.
customerRouterConfig (p. 165)

The customer router configuration.

Type: String

router (p. 165)

The details about the router.

Type: RouterType (p. 258) object

virtualInterfaceId (p. 165)

The ID assigned to the virtual interface.

Type: String

virtualInterfaceName (p. 165)

Provides the details about a virtual interface's router.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeTags

Describes the tags associated with the specified AWS Direct Connect resources.

Request Syntax

```
{
    "resourceArns": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**resourceArns (p. 167)**

The Amazon Resource Names (ARNs) of the resources.

Type: Array of strings

Required: Yes

Response Syntax

```
{
    "resourceTags": [
        {
            "resourceArn": "string",
            "tags": [
                {
                    "key": "string",
                    "value": "string"
                }
            ]
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**resourceTags (p. 167)**

Information about the tags.

Type: Array of ResourceTag (p. 256) objects
Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeVirtualGateways

Lists the virtual private gateways owned by the AWS account.

You can create one or more AWS Direct Connect private virtual interfaces linked to a virtual private gateway.

Response Syntax

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

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

`virtualGateways (p. 169)`

The virtual private gateways.

Type: Array of VirtualGateway (p. 261) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
See Also

- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeVirtualInterfaces

Displays all virtual interfaces for an AWS account. Virtual interfaces deleted fewer than 15 minutes before you make the request are also returned. If you specify a connection ID, only the virtual interfaces associated with the connection are returned. If you specify a virtual interface ID, then only a single virtual interface is returned.

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

Request Syntax

```json
{
    "connectionId": "string",
    "virtualInterfaceId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 171)**

The ID of the connection.

Type: String

Required: No

**virtualInterfaceId (p. 171)**

The ID of the virtual interface.

Type: String

Required: No

Response Syntax

```json
{
    "virtualInterfaces": [
        {
            "addressFamily": "string",
            "amazonAddress": "string",
            "amazonSideAsn": number,
            "asn": number,
            "authKey": "string",
            "awsDeviceV2": "string",
            "awsLogicalDeviceId": "string",
            "bgpPeers": [
                {
                    "addressFamily": "string",
                    "amazonAddress": "string",
                    "asn": number,
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**virtualInterfaces (p. 171)**

The virtual interfaces

Type: Array of VirtualInterface (p. 262) objects

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400
DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DisassociateConnectionFromLag

Disassociates a connection from a link aggregation group (LAG). The connection is interrupted and re-established as a standalone connection (the connection is not deleted; to delete the connection, use the DeleteConnection (p. 110) request). If the LAG has associated virtual interfaces or hosted connections, they remain associated with the LAG. A disassociated connection owned by an AWS Direct Connect Partner is automatically converted to an interconnect.

If disassociating the connection would cause the LAG to fall below its setting for minimum number of operational connections, the request fails, except when it's the last member of the LAG. If all connections are disassociated, the LAG continues to exist as an empty LAG with no physical connections.

Request Syntax

```json
{
  "connectionId": "string",
  "lagId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 174)**

The ID of the connection.

Type: String

Required: Yes

**lagId (p. 174)**

The ID of the LAG.

Type: String

Required: Yes

Response Syntax

```json
{
  "awsDevice": "string",
  "awsDeviceV2": "string",
  "awsLogicalDeviceId": "string",
  "bandwidth": "string",
  "connectionId": "string",
  "connectionName": "string",
  "connectionState": "string",
  "encryptionMode": "string",
  "hasLogicalRedundancy": "string",
  "jumboFrameCapable": "boolean",
  "lagId": "string",
  "loaIssueTime": "number",
}```
"location": "string",
"macSecCapable": boolean,
"macSecKeys": [ 
  { 
    "ckn": "string",
    "secretARN": "string",
    "startOn": "string",
    "state": "string"
  }
],
"ownerAccount": "string",
"partnerName": "string",
"providerName": "string",
"region": "string",
"tags": [ 
  { 
    "key": "string",
    "value": "string"
  }
],
"vlan": number
}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

awsDevice (p. 174)

   *This parameter has been deprecated.*

   The Direct Connect endpoint on which the physical connection terminates.

   Type: String

awsDeviceV2 (p. 174)

   The Direct Connect endpoint that terminates the physical connection.

   Type: String

awsLogicalDeviceId (p. 174)

   The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

   Type: String

bandwidth (p. 174)

   The bandwidth of the connection.

   Type: String

connectionId (p. 174)

   The ID of the connection.

   Type: String

connectionName (p. 174)

   The name of the connection.
connectionState (p. 174)

The state of the connection. The following are the possible values:

- **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.
- **deleting**: The connection is being deleted.
- **deleted**: The connection has been deleted.
- **rejected**: A hosted connection in the ordering state enters the rejected state if it is deleted by the customer.
- **unknown**: The state of the connection is not available.

encryptionMode (p. 174)

The MAC Security (MACsec) connection encryption mode.

The valid values are **no_encrypt**, **should_encrypt**, and **must_encrypt**.

hasLogicalRedundancy (p. 174)

Indicates whether the connection supports a secondary BGP peer in the same address family (IPv4/IPv6).

jumboFrameCapable (p. 174)

Indicates whether jumbo frames (9001 MTU) are supported.

lagId (p. 174)

The ID of the LAG.

loaIssueTime (p. 174)

The time of the most recent call to DescribeLoa (p. 161) for this connection.

location (p. 174)

The location of the connection.
macSecCapable (p. 174)
  Indicates whether the connection supports MAC Security (MACsec).
  Type: Boolean

macSecKeys (p. 174)
  The MAC Security (MACsec) security keys associated with the connection.
  Type: Array of MacSecKey (p. 241) objects

ownerAccount (p. 174)
  The ID of the AWS account that owns the connection.
  Type: String

partnerName (p. 174)
  The name of the AWS Direct Connect service provider associated with the connection.
  Type: String

portEncryptionStatus (p. 174)
  The MAC Security (MACsec) port link status of the connection.
  The valid values are Encryption Up, which means that there is an active Connection Key Name, or Encryption Down.
  Type: String

providerName (p. 174)
  The name of the service provider associated with the connection.
  Type: String

region (p. 174)
  The AWS Region where the connection is located.
  Type: String

tags (p. 174)
  The tags associated with the connection.
  Type: Array of Tag (p. 260) objects
  Array Members: Minimum number of 1 item.

vlan (p. 174)
  The ID of the VLAN.
  Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException
  One or more parameters are not valid.
HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DisassociateMacSecKey

Removes the association between a MAC Security (MACsec) security key and an AWS Direct Connect dedicated connection.

Request Syntax

```json
{
  "connectionId": "string",
  "secretARN": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

connectionId (p. 179)

The ID of the dedicated connection (dxcon-xxxx), or the ID of the LAG (dxlag-xxxx).

You can use DescribeConnections (p. 133) or DescribeLags (p. 158) to retrieve connection ID.

Type: String

Required: Yes

secretARN (p. 179)

The Amazon Resource Name (ARN) of the MAC Security (MACsec) secret key.

You can use DescribeConnections (p. 133) to retrieve the ARN of the MAC Security (MACsec) secret key.

Type: String

Required: Yes

Response Syntax

```json
{
  "connectionId": "string",
  "macSecKeys": [
    
    {
      "ckn": "string",
      "secretARN": "string",
      "startOn": "string",
      "state": "string"
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in JSON format by the service.

connectionId (p. 179)

The ID of the dedicated connection (dxcon-xxxx), or the ID of the LAG (dxlag-xxxx).

Type: String

macSecKeys (p. 179)

The MAC Security (MACsec) security keys no longer associated with the dedicated connection.

Type: Array of MacSecKey (p. 241) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
ListVirtualInterfaceTestHistory

Lists the virtual interface failover test history.

Request Syntax

```
{
  "bgpPeers": [ "string" ],
  "maxResults": number,
  "nextToken": "string",
  "status": "string",
  "testId": "string",
  "virtualInterfaceId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

bgpPeers (p. 181)

The BGP peers that were placed in the DOWN state during the virtual interface failover test.

Type: Array of strings

Required: No

maxResults (p. 181)

The maximum number of results to return with a single call. To retrieve the remaining results, make another call with the returned nextToken value.

If MaxResults is given a value larger than 100, only 100 results are returned.

Type: Integer

Required: No

nextToken (p. 181)

The token for the next page of results.

Type: String

Required: No

status (p. 181)

The status of the virtual interface failover test.

Type: String

Required: No

testId (p. 181)

The ID of the virtual interface failover test.
Type: String
Required: No

**virtualInterfaceId** (p. 181)
The ID of the virtual interface that was tested.

Type: String
Required: No

**Response Syntax**

```
{
   "nextToken": "string",
   "virtualInterfaceTestHistory": [
      {
         "bgpPeers": [ "string" ],
         "endTime": number,
         "ownerAccount": "string",
         "startTime": number,
         "status": "string",
         "testDurationInMinutes": number,
         "testId": "string",
         "virtualInterfaceId": "string"
      }
   ]
}
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

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

**nextToken** (p. 182)
The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

Type: String

**virtualInterfaceTestHistory** (p. 182)
The ID of the tested virtual interface.

Type: Array of VirtualInterfaceTestHistory (p. 266) objects

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400
DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
StartBgpFailoverTest

Starts the virtual interface failover test that verifies your configuration meets your resiliency requirements by placing the BGP peering session in the DOWN state. You can then send traffic to verify that there are no outages.

You can run the test on public, private, transit, and hosted virtual interfaces.

You can use ListVirtualInterfaceTestHistory to view the virtual interface test history.

If you need to stop the test before the test interval completes, use StopBgpFailoverTest.

Request Syntax

```json
{
   "bgpPeers": [ "string" ],
   "testDurationInMinutes": number,
   "virtualInterfaceId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**bgpPeers (p. 184)**

The BGP peers to place in the DOWN state.

Type: Array of strings

Required: No

**testDurationInMinutes (p. 184)**

The time in minutes that the virtual interface failover test will last.

Maximum value: 1,440 minutes (24 hours).

Default: 180 minutes (3 hours).

Type: Integer

Required: No

**virtualInterfaceId (p. 184)**

The ID of the virtual interface you want to test.

Type: String

Required: Yes

Response Syntax

```json
{
}
```
"virtualInterfaceTest": {  
  "bgpPeers": [ "string" ],  
  "endTime": number,  
  "ownerAccount": "string",  
  "startTime": number,  
  "status": "string",  
  "testDurationInMinutes": number,  
  "testId": "string",  
  "virtualInterfaceId": "string"  
}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**virtualInterfaceTest (p. 184)**

Information about the virtual interface failover test.

Type: VirtualInterfaceTestHistory (p. 266) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
StopBgpFailoverTest

Stops the virtual interface failover test.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**virtualInterfaceId (p. 186)**

The ID of the virtual interface you no longer want to test.

Type: String

Required: Yes

Response Syntax

```
{
    "virtualInterfaceTest": {
        "bgpPeers": [ "string" ],
        "endTime": number,
        "ownerAccount": "string",
        "startTime": number,
        "status": "string",
        "testDurationInMinutes": number,
        "testId": "string",
        "virtualInterfaceId": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**virtualInterfaceTest (p. 186)**

Information about the virtual interface failover test.

Type: VirtualInterfaceTestHistory (p. 266) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).
DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
TagResource

Adds the specified tags to the specified AWS Direct Connect resource. Each resource can have a maximum of 50 tags.

Each tag consists of a key and an optional value. If a tag with the same key is already associated with the resource, this action updates its value.

Request Syntax

```
{
  "resourceArn": "string",
  "tags": [ 
    
  ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

resourceArn (p. 188)

The Amazon Resource Name (ARN) of the resource.

Type: String

Required: Yes

tags (p. 188)

The tags to add.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.
HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

**DuplicateTagKeysException**

A tag key was specified more than once.

HTTP Status Code: 400

**TooManyTagsException**

You have reached the limit on the number of tags that can be assigned.

HTTP Status Code: 400

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
UntagResource

Removes one or more tags from the specified AWS Direct Connect resource.

Request Syntax

```json
{
   "resourceArn": "string",
   "tagKeys": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**resourceArn (p. 190)**

The Amazon Resource Name (ARN) of the resource.

Type: String

Required: Yes

**tagKeys (p. 190)**

The tag keys of the tags to remove.

Type: Array of strings


Pattern: ^([\p{L}\p{Z}\p{N}_\-/:=+\@]*)$

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
UpdateConnection

Updates the AWS Direct Connect dedicated connection configuration.

You can update the following parameters for a connection:

- The connection name
- The connection's MAC Security (MACsec) encryption mode.

Request Syntax

```
{
  "connectionId": "string",
  "connectionName": "string",
  "encryptionMode": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**connectionId (p. 192)**

The ID of the dedicated connection.

You can use DescribeConnections (p. 133) to retrieve the connection ID.

Type: String

Required: Yes

**connectionName (p. 192)**

The name of the connection.

Type: String

Required: No

**encryptionMode (p. 192)**

The connection MAC Security (MACsec) encryption mode.

The valid values are no_encrypt, should_encrypt, and must_encrypt.

Type: String

Required: No

Response Syntax

```
{
  "awsDevice": "string",
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

**awsDevice (p. 192)**

> *This parameter has been deprecated.*

The Direct Connect endpoint on which the physical connection terminates.

Type: String

**awsDeviceV2 (p. 192)**

The Direct Connect endpoint that terminates the physical connection.

Type: String

**awsLogicalDeviceId (p. 192)**

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

**bandwidth (p. 192)**

The bandwidth of the connection.
Type: String
connectionId (p. 192)

The ID of the connection.

Type: String
connectionName (p. 192)

The name of the connection.

Type: String
connectionState (p. 192)

The state of the connection. The following are the possible values:
- **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.
- **deleting**: The connection is being deleted.
- **deleted**: The connection has been deleted.
- **rejected**: A hosted connection in the ordering state enters the rejected state if it is deleted by the customer.
- **unknown**: The state of the connection is not available.

Type: String
encryptionMode (p. 192)

The MAC Security (MACsec) connection encryption mode.

The valid values are no_encrypt, should_encrypt, and must_encrypt.

Type: String
hasLogicalRedundancy (p. 192)

Indicates whether the connection supports a secondary BGP peer in the same address family (IPv4/IPv6).

Type: String
jumboFrameCapable (p. 192)

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean
lagId (p. 192)

The ID of the LAG.
Type: String  
**loaIssueTime (p. 192)**  
The time of the most recent call to DescribeLoa (p. 161) for this connection.

Type: Timestamp  
**location (p. 192)**  
The location of the connection.

Type: String  
**macSecCapable (p. 192)**  
Indicates whether the connection supports MAC Security (MACsec).

Type: Boolean  
**macSecKeys (p. 192)**  
The MAC Security (MACsec) security keys associated with the connection.

Type: Array of [MacSecKey (p. 241)] objects  
**ownerAccount (p. 192)**  
The ID of the AWS account that owns the connection.

Type: String  
**partnerName (p. 192)**  
The name of the AWS Direct Connect service provider associated with the connection.

Type: String  
**portEncryptionStatus (p. 192)**  
The MAC Security (MACsec) port link status of the connection.

The valid values are *Encryption Up*, which means that there is an active Connection Key Name, or *Encryption Down*.

Type: String  
**providerName (p. 192)**  
The name of the service provider associated with the connection.

Type: String  
**region (p. 192)**  
The AWS Region where the connection is located.

Type: String  
**tags (p. 192)**  
The tags associated with the connection.

Type: Array of [Tag (p. 260)] objects  
Array Members: Minimum number of 1 item.

**vlan (p. 192)**  
The ID of the VLAN.
Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
UpdateDirectConnectGateway

Updates the name of a current Direct Connect gateway.

Request Syntax

```json
{
  "directConnectGatewayId": "string",
  "newDirectConnectGatewayName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

directConnectGatewayId (p. 197)

The ID of the Direct Connect gateway to update.

Type: String

Required: Yes

newDirectConnectGatewayName (p. 197)

The new name for the Direct Connect gateway.

Type: String

Required: Yes

Response Syntax

```json
{
  "directConnectGateway": {
    "amazonSideAsn": number,
    "directConnectGatewayId": "string",
    "directConnectGatewayName": "string",
    "directConnectGatewayState": "string",
    "ownerAccount": "string",
    "stateChangeError": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

directConnectGateway (p. 197)

Information about a Direct Connect gateway, which enables you to connect virtual interfaces and virtual private gateway or transit gateways.
Type: DirectConnectGateway (p. 223) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
UpdateDirectConnectGatewayAssociation

Updates the specified attributes of the Direct Connect gateway association.
Add or remove prefixes from the association.

Request Syntax

```json
{
    "addAllowedPrefixesToDirectConnectGateway": [
        {
            "cidr": "string"
        }
    ],
    "associationId": "string",
    "removeAllowedPrefixesToDirectConnectGateway": [
        {
            "cidr": "string"
        }
    ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

- **addAllowedPrefixesToDirectConnectGateway (p. 199)**
  The Amazon VPC prefixes to advertise to the Direct Connect gateway.
  Type: Array of RouteFilterPrefix (p. 257) objects
  Required: No

- **associationId (p. 199)**
  The ID of the Direct Connect gateway association.
  Type: String
  Required: No

- **removeAllowedPrefixesToDirectConnectGateway (p. 199)**
  The Amazon VPC prefixes to no longer advertise to the Direct Connect gateway.
  Type: Array of RouteFilterPrefix (p. 257) objects
  Required: No

Response Syntax

```json
{
    "directConnectGatewayAssociation": {
```

API Version 2012-10-25
199
"allowedPrefixesToDirectConnectGateway": [  
  
  ],

"associatedGateway": {  
  "id": "string",
  "ownerAccount": "string",
  "region": "string",
  "type": "string"

},

"associationId": "string",
"associationState": "string",
"directConnectGatewayId": "string",
"directConnectGatewayOwnerAccount": "string",
"stateChangeError": "string",
"virtualGatewayId": "string",
"virtualGatewayOwnerAccount": "string",
"virtualGatewayRegion": "string"

}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

directConnectGatewayAssociation (p. 199)

Information about an association between a Direct Connect gateway and a virtual private gateway or transit gateway.

Type: DirectConnectGatewayAssociation (p. 225) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java V2
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V3
UpdateLag

Updates the attributes of the specified link aggregation group (LAG).

You can update the following LAG attributes:

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

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

- The tags

**Note**

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

Request Syntax

```json
{
    "encryptionMode": "string",
    "lagId": "string",
    "lagName": "string",
    "minimumLinks": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**encryptionMode (p. 202)**

The LAG MAC Security (MACsec) encryption mode.

AWS applies the value to all connections which are part of the LAG.

Type: String

Required: No

**lagId (p. 202)**

The ID of the LAG.

Type: String

Required: Yes

**lagName (p. 202)**

The name of the LAG.

Type: String
Required: No

**minimumLinks (p. 202)**

The minimum number of physical connections that must be operational for the LAG itself to be operational.

Type: Integer

Required: No

## Response Syntax

```json
{
    "allowsHostedConnections": boolean,
    "awsDevice": "string",
    "awsDeviceV2": "string",
    "awsLogicalDeviceId": "string",
    "connections": [
        {
            "awsDevice": "string",
            "awsDeviceV2": "string",
            "awsLogicalDeviceId": "string",
            "bandwidth": "string",
            "connectionId": "string",
            "connectionName": "string",
            "connectionState": "string",
            "encryptionMode": "string",
            "hasLogicalRedundancy": "string",
            "jumboFrameCapable": boolean,
            "lagId": "string",
            "loaIssueTime": number,
            "location": "string",
            "macSecCapable": boolean,
            "macSecKeys": [
                {
                    "ckn": "string",
                    "secretARN": "string",
                    "startOn": "string",
                    "state": "string"
                }
            ],
            "ownerAccount": "string",
            "partnerName": "string",
            "portEncryptionStatus": "string",
            "providerName": "string",
            "region": "string",
            "tags": [
                {
                    "key": "string",
                    "value": "string"
                }
            ],
            "vlan": number
        }
    ],
    "connectionsBandwidth": "string",
    "encryptionMode": "string",
    "hasLogicalRedundancy": "string",
    "jumboFrameCapable": boolean,
    "lagId": "string",
    "lagName": "string",
    "lagState": "string"
}
```
"location": "string",
"macSecCapable": boolean,
"macSecKeys": [
  {
    "ckn": "string",
    "secretARN": "string",
    "startOn": "string",
    "state": "string"
  }
],
"minimumLinks": number,
"numberOfConnections": number,
"ownerAccount": "string",
"providerName": "string",
"region": "string",
"tags": [
  {
    "key": "string",
    "value": "string"
  }
]
}

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

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

**allowsHostedConnections (p. 203)**

Indicates whether the LAG can host other connections.

Type: Boolean

**awsDevice (p. 203)**

*This parameter has been deprecated.*

The AWS Direct Connect endpoint that hosts the LAG.

Type: String

**awsDeviceV2 (p. 203)**

The AWS Direct Connect endpoint that hosts the LAG.

Type: String

**awsLogicalDeviceId (p. 203)**

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

**connections (p. 203)**

The connections bundled by the LAG.

Type: Array of Connection (p. 218) objects

**connectionsBandwidth (p. 203)**

The individual bandwidth of the physical connections bundled by the LAG. The possible values are 1Gbps and 10Gbps.
AWS Direct Connect API Reference
Response Elements

**Type: String**

**encryptionMode (p. 203)**

The LAG MAC Security (MACsec) encryption mode.

The valid values are no_encrypt, should_encrypt, and must_encrypt.

**Type: String**

**hasLogicalRedundancy (p. 203)**

Indicates whether the LAG supports a secondary BGP peer in the same address family (IPv4/IPv6).

**Type: String**

Valid Values: unknown | yes | no

**jumboFrameCapable (p. 203)**

Indicates whether jumbo frames (9001 MTU) are supported.

**Type: Boolean**

**lagId (p. 203)**

The ID of the LAG.

**Type: String**

**lagName (p. 203)**

The name of the LAG.

**Type: String**

**lagState (p. 203)**

The state of the LAG. The following are the possible values:

- requested: The initial state of a LAG. The LAG stays in the requested state until the Letter of Authorization (LOA) is available.
- pending: The LAG has been approved and is being initialized.
- available: The network link is established and the LAG is ready for use.
- down: The network link is down.
- deleting: The LAG is being deleted.
- deleted: The LAG is deleted.
- unknown: The state of the LAG is not available.

**Type: String**

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

**location (p. 203)**

The location of the LAG.

**Type: String**

**macSecCapable (p. 203)**

Indicates whether the LAG supports MAC Security (MACsec).

**Type: Boolean**
macSecKeys (p. 203)

The MAC Security (MACsec) security keys associated with the LAG.

Type: Array of MacSecKey (p. 241) objects

minimumLinks (p. 203)

The minimum number of physical dedicated connections that must be operational for the LAG itself to be operational.

Type: Integer

numberOfConnections (p. 203)

The number of physical dedicated connections bundled by the LAG, up to a maximum of 10.

Type: Integer

ownerAccount (p. 203)

The ID of the AWS account that owns the LAG.

Type: String

providerName (p. 203)

The name of the service provider associated with the LAG.

Type: String

region (p. 203)

The AWS Region where the connection is located.

Type: String

tags (p. 203)

The tags associated with the LAG.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

DirectConnectClientException

One or more parameters are not valid.

HTTP Status Code: 400

DirectConnectServerException

A server-side error occurred.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:
• AWS Command Line Interface
• AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java V2
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V3
UpdateVirtualInterfaceAttributes

Updates the specified attributes of the specified virtual private interface.

Setting the MTU of a virtual interface to 9001 (jumbo frames) can cause an update to the underlying physical connection if it wasn't updated to support jumbo frames. Updating the connection disrupts network connectivity for all virtual interfaces associated with the connection for up to 30 seconds. To check whether your connection supports jumbo frames, call DescribeConnections (p. 133). To check whether your virtual interface supports jumbo frames, call DescribeVirtualInterfaces (p. 171).

Request Syntax

```
{
    "enableSiteLink": boolean,
    "mtu": number,
    "virtualInterfaceId": "string",
    "virtualInterfaceName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 268).

The request accepts the following data in JSON format.

**enableSiteLink (p. 208)**

Indicates whether to enable or disable SiteLink.

Type: Boolean

Required: No

**mtu (p. 208)**

The maximum transmission unit (MTU), in bytes. The supported values are 1500 and 9001. The default value is 1500.

Type: Integer

Required: No

**virtualInterfaceId (p. 208)**

The ID of the virtual private interface.

Type: String

Required: Yes

**virtualInterfaceName (p. 208)**

The name of the virtual private interface.

Type: String

Required: No
Response Syntax

```json
{
   "addressFamily": "string",
   "amazonAddress": "string",
   "amazonSideAsn": number,
   "asn": number,
   "authKey": "string",
   "awsDeviceV2": "string",
   "awsLogicalDeviceId": "string",
   "bgpPeers": [
      {
         "addressFamily": "string",
         "amazonAddress": "string",
         "asn": number,
         "authKey": "string",
         "awsDeviceV2": "string",
         "awsLogicalDeviceId": "string",
         "bgpPeerId": "string",
         "bgpPeerState": "string",
         "bgpStatus": "string",
         "customerAddress": "string"
      }
   ],
   "connectionId": "string",
   "customerAddress": "string",
   "customerRouterConfig": "string",
   "directConnectGatewayId": "string",
   "jumboFrameCapable": boolean,
   "location": "string",
   "mtu": number,
   "ownerAccount": "string",
   "region": "string",
   "routeFilterPrefixes": [
      {
         "cidr": "string"
      }
   ],
   "siteLinkEnabled": boolean,
   "tags": [
      {
         "key": "string",
         "value": "string"
      }
   ],
   "virtualGatewayId": "string",
   "virtualInterfaceId": "string",
   "virtualInterfaceName": "string",
   "virtualInterfaceState": "string",
   "virtualInterfaceType": "string",
   "vlan": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

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

addressFamily (p. 209)

The address family for the BGP peer.
Response Elements

Type: String

Valid Values: ipv4 | ipv6

**amazonAddress (p. 209)**

The IP address assigned to the Amazon interface.

Type: String

**amazonSideAsn (p. 209)**

The autonomous system number (ASN) for the Amazon side of the connection.

Type: Long

**asn (p. 209)**

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

The valid values are 1-2147483647.

Type: Integer

**authKey (p. 209)**

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

Type: String

**awsDeviceV2 (p. 209)**

The Direct Connect endpoint that terminates the physical connection.

Type: String

**awsLogicalDeviceId (p. 209)**

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

**bgpPeers (p. 209)**

The BGP peers configured on this virtual interface.

Type: Array of BGPPeer (p. 216) objects

**connectionId (p. 209)**

The ID of the connection.

Type: String

**customerAddress (p. 209)**

The IP address assigned to the customer interface.

Type: String

**customerRouterConfig (p. 209)**

The customer router configuration.

Type: String
**directConnectGatewayId (p. 209)**

The ID of the Direct Connect gateway.

Type: String

**jumboFrameCapable (p. 209)**

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean

**location (p. 209)**

The location of the connection.

Type: String

**mtu (p. 209)**

The maximum transmission unit (MTU), in bytes. The supported values are 1500 and 9001. The default value is 1500.

Type: Integer

**ownerAccount (p. 209)**

The ID of the AWS account that owns the virtual interface.

Type: String

**region (p. 209)**

The AWS Region where the virtual interface is located.

Type: String

**routeFilterPrefixes (p. 209)**

The routes to be advertised to the AWS network in this Region. Applies to public virtual interfaces.

Type: Array of RouteFilterPrefix (p. 257) objects

**siteLinkEnabled (p. 209)**

Indicates whether SiteLink is enabled.

Type: Boolean

**tags (p. 209)**

The tags associated with the virtual interface.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

**virtualGatewayId (p. 209)**

The ID of the virtual private gateway. Applies only to private virtual interfaces.

Type: String

**virtualInterfaceId (p. 209)**

The ID of the virtual interface.

Type: String
virtualInterfaceName (p. 209)

The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).

Type: String

virtualInterfaceState (p. 209)

The state of the virtual interface. The following are the possible values:

- **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.
- **down**: A virtual interface that is BGP down.
- **deleting**: A virtual interface is in this state immediately after calling DeleteVirtualInterface (p. 129) 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 enters the Rejected state.
- **unknown**: The state of the virtual interface is not available.

Type: String

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

virtualInterfaceType (p. 209)

The type of virtual interface. The possible values are private and public.

Type: String

vlan (p. 209)

The ID of the VLAN.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors (p. 270).

**DirectConnectClientException**

One or more parameters are not valid.

HTTP Status Code: 400

**DirectConnectServerException**

A server-side error occurred.
HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
Data Types

The AWS 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 a data type structure is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- AssociatedGateway (p. 215)
- BGPPeer (p. 216)
- Connection (p. 218)
- CustomerAgreement (p. 222)
- DirectConnectGateway (p. 223)
- DirectConnectGatewayAssociation (p. 225)
- DirectConnectGatewayAssociationProposal (p. 227)
- DirectConnectGatewayAttachment (p. 229)
- Interconnect (p. 231)
- Lag (p. 234)
- Loa (p. 238)
- Location (p. 239)
- MacSecKey (p. 241)
- NewBGPPeer (p. 243)
- NewPrivateVirtualInterface (p. 244)
- NewPrivateVirtualInterfaceAllocation (p. 246)
- NewPublicVirtualInterface (p. 248)
- NewPublicVirtualInterfaceAllocation (p. 250)
- NewTransitVirtualInterface (p. 252)
- NewTransitVirtualInterfaceAllocation (p. 254)
- ResourceTag (p. 256)
- RouteFilterPrefix (p. 257)
- RouterType (p. 258)
- Tag (p. 260)
- VirtualGateway (p. 261)
- VirtualInterface (p. 262)
- VirtualInterfaceTestHistory (p. 266)
AssociatedGateway

Information about the associated gateway.

Contents

id

The ID of the associated gateway.
Type: String
Required: No

ownerAccount

The ID of the AWS account that owns the associated virtual private gateway or transit gateway.
Type: String
Required: No

region

The Region where the associated gateway is located.
Type: String
Required: No

type

The type of associated gateway.
Type: String

Valid Values: virtualPrivateGateway | transitGateway
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
BGPPeer

Information about a BGP peer.

Contents

addressFamily

The address family for the BGP peer.

Type: String

Valid Values: ipv4 | ipv6

Required: No

amazonAddress

The IP address assigned to the Amazon interface.

Type: String

Required: No

asn

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

Type: Integer

Required: No

authKey

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

Type: String

Required: No

awsDeviceV2

The Direct Connect endpoint that terminates the BGP peer.

Type: String

Required: No

awsLogicalDeviceId

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

Required: No

bgpPeerId

The ID of the BGP peer.

Type: String
bpgPeerState

The state of the BGP peer. The following are the possible values:

- verifying: The BGP peering addresses or ASN require validation before the BGP peer can be created. This state applies only to public virtual interfaces.
- pending: The BGP peer is created, and remains in this state until it is ready to be established.
- available: The BGP peer is ready to be established.
- deleting: The BGP peer is being deleted.
- deleted: The BGP peer is deleted and cannot be established.

Type: String

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

bpgStatus

The status of the BGP peer. The following are the possible values:

- up: The BGP peer is established. This state does not indicate the state of the routing function. Ensure that you are receiving routes over the BGP session.
- down: The BGP peer is down.
- unknown: The BGP peer status is not available.

Type: String

Valid Values: up | down | unknown

customerAddress

The IP address assigned to the customer interface.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
Connection

Information about an AWS Direct Connect connection.

Contents

awsDevice

This member has been deprecated.

The Direct Connect endpoint on which the physical connection terminates.

Type: String

Required: No

awsDeviceV2

The Direct Connect endpoint that terminates the physical connection.

Type: String

Required: No

awsLogicalDeviceId

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

Required: No

bandwidth

The bandwidth of the connection.

Type: String

Required: No

connectionId

The ID of the connection.

Type: String

Required: No

connectionName

The name of the connection.

Type: String

Required: No

connectionState

The state of the connection. The following are the possible values:

- 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.
- **deleting**: The connection is being deleted.
- **deleted**: The connection has been deleted.
- **rejected**: A hosted connection in the ordering state enters the rejected state if it is deleted by the customer.
- **unknown**: The state of the connection is not available.

Type: String

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

Required: No

**encryptionMode**

The MAC Security (MACsec) connection encryption mode.

The valid values are no_encrypt, should_encrypt, and must_encrypt.

Type: String

Required: No

**hasLogicalRedundancy**

Indicates whether the connection supports a secondary BGP peer in the same address family (IPv4/IPv6).

Type: String

Valid Values: unknown | yes | no

Required: No

**jumboFrameCapable**

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean

Required: No

**lagId**

The ID of the LAG.

Type: String

Required: No

**loaIssueTime**

The time of the most recent call to DescribeLoa (p. 161) for this connection.

Type: Timestamp

Required: No
location

The location of the connection.

Type: String

Required: No

macSecCapable

Indicates whether the connection supports MAC Security (MACsec).

Type: Boolean

Required: No

macSecKeys

The MAC Security (MACsec) security keys associated with the connection.

Type: Array of MacSecKey (p. 241) objects

Required: No

ownerAccount

The ID of the AWS account that owns the connection.

Type: String

Required: No

partnerName

The name of the AWS Direct Connect service provider associated with the connection.

Type: String

Required: No

portEncryptionStatus

The MAC Security (MACsec) port link status of the connection.

The valid values are Encryption Up, which means that there is an active Connection Key Name, or Encryption Down.

Type: String

Required: No

providerName

The name of the service provider associated with the connection.

Type: String

Required: No

region

The AWS Region where the connection is located.

Type: String

Required: No
tags

The tags associated with the connection.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

Required: No

vlan

The ID of the VLAN.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
CustomerAgreement

The name and status of a customer agreement.

Contents

agreementName

The name of the agreement.

Type: String

Length Constraints: Maximum length of 100.

Required: No

status

The status of the customer agreement. This will be either signed or unsigned

Type: String

Length Constraints: Maximum length of 30.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
DirectConnectGateway

Information about a Direct Connect gateway, which enables you to connect virtual interfaces and virtual private gateway or transit gateways.

Contents

amazonSideAsn

The autonomous system number (ASN) for the Amazon side of the connection.

Type: Long

Required: No

directConnectGatewayId

The ID of the Direct Connect gateway.

Type: String

Required: No

directConnectGatewayName

The name of the Direct Connect gateway.

Type: String

Required: No

directConnectGatewayState

The state of the Direct Connect gateway. The following are the possible values:

• pending: The initial state after calling CreateDirectConnectGateway (p. 72).
• available: The Direct Connect gateway is ready for use.
• deleting: The initial state after calling DeleteDirectConnectGateway (p. 115).
• deleted: The Direct Connect gateway is deleted and cannot pass traffic.

Type: String

Valid Values: pending | available | deleting | deleted

Required: No

ownerAccount

The ID of the AWS account that owns the Direct Connect gateway.

Type: String

Required: No

stateChangeError

The error message if the state of an object failed to advance.

Type: String

Required: No
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
DirectConnectGatewayAssociation

Information about an association between a Direct Connect gateway and a virtual private gateway or transit gateway.

Contents

allowedPrefixesToDirectConnectGateway

The Amazon VPC prefixes to advertise to the Direct Connect gateway.

Type: Array of RouteFilterPrefix (p. 257) objects

Required: No

associatedGateway

Information about the associated gateway.

Type: AssociatedGateway (p. 215) object

Required: No

associationId

The ID of the Direct Connect gateway association.

Type: String

Required: No

associationState

The state of the association. The following are the possible values:

- associating: The initial state after calling CreateDirectConnectGatewayAssociation (p. 74).
- associated: The Direct Connect gateway and virtual private gateway or transit gateway are successfully associated and ready to pass traffic.
- disassociating: The initial state after calling DeleteDirectConnectGatewayAssociation (p. 117).
- disassociated: The virtual private gateway or transit gateway is disassociated from the Direct Connect gateway. Traffic flow between the Direct Connect gateway and virtual private gateway or transit gateway is stopped.

Type: String

Valid Values: associating | associated | disassociating | disassociated | updating

Required: No

directConnectGatewayId

The ID of the Direct Connect gateway.

Type: String

Required: No

directConnectGatewayOwnerAccount

The ID of the AWS account that owns the associated gateway.

Type: String
stateChangeError

The error message if the state of an object failed to advance.

Type: String

virtualGatewayId

The ID of the virtual private gateway. Applies only to private virtual interfaces.

Type: String

virtualGatewayOwnerAccount

The ID of the AWS account that owns the virtual private gateway.

Type: String

virtualGatewayRegion

This member has been deprecated.

The AWS Region where the virtual private gateway is located.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
DirectConnectGatewayAssociationProposal

Information about the proposal request to attach a virtual private gateway to a Direct Connect gateway.

Contents

associatedGateway

Information about the associated gateway.

Type: AssociatedGateway (p. 215) object

Required: No
directConnectGatewayId

The ID of the Direct Connect gateway.

Type: String

Required: No
directConnectGatewayOwnerId

The ID of the AWS account that owns the Direct Connect gateway.

Type: String

Required: No
existingAllowedPrefixesToDirectConnectGateway

The existing Amazon VPC prefixes advertised to the Direct Connect gateway.

Type: Array of RouteFilterPrefix (p. 257) objects

Required: No

proposalId

The ID of the association proposal.

Type: String

Required: No

proposalState

The state of the proposal. The following are possible values:

- accepted: The proposal has been accepted. The Direct Connect gateway association is available to use in this state.
- deleted: The proposal has been deleted by the owner that made the proposal. The Direct Connect gateway association cannot be used in this state.
- requested: The proposal has been requested. The Direct Connect gateway association cannot be used in this state.

Type: String

Valid Values: requested | accepted | deleted

Required: No
requestedAllowedPrefixesToDirectConnectGateway

The Amazon VPC prefixes to advertise to the Direct Connect gateway.

Type: Array of RouteFilterPrefix (p. 257) objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
DirectConnectGatewayAttachment

Information about an attachment between a Direct Connect gateway and a virtual interface.

Contents

attachmentState

The state of the attachment. The following are the possible values:
- attaching: The initial state after a virtual interface is created using the Direct Connect gateway.
- attached: The Direct Connect gateway and virtual interface are attached and ready to pass traffic.
- detaching: The initial state after calling DeleteVirtualInterface (p. 129).
- detached: The virtual interface is detached from the Direct Connect gateway. Traffic flow between the Direct Connect gateway and virtual interface is stopped.

Type: String

Valid Values: attaching | attached | detaching | detached

Required: No

attachmentType

The type of attachment.

Type: String

Valid Values: TransitVirtualInterface | PrivateVirtualInterface

Required: No

directConnectGatewayId

The ID of the Direct Connect gateway.

Type: String

Required: No

stateChangeError

The error message if the state of an object failed to advance.

Type: String

Required: No

virtualInterfaceId

The ID of the virtual interface.

Type: String

Required: No

virtualInterfaceOwnerAccount

The ID of the AWS account that owns the virtual interface.

Type: String
Required: No

**virtualInterfaceRegion**

The AWS Region where the virtual interface is located.

Type: String

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
Interconnect

Information about an interconnect.

Contents

**awsDevice**

*This member has been deprecated.*

The Direct Connect endpoint on which the physical connection terminates.

Type: String

Required: No

**awsDeviceV2**

The Direct Connect endpoint that terminates the physical connection.

Type: String

Required: No

**awsLogicalDeviceId**

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

Required: No

**bandwidth**

The bandwidth of the connection.

Type: String

Required: No

**hasLogicalRedundancy**

Indicates whether the interconnect supports a secondary BGP in the same address family (IPv4/IPv6).

Type: String

Valid Values: unknown | yes | no

Required: No

**interconnectId**

The ID of the interconnect.

Type: String

Required: No

**interconnectName**

The name of the interconnect.
interconnectState

The state of the interconnect. The following are the possible values:

- **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 is approved, and is being initialized.
- **available**: The network link is up, and the interconnect is ready for use.
- **down**: The network link is down.
- **deleting**: The interconnect is being deleted.
- **deleted**: The interconnect is deleted.
- **unknown**: The state of the interconnect is not available.

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

**Required:** No

jumboFrameCapable

 Indicates whether jumbo frames (9001 MTU) are supported.

**Type:** Boolean

**Required:** No

lagId

The ID of the LAG.

**Type:** String

**Required:** No

loaIssueTime

The time of the most recent call to DescribeLoa (p. 161) for this connection.

**Type:** Timestamp

**Required:** No

location

The location of the connection.

**Type:** String

**Required:** No

providerName

The name of the service provider associated with the interconnect.

**Type:** String

**Required:** No
region

The AWS Region where the connection is located.

Type: String

Required: No

tags

The tags associated with the interconnect.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
Lag

Information about a link aggregation group (LAG).

Contents

allowsHostedConnections

Indicates whether the LAG can host other connections.

Type: Boolean

Required: No

awsDevice

*This member has been deprecated.*

The AWS Direct Connect endpoint that hosts the LAG.

Type: String

Required: No

awsDeviceV2

The AWS Direct Connect endpoint that hosts the LAG.

Type: String

Required: No

awsLogicalDeviceId

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.

Type: String

Required: No

connections

The connections bundled by the LAG.

Type: Array of Connection (p. 218) objects

Required: No

connectionsBandwidth

The individual bandwidth of the physical connections bundled by the LAG. The possible values are 1Gbps and 10Gbps.

Type: String

Required: No

encryptionMode

The LAG MAC Security (MACsec) encryption mode.

The valid values are no_encrypt, should_encrypt, and must_encrypt.
Type: String
Required: No

**hasLogicalRedundancy**

Indicates whether the LAG supports a secondary BGP peer in the same address family (IPv4/IPv6).

Type: String
Valid Values: unknown | yes | no
Required: No

**jumboFrameCapable**

Indicates whether jumbo frames (9001 MTU) are supported.

Type: Boolean
Required: No

**lagId**

The ID of the LAG.

Type: String
Required: No

**lagName**

The name of the LAG.

Type: String
Required: No

**lagState**

The state of the LAG. The following are the possible values:
- requested: The initial state of a LAG. The LAG stays in the requested state until the Letter of Authorization (LOA) is available.
- pending: The LAG has been approved and is being initialized.
- available: The network link is established and the LAG is ready for use.
- down: The network link is down.
- deleting: The LAG is being deleted.
- deleted: The LAG is deleted.
- unknown: The state of the LAG is not available.

Type: String
Valid Values: requested | pending | available | down | deleting | deleted | unknown
Required: No

**location**

The location of the LAG.

Type: String
Required: No
macSecCapable
Indicates whether the LAG supports MAC Security (MACsec).
Type: Boolean
Required: No

macSecKeys
The MAC Security (MACsec) security keys associated with the LAG.
Type: Array of MacSecKey (p. 241) objects
Required: No

minimumLinks
The minimum number of physical dedicated connections that must be operational for the LAG itself to be operational.
Type: Integer
Required: No

numberOfConnections
The number of physical dedicated connections bundled by the LAG, up to a maximum of 10.
Type: Integer
Required: No

ownerAccount
The ID of the AWS account that owns the LAG.
Type: String
Required: No

providerName
The name of the service provider associated with the LAG.
Type: String
Required: No

region
The AWS Region where the connection is located.
Type: String
Required: No

tags
The tags associated with the LAG.
Type: Array of Tag (p. 260) objects
Array Members: Minimum number of 1 item.
Required: No
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
Loa

Information about a Letter of Authorization - Connecting Facility Assignment (LOA-CFA) for a connection.

Contents

loaContent

The binary contents of the LOA-CFA document.
Type: Base64-encoded binary data object
Required: No

loaContentType

The standard media type for the LOA-CFA document. The only supported value is application/pdf.
Type: String
Valid Values: application/pdf
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
Location

Information about an AWS Direct Connect location.

Contents

availableMacSecPortSpeeds

The available MAC Security (MACsec) port speeds for the location.

Type: Array of strings

Required: No

availablePortSpeeds

The available port speeds for the location.

Type: Array of strings

Required: No

availableProviders

The name of the service provider for the location.

Type: Array of strings

Required: No

locationCode

The code for the location.

Type: String

Required: No

locationName

The name of the location. This includes the name of the colocation partner and the physical site of the building.

Type: String

Required: No

region

The AWS Region for the location.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
• AWS SDK for Java V2
• AWS SDK for Ruby V3
MacSecKey

Information about the MAC Security (MACsec) secret key.

Contents

ckn

The Connection Key Name (CKN) for the MAC Security secret key.

Type: String
Required: No

secretARN

The Amazon Resource Name (ARN) of the MAC Security (MACsec) secret key.

Type: String
Required: No

startOn

The date that the MAC Security (MACsec) secret key takes effect. The value is displayed in UTC format.

Type: String
Required: No

state

The state of the MAC Security (MACsec) secret key.

The possible values are:

- associating: The MAC Security (MACsec) secret key is being validated and not yet associated with the connection or LAG.
- associated: The MAC Security (MACsec) secret key is validated and associated with the connection or LAG.
- disassociating: The MAC Security (MACsec) secret key is being disassociated from the connection or LAG.
- disassociated: The MAC Security (MACsec) secret key is no longer associated with the connection or LAG.

Type: String
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
NewBGPPeer

Information about a new BGP peer.

Contents

addressFamily

The address family for the BGP peer.

Type: String

Valid Values: ipv4 | ipv6

Required: No

amazonAddress

The IP address assigned to the Amazon interface.

Type: String

Required: No

asn

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

Type: Integer

Required: No

authKey

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

Type: String

Required: No

customerAddress

The IP address assigned to the customer interface.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
NewPrivateVirtualInterface

Information about a private virtual interface.

Contents

addressFamily

The address family for the BGP peer.

Type: String

Valid Values: ipv4 | ipv6

Required: No

amazonAddress

The IP address assigned to the Amazon interface.

Type: String

Required: No

asn

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

The valid values are 1-2147483647.

Type: Integer

Required: Yes

authKey

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

Type: String

Required: No

customerAddress

The IP address assigned to the customer interface.

Type: String

Required: No

directConnectGatewayId

The ID of the Direct Connect gateway.

Type: String

Required: No

enableSiteLink

Indicates whether to enable or disable SiteLink.

Type: Boolean
mtu

The maximum transmission unit (MTU), in bytes. The supported values are 1500 and 9001. The default value is 1500.

Type: Integer

Required: No

tags

The tags associated with the private virtual interface.

Type: Array of Tag objects

Array Members: Minimum number of 1 item.

Required: No

virtualGatewayId

The ID of the virtual private gateway.

Type: String

Required: No

virtualInterfaceName

The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).

Type: String

Required: Yes

vlan

The ID of the VLAN.

Type: Integer

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
NewPrivateVirtualInterfaceAllocation

Information about a private virtual interface to be provisioned on a connection.

Contents

addressFamily

The address family for the BGP peer.

Type: String

Valid Values: ipv4 | ipv6

Required: No

amazonAddress

The IP address assigned to the Amazon interface.

Type: String

Required: No

asn

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

The valid values are 1-2147483647.

Type: Integer

Required: Yes

authKey

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

Type: String

Required: No

customerAddress

The IP address assigned to the customer interface.

Type: String

Required: No

mtu

The maximum transmission unit (MTU), in bytes. The supported values are 1500 and 9001. The default value is 1500.

Type: Integer

Required: No

tags

The tags associated with the private virtual interface.
Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

Required: No

**virtualInterfaceName**

The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).

Type: String

Required: Yes

**vlan**

The ID of the VLAN.

Type: Integer

Required: Yes

---

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
NewPublicVirtualInterface

Information about a public virtual interface.

Contents

addressFamily

The address family for the BGP peer.

Type: String

Valid Values: ipv4 | ipv6

Required: No

amazonAddress

The IP address assigned to the Amazon interface.

Type: String

Required: No

asn

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

The valid values are 1-2147483647.

Type: Integer

Required: Yes

authKey

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

Type: String

Required: No

customerAddress

The IP address assigned to the customer interface.

Type: String

Required: No

routeFilterPrefixes

The routes to be advertised to the AWS network in this Region. Applies to public virtual interfaces.

Type: Array of RouteFilterPrefix (p. 257) objects

Required: No

tags

The tags associated with the public virtual interface.

Type: Array of Tag (p. 260) objects
Array Members: Minimum number of 1 item.

Required: No

virtualInterfaceName

The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).

Type: String
Required: Yes

vlan

The ID of the VLAN.

Type: Integer
Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
NewPublicVirtualInterfaceAllocation

Information about a public virtual interface to be provisioned on a connection.

Contents

addressFamily

The address family for the BGP peer.

Type: String

Valid Values: ipv4 | ipv6

Required: No

amazonAddress

The IP address assigned to the Amazon interface.

Type: String

Required: No

asn

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

The valid values are 1-2147483647.

Type: Integer

Required: Yes

authKey

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

Type: String

Required: No

customerAddress

The IP address assigned to the customer interface.

Type: String

Required: No

routeFilterPrefixes

The routes to be advertised to the AWS network in this Region. Applies to public virtual interfaces.

Type: Array of RouteFilterPrefix (p. 257) objects

Required: No

tags

The tags associated with the public virtual interface.

Type: Array of Tag (p. 260) objects
Array Members: Minimum number of 1 item.

Required: No

virtualInterfaceName

The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).

Type: String

Required: Yes

vlan

The ID of the VLAN.

Type: Integer

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
NewTransitVirtualInterface

Information about a transit virtual interface.

Contents

addressFamily

The address family for the BGP peer.

Type: String

Valid Values: ipv4 | ipv6

Required: No

amazonAddress

The IP address assigned to the Amazon interface.

Type: String

Required: No

asn

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

The valid values are 1-2147483647.

Type: Integer

Required: No

authKey

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

Type: String

Required: No

customerAddress

The IP address assigned to the customer interface.

Type: String

Required: No

directConnectGatewayId

The ID of the Direct Connect gateway.

Type: String

Required: No

enableSiteLink

Indicates whether to enable or disable SiteLink.

Type: Boolean
Required: No

mtu
The maximum transmission unit (MTU), in bytes. The supported values are 1500 and 9001. The default value is 1500.
Type: Integer
Required: No

tags
The tags associated with the transitive virtual interface.
Type: Array of Tag (p. 260) objects
Array Members: Minimum number of 1 item.
Required: No

virtualInterfaceName
The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).
Type: String
Required: No

vlan
The ID of the VLAN.
Type: Integer
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
NewTransitVirtualInterfaceAllocation

Information about a transit virtual interface to be provisioned on a connection.

Contents

addressFamily

The address family for the BGP peer.
Type: String
Valid Values: ipv4 | ipv6
Required: No

amazonAddress

The IP address assigned to the Amazon interface.
Type: String
Required: No

asn

The autonomous system (AS) number for Border Gateway Protocol (BGP) configuration.
The valid values are 1-2147483647.
Type: Integer
Required: No

authKey

The authentication key for BGP configuration. This string has a minimum length of 6 characters and
and a maximum length of 80 characters.
Type: String
Required: No

customerAddress

The IP address assigned to the customer interface.
Type: String
Required: No

mtu

The maximum transmission unit (MTU), in bytes. The supported values are 1500 and 9001. The
default value is 1500.
Type: Integer
Required: No

tags

The tags associated with the transitive virtual interface.
Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

Required: No

virtualInterfaceName

The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).

Type: String

Required: No

vlan

The ID of the VLAN.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
ResourceTag

Information about a tag associated with an AWS Direct Connect resource.

Contents

resourceArn

The Amazon Resource Name (ARN) of the resource.

Type: String

Required: No

tags

The tags.

Type: Array of Tag (p. 260) objects

Array Members: Minimum number of 1 item.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
RouteFilterPrefix

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

Contents

cidr

The CIDR block for the advertised route. Separate multiple routes using commas. An IPv6 CIDR must use /64 or shorter.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
RouterType

Information about the virtual router.

Contents

platform

The virtual interface router platform.

Type: String

Required: No

routerTypeIdentifier

Identifies the router by a combination of vendor, platform, and software version. For example, CiscoSystemsInc-2900SeriesRouters-IOS124.

Type: String

Required: No

software

The router software.

Type: String

Required: No

vendor

The vendor for the virtual interface's router.

Type: String

Required: No

xsltTemplateName

The template for the virtual interface's router.

Type: String

Required: No

xsltTemplateNameForMacSec

The MAC Security (MACsec) template for the virtual interface's router.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
• AWS SDK for Java V2
• AWS SDK for Ruby V3
Tag

Information about a tag.

Contents

**key**

The key.

Type: String


Pattern: ^([\p{L}\p{Z}\p{N}_\-:.=\+/@]*$)

Required: Yes

**value**

The value.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 256.

Pattern: ^([\p{L}\p{Z}\p{N}_\-:.=\+/@]*$)

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
VirtualGateway

Information about a virtual private gateway for a private virtual interface.

Contents

virtualGatewayId

The ID of the virtual private gateway.

Type: String

Required: No

virtualGatewayState

The state of the virtual private gateway. The following are the possible values:

- pending: Initial state after creating the virtual private gateway.
- available: Ready for use by a private virtual interface.
- deleting: Initial state after deleting the virtual private gateway.
- deleted: The virtual private gateway is deleted. The private virtual interface is unable to send traffic over this gateway.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
VirtualInterface

Information about a virtual interface.

Contents

addressFamily

The address family for the BGP peer.

Type: String

Valid Values: ipv4 | ipv6

Required: No

amazonAddress

The IP address assigned to the Amazon interface.

Type: String

Required: No

amazonSideAsn

The autonomous system number (ASN) for the Amazon side of the connection.

Type: Long

Required: No

asn

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

The valid values are 1-2147483647.

Type: Integer

Required: No

authKey

The authentication key for BGP configuration. This string has a minimum length of 6 characters and a maximum length of 80 characters.

Type: String

Required: No

awsDeviceV2

The Direct Connect endpoint that terminates the physical connection.

Type: String

Required: No

awsLogicalDeviceId

The Direct Connect endpoint that terminates the logical connection. This device might be different than the device that terminates the physical connection.
Type: String  
Required: No  
bgpPeers  
The BGP peers configured on this virtual interface.  
Type: Array of BGPPeer (p. 216) objects  
Required: No  
connectionId  
The ID of the connection.  
Type: String  
Required: No  
customerAddress  
The IP address assigned to the customer interface.  
Type: String  
Required: No  
customerRouterConfig  
The customer router configuration.  
Type: String  
Required: No  
directConnectGatewayId  
The ID of the Direct Connect gateway.  
Type: String  
Required: No  
jumboFrameCapable  
Indicates whether jumbo frames (9001 MTU) are supported.  
Type: Boolean  
Required: No  
location  
The location of the connection.  
Type: String  
Required: No  
mtu  
The maximum transmission unit (MTU), in bytes. The supported values are 1500 and 9001. The default value is 1500.  
Type: Integer  
Required: No
ownerAccount

The ID of the AWS account that owns the virtual interface.

Type: String
Required: No

region

The AWS Region where the virtual interface is located.

Type: String
Required: No

routeFilterPrefixes

The routes to be advertised to the AWS network in this Region. Applies to public virtual interfaces.

Type: Array of RouteFilterPrefix (p. 257) objects
Required: No

siteLinkEnabled

Indicates whether SiteLink is enabled.

Type: Boolean
Required: No

tags

The tags associated with the virtual interface.

Type: Array of Tag (p. 260) objects
Array Members: Minimum number of 1 item.
Required: No

virtualGatewayId

The ID of the virtual private gateway. Applies only to private virtual interfaces.

Type: String
Required: No

virtualInterfaceId

The ID of the virtual interface.

Type: String
Required: No

virtualInterfaceName

The name of the virtual interface assigned by the customer network. The name has a maximum of 100 characters. The following are valid characters: a-z, 0-9 and a hyphen (-).

Type: String
Required: No
virtualInterfaceState

The state of the virtual interface. The following are the possible values:

- **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.
- **down**: A virtual interface that is BGP down.
- **deleting**: A virtual interface is in this state immediately after calling DeleteVirtualInterface (p. 129) 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 enters the **Rejected** state.
- **unknown**: The state of the virtual interface is not available.

Type: String
Valid Values: confirming | verifying | pending | available | down | deleting | deleted | rejected | unknown

Required: No

virtualInterfaceType

The type of virtual interface. The possible values are **private** and **public**.

Type: String

Required: No

vlan

The ID of the VLAN.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
VirtualInterfaceTestHistory

Information about the virtual interface failover test.

Contents

bgpPeers
The BGP peers that were put in the DOWN state as part of the virtual interface failover test.
Type: Array of strings
Required: No

endTime
The time that the virtual interface moves out of the DOWN state.
Type: Timestamp
Required: No

ownerAccount
The owner ID of the tested virtual interface.
Type: String
Required: No

startTime
The time that the virtual interface moves to the DOWN state.
Type: Timestamp
Required: No

status
The status of the virtual interface failover test.
Type: String
Required: No

testDurationInMinutes
The time that the virtual interface failover test ran in minutes.
Type: Integer
Required: No

testId
The ID of the virtual interface failover test.
Type: String
Required: No

virtualInterfaceId
The ID of the tested virtual interface.
Type: String
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signature Version 4 Signing Process in the Amazon Web Services General Reference.

Action

The action to be performed.

Type: string
Required: Yes

Version

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

Type: string
Required: Yes

X-Amz-Algorithm

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

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string
Valid Values: AWS4-HMAC-SHA256
Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: access_key/YYYYMMDD/region/service/aws4_request.

For more information, see Task 2: Create a String to Sign for Signature Version 4 in the Amazon Web Services General Reference.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string
Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format ("YYYYMMDD'T'HHMMSS'Z'"). For example, the following date time is a valid X-Amz-Date value: 20120325T120000Z.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is
not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Handling Dates in Signature Version 4 in the Amazon Web Services General Reference.

Type: string
Required: Conditional

**X-Amz-Security-Token**

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to AWS Services That Work with IAM in the IAM User Guide.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string
Required: Conditional

**X-Amz-Signature**

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string
Required: Conditional

**X-Amz-SignedHeaders**

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see Task 1: Create a Canonical Request For Signature Version 4 in the Amazon Web Services General Reference.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string
Required: Conditional
Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

**AccessDeniedException**

You do not have sufficient access to perform this action.

HTTP Status Code: 400

**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**

The AWS query string is malformed or 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**

The 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

**NotAuthorized**

You do not have permission to perform this action.

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

**ThrottlingException**

The request was denied due to request throttling.

HTTP Status Code: 400

**ValidationError**

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400