Amazon Virtual Private Cloud

**VPC Peering**

What is VPC Peering?

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network

that you've defined.

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic

between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate

with each other as if they are within the same network. You can create a VPC peering connection

between your own VPCs, or with a VPC in another AWS account. The VPCs can be in different regions

(also known as an inter-region VPC peering connection).

AWS uses the existing infrastructure of a VPC to create a VPC peering connection; it is neither a gateway

nor a VPN connection, and does not rely on a separate piece of physical hardware. There is no single

point of failure for communication or a bandwidth bottleneck.

A VPC peering connection helps you to facilitate the transfer of data. For example, if you have more than

one AWS account, you can peer the VPCs across those accounts to create a file sharing network. You can

also use a VPC peering connection to allow other VPCs to access resources you have in one of your VPCs.

You can establish peering relationships between VPCs across different AWS Regions (also called Inter-

Region VPC Peering). This allows VPC resources including EC2 instances, Amazon RDS databases and

Lambda functions that run in different AWS Regions to communicate with each other using private IP

addresses, without requiring gateways, VPN connections, or separate network appliances. The traffic

remains in the private IP space. All inter-region traffic is encrypted with no single point of failure,

or bandwidth bottleneck. Traffic always stays on the global AWS backbone, and never traverses the

public internet, which reduces threats, such as common exploits, and DDoS attacks. Inter-Region VPC

Peering provides a simple and cost-effective way to share resources between regions or replicate data for

geographic redundancy



VPC Peering Basics

To establish a VPC peering connection, you do the following:

1. The owner of the *requester VPC* sends a request to the owner of the *accepter VPC* to create the VPC

peering connection. The accepter VPC can be owned by you, or another AWS account, and cannot have

a CIDR block that overlaps with the requester VPC's CIDR block.

2. The owner of the accepter VPC accepts the VPC peering connection request to activate the VPC

peering connection.

3. To enable the flow of traffic between the VPCs using private IP addresses, the owner of each VPC in

the VPC peering connection must manually add a route to one or more of their VPC route tables that

points to the IP address range of the other VPC (the peer VPC).

4. If required, update the security group rules that are associated with your instance to ensure that traffic

to and from the peer VPC is not restricted. If both VPCs are in the same region, you can reference a

security group from the peer VPC as a source or destination for ingress or egress rules in your security

group rules.

5. By default, if instances on either side of a VPC peering connection address each other using a public

DNS hostname, the hostname resolves to the instance's public IP address. To change this behavior,

enable DNS hostname resolution for your VPC connection. After enabling DNS hostname resolution,

if instances on either side of the VPC peering connection address each other using a public DNS

hostname, the hostname resolves to the private IP address of the instance.

Multiple VPC Peering Connections

A VPC peering connection is a one to one relationship between two VPCs. You can create multiple VPC

peering connections for each VPC that you own, but transitive peering relationships are not supported.

You do not have any peering relationship with VPCs that your VPC is not directly peered with.

The following diagram is an example of one VPC peered to two different VPCs. There are two VPC

peering connections: VPC A is peered with both VPC B and VPC C. VPC B and VPC C are not peered, and

you cannot use VPC A as a transit point for peering between VPC B and VPC C. If you want to enable

routing of traffic between VPC B and VPC C, you must create a unique VPC peering connection between

them.

Pricing for a VPC Peering Connection

If the VPCs in the VPC peering connection are within the same region, the charges for transferring data

within the VPC peering connection are the same as the charges for transferring data across Availability

Zones. If the VPCs are in different regions, inter-region data transfer costs apply.



Viewing Your VPC Peering Connections

You can view all of your VPC peering connections in the Amazon VPC console. By default, the console

displays all VPC peering connections in different states, including those that may have been recently

deleted or rejected. For more information about the lifecycle of a VPC peering connection, see VPC

Peering Connection Lifecycle (p. 2).

**To view your VPC peering connections**

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

2. In the navigation pane, choose **Peering Connections**.

3. All of your VPC peering connections are listed. Use the filter search bar to narrow your results.

**To describe a VPC peering connection using the command line or an API**

• describe-vpc-peering-connections (AWS CLI)

• Get-EC2VpcPeeringConnections (AWS Tools for Windows PowerShell)

Rejecting a VPC Peering Connection

You can reject any VPC peering connection request that you've received that's in the pendingacceptance

state. You should only accept VPC peering connections from AWS accounts that you know

and trust; you can reject any unwanted requests.

**To reject a VPC peering connection**

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

2. In the navigation pane, choose **Peering Connections**.

3. Select the VPC peering connection, and choose **Actions**, **Reject Request**.

4. In the confirmation dialog box, choose **Yes, Reject**.

**To reject a VPC peering connection using the command line or an API**

• reject-vpc-peering-connection (AWS CLI)

• Deny-EC2VpcPeeringConnection (AWS Tools for Windows PowerShell)

• RejectVpcPeeringConnection (Amazon EC2 Query API)

Updating Your Route Tables for a VPC Peering

Connection

To send private IPv4 traffic from your instance to an instance in a peer VPC, you must add a route to the

route table that's associated with your subnet in which your instance resides. The route points to the

CIDR block (or portion of the CIDR block) of the peer VPC in the VPC peering connection, and specifies

the VPC peering connection as the target.

Similarly, if the VPCs in the VPC peering connection have associated IPv6 CIDR blocks, you can add a

route to your route table to enable communication with the peer VPC over IPv6.

If a subnet is not explicitly associated with a route table, it uses the main route table by default. For more

information, see Route Tables in the *Amazon VPC User Guide*.

You have a quota on the number of entries you can add per route table. If the number of VPC peering

connections in your VPC exceeds the route table entry quota for a single route table, consider using

multiple subnets that are each associated with a custom route table.

For more information about supported route table configurations for VPC peering connections, see VPC

Peering Configurations (p. 19).

You can add a route for a VPC peering connection that's in the pending-acceptance state. However,

the route will have a state of blackhole and have no effect until the VPC peering connection is in the

active state.

**Warning**

If you have a VPC peered with multiple VPCs that have overlapping or matching IPv4 CIDR

blocks, ensure that your route tables are configured to avoid sending response traffic from your

VPC to the incorrect VPC. AWS currently does not support unicast reverse path forwarding in

VPC peering connections that checks the source IP of packets and routes reply packets back to

the source. For more information, see Routing for Response Traffic (p. 44).

**To add an IPv4 route for a VPC peering connection**

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

2. In the navigation pane, choose **Route Tables**.

3. Select the route table that's associated with the subnet in which your instance resides.

**Note**

If you do not have a route table associated with that subnet, select the main route table for

the VPC, as the subnet then uses this route table by default.

4. Choose **Routes**, **Edit**, **Add Route**.

5. For **Destination**, enter the IPv4 address range to which the network traffic in the VPC peering

connection must be directed. You can specify the entire IPv4 CIDR block of the peer VPC, a

specific range, or an individual IPv4 address, such as the IP address of the instance with which to

communicate. For example, if the CIDR block of the peer VPC is 10.0.0.0/16, you can specify a

portion 10.0.0.0/28, or a specific IP address 10.0.0.7/32.

6. Select the VPC peering connection from **Target**, and then choose **Save**.

The owner of the peer VPC must also complete these steps to add a route to direct traffic back to your

VPC through the VPC peering connection.

If both VPCs in the VPC peering connection are in the same region, have IPv6 CIDR blocks, and the

resources in the VPC are enabled to use IPv6, you can also add a route for IPv6 communication



Updating Your Security Groups to Reference Peer

VPC Groups

You can update the inbound or outbound rules for your VPC security groups to reference security groups

in the peered VPC. Doing so allows traffic to flow to and from instances that are associated with the

referenced security group in the peered VPC.

**Requirements**

• The peer VPC can be a VPC in your account, or a VPC in another AWS account. To reference a security

group in another AWS account, include the account number in **Source** or **Destination** field; for

example, 123456789012/sg-1a2b3c4d.

• You cannot reference the security group of a peer VPC that's in a different region. Instead, use the CIDR

block of the peer VPC.

• To reference a security group in a peer VPC, the VPC peering connection must be in the active state.

**To update your security group rules using the console**

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

2. In the navigation pane, choose **Security Groups**.

3. Select the security group, and choose **Inbound Rules** to modify the inbound rules or **Outbound**

**Rules** to modify the outbound rules.

4. Choose **Edit**, **Add another rule**.

5. Specify the type, protocol, and port range as required. For **Source** (or **Destination** for an outbound

rule), type the ID of the security group in the peer VPC if it is in the same region or the CIDR block of

the peer VPC if it is in a different region.

**Note**

Security groups in a peer VPC are not automatically displayed.

6. Choose **Save**.

**To update inbound rules using the command line**

• authorize-security-group-ingress (AWS CLI)

• Grant-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

• Revoke-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

• revoke-security-group-ingress (AWS CLI)

**To update outbound rules using the command line**

• authorize-security-group-egress (AWS CLI)

• Grant-EC2SecurityGroupEgress (AWS Tools for Windows PowerShell)

• Revoke-EC2SecurityGroupEgress (AWS Tools for Windows PowerShell)

• revoke-security-group-egress (AWS CLI)

For example, to update your security group sg-aaaa1111 to allow inbound access over HTTP from sgbbbb2222

that's in a peer VPC, you can use the following AWS CLI command:

**aws ec2 authorize-security-group-ingress --group-id *sg-aaaa1111* --protocol tcp --port *80* --**

**source-group *sg-bbbb2222***

After you've updated the security group rules, use the describe-security-groups command to view the

referenced security group in your security group rules.

Identifying Your Referenced Security Groups

To determine if your security group is being referenced in the rules of a security group in a peer VPC, use

one of the following commands for one or more security groups in your account.

• describe-security-group-references (AWS CLI)

• Get-EC2SecurityGroupReference (AWS Tools for Windows PowerShell)

• DescribeSecurityGroupReferences (Amazon EC2 Query API)

In the following example, the response indicates that security group sg-bbbb2222 is being referenced

by a security group in VPC vpc-aaaaaaaa:

**aws ec2 describe-security-group-references --group-id *sg-bbbb2222***

***CODE:-***

{

"SecurityGroupsReferenceSet": [

{

"ReferencingVpcId": "vpc-aaaaaaaa",

"GroupId": "sg-bbbb2222",

"VpcPeeringConnectionId": "pcx-b04deed9"

}

]

}

If the VPC peering connection is deleted, or if the owner of the peer VPC deletes the referenced security

group, the security group rule becomes stale

**To describe your stale security group rules using the command line or an API**

• describe-stale-security-groups (AWS CLI)

• Get-EC2StaleSecurityGroup (AWS Tools for Windows PowerShell)

• DescribeStaleSecurityGroups (Amazon EC2 Query API)

In the following example, VPC A (vpc-aaaaaaaa) and VPC B were peered, and the VPC peering

connection was deleted. Your security group sg-aaaa1111 in VPC A references sg-bbbb2222 in VPC

B. When you run the describe-stale-security-groups command for your VPC, the response

indicates that security group sg-aaaa1111 has a stale SSH rule that references sg-bbbb2222.

**aws ec2 describe-stale-security-groups --vpc-id *vpc-aaaaaaaa***

***CODE:-***

{

"StaleSecurityGroupSet": [

{

"VpcId": "vpc-aaaaaaaa",

"StaleIpPermissionsEgress": [],

"GroupName": "Access1",

"StaleIpPermissions": [

{

"ToPort": 22,

"FromPort": 22,

"UserIdGroupPairs": [

{

"VpcId": "vpc-bbbbbbbb",

"PeeringStatus": "deleted",

"UserId": "123456789101",

"GroupName": "Prod1",

"VpcPeeringConnectionId": "pcx-b04deed9",

"GroupId": "sg-bbbb2222"

}

],

"IpProtocol": "tcp"

}

],

"GroupId": "sg-aaaa1111",

"Description": "Reference remote SG"

}

]

}.

