Amazon Virtual Private Cloud

VPC Peering Guide

API Version 2014-06-15

Amazon Web Services

Amazon Virtual Private Cloud: VPC Peering Guide

Amazon Web Services

What is VPC Peering?

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

This guide is for customers who wish to set up VPC peering connections between VPCs in their own account, or with a VPC in another AWS account.

A VPC peering connection allows you to route traffic between the peer VPCs using private IP addresses; as if they are part of the same network. For more information about creating and working with VPC peering connections in the VPC console, see VPC Peering in the *Amazon Virtual Private Cloud User Guide*.

This guide provides information about the different ways you can configure your VPCs and VPC peering connections to suit your networking requirements. It also provides the routing configurations for your route tables to enable network access between peer VPCs.

For more information, see the following topics:

- VPC Peering Overview (p. 2)
- VPC Peering Scenarios (p. 4)
- Configurations With Routes to an Entire CIDR Block (p. 6)
- Configurations With Routes to Specific Subnets or IP Addresses (p. 13)
- Invalid VPC Peering Connection Configurations (p. 21)

VPC Peering Overview

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IP 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 within a single region.

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.

Topics

VPC Peering Basics

To establish a VPC peering connection, the owner of the *requester VPC* (or *local VPC*) sends a request to the owner of the *peer VPC* to create the VPC peering connection. The peer VPC can be owned by you, or another AWS account. The owner of the peer VPC has to accept the VPC peering connection request to activate the VPC peering connection. To enable the flow of traffic between the peer VPCs using private IP addresses, add a route to one or more of your VPC's route tables that points to the IP address range of the peer VPC. The owner of the peer VPC adds a route to one of their VPC's route tables that points to the IP address range of your VPC.

For more information about creating and working with VPC peering connections in the VPC console, see VPC Peering in the *Amazon Virtual Private Cloud User Guide*.

You are charged for data transfer within a VPC peering connection at the same rate as you are charged for data transfer across Availability Zones. For more information, see Amazon EC2 Pricing.

VPC Peering Limitations

To create a VPC peering connection with another VPC, you need to be aware of the following limitations and rules:

- You cannot create a VPC peering connection between VPCs that have matching or overlapping CIDR blocks.
- You cannot create a VPC peering connection between VPCs in different regions.
- You have a limit on the number active and pending VPC peering connections that you can have per VPC. For more information about VPC limits, see Amazon VPC Limits in the *Amazon Virtual Private Cloud User Guide*.
- VPC peering does not support transitive peering relationships; in a VPC peering connection, your VPC will not have access to any other VPCs that the peer VPC may be peered with. This includes VPC peering connections that are established entirely within your own AWS account. For more information about unsupported peering relationships, see Invalid VPC Peering Connection Configurations (p. 21). For examples of supported peering relationships, see VPC Peering Scenarios (p. 4).
- You cannot have more than one VPC peering connection between the same two VPCs at the same time.
- The Maximum Transmission Unit (MTU) across a VPC peering connection is 1500 bytes.
- A placement group can span peered VPCs; however, you will not get full-bisection bandwidth between instances in peered VPCs. For more information about placement groups, see Placement Groups in the *Amazon Elastic Compute Cloud User Guide*.
- Unicast reverse path forwarding in VPC peering connections is not supported. For more information, see Routing for Response Traffic (p. 15).
- You cannot reference a security group from the peer VPC as a source or destination for ingress or egress rules in your security group. Instead, reference CIDR blocks of the peer VPC as the source or destination of your security group's ingress or egress rules.

VPC Peering Scenarios

There are a number of reasons you may need to set up VPC peering connection between your VPCs, or between a VPC that you own and a VPC in a different AWS account. The following scenarios can help you determine which configuration is best suited to your networking requirements.

Topics

Peering Two or More VPCs to Provide Full Access to Resources

In this scenario, you have two or more VPCs that you want to peer to enable full sharing of resources between all VPCs. The following are some examples:

- Your company has a VPC for the finance department, and another VPC for the accounting department. The finance department requires access to all resources that are in the accounting department, and the accounting department requires access to all resources in the finance department.
- Your company has multiple IT departments, each with their own VPC. Some VPCs are located within the same AWS account, and others in a different AWS account. You want to peer together all VPCs to enable the IT departments to have full access to each others' resources.

For more information about how to set up the VPC peering connection configuration and route tables for this scenario, see the following topics:

- Two VPCs Peered Together (p. 6)
- Three VPCs Peered Together (p. 8)
- Multiple VPCs Peered Together (p. 10)

For more information about creating and working with VPC peering connections in the VPC console, see VPC Peering in the *Amazon Virtual Private Cloud User Guide*.

Peering to One VPC to Access Centralized Resources

In this scenario, you have a central VPC that contains resources that you want to share with other VPCs. Your central VPC may require full or partial access to the peer VPCs, and similarly, the peer VPCs may require full or partial access to the central VPC. The following are some examples:

- Your company's IT department has a VPC for file sharing. You want to peer other VPCs to that central VPC, however, you do not want the other VPCs to send traffic to each other.
- Your company has a VPC that you want to share with your customers. Each customer can create a VPC peering connection with your VPC, however, your customers cannot route traffic to other VPCs that are peered to yours, nor are they aware of the other customers' routes.
- You have a central VPC that is used for Active Directory services. Specific instances in peer VPCs send requests to the Active Directory servers and require full access to the central VPC. The central VPC does not require full access to the peer VPCs; it only needs to route response traffic to the specific instances.

For more information about how to set up the VPC peering connection configuration and route tables for this scenario, see the following topics:

- One VPC Peered With Two VPCs (p. 7)
- One VPC Peered With Multiple VPCs (p. 8)
- Two VPCs Peered to Two Subnets in One VPC (p. 13)
- One VPC Peered to Specific Subnets in Two VPCs (p. 14)
- Instances in One VPC Peered to Instances in Two VPCs (p. 16)
- One VPC Peered With Two VPCs Using Longest Prefix Match (p. 17)

For more information about creating and working with VPC peering connections in the VPC console, see VPC Peering in the *Amazon Virtual Private Cloud User Guide*.

VPC Peering Configurations

The following sections describe supported VPC peering configurations. You may require a configuration that allows routing between the entire CIDR block of each VPC, or a configuration that limits routing to specific subnets or IP addresses

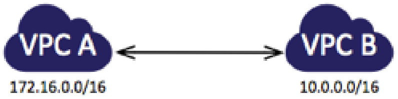
Topics

Configurations With Routes to an Entire CIDR Block

This section demonstrates the configuration for VPC peering connections in which you configure your route tables to access to the entire CIDR block of the peer VPC. For more information about scenarios in which you might need a specific VPC peering connection configuration, see VPC Peering Scenarios (p. 4). For more information about creating and working with VPC peering connections in the VPC console, see VPC Peering in the *Amazon Virtual Private Cloud User Guide*.

Two VPCs Peered Together

You have a VPC peering connection (`pcx-11112222`) between VPC A and VPC B, which are in the same AWS account, and do not have overlapping CIDR blocks.



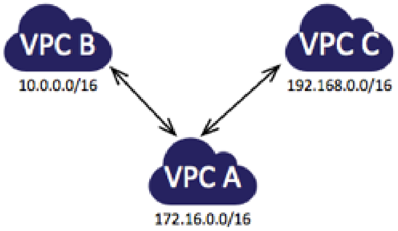
You may want to use this kind of configuration when you have a two VPCs that require access to each others' resources. For example, you set up VPC A for your accounting records, and VPC B for your financial records, and now you want each VPC to be able to access each others' resources without restriction.

The route tables for each VPC point to the relevant VPC peering connection to access the entire CIDR block of the peer VPC.

Route Tables	Destination	Target
VPC A's route table	172.16.0.0/16	Local
	10.0.0.0/16	pcx-11112222
VPC B's route table	10.0.0.0/16	Local
	172.16.0.0/16	pcx-11112222

One VPC Peered With Two VPCs

You have a central VPC (VPC A), and you have a VPC peering connection between VPC A and VPC B (`pcx-12121212`), and between VPC A and VPC C (`pcx-23232323`). The VPCs are in the same AWS account, and do not have overlapping CIDR blocks.



You may want to use this 'flying V' configuration when you have resources on a central VPC, such as a repository of services, that other VPCs need to access. The other VPCs do not need access to each others' resources; they only need access to resources on the central VPC.

> **Note**
> VPC B and VPC C cannot send traffic directly to each other through VPC A. VPC peering does not support transitive peering relationships, nor edge to edge routing. You must create a VPC peering connection between VPC B and VPC C in order to route traffic directly between them. For more information, see Three VPCs Peered Together (p. 8). For more information about unsupported peering scenarios, see Invalid VPC Peering Connection Configurations (p. 21).

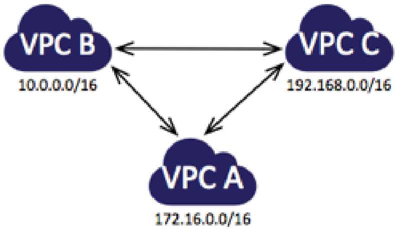
The route tables for each VPC point to the relevant VPC peering connection to access the entire CIDR block of the peer VPC.

Route Tables	Destination	Target
VPC A's route table	172.16.0.0/16	Local
	10.0.0.0/16	pcx-12121212
	192.168.0.0/16	pcx-23232323
VPC B's route table	10.0.0.0/16	Local
	172.16.0.0/16	pcx-12121212
VPC C's route table	192.168.0.0/16	Local
	172.16.0.0/16	pcx-23232323

Three VPCs Peered Together

You have peered three VPCs together in a full mesh configuration. The VPCs are in the same AWS account and do not have overlapping CIDR blocks:

- VPC A is peered to VPC B through VPC peering connection `pcx-aaaabbbb`
- VPC A is peered to VPC C through VPC peering connection `pcx-aaaacccc`
- VPC B is peered to VPC C through VPC peering connection `pcx-bbbbcccc`



You may want to use this full mesh configuration when you have separate VPCs that need to share resources with each other without restriction; for example, as a file sharing system.

The route tables for each VPC point to the relevant VPC peering connection to access the entire CIDR block of the peer VPCs.

Route Tables	Destination	Target
VPC A's route table	172.16.0.0/16	Local
	10.0.0.0/16	pcx-aaaabbbb
	192.168.0.0/16	pcx-aaaacccc
VPC B's route table	10.0.0.0/16	Local
	172.16.0.0/16	pcx-aaaabbbb
	192.168.0.0/16	pcx-bbbbcccc
VPC C's route table	192.168.0.0/16	Local
	172.16.0.0/16	pcx-aaaacccc
	10.0.0.0/16	pcx-bbbbcccc

One VPC Peered With Multiple VPCs

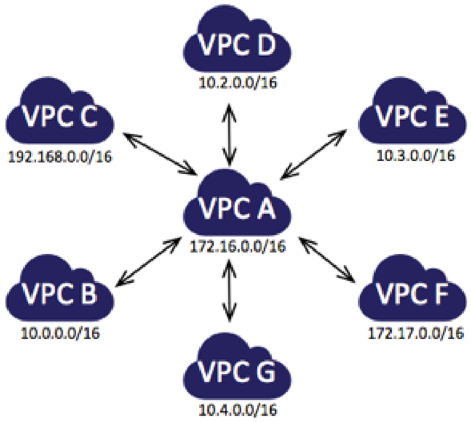
You have a central VPC (VPC A) that's peered to the following VPCs:

- VPC B through `pcx-aaaabbbb`
- VPC C through `pcx-aaaacccc`
- VPC D through `pcx-aaaadddd`
- VPC E through `pcx-aaaaeeee`
- VPC F through `pcx-aaaaffff`
- VPC G through `pcx-aaaagggg`

VPC A is peered with all other VPCs, but the other VPCs are not peered to each other. The VPCs are in the same AWS account and do not have overlapping CIDR blocks.

Note

None of the other VPCs can send traffic directly to each other through VPC A. VPC peering does not support transitive peering relationships, nor edge to edge routing. You must create a VPC peering connection between the other VPCs in order to route traffic between them. For more information, see Multiple VPCs Peered Together (p. 10). For more information about unsupported peering scenarios, see Invalid VPC Peering Connection Configurations (p. 21).



You may want to use this spoke configuration when you have resources on a central VPC, such as a repository of services, that other VPCs need to access. The other VPCs do not need access to each others' resources; they only need access to resources on the central VPC.

The route tables for each VPC point to the relevant VPC peering connection to access the entire CIDR block of the peer VPC.

Route Tables	Destination	Target
VPC A's route table	172.16.0.0/16	Local
	10.0.0.0/16	pcx-aaaabbbb
	192.168.0.0/16	pcx-aaaacccc
	10.2.0.0/16	pcx-aaaadddd
	10.3.0.0/16	pcx-aaaaeeee
	172.17.0.0/16	pcx-aaaaffff
	10.4.0.0/16	pcx-aaaagggg
VPC B's route table	10.0.0.0/16	Local
	172.16.0.0/16	pcx-aaaabbbb
VPC C's route table	192.168.0.0/16	Local
	172.16.0.0/16	pcx-aaaacccc
VPC D's route table	10.2.0.0/16	Local
	172.16.0.0/16	pcx-aaaadddd

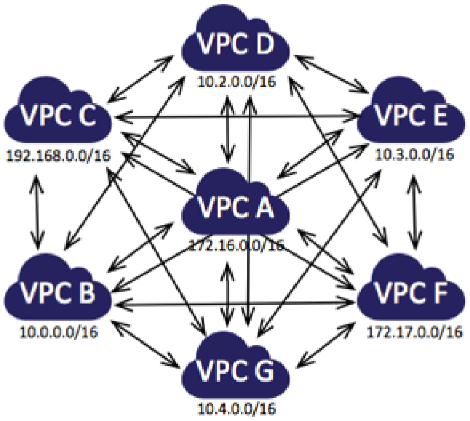
Route Tables	Destination	Target
VPC E's route table	10.3.0.0/16	Local
	172.16.0.0/16	pcx-aaaaeeee
VPC F's route table	172.17.0.0/16	Local
	172.16.0.0/16	pcx-aaaaaffff
VPC G's route table	10.4.0.0/16	Local
	172.16.0.0/16	pcx-aaaagggg

Multiple VPCs Peered Together

You have peered seven VPCs together in a full mesh configuration:

VPCs	VPC Peering Connection
A and B	pcx-aaaabbbb
A and C	pcx-aaaacccc
A and D	pcx-aaaadddd
A and E	pcx-aaaaeeee
A and F	pcx-aaaaffff
A and G	pcx-aaaagggg
B and C	pcx-bbbbcccc
B and D	pcx-bbbbdddd
B and E	pcx-bbbbeeee
B and F	pcx-bbbbffff
B and G	pcx-bbbbgggg
C and D	pcx-ccccdddd
C and E	pcx-cccceeee
C and F	pcx-ccccffff
C and G	pcx-ccccgggg
D and E	pcx-ddddeeee
D and F	pcx-ddddffff
D and G	pcx-ddddgggg
E and F	pcx-eeeeffff
E and G	pcx-eeeegggg
F and G	pcx-ffffgggg

The VPCs are in the same AWS account and do not have overlapping CIDR blocks.



You may want to use this full mesh configuration when you have multiple VPCs that must be able to access each others' resources without restriction; for example, as a file sharing network.

The route tables for each VPC point to the relevant VPC peering connection to access the entire CIDR block of the peer VPC.

Route Tables	Destination	Target
VPC A's route table	172.16.0.0/16	Local
	10.0.0.0/16	pcx-aaaabbbb
	192.168.0.0/16	pcx-aaaacccc
	10.2.0.0/16	pcx-aaaadddd
	10.3.0.0/16	pcx-aaaaeeee
	172.17.0.0/16	pcx-aaaaffff
	10.4.0.0/16	pcx-aaaagggg
VPC B's route table	10.0.0.0/16	Local
	172.16.0.0/16	pcx-aaaabbbb
	192.168.0.0/16	pcx-bbbbcccc
	10.2.0.0/16	pcx-bbbbdddd
	10.3.0.0/16	pcx-bbbbeeeee
	172.17.0.0/16	pcx-bbbbffff
	10.4.0.0/16	pcx-bbbbgggg

Route Tables	Destination	Target
VPC C's route table	192.168.0.0/16	Local
	172.16.0.0/16	pcx-aaaacccc
	10.0.0.0/16	pcx-ccccbbbb
	10.2.0.0/16	pcx-ccccdddd
	10.3.0.0/16	pcx-cccceeee
	172.17.0.0/16	pcx-ccccffff
	10.4.0.0/16	pcx-ccccgggg
VPC D's route table	10.2.0.0/16	Local
	172.16.0.0/16	pcx-aaaadddd
	10.0.0.0/16	pcx-bbbbdddd
	192.168.0.0/16	pcx-ccccdddd
	10.3.0.0/16	pcx-ddddeeee
	172.17.0.0/16	pcx-ddddffff
	10.4.0.0/16	pcx-ddddgggg
VPC E's route table	10.3.0.0/16	Local
	172.16.0.0/16	pcx-aaaaeeee
	10.0.0.0/16	pcx-bbbbeeee
	192.168.0.0/16	pcx-cccceeee
	10.2.0.0/16	pcx-ddddeeee
	172.17.0.0/16	pcx-eeeeffff
	10.4.0.0/16	pcx-eeeegggg
VPC F's route table	172.17.0.0/16	Local
	172.16.0.0/16	pcx-aaaaaffff
	10.0.0.0/16	pcx-bbbbffff
	192.168.0.0/16	pcx-ccccffff
	10.2.0.0/16	pcx-ddddffff
	10.3.0.0/16	pcx-eeeeffff
	10.4.0.0/16	pcx-ffffgggg

Route Tables	Destination	Target
VPC G's route table	10.4.0.0/16	Local
	172.16.0.0/16	pcx-aaaagggg
	10.0.0.0/16	pcx-bbbbgggg
	192.168.0.0/16	pcx-ccccgggg
	10.2.0.0/16	pcx-ddddgggg
	10.3.0.0/16	pcx-eeeegggg
	172.17.0.0/16	pcx-ffffgggg

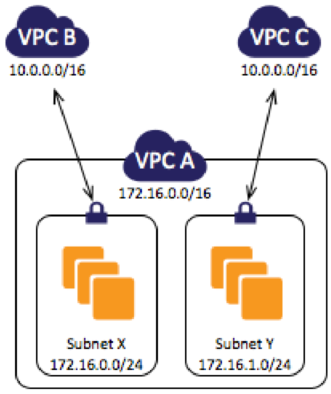
Configurations With Routes to Specific Subnets or IP Addresses

This section demonstrates the configurations for VPC peering connections in which you provide access to part of the CIDR block or a specific instance within the peer VPC. In these examples, a central VPC is peered to two or more VPCs that have overlapping CIDR blocks. For examples of scenarios in which you might need a specific VPC peering connection configuration, see VPC Peering Scenarios (p. 4). For more information about creating and working with VPC peering connections in the VPC console, see VPC Peering in the *Amazon Virtual Private Cloud User Guide*.

Topics

Two VPCs Peered to Two Subnets in One VPC

You have a central VPC (VPC A), and you have a VPC peering connection between VPC A and VPC B (`pcx-aaaabbbb`), and between VPC A and VPC C (`pcx-aaaacccc`). VPC A has two subnets - one for each VPC peering connection.



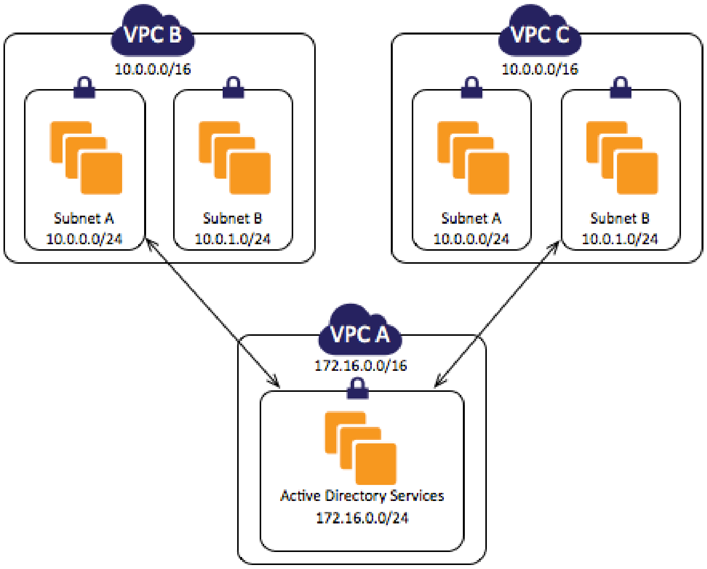
You may want to use this kind of configuration when you have a central VPC with separate sets of resources in different subnets. Other VPCs may require access to some of the resources, but not all of them.

The route table for subnet X points to VPC peering connection `pcx-aaaabbbb` to access the entire CIDR block of VPC B. VPC B's route table points to `pcx-aaaabbbb` to access the CIDR block of only subnet X in VPC A. Similarly, the route table for subnet Y points to VPC peering connection `pcx-aaaacccc` to access the entire CIDR block of VPC C. VPC C's route table points to `pcx-aaaacccc` to access the CIDR block of only subnet Y in VPC A.

Route Tables	Destination	Target
Subnet X's route table in VPC A	172.16.0.0/16	Local
	10.0.0.0/16	pcx-aaaabbbb
Subnet Y's route table in VPC A	172.16.0.0/16	Local
	10.0.0.0/16	pcx-aaaacccc
VPC B's route table	10.0.0.0/16	Local
	172.16.0.0/24	pcx-aaaabbbb
VPC C's route table	10.0.0.0/16	Local
	172.16.1.0/24	pcx-aaaacccc

One VPC Peered to Specific Subnets in Two VPCs

You have a central VPC (VPC A) with one subnet, and you have a VPC peering connection between VPC A and VPC B (`pcx-aaaabbbb`), and between VPC A and VPC C (`pcx-aaaacccc`). VPC B and VPC C each have two subnets, and only one in each is used for the peering connection with VPC A.



You may want to use this kind of configuration when you have a central VPC that has a single set of resources, such as Active Directory services, that other VPCs need to access. The central VPC does not require full access to the VPCs that it's peered with.

The route table for VPC A points to both VPC peering connections to access only specific subnets in VPC B and VPC C. The route tables for the subnets in VPC B and VPC C point to their VPC peering connections to access VPC A's subnet.

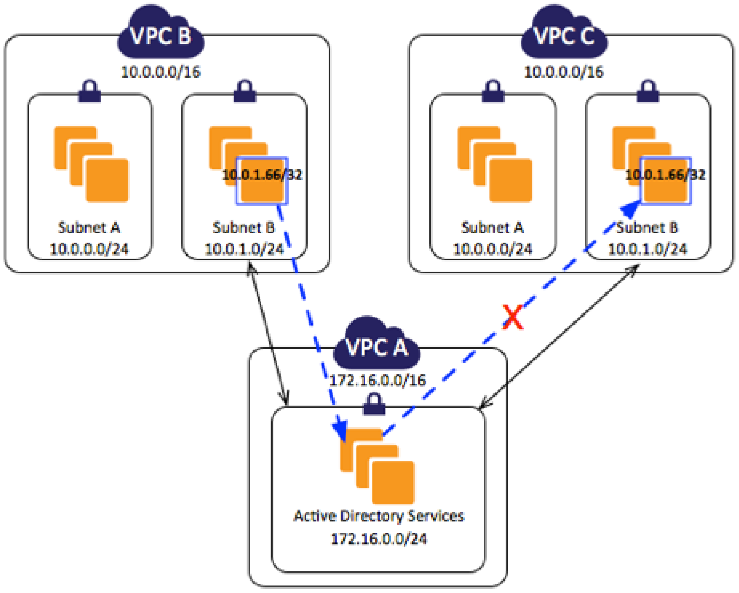
Route Tables	Destination	Target
VPC A's route table	172.16.0.0/16	Local
	10.0.0.0/24	pcx-aaaabbbb
	10.0.1.0/24	pcx-aaaacccc
Subnet A's route table in VPC B	10.0.0.0/24	Local
	172.16.0.0/24	pcx-aaaabbbb
Subnet B's route table in VPC C	10.0.1.0/24	Local
	172.16.0.0/24	pcx-aaaacccc

Routing for Response Traffic

If you have a VPC peered with multiple VPCs that have overlapping or matching 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 example, you have the same configuration as the example above - one VPC peered to specific subnets in two VPCS. VPC B and VPC C have matching CIDR blocks, and their subnets have matching CIDR blocks. The route tables for VPC A, subnet A in VPC B, and subnet B in VPC C remain unchanged. The route table for subnet B in VPC B points to the VPC peering connection `pcx-aaaabbbb` to access VPC A's subnet.

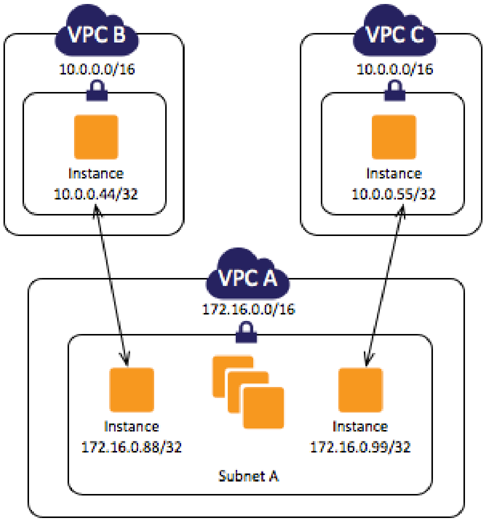
Route Table	Destination	Target
Subnet B's route table in VPC B	10.0.0.0/16	Local
	172.16.0.0/24	pcx-aaaabbbb



An instance in subnet B in VPC B with a private IP address of `10.0.1.66/32` sends traffic to the Active Directory server in VPC A using VPC peering connection `pcx-aaaabbbb`. VPC A sends the response traffic to `10.0.1.66/32`. However, VPC A's route table is configured to send all traffic within the `10.0.1.0/24` range of IP addresses to VPC peering connection `pcx-aaaacccc`, which is subnet B in VPC C. If subnet B in VPC C has an instance with an IP address of `10.0.1.66/32`, it will receive the response traffic from VPC A. The instance in subnet B in VPC B will not receive a response to its request to VPC A.

Instances in One VPC Peered to Instances in Two VPCs

You have a central VPC (VPC A) with one subnet, and you have a VPC peering connection between VPC A and VPC B (`pcx-aaaabbbb`), and between VPC A and VPC C (`pcx-aaaacccc`). VPC A has one subnet that has multiple instances; one for each of the VPCs that it's peered with. You may want to use this kind of configuration to limit peering traffic to specific instances.

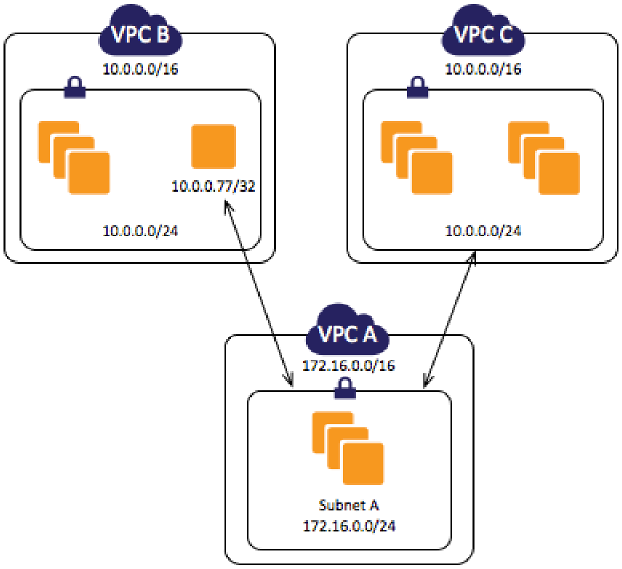


Each VPC's route table points to the relevant VPC peering connection to access a single IP address (and therefore a specific instance) in the peer VPC.

Route Tables	Destination	Target
VPC A's route table	172.16.0.0/16	Local
	10.0.0.44/32	pcx-aaaabbbb
	10.0.0.55/32	pcx-aaaacccc
VPC B's route table	10.0.0.0/16	Local
	172.16.0.88/32	pcx-aaaabbbb
VPC C's route table	10.0.0.0/16	Local
	172.16.0.99/32	pcx-aaaacccc

One VPC Peered With Two VPCs Using Longest Prefix Match

You have a central VPC (VPC A) with one subnet, and you have a VPC peering connection between VPC A and VPC B (`pcx-aaaabbbb`), and between VPC A and VPC C (`pcx-aaaacccc`). VPC B and VPC C have matching CIDR blocks. You want to use VPC peering connection `pcx-aaaabbbb` to route traffic between VPC A and specific instance in VPC B. All other traffic destined for the `10.0.0.0/16` IP address range is routed through `pcx-aaaacccc` between VPC A and VPC C.



VPC route tables use longest prefix match to select the most specific route across the intended VPC peering connection. All other traffic is routed through the next matching route, in this case, across the VPC peering connection `pcx-aaaacccc`.

Route Tables	Destination	Target
VPC A's route table	172.16.0.0/16	Local
	10.0.0.77/32	pcx-aaaabbbb
	10.0.0.0/16	pcx-aaaacccc
VPC B's route table	10.0.0.0/16	Local
	172.16.0.0/16	pcx-aaaabbbb
VPC C's route table	10.0.0.0/16	Local
	172.16.0.0/16	pcx-aaaacccc

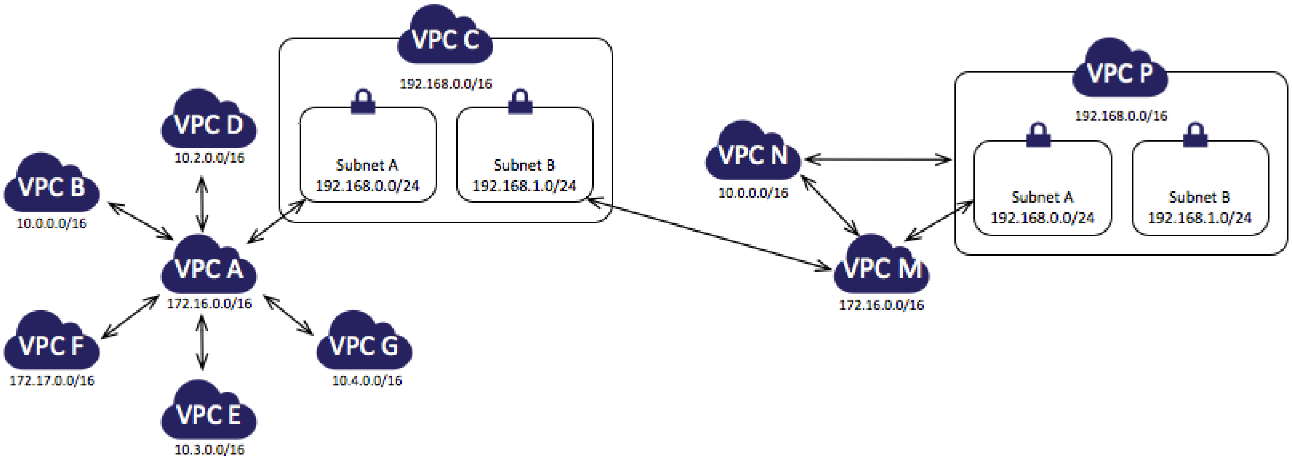
Important
If an instance other than `10.0.0.77/32` in VPC B sends traffic to VPC A, the response traffic may be routed to VPC C instead of VPC B. For more information, see Routing for Response Traffic (p. 15).

Multiple VPC Configurations

In this example, a central VPC (VPC A) is peered with multiple VPCs in a spoke configuration. For more information about this type of configuration, see One VPC Peered With Multiple VPCs (p. 8). You also have three VPCs (VPCs M, N and P) peered together in a full mesh configuration. For more information about this type of configuration, see Three VPCs Peered Together (p. 8).

VPC C also has a VPC peering connection with VPC M (`pcx-ccccmmmm`). VPC A and VPC M have overlapping CIDR blocks. This means that peering traffic between VPC A and VPC C is limited to a specific subnet (subnet A) in VPC C. This is to ensure that if VPC C receives a request from VPC A or VPC M, it sends the response traffic to the correct 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. 15).

Similarly, VPC C and VPC P have overlapping CIDR blocks. Peering traffic between VPC M and VPC C is limited to subnet B in VPC C, and peering traffic between VPC M and VPC P is limited to subnet A in VPC P. This is to ensure that if VPC M receives peering traffic from VPC C or VPC P, it sends the response traffic back to the correct VPC.



The route tables for VPCs B, D, E, F and G point to the relevant peering connections to access the full CIDR block for VPC A, and VPC A's route table points to the relevant peering connections for VPCs B, D, E, F and G to access their full CIDR blocks. For peering connection `pcx-aaaacccc`, VPC A's route table routes traffic only to subnet A in VPC C (`192.168.0.0/24`) and subnet A's route table in VPC C points to the full CIDR block of VPC A.

VPC N's route table points to the relevant peering connections to access the full CIDR blocks of VPC M and VPC P, and VPC P's route table points to the relevant peering connection to access the full CIDR block of VPC N. Subnet A's route table in VPC P points to the relevant peering connection to access the full CIDR block of VPC M. VPC M's route table points to the relevant peering connection to access subnet B in VPC C, and subnet A in VPC P.

Route Tables	Destination	Target
VPC A's route table	172.16.0.0/16	Local
	10.0.0.0/16	pcx-aaaabbbb
	192.168.0.0/24	pcx-aaaacccc
	10.2.0.0/16	pcx-aaaadddd
	10.3.0.0/16	pcx-aaaaeeee
	172.17.0.0/16	pcx-aaaaffff
	10.4.0.0/16	pcx-aaaagggg
VPC B's route table	10.0.0.0/16	Local
	172.16.0.0/16	pcx-aaaabbbb
Subnet A's route table in VPC C	192.168.0.0/24	Local
	172.16.0.0/16	pcx-aaaacccc
Subnet B's route table in VPC C	192.168.1.0/24	Local
	172.16.0.0/16	pcx-ccccmmmm
VPC D's route table	10.2.0.0/16	Local
	172.16.0.0/16	pcx-aaaadddd
VPC E's route table	10.3.0.0/16	Local
	172.16.0.0/16	pcx-aaaaeeee
VPC F's route table	172.17.0.0/16	Local
	172.16.0.0/16	pcx-aaaaaffff
VPC G's route table	10.4.0.0/16	Local
	172.16.0.0/16	pcx-aaaagggg
VPC M's route table	172.16.0.0/16	Local
	192.168.1.0/24	pcx-ccccmmmm
	10.0.0.0/16	pcx-mmmmnnnn
	192.168.0.0/24	pcx-mmmmpppp

Route Tables	Destination	Target
VPC N's route table	10.0.0.0/16	Local
	172.16.0.0/16	pcx-mmmmnnnn
	192.168.0.0/16	pcx-nnnnpppp
VPC P's route table	192.168.0.0/16	Local
	10.0.0.0/16	pcx-nnnnpppp
	172.16.0.0/16	pcx-mmmmpppp

Invalid VPC Peering Connection Configurations

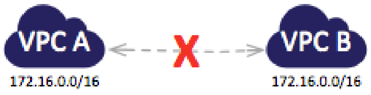
This section describes VPC peering connection configurations that are invalid.

For more information about VPC peering limitations, see VPC Peering Limitations (p. 2).

Topics

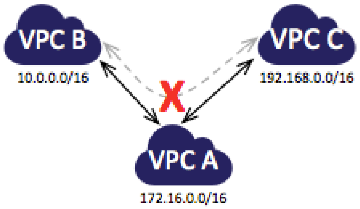
Overlapping CIDR blocks

You cannot create a VPC peering connection between VPCs with matching or overlapping CIDR blocks.



Transitive Peering

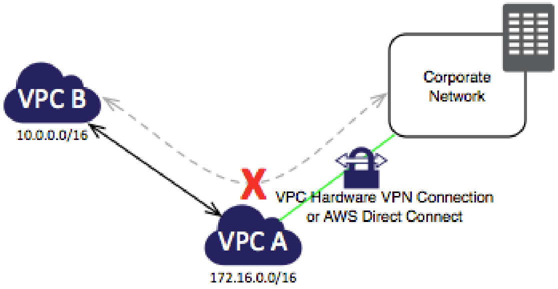
You have a VPC peering connection between VPC A and VPC B (`pcx-aaaabbbb`), and between VPC A and VPC C (`pcx-aaaacccc`). There is no VPC peering connection between VPC B and VPC C. You cannot route packets directly from VPC B to VPC C through VPC A.



To route packets directly between VPC B and VPC C, you can create a separate VPC peering connection between them (provided they do not have overlapping CIDR blocks). For more information, see Three VPCs Peered Together (p. 8).

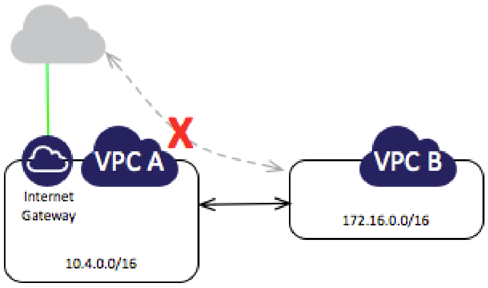
Edge to Edge Routing Through a VPN Connection

You have a VPC peering connection between VPC A and VPC B (`pcx-aaaabbbb`). VPC A also has a VPN connection to a corporate network. Edge to edge routing is not supported; you cannot use VPC A to extend the peering relationship to exist between VPC B and the corporate network. For example, traffic from the corporate network can't directly access VPC B by using the VPN connection to VPC A.



Edge to Edge Routing Through an Internet Gateway

You have a VPC peering connection between VPC A and VPC B (`pcx-abababab`). VPC A has an Internet gateway; VPC B does not. Edge to edge routing is not supported; you cannot use VPC A to extend the peering relationship to exist between VPC B and the Internet. For example, traffic from the Internet can't directly access VPC B by using the Internet gateway connection to VPC A.


Internet
Gateway
VPC A
X
10.4.0.0/16
VPC B
172.16.0.0/16