Virtual Private Cloud

Best Practices

Issue 01

Date 2024-08-07

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 VPC and Subnet Planning

Before using VPCs and subnets to build cloud networks, determine how many VPCs and subnets do you need and plan the necessary CIDR blocks and connectivity options. If you need to connect different VPCs or connect a VPC to an on-premises data center, ensure that their CIDR blocks do not conflict. Properly plan your VPCs and subnets based on the guidelines provided here to avoid CIDR block conflicts, which will make future network expansion easier.

How Do I Determine How Many VPCs I Need?

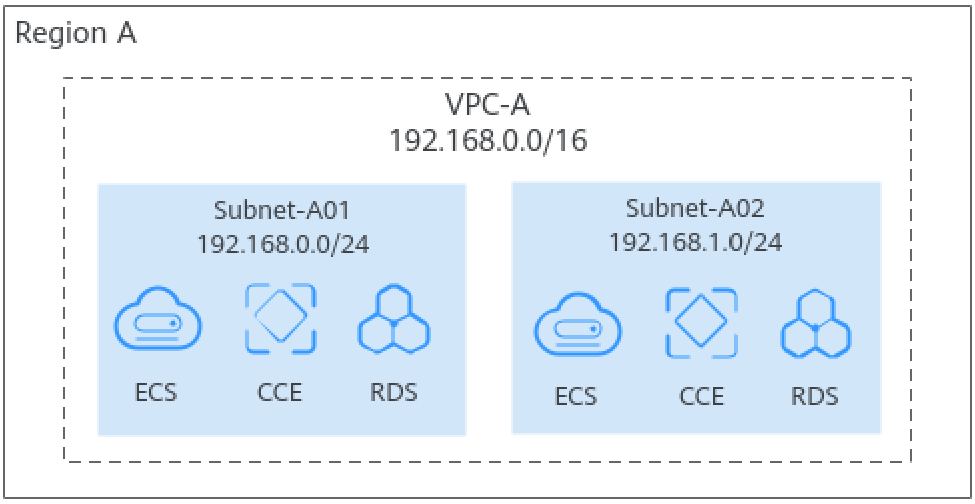
VPCs are region-specific. Cloud resources, such as ECSs, CCEs, and RDS instances, in a VPC must be in the same region as the VPC. By default, different VPCs are isolated from each other, but the subnets in a VPC can communicate with each other.

Planning a Single VPC

If your services are deployed in one region and do not have to handle a lot of traffic, you may not need network isolation. In this case, a single VPC should be enough.

You can create multiple subnets in a VPC for workloads with different requirements and associate route tables with these subnets to control traffic in and out of the subnets. In **Figure 1-1**, services are deployed on different subnets in a VPC (VPC-A in this example).

Figure 1-1 Planning a single VPC



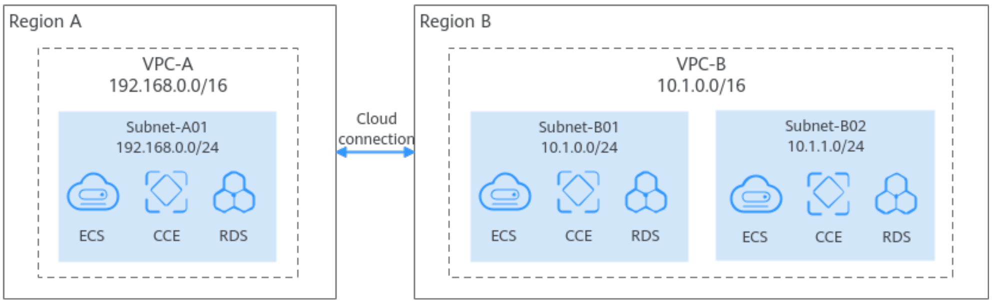
Planning Multiple VPCs

You need to plan multiple VPCs if you have:

- **Services that need to be deployed in different regions**

  VPC is a region-specific service, so services cannot be deployed across regions in a VPC. If your services are deployed in multiple regions, plan at least one VPC in each region.

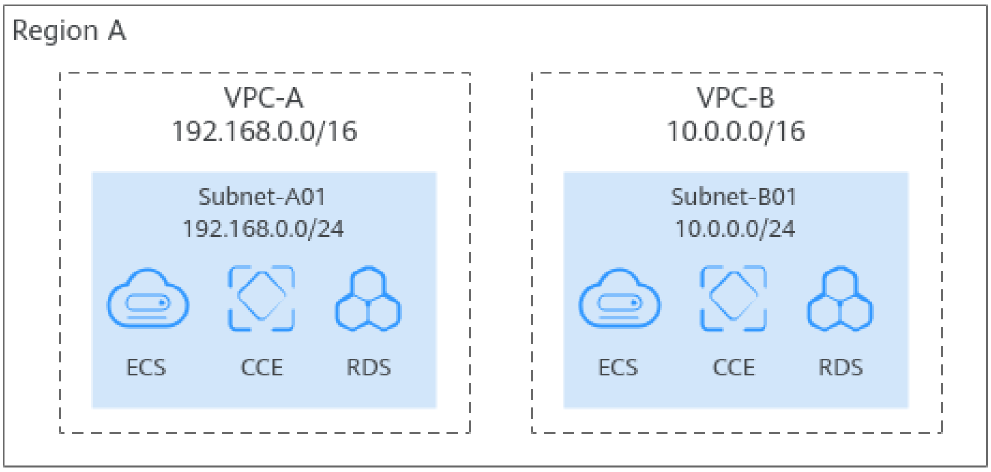
  Figure 1-2 Planning multiple VPCs

- **Services that are deployed in the same region but need network isolation.**

  If your services are deployed in the same region but need network isolation, you need to plan multiple VPCs in this region. Different VPCs are isolated from each other, so you can deploy different services in different VPCs, as shown in **Figure 1-3**. In the figure, some services are deployed in VPC-A, and some are deployed in VPC-B. The two VPCs are isolated from each other.

Figure 1-3 Planning multiple VPCs



NOTE

By default, you can create a maximum of five VPCs in each region. If this cannot meet your service requirements, **request a quota increase**.

How Do I Determine How Many Subnets I Need?

A subnet is a unique CIDR block with a range of IP addresses in a VPC. All cloud resources in a VPC must be deployed on subnets.

You can create different subnets for different services in a VPC. For example, you can create three subnets in a VPC, one subnet for web services, one for management services, and the third one for data services. Additionally, you can use network ACLs to control access to each subnet.

Note the following when selecting subnets and AZs for your resources:

- All instances in different subnets of the same VPC can communicate with each other by default, and the subnets can be located in different AZs. If you have a VPC with two subnets in it and they are located in different AZs, they can communicate with each other by default.
- A cloud resource can be in a different AZ from its subnet. For example, a cloud server in AZ 1 can be in a subnet in AZ 3. If AZ 3 becomes faulty, cloud servers in AZ 1 can still use the subnet in AZ 3, and your services are not interrupted.

NOTE

By default, you can create a maximum of 100 subnets in each region. If this cannot meet your service requirements, **request a quota increase**.

How Do I Plan CIDR Blocks for VPCs and Subnets?

After VPCs and subnets are created, their CIDR blocks cannot be changed. To ensure smooth service expansion and O&M, properly plan VPC and subnet CIDR blocks that best suit your service size and communication requirements.

NOTE

Both IPv4 and IPv6 CIDR blocks can be assigned to a subnet. You can customize IPv4 CIDR blocks but not IPv6 CIDR blocks. The system assigns an IPv6 CIDR block with a 64-bit mask to each subnet, for example, 2407:c080:802:1b32::/64.

Planning VPC CIDR Blocks

When creating a VPC, you need to specify an IPv4 CIDR block for it. Consider the following when selecting a CIDR block:

- Reserve sufficient IP addresses for subsequent service expansion.
- Avoid CIDR block conflicts. To enable communications between VPCs or between a VPC and an on-premises data center, ensure their CIDR blocks do not overlap.

When you create a VPC, you specify a primary IPv4 CIDR block for the VPC, which cannot be changed. You can **add a secondary IPv4 CIDR block to the VPC** if required.

When you create a VPC, we recommend that you use the private IPv4 address ranges specified in **RFC 1918** as the CIDR block, as described in **Table 1-1**.

Table 1-1 VPC CIDR blocks (RFC 1918)

VPC CIDR Block	IP Address Range	Netmask	Example CIDR Block
10.0.0.0/8-24	10.0.0.0–10.255.255.255	8-24	10.0.0.0/8
172.16.0.0/12-24	172.16.0.0–172.31.255.255	12-24	172.30.0.0/16
192.168.0.0/16-24	192.168.0.0–192.168.255.255	16-24	192.168.0.0/24

In addition to the preceding addresses, you can create a VPC with a publicly routable CIDR block that falls outside of the private IPv4 address ranges specified in RFC 1918. However, the reserved system and public CIDR blocks listed in **Table 1-2** must be excluded:

Table 1-2 Reserved system and public CIDR blocks

Reserved System CIDR Blocks	Reserved Public CIDR Blocks
• 100.64.0.0/10 • 214.0.0.0/7 • 198.18.0.0/15 • 169.254.0.0/16	• 0.0.0.0/8 • 127.0.0.0/8 • 240.0.0.0/4 • 255.255.255.255/32

Planning Subnet CIDR Blocks

- Subnet mask planning: The subnet CIDR block must be within the VPC CIDR block. Subnet CIDR blocks in a VPC must be unique. A subnet mask can be between the netmask of its VPC CIDR block and a /29 netmask. If a VPC CIDR block is 10.0.0.0/16, its subnet mask can be anything from 16 to 29.

  For example, if the CIDR block of a VPC is 10.0.0.0/16, you can specify 10.0.0.0/24 for a subnet in this VPC, 10.0.1.0/24 for the second subnet, and 10.0.2.0/24 for the third subnet.
- Planning CIDR block size: After a subnet is created, the CIDR block cannot be changed. You need to properly plan the CIDR block in advance based on the number of IP addresses required by your service.
  - The subnet CIDR block size cannot be too small. Ensure that the number of available IP addresses in the subnet meets service requirements. Remember that the first and last three addresses in a subnet CIDR block are reserved for system use. For example, in subnet 10.0.0.0/24, 10.0.0.1 is the gateway address, 10.0.0.253 is the system interface address, 10.0.0.254 is used by DHCP, and 10.0.0.255 is the broadcast address.
  - The subnet CIDR block cannot be too large, either. If you use a CIDR block that is too large, you may not have enough CIDR blocks available later for new subnets, which can be a problem when you want to scale out services.
- Avoiding subnet CIDR block conflicts: Avoid CIDR block conflicts if you need to connect two VPCs or connect a VPC to an on-premises data center.

  If the subnet CIDR blocks at both ends of the network conflict, **create a subnet**.

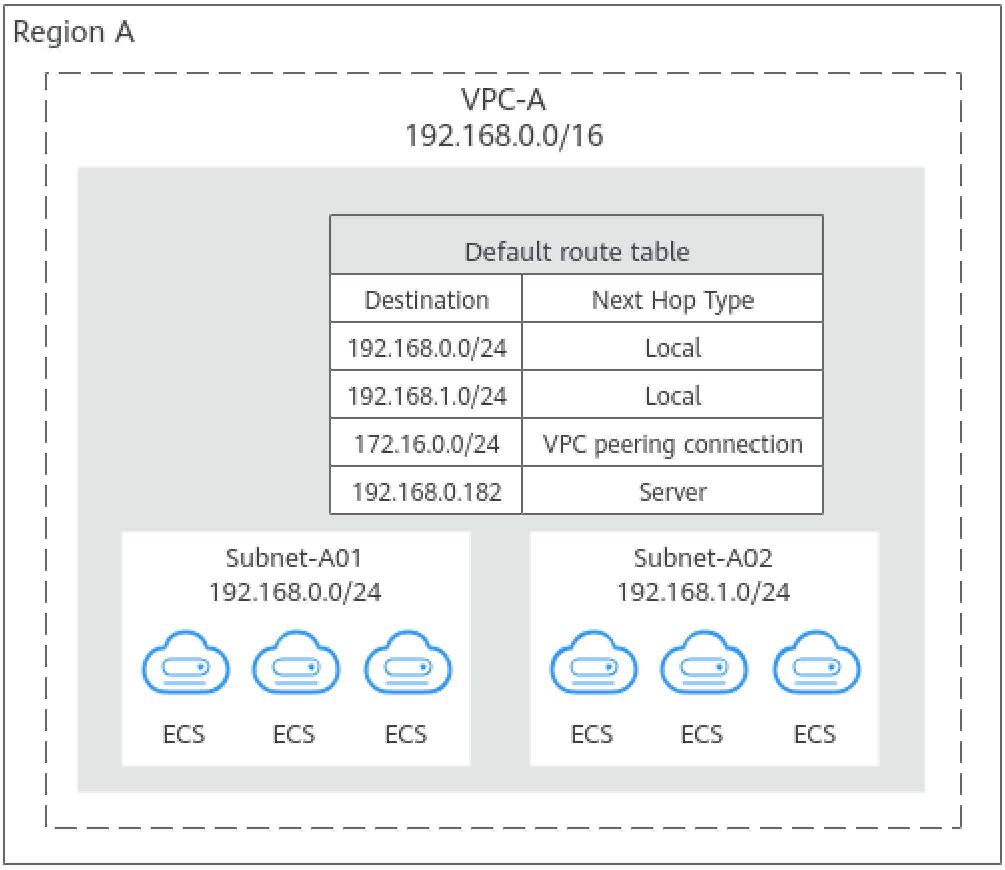
How Do I Plan How Many Route Tables I Need?

A route table contains a set of routes that are used to control the traffic in and out of your subnets in a VPC. You can configure destination, next hop, and other information for each route. A VPC can have multiple route tables. Plan route tables based on the following sections.

Planning One Route Table

If you have the same or similar requirements for controlling the network traffic to and from subnets in a VPC, you can create one route table and associate it with these subnets in this VPC. Each VPC comes with a default route table. If you create a subnet in the VPC, the subnet is associated with the default route table. You can add routes to the default route table to control where the traffic is directed. In **Figure 1-4**, VPC-A has only the default route table, and subnets Subnet-A01 and Subnet-A02 are associated with the default route table.

Figure 1-4 Planning one route table

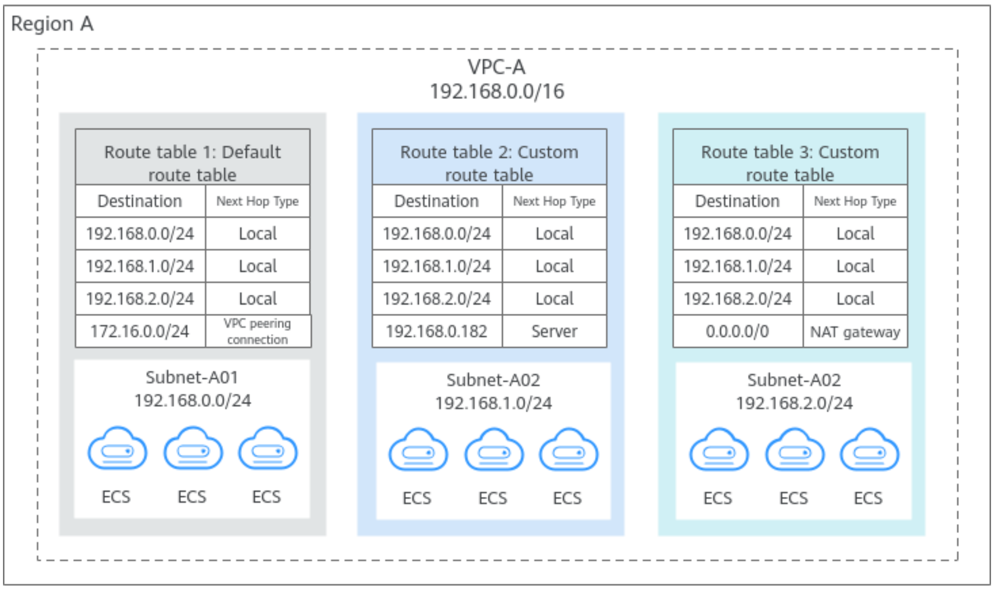


Default route table	
Destination	Next Hop Type
192.168.0.0/24	Local
192.168.1.0/24	Local
172.16.0.0/24	VPC peering connection
192.168.0.182	Server

Planning Multiple Route Tables

If you have different requirements for controlling the network traffic to and from subnets in a VPC, the default route table is not enough. You can create one or more custom route tables and associate them with these subnets in this VPC. In **Figure 1-5**, VPC-A has three route tables. Subnet-A01 is associated with default route table 1, Subnet-A02 is associated with custom route table 2, and Subnet-A03 is associated with custom route table 3.

Figure 1-5 Planning multiple route tables



Route table 1: Default route table

Destination	Next Hop Type
192.168.0.0/24	Local
192.168.1.0/24	Local
192.168.2.0/24	Local
172.16.0.0/24	VPC peering connection

Route table 2: Custom route table

Destination	Next Hop Type
192.168.0.0/24	Local
192.168.1.0/24	Local
192.168.2.0/24	Local
192.168.0.182	Server

Route table 3: Custom route table

Destination	Next Hop Type
192.168.0.0/24	Local
192.168.1.0/24	Local
192.168.2.0/24	Local
0.0.0.0/0	NAT gateway

How Do I Connect Two VPC or Connect a VPC to an On-premises Data Center?

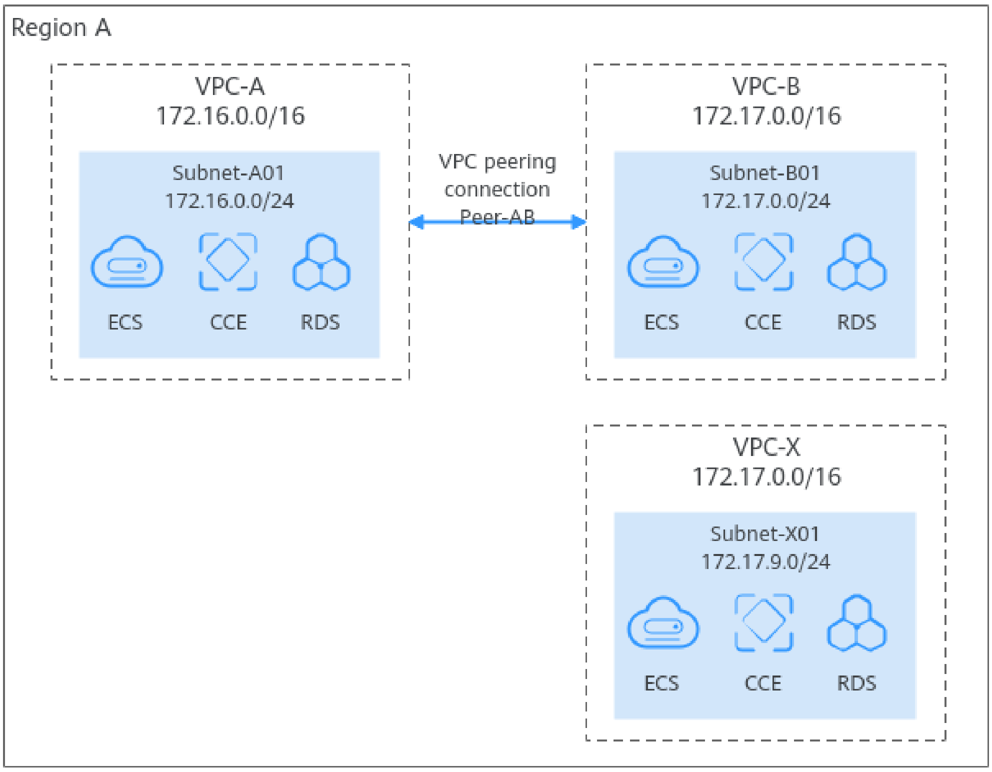
If you need to connect two VPCs or connect a VPC to an on-premises data center, ensure that their VPC CIDR blocks do not conflict.

Connecting Two VPCs

Connecting VPCs in the same region: In **Figure 1-6**, there are three VPCs in region A: VPC-A, VPC-B, and VPC-X. If you want to connect VPC-A and VPC-B, but isolate VPC-C from other VPCs:

- Ensure that the CIDR blocks of VPC-A and VPC-B connected by a peering connection (Peering-AB in this example) must be unique.
- You do not need to worry about VPC CIDR block conflicts because VPC-X does not need to communicate with other VPCs. If VPC-X and VPC-B need to communicate with each other, you can specify different CIDR blocks for the subnets in the two VPCs and create a VPC peering connection to connect the subnets.

Figure 1-6 Connecting VPCs in the same region

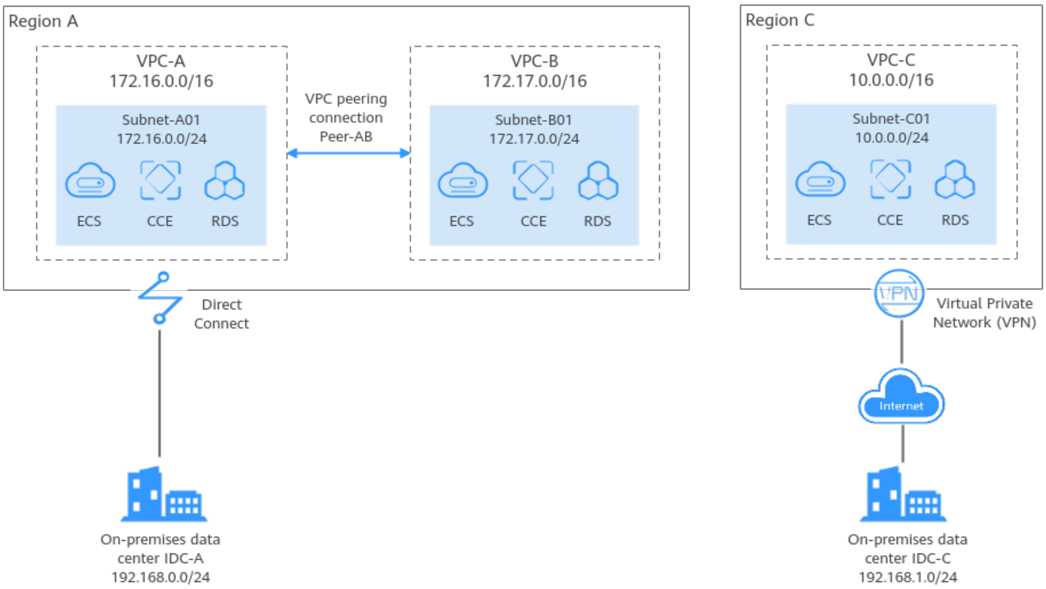


Connecting a VPC to an On-premises Data Center

In **Figure 1-7**, VPC-A and VPC-B in region A need to communicate with each other, and VPC-A needs to connect to on-premises data center IDC-A. In region C, VPC-C needs to connect to on-premises data center IDC-C.

- In region A, VPC-A and VPC-B have different CIDR blocks and can communicate with each other through a VPC peering connection. VPC-A and IDC-A have different CIDR blocks and are connected through a direct connection.
- In region C, VPC-C and IDC-C have different CIDR blocks and are connected through a VPC connection.

Figure 1-7 Connecting a VPC to an on-premises data center



Helpful Link

- You can create a VPC and an ECS to set up an IPv4 private network on the cloud and then bind an EIP to the ECS to allow the ECS to access the Internet. For details, see **Setting Up an IPv4 Network in a VPC**.
- You can create a VPC with an IPv4 and IPv6 CIDR block and create an ECS with both IPv4 and IPv6 addresses in the VPC. You can bind an EIP and add the IPv6 address of the ECS to a shared bandwidth to enable the ECS to communicate with the Internet over both IPv4 and IPv6 networks. For details, see **Setting Up an IPv4/IPv6 Dual-Stack Network in a VPC**.

2 VPC Connectivity

Accessing the Internet

Cloud resources in a VPC can use the following cloud services to connect to the Internet.

Table 2-1 Accessing the Internet

Cloud Service	Application Scenario	Description	Reference
EIP	Single ECS accesses the Internet.	You can assign an EIP and bind it to an ECS so that the ECS can access the Internet or provide services accessible from the Internet. An EIP can be bound to an ECS to enable Internet access, or unbound to disable access. Shared bandwidths can be used to lower costs.	**Configuring the VPC of ECSs That Access the Internet Using EIPs**

Cloud Service	Application Scenario	Description	Reference
NAT Gateway	Multiple ECSs share an EIP to access the Internet.	A NAT gateway offers both source network address translation (SNAT) and destination network address translation (DNAT). SNAT allows multiple ECSs in the same VPC to share EIPs to access the Internet. In this way, you can reduce management costs and prevent the EIPs of ECSs from being exposed to the Internet. DNAT implements port-level data forwarding. It maps EIP ports to ECS ports so that the ECSs in a VPC can share the same EIP and bandwidth to provide Internet-accessible services. However, DNAT does not balance traffic.	**Using SNAT to Access the Internet** **Using DNAT to Provide Services Accessible from the Internet**
ELB	Use load balancers provided by the ELB service to evenly distribute incoming traffic across multiple ECSs in high-concurrency scenarios, such as e-commerce.	Load balancers distribute traffic across multiple backend ECSs, balancing the workload on each ECS (at Layer 4 or Layer 7). You can bind EIPs to ECSs to allow the access from the Internet. ELB expands the service capabilities of your applications and improves availability by eliminating single points of failures.	**What Is ELB?**

Connecting VPCs

You can connect VPCs using the following cloud services.

Table 2-2 Connecting VPCs

Cloud Service	Application Scenario	Description	Reference
VPC Peering	Connect VPCs in the same region.	You can request a VPC peering connection with another VPC in your account or in another account, but the two VPCs must be in the same region. VPC peering connections are free of charge.	**Creating a VPC Peering Connection with Another VPC in Your Account** **Creating a VPC Peering Connection with a VPC in Another Account**
VPN	Use VPN to connect VPCs across regions at a low cost.	VPN uses an encrypted communications tunnel to connect VPCs in different regions and sends traffic over the Internet. It is inexpensive, easy to configure, and easy to use. However, VPN connections will be affected by the Internet quality.	-

Connecting to an On-premises Data Center (IDC)

If you have an on-premises data center and you do not want to migrate all of your services to the cloud, you can build a hybrid cloud, so that you can keep core data in your data center.

Table 2-3 Connecting to an on-premises data center

Cloud Service	Application Scenario	Description	Reference
VPN	Use VPN to connect a VPC to an on-premises data center at a low cost.	VPN uses an encrypted communications tunnel to connect a VPC on the cloud to an on-premises data center and sends traffic over the Internet. It is inexpensive, easy to configure, and easy to use. However, VPN connections will be affected by the Internet quality.	-

Cloud Service	Application Scenario	Description	Reference
Direct Connect	Use a physical connection to connect a VPC to an on-premises data center.	Direct Connect provides physical connections between VPCs and data centers. It features low latency and is very secure. Direct Connect is a good choice if you have strict requirements on network transmission quality and security.	**Accessing Multiple VPCs over the Same Connection**

3 Private Network Access

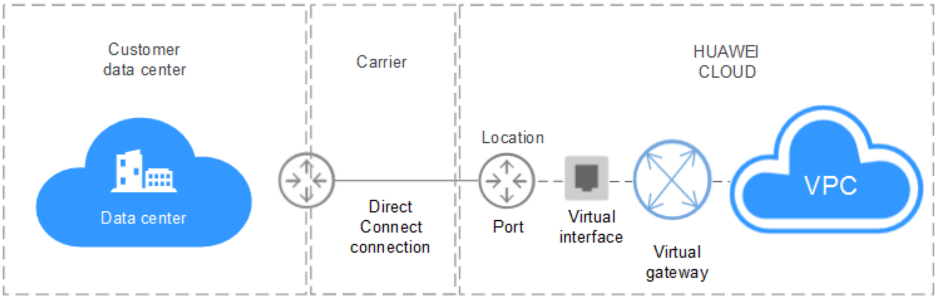
Connecting to an On-premises Data Center

You can connect a VPC to your on-premises data center. Once you have established this secure, reliable connection, you can move at scale to Huawei Cloud, a cloud with massive computing, storage, and network resources. With Huawei Cloud, you will be unaffected by sudden fluctuations in demand for services. Both Direct Connect and VPN support the connections between your data center and your VPCs on the cloud.

- Direct Connect

  Direct Connect provides high-speed, stable, and secure dedicated network connections that connect your data centers to VPCs. With Direct Connect, you can connect computers in your on-premises data center to cloud servers or hosting servers on Huawei Cloud. It maximizes cloud computing capacities and existing IT facilities to build a flexible, scalable hybrid cloud computing environment.

  Figure 3-1 Connecting to an on-premises data center with a Direct Connect connection

- VPN

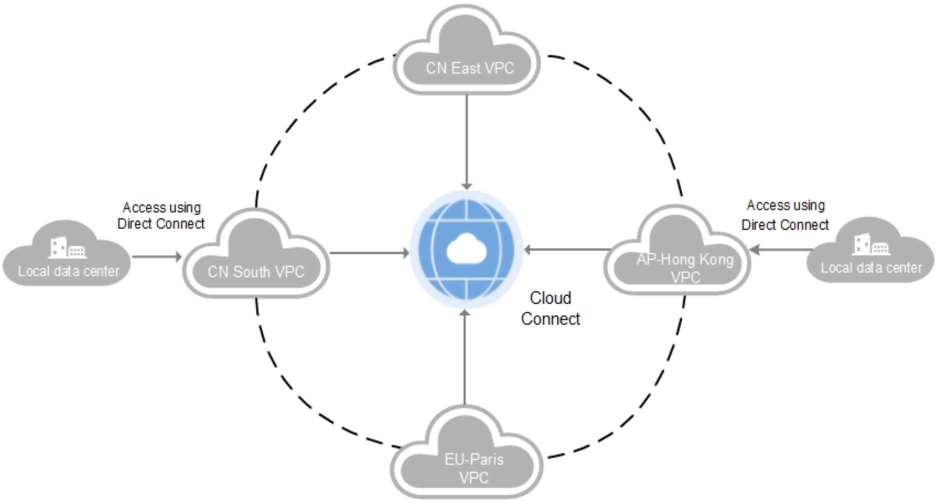
  VPN establishes a secure, encrypted communication tunnel between your local data center and your VPC on Huawei Cloud. With VPN, you can connect to a VPC and access the resources deployed there.

Connecting VPCs and Data Centers with Cloud Connect

Cloud Connect allows you to quickly build high-quality networks that can connect VPCs across regions and work with Direct Connect to connect VPCs and on-premises data centers.

With Cloud Connect, you can build a globally connected cloud network with enterprise-class scalability and communications capabilities.

Figure 3-2 Connecting VPCs and data centers with Cloud Connect



Connecting VPCs

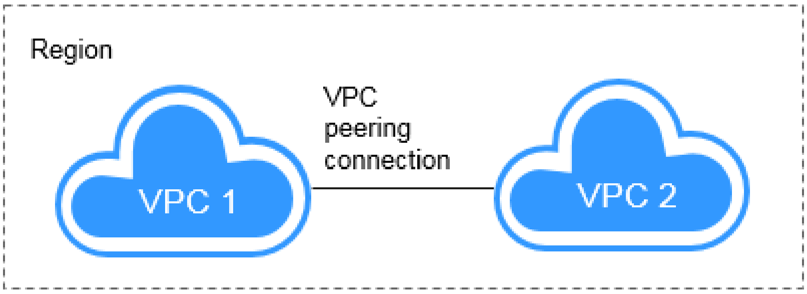
If you want to connect VPCs in the same region, you can use VPC peering connections.

If you want to connect VPCs in different regions and construct a service network across regions, you can use Direct Connect, VPN, or Cloud Connect.

- VPC peering

  You can use VPC peering connections to connect VPCs in the same region.

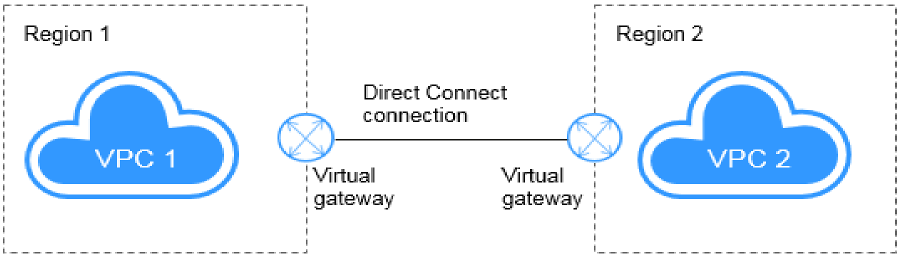
  Figure 3-3 Connecting VPCs in the same region with a VPC peering connection



- Direct Connect

  Direct Connect provides high-speed, stable, and secure dedicated network connections that connect your data centers to VPCs. With Direct Connect, you can connect computers in your on-premises data center to cloud servers or hosting servers on Huawei Cloud. It maximizes cloud computing capacities and existing IT facilities to build a flexible, scalable hybrid cloud computing environment. Direct Connect can also be used to connect VPCs in different regions.

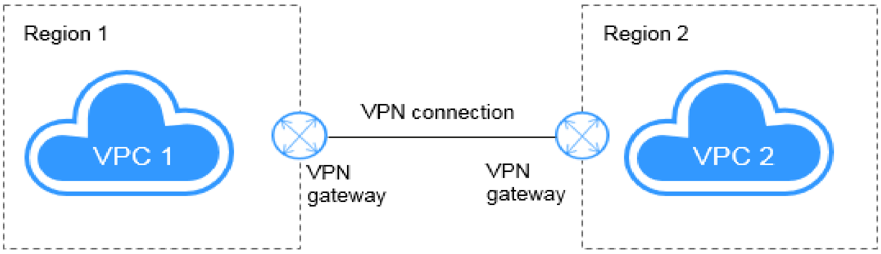
  Figure 3-4 Connecting VPCs in different regions with Direct Connect



- VPN

  VPN establishes a secure, encrypted communication tunnel between your local data center and your VPC on Huawei Cloud. With VPN, you can connect to a VPC and access the resources deployed there. VPN can connect VPCs in different regions.

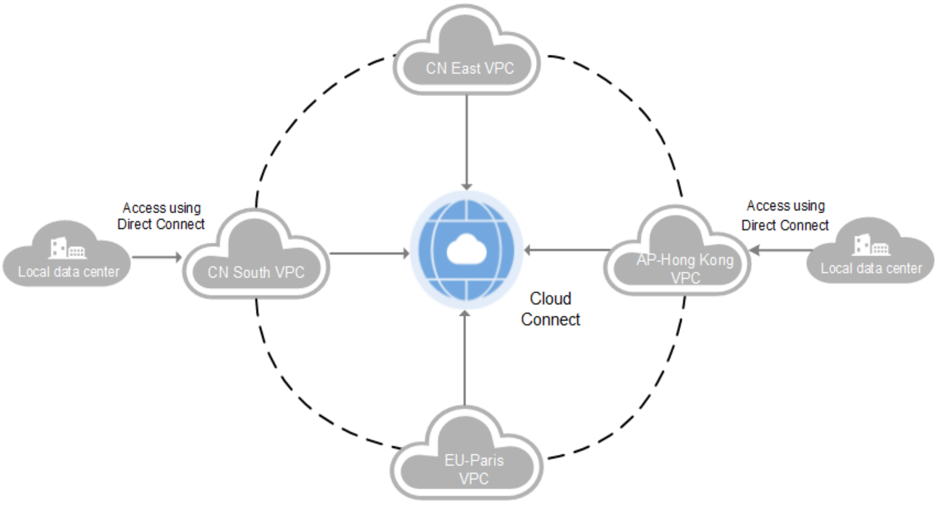
  Figure 3-5 Connecting VPCs in different regions with VPN



- Cloud Connect

  Cloud Connect allows you to quickly build high-quality networks that can connect VPCs across regions and work with Direct Connect to connect VPCs and on-premises data centers. With Cloud Connect, you can build a globally connected cloud network with enterprise-class scalability and communications capabilities.

Figure 3-6 Connecting VPCs in different regions with Cloud Connect



4 Public Network Access

Products

Cloud services, such as EIP, NAT Gateway, and ELB can be used to connect to the Internet.

- EIP

  The EIP service provides independent public IP addresses and bandwidth for Internet access. EIPs can be bound to or unbound from ECSs, BMSs, virtual IP addresses, NAT gateways, or load balancers. Various billing modes are provided to meet diverse service requirements.

- ELB

  ELB distributes access traffic among multiple ECSs to balance the application load, improving fault tolerance and expanding service capabilities of applications. You can create a load balancer, configure a listening protocol and port, and add backend servers to a load balancer. You can also check the running state of backend servers to ensure that requests are sent only to healthy servers.

- NAT Gateway

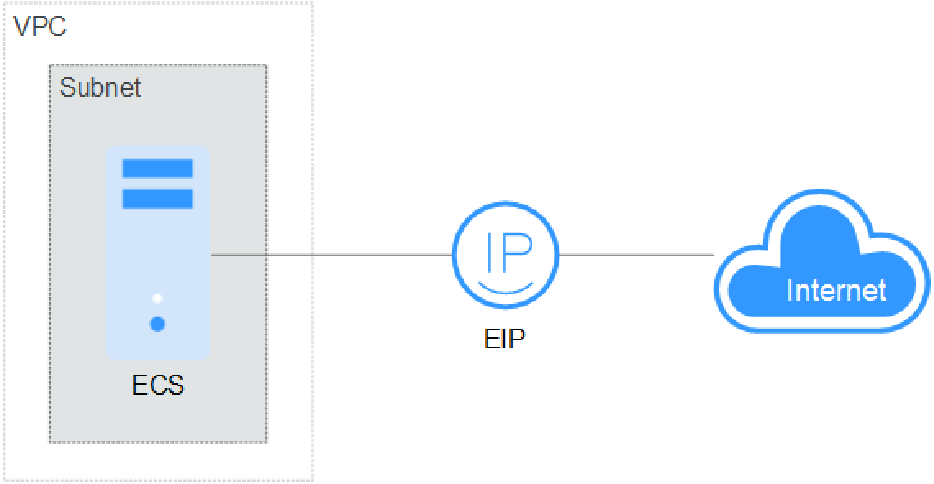
  NAT Gateway provides both SNAT and DNAT for your servers in a VPC and allows servers in your VPC to access or provide services accessible from the Internet.

Providing Services Accessible from the Internet

- Single ECS provides services accessible from the Internet.

  If you have only one application and the service traffic is small, you can assign an EIP and bind it to the ECS so that the ECS can provide services accessible from the Internet.

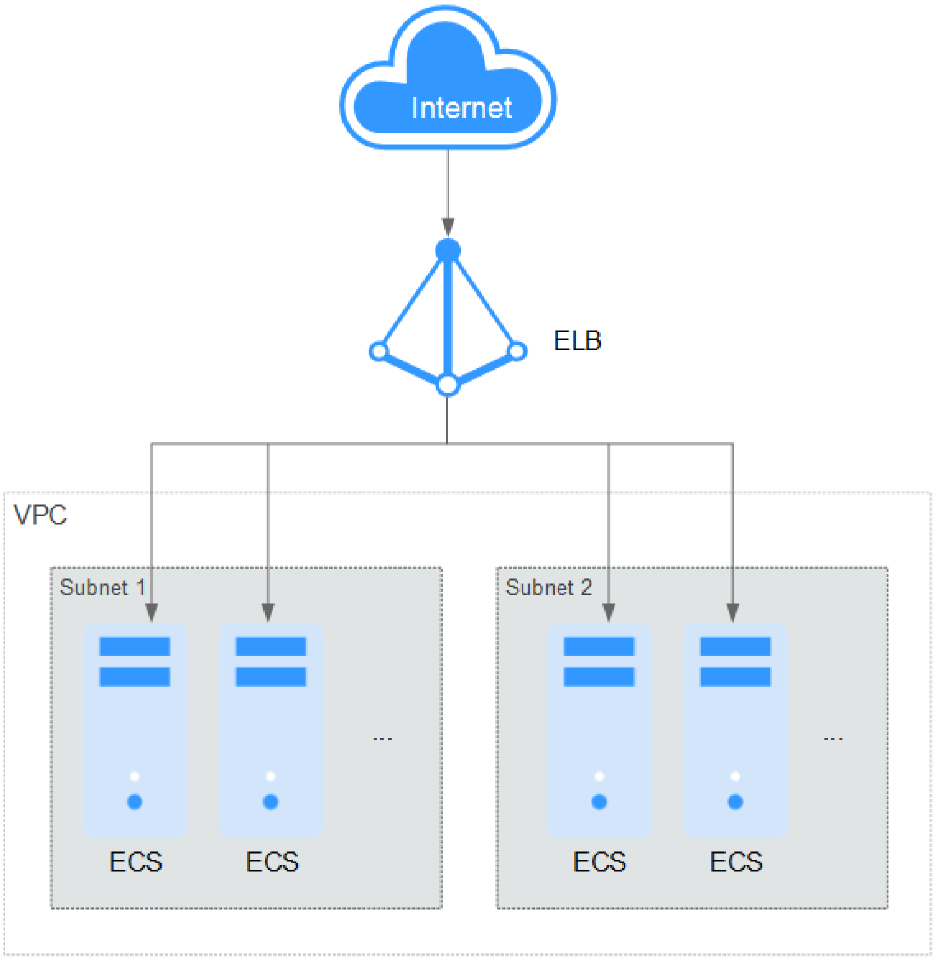
Figure 4-1 EIP



- Multiple ECSs balance workloads.

  In high-concurrency scenarios, such as e-commerce, you can use load balancers provided by the ELB service to evenly distribute incoming traffic across multiple ECSs, allowing a large number of users to concurrently access your business system or application. ELB deeply integrates with the Auto Scaling (AS) service, which enables automatic scaling based on service traffic and ensures service stability and reliability.

Figure 4-2 ELB

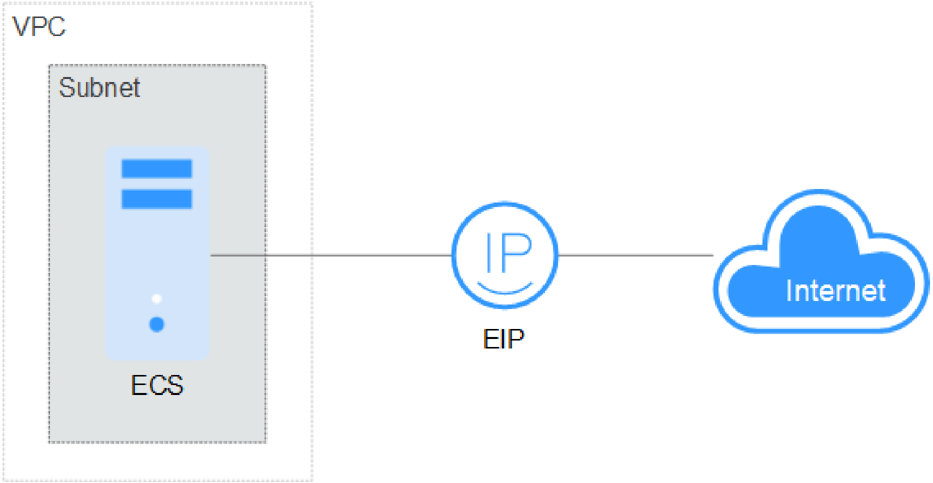


Accessing the Internet

- Single ECS accesses the Internet.

  When an ECS needs to access the Internet, you can bind an EIP to the ECS so that the ECS can access the Internet. Huawei Cloud allows your EIP to be billed on a pay-per-use basis. If you do not need to use the EIP, you can flexibly unbind it.

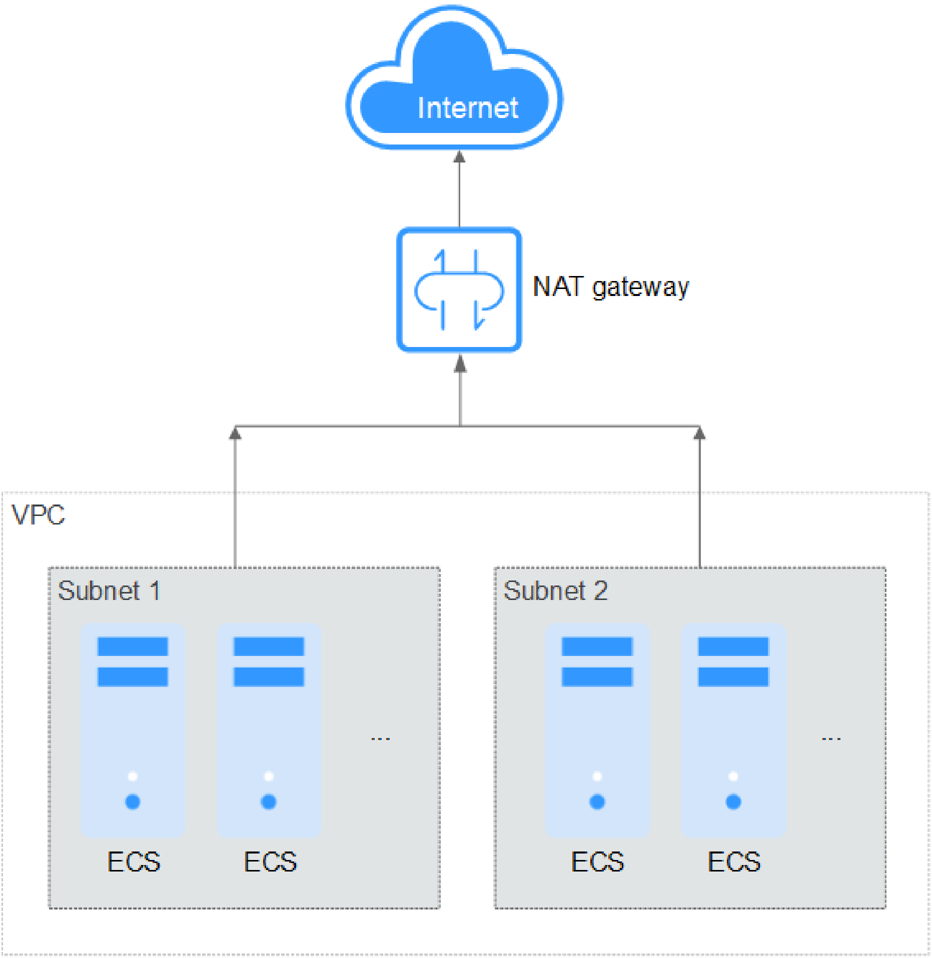
Figure 4-3 EIP



- Multiple ECSs access the Internet.

  If multiple ECSs in your VPC need to access the Internet, you can use a NAT gateway and configure SNAT rules by subnet to allow ECSs in the VPC to access the Internet. If you access to the Internet using an EIP but with no DNAT rules configured, external users cannot directly access the public network address of the NAT gateway through the Internet, ensuring ECS security.

Figure 4-4 NAT gateway



5 Lower Network Costs

You can select a proper product and billing mode based on your service requirements.

Dedicated Bandwidth

If you want to ensure the bandwidth available for a particular EIP, you are advised to purchase dedicated bandwidth. Dedicated bandwidth can only be used for a single, specific EIP. Dedicated bandwidth is not affected by other services.

An EIP can be billed by bandwidth or by traffic:

- Bandwidth: If your services use a large amount of traffic but are stable, an EIP billed by bandwidth is recommended.
- Traffic: If your services only use a relatively small amount of traffic, an EIP billed by traffic combined with a shared data package is recommended for a more favorable price.

If your traffic is stable, the yearly/monthly billing based on the bandwidth is more cost effective.

Shared Bandwidth

When you host a large number of applications on the cloud, if each EIP uses dedicated bandwidth, a lot of bandwidths are required, which incurs high costs. If all EIPs share the same bandwidth, your network operation costs will be lowered and your system O&M as well as resource statistics will be simplified. Multiple EIPs whose billing mode is pay-per-use can be added to a shared bandwidth. You can bind EIPs to products such as ECSs, NAT gateways, and load balancers so that these products can use the shared bandwidth.

A shared bandwidth can be billed by bandwidth.

If you use a large number of EIPs and their peak hours are different, use shared bandwidth to greatly reduce costs.

6 VPC Security

6.1 Using IP Address Groups to Reduce the Number of Security Group Rules

Scenarios

An IP address group is a collection of one or more IP addresses. You can use IP address groups when configuring security group rules. If you change the IP addresses in an IP address group, the security group rules are changed accordingly. You do not need to modify the security group rules one by one.

Finance and securities enterprises have high security requirements when planning cloud networks. Access to instances is often controlled based on IP addresses. To simplify security group rule configuration and control access based on IP addresses, you can use IP address groups to manage IP address ranges and IP addresses with the same security requirements. For more information about IP address groups, see **IP Address Group Overview**.

Suppose your enterprise has an online office system deployed on the cloud. To provide services for different departments, you associate office servers with different security groups based on security levels. These servers are accessed from a large number of IP addresses that may change from time to time.

- If IP address groups are not used, you need to configure multiple rules to control access from different sources. Once the IP addresses change, you need to adjust the rules in each security group one by one. The management workload increases with the number of security groups and rules.
- If IP address groups are used, you can add the IP addresses with the same security requirements to an IP address group and add rules with source set to this IP address group. When an IP address changes, you only need to change it in the IP address group. Then, the security group rules using the IP address group change accordingly. You do not need to modify the security group rules one by one. This simplifies security group management and improves efficiency.

Solution Architecture

In this practice, the instances are associated with three security groups based on different security requirements. In addition, these instances need to be accessed by specific IP addresses over SSH port 22. To simplify management, you can use IP address groups.

1. Create an IP address group and add IP addresses that need to access the instances.
2. Add inbound rules to allow traffic from the IP address group to the instances in the three security groups.

Table 6-1 Inbound rules

Direction	Action	Type	Protocol & Port	Source
Inbound	Allow	IPv4	TCP:22	IP address group

3. Change the IP addresses in the IP address group if any IP addresses change. Then, the rules using the IP address group change accordingly.

Constraints

Security group rules using IP address groups do not take effect for the following instances:

- General computing (S1, C1, and C2 ECSs)
- Memory-optimized (M1 ECSs)
- High-performance computing (H1 ECSs)
- Disk-intensive (D1 ECSs)
- GPU-accelerated (G1 and G2 ECSs)
- Large-memory (E1, E2, and ET2 ECSs)

Resource Planning

In this practice, the IP address group and security groups must be in the same region. For details, see **Table 6-2**. The following resource details are only examples. You can modify them as required.

Table 6-2 Resource planning

Resource	Quantity	Description
IP address group	1	Create an IP address group and add IP addresses that need to access the instances. ● **Name**: **ipGroup-A** ● **Max. IP Addresses**: Set it as required. In this practice, **20** is used. ● **IP Address Version**: Set it as required. In this practice, **IPv4** is used. ● **IP Addresses**: – 11.xx.xx.64/32 – 116.xx.xx.252/30 – 113.xx.xx.0/25 – 183.xx.xx.208/28
Security group	3	Add inbound rules to allow traffic from **ipGroup-A** to the instances in the three security groups, as shown in **Table 6-3**.

Table 6-3 Inbound rules

Direction	Action	Type	Protocol & Port	Source
Inbound	Allow	IPv4	TCP:22	ipGroup-A

Procedure

Step 1 Create IP address group **ipGroup-A** and add IP addresses that need to access the instances.

For details, see **Creating an IP Address Group**.

Step 2 Add inbound rules to allow traffic from **ipGroup-A** to the instances in the three security groups.

For details, see **Adding a Security Group Rule**.

After the rules are added, traffic from 11.xx.xx.64/32, 116.xx.xx.252/30, 113.xx.xx.0/25, and 183.xx.xx.208/28 are allowed to the Linux ECSs over SSH port 22.

Step 3 Change IP addresses in the IP address group.

After security group rules are added, you can add IP addresses to **ipGroup-A**. For example, you can add 117.xx.xx.0/25 to **ipGroup-A**, and the security groups rule is applied automatically, allowing traffic from 117.xx.xx.0/25 over SSH port 22.

For details, see **Managing IP Addresses in an IP Address Group**.

----End

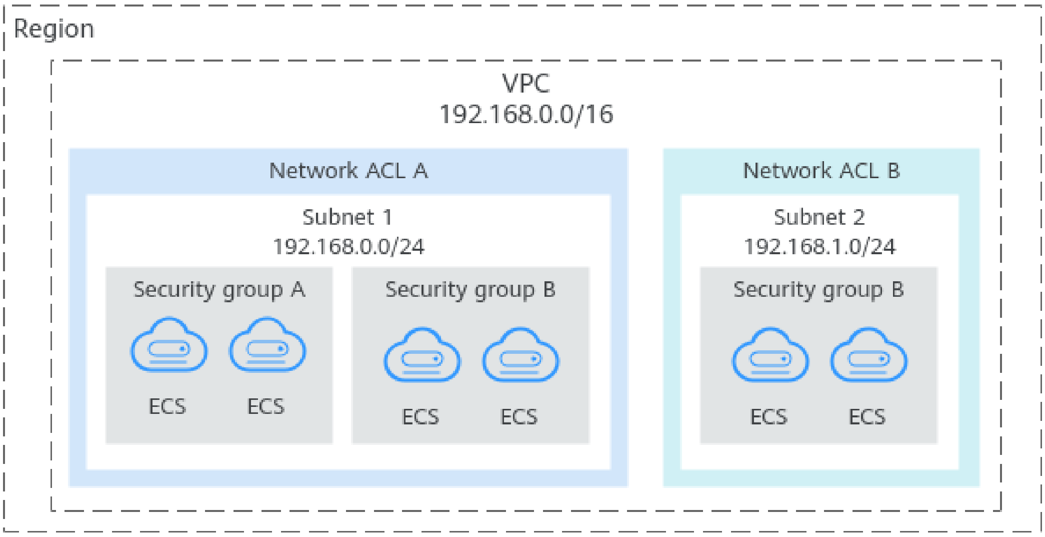
6.2 Using Security Groups and Network ACLs to Control Traffic

A VPC is your private network on the cloud. You can configure security groups and network ACL rules to ensure the security of instances, such as ECSs, databases, and containers, running in a VPC.

- A security group protects the instances in it.
- A network ACL protects associated subnets and all the resources in the subnets.

As shown in **Figure 6-1**, security groups A and B protect the network security of ECSs. Network ACLs A and B add an additional layer of defense to ECSs in subnets 1 and 2.

Figure 6-1 Security groups and network ACLs



Here are some common access control configuration examples:

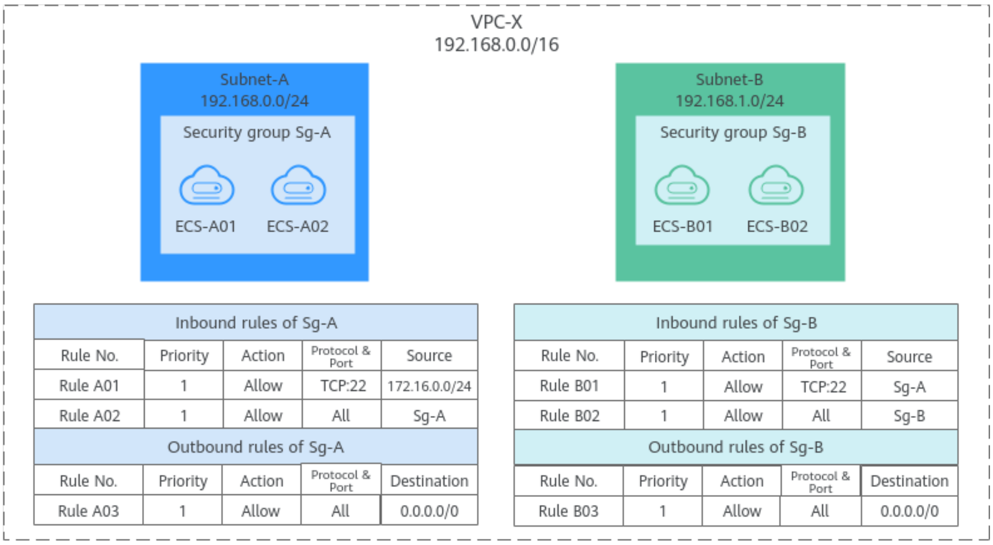
- Security group: **Allowing Traffic from Given IP Addresses or a Security Group**
- Security group: **Allowing Traffic from a Virtual IP Address**
- Security group: **Allowing Communications Between Instances in Two VPCs Connected by a VPC Peering Connection**
- Network ACL: **Controlling External Access to Instances in a Subnet**
- Network ACL: **Controlling Communications Between Instances in Different Subnets**

Allowing Traffic from Given IP Addresses or a Security Group

You can add inbound rules to allow traffic from specific IP addresses and other security groups. As you can see in **Figure 6-2**, there are two subnets (**Subnet-A** and **Subnet-B**) in **VPC-X**. ECSs in **Subnet-A** are associated with security group **Sg-A**, and ECSs in **Subnet-B** are associated with security group **Sg-B**.

- Add inbound rule A01 to **Sg-A** to allow traffic from IP addresses in **172.16.0.0/24** to access SSH port 22 on the ECSs in **Sg-A** for remotely logging in to these ECSs.
- Add inbound rule B01 to **Sg-B** to allow the ECSs in **Sg-A** to access SSH port 22 on the ECSs in **Sg-B** for remotely logging in to the ECSs in **Subnet-B**.

Figure 6-2 Allowing traffic from given IP addresses and security groups



Inbound rules of Sg-A				
Rule No.	Priority	Action	Protocol & Port	Source
Rule A01	1	Allow	TCP:22	172.16.0.0/24
Rule A02	1	Allow	All	Sg-A

Outbound rules of Sg-A				
Rule No.	Priority	Action	Protocol & Port	Destination
Rule A03	1	Allow	All	0.0.0.0/0

Inbound rules of Sg-B				
Rule No.	Priority	Action	Protocol & Port	Source
Rule B01	1	Allow	TCP:22	Sg-A
Rule B02	1	Allow	All	Sg-B

Outbound rules of Sg-B				
Rule No.	Priority	Action	Protocol & Port	Destination
Rule B03	1	Allow	All	0.0.0.0/0

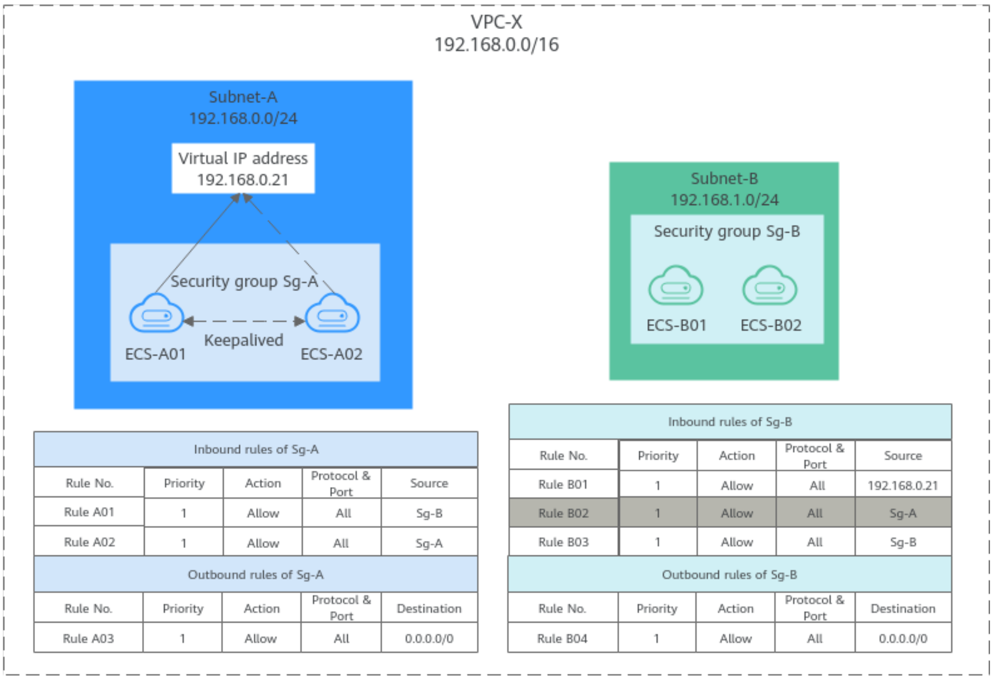
Allowing Traffic from a Virtual IP Address

You can add inbound rules to allow traffic from virtual IP addresses and other security groups. In **Figure 6-3**, there are two subnets (**Subnet-A** and **Subnet-B**) in **VPC-X**. ECSs in **Subnet-A** are associated with security group **Sg-A**, and ECSs in **Subnet-B** are associated with security group **Sg-B**.

- Add inbound rule A01 to **Sg-A** to allow the ECSs in **Sg-B** to access the ECSs in **Sg-A** using private IP addresses.
- Add inbound rule B01 to **Sg-B** to allow traffic from virtual IP address **192.168.0.21** to the ECSs in **Sg-B** using any protocol over any port. You can also set the source to the CIDR block of **Subnet-A** (192.168.0.0/24).

  Do not add rules like rule B02. This rule allows the ECSs in **Sg-A** to access the ECSs in **Sg-B** using private IP addresses but not virtual IP address **192.168.0.21**.

Figure 6-3 Allowing traffic from a virtual IP address



Inbound rules of Sg-A

Rule No.	Priority	Action	Protocol & Port	Source
Rule A01	1	Allow	All	Sg-B
Rule A02	1	Allow	All	Sg-A

Outbound rules of Sg-A

Rule No.	Priority	Action	Protocol & Port	Destination
Rule A03	1	Allow	All	0.0.0.0/0

Inbound rules of Sg-B

Rule No.	Priority	Action	Protocol & Port	Source
Rule B01	1	Allow	All	192.168.0.21
Rule B02	1	Allow	All	Sg-A
Rule B03	1	Allow	All	Sg-B

Outbound rules of Sg-B

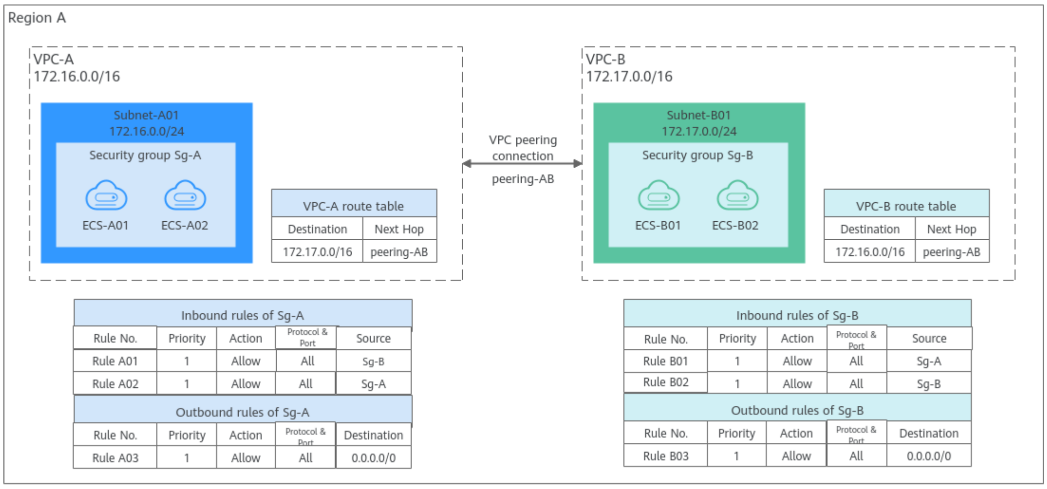
Rule No.	Priority	Action	Protocol & Port	Destination
Rule B04	1	Allow	All	0.0.0.0/0

Allowing Communications Between Instances in Two VPCs Connected by a VPC Peering Connection

In **Figure 6-4**, **VPC-A** and **VPC-B** are connected by VPC peering connection **peering-AB**. To allow ECSs in **Sg-A** and **Sg-B** to communicate with each other, you can add the following rules:

- Rule A01 with **Source** to **Sg-B** to allow ECSs in **Sg-B** to access ECSs in **Sg-A**.
- Rule B01 with **Source** to **Sg-A** to allow ECSs in **Sg-A** to access ECSs in **Sg-B**.

Figure 6-4 Allowing communications between ECSs in two VPCs connected by a VPC peering connection



VPC-A route table

Destination	Next Hop
172.17.0.0/16	peering-AB

VPC-B route table

Destination	Next Hop
172.16.0.0/16	peering-AB

Inbound rules of Sg-A

Rule No.	Priority	Action	Protocol & Port	Source
Rule A01	1	Allow	All	Sg-B
Rule A02	1	Allow	All	Sg-A

Outbound rules of Sg-A

Rule No.	Priority	Action	Protocol & Port	Destination
Rule A03	1	Allow	All	0.0.0.0/0

Inbound rules of Sg-B

Rule No.	Priority	Action	Protocol & Port	Source
Rule B01	1	Allow	All	Sg-A
Rule B02	1	Allow	All	Sg-B

Outbound rules of Sg-B

Rule No.	Priority	Action	Protocol & Port	Destination
Rule B03	1	Allow	All	0.0.0.0/0

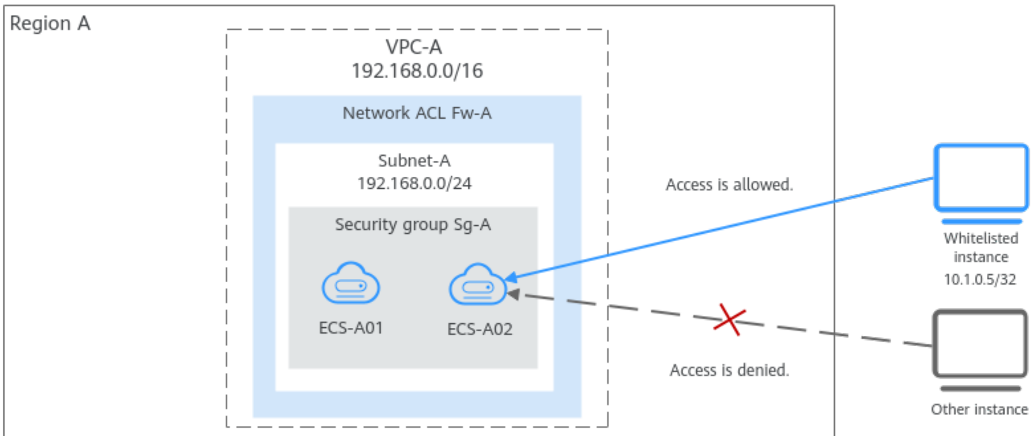
Controlling External Access to Instances in a Subnet

A network ACL controls traffic in and out of a subnet. If both security group and network ACL rules are configured, traffic matches network ACL rules first and then security group rules.

As shown in **Figure 6-5**, **ECS-A01** and **ECS-A02** in **Subnet-A** need to communicate with each other, and the instance with the IP address **10.1.0.5/32** needs to be whitelisted to allow it to remotely log in to **ECS-A01** and **ECS-A02** to perform O&M operations. The whitelisted instance can be a local PC, an instance in a different subnet of **VPC-A**, or an instance in another VPC. You need to configure network ACL and security group rules to allow the whitelisted instance to access ECSs in **VPC-A** and deny any other traffic.

- Network ACL rules:
  - Inbound rule: Custom rule **A01** allows the whitelisted instance to remotely log in to the instances in **Subnet-A** over SSH. The default rule denies any other traffic to the subnet.
  - Outbound rule: Network ACLs are stateful. The responses to inbound requests are allowed to leave the subnet. This means you do not need to additionally add outbound rules to allow such response traffic. The default rule denies any other outbound traffic.
- Security group rules:
  - Inbound rule: Rule **A01** allows the whitelisted instance to remotely log in to instances in **Subnet-A** over SSH. Rule **A02** allows instances in the security group to communicate with each other. Other traffic is denied to access the instances in security group **Sg-A**.
  - Outbound rule: Rule **A03** allows instances in **Sg-A** to access external resources.

Figure 6-5 Controlling external access to instances in a subnet



Inbound rules of Fw-A							
Rule No.	Priority	Action	Protocol	Source	Source Port Range	Destination	Destination Port Range
Custom rule A01	1	Allow	TCP	10.1.0.5/32	All	192.168.0.0/24	22
Default rule	*	Deny	All	0.0.0.0/0	All	0.0.0.0/0	All

Outbound rules of Fw-A							
Rule No.	Priority	Action	Protocol	Source	Source Port Range	Destination	Destination Port Range
Default rule	*	Deny	All	0.0.0.0/0	All	0.0.0.0/0	All

Inbound rules of Sg-A				
Rule No.	Priority	Action	Protocol & Port	Source
Rule A01	1	Allow	TCP:22	10.1.0.5/32
Rule A02	1	Allow	All	Sg-A

Outbound rules of Sg-A				
Rule No.	Priority	Action	Protocol & Port	Destination
Rule A03	1	Allow	All	0.0.0.0/0

Controlling Communications Between Instances in Different Subnets

In this example, **VPC-X** has two subnets: **Subnet-X01** and **Subnet-X02**. **ECS-01** and **ECS-02** work in **Subnet-X01**, and **ECS-03** works in **Subnet-X02**. Suppose you want to:

- Connect **ECS-02** to **ECS-03**.
- Isolate **ECS-01** from **ECS-03**.

To achieve this purpose, you need to configure security group and network ACL rules as follows:

1. Add inbound and outbound rules to **Sg-A** to ensure that the ECSs in this security group can communicate with each other.

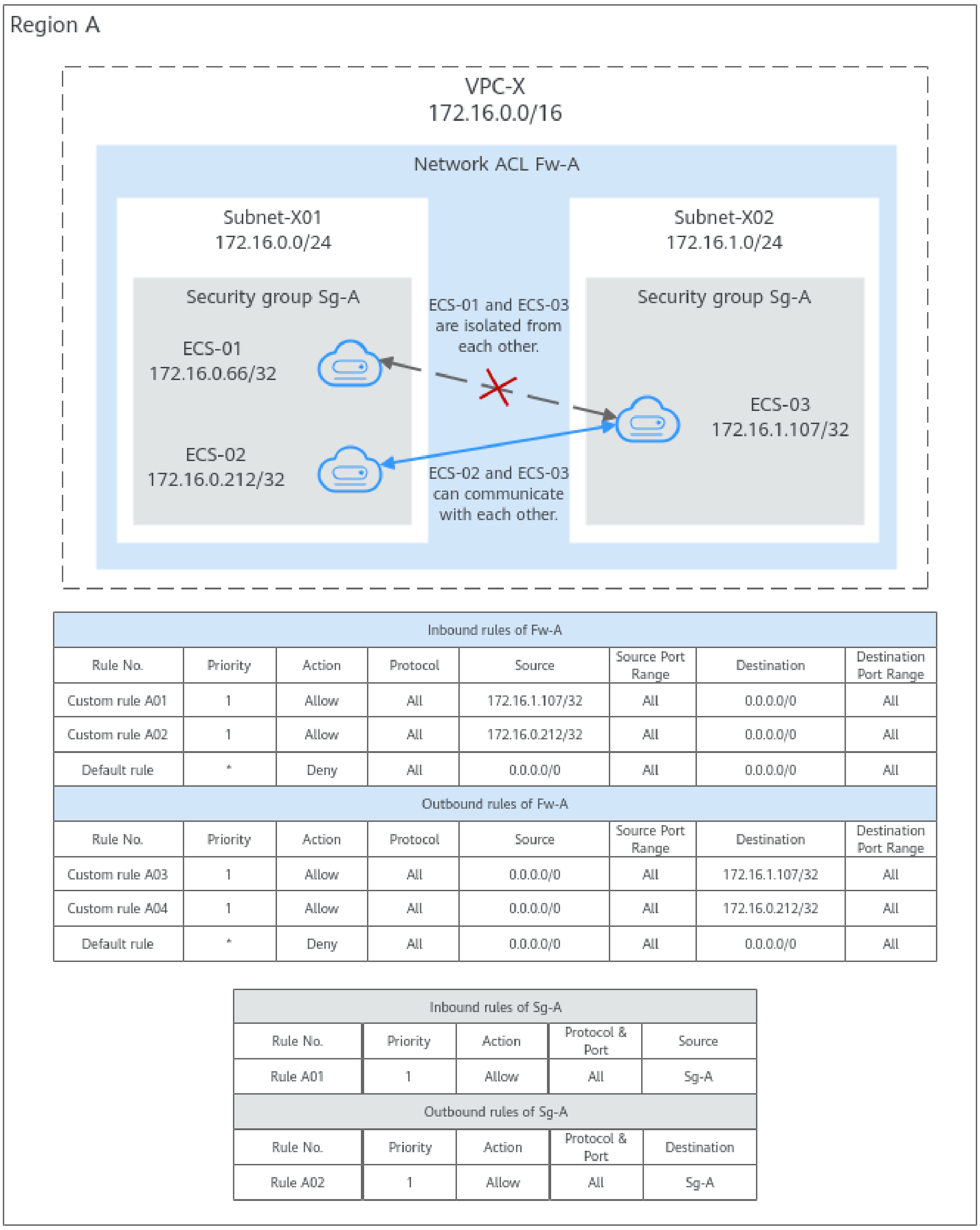
   The subnet has not been associated with a network ACL, so after the security group rules are added, both **ECS-01** and **ECS-02** can communicate with **ECS-03**.

2. Associate **Subnet-X01** and **Subnet-X02** with **Fw-A**.

   If there is only the default rule in **Fw-A**, instances in the same subnet can communicate with each other, while instances in different subnets are isolated from each other. In this case, **ECS-01** and **ECS-02** can communicate with each other, while **ECS-01** and **ECS-03** as well as **ECS-02** and **ECS-03** are isolated from each other.

3. Add custom rules to **Fw-A** to allow **ECS-02** to communicate with **ECS-03**.
   - Add custom rule A01 to allow **ECS-03** to access **Subnet-X01**.
   - Add custom rule A02 to allow **ECS-02** to access **Subnet-X02**.
   - Add custom rule A03 to allow traffic destined for **ECS-03** to leave **Subnet-X01**.
   - Add custom rule A04 to allow traffic destined for **ECS-02** to leave **Subnet-X02**.

Figure 6-6 Controlling communications between instances in different subnets



Inbound rules of Fw-A							
Rule No.	Priority	Action	Protocol	Source	Source Port Range	Destination	Destination Port Range
Custom rule A01	1	Allow	All	172.16.1.107/32	All	0.0.0.0/0	All
Custom rule A02	1	Allow	All	172.16.0.212/32	All	0.0.0.0/0	All
Default rule	*	Deny	All	0.0.0.0/0	All	0.0.0.0/0	All
Outbound rules of Fw-A							
Rule No.	Priority	Action	Protocol	Source	Source Port Range	Destination	Destination Port Range
Custom rule A03	1	Allow	All	0.0.0.0/0	All	172.16.1.107/32	All
Custom rule A04	1	Allow	All	0.0.0.0/0	All	172.16.0.212/32	All
Default rule	*	Deny	All	0.0.0.0/0	All	0.0.0.0/0	All

Inbound rules of Sg-A				
Rule No.	Priority	Action	Protocol & Port	Source
Rule A01	1	Allow	All	Sg-A
Outbound rules of Sg-A				
Rule No.	Priority	Action	Protocol & Port	Destination
Rule A02	1	Allow	All	Sg-A

6.3 Using a Third-Party Firewall to Scrub Traffic for VPCs Connected by VPC Peering Connections

Application Scenario

VPC allows you to configure and manage virtual networks. You can use security groups and network ACLs to control network access. You can also use third-party firewalls to ensure the security of cloud services.

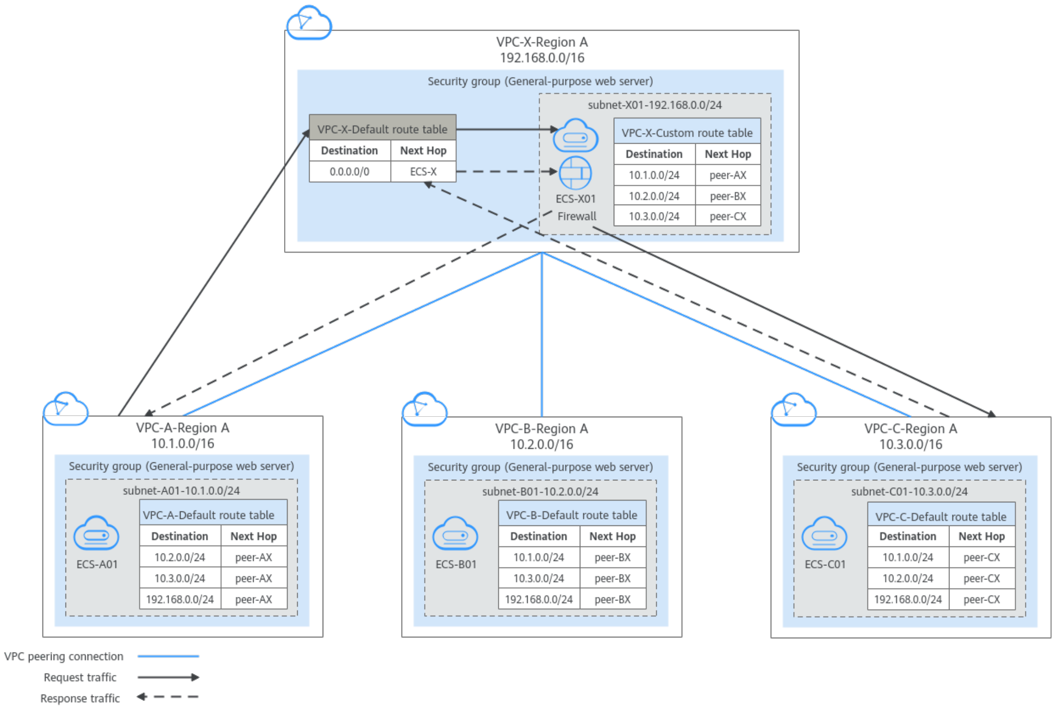
This section describes how to use a firewall to scrub traffic across VPCs that are connected using VPC peering connections.

Architecture

In this example, services are deployed in VPC-A, VPC-B, and VPC-C, and the firewall is deployed in VPC-X. These VPCs communicate with each other through VPC peering connections. The traffic across VPC-A, VPC-B, and VPC-C must flow through the firewall in VPC-X. The default route table of VPC-X directs all inbound traffic to the firewall. After being scrubbed by the firewall, the traffic is sent to a service VPC based on the custom route table.

Figure 6-7 shows how ecs-A01 accesses ecs-C01. You can view the request and response traffic paths.

Figure 6-7 Networking planning when a third-party firewall is used for scrubbing traffic across VPCs



VPC-X-Default route table

Destination	Next Hop
0.0.0.0/0	ECS-X

VPC-X-Custom route table

Destination	Next Hop
10.1.0.0/24	peer-AX
10.2.0.0/24	peer-BX
10.3.0.0/24	peer-CX

VPC-A-Default route table

Destination	Next Hop
10.2.0.0/24	peer-AX
10.3.0.0/24	peer-AX
192.168.0.0/24	peer-AX

VPC-B-Default route table

Destination	Next Hop
10.1.0.0/24	peer-BX
10.3.0.0/24	peer-BX
192.168.0.0/24	peer-BX

VPC-C-Default route table

Destination	Next Hop
10.1.0.0/24	peer-CX
10.2.0.0/24	peer-CX
192.168.0.0/24	peer-CX

Resource Planning

In this example, you need to create VPCs, ECSs, and VPC peering connections. For details about required resources, see **Table 6-4**.

NOTE

The following resource planning details are only examples for your reference. You need to plan resources based on actual service requirements.

Table 6-4 Required resources

Resource	Description
VPC	**Table 6-5** shows details about the required VPCs. In this example, there are four VPCs, including three VPCs where services are deployed and one VPC where the firewall is deployed. These VPCs are from the same region, and their subnet CIDR blocks do not overlap. ● Services are deployed in VPC-A, VPC-B, and VPC-C, and the firewall is deployed in VPC-X. These VPCs communicate with each other through VPC peering connections. ● VPC-A, VPC-B, VPC-C and VPC-X each have a subnet. ● The subnets of VPC-A, VPC-B, VPC-C are associated with their default route table. ● VPC X has a default route table and a custom route table. The subnet of VPC X is associated with the custom route table. The default route table controls the inbound traffic to VPC-X, and the custom route table controls the outbound traffic from VPC-X. **NOTICE** The subnet CIDR blocks of the VPCs that need to communicate with each other through a VPC peering connection cannot overlap. Otherwise, the VPC peering connection does not take effect. For details, see **Unsupported VPC Peering Configurations**.
ECS	**Table 6-6** shows details about the required ECSs. The four ECSs are in different VPCs. If the ECSs are in different security groups, add rules to the security groups to allow access to each other.
VPC peering connection	**Table 6-7** shows details about the required VPC peering connections. There are three VPC peering connections. ● peer-AX: connects VPC-A and VPC-X ● peer-BX: connects VPC-B and VPC-X ● peer-CX: connects VPC-C and VPC-X VPC peering connections are transitive. After routes are configured, VPC-A, VPC-B, and VPC-C can communicate with each other through VPC-X.

Table 6-5 VPC details

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Route Table	Subnet Is Used to Deploy
VPC-A	10.1.0.0/16	subnet-A01	10.1.0.0/24	Default route table	Services
VPC-B	10.2.0.0/16	subnet-B01	10.2.0.0/24	Default route table	Services
VPC-C	10.3.0.0/16	subnet-C01	10.3.0.0/24	Default route table	Services
VPC-X	192.168.0.0/16	subnet-X01	192.168.0.0/24	Custom route table	Firewall

Table 6-6 ECS details

ECS Name	VPC Name	Subnet Name	Private IP Address	Image	Security Group	ECS Is Used to Deploy
ecs-A01	VPC-A	subnet-A01	10.1.0.139	Public image: CentOS 8.2 64bit	sg-demo: General-purpose web server	Services
ecs-B01	VPC-B	subnet-B01	10.2.0.93			Services
ecs-C01	VPC-C	subnet-C01	10.3.0.220			Services
ecs-X01	VPC-X	subnet-X01	192.168.0.5			Firewall

Table 6-7 VPC peering connection details

Connection Name	Local VPC	Peer VPC
peer-AX	VPC-A	VPC-X
peer-BX	VPC-B	VPC-X
peer-CX	VPC-C	VPC-X

Route Configuration

You need to add routes to VPC route tables to allow communication between VPCs and scrub traffic through the firewall. For details, see **Table 6-8**.

NOTE

The following routes are only examples for your reference. You need to plan routes based on actual service requirements.

Table 6-8 Required route tables

Route Table	Description
Route tables of service VPCs	**Table 6-9** shows details about route tables of service VPCs. The default route tables of VPC-A, VPC-B, and VPC-C have routes with destinations set to other VPC subnets and with next hop set to VPC peering connection.
Route tables of firewall VPC	**Table 6-10** shows details about route tables of the firewall VPC-X. 1. In the default route table of VPC-X: ● If the firewall is deployed on an ECS, add a route with destination set to 0.0.0.0/0 and next hop set to ecs-X01 to direct traffic to the ECS with the firewall deployed. ● If the firewall is deployed on two ECSs and the ECSs communicate with external systems through a virtual IP address, the virtual IP address is dynamically switched to the standby ECS to continue providing services when the active ECS is faulty and cannot provide services. In this scenario, add a route with destination set to 0.0.0.0/0 and next hop set to the virtual IP address to direct traffic to the ECS with the firewall deployed. In this example, the firewall is deployed on an ECS. The traffic across VPC-A, VPC-B, and VPC-C needs to pass through VPC-X and be directed to the firewall for scrubbing. 2. In the custom route table of VPC-X, add routes with destination set to subnet CIDR blocks of service VPCs (VPC-A, VPC-B, and VPC-C) and next hop set to VPC peering connection.

Table 6-9 Details about route tables of service VPCs

VPC Name	Route Table	Destination	Next Hop Type	Next Hop	Route Type	Route Function
VPC-A	Default route table: rtb-vpc-A	10.2.0.0/24	VPC peering connection	peer-AX	Custom	• Destination: subnet-B01 in VPC-B • Connects subnet-A01 to subnet-B01
		10.3.0.0/24	VPC peering connection	peer-AX	Custom	• Destination: subnet-C01 in VPC-C • Connects subnet-A01 to subnet-C01
		192.168.0.0/24	VPC peering connection	peer-AX	Custom	• Destination: subnet-X01 in VPC-X • Connects subnet-A01 to subnet-X01
VPC-B	Default route table: rtb-vpc-B	10.1.0.0/24	VPC peering connection	peer-BX	Custom	• Destination: subnet-A01 in VPC-A • Connects subnet-A01 to subnet-B01
		10.3.0.0/24	VPC peering connection	peer-BX	Custom	• Destination: subnet-C01 in VPC-C • Connects subnet-B01 to subnet-C01
		192.168.0.0/24	VPC peering connection	peer-BX	Custom	• Destination: subnet-X01 in VPC-X • Connects subnet-B01 to subnet-X01

VPC Name	Route Table	Destination	Next Hop Type	Next Hop	Route Type	Route Function
VPC-C	Default route table: rtb-vpc-C	10.1.0.0/24	VPC peering connection	peer-CX	Custom	• Destination: subnet-A01 in VPC-A • Connects subnet-A01 to subnet-C01
		10.2.0.0/24	VPC peering connection	peer-CX	Custom	• Destination: subnet-B01 in VPC-B • Connects subnet-B01 to subnet-C01
		192.168.0.0/24	VPC peering connection	peer-CX	Custom	• Destination: subnet-X01 in VPC-X • Connects subnet-C01 to subnet-X01

Table 6-10 Details about route tables of firewall VPC

VPC Name	Route Table	Destination	Next Hop Type	Next Hop	Route Type	Route Function
VPC-X	Default route table: rtb-vpc-X	0.0.0.0/0	Server	ECS-X	Custom	- Destination: ecs-X with firewall deployed - Direct inbound traffic of VPC-X to the firewall. If your firewall is deployed on multiple ECSs and these ECSs communicate with external networks through a virtual IP address, set the next hop of the route to the virtual IP address.
	Custom route table: rtb-vpc-custom-X	10.1.0.0/24	VPC peering connection	peer-AX	Custom	- Destination: subnet-A01 in VPC-A - Connects subnet-A01 to subnet-X01
		10.2.0.0/24	VPC peering connection	peer-BX	Custom	- Destination: subnet-B01 in VPC-B - Connects subnet-B01 to subnet-X01
		10.3.0.0/24	VPC peering connection	peer-CX	Custom	- Destination: subnet-C01 in VPC-C - Connects subnet-C01 to subnet-X01

Notes and Constraints

- A VPC peering connection can only enable communication between VPCs in the same region.
- The subnet CIDR blocks of the VPCs that need to communicate with each other through a VPC peering connection cannot overlap.

- The subnet where the ECS deployed with a third-party firewall resides needs to be associated with a custom route table. Ensure that the region where your resources are located supports custom route tables.

  If **Route Tables** is displayed in the left pane of the network console, custom route tables are supported.

Procedure

Step 1 Create four VPCs and their subnets in region A.

For details, see **Creating a VPC**.

For details about VPCs and their subnets, see **Table 6-5**.

Step 2 Create a custom route table in VPC-X and associate subnet-X01 with the custom route table.

1. Create a custom route table in VPC-X.

   For details, see **Creating a Custom Route Table**.

2. Associate subnet-X01 with the custom route table created in **Step 2.1**.

   After subnet-X01 is created, it is automatically associated with the default route table of VPC-X. You need to associate the custom route table created in **Step 2.1** to subnet-X01.

   For details, see **Changing the Route Table Associated with a Subnet**.

Step 3 Create an ECS in each VPC.

For details, see **Creating an ECS**.

Step 4 Configure the NIC of ecs-X and install the third-party firewall on ecs-X.

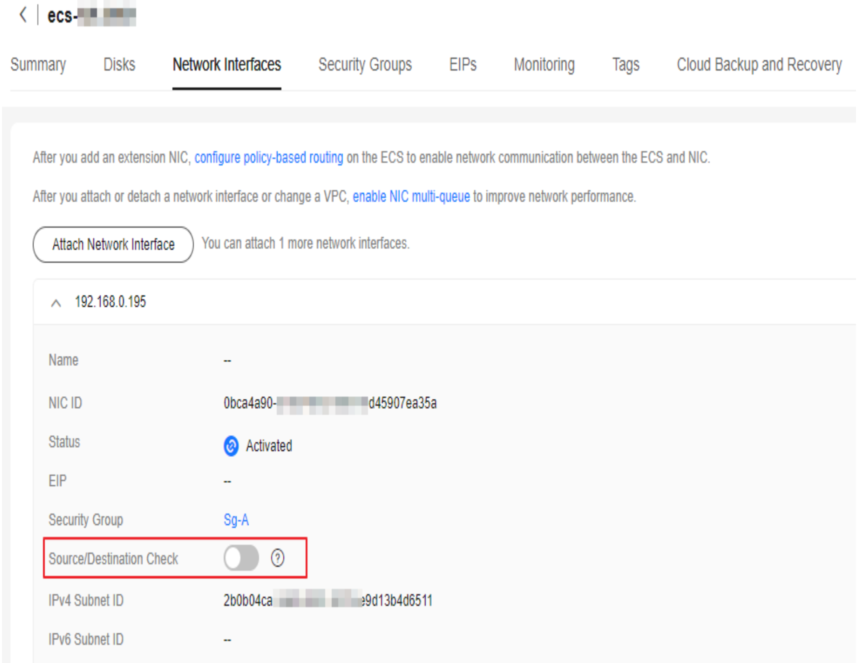
1. Disable source/destination check for the NIC of ecs-X.

   a. In the ECS list, click the name of the target ECS.

      The ECS details page is displayed.

   b. On the **Network Interfaces** tab, click ⌄ to expand the details area and check whether **Source/Destination Check** is disabled.

Figure 6-8 Disabling **Source/Destination Check**



After you add an extension NIC, configure policy-based routing on the ECS to enable network communication between the ECS and NIC.

After you attach or detach a network interface or change a VPC, enable NIC multi-queue to improve network performance.

You can attach 1 more network interfaces.

192.168.0.195

Name --

NIC ID 0bca4a90-[redacted]d45907ea35a

Status Activated

EIP --

Security Group Sg-A

Source/Destination Check ○

IPv4 Subnet ID 2b0b04ca[redacted]e9d13b4d6511

IPv6 Subnet ID --

2. Install a third-party firewall on ecs-X.

Step 5 (Optional) Configure a virtual IP address for ECSs.

You can create two ECSs in VPC-X and bind them to the same virtual IP address so that they can work in the active and standby mode. If the active ECS is faulty and cannot provide services, the virtual IP address will be dynamically switched to the standby ECS to continue providing services. Skip this step if the ECS where the firewall is deployed does not need to work in the active/standby mode.

1. Assign a virtual IP address in the VPC-X subnet.

   For details, see **Assigning a Virtual IP Address**.

2. Bind the virtual IP address to the active and standby ECSs where the firewall is deployed.

   For details, see **Binding a Virtual IP Address to an EIP or ECS**.

Step 6 Create three VPC peering connections and configure routes.

1. Create three VPC peering connections.

   – If your VPCs are in the same account, see **Creating a VPC Peering Connection with Another VPC in Your Account**.

   – If your VPCs are in different accounts, see **Creating a VPC Peering Connection with a VPC in Another Account**.

   For details about VPC peering connections, see **Table 6-7**.

2. In the default route tables of the three service VPCs, add routes with destination set to the other three VPCs and with next hop set to the VPC peering connection.

   For details, see **Adding a Custom Route**.

   In this example, add the routes planned in **Table 6-9** to the route tables of VPC-A, VPC-B, and VPC-C.

3. Add routes to the default and custom route tables of the firewall VPC.

   For details, see **Adding a Custom Route**.

   In this example, add the routes planned in **Table 6-10** to the default and custom route tables of VPC-X.

Step 7 Log in to the ECS and check whether the firewall takes effect.

Multiple methods are available for logging in to an ECS. For details, see **Logging In to an ECS**.

In this example, use VNC provided on the management console to log in to an ECS.

1. Log in to ecs-A01 and verify the network connectivity between VPC-A and VPC-B.

   ping *Private IP address of ecs-B01*

   Example command:

   ping 10.2.0.93

   If information similar to the following is displayed, the two VPCs can communicate with each other.

```
[root@ecs-A01 ~]# ping 10.2.0.93
PING 10.2.0.93 (10.2.0.93) 56(84) bytes of data.
64 bytes from 10.2.0.93: icmp_seq=1 ttl=64 time=0.849 ms
64 bytes from 10.2.0.93: icmp_seq=2 ttl=64 time=0.455 ms
64 bytes from 10.2.0.93: icmp_seq=3 ttl=64 time=0.385 ms
64 bytes from 10.2.0.93: icmp_seq=4 ttl=64 time=0.372 ms
...
--- 10.2.0.93 ping statistics ---
```

2. Keep the network connectivity between VPC-A and VPC-B in **Step 7.1** and log in to ecs-X01 to verify whether the traffic from VPC-A to VPC-B flows through ecs-X01.

3. On ecs-X01, check the traffic change on eth0.

   Run the following command at least twice consecutively to check whether the values of RX packets and TX packets change:

   ifconfig eth0

   If the packets change, the traffic flows through ecs-X01 and is scrubbed by the firewall.

```
[root@ecs-X01 ~]# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.5  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::f816:3eff:feb6:a632  prefixlen 64  scopeid 0x20<link>
        ether fa:16:3e:b6:a6:32  txqueuelen 1000  (Ethernet)
        RX packets 726222  bytes 252738526 (241.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 672597  bytes 305616882 (291.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@ecs-X01 ~]# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.5  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::f816:3eff:feb6:a632  prefixlen 64  scopeid 0x20<link>
        ether fa:16:3e:b6:a6:32  txqueuelen 1000  (Ethernet)
        RX packets 726260  bytes 252748508 (241.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 672633  bytes 305631756 (291.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

4. Repeat **Step 7.1** to **Step 7.3** to check the communication between other VPCs.

----End

6.4 Using Third-Party Firewalls to Filter Traffic When Connecting an On-premises Data Center to the Cloud

Scenarios

Your on-premises data center communicates with Huawei Cloud through Direct Connect or VPN. A third-party virtual firewall is deployed on the cloud to filter traffic.

This section describes how to use a third-party virtual firewall when connecting your on-premises data center to multiple VPCs.

Solution Advantages

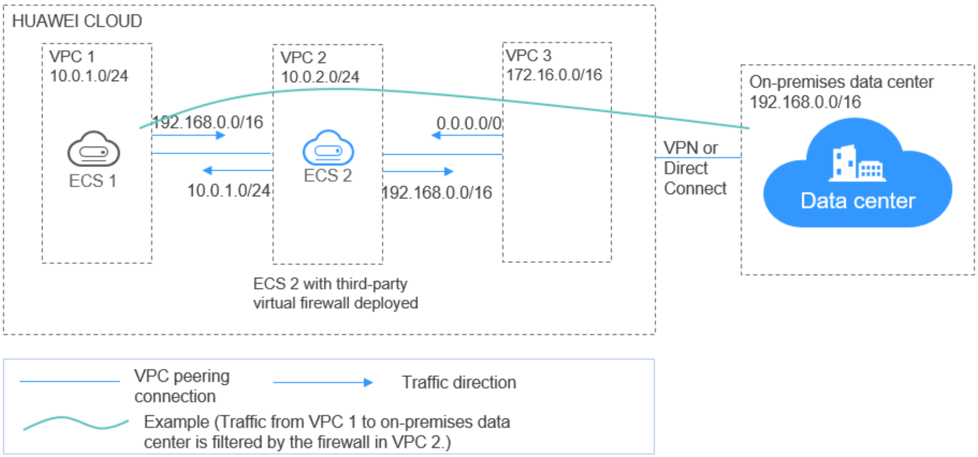
- You can use third-party firewalls.
- The traffic between the cloud and the on-premises data center will pass through the third-party virtual firewall.
- You can define security rules as required.

Typical Topology

Assume that your services are deployed in VPC 1, VPC 2, VPC 3, and your on-premises data center, and you need to use a third-party virtual firewall on the cloud. You can configure the virtual firewall on ECS 2 in VPC 2 and use VPC peering connections and configure routes to enable communication between the VPCs. In addition, you need to create a Direct Connect connection to enable communication between VPC 3 and the on-premises data center.

The deployment diagram is as follows:

Figure 6-9 Deployment diagram



Prerequisites

The subnet CIDR blocks of VPC 1, VPC 2, and VPC 3 cannot overlap with each other. Otherwise, communication through VPC peering connections will fail.

Procedure

Step 1 **Create VPCs.**

Create VPC 1, VPC 2, and VPC 3.

For details, see **Creating a VPC**.

NOTE

The CIDR blocks of VPC 1, VPC 2, and VPC 3 cannot overlap with each other. For example, the CIDR block of VPC 1 is 10.0.1.0/24, VPC 2 is 10.0.2.0/24, and VPC 3 is 172.16.0.0/16.

Step 2 **Create ECSs.**

1. Create ECS 1 and ECS 2, which belong to the VPC 1 subnet and VPC 2 subnet, respectively.

   For details, see **Creating an ECS**.

   NOTE

   The source/destination check must be disabled for the ECS 2 NIC.

2. Deploy a third-party virtual firewall on ECS 2.

Step 3 **Create VPC peering connections.**

Create VPC peering connections between VPC 1 and VPC 2, VPC 2 and VPC 3 to enable communications between them.

When creating a VPC peering connection, do not configure routes for the local and peer ends. Configure routes in **Step 7**.

For details about creating VPC peering connections, see **Creating a VPC Peering Connection with Another VPC in Your Account**.

Step 4 **Create a route table for a subnet.**

Create a custom route table and associate it with the VPC 2 subnet to control the outbound traffic.

For details, see **Creating a Custom Route Table**.

Step 5 **(Optional) Assign a virtual IP address and bind it to ECSs.**

You can create two ECSs in VPC 2 and bind them to the same virtual IP address so that they can work in the active and standby mode. If the active ECS is faulty and cannot provide services, the virtual IP address will be dynamically switched to the standby ECS to continue providing services. Skip this step if the standby ECS is not required.

1. Assign a virtual IP address in the VPC 2 subnet.

   For details, see **Assigning a Virtual IP Address**.

2. Bind the virtual IP address to ECS 2.

   For details, see **Binding a Virtual IP Address to an EIP or ECS**.

Step 6 **Create a Direct Connect connection.**

Use a Direct Connect connection to enable communication between VPC 3 and the on-premises data center. For details, see **Create a Connection**.

Step 7 **Configure routes.**

You can configure routes to forward traffic to a next hop and finally to a destination.

1. Add the following route to the default route table of VPC 1:

   Add a route to forward traffic from VPC 1 to the on-premises data center, set the destination of the route to the CIDR block of the on-premises data center, and the next hop of the route to the VPC peering connection between VPC 1 and VPC 2.

2. Add the following route to the default route table of VPC 2:

   Set the destination of the route to 0.0.0.0/0, and the next hop of the route to ECS 2.

   If there are two ECSs that use the same virtual IP address to work in the active and standby mode, the next hop should be the virtual IP address.

3. Add the following routes to the route table of VPC 2 subnet:

   a. Add a route to forward traffic from VPC 2 to VPC 1, set the destination of the route to the CIDR block of VPC 1, and the next hop of the route to the VPC peering connection between VPC 1 and VPC 2.

   b. Add a route to forward traffic from VPC 2 to the on-premises data center, set the destination of the route to the CIDR block of the on-premises data center, and the next hop of the route to the VPC peering connection between VPC 2 and VPC 3.

4. Add the following route to the default route table of VPC 3:

   Set the destination of the route to 0.0.0.0/0, and the next hop of the route to the VPC peering connection between VPC 2 and VPC 3.

   A Direct Connect connection has been created in **Step 6**. Thus, a route to the Direct Connect connection will be automatically delivered by the system.

----End

Verification

Log in to ECS 1 and then access your on-premises data center from ECS 1. Check whether ECS 2 can receive packets sent from ECS 1 to the data center. Check whether the packets pass through and are filtered by the firewall on ECS 2.

7 Deploying Containers that Can Communicate with Each Other on Huawei Cloud ECSs

Scenarios

You can deploy containers that are not provided by Huawei Cloud container services on Huawei Cloud ECSs and enable the containers on different ECSs but in the same subnet to communicate with each other.

Solution Advantages

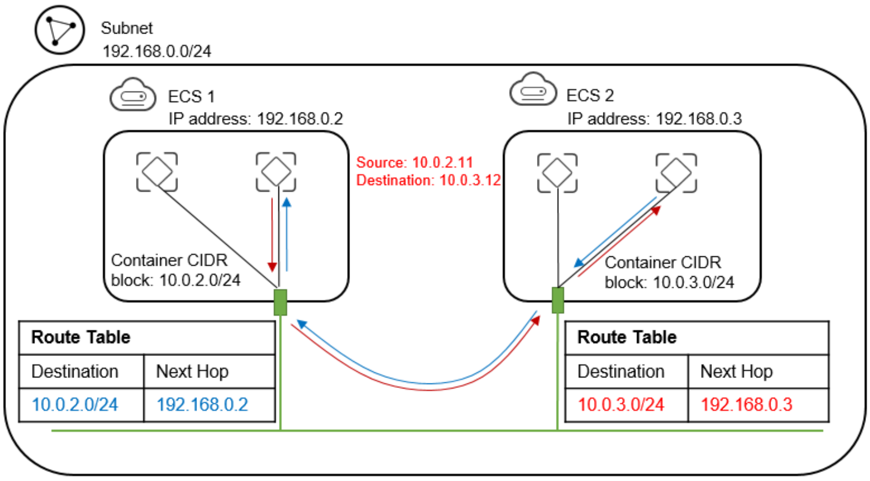
- Containers deployed on ECSs can use CIDR blocks that are not from those of the ECS VPCs, but use routes added to VPC route tables for data forwarding.
- You only need to add routes to the route tables to allow communications among containers, which is flexible and convenient.

Typical Topology

The network topology requirements are as follows:

- ECSs are in the same subnet. As shown in the following figure, the VPC subnet is 192.168.0.0/24, and the IP addresses of the ECS 1 and ECS 2 are 192.168.0.2 and 192.168.0.3, respectively.
- Containers are on CIDR blocks that are not from those of the VPC subnets that the ECSs belong to. Containers on the same ECS are on the same CIDR block, but containers on different ECSs are on different CIDR blocks. As shown in the following figure, the CIDR block of containers on ECS 1 is 10.0.2.0/24, and that on ECS 2 is 10.0.3.0/24.
- The next hop of the data packets sent to a container is the ECS where the container is deployed. As shown in the following figure, the next hop of the packets sent to CIDR block 10.0.2.0/24 is 192.168.0.2, and that of the packets sent to CIDR block 10.0.3.0/24 is 192.168.0.3.

Figure 7-1 Network topology



Route Table	
Destination	Next Hop
10.0.2.0/24	192.168.0.2

Route Table	
Destination	Next Hop
10.0.3.0/24	192.168.0.3

Procedure

Step 1 **Create VPCs.**

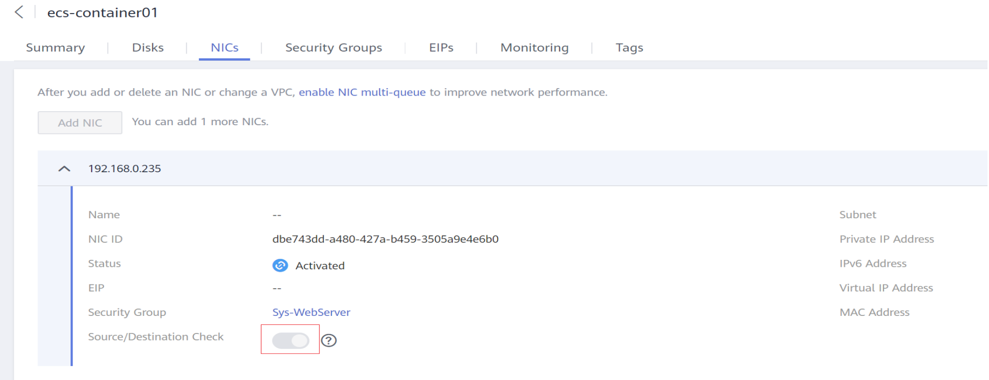
For details, see **Creating a VPC**.

Step 2 **Create ECSs.**

For details, see **Creating an ECS**.

After the ECS is created, disable source/destination check on the ECS NIC, as shown in **Figure 7-2**.

Figure 7-2 Disabling source/destination check


ecs-container01

Summary | Disks | NICs | Security Groups | EIPs | Monitoring | Tags

After you add or delete an NIC or change a VPC, enable NIC multi-queue to improve network performance.

Add NIC   You can add 1 more NICs.

192.168.0.235

Name	--	Subnet
NIC ID	dbe743dd-a480-427a-b459-3505a9e4e6b0	Private IP Address
Status	Activated	IPv6 Address
EIP	--	Virtual IP Address
Security Group	Sys-WebServer	MAC Address
Source/Destination Check		

Step 3 **Deploy containers on ECSs.**

You can use Docker CE to deploy containers. For details, see the documentation of Docker CE.

NOTE

Containers on the same ECS must be on the same CIDR block and the CIDR blocks of containers on different ECSs cannot overlap.

Step 4 **Add routes to the VPC route table.**

Set the next hop of the packets sent to CIDR block 10.0.2.0/24 to 192.168.0.2, and set the next hop of the packets sent to CIDR block 10.0.3.0/24 to 192.168.0.3.

NOTE

- By default, a VPC supports containers from a maximum of 50 different CIDR blocks. If containers from more different CIDR blocks need to be deployed in a VPC, apply for more route tables for the VPC.
- After a container is migrated to another ECS, you need to add routes to the route table of the ECS VPC.

Step 5 **Add security group rules.**

To use ping and traceroute commands to check the communications between containers, add the rules shown in **Table 7-1** to the security group of the ECSs to allow ICMP and UDP traffic.

For details, see **Adding a Security Group Rule**.

Table 7-1 Security group rules

Direction	Protocol	Port	Source
Inbound	ICMP	All	0.0.0.0/0
Inbound	UDP	All	0.0.0.0/0

----End

Verification

Use the ping command to check whether the containers deployed on two different ECSs can communicate with each other.

Run the following commands to create a network connection **my-net** on ECS 1, set the CIDR block to be used by a container on ECS 1 to 10.0.2.0/24, and create the container that uses **my-net**.

```
$ docker network create  --subnet 10.0.2.0/24 my-net
$ docker run -d --name nginx --net my-net -p 8080:80  nginx:alpine
```

Run the following commands to create a network connection and container on ECS 2, and set the CIDR block to be used by the container to 10.0.3.0/24.

```
$ docker network create  --subnet 10.0.3.0/24 my-net
$ docker run -d --name nginx --net my-net -p 8080:80  nginx:alpine
```

Run the following command to set the default policy of the FORWARD chain in the filter table of iptables on the ECS to ACCEPT.

NOTE

This operation is required because Docker sets the default policy of the FORWARD chain in the filter table of iptables to DROP for security purposes.

```
$ iptables -P FORWARD ACCEPT
```

Ping and traceroute 10.0.3.2 from 10.0.2.2. The ping and traceroute operations are successful, and the packet is tracerouted in the following sequence: 10.0.2.2 ->

10.0.2.1 -> 192.168.0.3 -> 10.0.3.2, which is consistent with the configured route forwarding rules.

```
[root@ecs1 ~]# docker exec -it nginx /bin/sh
/ # traceroute -d 10.0.3.2
traceroute to 10.0.3.2 (10.0.3.2), 30 hops max, 46 byte packets
 1  10.0.2.1 (10.0.2.1)  0.007 ms  0.004 ms  0.007 ms
 2  192.168.0.3 (192.168.0.3)  0.232 ms  0.165 ms  0.248 ms
 3  10.0.3.2 (10.0.3.2)  0.366 ms  0.308 ms  0.158 ms
/ # ping 10.0.3.2
PING 10.0.3.2 (10.0.3.2): 56 data bytes
64 bytes from 10.0.3.2: seq=0 ttl=62 time=0.570 ms
64 bytes from 10.0.3.2: seq=1 ttl=62 time=0.343 ms
64 bytes from 10.0.3.2: seq=2 ttl=62 time=0.304 ms
64 bytes from 10.0.3.2: seq=3 ttl=62 time=0.319 ms
```

8 Using a Virtual IP Address and Keepalived to Set Up a High-Availability Web Cluster

Scenarios

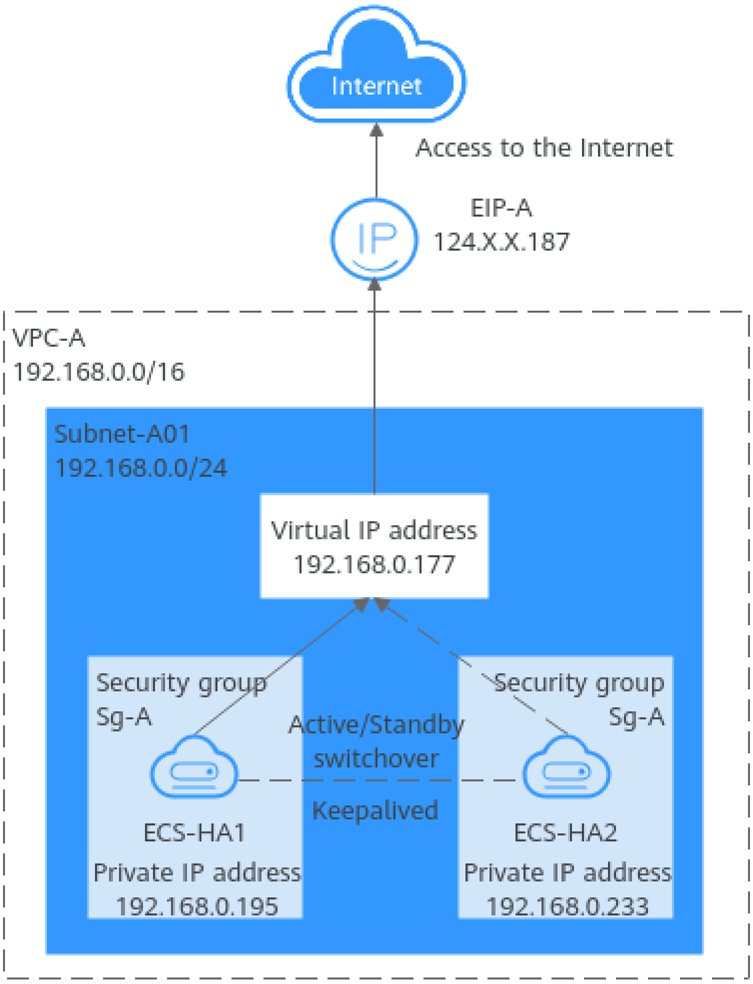
A virtual IP address is a private IP address assigned from a VPC subnet. You can use a virtual IP address and Keepalived to set up a high-availability active/standby web cluster. In such a cluster, if the active ECS goes down, the virtual IP address is bind to the standby ECS to provide services.

Architecture

Figure 8-1 shows a high-availability web cluster using Keepalived. In this architecture, virtual IP address **192.168.0.177** is bound to **ECS-HA1** and **ECS-HA2**. To allow **ECS-HA1** and **ECS-HA2** to access and be accessed from the Internet, an EIP (**EIP-A**) is bound to the virtual IP address. They work as follows:

1. **ECS-HA1** works as the active ECS and provides services accessible from the Internet using **EIP-A**. **ECS-HA2** works as the standby ECS, with no services deployed on it.
2. If **ECS-HA1** goes down, **ECS-HA2** takes over services, ensuring service continuity.

Figure 8-1 A high-availability web cluster using a virtual IP address and Keepalived



Advantages

A high-availability cluster can have one active ECS and one standby ECS or one active ECS and multiple standby ECSs. You can bind a virtual IP address to these ECSs. If the active ECS goes down, the standby ECS becomes the active ECS and continues to provide services.

Notes and Constraints

All servers of the HA cluster must be in the same subnet.

Resource Planning

In this example, the VPC, subnet, virtual IP address, EIP, and ECSs must be in the same region but can be in different AZs.

NOTE

The following resource details are only for your reference. You can modify them if needed.

Table 8-1 Resource planning

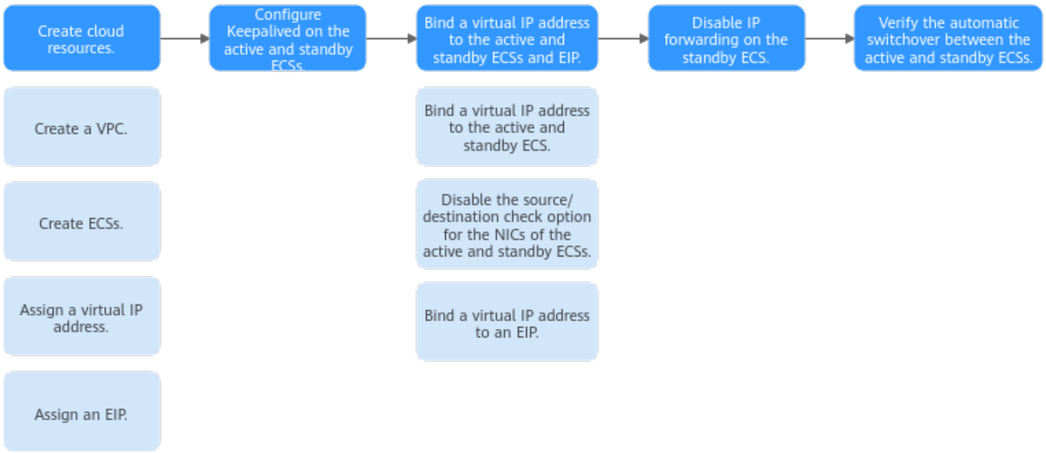
Resource Type	Quantity	Description
VPC and subnet	1	● VPC name: Set it as needed. In this example, **VPC-A** is used. ● VPC IPv4 CIDR block: Set it as needed. In this example, **192.168.0.0/16** is used. ● Subnet name: Set it as needed. In this example, **Subnet-A01** is used. ● Subnet IPv4 CIDR block: Set it as needed. In this example, **192.168.0.0/24** is used.
ECS	2	In this example, two ECSs are required for active/standby switchover. Configure the two ECSs as follows: ● **Name**: Set this parameter as needed. In this example, the two ECSs are named **ECS-HA1** and **ECS-HA2**. ● **Image**: Select an image as needed. In this example, a public image (CentOS 7.8 64bit) is used. ● **System Disk**: **General Purpose SSD** | 40 GiB ● **Data Disk**: In this example, no data disk is required. You can attach data disks based on service requirements and ensure data consistency between the two ECSs. ● Network parameters – VPC: Select a VPC. In this example, **VPC-A** is used. – Subnet: Select a subnet. In this example, **Subnet-A01** is used. ● **Security Group**: Select a security group as needed. In this example, **ECS-HA1** and **ECS-HA2** are associated with the same security group (**Sg-A**). ● Private IP address: Specify **192.168.0.195** for **ECS-HA1** and **192.168.0.233** for **ECS-HA2**.
Virtual IP address	1	Assign a virtual IP address from **Subnet-A01**. ● **Assignment Mode**: Set it as needed. In this example, **Automatic** is selected. ● Virtual IP address: **192.168.0.177** is used in this example. ● Instances: Bind **192.168.0.177** to **ECS-HA1** and **ECS-HA2**. ● EIP: Bind **192.168.0.177** to **EIP-A**.

Resource Type	Quantity	Description
EIP	1	- **Billing Mode:** Select a billing mode as needed. In this example, **Pay-per-use** is used. - **EIP Name**: Set it as needed. In this example, **EIP-A** is used. - **EIP**: The IP address is randomly assigned. In this example, **124.X.X.187** is used.

Procedure

You can follow the process in **Figure 8-2** to set up a high-availability web cluster using a virtual IP address and Keepalived

Figure 8-2 Process for setting up a high-availability web cluster


Create cloud resources.
Create a VPC.
Create ECSs.
Assign a virtual IP address.
Assign an EIP.
Configure Keepalived on the active and standby ECSs.
Bind a virtual IP address to the active and standby ECSs and EIP.
Bind a virtual IP address to the active and standby ECSs.
Disable the source/ destination check option for the NICs of the active and standby ECSs.
Bind a virtual IP address to an EIP.
Disable IP forwarding on the standby ECS.
Verify the automatic switchover between the active and standby ECSs.

Step 1: Create Cloud Resources

1. Create a VPC and subnet.

   For details, see **Creating a VPC and Subnet**.

2. Create two ECSs, one as the active ECS and the other as the standby ECS.

   For details, see **Purchasing an ECS**.

   Configure the ECSs as follows:

   – **Network**: Select **VPC-A** and **Subnet-A01** you have created.

   – **Security Group**: Create security group **Sg-A** and add inbound and outbound rules to it. Each security group comes with preset rules. You need to check and modify the rules as required.

     Add rules in **Table 8-2** to **Sg-A** and associate **Sg-A** with **ECS-HA1** and **ECS-HA2**.

Table 8-2 Sg-A rules

Direction	Action	Type	Protocol & Port	Source/ Destination	Description
Inbound	Allow	IPv4	TCP: 22	Source: 0.0.0.0/0	Allows remote logins to Linux ECSs over SSH port 22.
Inbound	Allow	IPv4	TCP: 3389	Source: 0.0.0.0/0	Allows remote logins to Windows ECSs over RDP port 3389.
Inbound	Allow	IPv4	TCP: 80	Source: 0.0.0.0/0	Allows external access to the website deployed on the ECSs over HTTP port 80.
Inbound	Allow	IPv4	All	Source: current security group (**Sg-A**)	Allows the ECSs in **Sg-A** to communicate with each other using IPv4 addresses.
Inbound	Allow	IPv6	All	Source: current security group (**Sg-A**)	Allows the ECSs in **sg-A** to communicate with each other using IPv6 addresses.
Outbound	Allow	IPv4	All	Destination: 0.0.0.0/0	Allows ECSs in **Sg-A** to access the Internet using IPv4 addresses.
Outbound	Allow	IPv6	All	Destination: ::/0	Allows ECSs in **Sg-A** to access the Internet using IPv6 addresses.

NOTICE

In this example, **Source** is set to **0.0.0.0/0**, which allows any external IP address to remotely log in to ECSs in **Sg-A**. To ensure security, you are advised to set **Source** to a specific IP address, for example, the IP address of your local PC.

If your ECSs are associated with different security groups, you need to add rules in **Table 8-3** to allow the ECSs in the two security groups to communicate with each other.

Table 8-3 Rules of security groups **Sg-A** and **Sg-B**

Security Group	Direction	Action	Type	Protocol & Port	Source/Destination	Description
Sg-A	Inbound	Allow	IPv4	All	Source: Sg-B	Allows ECSs in **Sg-B** to access those in **Sg-A** over any IPv4 protocol and port.
Sg-B	Inbound	Allow	IPv4	All	Source: Sg-A	Allows ECSs in **Sg-A** to access those in **Sg-B** over any IPv4 protocol and port.

– **EIP**: Select **Not required**.

3. Assign a virtual IP address from **Subnet-A01**.

   For details, see **Assigning a Virtual IP Address**.

4. Assign an EIP.

   For details, see **Assigning an EIP**.

Step 2: Configure Keepalived on ECS-HA1 and ECS-HA2.

1. Configure Keepalived on **ECS-HA1**.

   a. Bind **EIP-A** (**124.X.X.187**) to **ECS-HA1**.

      For details, see **Binding an EIP to an ECS**.

   b. Remotely log in to **ECS-HA1**.

      For details, see **Logging In to an ECS**.

   c. Run the following command to install the Nginx and Keepalived packages and related dependency packages:

      yum install nginx keepalived -y

      If information similar to the following is displayed, the installation is complete:

```
[root@ecs-ha1 ~]# yum install nginx keepalived -y
Loaded plugins: fastestmirror
Determining fastest mirrors
base
                              | 3.6 kB  00:00:00
epel
                              | 4.3 kB  00:00:00
extras
                              | 2.9 kB  00:00:00
updates
                                | 2.9 kB  00:00:00
(1/7): epel/x86_64/
group
         | 399 kB  00:00:00
(2/7): epel/x86_64/
updateinfo
           | 1.0 MB  00:00:00
(3/7): base/7/x86_64/
primary_db
```

```
         | 6.1 MB  00:00:00
(4/7): base/7/x86_64/
group_gz
        | 153 kB  00:00:00
(5/7): epel/x86_64/
primary_db
           | 8.7 MB  00:00:00
(6/7): extras/7/x86_64/
primary_db
       | 253 kB  00:00:00
(7/7): updates/7/x86_64/primary_db

.....
Dependency Installed:
  centos-indexhtml.noarch 0:7-9.el7.centos                    gperftools-libs.x86_64
0:2.6.1-1.el7                  lm_sensors-libs.x86_64 0:3.4.0-8.20160601gitf9185e5.el7_9.1
  net-snmp-agent-libs.x86_64 1:5.7.2-49.el7_9.4               net-snmp-libs.x86_64
1:5.7.2-49.el7_9.4               nginx-filesystem.noarch 1:1.20.1-10.el7
  openssl11-libs.x86_64 1:1.1.1k-7.el7

Complete!
```

d. Modify the Nginx configuration file.

   i. Run the following command to open the **/etc/nginx/nginx.conf** file:

      vim /etc/nginx/nginx.conf

   ii. Press **i** to enter the editing mode.

   iii. Replace the original content with the following:

```
user root;
worker_processes 1;
#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;
#pid logs/nginx.pid;
events {
      worker_connections 1024;
      }
http {
     include mime.types;
     default_type application/octet-stream;
     #log_format main '$remote_addr  - $remote_user [$time_local] "$request" '
     # '$status $body_bytes_sent  "$http_referer" '
     # '"$http_user_agent"  "$http_x_forwarded_for"';
     #access_log logs/access.log main;
     sendfile on;
     #tcp_nopush on;
     #keepalive_timeout 0;
     keepalive_timeout 65;
     #gzip on;
     server {
          listen 80;
          server_name localhost;
          #charset koi8-r;
          #access_log logs/host.access.log main;
          location / {
                    root html;
                    index index.html index.htm;
                    }
          #error_page 404  /404.html;
          # redirect server error pages to the static page /50x.html
          error_page 500 502 503 504 /50x.html;
          location =  /50x.html {
                              root html;
                              }
          }
}
```

   iv. Press **ESC** to exit and enter **:wq!** to save the configuration.

e. Modify the **index.html** file to verify whether the website is successfully accessed.

   i. Run the following command to open the **/usr/share/nginx/html/index.html** file:

      vim /usr/share/nginx/html/index.html

   ii. Press **i** to enter the editing mode.

   iii. Replace the original content with the following:

```
Welcome to ECS-HA1
```

   iv. Press **ESC** to exit and enter **:wq!** to save the configuration.

f. Run the following commands to set the automatic startup of Nginx upon ECS startup:

   systemctl enable nginx

   systemctl start nginx.service

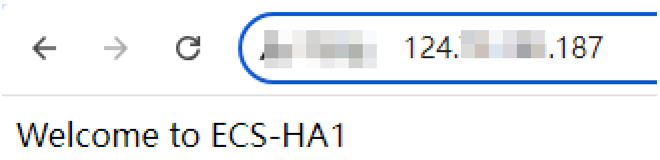
   Information similar to the following is displayed:

```
[root@ecs-ha1 ~]# systemctl enable nginx
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/
systemd/system/nginx.service.
[root@ecs-ha1 ~]# systemctl start nginx.service
```

g. Open a browser, enter the EIP address (**124.X.X.187**), and press **Enter** to verify the access to a single Nginx node.

   If the web page shown in the following figure is displayed, Nginx is successfully configured for **ECS-HA1**.

Figure 8-3 ECS-HA1 accessed



[redacted] 124.[redacted].187

Welcome to ECS-HA1

h. Modify the Keepalived configuration file.

   i. Run the following command to open the **/etc/keepalived/keepalived.conf** file:

      vim /etc/keepalived/keepalived.conf

   ii. Press **i** to enter the editing mode.

   iii. Replace the IP parameters in the configuration file as follows:

      ○ **mcast_src_ip** and **unicast_src_ip**: Change their values to the private IP address of an ECS. In this example, private IP address **192.168.0.195** of **ECS-HA1** is used.

      ○ **virtual_ipaddress**: Change the value to a virtual IP address. In this example, **192.168.0.177** is used.

```
! Configuration File for keepalived
global_defs {
router_id master-node
}
vrrp_script chk_http_port {
        script  "/etc/keepalived/chk_nginx.sh"
        interval 2
        weight -5
        fall 2
        rise 1
      }
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    mcast_src_ip 192.168.0.195
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
            auth_type PASS
            auth_pass 1111
            }
    unicast_src_ip 192.168.0.195
    virtual_ipaddress {
                192.168.0.177
                }
track_script {
    chk_http_port
    }
}
```

iv. Press **ESC** to exit and enter **:wq!** to save the configuration.

i. Configure the Nginx monitoring script.

   i. Run the following command to open the **/etc/keepalived/chk_nginx.sh** file:

   vim /etc/keepalived/chk_nginx.sh

   ii. Press **i** to enter the editing mode.

   iii. Replace the original content with the following:

```
#!/bin/bash
counter=$(ps -C nginx --no-heading|wc -l)
if [ "${counter}" = "0"  ]; then
    systemctl start nginx.service
    sleep 2
    counter=$(ps -C nginx  --no-heading|wc -l)
    if [ "${counter}" =  "0" ]; then
        systemctl stop keepalived.service
    fi
fi
```

   iv. Press **ESC** to exit and enter **:wq!** to save the configuration.

j. Run the following command to assign execute permissions to the **chk_nginx.sh** file:

   chmod +x /etc/keepalived/chk_nginx.sh

k. Run the following commands to set the automatic startup of Keepalived upon ECS startup:

   systemctl enable keepalived

   systemctl start keepalived.service

l. Unbind **EIP-A** from **ECS-HA1**.

   For details, see **Unbinding an EIP**.

2. Configure Keepalived on **ECS-HA2**.

   a. Bind **EIP-A** (**124.X.X.187**) to **ECS-HA2**.

   For details, see **Binding an EIP to an ECS**.

   b. Remotely log in to **ECS-HA2**.

   For details, see **Logging In to an ECS**.

   c. Run the following command to install the Nginx and Keepalived packages and related dependency packages:

yum install nginx keepalived -y

If information similar to the following is displayed, the installation is complete:

```
[root@ecs-ha2 ~]# yum install nginx keepalived -y
Loaded plugins: fastestmirror
Determining fastest mirrors
base
                          | 3.6 kB  00:00:00
epel
                          | 4.3 kB  00:00:00
extras
                          | 2.9 kB  00:00:00
updates
                            | 2.9 kB  00:00:00
(1/7): epel/x86_64/
group
        | 399 kB  00:00:00
(2/7): epel/x86_64/
updateinfo
          | 1.0 MB  00:00:00
(3/7): base/7/x86_64/
primary_db
         | 6.1 MB  00:00:00
(4/7): base/7/x86_64/
group_gz
        | 153 kB  00:00:00
(5/7): epel/x86_64/
primary_db
           | 8.7 MB  00:00:00
(6/7): extras/7/x86_64/
primary_db
       | 253 kB  00:00:00
(7/7): updates/7/x86_64/primary_db

.....
Dependency Installed:
  centos-indexhtml.noarch 0:7-9.el7.centos                    gperftools-libs.x86_64
0:2.6.1-1.el7                  lm_sensors-libs.x86_64 0:3.4.0-8.20160601gitf9185e5.el7_9.1
  net-snmp-agent-libs.x86_64 1:5.7.2-49.el7_9.4               net-snmp-libs.x86_64
1:5.7.2-49.el7_9.4               nginx-filesystem.noarch 1:1.20.1-10.el7
  openssl11-libs.x86_64 1:1.1.1k-7.el7

Complete!
```

d. Modify the Nginx configuration file.

   i. Run the following command to open the **/etc/nginx/nginx.conf** file:

      vim /etc/nginx/nginx.conf

   ii. Press **i** to enter the editing mode.

   iii. Replace the original content with the following:

```
user root;
worker_processes 1;
#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;
#pid logs/nginx.pid;
events {
      worker_connections 1024;
      }
http {
     include mime.types;
     default_type application/octet-stream;
     #log_format main '$remote_addr  - $remote_user [$time_local] "$request" '
     # '$status $body_bytes_sent  "$http_referer" '
     # '"$http_user_agent"  "$http_x_forwarded_for"';
     #access_log logs/access.log main;
     sendfile on;
```

```
    #tcp_nopush on;
    #keepalive_timeout 0;
    keepalive_timeout 65;
    #gzip on;
    server {
        listen 80;
        server_name localhost;
        #charset koi8-r;
        #access_log logs/host.access.log main;
        location / {
                root html;
                index index.html index.htm;
                }
        #error_page 404  /404.html;
        # redirect server error pages to the static page /50x.html
        error_page 500 502 503 504 /50x.html;
        location =  /50x.html {
                        root html;
                        }
        }
}
```

iv. Press **ESC** to exit and enter **:wq!** to save the configuration.

e. Modify the **index.html** file to verify whether the website is successfully accessed.

i. Run the following command to open the **/usr/share/nginx/html/index.html** file:

vim /usr/share/nginx/html/index.html

ii. Press **i** to enter the editing mode.

iii. Replace the original content with the following:

```
Welcome to ECS-HA2
```

iv. Press **ESC** to exit and enter **:wq!** to save the configuration.

f. Run the following commands to set the automatic startup of Nginx upon ECS startup:

systemctl enable nginx

systemctl start nginx.service

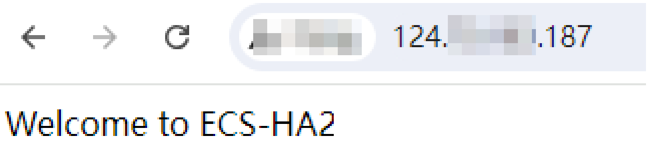
Information similar to the following is displayed:

```
[root@ecs-ha2 ~]# systemctl enable nginx
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/
systemd/system/nginx.service.
[root@ecs-ha2 ~]# systemctl start nginx.service
```

g. Open a browser, enter the EIP address (**124.X.X.187**), and press **Enter** to verify the access to a single Nginx node.

If the web page shown in the following figure is displayed, Nginx is successfully configured for **ECS-HA2**.

Figure 8-4 ECS-HA2 accessed



[redacted] 124.[redacted].187

Welcome to ECS-HA2

h. Modify the Keepalived configuration file.

   i. Run the following command to open the **/etc/keepalived/keepalived.conf** file:

      vim /etc/keepalived/keepalived.conf

   ii. Press **i** to enter the editing mode.

   iii. Replace the IP parameters in the configuration file as follows:

      ○ **mcast_src_ip** and **unicast_src_ip**: Change their values to the private IP address of an ECS. In this example, private IP address of **ECS-HA2** (**192.168.0.233**) is used.

      ○ **virtual_ipaddress**: Change the value to a virtual IP address. In this example, **192.168.0.177** is used.

```
! Configuration File for keepalived
global_defs {
router_id master-node
}
vrrp_script chk_http_port {
        script  "/etc/keepalived/chk_nginx.sh"
        interval 2
        weight -5
        fall 2
        rise 1
      }
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    mcast_src_ip 192.168.0.233
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
            auth_type PASS
            auth_pass 1111
            }
    unicast_src_ip 192.168.0.233
    virtual_ipaddress {
                192.168.0.177
                }
track_script {
    chk_http_port
    }
}
```

   iv. Press **ESC** to exit and enter **:wq!** to save the configuration.

i. Configure the Nginx monitoring script.

   i. Run the following command to open the **/etc/keepalived/chk_nginx.sh** file:

      vim /etc/keepalived/chk_nginx.sh

   ii. Press **i** to enter the editing mode.

   iii. Replace the original content with the following:

```
#!/bin/bash
counter=$(ps -C nginx --no-heading|wc -l)
if [ "${counter}" = "0"  ]; then
    systemctl start nginx.service
    sleep 2
    counter=$(ps -C nginx  --no-heading|wc -l)
    if [ "${counter}" =  "0" ]; then
        systemctl stop keepalived.service
    fi
fi
```

iv. Press **ESC** to exit and enter **:wq!** to save the configuration.

j. Run the following command to assign execute permissions to the **chk_nginx.sh** file:

chmod +x /etc/keepalived/chk_nginx.sh

k. Run the following commands to set the automatic startup of Keepalived upon ECS startup:

systemctl enable keepalived

systemctl start keepalived.service

l. Unbind **EIP-A** from **ECS-HA2**.

For details, see **Unbinding an EIP**.

Step 3: Bind the Virtual IP Address to the Active and Standby ECSs and EIP

1. Bind virtual IP address **192.168.0.177** to **ECS-HA1** and **ECS-HA2**.

   For details, see **Binding a Virtual IP Address to an Instance or EIP**.

2. Disable **Source/Destination Check** for the network interfaces of the active and standby ECSs.

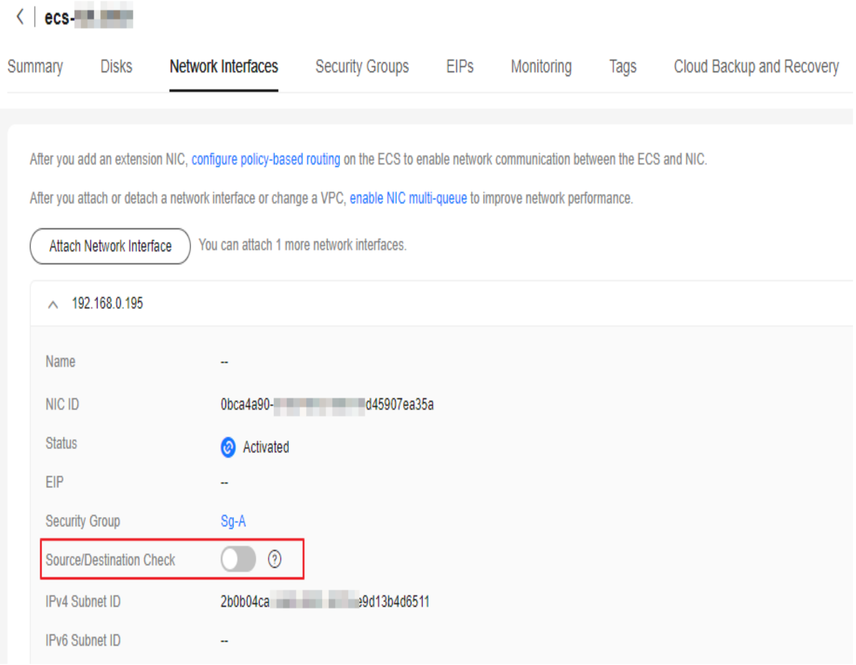
   When you bind a virtual IP address to an ECS, **Source/Destination Check** is disabled by default. You can perform the following operations to check whether the function is disabled. If the function is not disabled, disable it.

   a. In the ECS list, click the name of the target ECS.

   The ECS details page is displayed.

   b. On the **Network Interfaces** tab, click ⌄ to expand the details area and check whether **Source/Destination Check** is disabled.

Figure 8-5 Disabling **Source/Destination Check**



ecs-[redacted]

After you add an extension NIC, configure policy-based routing on the ECS to enable network communication between the ECS and NIC.

After you attach or detach a network interface or change a VPC, enable NIC multi-queue to improve network performance.

You can attach 1 more network interfaces.

192.168.0.195

Name --

NIC ID 0bca4a90-[redacted]d45907ea35a

Status Activated

EIP --

Security Group Sg-A

Source/Destination Check ○

IPv4 Subnet ID 2b0b04ca[redacted]9d13b4d6511

IPv6 Subnet ID --

3. Bind virtual IP address **192.168.0.177** to **EIP-A**.

   For details, see **Binding a Virtual IP Address to an Instance or EIP**.

Step 4: Disable IP Forwarding on the Standby ECS

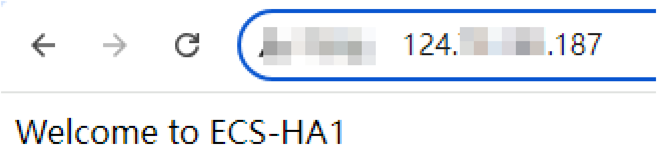
If a virtual IP address is bound to active/standby ECSs, you need to disable IP forwarding on the standby ECS. If an active/standby ECS switchover happens, ensure that IP forwarding of the new standby ECS is also disabled.

To make sure you do not miss any settings, it is better to disable IP forwarding on both of active and standby ECSs.

1. Open a browser, enter the EIP address (**124.X.X.187**), and press **Enter** to access the active ECS.

   If the following page is displayed, the **ECS-HA1** is used as the active ECS.

   Figure 8-6 The active ECS accessed

   [redacted] 124.[redacted].187

   Welcome to ECS-HA1

2. Remotely log in to the standby ECS (**ECS-HA2** in this example).

   For details, see **Logging In to an ECS**.

3. Disable IP forwarding by following the operations in **Table 8-4**. In this example, the ECS runs the Linux OS.

Table 8-4 Disabling IP forwarding

OS	Operations
Linux	1. Run the following command to switch to user **root**: **su root** 2. Run the following command to check whether IP forwarding is enabled: **cat /proc/sys/net/ipv4/ip_forward** In the command output, **1** indicates that IP forwarding is enabled, and **0** indicates that IP forwarding is disabled. The default value is **0**. ● If **0** is displayed, no further action is required. ● If **1** is displayed, go to the next step. 3. Use either of the following methods to modify the configuration file: Method 1 a. Run the following command to open the **/etc/sysctl.conf** file: **vim /etc/sysctl.conf** b. Press **i** to enter the editing mode. c. Set **net.ipv4.ip_forward** to **0**. d. Press **ESC** to exit and enter **:wq!** to save the configuration. Method 2 Run the **sed** command. An example command is as follows: **sed -i '/net.ipv4.ip_forward/s/1/0/g' /etc/sysctl.conf** 4. Run the following command to apply the modification: **sysctl -p /etc/sysctl.conf**
Windows	1. In the search box, enter **cmd** to open the **command prompt** window, and run the following command: **ipconfig/all** ● In the command output, if the value of **IP Routing Enabled** is **No**, IP forwarding is disabled. ● If **IP Routing Enabled** is **Yes**, IP forwarding is not disabled. Go to the next step. 2. Enter **regedit** in the search box to open the registry editor. 3. Set the value of **IPEnableRouter** under **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters** to **0**. ● If the value is set to **0**, IP forwarding will be disabled. ● If the value is set to **1**, IP forwarding will be enabled.

Step 5: Verify the Automatic Switchover Between the Active and Standby ECSs

1. Restart the active and standby ECSs.
   a. Remotely log in to **ECS-HA1**.

      For details, see **Logging In to an ECS**.

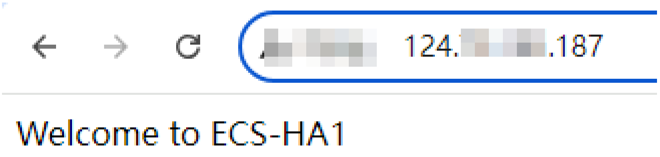
   b. Run the following command to restart **ECS-HA1**:

      reboot

   c. Repeat **1.a** to **1.b** to restart **ECS-HA2**.
2. Check whether the website on the active ECS can be accessed.
   a. Open a browser, enter the EIP address (**124.X.X.187**), and press **Enter**.

      If the following page is displayed, **ECS-HA1** is used as the active ECS and the website can be accessed.

      Figure 8-7 ECS-HA1 accessed



124.[redacted].187

Welcome to ECS-HA1

   b. Remotely log in to **ECS-HA1** and run the following command to check whether the virtual IP address is bound to the eth0 NIC of **ECS-HA1**:

      ip addr show

      If information similar to the following is displayed, the virtual IP address (**192.168.0.177**) has been bound to the eth0 NIC of **ECS-HA1**, and this ECS is the active one.

```
[root@ecs-ha1 ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    link/ether fa:16:3e:fe:56:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.195/24 brd 192.168.0.255 scope global noprefixroute dynamic eth0
       valid_lft 107898685sec preferred_lft 107898685sec
    inet 192.168.0.177/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fefe:5619/64 scope link
       valid_lft forever preferred_lft forever
```

   c. Run the following command to disable Keepalived on **ECS-HA1**:

      systemctl stop keepalived.service
3. Check whether **ECS-HA2** becomes the active ECS.
   a. Remotely log in to **ECS-HA2** and run the following command to check whether the virtual IP address is bound to the eth0 NIC of **ECS-HA2**:

      ip addr show

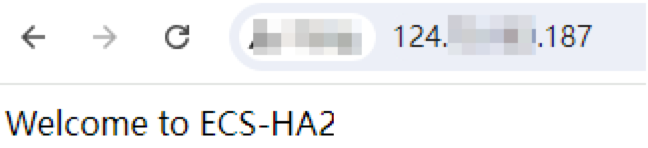
If information similar to the following is displayed, the virtual IP address (**192.168.0.177**) has been bound to the eth0 NIC of **ECS-HA2**, and this ECS becomes the active one.

```
[root@ecs-ha2 ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    link/ether fa:16:3e:fe:56:3f brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.233/24 brd 192.168.0.255 scope global noprefixroute dynamic eth0
       valid_lft 107898091sec preferred_lft 107898091sec
    inet 192.168.0.177/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fefe:563f/64 scope link
       valid_lft forever preferred_lft forever
```

b. Open a browser, enter the EIP address (**124.X.X.187**), and press **Enter** to check whether the website on the active ECS (**ECS-HA2**) can be accessed.

If the following page is displayed, **ECS-HA2** is used as the active ECS and the website can be accessed.

Figure 8-8 ECS-HA2 accessed



124.[redacted].187

Welcome to ECS-HA2

9 Configuring Policy-based Routes for an ECS with Multiple NICs

9.1 Overview

Background

If an ECS has multiple NICs, the primary NIC can communicate with external networks by default, but the extension NICs cannot. To enable extension NICs to communicate with external works either, you need to configure policy-based routes for these NICs.

Scenarios

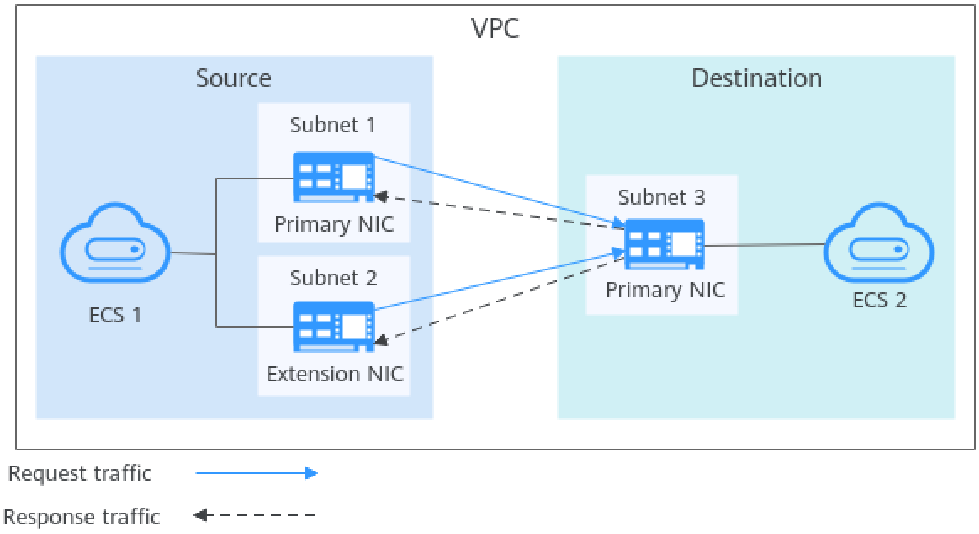
This example describes how to configure policy-based routes for an ECS with two NICs. **Figure 9-1** shows the networking. The details are as follows:

- The primary and extension NICs on the source ECS are in different subnets of the same VPC.
- The source and destination ECSs are in different subnets of the same VPC and the two ECSs can communicate with each other through primary NICs without configuring policy-based routes.
- After policy-based routes are configured for the two NICs of the source ECS, both the primary and extension NICs can communicate with the destination ECS.

NOTICE

You can select a destination IP address based on service requirements. Before configuring policy-based routes, ensure that the source ECS can use its primary NIC to communicate with the destination ECS.

Figure 9-1 Dual-NIC ECS networking



9.2 Collecting ECS Network Information

Scenarios

Before configuring policy-based routes for a multi-NIC ECS, you need to collect network information about the ECS.

- **Table 9-1** lists the information to be collected for a Linux ECS using IPv4.

Table 9-1 Linux ECS using IPv4

ECS	Primary NIC	Extension NIC	How to Obtain
Source	● NIC address: 10.0.0.115 ● Subnet: 10.0.0.0/24 ● Subnet gateway: 10.0.0.1	● NIC address: 10.0.1.183 ● Subnet: 10.0.1.0/24 ● Subnet gateway: 10.0.1.1	● **Obtaining ECS NIC Addresses** ● **Obtaining Subnet CIDR Blocks and Gateway Addresses**
Destination	NIC address: 10.0.2.12	N/A	

- **Table 9-2** lists the information to be collected for a Windows ECS using IPv4.

Table 9-2 Windows ECS using IPv4

ECS	Primary NIC	Extension NIC	How to Obtain
Source	- NIC address: 10.0.0.59 - Subnet gateway: 10.0.0.1	- NIC address: 10.0.1.104 - Subnet gateway: 10.0.1.1	- **Obtaining ECS NIC Addresses** - **Obtaining Subnet CIDR Blocks and Gateway Addresses**
Destination	NIC address: 10.0.2.12	N/A	

Obtaining ECS NIC Addresses

1. Log in to the management console.
2. Click 📍 in the upper left corner and select the desired region and project.
3. Click **Service List** and choose **Compute** > **Elastic Cloud Server**.
4. In the ECS list, click the target ECS name.

   The **Summary** tab page of the ECS is displayed.
5. In the **NICs** area, view the IP addresses of the primary and extension NICs.

Obtaining Subnet CIDR Blocks and Gateway Addresses

1. Log in to the management console.
2. Click 📍 in the upper left corner and select the desired region and project.
3. Click **Service List** and choose **Compute** > **Elastic Cloud Server**.
4. In the ECS list, click the target ECS name.

   The **Summary** tab page of the ECS is displayed.
5. In the **ECS Information** area, click the VPC hyperlink.

   The **Virtual Private Cloud** page is displayed.
6. Locate the target VPC and click the number in the **Subnets** column.

   The **Subnets** page is displayed.
7. In the subnet list, view the CIDR blocks of the subnets.
8. In the subnet list, click the subnet name.

   The **Summary** page is displayed.
9. Click the **IP Addresses** tab and view the gateway addresses of the subnet.

9.3 Configuring Policy-based Routes for a Linux ECS with Multiple NICs

Scenarios

This section describes how to configure policy-based routes for a dual-NIC ECS running CentOS 8.0 (64-bit).

For details about the background knowledge and networking of dual-NIC ECSs, see **Overview**.

Procedure (Linux ECS Using IPv4)

1. Collect the ECS network information required for configuring policy-based routes.

   For details, see **Collecting ECS Network Information**.

2. Log in to an ECS.

3. Check whether the source ECS can use its primary NIC to communicate with the destination ECS:

   ping -I *IP address of the primary NIC on the source ECS IP address of the destination ECS*

   In this example, run the following command:

   ping -I 10.0.0.115 10.0.2.12

   If information similar to the following is displayed, the source ECS can use its primary NIC to communicate with the destination ECS.

   ```
   [root@ecs-resource ~]# ping -I 10.0.0.115 10.0.2.12
   PING 10.0.2.12 (10.0.2.12) from 10.0.0.115 : 56(84) bytes of data.
   64 bytes from 10.0.2.12: icmp_seq=1 ttl=64 time=0.775 ms
   64 bytes from 10.0.2.12: icmp_seq=2 ttl=64 time=0.268 ms
   64 bytes from 10.0.2.12: icmp_seq=3 ttl=64 time=0.220 ms
   64 bytes from 10.0.2.12: icmp_seq=4 ttl=64 time=0.167 ms
   ^C
   --- 10.0.2.12 ping statistics ---
   ```

   NOTE

   Before configuring policy-based routes, ensure that the source ECS can use its primary NIC to communicate with the destination ECS.

4. Query the NIC names of the ECS:

   ifconfig

   Search for the NIC name based on the NIC address.

   – 10.0.0.115 is the IP address of the primary NIC, and the NIC name is eth0.

   – 10.0.1.183 is the IP address of the extension NIC, and the NIC name is eth1.

   ```
   [root@ecs-resource ~]# ifconfig
   eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
           inet 10.0.0.115  netmask 255.255.255.0  broadcast 10.0.0.255
           inet6 fe80::f816:3eff:fe92:6e0e  prefixlen 64  scopeid 0x20<link>
           ether fa:16:3e:92:6e:0e  txqueuelen 1000  (Ethernet)
           RX packets 432288  bytes 135762012 (129.4 MiB)
           RX errors 0  dropped 0  overruns 0  frame 1655
   ```

```
        TX packets 423744  bytes 106716932 (101.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.1.183  netmask 255.255.255.0  broadcast 10.0.1.255
        inet6 fe80::f816:3eff:febf:5818  prefixlen 64  scopeid 0x20<link>
        ether fa:16:3e:bf:58:18  txqueuelen 1000  (Ethernet)
        RX packets 9028  bytes 536972 (524.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 1915
        TX packets 6290  bytes 272473 (266.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5. Configure temporary routes for the ECS.

   NOTICE

   Temporary routes take effect immediately after being configured and will be lost after the ECS is restarted. To prevent network interruptions after the ECS is restarted, perform **6** after this step to configure persistent routes.

   a. Configure policy-based routes for both the primary and extension NICs:

      - Primary NIC

        ip route add default via *Subnet gateway* **dev** *NIC name* **table** *Route table name*

        ip route add *Subnet CIDR block* **dev** *NIC name* **table** *Route table name*

        ip rule add from *NIC address* **table** *Route table name*

      - Extension NIC

        ip route add default via *Subnet gateway* **dev** *NIC name* **table** *Route table name*

        ip route add *Subnet CIDR block* **dev** *NIC name* **table** *Route table name*

        ip rule add from *NIC address* **table** *Route table name*

      Configure the parameters as follows:

      - NIC name: Enter the name obtained in **4**.
      - Route table name: Customize a route table name using a number.
      - Other network information: Enter the IP addresses collected in **1**.

      In this example, run the following commands:

      - Primary NIC

        ip route add default via 10.0.0.1 dev eth0 table 10

        ip route add 10.0.0.0/24 dev eth0 table 10

        ip rule add from 10.0.0.115 table 10

      - Extension NIC

        ip route add default via 10.0.1.1 dev eth1 table 20

        ip route add 10.0.1.0/24 dev eth1 table 20

ip rule add from 10.0.1.183 table 20

NOTE

If the ECS has multiple NICs, configure policy-based routes for all NICs one by one.

b. Check whether the policy-based routes are successfully added.

ip rule

ip route show table *Route table name of the primary NIC*

ip route show table *Route table name of the extension NIC*

The route table name is customized in **5.a**.

In this example, run the following commands:

ip rule

ip route show table 10

ip route show table 20

If information similar to the following is displayed, the policy-based routes have been added.

```
[root@ecs-resource ~]# ip rule
0:      from all lookup local
32764:  from 10.0.1.183 lookup 20
32765:  from 10.0.0.115 lookup 10
32766:  from all lookup main
32767:  from all lookup default
[root@ecs-resource ~]# ip route show table 10
default via 10.0.0.1 dev eth0
10.0.0.0/24 dev eth0 scope link
[root@ecs-resource ~]# ip route show table 20
default via 10.0.1.1 dev eth1
10.0.1.0/24 dev eth1 scope link
```

c. Check whether the source ECS and the destination ECS can communicate with each other.

ping -I *IP address of the primary NIC on the source ECS IP address of the destination ECS*

ping -I *IP address of the extension NIC on the source ECS IP address of the destination ECS*

In this example, run the following commands:

ping -I 10.0.0.115 10.0.2.12

ping -I 10.0.1.183 10.0.2.12

If information similar to the following is displayed, both the NICs of the source ECS can communicate with the destination ECS.

```
[root@ecs-resource ~]# ping -I 10.0.0.115 10.0.2.12
PING 10.0.2.12 (10.0.2.12) from 10.0.0.115 : 56(84) bytes of data.
64 bytes from 10.0.2.12: icmp_seq=1 ttl=64 time=0.775 ms
64 bytes from 10.0.2.12: icmp_seq=2 ttl=64 time=0.268 ms
64 bytes from 10.0.2.12: icmp_seq=3 ttl=64 time=0.220 ms
64 bytes from 10.0.2.12: icmp_seq=4 ttl=64 time=0.167 ms
^C
--- 10.0.2.12 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 102ms
rtt min/avg/max/mdev = 0.167/0.357/0.775/0.244 ms
[root@ecs-resource ~]# ping -I 10.0.1.183 10.0.2.12
PING 10.0.2.12 (10.0.2.12) from 10.0.1.183 : 56(84) bytes of data.
64 bytes from 10.0.2.12: icmp_seq=1 ttl=64 time=2.84 ms
64 bytes from 10.0.2.12: icmp_seq=2 ttl=64 time=0.258 ms
64 bytes from 10.0.2.12: icmp_seq=3 ttl=64 time=0.234 ms
```

```
64 bytes from 10.0.2.12: icmp_seq=4 ttl=64 time=0.153 ms
^C
--- 10.0.2.12 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 92ms
rtt min/avg/max/mdev = 0.153/0.871/2.840/1.137 ms
```

6. Configure persistent routes for the ECS.

   a. Run the following command to open the **/etc/rc.local** file:

      vi /etc/rc.local

   b. Press **i** to enter the editing mode.

   c. Add the following content to the end of the file:

```
# wait for nics up
sleep 5
# Add v4 routes for eth0
ip route flush table 10
ip route add default via 10.0.0.1 dev eth0 table 10
ip route add 10.0.0.0/24 dev eth0 table 10
ip rule add from 10.0.0.115 table 10
# Add v4 routes for eth1
ip route flush table 20
ip route add default via 10.0.1.1 dev eth1 table 20
ip route add 10.0.1.0/24 dev eth1 table 20
ip rule add from 10.0.1.183 table 20
# Add v4 routes for cloud-init
ip rule add to 169.254.169.254 table main
```

      Parameters are described as follows:

      - wait for nics up: file startup time. Set the value to be the same as that in the preceding configurations.
      - Add v4 routes for eth0: policy-based routes of the primary NIC. Set the value to be the same as that configured in 5.a.
      - Add v4 routes for eth1: policy-based routes of the extension NIC. Set the value to be the same as that configured in 5.a.
      - Add v4 routes for cloud-init: Configure the Cloud-Init address. Set the value to be the same as that in the preceding configurations.

   d. Press **ESC** to exit and enter **:wq!** to save the configuration.

   e. Run the following command to assign execute permissions to the **/etc/rc.local** file:

      chmod +x /etc/rc.local

      NOTE

      If your operating system is Red Hat or EulerOS, run the following command after you perform 6.e:

      chmod +x /etc/rc.d/rc.local

   f. Run the following command to restart the ECS:

      reboot

NOTICE

Policy-based routes added to the **/etc/rc.local** file take effect only after the ECS is restarted. Ensure that workloads on the ECS will not be affected before restarting the ECS.

g. Repeat **5.b** to **5.c** to check whether the policy-based routes are added and whether the source ECS and the destination ECS can communicate with each other.

9.4 Configuring Policy-based Routes for a Windows ECS with Multiple NICs

Scenarios

This section describes how to configure policy-based routes for a dual-NIC ECS running Windows Server 2012 (64-bit).

For details about the background knowledge and networking of dual-NIC ECSs, see **Overview**.

Procedure (Windows ECS Using IPv4)

1. Collect the ECS network information required for configuring policy-based routes.

   For details, see **Collecting ECS Network Information**.

2. Log in to an ECS.

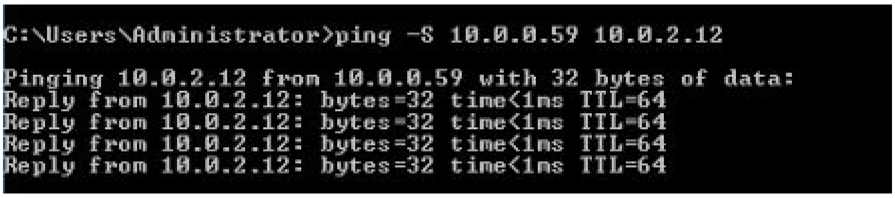
3. Check whether the source ECS can use its primary NIC to communicate with the destination ECS:

   ping -S *IP address of the primary NIC on the source ECS IP address of the destination ECS*

   In this example, run the following command:

   ping -S 10.0.0.59 10.0.2.12

   If information similar to the following is displayed, the source ECS can use its primary NIC to communicate with the destination ECS.



```
C:\Users\Administrator>ping -S 10.0.0.59 10.0.2.12

Pinging 10.0.2.12 from 10.0.0.59 with 32 bytes of data:
Reply from 10.0.2.12: bytes=32 time<1ms TTL=64
Reply from 10.0.2.12: bytes=32 time<1ms TTL=64
Reply from 10.0.2.12: bytes=32 time<1ms TTL=64
Reply from 10.0.2.12: bytes=32 time<1ms TTL=64
```

   NOTE

   Before configuring policy-based routes, ensure that the source ECS can use its primary NIC to communicate with the destination ECS.

4. Configure a policy-based route for the extension NIC.

   route add -p 0.0.0.0 mask 0.0.0.0 *Subnet gateway of the extension NIC* **metric** *Route priority*

   Configure the parameters as follows:

   - **0.0.0.0/0**: Default route. Do not change it.
   - Subnet gateway of the extension NIC: Enter the IP address collected in **1**.
   - Route priority: Set its value to 261. The priority of the extension NIC must be lower than that of the primary NIC. A larger value indicates a lower priority.

In this example, run the following command:

route add -p 0.0.0.0 mask 0.0.0.0 10.0.1.1 metric 261

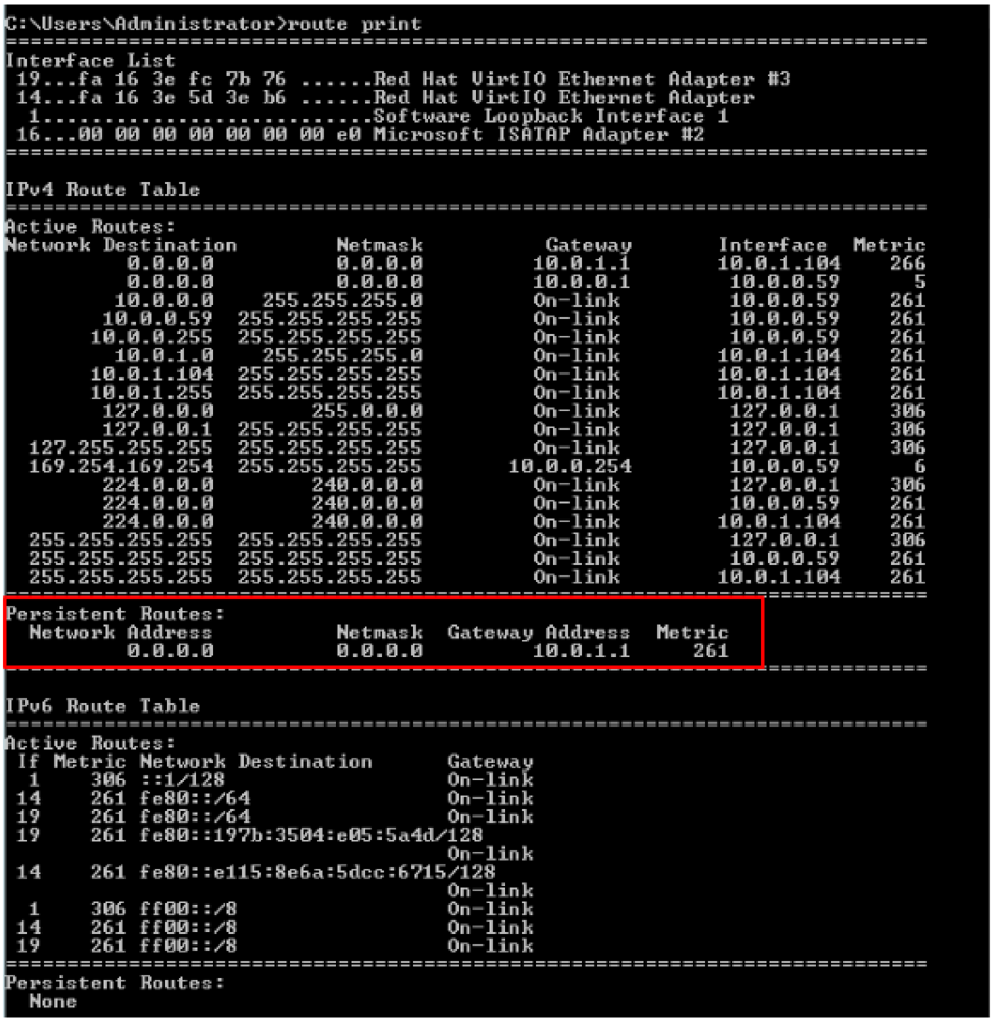
NOTE

- The primary NIC already has policy-based routes and you do not need to configure again.
- If the ECS has multiple extension NICs, configure policy-based routes for all extension NICs one by one.

5. Check whether the policy-based route is successfully added.

   route print

   If information similar to the following is displayed, the policy-based route has been added. The route is persistent and will not be lost after the ECS is restarted.



```
C:\Users\Administrator>route print
===========================================================================
Interface List
 19...fa 16 3e fc 7b 76 ......Red Hat VirtIO Ethernet Adapter #3
 14...fa 16 3e 5d 3e b6 ......Red Hat VirtIO Ethernet Adapter
  1...........................Software Loopback Interface 1
 16...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter #2
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0         10.0.1.1     10.0.1.104    266
          0.0.0.0          0.0.0.0         10.0.0.1      10.0.0.59      5
         10.0.0.0    255.255.255.0         On-link       10.0.0.59    261
        10.0.0.59  255.255.255.255         On-link       10.0.0.59    261
       10.0.0.255  255.255.255.255         On-link       10.0.0.59    261
         10.0.1.0    255.255.255.0         On-link      10.0.1.104    261
       10.0.1.104  255.255.255.255         On-link      10.0.1.104    261
       10.0.1.255  255.255.255.255         On-link      10.0.1.104    261
        127.0.0.0        255.0.0.0         On-link       127.0.0.1    306
        127.0.0.1  255.255.255.255         On-link       127.0.0.1    306
  127.255.255.255  255.255.255.255         On-link       127.0.0.1    306
  169.254.169.254  255.255.255.255       10.0.0.254      10.0.0.59      6
        224.0.0.0        240.0.0.0         On-link       127.0.0.1    306
        224.0.0.0        240.0.0.0         On-link       10.0.0.59    261
        224.0.0.0        240.0.0.0         On-link      10.0.1.104    261
  255.255.255.255  255.255.255.255         On-link       127.0.0.1    306
  255.255.255.255  255.255.255.255         On-link       10.0.0.59    261
  255.255.255.255  255.255.255.255         On-link      10.0.1.104    261
===========================================================================
Persistent Routes:
  Network Address          Netmask  Gateway Address  Metric
          0.0.0.0          0.0.0.0         10.0.1.1     261
===========================================================================

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  1    306 ::1/128                  On-link
 14    261 fe80::/64                On-link
 19    261 fe80::/64                On-link
 19    261 fe80::197b:3504:e05:5a4d/128
                                    On-link
 14    261 fe80::e115:8e6a:5dcc:6715/128
                                    On-link
  1    306 ff00::/8                 On-link
 14    261 ff00::/8                 On-link
 19    261 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None
```

6. Check whether the source ECS and the destination ECS can communicate with each other.

   ping -S *IP address of the primary NIC on the source ECS IP address of the destination ECS*

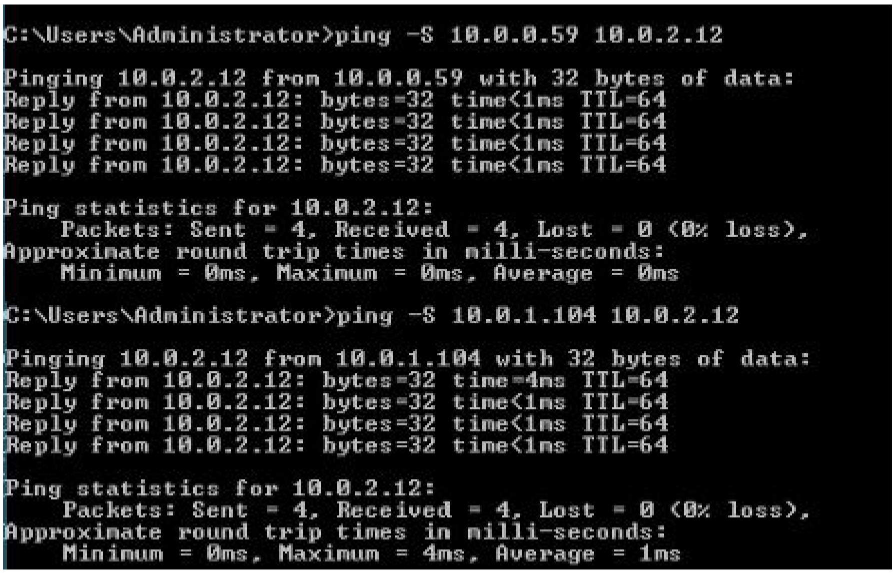
   ping -S *IP address of the extension NIC on the source ECS IP address of the destination ECS*

   In this example, run the following commands:

ping -S 10.0.0.59 10.0.2.12

ping -S 10.0.1.104 10.0.2.12

If information similar to the following is displayed, both the NICs of the source ECS can communicate with the destination ECS.



```
C:\Users\Administrator>ping -S 10.0.0.59 10.0.2.12

Pinging 10.0.2.12 from 10.0.0.59 with 32 bytes of data:
Reply from 10.0.2.12: bytes=32 time<1ms TTL=64
Reply from 10.0.2.12: bytes=32 time<1ms TTL=64
Reply from 10.0.2.12: bytes=32 time<1ms TTL=64
Reply from 10.0.2.12: bytes=32 time<1ms TTL=64

Ping statistics for 10.0.2.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Administrator>ping -S 10.0.1.104 10.0.2.12

Pinging 10.0.2.12 from 10.0.1.104 with 32 bytes of data:
Reply from 10.0.2.12: bytes=32 time=4ms TTL=64
Reply from 10.0.2.12: bytes=32 time<1ms TTL=64
Reply from 10.0.2.12: bytes=32 time<1ms TTL=64
Reply from 10.0.2.12: bytes=32 time<1ms TTL=64

Ping statistics for 10.0.2.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms
```

10 VPC Peering Connection Configurations

10.1 Overview

VPCs are isolated from each other. To connect two VPCs in the same region, you can use a VPC peering connection to route traffic between them using private IP addresses.

Here are some scenarios to help you determine which configuration is best suited to your networking requirements.

Table 10-1 VPC peering scenarios

Scenario	Configuration Description
● VPC CIDR blocks do not overlap. ● Subnet CIDR blocks do not overlap.	Create VPC peering connections to connect entire CIDR blocks of VPCs. For details, see **Connecting Entire CIDR Blocks of VPCs**.
● VPC CIDR blocks overlap. ● Some subnet CIDR blocks overlap.	Create VPC peering connections to connect specific subnets or ECSs from different VPCs. ● To connect specific subnets from two VPCs, the subnet CIDR blocks cannot overlap. For details, see **Connecting Specific Subnets from Different VPCs**. ● To connect specific ECSs from two VPCs, each ECS must have a unique private IP address. For details, see **Connecting Specific ECSs from Different VPCs**.
● VPC CIDR blocks overlap. ● All subnet CIDR blocks overlap.	VPC peering connections are not usable. For details, see **Unsupported VPC Peering Configurations**.

10.2 Using a VPC Peering Connection to Connect Two VPCs

You can configure a VPC peering connection and set the destination of the routes added to VPC route tables to the peer VPC CIDR block. In this way, all resources in the two VPCs are connected. **Table 10-2** shows example scenarios.

Table 10-2 Scenario description

Scenario	Scenario Description	IP Address Version	Example
Two VPCs peered together	You have two VPCs that require full access to each other's resources. For example, your company has VPC-A for the human resource department, and VPC-B for the finance department. The two departments require full access to each other's resources.	IPv4	**Two VPCs Peered Together (IPv4)**
		IPv6	**Two VPCs Peered Together (IPv6)**
Multiple VPCs peered together	You have multiple VPCs that require access to each other's resources. For example, your company has VPC-A for the human resource department, VPC-B for the finance department, and VPC-C for the marketing department. These departments require full access to each other's resources.	IPv4	**Multiple VPCs Peered Together (IPv4)**
		IPv4	**Multiple VPCs Peered Together Through Transitive Peering Connections (IPv4)**
		IPv6	**Multiple VPCs Peered Together (IPv6)**

Scenario	Scenario Description	IP Address Version	Example
One central VPC peered with two VPCs	You have a central VPC that requires access to two peer VPCs, and similarly, the peer VPCs require access to the central VPC. However, the two peer VPCs need to be isolated from each other. For example, public services (such as databases) are deployed on VPC-A. Both VPC-B and VPC-C need to access the databases, but they do not need to access each other.	IPv4	**One Central VPC Peered with Two VPCs (IPv4)**
		IPv6	**One Central VPC Peered with Two VPCs (IPv6)**
One central VPC with primary and secondary CIDR blocks peered with two VPCs	You have a central VPC that has both primary and secondary CIDR blocks. The central VPC needs to communicate with two peer VPCs, but the peer VPCs need to be isolated from each other.	IPv4	**One Central VPC with Primary and Secondary CIDR Blocks Peered with Two VPCs (IPv4)**
One central VPC peered with multiple VPCs	You have a central VPC that requires access to the multiple peer VPCs, and similarly, the peer VPCs require access to the central VPC. However, the peer VPCs need to be isolated from each other. For example, public services (such as databases) are deployed on your central VPC-A. VPC-B, VPC-C, VPC-D, VPC-E, VPC-F, and VPC-G need to access the databases, but these VPCs do not need to access each other.	IPv4	**One Central VPC Peered with Multiple VPCs (IPv4)**
		IPv6	**One Central VPC Peered with Multiple VPCs (IPv6)**

Notes and Constraints

If you create a VPC peering connection that connects entire CIDR blocks of two VPCs, the VPC CIDR blocks cannot overlap. Otherwise, the VPC peering connection does not take effect. For details, see **Invalid VPC Peering for Overlapping VPC CIDR Blocks**.

Even if you intend to use the VPC peering connection for IPv6 communication only, you cannot create a VPC peering connection if the VPCs have matching or overlapping IPv4 CIDR blocks. In all examples in this section, the IPv4 CIDR blocks of any VPCs connected by a VPC peering connection do not overlap.

Two VPCs Peered Together (IPv4)

Create Peering-AB between VPC-A and VPC-B. The CIDR blocks of VPC-A and VPC-B do not overlap.

- For details about resource planning, see **Table 10-3**.
- For details about VPC peering relationships, see **Table 10-4**.

Figure 10-1 Networking diagram (IPv4)

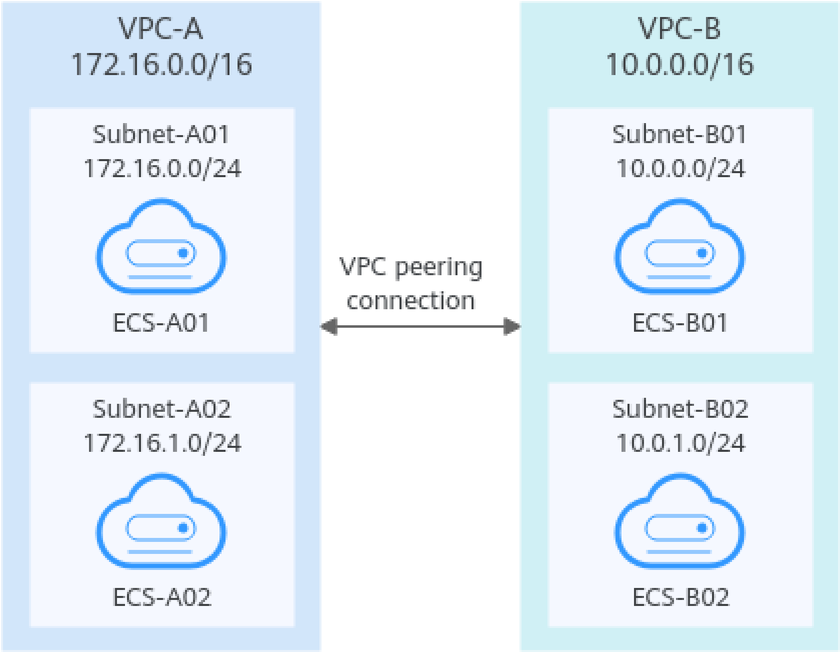


Table 10-3 Resource planning details (IPv4)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	172.16.0.0/24	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	172.16.0.111
		Subnet-A02	172.16.1.0/24	rtb-VPC-A	ECS-A02		172.16.1.91
VPC-B	10.0.0.0/16	Subnet-B01	10.0.0.0/24	rtb-VPC-B	ECS-B01		10.0.0.139
		Subnet-B02	10.0.1.0/24	rtb-VPC-B	ECS-B02		10.0.1.167

Table 10-4 Peering relationships (IPv4)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered with VPC-B.	Peering-AB	VPC-A	VPC-B

After the VPC peering connection is created, add the following routes to the route tables of the local and peer VPCs:

Table 10-5 VPC route tables (IPv4)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.1.0/24	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop.

NOTE

If the destination of a route in a route table of a VPC is set to the CIDR block of a peer VPC and the CIDR blocks of the two VPCs do not overlap, the VPCs can have full access to each other's resources.

Two VPCs Peered Together (IPv6)

Create Peering-AB between VPC-A and VPC-B. The subnets of VPC-A and VPC-B have both IPv4 and IPv6 CIDR blocks and their IPv4 CIDR blocks do not overlap.

- For details about resource planning, see **Table 10-6**.
- For details about VPC peering relationships, see **Table 10-7**.

Figure 10-2 Networking diagram (IPv6)

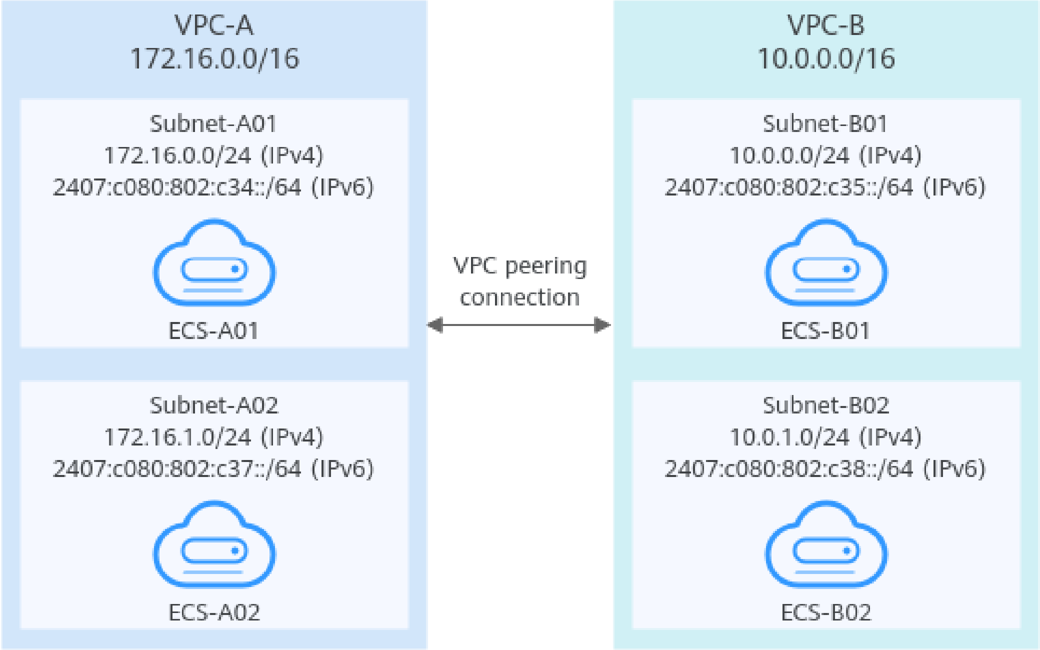


Table 10-6 Resource planning details (IPv6)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	- IPv4: 172.16.0.0/24 - IPv6: 2407:c080:802:c34::/64	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	- IPv4: 172.16.0.111 - IPv6: 2407:c080:802:c34:a925:f12e:cfa0:8edb
		Subnet-A02	- IPv4: 172.16.1.0/24 - IPv6: 2407:c080:802:c37::/64	rtb-VPC-A	ECS-A02		- IPv4: 172.16.1.91 - IPv6: 2407:c080:802:c37:594b:4c0f:2fcd:8b72

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-B	10.0.0.0/16	Subnet-B01	● IPv4: 10.0.0.0/24 ● IPv6: 2407:c080:802:c35::/64	rtb-VPC-B	ECS-B01		● IPv4: 10.0.0.139 ● IPv6: 2407:c080:802:c35:493:33f4:4531:5162
		Subnet-B02	● IPv4: 10.0.1.0/24 ● IPv6: 2407:c080:802:c38::/64	rtb-VPC-B	ECS-B02		● IPv4: 10.0.1.167 ● IPv6: 2407:c080:802:c38:b9a9:aa03:2700:c1cf

Table 10-7 Peering relationships (IPv6)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered with VPC-B.	Peering-AB	VPC-A	VPC-B

After the VPC peering connection is created, add the following routes to the route tables of the local and peer VPCs:

Table 10-8 VPC route tables (IPv6)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.

Route Table	Destination	Next Hop	Route Type	Description
	2407:c080:802:c34::/64	Local	System	
	172.16.1.0/24	Local	System	
	2407:c080:802:c37::/64	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop for IPv4 communication.
	2407:c080:802:c35::/64 (Subnet-B01)	Peering-AB	Custom	Add routes with the IPv6 CIDR blocks of Subnet-B01 and Subnet-B02 as the destinations and Peering-AB as the next hop for IPv6 communication.
	2407:c080:802:c38::/64 (Subnet-B02)	Peering-AB	Custom	
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c35::/64	Local	System	
	10.0.1.0/24	Local	System	
	2407:c080:802:c38::/64	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop for IPv4 communication.
	2407:c080:802:c34::/64 (Subnet-A01)	Peering-AB	Custom	Add routes with the IPv6 CIDR blocks of Subnet-A01 and Subnet-A02 as the destinations and Peering-AB as the next hop for IPv6 communication.
	2407:c080:802:c37::/64 (Subnet-A02)	Peering-AB	Custom	

NOTE

You can view IPv6 addresses of VPC subnets on the management console. To enable communication between entire CIDR blocks of two VPCs, you need to add routes with IPv6 CIDR blocks of all subnets in the VPCs as the destinations one by one.

Multiple VPCs Peered Together (IPv4)

If multiple VPCs need to communicate with each other, their CIDR blocks cannot overlap and you need to create a VPC peering connection between every two VPCs.

- For details about resource planning, see **Table 10-9**.
- For details about VPC peering relationships, see **Table 10-10**.

Figure 10-3 Networking diagram (IPv4)

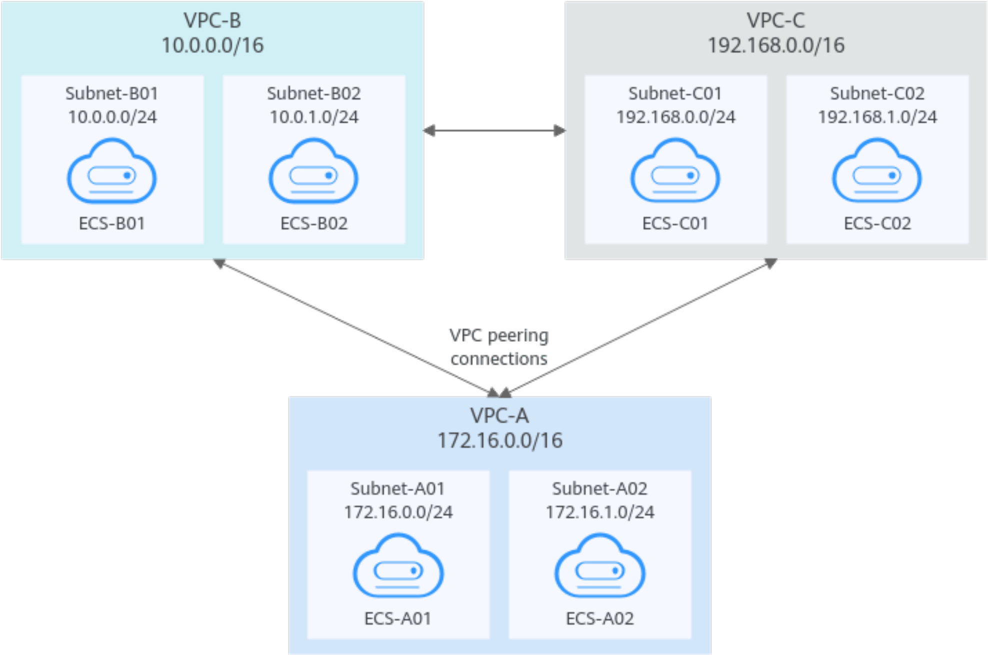


Table 10-9 Resource planning details (IPv4)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	172.16.0.0/24	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	172.16.0.111
		Subnet-A02	172.16.1.0/24	rtb-VPC-A	ECS-A02		172.16.1.91
VPC-B	10.0.0.0/16	Subnet-B01	10.0.0.0/24	rtb-VPC-B	ECS-B01		10.0.0.139

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
		Subnet-B02	10.0.1.0/24	rtb-VPC-B	ECS-B02		10.0.1.167
VPC-C	192.168.0.0/16	Subnet-C01	192.168.0.0/24	rtb-VPC-C	ECS-C01		192.168.0.194
		Subnet-C02	192.168.1.0/24	rtb-VPC-C	ECS-C02		192.168.1.200

Table 10-10 Peering relationships (IPv4)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered with VPC-B.	Peering-AB	VPC-A	VPC-B
VPC-A is peered with VPC-C.	Peering-AC	VPC-A	VPC-C
VPC-B is peered with VPC-C.	Peering-BC	VPC-B	VPC-C

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-11 VPC route tables (IPv4)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.1.0/24	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop.

Route Table	Destination	Next Hop	Route Type	Description
	192.168.0.0/16 (VPC-C)	Peering-AC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-AC as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop.
	192.168.0.0/16 (VPC-C)	Peering-BC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-BC as the next hop.
rtb-VPC-C	192.168.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	192.168.1.0/24	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AC	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AC as the next hop.
	10.0.0.0/16 (VPC-B)	Peering-BC	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-BC as the next hop.

NOTE

If the destination of a route in a route table of a VPC is set to the CIDR block of a peer VPC and the CIDR blocks of the two VPCs do not overlap, the VPCs can have full access to each other's resources.

Multiple VPCs Peered Together Through Transitive Peering Connections (IPv4)

VPC peering connections are transitive. As shown in **Figure 10-4**, there is a VPC peering connection between VPC-A and VPC-B, and between VPC-A and VPC-C. To enable communication between VPC-B and VP-C, you can use either of the following methods:

- Create a VPC peering connection between VPC-B and VPC-C. For details, see **Multiple VPCs Peered Together (IPv4)**.

- Add routes to direct traffic between VPC-B and VPC-C based on VPC-A. For details, see **Table 10-14**.

Figure 10-4 Transitive VPC peering connections

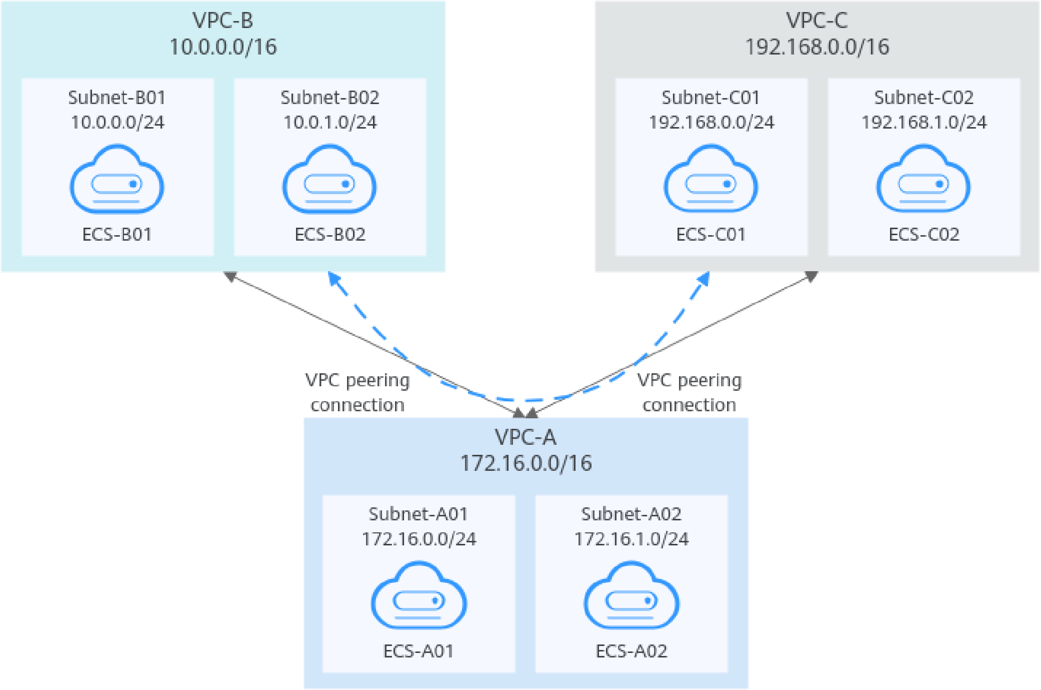


Table 10-12 Resource planning details (IPv4)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	172.16.0.0/24	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	172.16.0.111
		Subnet-A02	172.16.1.0/24	rtb-VPC-A	ECS-A02		172.16.1.91
VPC-B	10.0.0.0/16	Subnet-B01	10.0.0.0/24	rtb-VPC-B	ECS-B01		10.0.0.139
		Subnet-B02	10.0.1.0/24	rtb-VPC-B	ECS-B02		10.0.1.167
VPC-C	192.168.0.0/16	Subnet-C01	192.168.0.0/24	rtb-VPC-C	ECS-C01		192.168.0.194
		Subnet-C02	192.168.1.0/24	rtb-VPC-C	ECS-C02		192.168.1.200

Table 10-13 Peering relationships (IPv4)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered with VPC-B.	Peering-AB	VPC-A	VPC-B
VPC-A is peered with VPC-C.	Peering-AC	VPC-A	VPC-C

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-14 VPC route tables (IPv4)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.1.0/24	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop.
	192.168.0.0/16 (VPC-C)	Peering-AC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-AC as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop.
	192.168.0.0/16 (VPC-C)	Peering-AB	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-AB as the next hop.
rtb-VPC-C	192.168.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.

Route Table	Destination	Next Hop	Route Type	Description
	192.168.1.0/24	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AC	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AC as the next hop.
	10.0.0.0/16 (VPC-B)	Peering-AC	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AC as the next hop.

NOTE

If the destination of a route in a route table of a VPC is set to the CIDR block of a peer VPC and the CIDR blocks of the two VPCs do not overlap, the VPCs can have full access to each other's resources.

Multiple VPCs Peered Together (IPv6)

If multiple VPCs need to communicate with each other, you need to create a VPC peering connection between every two VPCs. In this example, subnets in VPC-A, VPC-B, and VPC-C have IPv6 CIDR blocks and the IPv4 CIDR blocks of VPC-A, VPC-B, and VPC-C cannot overlap.

- For details about resource planning, see **Table 10-15**.
- For details about VPC peering relationships, see **Table 10-16**.

Figure 10-5 Networking diagram (IPv6)

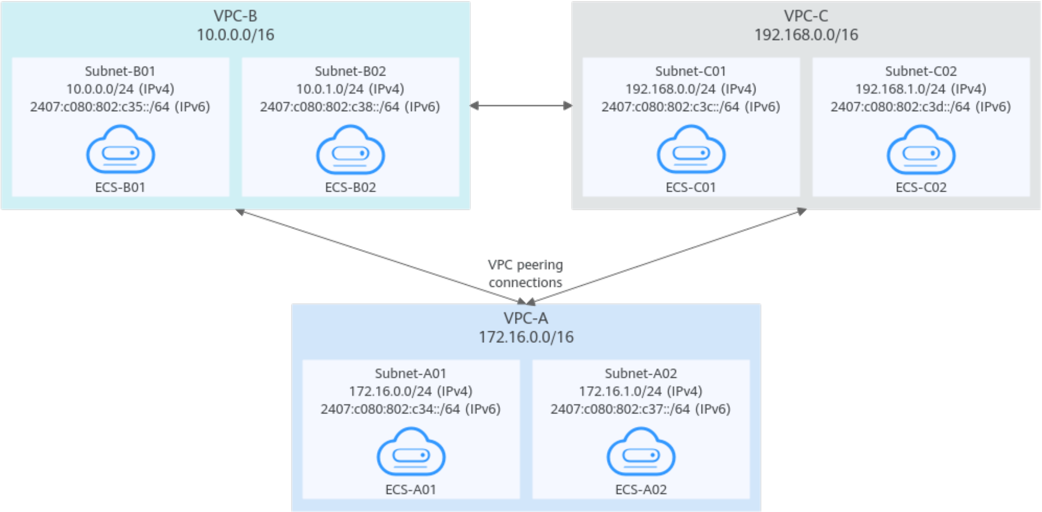


Table 10-15 Resource planning details (IPv6)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	● IPv4: 172.16.0.0/24 ● IPv6: 2407:c080:802:c34::/64	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	● IPv4: 172.16.0.111 ● IPv6: 2407:c080:802:c34:a925:f12e:cfa0:8edb
		Subnet-A02	● IPv4: 172.16.1.0/24 ● IPv6: 2407:c080:802:c37::/64	rtb-VPC-A	ECS-A02		● IPv4: 172.16.1.91 ● IPv6: 2407:c080:802:c37:594b:4c0f:2fcd:8b72
VPC-B	10.0.0.0/16	Subnet-B01	● IPv4: 10.0.0.0/24 ● IPv6: 2407:c080:802:c35::/64	rtb-VPC-B	ECS-B01		● IPv4: 10.0.0.139 ● IPv6: 2407:c080:802:c35:493:33f4:4531:5162
		Subnet-B02	● IPv4: 10.0.1.0/24 ● IPv6: 2407:c080:802:c38::/64	rtb-VPC-B	ECS-B02		● IPv4: 10.0.1.167 ● IPv6: 2407:c080:802:c38:b9a9:aa03:2700:c1cf

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-C	192.168.0.0/16	Subnet-C01	● IPv4: 192.168.0.0/24 ● IPv6: 2407:c080:802:c3c::/64	rtb-VPC-C	ECS-C01		● IPv4: 192.168.0.194 ● IPv6: 2407:c080:802:c3c:d2f3:d891:24f5:f4af
		Subnet-C02	● IPv4: 192.168.1.0/24 ● IPv6: 2407:c080:802:c3d::/64	rtb-VPC-C	ECS-C02		● IPv4: 192.168.1.200 ● IPv6: 2407:c080:802:c3d:e9ca:169a:390c:74d1

Table 10-16 Peering relationships (IPv6)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered with VPC-B.	Peering-AB	VPC-A	VPC-B
VPC-A is peered with VPC-C.	Peering-AC	VPC-A	VPC-C
VPC-B is peered with VPC-C.	Peering-BC	VPC-B	VPC-C

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-17 VPC route tables (IPv6)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c34::/64	Local	System	
	172.16.1.0/24	Local	System	
	2407:c080:802:c37::/64	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop for IPv4 communication.
	2407:c080:802:c35::/64 (Subnet-B01)	Peering-AB	Custom	Add routes with the IPv6 CIDR blocks of Subnet-B01 and Subnet-B02 as the destinations and Peering-AB as the next hop for IPv6 communication.
	2407:c080:802:c38::/64 (Subnet-B02)	Peering-AB	Custom	
	192.168.0.0/16 (VPC-C)	Peering-AC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-AC as the next hop for IPv4 communication.
	2407:c080:802:c3c::/64 (Subnet-C01)	Peering-AC	Custom	Add routes with the IPv6 CIDR blocks of Subnet-C01 and Subnet-C02 as the destinations and Peering-AC as the next hop for IPv6 communication.
	2407:c080:802:c3d::/64 (Subnet-C02)	Peering-AC	Custom	
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c35::/64	Local	System	
	10.0.1.0/24	Local	System	
	2407:c080:802:c38::/64	Local	System	

Route Table	Destination	Next Hop	Route Type	Description
	172.16.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop for IPv4 communication.
	2407:c080:802:c34::/64 (Subnet-A01)	Peering-AB	Custom	Add routes with the IPv6 CIDR blocks of Subnet-A01 and Subnet-A02 as the destinations and Peering-AB as the next hop for IPv6 communication.
	2407:c080:802:c37::/64 (Subnet-A02)	Peering-AB	Custom	
	192.168.0.0/16 (VPC-C)	Peering-BC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-BC as the next hop for IPv4 communication.
	2407:c080:802:c3c::/64 (Subnet-C01)	Peering-BC	Custom	Add routes with the IPv6 CIDR blocks of Subnet-C01 and Subnet-C02 as the destinations and Peering-BC as the next hop for IPv6 communication.
	2407:c080:802:c3d::/64 (Subnet-C02)	Peering-BC	Custom	
rtb-VPC-C	192.168.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c3c::/64	Local	System	
	192.168.1.0/24	Local	System	
	2407:c080:802:c3d::/64	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AC	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AC as the next hop for IPv4 communication.
	2407:c080:802:c34::/64 (Subnet-A01)	Peering-AC	Custom	Add routes with the IPv6 CIDR blocks of Subnet-A01 and Subnet-A02 as the destinations and Peering-AC as the next hop for IPv6 communication.
	2407:c080:802:c37::/64 (Subnet-A02)	Peering-AC	Custom	

Route Table	Destination	Next Hop	Route Type	Description
	10.0.0.0/16 (VPC-B)	Peering-BC	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-BC as the next hop for IPv4 communication.
	2407:c080:802:c35::/64 (Subnet-B01)	Peering-BC	Custom	Add routes with the IPv6 CIDR blocks of Subnet-B01 and Subnet-B02 as the destinations and Peering-BC as the next hop for IPv6 communication.
	2407:c080:802:c38::/64 (Subnet-B02)	Peering-BC	Custom	

NOTE

You can view IPv6 addresses of VPC subnets on the management console. To enable communication between entire CIDR blocks of two VPCs, you need to add routes with IPv6 CIDR blocks of all subnets in the VPCs as the destinations one by one.

One Central VPC Peered with Two VPCs (IPv4)

Create Peering-AB between VPC-A and VPC-B, and Peering-AC between VPC-A and VPC-C. The three VPCs do not have overlapping CIDR blocks.

- For details about resource planning, see **Table 10-18**.
- For details about VPC peering relationships, see **Table 10-19**.

Figure 10-6 Networking diagram (IPv4)

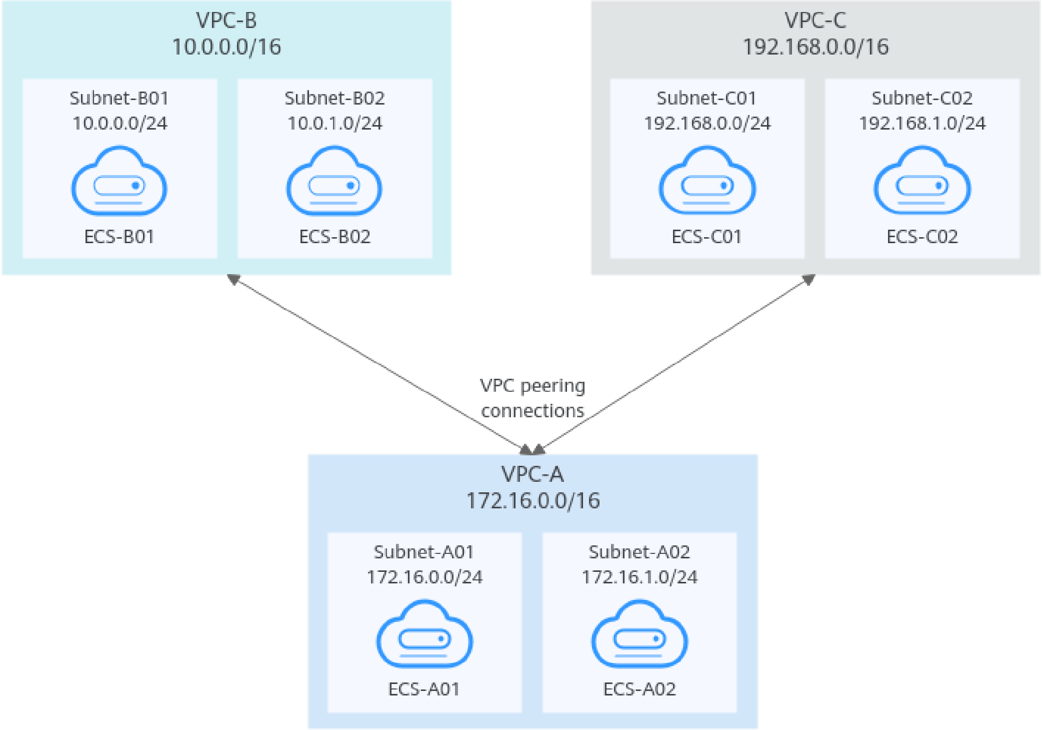


Table 10-18 Resource planning details (IPv4)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	172.16.0.0/24	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	172.16.0.111
		Subnet-A02	172.16.1.0/24	rtb-VPC-A	ECS-A02		172.16.1.91
VPC-B	10.0.0.0/16	Subnet-B01	10.0.0.0/24	rtb-VPC-B	ECS-B01		10.0.0.139
		Subnet-B02	10.0.1.0/24	rtb-VPC-B	ECS-B02		10.0.1.167
VPC-C	192.168.0.0/16	Subnet-C01	192.168.0.0/24	rtb-VPC-C	ECS-C01		192.168.0.194
		Subnet-C02	192.168.1.0/24	rtb-VPC-C	ECS-C02		192.168.1.200

Table 10-19 Peering relationships (IPv4)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered with VPC-B.	Peering-AB	VPC-A	VPC-B
VPC-A is peered with VPC-C.	Peering-AC	VPC-A	VPC-C

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-20 VPC route table details (IPv4)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.1.0/24	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop.
	192.168.0.0/16 (VPC-C)	Peering-AC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-AC as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop.
rtb-VPC-C	192.168.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	192.168.1.0/24	Local	System	

Route Table	Destination	Next Hop	Route Type	Description
	172.16.0.0/16 (VPC-A)	Peering-AC	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AC as the next hop.

NOTE

If the destination of a route in a route table of a VPC is set to the CIDR block of a peer VPC and the CIDR blocks of the two VPCs do not overlap, the VPCs can have full access to each other's resources.

One Central VPC Peered with Two VPCs (IPv6)

Create Peering-AB between VPC-A and VPC-B, and Peering-AC between VPC-A and VPC-C. Each VPC has IPv6 subnets. The IPv4 CIDR blocks of the three VPCs do not overlap with each other.

- For details about resource planning, see **Table 10-21**.
- For details about VPC peering relationships, see **Table 10-22**.

Figure 10-7 Networking diagram (IPv6)

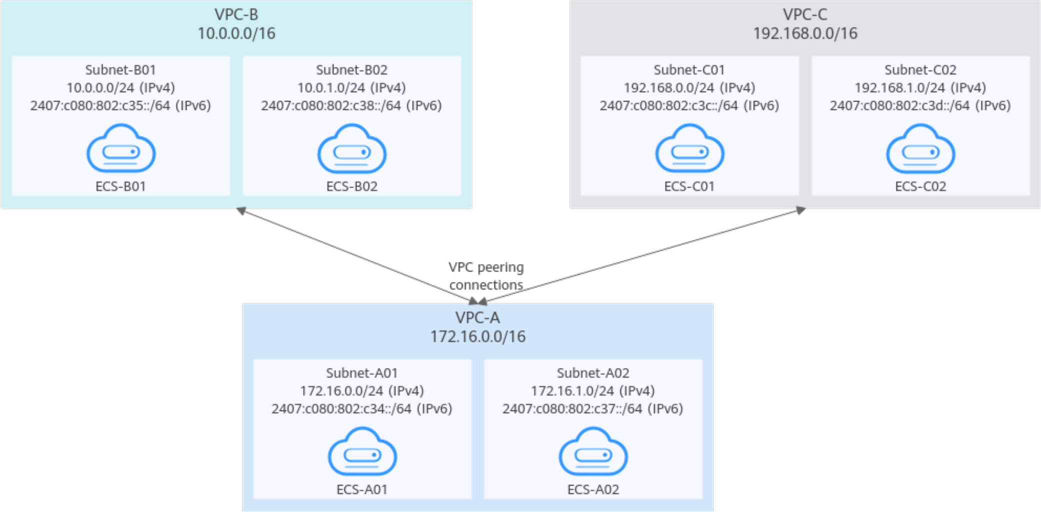


Table 10-21 Resource planning details (IPv6)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	● IPv4: 172.16.0.0/24 ● IPv6: 2407:c080:802:c34::/64	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	● IPv4: 172.16.0.111 ● IPv6: 2407:c080:802:c34:a925:f12e:cfa0:8edb
		Subnet-A02	● IPv4: 172.16.1.0/24 ● IPv6: 2407:c080:802:c37::/64	rtb-VPC-A	ECS-A02		● IPv4: 172.16.1.91 ● IPv6: 2407:c080:802:c37:594b:4c0f:2fcd:8b72
VPC-B	10.0.0.0/16	Subnet-B01	● IPv4: 10.0.0.0/24 ● IPv6: 2407:c080:802:c35::/64	rtb-VPC-B	ECS-B01		● IPv4: 10.0.0.139 ● IPv6: 2407:c080:802:c35:493:33f4:4531:5162
		Subnet-B02	● IPv4: 10.0.1.0/24 ● IPv6: 2407:c080:802:c38::/64	rtb-VPC-B	ECS-B02		● IPv4: 10.0.1.167 ● IPv6: 2407:c080:802:c38:b9a9:aa03:2700:c1cf

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-C	192.168.0.0/16	Subnet-C01	● IPv4: 192.168.0.0/24 ● IPv6: 2407:c080:802:c3c::/64	rtb-VPC-C	ECS-C01		● IPv4: 192.168.0.194 ● IPv6: 2407:c080:802:c3c:d2f3:d891:24f5:f4af
		Subnet-C02	● IPv4: 192.168.1.0/24 ● IPv6: 2407:c080:802:c3d::/64	rtb-VPC-C	ECS-C02		● IPv4: 192.168.1.200 ● IPv6: 2407:c080:802:c3d:e9ca:169a:390c:74d1

Table 10-22 Peering relationships (IPv6)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered with VPC-B.	Peering-AB	VPC-A	VPC-B
VPC-A is peered with VPC-C.	Peering-AC	VPC-A	VPC-C

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-23 VPC route table details (IPv6)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c34::/64	Local	System	
	172.16.1.0/24	Local	System	
	2407:c080:802:c37::/64	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop for IPv4 communication.
	2407:c080:802:c35::/64 (Subnet-B01)	Peering-AB	Custom	Add routes with the IPv6 CIDR blocks of Subnet-B01 and Subnet-B02 as the destinations and Peering-AB as the next hop for IPv6 communication.
	2407:c080:802:c38::/64 (Subnet-B02)	Peering-AB	Custom	
	192.168.0.0/16 (VPC-C)	Peering-AC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-AC as the next hop for IPv4 communication.
	2407:c080:802:c3c::/64 (Subnet-C01)	Peering-AC	Custom	Add routes with the IPv6 CIDR blocks of Subnet-C01 and Subnet-C02 as the destinations and Peering-AC as the next hop for IPv6 communication.
	2407:c080:802:c3d::/64 (Subnet-C02)	Peering-AC	Custom	
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c35::/64	Local	System	
	10.0.1.0/24	Local	System	
	2407:c080:802:c38::/64	Local	System	

Route Table	Destination	Next Hop	Route Type	Description
	172.16.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop for IPv4 communication.
	2407:c080:802:c34::/64 (Subnet-A01)	Peering-AB	Custom	Add routes with the IPv6 CIDR blocks of Subnet-A01 and Subnet-A02 as the destinations and Peering-AB as the next hop for IPv6 communication.
	2407:c080:802:c37::/64 (Subnet-A02)	Peering-AB	Custom	
rtb-VPC-C	192.168.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c3c::/64	Local	System	
	192.168.1.0/24	Local	System	
	2407:c080:802:c3d::/64	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AC	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AC as the next hop for IPv4 communication.
	2407:c080:802:c34::/64 (Subnet-A01)	Peering-AC	Custom	Add routes with the IPv6 CIDR blocks of Subnet-A01 and Subnet-A02 as the destinations and Peering-AC as the next hop for IPv6 communication.
	2407:c080:802:c37::/64 (Subnet-A02)	Peering-AC	Custom	

NOTE

You can view IPv6 addresses of VPC subnets on the management console. To enable communication between entire CIDR blocks of two VPCs, you need to add routes with IPv6 CIDR blocks of all subnets in the VPCs as the destinations one by one.

One Central VPC with Primary and Secondary CIDR Blocks Peered with Two VPCs (IPv4)

Create Peering-AB between VPC-A and VPC-B, and Peering-AC between VPC-A and VPC-C. VPC-A has both primary and secondary CIDR blocks. The three VPCs do not have overlapping CIDR blocks.

- For details about resource planning, see **Table 10-24**.
- For details about VPC peering relationships, see **Table 10-25**.

Figure 10-8 Networking diagram (IPv4)

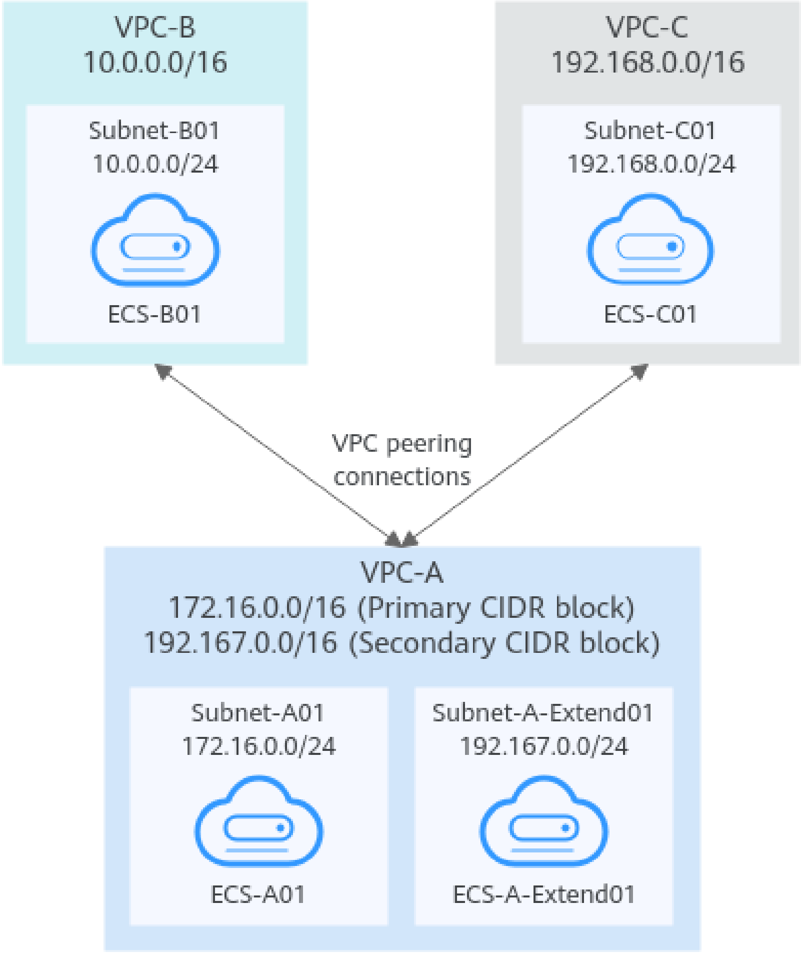


Table 10-24 Resource planning details

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-A	Primary CIDR block: 172.16.0.0/16	Subnet-A01	172.16.0.0/24	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	172.16.0.111
	Secondary CIDR block: 192.167.0.0/16	Subnet-A-Extend01	192.167.0.0/24	rtb-VPC-A	ECS-A-Extend01		192.167.0.100
VPC-B	10.0.0.0/16	Subnet-B01	10.0.0.0/24	rtb-VPC-B	ECS-B01		10.0.0.139
VPC-C	192.168.0.0/16	Subnet-C01	192.168.0.0/24	rtb-VPC-C	ECS-C01		192.168.0.194

Table 10-25 Peering relationships (IPv4)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered with VPC-B.	Peering-AB	VPC-A	VPC-B
VPC-A is peered with VPC-C.	Peering-AC	VPC-A	VPC-C

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-26 VPC route table details (IPv4)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	192.167.0.0/24	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop.
	192.168.0.0/16 (VPC-C)	Peering-AC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-AC as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.0.0/16 (Primary CIDR block of VPC-A)	Peering-AB	Custom	Add routes with the primary and secondary CIDR blocks of VPC-A as the destinations and Peering-AB as the next hop.
	192.167.0.0/16 (Secondary CIDR block of VPC-A)	Peering-AB	Custom	
rtb-VPC-C	192.168.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.0.0/16 (Primary CIDR block of VPC-A)	Peering-AC	Custom	Add routes with the primary and secondary CIDR blocks of VPC-A as the destinations and Peering-AC as the next hop.
	192.167.0.0/16 (Secondary CIDR block of VPC-A)	Peering-AC	Custom	

NOTE

If the destination of a route in a route table of a VPC is set to the CIDR block of a peer VPC and the CIDR blocks of the two VPCs do not overlap, the VPCs can have full access to each other's resources.

One Central VPC Peered with Multiple VPCs (IPv4)

Create a VPC peering connection between VPC-A and VPC-B, between VPC-A and VPC-C, between VPC-A and VPC-D, between VPC-A and VPC-E, between VPC-A and VPC-F, and between VPC-A and VPC-G. The CIDR blocks of these VPCs do not overlap.

- For details about resource planning, see **Table 10-27**.
- For details about VPC peering relationships, see **Table 10-28**.

Figure 10-9 Networking diagram (IPv4)

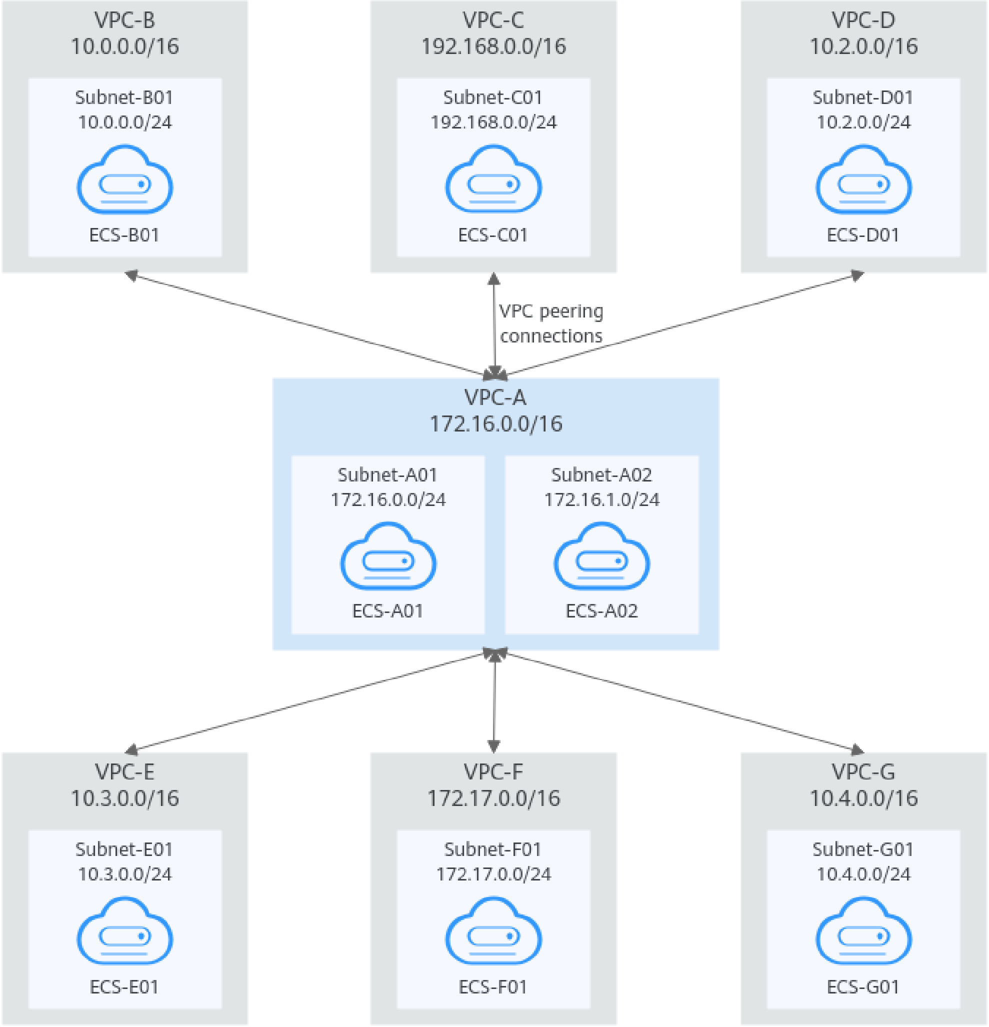


Table 10-27 Resource planning details (IPv4)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	172.16.0.0/24	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	172.16.0.111
		Subnet-A02	172.16.1.0/24	rtb-VPC-A	ECS-A02		172.16.1.91

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-B	10.0.0.0/16	Subnet-B01	10.0.0.0/24	rtb-VPC-B	ECS-B01		10.0.0.139
VPC-C	192.168.0.0/16	Subnet-C01	192.168.0.0/24	rtb-VPC-C	ECS-C01		192.168.0.194
VPC-D	10.2.0.0/16	Subnet-D01	10.2.0.0/24	rtb-VPC-D	ECS-D01		10.2.0.237
VPC-E	10.3.0.0/16	Subnet-E01	10.3.0.0/24	rtb-VPC-E	ECS-E01		10.3.0.87
VPC-F	172.17.0.0/16	Subnet-F01	172.17.0.0/24	rtb-VPC-F	ECS-F01		172.17.0.103
VPC-G	10.4.0.0/16	Subnet-G01	10.4.0.0/24	rtb-VPC-G	ECS-G01		10.4.0.10

Table 10-28 Peering relationships (IPv4)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered with VPC-B.	Peering-AB	VPC-A	VPC-B
VPC-A is peered with VPC-C.	Peering-AC	VPC-A	VPC-C
VPC-A is peered with VPC-D.	Peering-AD	VPC-A	VPC-D
VPC-A is peered with VPC-E.	Peering-AE	VPC-A	VPC-E
VPC-A is peered with VPC-F.	Peering-AF	VPC-A	VPC-F
VPC-A is peered with VPC-G.	Peering-AG	VPC-A	VPC-G

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-29 VPC route table details (IPv4)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.1.0/24	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop.
	192.168.0.0/16 (VPC-C)	Peering-AC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-AC as the next hop.
	10.2.0.0/16 (VPC-D)	Peering-AD	Custom	Add a route with the CIDR block of VPC-D as the destination and Peering-AD as the next hop.
	10.3.0.0/16 (VPC-E)	Peering-AE	Custom	Add a route with the CIDR block of VPC-E as the destination and Peering-AE as the next hop.
	172.17.0.0/16 (VPC-F)	Peering-AF	Custom	Add a route with the CIDR block of VPC-F as the destination and Peering-AF as the next hop.
	10.4.0.0/16 (VPC-G)	Peering-AG	Custom	Add a route with the CIDR block of VPC-G as the destination and Peering-AG as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop.
rtb-VPC-C	192.168.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.0.0/16 (VPC-A)	Peering-AC	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AC as the next hop.
rtb-VPC-D	10.2.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.

Route Table	Destination	Next Hop	Route Type	Description
	172.16.0.0/16 (VPC-A)	Peering-AD	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AD as the next hop.
rtb-VPC-E	10.3.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.0.0/16 (VPC-A)	Peering-AE	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AE as the next hop.
rtb-VPC-F	172.17.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.0.0/16 (VPC-A)	Peering-AF	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AF as the next hop.
rtb-VPC-G	10.4.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.0.0/16 (VPC-A)	Peering-AG	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AG as the next hop.

NOTE

If the destination of a route in a route table of a VPC is set to the CIDR block of a peer VPC and the CIDR blocks of the two VPCs do not overlap, the VPCs can have full access to each other's resources.

One Central VPC Peered with Multiple VPCs (IPv6)

Create a VPC peering connection between VPC-A and VPC-B, between VPC-A and VPC-C, between VPC-A and VPC-D, between VPC-A and VPC-E, between VPC-A and VPC-F, and between VPC-A and VPC-G. Each VPC has IPv6 subnets. The IPv4 CIDR blocks of these VPCs do not overlap.

- For details about resource planning, see **Table 10-30**.
- For details about VPC peering relationships, see **Table 10-31**.

Figure 10-10 Networking diagram (IPv6)

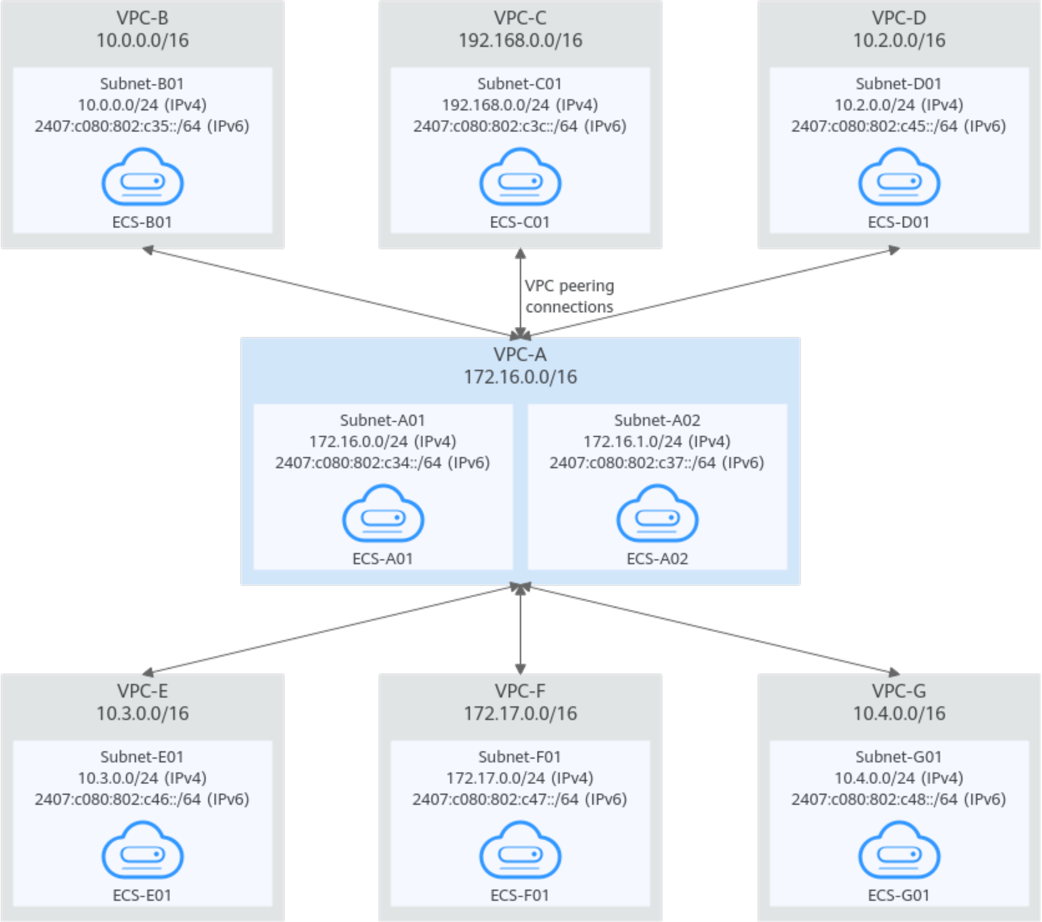


Table 10-30 Resource planning details (IPv6)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	- IPv4: 172.16.0.0/24 - IPv6: 2407:c080:802:c34::/64	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	- IPv4: 172.16.0.111 - IPv6: 2407:c080:802:c34:a925:f12e:cfa0:8edb

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
		Subnet-A02	● IPv4: 172.16.1.0/24 ● IPv6: 2407:c080:802:c37::/64	rtb-VPC-A	ECS-A02		● IPv4: 172.16.1.91 ● IPv6: 2407:c080:802:c37:594b:4c0f:2fcd:8b72
VPC-B	10.0.0.0/16	Subnet-B01	● IPv4: 10.0.0.0/24 ● IPv6: 2407:c080:802:c35::/64	rtb-VPC-B	ECS-B01		● IPv4: 10.0.0.139 ● IPv6: 2407:c080:802:c35:493:33f4:4531:5162
VPC-C	192.168.0.0/16	Subnet-C01	● IPv4: 192.168.0.0/24 ● IPv6: 2407:c080:802:c3c::/64	rtb-VPC-C	ECS-C01		● IPv4: 192.168.0.194 ● IPv6: 2407:c080:802:c3c:d2f3:d891:24f5:f4af
VPC-D	10.2.0.0/16	Subnet-D01	● IPv4: 10.2.0.0/24 ● IPv6: 2407:c080:802:c45::/64	rtb-VPC-D	ECS-D01		● IPv4: 10.2.0.237 ● IPv6: 2407:c080:802:c45:6bb7:f161:3596:6e4c

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	Subnet Route Table	ECS Name	Security Group	Private IP Address
VPC-E	10.3.0.0/16	Subnet-E01	● IPv4: 10.3.0.0/24 ● IPv6: 2407:c080:802:c46::/64	rtb-VPC-E	ECS-E01		● IPv4: 10.3.0.87 ● IPv6: 2407:c080:802:c46:2a2f:558a:85da:4c70
VPC-F	172.17.0.0/16	Subnet-F01	● IPv4: 172.17.0.0/24 ● IPv6: 2407:c080:802:c47::/64	rtb-VPC-F	ECS-F01		● IPv4: 172.17.0.103 ● IPv6: 2407:c080:802:c47:b5e2:e6f0:c42b:44fd
VPC-G	10.4.0.0/16	Subnet-G01	● IPv4: 10.4.0.0/24 ● IPv6: 2407:c080:802:c48::/64	rtb-VPC-G	ECS-G01		● IPv4: 10.4.0.10 ● IPv6: 2407:c080:802:c48:3020:f48c:4e54:aa17

Table 10-31 Peering relationships (IPv6)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered with VPC-B.	Peering-AB	VPC-A	VPC-B

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered with VPC-C.	Peering-AC	VPC-A	VPC-C
VPC-A is peered with VPC-D.	Peering-AD	VPC-A	VPC-D
VPC-A is peered with VPC-E.	Peering-AE	VPC-A	VPC-E
VPC-A is peered with VPC-F.	Peering-AF	VPC-A	VPC-F
VPC-A is peered with VPC-G.	Peering-AG	VPC-A	VPC-G

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-32 VPC route table details (IPv6)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c34::/64	Local	System	
	172.16.1.0/24	Local	System	
	2407:c080:802:c37::/64	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop for IPv4 communication.
	2407:c080:802:c35::/64 (Subnet-B01)	Peering-AB	Custom	Add a route with the IPv6 CIDR block of Subnet-B01 as the destination and Peering-AB as the next hop for IPv6 communication.

Route Table	Destination	Next Hop	Route Type	Description
	192.168.0.0/16 (VPC-C)	Peering-AC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-AC as the next hop for IPv4 communication.
	2407:c080:802:c3c::/64 (Subnet-C01)	Peering-AC	Custom	Add a route with the IPv6 CIDR block of Subnet-C01 as the destination and Peering-AC as the next hop for IPv6 communication.
	10.2.0.0/16 (VPC-D)	Peering-AD	Custom	Add a route with the CIDR block of VPC-D as the destination and Peering-AD as the next hop for IPv4 communication.
	2407:c080:802:c45::/64 (Subnet-D01)	Peering-AD	Custom	Add a route with the IPv6 CIDR block of Subnet-D01 as the destination and Peering-AD as the next hop for IPv6 communication.
	10.3.0.0/16 (VPC-E)	Peering-AE	Custom	Add a route with the CIDR block of VPC-E as the destination and Peering-AE as the next hop for IPv4 communication.
	2407:c080:802:c46::/64 (Subnet-E01)	Peering-AE	Custom	Add a route with the IPv6 CIDR block of Subnet-E01 as the destination and Peering-AE as the next hop for IPv6 communication.
	172.17.0.0/16 (VPC-F)	Peering-AF	Custom	Add a route with the CIDR block of VPC-F as the destination and Peering-AF as the next hop for IPv4 communication.
	2407:c080:802:c47::/64 (Subnet-F01)	Peering-AF	Custom	Add a route with the IPv6 CIDR block of Subnet-F01 as the destination and Peering-AF as the next hop for IPv6 communication.
	10.4.0.0/16 (VPC-G)	Peering-AG	Custom	Add a route with the CIDR block of VPC-G as the destination and Peering-AG as the next hop for IPv4 communication.
	2407:c080:802:c48::/64 (Subnet-G01)	Peering-AG	Custom	Add a route with the IPv6 CIDR block of Subnet-G01 as the destination and Peering-AG as the next hop for IPv6 communication.

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c35::/64	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop for IPv4 communication.
	2407:c080:802:c34::/64 (Subnet-A01)	Peering-AB	Custom	Add routes with the IPv6 CIDR blocks of Subnet-A01 and Subnet-A02 as the destinations and Peering-AB as the next hop for IPv6 communication.
	2407:c080:802:c37::/64 (Subnet-A02)	Peering-AB	Custom	
rtb-VPC-C	192.168.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c3c::/64	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AC	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AC as the next hop for IPv4 communication.
	2407:c080:802:c34::/64 (Subnet-A01)	Peering-AC	Custom	Add routes with the IPv6 CIDR blocks of Subnet-A01 and Subnet-A02 as the destinations and Peering-AC as the next hop for IPv6 communication.
	2407:c080:802:c37::/64 (Subnet-A02)	Peering-AC	Custom	
rtb-VPC-D	10.2.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c45::/64	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AD	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AD as the next hop for IPv4 communication.

Route Table	Destination	Next Hop	Route Type	Description
	2407:c080:802:c34::/64 (Subnet-A01)	Peering-AD	Custom	Add routes with the IPv6 CIDR blocks of Subnet-A01 and Subnet-A02 as the destinations and Peering-AD as the next hop for IPv6 communication.
	2407:c080:802:c37::/64 (Subnet-A02)	Peering-AD	Custom	
rtb-VPC-E	10.3.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c46::/64	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AE	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AE as the next hop.
	2407:c080:802:c34::/64 (Subnet-A01)	Peering-AE	Custom	Add routes with the IPv6 CIDR blocks of Subnet-A01 and Subnet-A02 as the destinations and Peering-AE as the next hop for IPv6 communication.
	2407:c080:802:c37::/64 (Subnet-A02)	Peering-AE	Custom	
rtb-VPC-F	172.17.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c47::/64	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AF	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AF as the next hop.
	2407:c080:802:c34::/64 (Subnet-A01)	Peering-AF	Custom	Add routes with the IPv6 CIDR blocks of Subnet-A01 and Subnet-A02 as the destinations and Peering-AF as the next hop for IPv6 communication.
	2407:c080:802:c37::/64 (Subnet-A02)	Peering-AF	Custom	
rtb-VPC-G	10.4.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c48::/64	Local	System	

Route Table	Destination	Next Hop	Route Type	Description
	172.16.0.0/16 (VPC-A)	Peering-AG	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AG as the next hop.
	2407:c080:802:c34::/64 (Subnet-A01)	Peering-AG	Custom	Add routes with the IPv6 CIDR blocks of Subnet-A01 and Subnet-A02 as the destinations and Peering-AG as the next hop for IPv6 communication.
	2407:c080:802:c37::/64 (Subnet-A02)	Peering-AG	Custom	

NOTE

You can view IPv6 addresses of VPC subnets on the management console. To enable communication between entire CIDR blocks of two VPCs, you need to add routes with IPv6 CIDR blocks of all subnets in the VPCs as the destinations one by one.

10.3 Using a VPC Peering Connection to Connect Subnets in Two VPCs

You can configure a VPC peering connection and set the destination of the routes added to VPC route tables to the subnet CIDR block of the peer VPC. In this way, all resources in the VPC subnets are connected. **Table 10-33** shows example scenarios.

Table 10-33 Scenario description

Scenario	Scenario Description	IP Address Version	Example
Two VPCs peered to two subnets in a central VPC	You have a central VPC that requires access to the multiple other VPCs. The other VPCs need to be isolated from each other. ● The central VPC has separate sets of resources in different subnets. ● The other VPCs require access to some of the resources, but not all of them.	IPv4	**Two VPCs Peered to Two Subnets in a Central VPC (IPv4)**
		IPv6/ IPv4	**Two VPCs Peered to Two Subnets in a Central VPC (IPv6/ IPv4)**
One central VPC peered to specific subnets in two VPCs	You have a central VPC that requires access to two other VPCs. The other VPCs need to be isolated from each other. ● The central VPC has public resources deployed and the other VPCs require access to all resources in the central VPC. ● Other VPCs have multiple subnets and only one in each VPC is used for accessing resources in the central VPC.	IPv4	**One Central VPC Peered to Specific Subnets in Two VPCs (IPv4)**
One central VPC peered to overlapping subnets from two VPCs	This scenario is similar to the preceding one. If two VPCs with overlapping subnets need to peer with the central VPC, traffic may fail to be forwarded to the required destination. To prevent this, plan the network according to this example.	IPv4	**One Central VPC Peered to Overlapping Subnets from Two VPCs (IPv4)**

Two VPCs Peered to Two Subnets in a Central VPC (IPv4)

The central VPC-A has two subnets, Subnet-A01 and Subnet-A02. The subnets are associated with different route tables. You need to create Peering-AB between Subnet-A01 and VPC-B, and Peering-AC between Subnet-A02 and VPC-C. VPC-B and VPC-C have the same CIDR block. However, there will be no route conflicts because the two subnets in VPC-A are associated with different route tables.

- For details about resource planning, see **Table 10-34**.
- For details about VPC peering relationships, see **Table 10-35**.

Figure 10-11 Networking diagram (IPv4)

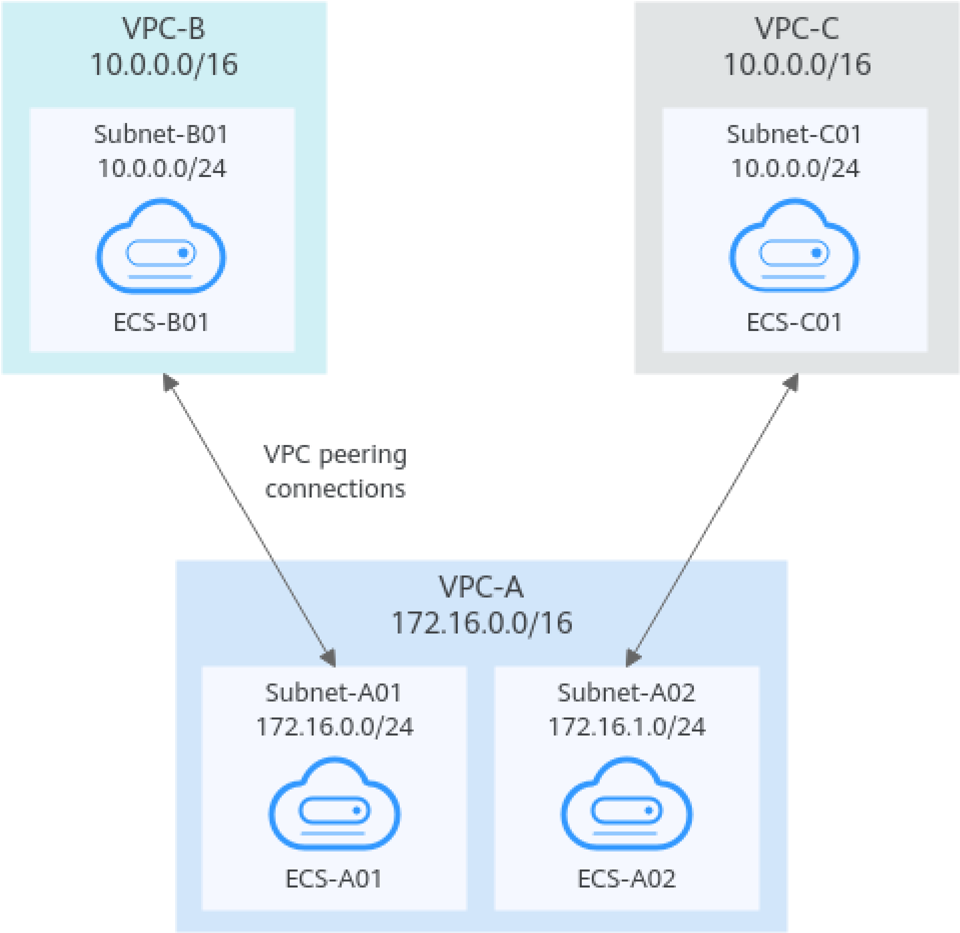


Table 10-34 Resource planning details (IPv4)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	VPC Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	172.16.0.0/24	rtb-VPC-A01	ECS-A01	sg-web: general-purpose web server	172.16.0.111
		Subnet-A02	172.16.1.0/24	rtb-VPC-A02	ECS-A02		172.16.1.91
VPC-B	10.0.0.0/16	Subnet-B01	10.0.0.0/24	rtb-VPC-B	ECS-B01		10.0.0.139
VPC-C	10.0.0.0/16	Subnet-C01	10.0.0.0/24	rtb-VPC-C	ECS-C01		10.0.0.71

NOTE

VPC-A has two route tables. Route table rtb-VPC-A01 is associated with Subnet-A01, and route table rtb-VPC-A02 is associated with Subnet-A02. The two subnets can communicate with each other.

Table 10-35 Peering relationships (IPv4)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
Subnet-A01 of VPC-A is peered to VPC-B.	Peering-AB	VPC-A	VPC-B
Subnet-A02 of VPC-A is peered to VPC-C.	Peering-AC	VPC-A	VPC-C

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-36 VPC route table details (IPv4)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A01	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.1.0/24	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop.
rtb-VPC-A02	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.1.0/24	Local	System	
	10.0.0.0/16 (VPC-C)	Peering-AC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-AC as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.

Route Table	Destination	Next Hop	Route Type	Description
	172.16.0.0/24 (Subnet-A01)	Peering-AB	Custom	Add a route with the CIDR block of Subnet-A01 as the destination and Peering-AB as the next hop.
rtb-VPC-C	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.1.0/24 (Subnet-A02)	Peering-AC	Custom	Add a route with the CIDR block of Subnet-A02 as the destination and Peering-AC as the next hop.

Two VPCs Peered to Two Subnets in a Central VPC (IPv6/IPv4)

The central VPC-A has two subnets, Subnet-A01 and Subnet-A02. The subnets are associated with different route tables. You need to create Peering-AB between Subnet-A01 and VPC-B for IPv6 communication, and Peering-AC between Subnet-A02 and VPC-C for IPv4 communication. VPC-B and VPC-C have the same CIDR block. However, there will be no route conflicts because the two subnets in VPC-A are associated with different route tables.

- For details about resource planning, see **Table 10-37**.
- For details about VPC peering relationships, see **Table 10-38**.

Figure 10-12 Networking diagram (IPv6/IPv4)

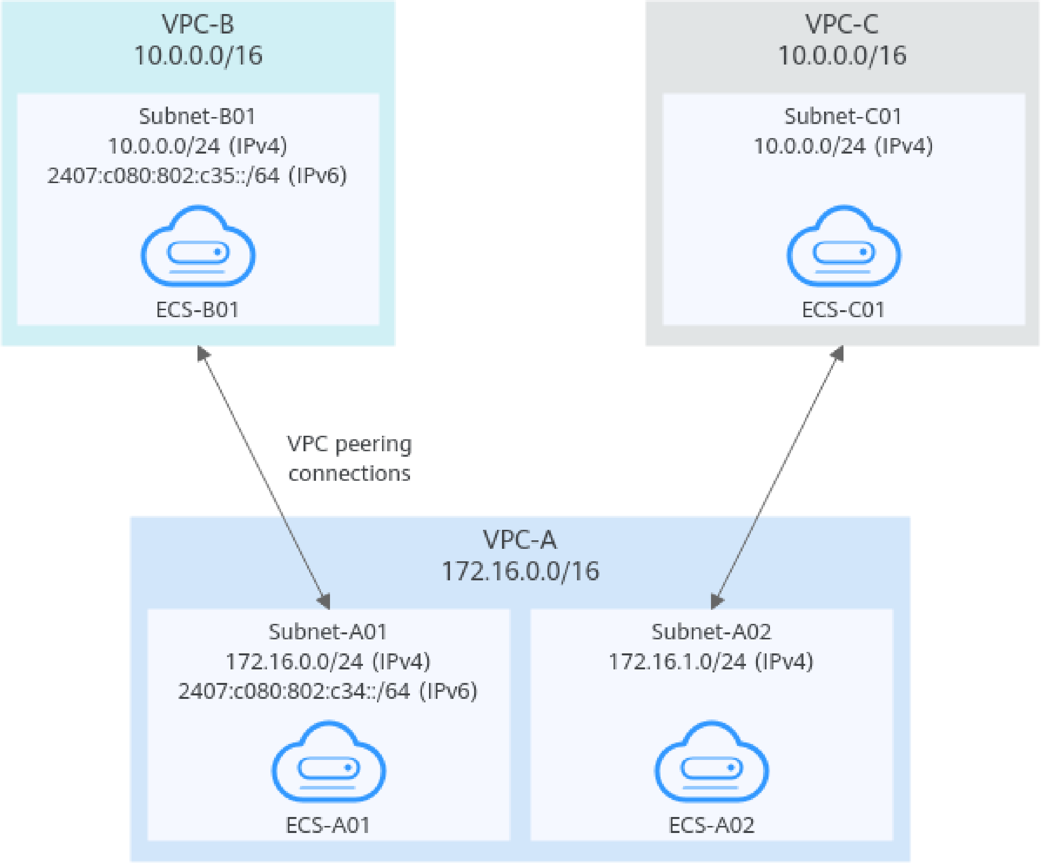


Table 10-37 Resource planning details (IPv6/IPv4)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	VPC Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	● IPv4: 172.16.0.0/24 ● IPv6: 2407:c080:802:c34::/64	rtb-VPC-A01	ECS-A01	sg-web: general-purpose web server	● IPv4: 172.16.0.111 ● IPv6: 2407:c080:802:c34:a925:f12e:cfa0:8edb
		Subnet-A02	172.16.1.0/24	rtb-VPC-A02	ECS-A02		172.16.1.91

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	VPC Route Table	ECS Name	Security Group	Private IP Address
VPC-B	10.0.0.0/16	Subnet-B01	• IPv4: 10.0.0.0/24 • IPv6: 2407:c080:802:c35::/64	rtb-VPC-B	ECS-B01		• IPv4: 10.0.0.139 • IPv6: 2407:c080:802:c35:493:33f4:4531:5162
VPC-C	10.0.0.0/16	Subnet-C01	10.0.0.0/24	rtb-VPC-C	ECS-C01		10.0.0.71

NOTE

VPC-A has two route tables. Route table rtb-VPC-A01 is associated with Subnet-A01, and route table rtb-VPC-A02 is associated with Subnet-A02. The two subnets can communicate with each other.

Table 10-38 Peering relationships (IPv6/IPv4)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
Subnet-A01 of VPC-A is peered to VPC-B. (IPv6)	Peering-AB	VPC-A	VPC-B
Subnet-A02 of VPC-A is peered to VPC-C. (IPv4)	Peering-AC	VPC-A	VPC-C

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-39 VPC route table details (IPv6/IPv4)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A01	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c34::/64	Local	System	
	172.16.1.0/24	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop for IPv4 communication.
	2407:c080:802:c35::/64 (Subnet-B01)	Peering-AB	Custom	Add a route with the IPv6 CIDR block of Subnet-B01 as the destination and Peering-AB as the next hop for IPv6 communication.
rtb-VPC-A02	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c34::/64	Local	System	
	172.16.1.0/24	Local	System	
	10.0.0.0/16 (VPC-C)	Peering-AC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-AC as the next hop for IPv4 communication.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	2407:c080:802:c35::/64	Local	System	
	172.16.0.0/24 (Subnet-A01)	Peering-AB	Custom	Add a route with the CIDR block of Subnet-A01 as the destination and Peering-AB as the next hop for IPv4 communication.
	2407:c080:802:c34::/64 (Subnet-A01)	Peering-AB	Custom	Add a route with the IPv6 CIDR block of Subnet-A01 as the destination and Peering-AB as the next hop for IPv6 communication.

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-C	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.1.0/24 (Subnet-A02)	Peering-AC	Custom	Add a route with the CIDR block of Subnet-A02 as the destination and Peering-AC as the next hop for IPv4 communication.

One Central VPC Peered to Specific Subnets in Two VPCs (IPv4)

You need to create Peering-AB between central VPC-A and Subnet-B01 in VPC-B, and Peering-AC between central VPC-A and Subnet-C02 in VPC-C. VPC-B and VPC-C have the same CIDR block, but the CIDR blocks of Subnet-B01 and Subnet-C02 do not overlap. Therefore, there will be no route conflicts.

- For details about resource planning, see **Table 10-40**.
- For details about VPC peering relationships, see **Table 10-41**.

Figure 10-13 Networking diagram (IPv4)

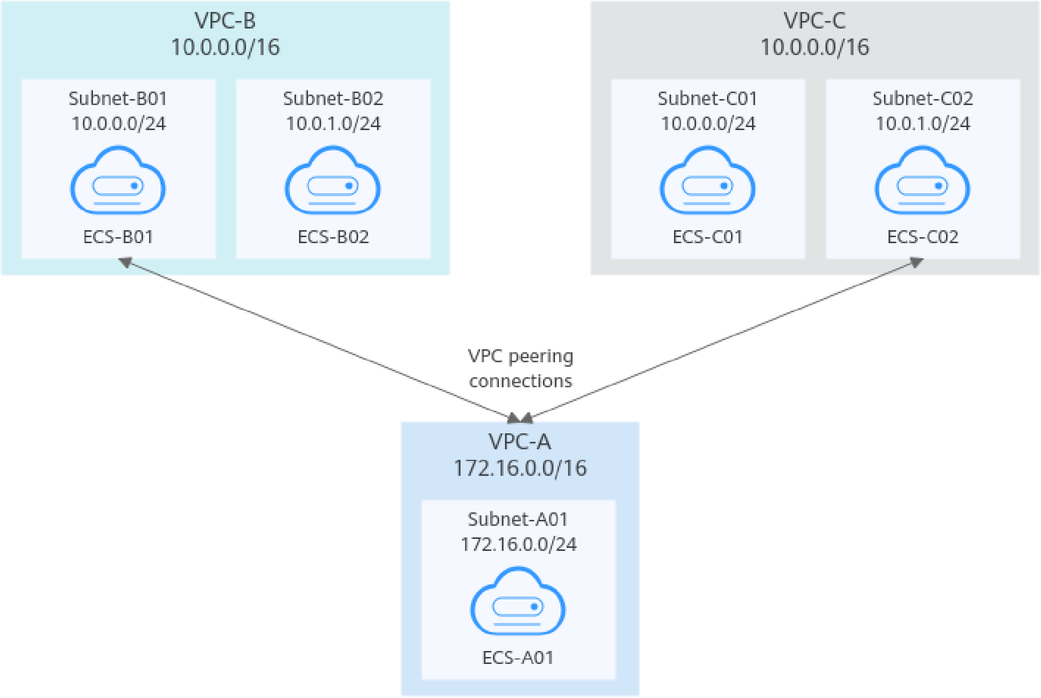


Table 10-40 Resource planning details (IPv4)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	VPC Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	172.16.0.0/24	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	172.16.0.111
VPC-B	10.0.0.0/16	Subnet-B01	10.0.0.0/24	rtb-VPC-B	ECS-B01		10.0.0.139
		Subnet-B02	10.0.1.0/24	rtb-VPC-B	ECS-B02		10.0.1.167
VPC-C	10.0.0.0/16	Subnet-C01	10.0.0.0/24	rtb-VPC-C	ECS-C01		10.0.0.71
		Subnet-C02	10.0.1.0/24	rtb-VPC-C	ECS-C02		10.0.1.116

Table 10-41 Peering relationships (IPv4)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered to Subnet-B01 of VPC-B.	Peering-AB	VPC-A	VPC-B
VPC-A is peered to Subnet-C02 of VPC-C.	Peering-AC	VPC-A	VPC-C

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-42 VPC route table details (IPv4)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.

Route Table	Destination	Next Hop	Route Type	Description
	10.0.0.0/24 (Subnet-B01)	Peering-AB	Custom	Add a route with the CIDR block of Subnet-B01 as the destination and Peering-AB as the next hop.
	10.0.1.0/24 (Subnet-C02)	Peering-AC	Custom	Add a route with the CIDR block of Subnet-C02 as the destination and Peering-AC as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop.
rtb-VPC-C	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AC	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AC as the next hop.

One Central VPC Peered to Overlapping Subnets from Two VPCs (IPv4)

If you want to create VPC peering connections between a VPC and multiple overlapping subnets from different VPCs, ensure that the destinations of the routes added for the peering connections do not conflict and traffic can be correctly forwarded.

In this example, you need to create Peering-AB between central VPC-A and Subnet-B02 in VPC-B, and Peering-AC between central VPC-A and Subnet-C02 in VPC-C. Subnet-B02 and Subnet-C02 have the same CIDR block, and ECS-B02 and ECS-C02 have the same private IP address (10.0.1.167/32).

- For details about resource planning, see **Table 10-43**.
- For details about VPC peering relationships, see **Table 10-44**.

Figure 10-14 Networking diagram (IPv4)

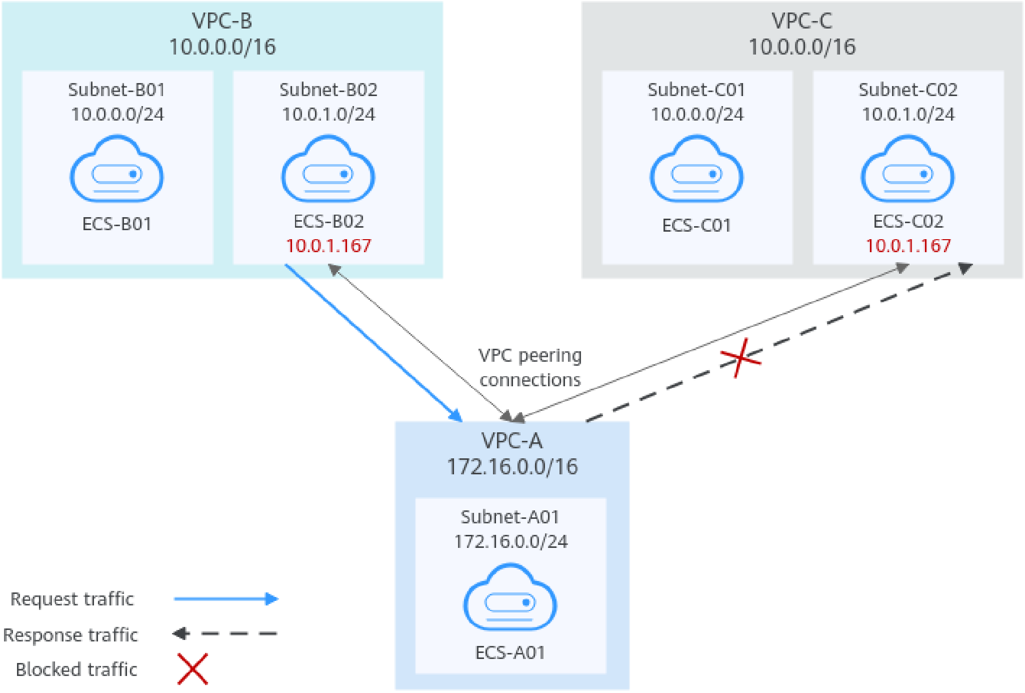


Table 10-43 Resource planning details (IPv4)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	VPC Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	172.16.0.0/24	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	172.16.0.111
VPC-B	10.0.0.0/16	Subnet-B01	10.0.0.0/24	rtb-VPC-B	ECS-B01		10.0.0.139
		Subnet-B02	10.0.1.0/24	rtb-VPC-B	ECS-B02		10.0.1.167
VPC-C	10.0.0.0/16	Subnet-C01	10.0.0.0/24	rtb-VPC-C	ECS-C01		10.0.0.71
		Subnet-C02	10.0.1.0/24	rtb-VPC-C	ECS-C02		10.0.1.167

Table 10-44 Peering relationships (IPv4)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered to Subnet-B02 of VPC-B.	Peering-AB	VPC-A	VPC-B
VPC-A is peered to Subnet-C02 of VPC-C.	Peering-AC	VPC-A	VPC-C

If you add routes to the route tables of the local and peer VPCs according to **Table 10-45**, the response traffic cannot be correctly forwarded. The details are as follows:

1. ECS-B02 in Subnet-B02 of VPC-B sends request traffic to VPC-A through the route with Peering-AB as the next hop in the rtb-VPC-B route table.
2. VPC-A receives the request traffic from ECS-B02 and expects to send the response traffic to ECS-B02. The rtb-VPC-A route table has the route with 10.0.1.167/32 as the destination, but its next hop is Peering-AC. The response traffic is incorrectly sent to VPC-C.
3. ECS-C02 in Subnet-C02 of VPC-C has the same private IP address (10.0.1.167/32) as ECS-B02. The response traffic is incorrectly sent to ECS-C02, and ECS-B02 cannot receive the response traffic.

Table 10-45 VPC route table details (IPv4)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24 (Subnet-C02)	Peering-AC	Custom	Add a route with the CIDR block of Subnet-C02 as the destination and Peering-AC as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop.

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-C	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24	Local	System	
	172.16.0.0/16 (VPC-A)	Peering-AC	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AC as the next hop.

If there are overlapping subnets, configure routes as follows to prevent traffic from being incorrectly forwarded:

- Suggestion 1: In the rtb-VPC-A route table, add a route with Peering-AB as the next hop and the private IP address of ECS-B02 (10.0.1.167/32) as the destination. The route with 10.0.1.167/32 as the destination is preferentially matched based on the longest prefix match rule to ensure that VPC-A sends the response traffic to ECS-B02. For more configurations, see **Using a VPC Peering Connection to Connect ECSs in Two VPCs**.

Table 10-46 VPC route table details

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.167/32 (ECS-B02)	Peering-AB	Custom	Add a route with the private IP address of ECS-B02 as the destination and Peering-AB as the next hop.
	10.0.1.0/24 (Subnet-C02)	Peering-AC	Custom	Add a route with the CIDR block of Subnet-C02 as the destination and Peering-AC as the next hop.

- Suggestion 2: In the rtb-VPC-A route table, change the destination of the route with Peering-AC as the next hop from Subnet-C02 to Subnet-C01. Add a route with Peering-AB as the next hop and Subnet-B02 as the destination to ensure that VPC-A can send the response traffic to Subnet-B02 in VPC-B.

Table 10-47 VPC route table details

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24 (Subnet-B02)	Peering-AB	Custom	Add a route with the CIDR block of Subnet-B02 as the destination and Peering-AB as the next hop.
	10.0.0.0/24 (Subnet-C01)	Peering-AC	Custom	Add a route with the CIDR block of Subnet-C01 as the destination and Peering-AC as the next hop.

10.4 Using a VPC Peering Connection to Connect ECSs in Two VPCs

You can configure a VPC peering connection and set the destination of the routes added to VPC route tables to the private IP address of ECS in the peer VPC. In this way, the two ECS are connected.

To enable traffic forwarding among these ECSs, you need to add routes with private IP addresses of these ECSs as the destinations and a VPC peering connection as the next hop to VPC route tables. **Table 10-48** shows example scenarios.

Table 10-48 Scenario description

Scenario	Scenario Description	IP Address Version	Example
ECS in a central VPC peered to ECSs in two other VPCs	You want a central VPC to communicate with the other two VPCs. However, you do not want the other two VPCs to communicate with each other. The other two VPCs have the same CIDR block and also include subnets that overlap. To prevent route conflicts in the central VPC, you can configure VPC peering connections to connect to specific ECSs in the other two VPCs.	IPv4	**ECS in a Central VPC Peered to ECSs in Two Other VPCs (IPv4)**
A central VPC peered with two other VPCs using longest prefix match	This scenario is similar to the preceding one. In addition to peering specific ECSs, you can create the following VPC peering connections based on the longest prefix match rule: ● Create a VPC peering connection between the central VPC and an ECS in VPC-B ● Create a VPC peering connection between the central VPC and a subnet in VPC-C This configuration expands the communication scope.	IPv4	**A Central VPC Peered with Two Other VPCs Using Longest Prefix Match (IPv4)**

ECS in a Central VPC Peered to ECSs in Two Other VPCs (IPv4)

You want to create a VPC peering connection between VPC-A and VPC-B, and between VPC-A and VPC-C. VPC-B and VPC-C have matching CIDR blocks. You can set the destinations of routes to private IP addresses of specific ECSs to limit traffic to these ECSs. If the destination of a route is not properly planned, traffic cannot be correctly forwarded. For details, see **One Central VPC Peered to Overlapping Subnets from Two VPCs (IPv4)**.

In this example, you need to create Peering-AB between ECS-A01-1 in VPC-A and ECS-B01 in VPC-B, and Peering-AC between ECS-A01-2 in VPC-A and ECS-C01 in VPC-C. Subnet-B01 and Subnet-C01 have matching CIDR blocks. The private IP addresses of ECS-B01 and ECS-C01 must be different. Otherwise, there will be route conflicts because the route table of VPC-A will have routes with the same destination.

- For details about resource planning, see **Table 10-49**.
- For details about VPC peering relationships, see **Table 10-50**.

Figure 10-15 Networking diagram (IPv4)

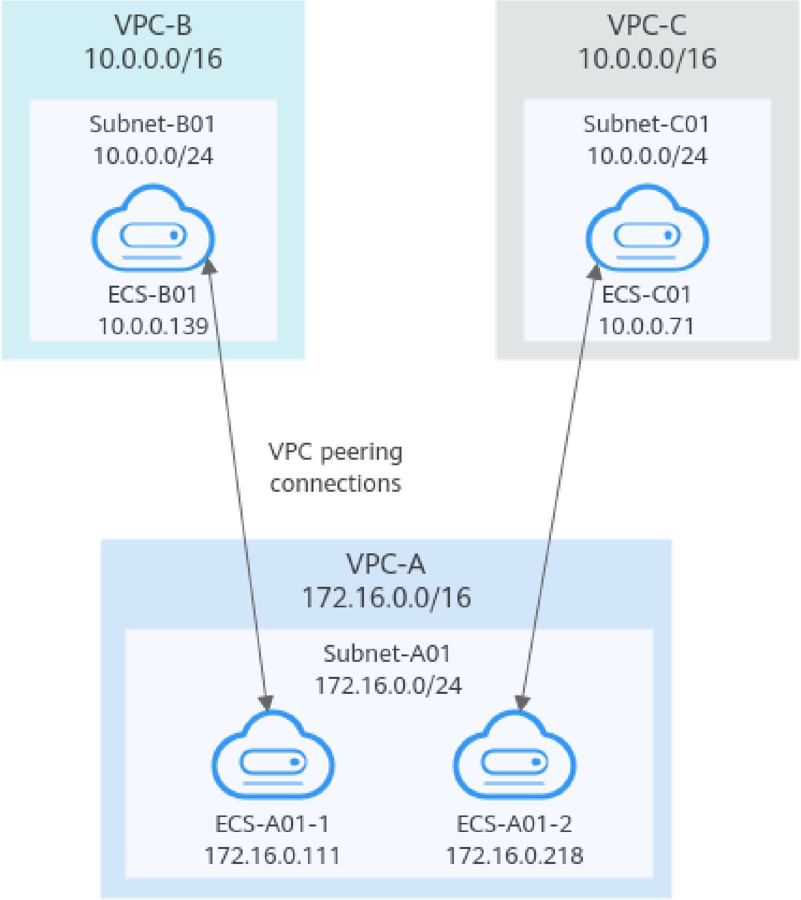


Table 10-49 Resource planning details (IPv4)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	VPC Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	172.16.0.0/24	rtb-VPC-A	ECS-A01-1	sg-web: general-purpose web server	172.16.0.111
					ECS-A01-2		172.16.0.218
VPC-B	10.0.0.0/16	Subnet-B01	10.0.0.0/24	rtb-VPC-B	ECS-B01		10.0.0.139

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	VPC Route Table	ECS Name	Security Group	Private IP Address
VPC-C	10.0.0.0/16	Subnet-C01	10.0.0.0/24	rtb-VPC-C	ECS-C01		10.0.0.71

Table 10-50 Peering relationships (IPv4)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
ECS-A01-1 in VPC-A is peered with ECS-B01 in VPC-B.	Peering-AB	VPC-A	VPC-B
ECS-A01-2 in VPC-A is peered with ECS-C01 in VPC-C.	Peering-AC	VPC-A	VPC-C

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-51 VPC route table details (IPv4)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.0.139/32 (ECS-B01)	Peering-AB	Custom	Add a route with the private IP address of ECS-B01 as the destination and Peering-AB as the next hop.
	10.0.0.71/32 (ECS-C01)	Peering-AC	Custom	Add a route with the private IP address of ECS-C01 as the destination and Peering-AC as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.

Route Table	Destination	Next Hop	Route Type	Description
	172.16.0.111/32 (ECS-A01-1)	Peering-AB	Custom	Add a route with the private IP address of ECS-A01-1 as the destination and Peering-AB as the next hop.
rtb-VPC-C	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.0.218/32 (ECS-A01-2)	Peering-AC	Custom	Add a route with the private IP address of ECS-A01-2 as the destination and Peering-AC as the next hop.

A Central VPC Peered with Two Other VPCs Using Longest Prefix Match (IPv4)

You want to create a VPC peering connection between VPC-A and VPC-B, and between VPC-A and VPC-C. VPC-B and VPC-C have matching CIDR blocks. You can set the destinations of routes to private IP addresses of specific ECSs to limit traffic to these ECSs. If the destination of a route is not properly planned, traffic cannot be correctly forwarded. For details, see **One Central VPC Peered to Overlapping Subnets from Two VPCs (IPv4)**.

In this example, you need to create Peering-AB between central VPC-A and ECS-B01 in VPC-B, and Peering-AC between central VPC-A and VPC-C. Subnet-B01 and Subnet-C01 have matching CIDR blocks. You can use the longest prefix match rule to control traffic forwarding.

- For details about resource planning, see **Table 10-52**.
- For details about VPC peering relationships, see **Table 10-53**.

Figure 10-16 Networking diagram (IPv4)

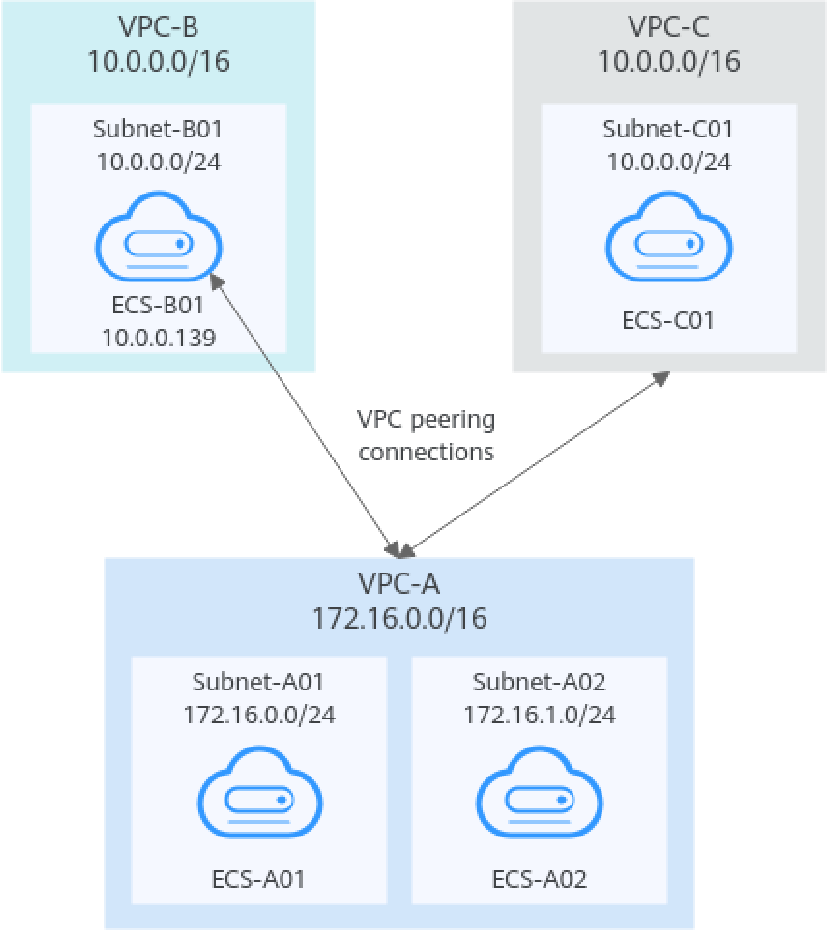


Table 10-52 Resource planning details (IPv4)

VPC Name	VPC CIDR Block	Subnet Name	Subnet CIDR Block	VPC Route Table	ECS Name	Security Group	Private IP Address
VPC-A	172.16.0.0/16	Subnet-A01	172.16.0.0/24	rtb-VPC-A	ECS-A01	sg-web: general-purpose web server	172.16.0.111
		Subnet-A02	172.16.1.0/24	rtb-VPC-A	ECS-A02		172.16.1.91
VPC-B	10.0.0.0/16	Subnet-B01	10.0.0.0/24	rtb-VPC-B	ECS-B01		10.0.0.139
VPC-C	10.0.0.0/16	Subnet-C01	10.0.0.0/24	rtb-VPC-C	ECS-C01		10.0.0.71

Table 10-53 Peering relationships (IPv4)

Peering Relationship	Peering Connection Name	Local VPC	Peer VPC
VPC-A is peered with ECS-B01 in VPC-B.	Peering-AB	VPC-A	VPC-B
VPC-A is peered with VPC-C.	Peering-AC	VPC-A	VPC-C

After the VPC peering connections are created, add the following routes to the route tables of the local and peer VPCs:

Table 10-54 VPC route table details (IPv4)

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	172.16.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.1.0/24	Local	System	
	10.0.0.139/32 (ECS-B01)	Peering-AB	Custom	Add a route with the private IP address of ECS-B01 as the destination and Peering-AB as the next hop.
	10.0.0.0/16 (VPC-C)	Peering-AC	Custom	Add a route with the CIDR block of VPC-C as the destination and Peering-AC as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop.
rtb-VPC-C	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	172.16.0.0/16 (VPC-A)	Peering-AC	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AC as the next hop.

10.5 Unsupported VPC Peering Configurations

Scenarios

The VPC peering connection configurations are not supported in **Table 10-55**.

Table 10-55 Scenarios that VPC peering connections are invalid

Scenario	Example
- If VPCs with the same CIDR block also include subnets that overlap, VPC peering connections are not usable. - If two VPCs have overlapping CIDR blocks but some of their subnets do not overlap, you cannot create a VPC peering connection to connect specific subnets that do not overlap.	**Invalid VPC Peering for Overlapping VPC CIDR Blocks** - **VPCs with the same CIDR block also include subnets that overlap.** - **Two VPCs have overlapping CIDR blocks but some of their subnets do not overlap.**
VPC peering connections cannot enable ECSs in their VPCs to share an EIP to access the Internet. If VPC-A and VPC-B are peered and ECS-A01 in VPC-A has an EIP, ECS-B01 in VPC-B cannot access the Internet using the EIP bound to ECS-A01.	**Invalid VPC Peering for Sharing an EIP**

Invalid VPC Peering for Overlapping VPC CIDR Blocks

If two VPCs have overlapping CIDR blocks, the VPC peering connection may not take effect due to route conflicts. The following describes the reasons and configuration suggestions.

- VPCs with the same CIDR block also include subnets that overlap.

  VPC peering connections are not usable. As shown in **Table 10-56**, VPC-A and VPC-B, and their subnets have the same CIDR block. If you create a VPC peering connection between VPC-A and VPC-B, their route tables are shown in **Table 10-56**.

  In the rtb-VPC-A route table, the custom route for routing traffic from VPC-A to VPC-B and the local route have overlapping destinations. The local route has a higher priority and traffic will be forwarded within VPC-A and cannot reach VPC-B.

Figure 10-17 Networking diagram (IPv4)

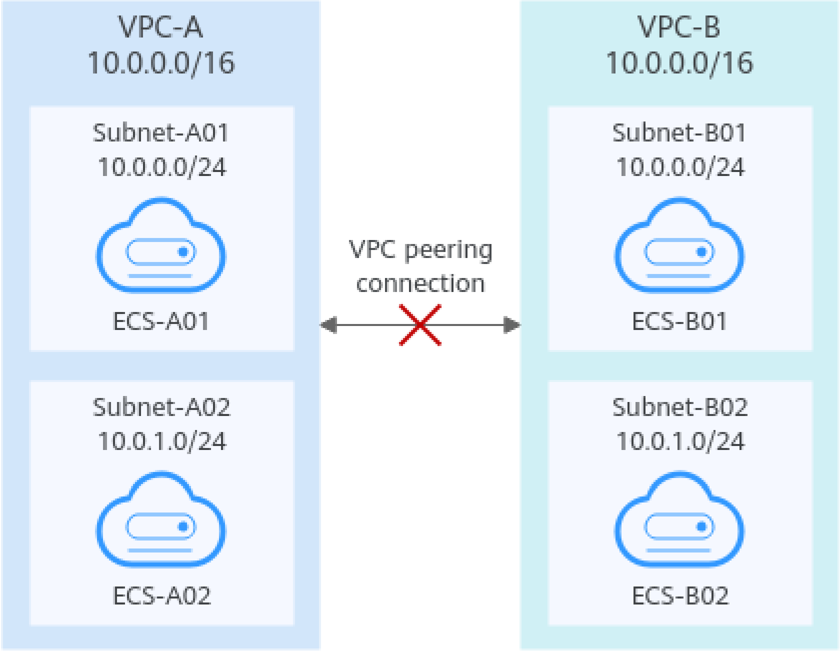
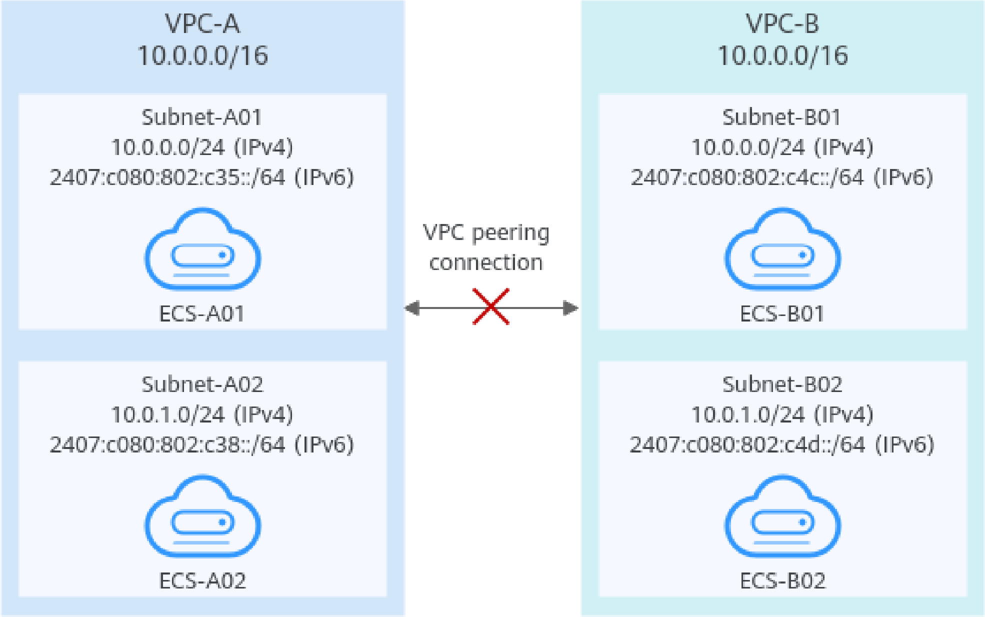


Table 10-56 VPC route table details

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24	Local	System	
	10.0.0.0/16 (VPC-B)	Peering-AB	Custom	Add a route with the CIDR block of VPC-B as the destination and Peering-AB as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24	Local	System	
	10.0.0.0/16 (VPC-A)	Peering-AB	Custom	Add a route with the CIDR block of VPC-A as the destination and Peering-AB as the next hop.

If two VPCs want to use their IPv6 CIDR blocks for communication by a VPC peering connection but the IPv4 CIDR blocks of the VPCs or subnets overlap, the connection is not usable.

Figure 10-18 Networking diagram (IPv6)



- Two VPCs have overlapping CIDR blocks but some of their subnets do not overlap.

  VPC peering connections will not take effect in the following scenarios:

  - Connecting overlapping CIDR blocks of VPCs

    As shown in **Figure 10-19**, if you create a VPC peering connection between VPC-A and VPC-B, the VPC peering connection will not take effect because the two VPCs have the same CIDR block.

  - Connecting overlapping subnets from different VPCs

    If you create a VPC peering connection between Subnet-A01 and Subnet-B02, the route tables are shown in **Table 10-57**. In the rtb-VPC-B route table, the custom route for routing traffic from Subnet-B02 to Subnet-A01 and the local route have overlapping destinations. The local route has a higher priority and traffic will be forwarded within Subnet-B02 and cannot reach Subnet-A01.

Figure 10-19 Networking diagram (IPv4)

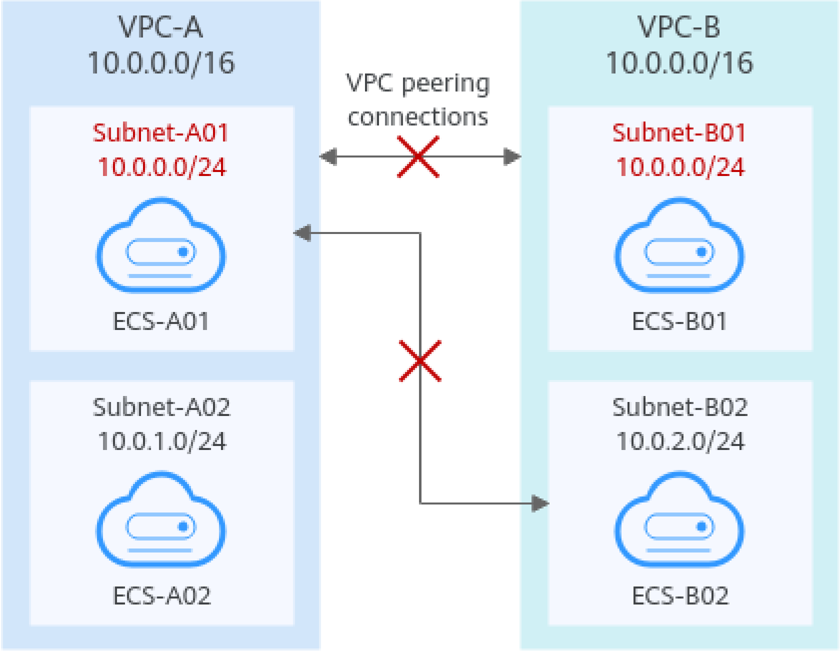
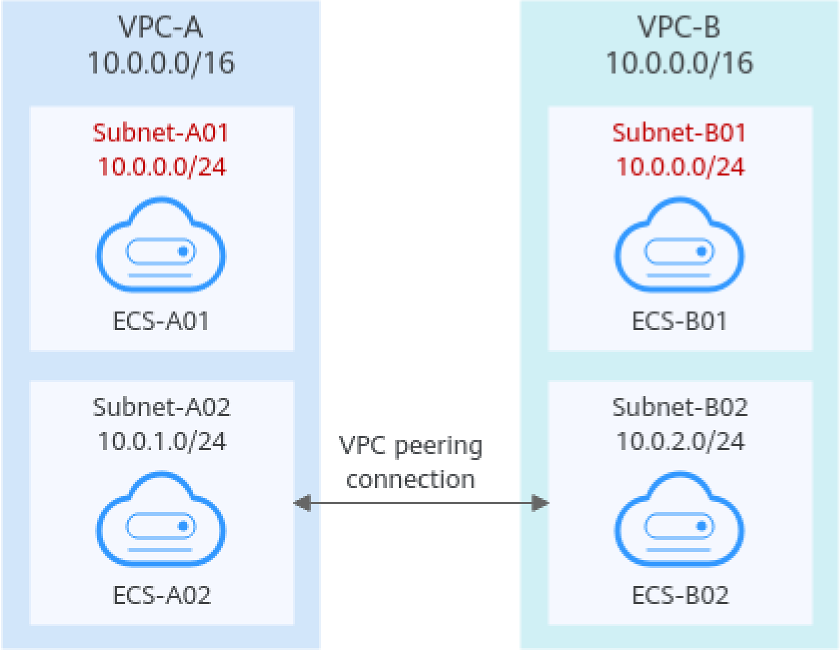


Table 10-57 VPC route table details

Route Table	Destination	Next Hop	Route Type	Description
rtb-VPC-A	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.1.0/24	Local	System	
	10.0.2.0/24 (Subnet-B02)	Peering-AB	Custom	Add a route with the CIDR block of Subnet-B02 as the destination and Peering-AB as the next hop.
rtb-VPC-B	10.0.0.0/24	Local	System	Local routes are automatically added for communications within a VPC.
	10.0.2.0/24	Local	System	
	10.0.0.0/24 (Subnet-A01)	Peering-AB	Custom	Add a route with the CIDR block of Subnet-A01 as the destination and Peering-AB as the next hop.

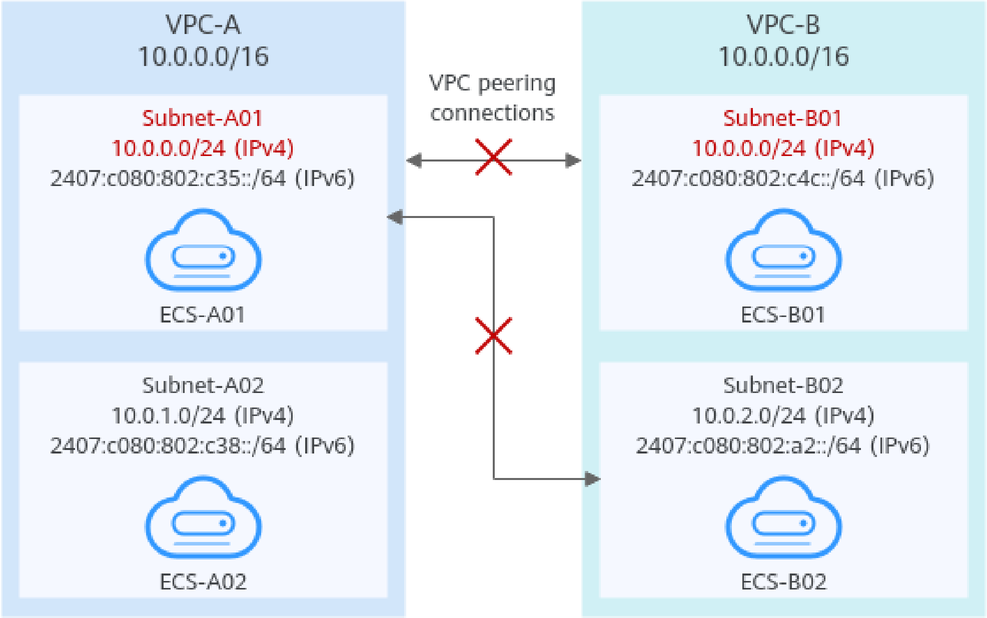
If the subnets connected by a VPC peering connection do not overlap, the connection will take effect. As shown in **Figure 10-20**, you can create a VPC peering connection between Subnet-A02 and Subnet-B02. In this case, the routes do not conflict and the VPC peering connection takes effect.

Figure 10-20 Networking diagram (IPv4)



If two VPCs want to use their IPv6 CIDR blocks for communication by a VPC peering connection but the IPv4 CIDR blocks of the VPCs or subnets overlap, the connection is not usable.

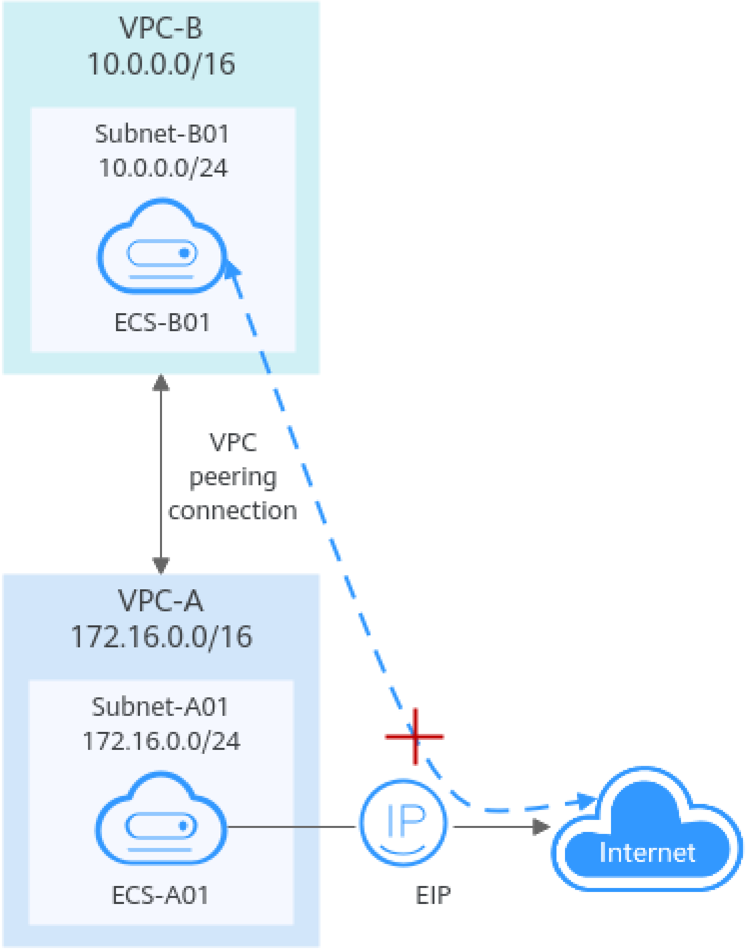
Figure 10-21 Networking diagram (IPv6)



Invalid VPC Peering for Sharing an EIP

As shown in **Figure 10-22**, although VPC-A and VPC-B are peered and ECS-A01 in VPC-A has an EIP, ECS-B01 in VPC-B cannot access the Internet using the EIP bound to ECS-A01.

Figure 10-22 Networking diagram


VPC-B
10.0.0.0/16
Subnet-B01
10.0.0.0/24
ECS-B01
VPC
peering
connection
VPC-A
172.16.0.0/16
Subnet-A01
172.16.0.0/24
ECS-A01
IP
EIP
Internet