Flat Network Tutorial

Version: ZStack 3.10.0

Issue: V3.10.0

ZStack

Copyright Statement

Trademark

Shanghai Yunzhou Information and Technology Ltd. reserves all rights to its trademarks, including , but not limited to ZStack and other trademarks in connection with Shanghai Yunzhou Information and Technology Ltd.

Other trademarks or registered trademarks presented in this document are owned or controlled solely by its proprietaries.

Notice

The products, services, or features that you purchased are all subject to the commercial contract and terms of Shanghai Yunzhou Information and Technology Ltd., but any part or all of the foregoing displayed in this document may not be in the scope of your purchase or use. Unless there are additional conventions, Shanghai Yunzhou Information and Technology Ltd. will disclaim any statement or warranty, whether implicit or explicit, on the contents of this document.

In an event of product version upgrades or other reasons, the contents of this document will be irregularly updated and released. Unless there are additional conventions, this document, considered solely as a reference guide, will not make any warranty, whether implicit or explicit, on all the statements, information, or suggestions.

Contents

1 Introduction

This Tutorial provides information about how to use and configure flat networks and a variety of practices to common usage scenarios.

A flat network is a network that does not offer any segmentation options. Any VM instance that is attached to this network can see the same broadcast traffic and can communicate with each other without requiring a router. A flat network has the following attributes:

- Hosts and VM instances reside on the same layer 2 broadcast domain.
- Services such as User Data, EIP, DHCP, and security group are provided.
- Both the distributed EIP and the distributed DHCP can avoid a single point of failure (SPOF). Hence, when high concurrency occurs, the entire concurrency of the system can be improved effectively.

A flat network provides the following network services:

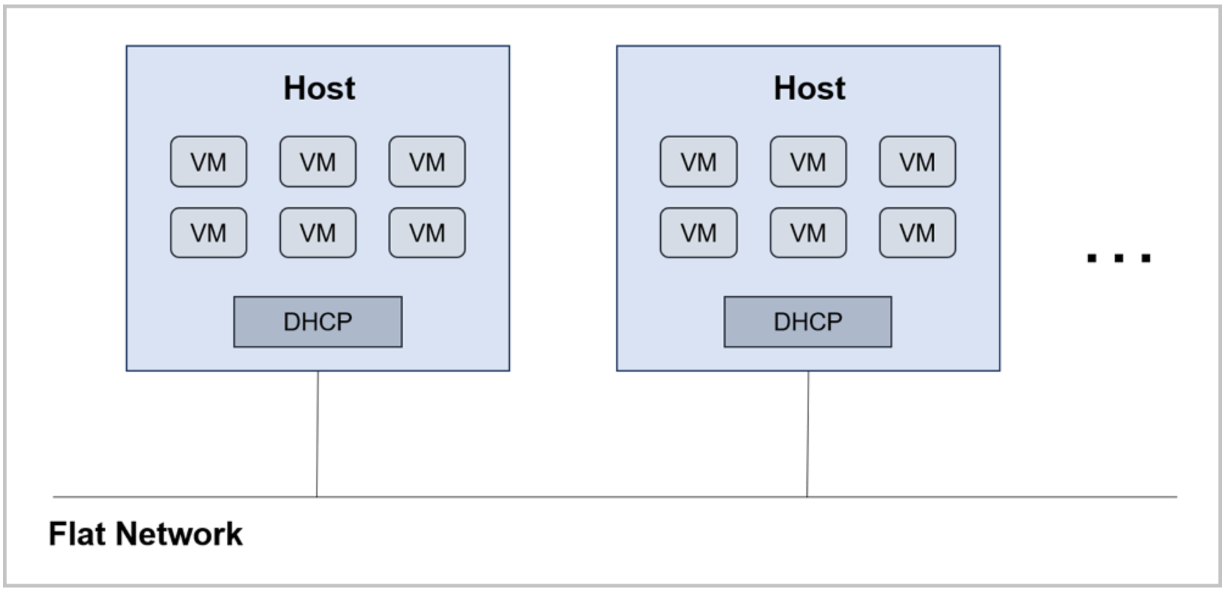
- User Data: Enable you to start, attach or perform specific operations for a VM instance by using `cloud-init`, such as injecting `ssh-key`.
- EIP: Refer to elastic IP addresses that can be used to reach internal, private networks via public networks.
- DHCP: Obtain IP addresses dynamically, which is achieved by the distributed DHCP.

  > **Note:**
  >
  > The DHCP service includes the DNS feature.

- Security group:
  - The security group network module provides the security group service.
  - VM firewalls can be securely controlled with iptables.

The flat network topology is shown in *Flat Network Topology*.

Figure 1-1: Flat Network Topology



2 Prerequisite

In this Tutorial, assume that you have installed the latest ZStack, and complete the basic cloud initialization, including adding a zone, cluster, host, backup storage, primary storage, and other basic resources. For more information, see installation and deployment topics and Wizard configuration topics in *User Guide*.

This Tutorial mainly describes basic deployments and typical usage scenarios of flat networks.

3 Basic Deployment

ZStack supports IPv4 flat networks, IPv6 flat networks, and public networks. This section describes basic deployments of IPv4 flat networks and IPv6 flat networks.

3.1 IPv4 Flat Network Deployment

IPv4, known as Internet Protocol version 4 which defines IP addresses in a 32-bit format, is the most popular Internet Protocol version across the globe. ZStack flat networks support the IPv4 protocol. This topic mainly describes the basic deployment of IPv4 flat networks.

Context

Assume that your environment is as follows:

Table 3-1: IPv4 Flat Network Configuration

Flat Network	Configurations
NIC	em01
VLAN ID	non-VLAN
IP range	*172.20.108.40-172.20.108.50*
Subnet mask	*255.255.0.0*
Gateway	*172.20.0.1*
DHCP IP	*172.20.180.41*

To create an IPv4 flat network

1. Create an L2 network corresponded by an IPv4 flat network, and attach this L2 network to the corresponding cluster.
2. Create an L3 network corresponded by an IPv4 flat network, and enter the corresponding IP range, subnet mask, gateway, and DNS.
3. Create a VM instance by using this IPv4 flat network.
4. Validate the connectivity of this IPv4 flat network.

Procedure

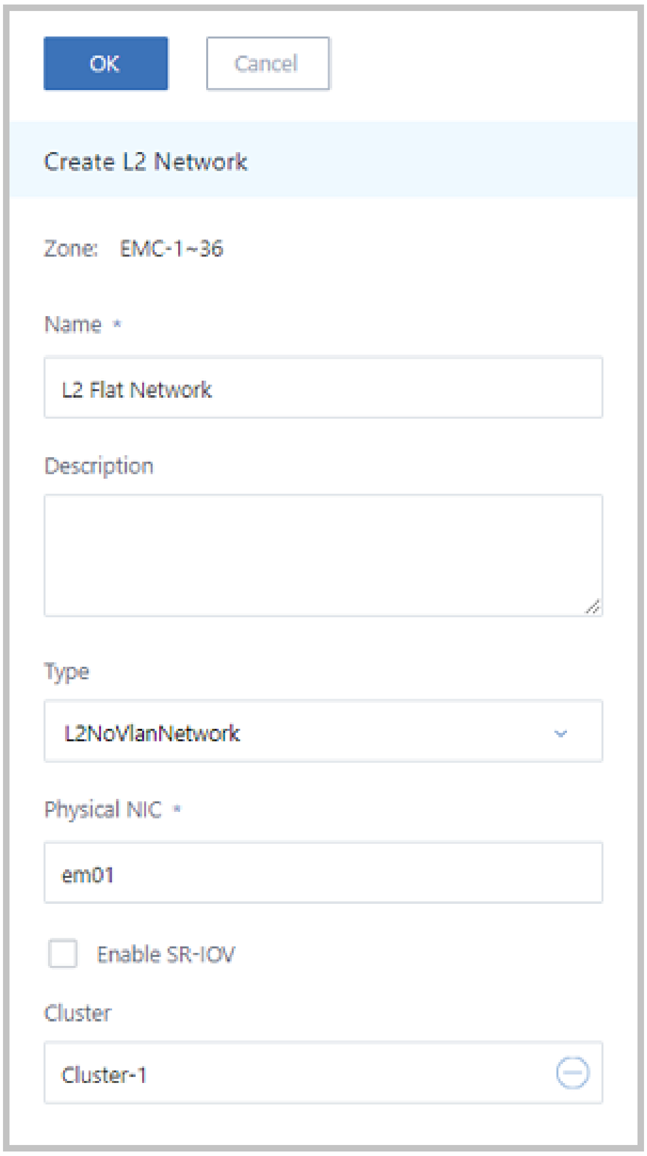
1. Create an L2 network corresponded by an IPv4 flat network, and attach this L2 network to the corresponding cluster.

In the navigation pane of the ZStack Private Cloud UI, choose **Network Resource** > **L2 Network Resource** > **L2 Network**. On the **L2 Network** page, click **Create L2 Network**. On the displayed **Create L2 Network** page, set the following parameters by referring to *IPv4 Flat Network Configuration*:

- **Name**: Enter a name for the L2 flat network.
- **Description**: Optional. Enter a description for the L2 flat network.
- **Type**: Select L2NoVlanNetwork.
- **Physical NIC**: Enter a name for the physical NIC, such as em01.
- **Enable SR-IOV**: Choose whether to enable SR-IOV.
  - By default, this checkbox is unchecked, which means that SR-IOV is disabled for the L2 flat network and is also disabled for the corresponding L3 network.
  - If checked, SR-IOV is enabled. You can also enable SR-IOV for the corresponding L3 network. In this case, make sure that the physical NICs used by the L2 flat network are virtually split via SR-IOV.
- **Cluster**: Select a cluster, such as Cluster-1.

Click **OK** to complete creating the L2 flat network, as shown in *Create L2 Flat Network*.

Figure 3-1: Create L2 Flat Network



OK Cancel

Create L2 Network

Zone: EMC-1~36

Name *

L2 Flat Network

Description

Type

L2NoVlanNetwork

Physical NIC *

em01

□ Enable SR-IOV

Cluster

Cluster-1

2. Create an L3 network corresponded by an IPv4 flat network, and enter the corresponding IP range, subnet mask, gateway, and DNS.

   In the navigation pane of the ZStack Private Cloud UI, choose **Network Resource** > **L3 Network** > **Private Network**. On the **Private Network** page, click **Create Private Network**. On the displayed **Create Private Network** page, set the following parameters by referring to *IPv4 Flat Network Configuration*:

   - **Name**: Enter a name for the L3 flat network.
   - **Description**: Optional. Enter a description for the L3 flat network.
   - **L2 Network**: Select an L2 flat network that you created.

- **Stop DHCP server**: Choose whether to enable the DHCP service.

> **Note:**
>
> - By default, this checkbox is unchecked, which means that the DHCP service is enabled , and IP addresses will be automatically assigned to VM instances. In this case, you can customize a DHCP IP address, or let the system randomly specify a DHCP IP address.
> - If selected, the DHCP service will be disabled, which means that VM instances that use this network cannot obtain IP addresses automatically, and need to be configured manually with IP addresses. In that case, you cannot customize the DHCP IP address. In addition, the system cannot randomly specify a DHCP IP address.

- **Network Type**: Select Flat network.
- **Add IP Range**: Select IPv4 and IP Range.

> **Note:**
>
> ZStack supports two types of IP addresses: IPv4 and IPv6. A network range can be IP range or CIDR. For the purpose of this Tutorial, IPv4 address and IP range are taken as an example.

- **Start IP**: Enter a start IP address, such as *172.20.108.40*.
- **End IP**: Enter an end IP address, such as *172.20.108.50*.
- **Netmask**: Enter a netmask, such as *255.255.0.0*.
- **Gateway**: Enter a gateway such as *172.20.0.1*.
- **DHCP IP**: Optional. Enter a DHCP IP address as needed.

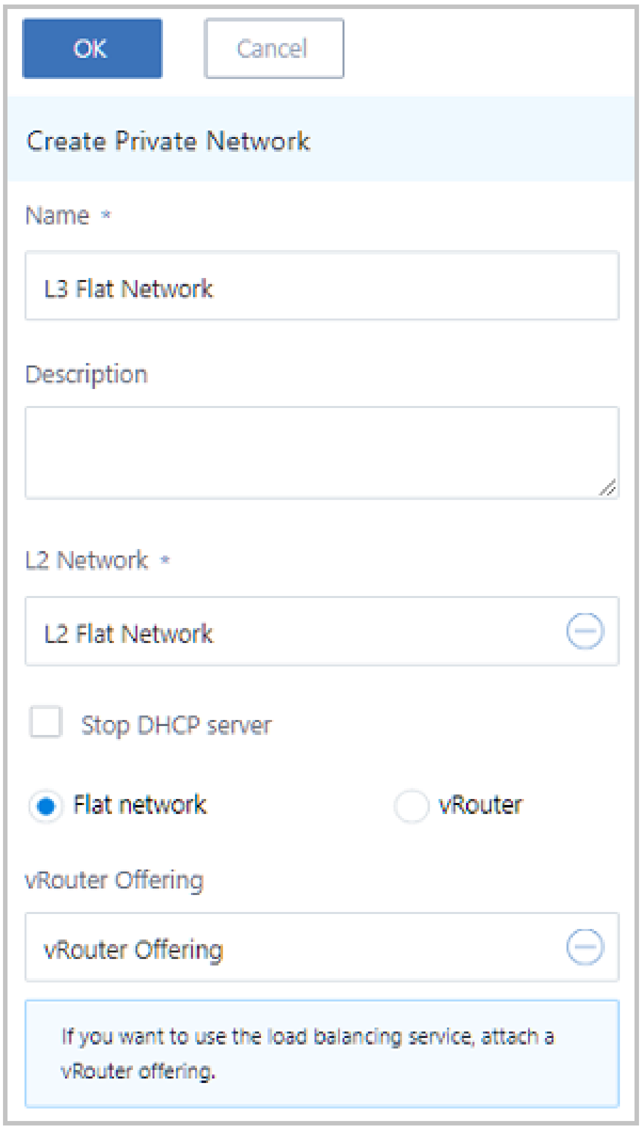
> **Note:**
>
> - If you create an L3 network and enable the DHCP service for the first time, or if you add the first IP range for the L3 network that has enabled the DHCP service, you can customize the DHCP IP address.
> - If the L3 network has a DHCP IP address, you cannot customize the DHCP IP address when you add an IP range.
> - The DHCP IP address can be included or excluded in or from the added IP range. However, the DHCP IP address must be within the CIDR to which the added IP range belongs, and must not be occupied.

- The IP range specified within the start IP address and end IP address cannot contain IP addresses of the link-local address (169.254.0.0/16).
- If not specified, the system will randomly specify an IP address within the IP range that you added.

- **DNS**: Optional. Enter a DNS, such as *114.114.114.114*.

Click **OK** to complete creating the L3 flat network, as shown in *Create L3 Flat Network*.

Figure 3-2: Create L3 Flat Network



Create Private Network

Name *

L3 Flat Network

Description

L2 Network *

L2 Flat Network

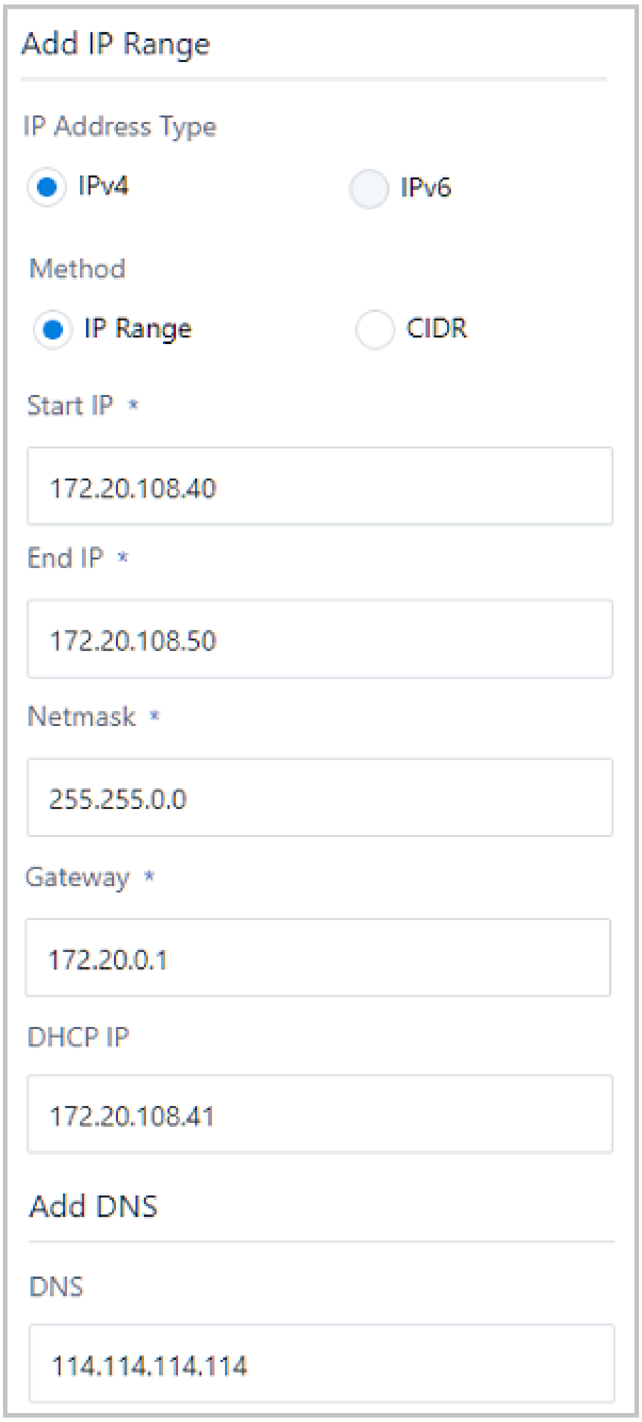
□ Stop DHCP server

● Flat network ○ vRouter

vRouter Offering

vRouter Offering

If you want to use the load balancing service, attach a vRouter offering.



Add IP Range

IP Address Type

● IPv4 ○ IPv6

Method

● IP Range ○ CIDR

Start IP *

172.20.108.40

End IP *

172.20.108.50

Netmask *

255.255.0.0

Gateway *

172.20.0.1

DHCP IP

172.20.108.41

Add DNS

DNS

114.114.114.114

3. Create a Private Cloud VM instance by using this IPv4 flat network.

   In the navigation pane of the ZStack Private Cloud UI, choose **Resource Pool** > **VM Instance**. On the **VM Instance** page, click **Create VM Instance**. On the displayed **Create VM Instance** page, set the following parameters:

   - **Add Type**: Select Single.

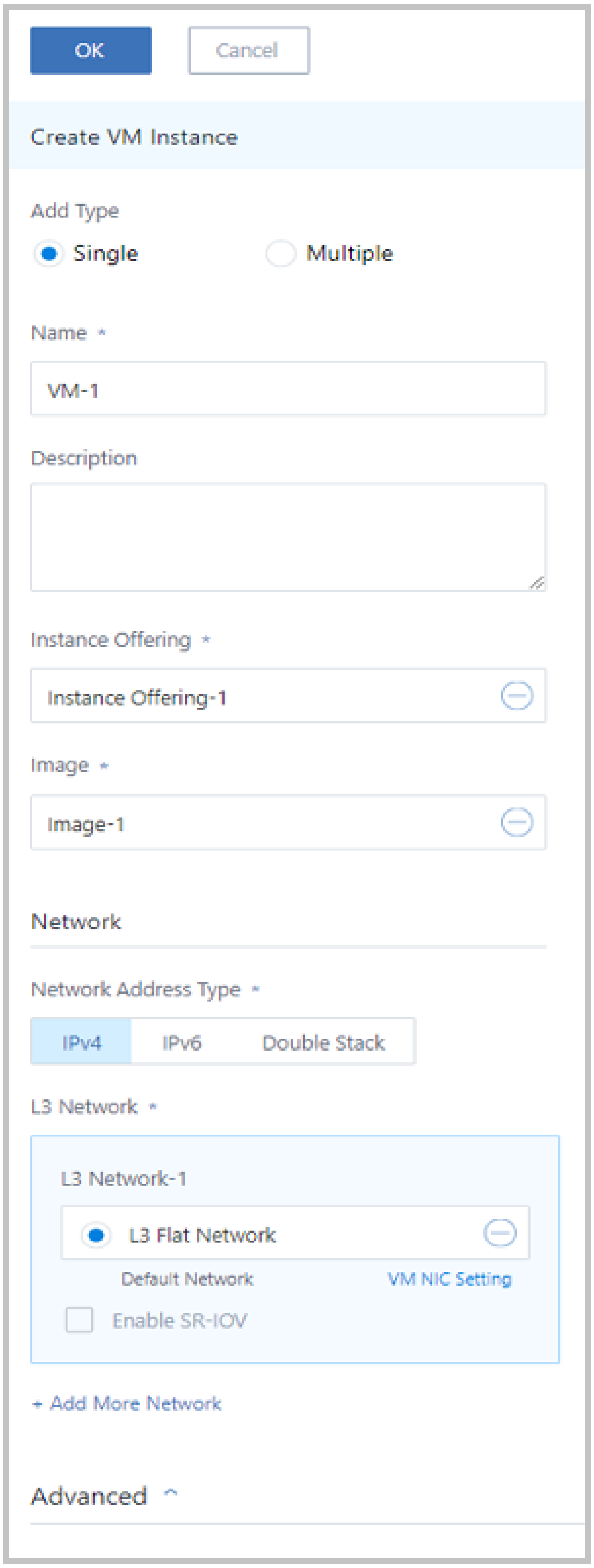
Note:

To create VM instances in bulk, select **Multiple**, and enter the VM count.

- **Name**: Enter a name for the Private Cloud VM instance, such as VM-1.
- **Description**: Optional. Enter a description for the VM instance.
- **Instance Offering**: Select an instance offering that you selected.
- **Image**: Select an image that you added.
- **Network**: Select an IPv4 flat network.

Click **OK** to complete creating the Private Cloud VM instance, as shown in *Create VM-1*.

Figure 3-3: Create VM-1



Create VM Instance

Add Type

● Single ○ Multiple

Name *

VM-1

Description

Instance Offering *

Instance Offering-1

Image *

Image-1

Network

Network Address Type *

[✓ IPv4] IPv6 Double Stack

L3 Network *

L3 Network-1

● L3 Flat Network

Default Network VM NIC Setting

□ Enable SR-IOV

Similarly, use the flat network to create another VM instance, such as VM-2.

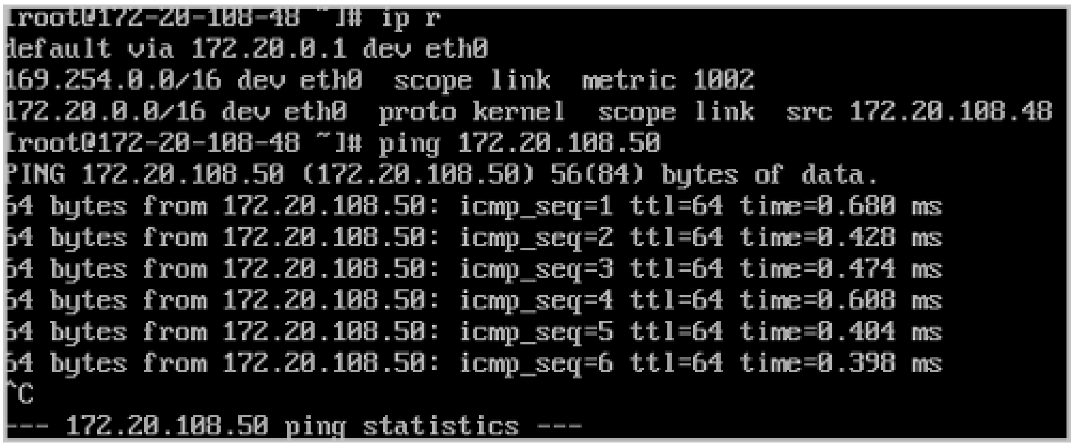
4. **4.** Validate the connectivity of this IPv4 flat network.

Expected result: The two VM instances (such as VM-1 and VM-2) on the same network range can communicate with each other.

Validate the connectivity:

- Log in to VM-1 and validate whether VM-1 can `ping` VM-2, as shown in *VM-1 Pings VM-2*.

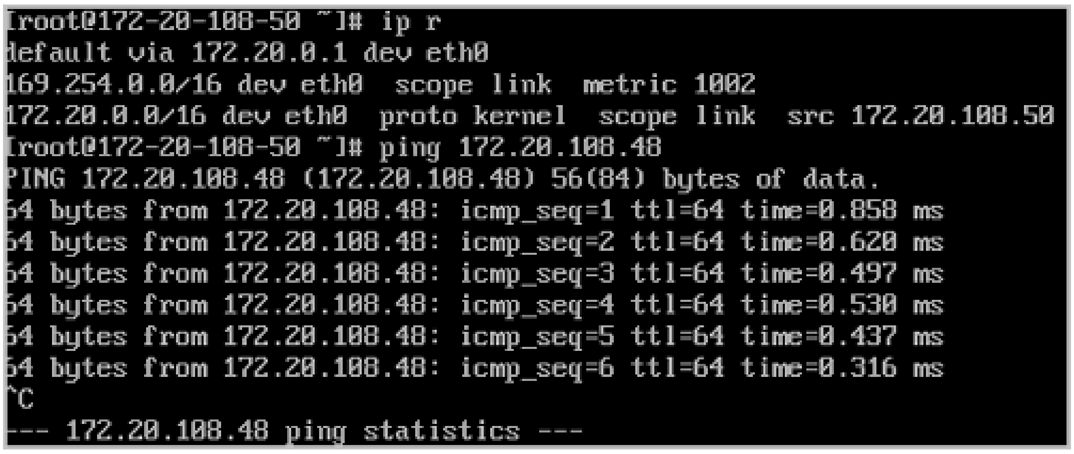
Figure 3-4: VM-1 Pings VM-2



```
[root@172-20-108-48 ~]# ip r
default via 172.20.0.1 dev eth0
169.254.0.0/16 dev eth0  scope link  metric 1002
172.20.0.0/16 dev eth0  proto kernel  scope link  src 172.20.108.48
[root@172-20-108-48 ~]# ping 172.20.108.50
PING 172.20.108.50 (172.20.108.50) 56(84) bytes of data.
64 bytes from 172.20.108.50: icmp_seq=1 ttl=64 time=0.680 ms
64 bytes from 172.20.108.50: icmp_seq=2 ttl=64 time=0.428 ms
64 bytes from 172.20.108.50: icmp_seq=3 ttl=64 time=0.474 ms
64 bytes from 172.20.108.50: icmp_seq=4 ttl=64 time=0.608 ms
64 bytes from 172.20.108.50: icmp_seq=5 ttl=64 time=0.404 ms
64 bytes from 172.20.108.50: icmp_seq=6 ttl=64 time=0.398 ms
^C
--- 172.20.108.50 ping statistics ---
```

- Log in to VM-2 and validate whether VM-2 can `ping` VM-1, as shown in *VM-2 Pings VM-1*.

Figure 3-5: VM-2 Pings VM-1



```
[root@172-20-108-50 ~]# ip r
default via 172.20.0.1 dev eth0
169.254.0.0/16 dev eth0  scope link  metric 1002
172.20.0.0/16 dev eth0  proto kernel  scope link  src 172.20.108.50
[root@172-20-108-50 ~]# ping 172.20.108.48
PING 172.20.108.48 (172.20.108.48) 56(84) bytes of data.
64 bytes from 172.20.108.48: icmp_seq=1 ttl=64 time=0.858 ms
64 bytes from 172.20.108.48: icmp_seq=2 ttl=64 time=0.620 ms
64 bytes from 172.20.108.48: icmp_seq=3 ttl=64 time=0.497 ms
64 bytes from 172.20.108.48: icmp_seq=4 ttl=64 time=0.530 ms
64 bytes from 172.20.108.48: icmp_seq=5 ttl=64 time=0.437 ms
64 bytes from 172.20.108.48: icmp_seq=6 ttl=64 time=0.316 ms
^C
--- 172.20.108.48 ping statistics ---
```

-

So far, we have introduced the basic deployments of the IPv4 flat network.

3.2 IPv6 Flat Network Deployment

IPv6 is Internet Protocol version 6 that defines IP addresses in a 128-bit format. IPv6 resolves the long-anticipated problem of IPv4 address exhaustion, so many devices can be connected to the Internet. ZStack flat networks support the IPv6 protocol. This topic describes the basic deployment of IPv6 flat networks.

Context

Assume that your environment is as follows:

Table 3-2: IPv6 Flat Network Configuration

Flat Network	Configurations
NIC	em1
VLAN ID	2002
IP Range	*234e:0:4567::2-234e:0:4567:0:ffff:ffff:ffff:ffff*
Prefix length	64
Gateway	*234e:0:4567::1*
DHCP IP	*234e:0:4567::3*
DNS	*240c::6644*

To create an IPv6 flat network

1. Create an L2 network corresponded by an IPv6 flat network, and attach this L2 network to the corresponding cluster.
2. Create an L3 network corresponded by the IPv6 flat network.
3. Create two VM instances by using the IPv6 flat network.
4. Obtain IPv6 addresses of the VM instances.
5. Validate the connectivity of the IPv6 flat network.

Procedure

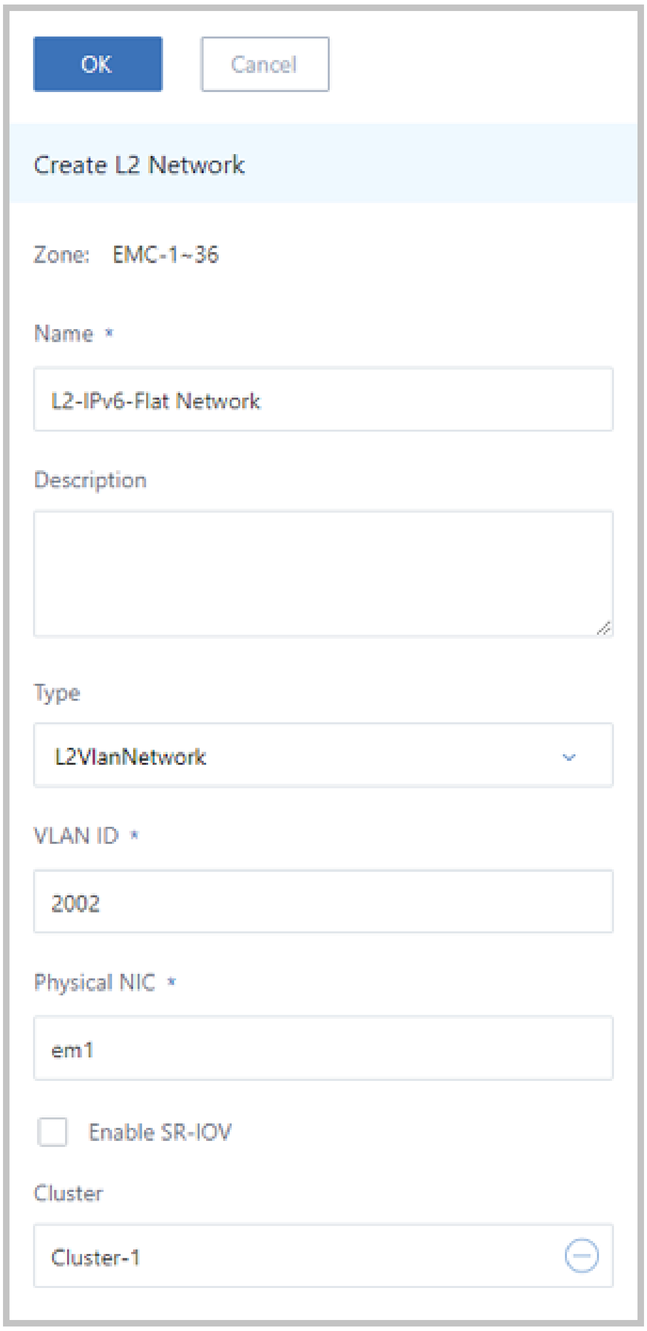
1. Create an L2 network corresponded by an IPv6 flat network, and attach this L2 network to the corresponding cluster.

   In the navigation pane of the ZStack Private Cloud UI, choose **Network Resource** > **L2 Network Resource** > **L2 Network**. On the **L2 Network** page, click **Create L2 Network**. On the displayed **Create L2 Network** page, set the following parameters by referring to *IPv6 Flat Network Configuration*:

- **Name**: Enter a name for the L2 flat network, such as L2-IPv6-Flat Network.
- **Description**: Optional. Enter a description for the L2 flat network.
- **Type**: Select L2VlanNetwork.
- **VLAN ID**: Enter a VLAN ID, such as 2002.
- **Physical NIC**: Enter a name for the physical NIC, such as em1.
- **Enable SR-IOV**: Choose whether to enable SR-IOV. Currently, SR-IOV is not supported by IPv6 L3 networks.
- **Cluster**: Select a cluster attached by the NIC, such as Cluster-1.

Click **OK** to complete creating the L2-IPv6-Flat Network, as shown in *Create L2-IPv6-Flat Network*.

Figure 3-6: Create L2-IPv6-Flat Network



Create L2 Network

Zone: EMC-1~36

Name *

L2-IPv6-Flat Network

Description

Type

L2VlanNetwork

VLAN ID *

2002

Physical NIC *

em1

□ Enable SR-IOV

Cluster

Cluster-1

2. Create an L3 network corresponded by the IPv6 flat network.

   In the navigation pane of the ZStack Private Cloud UI, choose **Network Resource** > **L3 Network** > **Private Network**. On the **Private Network** page, click **Create Private Network**. On the displayed **Create Private Network** page, set the following parameters by referring to *IPv6 Flat Network Configuration*.

   - **Name**: Enter a name for the L3 network, such as L3-IPv6-Flat Network.

- **Description**: Optional. Enter a description for the L3 flat network.
- **L2 Network**: Select an L2 flat network that you created, such as L2-IPv6-Flat Network.
- **Stop DHCP server**: Choose whether to enable the DHCP service.

  Note:

  - By default, this checkbox is unchecked, which means that the DHCP service is enabled , and IP addresses will be automatically assigned to VM instances. In this case, you can customize a DHCP IP address, or let the system randomly specify a DHCP IP address.
  - If selected, the DHCP service will be disabled, which means that VM instances that use this network cannot obtain IP addresses automatically, and need to be configured manually with IP addresses. In that case, you cannot customize the DHCP IP address. In addition, the system cannot randomly specify a DHCP IP address.
- **Network Type**: Select Flat network.
- **vRouter Offering**: Leave this field unspecified. IPv6 flat networks do not support the load balancing service.
- **Add IP Range**: To add a network range, set the following parameters:

  - **IP Address Type**: Select IPv6.
  - **Method**: Select IP Range.

    Note:

    IPv6 supports IP Range and CIDR. The following are the supported formats:

    - Colon hexadecimal notation: *X:X:X:X:X:X:X:X*. Specifically, each X is a 16-bit section that can be represented with hexadecimal digits, such as *234e:0:4567:0:ffff:ffff:ffff:ffff*.
    - Zero compression: If a long range of the number 0 is included in an IPv6 address, this continuous range of 0 can be compressed into **::**. However, to ensure the uniqueness of address resolutions, **::** in the address can only be appeared once, such as *234e:0:4567::2*.
    - CIDR notation: *X:X:X:X:X:X:X:X/N*. Specifically, N represents the prefix length.
  - **Mode**: Select Stateful-DHCP.

    Note:

    IPv6 supports the following IP allocations:

- Stateful-DHCP: Stateful DHCP configurations, which means that interface addresses and other parameters are all configured via the DHCP protocol. Only the IP Range method supports this allocation.
- Stateless-DHCP: Stateless DHCP configurations, which means that interface addresses are automatically deduced via prefixes of routing advertisements, while other parameters are configured via the DHCP protocol.
- SLAAC: Stateless address autoconfigurations, which means that interface addresses are automatically deduced via prefixes of routing advertisements, while other parameters are attached in the routing advertisements.

— **Start IP**: Enter a start IP for the L3 network, such as *234e:0:4567::2*.

— **End IP**: Enter an end IP for the L3 network, such as *234e:0:4567:0:ffff:ffff:ffff:ffff*.

— **Prefix Length**: Enter a prefix length for the L3 network, such as 64. Range: 64-126.

— **Gateway**: Enter a gateway for the L3 network, such as *234e:0:4567::1*.

— **DHCP IP**: Optional. Enter a DHCP IP address, such as *234e:0:4567::3*.

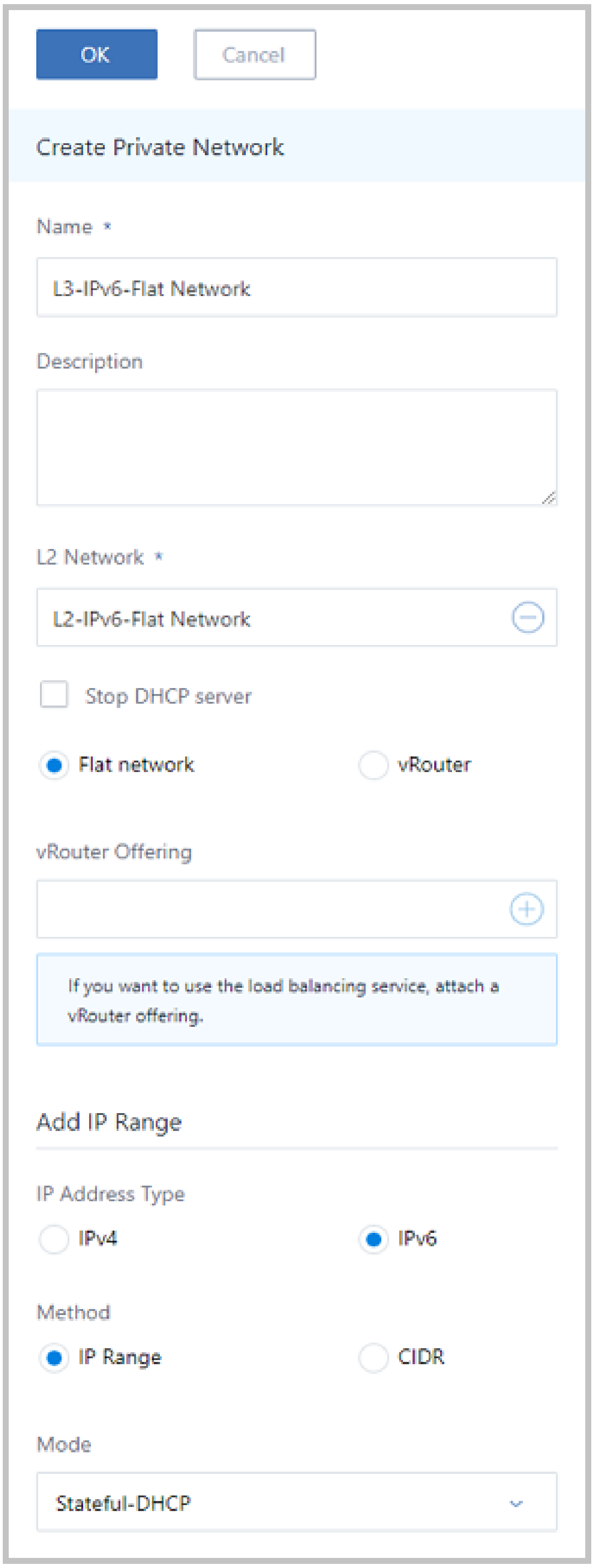
Note:

- If you create an L3 network and enable the DHCP service for the first time, or if you add the first IP range for the L3 network that has enabled the DHCP service, you can customize the DHCP IP address.
- If the L3 network has a DHCP IP address, you cannot customize the DHCP IP address when you add an IP range.
- The DHCP IP address can be included or excluded in or from the added IP range . However, the DHCP IP address must be within the CIDR to which the added IP range belongs, and must not be occupied.
- The IP range specified within the start IP address and end IP address cannot contain IP addresses of the link-local address (169.254.0.0/16).
- If not specified, the system will randomly specify an IP address within the IP range that you added.

- **DNS**: Set a DNS address for the L3 network. If null, the DNS address defaults to *240c::6644* .

Click **OK** to complete creating the L3-IPv6-Flat Network, as shown in *Create L3-IPv6-Flat Network*.

Figure 3-7: Create L3-IPv6-Flat Network



Create Private Network

Name *

L3-IPv6-Flat Network

Description

L2 Network *

L2-IPv6-Flat Network

□ Stop DHCP server

● Flat network ○ vRouter

vRouter Offering

If you want to use the load balancing service, attach a vRouter offering.

Add IP Range

IP Address Type

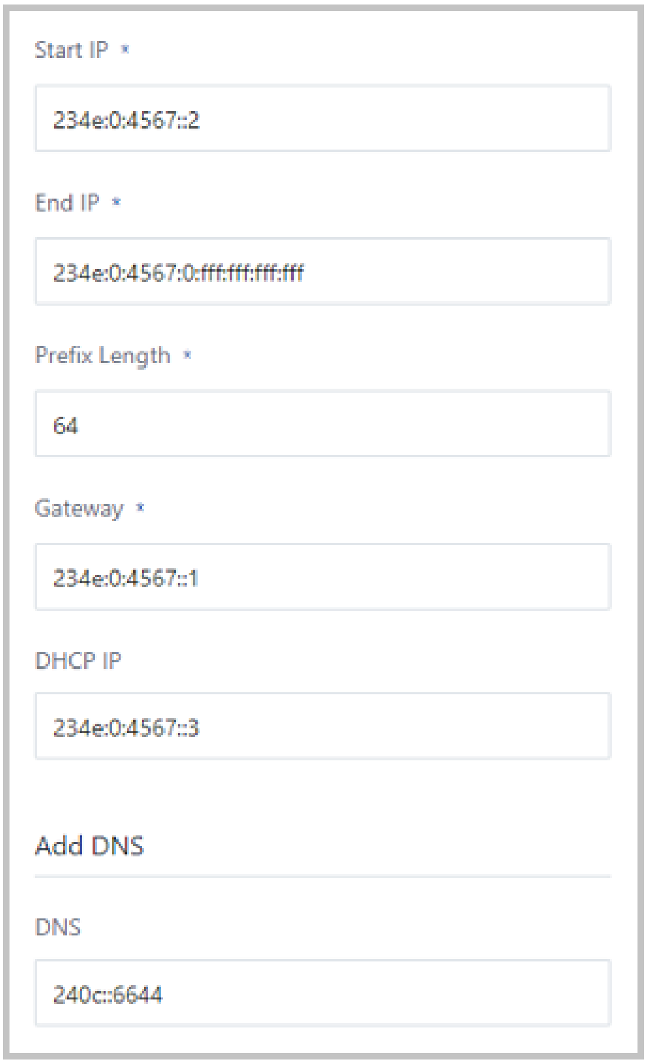
○ IPv4 ● IPv6

Method

● IP Range ○ CIDR

Mode

Stateful-DHCP



Start IP *

234e:0:4567::2

End IP *

234e:0:4567:0:fff:fff:fff:fff

Prefix Length *

64

Gateway *

234e:0:4567::1

DHCP IP

234e:0:4567::3

Add DNS

DNS

240c::6644

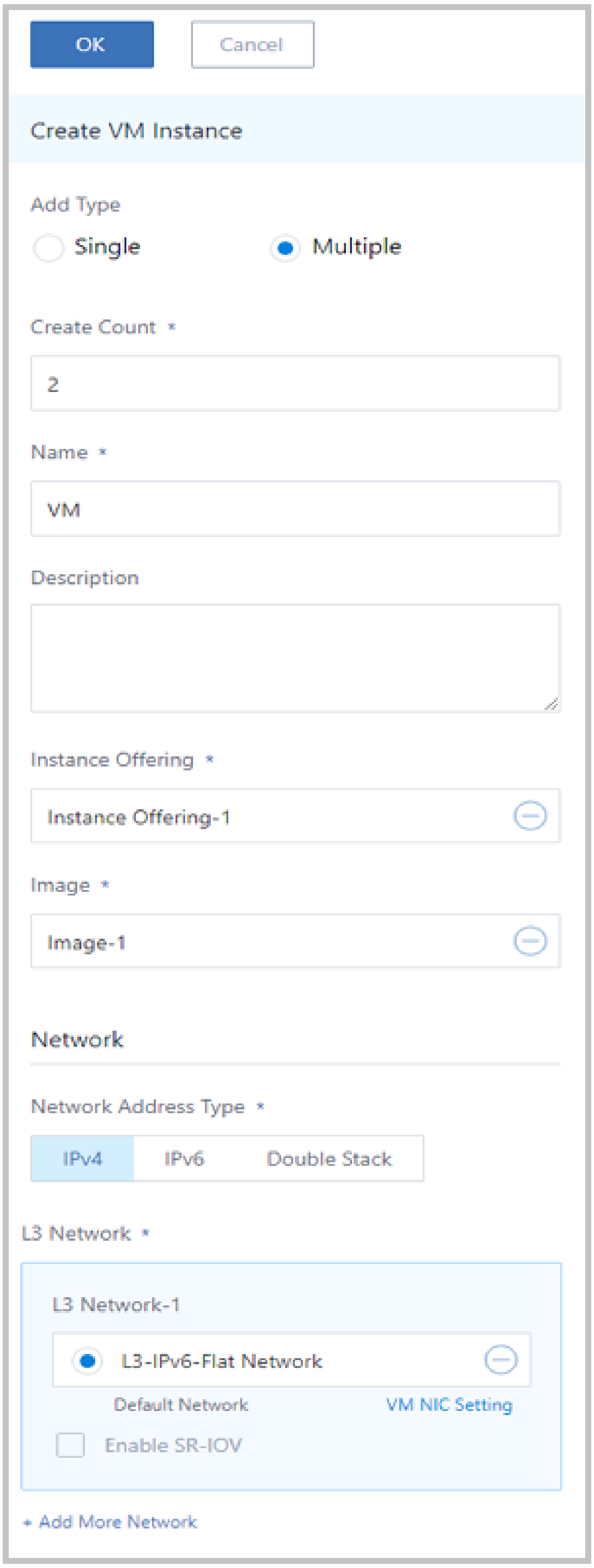
3. Create two VM instances by using the IPv6 flat network.

   In the navigation pane of the ZStack Private Cloud UI, choose **Resource Pool** > **VM Instance**. On the **VM Instance** page, click **Create VM Instance**. On the displayed **Create VM Instance** page, set the following parameters (Take multiple VM instances as an example),

   - **Add Type**: Select Multiple.
   - **Create Count**: Enter 2.
   - **Name**: Enter a name for these two VM instances, such as VM.
   - **Description**: Optional. Enter a description for these VM instances.
   - **Instance Offering**: Select an instance offering for these VM instances.
   - **Image**: Select an image for these VM instances.
   - **Network**: Select the L3 flat network with IPv6, such as L3-IPv6-Flat-Network.

   Click **OK** to complete creating these two VM instances with IPv6 networks, as shown in *Create VM Instance*.

Figure 3-8: Create VM Instance



Create VM Instance

Add Type

○ Single ● Multiple

Create Count *

2

Name *

VM

Description

Instance Offering *

Instance Offering-1

Image *

Image-1

Network

Network Address Type *

IPv4 | IPv6 | Double Stack

L3 Network *

L3 Network-1

● L3-IPv6-Flat Network

Default Network VM NIC Setting

□ Enable SR-IOV

+ Add More Network

4. Obtain IPv6 addresses of the VM instances.

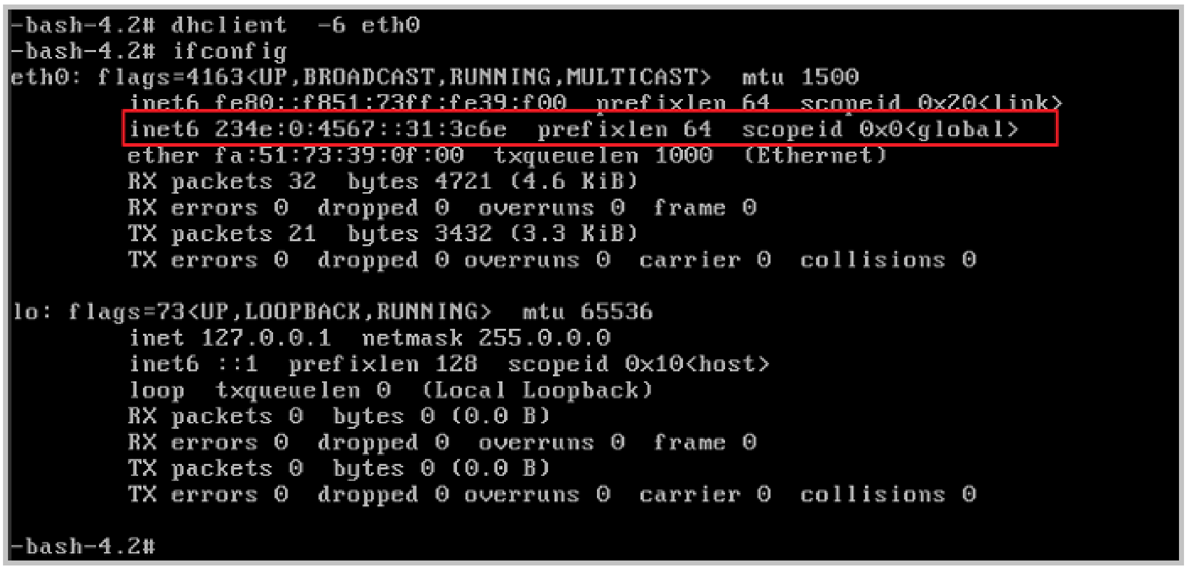
   By default, ZStack can automatically obtain IP addresses for the IPv4 network, while you must manually configure IP addresses for VM instances that use the IPv6 network. Open the consoles of these two VM instances respectively, and run the following commands to obtain IPv6 addresses:

   ```
   -bash-4.2# dhclient -6 eth0  # eth0 is the NIC name.
   ```

   > **Note:**
   >
   > The address that begins with FE80 is the link-local address instead of the expected address.

   Obtain the IPv6 addresses, as shown in *Obtain IPv6 Address*

   Figure 3-9: Obtain IPv6 Address

   ```
   -bash-4.2# dhclient  -6 eth0
   -bash-4.2# ifconfig
   eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
           inet6 fe80::f851:73ff:fe39:f00  prefixlen 64  scopeid 0x20<link>
           inet6 234e:0:4567::31:3c6e  prefixlen 64  scopeid 0x0<global>
           ether fa:51:73:39:0f:00  txqueuelen 1000  (Ethernet)
           RX packets 32  bytes 4721 (4.6 KiB)
           RX errors 0  dropped 0  overruns 0  frame 0
           TX packets 21  bytes 3432 (3.3 KiB)
           TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

   lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
           inet 127.0.0.1  netmask 255.0.0.0
           inet6 ::1  prefixlen 128  scopeid 0x10<host>
           loop  txqueuelen 0  (Local Loopback)
           RX packets 0  bytes 0 (0.0 B)
           RX errors 0  dropped 0  overruns 0  frame 0
           TX packets 0  bytes 0 (0.0 B)
           TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

   -bash-4.2#
   ```

   In this scenario, you will obtain the following IPv6 addresses:

   - VM-1 IP address: *234e:0:4567::63:ab4d*
   - VM-2 IP address: *234e:0:4567::31:3c6e*

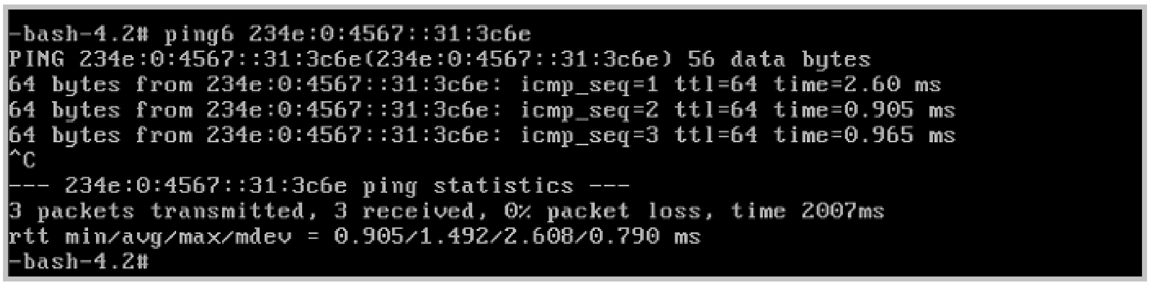
5. Validate the connectivity of the IPv6 flat network.

   Expected result: The two VM instances (such as VM-1 and VM-2) on the same network range can communicate with each other.

   Validate the connectivity:

   - Log in to VM-1 and validate whether VM-1 can `ping` VM-2, as shown in *VM-1 Pings VM-2*.

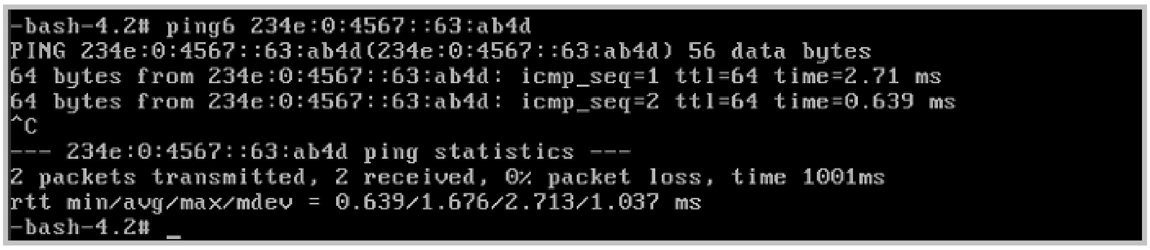
Figure 3-10: VM-1 Pings VM-2



```
-bash-4.2# ping6 234e:0:4567::31:3c6e
PING 234e:0:4567::31:3c6e(234e:0:4567::31:3c6e) 56 data bytes
64 bytes from 234e:0:4567::31:3c6e: icmp_seq=1 ttl=64 time=2.60 ms
64 bytes from 234e:0:4567::31:3c6e: icmp_seq=2 ttl=64 time=0.905 ms
64 bytes from 234e:0:4567::31:3c6e: icmp_seq=3 ttl=64 time=0.965 ms
^C
--- 234e:0:4567::31:3c6e ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 0.905/1.492/2.608/0.790 ms
-bash-4.2#
```

- Log in to VM-2 and validate whether VM-2 can `ping` VM-1, as shown in *VM-2 Pings VM-1*.

Figure 3-11: VM-2 Pings VM-1



```
-bash-4.2# ping6 234e:0:4567::63:ab4d
PING 234e:0:4567::63:ab4d(234e:0:4567::63:ab4d) 56 data bytes
64 bytes from 234e:0:4567::63:ab4d: icmp_seq=1 ttl=64 time=2.71 ms
64 bytes from 234e:0:4567::63:ab4d: icmp_seq=2 ttl=64 time=0.639 ms
^C
--- 234e:0:4567::63:ab4d ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.639/1.676/2.713/1.037 ms
-bash-4.2# _
```

So far, we have introduced the basic deployments of the IPv6 flat network.

4 Usage Scenario

IPv4 flat networks and IPv6 flat networks can all be applied to the following typical usage scenarios:

- IPv4+IPv6 double stack
- L2 connected network
- Security group
- EIP
- Load Balancing
- Hardware SDN

For the purpose of this Tutorial, this section mainly describes IPv4 flat networks as an example.

4.1 IPv4+IPv6 Double Stack

An IPv4+IPv6 double stack is one NIC with two types of IP addresses: IPv4 and IPv6, and takes full advantage of both IPv4 and IPv6. With the IPv4+IPv6 double stack, you can build different business scenarios.

Context

Assume that your environment is as follows:

1. IPv4 range

Table 4-1: IPv4 Range Configuration

Flat Network	Configurations
NIC	em1
VLAN ID	2002
IP range	*192.168.2.2-192.168.2.254*
Subnet mask	*255.255.255.0*
Gateway	*192.168.2.1*
DHCP IP	*192.168.2.3*
DNS	*223.5.5.5*

2. IPv6 range

Table 4-2: IPv6 Range Configuration

Flat Network	Configurations
IP range	*234e:0:4568::2-234e:0:4568:0:ffff:ffff:ffff:ffff*
Prefix length	64
Gateway	*234e:0:4568::1*
DHCP IP	*234e:0:4567::3*
DNS	*240c::6644*

To create an IPv4+IPv6 double stack

1. Create an IPv4 networking environment.
2. Add an IPv6 range.
3. Add an IPv6 DNS address.
4. Create VM instances by using a double stack network.
5. Obtain IPv6 addresses of the VM instances.
6. Validate the network connectivity.

Procedure

1. Create an IPv4 networking environment.

   Create a flat network with an IPv4 address. Assume that the flat network is L3-Flat Network. At this time, this network is an IPv4 network. For more information, see *IPv4 Range Configuration* and *IPv6 Range Configuration*.

   Note:
   You can also create an IPv6 flat network first, and then add an IPv4 range to a flat network.

2. Add an IPv6 range.

   Add an IPv6 range to the existing IPv4 network. Hence, an IPv4+IPv6 double stack network is created.

   On the **Private Network** page, select the IPv4 network, choose **Actions** > **Add IPv6 IP Range**, and set the following parameters:

   - **Method**: Select IP Range. Options: IP Range and CIDR.

     Note:

IPv6 supports IP Range and CIDR. The following are the supported formats:

- Colon hexadecimal notation: *X:X:X:X:X:X:X:X*. Specifically, each X is a 16-bit section that can be represented with hexadecimal digits, such as *234e:0:4567:0:ffff:ffff:ffff:ffff*.
- Zero compression: If a long range of the number 0 is included in an IPv6 address, this continuous range of 0 can be compressed into **::**. However, to ensure the uniqueness of address resolutions, **::** in the address can only be appeared once, such as *234e:0:4567 ::2*.
- CIDR notation: *X:X:X:X:X:X:X:X/N*. Specifically, N represents the prefix length.

- **Mode**: Select Stateful-DHCP.

Note:

IPv6 supports the following IP allocations:

- Stateful-DHCP: Stateful DHCP configurations, which means that interface addresses and other parameters are all configured via the DHCP protocol. Only the IP Range method supports this allocation.
- Stateless-DHCP: Stateless DHCP configurations, which means that interface addresses are automatically deduced via prefixes of routing advertisements, while other parameters are configured via the DHCP protocol.
- SLAAC: Stateless address autoconfigurations, which means that interface addresses are automatically deduced via prefixes of routing advertisements, while other parameters are attached in the routing advertisements.

- **Start IP**: Enter a start IP for the IPv4 network, such as *234e:0:4568::2*.
- **End IP**: Enter an end IP for the IPv4 network, such as *234e:0:4568:0:ffff:ffff:ffff:ffff*.
- **Prefix Length**: Enter a prefix length for the IPv4 network, such as 64. Range: 64-126.
- **Gateway**: Enter a gateway for the IPv4 network, such as *234e:0:4568::1*.
- **DHCP IP**: Optional. Enter a DHCP IP address, such as *234e:0:4568::3*.

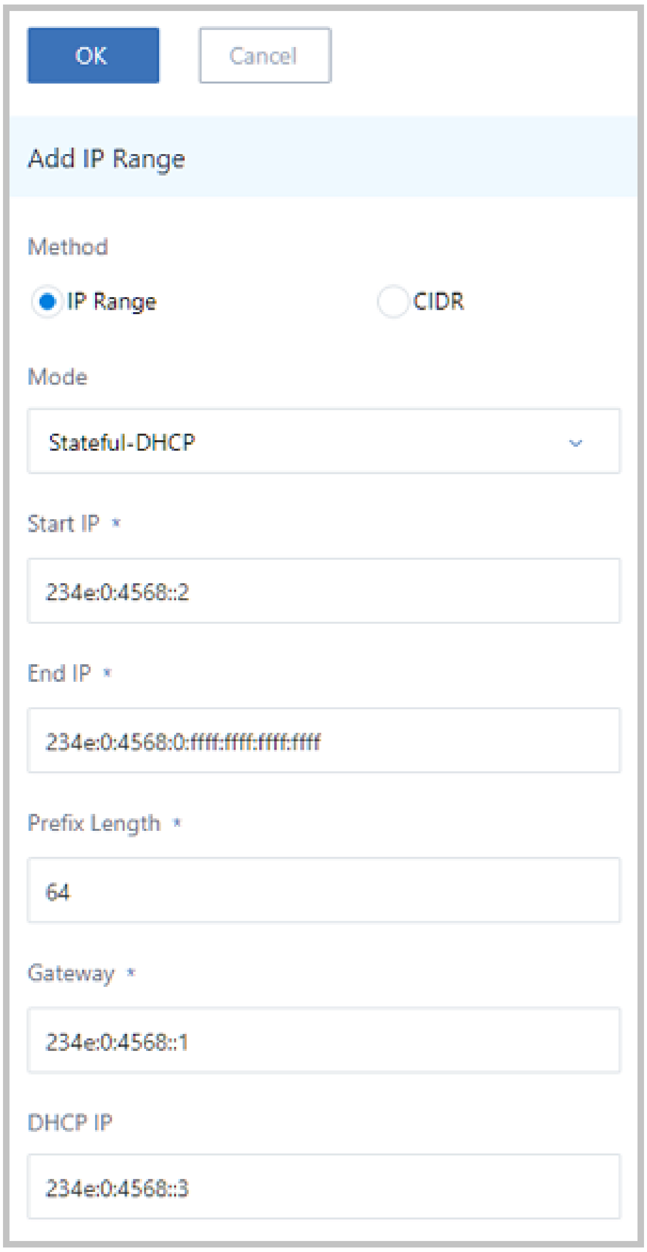
Note:

- If you create an L3 network and enable the DHCP service for the first time, or if you add the first IP range for the L3 network that has enabled the DHCP service, you can customize the DHCP IP address.
- If the L3 network has a DHCP IP address, you cannot customize the DHCP IP address when you add an IP range.

- The DHCP IP address can be included or excluded in or from the added IP range. However, the DHCP IP address must be within the CIDR to which the added IP range belongs, and must not be occupied.
- The IP range specified within the start IP address and end IP address cannot contain IP addresses of the link-local address (169.254.0.0/16).
- If not specified, the system will randomly specify an IP address within the IP range that you added.

Click **OK** to complete adding the IPv6 range for the IPv4 flat network, as shown in *Add IPv6 Range*. Hence, this network has changed into an IPv4+IPv6 network.

Figure 4-1: Add IPv6 Range



Add IP Range

Method

● IP Range ○ CIDR

Mode

Stateful-DHCP

Start IP *

234e:0:4568::2

End IP *

234e:0:4568:0:ffff:ffff:ffff:ffff

Prefix Length *

64

Gateway *

234e:0:4568::1

DHCP IP

234e:0:4568::3

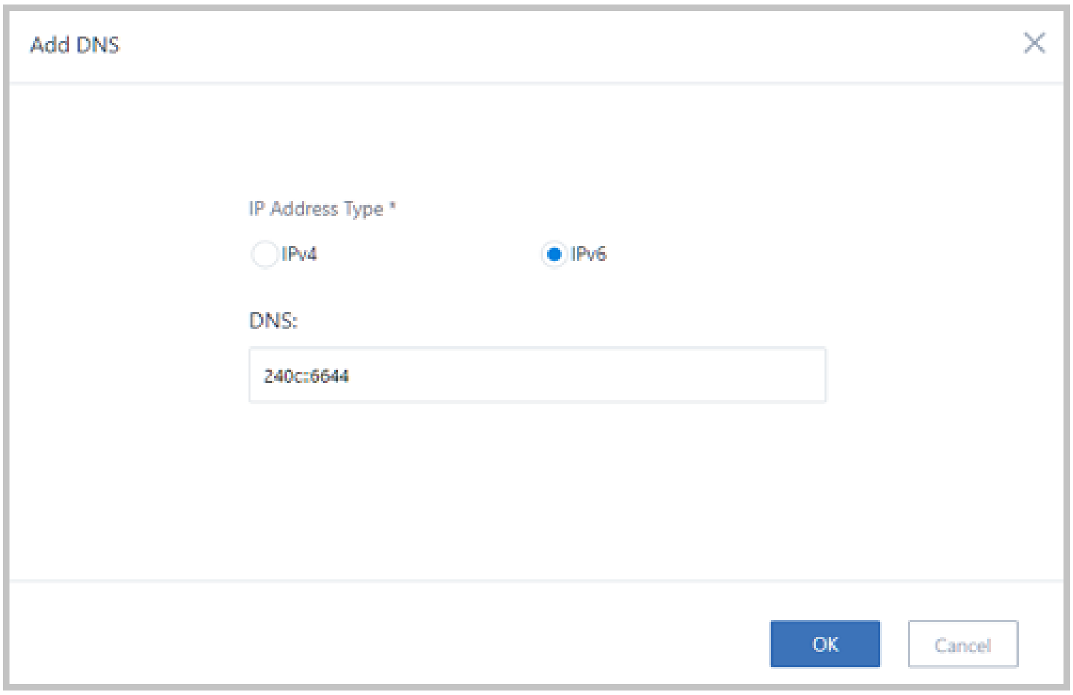
3. Add an IPv6 DNS address.

On the **DNS** tab page of this network, click **Add DNS**, and set the following parameters:

- **IP Address Type**: Select IPv6.
- **DNS**: Enter the IPv6 DNS address, such as *240c::6644*.

Click **OK** to complete adding the DNS address, as shown in *Add IPv6 DNS Address*.

Figure 4-2: Add IPv6 DNS Address



Add DNS

IP Address Type *

○ IPv4 ● IPv6

DNS:

240c::6644

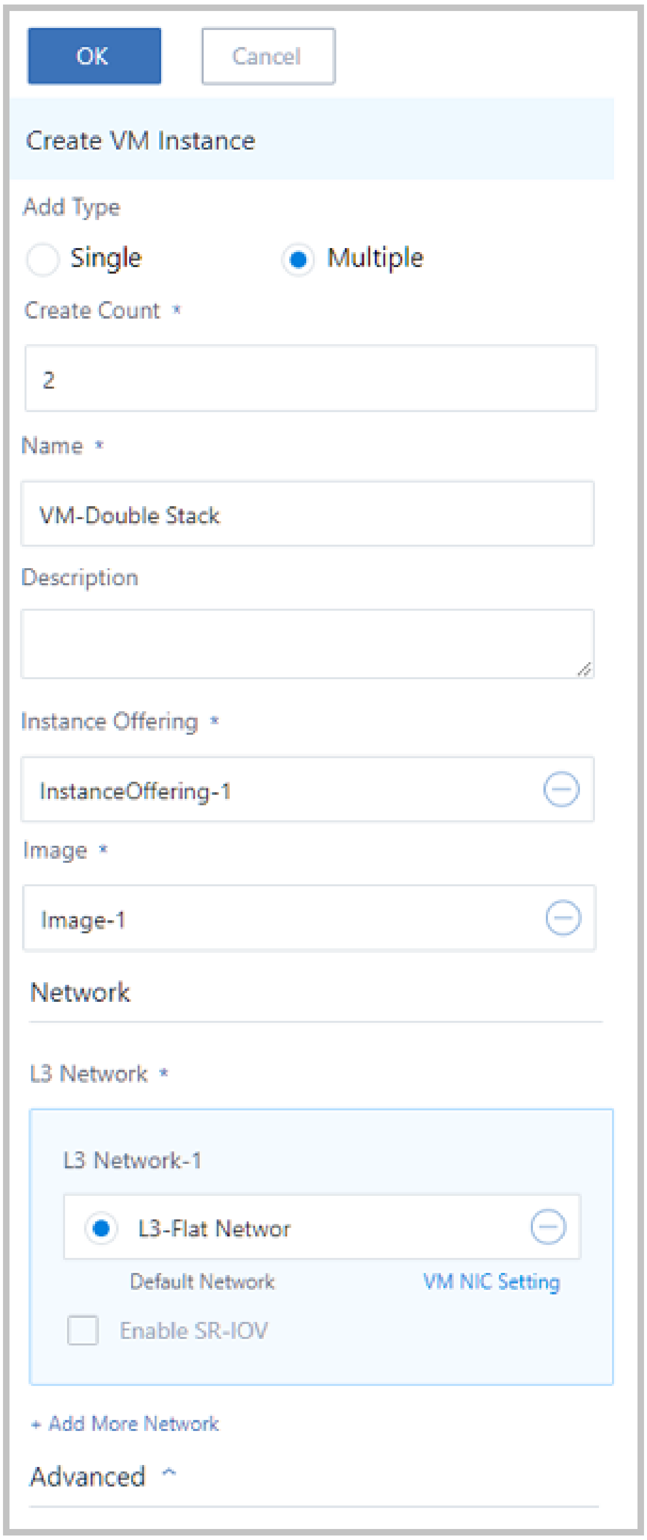
4. Create VM instances by using a double stack network.

   In the navigation pane of the ZStack Private Cloud UI, choose **Resource Pool** > **VM Instance**. On the **VM Instance** page, click **Create VM Instance**. On the displayed **Create VM Instance** page, set the following parameters:

   - **Add Type**: Select Multiple.
   - **Create Count**: Enter 2.
   - **Name**: Enter a name for the VM instances, such as VM-Double Stack.
   - **Description**: Optional. Enter a description for the VM instances.
   - **Instance Offering**: Select an instance offering for the VM instances.
   - **Image**: Select an image for the VM instances.
   - **Network**: Select a double-stack L3 network.

   Click **OK** to complete creating these two double stack VM instances (VM-Double Stack-1 and VM-Double Stack-2), as shown in *Create VM Instance*.

Figure 4-3: Create VM Instance



Create VM Instance

Add Type

○ Single ● Multiple

Create Count * 2

Name * VM-Double Stack

Description

Instance Offering * InstanceOffering-1

Image * Image-1

Network

L3 Network *

L3 Network-1

● L3-Flat Networ

Default Network VM NIC Setting

□ Enable SR-IOV

+ Add More Network

Advanced ^

5. Obtain IPv6 addresses of the VM instances.

ZStack defaults to automatically obtain IP addresses for the IPv4 network, while you must manually obtain IPv6 addresses for the IPv6 network. Open the consoles of these two VM instances respectively, and run the following commands to obtain IP addresses:

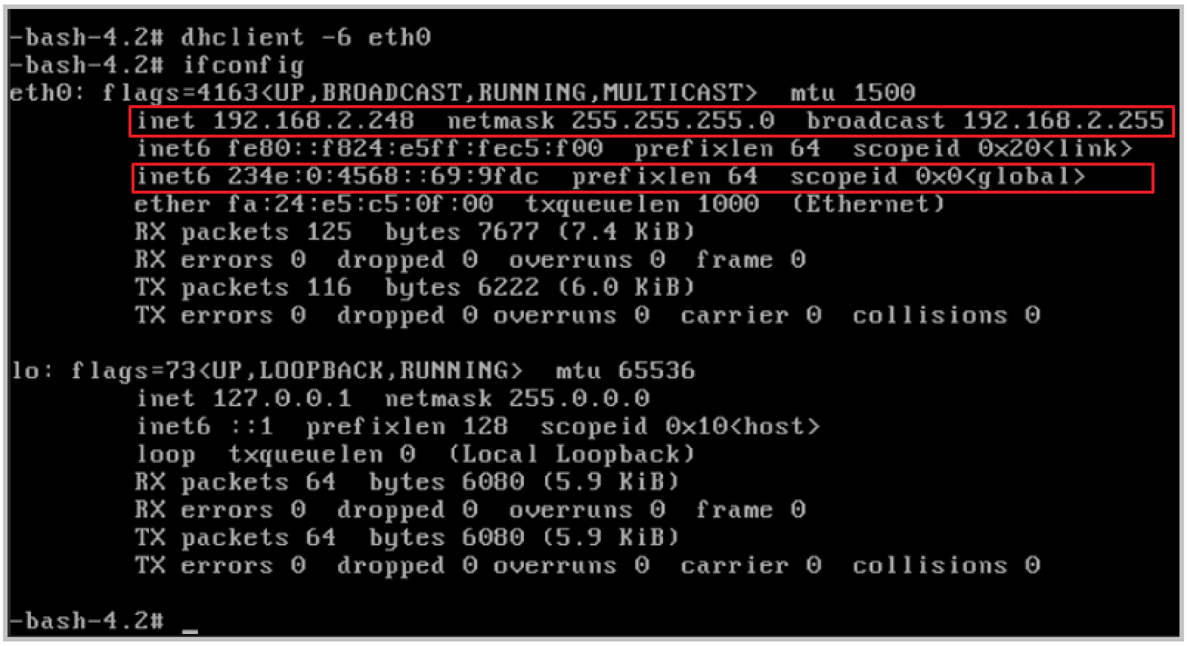
```
-bash-4.2# dhclient -6 eth0  # eth0 is the NIC name.
```

Note:

The address that begins with FE80 is the link-local address instead of the expected address.

Obtain IP addresses, as shown in *Obtain IP Address*.

Figure 4-4: Obtain IP Address



```
-bash-4.2# dhclient -6 eth0
-bash-4.2# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.248  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::f824:e5ff:fec5:f00  prefixlen 64  scopeid 0x20<link>
        inet6 234e:0:4568::69:9fdc  prefixlen 64  scopeid 0x0<global>
        ether fa:24:e5:c5:0f:00  txqueuelen 1000  (Ethernet)
        RX packets 125  bytes 7677 (7.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 116  bytes 6222 (6.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 0  (Local Loopback)
        RX packets 64  bytes 6080 (5.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 64  bytes 6080 (5.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

-bash-4.2# _
```

In this scenario, after running **`ifconfig`**, you will obtain the following addresses:

- VM-Double Stack-1 IPv4 address: *192.168.2.248*
- VM-Double Stack-1 IPv6 address: *234e:0:4568::69:9fdc*
- VM-Double Stack-2 IPv4 address: *192.168.2.183*
- VM-Double Stack-2 IPv6 address: *234e:0:4568::23:c59b*

6. Validate the network connectivity.

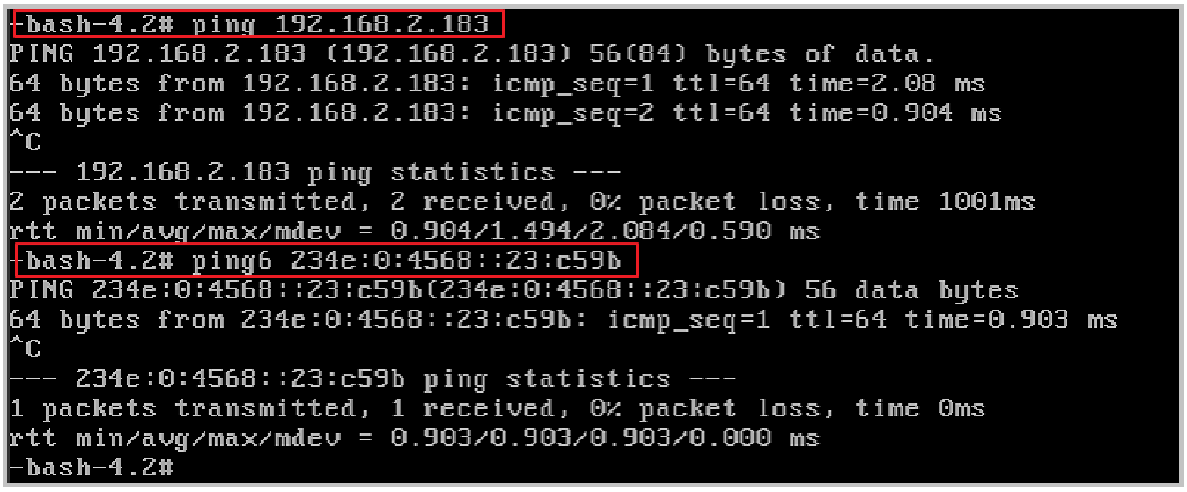
   Expected result:

   - Log in to the VM-Double Stack-1. Use the IPv4 address and the IPv6 address respectively to validate whether these two IP addresses can `ping` VM-Double Stack-2.

- Log in to the VM-Double Stack-2. Use the IPv4 address and the IPv6 address respectively to validate whether these two IP addresses can `ping` VM-Double Stack-1.

Validate the network connectivity: Log in to the VM-Double Stack-1. Use the IPv4 address and the IPv6 address to validate whether these two IP addresses can `ping` VM-Double Stack-2, as shown in *Validate Network Connectivity*.

Figure 4-5: Validate Network Connectivity



```
-bash-4.2# ping 192.168.2.183
PING 192.168.2.183 (192.168.2.183) 56(84) bytes of data.
64 bytes from 192.168.2.183: icmp_seq=1 ttl=64 time=2.08 ms
64 bytes from 192.168.2.183: icmp_seq=2 ttl=64 time=0.904 ms
^C
--- 192.168.2.183 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.904/1.494/2.084/0.590 ms
-bash-4.2# ping6 234e:0:4568::23:c59b
PING 234e:0:4568::23:c59b(234e:0:4568::23:c59b) 56 data bytes
64 bytes from 234e:0:4568::23:c59b: icmp_seq=1 ttl=64 time=0.903 ms
^C
--- 234e:0:4568::23:c59b ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.903/0.903/0.903/0.000 ms
-bash-4.2#
```

Similarly, log in to the VM-Double Stack-2. Use the IPv4 address and the IPv6 address to validate whether these two IP addresses `ping` VM-Double Stack-1.

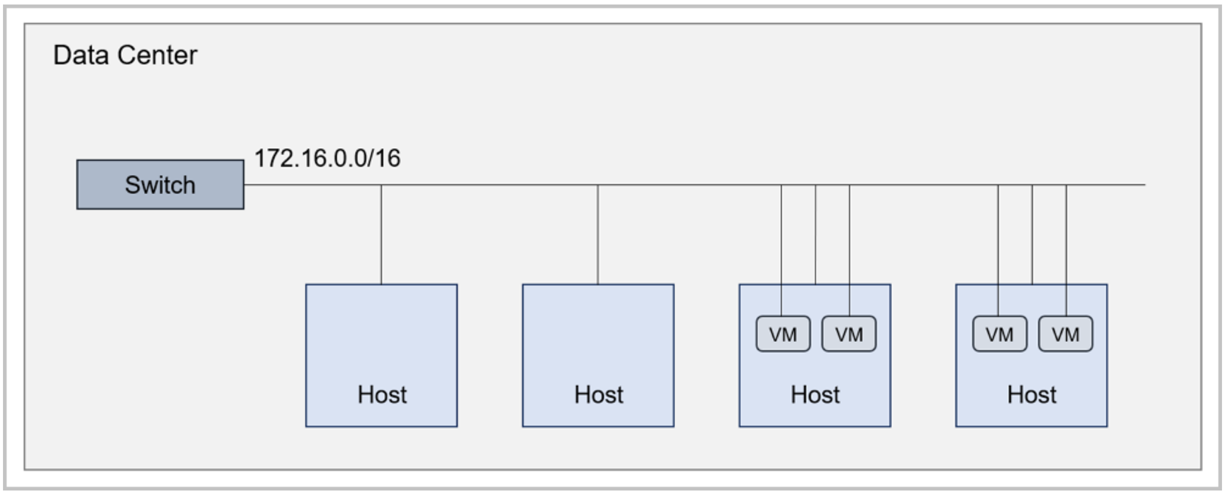
So far, we have introduced how to use a double stack (IPv4+IPv6) flat network.

4.2 L2 Connected Network

A typical L2 flat network is a layer 2 connected network. Specifically, in a data center of a private cloud, all hosts and VM instances are on one L2 network, while the IP addresses of these hosts and VM instances are on the same L3 network. Mutual accesses between these hosts and VM instances are not routed via gateways.

IP addresses of all compute nodes are assigned from *172.16.0.0/16*, as shown in *L2 Flat Network*.

Figure 4-6: L2 Flat Network



L2 flat networks are preferably applied to small and medium-sized enterprises. Due to the simplicity of the network topology architecture, staff computers can reach each other directly. Computers of all staffs are on one L2 network, so network access controls usually are guaranteed by security groups (distributed firewalls) of the private cloud.

In actual deployments, gateway addresses of L3 networks must be set as gateway addresses of your company. Besides, IP addresses that are assigned to VM instances must avoid duplicating IP addresses associated to hosts with attended assignment and isolation.

4.3 Security Group

Prerequisites

A security group serves as a virtual firewall for your VM instances to allow or deny incoming network traffic to, or outgoing network traffic from, multiple types of cloud resources. L3 network security controls are provided over your VM instances, and TCP, UDP, or ICMP data packets are managed for effective filtering. With the security group, you can effectively control specified VM instances on specified networks according to specified security rules.

- Flat networks, vRouter networks, and VPC support the security group service. The security group service is provided by the security group network service module. By using iptables, you can perform security controls over VM instances. This method also applies to flat networks, vRouter networks, and VPC.
- A security group is actually a distributed firewall. When you modify a rule, or when you add or delete a NIC, note that firewall rules in VM instances are updated as well.

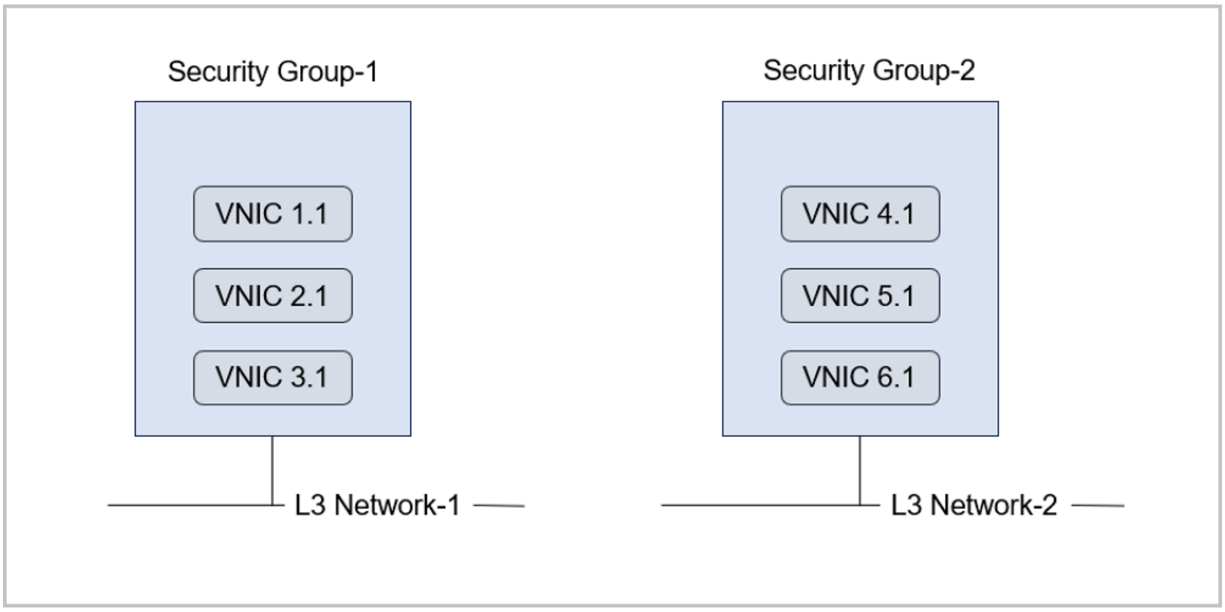
Security group rule:

- A security group rule has the following two types of traffics according the direction of data packets:
  - Ingress: Represents inbound data packets that access a VM instance.
  - Egress: Represents outbound data packets that are sent from a VM instance.
- A security group rule supports the following protocol types:
  - ALL: Includes all protocol types, indicating that you cannot specify a port.
  - TCP: Supports ports 1-65535.
  - UDP: Supports ports 1-65535.
  - ICMP: By default, both the start port and end port are all -1, indicating that all ICMP protocols are supported.
- A security group rule can limit data sources that comes either from inside or outside of VM instances. Currently, sources can be set as source CIDR or source security group.
  - Source CIDR: Allows only the specified CIDR.
  - Source security group: Allows only the VM instances in a specified security group.

> **Note:**
>
> If you set both CIDR and the security group, note that only the intersection of them can take effect.

A security group topology is shown in *Figure 4-7: Security Group*.

Figure 4-7: Security Group



Context

The basic workflow about how to use a security group is as follows: Select an L3 network, set the corresponding security group rule, and add specified VM instances to the rule.

The following two scenarios are introduced as how to create a security group in a flat network environment:

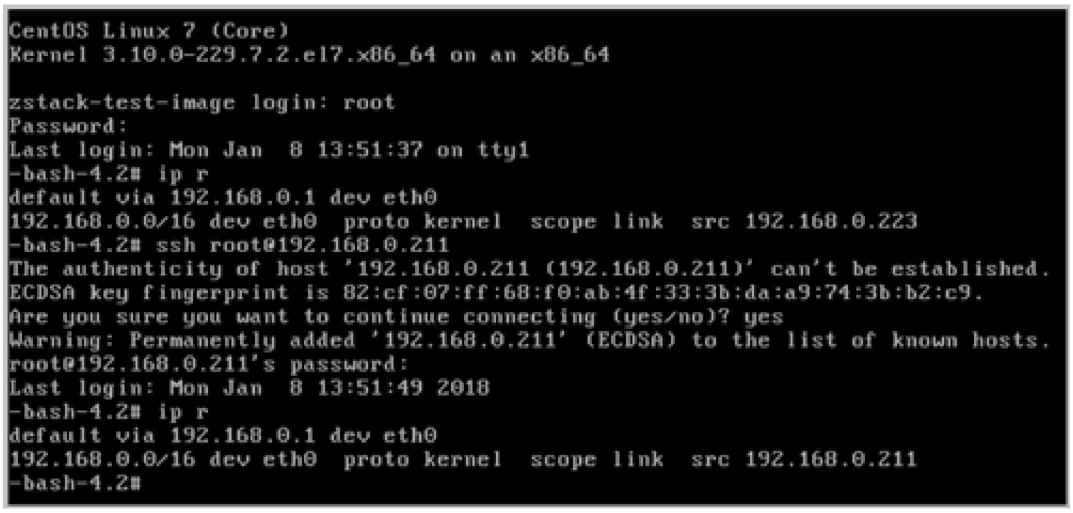
- Set an ingress rule for VM instances.
- Set an egress rule for VM instances.

Procedure

1. Create a flat network, and create two VM instances named after VM-1 and VM-2. For more information, see *Basic Deployment*.

   Log in to VM-1, and remotely connect to VM-2 via port 22 defaulted by SSH, as shown in *Successful Login via SSH*.

   Figure 4-8: Successful Login via SSH

```
CentOS Linux 7 (Core)
Kernel 3.10.0-229.7.2.el7.x86_64 on an x86_64

zstack-test-image login: root
Password:
Last login: Mon Jan  8 13:51:37 on tty1
-bash-4.2# ip r
default via 192.168.0.1 dev eth0
192.168.0.0/16 dev eth0  proto kernel  scope link  src 192.168.0.223
-bash-4.2# ssh root@192.168.0.211
The authenticity of host '192.168.0.211 (192.168.0.211)' can't be established.
ECDSA key fingerprint is 82:cf:07:ff:68:f0:ab:4f:33:3b:da:a9:74:3b:b2:c9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.211' (ECDSA) to the list of known hosts.
root@192.168.0.211's password:
Last login: Mon Jan  8 13:51:49 2018
-bash-4.2# ip r
default via 192.168.0.1 dev eth0
192.168.0.0/16 dev eth0  proto kernel  scope link  src 192.168.0.211
-bash-4.2#
```

2. Set an ingress rule for VM-1.

   a) Create a security group.

   In the navigation pane of the ZStack Private Cloud UI, choose **Network Service** > **Security Group**. On the **Security Group** page, click **Create Security Group**. On the displayed **Create Security Group** page, set the following parameters:

   - **Name**: Enter a name for the security group.
   - **Description**: Optional. Enter a description for the security group.
   - **IP Address Type**: Select an IP address type. Options: IPv4 | IPv6.

- **Network**: Select an existing L3 network according to the IP address type that you selected.
  - IPv4 supports three types of L3 networks: public network, private network, and VPC network.
  - IPv6 supports two types of L3 networks: public network and private network.
  - You can add multiple same types of L3 networks, but cannot add different types of L3 networks at the same time.
- **Rule**: Optional. Either set a firewall rule directly when creating a security group or set the firewall rule after creating the security group.

  Note:

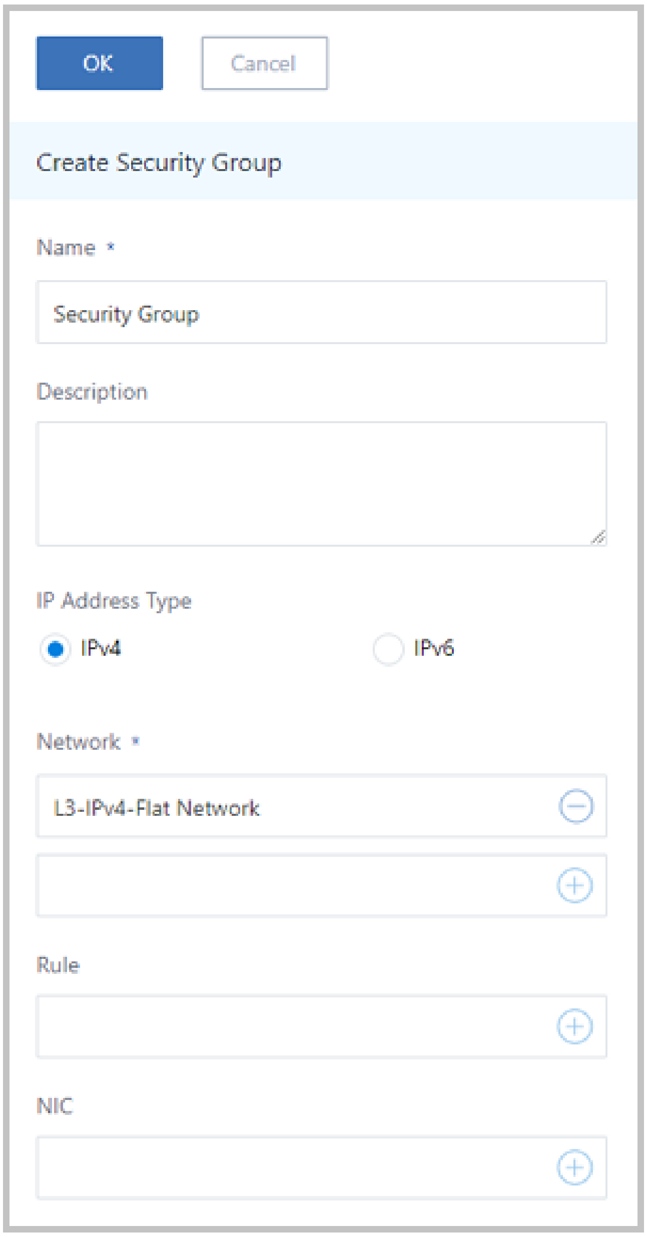
  For more information, see *Set Ingress Rule* and *Set Egress Rule*.
- **NIC**: Optional. Add a VM NIC to the security group. A VM NIC can be either added directly to the security group when you create the security group or added to the security group after you create the security group.

  Note:

  For more information, see *Add VM NIC to Security Group*.

Click **OK** to complete creating the security group, as shown in *Create Security Group*.

Figure 4-9: Create Security Group



Create Security Group

Name *

Security Group

Description

IP Address Type

● IPv4 ○ IPv6

Network *

L3-IPv4-Flat Network

Rule

NIC

b) Set an ingress rule.

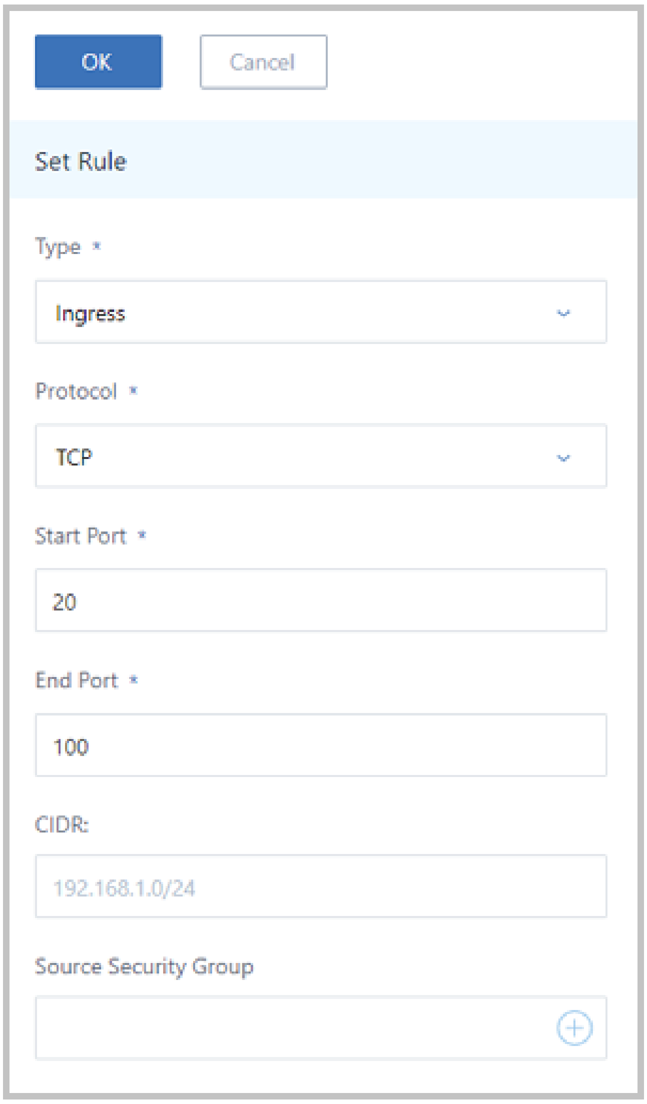
The following example is about a security group rule that is set after a security group is created. On the **Security Group** page, select a security group that you created, expand the details page of the security group, and click **Rule**. On the **Rule** tab page, choose **Actions** > **Add Rule**. On the displayed **Set Rule** page, set the following parameters:

- **Type**: Select Ingress.
- **Protocol**: Select TCP.
- **Start Port**: Enter 20.

- **End Port**: Enter 100.
- **CIDR**: Optional. Only the specified CIDR is allowed.
- **Source Security Group**: Optional. Only the specified VM instance within the security group is allowed.

Click **OK** to complete setting the ingress rule, as shown in *Set Ingress Rule*.

Figure 4-10: Set Ingress Rule



Set Rule

Type *

Ingress

Protocol *

TCP

Start Port *

20

End Port *

100

CIDR:

192.168.1.0/24

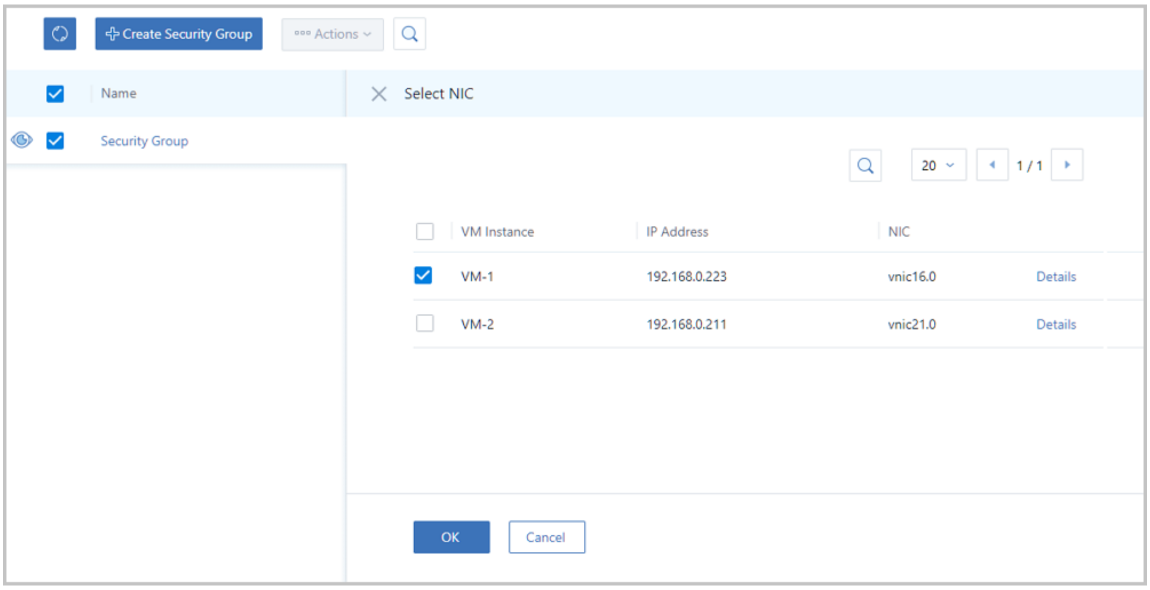
Source Security Group

c) Add a VM NIC to the security group.

The following example is about a VM NIC that is added after a security group is created. On the **Security Group** page, select a security group that you created, expand the details page of the security group, and click **VM NIC**. On the **VM NIC** tab page, choose **Actions** > **Bind**

VM NIC. On the displayed **Select NIC** page, select a VM NIC, such as VM-1, as shown in *Add VM NIC to Security Group*.

Figure 4-11: Add VM NIC to Security Group



☐	VM Instance	IP Address	NIC	
☑	VM-1	192.168.0.223	vnic16.0	Details
☐	VM-2	192.168.0.211	vnic21.0	Details

d) Verify the ingress rule.

Now, VM-1 can be only reached via port 20-100.

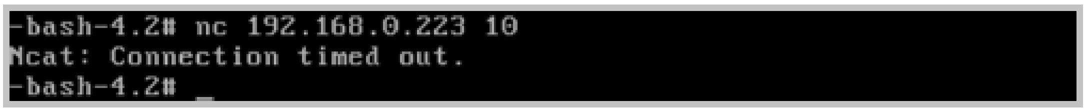
Log in to VM-2, and try to run **nc** to establish connections to VM-1.

Note:

The iptables rule in VM-1 must be cleaned by running `iptables -F`.

1. For example, if you use port 10 that is out of the rule range, VM-2 and VM-1 fail to communicate with each other, as shown in *VM-2 Fails to Connect VM-1 on Port 10*.

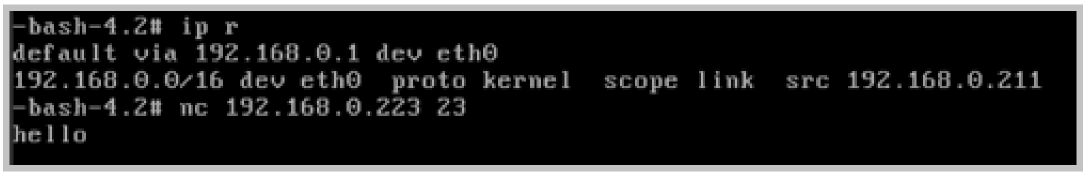
   Figure 4-12: VM-2 Fails to Connect VM-1 on Port 10

   ```
   -bash-4.2# nc 192.168.0.223 10
   Ncat: Connection timed out.
   -bash-4.2# _
   ```

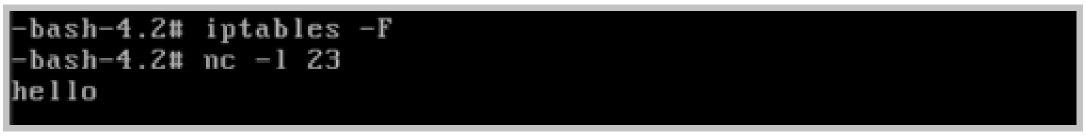
2. For example, if you use port 23 within the rule range, both VM-2 and VM-1 communicate with each other successfully, as shown in *VM-2 Sends Message to VM-1 on Port 23* and *VM-1 Receives Message Successfully on Port 23*.

Figure 4-13: VM-2 Sends Message to VM-1 on Port 23



```
-bash-4.2# ip r
default via 192.168.0.1 dev eth0
192.168.0.0/16 dev eth0  proto kernel  scope link  src 192.168.0.211
-bash-4.2# nc 192.168.0.223 23
hello
```

Figure 4-14: VM-1 Receives Message Successfully on Port 23



```
-bash-4.2# iptables -F
-bash-4.2# nc -l 23
hello
```

3. Set an egress rule for VM-1.

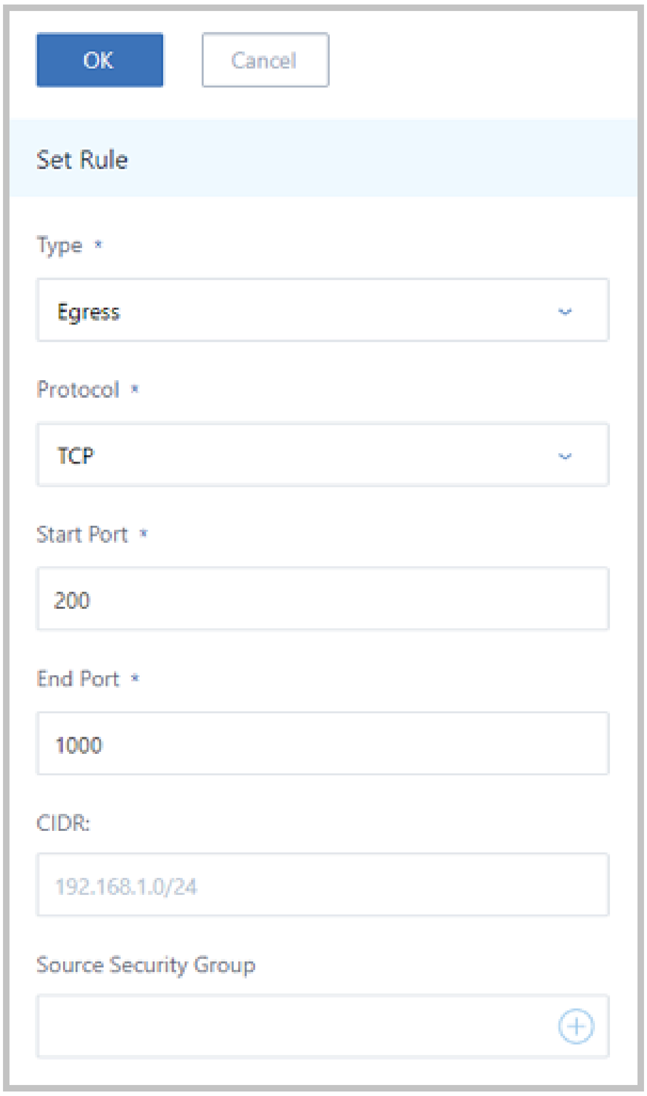
   a) Set an egress rule.

   The following example is about a security group rule that is set after a security group is created. On the **Security Group** page, select a security group that you created, expand the details page of the security group, and click **Rule**. On the **Rule** tab page, choose **Actions** > **Add Rule**. On the displayed **Set Rule** page, set the following parameters:

   - **Type**: Select Egress.
   - **Protocol**: Select TCP.
   - **Start Port**: Enter 200.
   - **End Port**: Enter 1000.
   - **CIDR**: Optional. Only specified CIDR is allowed.
   - **Source Security Group**: Optional. Only the specified VM instance within the security group is allowed.

   Click **OK** to complete setting the egress rule, as shown in *Set Egress Rule*.

Figure 4-15: Set Egress Rule



Set Rule

Type *

Egress

Protocol *

TCP

Start Port *

200

End Port *

1000

CIDR:

192.168.1.0/24

Source Security Group

b) Verify the egress rule.

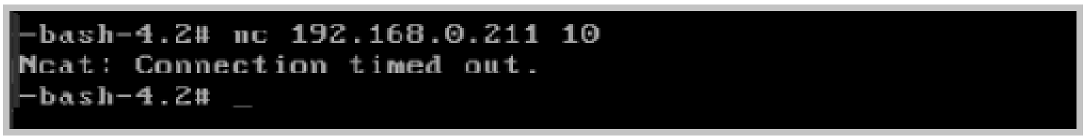
Now, only VM-1 can be reached via port 200-1000.

Log in to VM-2, and try to run **nc** to establish connections to VM-1.

> **Note:**
>
> The iptables in VM-1 can be cleaned by running `iptables -F`.

1. For example, if you use port 10 that is out of the rule range, VM-2 and VM-1 fail to communicate with each other, as shown in *VM-2 Fails to Connect to VM-1 on Port 10*

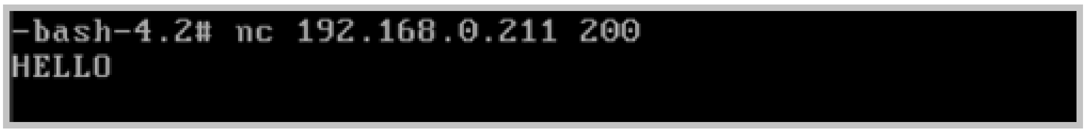
Figure 4-16: VM-2 Fails to Connect to VM-1 on Port 10



```
-bash-4.2# nc 192.168.0.211 10
Ncat: Connection timed out.
-bash-4.2# _
```

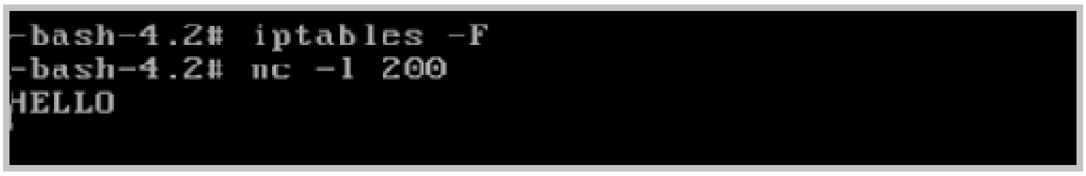
2. For example, if you use port 200 within the rule range, VM-2 and VM-1 communicate with each other successfully, as shown in *VM-1 Sends Message to VM-2 on Port 200* and *VM-2 Receives Message Successfully on Port 200*.

Figure 4-17: VM-1 Sends Message to VM-2 on Port 200



```
-bash-4.2# nc 192.168.0.211 200
HELLO
```

Figure 4-18: VM-2 Receives Message Successfully on Port 200



```
-bash-4.2# iptables -F
-bash-4.2# nc -l 200
HELLO
```

What's next

The constraints of a security group are as follows:

- A security group can be attached to more than one VM instance. These VM instances will share the same security group rules.
- A security group can be attached to more than one L3 network. These L3 networks will share the same security group rules.
- A security group supports whitelists. That is, you can set all security group rules to Allow. Once you set an allow rule for a port, other ports will not be allowed.
- When you create a security group, the system automatically configures two rules (an inbound rule and an outbound rule whose protocol types are both ALL) for communications in the security group. You can delete these two default rules to cancel the intra-group communication.
- When you create a security group, if you did not set any rule, incoming traffics are not allowed to access VM instances in the security group. However, outgoing traffics from VM instances in the security group are allowed.

- If you are using simultaneously the security group with other network services, such as load balancing and vRouter table, make sure that the corresponding rules required by these network services are added to the security group.

So far, we have introduced how to use the security group.

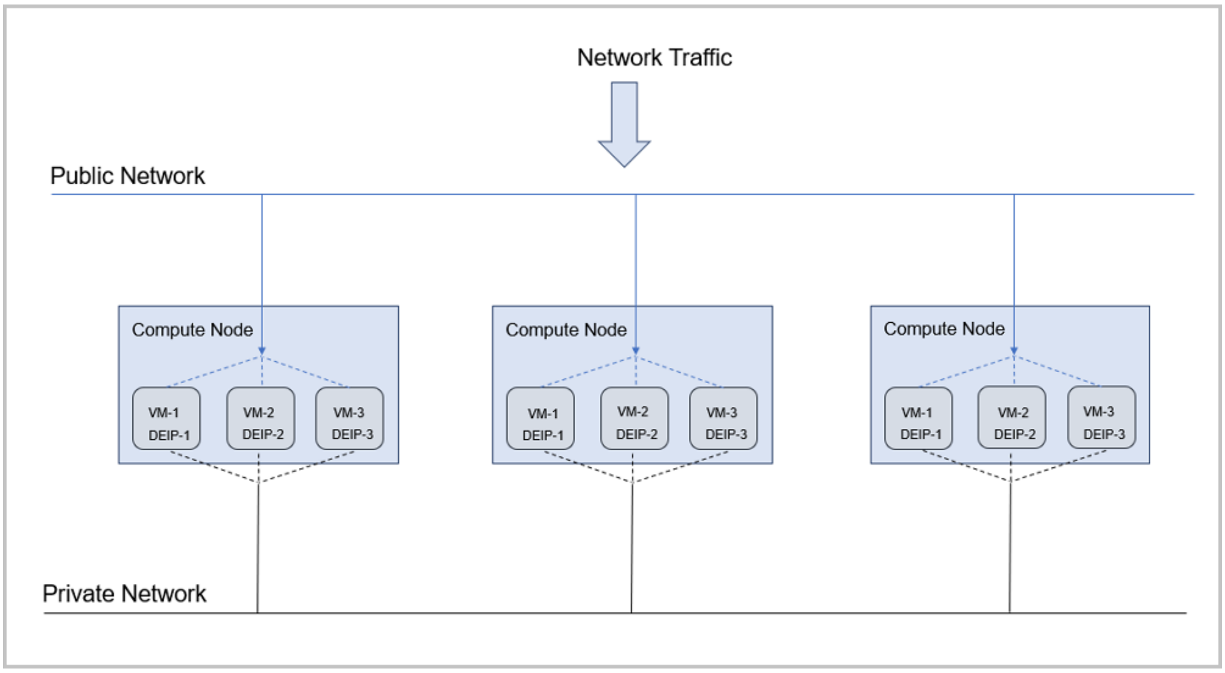
4.4 EIP

Prerequisites

An elastic IP address (EIP) is a method to access a private network through other networks. An EIP converts the IP address of a network into the IP address of another network based on the network address translation (NAT) function.

The following is an example of an EIP usage scenario in flat networks, as shown in *EIP Usage Scenario in Flat Network*.

Figure 4-19: EIP Usage Scenario in Flat Network



Context

The following two scenarios are introduced as how to create an EIP in a flat network environment:

- Create an EIP and bind it to a VM instance.
- Bind the EIP to another VM instance.

Procedure

1. Create a flat network, and create a VM instance named after VM-1by using this flat network. For more information, see *Basic Deployment*.
2. Create an EIP.

   In the navigation pane of the ZStack Private Cloud UI, choose **Network Service** > **EIP**. On the **EIP** page, click **Create EIP**. On the displayed **Create EIP** page, set the following parameters:

   - **Name**: Enter a name for the EIP, such as EIP-1.
   - **Description**: Optional. Enter a description for the EIP.
   - **Select EIP**: Provide the EIP service via a VIP.

     To use a VIP, select one of the two methods:

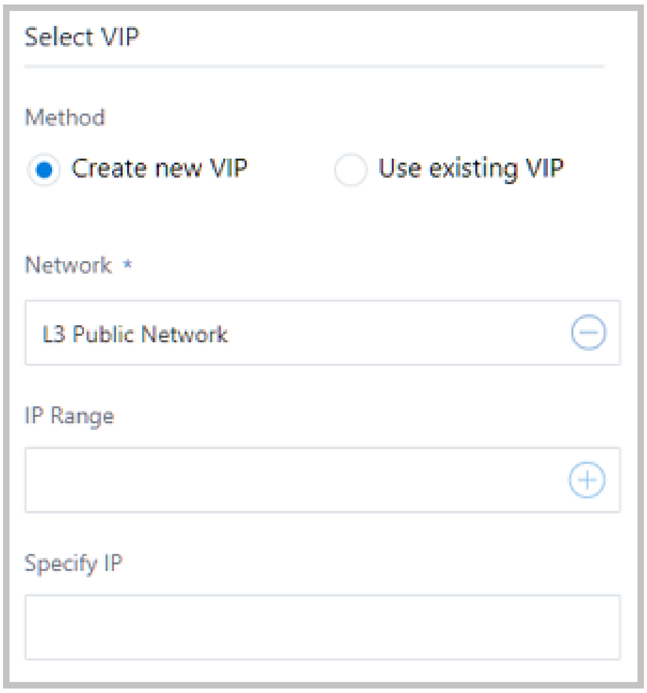
     - **Create new VIP**:

       To create a new VIP, set the following parameters:

       - **Network**: Select a network that provides VIPs. Options: public network | flat network.
       - **IP Range**: Optional. Specify an IP range. Notice that an IPv4 public network allows you to select a normal IP range or the IP range of an IP address pool.
       - **Specified IP**: Optional. Specify a VIP. If not specified, the system will automatically assign a VIP for the EIP.

       Create a new VIP, as shown in *Figure 4-20: Create new VIP*.

Figure 4-20: Create new VIP



Select VIP

Method

● Create new VIP ○ Use existing VIP

Network *

L3 Public Network

IP Range

Specify IP

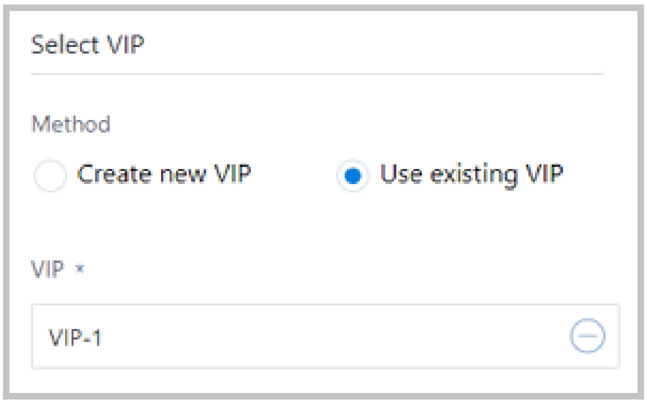
- **Use existing VIP**:

  To use an existing VIP, set the following parameter:

  - **VIP**: Select an existing VIP.

  Select an existing VIP, as shown in *Figure 4-21: Use existing IP*.

Figure 4-21: Use existing IP



Select VIP

Method

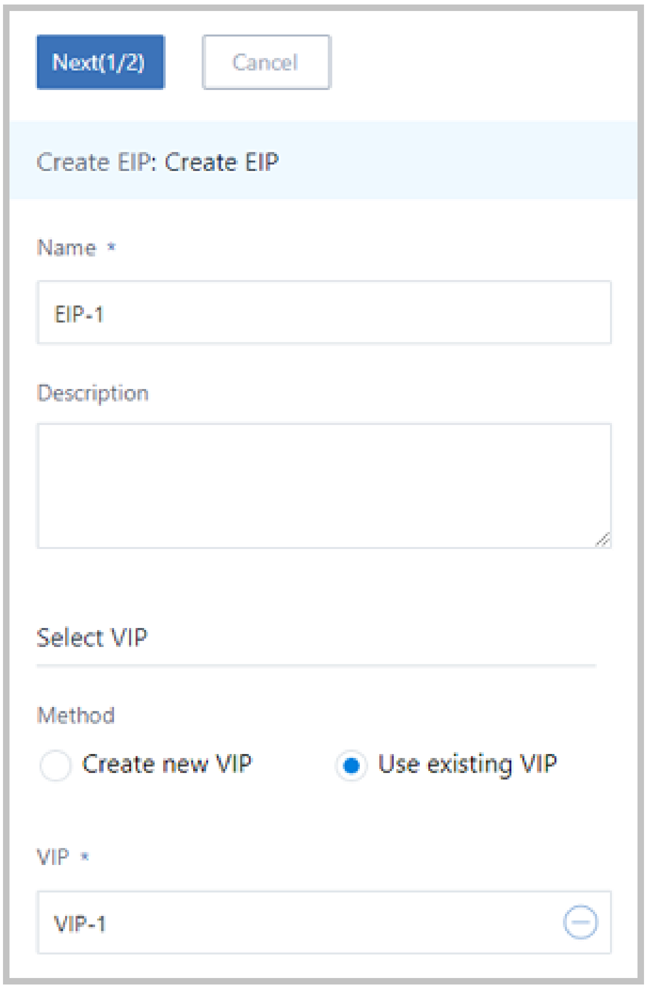
○ Create new VIP ● Use existing VIP

VIP *

VIP-1

Complete creating the EIP, as shown in *Create EIP*.

Figure 4-22: Create EIP



Create EIP: Create EIP

Name *

EIP-1

Description

Select VIP

Method

○ Create new VIP ● Use existing VIP

VIP *

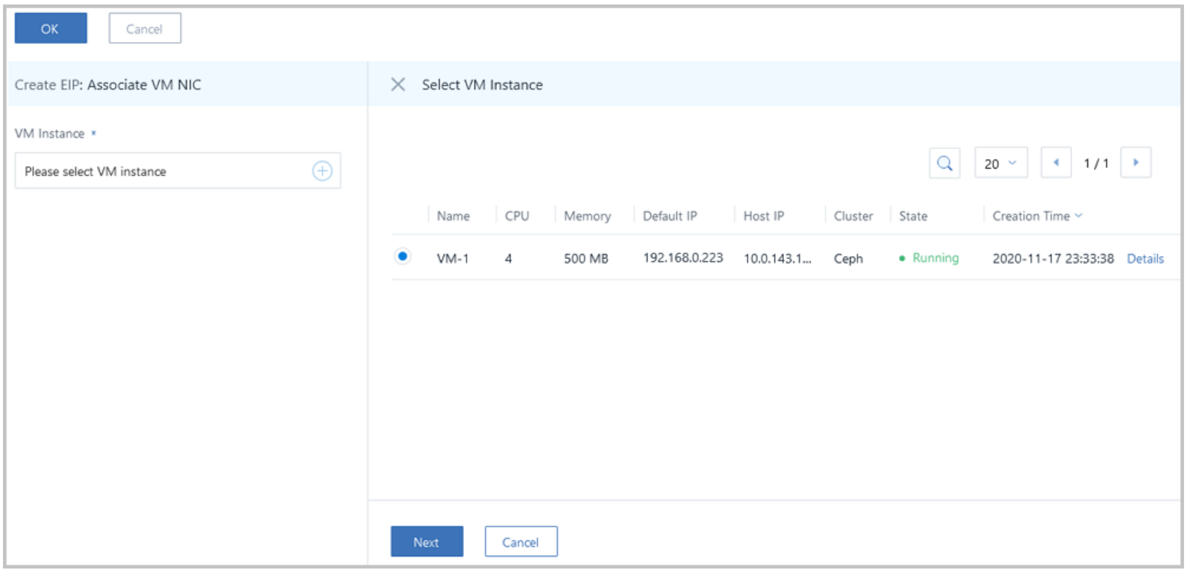
VIP-1

3. Bind EIP-1 to VM-1.

   A VM NIC can be either directly added to an EIP when the EIP is created or added to the VM instance after the EIP is created.

   The following example is about a VM NIC that is bound to an EIP when the EIP is created. On the **Create EIP** page, click **OK**. On the displayed **Associate VM NIC** page, click the **VM Instance** Plus sign. On the **Select VM Instance** page, select a VM instance that you need to associate, such as VM-1, and click **OK**, as shown in *Select VM-1* and *Bind EIP-1 to VM-1*.

Figure 4-23: Select VM-1



Create EIP: Associate VM NIC

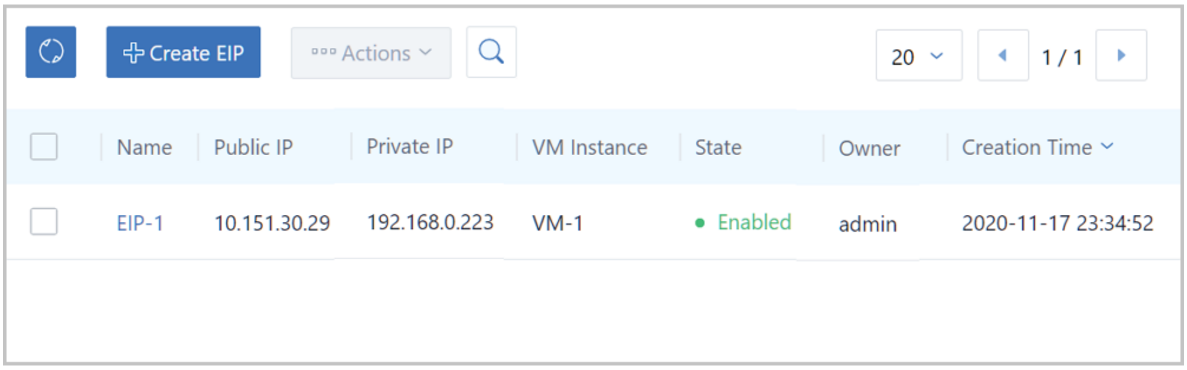
VM Instance *

Please select VM instance

20 1 / 1

	Name	CPU	Memory	Default IP	Host IP	Cluster	State	Creation Time	
●	VM-1	4	500 MB	192.168.0.223	10.0.143.1...	Ceph	● Running	2020-11-17 23:33:38	Details

Figure 4-24: Bind EIP-1 to VM-1

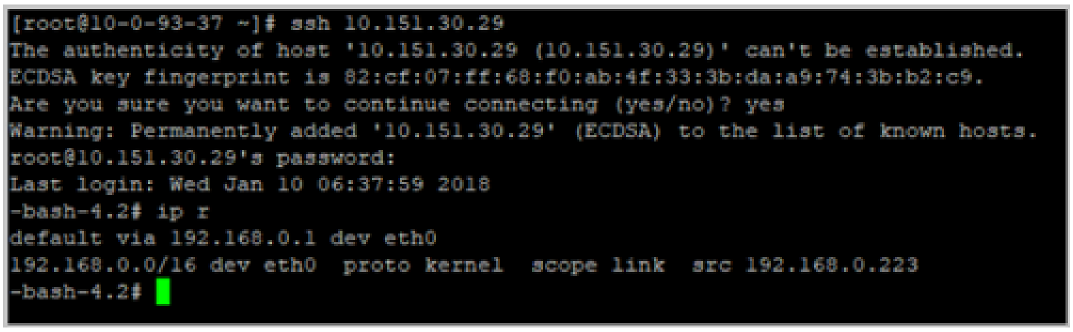


□	Name	Public IP	Private IP	VM Instance	State	Owner	Creation Time
□	EIP-1	10.151.30.29	192.168.0.223	VM-1	● Enabled	admin	2020-11-17 23:34:52

4. Log in to VM-1 via EIP-1.

   Log in to a VM instance within a public network range (*10.151.30.0-10.151.30.30*) that can reach the flat network, and use SSH to reach EIP-1 (*10.151.30.29*). That is, log in to VM-1 via the private network: *192.168.0.223*, as shown in *Log in to VM-1 via EIP-1*.

Figure 4-25: Log in to VM-1 via EIP-1



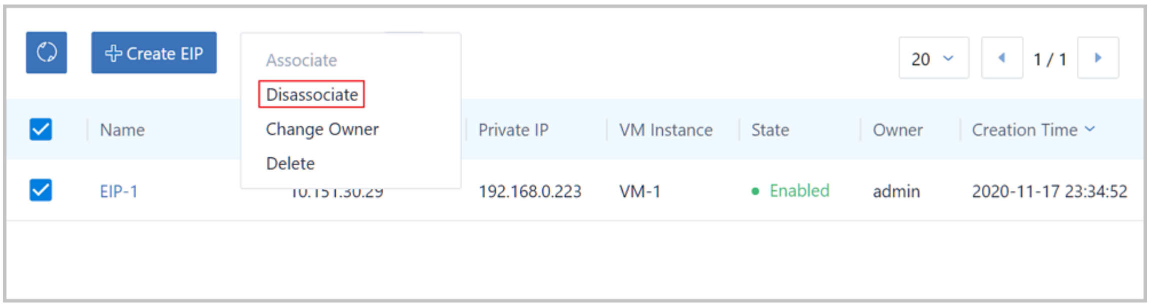
```
[root@10-0-93-37 ~]# ssh 10.151.30.29
The authenticity of host '10.151.30.29 (10.151.30.29)' can't be established.
ECDSA key fingerprint is 82:cf:07:ff:68:f0:ab:4f:33:3b:da:a9:74:3b:b2:c9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.151.30.29' (ECDSA) to the list of known hosts.
root@10.151.30.29's password:
Last login: Wed Jan 10 06:37:59 2018
-bash-4.2# ip r
default via 192.168.0.1 dev eth0
192.168.0.0/16 dev eth0  proto kernel  scope link  src 192.168.0.223
-bash-4.2# 
```

5. Bind EIP-1 to another VM instance.

   a) Unbind EIP-1 from VM-1.

   On the **EIP** page, select EIP-1, and click **Actions** > **Disassociate**. On the displayed **Disassociate VM Instance** confirmation page, click **OK**, as shown in *Unbind EIP-1 from VM-1*.

   Figure 4-26: Unbind EIP-1 from VM-1



☑	Name		Private IP	VM Instance	State	Owner	Creation Time
☑	EIP-1	10.151.30.29	192.168.0.223	VM-1	● Enabled	admin	2020-11-17 23:34:52

   b) Bind EIP-1 to another VM instance.

   After you unbind the EIP, click **Associate** to rebind the EIP to another VM instance.

So far, we have introduced how to use the EIP of the flat network.

4.5 Load Balancing

Prerequisites

Load balancing (LB) distributes inbound traffics from a VIP to a group of backend VM instances, and then automatically detects and isolates unavailable VM instances, whereby enhancing the service capability and availability of your businesses.

- Load balancing automatically distributes your inbound application traffics to the preconfigured backend VM instances, thereby providing highly concurrent and highly reliable access services.

- In your practice, you can adjust the VM instances in load balancing listeners to improve your service capability, which will not affect your normal business access.
- A load balancing listener supports four types of protocols: TCP, HTTP, HTTPS, and UDP.
- If the listener protocol is HTTPS, you need to bind a certificate. Note that you can upload a certificate or a certificate link.
- A load balancer allows you to flexibly configure multiple forwarding policies to achieve advanced forwarding controlling.
- Load balancing allows you to display real-time SLB business traffics and connections in monitoring data.

Definitions related to load balancing:

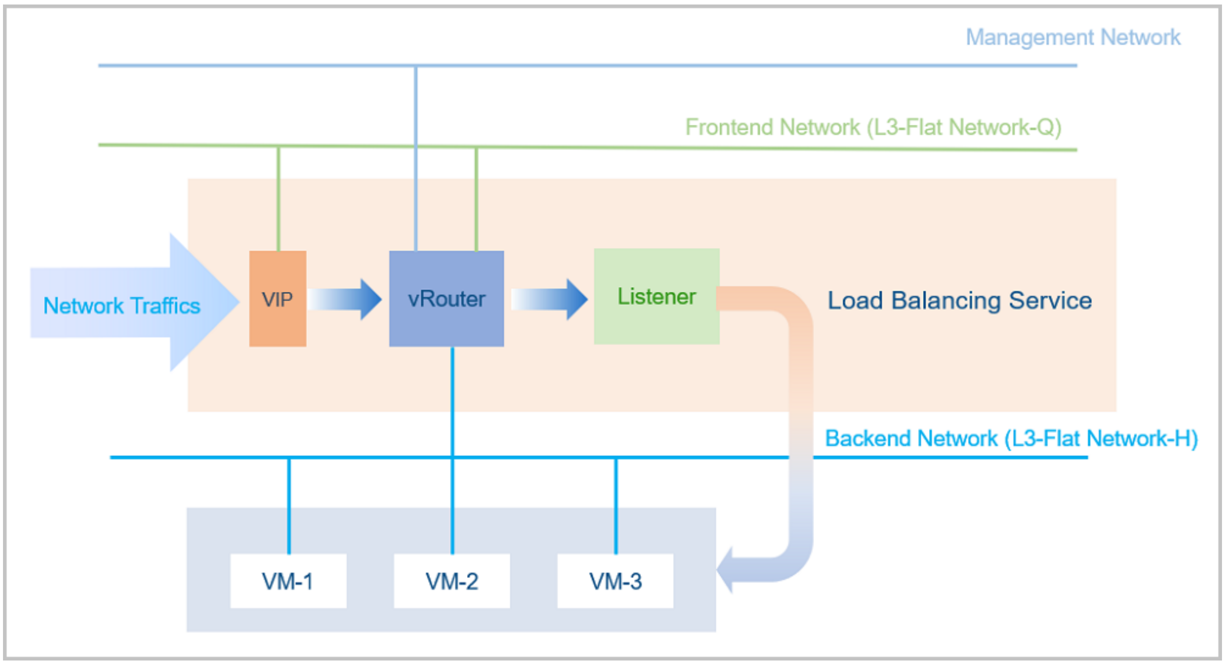
- Frontend network: Provide VIP networks in load balancing network services. Public networks, flat networks, and VPC networks can be used as frontend networks.
- Backend network: Create a private network for backend VM instances in load balancing network services. Flat networks, vRouter networks, and VPC networks can be used as backend networks.
- Private flat network load balancing: Act as a frontend network used to provide intranet load balancing services via a vRouter.

A flat network load balancing supports the following scenarios:

- Scenario 1: Use a flat network to act as a backend network used to provide the public load balancing service based on the flat network. Ensure that both the frontend network and the vRouter use the same L3 network.
- Scenario 2: Use other types of flat networks to act as backend networks used to provide the load balancing service based on the flat network. Ensure that both the frontend networks and the vRouters use the same L3 network.
- Scenario 3: Use the frontend network to act as the backend network used to provide the flat network based load balancing service. At this time, the frontend network defined by the vRouter offering can be a public network or flat network.

For the purpose of this Tutorial, Scenario 2 is taken as an example. The basic workflow about how it works is shown in *Private Flat Network Load Balancing*.

Figure 4-27: Private Flat Network Load Balancing



Context

To create a private flat network load balancing

1. Create a frontend network.
2. Create a vRouter offering that is attached to the frontend network.
3. Create a backend network and attach this vRouter offering to the backend network.
4. Create backend VM instances.
5. Create a load balancer.
6. Create and add a listener, and attach it to the load balancer.
7. Bind the NICs of the VM instances to this listener.
8. Validate this scenario.

Assume that the usage scenario is as follows:

Create a load balancer, add a listener and bind three VM instances to this listener. The load balancing service will be provided for these three VM instances based on the defaulted round-robin algorithm.

Procedure

1. Create a frontend network.

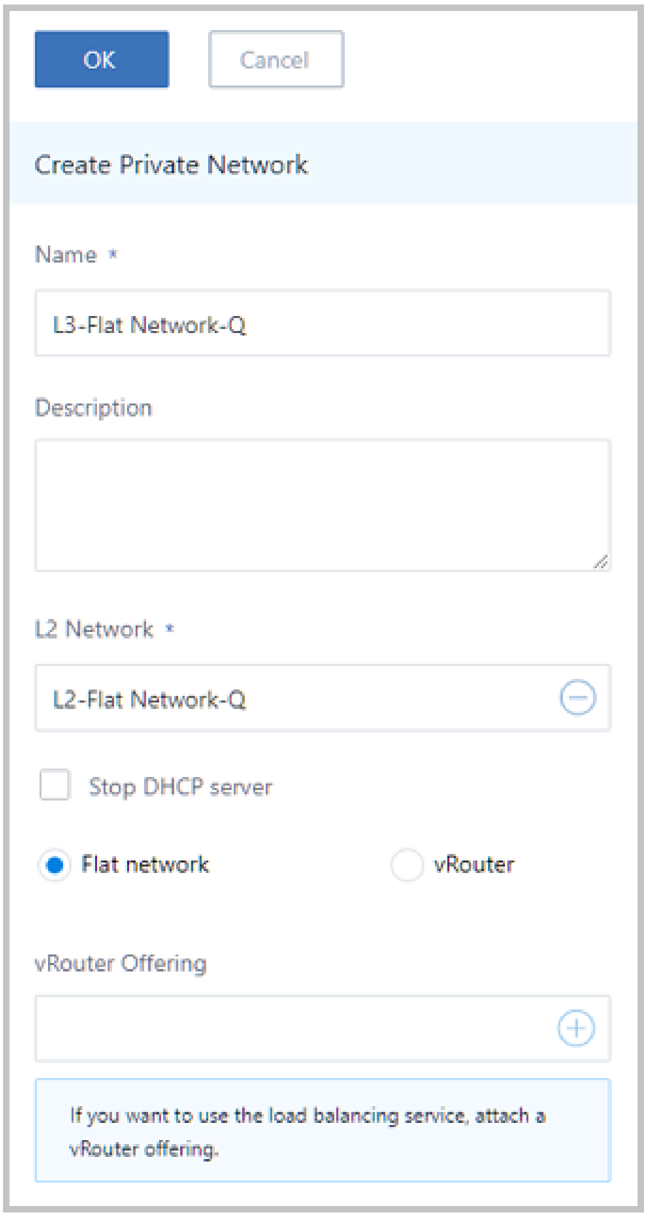
By referring to *IPv4 Flat Network Deployment*, create a flat network such L3-Flat Network-Q, and use this flat network to serve as a frontend network.

Note:

A frontend network can be used to create a VIP and a vRouter without attaching a vRouter offering.

Click **OK** to complete creating the frontend network, as shown in *Figure 4-28: Create Frontend Network*.

Figure 4-28: Create Frontend Network



Create Private Network

Name *

L3-Flat Network-Q

Description

L2 Network *

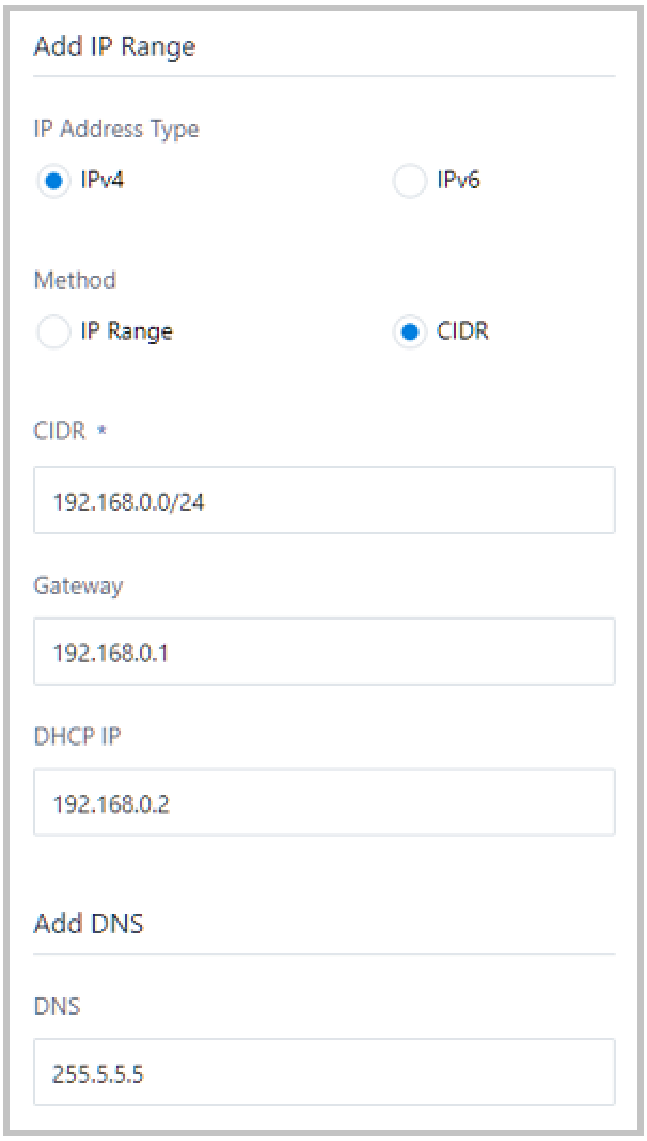
L2-Flat Network-Q

□ Stop DHCP server

● Flat network ○ vRouter

vRouter Offering

If you want to use the load balancing service, attach a vRouter offering.



Add IP Range

IP Address Type

● IPv4 ○ IPv6

Method

○ IP Range ● CIDR

CIDR *

192.168.0.0/24

Gateway

192.168.0.1

DHCP IP

192.168.0.2

Add DNS

DNS

255.5.5.5

2. Create a vRouter offering that is attached to the frontend network.

a) Add a vRouter image.

In the navigation pane of ZStack Private Cloud, choose **Network Resource** > **vRouter Resource** > **vRouter Image**. On the **vRouter Image**, click **Add vRouter Image**, and then set the following parameters:

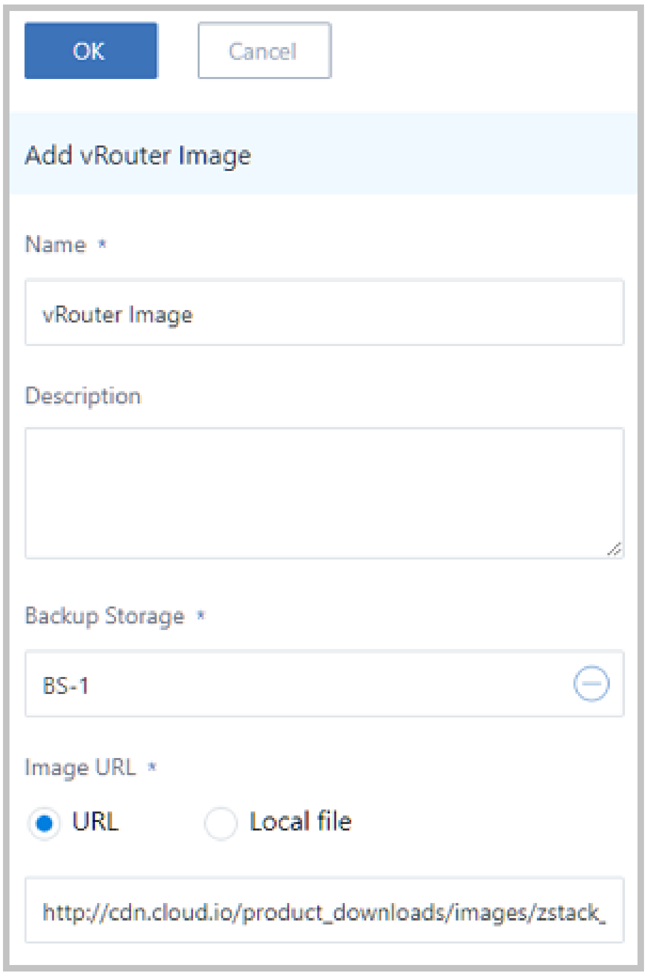
- **Name**: Enter a name for the vRouter image.
- **Description**: Optional. Enter a description for the vRouter image.
- **Backup Storage**: Select a backup storage to store the vRouter image.
- **Image URL**: Add the vRouter image via URL or local file.

  ZStack provides dedicated vRouter images and lets you download the latest vRouter images via *ZStack*.

- File name: zstack-vrouter-3.10.0.qcow2
- Download address: Click *ZStack Downloads*.

Click **OK** to complete adding the vRouter image, as shown in *Add vRouter Image*.

Figure 4-29: Add vRouter Image



Add vRouter Image

Name *

vRouter Image

Description

Backup Storage *

BS-1

Image URL *

● URL ○ Local file

http://cdn.cloud.io/product_downloads/images/zstack_

b) Create an L2 management network.

The management network is used to create vRouters. Plan for your networks as needed. On the Cloud, create an L2 management network and then an L3 management network.

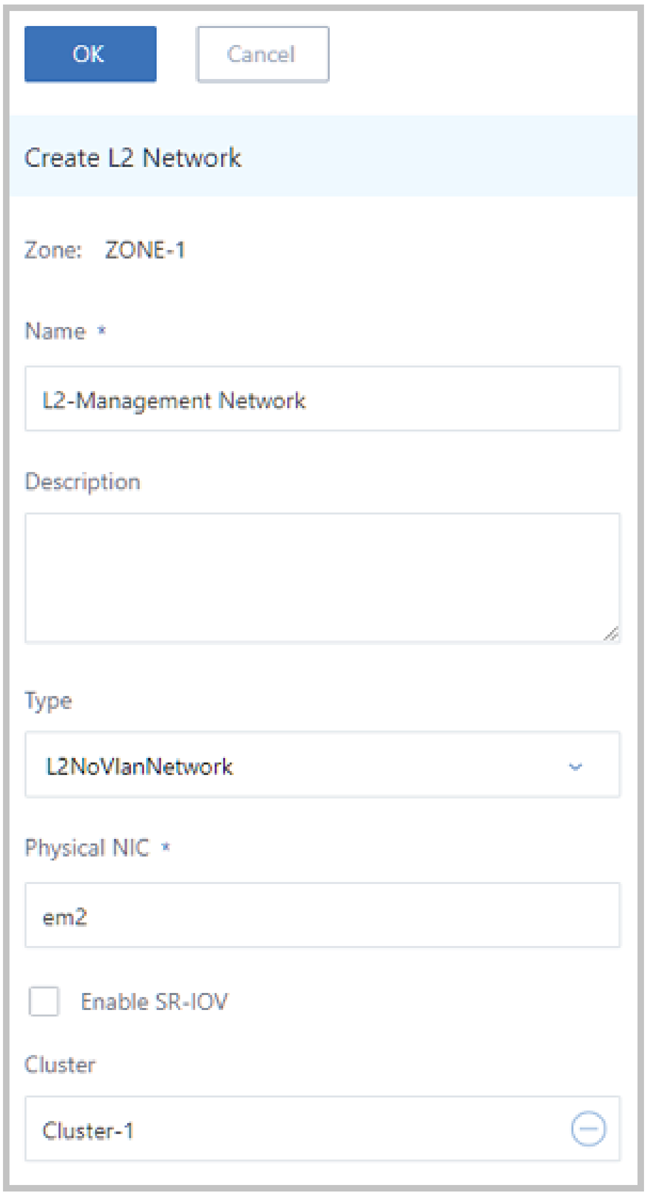
In the navigation pane of ZStack Private Cloud, choose **Network Resource** > **L2 Network Resource** > **L2 Network**. On the **L2 Network**, click **Create L2 Network**. On the displayed **Create L2 Network**, set the following parameters:

- **Name**: Enter a name for the L2 management network.
- **Description**: Optional. Enter a description for the L2 management network.
- **Type**: Select a network type as needed, such as L2NoVlanNetwork.

- **Physical NIC**: Enter a NIC name, such as em2.
- **Enable SR-IOV**: Choose whether to enable the SR-IOV feature. In this scenario, this checkbox is left unchecked.
- **Cluster**: Select a cluster, such as Cluster-1.

Click **OK** to complete creating the L2 management network, as shown in *Create L2 Management Network*.

Figure 4-30: Create L2 Management Network



Create L2 Network

Zone: ZONE-1

Name *

L2-Management Network

Description

Type

L2NoVlanNetwork

Physical NIC *

em2

□ Enable SR-IOV

Cluster

Cluster-1

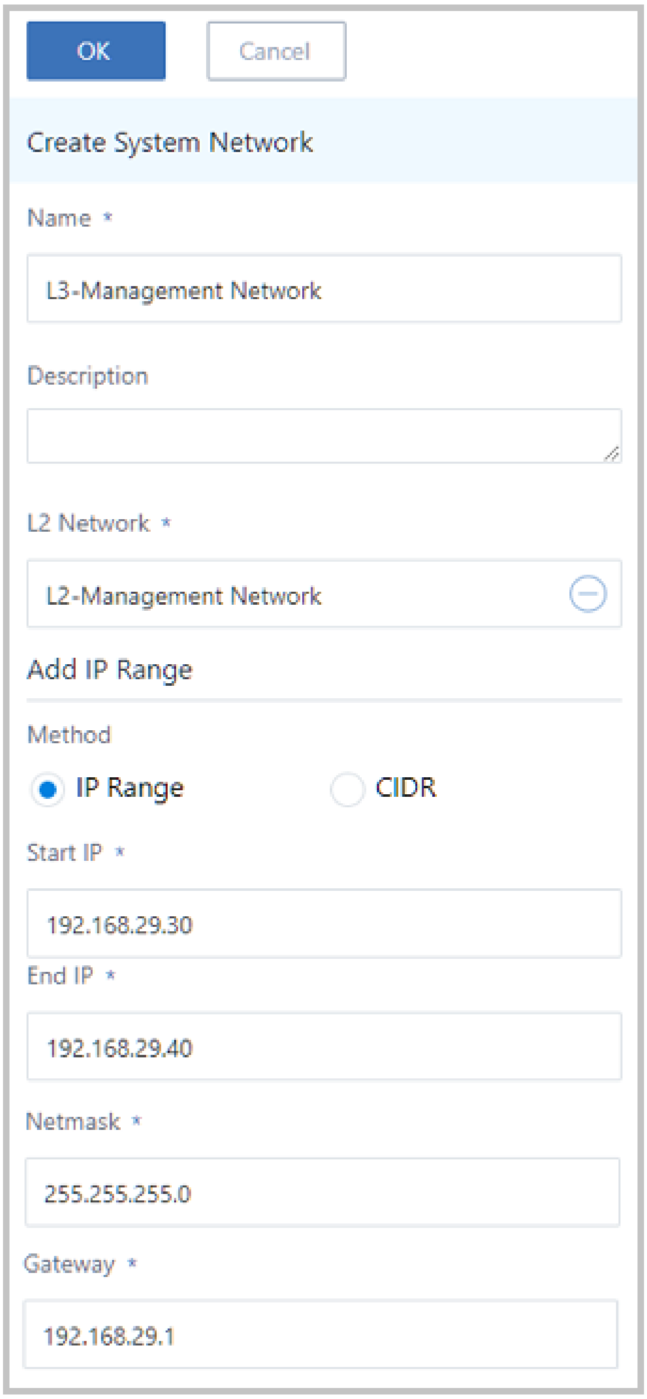
c) Create an L3 management network.

In the navigation pane of ZStack Private Cloud, choose **Network Resource** > **L3 Network** > **System Network**. On the **System Network**, click **Create System Network**. On the displayed **Create System Network**, and then set the following parameters:

- **Name**: Enter a name for the L3 management network.
- **Description**: Optional. Enter a description for the L3 management network.
- **L2 Network**: Select the created L2 management network.
- **Add IP Range**: Select IP Range or CIDR according to your network planning.
- **Gateway**: Set the gateway IP address as needed.

Click **OK** to complete creating the L3 management network, as shown in *Create L3 Management Network*.

Figure 4-31: Create L3 Management Network



Create System Network

Name *

L3-Management Network

Description

L2 Network *

L2-Management Network

Add IP Range

Method

● IP Range ○ CIDR

Start IP *

192.168.29.30

End IP *

192.168.29.40

Netmask *

255.255.255.0

Gateway *

192.168.29.1

d) Create a vRouter offering.

In the navigation pane of ZStack, choose **Network Resource** > **vRouter Resource** > **vRouter Offering**. On the **vRouter Offering** page, click **Create vRouter Offering**, set the following parameters:

- **Name**: Enter a name for the vRouter offering.

- **Description**: Optional. Enter a description for the vRouter offering.
- **CPU**: Set the CPU count. We recommend that you set the count to 8 or above in your production environment.
- **Memory**: Set the memory size. Unit: M | G | T. We recommend that you set the size to 8 G or above in your production environment.
- **Image**: Select the vRouter image.
- **Management Network**: Select the L3 management network.

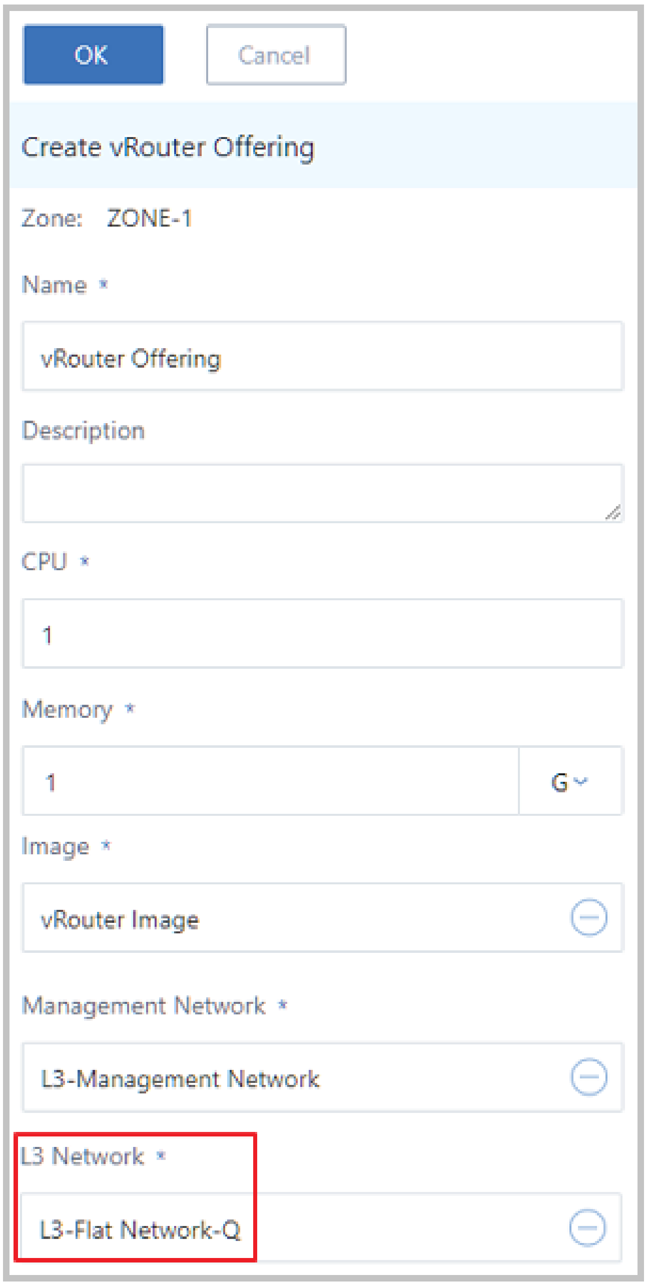
  > **Note:**
  >
  > Assume that you use a single network environment (the management network and the public network use the same network) on your Cloud. When you create a vRouter offering, the public network can be used as the management network.

- **L3 Network**: Select the L3 network. In this scenario, select the frontend network.

  > **Note:**
  >
  > Ensure that this L3 network and the frontend network are the same.

Click **OK** to complete creating the vRouter offering, as shown in *Create vRouter Offering*.

Figure 4-32: Create vRouter Offering



Create vRouter Offering

Zone: ZONE-1

Name *

vRouter Offering

Description

CPU *

1

Memory *

1 G

Image *

vRouter Image

Management Network *

L3-Management Network

L3 Network *

L3-Flat Network-Q

3. Create a backend network and attach this vRouter offering to the backend network.

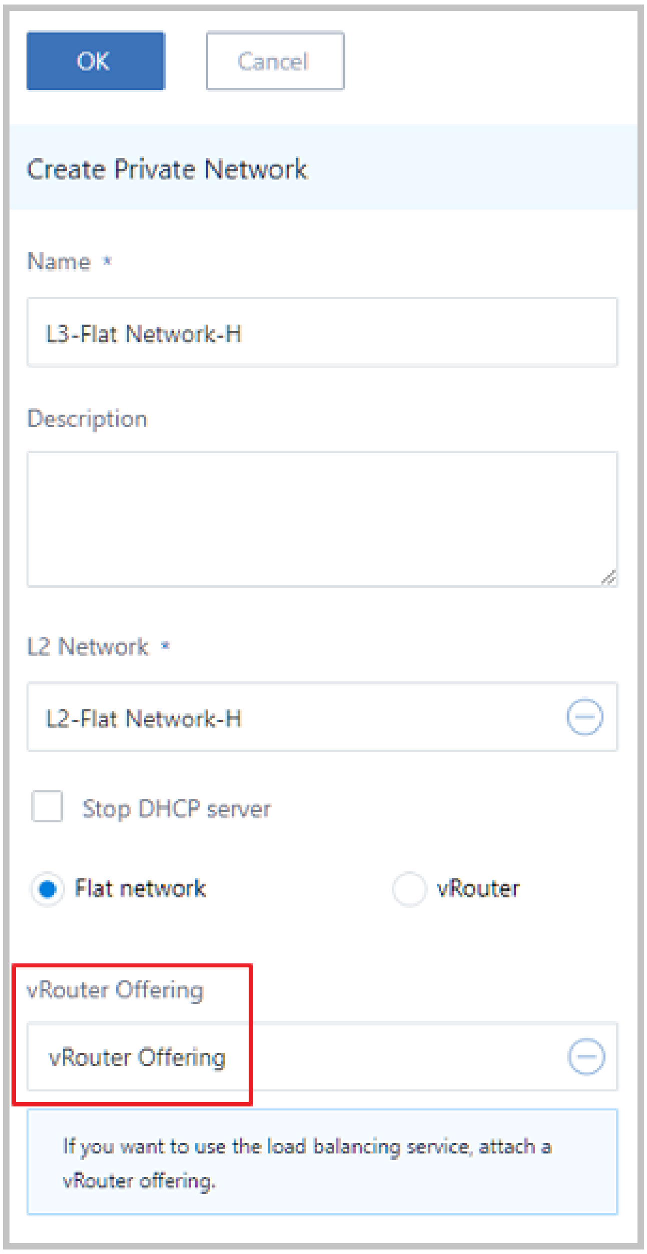
   By referring to *IPv4 Flat Network Deployment*, create a flat network such as L3-Flat Network-H, and use this flat network to act as a backend network.

   Note:

   ZStack lets you create a vRouter by using the vRouter offering attached by the backend network to provide the load balancing service. Hence, the backend network must attach this vRouter offering in advance.

Click **OK** to complete creating the backend network, as shown in *Create Backend Network*.

Figure 4-33: Create Backend Network



Create Private Network

Name *

L3-Flat Network-H

Description

L2 Network *

L2-Flat Network-H

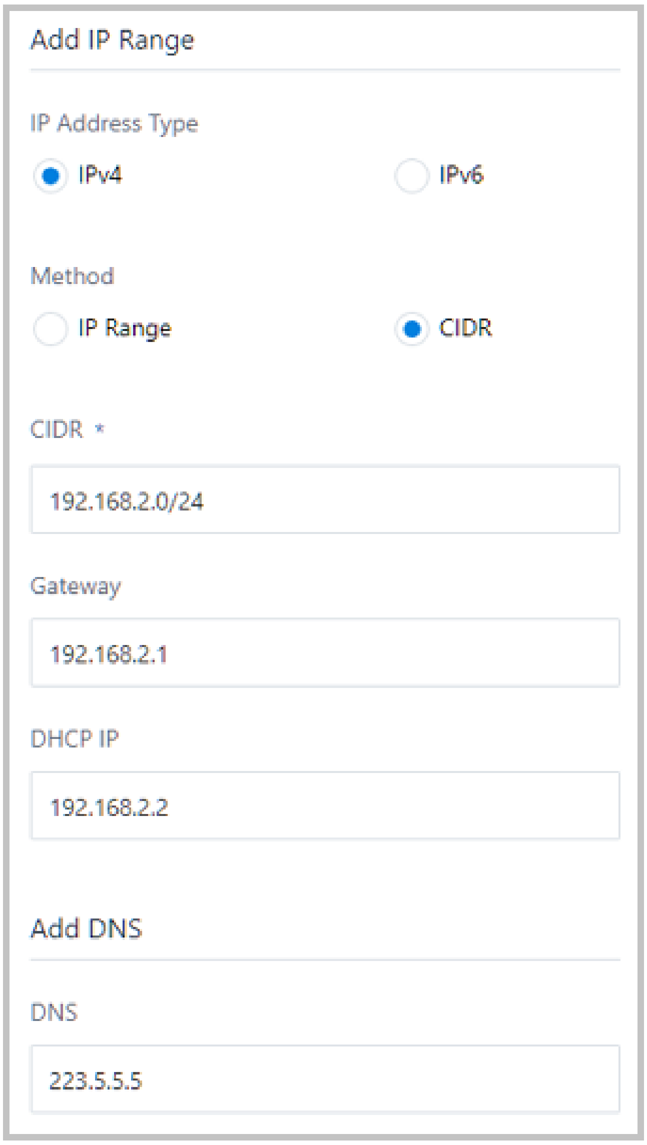
□ Stop DHCP server

● Flat network ○ vRouter

vRouter Offering

vRouter Offering

If you want to use the load balancing service, attach a vRouter offering.



Add IP Range

IP Address Type

● IPv4 ○ IPv6

Method

○ IP Range ● CIDR

CIDR *

192.168.2.0/24

Gateway

192.168.2.1

DHCP IP

192.168.2.2

Add DNS

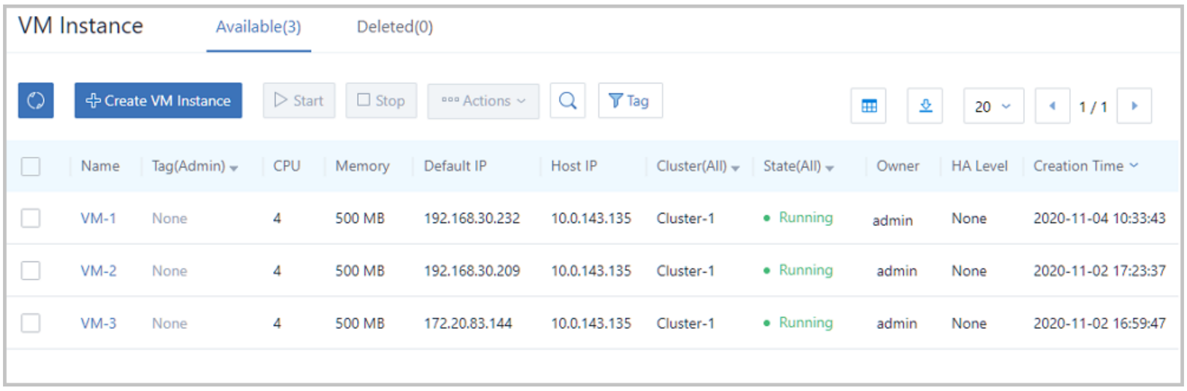
DNS

223.5.5.5

4. Create backend VM instances.

   Use the backend network to create three VM instances (such as VM-1, VM-2, and VM-3) to serve as the backend VM instances for the load balancing service, as shown in *Backend VM Instances*.

Figure 4-34: Backend VM Instances



□	Name	Tag(Admin)	CPU	Memory	Default IP	Host IP	Cluster(All)	State(All)	Owner	HA Level	Creation Time
□	VM-1	None	4	500 MB	192.168.30.232	10.0.143.135	Cluster-1	● Running	admin	None	2020-11-04 10:33:43
□	VM-2	None	4	500 MB	192.168.30.209	10.0.143.135	Cluster-1	● Running	admin	None	2020-11-02 17:23:37
□	VM-3	None	4	500 MB	172.20.83.144	10.0.143.135	Cluster-1	● Running	admin	None	2020-11-02 16:59:47

5. Create a load balancer.

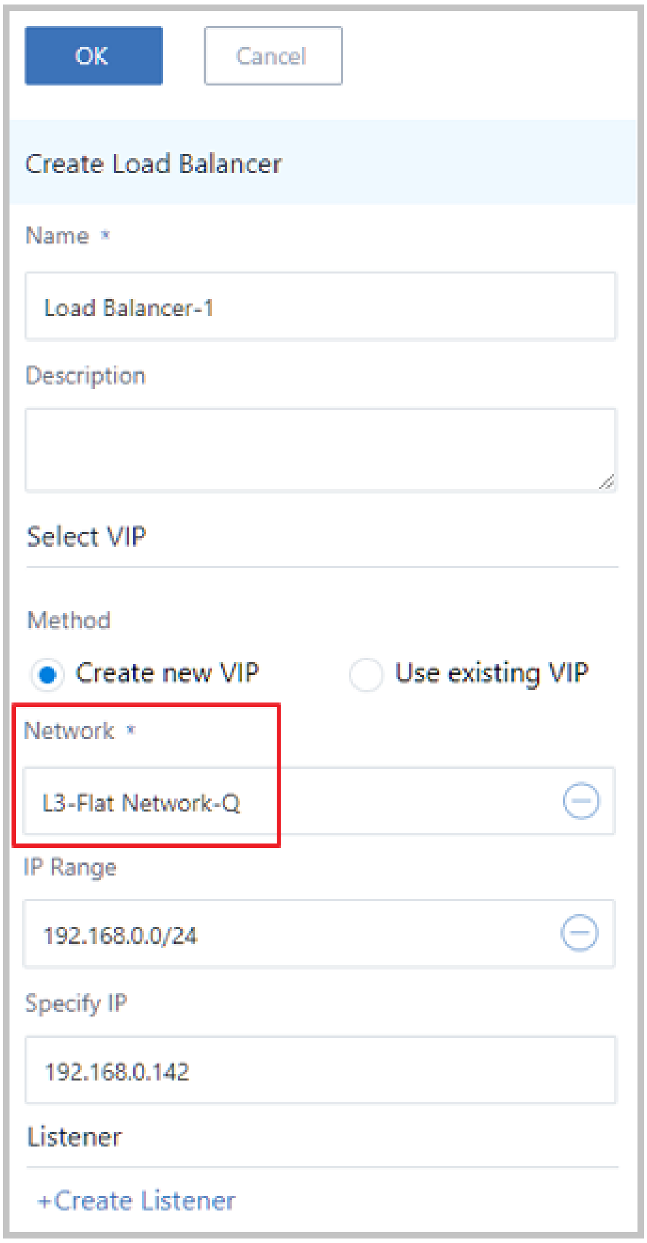
In the navigation pane of ZStack Private Cloud, choose **Network Service** > **Load Balancing** > **Load Balancer**. On the **Load Balancer** page, click **Create Load Balancer**. On the displayed **Create Load Balancer**, set the following parameters:

- **Name**: Enter a name for the load balancer, such as Load Balancer-1.
- **Description**: Optional. Enter a description for the load balancer.
- **Select VIP**: Select a VIP that used by the load balancing service. In this scenario, the Create new VIP method is used. Set the following parameters:
  - **Network**: Select a frontend network, such as L3-Flat Network-Q.
  - **IP Range**: Optional. Select the IP range where the VIP resides.
  - **Specify IP**: Optional. Specify an IP address for the VIP.
- **Listener**: Optional. Either select **Create Listener** to create a listener, or add a listen after creating the load balancer.

> **Note:**
>
> In this scenario, we recommend that you not add any listener. To add a listener, refer to the next step.

Click **OK** to complete creating the load balancer, as shown in *Create Load Balancer*.

Figure 4-35: Create Load Balancer



Create Load Balancer

Name *

Load Balancer-1

Description

Select VIP

Method

● Create new VIP ○ Use existing VIP

Network *

L3-Flat Network-Q

IP Range

192.168.0.0/24

Specify IP

192.168.0.142

Listener

6. Create and add a listener, and attach it to the load balancer.

   On the **Listener** tap page in the load balancer details page, choose **Actions** > **Create Listener**. On the displayed **Create Listener**, set the following parameters:

   - **Name**: Enter a name for the listener, such as Listener-1.
   - **Description**: Optional. Enter a description for the listener.
   - **Protocol**: Select a protocol type. Options: TCP | HTTP | HTTPS | UDP. These types of protocols all support ports from 1-65535.

- **Load Balancer Port**: Select one port to act as the load balancer port from port 1-65535, such as port 80.
- **VM Port**: Select one port to act as the VM port from port 1-65535, such as port 100.

> **Note:**
>
> For example, assume that the load balancer port is 80 and the VM port is 100. Data can be accessed from port 80 of a VIP and are forwarded to port 100 of backend VM instances via load balancing.

- **Advanced**: Make advanced settings, such health check protocol and load balancing algorithm. In this scenario, the default settings are used.
  - **Health Check Protocol**: Set the health check protocol. Options: TCP | HTTP | UDP. The health check protocol can be different from the listener protocol.
    - If the listener protocol is TCP, HTTP, or HTTPS, the health check protocol can be TCP or HTTP.
    - If the listener protocol is UDP, the health check protocol can be UDP.
    - If you select HTTP, you can configure the normal status code, health check path, and health check method.
      - **Normal Status Code**: The HTTP status code when the health check passes. You can select more than one code as needed. Options: http_2xx | http_3xx | http_4xx | http_5xx.
      - **Health Check Path**: The URI of the page on which health checks are performed. For example, `/healthcheck.html`. We recommend that you set health check for static pages. When you set a health check path, make sure that:
        - The health check path must be 2 to 80 characters long.
        - The health check path can contain only letters, numbers, special symbols (-/.%?#&), or a combination of these three types of characters.
        - The health check path must start with a forward slash (/).
      - **Health Check Method**: Check whether the server application is healthy by sending a HEAD or GET request to simulate the access behavior of a browser. Default method: HEAD.
  - **Idle Connection Timeout**: The amount of time that the load balancer terminates the connection between the server and the client when no data is transmitted. Default value: 60 seconds.

- **Health Check Threshold**: The number of consecutive health check successes required before considering an unhealthy VM instance healthy. Default value: 2.
- **Health Check Port**: Default value: default, indicating that the health check port is the same as that of the VM instance. You can also specify other ports as needed.
- **Unhealth Check Threshold**: The number of consecutive failed health checks required before considering a VM instance unhealthy. Default value: 2.
- **Health Check Interval**: The amount of time between health checks of an individual VM instance. Default value: 5 seconds.
- **Max Connection**: The maximum number of connections of the load balancer. Default value: 100000. Value range: 1-100,000.
- **Load Balancer Algorithm**: The routing algorithm that the load balancer uses to handle data packets. Default value: **roundrobin** (round robin).

  The supported load balancer algorithms are as follows:

  - roundrobin (round robin)

    Sequentially distributes external requests to VM instances specified by the load balancing algorithm. Each VM instance is treated equally without regardless of the actual number of connections and system load.
  - leastconn (least number of connections)

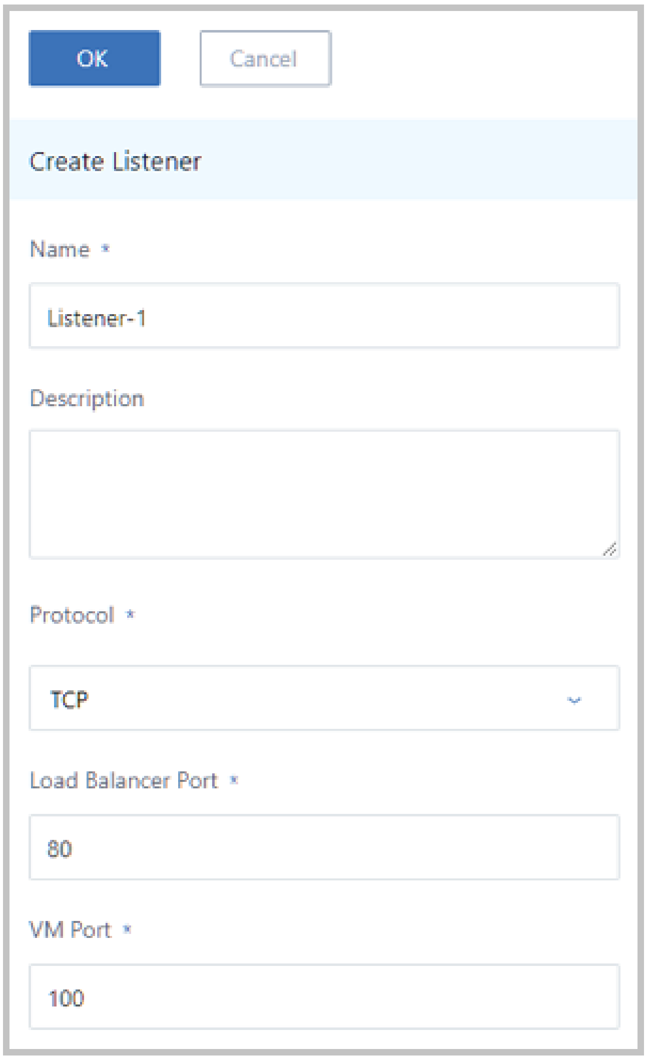
    Dynamically schedules network requests to the VM instances with the least number of established connections. If the servers (VM instances) in the cluster have similar system performance, the **leastconn** algorithm can balance the loads better.
  - source (source hashing scheduling)

    Finds out target servers from a hash table according to the source IP address (as hash key). If the target servers are available and not overloaded, requests will be sent to these servers. Otherwise, the response is null.
  - weightroundrobin (weighted round robin)

    Is a generalization of round robin scheduling, and distributes external requests to VM instances specified by the load balancing algorithm according the VM weight. VM instances with higher weight value have higher priority.

Click **OK** to complete creating the listener and adding it to the load balancer, as shown in *Create Listener*.

Figure 4-36: Create Listener



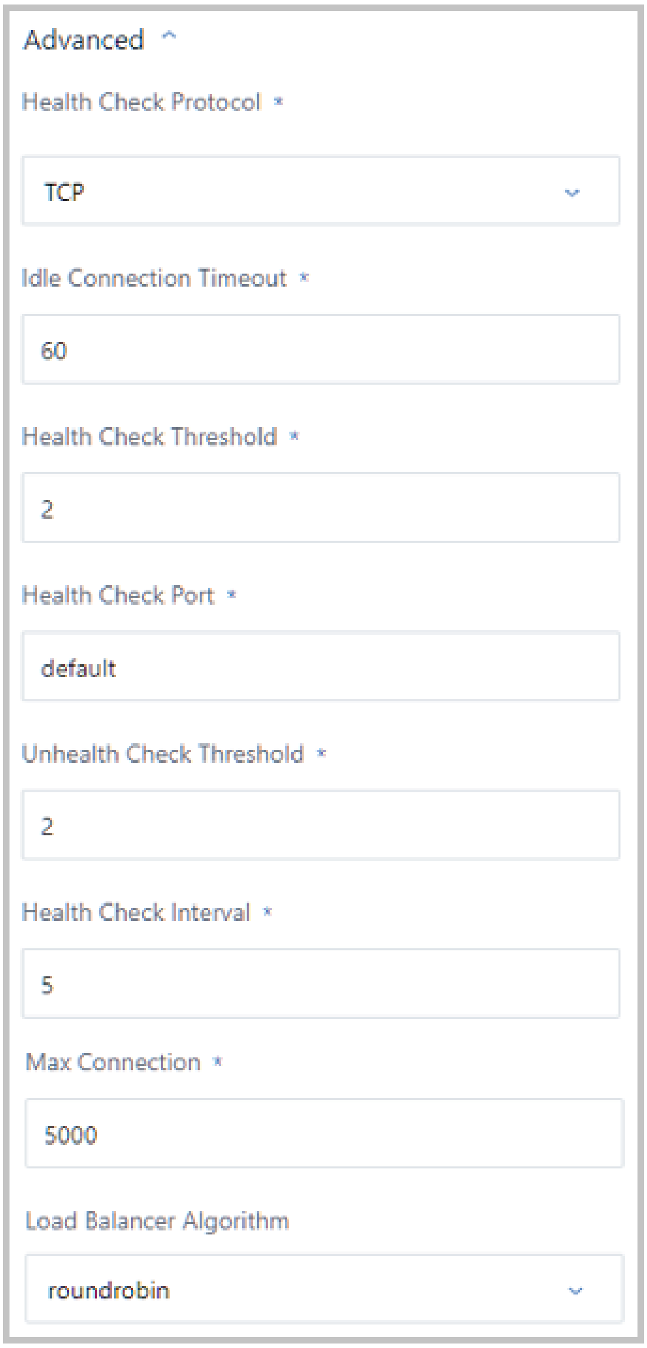
Name *: Listener-1

Description:

Protocol *: TCP

Load Balancer Port *: 80

VM Port *: 100



Advanced ^

Health Check Protocol *

TCP

Idle Connection Timeout *

60

Health Check Threshold *

2

Health Check Port *

default

Unhealth Check Threshold *

2

Health Check Interval *

5

Max Connection *

5000

Load Balancer Algorithm

roundrobin

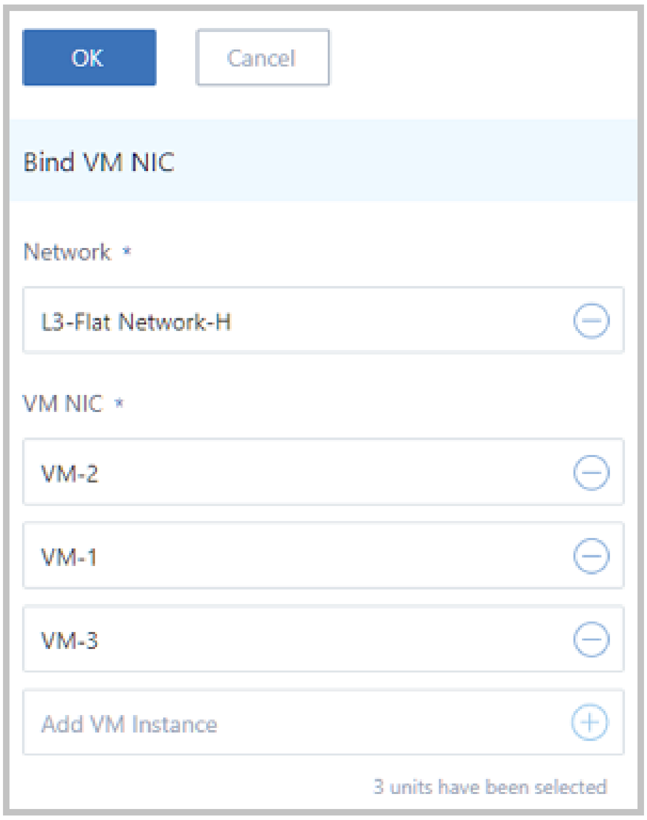
7. Bind the NICs of the VM instances to this listener.

   In the navigation pane of ZStack Private Cloud, choose **Network Service** > **Load Balancing** > **Listener**. On the **Listener** page, select a listener, and choose **Actions** > **Bind VM NIC**. On the displayed **Bind VM NIC**, set the following parameters:

   - **Network**: Select the backend network, such as L3-Backend Network-H.
   - **VM Instance**: Select the backend VM instances, such as VM-1, VM-2, and VM-3.

Click **OK** to complete adding this listener to the load balancer, as shown in *Bind VM NIC*. So far, you have completed configuring the load balancing service. Now, this service can work properly.

Figure 4-37: Bind VM NIC



Note:

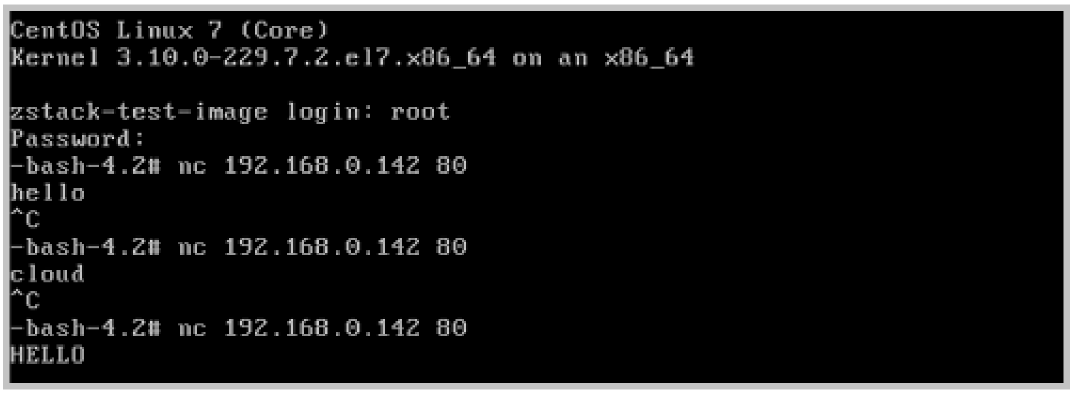
If you bind VM NICs to a listener for the first time, a vRouter will be created according to the vRouter offering attached by the backend network. This vRouter can be used to provide the load balancing service.

8. Validate this scenario.

   Based on the preceding scenario, send three messages to port 80 of the VIP (*192.168.0.142*). Then, port 100 of VM-1 (*192.168.2.233*), VM-2 (*192.168.2.237*), and VM-3 (*192.168.2.245*) will receive one message via the Round-robin method.

   Send three messages to the VIP, as shown in *VIP Receives Messages*.

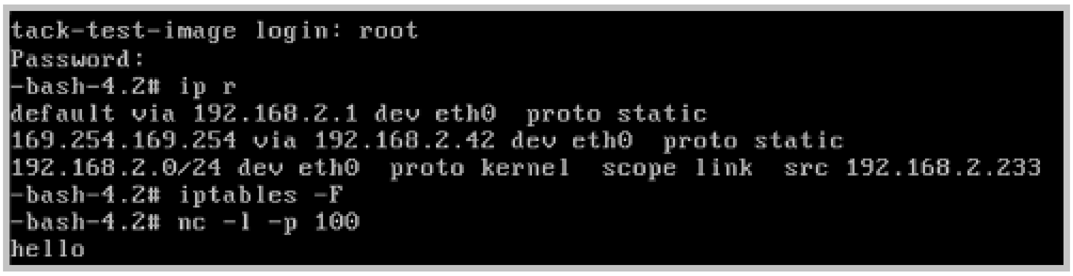
Figure 4-38: VIP Receives Messages



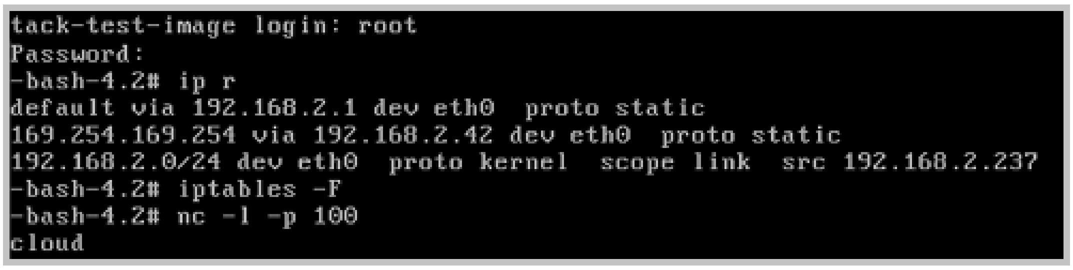
```
CentOS Linux 7 (Core)
Kernel 3.10.0-229.7.2.el7.x86_64 on an x86_64

zstack-test-image login: root
Password:
-bash-4.2# nc 192.168.0.142 80
hello
^C
-bash-4.2# nc 192.168.0.142 80
cloud
^C
-bash-4.2# nc 192.168.0.142 80
HELLO
```

Three VM instances receive one message via the Round-robin method, as shown in *Backend VM Instance Receives Message*.

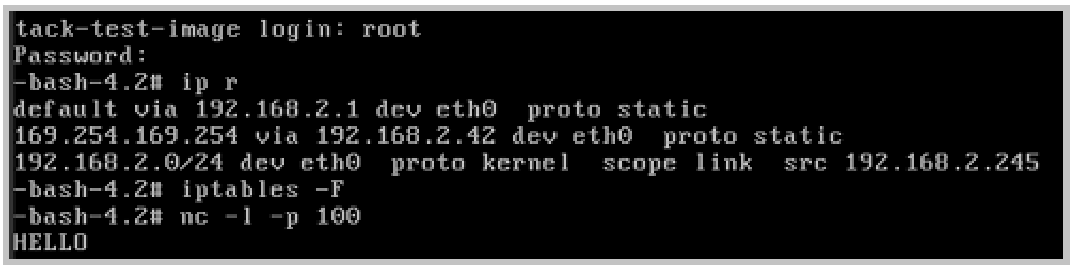
Figure 4-39: Backend VM Instance Receives Message



```
tack-test-image login: root
Password:
-bash-4.2# ip r
default via 192.168.2.1 dev eth0  proto static
169.254.169.254 via 192.168.2.42 dev eth0  proto static
192.168.2.0/24 dev eth0  proto kernel  scope link  src 192.168.2.233
-bash-4.2# iptables -F
-bash-4.2# nc -l -p 100
hello
```



```
tack-test-image login: root
Password:
-bash-4.2# ip r
default via 192.168.2.1 dev eth0  proto static
169.254.169.254 via 192.168.2.42 dev eth0  proto static
192.168.2.0/24 dev eth0  proto kernel  scope link  src 192.168.2.237
-bash-4.2# iptables -F
-bash-4.2# nc -l -p 100
cloud
```



```
tack-test-image login: root
Password:
-bash-4.2# ip r
default via 192.168.2.1 dev eth0  proto static
169.254.169.254 via 192.168.2.42 dev eth0  proto static
192.168.2.0/24 dev eth0  proto kernel  scope link  src 192.168.2.245
-bash-4.2# iptables -F
-bash-4.2# nc -l -p 100
HELLO
```

So far, we have introduced how to use the load balancing service based on a flat network.

4.6 Hardware SDN

Prerequisites

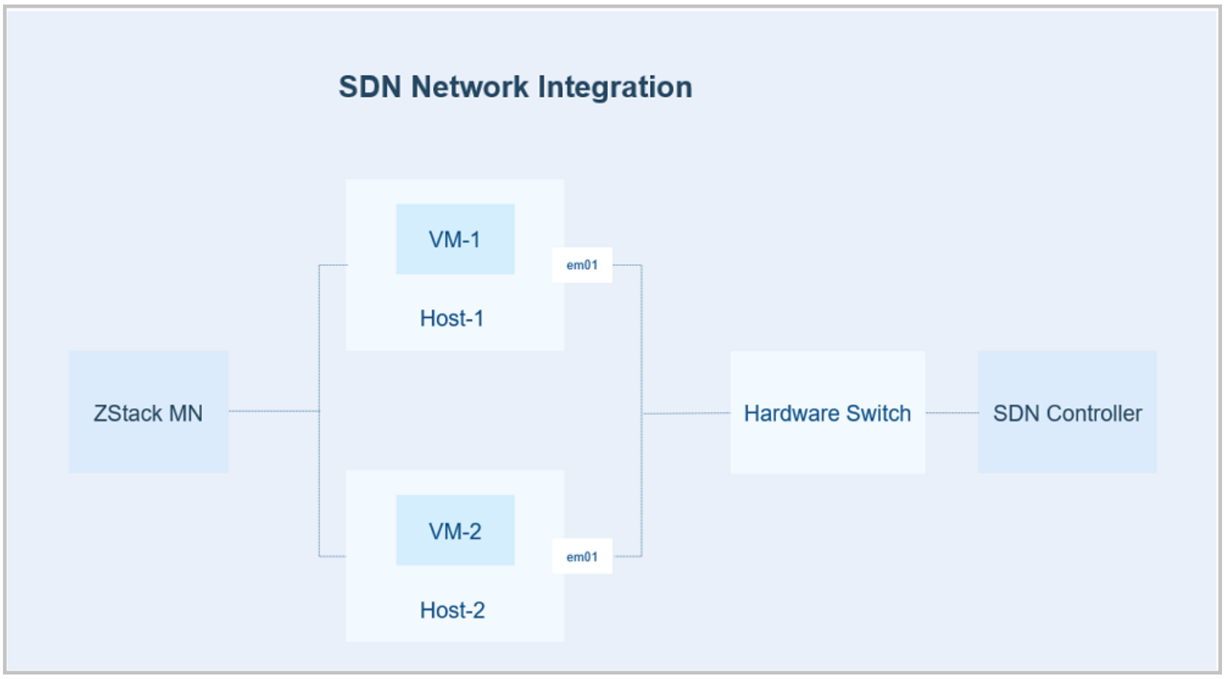
By adding an SDN controller, you can take over SDN networks of hardware switches to reduce network latencies and improve VXLAN network performances.

- To add an SDN controller to the Cloud, plan for management networks in advance, and complete preparing the basic configurations for the SDN controller.
- Currently, the Cloud only allows you to add an H3C SDN controller: VCFC.

> **Note:**
> If you use VCFC to configure hardware SDN, configure the mapping between VLAN and VXLAN on VCFC in advance.

The manipulated SDN networks of a hardware switch only support flat networks and other network services, but do not support vRouter networks or VPC, as shown in *Take over Hardware Switch SDN Network via the Cloud*.

Figure 4-40: Take over Hardware Switch SDN Network via the Cloud



Context

To take over SDN networks owned by a hardware switch

1. Plan for a management network, and complete basic configurations for an SDN controller.
2. Add an SDN controller to the Cloud.
3. Create a hardware SDN VXLAN Pool, and attach the VXLAN Pool to the corresponding cluster.
4. Create a HardwareVxlanNetwork VXLAN network based on the VXLAN Pool.

5. Create a private L3 network with the flat network type by using the VXLAN network.
6. Create two VM instances named after VM-1 and VM-2 by using the private L3 network and ensure that both VM-1 and VM-2 reside on the different hosts.
7. Test the network bandwidth intercommunicated by both VM instances, and verify performance optimizations of the VXLAN network.

Procedure

1. Plan for a management network, and complete basic configurations for an SDN controller.

   Deploy your hardware environment in advance, plan for a management network, and complete basic configurations for an SDN controller. These basic configurations include configuring virtual distributed switches and the VLAN-VXLAN mapping table.

2. Add an SDN controller to the Cloud.

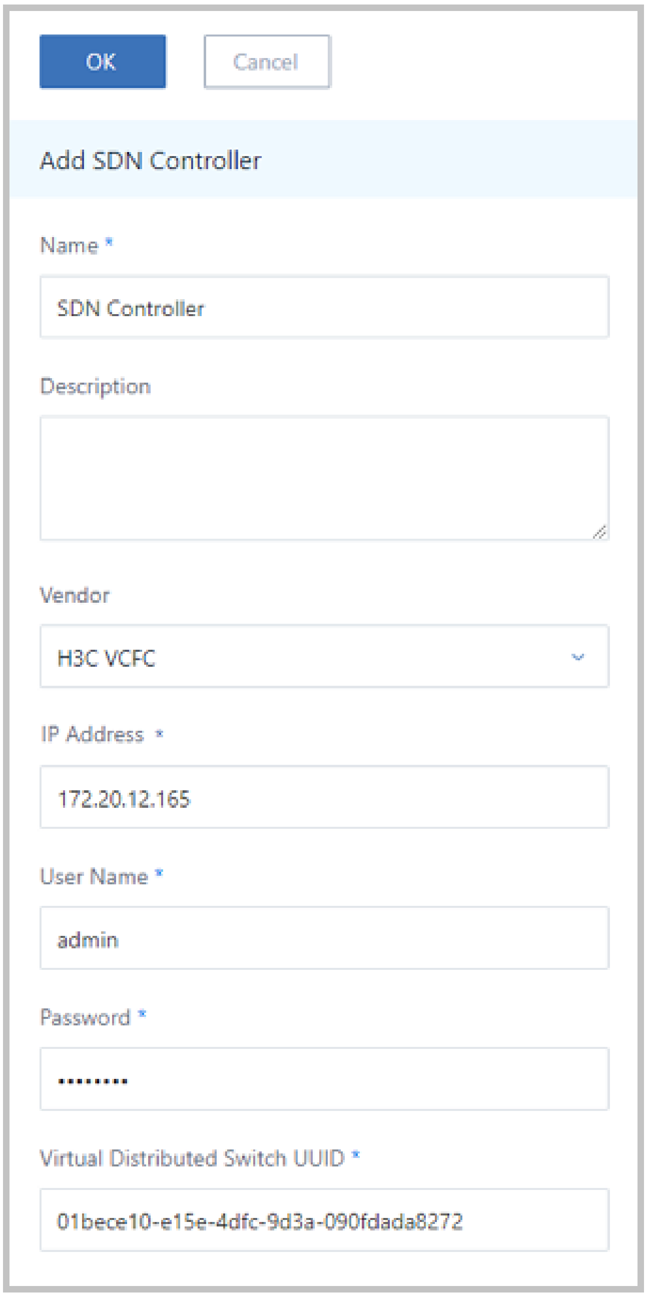
   In the navigation pane of the ZStack Private Cloud UI, choose **Network Resource** > **SDN Controller**. On the **SDN Controller** page, click **Add SDN Controller**. On the displayed **Add SDN Controller** page, set the following parameters:

   - **Name**: Enter a name for the SDN controller.
   - **Description**: Optional. Enter a description for the SDN controller.
   - **Vendor**: Select an SDN controller vendor. Currently, you are only allowed to add an H3C SDN controller: VCFC.
   - **IP**: Enter the IP address of the SDN controller.
   - **User Name**: Enter the user name of the SDN controller.
   - **Password**: Enter the password of the SDN controller.
   - **Virtual Distributed Switch UUID**: Enter a virtual distributed switch UUID.

     > **Note:**
     >
     > - Make sure that you complete configuring a virtual distributed switch in the SDN controller in advance.
     > - With this virtual distributed switch, an available VNI range of the hardware SDN VXLAN Pool can be determined.

   You can add an SDN controller, as shown in *Add SDN Controller*.

Figure 4-41: Add SDN Controller



Add SDN Controller

Name *

SDN Controller

Description

Vendor

H3C VCFC

IP Address *

172.20.12.165

User Name *

admin

Password *

••••••••

Virtual Distributed Switch UUID *

01bece10-e15e-4dfc-9d3a-090fdada8272

3. Create a hardware SDN VXLAN Pool, and attach the VXLAN Pool to the corresponding cluster.

   In the navigation pane of the ZStack Private Cloud UI, choose **Network Resource** > **L2 Network Resource** > **VXLAN Pool**. On the **VXLAN Pool** page, click **Create VXLAN Pool**. On the displayed **Create VXLAN Pool** page, set the following parameters:

   - **Name**: Enter a name for the VXLAN Pool.
   - **Description**: Optional. Enter a description for the VXLAN Pool.
   - **Type**: Select Hardware SDN.

- **SDN Controller**: Add an SDN controller to the Cloud in advance.
- **Start Vni**: Enter the start ID of HardwareVxlanNetwork.
- **End Vni**: Enter the end ID of HardwareVxlanNetwork. This end ID must be larger or equal to the start VINI.

  Note:

  The VINI range supported by a hardware SDN VXLAN Pool depends on the virtual distributed switch.
- **Cluster**: Optional. Select a cluster that you need to attach.

  Note:

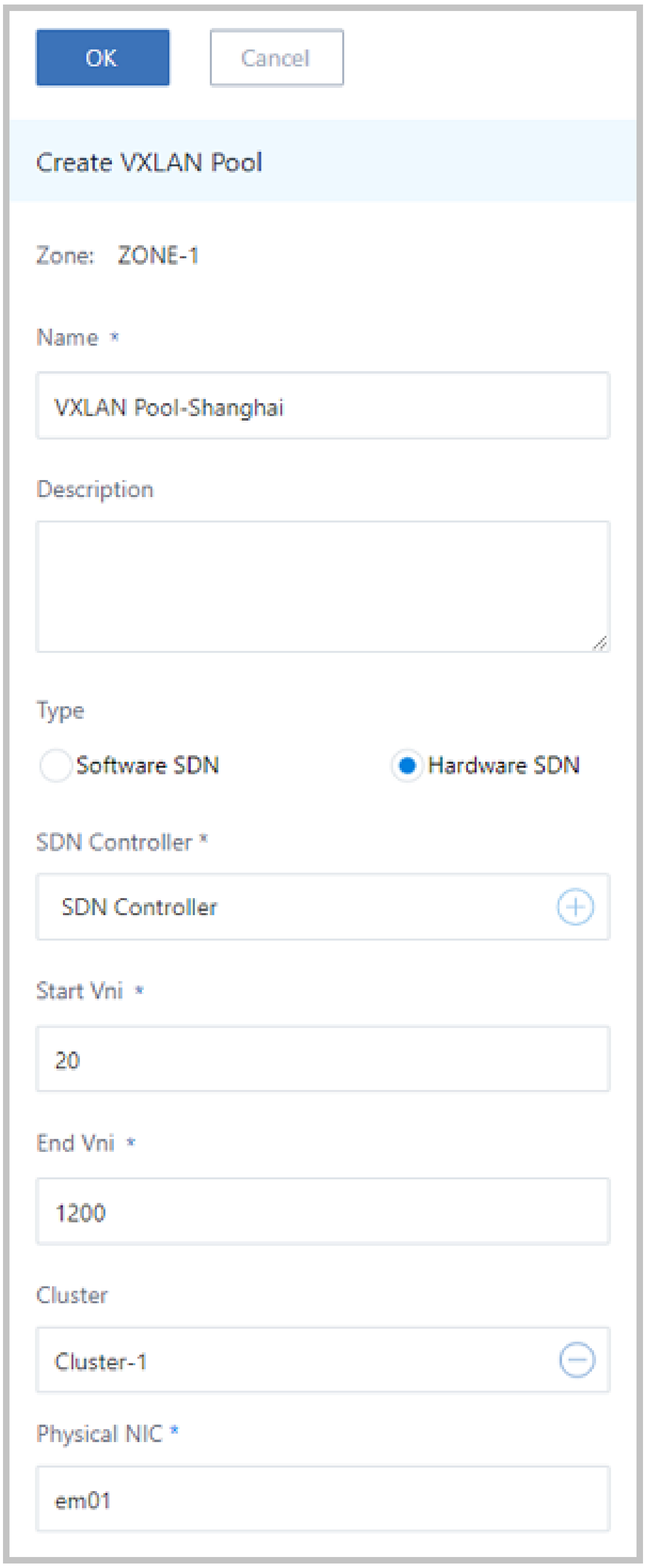
  Either attach a cluster to a VXLAN Pool when you create the VXLAN Pool or attach the cluster to the VXLAN Pool after you create the VXLAN Pool.
- **Physical NIC**: Enter a physical NIC for the VXLAN Pool.

  Note:

  The physical NIC of the cluster must connect to the switch managed by the SDN controller.

You can create a hardware SDN VXLAN Pool, as shown in *Create Hardware SDN VXLAN Pool*.

Figure 4-42: Create Hardware SDN VXLAN Pool



Create VXLAN Pool

Zone: ZONE-1

Name *

VXLAN Pool-Shanghai

Description

Type

○ Software SDN ● Hardware SDN

SDN Controller *

SDN Controller

Start Vni *

20

End Vni *

1200

Cluster

Cluster-1

Physical NIC *

em01

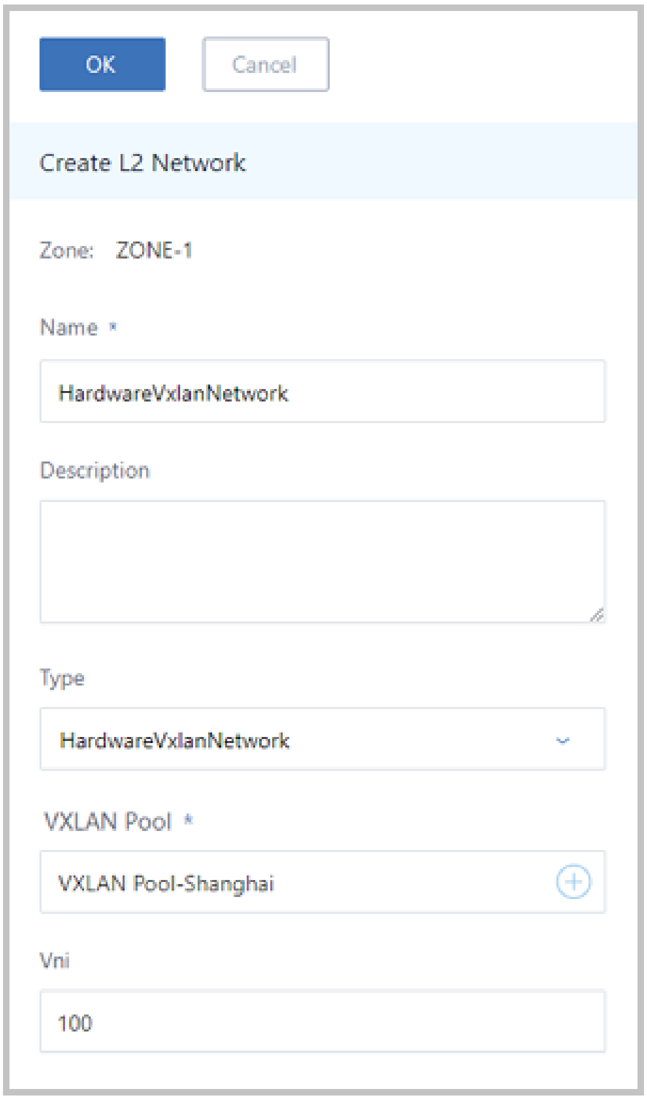
4. Create a HardwareVxlanNetwork VXLAN network based on the VXLAN Pool.

In the navigation pane of the ZStack Private Cloud UI, choose **Network Resource** > **L2 Network Resource** > **L2 Network**. On the **L2 Network** page, click **Create L2 Network**. On the displayed **Create L2 Network** page, set the following parameters:

- **Name**: Enter a name for the HardwareVxlanNetwork L2 network.
- **Description**: Optional. Enter a description for the HardwareVxlanNetwork L2 network.
- **Type**: Select HardwareVxlanNetwork.
- **VXLAN Pool**: Select a VXALN Pool with the hardware SDN type.
- **Vni**: Optional. Select the specified VNI in the VXLAN Pool. If null, the Cloud will randomly assign a VINI for you.

You can create an L2 network with the HardwareVxlanNetwork type, as shown in *Create HardwareVxlanNetwork*.

Figure 4-43: Create HardwareVxlanNetwork



Create L2 Network

Zone: ZONE-1

Name *

HardwareVxlanNetwork

Description

Type

HardwareVxlanNetwork

VXLAN Pool *

VXLAN Pool-Shanghai

Vni

100

5. Create a private L3 network with the flat network type by using the VXLAN network.

   In the navigation pane of the ZStack Private Cloud UI, choose **Network Resource** > **L3 Network** > **Private Network**. On the**Private Network** page, click **Create Private Network**. On the displayed **Create Private Network** page, set the following parameters:

   - **Name**: Enter a name for the private network.
   - **Description**: Optional. Enter a description for the private network.
   - **L2 Network**: Select an L2 network corresponded by the private network. In this scenario, select HardwareVxlanNetwork.
   - **Stop DHCP server**: Enable the DHCP service if unchecked.
   - **Network Type**: Select Flat network.

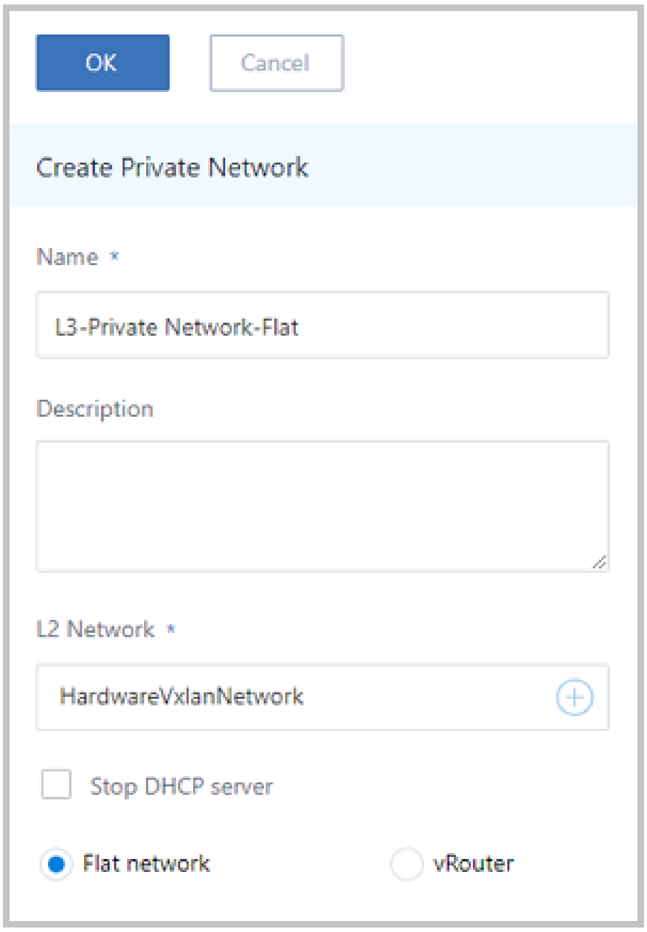
Note:

If your L2 network is with HadrewareVxlanNetwork, the private network only supports both flat networks and the corresponding network services rather than vRouter networks.

- If you select an IPv4 network address and add a network range via CIDR, set the following parameters:
  - **CIDR**: Enter a CIDR for the private network, such as *192.168.11.0/24*.
  - **Gateway**: Enter a gateway for the private network, such as *192.168.11.1*.
  - **DHCP IP**: Optional. Set a DHCP IP as needed. In this scenario, select system random allocations.
  - **Add DNS**: Set a DNS address, such as *223.5.5.5*.

You can create the L3-Private Network, as shown in *Create L3-Private Network*.

Figure 4-44: Create L3-Private Network



Create Private Network

Name *

L3-Private Network-Flat

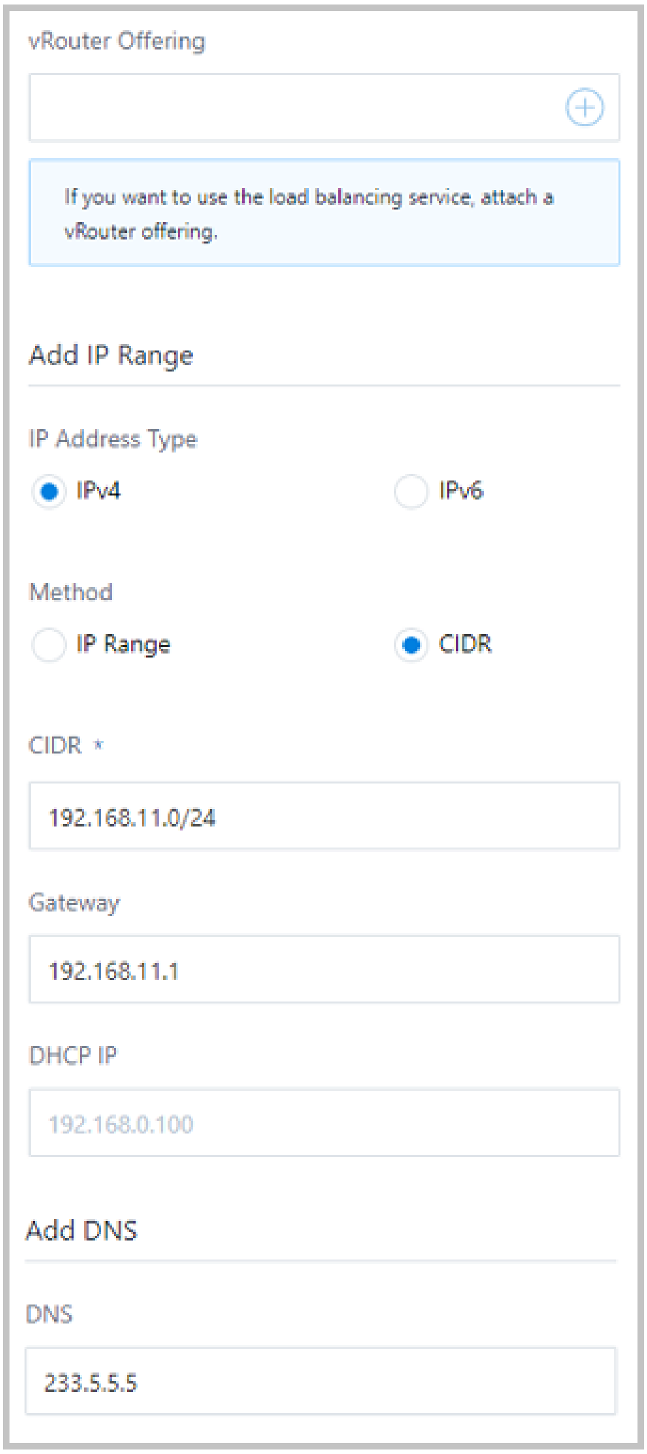
Description

L2 Network *

HardwareVxlanNetwork

□ Stop DHCP server

● Flat network ○ vRouter



vRouter Offering

If you want to use the load balancing service, attach a vRouter offering.

Add IP Range

IP Address Type

● IPv4 ○ IPv6

Method

○ IP Range ● CIDR

CIDR *

192.168.11.0/24

Gateway

192.168.11.1

DHCP IP

192.168.0.100

Add DNS

DNS

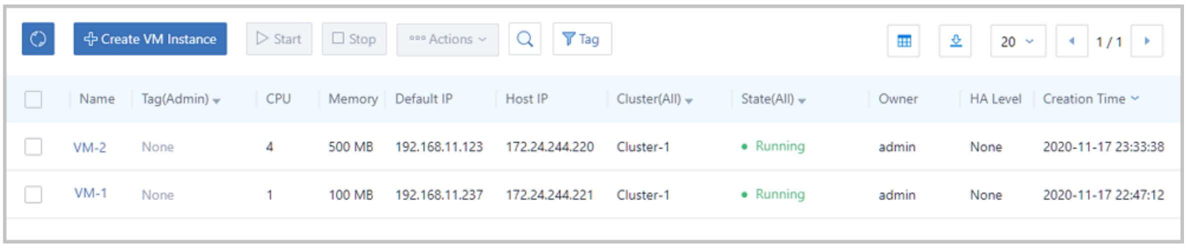
233.5.5.5

6. Create two VM instances named after VM-1 and VM-2 by using the L3 private network and ensure that both VM-1 and VM-2 reside on the different hosts.

   Create VM instances based on the flat network. For more information, see *IPv4 Flat Network Deployment* in Basic Deployment.

   The two VM instances that you created are shown in *Figure 4-45: VM-1 VM-2*.

Figure 4-45: VM-1 VM-2

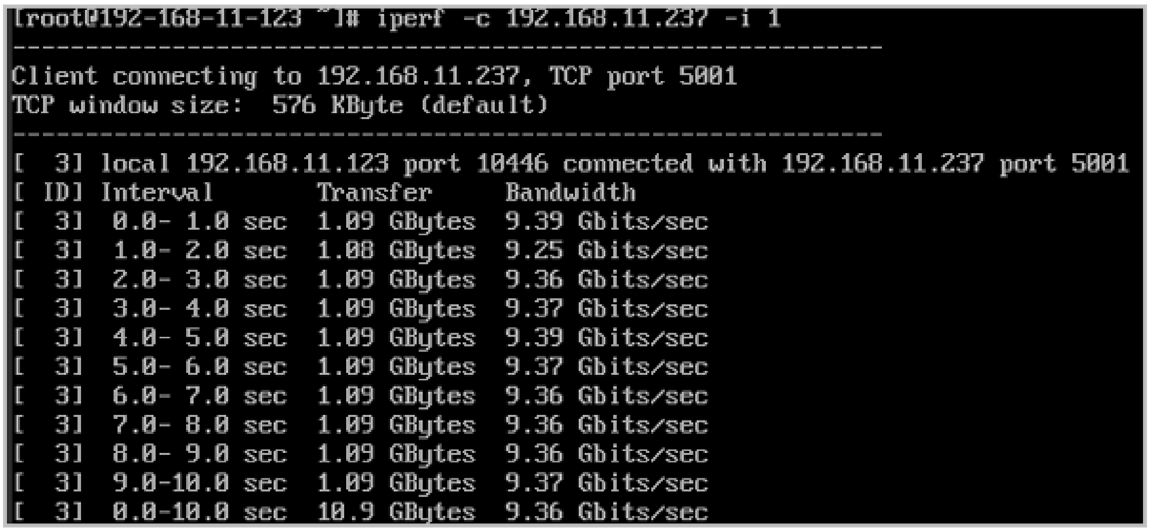


□	Name	Tag(Admin)	CPU	Memory	Default IP	Host IP	Cluster(All)	State(All)	Owner	HA Level	Creation Time
□	VM-2	None	4	500 MB	192.168.11.123	172.24.244.220	Cluster-1	● Running	admin	None	2020-11-17 23:33:38
□	VM-1	None	1	100 MB	192.168.11.237	172.24.244.221	Cluster-1	● Running	admin	None	2020-11-17 22:47:12

7. Test the network bandwidth intercommunicated by both VM instances, and verify performance optimizations of the VXLAN network.

   1. Log in to VM-1, and run `iperf -s`.
   2. Log in to VM-2, and run `iperf -c 192.168.11.237 -i 1`.
   3. The actual test result is as shown in *VM-2 Console*.

      Figure 4-46: VM-2 Console

```
[root@192-168-11-123 ~]# iperf -c 192.168.11.237 -i 1
------------------------------------------------------------
Client connecting to 192.168.11.237, TCP port 5001
TCP window size:  576 KByte (default)
------------------------------------------------------------
[  3] local 192.168.11.123 port 10446 connected with 192.168.11.237 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  1.09 GBytes  9.39 Gbits/sec
[  3]  1.0- 2.0 sec  1.08 GBytes  9.25 Gbits/sec
[  3]  2.0- 3.0 sec  1.09 GBytes  9.36 Gbits/sec
[  3]  3.0- 4.0 sec  1.09 GBytes  9.37 Gbits/sec
[  3]  4.0- 5.0 sec  1.09 GBytes  9.39 Gbits/sec
[  3]  5.0- 6.0 sec  1.09 GBytes  9.37 Gbits/sec
[  3]  6.0- 7.0 sec  1.09 GBytes  9.36 Gbits/sec
[  3]  7.0- 8.0 sec  1.09 GBytes  9.36 Gbits/sec
[  3]  8.0- 9.0 sec  1.09 GBytes  9.36 Gbits/sec
[  3]  9.0-10.0 sec  1.09 GBytes  9.37 Gbits/sec
[  3]  0.0-10.0 sec  10.9 GBytes  9.36 Gbits/sec
```

   4. Conclusion:

      In this scenario, the physical NIC is 10 gigabit NIC. Specifically, the network bandwidth intercommunicated by both VM instances reach almost 10 gigabit. In addition, the VXLAN network performance has significantly improved.

Glossary

Zone

A zone is a logical group of resources such as clusters, L2 networks, and primary storages. Zone is the largest resource scope defined in ZStack.

Cluster

A cluster is a logical group of analogy hosts (compute nodes). Hosts in the same cluster must be installed with the same operating system, have the same network configuration, and be able to access the same primary storage. In a real data center, a cluster usually maps to a rack.

Management Node

A management node is a host with operating system installed to provide UI management and Cloud deployment.

Compute Node

A compute node is a physical server (also known as a host) that provides VM instances with compute, network, and storage resources.

Primary Storage

A primary storage is a storage server used to store disk files in VM instances. Local storage, NFS, Ceph, Shared Mount Point, and Shared Block are supported.

Backup Storage

A backup storage is a storage server used to store image template files. ImageStore, SFTP (Community Edition), and Ceph are supported. We recommend that you deploy backup storage separately.

ImageStore

ImageStore is a type of backup storage. You can use ImageStore to create images for VM instances that are in the running state and manage image version updates and release. ImageStore allows you quickly upload, download, export images, and create image snapshots as needed.

VM Instance

A VM instance is a virtual machine instance running on a host. A VM instance has its own IP address to access public network and run application services.

Image

An image is an image template used by a VM instance or volume. Image templates include system volume images and data volume images.

Volume

A volume can either be a data volume or a root volume. A volume provides storage to a VM instance. A shared volume can be attached to one or more VM instances.

Instance Offering

An instance offering is a specification of the VM instance CPU and memory, and defines the host allocator strategy, disk bandwidth, and network bandwidth.

Disk Offering

A disk offering is a specification of a volume, which defines the size of a volume and how the volume will be created.

L2 Network

An L2 network is a layer 2 broadcast domain used for layer 2 isolation. Generally, L2 networks are identified by names of devices on the physical network.

L3 Network

An L3 network is a collection of network configurations for VM instances, including the IP range, gateway, and DNS.

Public Network

A public network is generally allocated with a public IP address by Network Information Center (NIC) and can be connected to IP addresses on the Internet.

Private Network

A private network is the internal network that can be connected and accessed by VM instances.

L2NoVlanNetwork

L2NoVlanNetwork is a network type for creating an L2 network. If L2NoVlanNetwork is selected, VLAN settings are not used for host connection.

L2VlanNetwork

L2VlanNetwork is a network type for creating an L2 network. If L2VlanNetwork is selected, VLAN settings are used for host connection and need to be configured on the corresponding switches in advance.

VXLAN Pool

A VXLAN pool is an underlay network in VXLAN. You can create multiple VXLAN overlay networks (VXLAN) in a VXLAN pool. The overlay networks can operate on the same underlay network device.

VXLAN

A VXLAN network is a L2 network encapsulated by using the VXLAN protocol. A VXLAN network belongs to a VXLAN pool. Different VXLAN networks are isolated from each other on the L2 network.

vRouter

A vRouter is a custom Linux VM instance that provides various network services.

Security Group

A security group provides L3 network firewall control over the VM instances. It can be used to set different security rules to filter IP addresses, network packet types, and the traffic flow of network packets.

EIP

An elastic IP address (EIP) is a method to access a private network through a public network.

Snapshot

A snapshot is a point-in-time capture of data status in a disk. A snapshot can be either an automatic snapshot or a manual snapshot.