Huawei Certification Training

# HCIE-Datacom

# Lab Guide

Version: V1.0


HUAWEI TECHNOLOGIES CO., LTD

**Trademarks and Permissions**


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

**Notice**

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

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

**Huawei Technologies Co., Ltd.**

Address: Huawei Industrial Base Bantian, Longgang Shenzhen 518129 People's Republic of China

Website: http://e.huawei.com

# Huawei Certification System

Huawei Certification is a part of the company's "Platform + Ecosystem" strategy, and it supports the ICT infrastructure featuring "Cloud-Pipe-Device". It evolves to reflect the latest trends of ICT development. Huawei Certification consists of two categories: ICT Infrastructure Certification, and Cloud Service & Platform Certification.

Huawei offers three levels of certification: Huawei Certified ICT Associate (HCIA), Huawei Certified ICT Professional (HCIP), and Huawei Certified ICT Expert (HCIE).

Huawei Certification covers all ICT fields and adapts to the industry trend of ICT convergence. With its leading talent development system and certification standards, it is committed to fostering new ICT talent in the digital era, and building a sound ICT talent ecosystem.

HCIE-Datacom V1.0 is designed to cultivate and certificate network experts with professional knowledge and skills of converged cross-field solutions in the datacom domain.

Passing HCIE-Datacom V1.0 means that you have mastered the following knowledge and skills: advanced routing & switching technology, panorama of the enterprise network architecture, typical campus network architecture and technology, CloudCampus Solution design and deployment, typical WAN architecture and technology, SD-WAN Solution design and deployment, typical bearer WAN architecture and technology, CloudWAN Solution design and deployment, and network automation technology and practice. Through HCIE-Datacom V1.0, you will obtain solid theoretical knowledge of the enterprise network cross-field solutions, and will be able to plan, construct, maintain, and optimize the enterprise campus network, WAN, and bearer WAN based on Huawei datacom products and solutions. You will be competent in all-scenario expert positions for enterprise networks, including customer managers, project managers, presales experts, post-sales experts, and network architects.


Huawei Certification
Huawei Certified ICT Expert
HCIE
Huawei Certified ICT Professional
HCIP
Huawei Certified ICT Associate
HCIA
Cloud Service & Platform
云服务与平台
HarmonyOS(To Be)
Intelligent Driving (To Be)
IoT
GaussDB
Big Data
AI
openGauss (To Be)
openEuler
Kunpeng Application Developer
Cloud(Cloud Computing + Cloud Service)
ICT Infrastructure
ICT技术架构与应用
Storage
Intelligent Computing
Enterprise Communication
Intelligent Vision
Data Center
Datacom
WLAN
Security
SDN
Transmission
Access
LTE
5G

# About This Document

## Overview

This document is applicable to the candidates who are preparing for the HCIE-Datacom exam and anyone who wants to master the following knowledge and skills: advanced routing & switching technology, panorama of the enterprise network architecture, typical campus network architecture and technology, CloudCampus Solution design and deployment, typical WAN architecture and technology, SD-WAN Solution design and deployment, typical bearer WAN architecture and technology, CloudWAN Solution design and deployment, and network automation technology and practice.

## Description

This lab guide consists of the following labs:

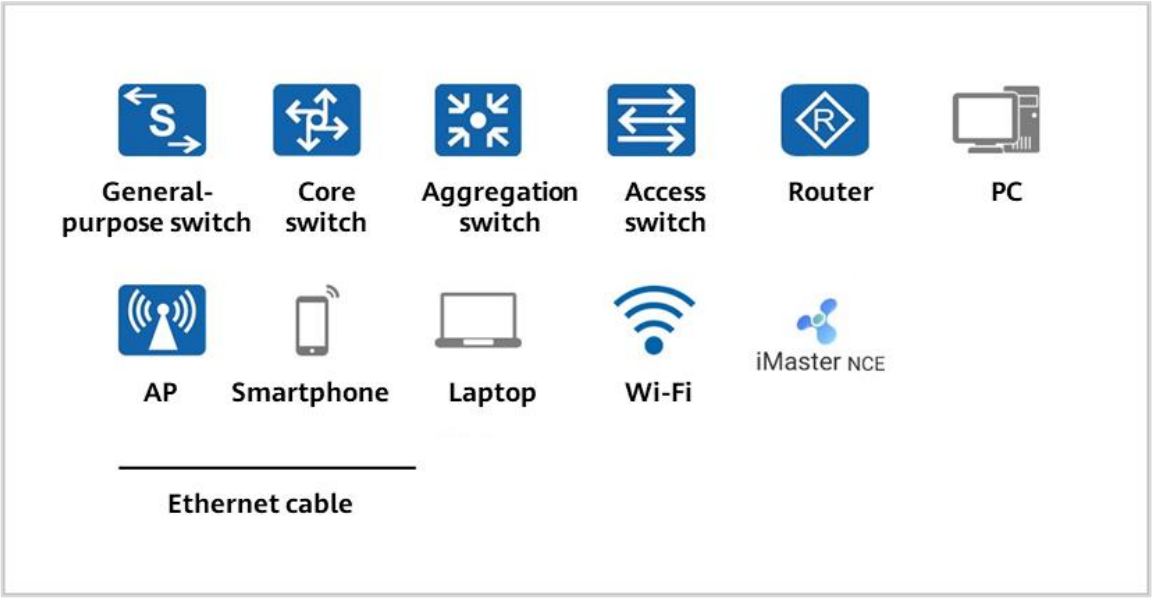
- IPv4 routing
- IPv6 routing
- MPLS VPN
- EVPN
- VXLAN
- VXLAN-based virtualized campus network deployment
- WAN interconnection network deployment
- SR-MPLS
- SRv6
- Open network programmability

## Background Knowledge Required

This course is an HCIE course. The intended audience is expected to:

- Have basic computer skills.
- Be familiar with the principles of the TCP/IP protocol stack.
- Be familiar with the basic working principles of Ethernet switches and routers.
- Have knowledge and skills described in the HCIP-Datacom-Core Technology course.

# Common Icons




# Lab Environment Overview

## Networking Introduction

This lab environment is prepared for datacom engineers who are preparing for the HCIE-Datacom exam. Each suite of lab environment includes four campus switches (PoE-incapable), one PoE switch, one interconnection switch, one AP, three enterprise egress routers, six bearer WAN routers, and some servers.

## Device Introduction

The following table lists devices recommended for HCIE-Datacom labs and the mappings between the device name, model, and software version.

| Device Name | Model | Software Version |
|---|---|---|
| Switch | CloudEngine S5731-H24T4XC | V200R020C10SPC500 or later |
| PoE switch | CloudEngine S5731-H24P4XC | V200R020C10SPC500 or later |
| Interconnection switch | S5720-52X-LI-AC | - |
| AP | AirEngine 5760-51 | - |
| Enterprise egress router | NetEngine AR6120 | V300R019C10SPC300 or later |
| Bearer WAN routers | NetEngine 8000-M6 | V800R012C10SPC300 or later |

Node: The port, output, and configuration information of devices in this document is provided based on the recommended topology. The actual information may vary according to the lab environment.

You can use a switch supporting Layer 3 functions as an interconnection switch, with no specific requirement on the version.

# Contents

**About This Document ..... 3**
Overview ..... 3
Description ..... 3
Background Knowledge Required ..... 3
Common Icons ..... 4
Lab Environment Overview ..... 4
**1 IPv4 Routing ..... 9**
1.1 Advanced IGP Features ..... 9
1.1.1 About This Lab ..... 9
1.1.2 Lab Task ..... 10
1.1.3 Quiz ..... 30
1.2 Advanced BGP Features ..... 31
1.2.1 About This Lab ..... 31
1.2.2 Lab Task ..... 32
1.2.3 Quiz ..... 48
**2 IPv6 Routing ..... 49**
2.1 IPv6 Routing ..... 49
2.1.1 About This Lab ..... 49
2.1.2 Lab Task ..... 49
2.1.3 Quiz ..... 66
**3 MPLS VPN ..... 67**
3.1 MPLS VPN ..... 67
3.1.1 About This Lab ..... 67
3.1.2 Lab Task ..... 67
3.1.3 Quiz ..... 83
**4 EVPN ..... 84**
4.1 EVPN L3VPNv4 over MPLS ..... 84
4.1.1 About This Lab ..... 84
4.1.2 Lab Task ..... 84
4.1.3 Quiz ..... 101
**5 VXLAN Lab ..... 102**
5.1 Layer 2 Interconnection Through a Static VXLAN Tunnel ..... 102
5.1.1 About This Lab ..... 102
5.1.2 Lab Task ..... 103

5.1.3 Quiz ..... 111
5.2 Centralized VXLAN Gateway ..... 112
5.2.1 About This Lab ..... 112
5.2.2 Configuration Procedure ..... 113
5.2.3 Quiz ..... 123
5.3 Distributed VXLAN Gateway ..... 124
5.3.1 About This Lab ..... 124
5.3.2 Configuration Procedure ..... 125
5.3.3 Quiz ..... 144
**6 VXLAN-based Virtualized Campus Network Deployment ..... 145**
6.1 VXLAN-based Virtualized Campus Network Deployment ..... 145
6.1.1 About This Lab ..... 145
6.1.2 Configuration for Lab Tasks ..... 162
6.1.3 (Optional) Clearing Configurations ..... 230
6.1.4 Quiz ..... 234
**7 WAN Interconnection Network Deployment ..... 235**
7.1 WAN Interconnection Network Deployment ..... 235
7.1.1 About This Lab ..... 235
7.1.2 Lab Task ..... 243
7.1.3 (Optional) Clearing Configurations ..... 307
7.1.4 Quiz ..... 311
**8 SR-MPLS ..... 312**
8.1 L3VPNv4 over SR-MPLS BE ..... 312
8.1.1 About This Lab ..... 312
8.1.2 Lab Task ..... 313
8.1.3 Quiz ..... 327
8.2 L3VPNv4 over SR-MPLS TE ..... 328
8.2.1 About This Lab ..... 328
8.2.2 Lab Task ..... 328
8.2.3 Quiz ..... 346
8.3 L3VPNv4 over SR-MPLS Policy ..... 347
8.3.1 About This Lab ..... 347
8.3.2 Lab Task ..... 347
8.3.3 Quiz ..... 369
**9 SRv6 ..... 370**
9.1 L3VPNv4 over SRv6 BE ..... 370
9.1.1 About This Lab ..... 370
9.1.2 Lab Task ..... 370

9.1.3 Quiz ..... 381
9.2 EVPN L3VPN over SRv6 Policy ..... 382
9.2.1 About This Lab ..... 382
9.2.2 Lab Task ..... 382
9.2.3 Quiz ..... 407
**10 Open Network Programmability ..... 408**
10.1 SSH Lab ..... 408
10.1.1 About This Lab ..... 408
10.1.2 Lab Task ..... 408
10.1.3 Quiz ..... 420
10.2 NETCONF Lab ..... 421
10.2.1 About This Lab ..... 421
10.2.2 Lab Task ..... 421
10.2.3 Quiz ..... 429
10.3 OPS Lab ..... 430
10.3.1 About This Lab ..... 430
10.3.2 Lab Task ..... 430
10.3.3 Quiz ..... 436
Reference Answers to Quiz ..... 437

# 1 IPv4 Routing

## 1.1 Advanced IGP Features

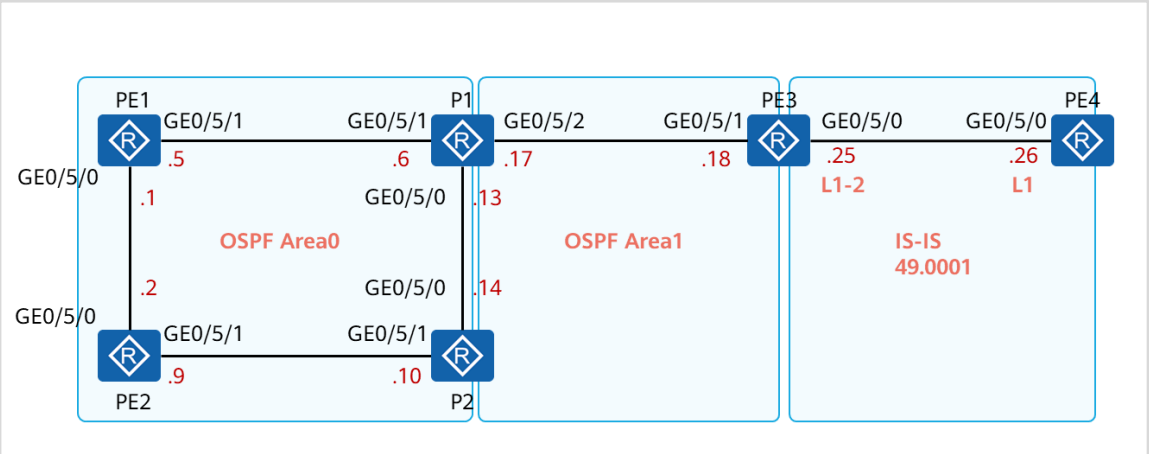
### 1.1.1 About This Lab

#### 1.1.1.1 Objectives

- Configure fast reroute (FRR) and Bidirectional Forwarding Detection (BFD) to accelerate OSPF convergence.
- Adjust the cost of OSPF routes using routing policies.
- Configure inter-area route filtering to reduce the size of link state database (LSDB) for Open Shortest Path First (OSPF).

#### 1.1.1.2 Networking Description

**Figure 1-1 Advanced IGP features**




The figure shows the device connection and IP address planning. The interface interconnection addresses are in the format of 10.0.0.*Y*/30, and the values represented by Y are shown in the figure. Loopback0 is created on all devices, and loopback interface IP addresses are in the format of 10.0.*X.X*. The specific IP addresses are listed in a table in the section "Configuration Procedure".

The interconnection interfaces of PE1, PE2, P1, and P2 belong to OSPF area 0. Loopback0 interfaces of the four devices also belong to OSPF area 0. The interconnection interfaces of P1 and PE3 as well as PE3's Loopback0 belong to OSPF area 1.

PE3 and PE4 belong to IS-IS area 49.0001. PE4 is an IS-IS Level-1 router and PE3 is an IS-IS Level-1-2 router.

This lab covers the following tasks:

1. Adjust the OSPF cost of interfaces so that traffic between PE2 and P1 in area 0 is preferentially transmitted over the link with higher bandwidth, that is, transmitted through PE1.
2. Enable OSPF IP FRR on PE2 so that OSPF can generate a backup route to the loopback interface of P1.
3. Enable BFD for OSPF in the entire OSPF area 0.
4. Configure route filtering on P1 to limit the routes that enter OSPF area 0, so as to control the number of routing entries in area 0.
5. Create a default route on PE3 that simulates the egress of the entire network, and advertise the route to the IS-IS.

## 1.1.2 Lab Task

### 1.1.2.1 Configuration Roadmap

1. Configure IP addresses for the devices.
2. Configure OSPF as planned.
3. Configure IS-IS as planned.
4. Adjust the OSPF cost of the interfaces between PE2 and P1 so that PE2-to-P1 traffic is preferentially forwarded through PE1. Enable OSPF IP FRR on PE2 to generate a backup route to P1.
5. Enable BFD in the OSPF area 0 to speed up OSPF convergence.
6. Create Loopback2 interfaces with the same IP address on PE1 and P2, and enable OSPF on these interfaces. Check whether equal-cost routes exist in the OSPF routing table on P1 and limit the number of equal-cost routes to 1. Create a Loopback3 interface on PE3 and enable OSPF. Configure inter-area route filtering on ABR P1 to prevent the Loopback3 route on PE3 from being transmitted to area 0.
7. Advertise a default route in IS-IS process 1 on PE3.

### 1.1.2.2 Configuration Procedure

Step 1 Complete basic device configuration.

Configure the configuration validation mode as immediate validation, and configure IP addresses for interconnection and Loopback0 interfaces. Loopback0 addresses must be configured as planned in the following table.

**Table 1-1 Loopback0 IP addresses**

| Device | X value | Loopback0 IP Address |
|---|---|---|
| PE1 | 1 | 10.0.1.1 |
| PE2 | 2 | 10.0.2.2 |
| PE3 | 3 | 10.0.3.3 |

| Device | X value | Loopback0 IP Address |
|---|---|---|
| PE4 | 4 | 10.0.4.4 |
| P1 | 5 | 10.0.5.5 |
| P2 | 6 | 10.0.6.6 |

# Name the devices.

N/A

# Disable unnecessary interfaces in this lab.

N/A

# Set the configuration validation mode to immediate validation.

```
<PE1>system-view immediately
```

```
<PE2>system-view immediately
```

```
<PE3>system-view immediately
```

```
<PE4>system-view immediately
```

```
<P1>system-view immediately
```

```
<P2>system-view immediately
```

# Disable data communication network (DCN) globally on each device.

```
[PE1] undo dcn
Warning: This operation will disable DCN function. Continue? [Y/N]:y
```

Here, PE1 is used as an example. Repeat this operation on other devices.

By default, DCN is enabled on NE router interfaces. To facilitate experiments in the lab, disable DCN globally on all devices.

# Configure IP addresses for the interconnection interfaces and Loopback0 interface on PE1.

```
[PE1]interface LoopBack0
[PE1-LoopBack0] ip address 10.0.1.1 32
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0]ip address 10.0.0.1 30
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
[PE1-GigabitEthernet0/5/1]ip address 10.0.0.5 30
[PE1-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection interfaces and Loopback0 interface on PE2.

```
[PE2]interface LoopBack0
[PE2-LoopBack0] ip address 10.0.2.2 32
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ip address 10.0.0.2 30
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ip address 10.0.0.9 30
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection interfaces and Loopback0 interface on PE3.

```
[PE3]interface LoopBack0
[PE3-LoopBack0] ip address 10.0.3.3 32
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] ip address 10.0.0.25 30
[PE3-GigabitEthernet0/5/0] quit
[PE3]interface GigabitEthernet0/5/1
[PE3-GigabitEthernet0/5/1] ip address 10.0.0.18 30
[PE3-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection interface and Loopback0 interface on PE4.

```
[PE4]interface LoopBack0
[PE4-LoopBack0] ip address 10.0.4.4 32
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] ip address 10.0.0.26 30
[PE4-GigabitEthernet0/5/0] quit
```

# Configure IP addresses for the interconnection interfaces and Loopback0 interface on P1.

```
[P1]interface LoopBack0
[P1-LoopBack0] ip address 10.0.5.5 32
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
```

```
[P1-GigabitEthernet0/5/0] ip address 10.0.0.13 30
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] ip address 10.0.0.6 30
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] ip address 10.0.0.17 30
[P1-GigabitEthernet0/5/2] quit
```

# Configure IP addresses for the interconnection interfaces and Loopback0 interface on P2.

```
[P2]interface LoopBack0
[P2-LoopBack0] ip address 10.0.6.6 32
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ip address 10.0.0.14 30
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ip address 10.0.0.10 30
[P2-GigabitEthernet0/5/1] quit
```

# Test connectivity between interconnection interfaces on PE1, P2, and PE3.

```
[PE1]ping -c 1 10.0.0.6
  PING 10.0.0.6: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.6: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.6 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

[PE1]ping -c 1 10.0.0.2
  PING 10.0.0.2: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.2: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.2 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 1/1/1 ms


[P2]ping -c 1 10.0.0.9
  PING 10.0.0.9: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.9: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.9 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

```
[P2]ping -c 1 10.0.0.13
  PING 10.0.0.13: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.13: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.13 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

[PE3]ping -c 1 10.0.0.17
  PING 10.0.0.17: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.17: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.17 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

[PE3]ping -c 1 10.0.0.26
  PING 10.0.0.26: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.26: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.26 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

The interconnection interfaces can communicate with each other properly.

Step 2 Configure OSPF.

Configure OSPF on PE1, PE2, P1, P2, and PE3 as planned. Configure Loopback0 IP addresses as router IDs and set the OSPF process ID to 1. Enable OSPF on the corresponding interfaces.

# Configure OSPF on PE1, and enable OSPF on the interconnection and Loopback0 interfaces.

```
[PE1]ospf 1 router-id 10.0.1.1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
[PE1]interface LoopBack0
[PE1-LoopBack0] ospf enable 1 area 0
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0] ospf enable 1 area 0
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
[PE1-GigabitEthernet0/5/1] ospf enable 1 area 0
[PE1-GigabitEthernet0/5/1] quit
```

# Configure OSPF on PE2, and enable OSPF on the interconnection and Loopback0 interfaces.

```
[PE2]ospf 1 router-id 10.0.2.2
[PE2-ospf-1] area 0
[PE2-ospf-1-area-0.0.0.0] quit
[PE2-ospf-1] quit
[PE2]interface LoopBack0
[PE2-LoopBack0] ospf enable 1 area 0
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ospf enable 1 area 0
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ospf enable 1 area 0
[PE2-GigabitEthernet0/5/1] quit
```

# Configure OSPF on PE3, and enable OSPF on the interconnection and Loopback0 interfaces.

```
[PE3]ospf 1 router-id 10.0.3.3
[PE3-ospf-1] area 1
[PE3-ospf-1-area-0.0.0.1] quit
[PE3-ospf-1] quit
[PE3]interface LoopBack0
[PE3-LoopBack0] ospf enable 1 area 1
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/1
[PE3-GigabitEthernet0/5/1] ospf enable 1 area 1
[PE3-GigabitEthernet0/5/1] quit
```

# Configure OSPF on P1, and enable OSPF on the interconnection and Loopback0 interfaces.

```
[P1]ospf 1 router-id 10.0.5.5
[P1-ospf-1] area 0
[P1-ospf-1-area-0.0.0.0]quit
[P1-ospf-1] area 1
[P1-ospf-1-area-0.0.0.1] quit
[P1-ospf-1] quit
[P1]interface LoopBack0
[P1-LoopBack0] ospf enable 1 area 0
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] ospf enable 1 area 0
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] ospf enable 1 area 0
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] ospf enable 1 area 1
[P1-GigabitEthernet0/5/2] quit
```

# Configure OSPF on P2, and enable OSPF on the interconnection and Loopback0 interfaces.

```
[P2]ospf 1 router-id 10.0.6.6
[P2-ospf-1] area 0
[P2-ospf-1-area-0.0.0.0] quit
[P2-ospf-1] quit
[P2]interface LoopBack0
[P2-LoopBack0] ospf enable 1 area 0
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ospf enable 1 area 0
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ospf enable 1 area 0
[P2-GigabitEthernet0/5/1] quit
```

# Check OSPF neighbor relationships on P1 and PE2.

```
[P1]display ospf peer brief
(M) Indicates MADJ neighbor


             OSPF Process 1 with Router ID 10.0.5.5
                     Peer Statistic Information
Total number of peer(s): 3
 Peer(s) in full state: 3
--------------------------------------------------------------------------------
 Area Id          Interface                     Neighbor id          State
 0.0.0.0          GE0/5/0                       10.0.6.6             Full
 0.0.0.0          GE0/5/1                       10.0.1.1             Full
 0.0.0.1          GE0/5/2                       10.0.3.3             Full
--------------------------------------------------------------------------------
```

```
[PE2]display ospf peer brief
(M) Indicates MADJ neighbor


             OSPF Process 1 with Router ID 10.0.2.2
                     Peer Statistic Information
Total number of peer(s): 2
 Peer(s) in full state: 2
--------------------------------------------------------------------------------
 Area Id          Interface                     Neighbor id          State
 0.0.0.0          GE0/5/0                       10.0.1.1             Full
 0.0.0.0          GE0/5/1                       10.0.6.6             Full
--------------------------------------------------------------------------------
```

OSPF neighbor relationships have been established.

# Check OSPF routes on PE2.

```
[PE2]display ospf routing
        OSPF Process 1 with Router ID 10.0.2.2
                Routing Tables

Routing for Network
Destination        Cost    Type       NextHop      AdvRouter      Area
10.0.0.0/30        1       Direct     10.0.0.2     10.0.2.2       0.0.0.0
10.0.0.4/30        2       Transit    10.0.0.1     10.0.5.5       0.0.0.0
10.0.0.8/30        1       Direct     10.0.0.9     10.0.2.2       0.0.0.0
10.0.0.12/30       2       Transit    10.0.0.10    10.0.5.5       0.0.0.0
10.0.0.16/30       3       Inter-area 10.0.0.10    10.0.5.5       0.0.0.0
10.0.0.16/30       3       Inter-area 10.0.0.1     10.0.5.5       0.0.0.0
10.0.1.1/32        1       Stub       10.0.0.1     10.0.1.1       0.0.0.0
10.0.2.2/32        0       Direct     10.0.2.2     10.0.2.2       0.0.0.0
10.0.3.3/32        3       Inter-area  10.0.0.10   10.0.5.5       0.0.0.0
10.0.3.3/32        3       Inter-area  10.0.0.1    10.0.5.5       0.0.0.0
10.0.5.5/32        2       Stub        10.0.0.10   10.0.5.5       0.0.0.0
10.0.5.5/32        2       Stub        10.0.0.1    10.0.5.5       0.0.0.0
10.0.6.6/32        1       Stub        10.0.0.10   10.0.6.6       0.0.0.0

Total Nets:10
Intra Area: 8  Inter Area: 2  ASE:0  NSSA: 0
```

PE2 has learned routes in the entire OSPF area 0. PE2 has two equal-cost routes to Loopback0 of PE3.

Step 3 Configure IS-IS.

Configure IS-IS processes on routers one by one according to the topology design. Set the process ID to 1, the network entity title (NET) of PE3 to 3, and the NET of PE4 to 4. For example, the NET of PE3 is: 49.0001.0000.0000.0003.00.

# Configure IS-IS on PE3.

```
[PE3]isis 1
[PE3-isis-1] is-level level-1-2
[PE3-isis-1] network-entity 49.0001.0000.0000.0003.00
[PE3-isis-1] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] isis enable 1
[PE3-GigabitEthernet0/5/0] quit
```

# Configure IS-IS on PE4.

```
[PE4]isis 1
[PE4-isis-1] is-level level-1
[PE4-isis-1] network-entity 49.0001.0000.0000.0004.00
[PE4-isis-1] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] isis enable 1
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface LoopBack 0
[PE4-LoopBack0] isis enable 1
[PE4-LoopBack0] quit
```

# Check IS-IS neighbor relationships and the IS-IS routing table on PE3.

```
[PE3]display isis peer

                    Peerinformation forISIS(1)

 SystemId     Interface          CircuitId            State HoldTime    Type     PRI
------------------------------------------------------------------------------
0000.0000.0004  GE0/5/0        0000.0000.0004.01  Up    9s          L1       64

TotalPeer(s):1
```

PE3 and PE4 have established an IS-IS neighbor relationship.

```
[PE3]display isis route 10.0.4.4

                          Route information for ISIS(1)
                          ----------------------------

                         ISIS(1) Level-1 Forwarding Table
                         --------------------------------

IPV4 Destination     IntCost      ExtCost ExitInterface      NextHop            Flags
-------------------------------------------------------------------------------
10.0.4.4/32          10           NULL    GE0/5/0        10.0.0.26          A/-/L/-
      Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
             U-Up/Down Bit Set, LP-Local Prefix-Sid
      Protect Type: L-Link Protect, N-Node Protect
```

PE3 has learned the route generated by Loopback0 of PE4 through IS-IS.

## Step 4 Adjust the OSPF cost and enable OSPF IP FRR.

Adjust the OSPF cost of the interfaces between PE2 and P2 so that PE2-to-P1 traffic is preferentially forwarded through PE1. Enable OSPF IP FRR on PE2 to generate a backup route to P1.

# Check the OSPF route to 10.0.5.5 on PE2 before the adjustment.

```
[PE2]display ospf routing 10.0.5.5 32
        OSPF Process 1withRouter ID 10.0.2.2

Destination  :10.0.5.5/32
AdverRouter  :10.0.5.5          Area          :0.0.0.0
Cost         :2                 Type          :Stub
NextHop      :10.0.0.10         Interface     :GE0/5/1
Priority     :Medium            Age           :00h00m01s

Destination  :10.0.5.5/32
AdverRouter  :10.0.5.5          Area          :0.0.0.0
Cost         :2                 Type          :Stub
NextHop      :10.0.0.1          Interface     :GE0/5/0
Priority     :Medium            Age           :00h00m01s
```

According to the command output, there are two equal-cost routes to 10.0.5.5, with the next hops being PE1 and P2 respectively.

# Adjust the OSPF cost of the interface connecting PE2 to P2.

```
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ospf cost 2
```

# Check the OSPF route to 10.0.5.5 on PE2 again.

```
[PE2]display ospf routing 10.0.5.5 32

        OSPF Process 1 with Router ID 10.0.2.2

Destination   : 10.0.5.5/32
AdverRouter   : 10.0.5.5          Area          : 0.0.0.0
Cost          : 2                 Type          : Stub
NextHop       : 10.0.0.1          Interface     : GE0/5/0
Priority      : Medium            Age           : 00h01m28s
```

Only one route is available, with the next hop being PE1.

# Configure OSPF IP FRR on PE2.

```
[PE2]ospf 1
[PE2-ospf-1] frr
[PE2-ospf-1-frr] loop-free-alternate
[PE2-ospf-1-frr] quit
[PE2-ospf-1] quit
```

# Check the OSPF route to 10.0.5.5 on PE2 again.

```
[PE2]display ospf routing 10.0.5.5 32

        OSPF Process 1 with Router ID 10.0.2.2

Destination     : 10.0.5.5/32
AdverRouter     : 10.0.5.5          Area          : 0.0.0.0
Cost            : 2                 Type          : Stub
NextHop         : 10.0.0.1          Interface     : GE0/5/0
Priority        : Medium            Age           : 00h00m54s
BackupNextHop   : 10.0.0.10         BackupInterface : GE0/5/1
BackupType      : LFA LINK-NODE
```

PE2 has generated a backup route to P1's Loopback0 interface. The next hop address is 10.0.0.10 and the outbound interface is GE0/5/1. This means the backup route goes to P1's Loopback0 interface through P2.

Step 5 Configure BFD for OSPF.

To accelerate OSPF convergence, enable BFD on all routers in the OSPF area 0, configure BFD on the interconnection interfaces, and configure GE0/5/0 of PE2 to filter incoming

BFD packets. Check whether PE2 can quickly detect the BFD session interruption and trigger OSPF route switchover.

# Enable BFD globally on all devices and enable BFD for OSPF.

```
[PE1] bfd
[PE1-bfd] quit
[PE1] ospf 1
[PE1-ospf-1] bfd all-interfaces enable
```

PE1 is used as an example. Configurations on other devices are the same and are not provided here.

# Enable BFD for interfaces. Set the minimum interval at which BFD packets are sent to and received from a neighbor to 500 ms, and set the local detection multiple to 3.

```
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0] ospf bfd enable
[PE1-GigabitEthernet0/5/0] ospf bfd min-tx-interval 500 min-rx-interval 500 detect-multiplier 3
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
[PE1-GigabitEthernet0/5/1] ospf bfd enable
[PE1-GigabitEthernet0/5/1] ospf bfd min-tx-interval 500 min-rx-interval 500 detect-multiplier 3
[PE1-GigabitEthernet0/5/1] quit
```

PE1 is used as an example. The configurations of other devices are the same.

# Check the OSPF BFD session status on P1 and PE2.

```
[P1]display ospf bfd session all
            OSPF Process 1 with Router ID 10.0.5.5

  Area 0.0.0.0 interface 10.0.0.13 (GE0/5/0)'s BFD Sessions

 NeighborId      :10.0.6.6           AreaId:0.0.0.0              Interface:GE0/5/0
 BFDState        :Up                  rx      :500                   tx          :500
 Multiplier      :3                   BFD Local Dis:16385     LocalIpAdd:10.0.0.13
 RemoteIpAdd :10.0.0.14          Diagnostic Info:No diagnostic information

  Area 0.0.0.0 interface 10.0.0.6 (GE0/5/1)'s BFD Sessions

 NeighborId      :10.0.1.1           AreaId:0.0.0.0              Interface:GE0/5/1
 BFDState        :Up                  rx      :500                   tx          :500
 Multiplier       :3                   BFD Local Dis:16386     LocalIpAdd:10.0.0.6
 RemoteIpAdd   :10.0.0.5          Diagnostic Info:No diagnostic information

  Area 0.0.0.1 interface 10.0.0.17 (GE0/5/2)'s BFD Sessions

 NeighborId     :10.0.3.3           AreaId:0.0.0.1              Interface:GE0/5/2
 BFDState       :Up                  rx      :500                   tx          :500
 Multiplier      :3                   BFD Local Dis:16387     LocalIpAdd:10.0.0.17
 RemoteIpAdd :10.0.0.18          Diagnostic Info:No diagnostic information

     Total UP/DOWN/UNKNOWN BFD Session Number : 3 / 0 / 0
```

```
[PE2]display ospf bfd session all
            OSPF Process 1 with Router ID 10.0.2.2

  Area 0.0.0.0 interface 10.0.0.2 (GE0/5/0)'s BFD Sessions

 NeighborId      :10.0.1.1          AreaId:0.0.0.0              Interface:GE0/5/0
 BFDState        :Up                rx      :500                tx          :500
 Multiplier      :3                 BFD Local Dis:16385         LocalIpAdd:10.0.0.2
 RemoteIpAdd     :10.0.0.1          Diagnostic Info:No diagnostic information

  Area 0.0.0.0 interface 10.0.0.9 (GE0/5/1)'s BFD Sessions

 NeighborId      :10.0.6.6          AreaId:0.0.0.0              Interface:GE0/5/1
 BFDState        :Up                rx      :500                tx          :500
 Multiplier      :3                 BFD Local Dis:16386         LocalIpAdd:10.0.0.9
 RemoteIpAdd     :10.0.0.10         Diagnostic Info:No diagnostic information

    Total UP/DOWN/UNKNOWN BFD Session Number : 2 / 0 / 0
```

The OSPF BFD sessions of all interfaces on P1 and PE2 are normal.

# Configure a traffic policy to filter BFD packets.

```
[PE2]acl number 3000
[PE2-acl4-advance-3000] rule 1 permit udp destination-port eq 3784
[PE2-acl4-advance-3000] quit
[PE2]traffic classifier bfd operator or
[PE2-classifier-bfd] if-match acl 3000
[PE2-classifier-bfd] quit
[PE2]traffic behavior bfd
[PE2-behavior-bfd] deny
[PE2-behavior-bfd] quit
[PE2]traffic policy bfd
[PE2-trafficpolicy-bfd] classifier bfd behavior bfd
[PE2-trafficpolicy-bfd] quit
```

Create ACL 3000 to match BFD packets (UDP port 3784 is the destination port of BFD control packets for single-hop detection.), and reference this ACL in the traffic policy.

# Apply the traffic policy to the inbound direction of GE0/5/0 on PE2.

```
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0]traffic-policy bfd inbound
```

# Check the BFD session of GE0/5/0 on PE2.

```
[PE2]display ospf bfd session GigabitEthernet 0/5/0

            OSPF Process 1 with Router ID 10.0.2.2

  Area 0.0.0.0 interface 10.0.0.2 (GE0/5/0)'s BFD Sessions

 NeighborId      :10.0.1.1          AreaId:0.0.0.0              Interface:GE0/5/0
```

```
 BFDState        :Down              rx      :-                    tx          :-
 Multiplier      :-                   BFD Local Dis:16385       LocalIpAdd:10.0.0.2
 RemoteIpAdd     :10.0.0.1          Diagnostic Info:Control   Detection Time Expired

     Total UP/DOWN/UNKNOWN BFD Session Number : 0 / 1 / 0
```

The BFD session of GE0/5/0 on PE2 is Down.

# Check the status of GE0/5/0 on PE2.

```
[PE2]display interface brief | include 0/5/0
PHY: Physical
*down: administratively down
^down: standby
(l): loopback
(s): spoofing
(E): E-Trunk down
(b): BFD down
(B): Bit-error-detection down
(e): ETHOAM down
(d): Dampening Suppressed
(p): port alarm down
(ld): loop-detect trigger down
(mf): mac-flapping blocked
(c): CFM down
(sd): STP instance discarding
InUti/OutUti: input utility/output utility
Interface                   PHY    Protocol  InUti OutUti    inErrors   outErrors
GigabitEthernet0/5/0        up     up         0.01% 0.01%    0          0
```

The physical status of the interface is still Up.

# Check the OSPF neighbor status on PE2.

```
[PE2]display ospf peer brief
(M) Indicates MADJ neighbor


            OSPF Process 1 with Router ID 10.0.2.2
                    Peer Statistic Information
Total number of peer(s): 2
 Peer(s) in full state: 2
--------------------------------------------------------------------------------
 Area Id          Interface                     Neighbor id          State
 0.0.0.0          GE0/5/0                       10.0.1.1             Full
 0.0.0.0          GE0/5/1                       10.0.6.6             Full
--------------------------------------------------------------------------------
```

The OSPF neighbor relationship between PE2 and PE1 is normal and not Down.

Configure BFD in the OSPF view and interface view. When the BFD session goes Down, OSPF is instructed to perform convergence again (Hello packets are sent immediately and the dead interval is changed to a small value), instead of directly setting the OSPF neighbor state to Down.

In this lab, only BFD packets on an interface are filtered. The OSPF packets received by the interface are not filtered. Therefore, OSPF does not detect the disconnection of the neighbor relationship.

# Cancel the traffic policy on GE0/5/0 of PE2.

```
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0]undo traffic-policy bfd inbound
```

# Add rule 2 to ACL 3000 to match OSPF packets.

```
[PE2]acl 3000
[PE2-acl4-advance-3000]rule 2 permit ospf
```

# Apply the traffic policy on GE0/5/0 of PE2 again.

```
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0]traffic-policy bfd inbound
```

PE2 cannot receive BFD or OSPF packets. After the BFD session goes Down, re-establishment of the OSPF neighbor relationship is triggered. However, GE0/5/0 on PE2 can only send OSPF packets but cannot receive OSPF packets. Therefore, the OSPF neighbor relationship fails to be established.

# Check OSPF neighbors on PE2.

```
[PE2]display ospf peer brief

(M) Indicates MADJ    neighbor


             OSPF Process 1 with Router ID 10.0.2.2
                       Peer Statistic Information
Total number of peer(s): 2
 Peer(s) in full state: 1
--------------------------------------------------------------------------
 Area Id          Interface                     Neighbor id          State
 0.0.0.0          GE0/5/0                       10.0.1.1             ExStart
 0.0.0.0          GE0/5/1                       10.0.6.6             Full
--------------------------------------------------------------------------
```

The neighbor relationship between PE2 and PE1 is **Exstart**, instead of **Full**. The interval is far from the dead interval of OSPF, but re-establishment of the OSPF neighbor relationship is triggered.

# Check the OSPF routing table on PE2.

```
[PE2]display ospf routing
        OSPF Process 1 with Router ID 10.0.2.2
                Routing Tables

Routing for Network
Destination      Cost   Type       NextHop         AdvRouter       Area
10.0.0.0/30      1      Direct     10.0.0.2        10.0.2.2        0.0.0.0
```

```
10.0.0.4/30     4    Transit    10.0.0.10       10.0.5.5        0.0.0.0
10.0.0.8/30     2    Direct     10.0.0.9        10.0.2.2        0.0.0.0
10.0.0.12/30    3    Transit    10.0.0.10       10.0.5.5        0.0.0.0
10.0.0.16/30    4    Inter-area 10.0.0.10       10.0.5.5        0.0.0.0
10.0.1.1/32     4    Stub       10.0.0.10       10.0.1.1        0.0.0.0
10.0.2.2/32     0    Direct     10.0.2.2        10.0.2.2        0.0.0.0
10.0.3.3/32     4    Inter-area 10.0.0.10       10.0.5.5        0.0.0.0
10.0.5.5/32     3    Stub       10.0.0.10       10.0.5.5        0.0.0.0
10.0.6.6/32     2    Stub       10.0.0.10       10.0.6.6        0.0.0.0
```

The routing table shows that the next hop address of the OSPF route to P1 is 10.0.0.10 (IP address of GE0/5/1 on P2). This means the OSPF route to P1 has been switched to the backup path.

This step shows how BFD accelerates OSPF convergence. After completing this step, cancel the traffic policy on GE0/5/0 of PE2.

```
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0]undo traffic-policy bfd inbound
```

# Check OSPF neighbors on PE2.

```
[PE2]display ospf peer brief

(M) Indicates MADJ    neighbor


             OSPF Process 1 with Router ID 10.0.2.2
                       Peer Statistic Information
Total number of peer(s): 2
 Peer(s) in full state: 2
--------------------------------------------------------------------------------
 Area Id          Interface                     Neighbor id          State
 0.0.0.0          GE0/5/0                       10.0.1.1             Full
 0.0.0.0          GE0/5/1                       10.0.6.6             Full
--------------------------------------------------------------------------------
```

Ensure that the neighbor relationship between PE2 and PE1 has recovered.

## Step 6 Configure OSPF route limitation.

Create Loopback2 interfaces on PE1 and P2, and set their IP addresses to 10.2.16.16/32. Enable OSPF on Loopback2 interfaces. Limit the number of equal-cost routes on P1 to 1.

Create a Loopback3 interface with the IP address of 10.3.3.3/24 on PE3, change the OSPF interface type to broadcast, enable OSPF on Loopback3 interface, and filter inter-area Type-3 LSAs on ABR P1 to prevent OSPF inter-area route to 10.3.3.0/24 from being advertised to area 0.

# Create Loopback2 interfaces on PE1 and P2, and enable OSPF.

```
[PE1]interface LoopBack2
[PE1-LoopBack2] ip address 10.2.16.16 32
[PE1-LoopBack2] ospf enable 1 area 0
[PE1-LoopBack2] quit
```

```
[P2]interface LoopBack2
[P2-LoopBack2] ip address 10.2.16.16 32[P2-LoopBack2] ospf enable 1 area 0
[P2-LoopBack2] quit
```

# Check the OSPF routing table on P1.

```
[P1]display ospf routing

        OSPF Process 1 withRouter ID 10.0.5.5
              Routing Tables

Routing forNetwork
Destination      Cost   Type      NextHop      AdvRouter      Area
10.0.0.0/30      2      Transit   10.0.0.5     10.0.2.2       0.0.0.0
10.0.0.4/30      1      Direct    10.0.0.6     10.0.5.5       0.0.0.0
10.0.0.8/30      2      Transit   10.0.0.14    10.0.6.6       0.0.0.0
10.0.0.12/30     1      Direct    10.0.0.13    10.0.5.5       0.0.0.0
10.0.0.16/30     1      Direct    10.0.0.17    10.0.5.5       0.0.0.1
10.0.1.1/32      1      Stub      10.0.0.5     10.0.1.1       0.0.0.0
10.0.2.2/32      2      Stub      10.0.0.5     10.0.2.2       0.0.0.0
10.0.2.2/32      2      Stub      10.0.0.14    10.0.2.2       0.0.0.0
10.0.3.3/32      1      Stub      10.0.0.18    10.0.3.3       0.0.0.1
10.0.5.5/32      0      Direct    10.0.5.5     10.0.5.5       0.0.0.0
10.0.6.6/32      1      Stub      10.0.0.14    10.0.6.6       0.0.0.0
10.2.16.16/32    1      Stub      10.0.0.5     10.0.1.1       0.0.0.0
10.2.16.16/32    1      Stub      10.0.0.14    10.0.6.6       0.0.0.0

TotalNets:11
Intra Area: 11  Inter Area: 0  ASE:0  NSSA: 0
```

P1 has two OSPF routes to 10.2.16.16/32 with the next hops being PE1 and P2 respectively. The routes work in load balancing mode.

# Limit the number of equal-cost routes to 1 on P1.

```
[P1]ospf 1
[P1-ospf-1] maximum load-balancing 1
```

# Check the OSPF routing table on P1 again.

```
[P1]display ospf routing

        OSPF Process 1 with Router ID 10.0.5.5
              Routing Tables

 Routing for Network
 Destination      Cost   Type      NextHop      AdvRouter      Area
 10.0.0.0/30      2      Transit   10.0.0.5     10.0.2.2       0.0.0.0
 10.0.0.4/30      1      Direct    10.0.0.6     10.0.5.5       0.0.0.0
 10.0.0.8/30      2      Transit   10.0.0.14    10.0.6.6       0.0.0.0
 10.0.0.12/30     1      Direct    10.0.0.13    10.0.5.5       0.0.0.0
 10.0.0.16/30     1      Direct    10.0.0.17    10.0.5.5       0.0.0.1
```

```
10.0.1.1/32        1        Stub      10.0.0.5        10.0.1.1        0.0.0.0
10.0.2.2/32        2        Stub      10.0.0.5        10.0.2.2        0.0.0.0
10.0.2.2/32        2        Stub      10.0.0.14       10.0.2.2        0.0.0.0
10.0.3.3/32        1        Stub      10.0.0.18       10.0.3.3        0.0.0.1
10.0.5.5/32        0        Direct    10.0.5.5        10.0.5.5        0.0.0.0
10.0.6.6/32        1        Stub      10.0.0.14       10.0.6.6        0.0.0.0
10.2.16.16/32      1        Stub      10.0.0.5        10.0.1.1        0.0.0.0

Total Nets: 11
Intra Area: 11    Inter Area: 0    ASE: 0    NSSA: 0
```

P1 has only one OSPF route to 10.2.16.16/32, with the next hop being PE1.

If the number of equal-cost routes exceeds the limit specified in the **maximum load-balancing** command, valid routes are selected for load balancing based on the following criteria:

1. Route priority: Routes with the highest priority (lowest weight) are selected for load balancing.
2. Interface index: If routes have the same priority, the routes with the largest interface index are selected for load balancing.
3. Next-hop IP address: If routes have the same priority and interface index, the routes with the largest next-hop IP addresses are selected for load balancing.

The index of the interface (GE0/5/1) connecting P1 to PE1 is greater than that of the interface (GE0/5/0) connecting P1 to P2. Therefore, the OSPF routing to 10.2.16.16/32 sent from PE1 is selected as a valid routing.

# Create a Loopback3 interface on PE3 and enable OSPF.

```
[PE3]interface LoopBack3
[PE3-LoopBack3] ip address 10.3.3.3 255.255.255.0
[PE3-LoopBack3] ospf network-type broadcast
[PE3-LoopBack3] ospf enable 1 area 0.0.0.1
```

# Check the OSPF routing table on PE1.

```
<PE1>display ospf routing
          OSPF Process 1 with Router ID 10.0.1.1
                Routing Tables

Routing for Network
Destination       Cost    Type       NextHop      AdvRouter      Area
10.0.0.0/30       1       Direct     10.0.0.1     10.0.1.1       0.0.0.0
10.0.0.4/30       1       Direct     10.0.0.5     10.0.1.1       0.0.0.0
10.0.0.8/30       3       Transit    10.0.0.6     10.0.6.6       0.0.0.0
10.0.0.8/30       3       Transit    10.0.0.2     10.0.6.6       0.0.0.0
10.0.0.12/30      2       Transit    10.0.0.6     10.0.5.5       0.0.0.0
10.0.0.16/30      2       Inter-area 10.0.0.6     10.0.5.5       0.0.0.0
10.0.1.1/32       0       Direct     10.0.1.1     10.0.1.1       0.0.0.0
10.0.2.2/32       1       Stub       10.0.0.2     10.0.2.2       0.0.0.0
10.0.3.3/32       2       Inter-area 10.0.0.6     10.0.5.5       0.0.0.0
10.0.5.5/32       1       Stub       10.0.0.6     10.0.5.5       0.0.0.0
10.0.6.6/32       2       Stub       10.0.0.6     10.0.6.6       0.0.0.0
10.2.16.16/32     0       Direct     10.2.16.16   10.0.1.1       0.0.0.0
```

```
10.3.3.0/24        2       Inter-area 10.0.0.6       10.0.5.5        0.0.0.0

TotalNets:12
Intra Area: 9  Inter Area: 3  ASE:0  NSSA: 0
```

PE1 has learned the route to the network segment to which the IP address of PE3's Loopback3 belongs.

# Check the OSPF LSDB on P1.

```
[P1]display ospf lsdb
         OSPF Process 1 with Router ID 10.0.5.5
               Link State Database

                    Area: 0.0.0.0
Type      LinkState ID    AdvRouter       Age  Len  Sequence    Metric
Router    10.0.1.1        10.0.1.1        653  72   80000065    1
Router    10.0.2.2        10.0.2.2        1352 60   80000066    2
Router    10.0.5.5        10.0.5.5        1712 60   8000005b    1
Router    10.0.6.6        10.0.6.6        677  72   8000005b    1
Network   10.0.0.2        10.0.2.2        1352 32   80000001    0
Network   10.0.0.6        10.0.5.5        1712 32   80000051    0
Network   10.0.0.10       10.0.6.6        1703 32   80000051    0
Network   10.0.0.13       10.0.5.5        749  32   80000056    0
Sum-Net   10.0.0.16       10.0.5.5        850  28   80000056    1
Sum-Net   10.0.3.3        10.0.5.5        791  28   80000056    1
Sum-Net   10.3.3.0        10.0.5.5        191  28   80000001    1

                    Area: 0.0.0.1
Type      LinkState ID    AdvRouter       Age  Len  Sequence    Metric
Router    10.0.3.3        10.0.3.3        192  60   80000058    1
Router    10.0.5.5        10.0.5.5        801  36   80000057    1
Network   10.0.0.17       10.0.5.5        801  32   80000056    0
Sum-Net   10.0.0.0        10.0.5.5        1345 28   80000054    2
Sum-Net   10.0.0.4        10.0.5.5        1752 28   80000051    1
Sum-Net   10.0.0.8        10.0.5.5        1744 28   80000051    2
Sum-Net   10.0.0.12       10.0.5.5        850  28   80000056    1
Sum-Net   10.0.1.1        10.0.5.5        1703 28   80000052    1
Sum-Net   10.0.2.2        10.0.5.5        1703 28   80000051    2
Sum-Net   10.0.5.5        10.0.5.5        850  28   80000056    0
Sum-Net   10.0.6.6        10.0.5.5        749  28   80000056    1
Sum-Net   10.2.16.16      10.0.5.5        676  28   80000001    1
```

The LSDB of area 0 on ABR P1 contains a Type 3 LSA with the link state ID of 10.3.3.0.

# Configure Type 3 LSA filtering on ABR P1 to prevent the OSPF inter-area route to 10.3.3.0/24 from being transmitted to area 0.

```
[P1]ip ip-prefix 1 index 10 deny 10.3.3.0 24 greater-equal 24 less-equal 24
[P1]ip ip-prefix 1 index 20 permit 0.0.0.0 0 less-equal 32
[P1]ospf 1
[P1-ospf-1] area 1
[P1-ospf-1-area-0.0.0.1] filter ip-prefix 1 export
```

# Check the OSPF LSDB on P1 again.

```
[P1]display ospf lsdb
        OSPF Process 1 with Router ID 10.0.5.5
             Link State Database

                  Area: 0.0.0.0
Type      LinkState ID    AdvRouter       Age  Len  Sequence    Metric
Router    10.0.1.1        10.0.1.1        653  72   80000065    1
Router    10.0.2.2        10.0.2.2        1352 60   80000066    2
Router    10.0.5.5        10.0.5.5        1712 60   8000005b    1
Router    10.0.6.6        10.0.6.6        677  72   8000005b    1
Network   10.0.0.2        10.0.2.2        1352 32   80000001    0
Network   10.0.0.6        10.0.5.5        1712 32   80000051    0
Network   10.0.0.10       10.0.6.6        1703 32   80000051    0
Network   10.0.0.13       10.0.5.5        749  32   80000056    0
Sum-Net   10.0.0.16       10.0.5.5        850  28   80000056    1
Sum-Net   10.0.3.3        10.0.5.5        791  28   80000056    1

                  Area: 0.0.0.1
Type      LinkState ID    AdvRouter       Age  Len  Sequence    Metric
Router    10.0.3.3        10.0.3.3        192  60   80000058    1
Router    10.0.5.5        10.0.5.5        801  36   80000057    1
Network   10.0.0.17       10.0.5.5        801  32   80000056    0
Sum-Net   10.0.0.0        10.0.5.5        1345 28   80000054    2
Sum-Net   10.0.0.4        10.0.5.5        1752 28   80000051    1
Sum-Net   10.0.0.8        10.0.5.5        1744 28   80000051    2
Sum-Net   10.0.0.12       10.0.5.5        850  28   80000056    1
Sum-Net   10.0.1.1        10.0.5.5        1703 28   80000052    1
Sum-Net   10.0.2.2        10.0.5.5        1703 28   80000051    2
Sum-Net   10.0.5.5        10.0.5.5        850  28   80000056    0
Sum-Net   10.0.6.6        10.0.5.5        749  28   80000056    1
Sum-Net   10.2.16.16      10.0.5.5        676  28   80000001    1
```

The LSDB of area 0 on P1 does not contain the Type 3 LSA with the link state ID of 10.3.3.0.

# Check the OSPF route to 10.3.3.0/24 on PE1.

```
[PE1]display ospf routing 10.3.3.0
[PE1]
```

The OSPF route to 10.3.3.0/24 does not exist.

Step 7 Configure a default IS-IS route.

Advertise a default route on PE3 (an IS-IS Level-1-2 device) and check default routes on PE4 (an IS-IS Level-1 device).

# Advertise a default route on PE3.

```
[PE3]isis 1
[PE3-isis-1]default-route-advertise always level-1-2
```

# Check IS-IS routes on PE4.

```
[PE4]display isis route
```

```
                         Route information for ISIS(1)
                         -----------------------------

                        ISIS(1) Level-1 Forwarding Table
                        --------------------------------

IPV4 Destination     IntCost      ExtCost ExitInterface      NextHop            Flags
-------------------------------------------------------------------------------
0.0.0.0/0            10           NULL    GE0/5/0            10.0.0.25          A/-/-/-
10.0.0.24/30         10           NULL    GE0/5/0            Direct             D/-/L/-
10.0.4.4/32          0            NULL    Loop0              Direct             D/-/L/-
      Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
             U-Up/Down Bit Set, LP-Local Prefix-Sid
      Protect Type: L-Link Protect, N-Node Protect
```

PE4 has learned the default route advertised by PE3.

# Check the IS-IS LSDB on PE4.

```
<PE4>display isis lsdb verbose
                        Database information for ISIS(1)
                        --------------------------------


                          Level-1 Link State Database

LSPID                 Seq Num      Checksum     HoldTime         Length     ATT/P/OL
-------------------------------------------------------------------------------
0000.0000.0003.00-00  0x000000b4 0x662e       1088             82         0/0/0
  SOURCE          0000.0000.0003.00
  NLPID           IPV4
  AREA ADDR       49.0001
  INTF ADDR       10.0.0.25
  NBR  ID         0000.0000.0004.01   COST: 10
  IP-Internal     10.0.0.24           255.255.255.252  COST: 10
  IP-Internal     0.0.0.0             0.0.0.0          COST: 0

0000.0000.0004.00-00* 0x000000b4 0x0c60       359              86         0/0/0
  SOURCE          0000.0000.0004.00
  NLPID           IPV4
  AREA ADDR       49.0001
  INTF ADDR       10.0.0.26
  INTF ADDR       10.0.4.4
  NBR  ID         0000.0000.0004.01   COST: 10
  IP-Internal     10.0.0.24           255.255.255.252  COST: 10
  IP-Internal     10.0.4.4            255.255.255.255  COST: 0

0000.0000.0004.01-00* 0x000000af 0x8f48       357              55         0/0/0
  SOURCE          0000.0000.0004.01
  NLPID           IPV4
  NBR  ID         0000.0000.0004.00   COST: 0
  NBR  ID         0000.0000.0003.00   COST: 0

Total LSP(s): 3
    *(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),
```

```
                ATT-Attached, P-Partition, OL-Overload
```

The LSP 0000.0000.0003.00-00 from PE3 carries the default route.

**----End**

## 1.1.3 Quiz

OSPF inter-area routes can be filtered by running the **filter ip-prefix ip-prefix-name export** command on an ABR but intra-area routes cannot be filtered by running the command in an OSPF area. Why?

# 1.2 Advanced BGP Features

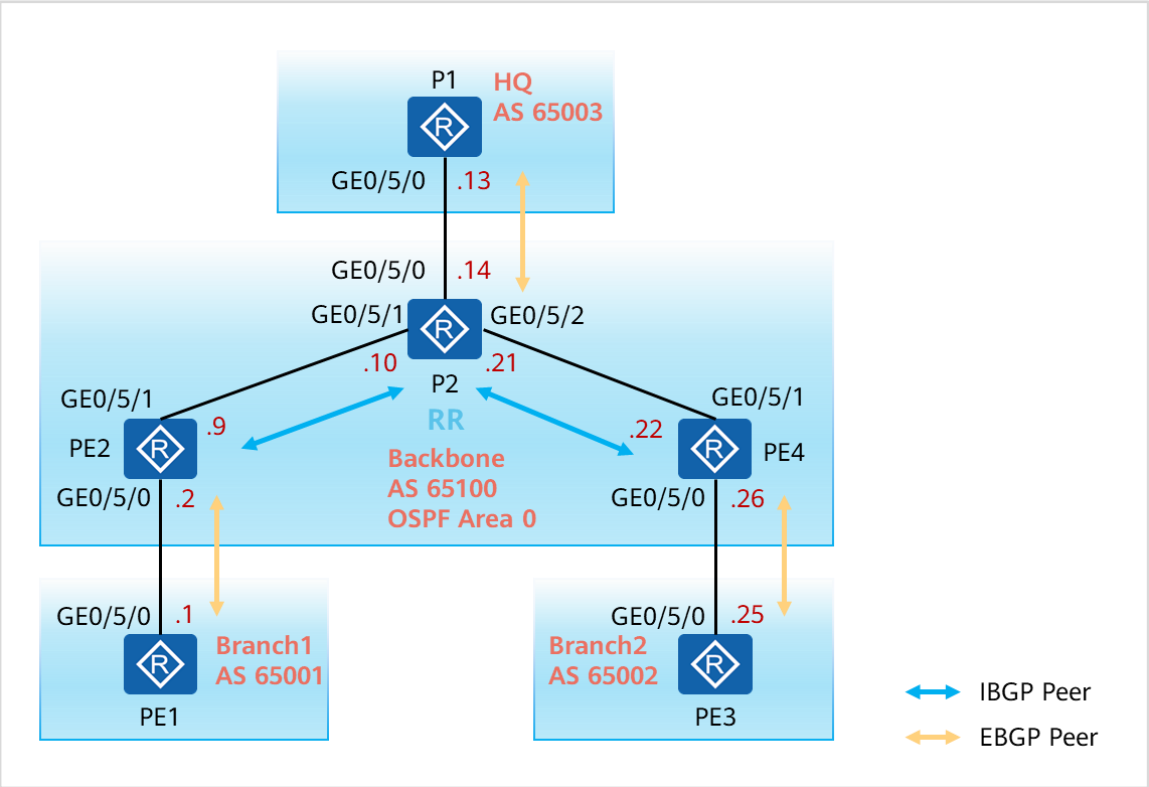
## 1.2.1 About This Lab

### 1.2.1.1 Objectives

- Configure Border Gateway Protocol (BGP) routing policies.
- Configure BGP security features.
- Configure BGP outbound route filtering (ORF) features.

### 1.2.1.2 Networking Description

**Figure 1-2 Lab topology of advanced BGP features**




The figure shows the device connections, IP address planning, BGP AS numbers, and BGP peer relationships. The interface interconnection addresses are in the format of 10.0.0.*Y*/30, and the values represented by Y are shown in the figure. Loopback0 is created on all devices, and loopback interface IP addresses are in the format of 10.0.*X.X*. The specific IP addresses are listed in a table in the section "Configuration Procedure".

In this lab, P2, PE2, and PE4 are used to simulate a backbone network, P1 is used to simulate the enterprise headquarters, and PE1 and PE3 are used to simulate enterprise branches. Different loopback interfaces are created on PE1, PE3, and P1 to simulate users. The Community Filter, AS_Path Filter, and ORF features are used to control transmission of the service routes.

# 1.2.2 Lab Task

## 1.2.2.1 Configuration Roadmap

1. Configure IP addresses for the devices.
2. Configure OSPF in the backbone area to construct an underlying network.
3. Configure BGP between the backbone network and the enterprise network, and configure Generalized TTL Security Mechanism (GTSM) and BGP authentication to ensure BGP network security.
4. Configure IBGP peer relationships on the backbone network. To simplify the configuration, use a peer group to establish peer relationships. Configure P2 as a route reflector (RR), and configure PE2 and PE4 as RR clients.
5. On P1, PE1, and PE3, create Loopback1 and Loopback2 to simulate service network segments of the OA and finance departments, respectively. Configure a routing policy to assign a community value to the route destined for Loopback1 route, so as to mark the originating AS of the route.
6. Configure a routing policy on P2, PE2, and PE4 to control the transmission of routes destined for OA and financial network segments.
7. Configure ORF on P1 and P2 to limit the number of routes received by P1.

## 1.2.2.2 Configuration Procedure

Step 1 Complete basic device configuration.

Configure the configuration validation mode as immediate validation, and configure IP addresses for interconnection and Loopback0 interfaces. Loopback0 addresses must be configured as planned in the following table.

**Table 1-2 Loopback0 IP addresses**

| Device | Value of *X* | Loopback0 IP Address |
|---|---|---|
| PE1 | 1 | 10.0.1.1 |
| PE2 | 2 | 10.0.2.2 |
| PE3 | 3 | 10.0.3.3 |
| PE4 | 4 | 10.0.4.4 |
| P1 | 5 | 10.0.5.5 |
| P2 | 6 | 10.0.6.6 |

# Name the devices.

N/A

# Disable unnecessary interfaces in this lab.

N/A

# Set the configuration validation mode to immediate validation.

```
<PE1>system-view immediately
```

```
<PE2>system-view immediately
```

```
<PE3>system-view immediately
```

```
<PE4>system-view immediately
```

```
<P1>system-view immediately
```

```
<P2>system-view immediately
```

# Disable DCN globally on each device.

```
[PE1] undo dcn
Warning: This operation will disable DCN function. Continue? [Y/N]:y
```

Here, PE1 is used as an example. Repeat this operation on other devices.

By default, DCN is enabled on NE router interfaces. To facilitate experiments in the lab, disable DCN globally on all devices.

# Configure IP addresses for the interconnection interface and Loopback0 on PE1.

```
[PE1]interface LoopBack0
[PE1-LoopBack0] ip address 10.0.1.1 32
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0]ip address 10.0.0.1 30
[PE1-GigabitEthernet0/5/0] quit
```

# Configure IP addresses for the interconnection interfaces and Loopback0 interface on PE2.

```
[PE2]interface LoopBack0
[PE2-LoopBack0] ip address 10.0.2.2 32
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ip address 10.0.0.2 30
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
```

```
[PE2-GigabitEthernet0/5/1] ip address 10.0.0.9 30
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection interface and Loopback0 interface on PE3.

```
[PE3]interface LoopBack0
[PE3-LoopBack0] ip address 10.0.3.3 32
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] ip address 10.0.0.25 30
[PE3-GigabitEthernet0/5/0] quit
```

# Configure IP addresses for the interconnection interfaces and Loopback0 interface on PE4.

```
[PE4]interface LoopBack0
[PE4-LoopBack0] ip address 10.0.4.4 32
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] ip address 10.0.0.26 30
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] ip address 10.0.0.22 255.255.255.252
```

# Configure IP addresses for the interconnection interface and Loopback0 interface on P1.

```
[P1]interface LoopBack0
[P1-LoopBack0] ip address 10.0.5.5 32
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] ip address 10.0.0.13 30
[P1-GigabitEthernet0/5/0] quit
```

# Configure IP addresses for the interconnection interfaces and Loopback0 interface on P2.

```
[P2]interface LoopBack0
[P2-LoopBack0] ip address 10.0.6.6 32
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ip address 10.0.0.14 30
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ip address 10.0.0.10 30
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] ip address 10.0.0.21 255.255.255.252
[P2-GigabitEthernet0/5/2] quit
```

# Test connectivity between interconnection interfaces on PE1, P2, and PE3.

```
[P2]ping -c 1 10.0.0.22
  PING 10.0.0.22: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.22: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.22 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

[P2]ping -c 1 10.0.0.9
  PING 10.0.0.9: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.9: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.9 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

[P2]ping -c 1 10.0.0.13
  PING 10.0.0.13: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.13: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.13 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

```
[PE1]ping -c 1 10.0.0.2
  PING 10.0.0.2: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.2: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.2 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
[PE3]ping -c 1 10.0.0.26
  PING 10.0.0.26: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.26: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.26 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

Step 2 Configure OSPF in the backbone area.

Enable OSPF on P2, PE2, and PE4 in the backbone area. Set the process ID to 1. Use the IP address of Loopback0 as the router ID. Enable OSPF on the interconnection and Loopback0 interfaces of devices in the backbone area.

# Configure OSPF on PE2, and enable OSPF on the interconnection and Loopback0 interfaces.

```
[PE2]ospf 1 router-id 10.0.2.2
[PE2-ospf-1] area 0
[PE2-ospf-1-area-0.0.0.0] quit
[PE2-ospf-1] quit
[PE2]interface LoopBack0
[PE2-LoopBack0] ospf enable 1 area 0
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ospf enable 1 area 0
[PE2-GigabitEthernet0/5/1] quit
```

# Configure OSPF on P2, and enable OSPF on the interconnection and Loopback0 interfaces.

```
[P2]ospf 1 router-id 10.0.6.6
[P2-ospf-1] area 0
[P2-ospf-1-area-0.0.0.0] quit
[P2-ospf-1] quit
[P2]interface LoopBack0
[P2-LoopBack0] ospf enable 1 area 0
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ospf enable 1 area 0
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] ospf enable 1 area 0
[P2-GigabitEthernet0/5/2] quit
```

# Configure OSPF on PE4, and enable OSPF on the interconnection and Loopback0 interfaces.

```
[PE4]ospf 1 router-id 10.0.4.4
[PE4-ospf-1]area 0
[PE4-ospf-1-area-0.0.0.0] quit
[PE4-ospf-1]quit
[PE4]interface Loopback0
[PE4-LoopBack0] ospf enable 1 are 0
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] ospf enable 1 area 0
[PE4-GigabitEthernet0/5/1] quit
```

# Check OSPF neighbor relationships on P2.

```
[P2]display ospf peer brief

(M) Indicates MADJ neighbor
```

```
          OSPF Process 1 with Router ID 10.0.6.6
                    Peer Statistic Information
Total number of peer(s): 2
 Peer(s) in full state: 2
--------------------------------------------------------------------------------
 Area Id          Interface                     Neighbor id          State
 0.0.0.0          GE0/5/1                       10.0.2.2             Full
 0.0.0.0          GE0/5/2                       10.0.4.4             Full
--------------------------------------------------------------------------------
```

OSPF neighbor relationships are successfully established.

# Check OSPF routes on P2.

```
[P2]display ospf routing
        OSPF Process 1 with Router ID 10.0.6.6
              Routing Tables

Routing for Network
Destination       Cost    Type      NextHop       AdvRouter       Area
10.0.0.8/30       1       Direct    10.0.0.10     10.0.6.6        0.0.0.0
10.0.0.20/30      1       Direct    10.0.0.21     10.0.6.6        0.0.0.0
10.0.2.2/32       1       Stub      10.0.0.9      10.0.2.2        0.0.0.0
10.0.4.4/32       1       Stub      10.0.0.22     10.0.4.4        0.0.0.0
10.0.6.6/32       0       Direct    10.0.6.6      10.0.6.6        0.0.0.0

Total Nets:5
Intra Area: 5  Inter Area: 0  ASE:0  NSSA: 0
```

P2 has learned the routes to the Loopback0 interfaces of PE2 and PE4.

Step 3 Deploy BGP between the backbone network and the enterprise egress.

Configure BGP between the backbone network and the enterprise egress. Use the IP address of the Loopback0 interface as the source IP address and router ID for establishing BGP peer relationships. Configure static routes to ensure the Loopback0 interfaces are reachable to each other.

Configure GTSM and BGP authentication to ensure BGP network security. Set the authentication password to **Huawei@123** and GTSM to 255.

# Configure static routes on P1 and P2.

```
[P1]ip route-static 10.0.6.6 32 10.0.0.14
```

```
[P2]ip route-static 10.0.5.5 32 10.0.0.13
```

# Configure static routes on PE1 and PE2.

```
[PE1]ip route-static 10.0.2.2 32 10.0.0.2
```

```
[PE2]ip route-static 10.0.1.1 32 10.0.0.1
```

# Configure static routes on PE3 and PE4.

```
[PE3]ip route-static 10.0.4.4 32 10.0.0.26
```

```
[PE4]ip route-static 10.0.3.3 32 10.0.0.25
```

# Establish an EBGP peer relationship between P1 and P2

```
[P1]bgp 65003
[P1-bgp] router-id 10.0.5.5
[P1-bgp] peer 10.0.6.6 as-number 65100
[P1-bgp] peer 10.0.6.6 connect-interface LoopBack0
[P1-bgp] peer 10.0.6.6 password cipher Huawei@123
[P1-bgp] peer 10.0.6.6 valid-ttl-hops 255
```

```
[P2] bgp 65100
[P2-bgp] router-id 10.0.6.6
[P2-bgp] peer 10.0.5.5 as-number 65003
[P2-bgp] peer 10.0.5.5 connect-interface LoopBack0
[P2-bgp] peer 10.0.5.5 password cipher Huawei@123
[P2-bgp] peer 10.0.5.5 valid-ttl-hops 255
```

# Establish an EBGP peer relationship between PE1 and PE2.

```
[PE1]bgp 65001
[PE1-bgp] router-id 10.0.1.1
[PE1-bgp] peer 10.0.2.2 as-number 65100
[PE1-bgp] peer 10.0.2.2 connect-interface LoopBack0
[PE1-bgp] peer 10.0.2.2 password cipher Huawei@123
[PE1-bgp] peer 10.0.2.2 valid-ttl-hops 255
```

```
[PE2] bgp 65100
[PE2-bgp] router-id 10.0.2.2
[PE2-bgp] peer 10.0.1.1 as-number 65001
[PE2-bgp] peer 10.0.1.1 connect-interface LoopBack0
[PE2-bgp] peer 10.0.1.1 password cipher Huawei@123
[PE2-bgp] peer 10.0.1.1 valid-ttl-hops 255
```

# Establish an EBGP peer relationship between PE3 and PE4.

```
[PE3]bgp 65002
[PE3-bgp] router-id 10.0.3.3
```

```
[PE3-bgp] peer 10.0.4.4 as-number 65100
[PE3-bgp] peer 10.0.4.4 connect-interface LoopBack0
[PE3-bgp] peer 10.0.4.4 password cipher Huawei@123
[PE3-bgp] peer 10.0.4.4 valid-ttl-hops 255
```

```
[PE4] bgp 65100
[PE4-bgp] router-id 10.0.4.4
[PE4-bgp] peer 10.0.3.3 as-number 65002
[PE4-bgp] peer 10.0.3.3 connect-interface LoopBack0
[PE4-bgp] peer 10.0.3.3 password cipher Huawei@123
[PE4-bgp] peer 10.0.3.3 valid-ttl-hops 255
```

# Check the EBGP peer relationships on the devices in the backbone area.

```
[PE2]display bgp peer

BGP local router ID : 10.0.2.2
Local AS number : 65100
Total number of peers : 1                 Peers in established state : 1

 Peer            V     AS      MsgRcvd  MsgSent  OutQ   Up/Down     State        PrefRcv
 10.0.1.1        4     65001   3        3        0      00:00:10    Established  0
```

```
[PE4]display bgp peer

BGP local router ID : 10.0.4.4
Local AS number : 65100
Total number of peers : 1                 Peers in established state : 1

 Peer            V     AS      MsgRcvd  MsgSent  OutQ   Up/Down     State        PrefRcv
 10.0.3.3        4     65002   3        3        0      00:00:10    Established  0
```

```
[P2]display bgp peer

BGP local router ID : 10.0.6.6
Local AS number : 65100
Total number of peers : 1                 Peers in established state : 1

 Peer            V     AS      MsgRcvd  MsgSent  OutQ   Up/Down     State        PrefRcv
 10.0.5.5        4     65003   3        3        0      00:00:10    Established  0
```

The BGP peer relationships between PE2, PE4, and P2 and their peers are in **Established** state.

Step 4 Deploy BGP in the backbone area.

Configure IBGP peer relationships in the backbone area. Use the IP address of Loopback0 as the source address and router ID for establishing BGP peer relationships. Configure P2 as an RR, and configure PE2 and PE4 as RR clients. To simplify the configurations, use a peer group to configure BGP peer relationships on P2.

# Configure BGP on P2.

```
[P2]bgp 65100
[P2-bgp] group BB internal
[P2-bgp] peer BB connect-interface LoopBack 0
[P2-bgp] peer BB next-hop-local
[P2-bgp] peer 10.0.2.2 group BB
[P2-bgp] peer 10.0.4.4 group BB
[P2-bgp] ipv4-family unicast
[P2-bgp-af-ipv4] peer BB reflect-client
```

# Configure BGP on PE2.

```
[PE2]bgp 65100
[PE2-bgp] peer 10.0.6.6 as-number 65100
[PE2-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE2-bgp] peer 10.0.6.6 next-hop-local
```

# Configure BGP on PE4.

```
[PE4]bgp 65100
[PE4-bgp] peer 10.0.6.6 as-number 65100
[PE4-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE4-bgp] peer 10.0.6.6 next-hop-local
```

# Check the IBGP peer relationship on P2.

```
[P2]display bgp peer

BGP local router ID : 10.0.6.6
Local AS number : 65100
Total number of peers : 1                 Peers in established state : 1

 Peer            V    AS      MsgRcvd  MsgSent  OutQ   Up/Down      State       PrefRcv
10.0.2.2         4    65100   3        3        0      00:00:52     Established 0
10.0.4.4         4    65100   3        3        0      00:00:44     Established 0
10.0.5.5         4    65003   3        3        0      00:05:08     Established 0
```

An IBGP peer relationship has been successfully established.

Step 5 Advertise BGP routes.

Create Loopback1 and Loopback2 on the enterprise routers. IP addresses of these loopback interfaces are in the format of 10.*Y*.*X*.*X*/32, where *X* is listed in Table 1-2 and *Y* is the interface number. For example, the IP address of Loopback1 on P1 is 10.1.5.5.

Advertise the routes to Loopback1 and Loopback2 to BGP and apply a community value to route destined for Loopback1 by a routing policy.

# Create Loopback1 and Loopback2 interfaces.

```
[P1]interface Loopback1
[P1-LoopBack1] ip address 10.1.5.5 32
[P1-LoopBack1] quit
[P1]interface Loopback2
[P1-LoopBack2] ip address 10.2.5.5 32
[P1-LoopBack2] quit
[PE1]interface Loopback1
[PE1-LoopBack1] ip address 10.1.1.1 32
[PE1-LoopBack1] quit
[PE1]interface Loopback2
[PE1-LoopBack2] ip address 10.2.1.1 32
[PE1-LoopBack2] quit
```

```
[PE3]interface Loopback1
[PE3-LoopBack1] ip address 10.1.3.3 32
[PE3-LoopBack1] quit
[PE3]interface Loopback2
[PE3-LoopBack2] ip address 10.2.3.3 32
[PE3-LoopBack2] quit
```

# Advertise service network segments using the **network** command on enterprise routers.

```
[P1]bgp 65003
[P1-bgp]network 10.1.5.5 32
[P1-bgp]network 10.2.5.5 32
```

```
[PE1]bgp 65001
[PE1-bgp]network 10.1.1.1 32
[PE1-bgp]network 10.2.1.1 32
```

```
[PE3]bgp 65002
[PE3-bgp]network 10.1.3.3 32
[PE3-bgp]network 10.2.3.3 32
```

# Check BGP routes on P1.

```
[P1]display bgp routing-table

 BGP Local router ID is 10.0.5.5
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,   i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found


 Total Number of Routes: 6
```

```
      Network          NextHop        MED        LocPrf     PrefVal Path/Ogn
 *>   10.1.1.1/32      10.0.6.6                             0       65100 65001i
 *>   10.1.3.3/32      10.0.6.6                             0       65100 65002i
 *>   10.1.5.5/32      0.0.0.0          0                   0       i
 *>   10.2.1.1/32      10.0.6.6                             0       65100 65001i
 *>   10.2.3.3/32      10.0.6.6                             0       65100 65002i
 *>   10.2.5.5/32      0.0.0.0          0                   0       i
```

P1 has learned the routes to service network segments of PE1 and PE3.

# Check BGP routes on PE1.

```
[PE1]display bgp routing-table

 BGP Local router ID is 10.0.1.1
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,   i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found


 Total Number of Routes: 6
      Network          NextHop        MED    LocPrf PrefVal Path/Ogn
 *>   10.1.1.1/32      0.0.0.0          0             0       i
 *>   10.1.3.3/32      10.0.2.2                       0       65100 65002i
 *>   10.1.5.5/32      10.0.2.2                       0       65100 65003i
 *>   10.2.1.1/32      0.0.0.0          0             0       i
 *>   10.2.3.3/32      10.0.2.2                       0       65100 65002i
 *>   10.2.5.5/32      10.0.2.2                       0       65100 65003i
```

PE1 has learned the routes to service network segments of P1 and PE3.

# Check BGP routes on PE3.

```
[PE3]display bgp routing-table

 BGP Local router ID is 10.0.3.3
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,   i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found


 Total Number of Routes: 6
      Network          NextHop        MED        LocPrf     PrefVal Path/Ogn
 *>   10.1.1.1/32      10.0.4.4                             0       65100 65001i
 *>   10.1.3.3/32      0.0.0.0          0                   0       i
 *>   10.1.5.5/32      10.0.4.4                             0       65100 65003i
 *>   10.2.1.1/32      10.0.4.4                             0       65100 65001i
 *>   10.2.3.3/32      0.0.0.0          0                   0       i
 *>   10.2.5.5/32      10.0.4.4                             0       65100 65003i
```

PE3 has learned the routes to service network segments of P1 and PE2.

Assign community values to the routes destined for Loopback1 interfaces on P1, PE1, and PE3 by using routing policies. Enable devices to advertise the community attribute to their peers. By default, this function is disabled.

The community attribute is generally in the format of AS:*NN*, which is defined as AS:01 in this lab.

# Enable the capability of advertising the community attribute to peers on all routers.

```
[P1]bgp 65003
[P1-bgp] peer 10.0.6.6 advertise-community
```

```
[P2]bgp 65100
[P2-bgp] peer 10.0.5.5 advertise-community
[P2-bgp] peer BB advertise-community
```

```
[PE1]bgp 65001
[PE1-bgp] peer 10.0.2.2 advertise-community
```

```
[PE2]bgp 65100
[PE2-bgp] peer 10.0.6.6 advertise-community
[PE2-bgp] peer 10.0.1.1 advertise-community
```

```
[PE3]bgp 65002
[PE3-bgp] peer 10.0.4.4 advertise-community
```

```
[PE4]bgp 65100
[PE4-bgp] peer 10.0.6.6 advertise-community
[PE4-bgp] peer 10.0.3.3 advertise-community
```

# Configure routing policies on P1, PE1, and PE3 to apply community values to the routes destined for Loopback1 interfaces.

```
[P1]ip ip-prefix Com index 10 permit 10.1.5.5 32
[P1]route-policy Attr permit node 10
[P1-route-policy] if-match ip-prefix Com
[P1-route-policy] apply community 65003:1
[P1-route-policy] quit
[P1]route-policy Attr permit node 100
[P1-route-policy] quit
[P1]bgp 65003
[P1-bgp] peer 10.0.6.6 route-policy Attr export
```

```
[PE1]ip ip-prefix Com index 10 permit 10.1.1.1 32
[PE1]route-policy Attr permit node 10
[PE1-route-policy] if-match ip-prefix Com
[PE1-route-policy] apply community 65001:1
[PE1-route-policy] quit
[PE1]route-policy Attr permit node 100
[PE1-route-policy] quit
[PE1]bgp 65001
[PE1-bgp] peer 10.0.2.2 route-policy Attr export
```

```
[PE3]ip ip-prefix Com index 10 permit 10.1.3.3 32
[PE3]route-policy Attr permit node 10
[PE3-route-policy] if-match ip-prefix Com
[PE3-route-policy] apply community 65002:1
[PE3-route-policy] quit
[PE3]route-policy Attr permit node 100
[PE3-route-policy] quit
[PE3]bgp 65002
[PE3-bgp] peer 10.0.4.4 route-policy Attr export
```

# Check the configuration.

```
[P1]display bgp routing-table 10.1.1.1

 BGP local router ID : 10.0.5.5
 Local AS number : 65003
 Paths:    1 available, 1 best, 1 select, 0 best-external, 0 add-path
 BGP routing table entry information of 10.1.1.1/32:
 From: 10.0.6.6 (10.0.6.6)
 Route Duration: 0d00h02m21s
 Direct Out-interface: GigabitEthernet0/5/0
 Original nexthop: 10.0.6.6
 Qos information : 0x0
 Community: <65001:1>
 AS-path 65100 65001, origin igp, pref-val 0, valid, external, best, select, pre 255
 Not advertised to any peer yet
[P1]display bgp routing-table 10.1.3.3

 BGP local router ID : 10.0.5.5
 Local AS number : 65003
 Paths:    1 available, 1 best, 1 select, 0 best-external, 0 add-path
 BGP routing table entry information of 10.1.3.3/32:
 From: 10.0.6.6 (10.0.6.6)
 Route Duration: 0d00h01m41s
 Direct Out-interface: GigabitEthernet0/5/0
 Original nexthop: 10.0.6.6
 Qos information : 0x0
 Community: <65002:1>
 AS-path 65100 65002, origin igp, pref-val 0, valid, external, best, select, pre 255
 Not advertised to any peer yet
```

P1 is used as an example. The check methods on PE1 and PE3 are similar.

Step 6 Configure routing policies to control service route transmission.

The service route learning requirements are as follows: Only the headquarters can learn the routes to Loopback2 interfaces of the branches, and the branches cannot learn the routes to Loopback2 interfaces from each other.

To simplify the filtering configuration, you can use **AS-Path Filter** and **Route-Policy** to restrict the routes advertised by PE2 and PE4 to their peers.

When controlling routes, do not filter out the routes to Loopback0 interfaces. You can use **Community Filter** to allow the routes destined for Loopback1 interfaces to be advertised and then filter out the routes destined for Loopback2 interfaces.

# Configure PE2.

```
[PE2]ip community-filter basic OA index 10 permit 65002:1
[PE2]ip community-filter basic OA index 20 permit 65003:1
[PE2]ip as-path-filter Finance permit 65002$
[PE2]route-policy Finance permit node 10
[PE2-route-policy] if-match community-filter OA
[PE2-route-policy] quit
[PE2]route-policy Finance deny node 20
[PE2-route-policy] if-match as-path-filter Finance
[PE2-route-policy] quit
[PE2]route-policy Finance permit node 30
[PE2-route-policy] quit
[PE2]bgp 65100
[PE2-bgp] peer 10.0.1.1 route-policy Finance export
```

# Configure PE4.

```
[PE4]ip community-filter basic OA index 10 permit 65001:1
[PE4]ip community-filter basic OA index 20 permit 65003:1
[PE4]ip as-path-filter Finance permit 65001$
[PE4]route-policy Finance permit node 10
[PE4-route-policy] if-match community-filter OA
[PE4-route-policy] quit
[PE4]route-policy Finance deny node 20
[PE4-route-policy] if-match as-path-filter Finance
[PE4-route-policy] quit
[PE4]route-policy Finance permit node 30
[PE4-route-policy] quit
[PE4]bgp 65100
[PE4-bgp] peer 10.0.3.3 route-policy Finance export
```

# Check the BGP routing tables on PE1 and PE3.

```
[PE1]display bgp routing-table

 BGP Local router ID is 10.0.1.1
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,   i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
```

```
RPKI validation codes: V - valid, I - invalid, N - not-found


Total Number of Routes: 5
         Network            NextHop      MED    LocPrf     PrefVal Path/Ogn
*>       10.1.1.1/32        0.0.0.0        0                  0       i
*>       10.1.3.3/32        10.0.2.2                          0       65100 65002i
*>       10.1.5.5/32        10.0.2.2                          0       65100 65003i
*>       10.2.1.1/32        0.0.0.0        0                  0       i
*>       10.2.5.5/32        10.0.2.2                          0       65100 65003i
```

```
[PE3]display bgp routing-table

 BGP Local router ID is 10.0.3.3
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,    i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found


 Total Number of Routes: 5
         Network            NextHop      MED    LocPrf     PrefVal Path/Ogn
*>       10.1.1.1/32        10.0.4.4                          0       65100 65001i
*>       10.1.3.3/32        0.0.0.0        0                  0       i
*>       10.1.5.5/32        10.0.4.4                          0       65100 65003i
*>       10.2.3.3/32        0.0.0.0        0                  0       i
*>       10.2.5.5/32        10.0.4.4                          0       65100 65003i
```

The command output shows that PE1 and PE3 do not learn the routes to Loopback1 interfaces of their peers

Step 7 Configure ORF.

If branches incorrectly advertise routes, the headquarters learn unnecessary routes. To prevent this, configure ORF on P2 and P1 so that P1 learns only the planned routes that should be advertised by the branches.

# Create Loopback3 on PE3, set its IP address to 10.3.3.3/32, and advertise the route to this IP address to BGP.

```
[PE3]interface LoopBack 3
[PE3-LoopBack3] ip address 10.3.3.3 32
[PE3-LoopBack3] quit
```

```
[PE3]bgp 65002
[PE3-bgp] network 10.3.3.3 32
```

# Check whether P1 has learned the route advertised by PE3.

```
[P1]display bgp routing-table | include 10.3.3.3
```

```
Info: It will take a long time if the content you search is too much or the string you input is too long,
you can press CTRL_C to break.

 BGP Local router ID is 10.0.5.5
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,   i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found


 Total Number of Routes: 7
      Network            NextHop        MED        LocPrf     PrefVal Path/Ogn

 *>   10.3.3.3/32        10.0.6.6                                 0       65100 65002i
```

P1 has learned the route.

# Configure an IP prefix list named **Branch** on P1 to match the planned branch route.

```
[P1]ip ip-prefix Branch index 10 permit 10.1.1.1 32
[P1]ip ip-prefix Branch index 20 permit 10.2.1.1 32
[P1]ip ip-prefix Branch index 30 permit 10.1.3.3 32
[P1]ip ip-prefix Branch index 40 permit 10.2.3.3 32
```

Do not match the route to Loopback3 route of PE3.

# Configure ORF on P1 and P2.

```
[P1]bgp 65003
[P1-bgp] peer 10.0.6.6 ip-prefix Branch import
[P1-bgp] peer 10.0.6.6 capability-advertise orf ip-prefix send
```

```
[P2]bgp 65100
[P2-bgp] peer 10.0.5.5 capability-advertise orf ip-prefix receive
```

# Check the ORF information on P2.

```
[P2]display bgp peer 10.0.5.5 orf ip-prefix
 Total number of ip-prefix received: 4
 Index   Action   Prefix            MaskLen   MinLen   MaxLen
 10      Permit   10.1.1.1          32
 20      Permit   10.2.1.1          32
 30      Permit   10.1.3.3          32
 40      Permit   10.2.3.3          32
```

The command output shows that P2 has imported the route prefixes permitted by P1.

# Check the BGP route to 10.3.3.3 on P1.

```
[P1]display bgp routing-table | include 10.3.3.3
Info: It will take a long time if the content you search is too much or the string you input is too long,
you can press CTRL_C to break.
```

```
BGP Local router ID is 10.0.5.5
Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
              h - history,   i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V - valid, I - invalid, N - not-found


Total Number of Routes: 6
     Network            NextHop        MED        LocPrf     PrefVal Path/Ogn
```

The route does not exist in the BGP routing table on P1.

**----End**

## 1.2.3 Quiz

In addition to ORF, which method can also be used to filter received routes?

# 2 IPv6 Routing

## 2.1 IPv6 Routing

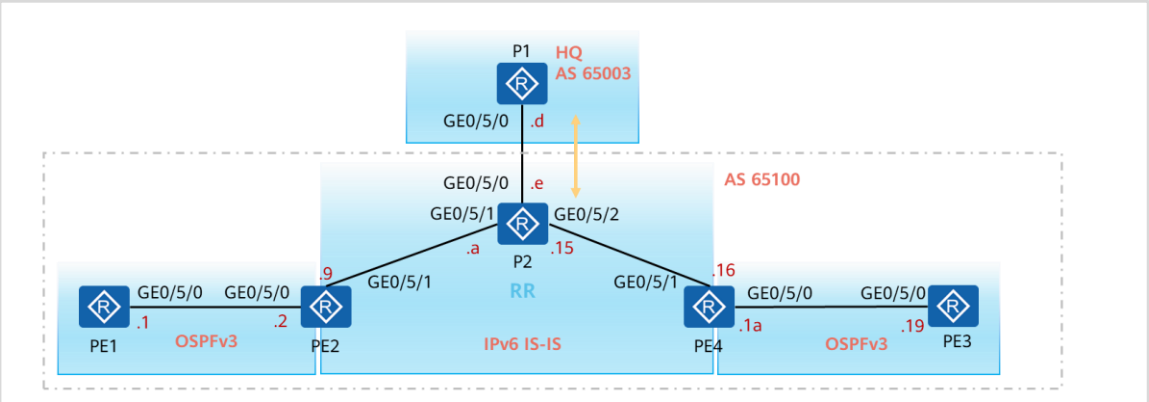
### 2.1.1 About This Lab

#### 2.1.1.1 Objectives

- Configure IPv6 Intermediate System to Intermediate System (IS-IS).
- Configure OSPFv3.
- Configure Multiprotocol Extensions for BGP (MP-BGP).

#### 2.1.1.2 Networking Description

**Figure 2-1 IPv6 routing topology**




The figure shows the device connection and IP address planning. The interface interconnection addresses are in the format of 2001::*Y*/126, and the values represented by Y are shown in the figure. Loopback0 is created on all devices, and loopback interface IP addresses are in the format of 2001::*X:X*/128. The specific IP addresses are listed in a table in the section "Configuration Procedure".

In AS 65100, OSPFv3 and IS-IS are used to construct an underlying network. MP-IBGP runs among PE1, P2, and PE3 to transmit BGP routes. MP-EBGP runs between P1 and P2, and bidirectional route redistribution between IS-IS and BGP is performed on P2.

### 2.1.2 Lab Task

#### 2.1.2.1 Configuration Roadmap

1. Configure IPv6 addresses for devices.
2. Deploy OSPFv3.

3. Deploy IPv6 IS-IS.
4. Configure bidirectional route import between IPv6 IS-IS and OSPFv3.
5. Establish IBGP peer relationships among PE1, P2, and PE3, and configure P2 as an RR.
6. Establish an EBGP peer relationship between P1 and P2 using the IP addresses of Loopback0 interfaces as the source addresses. Create Loopback1 on P1 and advertise the route destined for this interface to BGP.
7. Perform bidirectional route redistribution between IS-IS and BGP on P2.

## 2.1.2.2 Configuration Procedure

Step 1 Complete basic device configuration.

Configure the configuration validation mode as immediate validation, and configure IP addresses for interconnection and Loopback0 interfaces. Loopback0 addresses must be configured as planned in the following table.

**Table 2-1 Loopback0 IP addresses**

| Device | Value of X | Loopback0 IP Address |
|---|---|---|
| PE1 | 1 | 2001::1:1 |
| PE2 | 2 | 2001::2:2 |
| PE3 | 3 | 2001::3:3 |
| PE4 | 4 | 2001::4:4 |
| P1 | 5 | 2001::5:5 |
| P2 | 6 | 2001::6:6 |

# Name the devices.

N/A

# Disable unnecessary interfaces in this lab.

N/A

# Set the configuration validation mode to immediate validation.

```
<PE1>system-view immediately
```

```
<PE2>system-view immediately
```

```
<PE3>system-view immediately
```

```
<PE4>system-view immediately
```

```
<P2>system-view immediately
```

# Disable DCN globally on each device.

```
[PE1] undo dcn
Warning: This operation will disable DCN function. Continue? [Y/N]:y
```

Here, PE1 is used as an example. Repeat this operation on other devices.

By default, DCN is enabled on NE router interfaces. To facilitate experiments in the lab, disable DCN globally on all devices.

# Configure IPv6 addresses for the interconnection interface and Loopback0 interface on PE1.

```
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0]ipv6 enable
[PE1-GigabitEthernet0/5/0]ipv6 address 2001::1 126
[PE1]interface LoopBack 0
[PE1-LoopBack0] ipv6 enable
[PE1-LoopBack0] ipv6 address 2001::1:1 128
```

# Configure IPv6 addresses for the interconnection interfaces and Loopback0 interface on PE2.

```
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ipv6 enable
[PE2-GigabitEthernet0/5/0] ipv6 address 2001::2/126
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ipv6 enable
[PE2-GigabitEthernet0/5/1] ipv6 address 2001::9/126
[PE2-GigabitEthernet0/5/1] quit
[PE2]interface LoopBack0
[PE2-LoopBack0] ipv6 enable
[PE2-LoopBack0] ipv6 address 2001::2:2/128
[PE2-LoopBack0] quit
```

# Configure IPv6 addresses for the interconnection interface and Loopback0 interface on P1.

```
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] ipv6 enable
[P1-GigabitEthernet0/5/0] ipv6 address 2001::d 126
[P1-GigabitEthernet0/5/0] quit
[P1] interface LoopBack 0
```

```
[P1-LoopBack0] ipv6 enable
[P1-LoopBack0] ipv6 address 2001::5:5 128
[P1-LoopBack0] quit
```

# Configure IPv6 addresses for the interconnection interfaces and Loopback0 interface on P2.

```
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ipv6 enable
[P2-GigabitEthernet0/5/0] ipv6 address 2001::e/126
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ipv6 enable
[P2-GigabitEthernet0/5/1] ipv6 address 2001::a/126
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] ipv6 enable
[P2-GigabitEthernet0/5/2] ipv6 address 2001::15/126
[P2-GigabitEthernet0/5/2] quit
[P2]interface LoopBack0
[P2-LoopBack0] ipv6 enable
[P2-LoopBack0] ipv6 address 2001::6:6/128
[P2-LoopBack0] quit
```

# Configure IPv6 addresses for the interconnection interfaces and Loopback0 interface on PE4.

```
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] ipv6 enable
[PE4-GigabitEthernet0/5/0] ipv6 address 2001::1a/126
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] ipv6 enable
[PE4-GigabitEthernet0/5/1] ipv6 address 2001::16/126
[PE4-GigabitEthernet0/5/1] quit
[PE4]interface LoopBack0
[PE4-LoopBack0] ipv6 enable
[PE4-LoopBack0] ipv6 address 2001::4:4/128
[PE4-LoopBack0] quit
```

# Configure IPv6 addresses for the interconnection interface and Loopback0 interface on PE3.

```
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] ipv6 enable
[PE3-GigabitEthernet0/5/0] ipv6 address 2001::19/126
[PE3-GigabitEthernet0/5/0] quit
[PE3]interface LoopBack0
[PE3-LoopBack0] ipv6 enable
[PE3-LoopBack0] ipv6 address 2001::3:3/128
[PE3-LoopBack0] quit
```

# Test IPv6 connectivity between interconnection interfaces on PE1, P2, and PE3.

```
[P2]ping ipv6 -c 1 2001::9
  PING 2001::9 : 56  data bytes, press CTRL_C to break
    Reply from 2001::9
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::9 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=1/1/1 ms

[P2]ping ipv6 -c 1 2001::16
  PING 2001::16 : 56  data bytes, press CTRL_C to break
    Reply from 2001::16
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::16 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
round-trip min/avg/max=1/1/1 ms

[P2]ping ipv6 -c 1 2001::d
  PING 2001::D : 56  data bytes, press CTRL_C to break
    Reply from 2001::D
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::D ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=1/1/1 ms
```

```
[PE1]ping ipv6 -c 1 2001::2
  PING 2001::2 : 56  data bytes, press CTRL_C to break
    Reply from 2001::2
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::2 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=1/1/1 ms
```

```
[PE3]ping ipv6 -c 1 2001::1a
  PING 2001::1A : 56  data bytes, press CTRL_C to break
    Reply from 2001::1A
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::1A ping statistics---
    1 packet(s) transmitted
```

```
  1 packet(s) received
  0.00% packet loss
  round-trip min/avg/max=1/1/1 ms
```

Step 2 Deploy OSPFv3.

Deploy OSPFv3 between PE1 and PE2, and between PE3 and PE4. Set the OSPF process ID to 1, the area ID to 0, and the instance ID to 1. The router IDs are in the format of 10.0.*X.X*. For the value of *X*, see Table 2-1.

Enable OSPFv3 on all interconnection and Loopback0 interfaces.

# Configure OSPFv3 on PE1.

```
[PE1] ospfv3 1
[PE1-ospfv3-1] router-id 10.0.1.1
[PE1-ospfv3-1] quit
[PE1] interface LoopBack 0
[PE1-LoopBack0] ospfv3 1 area 0 instance 1
[PE1-LoopBack0] quit
[PE1] interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0] ospfv3 1 area 0 instance 1
```

# Configure OSPFv3 on PE2.

```
[PE2] ospfv3 1
[PE2-ospfv3-1] router-id 10.0.2.2
[PE2-ospfv3-1] quit
[PE2] interface LoopBack 0
[PE2-LoopBack0] ospfv3 1 area 0 instance 1
[PE2-LoopBack0] quit
[PE2] interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ospfv3 1 area 0 instance 1
```

# Configure OSPFv3 on PE3.

```
[PE3] ospfv3 1
[PE3-ospfv3-1] router-id 10.0.3.3
[PE3-ospfv3-1] quit
[PE3] interface LoopBack 0
[PE3-LoopBack0] ospfv3 1 area 0 instance 1
[PE3-LoopBack0] quit
[PE3] interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] ospfv3 1 area 0 instance 1
```

# Configure OSPFv3 on PE4.

```
[PE4] ospfv3 1
[PE4-ospfv3-1] router-id 10.0.4.4
[PE4-ospfv3-1] quit
[PE4] interface LoopBack 0
[PE4-LoopBack0] ospfv3 1 area 0 instance 1
[PE4-LoopBack0] quit
[PE4] interface GigabitEthernet0/5/0
```

```
[PE4-GigabitEthernet0/5/0] ospfv3 1 area 0 instance 1
```

# Check OSPFv3 neighbor relationships.

```
[PE2]display ospfv3 peer

OSPFv3 Process (1)
OSPFv3 Area (0.0.0.0)
Neighbor ID        Pri State            Dead Time   Interface          Instance ID
10.0.1.1             1 Full/DR          00:00:31    GE0/5/0                 1
```

```
[PE4]display ospfv3 peer

OSPFv3 Process (1)
OSPFv3 Area (0.0.0.0)
Neighbor ID        Pri State            Dead Time   Interface          Instance ID
10.0.3.3             1 Full/DR          00:00:37    GE0/5/0                 1
```

OSPFv3 neighbor relationships have been established.

# Check OSPFv3 routing tables.

```
[PE2]display ospfv3 routing

Codes : E2 - Type 2 External, E1 - Type 1 External, IA - Inter-Area,
        N - NSSA
Flags : A - Added to URT6, LT - Locator Routing

OSPFv3 Process (1)
      Destination                                                 Metric
        Next-hop
      2001::/126                                                     1
        directly connected, GE0/5/0, Flags : A
      2001::1:1/128                                                  1
        via FE80::F29B:B8FF:FECC:7452, GE0/5/0, Flags : A
      2001::2:2/128                                                  0
        directly connected, Loop0, Flags : A
```

```
[PE4]display ospfv3 routing

Codes : E2 - Type 2 External, E1 - Type 1 External, IA - Inter-Area,
        N - NSSA
Flags : A - Added to URT6, LT - Locator Routing

OSPFv3 Process (1)
      Destination                                                 Metric
        Next-hop
      2001::18/126                                                   1
        directly connected, GE0/5/0, Flags : A
      2001::3:3/128                                                  1
```

```
    via FE80::F29B:B8FF:FECC:70C2, GE0/5/0, Flags : A
2001::4:4/128                                                0
    directly connected, Loop0, Flags : A
```

OSPFv3 routes have been learned.

Step 3 Configure IS-IS.

Configure IS-IS among PE2, P2, and PE4 and set the process ID to 1. All IS-IS routers are Level-2 routers. The cost type is set to **wide** mode. The area ID of the NET address is set to 49.0001. The system ID is assigned as follows: 49.0001.000*X*.000*X*.000*X*, where *X* indicates the device ID. For details, see Table 2-1. The IS-IS host name must be the same as the device name.

Enable IS-IS on the interconnection interfaces between PE2 and PE4, and enable IS-IS on the interconnection and Loopback0 interfaces on P2.

# Configure IS-IS on PE2.

```
[PE2]isis 1
[PE2-isis-1] is-level level-2
[PE2-isis-1] cost-style wide
[PE2-isis-1] network-entity 49.0001.0002.0002.0002.00
[PE2-isis-1] is-name PE2
[PE2-isis-1] ipv6 enable topology ipv6
[PE2-isis-1] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] isis ipv6 enable 1
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on P2.

```
[P2]isis 1
[P2-isis-1] is-level level-2
[P2-isis-1] cost-style wide
[P2-isis-1] network-entity 49.0001.0006.0006.0006.00
[P2-isis-1] is-name P2
[P2-isis-1] ipv6 enable topology ipv6
[P2-isis-1] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] isis ipv6 enable 1
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] isis ipv6 enable 1
[P2-GigabitEthernet0/5/2] quit
[P2]interface LoopBack 0
[P2-LoopBack0] isis ipv6 enable 1
[P2-LoopBack0] quit
```

# Configure IS-IS on PE4.

```
[PE4]isis 1
[PE4-isis-1] is-level level-2
[PE4-isis-1] cost-style wide
[PE4-isis-1] network-entity 49.0001.0004.0004.0004.00
```

```
[PE4-isis-1] is-name PE4
[PE4-isis-1] ipv6 enable topology ipv6
[PE4-isis-1] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] isis ipv6 enable 1
[PE4-GigabitEthernet0/5/1] quit
```

# Check the IS-IS neighbor status on P2.

```
[P2]display isis peer

                        Peer information for ISIS(1)

  System Id     Interface          Circuit Id         State HoldTime   Type     PRI
------------------------------------------------------------------------------
PE2*            GE0/5/1            PE2.01             Up    7s         L2       64
PE4*            GE0/5/2            PE4.01             Up    9s         L2       64

Total Peer(s): 2
```

IS-IS neighbor relationships have been successfully established.

# Check IS-IS routes on PE2 and PE4.

```
[PE2]display isis route

                  Routeinformation forISIS(1)
                  ----------------------------

                 ISIS(1) Level-2Forwarding Table
                 --------------------------------

IPV6 Dest.    ExitInterface       NextHop                    Cost    Flags
------------------------------------------------------------------------------
2001::8/126   GE0/5/1             Direct                      10      D/-/L/-
2001::14/126  GE0/5/1             FE80::F29B:B8FF:FECC:740B  20      A/-/-/-
2001::6:6/128 GE0/5/1             FE80::F29B:B8FF:FECC:740B  10      A/-/-/-
   Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
          U-Up/Down Bit Set, LP-Local Prefix-Sid
   Protect Type: L-Link Protect, N-Node Protect
```

```
[PE4]display isis route

                  Routeinformation forISIS(1)
                  ----------------------------

                 ISIS(1) Level-2Forwarding Table
                 --------------------------------

IPV6 Dest.    ExitInterface       NextHop                    Cost    Flags
------------------------------------------------------------------------------
2001::8/126   GE0/5/1             FE80::F29B:B8FF:FECC:740C  20      A/-/-/-
2001::14/126  GE0/5/1             Direct                      10      D/-/L/-
```

```
2001::6:6/128   GE0/5/1            FE80::F29B:B8FF:FECC:740C   10       A/-/-/-
    Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
           U-Up/Down Bit Set, LP-Local Prefix-Sid
    Protect Type: L-Link Protect, N-Node Protect
```

PE2 and PE4 have learned the route to Loopback0 of P2.

Step 4 Configure bidirectional route import between IS-IS and OSPFv3.

Configure bidirectional route import between OSPFv3 and IS-IS on PE2 and PE4 to implement underlying network interworking for MP-BGP.

# Configure PE2.

```
[PE2]ospfv3 1
[PE2-ospfv3-1] import-route isis 1
[PE2-ospfv3-1] quit
[PE2]isis 1
[PE2-isis-1] ipv6 import-route ospfv3 1
[PE2-isis-1] quit
```

```
[PE4]ospfv3 1
[PE4-ospfv3-1] import-route isis 1
[PE4-ospfv3-1] quit
[PE4]isis 1
[PE4-isis-1] ipv6 import-route ospfv3 1
[PE4-isis-1] quit
```

# Check the OSPFv3 routing tables on PE1 and PE3.

```
[PE1]display ospfv3 routing

Codes : E2 - Type 2 External, E1 - Type 1 External, IA - Inter-Area,
          N - NSSA
Flags : A - Added to URT6, LT - Locator Routing

OSPFv3 Process (1)
        Destination                                          Metric
          Next-hop
        2001::/126                                              1
          directly connected, GE0/5/0, Flags : A
E2      2001::8/126                                             1
          via FE80::F29B:B8FF:FECC:743A, GE0/5/0, Flags : A
E2      2001::14/126                                            1
          via FE80::F29B:B8FF:FECC:743A, GE0/5/0, Flags : A
E2      2001::18/126                                            1
          via FE80::F29B:B8FF:FECC:743A, GE0/5/0, Flags : A
        2001::1:1/128                                           0
          directly connected, Loop0, Flags : A
        2001::2:2/128                                           1
          via FE80::F29B:B8FF:FECC:743A, GE0/5/0, Flags : A
```

```
E2    2001::3:3/128                                                  1
        via FE80::F29B:B8FF:FECC:743A, GE0/5/0, Flags : A
E2    2001::4:4/128                                                  1
        via FE80::F29B:B8FF:FECC:743A, GE0/5/0, Flags : A
E2    2001::6:6/128                                                  1
        via FE80::F29B:B8FF:FECC:743A, GE0/5/0, Flags : A
```

```
[PE3]display ospfv3 routing

Codes : E2 - Type 2 External, E1 - Type 1 External, IA - Inter-Area,
          N - NSSA
Flags : A - Added to URT6, LT - Locator Routing

OSPFv3 Process (1)
       Destination                                                Metric
         Next-hop
E2    2001::/126                                                    1
        via FE80::F29B:B8FF:FECC:77B2, GE0/5/0, Flags : A
E2    2001::8/126                                                   1
        via FE80::F29B:B8FF:FECC:77B2, GE0/5/0, Flags : A
E2    2001::14/126                                                  1
        via FE80::F29B:B8FF:FECC:77B2, GE0/5/0, Flags : A
      2001::18/126                                                  1
        directly connected, GE0/5/0, Flags : A
E2    2001::1:1/128                                                 1
        via FE80::F29B:B8FF:FECC:77B2, GE0/5/0, Flags : A
E2    2001::2:2/128                                                 1
        via FE80::F29B:B8FF:FECC:77B2, GE0/5/0, Flags : A
      2001::3:3/128                                                 0
        directly connected, Loop0, Flags : A
      2001::4:4/128                                                 1
        via FE80::F29B:B8FF:FECC:77B2, GE0/5/0, Flags : A
E2    2001::6:6/128                                                 1
        via FE80::F29B:B8FF:FECC:77B2, GE0/5/0, Flags : A
```

PE1 and PE3 have learned the route to Loopback0 of P2.

# Check the IS-IS routing table on P2.

```
[P2]display isis route

                  Routeinformation forISIS(1)
                  ---------------------------

                 ISIS(1) Level-2Forwarding Table
                 -------------------------------

IPV6 Dest.     ExitInterface        NextHop                 Cost    Flags
--------------------------------------------------------------------------------
2001::/126     GE0/5/1              FE80::F29B:B8FF:FECC:743B 10     A/-/-/-
2001::8/126    GE0/5/1              Direct                   10     D/-/L/-
2001::14/126   GE0/5/2              Direct                   10     D/-/L/-
2001::18/126   GE0/5/2              FE80::F29B:B8FF:FECC:77B3 10     A/-/-/-
```

```
2001::1:1/128   GE0/5/1          FE80::F29B:B8FF:FECC:743B  10      A/-/-/-
2001::2:2/128   GE0/5/1          FE80::F29B:B8FF:FECC:743B  10      A/-/-/-
2001::3:3/128   GE0/5/2          FE80::F29B:B8FF:FECC:77B3  10      A/-/-/-
2001::4:4/128   GE0/5/2          FE80::F29B:B8FF:FECC:77B3  10      A/-/-/-
2001::6:6/128   Loop0            Direct                     0       D/-/L/-
    Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
           U-Up/Down Bit Set, LP-Local Prefix-Sid
    Protect Type: L-Link Protect, N-Node Protect
```

P2 has learned the routes to Loopback0 interfaces of PE1 and PE3.

# Use the IPv6 address of Loopback0 on P2 as the source address to test the connectivity between the Loopback0 interfaces of PE1 and PE3.

```
[P2]ping ipv6 -c 1 -a 2001::6:6 2001::1:1
  PING 2001::1:1 : 56  data bytes, press CTRL_C to break
    Reply from 2001::1:1
    bytes=56 Sequence=1 hop limit=63 time=1 ms

  --- 2001::1:1 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=1/1/1 ms

[P2]ping ipv6 -c 1 -a 2001::6:6 2001::3:3
  PING 2001::3:3 : 56  data bytes, press CTRL_C to break
    Reply from 2001::3:3
    bytes=56 Sequence=1 hop limit=63 time=1 ms

  --- 2001::3:3 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=1/1/1 ms
```

The communication is normal.

Step 5 Configure MP-BGP.

Configure PE1, P2, and PE3 to establish IBGP peer relationships by using the IP addresses of their Loopback0 interfaces as the source addresses, and deploy P2 as an RR.

Configure P1 and P2 to establish an EBGP peer relationship using the IP addresses of their Loopback0 interfaces as the source addresses. Create a Loopback1 interface on P1, and advertise the route destined for this interface to BGP. Import IS-IS routes to BGP on P2.

All BGP peers use 10.0.*X.X* as the router ID. For the value of *X*, see Table 2-1.

# Configure static routes on P1 and P2.

```
[P1]ipv6 route-static 2001::6:6 128 2001::e
```

```
[P2]ipv6 route-static 2001::5:5 128 2001::d
```

To ensure that Loopback0 interfaces between P1 and P2 can communicate with each other, configure IPv6 static routes.

# On P1, test connectivity between Loopback0 interfaces of P1 and P2.

```
[P1]ping ipv6 -c 1 -a 2001::5:5 2001::6:6
  PING 2001::6:6 : 56  data bytes, press CTRL_C to break
    Reply from 2001::6:6
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::6:6 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=1/1/1 ms
```

# Establish an MP-EBGP peer relationship between P1 and P2.

```
[P1]bgp 65003
[P1-bgp] router-id 10.0.5.5
[P1-bgp] peer 2001::6:6 as-number 65100
[P1-bgp] peer 2001::6:6 ebgp-max-hop 255
[P1-bgp] peer 2001::6:6 connect-interface LoopBack0
[P1-bgp] ipv6-family unicast
[P1-bgp-af-ipv6]peer 2001::6:6 enable
```

```
[P2]bgp 65100
[P2-bgp] router-id 10.0.6.6
[P2-bgp] peer 2001::5:5 as-number 65003
[P2-bgp] peer 2001::5:5 ebgp-max-hop 255
[P2-bgp] peer 2001::5:5 connect-interface LoopBack0
[P2-bgp] ipv6-family unicast
[P2-bgp-af-ipv6] peer 2001::5:5 enable
```

# Check the MP-EBGP peer relationship on P1.

```
[P1]display bgp ipv6 peer

BGP local router ID : 10.0.5.5
Local AS number : 65003
Total number of peers : 1                 Peers in established state : 1

 Peer                    V     AS  MsgRcvd  MsgSent  OutQ  Up/Down       State       PrefRcv
 2001::6:6               4  65100        5        7     0  00:01:57      Established         0
```

The MP-EBGP peer relationship has been successfully established.

# Create a Loopback1 interface on P1, set its IPv6 address to 2001::1:5:5/128, and advertise the route to this interface to BGP.

```
[P1]interface LoopBack1
[P1-LoopBack1] ipv6 enable
[P1-LoopBack1] ipv6 address 2001::1:5:5/128
[P1-LoopBack1] quit
[P1]bgp 65003
[P1-bgp]ipv6-family unicast
[P1-bgp-af-ipv6]network 2001::1:5:5 128
```

# Check the BGP4+ routing table on P2.

```
[P2]display bgp ipv6 routing-table

 BGP Local router ID is 10.0.6.6
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,   i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found


TotalNumber of Routes:1
*>    Network  :2001::1:5:5                PrefixLen :128
      NextHop  :2001::5:5                  LocPrf    :
      MED      :0                          PrefVal   :0
      Label    :
      Path/Ogn :65003i
```

P2 has learned the route advertised by P1.

Configure PE1, P2, and PE3 to establish MP-IBGP peer relationships.

# Configure MP-IBGP peers on PE1 and PE3.

```
[PE1] bgp 65100
[PE1-bgp] router-id 10.0.1.1
[PE1-bgp] peer 2001::6:6 as-number 65100
[PE1-bgp] peer 2001::6:6 connect-interface LoopBack0
[PE1-bgp] ipv6-family unicast
[PE1-bgp-af-ipv6] peer 2001::6:6 enable
```

```
[PE3] bgp 65100
BGP is creating configuration data now. Please wait.....done.
[PE3-bgp] router-id 10.0.3.3
[PE3-bgp] peer 2001::6:6 as-number 65100
[PE3-bgp] peer 2001::6:6 connect-interface LoopBack0
[PE3-bgp] ipv6-family unicast
[PE3-bgp-af-ipv6] peer 2001::6:6 enable
```

# Configure MP-IBGP peers on P2.

```
[P2] bgp 65100
[P2-bgp] peer 2001::1:1 as-number 65100
[P2-bgp] peer 2001::1:1 connect-interface LoopBack0
[P2-bgp] peer 2001::3:3 as-number 65100
```

```
[P2-bgp] peer 2001::3:3 connect-interface LoopBack0
[P2-bgp] ipv6-family unicast
[P2-bgp-af-ipv6] peer 2001::1:1 enable
[P2-bgp-af-ipv6] peer 2001::1:1 next-hop-local
[P2-bgp-af-ipv6] peer 2001::1:1 reflect-client
[P2-bgp-af-ipv6] peer 2001::3:3 enable
[P2-bgp-af-ipv6] peer 2001::3:3 next-hop-local
[P2-bgp-af-ipv6] peer 2001::3:3 reflect-client
```

PE1 and PE3 are configured as RR clients.

# Check the MP-IBGP peer relationships on P2.

```
[P2]display bgp ipv6 peer

 BGP local router ID : 10.0.6.6
 Local AS number : 65100
 Total number of peers : 3                 Peers in established state : 3

  Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv
  2001::1:1       4       65100        5        8     0  00:01:48 Established       0
  2001::3:3       4       65100        5       13     0  00:02:34 Established       0
  2001::5:5       4       65003       46       45     0  00:36:35 Established       1
```

The peer relationships have been established.

# Check the BGP4+ routing tables on PE1 and PE3.

```
[PE1]display bgp ipv6 routing-table

 BGP Local router ID is 10.0.1.1
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found


 Total Number of Routes: 1
 *>      Network  : 2001::1:5:5                  PrefixLen : 128
         NextHop  : 2001::6:6                    LocPrf    :   100
         MED      : 0                            PrefVal   : 0
         Label    :
         Path/Ogn : 65003i
```

```
[PE3]display bgp ipv6 routing-table

 BGP Local router ID is 10.0.3.3
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found
```

```
 Total Number of Routes: 1
 *>      Network     : 2001::1:5:5                          PrefixLen   : 128
         NextHop     : 2001::6:6                            LocPrf      :    100
         MED         : 0                                    PrefVal     : 0
         Label       :
         Path/Ogn    : 65003i
```

PE1 and PE3 have learned the route to Loopback1 of P1.

# Import IS-IS routes to BGP on P2.

```
[P2]bgp 65100
[P2-bgp]ipv6-family unicast
[P2-bgp-af-ipv6]import-route isis 1
```

# Check the BGP4+ routing table on P1.

```
[P1]display bgp ipv6 routing-table

 BGP Local router ID is 10.0.5.5
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,   i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found


 Total Number of Routes: 10
 *>      Network     : 2001::                               PrefixLen : 126
         NextHop     : 2001::6:6                            LocPrf    :
         MED         : 10                                   PrefVal   : 0
         Label       :
         Path/Ogn    : 65100?
 *>      Network     : 2001::8                              PrefixLen : 126
         NextHop     : 2001::6:6                            LocPrf    :
         MED         : 0                                    PrefVal   : 0
         Label       :
         Path/Ogn    : 65100?
 *>      Network     : 2001::14                             PrefixLen : 126
         NextHop     : 2001::6:6                            LocPrf    :
         MED         : 0                                    PrefVal   : 0
         Label       :
         Path/Ogn    : 65100?
 *>      Network     : 2001::18                             PrefixLen : 126
         NextHop     : 2001::6:6                            LocPrf    :
         MED         : 10                                   PrefVal   : 0
         Label       :
         Path/Ogn    : 65100?
 *>      Network     : 2001::1:1                            PrefixLen : 128
         NextHop     : 2001::6:6                            LocPrf    :
         MED         : 10                                   PrefVal   : 0
         Label       :
         Path/Ogn    : 65100?
 *>      Network     : 2001::2:2                            PrefixLen : 128
         NextHop     : 2001::6:6                            LocPrf    :
         MED         : 10                                   PrefVal   : 0
```

```
          Label        :
          Path/Ogn     : 65100?
 *>       Network      : 2001::3:3                    PrefixLen : 128
          NextHop      : 2001::6:6                    LocPrf    :
          MED          : 10                           PrefVal   : 0
          Label        :
          Path/Ogn     : 65100?
 *>       Network      : 2001::4:4                    PrefixLen : 128
          NextHop      : 2001::6:6                    LocPrf    :
          MED          : 10                           PrefVal   : 0
          Label        :
          Path/Ogn     : 65100?
          Network      : 2001::6:6                    PrefixLen : 128
          NextHop      : 2001::6:6                    LocPrf    :
          MED          : 0                            PrefVal   : 0
          Label        :
          Path/Ogn     : 65100?
 *>       Network      : 2001::1:5:5                  PrefixLen : 128
          NextHop      : ::                           LocPrf    :
          MED          : 0                            PrefVal   : 0
          Label        :
          Path/Ogn     :  i
```

P1 has learned the routes in AS 65100.

By now, PE1 and PE3 still cannot communicate with the Loopback1 on P1 because the transit nodes PE2 and PE4 do not have a route to the Loopback1 on P1. To solve this problem, you need to redistribute BGP routes to IS-IS on P2.

# Redistribute BGP routes to IS-IS on P2.

```
[P2-isis-1]
[P2-isis-1]ipv6 import-route bgp
```

# Check whether the IS-IS routing tables on PE2 and PE4 contain routes to Loopback1 of P1.

```
[PE2]display isis route ipv6 2001::1:5:5

                      Routeinformation forISIS(1)
                      ---------------------------

                     ISIS(1) Level-2Forwarding Table
                     -------------------------------

IPV6 Dest.         ExitInterface       NextHop                      Cost     Flags
--------------------------------------------------------------------------------
2001::1:5:5/128    GE0/5/1             FE80::F29B:B8FF:FECC:740B    10       A/-/-/-
    Flags:D-Direct, A-AddedtoURT, L-Advertised in LSPs,S-IGP Shortcut,
          U-Up/Down Bit Set,LP-Local Prefix-Sid
    ProtectType: L-Link Protect, N-NodeProtect
```

```
[PE4]display isis route ipv6 2001::1:5:5

                          Route information for ISIS(1)
                          ----------------------------

                         ISIS(1) Level-2 Forwarding Table
                         --------------------------------

 IPV6 Dest.       ExitInterface          NextHop                       Cost       Flags
-------------------------------------------------------------------------------
2001::1:5:5/128 GE0/5/1                FE80::F29B:B8FF:FECC:740C   10         A/-/-/-
       Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
              U-Up/Down Bit Set, LP-Local Prefix-Sid
       Protect Type: L-Link Protect, N-Node Protect
```

# Test the connectivity among PE1, PE3, and P1's Loopback1 interfaces.

```
[PE1]ping ipv6 -c 1 2001::1:5:5
  PING 2001::1:5:5 : 56  data bytes, press CTRL_C to break
    Reply from 2001::1:5:5
    bytes=56 Sequence=1 hop limit=62 time=2 ms

  --- 2001::1:5:5 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=2/2/2 ms
```

```
[PE3]ping ipv6 -c 1 2001::1:5:5
  PING 2001::1:5:5 : 56  data bytes, press CTRL_C to break
    Reply from 2001::1:5:5
    bytes=56 Sequence=1 hop limit=62 time=1 ms

  --- 2001::1:5:5 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
round-trip min/avg/max=1/1/1 ms
```

**----End**

## 2.1.3 Quiz

What are the differences between BGP IPv4 route transmission and BGP4+ route transmission?

# 3 MPLS VPN

## 3.1 MPLS VPN

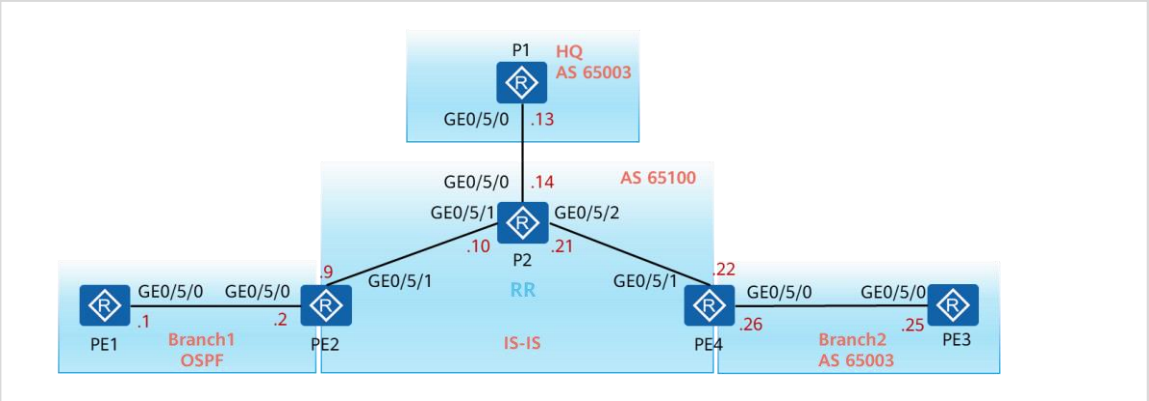
### 3.1.1 About This Lab

#### 3.1.1.1 Objectives

- Configure MPLS.
- Configure MPLS LDP.
- Configure MPLS VPN.
- Configure route exchange between CEs and PEs using different methods.
- Control route learning between CEs based on RT values.

#### 3.1.1.2 Networking Description

**Figure 3-1 MPLS VPN lab topology**




The figure shows the device connection and IP address planning. The interface interconnection addresses are in the format of 10.0.0.*Y*/30, and the values represented by Y are shown in the figure. Loopback0 interfaces are created on all devices, and the IP addresses of Loopback interfaces are in the format of 10.0.*X*.*X*. The values indicated by *X* are shown in the tables related to corresponding configuration procedures.

This lab introduces how to implement route learning between the HQ and the branches by controlling RT values.

### 3.1.2 Lab Task

#### 3.1.2.1 Configuration Roadmap

1. Complete the basic configuration of device IP addresses.

2. Configure IS-IS in AS 65100 and enable IS-IS on interconnection and Loopback0 interfaces for IGP intercommunication in this AS.
3. Configure MPLS in AS 65100 and enable MPLS globally. Configure MPLS LSR IDs and enable MPLS and MPLS LDP on interconnection interfaces.
4. Create VPN instances on P1 to configure the EBGP peer between P1 and P2.
5. Create a VPN instance on PE4 to configure the EBGP peer between PE4 and PE3.
6. Create a VPN instance on PE2 to configure OSPF between PE2 and PE1.

### 3.1.2.2 Configuration Procedure

Step 1 Complete the basic configuration of devices.

Set the command validation mode to immediate validation and configure IP addresses for interconnection and Loopback0 interfaces. Loopback0 interface addresses need to be configured according to the following table.

**Table 3-1 Loopback0 IP addresses**

| Device ID | X value | Loopback0 IP Address |
|---|---|---|
| PE1 | 1 | 10.0.1.1 |
| PE2 | 2 | 10.0.2.2 |
| PE3 | 3 | 10.0.3.3 |
| PE4 | 4 | 10.0.4.4 |
| P1 | 5 | 10.0.5.5 |
| P2 | 6 | 10.0.6.6 |

# Name the devices.

N/A

# Disable unnecessary interfaces in this lab.

N/A

# Configure the command validation mode as immediate validation.

```
<PE1>system-view immediately
```

```
<PE2>system-view immediately
<PE3>system-view immediately
```

```
<PE4>system-view immediately
```

```
<P1>system-view immediately
```

```
<P2>system-view immediately
```

# Disable the DCN function globally on all devices.

```
[PE1] undo dcn
Warning: This operation will disable DCN function. Continue? [Y/N]:y
```

Here, PE1 is used as an example. Repeat the same operations for other devices.

By default, the DCN function is enabled on NE router interfaces. To facilitate this lab, disable the DCN function globally on all devices.

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE1.

```
[PE1]interface LoopBack0
[PE1-LoopBack0] ip address 10.0.1.1 32
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0]ip address 10.0.0.1 30
[PE1-GigabitEthernet0/5/0] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE2.

```
[PE2]interface LoopBack0
[PE2-LoopBack0] ip address 10.0.2.2 32
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ip address 10.0.0.2 30
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ip address 10.0.0.9 30
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE3.

```
[PE3]interface LoopBack0
[PE3-LoopBack0] ip address 10.0.3.3 32
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] ip address 10.0.0.25 30
[PE3-GigabitEthernet0/5/0] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE4.

```
[PE4]interface LoopBack0
[PE4-LoopBack0] ip address 10.0.4.4 32
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
```

```
[PE4-GigabitEthernet0/5/0] ip address 10.0.0.26 30
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] ip address 10.0.0.22 255.255.255.252
[PE4-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on P1.

```
[P1]interface LoopBack0
[P1-LoopBack0] ip address 10.0.5.5 32
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] ip address 10.0.0.13 30
[P1-GigabitEthernet0/5/0] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on P2.

```
[P2]interface LoopBack0
[P2-LoopBack0] ip address 10.0.6.6 32
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ip address 10.0.0.14 30
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ip address 10.0.0.10 30
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] ip address 10.0.0.21 255.255.255.252
```

Step 2 Configure IS-IS.

(Optional) Configure IS-IS in AS 65100. Set the IS-IS area ID to 49.0001 and the process ID to 1. All the devices are Level-2 devices. Configure NET to 49.0001.000*X*.000*X*.000*X*.00 and the values indicated by *X* are listed in the table in Step 1. Enable IS-IS on Loopback0 and interconnection interfaces, and configure the network type of the interfaces to P2P.

In this case, you need to set **cost-style** to **wide** to support IS-IS extensions.

# Configure IS-IS on PE2.

```
[PE2]isis 1
[PE2-isis-1] is-level level-2
[PE2-isis-1] cost-style wide
[PE2-isis-1] network-entity 49.0001.0002.0002.0002.00
[PE2-isis-1] is-name PE2
[PE2-isis-1]interface LoopBack0
[PE2-LoopBack0] isis enable 1
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] isis enable 1
[PE2-GigabitEthernet0/5/1] isis circuit-type p2p
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on PE4.

```
[PE4]isis 1
[PE4-isis-1] is-level level-2
[PE4-isis-1] cost-style wide
[PE4-isis-1] network-entity 49.0001.0004.0004.0004.00
[PE4-isis-1] is-name PE4
[PE4-isis-1] quit
[PE4]interface LoopBack0
[PE4-LoopBack0] isis enable 1
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] isis enable 1
[PE4-GigabitEthernet0/5/1] isis circuit-type p2p
[PE4-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on P2.

```
[P2]isis 1
[P2-isis-1] is-level level-2
[P2-isis-1] cost-style wide
[P2-isis-1] network-entity 49.0001.0006.0006.0006.00
[P2-isis-1] is-name P2
[P2-isis-1] quit
[P2]interface LoopBack0
[P2-LoopBack0] isis enable 1
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] isis enable 1
[P2-GigabitEthernet0/5/1] isis circuit-type p2p
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] isis enable 1
[P2-GigabitEthernet0/5/2] isis circuit-type p2p
[P2-GigabitEthernet0/5/2] quit
```

# Check the IS-IS neighbor relationship on P2.

```
[P2]display isis peer
                        Peer information for ISIS(1)

  System Id      Interface          Circuit Id         State HoldTime Type      PRI
------------------------------------------------------------------------------
PE2              GE0/5/1            0000000007         Up    25s      L2        --
PE4              GE0/5/2            0000000007         Up    30s      L2        --

Total Peer(s): 2
```

The neighbor relationship status is normal.

# Check IS-IS routes on P2.

```
[P2]display isis route

                Routeinformation forISIS(1)
                ---------------------------
```

```
                    ISIS(1) Level-2 Forwarding Table
                    --------------------------------

IPV4 Destination   IntCost    ExtCost ExitInterface     NextHop         Flags
-------------------------------------------------------------------------------
10.0.0.8/30        10         NULL    GE0/5/1           Direct          D/-/L/-
10.0.0.20/30       10         NULL    GE0/5/2           Direct          D/-/L/-
10.0.2.2/32        10         NULL    GE0/5/1           10.0.0.9        A/-/-/-
10.0.4.4/32        10         NULL    GE0/5/2           10.0.0.22       A/-/-/-
10.0.6.6/32        0          NULL    Loop0             Direct          D/-/L/-
    Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
           U-Up/Down Bit Set, LP-Local Prefix-Sid
    Protect Type: L-Link Protect, N-Node Protect
```

P2 has learned the routes.

Step 3 Configure MPLS.

Enable MPLS on all devices, configure MPLS LSR IDs (by using the IP addresses of Loopback0), and enable MPLS and MPLS LDP on the interconnection interfaces.

# Configure MPLS LSR IDs and enable MPLS globally.

```
[PE2]mpls lsr-id 10.0.2.2
[PE2]mpls
[PE2]mpls ldp
```

```
[PE4]mpls lsr-id 10.0.4.4
[PE4]mpls
[PE4]mpls ldp
```

```
[P2]mpls lsr-id 10.0.6.6
[P2]mpls
[P2]mpls ldp
```

# Enable MPLS and MPLS LDP on the interconnection interfaces.

```
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] mpls
[P2-GigabitEthernet0/5/1] mpls ldp
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] mpls
[P2-GigabitEthernet0/5/2] mpls ldp
[P2-GigabitEthernet0/5/2] quit
```

```
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] mpls
[PE2-GigabitEthernet0/5/1] mpls ldp
```

```
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] mpls
[PE4-GigabitEthernet0/5/1] mpls ldp
```

# Check MPLS LDP session status on P2.

```
[P2]display mpls ldp adjacency all
 LDP Adjacency Information
 Codes: R: Remote Adjacency, L: Local Adjacency
 An asterisk (*) before an adjacency means the adjacency is being deleted.
 ------------------------------------------------------------------------------
 SN    SourceAddr      PeerID          VrfID AdjAge(DDDD:HH:MM) RcvdHello Type
 ------------------------------------------------------------------------------
 1     10.0.0.9        10.0.2.2        0     0000:00:01         17         L
 2     10.0.0.22       10.0.4.4        0     0000:00:00         13         L
 ------------------------------------------------------------------------------
 TOTAL: 2 Record(s) Found.
```

## Step 4 Configure L3VPN.

Configure the MP-BGP peers in AS 65100. Use the IP address of Loopback0 as the router ID as well as the source IP address for establishing BGP peer relationships. Configure P2 as an RR to transmit VPNv4 routes.

# Configure PE2.

```
[PE2]bgp 65100
[PE2-bgp] router-id 10.0.2.2
[PE2-bgp] peer 10.0.6.6 as-number 65100
[PE2-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[PE4]bgp 65100
[PE4-bgp] router-id 10.0.4.4
[PE4-bgp] peer 10.0.6.6 as-number 65100
[PE4-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE4-bgp] ipv4-family vpnv4
[PE4-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[P2]bgp 65100
[P2-bgp] router-id 10.0.6.6
[P2-bgp] peer 10.0.2.2 as-number 65100
[P2-bgp] peer 10.0.2.2 connect-interface LoopBack0
[P2-bgp] peer 10.0.4.4 as-number 65100
[P2-bgp] peer 10.0.4.4 connect-interface LoopBack0
[P2-bgp] ipv4-family vpnv4
[P2-bgp-af-vpnv4] undo policy vpn-target
```

```
[P2-bgp-af-vpnv4] peer 10.0.2.2 enable
[P2-bgp-af-vpnv4] peer 10.0.2.2 reflect-client
[P2-bgp-af-vpnv4] peer 10.0.4.4 enable
[P2-bgp-af-vpnv4] peer 10.0.4.4 reflect-client
```

When configuring an RR, disable the RT check on VPNv4 routes.

# Check the VPNv4 peer relationship status on P2.

```
[P2]display bgp vpnv4 all peer
BGP local router ID : 10.0.6.6
Local AS number : 65100
Total number of peers : 2                 Peers in established state : 2

 Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State       PrefRcv
 10.0.2.2        4       65100        6        6     0  00:02:05   Established         0
 10.0.4.4        4       65100        6        6     0  00:02:04   Established         0
```

Step 5 The configuration between CEs and PEs in the HQ is completed.

From this step, you need to configure the routing protocol between CEs and PEs and the following results are reached via configuring IRT and ERT:

1. The HQ can normally learn the routes from Branch1 and Branch2.
2. The next hop of the peer routes learned between the branches is the HQ.
3. The branches implement mutual access through the HQ.

To meet the preceding requirements, you need to create two VPN instances between P1 and P2 to receive and send routes. Therefore, create a sub-interface in addition to the physical interfaces.

# Configure the sub-interface between P1 and P2.

```
[P1]interface GigabitEthernet0/5/0.1
[P1-GigabitEthernet0/5/0.1] vlan-type dot1q 10
[P1-GigabitEthernet0/5/0.1] ip address 10.0.0.113 255.255.255.252
[P1-GigabitEthernet0/5/0.1] quit
```

```
[P2]interface GigabitEthernet0/5/0.1
[P2-GigabitEthernet0/5/0.1] vlan-type dot1q 10
[P2-GigabitEthernet0/5/0.1] ip address 10.0.0.114 255.255.255.252
[P2-GigabitEthernet0/5/0.1] quit
```

# Test the connectivity of the sub-interface.

```
[P2]ping 10.0.0.113
  PING 10.0.0.113: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.113: bytes=56 Sequence=1 ttl=255 time=1 ms
    Reply from 10.0.0.113: bytes=56 Sequence=2 ttl=255 time=1 ms
    Reply from 10.0.0.113: bytes=56 Sequence=3 ttl=255 time=1 ms
    Reply from 10.0.0.113: bytes=56 Sequence=4 ttl=255 time=1 ms
    Reply from 10.0.0.113: bytes=56 Sequence=5 ttl=255 time=1 ms
```

```
--- 10.0.0.113 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/1/1 ms
```

# Configure BGP for P1, use the IP address of Loopback0 as the router ID, and establish an EBGP peer relationship through directly connected interfaces.

```
[P1]bgp 65003
[P1-bgp] router-id 10.0.5.5
[P1-bgp] peer 10.0.0.14 as-number 65100
[P1-bgp] peer 10.0.0.114 as-number 65100
```

# Create VPN instance **vpna_in** on P2 to receive routes and **vpna_out** to send routes.

In the entire network, configure VPN instances on the PE devices in MPLS VPN according to the following planning.

**Table 3-2 VPN instance planning**

| Device ID | VPN Instance | RD | IRT | ERT |
|---|---|---|---|---|
| P2 | vpna_in | 10:60 | 10:46, 10:26 | - |
| P2 | vpna_out | 10:6060 | - | 10:624 |
| PE2 | vpna | 10:20 | 10:624 | 10:26 |
| PE4 | vpna | 10:40 | 10:624 | 10:46 |

```
[P2]ip vpn-instance vpna_in
[P2-vpn-instance-vpna_in] ipv4-family
[P2-vpn-instance-vpna_in-af-ipv4] route-distinguisher 10:60
[P2-vpn-instance-vpna_in-af-ipv4] vpn-target 10:46 import-extcommunity
[P2-vpn-instance-vpna_in-af-ipv4] vpn-target 10:26 import-extcommunity
[P2-vpn-instance-vpna_in-af-ipv4] quit
[P2-vpn-instance-vpna_in] quit
```

```
[P2]ip vpn-instance vpna_out
[P2-vpn-instance-vpna_out] ipv4-family
[P2-vpn-instance-vpna_out-af-ipv4] route-distinguisher 10:6060
[P2-vpn-instance-vpna_out-af-ipv4] vpn-target 10:624 export-extcommunity
[P2-vpn-instance-vpna_out-af-ipv4] quit
[P2-vpn-instance-vpna_out] quit
```

# After configuring the VPN instances, add the interfaces into corresponding VPN instances.

```
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ip binding vpn-instance vpna_out
[P2-GigabitEthernet0/5/0] ip address 10.0.0.14 255.255.255.252
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/0.1
[P2-GigabitEthernet0/5/0.1] vlan-type dot1q 10
[P2-GigabitEthernet0/5/0.1] ip binding vpn-instance vpna_in
[P2-GigabitEthernet0/5/0.1] ip address 10.0.0.114 255.255.255.252
```

Reconfigure IP addresses for the interfaces.

# Configure EBGP on P2.

```
[P2]bgp 65100
[P2-bgp] ipv4-family vpn-instance vpna_in
[P2-bgp-vpna_in] peer 10.0.0.113 as-number 65003
[P2-bgp-vpna_in] peer 10.0.0.113 substitute-as
[P2-bgp-vpna_in] ipv4-family vpn-instance vpna_out
[P2-bgp-vpna_out] peer 10.0.0.13 as-number 65003
[P2-bgp-vpna_out] peer 10.0.0.13 allow-as-loop 2
```

The AS_Path attribute of the BGP VPNv4 routes received in the VPN instance **vpna_in** carries AS 65003. P1 discards the routes due to loop prevention mechanism. Therefore, you need to configure the **substitute-as** parameter to substitute the AS.

The branch routes learned in **vpna_in** are advertised to the **vpna_out** instance in the form of BGP routes through the CE (P1) of the HQ. However, P2 does not learn the routes because they carry the local AS ID. Therefore, the **allow-as-loop** parameter must be configured.

# Check the BGP peer relationship status on P1.

```
[P1]display bgp peer
BGP local router ID : 10.0.5.5
Local AS number : 65003
Total number of peers : 2                 Peers in established state : 2

 Peer            V     AS  MsgRcvd  MsgSent  OutQ  Up/Down       State       PrefRcv
 10.0.0.14       4  65100     1258     1263     0  18:13:13  Established         1
 10.0.0.114      4  65100     1290     1298     0  18:38:11  Established         1
```

The peer relationship has been normally established.

# Create Loopback1 on P1 to simulate a service network segment.

```
[P1]interface LoopBack1
[P1-LoopBack1] ip address 10.1.5.5 32
```

Configure its IP address to 10.1.5.5/32.

# Advertise the route into BGP on P1.

```
[P1] bgp 65003
[P1-bgp] network 10.1.5.5 255.255.255.255
```

Step 6 Complete the configuration between CEs and PEs in Branch2.

Configure a VPN instance on PE4 according to the planning, and add an interface to the VPN instance. Configure EBGP between PE4 and PE3. Create Loopback1 interface on PE3 to simulate a service network segment, and import it into BGP.

#Create a VPN instance on PE4.

```
[PE4]ip vpn-instance vpna
[PE4-vpn-instance-vpna] ipv4-family
[PE4-vpn-instance-vpna-af-ipv4] route-distinguisher 10:40
[PE4-vpn-instance-vpna-af-ipv4] vpn-target 10:46 export-extcommunity
[PE4-vpn-instance-vpna-af-ipv4] vpn-target 10:624 import-extcommunity
[PE4-vpn-instance-vpna-af-ipv4] quit
[PE4-vpn-instance-vpna] quit
```

# Add the interface that connects the CEs to the VPN instance.

```
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] ip binding vpn-instance vpna
[PE4-GigabitEthernet0/5/0] ip address 10.0.0.26 255.255.255.252
```

# Complete EBGP configuration on PE4 and establish a BGP peer relationship through directly connected interfaces.

```
[PE4]bgp 65100
[PE4-bgp] ipv4-family vpn-instance vpna
[PE4-bgp-vpna] peer 10.0.0.25 as-number 65003
[PE4-bgp-vpna] peer 10.0.0.25 substitute-as
```

Configure the **substitute-as** parameter.

# Configure EBGP on PE3 and use the IP address of Loopback0 as the router ID.

```
[PE3]bgp 65003
[PE3-bgp] router-id 10.0.3.3
[PE3-bgp] peer 10.0.0.26 as-number 65100
```

# Check the EBGP peer relationship status on PE4.

```
[PE4]display bgp vpnv4 vpn-instance vpna peer

BGP local router ID : 10.0.4.4
Local AS number : 65100

VPN-Instance vpna, Router ID 10.0.4.4:
Total number of peers : 1                 Peers in established state : 1

 Peer                    V          S       MsgRcvd   MsgSent   OutQ  Up/Down       State       PrefRcv
 10.0.0.25               4          65003   1290      1298      0     18:42:33      Established 1
```

The peer relationship has been normally established.

# Create Loopback1 interface on PE3 and advertise it into BGP.

```
[PE3]interface LoopBack1
[PE3-LoopBack1] ip address 10.1.3.3 32
[PE3-LoopBack1] quit
[PE3]bgp 65003
[PE3-bgp] network 10.1.3.3 255.255.255.255
```

Step 7 Complete the configuration between CEs and PEs in Branch1.

Configure a VPN instance on PE2 according to the planning, and add an interface to the VPN instance. Configure OSPF between PE2 and PE1. Create Loopback1 interface on PE1 to simulate a service network segment, and import it into OSPF.

# Create a VPN instance on PE2.

```
[PE2]ip vpn-instance vpna
[PE2-vpn-instance-vpna] ipv4-family
[PE2-vpn-instance-vpna-af-ipv4] route-distinguisher 10:20
[PE2-vpn-instance-vpna-af-ipv4] apply-label per-instance
[PE2-vpn-instance-vpna-af-ipv4] vpn-target 10:26 export-extcommunity
[PE2-vpn-instance-vpna-af-ipv4] vpn-target 10:624 import-extcommunity
```

# Add PE2 interfaces to the VPN instance.

```
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ip binding vpn-instance vpna
[PE2-GigabitEthernet0/5/0] ip address 10.0.0.2 255.255.255.252
```

Reconfigure IP addresses for the interfaces.

# Configure OSPF on PE2.

```
[PE2]ospf 1 router-id 10.0.2.2 vpn-instance vpna
[PE2-ospf-1] import-route bgp cost 20 type 2
[PE2-ospf-1] area 0.0.0.0
```

Import BGP routes.

# Enable OSPF on the interconnection interfaces.

```
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ospf enable 1 area 0.0.0.0
```

# Configure OSPF on PE1.

```
[PE1]ospf 1 router-id 10.0.1.1
[PE1-ospf-1] area 0.0.0.0
```

# Enable OSPF on the interconnection interfaces.

```
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0] ospf enable 1 area 0.0.0.0
```

# Create Loopback1 interface on PE1 and enable OSPF on the interface.

```
[PE1]interface LoopBack1
[PE1-LoopBack1] ip address 10.1.1.1 32
[PE1-LoopBack1] ospf enable 1 area 0.0.0.0
```

# Check the OSPF neighbor relationship status on PE2.

```
[PE2]display ospf peer brief
(M) Indicates MADJ neighbor


             OSPF Process 1 with Router ID 10.0.2.2
                     Peer Statistic Information
Total number of peer(s): 1
 Peer(s) in full state: 1
--------------------------------------------------------------------------------
 Area Id          Interface          Neighbor id      State
 0.0.0.0          GE0/5/0            10.0.1.1           Full
--------------------------------------------------------------------------------
```

The neighbor relationship status is normal.

# Restrict OSPF routes imported into BGP by configuring IP prefix list and route-policy.

```
[PE2]ip ip-prefix Loopback1 index 10 permit 10.1.1.1 32
[PE2]route-policy O2B permit node 10
[PE2-route-policy] if-match ip-prefix Loopback1
```

# Import OSPF routes in the BGP VPN instance view.

```
[PE2]bgp 65100
[PE2-bgp] ipv4-family vpn-instance vpna
[PE2-bgp-vpna] import-route ospf 1 route-policy O2B
```

## Step 8 Verify routes.

Check the BGP VPNv4 routes of the CEs and PEs from the branches and the HQ, as well as their corresponding IGP and BGP protocol routing tables.

# Check the VPNv4 routing table on P2 in the HQ.

```
[P2]display bgp vpnv4 vpn-instance vpna_in routing-table
 BGP Local router ID is 10.0.6.6
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,   i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found


 VPN-Instance   vpna_in, Router ID 10.0.6.6:

 Total Number of Routes: 4
      Network            NextHop           MED        LocPrf    PrefVal Path/Ogn
 *>   10.1.5.5/32        10.0.0.113        0                    0       65003i
 *>i  10.1.1.1/32        10.0.2.2          2          100       0       ?
 *>i  10.1.3.3/32        10.0.4.4          0          100       0       65003i
```

In VPN instance **vpna_in**, branch routes 10.1.1.1/32 and 10.1.3.3/32 have been learned.

```
[P2]display bgp vpnv4 vpn-instance vpna_out routing-table
BGP Local router ID is 10.0.6.6
Status codes: * - valid, >- best, d- damped,x - best external,a - add path,
              h- history, i- internal, s - suppressed,S - Stale
              Origin : i- IGP, e - EGP, ? - incomplete
RPKI validation codes: V- valid, I- invalid, N - not-found


VPN-Instance vpna_out, Router ID 10.0.6.6:

Total Number of Routes:4
     Network            NextHop          MED        LocPrf    PrefVal Path/Ogn

*>   10.1.5.5/32        10.0.0.13        0                    0      65003i
*>   10.1.1.1/32        10.0.0.13                             0      65003 65100?
*>   10.1.3.3/32        10.0.0.13                             0      65003 65100 65100i
```

In VPN instance **vpna_out**, routes from the HQ (10.1.1.1/32 and 10.1.3.3/32) have been learned. In this case, the next hop is 10.0.0.13, that is P1.

# Check the BGP VPNv4 routing tables on PE2 and PE4.

```
[PE2]display bgp vpnv4 all routing-table
BGP Local router ID is 10.0.2.2
Status codes: * - valid, >- best, d- damped,x - best external,a - add path,
              h- history, i- internal, s - suppressed,S - Stale
              Origin : i- IGP, e - EGP, ? - incomplete
RPKI validation codes: V- valid, I- invalid, N - not-found

Total number of routes from all PE: 4
Route Distinguisher: 10:20


     Network            NextHop          MED        LocPrf    PrefVal Path/Ogn

*>   10.1.1.1/32        0.0.0.0          2                    0      ?
Route Distinguisher: 10:6060

     Network            NextHop          MED        LocPrf    PrefVal Path/Ogn

*>i  10.1.1.1/32        10.0.6.6                    100        0      65003 65100?
*>i  10.1.3.3/32        10.0.6.6                    100        0      65003 65100 65100i
*>i  10.1.5.5/32        10.0.6.6         0          100        0      65003i

VPN-Instance  vpna, Router ID 10.0.2.2:

Total Number of Routes:4
     Network            NextHop          MED        LocPrf    PrefVal Path/Ogn

*>   10.1.1.1/32        0.0.0.0          2                    0      ?
* i                     10.0.6.6                    100        0      65003 65100?
*>i  10.1.3.3/32        10.0.6.6                    100        0      65003 65100 65100i
*>i  10.1.5.5/32        10.0.6.6         0          100        0      65003i
```

PE2 has learned the service routes to the other branch and the HQ, and the next hops are the PEs of the HQ.

```
[PE4]display bgp vpnv4 all routing-table
BGP Local router ID is 10.0.4.4
Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V - valid, I - invalid, N - not-found


Total number of routes from all PE: 4
Route Distinguisher: 10:40


      Network         NextHop                MED        LocPrf    PrefVal Path/Ogn

*>    10.1.3.3/32     10.0.0.25               0                0      65003i
Route Distinguisher: 10:6060


      Network       NextHop                MED        LocPrf    PrefVal Path/Ogn

*>i   10.1.1.1/32     10.0.6.6                          100        0      65003 65100?
*>i   10.1.3.3/32     10.0.6.6                          100        0      65003 65100 65100i
*>i   10.1.5.5/32     10.0.6.6                0         100        0      65003i

VPN-Instance vpna, Router ID 10.0.4.4:

Total Number of Routes: 4
      Network         NextHop                MED        LocPrf    PrefVal Path/Ogn

*>i   10.1.1.1/32     10.0.6.6                          100        0      65003 65100?
*>    10.1.3.3/32     10.0.0.25               0                  0      65003i
* i                   10.0.6.6                          100        0      65003 65100 65100i
*>i   10.1.5.5/32     10.0.6.6                0         100        0      65003i
```

PE4 has learned the service routes to the other branch and the HQ, and the next hops are the PEs of the HQ.

# Check the IGP protocol routing tables on PE1 and PE3.

```
[PE1]display ospf routing
            OSPF Process 1 with Router ID 10.0.1.1
                  Routing Tables

 Routing for Network
 Destination        Cost      Type       NextHop         AdvRouter       Area
 10.0.0.0/30        1         Direct     10.0.0.1        10.0.1.1        0.0.0.0
 10.1.1.1/32        0         Direct     10.1.1.1        10.0.1.1        0.0.0.0

 Routing for ASEs
 Destination        Cost      Type       Tag             NextHop         AdvRouter
 10.1.3.3/32        20        Type2      3489726028      10.0.0.2        10.0.2.2
```

```
 10.1.5.5/32          20           Type2       3489726028    10.0.0.2        10.0.2.2

 Total Nets: 4
 Intra Area: 2  Inter Area: 0  ASE: 2  NSSA: 0
```

The OSPF routing table on PE1 has contained the routes to the HQ and the branches.

```
[PE3]display bgp routing-table
BGP Local router ID is 10.0.3.3
Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V - valid, I - invalid, N - not-found


Total Number of Routes: 3
     Network         NextHop                MED        LocPrf    PrefVal Path/Ogn

*>   10.1.1.1/32     10.0.0.26                             0       65100 65100 65100?
*>   10.1.3.3/32     0.0.0.0                0              0       i
*>   10.1.5.5/32     10.0.0.26                             0       65100 65100i
```

The BGP routing table on PE3 has contained the routes to the HQ and the branches.

# Test service connectivity on PE3.

```
[PE3]ping -a 10.1.3.3 10.1.1.1
  PING 10.1.1.1: 56  data bytes, press CTRL_C to break
    Reply from 10.1.1.1: bytes=56 Sequence=1 ttl=250 time=1 ms
    Reply from 10.1.1.1: bytes=56 Sequence=2 ttl=250 time=1 ms
    Reply from 10.1.1.1: bytes=56 Sequence=3 ttl=250 time=1 ms
    Reply from 10.1.1.1: bytes=56 Sequence=4 ttl=250 time=1 ms
    Reply from 10.1.1.1: bytes=56 Sequence=5 ttl=250 time=1 ms

  --- 10.1.1.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

The communication is normal. The source addresses must be local service network addresses.

# Check the service traffic paths.

```
[PE3]tracert -a 10.1.3.3 10.1.1.1
 traceroute to 10.1.1.1(10.1.1.1), max hops: 30, packet length: 40, press CTRL_C to break
 1 10.0.0.26 27 ms  2 ms  3 ms
 2 10.0.0.14 4 ms  2 ms  3 ms
 3 10.0.0.13 2 ms  2 ms  2 ms
 4 10.0.0.114 3 ms  3 ms  4 ms
 5 10.0.0.2 6 ms  2 ms  4 ms
 6 10.1.1.1 3 ms  2 ms  2 ms
```

The command output shows that the traffic is forwarded through P1 of the HQ. If the HQ wants to restrict the communication between branches, you can perform related operations.

**----End**

## 3.1.3 Quiz

The tag value in the OSPF routing table on PE1 is 3489726028, how is the value generated?

# 4 EVPN

## 4.1 EVPN L3VPNv4 over MPLS

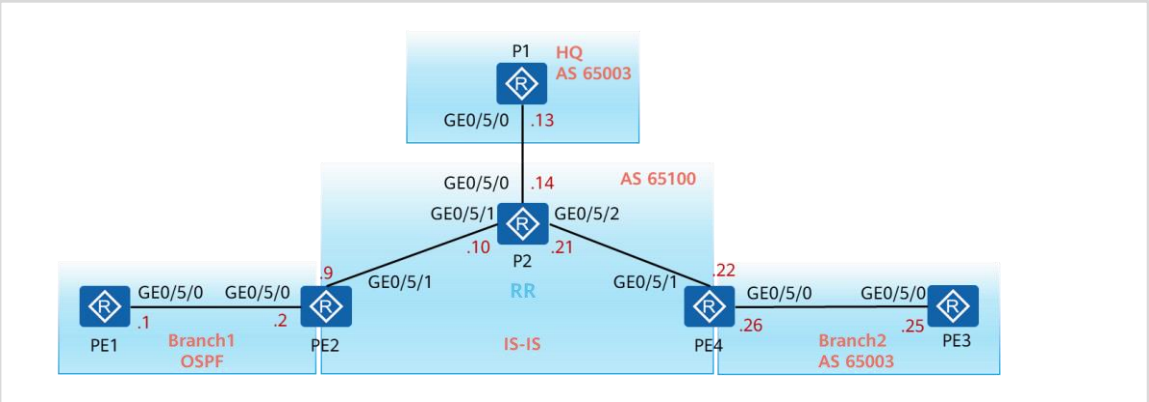
### 4.1.1 About This Lab

#### 4.1.1.1 Objectives

- Configure MPLS.
- Configure MPLS LDP.
- Carry L3VPNv4 over EVPN.

#### 4.1.1.2 Networking Description

**Figure 4-1 EVPN L3VPNv4 over MPLS lab topology**




The figure shows the device connection and IP address planning. The interface interconnection addresses are in the format of 10.0.0.*Y*/30, and the values represented by Y are shown in the figure. Loopback0 interfaces are created for all devices, and the IP addresses of Loopback interfaces are in the format of 10.0.*X*.*X*. The values indicated by *X* are shown in the tables related to corresponding configuration procedures.

This lab introduces how to implement route learning between the HQ and the branches by controlling RT values. Also, it involves how to transmit routes between different branches by using EVPN as the transport protocol.

### 4.1.2 Lab Task

#### 4.1.2.1 Configuration Roadmap

1. Complete the basic configuration of device IP addresses.

2. Configure IS-IS in AS 65100 and enable IS-IS on interconnection and Loopback0 interfaces for IGP intercommunication in this AS.
3. Configure MPLS in AS 65100 and enable MPLS globally. Configure MPLS LSR IDs and enable MPLS and MPLS LDP on interconnection interfaces.
4. Create VPN instances on P1 to configure the EBGP peer between P1 and P2.
5. Create a VPN instance on PE4 to configure the EBGP peer between PE4 and PE3.
6. Create a VPN instance on PE2 to configure OSPF between PE2 and PE1.

### 4.1.2.2 Configuration Procedure

Step 1 Complete the basic configuration of devices.

Set the command validation mode to immediate validation and configure IP addresses for interconnection and Loopback0 interfaces. Loopback0 interface addresses need to be configured according to the following table.

**Table 4-1 Loopback0 IP addresses**

| Device ID | X value | Loopback0 IP Address |
|---|---|---|
| PE1 | 1 | 10.0.1.1 |
| PE2 | 2 | 10.0.2.2 |
| PE3 | 3 | 10.0.3.3 |
| PE4 | 4 | 10.0.4.4 |
| P1 | 5 | 10.0.5.5 |
| P2 | 6 | 10.0.6.6 |

# Name the devices.

N/A

# Disable unnecessary interfaces.

N/A

# Configure the command validation mode as immediate validation.

```
<PE1>system-view immediately
```

```
<PE2>system-view immediately
```

```
<PE3>system-view immediately
```

```
<PE4>system-view immediately
```

```
<P1>system-view immediately
```

```
<P2>system-view immediately
```

# Disable the DCN function globally on all devices.

```
[PE1] undo dcn
Warning: This operation will disable DCN function. Continue? [Y/N]:y
```

Here, PE1 is used as an example. Repeat the same operations for other devices.

By default, the DCN function is enabled on NE router interfaces. To facilitate this lab, disable the DCN function globally on all devices.

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE1.

```
[PE1]interface LoopBack0
[PE1-LoopBack0] ip address 10.0.1.1 32
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0]ip address 10.0.0.1 30
[PE1-GigabitEthernet0/5/0] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE2.

```
[PE2]interface LoopBack0
[PE2-LoopBack0] ip address 10.0.2.2 32
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ip address 10.0.0.2 30
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ip address 10.0.0.9 30
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE3.

```
[PE3]interface LoopBack0
[PE3-LoopBack0] ip address 10.0.3.3 32
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] ip address 10.0.0.25 30
[PE3-GigabitEthernet0/5/0] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE4.

```
[PE4]interface LoopBack0
```

```
[PE4-LoopBack0] ip address 10.0.4.4 32
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] ip address 10.0.0.26 30
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] undo shutdown
[PE4-GigabitEthernet0/5/1] ip address 10.0.0.22 255.255.255.252
[PE4-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on P1.

```
[P1]interface LoopBack0
[P1-LoopBack0] ip address 10.0.5.5 32
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] ip address 10.0.0.13 30
[P1-GigabitEthernet0/5/0] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on P2.

```
[P2]interface LoopBack0
[P2-LoopBack0] ip address 10.0.6.6 32
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ip address 10.0.0.14 30
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ip address 10.0.0.10 30
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] undo shutdown
[P2-GigabitEthernet0/5/2] ip address 10.0.0.21 255.255.255.252
```

## Step 2 Configure IS-IS.

(Optional) Configure IS-IS in AS 65100. Set the IS-IS area ID to 49.0001 and the process ID to 1. All the devices are Level-2 devices. Configure NET to 49.0001.000*X*.000*X*.000*X*.00. The values indicated by *X* are shown in the table in Step 1. Enable IS-IS on Loopback0 and interconnection interfaces, and configure the network type of the interfaces to P2P.

In this case, you need to set **cost-style** to **wide** to support IS-IS extensions.

# Configure IS-IS on PE2.

```
[PE2]isis 1
[PE2-isis-1] is-level level-2
[PE2-isis-1] cost-style wide
[PE2-isis-1] network-entity 49.0001.0002.0002.0002.00
[PE2-isis-1] is-name PE2
[PE2-isis-1]interface LoopBack0
[PE2-LoopBack0] isis enable 1
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] isis enable 1
```

```
[PE2-GigabitEthernet0/5/1] isis circuit-type p2p
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on PE4.

```
[PE4]isis 1
[PE4-isis-1] is-level level-2
[PE4-isis-1] cost-style wide
[PE4-isis-1] network-entity 49.0001.0004.0004.0004.00
[PE4-isis-1] is-name PE4
[PE4-isis-1] quit
[PE4]interface LoopBack0
[PE4-LoopBack0] isis enable 1
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] isis enable 1
[PE4-GigabitEthernet0/5/1] isis circuit-type p2p
[PE4-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on P2.

```
[P2]isis 1
[P2-isis-1] is-level level-2
[P2-isis-1] cost-style wide
[P2-isis-1] network-entity 49.0001.0006.0006.0006.00
[P2-isis-1] is-name P2
[P2-isis-1] quit
[P2]interface LoopBack0
[P2-LoopBack0] isis enable 1
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] isis enable 1
[P2-GigabitEthernet0/5/1] isis circuit-type p2p
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] isis enable 1
[P2-GigabitEthernet0/5/2] isis circuit-type p2p
[P2-GigabitEthernet0/5/2] quit
```

# Check the IS-IS neighbor relationship on P2.

```
[P2]display isis peer
                        Peer information for ISIS(1)

  System Id     Interface          Circuit Id         State HoldTime Type     PRI
------------------------------------------------------------------------------
PE2             GE0/5/1            0000000007         Up    25s      L2       --
PE4             GE0/5/2            0000000007         Up    30s      L2       --

Total Peer(s): 2
```

The neighbor relationship status is normal.

# Check IS-IS routes on P2.

```
[P2]display isis route

                    Routeinformation forISIS(1)
                    ---------------------------

                   ISIS(1) Level-2Forwarding Table
                   -------------------------------

IPV4Destination   IntCost    ExtCost ExitInterface     NextHop         Flags
--------------------------------------------------------------------------------
10.0.0.8/30       10          NULL    GE0/5/1           Direct          D/-/L/-
10.0.0.20/30      10          NULL    GE0/5/2           Direct          D/-/L/-
10.0.2.2/32       10          NULL    GE0/5/1           10.0.0.9        A/-/-/-
10.0.4.4/32       10          NULL    GE0/5/2           10.0.0.22       A/-/-/-
10.0.6.6/32       0           NULL    Loop0             Direct          D/-/L/-
    Flags:D-Direct, A-AddedtoURT, L-Advertised in LSPs,S-IGP Shortcut,
          U-Up/Down Bit Set,LP-Local Prefix-Sid
    Protect Type: L-Link Protect, N-Node Protect
```

P2 has learned the routes.

Step 3 Configure MPLS.

Enable MPLS on all devices, configure MPLS LSR IDs (by using the IP addresses of Loopback0), and enable MPLS and MPLS LDP on the interconnection interfaces.

# Configure MPLS LSR IDs and enable MPLS globally.

```
[PE2]mpls lsr-id 10.0.2.2
[PE2]mpls
[PE2]mpls ldp
```

```
[PE4]mpls lsr-id 10.0.4.4
[PE4]mpls
[PE4]mpls ldp
```

```
[P2]mpls lsr-id 10.0.6.6
[P2]mpls
[P2]mpls ldp
```

# Enable MPLS and MPLS LDP on the interconnection interfaces.

```
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] mpls
[P2-GigabitEthernet0/5/1] mpls ldp
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] mpls
[P2-GigabitEthernet0/5/2] mpls ldp
[P2-GigabitEthernet0/5/2] quit
```

```
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] mpls
[PE2-GigabitEthernet0/5/1] mpls ldp
```

```
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] mpls
[PE4-GigabitEthernet0/5/1] mpls ldp
```

# Check MPLS LDP session status on P2.

```
[P2]display mpls ldp adjacency all
LDP Adjacency Information
Codes:R: Remote Adjacency, L: Local Adjacency
An asterisk (*) before an adjacency means the adjacency is being deleted.
--------------------------------------------------------------------------------
SN  SourceAddr      PeerID       VrfID AdjAge(DDDD:HH:MM) RcvdHello Type
--------------------------------------------------------------------------------
1   10.0.0.9        10.0.2.2     0     0000:00:01         17       L
2   10.0.0.22       10.0.4.4     0     0000:00:00         13       L
--------------------------------------------------------------------------------
TOTAL:2 Record(s) Found.
```

## Step 4 Configure EVPN L3VPNv4.

Configure the MP-BGP peers in AS 65100. Use the IP address of Loopback0 as the router ID as well as the source IP address for establishing BGP peer relationships. Configure P2 as an RR to transmit EVPN routes.

# Configure PE2.

```
[PE2]bgp 65100
[PE2-bgp] router-id 10.0.2.2
[PE2-bgp] peer 10.0.6.6 as-number 65100
[PE2-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE2-bgp] l2vpn-family evpn
[PE2-bgp-af-evpn] policy vpn-target
[PE2-bgp-af-evpn] peer 10.0.6.6 enable
```

# Configure P2.

```
[P2]bgp 65100
[P2-bgp] router-id 10.0.6.6
[P2-bgp] peer 10.0.2.2 as-number 65100
[P2-bgp] peer 10.0.2.2 connect-interface LoopBack0
[P2-bgp] peer 10.0.4.4 as-number 65100
[P2-bgp] peer 10.0.4.4 connect-interface LoopBack0
[P2-bgp] l2vpn-family evpn
[P2-bgp-af-evpn] undo policy vpn-target
[P2-bgp-af-evpn] peer 10.0.2.2 enable
[P2-bgp-af-evpn] peer 10.0.2.2 reflect-client
```

```
[P2-bgp-af-evpn] peer 10.0.4.4 enable
[P2-bgp-af-evpn] peer 10.0.4.4 reflect-client
```

# Configure PE4.

```
[PE4]bgp 65100
[PE4-bgp] router-id 10.0.4.4
[PE4-bgp] peer 10.0.6.6 as-number 65100
[PE4-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE4-bgp] l2vpn-family evpn
[PE4-bgp-af-evpn] policy vpn-target
[PE4-bgp-af-evpn] peer 10.0.6.6 enable
```

# Check the MP-BGP EVPN peer relationship on P2.

```
[P2]display bgp evpn peer
BGP local router ID : 10.0.6.6
Local AS number : 65100
Total number of peers : 2                 Peers in established state : 2

 Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State    PrefRcv
 10.0.2.2        4       65100        4        4     0  00:00:37 Established          0
 10.0.4.4        4       65100        4        4     0  00:00:11 Established          0
```

## Step 5 Complete the configuration between CEs and PEs in the HQ.

From this step, you need to configure the routing protocol between CEs and PEs and the following results are reached via configuring IRT and ERT:

1. The HQ can normally learn the routes from Branch1 and Branch2.
2. The next hop of the peer routes learned between the branches is the HQ.
3. The branches implement mutual access through the HQ.

To meet the preceding requirements, you need to create two VPN instances between P1 and P2 to receive and send routes. Therefore, create a sub-interface in addition to the physical interfaces.

# Configure the sub-interface between P1 and P2.

```
[P1]interface GigabitEthernet0/5/0.1
[P1-GigabitEthernet0/5/0.1] vlan-type dot1q 10
[P1-GigabitEthernet0/5/0.1] ip address 10.0.0.113 255.255.255.252
[P1-GigabitEthernet0/5/0.1] quit
```

```
[P2]interface GigabitEthernet0/5/0.1
[P2-GigabitEthernet0/5/0.1] vlan-type dot1q 10
[P2-GigabitEthernet0/5/0.1] ip address 10.0.0.114 255.255.255.252
[P2-GigabitEthernet0/5/0.1] quit
```

# Test the connectivity of the sub-interface.

```
[P2]ping 10.0.0.113
  PING 10.0.0.113: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.113: bytes=56 Sequence=1 ttl=255 time=1 ms
    Reply from 10.0.0.113: bytes=56 Sequence=2 ttl=255 time=1 ms
    Reply from 10.0.0.113: bytes=56 Sequence=3 ttl=255 time=1 ms
    Reply from 10.0.0.113: bytes=56 Sequence=4 ttl=255 time=1 ms
    Reply from 10.0.0.113: bytes=56 Sequence=5 ttl=255 time=1 ms

  --- 10.0.0.113 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

# Configure BGP for P1, use the IP address of Loopback0 as the router ID, and establish an EBGP peer relationship through directly connected interfaces.

```
[P1]bgp 65003
[P1-bgp] router-id 10.0.5.5
[P1-bgp] peer 10.0.0.14 as-number 65100
[P1-bgp] peer 10.0.0.114 as-number 65100
```

# Create VPN instance **vpna_in** on P2 to receive routes and **vpna_out** to send routes.

In the entire network, configure VPN instances on the PE devices in MPLS VPN according to the following planning.

**Table 4-2 VPN instance planning**

| Device ID | VPN Instance | RD | IRT | ERT |
|---|---|---|---|---|
| P2 | vpna_in | 10:60 | 10:46, 10:26 | - |
| P2 | vpna_out | 10:6060 | - | 10:624 |
| PE2 | vpna | 10:20 | 10:624 | 10:26 |
| PE4 | vpna | 10:40 | 10:624 | 10:46 |

```
[P2]ip vpn-instance vpna_in
[P2-vpn-instance-vpna_in] ipv4-family
[P2-vpn-instance-vpna_in-af-ipv4] route-distinguisher 10:60
[P2-vpn-instance-vpna_in-af-ipv4] vpn-target 10:46 import-extcommunity evpn
[P2-vpn-instance-vpna_in-af-ipv4] vpn-target 10:26 import-extcommunity evpn
[P2-vpn-instance-vpna_in-af-ipv4] evpn mpls routing-enable
[P2-vpn-instance-vpna_in-af-ipv4] quit
[P2-vpn-instance-vpna_in] quit
```

Add the **evpn** parameter to RT values, and enable EVPN to generate and advertise IP prefix and IRB routes.

```
[P2]ip vpn-instance vpna_out
[P2-vpn-instance-vpna_out] ipv4-family
[P2-vpn-instance-vpna_out-af-ipv4] route-distinguisher 10:6060
```

```
[P2-vpn-instance-vpna_out-af-ipv4] vpn-target 10:624 export-extcommunity evpn
[P2-vpn-instance-vpna_out-af-ipv4] evpn mpls routing-enable
[P2-vpn-instance-vpna_out-af-ipv4] quit
```

# After configuring the VPN instances, add the interfaces into corresponding VPN instances.

```
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ip binding vpn-instance vpna_out
[P2-GigabitEthernet0/5/0] ip address 10.0.0.14 255.255.255.252
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/0.1
[P2-GigabitEthernet0/5/0.1] vlan-type dot1q 10
[P2-GigabitEthernet0/5/0.1] ip binding vpn-instance vpna_in
[P2-GigabitEthernet0/5/0.1] ip address 10.0.0.114 255.255.255.252
```

Reconfigure IP addresses for the interfaces.

# Configure EBGP on P2.

```
[P2]bgp 65100
[P2-bgp] ipv4-family vpn-instance vpna_in
[P2-bgp-vpna_in] peer 10.0.0.113 as-number 65003
[P2-bgp-vpna_in] peer 10.0.0.113 substitute-as
[P2-bgp-vpna_in] advertise l2vpn evpn
[P2-bgp-vpna_in] ipv4-family vpn-instance vpna_out
[P2-bgp-vpna_out] peer 10.0.0.13 as-number 65003
[P2-bgp-vpna_out] peer 10.0.0.13 allow-as-loop 2
[P2-bgp-vpna_out] advertise l2vpn evpn
```

The AS_Path attribute of the BGP VPNv4 routes received in the VPN instance **vpna_in** carries AS 65003. P1 discards the routes due to loop prevention mechanism. Therefore, you need to configure the **substitute-as** parameter to substitute the AS.

The branch routes learned in **vpna_in** are advertised to **vpna_out** in the form of BGP routes through the CE (P1) of the HQ. However, P2 does not learn the routes because they carry the local AS ID. Therefore, the **allow-as-loop** parameter must be configured.

The **advertise l2vpn evpn** command must be configured in VPN instances to configure IP prefix route advertisement.

# Check the BGP peer relationship status on P1.

```
[P1]display bgp peer
BGP local router ID : 10.0.5.5
Local AS number : 65003
Total number of peers : 2                 Peers in established state : 2

 Peer            V     AS  MsgRcvd  MsgSent  OutQ  Up/Down       State      PrefRcv
 10.0.0.14       4  65100     1258     1263     0  18:13:13  Established        1
 10.0.0.114      4  65100     1290     1298     0  18:38:11  Established        1
```

The peer relationship has been normally established.

# Create Loopback1 on P1 to simulate a service network segment.

```
[P1]interface LoopBack1
[P1-LoopBack1] ip address 10.1.5.5 32
```

Configure its IP address to 10.1.5.5/32.

# Advertise the route into BGP on P1.

```
[P1] bgp 65003
[P1-bgp] network 10.1.5.5 255.255.255.255
```

Step 6 Complete the configuration between CEs and PEs in Branch2.

Configure a VPN instance on PE4 according to the planning, and add an interface to the VPN instance. Configure EBGP between PE4 and PE3. Create Loopback1 interface on PE3 to simulate a service network segment, and import it into BGP.

#Create a VPN instance on PE4.

```
[PE4]ip vpn-instance vpna
[PE4-vpn-instance-vpna] ipv4-family
[PE4-vpn-instance-vpna-af-ipv4] route-distinguisher 10:40
[PE4-vpn-instance-vpna-af-ipv4] vpn-target 10:46 export-extcommunity evpn
[PE4-vpn-instance-vpna-af-ipv4] vpn-target 10:624 import-extcommunity evpn
[PE4-vpn-instance-vpna-af-ipv4] evpn mpls routing-enable
[PE4-vpn-instance-vpna-af-ipv4] quit
[PE4-vpn-instance-vpna] quit
```

# Add the interface that connects the CEs to the VPN instance.

```
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] ip binding vpn-instance vpna
[PE4-GigabitEthernet0/5/0] ip address 10.0.0.26 255.255.255.252
```

# Complete EBGP configuration on PE4 and establish a BGP peer relationship through directly connected interfaces.

```
[PE4]bgp 65100
[PE4-bgp] ipv4-family vpn-instance vpna
[PE4-bgp-vpna] peer 10.0.0.25 as-number 65003
[PE4-bgp-vpna] peer 10.0.0.25 substitute-as
[PE4-bgp-vpna] advertise l2vpn evpn
```

Configure the **substitute-as** parameter.

# Configure EBGP on PE3 and use the IP address of Loopback0 as the router ID.

```
[PE3]bgp 65003
[PE3-bgp] router-id 10.0.3.3
[PE3-bgp] peer 10.0.0.26 as-number 65100
```

# Check the EBGP peer relationship status on PE4.

```
[PE4]display bgp vpnv4 vpn-instance vpna peer
```

```
BGP local router ID : 10.0.4.4
Local AS number : 65100

VPN-Instance vpna, Router ID 10.0.4.4:
Total number of peers : 1                 Peers in established state : 1

 Peer                    V          S       MsgRcvd   MsgSent   OutQ  Up/Down       State       PrefRcv
 10.0.0.25               4          65003   1290      1298      0     18:42:33      Established 1
```

The peer relationship has been normally established.

# Create Loopback1 interface on PE3 and advertise it into BGP.

```
[PE3]interface LoopBack1
[PE3-LoopBack1] ip address 10.1.3.3 32
[PE3-LoopBack1] quit
[PE3]bgp 65003
[PE3-bgp] network 10.1.3.3 255.255.255.255
```

Step 7 Complete the configuration between CEs and PEs in Branch1.

Configure a VPN instance on PE2 according to the planning, and add an interface to the VPN instance. Configure EBGP between PE2 and PE1. Create Loopback1 interface on PE1 to simulate a service network segment, and import it into OSPF.

# Create a VPN instance on PE2.

```
[PE2]ip vpn-instance vpna
[PE2-vpn-instance-vpna] ipv4-family
[PE2-vpn-instance-vpna-af-ipv4] route-distinguisher 10:20
[PE2-vpn-instance-vpna-af-ipv4] apply-label per-instance
[PE2-vpn-instance-vpna-af-ipv4] vpn-target 10:26 export-extcommunity evpn
[PE2-vpn-instance-vpna-af-ipv4] vpn-target 10:624 import-extcommunity evpn
[PE2-vpn-instance-vpna-af-ipv4] evpn mpls routing-enable
```

# Add PE2 interfaces to the VPN instance.

```
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ip binding vpn-instance vpna
[PE2-GigabitEthernet0/5/0] ip address 10.0.0.2 255.255.255.252
```

Reconfigure IP addresses for the interfaces.

# Configure OSPF on PE2.

```
[PE2]ospf 1 router-id 10.0.2.2 vpn-instance vpna
[PE2-ospf-1] import-route bgp cost 20 type 2
[PE2-ospf-1] area 0.0.0.0
```

Import BGP routes.

# Enable OSPF on the interconnection interfaces.

```
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ospf enable 1 area 0.0.0.0
```

# Configure OSPF on PE1.

```
[PE1]ospf 1 router-id 10.0.1.1
[PE1-ospf-1] area 0.0.0.0
```

# Enable OSPF on the interconnection interfaces.

```
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0] ospf enable 1 area 0.0.0.0
```

# Create Loopback1 interface on PE1 and enable OSPF on the interface.

```
[PE1]interface LoopBack1
[PE1-LoopBack1] ip address 10.1.1.1 32
[PE1-LoopBack1] ospf enable 1 area 0.0.0.0
```

# Check the OSPF neighbor relationship status on PE2.

```
[PE2]display ospf peer brief
(M) Indicates MADJ neighbor


             OSPF Process 1 with Router ID 10.0.2.2
                     Peer Statistic Information
Total number of peer(s): 1
 Peer(s) in full state: 1
--------------------------------------------------------------------------------
 Area Id          Interface                Neighbor id      State
 0.0.0.0          GE0/5/0                  10.0.1.1           Full
--------------------------------------------------------------------------------
```

The neighbor relationship status is normal.

# Restrict OSPF routes imported into BGP by configuring IP prefix list and route-policy.

```
[PE2]ip ip-prefix Loopback1 index 10 permit 10.1.1.1 32
[PE2]route-policy O2B permit node 10
[PE2-route-policy] if-match ip-prefix Loopback1
```

# Import OSPF routes in the BGP VPN instance view.

```
[PE2]bgp 65100
[PE2-bgp] ipv4-family vpn-instance vpna
[PE2-bgp-vpna] import-route ospf 1 route-policy O2B
[PE2-bgp-vpna] advertise l2vpn evpn
```

## Step 8 Verify routes.

Check the BGP EVPN routing tables of the branches and the HQ. Check EVPN and CE-side route learning results.

# Check the EVPN routing table on P2.

```
[P2]display bgp evpn all routing-table
```

```
Local AS number : 65100

 BGP Local router ID is 10.0.6.6
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,   i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete


 EVPN address family:
 Number of Ip Prefix Routes: 6
 Route Distinguisher: 10:20
      Network(EthTagId/IpPrefix/IpPrefixLen)          NextHop
 *>i  0:10.1.1.1:32                                    10.0.2.2
 Route Distinguisher: 10:40
      Network(EthTagId/IpPrefix/IpPrefixLen)          NextHop
 *>i  0:10.1.3.3:32                                    10.0.4.4
 Route Distinguisher: 10:60
      Network(EthTagId/IpPrefix/IpPrefixLen)          NextHop
 *>   0:10.1.5.5:32                                    10.0.0.113
 Route Distinguisher: 10:6060
      Network(EthTagId/IpPrefix/IpPrefixLen)          NextHop
 *>   0:10.1.1.1:32                                    10.0.0.13
 *>   0:10.1.3.3:32                                    10.0.0.13
 *>   0:10.1.5.5:32                                    10.0.0.13
```

The EVPN Type 5 routes are displayed.

According to RD values, the service routes created on P1, PE1 and PE3 have been learned by P2. The service routes that are learned by P1 and then transmitted to P2 are learned in VPN instance **vpna_out** and the next hop of these routes is 10.0.0.13.

# Check the BGP routing table of VPN instance **vpna_in** on P2.

```
[P2]display bgp vpnv4 vpn-instance vpna_in routing-table
BGP Local router ID is 10.0.6.6
Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V - valid, I - invalid, N - not-found


VPN-Instance vpna_in, Router ID 10.0.6.6:

Total Number of Routes: 3
     Network            NextHop          MED        LocPrf    PrefVal Path/Ogn

*>i  10.1.1.1/32        10.0.2.2          2          100       0       ?
*>i  10.1.3.3/32        10.0.4.4          0          100       0       65003i
*>   10.1.5.5/32        10.0.0.113        0                    0       65003i
```

The service routes learned by VPN instance **vpna_in** are created by different PEs.

# Check the BGP routing table of VPN instance **vpna_out** on P2.

```
[P2]display bgp vpnv4 vpn-instance vpna_in routing-table
```

```
BGP Local router ID is 10.0.6.6
Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
              h - history, i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V - valid, I - invalid, N - not-found


VPN-Instance vpna_in, Router ID 10.0.6.6:

Total Number of Routes: 3
     Network            NextHop          MED        LocPrf    PrefVal Path/Ogn

*>i  10.1.1.1/32        10.0.0.13                   0         65003 65100?
*>i  10.1.3.3/32        10.0.0.13                   0         65003 65100 65100i
*>   10.1.5.5/32        10.0.0.13        0          0         65003i
```

All the service routes from P1 are learned by VPN instance **vpna_in** and the next hop is P1.

# Check the EVPN routing table on PE2.

```
[PE2]display bgp evpn all routing-table
 Local AS number : 65100

 BGP Local router ID is 10.0.2.2
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete


 EVPN address family:
 Number of Ip Prefix Routes: 4
 Route Distinguisher: 10:20
      Network(EthTagId/IpPrefix/IpPrefixLen)          NextHop
 *>   0:10.1.1.1:32                                     0.0.0.0
 Route Distinguisher: 10:6060
      Network(EthTagId/IpPrefix/IpPrefixLen)          NextHop
 *>i  0:10.1.1.1:32                                     10.0.6.6
 *>i  0:10.1.3.3:32                                     10.0.6.6
 *>i  0:10.1.5.5:32                                     10.0.6.6
```

The next hop of all EVPN Type 5 routes is P2, indicating that the routes go to the other branch through the HQ.

# Check the BGP routing table of VPN instance **vpna** on PE2.

```
[PE2]display bgp vpnv4 vpn-instance vpna routing-table

 BGP Local router ID is 10.0.2.2
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found
```

```
 VPN-Instance vpna, Router ID 10.0.2.2:

 Total Number of Routes: 4
      Network         NextHop              MED       LocPrf    PrefValPath/Ogn

*>i   10.1.1.1/32     0.0.0.0                2                   0     ?
* i                   10.0.6.6                         100      0     65003 65100?
*>i   10.1.3.3/32     10.0.6.6                         100      0     65003 65100 65100i
*>    10.1.5.5/32     10.0.6.6               0         100      0     65003i
```

All the service routes are learned by VPN instance **vpna**

# Check the OSPF routing table on PE1.

```
[PE1]display ospf routing
            OSPF Process 1 with Router ID 10.0.1.1
                    Routing Tables

 Routing for Network
 Destination          Cost      Type        NextHop         AdvRouter        Area
 10.0.0.0/30          1         Direct      10.0.0.1        10.0.1.1         0.0.0.0
 10.1.1.1/32          0         Direct      10.1.1.1        10.0.1.1         0.0.0.0

 Routing for ASEs
 Destination          Cost        Type        Tag             NextHop         AdvRouter
 10.1.3.3/32          20          Type2       3489726028      10.0.0.2        10.0.2.2
 10.1.5.5/32          20          Type2       3489726028      10.0.0.2        10.0.2.2

 Total Nets: 4
 Intra Area: 2   Inter Area: 0   ASE: 2   NSSA: 0
```

The OSPF routing table on PE1 has contained the routes to the HQ and the branches.

# Check the EVPN routing table on PE4.

```
[PE4]display bgp evpn all routing-table
 Local AS number : 65100

 BGP Local router ID is 10.0.4.4
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,   i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete


 EVPN address family:
 Number of Ip Prefix Routes: 4
 Route Distinguisher: 10:40
        Network(EthTagId/IpPrefix/IpPrefixLen)          NextHop
 *>     0:10.1.3.3:32                                    10.0.0.25
 Route Distinguisher: 10:6060
        Network(EthTagId/IpPrefix/IpPrefixLen)          NextHop
 *>i    0:10.1.1.1:32                                    10.0.6.6
 *>i    0:10.1.3.3:32                                    10.0.6.6
 *>i    0:10.1.5.5:32                                    10.0.6.6
```

The next hop of all EVPN Type 5 routes is P2, indicating that the routes go to the other branch through the HQ.

# Check the BGP routing table of VPN instance **vpna** on PE4.

```
[PE4]display bgp vpnv4 vpn-instance vpna routing-table

 BGP Local router ID is 10.0.4.4
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found


 VPN-Instance vpna, Router ID 10.0.4.4:
 Total Number of Routes: 4
      Network         NextHop                MED       LocPrf    PrefValPath/Ogn

*>i   10.1.1.1/32     10.0.6.6                           100       0     65003 65100?
*>   10.1.3.3/32      10.0.0.25                0                   0     65003i
* i                   10.0.6.6                           100       0     65003 65100 65100i
*>i   10.1.5.5/32     10.0.6.6                 0         100       0     65003i
```

All the service routes are learned by VPN instance **vpna**

# Check the BGP routing table on PE3.

```
[PE3]display bgp routing-table

 BGP Local router ID is 10.0.3.3
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found
Total Number of Routes: 3
      Network         NextHop                MED       LocPrf    PrefValPath/Ogn

*>    10.1.1.1/32     10.0.0.26                                    0     65100 65100 65100?
*>    10.1.3.3/32     0.0.0.0                  0                   0      i
*>    10.1.5.5/32     10.0.0.26                                    0     65100 65100i
```

The routes to the HQ and the other branch have been normally learned.

# Perform a connectivity test.

```
[PE3]ping -a 10.1.3.3 10.1.1.1
  PING 10.1.1.1: 56  data bytes, press CTRL_C to break
    Reply from 10.1.1.1: bytes=56 Sequence=1 ttl=250 time=1 ms
    Reply from 10.1.1.1: bytes=56 Sequence=2 ttl=250 time=1 ms
    Reply from 10.1.1.1: bytes=56 Sequence=3 ttl=250 time=1 ms
    Reply from 10.1.1.1: bytes=56 Sequence=4 ttl=250 time=1 ms
    Reply from 10.1.1.1: bytes=56 Sequence=5 ttl=250 time=1 ms

  --- 10.1.1.1 ping statistics ---
    5 packet(s) transmitted
```

```
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

# Check packet forwarding paths.

```
[PE3]tracert -a 10.1.3.3 10.1.1.1
 traceroute to 10.1.1.1(10.1.1.1), max hops: 30, packet length: 40, press CTRL_C to break
 1 10.0.0.26 3 ms   2 ms   2 ms
 2 10.0.0.14 5 ms   3 ms   3 ms
 3 10.0.0.13 3 ms   2 ms   2 ms
 4 10.0.0.114 2 ms   2 ms   2 ms
 5 10.0.0.2 5 ms   3 ms   2 ms
 6 10.1.1.1 2 ms   2 ms   2 ms
```

Packets are forwarded from P2 to P1 and finally forwarded to PE1 through PE2.

By now, the configuration procedure of using EVPN as the control plane MPLS VPN is complete.

**----End**

## 4.1.3 Quiz

If EVPN is used to implement L2VPN like VPLS, which type of route is used to transmit routing information?

# 5 VXLAN Lab

## 5.1 Layer 2 Interconnection Through a Static VXLAN Tunnel

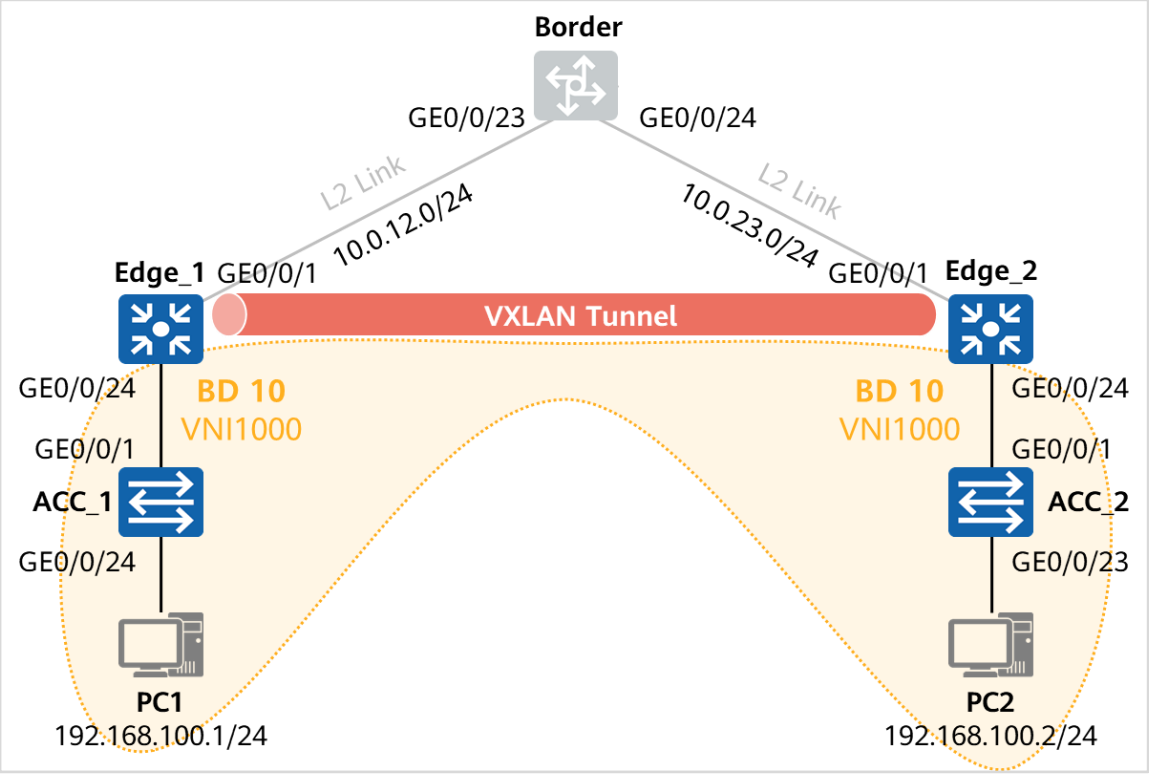
### 5.1.1 About This Lab

#### 5.1.1.1 Objectives

- Create a static VXLAN tunnel through the CLI.
- Implement Layer 2 interconnection through a VXLAN tunnel.
- Implement VXLAN service access through VLAN binding.

#### 5.1.1.2 Networking Description

**Figure 5-1 Implement Layer 2 interconnection through a static VXLAN tunnel**




The preceding figure shows the device interconnection. Loopback0 interfaces are created on Border, Edge_1, and Edge_2, and their IP addresses are in the format of 10.0.*x.x*. In the

format, *x* indicates the device ID and is marked in tables in the corresponding configuration procedures.

Configure the interfaces connecting Edge_1 and Border as trunk interfaces and allow packets from VLAN 12 to pass through. VLANIF 12 is created on Edge_1 and Border for Layer 3 interconnection, and the IP address is 10.0.12.*x*/24.

Configure the interfaces connecting Border and Edge_2 as trunk interfaces and allow packets from VLAN 23 to pass through. VLANIF 23 is created on Border and Edge_2 for Layer 3 interconnection, and the IP address is 10.0.23.*x*/24.

OSPF runs between Edge_1, Border, and Edge_2, the IP address of Loopback0 is used as the router ID, and OSPF is enabled on loopback and interconnection interfaces (the process ID is 1).

Edge_1 and Edge_2 use the address of Loopback0 as the source address of the VXLAN NVE interface to establish a static VXLAN tunnel.

ACC_1 and ACC_2 function as access switches and interconnect with PC1 and PC2, respectively, through access interfaces. Edge_1 and Edge_2 interconnect with ACC_1 and ACC_2, respectively, through Layer 2 sub-interfaces, and allow packets from VLAN 100 to pass through. Layer 3 sub-interfaces are bound to BD 10. This allows packets from terminals to be transmitted to a VXLAN network through Layer 2 sub-interfaces.

## 5.1.2 Lab Task

### 5.1.2.1 Configuration Roadmap

1. Complete the basic configuration of connectivity.
2. Configure service access points.
3. Configure a static VXLAN tunnel.

### 5.1.2.2 Configuration Procedure

Step 1 Complete the basic configuration of connectivity.

Configure interconnection interface IP addresses and OSPF on Edge_1, Border, and Edge_2 according to the following table.

**Table 5-1 Device ID**

| Device Name | Device ID (X) |
|---|---|
| Edge_1 | 1 |
| Border | 2 |
| Edge_2 | 3 |

# Name the devices.

N/A

# Configure a VLAN and interfaces on Edge_1.

```
[Edge_1] vlan 12
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_1]
[Edge_1] interface GigabitEthernet0/0/1
[Edge_1-GigabitEthernet0/0/1] port link-type trunk
[Edge_1-GigabitEthernet0/0/1] port trunk allow-pass vlan 12
[Edge_1-GigabitEthernet0/0/1] quit
[Edge_1]
[Edge_1] interface Vlanif12
[Edge_1-Vlanif12] ip address 10.0.12.1 255.255.255.0
[Edge_1-Vlanif12] quit
[Edge_1]
[Edge_1] interface LoopBack 0
[Edge_1-LoopBack0] ip address 10.0.1.1 32
[Edge_1-LoopBack0] quit
```

# Configure VLANs and interfaces on Border.

```
[Border] vlan batch 12 23
Info: This operation may take a few seconds. Please wait for a moment...done.
[Border]
[Border] interface GigabitEthernet0/0/23
[Border-GigabitEthernet0/0/23] port link-type trunk
[Border-GigabitEthernet0/0/23] port trunk allow-pass vlan 12
[Border-GigabitEthernet0/0/23] quit
[Border] interface GigabitEthernet0/0/24
[Border-GigabitEthernet0/0/24] port link-type trunk
[Border-GigabitEthernet0/0/24] port trunk allow-pass vlan 23
[Border-GigabitEthernet0/0/24] quit
[Border]
[Border] interface Vlanif12
[Border-Vlanif12] ip address 10.0.12.2 255.255.255.0
[Border-Vlanif12] quit
[Border] interface Vlanif23
[Border-Vlanif23] ip address 10.0.23.2 255.255.255.0
[Border-Vlanif23] quit
[Border]
[Border] interface LoopBack 0
[Border-LoopBack0] ip address 10.0.2.2 32
[Border-LoopBack0] quit
```

# Configure a VLAN and interfaces on Edge_2.

```
[Edge_2] vlan 23
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_2]
[Edge_2] interface GigabitEthernet0/0/1
[Edge_2-GigabitEthernet0/0/1] port link-type trunk
[Edge_2-GigabitEthernet0/0/1] port trunk allow-pass vlan 23
[Edge_2-GigabitEthernet0/0/1] quit
[Edge_2]
[Edge_2] interface Vlanif 23
[Edge_2-Vlanif23] ip address 10.0.23.3 255.255.255.0
[Edge_2-Vlanif23] quit
[Edge_2]
```

```
[Edge_2] interface LoopBack 0
[Edge_2-LoopBack0] ip address 10.0.3.3 32
[Edge_2-LoopBack0] quit
```

# Test the connectivity of the interconnection interfaces.

```
[Border]ping 10.0.12.1
  PING 10.0.12.1: 56  data bytes, press CTRL_C to break
    Reply from 10.0.12.1: bytes=56 Sequence=1 ttl=254 time=1 ms
    Reply from 10.0.12.1: bytes=56 Sequence=2 ttl=254 time=1 ms
    Reply from 10.0.12.1: bytes=56 Sequence=3 ttl=254 time=1 ms
    Reply from 10.0.12.1: bytes=56 Sequence=4 ttl=254 time=1 ms
    Reply from 10.0.12.1: bytes=56 Sequence=5 ttl=254 time=1 ms

  --- 10.0.12.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

[Border]ping 10.0.23.3
  PING 10.0.23.3: 56  data bytes, press CTRL_C to break
    Request time out
    Reply from 10.0.23.3: bytes=56 Sequence=2 ttl=254 time=1 ms
    Reply from 10.0.23.3: bytes=56 Sequence=3 ttl=254 time=1 ms
    Reply from 10.0.23.3: bytes=56 Sequence=4 ttl=254 time=1 ms
    Reply from 10.0.23.3: bytes=56 Sequence=5 ttl=254 time=1 ms

  --- 10.0.23.3 ping statistics ---
    5 packet(s) transmitted
    4 packet(s) received
    20.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

Test the connectivity of VLANIF interfaces to Edge_1 and Edge_2 on Border.

# Configure OSPF on Edge_1.

```
[Edge_1]ospf 1 router-id 10.0.1.1
[Edge_1-ospf-1] area 0.0.0.0
[Edge_1-ospf-1-area-0.0.0.0] network 10.0.1.1 0.0.0.0
[Edge_1-ospf-1-area-0.0.0.0] network 10.0.12.0 0.0.0.255
[Edge_1-ospf-1-area-0.0.0.0] quit
```

Use the IP address of Loopback0 as the router ID, and enable OSPF on Loopback0 and VLANIF 12.

# Configure OSPF on Border.

```
[Border]ospf 1 router-id 10.0.2.2
[Border-ospf-1] area 0.0.0.0
[Border-ospf-1-area-0.0.0.0] network 10.0.2.2 0.0.0.0
[Border-ospf-1-area-0.0.0.0] network 10.0.12.0 0.0.0.255
[Border-ospf-1-area-0.0.0.0] network 10.0.23.0 0.0.0.255
[Border-ospf-1-area-0.0.0.0] quit
```

Use the IP address of Loopback0 as the router ID, and enable OSPF on Loopback0, VLANIF 12, and VLANIF 23.

# Configure OSPF on Edge_2.

```
[Edge_2]ospf 1 router-id 10.0.3.3
[Edge_2-ospf-1] area 0.0.0.0
[Edge_2-ospf-1-area-0.0.0.0] network 10.0.3.3 0.0.0.0
[Edge_2-ospf-1-area-0.0.0.0] network 10.0.23.0 0.0.0.255
[Edge_2-ospf-1-area-0.0.0.0] quit
```

# Check the OSPF neighbor relationship and OSPF routing table on Border.

```
[Border]display ospf peer

            OSPF Process 1 with Router ID 10.0.2.2
                    Neighbors

 Area 0.0.0.0 interface   10.0.12.2(Vlanif12)'s neighbors
 Router ID: 10.0.1.1              Address: 10.0.12.1
   State: Full   Mode:Nbr is   Slave   Priority: 1
   R: 10.0.12.1   BDR: 10.0.12.2   MTU: 0
   Dead timer due in 36   sec
   Retrans timer interval: 5
   Neighbor is up for 00:01:57
   Authentication Sequence: [ 0 ]

                    Neighbors

 Area 0.0.0.0 interface 10.0.23.2(Vlanif23)'s neighbors
 Router ID: 10.0.3.3              Address: 10.0.23.3
   State: Full   Mode:Nbr is   Master   Priority: 1
   DR: 10.0.23.2   BDR: 10.0.23.3   MTU: 0
   Dead timer due in 28   sec
   Retrans timer interval: 5
   Neighbor is up for 00:01:24
   Authentication Sequence: [ 0 ]
```

The OSPF neighbor relationship has been established.

# Check the OSPF routing table on Border.

```
[Border]display ospf routing

       OSPF Process 1 withRouter ID 10.0.2.2
            Routing Tables

Routing forNetwork
Destination       Cost  Type       NextHop         AdvRouter       Area
10.0.2.2/32       0     Stub       10.0.2.2        10.0.2.2        0.0.0.0
10.0.12.0/24      1     Transit    10.0.12.2       10.0.2.2        0.0.0.0
10.0.23.0/24      1     Transit    10.0.23.2       10.0.2.2        0.0.0.0
10.0.1.1/32       1     Stub       10.0.12.1       10.0.1.1        0.0.0.0
10.0.3.3/32       1     Stub       10.0.23.3       10.0.3.3        0.0.0.0
```

```
TotalNets:5
Intra Area: 5  Inter Area: 0  ASE:0  NSSA: 0
```

Border has learned the routes generated by Loopback0 interfaces on Edge_1 and Edge_2.

# Test the connectivity of Loopback0 interfaces between Edge_1 and Edge_2.

```
[Edge_1]ping -a 10.0.1.1 10.0.3.3
  PING 10.0.3.3: 56  data bytes, press CTRL_C to break
    Reply from 10.0.3.3: bytes=56 Sequence=1 ttl=253 time=1 ms
    Reply from 10.0.3.3: bytes=56 Sequence=2 ttl=253 time=1 ms
    Reply from 10.0.3.3: bytes=56 Sequence=3 ttl=253 time=1 ms
    Reply from 10.0.3.3: bytes=56 Sequence=4 ttl=253 time=1 ms
    Reply from 10.0.3.3: bytes=56 Sequence=5 ttl=253 time=1 ms

  --- 10.0.3.3 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 1/1/1 ms
```

Edge_1 and Edge_2 can communicate with each other through their Loopback0 interfaces.

Step 2 Configure service access points.

Configure a service access point on Edge_1 and Edge_2, create a sub-interface to interconnect with ACC_1 and ACC_2 and terminate packets from VLAN 100, and bind the sub-interfaces to BD 10.

Configure the interfaces connecting ACC_1 and ACC_2 to PCs as access interfaces and set the PVID to 100. Configure the interfaces connecting ACC_1 and ACC_2 to Edge_1 and Edge_2 as trunk interfaces, and allow packets from VLAN 100 to pass through.

# Configure Edge_1.

```
[Edge_1] bridge-domain 10
[Edge_1-bd10] vxlan vni 1000
[Edge_1-bd10] quit
[Edge_1] vcmp role silent
[Edge_1] interface GigabitEthernet0/0/24
[Edge_1-GigabitEthernet0/0/24] port link-type trunk
Info: This operation may take a few seconds. Please wait for a moment...
[Edge_1-GigabitEthernet0/0/24] quit
[Edge_1] interface GigabitEthernet0/0/24.100 mode l2
[Edge_1-GigabitEthernet0/0/24.100] encapsulation dot1q vid 100
Info: This operation may take a few seconds. Please wait for a moment...
[Edge_1-GigabitEthernet0/0/24.100] bridge-domain 10
[Edge_1-GigabitEthernet0/0/24.100] quit
```

By default, after VCMP is configured, the encapsulation mode of packets allowed to pass a Layer 2 sub-interface cannot be set to Dot1q. In this case, you need to run the **vcmp role silent** command.

# Configure Edge_2.

```
[Edge_2] bridge-domain 10
[Edge_2-bd10] vxlan vni 1000
[Edge_2-bd10] quit
[Edge_2] vcmp role silent
[Edge_2] interface GigabitEthernet0/0/24
[Edge_2-GigabitEthernet0/0/24] port link-type trunk
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_2-GigabitEthernet0/0/24] quit
[Edge_2] interface GigabitEthernet0/0/24.100 mode l2
[Edge_2-GigabitEthernet0/0/24.100] encapsulation dot1q vid 100
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_2-GigabitEthernet0/0/24.100] bridge-domain 10
[Edge_2-GigabitEthernet0/0/24.100] quit
```

# Configure ACC_1.

```
[ACC_1] vlan 100
Info: This operation may take a few seconds. Please wait for a moment...done.
[ACC_1-vlan100] interface GigabitEthernet0/0/24
[ACC_1-GigabitEthernet0/0/24] port link-type access
[ACC_1-GigabitEthernet0/0/24] port default vlan 100
[ACC_1-GigabitEthernet0/0/24] quit
[ACC_1] interface GigabitEthernet0/0/1
[ACC_1-GigabitEthernet0/0/1] port link-type trunk
[ACC_1-GigabitEthernet0/0/1] port trunk allow-pass vlan 100
[ACC_1-GigabitEthernet0/0/1] quit
```

# Configure ACC_2.

```
[ACC_2] vlan 100
Info: This operation may take a few seconds. Please wait for a moment...done.
[ACC_2-vlan100]interface GigabitEthernet0/0/23
[ACC_2-GigabitEthernet0/0/23] port link-type access
Info: This operation may take a few seconds. Please wait for a moment...done.
[ACC_2-GigabitEthernet0/0/23] port default vlan 100
[ACC_2-GigabitEthernet0/0/23] quit
[ACC_2] interface GigabitEthernet0/0/1
[ACC_2-GigabitEthernet0/0/1] port link-type trunk
[ACC_2-GigabitEthernet0/0/1] port trunk allow-pass vlan 100
[ACC_2-GigabitEthernet0/0/1] quit
```

## Step 3 Configure a static VXLAN tunnel.

On Edge_1 and Edge_2, use the address of Loopback0 as the source address of the NVE interface to establish a static VXLAN tunnel for the interconnection between PC1 and PC2.

# Create an NVE interface on Edge_1.

```
[Edge_1]interface Nve1
[Edge_1-Nve1] source 10.0.1.1
[Edge_1-Nve1] vni 1000 head-end peer-list 10.0.3.3
[Edge_1-Nve1] quit
```

# Create an NVE interface on Edge_2.

```
[Edge_2]interface Nve1
[Edge_2-Nve1] source 10.0.3.3
[Edge_2-Nve1] vni 1000 head-end peer-list 10.0.1.1
[Edge_2-Nve1] quit
```

# Check the VXLAN tunnel on Edge_1 and Edge_2.

```
[Edge_1]display vxlan tunnel
 Tunnel ID         Source              Destination                 State       Type
 ------------------------------------------------------------------------------------------------------------------------
 4026531841        10.0.1.1            10.0.3.3                    up          static
 ------------------------------------------------------------------------------------------------------------------------
 Number of vxlan tunnel :
 Total: 1      Static: 1      L2 dynamic: 0      L3 dynamic: 0
```

```
[Edge_2]display vxlan tunnel
 Tunnel ID         Source              Destination                 State       Type
 ------------------------------------------------------------------------------------------------------------------------
4026531841         10.0.3.3            10.0.1.1                    up          static
 ------------------------------------------------------------------------------------------------------------------------
 Number of vxlan tunnel :
 Total: 1      Static: 1      L2 dynamic: 0      L3 dynamic: 0
```

Static VXLAN tunnels in Up state have been created on Edge_1 and Edge_2.

## Step 4 Verify configurations.

Access PC2 from PC1 and check VXLAN-related forwarding entries.

# Ping PC2 from PC1.

```
C:\Users\PC1>ping 192.168.100.2

Pinging 192.168.100.2 with 32 bytes of data:
Reply from 192.168.100.2: bytes=32 time=1ms TTL=128
Reply from 192.168.100.2: bytes=32 time<1ms TTL=128
Reply from 192.168.100.2: bytes=32 time<1ms TTL=128
Reply from 192.168.100.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.100.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

The command output shows that PC1 can communicate with PC2.

# Check the MAC address entries on Edge_1 and Edge_2.

```
[Edge_1]display mac-address
------------------------------------------------------------------------------
MAC Address    VLAN/VSI/BD          Learned-From      Type
```

```
--------------------------------------------------------------------------------
000c-292e-b4a3  -/-/10              10.0.3.3            dynamic
000c-29b3-efea  -/-/10              GE0/0/24.100        dynamic
9400-b049-9efd  12/-/-              GE0/0/1             dynamic
--------------------------------------------------------------------------------
Total items displayed = 3
```

On Edge_1, you can find that the MAC addresses of PC1 and PC2 have been learned from the sub-interface GE0/0/24.100 and the remote VTEP (10.0.3.3), respectively.

```
[Edge_2]display mac-address
--------------------------------------------------------------------------------
MAC Address    VLAN/VSI/BD          Learned-From      Type
--------------------------------------------------------------------------------
000c-292e-b4a3  -/-/10              GE0/0/24.100      dynamic
000c-29b3-efea  -/-/10              10.0.1.1          dynamic
9400-b049-9ef9  23/-/-              GE0/0/1           dynamic
--------------------------------------------------------------------------------
Total items displayed = 3
```

Similarly, on Edge_2, you can find the MAC addresses learned from the local sub-interface and remote VTEP (10.0.1.1).

# Create ACL 3000 on Border for matching VXLAN-encapsulated packets for communication between PC1 and PC2.

```
[Border]acl number 3000
[Border-acl-adv-3000] rule 1 permit udp source 10.0.1.1 0 destination 10.0.3.3 0 destination-port eq 4789
```

The ACL matches packets with the source IP address being 10.0.1.1, destination IP addresses being 10.0.3.3, protocol being UDP, and destination port number being 4789.

# Run the **capture-packet** command on Border to obtain the packets on GE0/0/24.

```
[Border]capture-packet acl 3000 interface GigabitEthernet 0/0/24 destination file vxlan.cap outbound packet-num 10
Info: Packet getting is configured. Saved to flash:/vxlan.cap.
```

# Ping PC2 from PC1 again.

```
C:\Users\PC1>ping 192.168.100.2

Pinging 192.168.100.2 with 32 bytes of data:
Reply from 192.168.100.2: bytes=32 time<1ms TTL=128
Reply from 192.168.100.2: bytes=32 time<1ms TTL=128
Reply from 192.168.100.2: bytes=32 time<1ms TTL=128
Reply from 192.168.100.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.100.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

# After the packets have been obtained, the command output similar to the following is displayed:

```
-----------------packet getting report-----------------
file: flash:/vxlan.cap
packets getting: interface GigabitEthernet0/0/24
ACL: 3000
vlan: -    cvlan: -
car: 64kbps timeout: 60s
packets: 10 (expected) 10 (actual)
length without tunnel header: 64 (expected)
-------------------------------------------------------
```

Information about the path of the obtained packets is displayed in the command output. You can download the file through FTP or SFTP. For details about how to enable FTP or SFTP on the device, see the related product documentation.

# Check the obtained packets on Border.

```
Ethernet II, Src: 94:00:b0:49:9e:f9 (94:00:b0:49:9e:f9), Dst: 80:e1:bf:0e:56:59 (80:e1:bf:0e:56:59)
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 23
Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.3.3
User Datagram Protocol, Src Port: 19456, Dst Port: 4789
Virtual eXtensible Local Area Network
    Flags: 0x0800, VXLAN Network ID (VNI)
    Group Policy ID: 0
    VXLAN Network Identifier (VNI): 1000
    Reserved: 0
Ethernet II, Src: 00:0c:29:b3:ef:ea, Dst: 00:0c:29:2e:b4:a3
Internet Protocol Version 4, Src: 192.168.100.1, Dst: 192.168.100.2
Internet Control Message Protocol
```

Obtain the packets on Border. You can find that, in the outer IP header, the source and destination IP addresses are respectively 10.0.1.1 and 10.0.3.3, which are the NVE interface IP addresses of Edge_1 and Edge_2, respectively.

The outer IP header is followed by the UDP header, in which the destination port number is 4789. This indicates that the subsequent packet is encapsulated using VXLAN.

In the VXLAN header, the VNI is 1000.

The VXLAN header is followed by the original data frame and IP header for communication between PC1 and PC2.

**----End**

## 5.1.3 Quiz

How will a VTEP process the broadcast packets in the local BD from hosts?

# 5.2 Centralized VXLAN Gateway

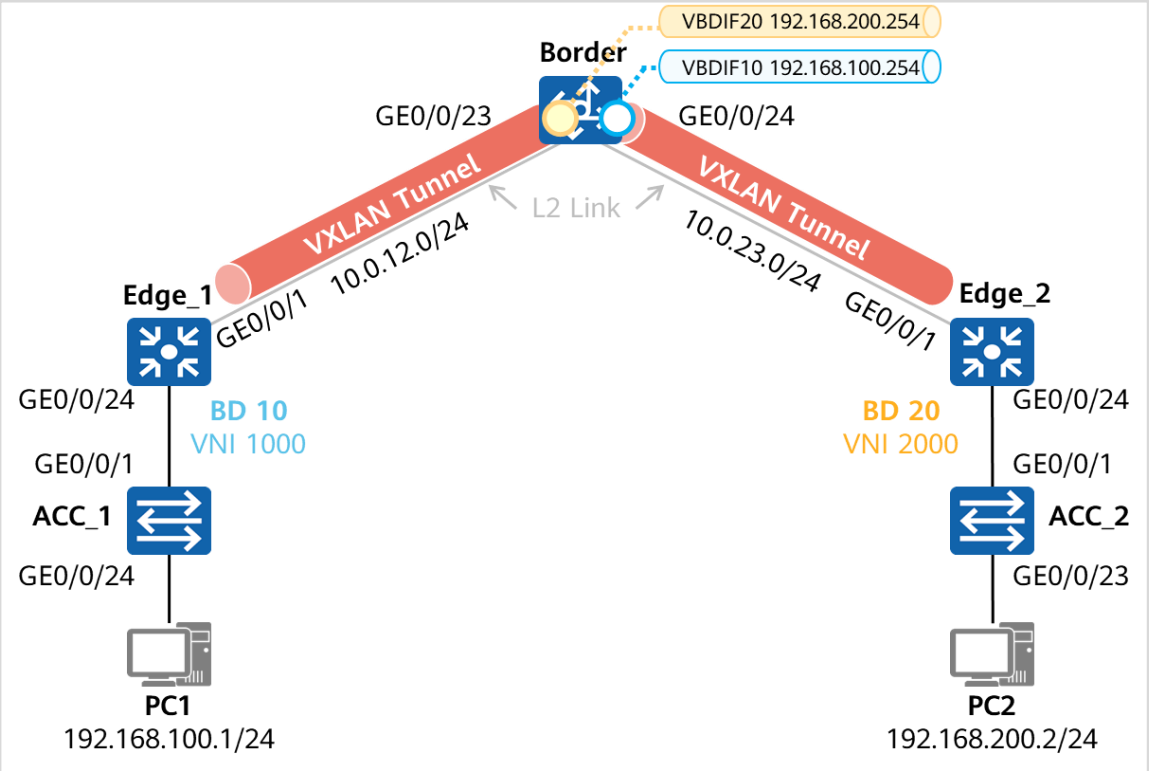
## 5.2.1 About This Lab

### 5.2.1.1 Objectives

- Configure a centralized VXLAN gateway to implement Layer 3 interconnection.
- Describe the packet forwarding process in the centralized VXLAN gateway scenario.

### 5.2.1.2 Networking Description

**Figure 5-2 Centralized VXLAN gateway topology**




The figure shows the device interconnection. Loopback0 interfaces are created on Border, Edge_1, and Edge_2, and their IP addresses are in the format of 10.0.*x.x*. In the format, *x* indicates the device ID and is marked in tables in the corresponding configuration procedures.

Configure the interfaces connecting Edge_1 and Border as trunk interfaces and allow packets from VLAN 12 to pass through. VLANIF 12 is created on Edge_1 and Border for Layer 3 interconnection, and the IP address is 10.0.12.*x*/24.

Configure the interfaces connecting Border and Edge_2 as trunk interfaces and allow packets from VLAN 23 to pass through. VLANIF 23 is created on Border and Edge_2 for Layer 3 interconnection, and the IP address is 10.0.23.*x*/24.

OSPF runs between Edge_1, Border, and Edge_2, the IP address of Loopback0 is used as the router ID, and OSPF is enabled on loopback and interconnection interfaces.

Edge_1, Border, and Edge_2 use the address of Loopback0 as the source address of the VXLAN NVE interface. A VXLAN tunnel is established between Edge_1 and Border to

transmit traffic of BD 10, and a VXLAN tunnel is established between Border and Edge_2 to transmit traffic of BD 20.

ACC_1 and ACC_2 function as access switches and interconnect with PC1 and PC2, respectively, through access interfaces. PC1 and PC2 are added to VLAN 100 and VLAN 200, respectively. Edge_1 and Border interconnect with ACC_1 and ACC_2 through Layer 2 sub-interfaces to terminate packets from VLAN 100 and VLAN 200 respectively. The sub-interfaces are associated with BD 10 and BD 20, respectively.

On Border, create VBDIF 10 and VBDIF 20 as gateways for terminals in BD 10 and BD 20 to implement Layer 3 interconnection between terminals in BD 10 and BD 20.

## 5.2.2 Configuration Procedure

### 5.2.2.1 Configuration Roadmap

1. Complete the basic configuration of connectivity.
2. Configure service access points.
3. Configure static VXLAN tunnels connecting Border to Edge_1 and Edge_2.
4. Create VBDIF interfaces.

### 5.2.2.2 Configuration Procedure

Step 1 Complete the basic configuration of connectivity.

Configure interconnection interface IP addresses and OSPF on Edge_1, Border, and Edge_2 according to the following table.

**Table 5-2 Device ID**

| Device Name | Device ID (X) |
|---|---|
| Edge_1 | 1 |
| Border | 2 |
| Edge_2 | 3 |

# Name the devices.

N/A

# Configure a VLAN and interfaces on Edge_1.

```
[Edge_1]vlan 12
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_1]
[Edge_1]interface GigabitEthernet0/0/1
[Edge_1-GigabitEthernet0/0/1] port link-type trunk
[Edge_1-GigabitEthernet0/0/1] port trunk allow-pass vlan 12
[Edge_1-GigabitEthernet0/0/1] quit
[Edge_1]
[Edge_1]interface Vlanif12
[Edge_1-Vlanif12] ip address 10.0.12.1 255.255.255.0
```

```
[Edge_1-Vlanif12] quit
[Edge_1]
[Edge_1]interface LoopBack 0
[Edge_1-LoopBack0] ip address 10.0.1.1 32
[Edge_1-LoopBack0] quit
```

# Configure VLANs and interfaces on Border.

```
[Border]vlan batch 12 23
Info: This operation may take a few seconds. Please wait for a moment...done.
[Border]
[Border]interface GigabitEthernet0/0/23
[Border-GigabitEthernet0/0/23] port link-type trunk
[Border-GigabitEthernet0/0/23] port trunk allow-pass vlan 12
[Border-GigabitEthernet0/0/23] quit
[Border]interface GigabitEthernet0/0/24
[Border-GigabitEthernet0/0/24] port link-type trunk
[Border-GigabitEthernet0/0/24] port trunk allow-pass vlan 23
[Border-GigabitEthernet0/0/24] quit
[Border]
[Border]interface Vlanif12
[Border-Vlanif12] ip address 10.0.12.2 255.255.255.0
[Border-Vlanif12] quit
[Border]interface Vlanif23
[Border-Vlanif23] ip address 10.0.23.2 255.255.255.0
[Border-Vlanif23] quit
[Border]
[Border]interface LoopBack 0
[Border-LoopBack0] ip address 10.0.2.2 32
[Border-LoopBack0] quit
```

# Configure a VLAN and interfaces on Edge_2.

```
[Edge_2]vlan 23
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_2]
[Edge_2]interface GigabitEthernet0/0/1
[Edge_2-GigabitEthernet0/0/1] port link-type trunk
[Edge_2-GigabitEthernet0/0/1] port trunk allow-pass vlan 23
[Edge_2-GigabitEthernet0/0/1] quit
[Edge_2]
[Edge_2]interface Vlanif 23
[Edge_2-Vlanif23] ip address 10.0.23.3 255.255.255.0
[Edge_2-Vlanif23] quit
[Edge_2]
[Edge_2]interface LoopBack 0
[Edge_2-LoopBack0] ip address 10.0.3.3 32
[Edge_2-LoopBack0] quit
```

# Test the connectivity of the interconnection interfaces.

```
[Border]ping 10.0.12.1
  PING 10.0.12.1: 56  data bytes, press CTRL_C to break
    Reply from 10.0.12.1: bytes=56 Sequence=1 ttl=254 time=1 ms
```

```
    Reply from 10.0.12.1: bytes=56 Sequence=2 ttl=254 time=1 ms
    Reply from 10.0.12.1: bytes=56 Sequence=3 ttl=254 time=1 ms
    Reply from 10.0.12.1: bytes=56 Sequence=4 ttl=254 time=1 ms
    Reply from 10.0.12.1: bytes=56 Sequence=5 ttl=254 time=1 ms

  --- 10.0.12.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

[Border]ping 10.0.23.3
  PING 10.0.23.3: 56  data bytes, press CTRL_C to break
    Request time out
    Reply from 10.0.23.3: bytes=56 Sequence=2 ttl=254 time=1 ms
    Reply from 10.0.23.3: bytes=56 Sequence=3 ttl=254 time=1 ms
    Reply from 10.0.23.3: bytes=56 Sequence=4 ttl=254 time=1 ms
    Reply from 10.0.23.3: bytes=56 Sequence=5 ttl=254 time=1 ms

  --- 10.0.23.3 ping statistics ---
    5 packet(s) transmitted
    4 packet(s) received
    20.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

Test the connectivity of VLANIF interfaces to Edge_1 and Edge_2 on Border.

# Configure OSPF on Edge_1.

```
[Edge_1]ospf 1 router-id 10.0.1.1
[Edge_1-ospf-1] area 0.0.0.0
[Edge_1-ospf-1-area-0.0.0.0] network 10.0.1.1 0.0.0.0
[Edge_1-ospf-1-area-0.0.0.0] network 10.0.12.0 0.0.0.255
[Edge_1-ospf-1-area-0.0.0.0] quit
```

Use the IP address of Loopback0 as the router ID, and enable OSPF on Loopback0 and VLANIF 12.

# Configure OSPF on Border.

```
[Border]ospf 1 router-id 10.0.2.2
[Border-ospf-1] area 0.0.0.0
[Border-ospf-1-area-0.0.0.0] network 10.0.2.2 0.0.0.0
[Border-ospf-1-area-0.0.0.0] network 10.0.12.0 0.0.0.255
[Border-ospf-1-area-0.0.0.0] network 10.0.23.0 0.0.0.255
[Border-ospf-1-area-0.0.0.0] quit
```

Use the IP address of Loopback0 as the router ID, and enable OSPF on Loopback0, VLANIF 12, and VLANIF 23.

# Configure OSPF on Edge_2.

```
[Edge_2]ospf 1 router-id 10.0.3.3
[Edge_2-ospf-1] area 0.0.0.0
[Edge_2-ospf-1-area-0.0.0.0] network 10.0.3.3 0.0.0.0
```

```
[Edge_2-ospf-1-area-0.0.0.0] network 10.0.23.0 0.0.0.255
[Edge_2-ospf-1-area-0.0.0.0] quit
```

# Check the OSPF neighbor relationship and OSPF routing table on Border.

```
[Border]display ospf peer

         OSPF Process 1 with Router ID 10.0.2.2
                 Neighbors

 Area 0.0.0.0 interface 10.0.12.2(Vlanif12)'s neighbors
 Router ID: 10.0.1.1             Address: 10.0.12.1
   State: Full  Mode:Nbr is  Slave  Priority: 1
   R: 10.0.12.1  BDR: 10.0.12.2  MTU: 0
   Dead timer due in 36  sec
   Retrans timer interval: 5
   Neighbor is up for 00:01:57
   Authentication Sequence: [ 0 ]

                 Neighbors

 Area 0.0.0.0 interface 10.0.23.2(Vlanif23)'s neighbors
 Router ID: 10.0.3.3             Address: 10.0.23.3
   State: Full  Mode:Nbr is  Master  Priority: 1
   DR: 10.0.23.2  BDR: 10.0.23.3  MTU: 0
   Dead timer due in 28  sec
   Retrans timer interval: 5
   Neighbor is up for 00:01:24
   Authentication Sequence: [ 0 ]
```

The OSPF neighbor relationship has been established.

# Check the OSPF routing table on Border.

```
[Border]display ospf routing

      OSPF Process 1 with Router ID 10.0.2.2
           Routing Tables

 Routing for Network
 Destination      Cost  Type       NextHop         AdvRouter       Area
 10.0.2.2/32      0     Stub       10.0.2.2        10.0.2.2        0.0.0.0
 10.0.12.0/24     1     Transit    10.0.12.2       10.0.2.2        0.0.0.0
 10.0.23.0/24     1     Transit    10.0.23.2       10.0.2.2        0.0.0.0
 10.0.1.1/32      1     Stub       10.0.12.1       10.0.1.1        0.0.0.0
 10.0.3.3/32      1     Stub       10.0.23.3       10.0.3.3        0.0.0.0

 Total Nets: 5
 Intra Area: 5  Inter Area: 0  ASE: 0  NSSA: 0
```

Border has learned the routes generated by Loopback0 interfaces on Edge_1 and Edge_2.

# Test the connectivity of Loopback0 interfaces between Edge_1 and Edge_2.

```
[Edge_1]ping -a 10.0.1.1 10.0.3.3
```

```
  PING 10.0.3.3: 56  data bytes, press CTRL_C to break
    Reply from 10.0.3.3: bytes=56 Sequence=1 ttl=253 time=1 ms
    Reply from 10.0.3.3: bytes=56 Sequence=2 ttl=253 time=1 ms
    Reply from 10.0.3.3: bytes=56 Sequence=3 ttl=253 time=1 ms
    Reply from 10.0.3.3: bytes=56 Sequence=4 ttl=253 time=1 ms
    Reply from 10.0.3.3: bytes=56 Sequence=5 ttl=253 time=1 ms

  --- 10.0.3.3 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 1/1/1 ms
```

Edge_1 and Edge_2 can communicate with each other through their Loopback0 interfaces.

Step 2 Configure service access points.

Configure a service access point on Edge_1 and Edge_2, create sub-interfaces to interconnect with ACC_1 and ACC_2 and terminate packets from VLAN 100 and VLAN 200, and bind the sub-interfaces to BD 10 and BD 20.

Configure the interfaces connecting ACC_1 and ACC_2 to PCs as access interfaces and set PVIDs to 100 and 200. Configure the interfaces connecting ACC_1 and ACC_2 to Edge_1 and Edge_2 as trunk interfaces, and allow packets from VLAN 100 and VLAN 200 to pass through.

# Configure Edge_1.

```
[Edge_1]bridge-domain 10
[Edge_1-bd10] vxlan vni 1000
[Edge_1-bd10] quit
[Edge_1]vcmp role silent
[Edge_1]interface GigabitEthernet0/0/24
[Edge_1-GigabitEthernet0/0/24]port link-type trunk
Info: This operation may take a few seconds. Please wait for a moment...
[Edge_1-GigabitEthernet0/0/24]quit
[Edge_1]interface GigabitEthernet0/0/24.100 mode l2
[Edge_1-GigabitEthernet0/0/24.100] encapsulation dot1q vid 100
Info: This operation may take a few seconds. Please wait for a moment...
[Edge_1-GigabitEthernet0/0/24.100] bridge-domain 10
[Edge_1-GigabitEthernet0/0/24.100] quit
```

By default, after VCMP is configured, the encapsulation mode of packets allowed to pass a Layer 2 sub-interface cannot be set to Dot1q. In this case, you need to run the **vcmp role silent** command.

# Configure Edge_2.

```
[Edge_2]bridge-domain 20
[Edge_2-bd20] vxlan vni 2000
[Edge_2-bd20] quit
[Edge_2]vcmp role silent
[Edge_2]interface GigabitEthernet0/0/24
[Edge_2-GigabitEthernet0/0/24]port link-type trunk
```

```
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_2-GigabitEthernet0/0/24]quit
[Edge_2]interface GigabitEthernet0/0/24.200 mode l2
[Edge_2-GigabitEthernet0/0/24.200] encapsulation dot1q vid 200
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_2-GigabitEthernet0/0/24.200] bridge-domain 20
[Edge_2-GigabitEthernet0/0/24.200] quit
```

# Configure ACC_1.

```
[ACC_1]vlan 100
Info: This operation may take a few seconds. Please wait for a moment...done.
[ACC_1-vlan100]interface GigabitEthernet0/0/24
[ACC_1-GigabitEthernet0/0/24] port link-type access
[ACC_1-GigabitEthernet0/0/24] port default vlan 100
[ACC_1-GigabitEthernet0/0/24] quit
[ACC_1]interface GigabitEthernet0/0/1
[ACC_1-GigabitEthernet0/0/1] port link-type trunk
[ACC_1-GigabitEthernet0/0/1] port trunk allow-pass vlan 100
[ACC_1-GigabitEthernet0/0/1] quit
```

# Configure ACC_2.

```
[ACC_2]vlan 200
Info: This operation may take a few seconds. Please wait for a moment...d
[ACC_2]interface GigabitEthernet0/0/23
[ACC_2-GigabitEthernet0/0/23] port link-type access
[ACC_2-GigabitEthernet0/0/23] port default vlan 200
[ACC_2-GigabitEthernet0/0/23] quit
[ACC_2]interface GigabitEthernet0/0/1
[ACC_2-GigabitEthernet0/0/1] port link-type trunk
[ACC_2-GigabitEthernet0/0/1] port trunk allow-pass vlan 200
[ACC_2-GigabitEthernet0/0/1] quit
```

Step 3 Configure static VXLAN tunnels and VBDIF interfaces.

Edge_1, Border, and Edge_2 use the address of Loopback0 as the source address of the NVE interface to establish VXLAN tunnels. A VXLAN tunnel is established between Edge_1 and Border to transmit traffic of BD 10, and a VXLAN tunnel is established between Border and Edge_2 to transmit traffic of BD 20.

On Border, create VBDIF 10 and VBDIF 20 as the gateways for PC1 and PC2.

# Configure NVE interfaces on Edge_1 and Border to establish a static VXLAN tunnel.

```
[Edge_1] interface Nve1
[Edge_1-Nve1] source 10.0.1.1
[Edge_1-Nve1] vni 1000 head-end peer-list 10.0.2.2
[Edge_1-Nve1] quit
```

Create BD 10 on Border.

```
[Border]bridge-domain 10
[Border-bd10] vxlan vni 1000
```

```
[Border-bd10] quit
[Border]interface Nve1
[Border-Nve1] source 10.0.2.2
[Border-Nve1] vni 1000 head-end peer-list 10.0.1.1
[Border-Nve1] quit
```

# Configure NVE interfaces on Edge_2 and Border to establish a static VXLAN tunnel.

```
[Edge_2]interface Nve1
[Edge_2-Nve1] source 10.0.3.3
[Edge_2-Nve1] vni 2000 head-end peer-list 10.0.2.2
[Edge_2-Nve1] quit
```

Create BD 20 on Border.

```
[Border] bridge-domain 20
[Border-bd20] vxlan vni 2000
[Border-bd20] quit
[Border] interface Nve 1
[Border-Nve1] vni 2000 head-end peer-list 10.0.3.3
[Border-Nve1] quit
```

# Check the status of VXLAN tunnels on Edge_1, Edge_2, and Border.

```
[Edge_1]display vxlan tunnel
 Tunnel ID         Source     Destination   State       Type
 ------------------------------------------------------------------------------------------------------------------------
 4026531842        10.0.1.1   10.0.2.2      up            static
 ------------------------------------------------------------------------------------------------------------------------
 Number of vxlan tunnel :
 Total: 1      Static: 1      L2 dynamic: 0      L3 dynamic: 0

 [Edge_2]display vxlan tunnel
 Tunnel ID         Source     Destination   State       Type
 ------------------------------------------------------------------------------------------------------------------------
 4026531842        10.0.3.3   10.0.2.2        up          static
 ------------------------------------------------------------------------------------------------------------------------
 Number of vxlan tunnel :
 Total: 1      Static: 1      L2 dynamic: 0      L3 dynamic: 0

 [Border]display vxlan tunnel
 Tunnel ID         Source     Destination   State       Type
 ------------------------------------------------------------------------------------------------------------------------
 4026531841        10.0.2.2   10.0.1.1        up          static
 4026531842        10.0.2.2   10.0.3.3        up          static
 ------------------------------------------------------------------------------------------------------------------------
 Number of vxlan tunnel :
 Total: 2      Static: 2      L2 dynamic: 0      L3 dynamic: 0
```

The VXLAN tunnels are in Up state on the three devices.

# Create VBDIF 10 and VBDIF 20 on Border and configure their IP addresses according to the planning.

```
[Border]interface Vbdif10
[Border-Vbdif10] ip address 192.168.100.254 255.255.255.0
[Border-Vbdif10] quit
[Border]interface Vbdif20
[Border-Vbdif20] ip address 192.168.200.254 255.255.255.0
[Border-Vbdif20] quit
```

Step 4 Verify configurations.

Ping PC2 from PC1. Check and analyze the entire forwarding process.

# Test the connectivity of PC1 and PC2 to gateway addresses.

```
C:\Users\PC1>ping 192.168.100.254

Pinging 192.168.100.254 with 32 bytes of data:
Reply from 192.168.100.254: bytes=32 time<1ms TTL=254
Reply from 192.168.100.254: bytes=32 time<1ms TTL=254
Reply from 192.168.100.254: bytes=32 time<1ms TTL=254
Reply from 192.168.100.254: bytes=32 time<1ms TTL=254

Ping statistics for 192.168.100.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\Users\PC2>ping 192.168.200.254

Pinging 192.168.200.254 with 32 bytes of data:
Reply from 192.168.200.254: bytes=32 time<1ms TTL=254
Reply from 192.168.200.254: bytes=32 time<1ms TTL=254
Reply from 192.168.200.254: bytes=32 time<1ms TTL=254
Reply from 192.168.200.254: bytes=32 time<1ms TTL=254

Ping statistics for 192.168.200.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

# Check the MAC addresses of VBDIF 10 and VBDIF 20 on Border.

```
[Border]display interface Vbdif 10 | in Hardware
IP Sending Frames' Format is PKTFMT_ETHNT_2, Hardware address is 9400-b049-9efb
[Border]display interface Vbdif 20 | in Hardware
IP Sending Frames' Format is PKTFMT_ETHNT_2, Hardware address is 9400-b049-9ef6
```

Similar to VLANIF interfaces, VBDIF 10 and VBDIF 20 have the same MAC address, which is 9400-b049-9ef6.

# Check the MAC address entries in BD 10 on Edge_1.

```
[Edge_1]display mac-address bridge-domain 10
--------------------------------------------------------------------------------
```

```
MAC Address      VLAN/VSI/BD     Learned-From          Type
-------------------------------------------------------------------------------
000c-29b3-efea     -/-/10          GE0/0/24.100          dynamic
9400-b049-9efb     -/-/10          10.0.2.2              dynamic
-------------------------------------------------------------------------------
Total items displayed = 2
```

There are two MAC address entries in BD 10 on Edge_1, which are learned respectively from the sub-interface GE0/0/24.100 and the remote VTEP (10.0.2.2). The learned two MAC addresses are the MAC address of the NIC on PC1 and the MAC address of VBDIF 10 on Border.

# Obtain the VXLAN packets received by Border from GE0/0/23.

```
[Border]capture-packet interface GigabitEthernet 0/0/23 destination file VXLAN.cap inbound packet-num 50
```

In this case, Border is not the intermediate device for forwarding VXLAN packets but the termination device for VXLAN packets. Therefore, if you use filter conditions to obtain packets, no packet will be matched.

# Ping PC2 from PC1.

```
C:\Users\PC1>ping 192.168.200.2

Pinging 192.168.200.2 with 32 bytes of data:
Reply from 192.168.200.2: bytes=32 time<1ms TTL=127
Reply from 192.168.200.2: bytes=32 time<1ms TTL=127
Reply from 192.168.200.2: bytes=32 time<1ms TTL=127
Reply from 192.168.200.2: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.200.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

The command output shows that the communication is normal.

# After the packets have been obtained, the command output similar to the following is displayed:

```
  -----------------packet getting report-----------------
  file: flash:/VXLAN.cap
  packets getting: interface GigabitEthernet0/0/24
  ACL: 3000
  vlan: -    cvlan: -
  car: 64kbps timeout: 60s
  packets: 10 (expected) 10 (actual)
  length without tunnel header: 64 (expected)
  -------------------------------------------------------
```

Information about the path of the obtained packets is displayed in the command output. You can download the file through FTP or SFTP. For details about how to enable FTP or SFTP on the device, see the related product documentation.

# Check the obtained packets on Border.

```
Ethernet II, Src: d4:46:49:82:34:8d (d4:46:49:82:34:8d), Dst: 94:00:b0:49:9e:fd (94:00:b0:49:9e:fd)
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 12
Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.2.2
User Datagram Protocol, Src Port: 256, Dst Port: 4789
Virtual eXtensible Local Area Network
    Flags: 0x0800, VXLAN Network ID (VNI)
    Group Policy ID: 0
    VXLAN Network Identifier (VNI): 1000
    Reserved: 0
Ethernet II, Src: 00:0c:29:b3:ef:ea, Dst: 94:00:b0:49:9e:fb
Internet Protocol Version 4, Src: 192.168.100.1, Dst: 192.168.200.2
Internet Control Message Protocol
```

Filter and check the VXLAN packets manually.

This packet encapsulates the original data frame for PC1 to ping PC2. Encapsulate the VXLAN header (the VNI is 1000) on Edge_1. Use the source address (10.0.1.1) of interface NVE1 on Edge_1 as the source IP address of the IP header and use the source address of interface NVE1 on Border as the destination address (10.0.2.2) of the IP header.

After receiving the data frame, Border checks the destination MAC address (94:00:b0:49:9e:fd, which is the MAC address of VLANIF 12) of the data frame and determines that the data frame needs to be terminated locally.

After decapsulating the data frame, Border finds that the destination IP address is 10.0.2.2, which is the IP address of the local Loopback0. Next, Border further decapsulates the data frame to check the upper-layer data and finds that the packet is encapsulated using VXLAN. After that, Border checks the VNI in the VXLAN header and determines that the inner data frame needs to be searched and forwarded in BD 10 based on VNI 1000.

Subsequently, Border checks the destination MAC address (94:00:b0:49:9e:fb) of the inner data frame in BD 10. The destination MAC address is the MAC address of VBDIF 10. Therefore, Border determines that the data frame needs to be processed by VBDIF 10. Border then further decapsulates the data frame to check the upper-layer data and finds that the destination IP address of the inner data frame at the network layer is 192.168.200.2, which is not the IP address of the local interface. In this case, Border checks the local IP routing table before Layer 3 forwarding.

# Check the IP routing table on Border.

```
[Border]display ip routing-table 192.168.200.2
Route Flags: R - relay, D - download to fib, T - to vpn-instance
------------------------------------------------------------------------------
Routing Table : Public
Summary Count : 1
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        192.168.200.0/24  Direct  0    0           D   192.168.200.254  Vbdif20
```

Run the **display ip routing-table 192.168.200.2** command to check the result of searching the routing table for the packets destined for 192.168.200.2 by Border. The command output shows that a direct route generated by VBDIF 20 is matched. In this

case, Border determines the forwarding interface based on the ARP and MAC address entries.

# Check the ARP and MAC address entries on Border.

```
[Border]display arp | include 192.168.200.2
IP ADDRESS      MAC ADDRESS     EXPIRE(M) TYPE       INTERFACE   VPN-INSTANCE
                                VLAN/CEVLAN(SIP/DIP)
--------------------------------------------------------------------------------
192.168.200.254 9400-b049-9ef6              I-              Vbdif20
192.168.200.2   000c-292e-b4a3  11          D-0             Vbdif20
--------------------------------------------------------------------------------
Total:9         Dynamic:4       Static:0    Interface:5

[Border]display mac-address 000c-292e-b4a3
--------------------------------------------------------------------------------
MAC Address   VLAN/VSI/BD   Learned-From        Type
--------------------------------------------------------------------------------
000c-292e-b4a3  -/-/20          10.0.3.3            dynamic

--------------------------------------------------------------------------------
Total items displayed = 1
```

According to the ARP and MAC address entries, Border needs to re-encapsulate the data frame and forward it to the remote VTEP (10.0.3.3).

In this case, Border re-encapsulates the inner data frame (by replacing the source and destination MAC addresses), and then sends the data frame to Edge_2 through the static VXLAN tunnel.

**----End**

## 5.2.3 Quiz

After Edge_2 receives VXLAN packets (in which the original data frame for PC1 to ping PC2 is encapsulated) from Border, how will Edge_2 process the packets?

# 5.3 Distributed VXLAN Gateway

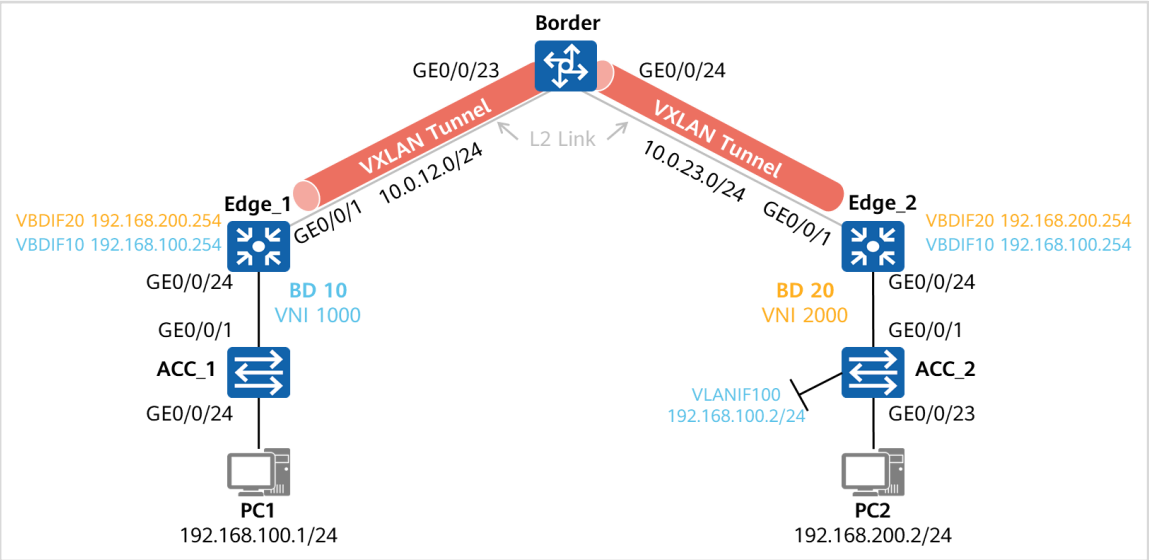
## 5.3.1 About This Lab

### 5.3.1.1 Objectives

- Configure distributed VXLAN gateways.
- Analyze the packet forwarding process on the same network segment and across network segments in the distributed VXLAN gateway scenario.

### 5.3.1.2 Networking Description

**Figure 5-3 Distributed VXLAN gateway topology**




The figure shows the device interconnection. Loopback0 interfaces are created on Border, Edge_1, and Edge_2, and their IP addresses are in the format of 10.0.*x.x*. In the format, *x* indicates the device ID and is marked in tables in the corresponding configuration procedures.

Configure the interfaces connecting Edge_1 and Border as trunk interfaces and allow packets from VLAN12 to pass through. VLANIF12 is created on Edge_1 and Border for Layer 3 interconnection, and the IP address is 10.0.12.*x*/24.

Configure the interfaces connecting Border and Edge_2 as trunk interfaces and allow packets from VLAN23 to pass through. VLANIF23 is created on Border and Edge_2 for Layer 3 interconnection, and the IP address is 10.0.23.*x*/24.

OSPF runs between Edge_1, Border, and Edge_2, the IP address of Loopback0 is used as the router ID, and OSPF is enabled on loopback and interconnection interfaces.

Edge_1 and Edge_2 use the address of Loopback0 as the source address of the VXLAN NVE interface to transmit the traffic of BD 10 and BD 20.

ACC_1 and ACC_2 function as access switches and interconnect with PC1 and PC2, respectively, through access interfaces. PC1 and PC2 are added to VLAN 100 and VLAN 200, respectively. Create VLANIF 100 on ACC_2 to simulate a terminal in VLAN 100. Edge_1 and Edge_2 interconnect with ACC_1 and ACC_2 through Layer 2 sub-interfaces.

Edge_1 terminates the packets from VLAN 100 and Edge_2 terminates the packets from VLAN 100 and VLAN 200, respectively. The sub-interfaces are associated with BD10 and BD20, respectively.

VBDIF 10 and VBDIF 20 are created on Edge_1 and Edge_2 to function as the gateways (distributed gateways) for terminals in BD 10 and BD 20. A BGP EVPN peer relationship is established between Edge_1 and Edge_2 to transmit Type 2 routes, thereby transmitting host routes and MAC addresses.

# 5.3.2 Configuration Procedure

## 5.3.2.1 Configuration Roadmap

1. Complete the basic configuration of connectivity.
2. Configure service access points.
3. Create a VBDIF interface and configure an NVE interface.
4. Configure EVPN and IP VPN instances and bind them to the BD and VBDIF interfaces, respectively.
5. Establish a BGP EVPN peer relationship.

## 5.3.2.2 Configuration Procedure

Step 1 Complete the basic configuration of connectivity.

Configure interconnection interface IP addresses and OSPF on Edge_1, Border, and Edge_2 according to the following table.

**Table 5-3 Device ID**

| Device Name | Device ID (X) |
|---|---|
| Edge_1 | 1 |
| Border | 2 |
| Edge_2 | 3 |

# Name the devices.

N/A

# Configure a VLAN and interfaces on Edge_1.

```
[Edge_1]vlan 12
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_1]interface GigabitEthernet0/0/1
[Edge_1-GigabitEthernet0/0/1] port link-type trunk
[Edge_1-GigabitEthernet0/0/1] port trunk allow-pass vlan 12
[Edge_1-GigabitEthernet0/0/1] quit
[Edge_1]interface Vlanif12
[Edge_1-Vlanif12] ip address 10.0.12.1 255.255.255.0
[Edge_1-Vlanif12] quit
[Edge_1]interface LoopBack 0
```

```
[Edge_1-LoopBack0] ip address 10.0.1.1 32
[Edge_1-LoopBack0] quit
```

# Configure VLANs and interfaces on Border.

```
[Border]vlan batch 12 23
Info: This operation may take a few seconds. Please wait for a moment...done.
[Border]interface GigabitEthernet0/0/23
[Border-GigabitEthernet0/0/23] port link-type trunk
[Border-GigabitEthernet0/0/23] port trunk allow-pass vlan 12
[Border-GigabitEthernet0/0/23] quit
[Border]interface GigabitEthernet0/0/24
[Border-GigabitEthernet0/0/24] port link-type trunk
[Border-GigabitEthernet0/0/24] port trunk allow-pass vlan 23
[Border-GigabitEthernet0/0/24] quit
[Border]interface Vlanif12
[Border-Vlanif12] ip address 10.0.12.2 255.255.255.0
[Border-Vlanif12] quit
[Border]interface Vlanif23
[Border-Vlanif23] ip address 10.0.23.2 255.255.255.0
[Border-Vlanif23] quit
[Border]interface LoopBack 0
[Border-LoopBack0] ip address 10.0.2.2 32
[Border-LoopBack0] quit
```

# Configure a VLAN and interfaces on Edge_2.

```
[Edge_2]vlan 23
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_2]
[Edge_2]interface GigabitEthernet0/0/1
[Edge_2-GigabitEthernet0/0/1] port link-type trunk
[Edge_2-GigabitEthernet0/0/1] port trunk allow-pass vlan 23
[Edge_2-GigabitEthernet0/0/1] quit
[Edge_2]
[Edge_2]interface Vlanif 23
[Edge_2-Vlanif23] ip address 10.0.23.3 255.255.255.0
[Edge_2-Vlanif23] quit
[Edge_2]
[Edge_2]interface LoopBack 0
[Edge_2-LoopBack0] ip address 10.0.3.3 32
[Edge_2-LoopBack0] quit
```

# Test the connectivity of the interconnection interfaces.

```
[Border]ping 10.0.12.1
  PING 10.0.12.1: 56  data bytes, press CTRL_C to break
    Reply from 10.0.12.1: bytes=56 Sequence=1 ttl=254 time=1 ms
    Reply from 10.0.12.1: bytes=56 Sequence=2 ttl=254 time=1 ms
    Reply from 10.0.12.1: bytes=56 Sequence=3 ttl=254 time=1 ms
    Reply from 10.0.12.1: bytes=56 Sequence=4 ttl=254 time=1 ms
    Reply from 10.0.12.1: bytes=56 Sequence=5 ttl=254 time=1 ms

  --- 10.0.12.1 ping statistics ---
```

```
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

[Border]ping 10.0.23.3
  PING 10.0.23.3: 56  data bytes, press CTRL_C to break
    Request time out
    Reply from 10.0.23.3: bytes=56 Sequence=2 ttl=254 time=1 ms
    Reply from 10.0.23.3: bytes=56 Sequence=3 ttl=254 time=1 ms
    Reply from 10.0.23.3: bytes=56 Sequence=4 ttl=254 time=1 ms
    Reply from 10.0.23.3: bytes=56 Sequence=5 ttl=254 time=1 ms

  --- 10.0.23.3 ping statistics ---
    5 packet(s) transmitted
    4 packet(s) received
    20.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

Test the connectivity of VLANIF interfaces to Edge_1 and Edge_2 on Border.

# Configure OSPF on Edge_1.

```
[Edge_1]ospf 1 router-id 10.0.1.1
[Edge_1-ospf-1] area 0.0.0.0
[Edge_1-ospf-1-area-0.0.0.0] network 10.0.1.1 0.0.0.0
[Edge_1-ospf-1-area-0.0.0.0] network 10.0.12.0 0.0.0.255
[Edge_1-ospf-1-area-0.0.0.0] quit
```

Use the IP address of Loopback0 as the router ID, and enable OSPF on Loopback0 and VLANIF 12.

# Configure OSPF on Border.

```
[Border]ospf 1 router-id 10.0.2.2
[Border-ospf-1] area 0.0.0.0
[Border-ospf-1-area-0.0.0.0] network 10.0.2.2 0.0.0.0
[Border-ospf-1-area-0.0.0.0] network 10.0.12.0 0.0.0.255
[Border-ospf-1-area-0.0.0.0] network 10.0.23.0 0.0.0.255
[Border-ospf-1-area-0.0.0.0] quit
```

Use the IP address of Loopback0 as the router ID, and enable OSPF on Loopback0, VLANIF 12, and VLANIF 23.

# Configure OSPF on Edge_2.

```
[Edge_2]ospf 1 router-id 10.0.3.3
[Edge_2-ospf-1] area 0.0.0.0
[Edge_2-ospf-1-area-0.0.0.0] network 10.0.3.3 0.0.0.0
[Edge_2-ospf-1-area-0.0.0.0] network 10.0.23.0 0.0.0.255
[Edge_2-ospf-1-area-0.0.0.0] quit
```

# Check the OSPF neighbor relationship and OSPF routing table on Border.

```
[Border]display ospf peer
```

```
          OSPF Process 1 with Router ID 10.0.2.2
                  Neighbors

 Area 0.0.0.0 interface 10.0.12.2(Vlanif12)'s neighbors
 Router ID: 10.0.1.1             Address: 10.0.12.1
   State: Full   Mode:Nbr is   Slave   Priority: 1
   R: 10.0.12.1   BDR: 10.0.12.2   MTU: 0
   Dead timer due in 36   sec
   Retrans timer interval: 5
   Neighbor is up for 00:01:57
   Authentication Sequence: [ 0 ]

                  Neighbors

 Area 0.0.0.0 interface 10.0.23.2(Vlanif23)'s neighbors
 Router ID: 10.0.3.3             Address: 10.0.23.3
   State: Full   Mode:Nbr is   Master   Priority: 1
   DR: 10.0.23.2   BDR: 10.0.23.3   MTU: 0
   Dead timer due in 28   sec
   Retrans timer interval: 5
   Neighbor is up for 00:01:24
   Authentication Sequence: [ 0 ]
```

The OSPF neighbor relationship has been established.

# Check the OSPF routing table on Border.

```
[Border]display ospf routing

      OSPF Process 1 withRouter ID 10.0.2.2
           Routing Tables

Routing forNetwork
Destination       Cost  Type      NextHop        AdvRouter       Area
10.0.2.2/32       0     Stub      10.0.2.2       10.0.2.2        0.0.0.0
10.0.12.0/24      1     Transit   10.0.12.2      10.0.2.2        0.0.0.0
10.0.23.0/24      1     Transit   10.0.23.2      10.0.2.2        0.0.0.0
10.0.1.1/32       1     Stub      10.0.12.1      10.0.1.1        0.0.0.0
10.0.3.3/32       1     Stub      10.0.23.3      10.0.3.3        0.0.0.0

TotalNets:5
Intra Area: 5  Inter Area: 0  ASE:0  NSSA: 0
```

Border has learned the routes generated by Loopback0 interfaces on Edge_1 and Edge_2.

# Test the connectivity of Loopback0 interfaces between Edge_1 and Edge_2.

```
[Edge_1]ping -a 10.0.1.1 10.0.3.3
  PING 10.0.3.3: 56   data bytes, press CTRL_C to break
    Reply from 10.0.3.3: bytes=56 Sequence=1 ttl=253 time=1 ms
    Reply from 10.0.3.3: bytes=56 Sequence=2 ttl=253 time=1 ms
    Reply from 10.0.3.3: bytes=56 Sequence=3 ttl=253 time=1 ms
    Reply from 10.0.3.3: bytes=56 Sequence=4 ttl=253 time=1 ms
    Reply from 10.0.3.3: bytes=56 Sequence=5 ttl=253 time=1 ms
```

```
  --- 10.0.3.3 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 1/1/1 ms
```

Edge_1 and Edge_2 can communicate with each other through their Loopback0 interfaces.

Step 2 Configure service access points.

Configure a service access point on Edge_1 and Edge_2, create sub-interfaces to interconnect with ACC_1 and ACC_2 and terminate packets from VLAN 100 and VLAN 200, and bind the sub-interfaces to BD 10 and BD 20.

Configure the interfaces connecting ACC_1 and ACC_2 to PCs as access interfaces and set PVIDs to 100 and 200. Configure the interfaces connecting ACC_1 and ACC_2 to Edge_1 and Edge_2 as trunk interfaces, and allow packets from VLAN 100 and VLAN 200 to pass through.

Create VLANIF 100 on ACC_2 and assign IP address 192.168.100.2/24 to it to simulate a terminal user.

# Configure Edge_1.

```
[Edge_1]bridge-domain 10
[Edge_1-bd10] vxlan vni 1000
[Edge_1-bd10] quit
[Edge_1]bridge-domain 20
[Edge_1-bd20] vxlan vni 2000
[Edge_1-bd20] quit
[Edge_1]vcmp role silent
[Edge_1]interface GigabitEthernet0/0/24
[Edge_1-GigabitEthernet0/0/24]port link-type trunk
Info: This operation may take a few seconds. Please wait for a moment...
[Edge_1-GigabitEthernet0/0/24]quit
[Edge_1]interface GigabitEthernet0/0/24.100 mode l2
[Edge_1-GigabitEthernet0/0/24.100] encapsulation dot1q vid 100
Info: This operation may take a few seconds. Please wait for a moment...
[Edge_1-GigabitEthernet0/0/24.100] bridge-domain 10
[Edge_1-GigabitEthernet0/0/24.100] quit
```

By default, after VCMP is configured, the encapsulation mode of packets allowed to pass a Layer 2 sub-interface cannot be set to Dot1q. In this case, you need to run the **vcmp role silent** command.

# Configure Edge_2.

```
[Edge_2]bridge-domain 10
[Edge_2-bd10] vxlan vni 1000
[Edge_2-bd10] quit
[Edge_2]bridge-domain 20
[Edge_2-bd20] vxlan vni 2000
[Edge_2-bd20] quit
[Edge_2]vcmp role silent
```

```
[Edge_2]interface GigabitEthernet0/0/24
[Edge_2-GigabitEthernet0/0/24]port link-type trunk
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_2-GigabitEthernet0/0/24]quit
[Edge_2]interface GigabitEthernet0/0/24.200 mode l2
[Edge_2-GigabitEthernet0/0/24.200] encapsulation dot1q vid 200
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_2-GigabitEthernet0/0/24.200] bridge-domain 20
[Edge_2-GigabitEthernet0/0/24.200] quit
[Edge_2]interface GigabitEthernet0/0/24.100 mode l2
[Edge_2-GigabitEthernet0/0/24.100] encapsulation dot1q vid 100
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_2-GigabitEthernet0/0/24.100] bridge-domain 10
[Edge_2-GigabitEthernet0/0/24.100] quit
```

Create BD 10 and BD 20. Create sub-interfaces to terminate packets from VLAN 100 and VLAN 200.

# Configure ACC_1.

```
[ACC_1]vlan batch 100 200
Info: This operation may take a few seconds. Please wait for a moment...done.
[ACC_1-vlan100]interface GigabitEthernet0/0/24
[ACC_1-GigabitEthernet0/0/24] port link-type access
[ACC_1-GigabitEthernet0/0/24] port default vlan 100
[ACC_1-GigabitEthernet0/0/24] quit
[ACC_1]interface GigabitEthernet0/0/1
[ACC_1-GigabitEthernet0/0/1] port link-type trunk
[ACC_1-GigabitEthernet0/0/1] port trunk allow-pass vlan 100 200
[ACC_1-GigabitEthernet0/0/1] quit
```

# Configure ACC_2.

```
[ACC_2] vlan batch 100 200
Info: This operation may take a few seconds. Please wait for a moment...done.
[ACC_2] interface GigabitEthernet0/0/23
[ACC_2-GigabitEthernet0/0/23] port link-type access
[ACC_2-GigabitEthernet0/0/23] port default vlan 200
[ACC_2-GigabitEthernet0/0/23] quit
[ACC_2] interface GigabitEthernet0/0/1
[ACC_2-GigabitEthernet0/0/1] port link-type trunk
[ACC_2-GigabitEthernet0/0/1] port trunk allow-pass vlan 100 200
[ACC_2-GigabitEthernet0/0/1] quit
[ACC_2] interface vlanif100
[ACC_2-Vlanif100] ip address 192.168.100.2 255.255.255.0
[ACC_2-Vlanif100] quit
[ACC_2] ip route-static 192.168.200.0 255.255.255.0 192.168.100.254
```

Create VLANIF 100, and configure a static route destined to 192.168.200.0/24 with the next hop being 192.168.100.254.

Step 3 Create a VBDIF interface and configure an NVE interface.

Create NVE interfaces on Edge_1 and Edge_2, and set the ingress replication list protocol to BGP.

# Configure an NVE interface on Edge_1.

```
[Edge_1] interface Nve1
[Edge_1-Nve1] source 10.0.1.1
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_1-Nve1] vni 1000 head-end peer-list protocol bgp
[Edge_1-Nve1] vni 2000 head-end peer-list protocol bgp
[Edge_1-Nve1] quit
```

# Configure an NVE interface on Edge_2.

```
[Edge_2]interface Nve1
[Edge_2-Nve1] source 10.0.3.3
Info: This operation may take a few seconds. Please wait for a moment...done.
[Edge_2-Nve1] vni 1000 head-end peer-list protocol bgp
[Edge_2-Nve1] vni 2000 head-end peer-list protocol bgp
[Edge_2-Nve1] quit
```

# Create VBDIF 10 and VBDIF 20 on Edge_1.

```
[Edge_1]interface Vbdif10
[Edge_1-Vbdif10] ip address 192.168.100.254 255.255.255.0
[Edge_1-Vbdif10] mac-address 0000-5e00-0110
[Edge_1-Vbdif10] arp collect host enable
[Edge_1-Vbdif10] quit
[Edge_1]interface Vbdif20
[Edge_1-Vbdif20] ip address 192.168.200.254 255.255.255.0
[Edge_1-Vbdif20] mac-address 0000-5e00-0120
[Edge_1-Vbdif20] arp collect host enable
[Edge_1-Vbdif20] quit
```

# Create VBDIF 10 and VBDIF 20 on Edge_2.

```
[Edge_2]interface Vbdif10
[Edge_2-Vbdif10] ip address 192.168.100.254 255.255.255.0
[Edge_2-Vbdif10] mac-address 0000-5e00-0110
[Edge_2-Vbdif10] arp collect host enable
[Edge_2-Vbdif10] quit
[Edge_2]interface Vbdif20
[Edge_2-Vbdif20] ip address 192.168.200.254 255.255.255.0
[Edge_2-Vbdif20] mac-address 0000-5e00-0120
[Edge_2-Vbdif20] arp collect host enable
[Edge_2-Vbdif20] quit
```

Create VBDIF 10 and VBDIF 20 on both Edge_1 and Edge_2. Then, change the MAC addresses of the interfaces to 0000-5e00-0110 and 0000-5e00-0120 respectively, and run the **arp collect host enable** command on the VBDIF interfaces to transmit Type 2 routes generated based on ARP information of hosts through BGP EVPN. By doing so, the MAC addresses of the gateways remain unchanged and the hosts do not need to re-learn the ARP entries when the hosts are migrated between different aggregation switches.

Step 4 Configure EVPN and IP VPN instances and bind them to the BD and VBDIF interfaces, respectively.

Create an EVPN instance and an IP VPN instance on Edge_1 and Edge_2, and bind the instances to BD and VBDIF interfaces, respectively. To ensure that EVPN routes and IP routes can be learned mutually, plan EVPN and IP VPN instances as follows.

**Table 5-4 EVPN instance planing on Edge_1**

| EVPN Instance Name | BD | RD | Export RT | Import RT |
|---|---|---|---|---|
| Edge_1_BD_10 | 10 | 21:10 | 213:10, 313:12 | 213:10 |
| Edge_1_BD_20 | 20 | 21:20 | 213:20, 313:12 | 213:20 |

**Table 5-5 EVPN instance planing on Edge_2**

| EVPN Instance Name | BD | RD | Export RT | Import RT |
|---|---|---|---|---|
| Edge_2_BD_10 | 10 | 23:10 | 213:10, 313:12 | 213:10 |
| Edge_2_BD_20 | 20 | 23:20 | 213:20, 313:12 | 213:20 |

**Table 5-6 IP VPN instance planing on Edge_1**

| VPN Instance Name | VBDIF | L3 VNI | RD | Export RT | Import RT |
|---|---|---|---|---|---|
| Edge_1_VPN_10_20 | 10 | 1020 | 31:12 | 313:12 | 313:12 |
| | 20 | | | | |

**Table 5-7 IP VPN instance planing on Edge_2**

| VPN Instance Name | VBDIF | L3 VNI | RD | Export RT | Import RT |
|---|---|---|---|---|---|
| Edge_2_VPN_10_20 | 10 | 1020 | 33:12 | 313:12 | 313:12 |
| | 20 | | | | |

EVPN instance planning (In the following planning, the IDs of Edge_1, Border, and Edge_2 are 1, 2 and 3, respectively.):

1. RD: 2Y:Z, where 2 indicates an EVPN instance, Y indicates the device ID, and Z indicates the BD. Take BD 10 on Edge_1 as an example. The RD for the corresponding EVPN instance is 21:10.
2. Import RT:213:Z, where 2 indicates an EVPN instance, 13 indicates that the RT is used to transmit EVPN routes between Edge_1 and Edge_2, and Z indicates the BD. Take BD 10 on Edge_1 as an example. The import RT for the corresponding EVPN instance is 213:10.

3. Export RT: Except for the one that is the same as the import RT 213:Z, the other export RT is 313:12, which is used to import IP routes into the IP VPN instance routing table.

IP VPN instance planning:

1. RD: 3X:12, where 3 indicates an IP VPN instance, X indicates the device ID, and 12 indicates that the IP VPN instance is used by VBDIF 10 and VBDIF 20.
2. Export RT and import RT: 313:12, where 3 indicates an IP VPN instance, 13 indicates that the IP VPN instance is used between Edge_1 and Edge_2, and 12 indicates VBDIF 10 and VBDIF 20.
3. L3VNI: 1020, which indicates the L3VNI used for communication between BD 10 and BD 20.

# On Edge_1, create EVPN instances and bind them to a BD.

```
[Edge_1]evpn vpn-instance Edge_1_BD_10 bd-mode
[Edge_1-evpn-instance-Edge_1_BD_10] route-distinguisher 21:10
[Edge_1-evpn-instance-Edge_1_BD_10] vpn-target 213:10 export-extcommunity
[Edge_1-evpn-instance-Edge_1_BD_10] vpn-target 313:12 export-extcommunity
[Edge_1-evpn-instance-Edge_1_BD_10] vpn-target 213:10 import-extcommunity
[Edge_1-evpn-instance-Edge_1_BD_10] quit
[Edge_1]evpn vpn-instance Edge_1_BD_20 bd-mode
[Edge_1-evpn-instance-Edge_1_BD_20] route-distinguisher 21:20
[Edge_1-evpn-instance-Edge_1_BD_20] vpn-target 213:20 export-extcommunity
[Edge_1-evpn-instance-Edge_1_BD_20] vpn-target 313:12 export-extcommunity
[Edge_1-evpn-instance-Edge_1_BD_20] vpn-target 213:20 import-extcommunity
[Edge_1-evpn-instance-Edge_1_BD_20] quit
[Edge_1]bridge-domain 10
[Edge_1-bd10]evpn binding vpn-instance Edge_1_BD_10
[Edge_1-bd10]quit
[Edge_1]bridge-domain 20
[Edge_1-bd20]evpn binding vpn-instance Edge_1_BD_20
[Edge_1-bd20]quit
```

# On Edge_2, create EVPN instances and bind them to a BD.

```
[Edge_2]evpn vpn-instance Edge_2_BD_10 bd-mode
[Edge_2-evpn-instance-Edge_2_BD_10] route-distinguisher 23:10
[Edge_2-evpn-instance-Edge_2_BD_10] vpn-target 213:10 export-extcommunity
[Edge_2-evpn-instance-Edge_2_BD_10] vpn-target 313:12 export-extcommunity
[Edge_2-evpn-instance-Edge_2_BD_10] vpn-target 213:10 import-extcommunity
[Edge_2-evpn-instance-Edge_2_BD_10] quit
[Edge_2]evpn vpn-instance Edge_2_BD_20 bd-mode
[Edge_2-evpn-instance-Edge_2_BD_20] route-distinguisher 23:20
[Edge_2-evpn-instance-Edge_2_BD_20] vpn-target 213:20 export-extcommunity
[Edge_2-evpn-instance-Edge_2_BD_20] vpn-target 313:12 export-extcommunity
[Edge_2-evpn-instance-Edge_2_BD_20] vpn-target 213:20 import-extcommunity
[Edge_2-evpn-instance-Edge_2_BD_20] quit
[Edge_2]bridge-domain 10
[Edge_2-bd10]evpn binding vpn-instance Edge_2_BD_10
[Edge_2-bd10]quit
[Edge_2]bridge-domain 20
[Edge_2-bd20]evpn binding vpn-instance Edge_2_BD_20
```

```
[Edge_2-bd20]quit
```

# On Edge_1, create an IP VPN instance and bind it to a VBDIF.

```
[Edge_1]ip vpn-instance Edge_1_VPN_10_20
[Edge_1-vpn-instance-Edge_1_VPN_10_20] ipv4-family
[Edge_1-vpn-instance-Edge_1_VPN_10_20-af-ipv4] route-distinguisher 31:12
[Edge_1-vpn-instance-Edge_1_VPN_10_20-af-ipv4] vpn-target 313:12 export-extcommunity evpn
[Edge_1-vpn-instance-Edge_1_VPN_10_20-af-ipv4] vpn-target 313:12 import-extcommunity evpn
[Edge_1-vpn-instance-Edge_1_VPN_10_20-af-ipv4] quit
[Edge_1-vpn-instance-Edge_1_VPN_10_20] vxlan vni 1020
[Edge_1-vpn-instance-Edge_1_VPN_10_20] quit
[Edge_1]interface Vbdif 10
[Edge_1-Vbdif10]ip binding vpn-instance Edge_1_VPN_10_20
Info: IGMP/MLD snooping configurations in the corresponding VLAN are cleared!
Info: All IPv4 related configurations on this interface are removed!
Info: All IPv6 related configurations on this interface are removed!
[Edge_1-Vbdif10]quit
[Edge_1]interface Vbdif 20
[Edge_1-Vbdif20]ip binding vpn-instance Edge_1_VPN_10_20
Info: IGMP/MLD snooping configurations in the corresponding VLAN are cleared!
Info: All IPv4 related configurations on this interface are removed!
Info: All IPv6 related configurations on this interface are removed!
[Edge_1-Vbdif20]quit
[Edge_1]interface Vbdif10
[Edge_1-Vbdif10] ip address 192.168.100.254 255.255.255.0
[Edge_1-Vbdif10] quit
[Edge_1]interface Vbdif20
[Edge_1-Vbdif20] ip address 192.168.200.254 255.255.255.0
[Edge_1-Vbdif20] quit
```

Note: After an IP VPN instance is bound to an interface, the IP address configuration of the interface will be cleared. In this case, you need to reconfigure an IP address for the interface.

# On Edge_2, create an IP VPN instance and bind it to a VBDIF.

```
[Edge_2]ip vpn-instance Edge_2_VPN_10_20
[Edge_2-vpn-instance-Edge_2_VPN_10_20] ipv4-family
[Edge_2-vpn-instance-Edge_2_VPN_10_20-af-ipv4] route-distinguisher 33:12
[Edge_2-vpn-instance-Edge_2_VPN_10_20-af-ipv4] vpn-target 313:12 export-extcommunity
[Edge_2-vpn-instance-Edge_2_VPN_10_20-af-ipv4] vpn-target 313:12 export-extcommunity evpn
[Edge_2-vpn-instance-Edge_2_VPN_10_20-af-ipv4] vpn-target 313:12 import-extcommunity
[Edge_2-vpn-instance-Edge_2_VPN_10_20-af-ipv4] vpn-target 313:12 import-extcommunity evpn
[Edge_2-vpn-instance-Edge_2_VPN_10_20-af-ipv4] quit
[Edge_2-vpn-instance-Edge_2_VPN_10_20] vxlan vni 1020
[Edge_2-vpn-instance-Edge_2_VPN_10_20] quit
[Edge_2]interface Vbdif 10
[Edge_2-Vbdif10]ip binding vpn-instance Edge_2_VPN_10_20
Info: IGMP/MLD snooping configurations in the corresponding VLAN are cleared!
Info: All IPv4 related configurations on this interface are removed!
Info: All IPv6 related configurations on this interface are removed!
[Edge_2-Vbdif10]quit
[Edge_2]interface Vbdif 20
[Edge_2-Vbdif20]ip binding vpn-instance Edge_2_VPN_10_20
```

```
Info: IGMP/MLD snooping configurations in the corresponding VLAN are cleared!
Info: All IPv4 related configurations on this interface are removed!
Info: All IPv6 related configurations on this interface are removed!
[Edge_2-Vbdif20]quit
[Edge_2]interface Vbdif10
[Edge_2-Vbdif10] ip address 192.168.100.254 255.255.255.0
[Edge_2-Vbdif10] quit
[Edge_2]interface Vbdif20
[Edge_2-Vbdif20] ip address 192.168.200.254 255.255.255.0
[Edge_2-Vbdif20] quit
```

Note: After an IP VPN instance is bound to an interface, the IP address configuration of the interface will be cleared. In this case, you need to reconfigure an IP address for the interface.

Step 5 Establish a BGP EVPN peer relationship.

Establish an IBGP peer relationship between Edge_1 and Edge_2, use the address of the Loopback interface as the source address of the BGP session, enable IRB route advertisement in the EVPN address family, and enable L2VPN EVPN route in the VPN instance.

# Configure BGP EVPN on Edge_1.

```
[Edge_1] bgp 100
[Edge_1-bgp] router-id 10.0.1.1
[Edge_1-bgp] peer 10.0.3.3 as-number 100
[Edge_1-bgp] peer 10.0.3.3 connect-interface LoopBack0
[Edge_1-bgp] ipv4-family vpn-instance Edge_1_VPN_10_20
[Edge_1-bgp-Edge_1_VPN_10_20] advertise l2vpn evpn
[Edge_1-bgp-Edge_1_VPN_10_20] quit
[Edge_1-bgp] l2vpn-family evpn
[Edge_1-bgp-af-evpn] policy vpn-target
[Edge_1-bgp-af-evpn] peer 10.0.3.3 enable
[Edge_1-bgp-af-evpn] peer 10.0.3.3 advertise irb
[Edge_1-bgp-af-evpn] quit
```

# Configure BGP EVPN on Edge_2.

```
[Edge_2] bgp 100
[Edge_2-bgp] router-id 10.0.3.3
[Edge_2-bgp] peer 10.0.1.1 as-number 100
[Edge_2-bgp] peer 10.0.1.1 connect-interface LoopBack0
[Edge_2-bgp] ipv4-family vpn-instance Edge_2_VPN_10_20
[Edge_2-bgp-Edge_2_VPN_10_20] advertise l2vpn evpn
[Edge_2-bgp-Edge_2_VPN_10_20] quit
[Edge_2-bgp] l2vpn-family evpn
[Edge_2-bgp-af-evpn] policy vpn-target
[Edge_2-bgp-af-evpn] peer 10.0.1.1 enable
[Edge_2-bgp-af-evpn] peer 10.0.1.1 advertise irb
[Edge_2-bgp-af-evpn] quit
```

# Check the BGP EVPN peer relationship status on Edge_1 and Edge_2.

```
[Edge_1]display bgp evpn peer

Status codes: * - Dynamic

BGP local router ID : 10.0.1.1
Local AS number : 100
Total number of peers : 1                 Peers in established state : 1
Total number of dynamic peers : 0

 Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State    PrefRcv

 10.0.3.3        4         100         4        4             0       00:00:04    Established         2
```

```
[Edge_2]display bgp evpn peer

Status codes: * - Dynamic

BGP local router ID : 10.0.3.3
Local AS number : 100
Total number of peers : 1                 Peers in established state : 1
Total number of dynamic peers : 0

 Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State    PrefRcv

 10.0.1.1        4         100         4        4             0       00:00:08    Established         2
```

The BGP EVPN peer relationship between Edge_1 and Edge_2 is normally established.

Step 6 Verify configurations and check the communication process.

On PC1, ping 192.168.100.2 (VLANIF 100 on Edge_2) on the same network segment and ping PC2 on a different network segment. Check VXLAN tunnels established by using BGP EVPN, and check the ingress replication list. Analyze the intra-subnet and the inter-subnet data forwarding processes.

# Use PC1, PC2, and VLANIF 100 to ping their respective gateways.

```
C:\Users\PC1>ping 192.168.100.254

Pinging 192.168.100.254 with 32 bytes of data:
Reply from 192.168.100.254: bytes=32 time<1ms TTL=254
Reply from 192.168.100.254: bytes=32 time<1ms TTL=254
Reply from 192.168.100.254: bytes=32 time<1ms TTL=254
Reply from 192.168.100.254: bytes=32 time<1ms TTL=254

Ping statistics for 192.168.100.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\Users\PC2>ping 192.168.200.254
```

```
Pinging 192.168.200.254 with 32 bytes of data:
Reply from 192.168.200.254: bytes=32 time<1ms TTL=254
Reply from 192.168.200.254: bytes=32 time<1ms TTL=254
Reply from 192.168.200.254: bytes=32 time<1ms TTL=254
Reply from 192.168.200.254: bytes=32 time<1ms TTL=254

Ping statistics for 192.168.200.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
<ACC_2>ping 192.168.100.254
  PING 192.168.100.254: 56  data bytes, press CTRL_C to break
    Reply from 192.168.100.254: bytes=56 Sequence=1 ttl=254 time=1 ms
    Reply from 192.168.100.254: bytes=56 Sequence=2 ttl=254 time=1 ms
    Reply from 192.168.100.254: bytes=56 Sequence=3 ttl=254 time=1 ms
    Reply from 192.168.100.254: bytes=56 Sequence=4 ttl=254 time=1 ms
    Reply from 192.168.100.254: bytes=56 Sequence=5 ttl=254 time=1 ms

  --- 192.168.100.254 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

The command outputs show that the communication with corresponding VBDIF gateways is normal.

# Check VXLAN tunnel information on Edge_1.

```
[Edge_1]display vxlan tunnel
Tunnel ID       Source                  Destination             State   Type
--------------------------------------------------------------------------------
4026531841      10.0.1.1                10.0.3.3                up      l2 dynamic
3               10.0.1.1                10.0.3.3                up      l3 dynamic
--------------------------------------------------------------------------------
Number of vxlan tunnel:
Total: 2    Static: 0    L2 dynamic: 1    L3 dynamic: 1
```

Layer 2 and Layer 3 VXLAN tunnels have been established between Edge_1 and Edge_2.

# Check Type 3 BGP EVPN routes on Edge_1.

```
<Edge_1> display bgp evpn all routing-table inclusive-route
 Local AS number : 100

 BGP  Local router  ID is 10.0.1.1
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

 EVPN address family:
```

```
  Number of Inclusive Multicast Routes: 4
 Route Distinguisher: 21:10
         Network(EthTagId/IpAddrLen/OriginalIp)                NextHop
 *>       0:32:10.0.1.1                                       0.0.0.0
 Route Distinguisher: 21:20
         Network(EthTagId/IpAddrLen/OriginalIp)                NextHop
 *>       0:32:10.0.1.1                                       0.0.0.0
 Route Distinguisher: 23:10
         Network(EthTagId/IpAddrLen/OriginalIp)                NextHop
 *>i      0:32:10.0.3.3                                       10.0.3.3
 Route Distinguisher: 23:20
         Network(EthTagId/IpAddrLen/OriginalIp)                NextHop
 *>i      0:32:10.0.3.3                                       10.0.3.3
```

The command output shows that there are two Type 3 routes from 10.0.3.3. After receiving these Type 3 routes, the device creates a Layer 2 ingress replication list.

# Check the VXLAN peer relationship on Edge_1.

```
[Edge_1]display vxlan peer
Vni ID      Source                  Destination             Type
--------------------------------------------------------------------------------
1000        10.0.1.1                10.0.3.3                l2 dynamic
2000        10.0.1.1                10.0.3.3                l2 dynamic
-           10.0.1.1                10.0.3.3                l3 dynamic
--------------------------------------------------------------------------------
Number of peers :
Total: 3   Static: 0   L2 dynamic: 2   L3 dynamic: 1
```

Layer 2 and Layer 3 VXLAN peer relationships have been established.

# Test the connectivity between PC1 and VLANIF 100 on Edge_2.

```
C:\Users\PC1>ping 192.168.100.2

Pinging 192.168.100.2 with 32 bytes of data:
Reply from 192.168.100.2: bytes=32 time<1ms TTL=254
Reply from 192.168.100.2: bytes=32 time<1ms TTL=254
Reply from 192.168.100.2: bytes=32 time<1ms TTL=254
Reply from 192.168.100.2: bytes=32 time<1ms TTL=254

Ping statistics for 192.168.100.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

The command output shows that the communication is normal.

# On PC1, check the ARP information about 192.168.100.2. (Irrelevant entries have been omitted.)

```
C:\Users\PC1>arp -a

Interface: 192.168.100.1 --- 0xc
  Internet Address      Physical Address      Type
```

```
  192.168.100.2           b0-08-75-0e-fb-db       dynamic
```

The MAC address corresponding to 192.168.100.2 is b0-08-75-0e-fb-db.

# Check the MAC address entries in BD 10 on Edge_1.

```
<Edge_1>display mac-address bridge-domain 10
--------------------------------------------------------------------------------
MAC Address         VLAN/VSI/BD                 Learned-From         Type
--------------------------------------------------------------------------------
000c-29b3-efea       -/-/10                     GE0/0/24.100         dynamic
b008-750e-fbdb       -/-/10                     10.0.3.3             dynamic

--------------------------------------------------------------------------------
Total items displayed = 2
```

The MAC address entry b008-750e-fbdb on Edge_1 is learned from the remote VTEP (10.0.3.3).

In this case, if PC1 sends a unicast frame to 192.168.100.2, Edge_1 will check the destination MAC address and search for the MAC address in corresponding BD (BD 10) after Edge_1 has received the frame. It is found that the matched MAC address entry is the one learned from the remote VTEP and that the entry is generated based on the route learned through BGP EVPN.

# Check BGP EVPN Type 2 routes on Edge_1.

```
<Edge_1>display bgp evpn all routing-table mac-route

 Local AS number :100

 BGP Local router ID is 10.0.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,   i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete

 EVPN address family:
 Number of Mac Routes : 3
 Route Distinguisher: 21:10
          Network(EthTagId/MacAddrLen/MacAddr/IpAddrLen/IpAddr)   NextHop
 *>       0:48:000c-29b3-efea:32:192.168.100.1                      0.0.0.0
 Route Distinguisher: 23:10
          Network(EthTagId/MacAddrLen/MacAddr/IpAddrLen/IpAddr)   NextHop
 *>i      0:48:b008-750e-fbdb:32:192.168.100.2                       10.0.3.3

 Route Distinguisher: 23:20
          Network(EthTagId/MacAddrLen/MacAddr/IpAddrLen/IpAddr)   NextHop
 *>i      0:48:000c-292e-b4a3:32:192.168.200.2                       10.0.3.3
 VPN-Instance Edge_1_VPN_10_20, Router ID 10.0.1.1:
 Total Number of Routes: 2
       Network              NextHop          MED          LocPrf      PrefVal Path/Ogn
 *>i   192.168.100.2/32     10.0.3.3                      100         0       ?
 *>i   192.168.200.2/32     10.0.3.3                      100         0       ?
```

The command output shows that there is a Type 2 route from 10.0.3.3, which contains the MAC address corresponding to 192.168.100.2.

# Check detailed information about the route.

```
<Edge_1>display bgp evpn all routing-table mac-route 0:48:b008-750e-fbdb:32:192.168.100.2

 BGP local router ID : 10.0.1.1
 Local AS number : 100

 Total routes of Route Distinguisher(23:10): 1
 BGP routing table entry information of 0:48:b008-750e-fbdb:32:192.168.100.2:
 Label information (Received/Applied): 1000 1020/NULL
 From: 10.0.3.3 (10.0.3.3)
 Route Duration: 00h29m22s
 Relay IP Nexthop: 10.0.12.2
 Relay IP Out-Interface: Vlanif12
 Original nexthop: 10.0.3.3
 Qos information : 0x0
 Ext-Community:RT <213 : 10>, RT <313 : 12>,
                    Tunnel Type <VxLan(8)>, MAC Mobility <flag:0 seq:1 res:0>,
                    Router's MAC <80e1-bf0e-5650>
 AS-path Nil, origin incomplete, localpref 100, pref-val 0, valid, internal, best, select, pre 255, IGP cost
2
 Route Type: 2 (MAC Advertisement Route)
 Ethernet Tag ID:0, MAC Address/Len: b008-750e-fbdb/48, IP Address/Len: 192.168.100.2/32, ESI:
0000.0000.0000.0000.0000
 Not advertised to any peer yet
```

In the label information, it is found that the first VNI is the L2VNI (1000) and the second VNI is the L3VNI (1020). The RT values are 213:10 and 313:12.

This route is generated by Edge_2 based on the host ARP information in BD 10. (After the **arp collect host enable** command is run on the VBDIF interface, Edge_2 converts the learned ARP information of a downstream terminal into an EVPN Type 2 route and externally sends the route out.) The route contains the information about the host IP address, host MAC address, L2VNI, and L3VNI.

In this case, Edge_1 performs Layer 2 forwarding. That is, Edge_1 encapsulates the data frame into a VXLAN header based on the result of searching the MAC address table, and then forwards the VXLAN packet to Edge_2. The VNI carried in the VXLAN packet is the L2VNI, which is the VNI (1000) bound to BD 10.

# Check the communication process between PC1 and PC2.

```
C:\Users\PC1>ping 192.168.200.2

Pinging 192.168.200.2 with 32 bytes of data:
Reply 192.168.200.2: bytes=32 time<1ms TTL=126
Reply 192.168.200.2: bytes=32 time<1ms TTL=126
Reply 192.168.200.2: bytes=32 time<1ms TTL=126
Reply 192.168.200.2: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.200.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
      Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

PC1 and PC2 communicate with each other normally.

# Check the IP routing table on Edge_1.

```
<Edge_1>display ip routing-table vpn-instance Edge_1_VPN_10_20
Route Flags: R - relay, D - download to fib, T - to vpn-instance
------------------------------------------------------------------------------
Routing Tables: Edge_1_VPN_10_20
         Destinations : 6        Routes : 6

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

   192.168.100.0/24  Direct  0    0           D   192.168.100.254 Vbdif10
   192.168.100.2/32  IBGP    255  0           RD  10.0.3.3        VXLAN
192.168.100.254/32   Direct  0    0           D   127.0.0.1       Vbdif10
   192.168.200.0/24  Direct  0    0           D   192.168.200.254 Vbdif20
   192.168.200.2/32  IBGP    255  0           RD  10.0.3.3        VXLAN
192.168.200.254/32   Direct  0    0           D   127.0.0.1       Vbdif20
```

PC1 and PC2 communicate with each other across subnets. The packet from PC1 to PC2 is first sent to the gateway of PC1 (VBDIF 10 on Edge_1). In this case, the destination MAC address of the data frame is 0000-5e00-0110, which is the MAC address of a local interface on Edge_1. Then, Edge_1 checks the destination IP address of the packet, finding that the IP address is not the address of a local interface. Therefore, Edge_1 determines that the packet needs to be further forwarded. Because the Layer 3 interface (VBDIF 10) that receives the packet is bound to the IP VPN instance Edge_1_VPN_10_20, Edge_1 searches for a route in the IP VPN instance. An IBGP route with the next hop being 10.0.3.3 is matched, whose outbound interface is the VXLAN logical interface. In this case, Edge_1 encapsulates the data frame in the VXLAN header and sends it to 10.0.3.3. The VNI in the VXLAN header is determined by checking the BGP EVPN routing entry.

# Check BGP VPNv4 routes on Edge_1.

```
<Edge_1>display bgp vpnv4 vpn-instance Edge_1_VPN_10_20 routing-table 192.168.200.2

 BGP local router ID : 10.0.1.1
 Local AS number : 100

 VPN-Instance Edge_1_VPN_10_20, Router ID 10.0.1.1:
 Paths:    1 available, 1 best, 1 select
 BGP routing table entry information of 192.168.200.2/32:
 Label information (Received/Applied): 1020/NULL
 From: 10.0.3.3 (10.0.3.3)
 Route Duration: 00h48m07s
 Relay Tunnel Out-Interface: VXLAN
 Relay token: 0x3
 Original nexthop: 10.0.3.3
 Qos information : 0x0
 Ext-Community:RT <213 : 20>, RT <313 : 12>,
                    Tunnel Type <VxLan(8)>, MAC Mobility <flag:0 seq:3 res:0>,
```

```
                    Router's MAC <80e1-bf0e-5650>
AS-path Nil, origin incomplete, localpref 100, pref-val 0, valid, internal, best, select, active, pre 255
Not advertised to any peer yet
```

In the BGP VPNv4 routing entry, it is found that the label (L3VNI) of routing entry 192.168.200.2 is 1020, and the router's MAC address is 80e1-bf0e-5650. In this case, the VNI and the MAC address are respectively the VNI after the re-encapsulation of the data frame and the destination MAC address of the inner data frame.

Next, check BGP EVPN Type 2 routes.

# Check BGP EVPN Type 2 routes on Edge_1.

```
<Edge_1>display bgp evpn all routing-table mac-route

 Local AS number :100

 BGP Local router ID is 10.0.1.1
 Status codes: * - valid, > - best, d - damped,
                    h - history,   i - internal, s - suppressed, S - Stale
                    Origin : i - IGP, e - EGP, ? - incomplete

 EVPN address family:
 Number of Mac Routes : 3
 Route Distinguisher: 21:10
          Network(EthTagId/MacAddrLen/MacAddr/IpAddrLen/IpAddr)    NextHop
 *>       0:48:000c-29b3-efea:32:192.168.100.1                     0.0.0.0

 Route Distinguisher: 23:10
          Network(EthTagId/MacAddrLen/MacAddr/IpAddrLen/IpAddr)    NextHop
 *>i      0:48:b008-750e-fbdb:32:192.168.100.2                     10.0.3.3

 Route Distinguisher: 23:20
          Network(EthTagId/MacAddrLen/MacAddr/IpAddrLen/IpAddr)    NextHop
 *>i      0:48:000c-292e-b4a3:32:192.168.200.2                     10.0.3.3

 VPN-Instance Edge_1_VPN_10_20, Router ID 10.0.1.1:
 Total Number of Routes: 2
       Network              NextHop          MED          LocPrf      PrefVal Path/Ogn

 *>i   192.168.100.2/32     10.0.3.3                      100          0       ?
 *>i   192.168.200.2/32     10.0.3.3                      100          0       ?
```

The command output shows a Type 2 route from 10.0.3.3, which contains the host MAC address (000c-292e-b4a3) and host IP address (192.168.200.2), with the RD value being 23:20.

# Check detailed information about the Type 2 route 0:48:000c-292e-b4a3:32:192.168.200.2.

```
<Edge_1>display bgp evpn all routing-table mac-route 0:48:000c-292e-b4a3:32:192.168.200.2

 BGP local router ID : 10.0.1.1
 Local AS number : 100
```

```
 Total routes of Route Distinguisher(23:20): 1
 BGP routing table entry information of 0:48:000c-292e-b4a3:32:192.168.200.2:
 Label information (Received/Applied): 2000 1020/NULL
 From: 10.0.3.3 (10.0.3.3)
 Route Duration: 00h52m19s
 Relay IP Nexthop: 10.0.12.2
 Relay IP Out-Interface: Vlanif12
 Original nexthop: 10.0.3.3
 Qos information : 0x0
 Ext-Community:RT <213 : 20>, RT <313 : 12>,
                    Tunnel Type <VxLan(8)>, MAC Mobility <flag:0 seq:3 res:0>,
                    Router's MAC <80e1-bf0e-5650>
 AS-path Nil, origin incomplete, localpref 100, pref-val 0, valid, internal, best, select, pre 255, IGP cost
2
 Route Type: 2 (MAC Advertisement Route)
 Ethernet Tag ID:0, MAC Address/Len: 000c-292e-b4a3/48, IP Address/Len: 192.168.200.2/32, ESI:
0000.0000.0000.0000.0000
 Not advertised to any peer yet
```

The command output shows that the route carries the L2VNI and L3VNI, as well as the export RT bound to the EVPN instances on Edge_2. After receiving the route, Edge_1 compares the RT with the import RT values of the local EVPN instance and IP VPN instance. The RT of the EVPN instance and IP VPN instance (Edge_1_VPN_10_20) bound to BD 20 is matched. Because the route carries the host IP address and host MAC address, Edge_1 adds the host MAC address to the MAC address table of BD 20 and adds the host IP address to the routing table of the corresponding IP VPN instance.

By controlling the RT value, Edge_1 learns the MAC address entries and IP routes through Type 2 IRB routes.

In this case, the data frame for communication between PC1 and PC2 is encapsulated by Edge_1. The VNI carried in the VXLAN header is 1020, which is used to inform Edge_2 of the IP VPN instance of packets. The destination MAC address of the inner data frame is the router MAC address carried by BGP EVPN routes. Similar to the bridge MAC address of the device, the MAC address is used to identify the device itself.

# Check the corresponding routing table of IP VPN instance Edge_2_VPN_10_20 on Edge_2.

```
[Edge_2]display ip routing-table vpn-instance Edge_2_VPN_10_20
Route Flags: R - relay, D - download to fib, T - to vpn-instance
------------------------------------------------------------------------------
Routing Tables: Edge_2_VPN_10_20
         Destinations : 5        Routes : 5

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

 192.168.100.0/24   Direct  0    0           D   192.168.100.254 Vbdif10
 192.168.100.1/32   IBGP    255  0           RD  10.0.1.1        VXLAN
192.168.100.254/32  Direct  0    0           D   127.0.0.1       Vbdif10
 192.168.200.0/24   Direct  0    0           D   192.168.200.254 Vbdif20
192.168.200.254/32  Direct  0    0           D   127.0.0.1       Vbdif20
```

After receiving the packet from Edge_1, Edge_2 checks the VNI in the VXLAN header and the destination MAC address in the inner data frame, and finds that it needs to search the local routing table before Layer 3 forwarding. Then, Edge_2 searches the routing table in the IP VPN instance corresponding to VNI 1020 and finds a direct route.

# Check the ARP and MAC address tables on Edge_2.

```
[Edge_2]display arp all | in 192.168.200.2
IP ADDRESS      MAC ADDRESS     EXPIRE(M) TYPE        INTERFACE     VPN-INSTANCE
                                VLAN/CEVLAN(SIP/DIP)
--------------------------------------------------------------------------------
192.168.200.254 0000-5e00-0120             I -         Vbdif20       Edge_2_VPN_10_20
192.168.200.2   000c-292e-b4a3  4          D-0         GE0/0/24.200  Edge_2_VPN_10_20
--------------------------------------------------------------------------------
Total:7         Dynamic:3       Static:0      Interface:4
```

```
[Edge_2]display mac-address 000c-292e-b4a3
--------------------------------------------------------------------------------
MAC Address     VLAN/VSI/BD                Learned-From          Type
--------------------------------------------------------------------------------
000c-292e-b4a3  -/-/20                     GE0/0/24.200          dynamic

--------------------------------------------------------------------------------
Total items displayed = 1
```

According to the search result, the data frame is re-encapsulated and sent out from the sub-interface GE0/0/24.200 to ACC_2 for processing.

In this case, the destination MAC address of the data frame is the MAC address of PC2.

By now, the packet forwarding process from PC1 to PC2 is complete. During the entire process, both the ingress VTEP (Edge_1) and egress VTEP (Edge_2) search the Layer 3 routing table and forward the packets based on the search result. When packets are forwarded between VTEPs, the VNI carried in the inner data frame is the L3VNI.

**----End**

## 5.3.3 Quiz

In distributed gateways, whether Layer 3 forwarding between VTEPs can be implemented during communication in the same network segment?

# 6 VXLAN-based Virtualized Campus Network Deployment

## 6.1 VXLAN-based Virtualized Campus Network Deployment

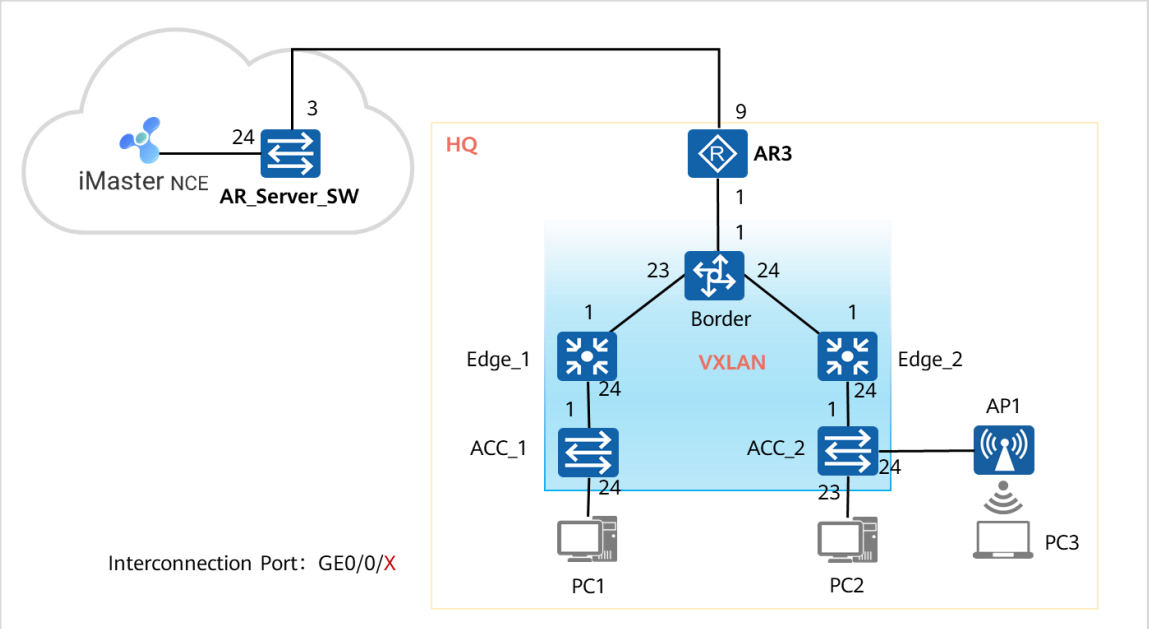
### 6.1.1 About This Lab

#### 6.1.1.1 Objectives

- Master the process of creating sites and onboarding devices.
- Master the process of creating a fabric network, deploying resources, and automatically deploying an underlay network.
- Master the process of deploying an overlay network.
- Master the process of deploying free mobility and access control.
- Master the process of deploying WLAN services.

#### 6.1.1.2 Networking Description

**Figure 6-1 VXLAN-based virtualized campus network deployment**




As shown in the figure, the network consists of two parts: HQ and cloud.

**HQ**: ACC_1 and ACC_2 function as access devices that connect to wired terminals and provide network services for wired users. AP1 is connected to ACC_2 to provide network services for wireless users. Edge_1 and Edge_2 serve as aggregation devices, and Border functions as the core device. AR3 works as both the campus egress and the DHCP server, which allocates IP addresses to other devices and user terminals at the HQ.

**Cloud**: AR_Server_SW is used to simulate the cloud. It connects the HQ and iMaster NCE, and also functions as the gateway of iMaster NCE.

**VXLAN network**: The network topology for this lab uses a distributed gateway model, in which VXLAN is deployed across core and aggregation layers. Edge_1 and Edge_2 function as the edge nodes of the VXLAN network, whereas Border functions as the border node of the VXLAN network.

After completing this lab, you can use the controller to centrally manage networks and services and deliver configurations to devices.

This lab is relevant to the WAN deployment lab. Restore the environment after completing both this lab and the WAN deployment lab.

## 6.1.1.3 Lab Plan

This section provides the plan for all the data required in subsequent steps. It is recommended that you get familiar with the data plan.

### 6.1.1.3.1 AR3 Configuration Plan

Set an IP address for GE0/0/9 and configure a static route destined to iMaster NCE on AR3. AR3 works as the DHCP server, which allocates IP addresses to downstream devices and user terminals, and notifies downstream devices of iMaster NCE's IP address and port number through DHCP Option 148.

**Table 6-1 AR3 VLAN port type and parameter plan**

| Device | Port | Port Type | VLAN |
|---|---|---|---|
| AR3 | GE0/0/1 | Hybrid | PVID: VLAN 10<br>Tagged VLAN: 130, 140, 150<br>Untagged VLAN: 10 |

VLAN 10 is used as the PVID to respond to the DHCP requests from the switches. VLANIF 130, VLANIF 140, and VLANIF 150 are the interconnection interfaces for connecting to external networks and network service resources for VNs on Border.

**Table 6-2 IP address plan**

| Device | Port | IP Address |
|---|---|---|
| AR3 | GE0/0/9 | 65.0.0.3/24 |
| | VLANIF 10 | 172.16.10.254/24 |
| iMaster NCE | / | 172.99.0.99 |

#### 6.1.1.3.2 Device Onboarding Plan

You can add switches and APs with the same versions and equipment serial numbers (ESNs) as those in this lab to sites on iMaster NCE to manage these devices.

**Table 6-3 Device ESN**

| Device | ESN | Model | Site | Device Type | Role |
|---|---|---|---|---|---|
| Border | W02140038942 | S5731-H24T4XC | HQ | LSW | Core |
| Edge_1 | W02140014081 | S5731-H24T4XC | | LSW | Aggregation |
| Edge_2 | W02130010540 | S5731-H24T4XC | | LSW | Aggregation |
| ACC_1 | W02140038919 | S5731-H24T4XC | | LSW | Access |
| ACC_2 | DM20A9900120 | S5731-H24P4XC | | LSW | Access |
| AP1 | 2102352UBR10L6001315 | AirEngine5760-1 | | AP | AP |

Use the actual ESNs of devices in the environment.

#### 6.1.1.3.3 Campus Fabric and Underlay Network Plan

Before creating a fabric network based on a physical network, you need to configure the resources used in the fabric network, including the network resource pool (consisting of VLAN, IP address, BD, and VNI resources).

In addition, you need to plan external networks, network service resources, and templates (including server templates and authentication profiles), and they will be used to create VNs and configure access management.

The network resource pool plan is as follows.

**Table 6-4 Plan for the global resource pool on the fabric network**

| Network Resource | Value | Description |
|---|---|---|
| VLAN | 100-300 | Service VLAN ID pool, including the VLANs for connecting to external networks, VLANs for connecting to network service resources, policy association management VLAN, and access VLANs of terminals. |
| Bridge Domain (BD) | 1-1000 | Each BD is identified by a BD ID. The controller automatically selects a BD ID when delivering configurations to a device. |
| VXLAN Network Identifier (VNI) | 1-1000 | VXLAN network identifier, which is used to distinguish VXLAN network isolation domains. |

**Table 6-5 Plan for the automation resource pool on the underlay network**

| Resource | Value | Description |
| --- | --- | --- |
| Interconnection VLAN | 10-20 | Used for the interconnection between the border and edge nodes on the fabric network. |
| Interworking IP address | 172.20.0.0-172.20.0.0/16 | Used for the interconnection between the border and edge nodes on the fabric network. |
| Loopback interface IP | 3.3.3.0-3.3.3.0/24 | IP address of a loopback interface.<br>Loopback interface IP addresses are used to establish BGP EVPN peer relationships when the underlay routing domain is automatically configured and the fabric network is automatically connected to network service resources. |

The policy template plan is as follows (including authentication templates and servers). The plan is used to perform access authentication for users.

**Table 6-6 Plan for the RADIUS server template**

| Parameter | Value |
| --- | --- |
| Name | RADIUS (customizable) |
| Use the built-in server | Enabled |
| Authentication component | Built-in authentication service |
| Key | Huawei@123 (customizable) |

**Table 6-7 Plan for the Portal server template**

| Parameter | Value |
| --- | --- |
| Name | Portal (customizable) |
| Use the built-in server | Enabled |
| Page push protocol | HTTPS |
| Authentication component | Built-in authentication service |
| Key | Huawei@123 (customizable) |

**Table 6-8 Plan for the 802.1X and MAC address authentication profile**

| Parameter | Value |
|---|---|
| Name | MAC_802.1X (customizable) |
| Authentication mode | MAC, 802.1X |
| RADIUS server template | RADIUS (using a created template) |

**Table 6-9 Plan for the Portal authentication profile**

| Parameter | Value |
|---|---|
| Name | Portal |
| Authentication mode | Portal |
| RADIUS server template | RADIUS (using a created template) |
| Primary portal server template | Portal (using a created template) |

A fabric network based on the campus network will be built in this lab. VXLAN is deployed across core and aggregation switches on the fabric network, and the distributed VXLAN gateway networking is used.

The plan for a fabric network is as follows.

**Table 6-10 Plan for a fabric network**

| Parameter | Value |
|---|---|
| Name | HQ (customizable) |
| Networking type | Distributed gateway |
| Wireless WAC location | Border (the border node) |
| Automatic routing domain configuration | Enabled |
| RR cluster ID | 1 |

**Table 6-11 Role plan for devices on the fabric network**

| Device | Role |
|---|---|
| Border | Border |
| Edge_1 | Edge |

| Device | Role |
| --- | --- |
| Edge_2 | Edge |
| ACC_1 | Extended |
| ACC_2 | Extended |
| Configuration Switch | Enabled |
| Domain | Single area (customizable) |
| Encryption | None (customizable) |

In this lab, create two VNs based on the campus network: the OA network (for non-R&D office) and RD network (for R&D office). Then create two external networks for users on the preceding VNs to access external networks.

Moreover, configure DHCP network service resources for users on the preceding VNs to obtain IP addresses.

Finally, configure an external server whose type is other and use the server to simulate an E-mail server.

The external networks and network service resource plans are as follows.

**Table 6-12 Plan for the external network of the OA network**

| Parameter | Value |
| --- | --- |
| Connection between the fabric and external network | L3 exclusive egress |
| Name | OA (customizable) |
| Internet connection | Enabled |
| Border device | Border |
| Interconnection port | GigabitEthernet0/0/1 |
| VLAN | 130 |
| IP address type | IPv4 |
| Local IPv4 address | 13.1.1.2 |
| Remote IPv4 address | 13.1.1.1 |
| IPv4 address mask | 30 |

**Table 6-13 Plan for the external network of the RD network**

| Parameter | Value |
|---|---|
| Connection between the fabric and external network | Layer 3 exclusive egress |
| Name | RD (customizable) |
| Internet connection | Enabled |
| Border device | Border |
| Interconnection port | GigabitEthernet0/0/1 |
| VLAN | 140 |
| IP address type | IPv4 |
| Local IPv4 address | 14.1.1.2 |
| Remote IPv4 address | 14.1.1.1 |
| IPv4 address mask | 30 |

**Table 6-14 Plan for the network service resource (the DHCP server and other server)**

| Parameter | Value |
|---|---|
| Name | DHCP_Email (customizable) |
| Server type | DHCP or other |
| DHCP server | 192.168.150.1 |
| Other server | 172.17.3.3/32 |
| Server interconnection address pool | / |
| Scenario | Directly connect to a switch |
| Interconnection device | Border |
| Interconnection port | GigabitEthernet0/0/1 |
| Interconnection VLAN | 150 |
| Interconnection IPv4 address | 192.168.150.2 |
| Peer IPv4 address | 192.168.150.1 |
| Mask | 30 |

In this lab, the 802.1X or MAC address authentication will be performed on wired users, and the Portal authentication will be performed on the wireless users. Only wired users are connected to ACC_1 and both wired and wireless users are connected to ACC_2.

The plan for access management is as follows.

**Table 6-15 Plan for access management on Edge_1**

| Parameter | Value |
|---|---|
| Authentication Control Point | Edge_1 |
| Authentication Control Point Management Parameter | |
| Management VLAN of CAPWAP | 160 |
| Management IP address of CAPWAP | 172.16.16.1/23 |
| Port Name | GigabitEthernet0/0/24 |
| Connected Device Type | Extended access switch |
| Authentication Template | MAC_802.1X |
| Authentication enforcement point | |
| Device Name | ACC_1 |
| Port Name | GigabitEthernet0/0/24 |
| Connected Device Type | Terminal (PC, phone, dumb terminal, non-fabric extended switch/AP) |
| Inherit Authentication Template on Authentication Control Point Port | Yes |

**Table 6-16 Plan for access management on Edge_2**

| Parameter | Value |
|---|---|
| Authentication Control Point | Edge_2 |
| Authentication Control Point Management Parameter | |
| Management VLAN of CAPWAP | 180 |
| Management IP address of CAPWAP | 172.16.18.1/23 |
| Port Name | GigabitEthernet0/0/24 |
| Connected Device Type | Extended access switch |

| Parameter | Value |
| --- | --- |
| Authentication Template | MAC_802.1X |
| Authentication enforcement point | |
| Device Name | ACC_1 |
| Port Name | GigabitEthernet0/0/23 |
| Connected Device Type | Terminal (PC, phone, dumb terminal, non-fabric extended switch/AP) |
| Inherit Authentication Template on Authentication Control Point Port | Yes |
| Port Name | GigabitEthernet0/0/24 |
| Connected Device Type | Terminal (PC, phone, dumb terminal, non-fabric extended switch/AP) |
| Inherit Authentication Template on Authentication Control Point Port | No |

#### 6.1.1.3.4 Overlay Network Automation Plan

Create two VNs (OA and RD) and associate them with the created external networks. Ensure that users on the two VNs can obtain IP addresses from external DHCP servers.

The plan for the VNs is as follows.

**Table 6-17 Data plan for the OA network**

| Parameter | Value |
| --- | --- |
| Creating a VN | |
| Name | OA (customizable) |
| User gateway location | Inside the fabric |
| User-defined VRF name | OA |
| External network | OA (use the created external network) |
| Network service resources | DHCP_E-mail (use the created network service resources) |
| Manually Specified | |
| User gateway 1 | |
| Name | Sales_Wired (customizable) |

| Parameter | Value |
| --- | --- |
| VLAN Type | Dynamic VLAN |
| VLAN | 100 |
| Subnet | 172.17.10.0/24 |
| Gateway Address | 172.17.10.254 |
| User gateway 2 | |
| Name | Market_Wired (customizable) |
| VLAN Type | Dynamic VLAN |
| VLAN | 200 |
| Subnet | 172.17.20.0/24 |
| Gateway Address | 172.17.20.254 |
| User gateway 3 | |
| Name | Sales_Wireless (customizable) |
| VLAN Type | Dynamic VLAN |
| VLAN | 110 |
| Subnet | 172.17.11.0/24 |
| Gateway Address | 172.17.11.254 |
| User gateway 4 | |
| Name | Market_Wireless (customizable) |
| VLAN Type | Dynamic VLAN |
| VLAN | 210 |
| Subnet | 172.17.21.0/24 |
| Gateway Address | 172.17.21.254 |
| Wired access | |
| Name | OA (customizable) |
| Device name: ACC_1 | Port: GigabitEthernet0/0/24<br>Service VLAN: Dynamic VLAN |
| Device name: ACC_2 | Port: GigabitEthernet0/0/23<br>Service VLAN: Dynamic VLAN |

| Parameter | Value |
| --- | --- |
| Wireless access | |
| Site | HQ |
| Device Name | Border |

**Table 6-18 Data plan for the RD network**

| Parameter | Value |
| --- | --- |
| Creating a VN | |
| Name | RD (customizable) |
| User gateway location | Inside the fabric |
| User-defined VRF name | RD |
| External network | RD (use the created external network) |
| Network service resources | DHCP_E-mail (use the created network service resources) |
| Configuring a user gateway | |
| Name | RD (customizable) |
| VLAN Type | Dynamic VLAN |
| VLAN | 300 |
| Subnet | 172.17.30.0/24 |
| Gateway Address | 172.17.30.254 |
| Wired access | |
| Name | RD (customizable) |
| Device name: ACC_1 | Port: GigabitEthernet0/0/24<br>Service VLAN: Dynamic VLAN |
| Device name: ACC_2 | Port: GigabitEthernet0/0/23<br>Service VLAN: Dynamic VLAN |

The plan for communication between VNs is as follows.

**Table 6-19 Plan for communication between VNs**

| Parameter | Value |
| --- | --- |
| Name | OA_to_RD (customizable) |
| Interconnection device | Border |
| Interworking mode | Partial interwork |
| IP address type | IPv4 |
| Source VN | OA |
| Source IPv4 Prefix | Market_Wired(172.17.20.0/24) |
| Destination VN | RD |
| Destination IPv4 Prefix | RD(172.17.30.0/24) |

| Parameter | Value |
| --- | --- |
| RD user group | |
| Username | kris (customizable) |
| Password | Huawei@123 (customizable) |
| Change password upon next login | Disabled |
| Available login mode | 802.1X |
| Sales user group | |
| Username | sales (customizable) |
| Password | Huawei@123 (customizable) |
| Change password upon next login | Disabled |
| Available login mode | 802.1X, Portal |
| Market user group | |
| Username | market (customizable) |
| Password | Huawei@123 (customizable) |
| Change password upon next login | Disabled |
| Available login mode | 802.1X, Portal |

#### 6.1.1.3.5 Free Mobility and Access Authentication Plan

Create five security groups (Sales_Wired, Sales_Wireless, Market_Wired, Market_Wireless, and RD). They respectively indicate the wired sales users, wireless sales users, wired marketing users, wireless marketing users, and R&D users on the OA and RD networks.

The plan for access authentication is as follows.

**Table 6-20 Plan for user authentication accounts**

| Parameter | Value |
|---|---|
| RD user group | |
| Username | kris (customizable) |
| Password | Huawei@123 (customizable) |
| Change password upon next login | Disabled |
| Available login mode | 802.1X |
| Sales user group | |
| Username | sales (customizable) |
| Password | Huawei@123 (customizable) |
| Change password upon next login | Disabled |
| Available login mode | 802.1X, Portal |
| Market user group | |
| Username | market (customizable) |
| Password | Huawei@123 (customizable) |
| Change password upon next login | Disabled |
| Available login mode | 802.1X, Portal |

**Table 6-21 Plan for the authorization result**

| Parameter | Value |
|---|---|
| Sales_Wired (802.1X authentication) | |
| Name | Sales_Wired_Result (customizable) |
| Security Group | Sales_Wired (use the created security group) |

| Parameter | Value |
| --- | --- |
| VLAN | 100 |
| Sales_Wireless (Portal authentication) | |
| Name | Sales_Wireless_Result (customizable) |
| Security Group | Sales_Wireless (use the created security group) |
| VLAN | 110 |
| Market_Wired (802.1X authentication) | |
| Name | Market_Wired_Result (customizable) |
| Security Group | Market_Wired (use the created security group) |
| VLAN | 200 |
| Market_Wireless (Portal authentication) | |
| Name | Market_Wireless_Result (customizable) |
| Security Group | Market_Wireless (use the created security group) |
| VLAN | 210 |
| RD (802.1X authentication) | |
| Name | RD_Result (customizable) |
| Security Group | RD (use the created security group) |
| VLAN | 300 |

**Table 6-22 Plan for the authorization rule**

| Parameter | Value |
| --- | --- |
| Sales_Wired (802.1X authentication) | |
| Name | Sales_Wired_Rule (customizable) |
| Authentication mode | User access authentication |
| Portal-HACA | Disabled |
| Access mode | Wired |
| Match user groups | Enabled<br>Choosing the Sales user group |
| Authorization result | Sales_Wired_Result |

| Parameter | Value |
| --- | --- |
| Sales_Wireless (Portal authentication) | |
| Name | Sales_Wireless_Rule (customizable) |
| Authentication mode | User access authentication |
| Portal-HACA | Disabled |
| Access mode | Wi-Fi |
| Match user groups | Enabled<br>Choosing the Sales user group |
| Match SSIDs | Sales |
| Authorization result | Sales_Wireless_Result |
| Market_Wired (802.1X authentication) | |
| Name | Market_Wired_Rule (customizable) |
| Authentication mode | User access authentication |
| Portal-HACA | Disabled |
| Access mode | Wired |
| Match user groups | Enabled<br>Choosing the Market user group |
| Authorization result | Market_Wired_Result |
| Market_Wireless (Portal authentication) | |
| Name | Market_Wireless_Rule (customizable) |
| Authentication mode | User access authentication |
| Portal-HACA | Disabled |
| Access mode | Wi-Fi |
| Match user groups | Enabled<br>Choosing the Market user group |
| Match SSIDs | Market |
| Authorization result | Market_Wireless_Result |
| RD (802.1X authentication) | |
| Name | RD_Rule (customizable) |

| Parameter | Value |
| --- | --- |
| Authentication mode | User access authentication |
| Portal-HACA | Disabled |
| Access mode | Wired |
| Match user groups | Enabled<br>Choosing the RD user group |
| Authorization result | RD_Result |

### 6.1.1.3.6 WLAN Service Plan

To onboard the AP and wireless users, complete the WLAN service configurations including AP onboarding configurations: the PnP configuration, subnet configuration on the Border side (the configuration of DHCP Option 148 and controller's address), and CAPWAP source address configuration.

**Table 6-23 AP onboarding configuration**

| Parameter | Value |
| --- | --- |
| Position of the native AC | Border |
| AP onboarding mode | Configuring an independent PnP VLAN |
| Management VLAN for the AP | 2 |
| Management network segment for the AP | 172.16.20.0/24 |
| Address of the AC | 172.16.20.254 |
| Mode in which an AP obtains the address of the controller | Negotiating the address of the controller automatically |

**Table 6-24 Wireless authentication configuration on the controller**

| Parameter | Value |
| --- | --- |
| Name: Sales | |
| SSID | Sales |
| Authentication mode | Open network (Portal authentication) |
| Page pusher | Built-in authentication by cloud platform |
| Portal protocol | Portal 2.0 |

| Parameter | Value |
| --- | --- |
| Primary portal server | Portal |
| RADIUS server | RADIUS |
| Push mode | Fast |
| Push page | Default username and password authentication page |
| Portal authentication-free | Enabled |
| Bypass policy | Enabled by default |
| Device | Border |
| Name: Market | |
| SSID | Market |
| Authentication mode | Open network |
| Page pusher | Built-in authentication of the cloud platform |
| Portal protocol | Portal 2.0 |
| Primary portal server | Portal |
| RADIUS server | RADIUS |
| Push mode | Fast |
| Push page | Default user name and password authentication customization page |
| Portal authentication-free | Enabled |
| Bypass policy | Enabled by default |
| Device | Border |

**Table 6-25 WLAN service plan**

| Parameter | Value |
| --- | --- |
| Service name: Sales_Wireless | |
| Service VLAN | 110 |
| SSID profile | Name: Sales (customizable)<br>SSID name: Sales |
| VAP profile | Name: Sales (customizable)<br>Service VLAN: VLAN 110 |

| Parameter | Value |
| --- | --- |
| | Forwarding mode: tunnel forwarding<br>Referenced profiles: SSID profile "Sales", with a profile of the controller being used as the authentication profile |
| Service name: Market_Wireless | |
| Service VLAN | 210 |
| SSID profile | Name: Market (customizable)<br>SSID name: Market |
| VAP profile | Name: Market (customizable)<br>Service VLAN: VLAN 210<br>Forwarding mode: tunnel forwarding<br>Referenced profiles: SSID profile "Market", and use a profile of the controller as the authentication profile. |

## 6.1.2 Configuration for Lab Tasks

### 6.1.2.1 Configuration Roadmap

1. Complete pre-configuration, so that AR3 and the controller can communicate with each other. Complete NAT configuration (used for device onboarding), configure AR3 as a DHCP server for device onboarding and end users, configure the simulated external network, and create a loopback interface for external E-mail server simulating. Ensure that there are reachable routes between AR3 and the external network.
2. Create a site named HQ, and ensure that all the devices can obtain the controller's IP address through the DHCP service of AR3 and communicate with the controller through the NAT service of AR3.
3. After onboarding the device, configure a global fabric resource pool and an underlay resource pool according to the plan.
4. Configure policy templates, including the RADIUS server template, Portal server template and authentication profiles (Portal and 802.1X authentication).
5. Configure a fabric network using distributed networking, deploy VXLAN across core and aggregation layers, and configure Border as the native AC to manage APs.
6. Configure external networks OA and RD for VNs (to be created).
7. Configure network server resources, including a DHCP server and a server whose type being other.
8. Configure access management, and set **Connected Device Type** to **Extended access switch**.
9. Create VNs OA and RD. Create user gateways for wireless and wired users of the market (Market) and sales department (Sales) on the OA network. Create user

gateways only for wired users on the RD network and associate OA and RD networks with external networks and the corresponding network resource servers.

10. Configure communication between VNs in partial interconnection mode, and ensure that there are reachable routes between Market and RD networks.
11. Free mobility configuration: Configure security groups for wireless and wired users of Market and Sales on the OA network and for the wired users on the RD network. Then create a resource group E_Mail (correspond to the server whose type is other in network server resources). Create a policy matrix to ensure that wired users in Market can access the RD network, but users on the RD network cannot access E_Mail.
12. Access authentication: create user groups respectively for the market department (Market), sales department (Sales) and RD department (RD), create authentication results and rules for wireless and wired users of Market and Sales on the OA network and for the wired users on the RD network, create the Portal authentication exemption policy and configure Portal perception-free reauthentication for wireless users in Market and Sales.
13. WLAN services: Configure Border as the native AC, as well as create different SSIDs and configure different service VLANs for wireless users in Market and Sales.

## 6.1.2.2 (Optional) Initialization

Before the lab, restore the factory settings of iMaster NCE and network devices (Border, Edge_1, Edge_2, ACC_1, and ACC_2).

Note: Delete the configurations of iMaster NCE and then those of network devices.

For details, see **错误!未找到引用源。**"**错误!未找到引用源。**."

Note: If the lab environment has been initialized, skip this step.

## 6.1.2.3 Pre-configuration for AR3

AR3 functions as the egress router of the HQ campus, so it should be pre-configured first to interconnect with other devices in the campus. The pre-configurations include: VLAN, VLANIF, IP routing, DHCP server, external networks, LLDP, and NAT configurations.

### 6.1.2.3.1 Pre-configuration for the DHCP Server (Used for Device Plug and Play)

AR3 functions as a DHCP server to allocate management IP addresses to the switches at the HQ site. In addition, AR3 needs to notify the switches of the iMaster NCE's IP address and port information.

During the pre-configuration, VLAN 10 is used for the interconnection between AR3 and Border, and the downstream switch of AR3 will obtain an IP address from the interface address pool of VLANIF 10 and obtain the IP address and port information of iMaster NCE.

Step 1 Configure a VLAN.

# Create VLAN 10 on AR3.

```
[AR3]vlan 10
```

# Configure the interface connecting AR3 to Border as a hybrid interface, add it to VLAN 10 in untagged mode, and set PVID to 10.

```
[AR3]interface GigabitEthernet 0/0/1
[AR3-GigabitEthernet0/0/1] portswitch
[AR3-GigabitEthernet0/0/1] port link-type hybrid
[AR3-GigabitEthernet0/0/1] port hybrid pvid vlan 10
[AR3-GigabitEthernet0/0/1] undo port hybrid vlan 1
[AR3-GigabitEthernet0/0/1] port hybrid untagged vlan 10
```

Step 2 Perform DHCP-related operations.

Enable the DHCP service globally on AR3, configure an IP address for VLANIF 10, select an interface address pool, and configure DHCP static binding entries and Option 148.

# Enable the DHCP service globally.

```
[AR3]dhcp enable
```

# Create VLANIF 10 on AR3 and configure an IP address for it.

```
[AR3]interface Vlanif10
[AR3-Vlanif10] ip address 172.16.10.254 255.255.255.0
[AR3-Vlanif10] dhcp select interface
[AR3-Vlanif10] dhcp server static-bind ip-address 172.16.10.1 mac-address 9400-b049-9ef2
[AR3-Vlanif10] dhcp server static-bind ip-address 172.16.10.2 mac-address d446-4982-348d
[AR3-Vlanif10] dhcp server static-bind ip-address 172.16.10.3 mac-address 80e1-bf0e-5652
[AR3-Vlanif10] dhcp server static-bind ip-address 172.16.10.4 mac-address 9400-b049-9d82
[AR3-Vlanif10] dhcp server static-bind ip-address 172.16.10.5 mac-address b008-750e-fbd0
[AR3-Vlanif10] dhcp server option 148 ascii agilemode=agile-cloud;agilemanage-mo
de=ip;agilemanage-domain=172.99.0.99;agilemanage-port=10020;
```

Configure the device to allocate IP addresses to downstream devices from the interface address pool and notify the devices of the controller's IP address through Option 148.

Binding static IP addresses to downstream devices is optional. You can view the MAC address of the VLANIF interface on a downstream device and use the address as a binding MAC address.

### 6.1.2.3.2 Pre-configuration for the DHCP Server (Used for Allocating IP Addresses to Hosts)

AR3 functions as a DHCP server to allocate IP addresses to hosts at the HQ site.

Step 1 Create a DHCP address pool.

Create DHCP address pools for hosts on OA and RD VNs respectively.

# Create a wired network address pool for Sales.

```
[AR3]ip pool Sales_Wired
[AR3-ip-pool-Sales_Wired] gateway-list 172.17.10.254
[AR3-ip-pool-Sales_Wired] network 172.17.10.0 mask 255.255.255.0
[AR3-ip-pool-Sales_Wired] quit
```

# Create a wireless network address pool for Sales.

```
[AR3]ip pool Sales_Wireless
[AR3-ip-pool-Sales_Wireless] gateway-list 172.17.11.254
[AR3-ip-pool-Sales_Wireless] network 172.17.11.0 mask 255.255.255.0
[AR3-ip-pool-Sales_Wireless] quit
```

# Create a wired network address pool for Market.

```
[AR3]ip pool Market_Wired
[AR3-ip-pool-Market_Wired] gateway-list 172.17.20.254
[AR3-ip-pool-Market_Wired] network 172.17.20.0 mask 255.255.255.0
[AR3-ip-pool-Market_Wired] quit
```

# Create a wireless network address pool for Market.

```
[AR3]ip pool Market_Wireless
[AR3-ip-pool-Market_Wireless] gateway-list 172.17.21.254
[AR3-ip-pool-Market_Wireless] network 172.17.21.0 mask 255.255.255.0
[AR3-ip-pool-Market_Wireless] quit
```

# Create an address pool for the RD network.

```
[AR3]ip pool RD
[AR3-ip-pool-RD] gateway-list 172.17.30.254
[AR3-ip-pool-RD] network 172.17.30.0 mask 255.255.255.0
[AR3-ip-pool-RD] quit
```

Step 2 Configure interconnection.

To allocate addresses to hosts, create a corresponding interconnection interface (VLANIF interface) on AR3 and allow the corresponding VLAN on the interface.

For configuration details, see Table 6-14.

# Create a VLAN.

```
[AR3]vlan 150
```

# Create a VLANIF interface.

```
[AR3]interface Vlanif150
[AR3-Vlanif150] ip address 192.168.150.1 255.255.255.252
[AR3-Vlanif150] dhcp select global
[AR3-Vlanif150] quit
```

Enable the interface to use the global address pool.

# Add the interconnection interface to a VLAN.

```
[AR3]interface GigabitEthernet0/0/1
[AR3-GigabitEthernet0/0/1] port hybrid tagged vlan 150
```

### 6.1.2.3.3 Pre-configuration for External Networks

Complete external network-related configurations on AR3, including the configuration of the interconnection between AR3 and Border as well as the configuration of the loopback interface simulating an E-mail Server.

# Create LoopBack1.

```
[AR3]interface LoopBack1
[AR3-LoopBack1] ip address 172.17.3.3 32
```

# Complete OA VN-related configurations.

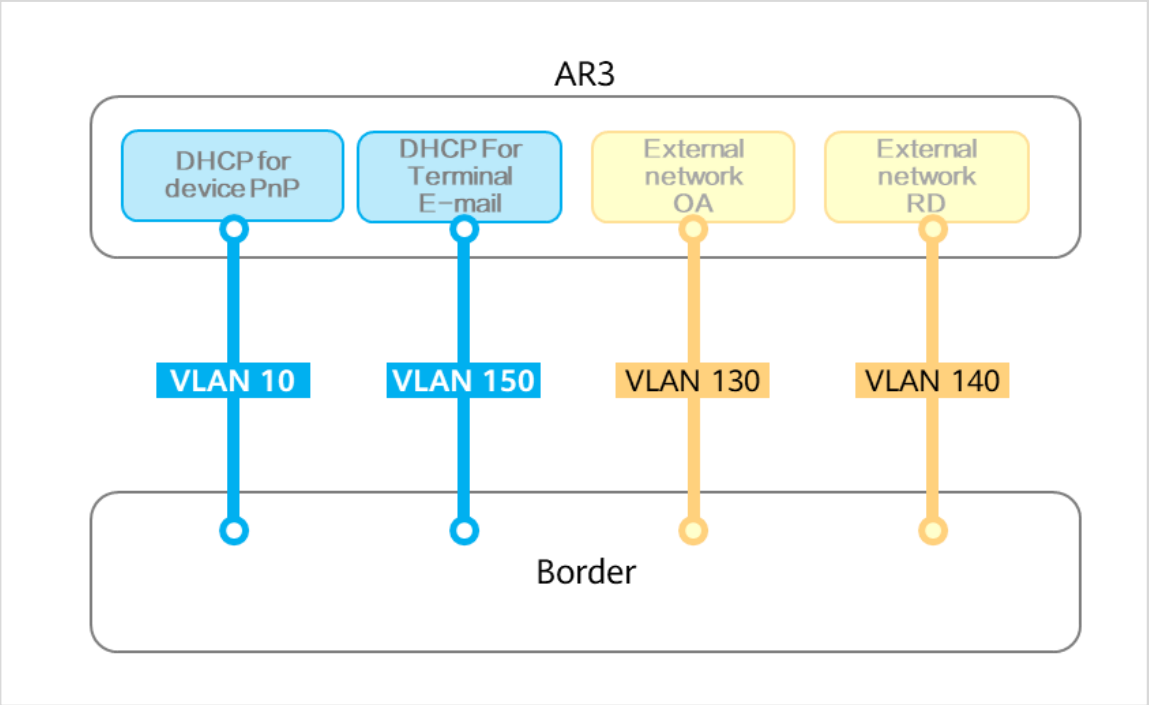
```
[AR3]vlan 130
[AR3]interface Vlanif130
[AR3-Vlanif130] ip address 13.1.1.1 255.255.255.252
[AR3-Vlanif130] quit
[AR3]interface GigabitEthernet0/0/1
[AR3-GigabitEthernet0/0/1] port hybrid tagged vlan 130
```

Create a VLANIF interface for interconnection with the OA VN and allow the corresponding VLAN on the physical interface.

# Complete RD VN-related configurations.

```
[AR3]vlan 140
[AR3]interface Vlanif140
[AR3-Vlanif140] ip address 14.1.1.1 255.255.255.252
[AR3-Vlanif140] quit
[AR3]interface GigabitEthernet0/0/1
[AR3-GigabitEthernet0/0/1] port hybrid tagged 140
```

Multiple VLANs have been used in the pre-configuration of AR3. For details about their functions, see the following figure.




### 6.1.2.3.4 Configuration for AR3 LLDP

Enable LLDP on AR3 so that iMaster NCE can discover interconnection links between the fabric network and external networks.

```
[AR3] lldp enable
```

### 6.1.2.3.5 Configuration for the connectivity of AR3

Complete the configuration for the interconnection between AR3 and external networks to enable AR3 to communicate with iMaster NCE, and to enable downstream switches to communicate with iMaster NCE through the source NAT technology.

In addition, configure return routes destined to the host's network segment on AR3.

# Configure the IP address of the interconnection interface between AR3 and the external network.

```
[AR3]interface GigabitEthernet0/0/9
[AR3-GigabitEthernet0/0/9] ip address 65.0.0.3 255.255.255.0
[AR3-GigabitEthernet0/0/9] quit
```

# Configure source NAT.

```
[AR3]acl 3000
[AR3-acl-adv-3000]rule permit ip
[AR3-acl-adv-3000]quit
[AR3]interface GigabitEthernet0/0/9
[AR3-GigabitEthernet0/0/9]nat outbound 3000
```

# Configure return routes destined to the host's network segment on the OA network.

```
[AR3]ip route-static 172.17.10.0 255.255.255.0 13.1.1.2
```

```
[AR3]ip route-static 172.17.11.0 255.255.255.0 13.1.1.2
[AR3]ip route-static 172.17.20.0 255.255.255.0 13.1.1.2
[AR3]ip route-static 172.17.21.0 255.255.255.0 13.1.1.2
```

# Configure a return route destined to the host's network segment on the RD network.

```
[AR3]ip route-static 172.17.30.0 255.255.255.0 14.1.1.2
```

# Configure the default route of AR3 destined to external networks.

```
[AR3]ip route-static 0.0.0.0 0.0.0.0 65.0.0.254
```

## 6.1.2.4 Creating a Site and Onboarding Devices

In actual projects, there are a large number of switches. After being powered on, the switches need to be initialized and configured with basic functions such as IP connectivity. This initialization process is called deployment. Traditional deployment requires a large number of manual operations, which is inefficient. The CloudCampus Solution supports Zero Touch Provisioning (ZTP) of network devices — also known as device plug-and-play. With this function, the devices will be automatically deployed after they are powered on and connected to the network with factory settings, greatly reducing O&M and management costs.

Currently, two deployment scenarios are supported: onboarding before configuration and configuration before onboarding. This document describes the first one.

Step 1 Obtain the ESN.

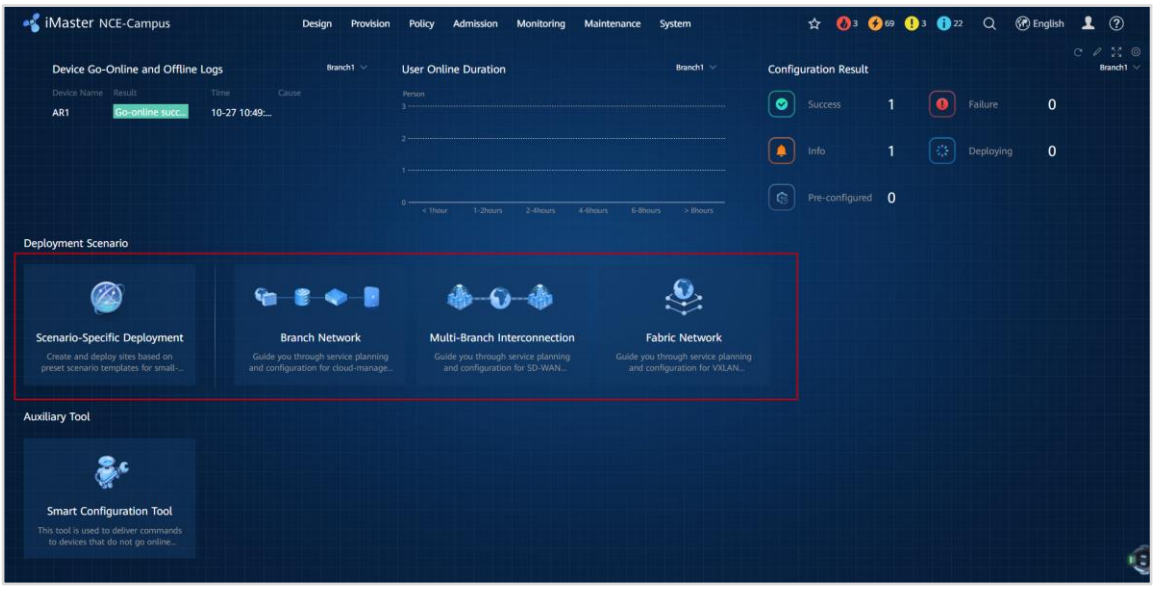
Check the ESNs of switches and APs based on the practice environment. This lab does not describe how to log in to a switch or AP and view the ESN. For details, see the product documentation of the corresponding device.

Step 2 Log in to iMaster NCE to start deployment.

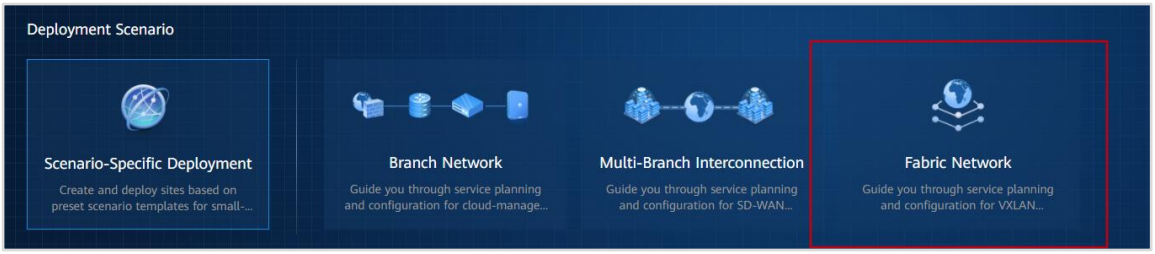
The home page is displayed as shown in the following figure.

Pay attention to the two areas in red boxes. The items in the navigation area on the top cover full lifecycle management of the campus network. To manage networks or provision services, choose corresponding items to access the configuration page.

To simplify operations, iMaster NCE provides a configuration navigation oriented to typical deployment scenarios. For example, by performing operations in **Deployment Scenario** > **Fabric Network**, you can complete the service plan and configuration of the VXLAN-based campus network quickly. By using the items in the navigation area in combination, you can also complete the VXLAN-related plan and configuration.

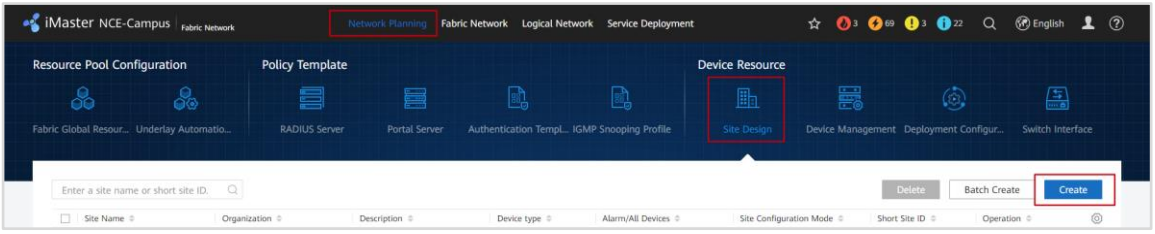



# Choose **Deployment Scenario** > **Fabric Network** on the home page to access the fabric network configuration page, as shown in the following figure.




Step 3 Create a site and add devices.

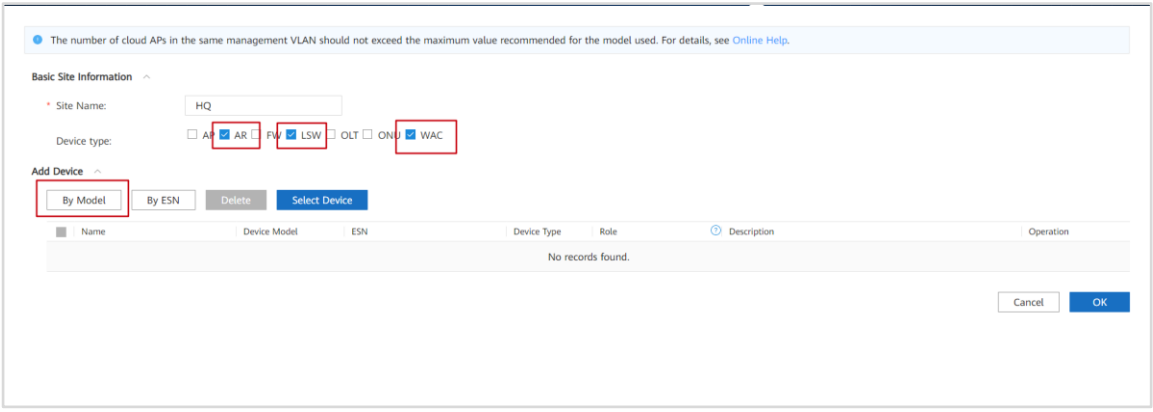
# Create a site.




Choose **Network Planning** > **Site Design** to access the site design page. Click **Create** on this page to create a site and add devices.

To ensure that iMaster NCE can manage the devices on the campus network successfully, create a site and add devices at first. The devices added here should correspond to physical devices on the campus network. You can add devices one by one manually, or import devices in batches.

# Select device types.

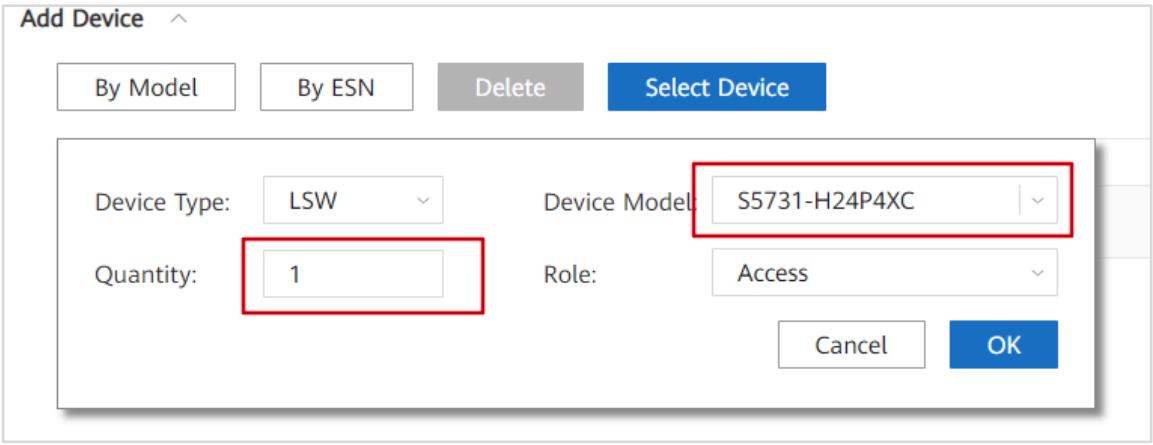


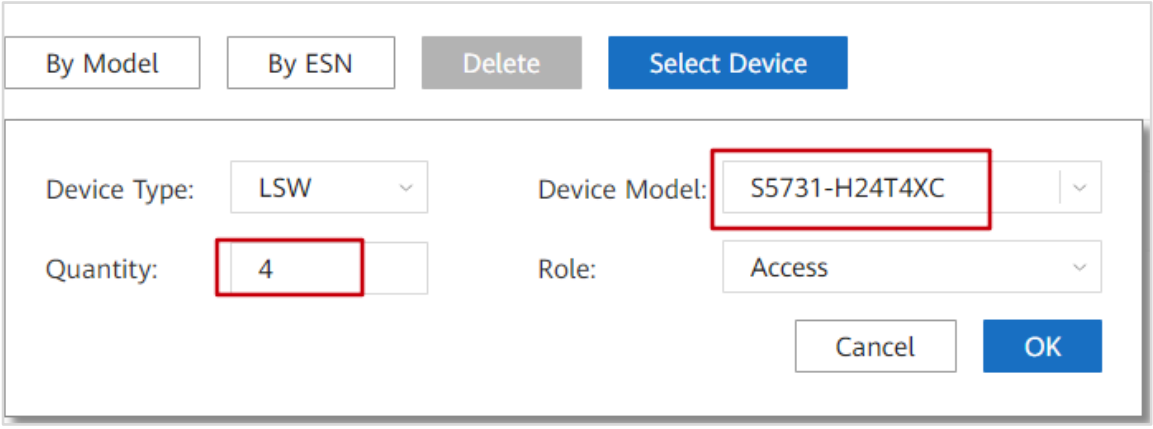

Note that **WAC** and **LSW** are mandatory in **Device type**. You are advised to select **AR** for the SD-WAN lab in future.

Devices can be added by setting ESNs or device models.

S5731-H24P4XC and S5731-H24T4XC in this lab should be added by setting device models.

# Add S5731-H24P4XC and S5731-H24T4XC respectively by setting device models.




Do not set **Role** when adding S5731-H24T4XC.

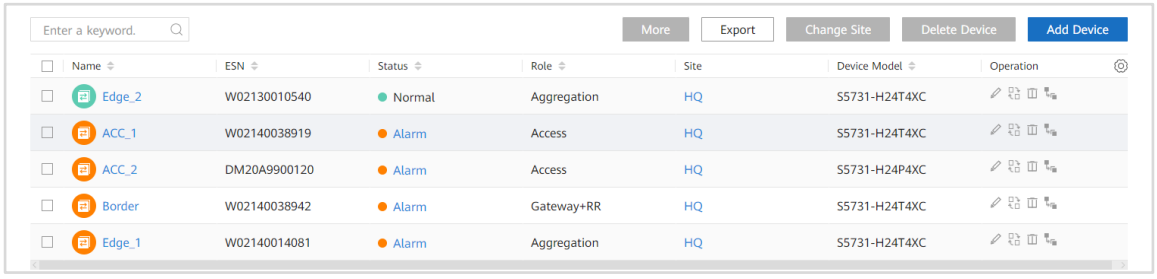
# Set **Name**, **Role**, and **ESN**.


Add information according to the plan.

Ensure that configurations of all switches are cleared, and the switches are restarted. Do not run any command during the startup phase, and you can run commands only after the device onboarding.

Step 4 Check the device registration status.

# Check whether a device is online.




Choose **Deployment Scenario** > **Fabric Network** > **Network Planning** > **Device Management** to access the device management page. The device status is normal (or an alarm exists). Because the devices have been initialized, they can successfully register with iMaster NCE after a site is created and devices are added to iMaster NCE.

Node: An AP cannot be onboarded because the access management configuration is not completed. Therefore, how to onboard an AP is skipped in this section.

## 6.1.2.5 Campus Fabric and Underlay Network Automation

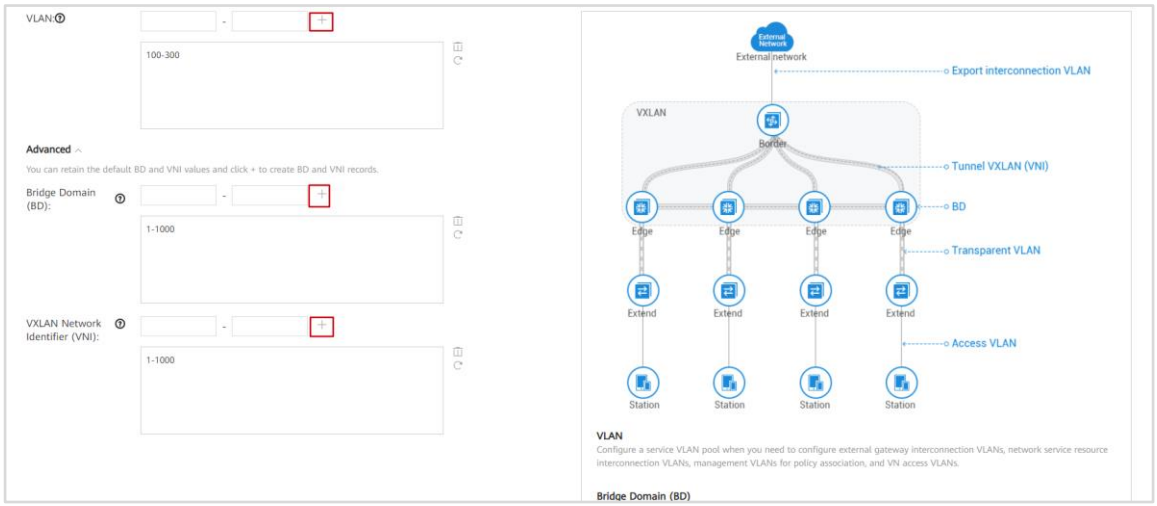
### 6.1.2.5.1 Configuration for Network Resources and Policy Templates

In this section, we will complete configurations of **Policy Template** and **Resource Pool Configuration** in **Network Planning**. For details about parameters, see corresponding templates in the lab plan.

Step 1 Configure a fabric global resource pool.

Configure a fabric global resource pool according to the plan.

# Configure a fabric resource pool.



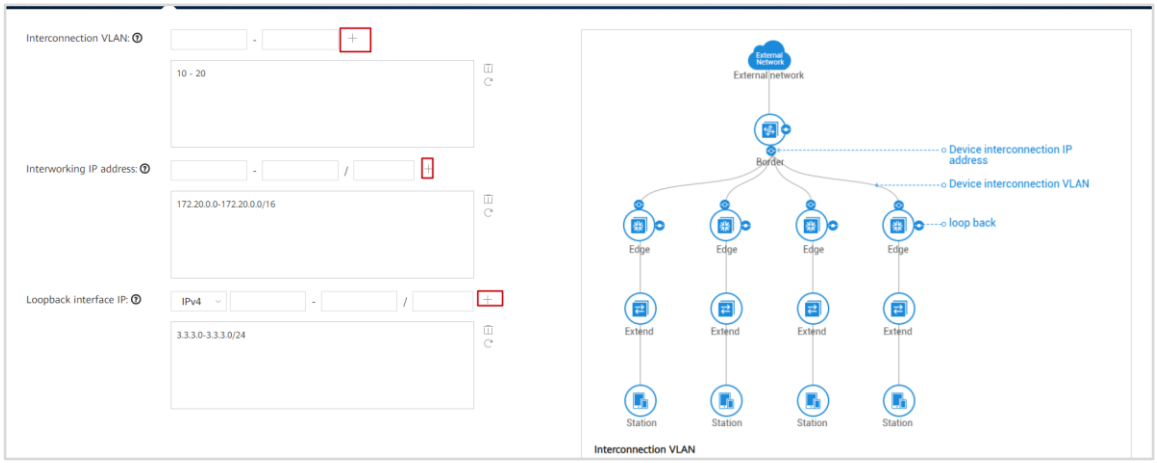

On the home page of iMaster NCE, choose **Deployment Scenario** > **Fabric Network** > **Network Planning** > **Fabric Global Resource Pool** to access the configuration page of a fabric global resource pool.

After adding a parameter setting, click **+** next to it.

Step 2 Configure an underlay automation resource pool.

Configure an underlay automation resource pool according to the plan.

# Configure an underlay automation resource pool.

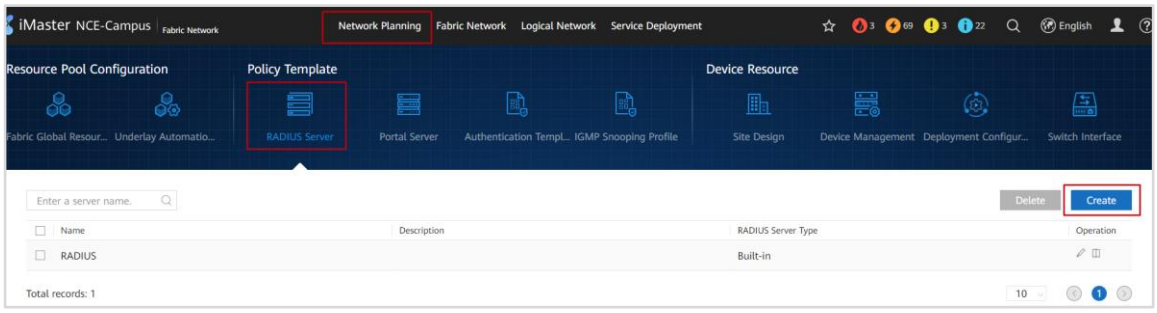


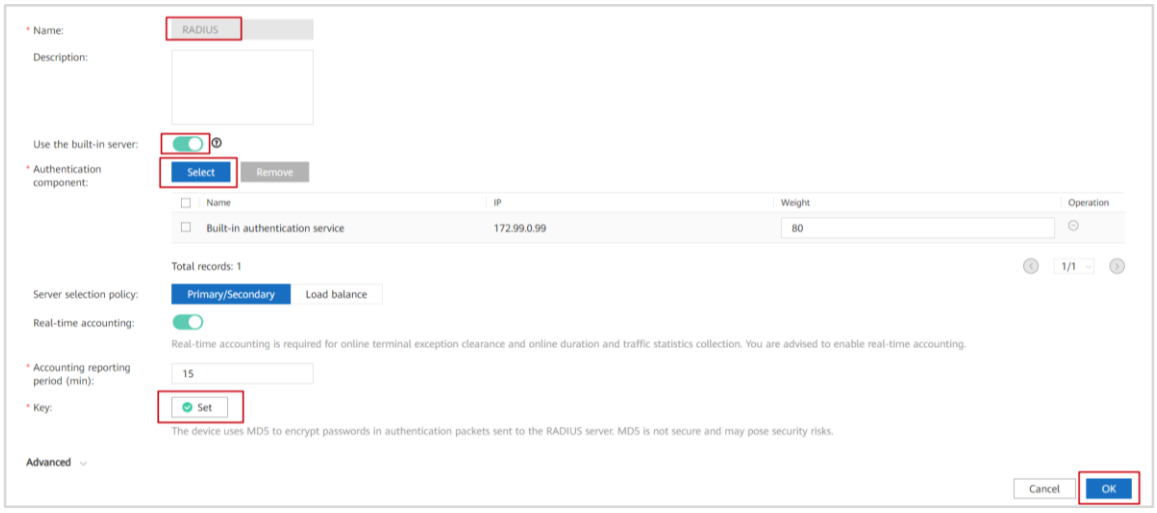

Click **Underlay Automation Resource Pool** in the navigation on the top, set corresponding parameters in the configuration page of the underlay automation resource pool, and click **+**.

Step 3 Configure a policy template.

Complete the configurations of **RADIUS Server**, **Portal Server**, and **Authentication Template** in **Policy Template**.

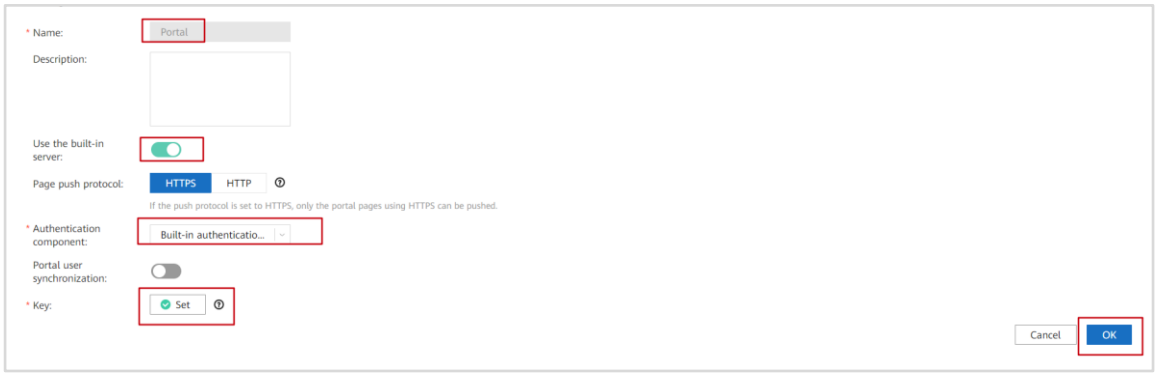
# Configure a RADIUS server.



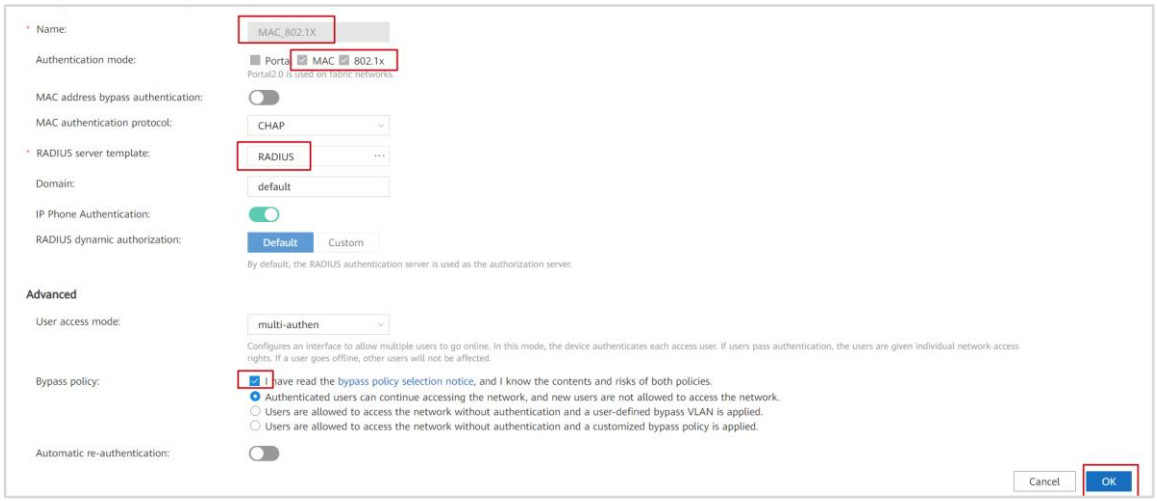

Click **RADIUS Server** and **Create** in the navigation bar on the top. Complete the configuration of the RADIUS server in the displayed window and click **OK**.

# Configure a Portal server.



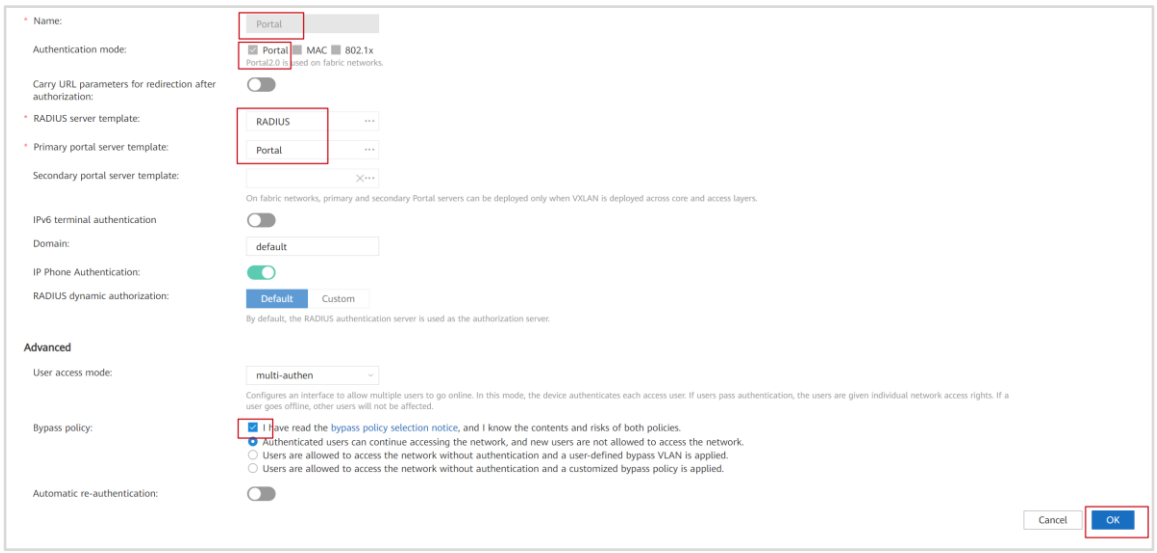

Click **Portal Server** and **Create** on the current page. Complete the configuration of the Portal server in the displayed window and click **OK**.

# Create an authentication template named **MAC_802.1X**.




Click **Authentication Template** and **Create** on the current page to create the 802.1X and MAC address authentication template.

# Create an authentication template named **Portal**.



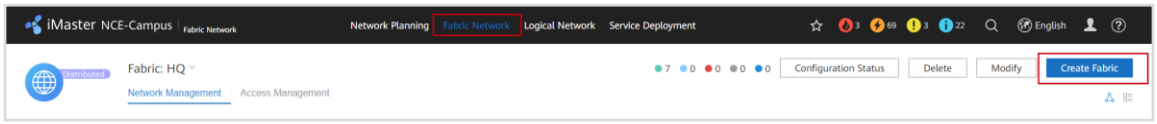

Click **Authentication Template** and **Create** on the current page to create a Portal authentication template.

## 6.1.2.5.2 Configuration for a Fabric Network

Step 1 Create a fabric network and complete automatic deployment of an underlay network.

Create a fabric network according to the plan.

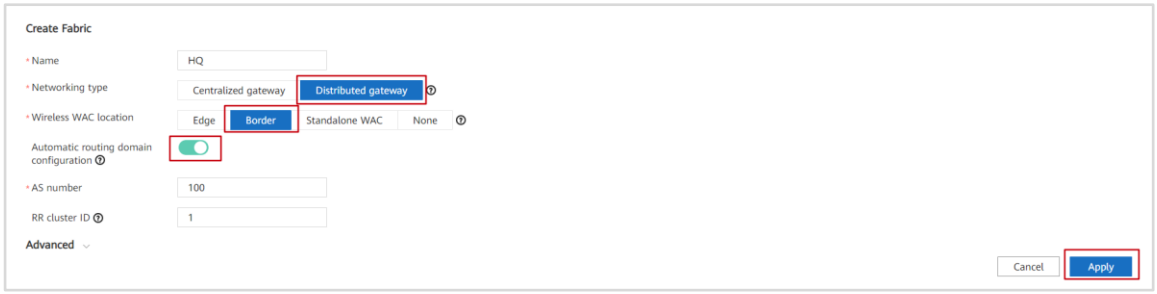
# Create a fabric network.




Click **Fabric Network** on the top of the page to access the fabric management page, and click **Create Fabric**.

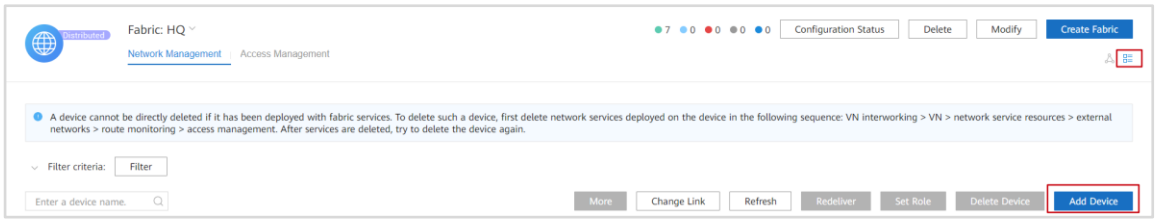
If there is no fabric network on the controller, a message will be displayed, indicating that a fabric network should be created.

# Create a fabric network according to the plan.



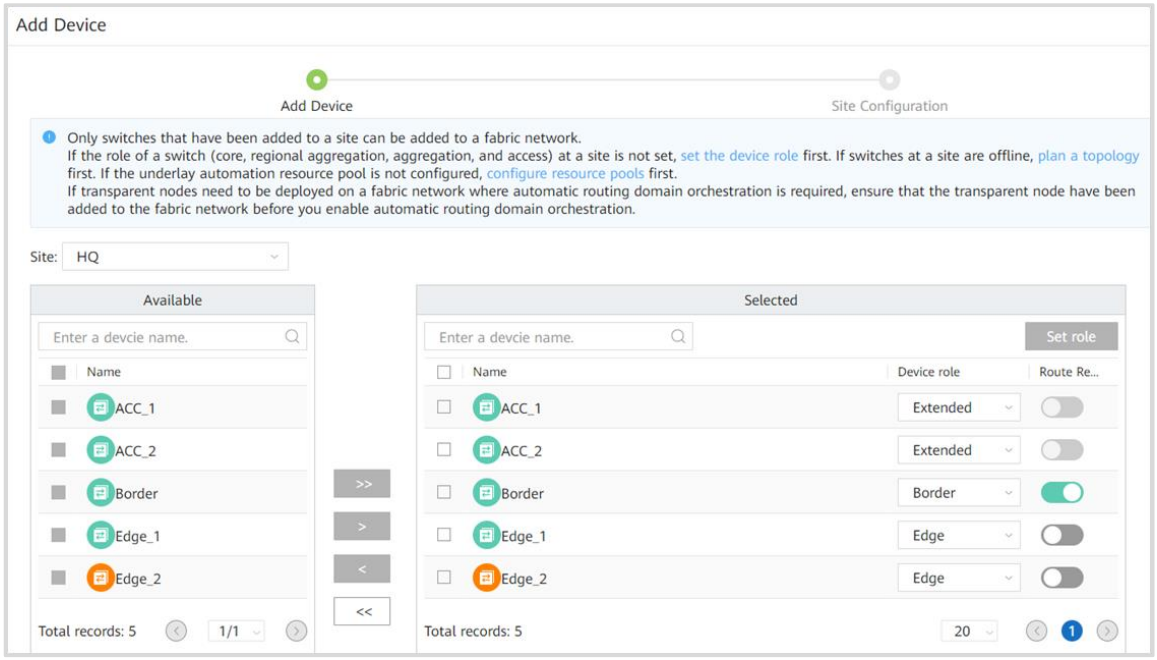

Click **Apply**.

# Add devices.




Click the icon numbered **1** to switch the view. Click **Add Device** to add switches to the fabric network.

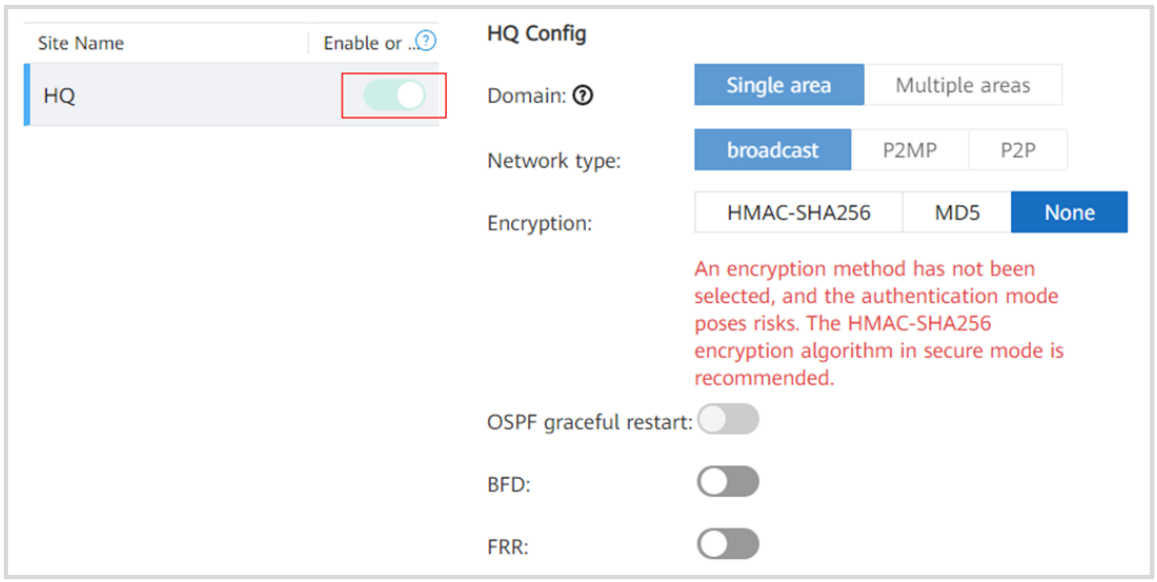
# Configure device roles.




On the displayed page, add the devices on the left to the right (select a device and click the button numbered **1**), and set the roles of switches according to the plan.

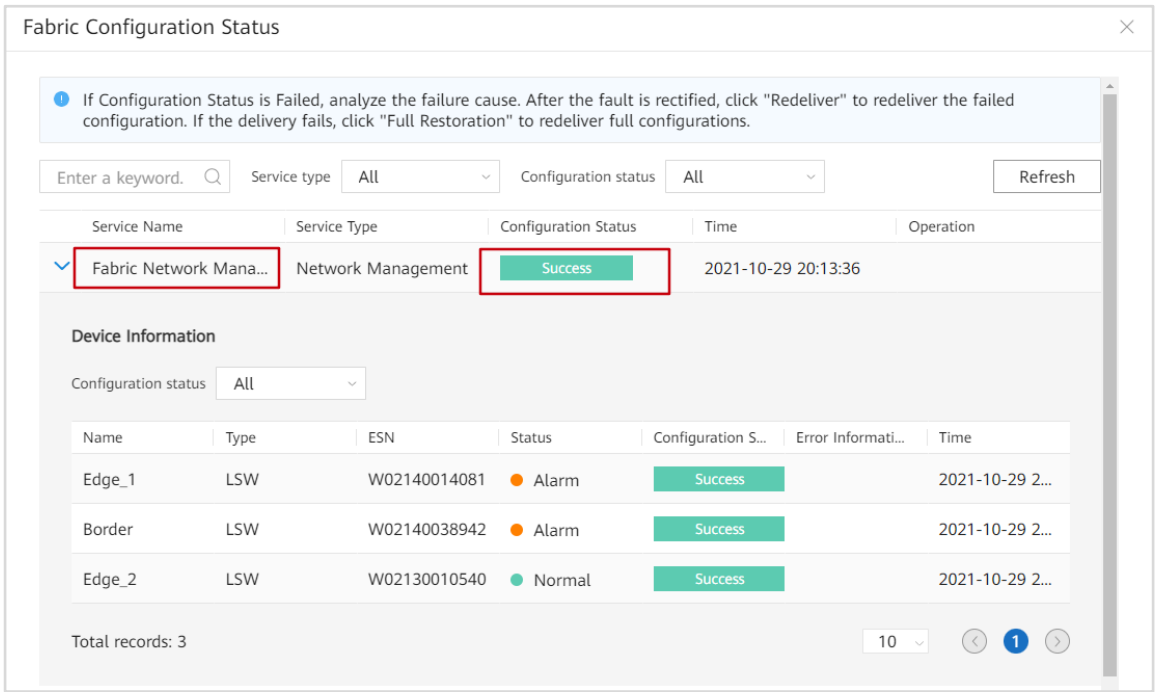
In this lab, Edge_1 and Edge_2 need to be specified as edge nodes (edge nodes on the VXLAN), ACC_1 and ACC_2 need to be specified as extended nodes, and the core switch Border needs to be specified as the border node which connects the fabric network with external networks.

# Complete automatic physical network deployment.




On the following page, enable **Config Switch** next to **HQ** and disable the authentication between OSPF neighbors, then click **Apply**.

# Verify the configuration.

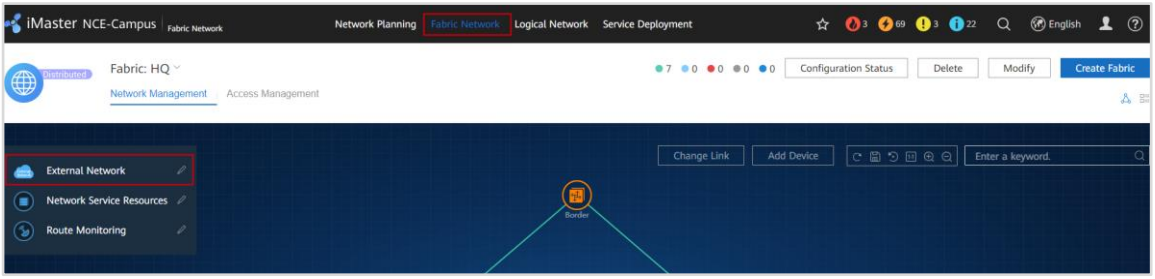



Click **Configuration Status** on the current page, and ensure that **Configuration Status** of all configuration items is **Success**.

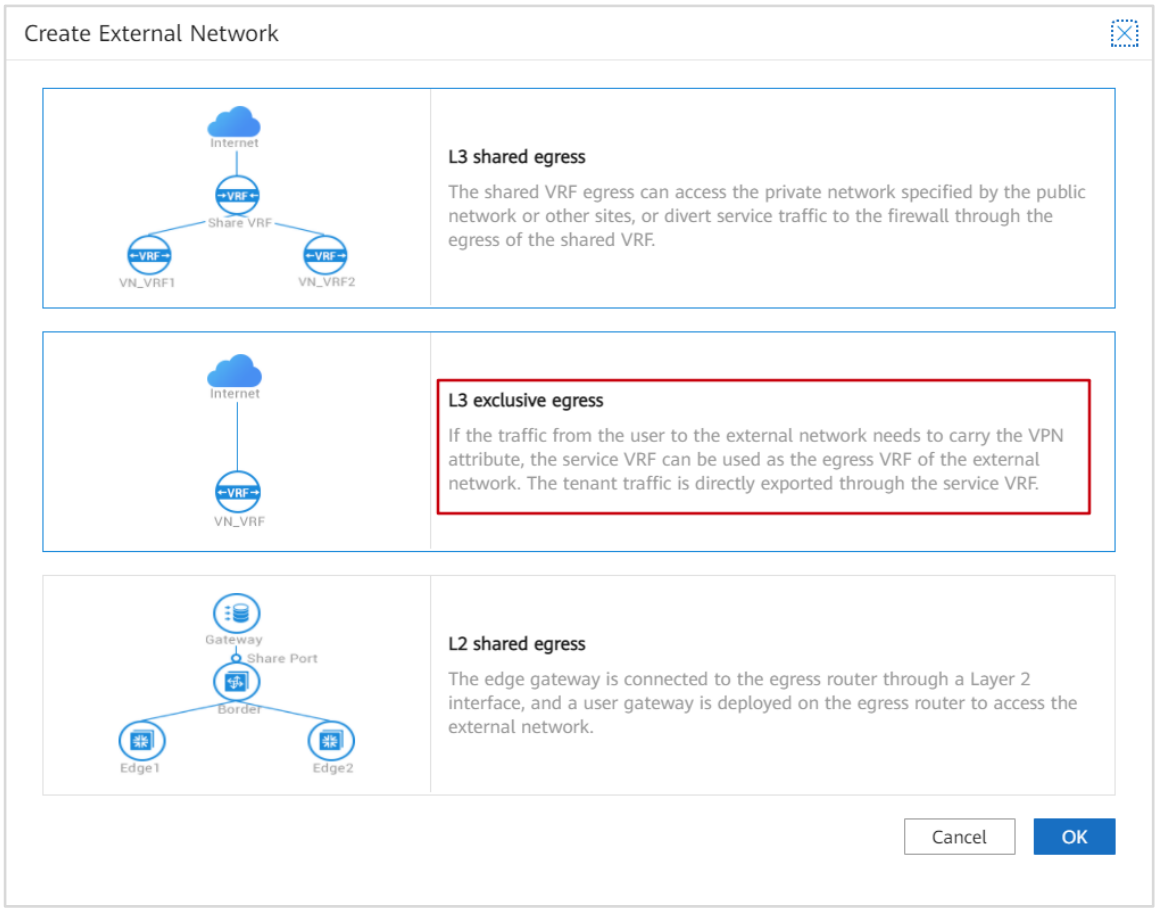
## Step 2 Configure external networks.

In this step, we will create external networks.

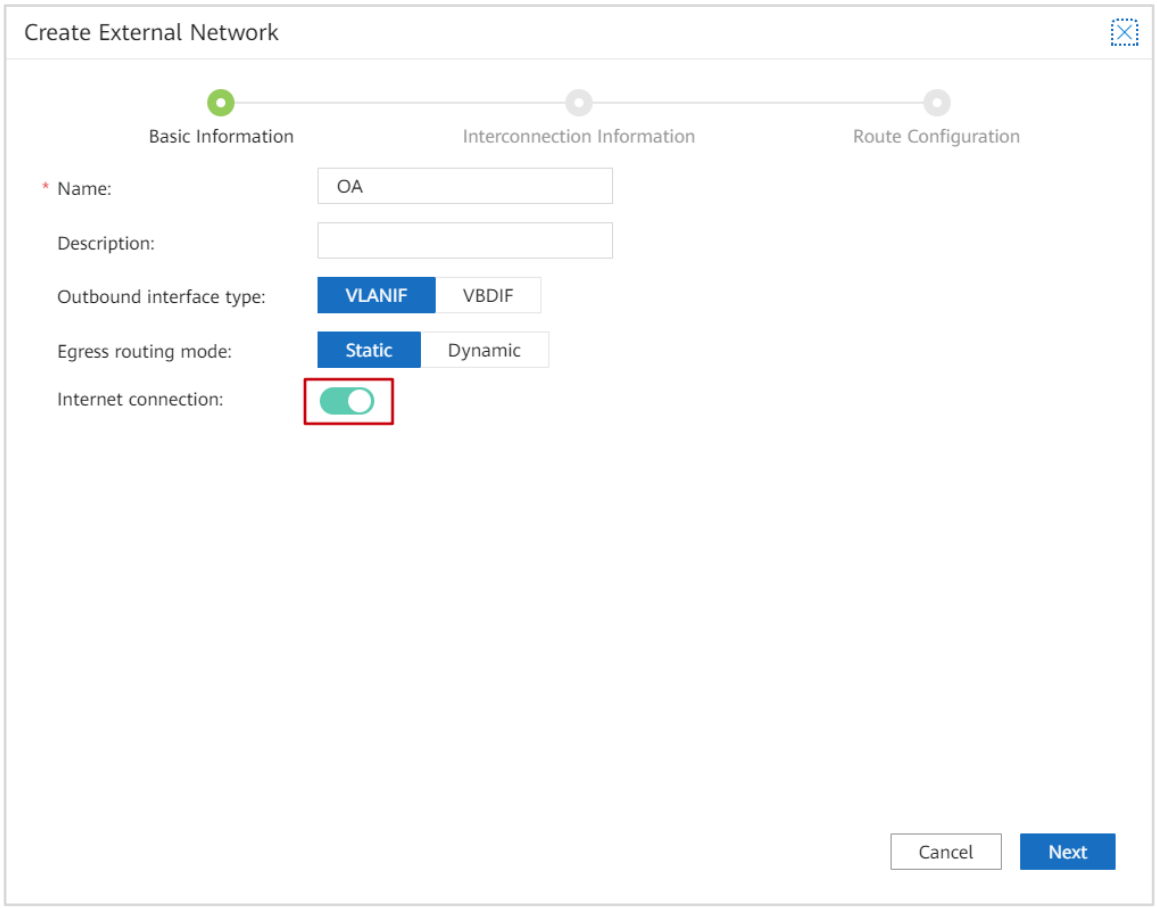
# Configure the external network OA.



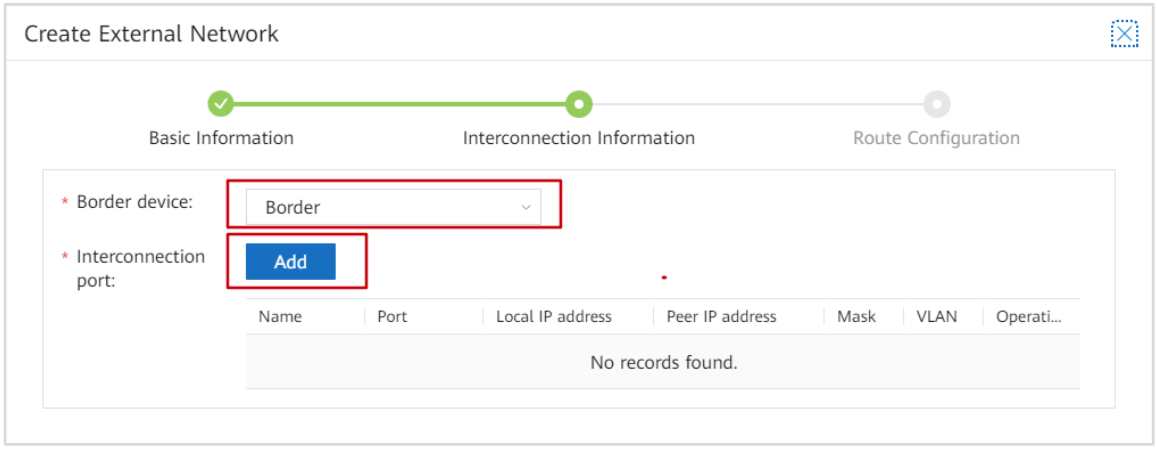

Choose **Deployment Scenario** > **Fabric Network** to access the configuration page of the fabric network, and click the edit icon next to **External Network** and then click **Create**.




Select **L3 exclusive egress** and click **OK**.



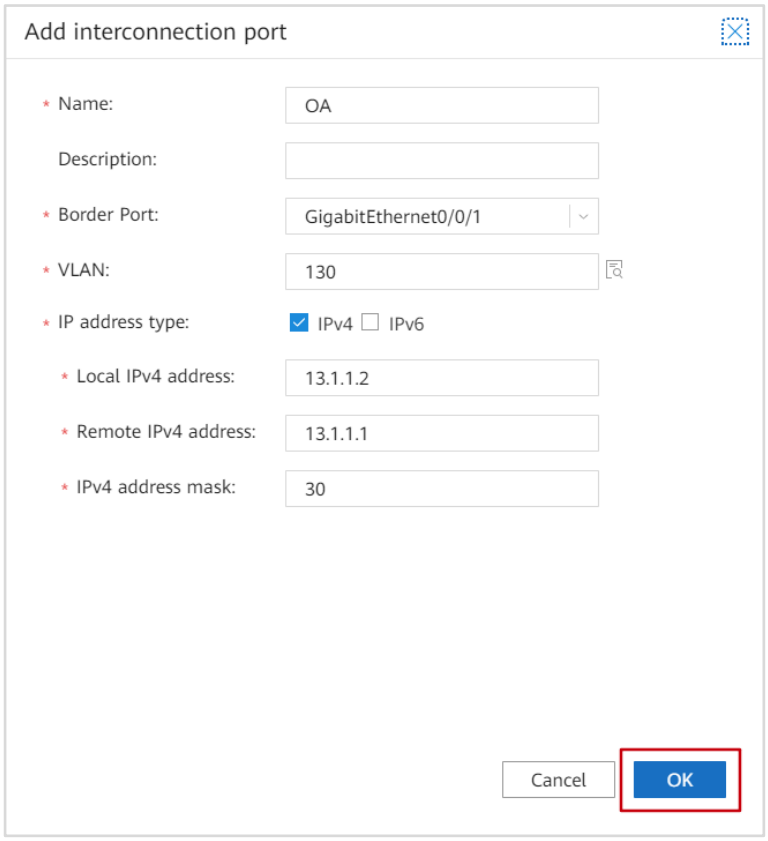

Configure the name of the external network, and enable **Internet connection**.




Set **Core device** to **Border** and click **Add** to configure the interconnection port.

Set **Border device** to **Border** on this page, indicating that **Border** is used as the border device to connect to the external network.

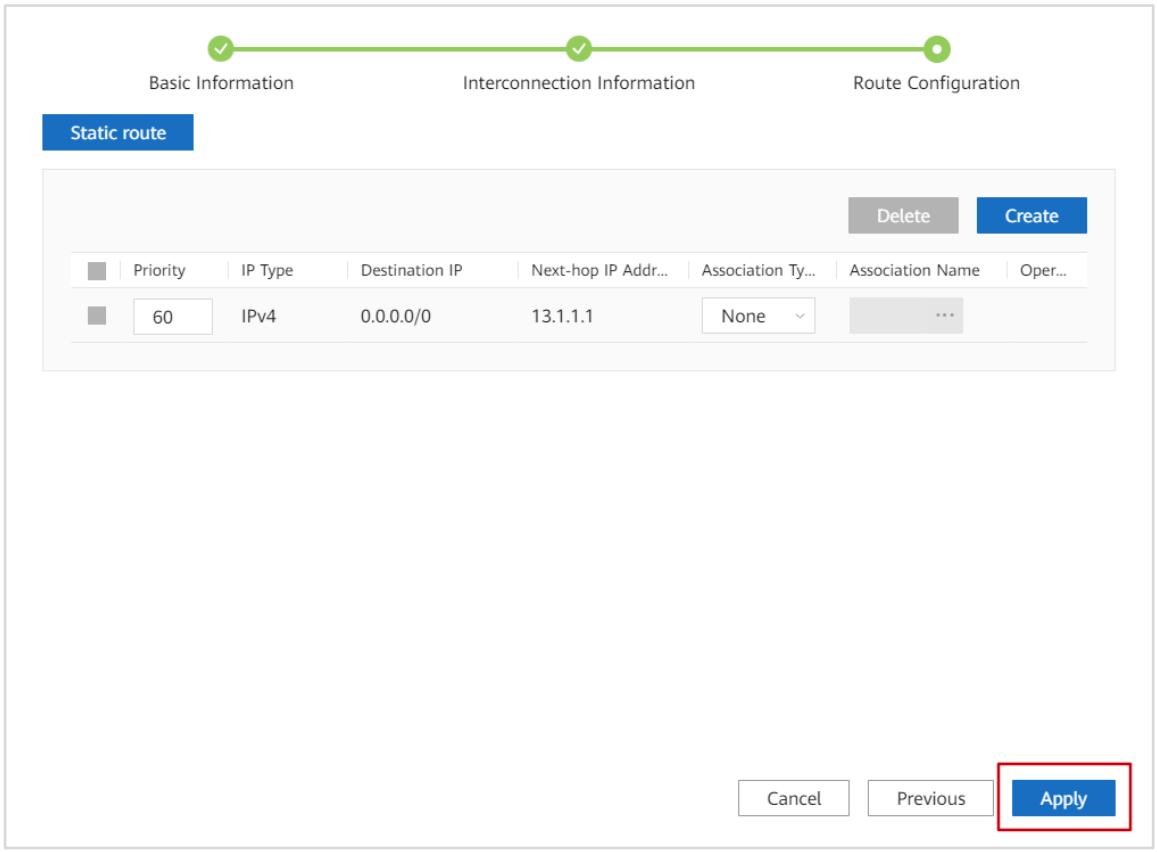
Then click **Add** next to **Interconnection port** to configure the port used by the border device to communicate with the external network, and then configure the connection information based on the data plan.




Parameter descriptions:

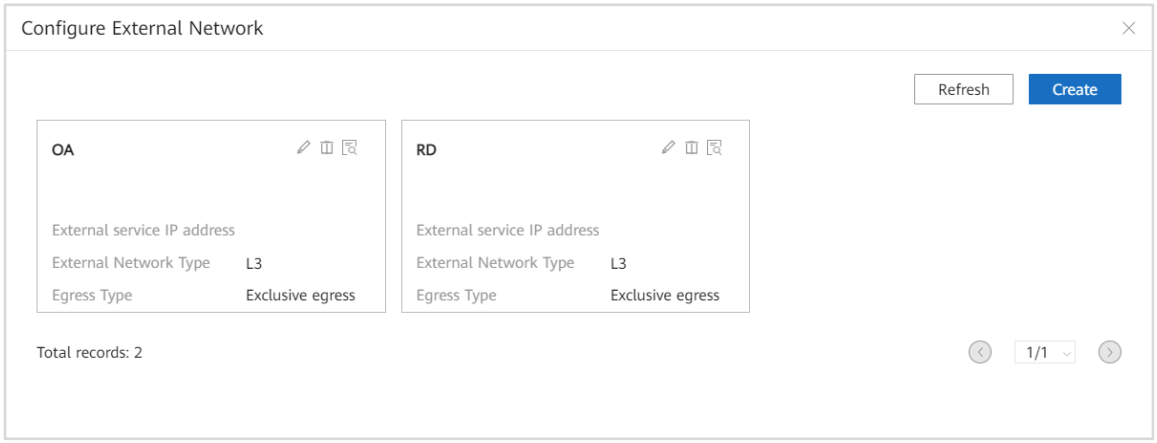
1. Name: customizable, OA is used as an example here.
2. Border Port: port used by Border to connect to the external device (AR3 in this example) on the external network.
3. VLAN: VLAN used by Border to connect to the external device on the external network.
4. IP address type: IPv4 is used as an example here, and you can select IPv4 or IPv6 as needed.
5. Local IPv4 address: IP address of the local interface used by Border to connect to the external device on the external network. That is, the IP address of the VLANIF interface for the VLAN used by Border to connect to the external network.
6. Remote IPv4 address: the peer device's IP address on the external network.
7. IPv4 address mask: mask of the IPv4 address.

Configure the interconnection port based on the plan, and click **OK** then **Next**.




Click **Apply** in **Route Configuration**. The external network OA is created.

# Configure the external network RD.




The procedure for creating the external network RD is the same as that for the external network OA, and is not described here. Configure it according to the plan.

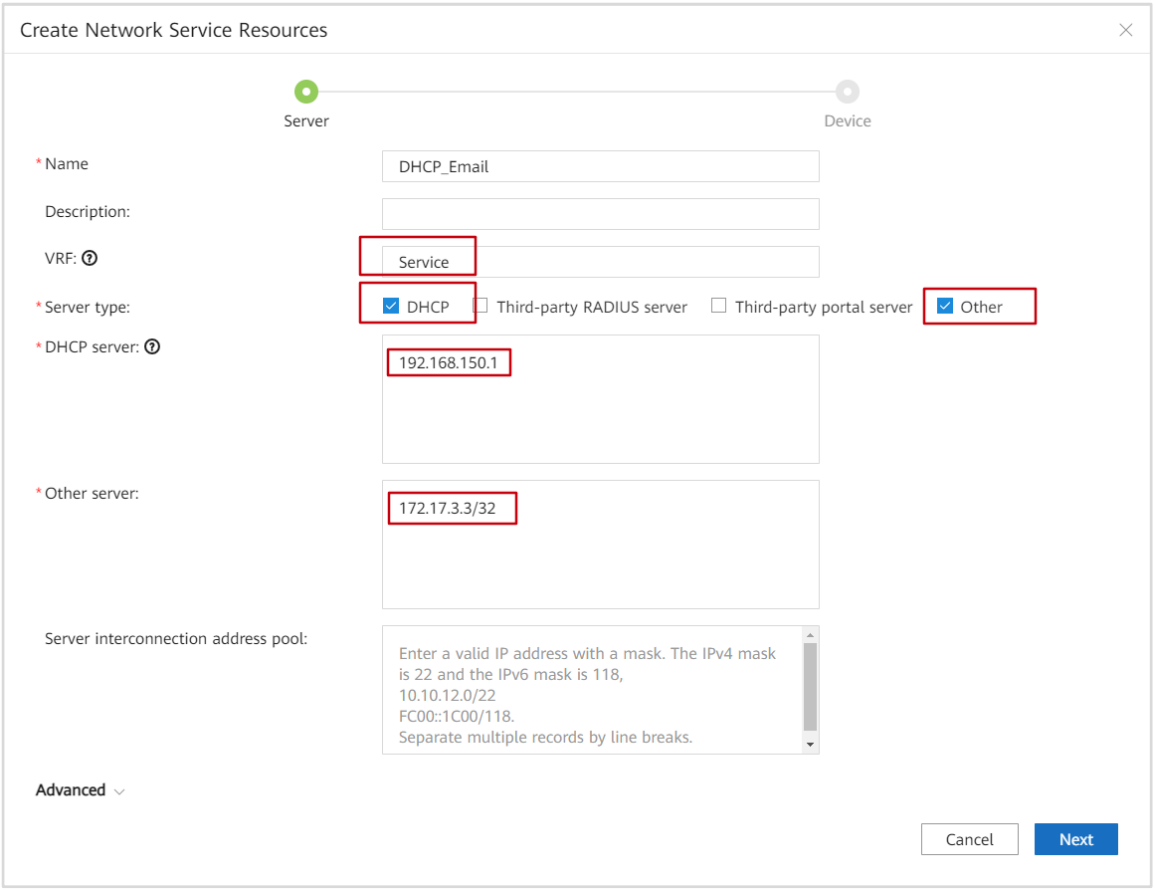
Step 3 Configure network service resources.

In this step, we will configure the connectivity between the fabric network and network service resources (DHCP server, RADIUS server, Portal server, and other servers). The connectivity configuration includes the device IP address, interconnection VLAN, interconnection IP address, peer IP address, and interconnection port.

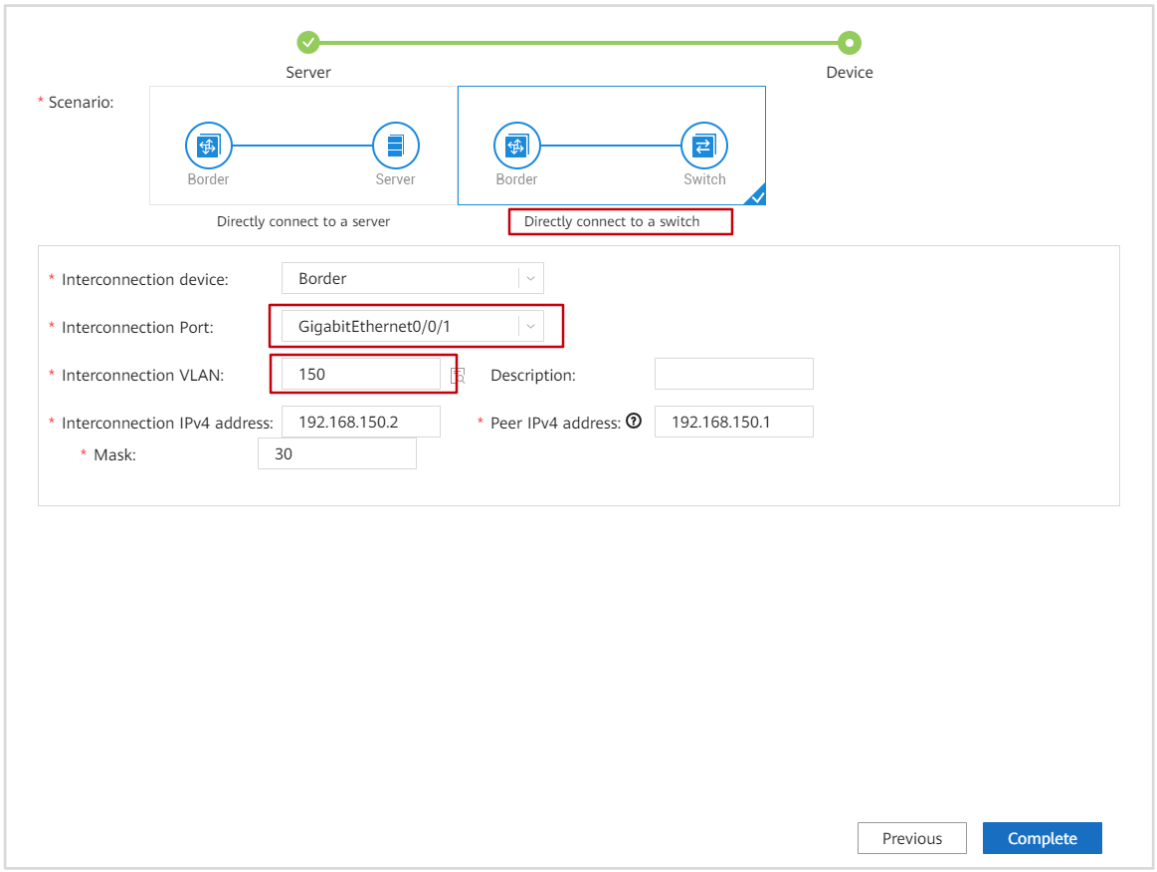
In this example, network service resources include DHCP servers which allocate IP addresses to user terminals and servers used as E-mail servers whose type are other.

# Create network service resources.


Choose **Deployment Scenario** > **Fabric Network** to access the configuration page of the fabric network, and click the edit icon next to **Network Service Resources** and then click **Create**.

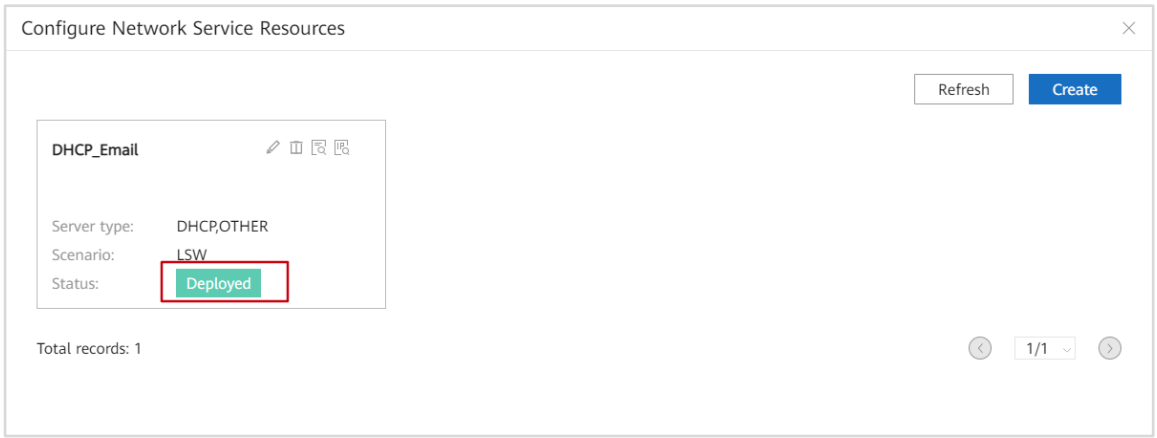



Configure the DHCP server and E-mail server as planned, and click **Next**.




Complete the interconnection configuration as planned. Set **Scenario** to **Directly connect to a switch** and click **Complete** to complete the creation of network service resources.

# Check the deployment status of network service resources.




If the status is **Deployed**, the deployment is successful.

Step 4 Perform the access management configuration.

In this lab, network access control is required on the campus network, including 802.1X authentication, MAC address authentication, and Portal authentication. Among the three authentication modes, a RADIUS server is required in 802.1X authentication and MAC address authentication whereas a RADIUS server and a Portal server are required in Portal authentication.

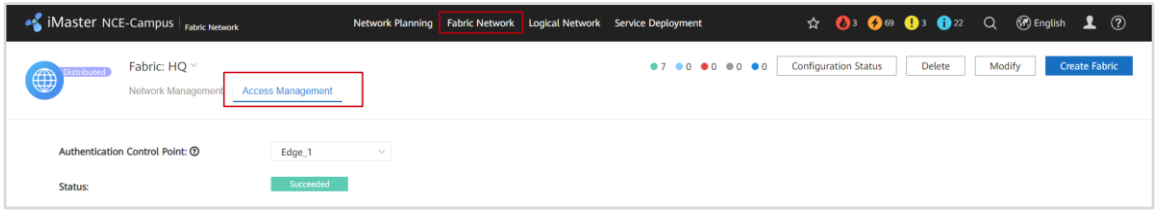
# Switch the working mode of ACC_1 and ACC_2.

```
[ACC_1]undo as-mode disable
Warning: Changing the AS mode will clear current configuration and reboot the system, causing
WLAN services unavailable. Continue? [Y/N]:y
```

```
[ACC_2]undo as-mode disable
Warning: Changing the AS mode will clear current configuration and reboot the system, causing
WLAN services unavailable. Continue? [Y/N]:y
```

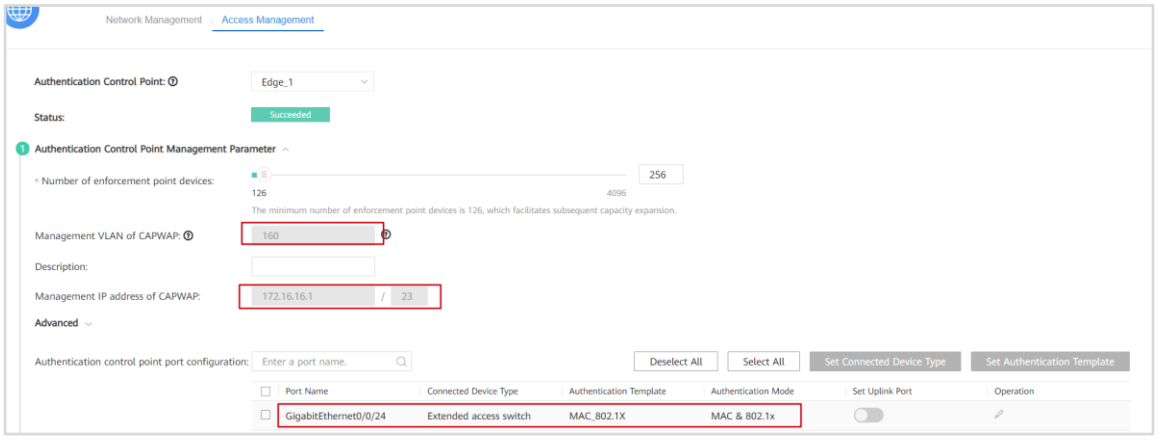
To configure policy association between Edge_1/Edge_2 and ACC_1/ACC_2, you need to switch the working mode of ACC_1/ACC_2 to the AS mode. By default, the working mode of S5731-H series switches is parent, so you need to run the **undo as-mode disable** command to switch the working mode.

# Access the **Access Management** page.




Choose **Deployment Scenario** > **Fabric Network** to access the fabric configuration page. Click **Access Management** to access the access management configuration page. By default, an authentication control point has been selected.

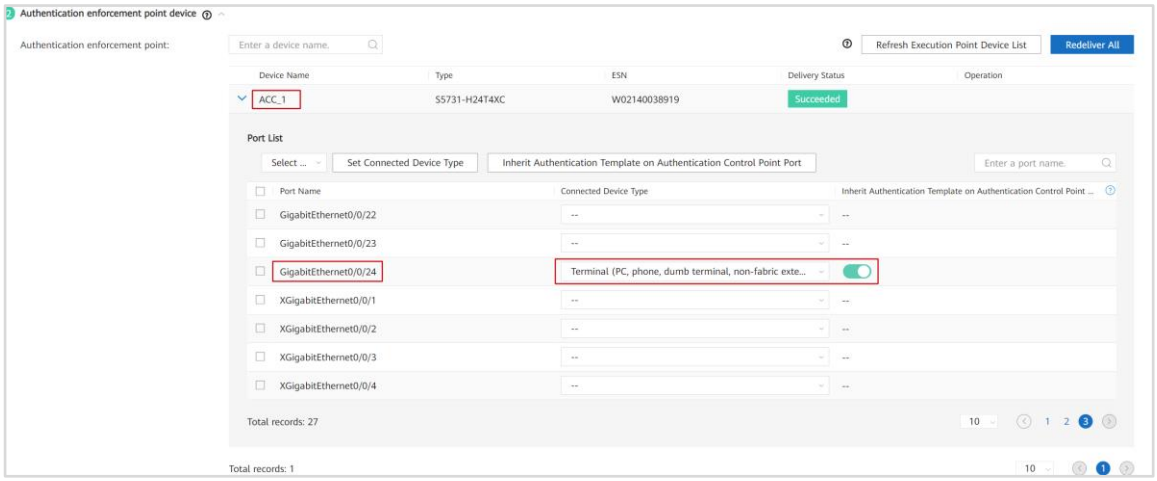
# Configure Edge_1 as an authentication control point.




Configure Edge_1 based on the access management plan table.

ACC switches will authenticate the access terminals together with edge switches through the policy association VLAN.

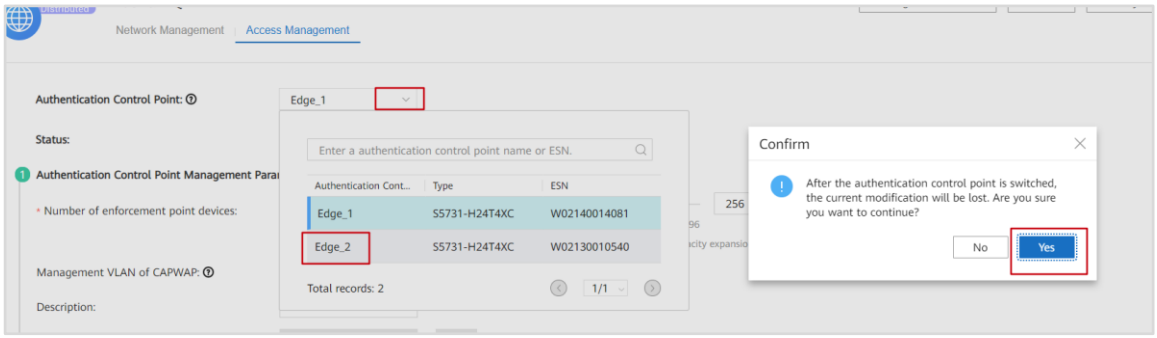
Because policy association is used, set the interface connected to ACC_1 to **Extended access switch** and set the authentication template to **MAC_802.1X**.



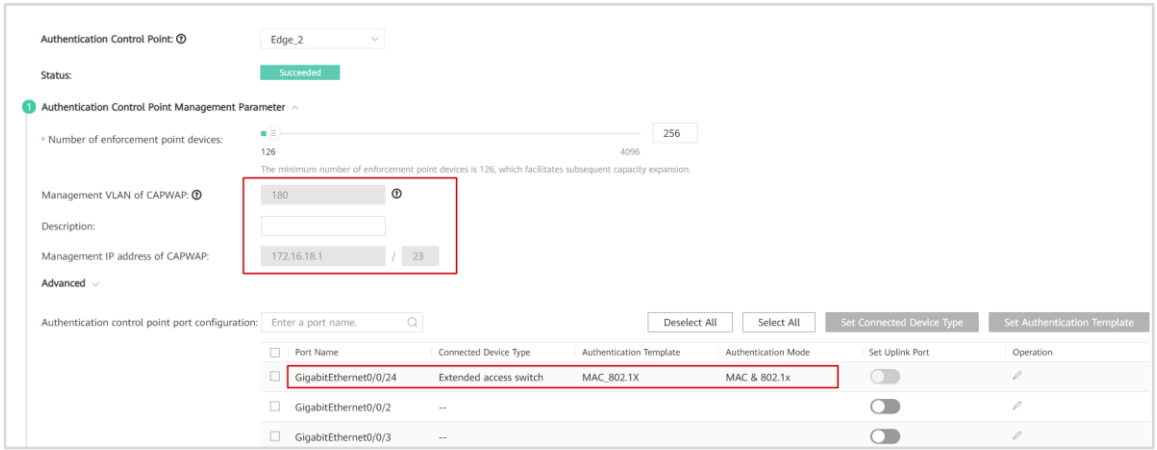

Select **GE0/0/24** (the interface connecting ACC_1 to terminals) in the **Authentication enforcement point device** area and set **Connected Device Type** to **Terminal (PC, phone, dumb terminal, non-fabric extended switch/AP)**. Enable **Inherit Authentication Template on Authentication Control Point Port**. GE0/0/24 on ACC_1 will use the authentication template of Edge_1 and work with Edge_1 to complete the access authentication on terminals.

Click **Apply** in the lower right corner of the page to complete the access management configuration of Edge_1.

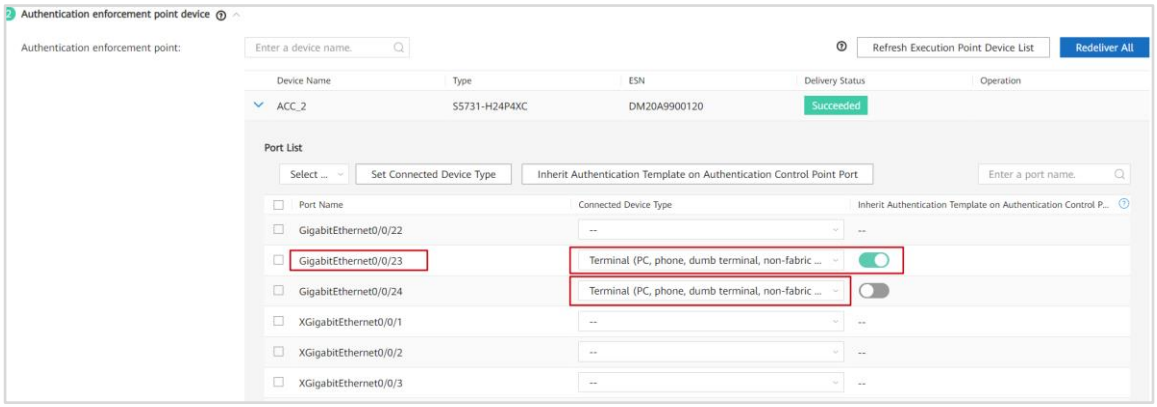
# Configure Edge_2 as an authentication control point.




Select **Edge_2** from the **Authentication Control Point** drop-down list box to access the configuration page of Edge_2. Ensure that you have applied the configuration of Edge_1.

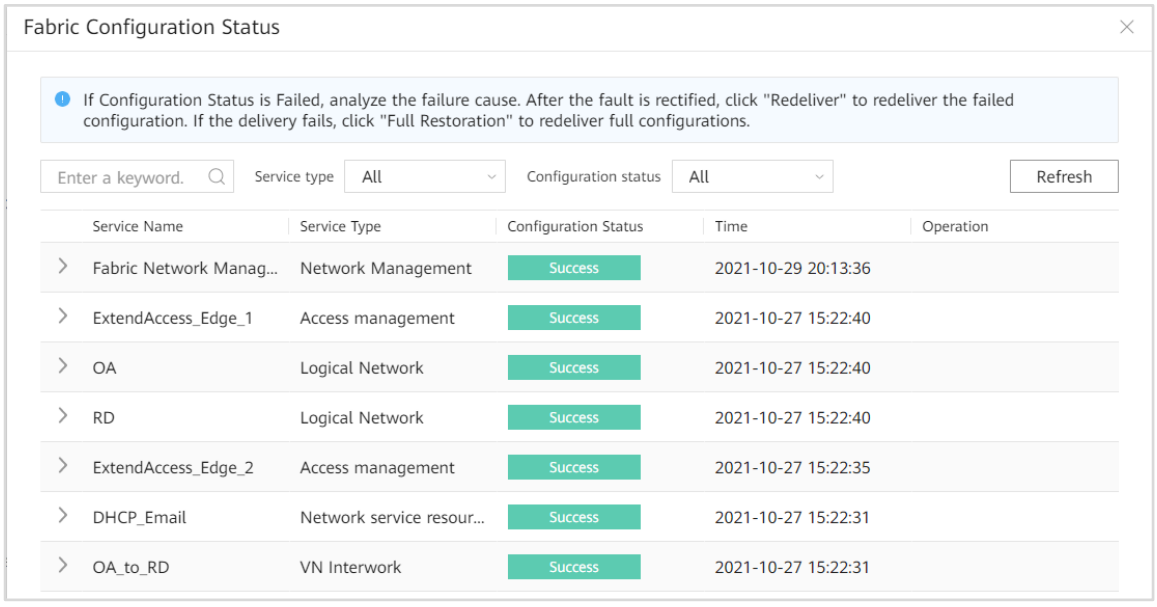



Configure Edge_2 based on the access management plan table.




Select **GE0/0/23** (the interface connecting ACC_2 to terminals) in the **Authentication enforcement point device** area and set **Connected Device Type** to **Terminal (PC, phone, dumb terminal, non-fabric extended switch/AP)**. Enable **Inherit Authentication Template on Authentication Control Point Port**. For an interface connected to an AP, configure only the connected device type.

# Check the fabric configuration delivery result.




Click **Configuration Status** in the upper right corner of the page to check the configuration delivery result.

## 6.1.2.6 Overlay Network Automation

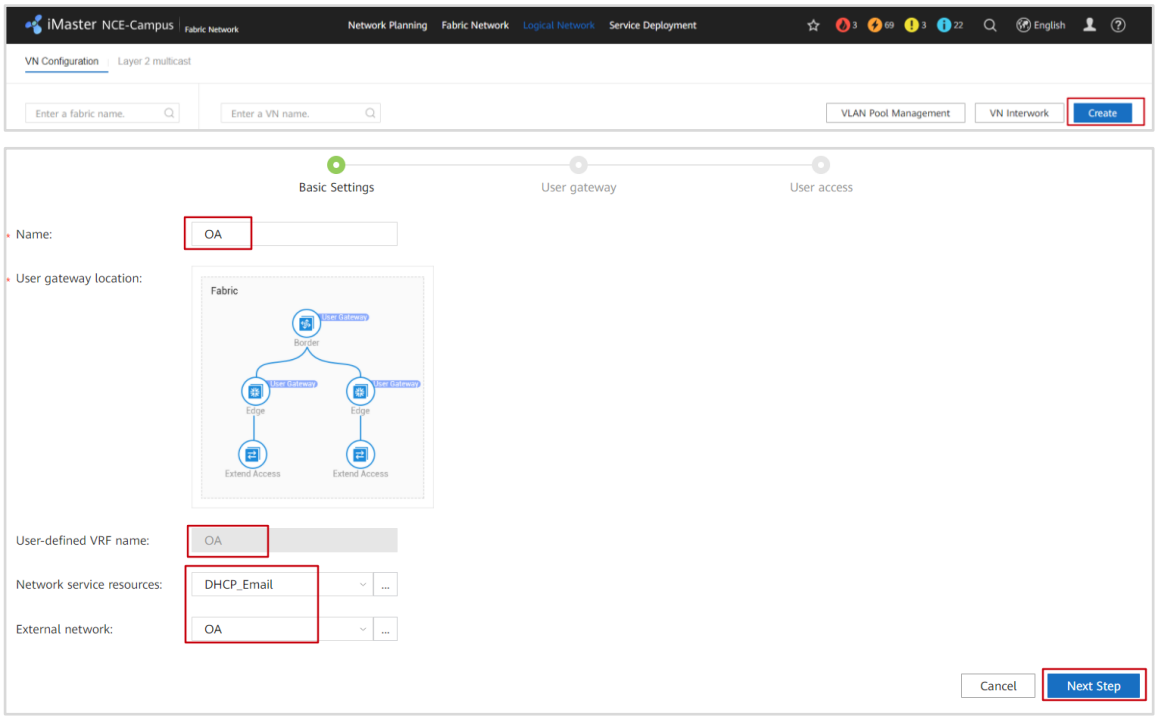
### 6.1.2.6.1 VN

In this part, create two VNs named **OA** and **RD**, associate them with corresponding external networks and network service resources, and create gateways for users.

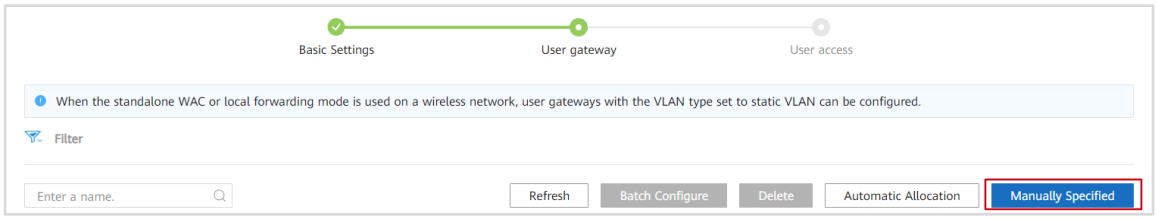
Step 1 Create a VN named **OA**.

Create the OA VN based on the plan table.

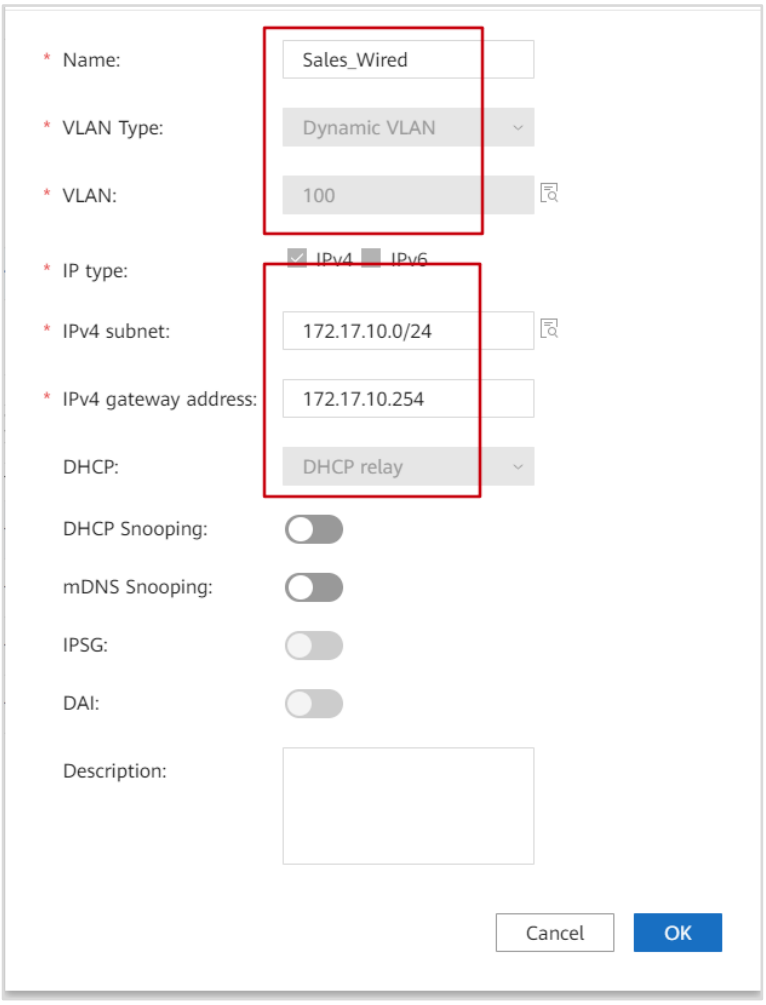
# Create a VN named **OA**.



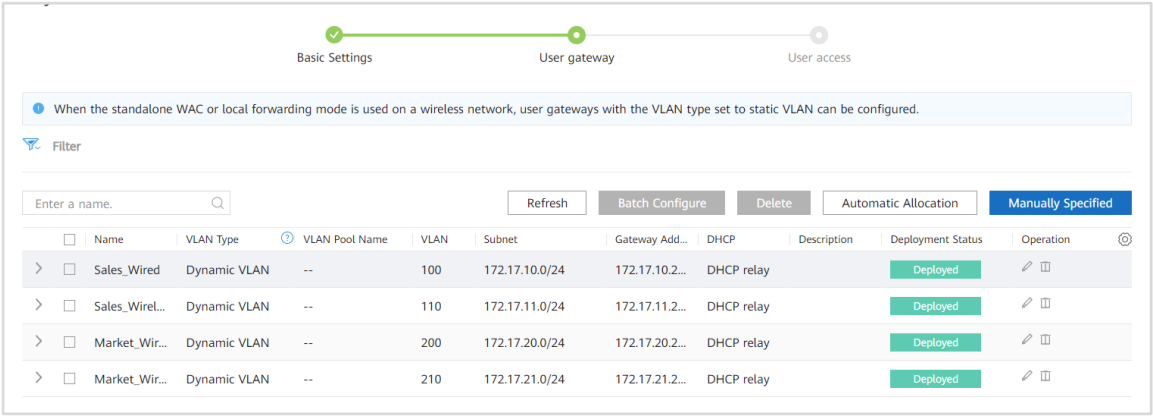

Switch the page to **Logical Network**, and click **Create**. On the **Create VN** page, set **Name** to **OA**, **User-defined VRF name** to **OA**, **Network service resources** to **DHCP_Email**, and **External network** to **OA**.



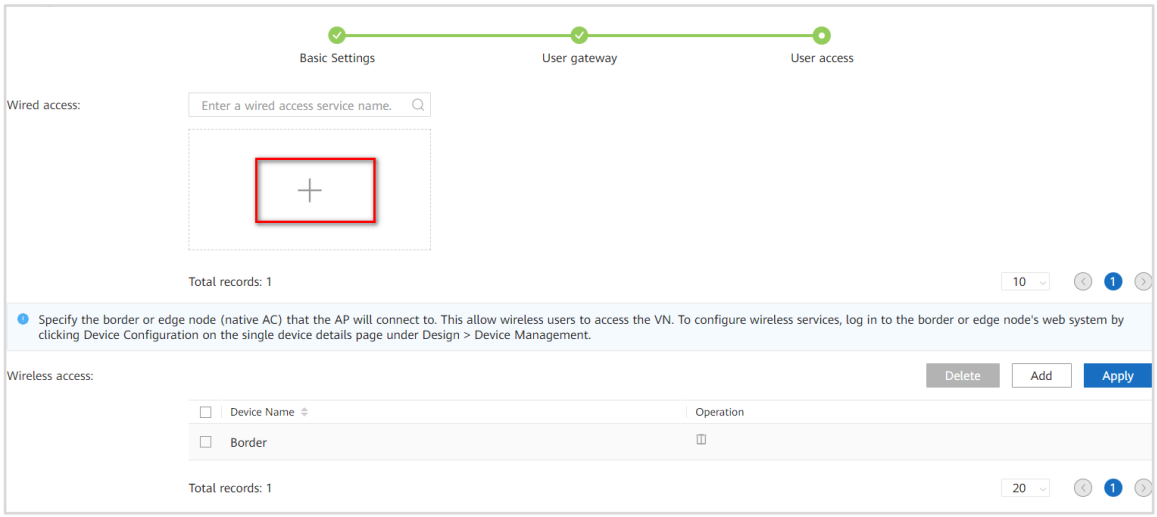

Click **Next** under **User gateway**, click **Manually Specified** to configure the user gateway on the OA VN.



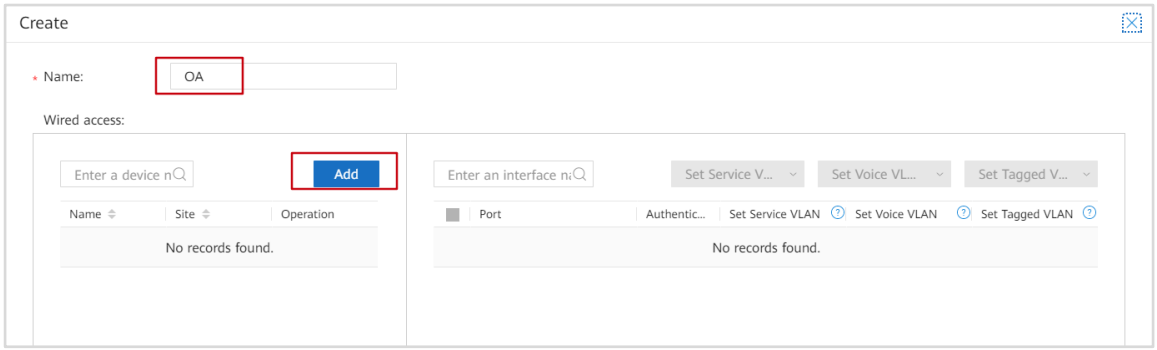

The user gateway configuration of Sales_Wired is shown in the preceding figure. Repeat the procedure to configure the other three user gateways.



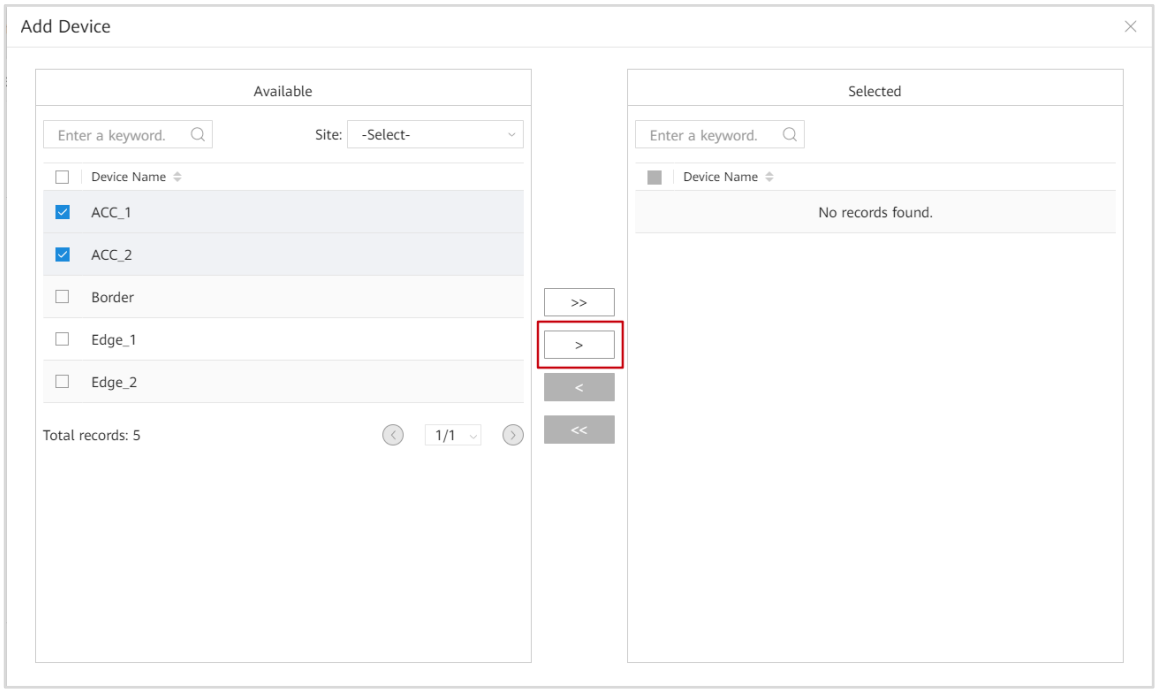

Click **Next Step** to configure user access.



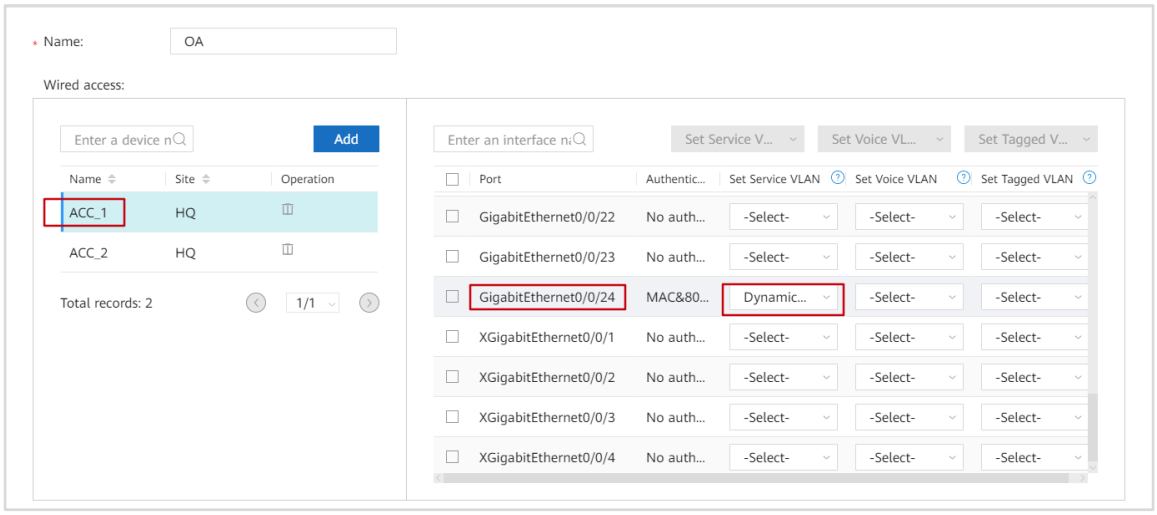

Click the **+** button to add wired access.



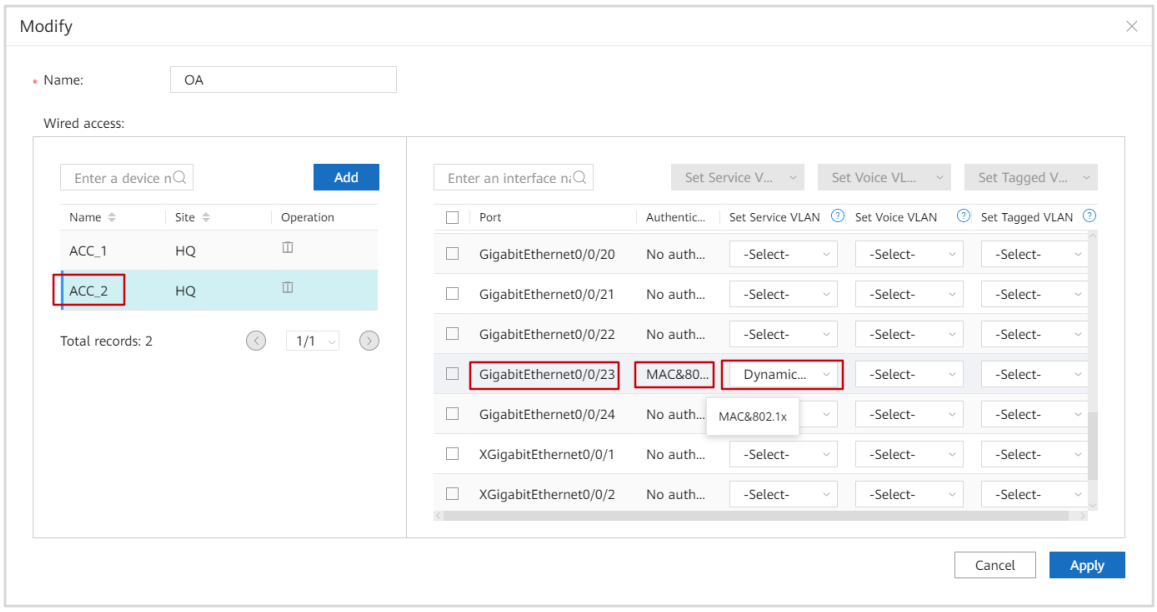

Set **Name** to **OA**. Click **Add**.




Select **ACC_1** and **ACC_2**. Add them to the **Selected** area and click **OK**.

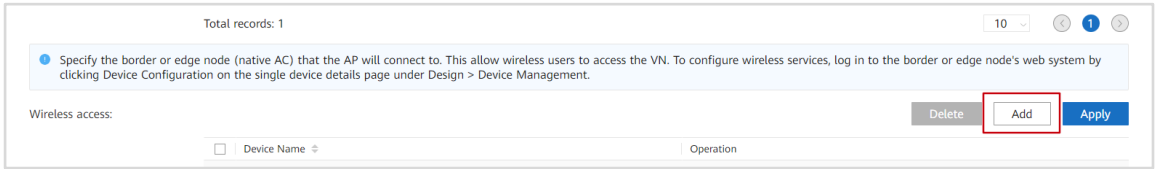


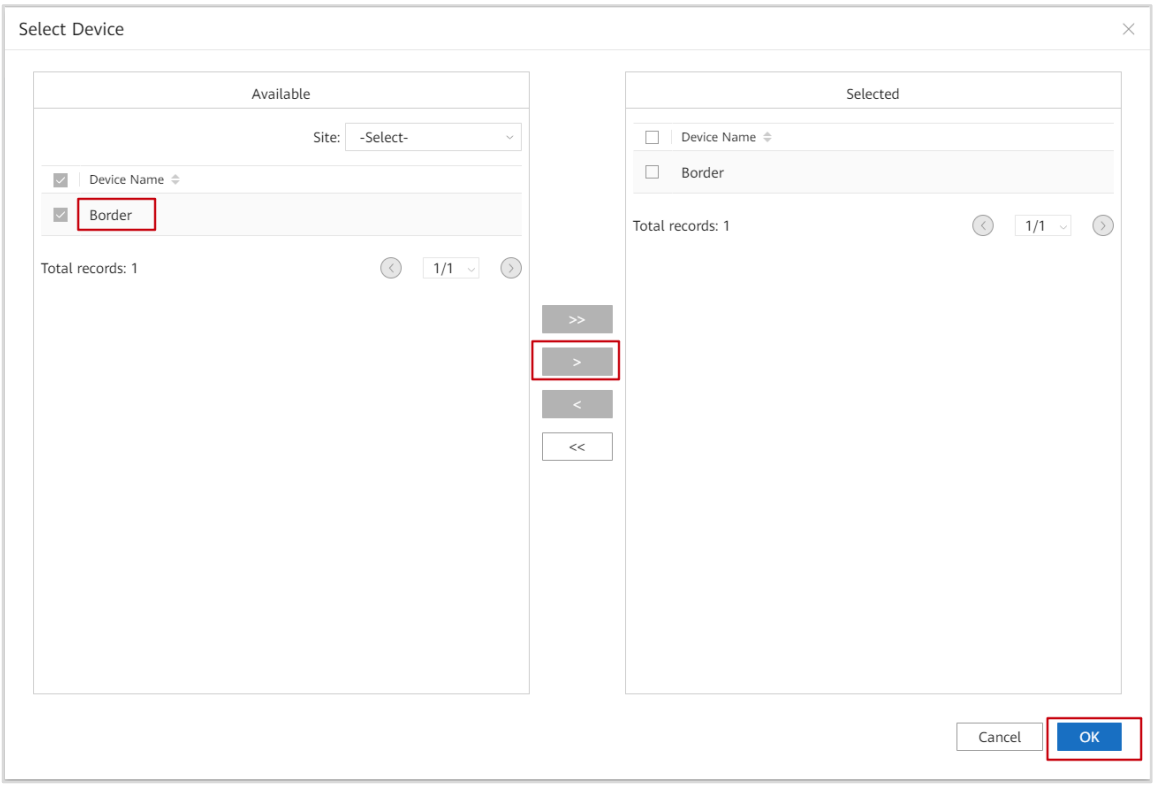

Set the service VLAN for GE0/0/24 on ACC_1 to **Dynamic VLAN**.




Set the service VLAN for GE0/0/23 on ACC_2 to **Dynamic VLAN**. Click **Apply**.

# Configure wireless access.






Click **Add**. Select **Border** and click **OK**

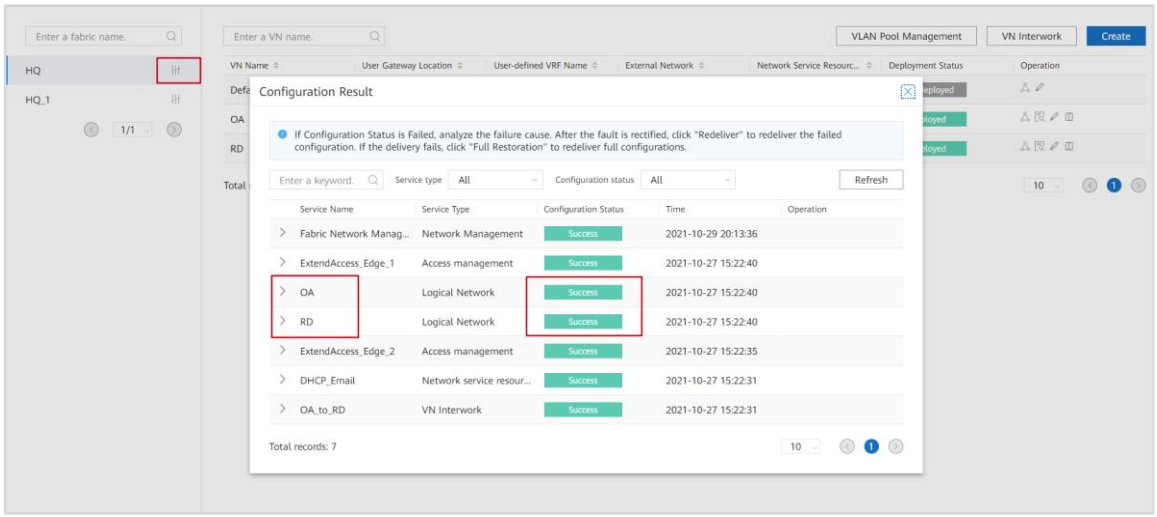

Click **Apply** to finish the OA VN configuration.

Step 2 Create a VN named **RD**.

Create an RD VN by referring to the procedure of creating the OA VN. See the data plan of the RD VN for related parameters. Note that wireless users are not allowed to access the RD VN.

Step 3 Check the configuration.

On the VN management page, click the configuration result icon to check the configuration result, as shown in the following figure.

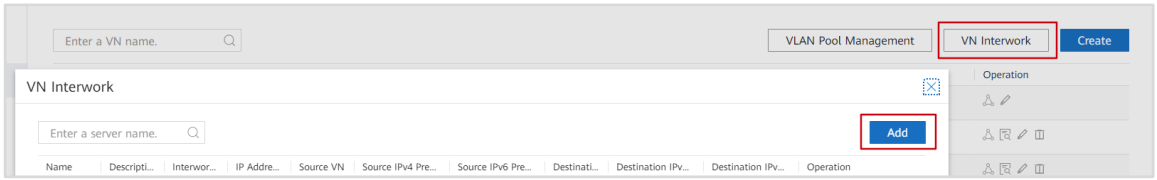



## 6.1.2.6.2 Configuring Communication Between VNs

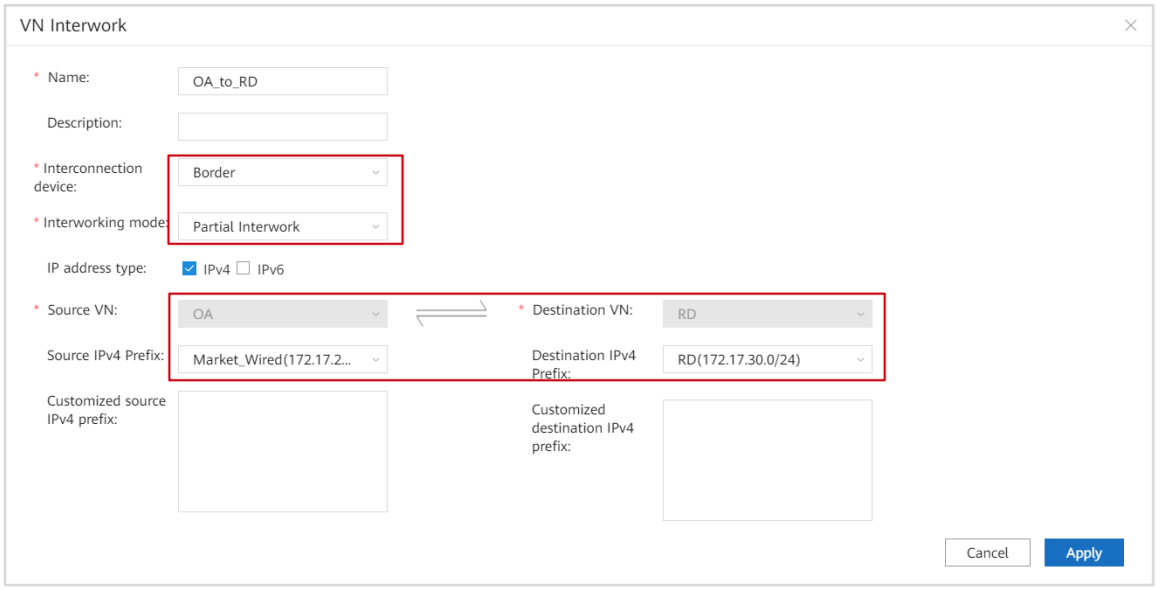
By default, VNs are isolated from each other. However, in some scenarios, they need to communicate with each other, so reachable routes must be configured between them.

For example, in this lab, assume that marketing personnel (in the Market_Wired user group) who access the OA VN can access the RD VN to obtain corresponding device plans. It is necessary to configure communication between network segment 172.16.20.0/24 on the OA VN and network segment 172.16.30.0/24 on the RD VN.

# Create VN communication.

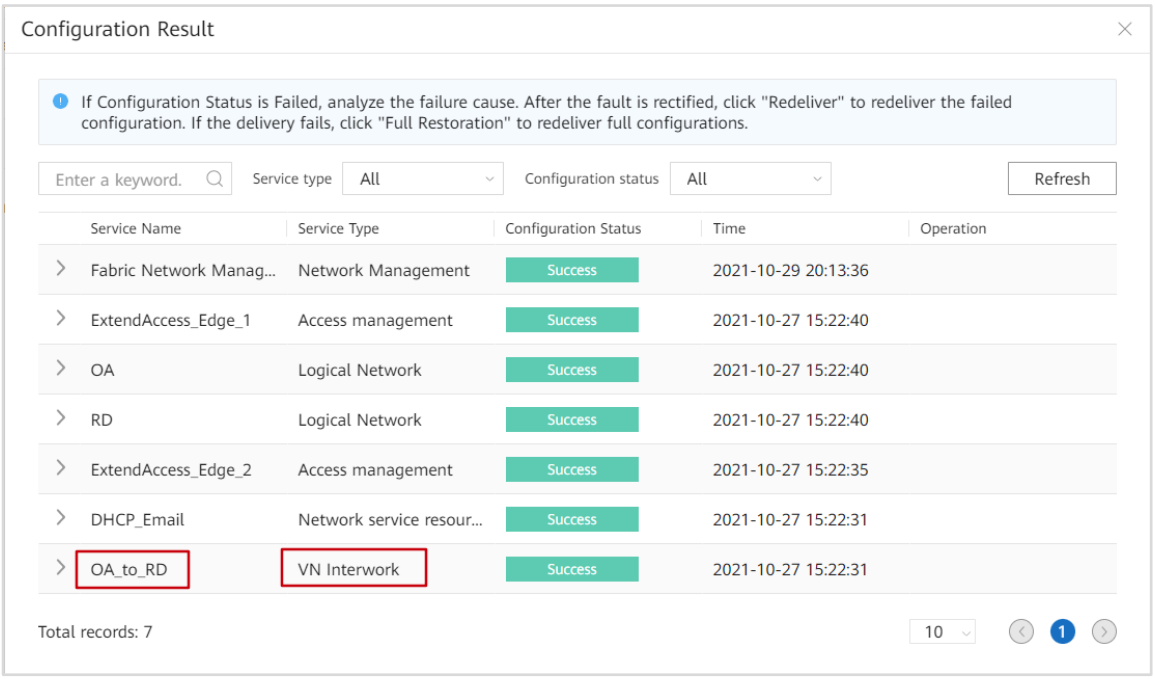



On the current page, click **VN Interwork**. Click **Add**.




Configure VN communication as planned and click **Apply**.

# Check the configuration.




The configuration is completed.

## 6.1.2.7 Free Mobility and Access Authentication

On a large campus network, employees are usually allowed to access the network from any location, any VLAN, and any IP network segment with controlled network access permission. Therefore, free mobility is introduced. Through the controller and agile switches, network access permission can automatically move with users, improving the mobile office experience.

Free mobility solves the following problems faced by the traditional campus networks from three aspects:

1. Service policies and IP address decoupling
2. Centralized management of user information
3. Centralized policy management

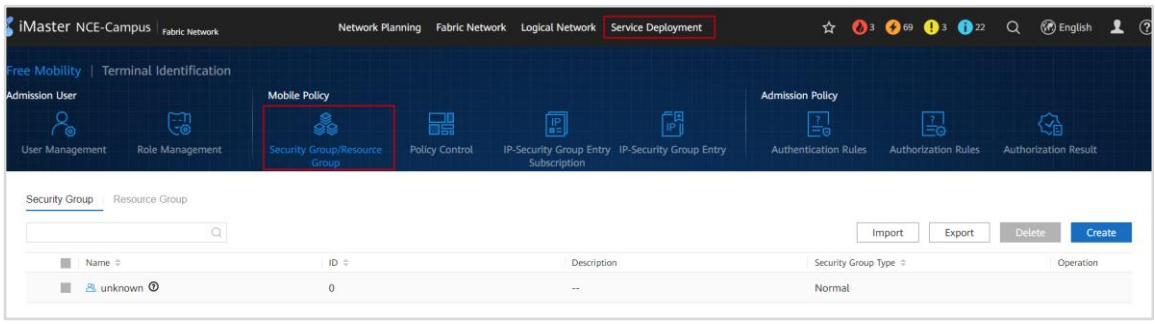
The free mobility function allows the administrator to define logical groups (security groups) on the campus network. The administrator configures the communication rules (policy matrix) between groups based on the GUI of iMaster NCE, and delivers communication rules to policy enforcement points (switches) on the network. Combined with access authentication deployed subsequently, free mobility allows users to obtain the authorization of the corresponding security group when they access the campus network and pass access authentication. In this way, the traffic exchanged between users will be identified as the traffic exchanged between security groups. At the policy enforcement point, the device executes policies for the traffic according to the policy matrix defined by the administrator.

In this lab, we will create one resource group E_mail and five security groups: Sales_Wired, Sales_Wireless, Market_Wired, Market_Wireless, and RD.

### 6.1.2.7.1 Free Mobility

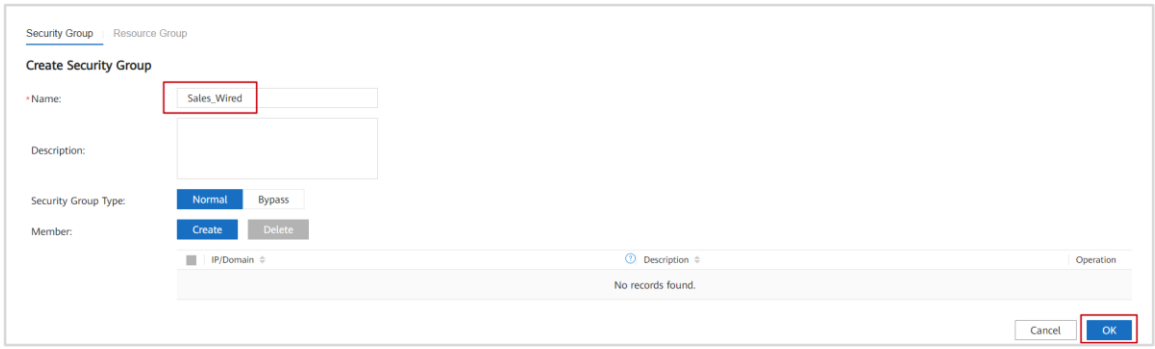
Step 1 Create security groups.

# Switch pages.



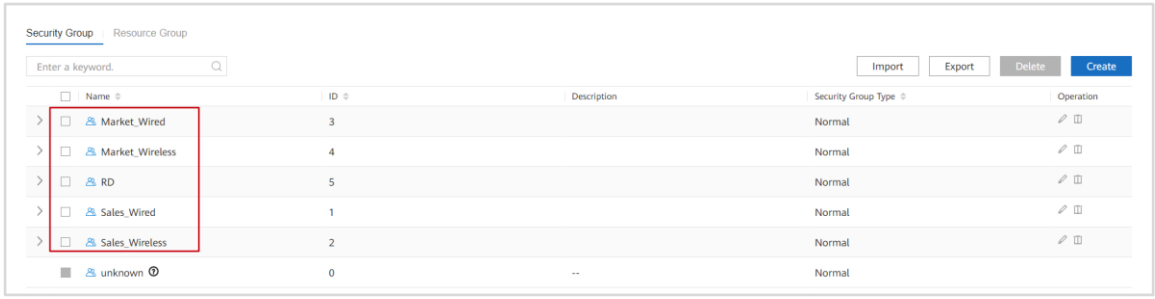

On the home page of iMaster NCE, choose **Deployment Scenario** > **Fabric Network**, click **Service Deployment** to access the service deployment page, and click **Security Group/Resource Group** on the page.

# Create security groups.

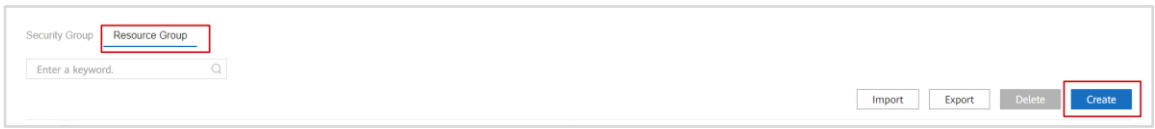


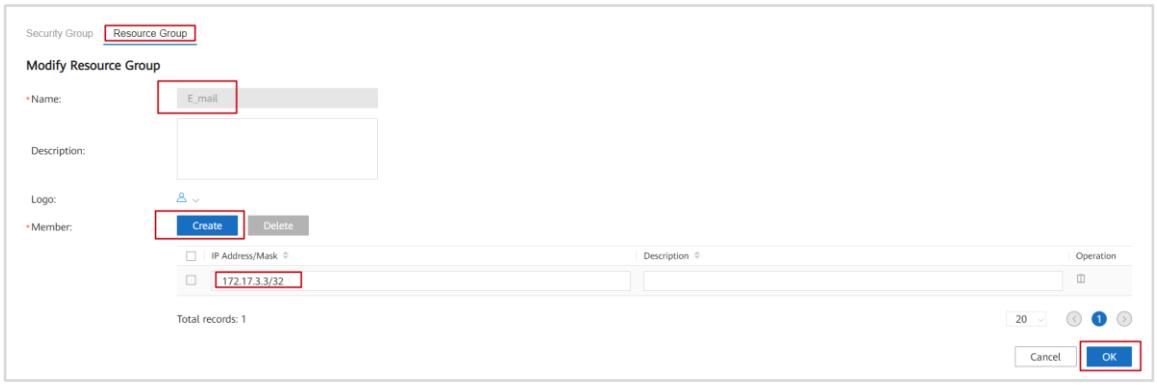

Click **Create** to access the **Create Security Group** page, and set **Name** to **Sales_Wired**. Click **OK**.

Repeat the steps to create the other four security groups.




# Create a resource group.





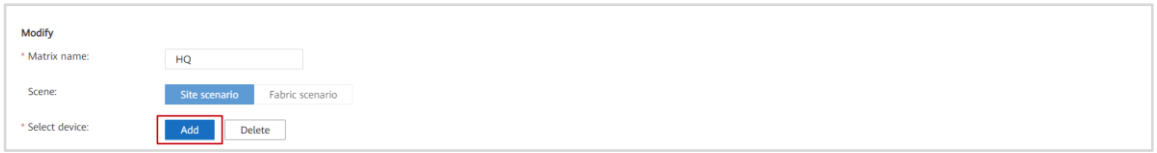

Create a resource group named **E_mail** whose member is 172.17.3.3/32.

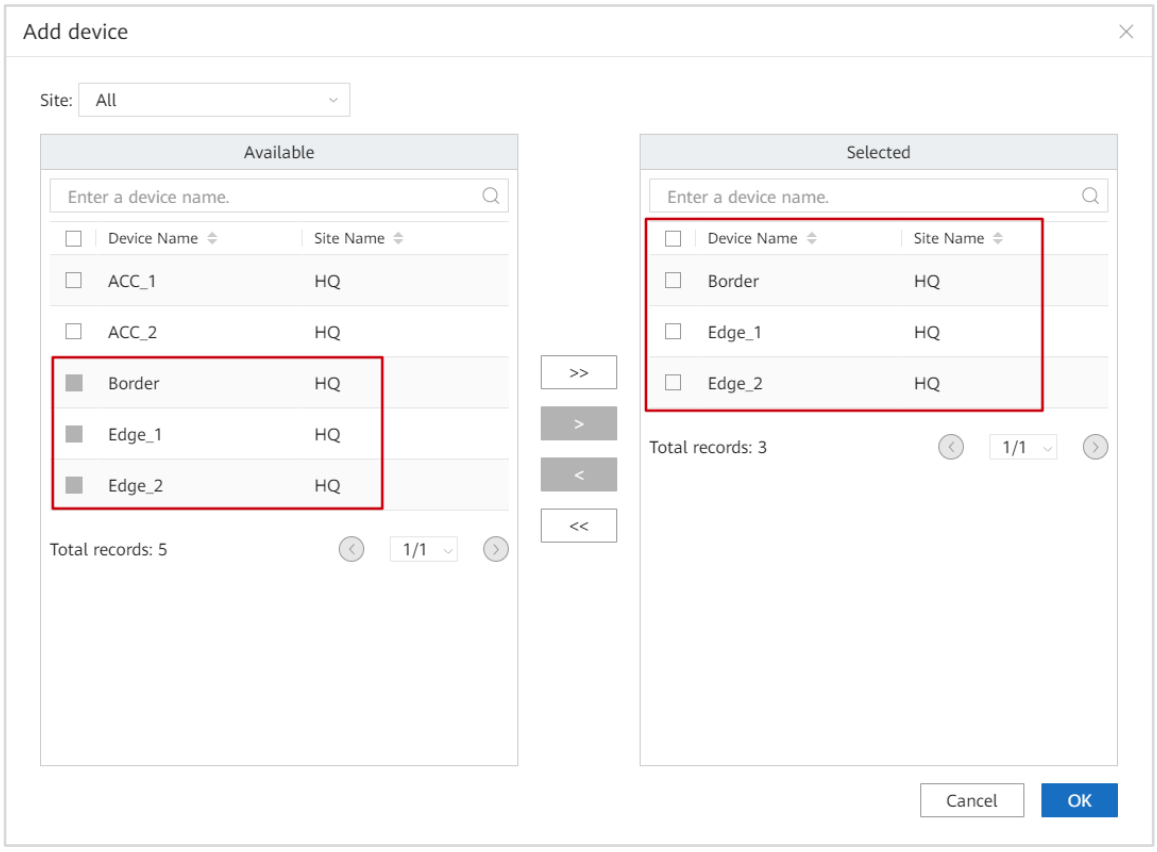
Step 2 Create a policy matrix.

# Switch pages.


Click **Policy Control** to access the **Policy Control Configuration** page. Click **Create** to create a policy matrix.

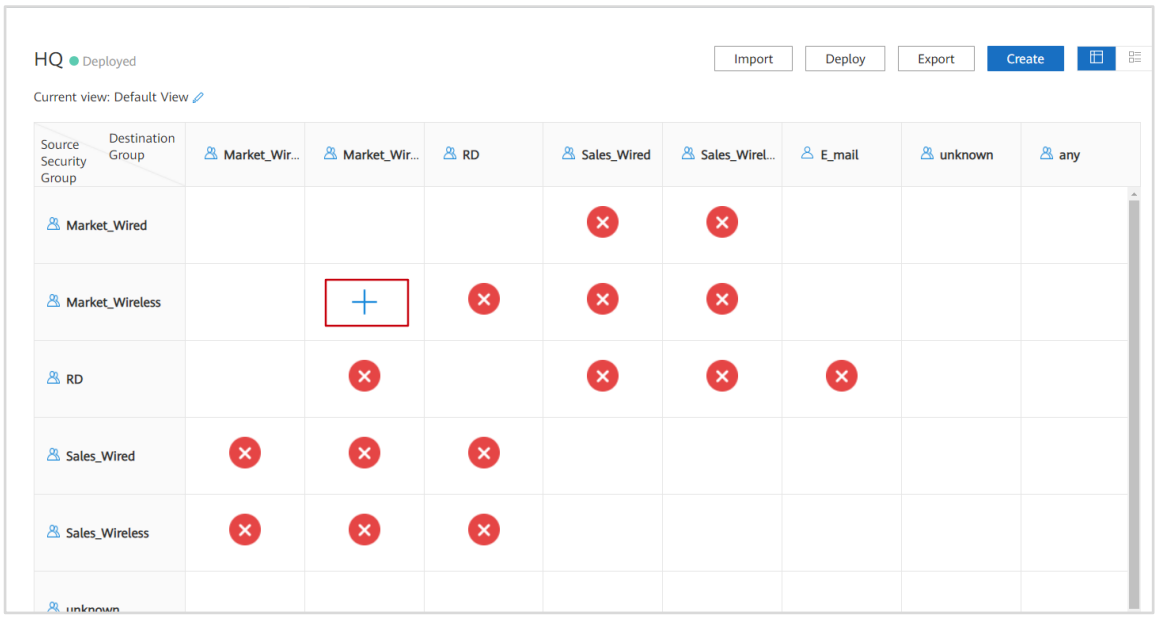
# Configure a policy matrix.



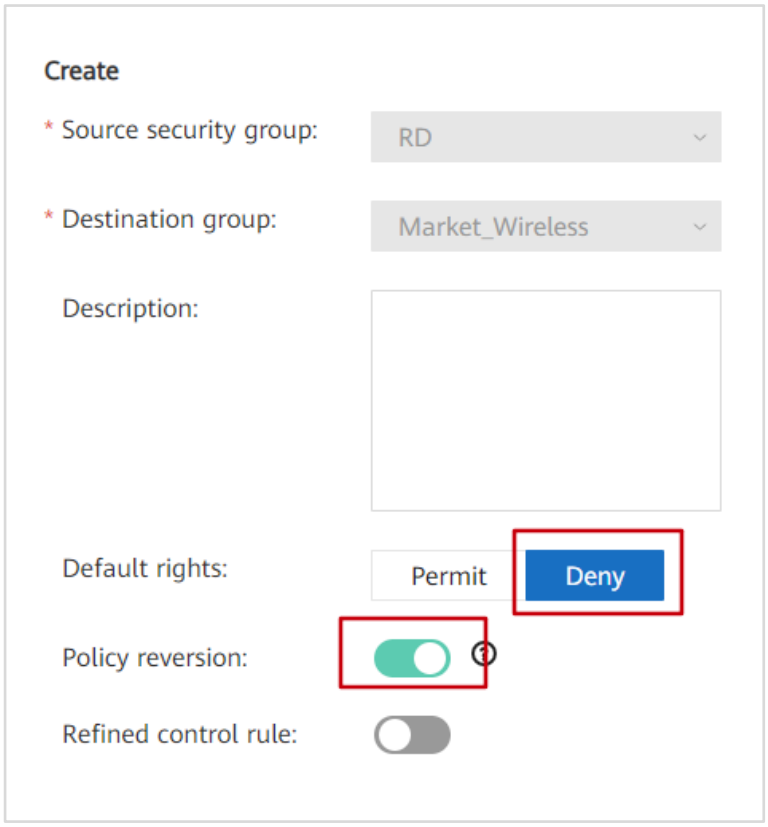



Choose **Border**, **Edge_1**, and **Edge_2** from **Device name**. The communication between wired users is applied through the policy matrix of Edge_1 and Edge_2. In this lab, Border functions as the AC, and WLAN services are forwarded through a tunnel. Therefore, the policy matrix should also be implemented on Border.

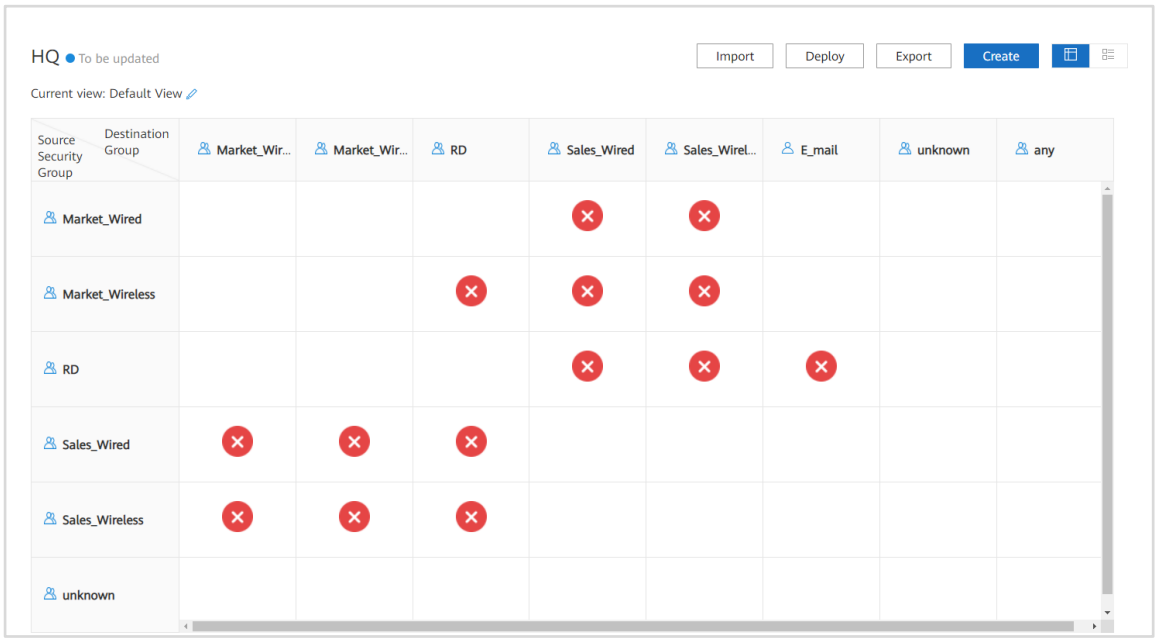
# Configure a policy matrix.




Click **+** in the blank area, and establish a mutual access rule.




Set **Default rights** to **Deny**. Select **Policy reversion** and click **OK**.




Click **Deploy** to deploy the policy matrix to the device.

As planned, users in the Market_Wired group can access the RD VN, but users on the RD VN cannot access E_mail resource group through the policy matrix.

Note that inter-VN communication is not configured between two security groups in the sales department and the RD, and they cannot communicate with each other on the

fabric network. But they can communicate with each other through the external network (AR3). So prohibitions should be defined in the policy matrix.

In the policy matrix, only the allowed communication should be bypassed.

Step 3 Subscribe to IP-security group entries.

It is not required in this scenario.

### 6.1.2.7.2 Access Authentication

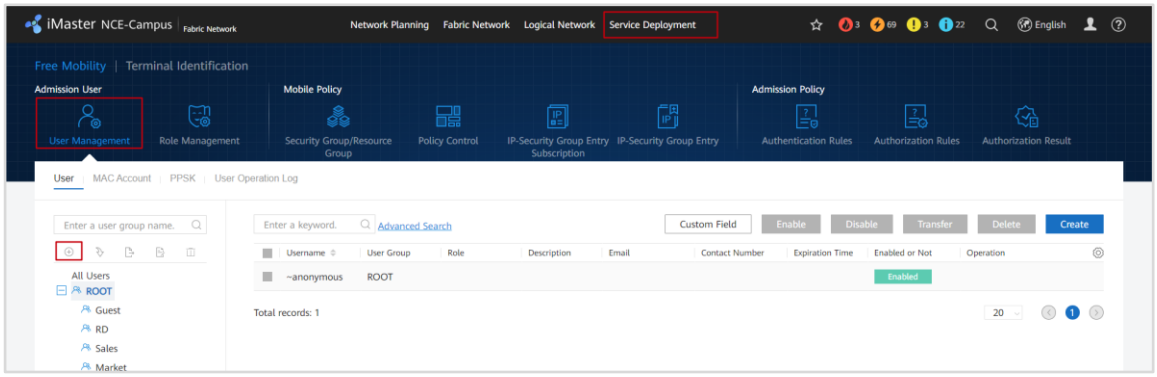
For user access authentication, configure user groups and authentication accounts on the user management page of the controller. Users can use the configured accounts for access authentication.

After configuring user accounts, configure authentication and authorization, including authentication rules, authorization results, and authorization rules.

Step 1 Create user groups and user accounts.

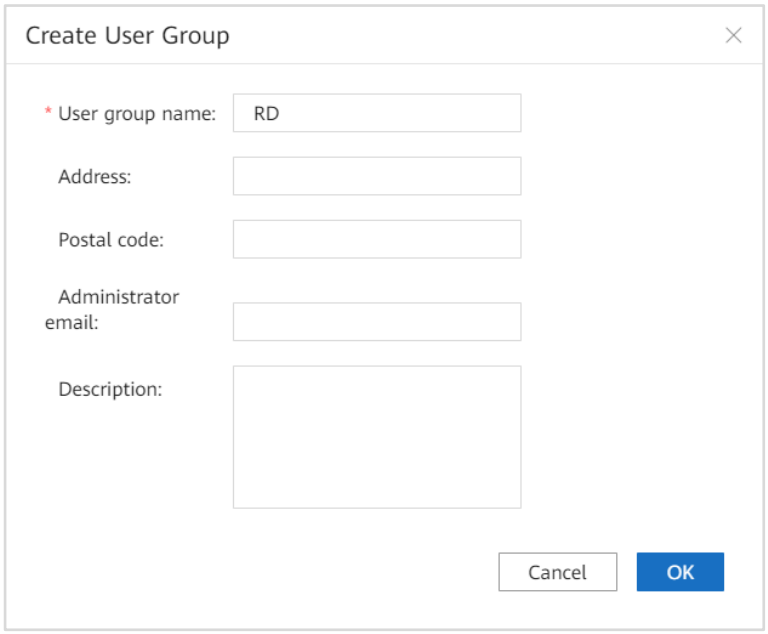
Create user groups and users by referring to user authentication account plans.

# Access the **User Management** page.



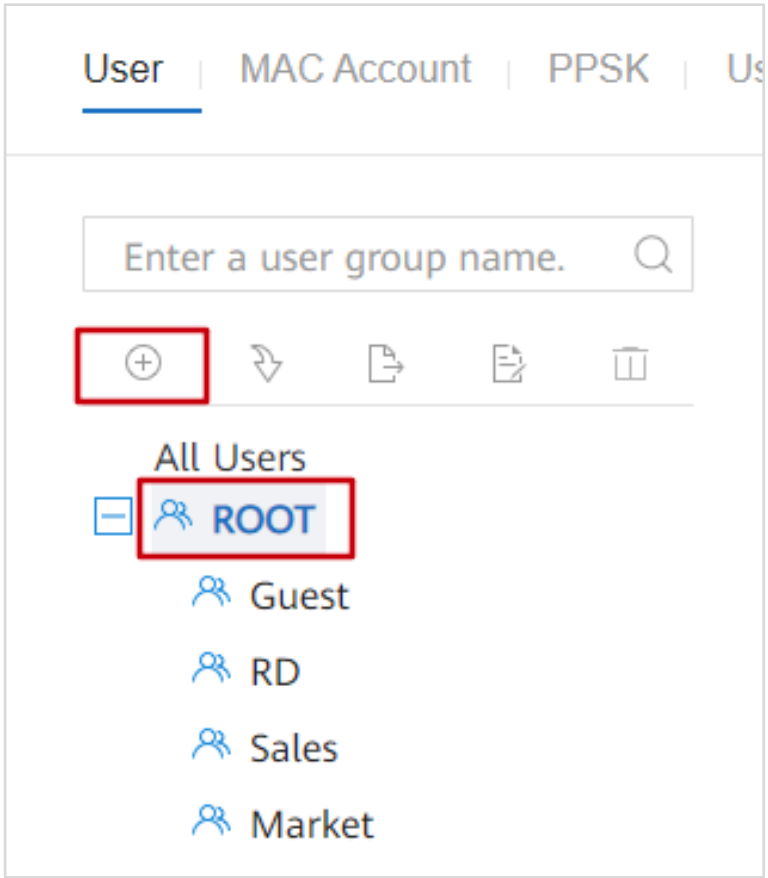

On the home page of iMaster NCE, choose **Deployment Scenario** > **Fabric Network**. Click **Service Deployment**. Click **User Management** to access the user management page.

# Create user groups.



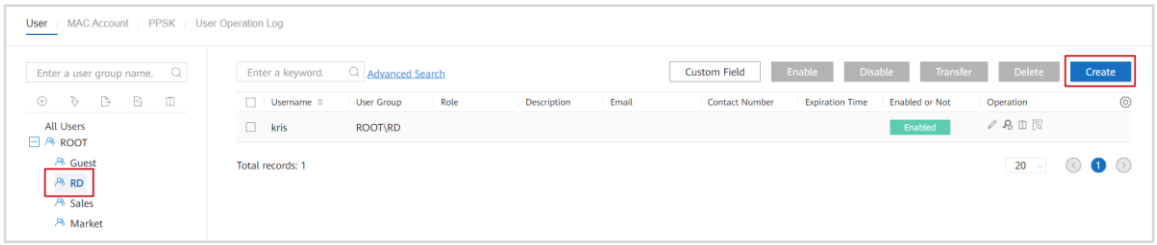

Create a user group **RD**.

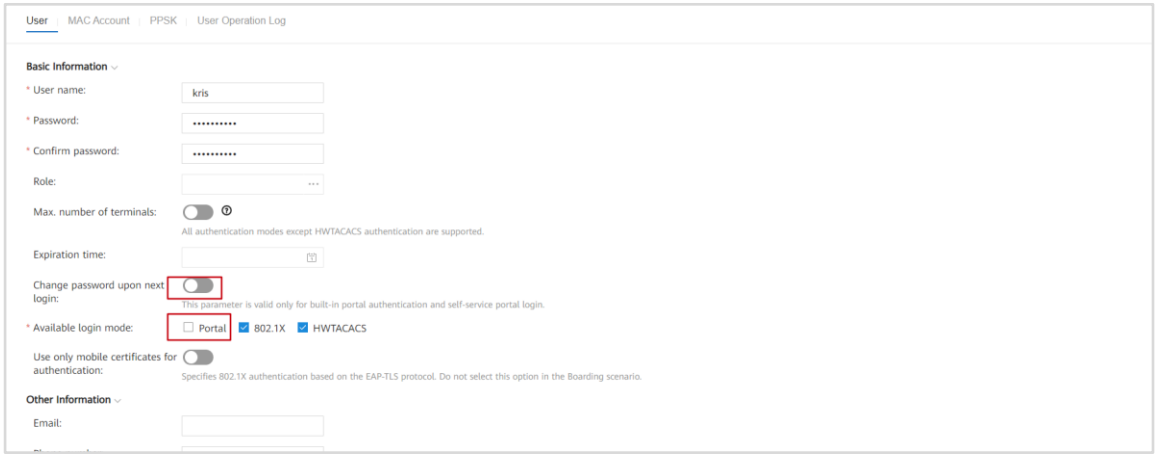
Use the same method to create user groups **Sales** and **Market**.




Note: After creating a user group, click **ROOT** so that the newly created user group belongs to **ROOT**. Otherwise, the user group is nested and created in the previous group.

# Create user accounts.




Create an account in the corresponding user group.

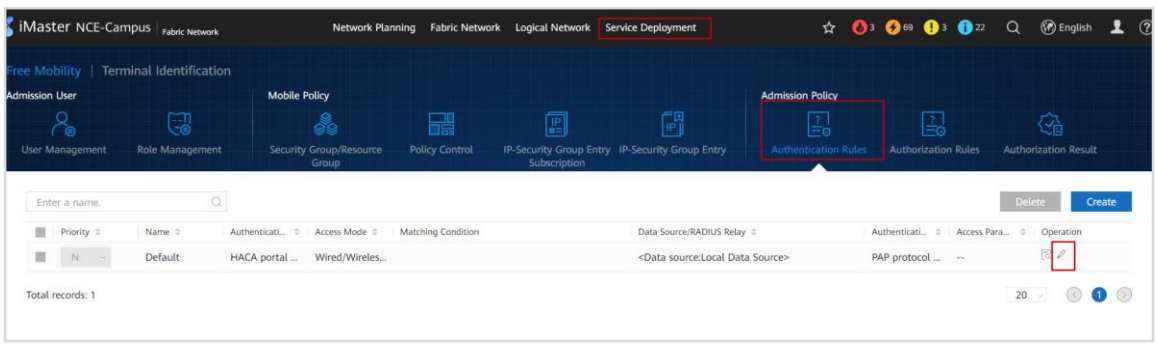
Take user kris (in the RD user group) as an example. Deselect **Change password upon next login**. For users on the RD VN, Portal authentication is unnecessary, so deselect **Portal** in **Available login mode**.

Repeat the steps to create another two accounts in the corresponding user groups.

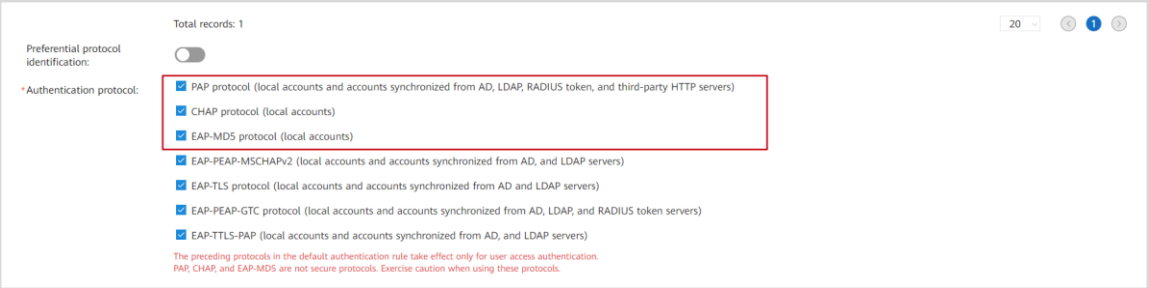
Step 2 Create authentication rules.

Complete the configuration of authentication rules according to the plan.

# Switch pages.

In this lab, you do not need to manually create authentication rules. The Default authentication rules of the controller can be used as the authentication rules for all users directly.



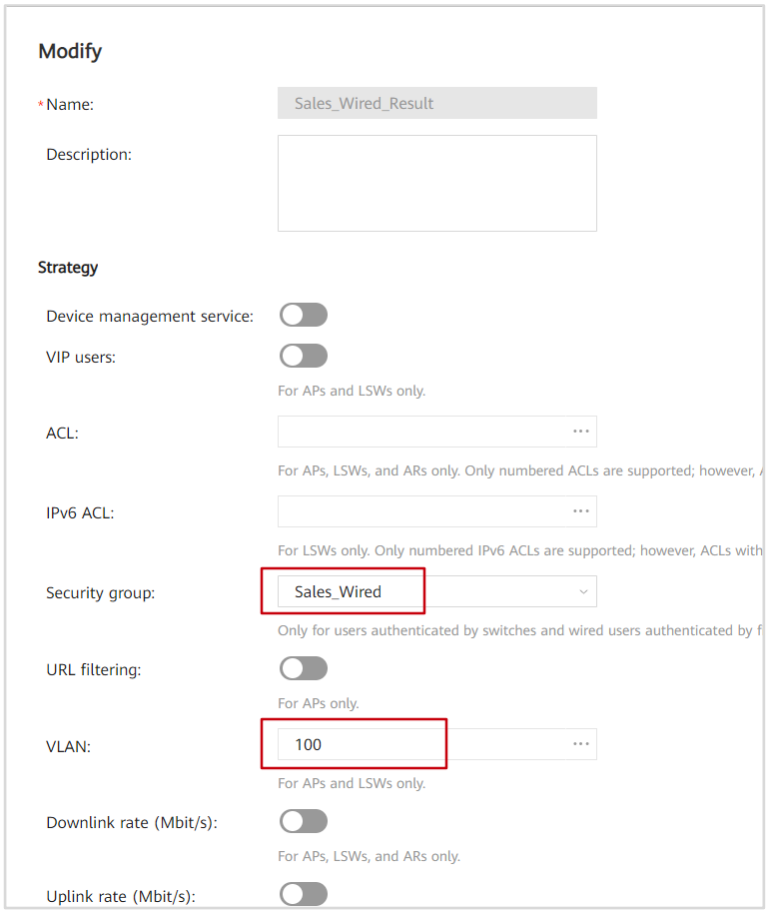

However, the authentication rules should be modified in this example. Click **Modify** in the **Operation** column next to the rule **Default**, and edit the default rules. Select the three authentication protocols shown in the preceding figure.

Step 3 Create an authorization result.

According to the plan, complete the creation of authorization results. In this step, create different authorization results for the wired and wireless users of sales and marketing departments as well as RD.

# Switch pages.

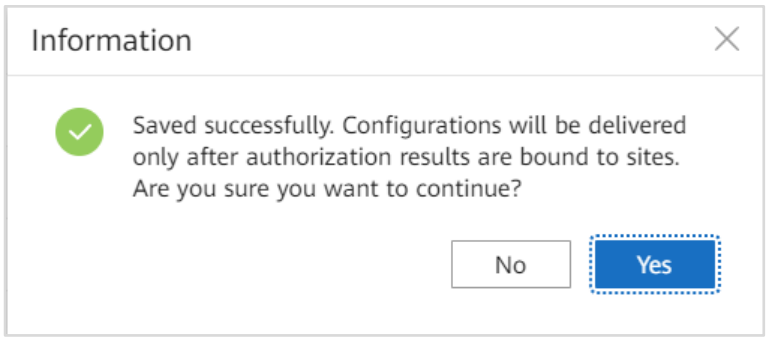

Click **Create** to create new authorization results.

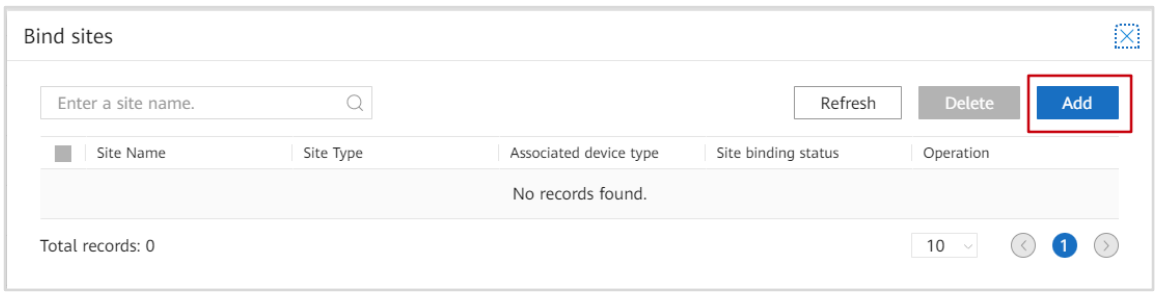


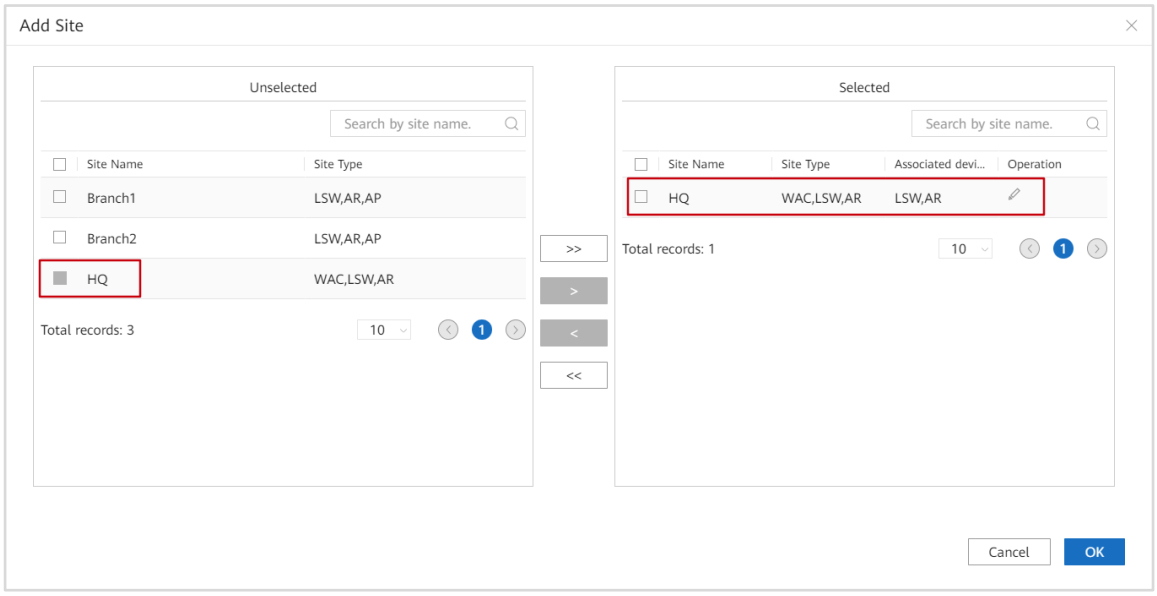

Authorize security groups and VLANs in the authorization results.

Click **OK**.

The system then prompts you to bind the authorization results to specified sites. Click **Add**, as shown in the following figure.



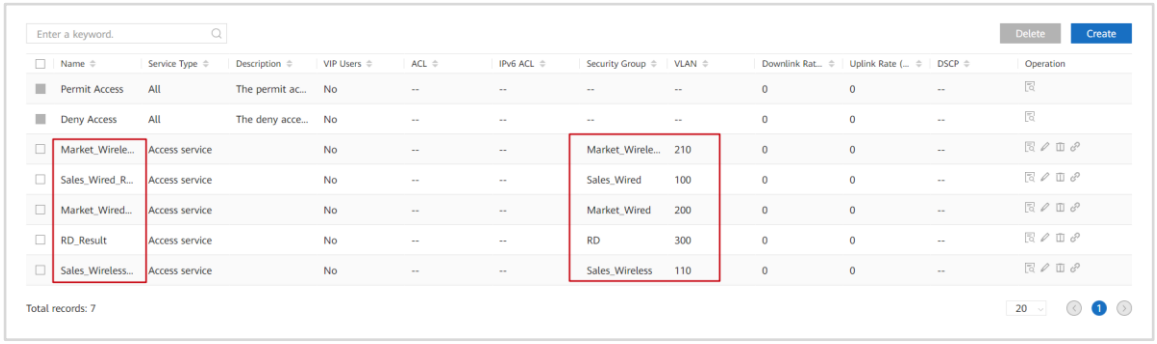






Choose **HQ**, and add it to the **Selected** area. Then click **OK**.

The Sales_Wired_Result configuration is complete.

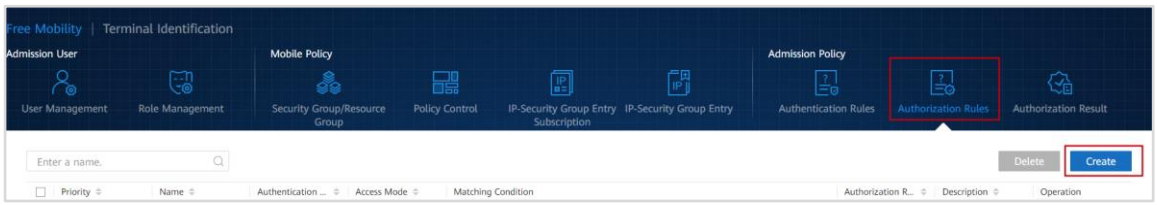
Repeat the same steps to create the other four authorization results and bind them to site **HQ**. Note that the authorized VLANs and security groups in the four authorization results are different. You must enter information according to the data plan of authorization results.



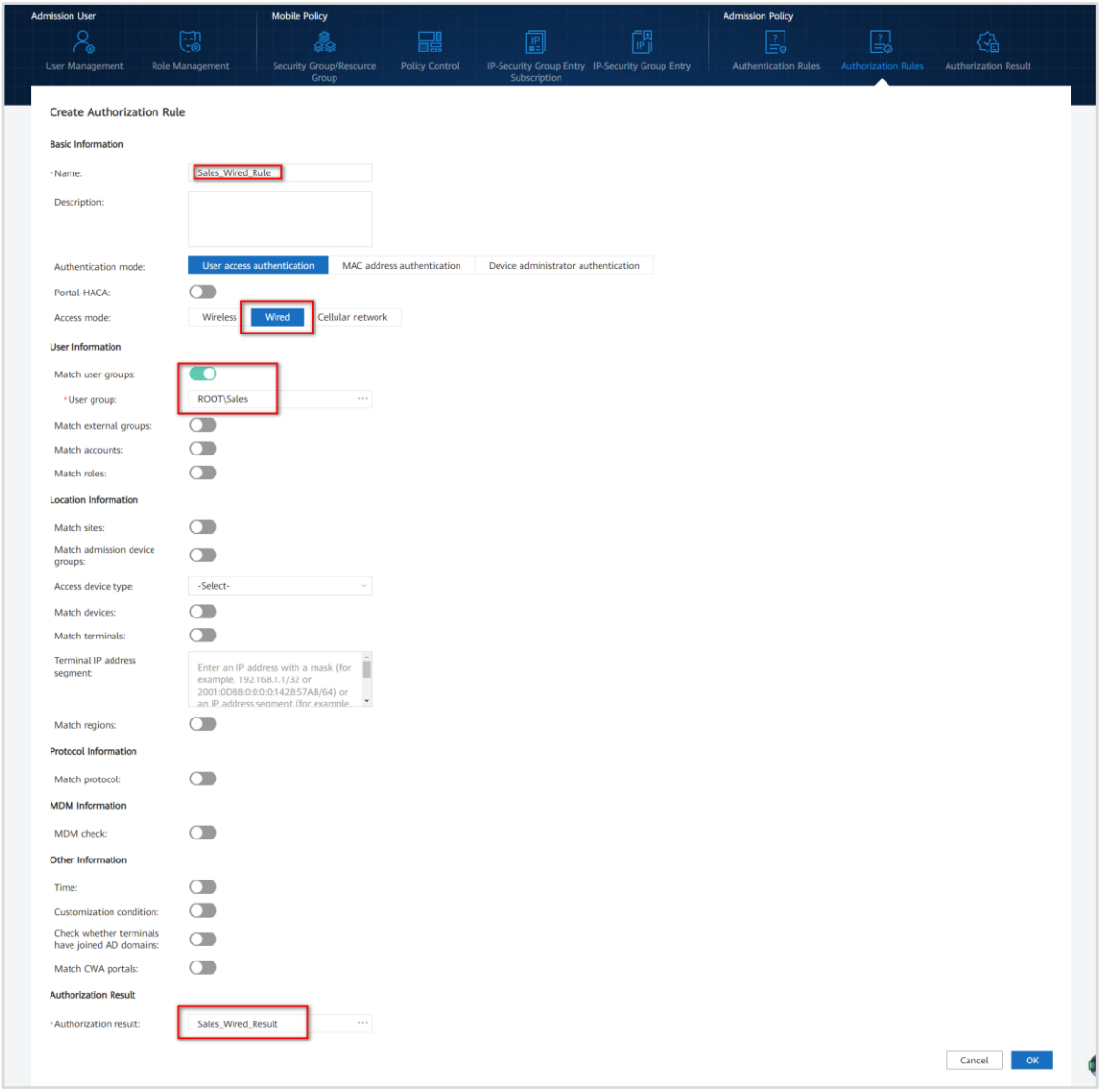

## Step 4 Create authorization rules.

Create authorization rules as planned. In this step, create different authorization rules for the wired and wireless users in the Sales, Market, and RD user groups.

# Switch pages.

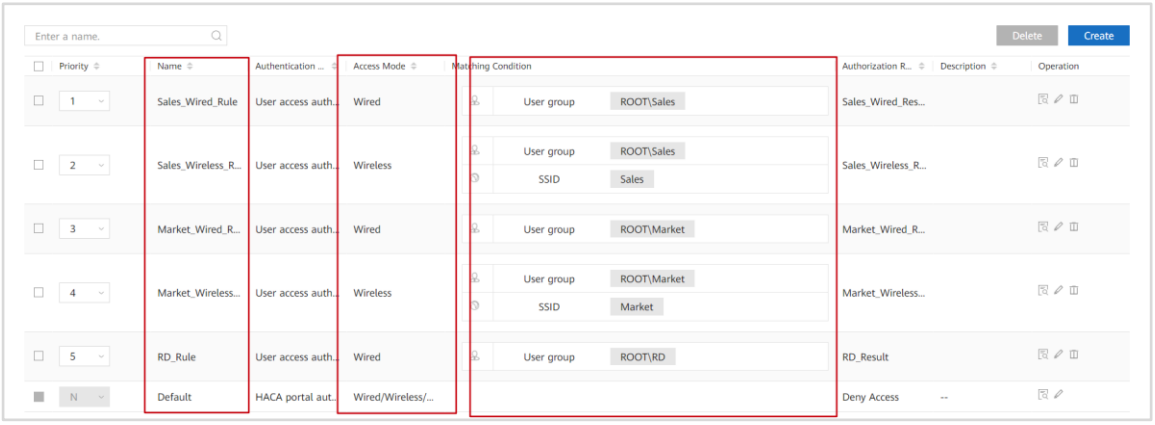



On the current page, click **Authorization Rules**.




Set **User access authorization** to **Wired**. Enable **Match user groups**. Set **Authorization result** to **Sales_Wired_Result**. After the configuration is complete, click **OK**.

Configure the other authorization rules based on the plan.




The configuration of authorization rules is complete.

## 6.1.2.8 WLAN Services

In this lab, use the native AC of Border as the AC for AP onboarding.

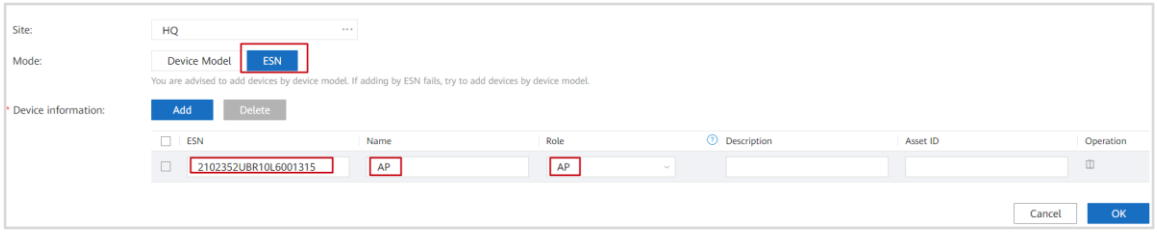
To configure Border as the AC that manages APs, perform the following configurations:

1. In the device management, add an AP to the HQ site.
2. In the site configuration, allocate the AP to Border.
3. In the site configuration, create a network segment for AP onboarding and management, and set the VLAN ID of the network segment to 2.
4. In the site configuration, configure VLAN 2 as the wireless PnP management VLAN of Border.
5. On the web page of Border, manually configure VLANIF 2 as the source interface of the CAPWAP tunnel.

Then log in to the web page of Border based on the plan, and configure WLAN services.

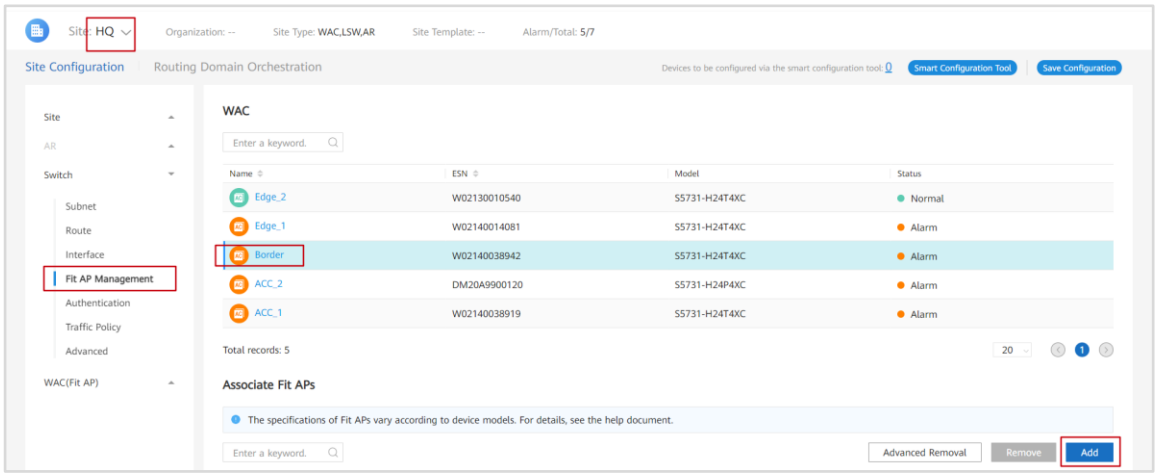
Step 1 Configure AP onboarding.

# Add an AP.

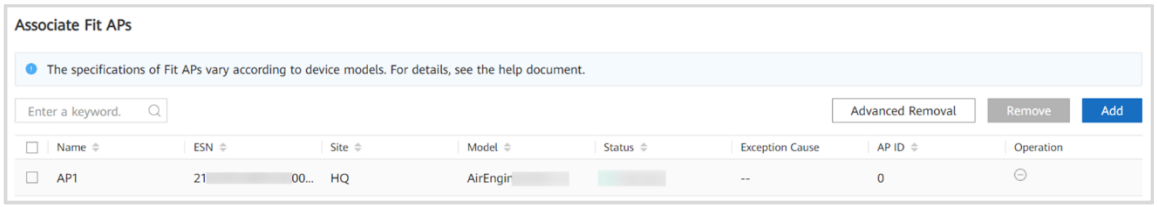



Choose **Design** > **Device Management** to add devices to the HQ site (select the NETCONF protocol). Add an AP by setting **ESN**. It is normal that the AP cannot be onboarded and its status is unregistered because related configurations are incomplete.

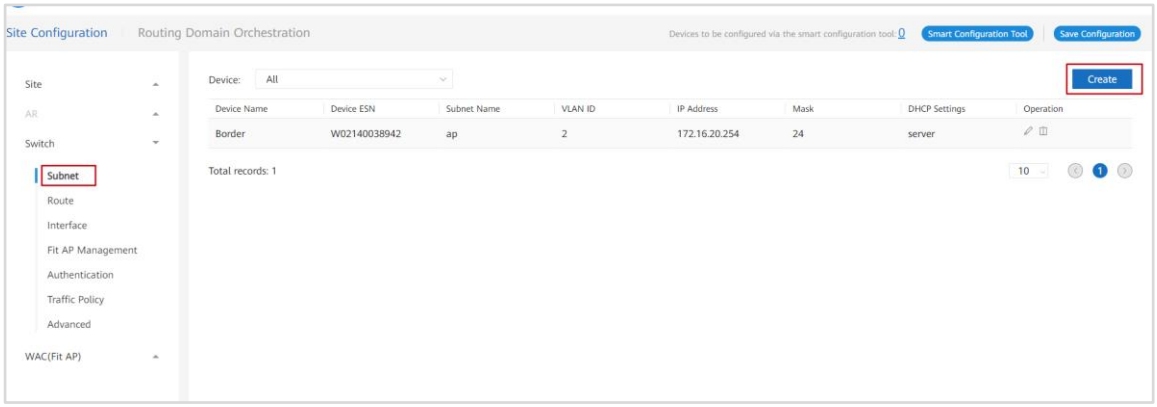
# Allocate an AP.



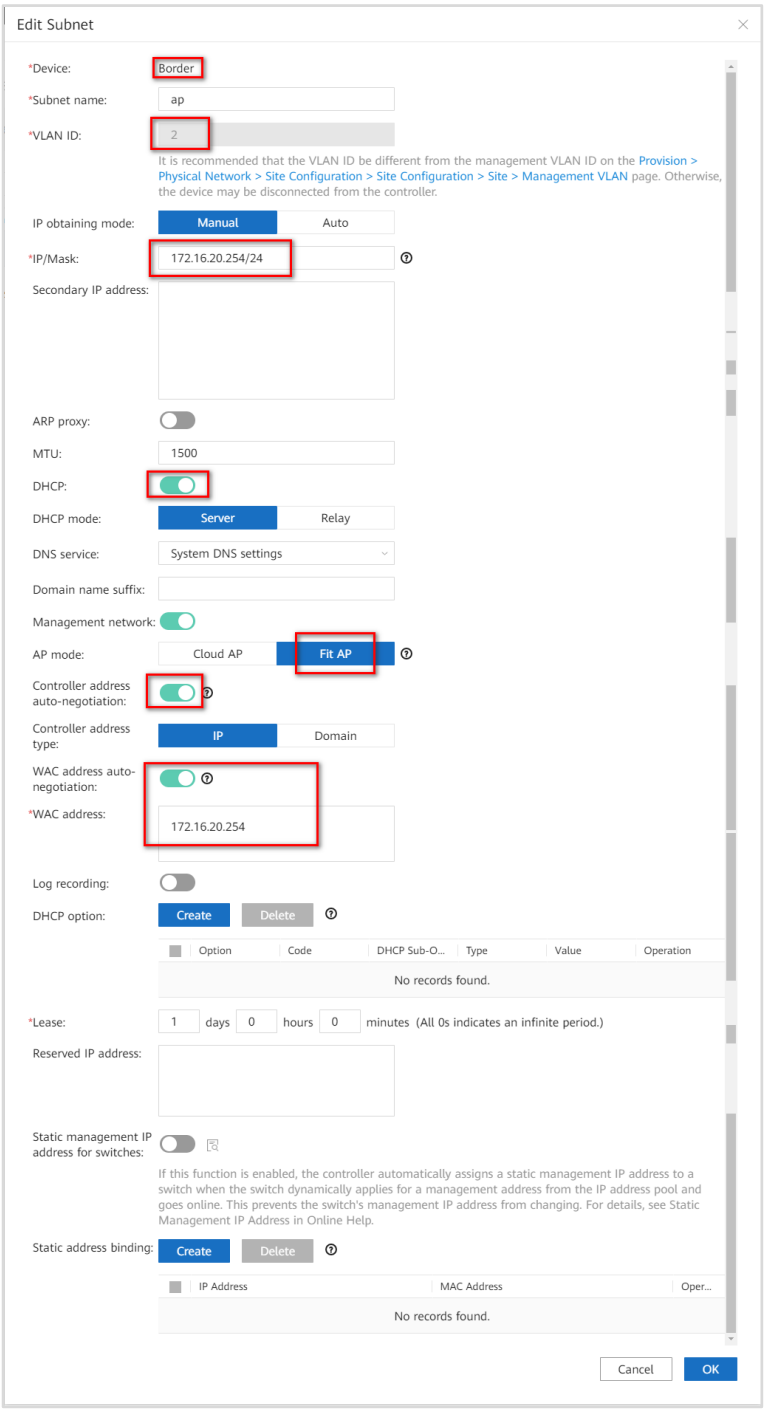

Choose **Provision** > **Site Configuration** > **Switch** > **Fit AP Management**. Click **Border**, and click **Add** to allocate AP1 to Border.




# Create a subnet.

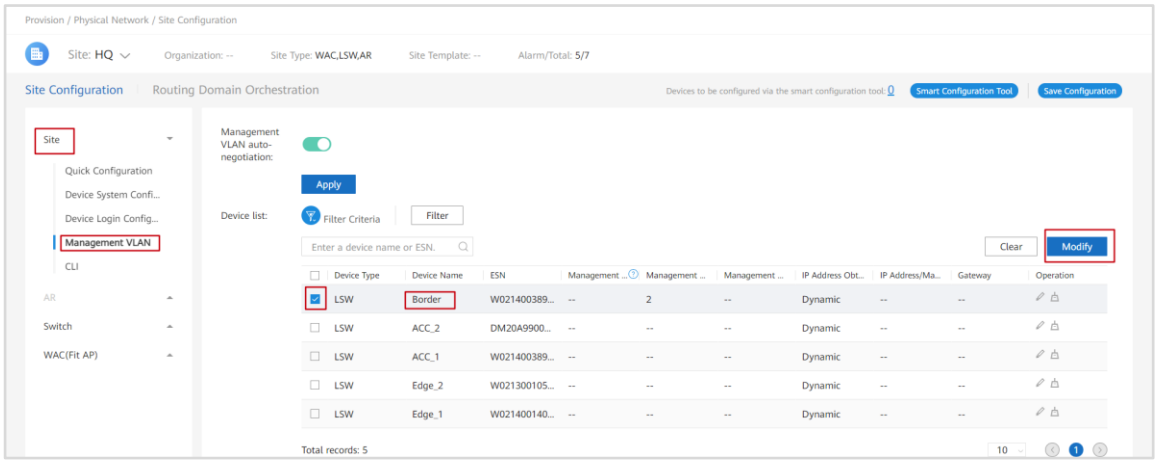



On the current page, click **subnet**. Then click **Create** to create a subnet for AP onboarding.

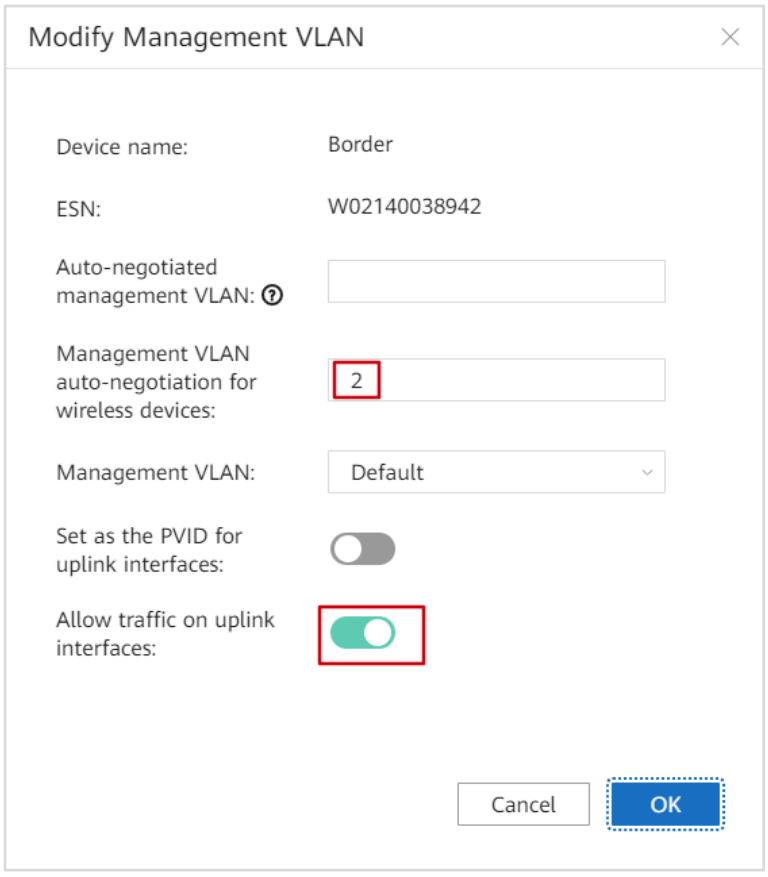



Set the subnet parameters according to the preceding figure. Through the subnet, the AP will obtain IP addresses of the AC and the controller.

# Set the wireless PnP VLAN of Border.

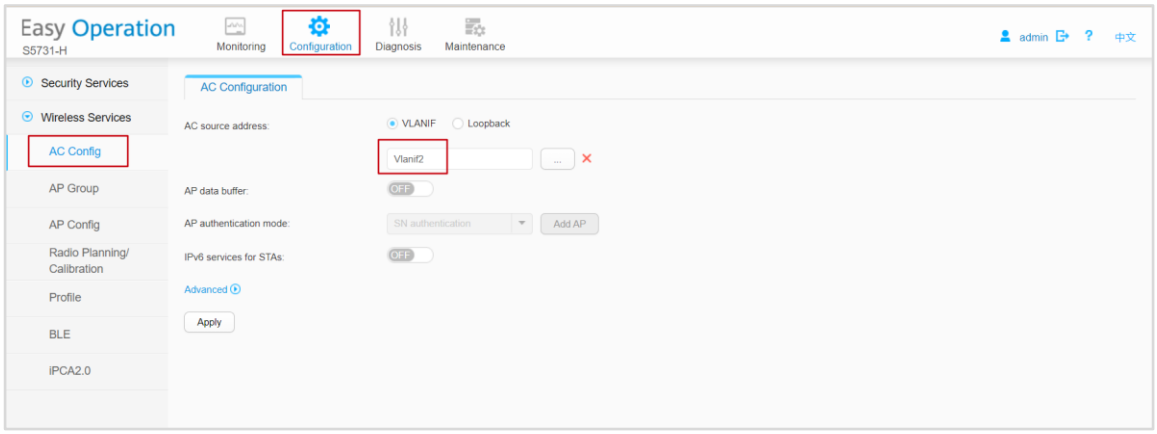



On the current page, choose **Site** > **Management VLAN**, select **Border**, and click **Modify**.




Set **Management VLAN auto-negotiation for wireless devices** to 2. Border will deliver the configuration to downstream devices through the LLDP protocol. After ACC_2 recognizes that GE0/0/24 is connected to the AP, it automatically adds the AP to VLAN 2, and devices between ACC_2 and Border will allow packets from VLAN 2 to pass through.

# Set the source address of the CAPWAP tunnel on the web page of Border.

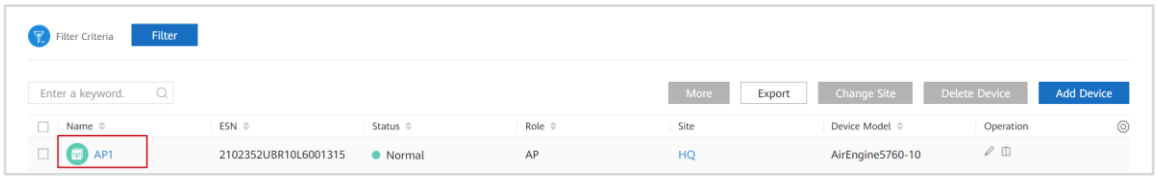



Set the CAPWAP communication source interface to VLANIF 2.

# Check the AP onboarding status.

```
[Border]display ap all
Total AP information:
nor   : normal          [1]
ExtraInfo : Extra information
P     : insufficient power supply
-------------------------------------------------------------------------------------------------
ID   MAC             Name Group  IP              Type              State  STA  Uptime     ExtraInfo
-------------------------------------------------------------------------------------------------
0    f4de-af36-b580  AP1  default 172.16.20.183 AirEngine5760-10  nor    0    5H:13M:30S  -
-------------------------------------------------------------------------------------------------
```

AP1 is recognized and added to VLAN 2. After AP1 obtains the IP address and AC address from the address pool on VLANIF 2, it registers with Border.




In the device management, AP1 also registers with the controller.

## Step 2 Perform Portal service-related configurations.

In the lab, iMaster NCE is used as the Portal server. The authentication device communicates with the Portal server through AR3 source address translation. To ensure that the Portal server can communicate with the authentication device, the port used by the authentication device to listen to Portal server messages needs to be mapped to the public network.

In the lab, Border is used as the WAC. Therefore, the IP address used for communication between Border and the Portal server and the corresponding port number (UDP port 2000) need to be mapped to the public network.

# Configure static NAT on egress router AR3.

```
[AR3]interface GigabitEthernet0/0/9
```

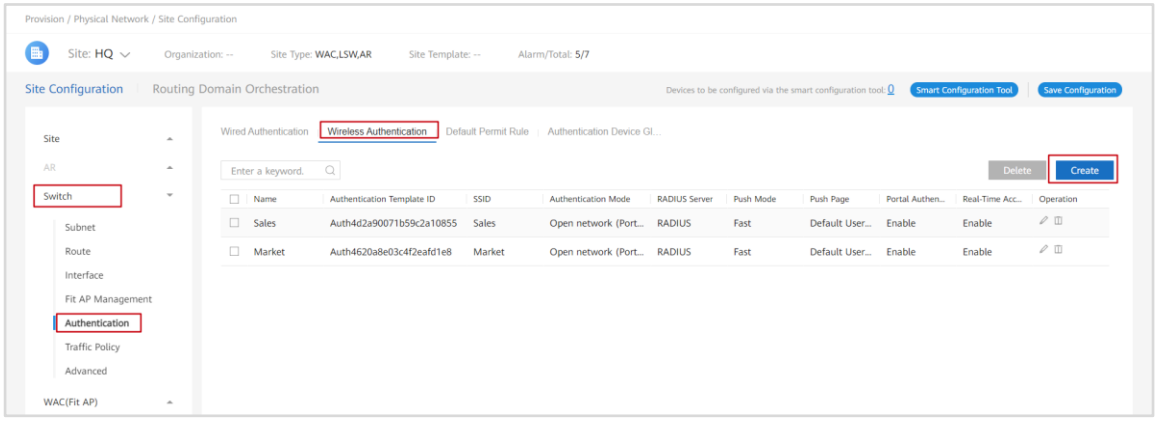
```
[AR3-GigabitEthernet0/0/9] nat static protocol udp global current-interface 2000 inside 172.16.10.1 2000 netmask 255.255.255.255
```

172.16.10.1 is the VLANIF interface address of Border (obtained through DHCP). Static address binding is configured on AR3 to use a fixed private IP address during mapping.

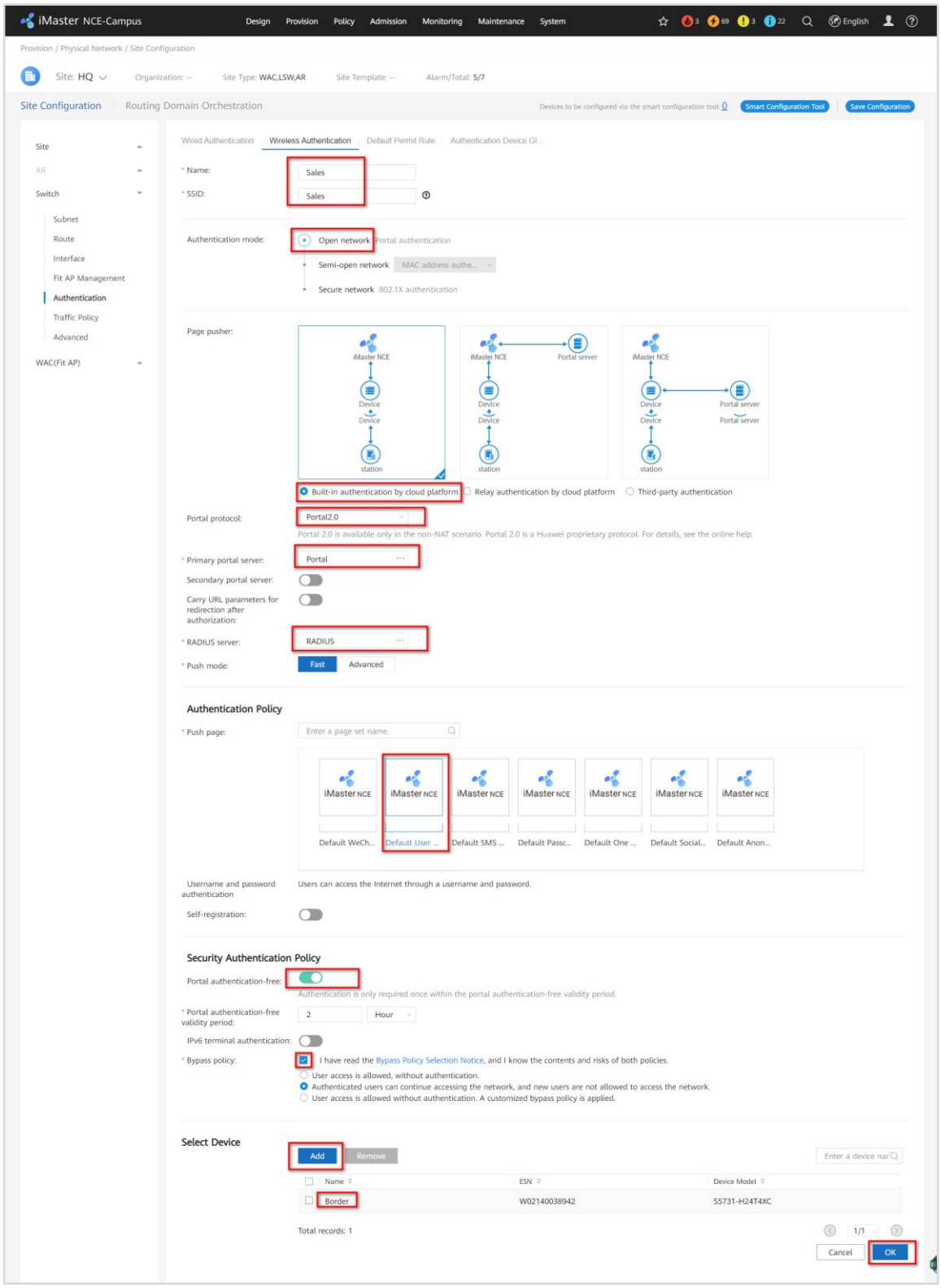
Step 3 Configure wireless services on the controller.

Complete wireless configurations on the controller and deliver Portal authentication configurations to Border.

# Configure wireless authentication.

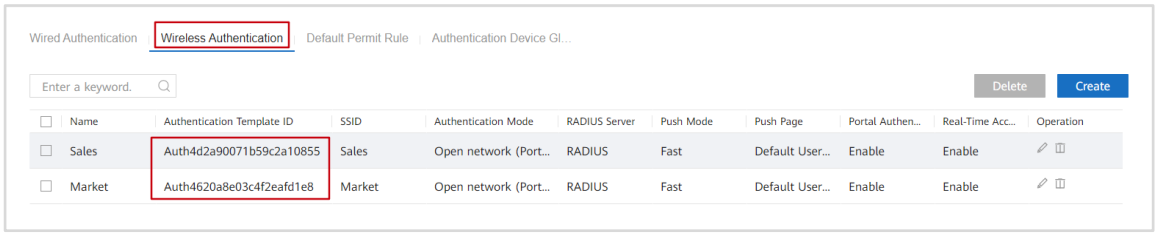



Choose **Site Configuration** > **Switch** > **Authentication**, and click **Wireless Authentication**. Configure wireless authentication.




Complete Sales wireless authentication configurations as planned. Configurations for Market are the same as that for Sales and are not described here.

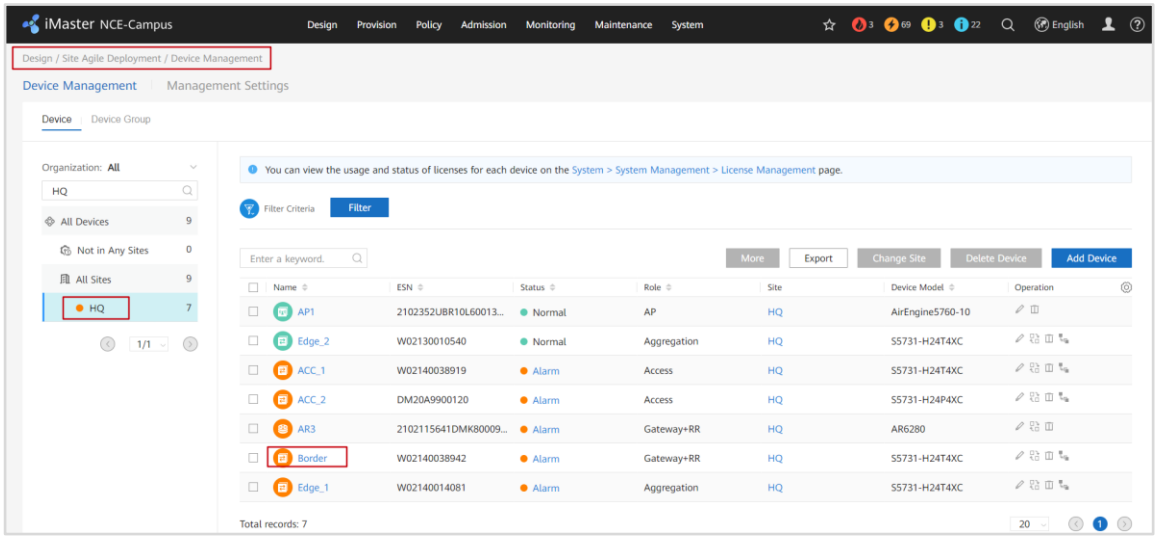
# Check the configuration of wireless authentication.



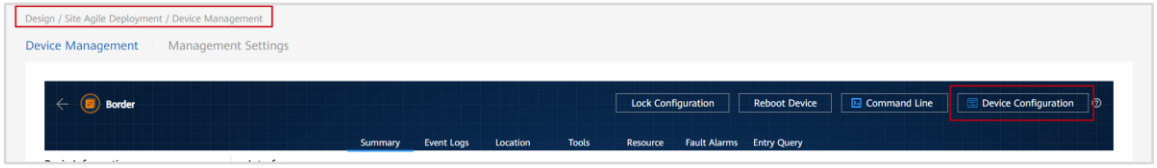

To check the configuration of wireless authentication, pay attention to the authentication profile ID. Associate the following-up configurations of wireless services on Border with corresponding profiles, and remember the corresponding profile ID generated for different service SSIDs.

Step 4 Log in to the web page of Border.

# Access the device management page.




Choose **Design** > **Device Management**, select **HQ**, and click **Border** to access the device management page.



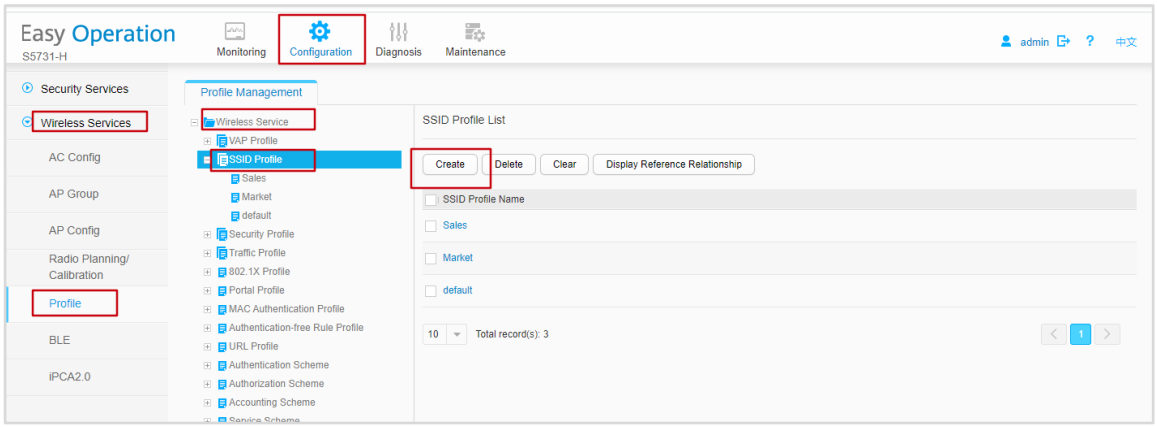

Click **Device Configuration** in the upper right corner. The device management page is displayed. (Some browsers may block the pop-up pages, pay attention to the prompt).

Then enter the user name and password to access the device management page.

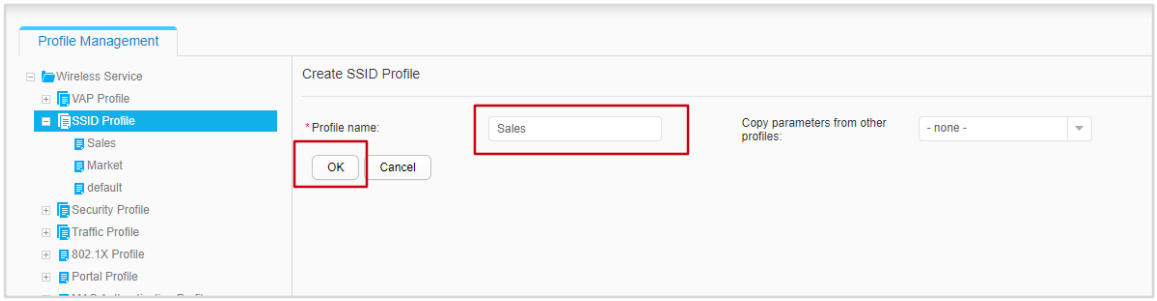
Step 5 Create and configure SSID profiles.

Create different SSID profiles for Sales and Market, and complete configurations as planned.

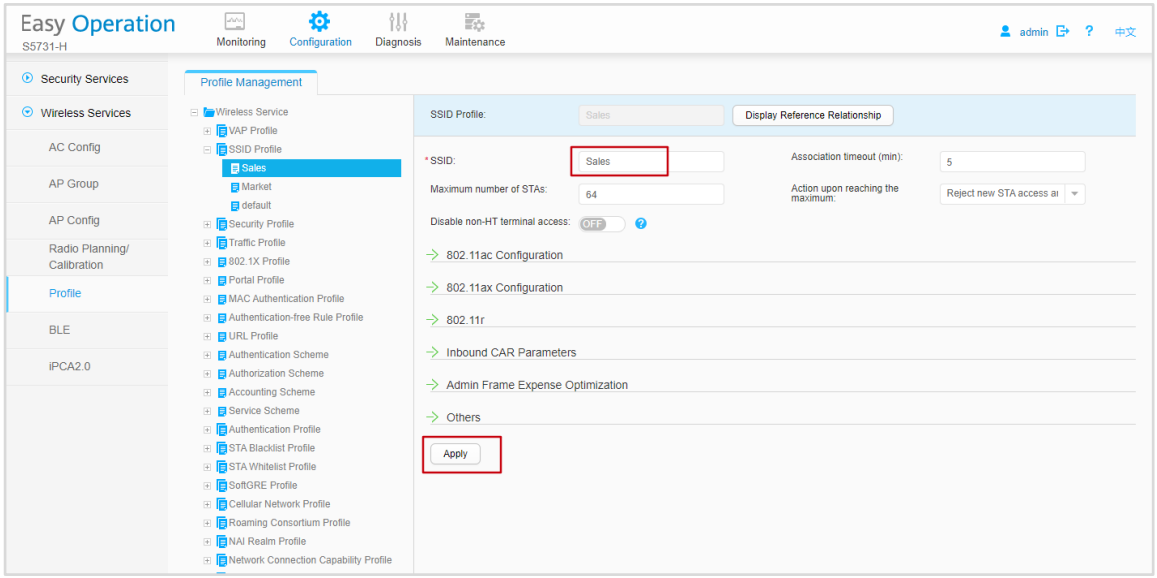
# Create an SSID profile.




Click **Create**.




Create an SSID profile named **Sales**, and click **OK**.



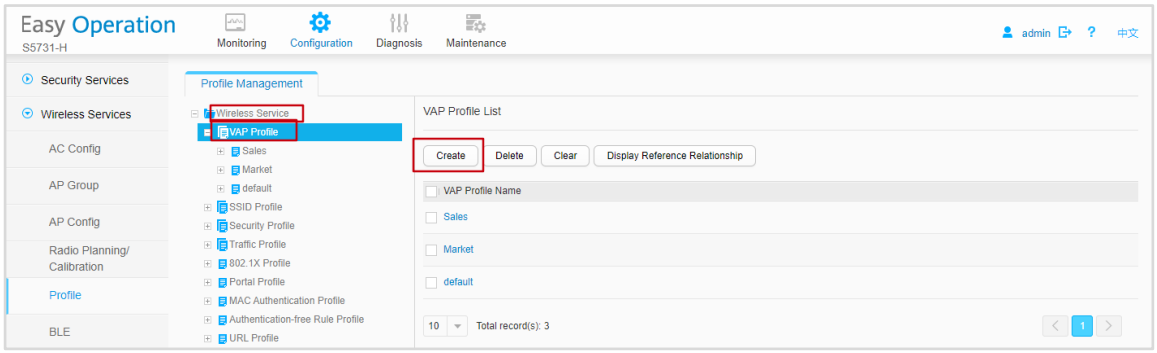

On the page that is displayed, set **SSID name** to **Sales**, and click **Apply**.

The SSID profile **Sales** is created. Use the same method to create an SSID profile **Market**.

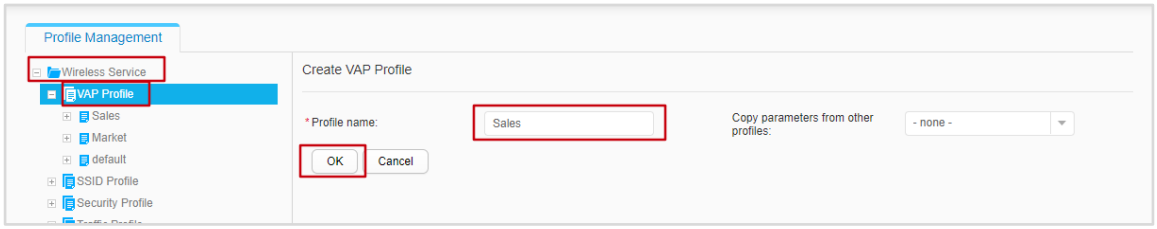
Step 6 Create and configure VAP profiles.

Create different VAP profiles for Sales and Market, and complete configurations as planned.

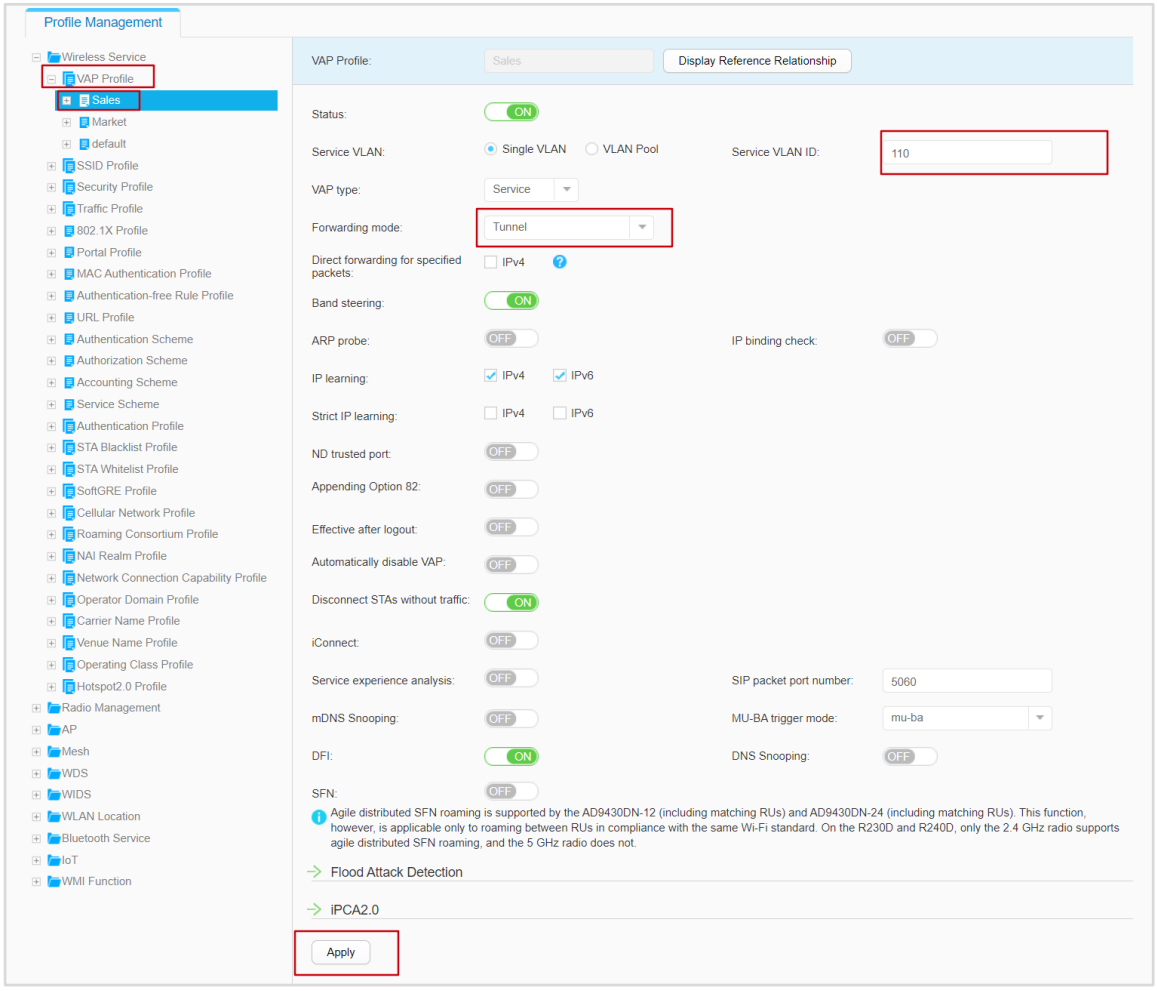
# Create VAP profiles.




Click **Create**.

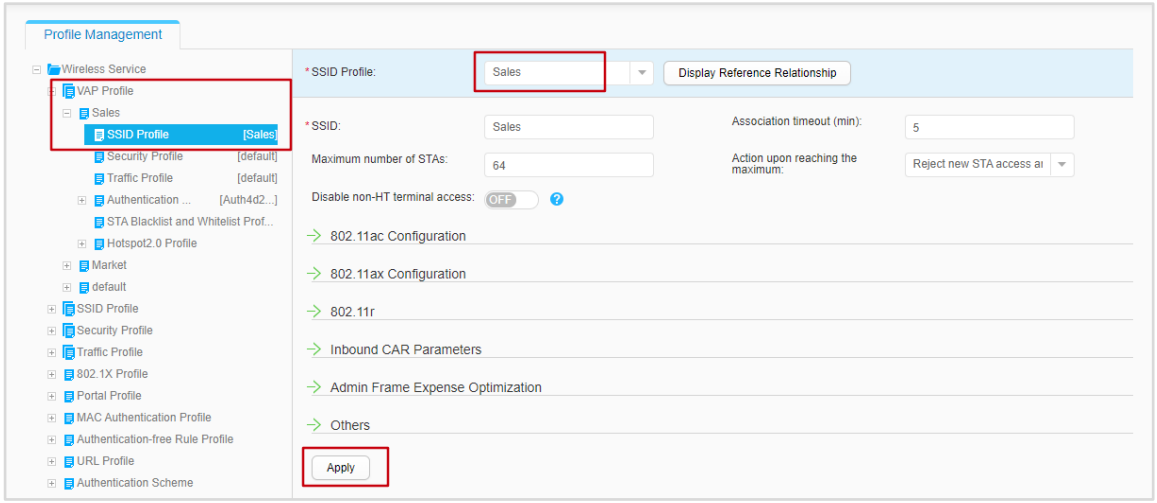



Create a VAP profile named **Sales**, and click **OK**.



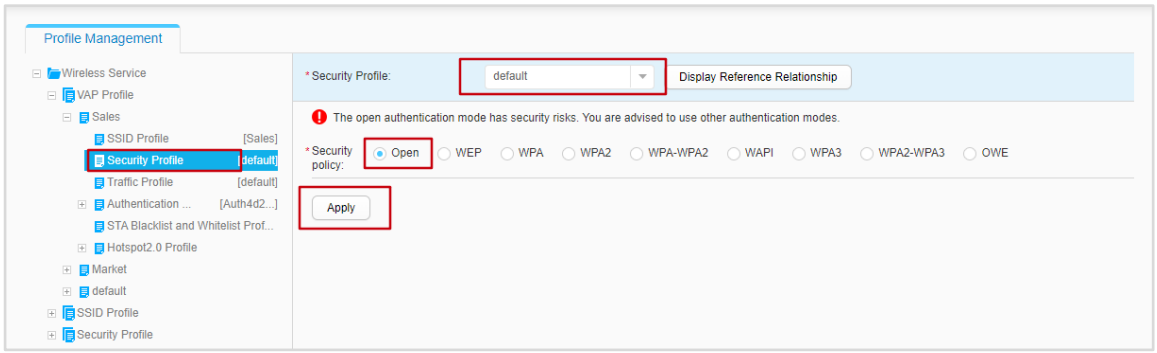
On the page that is displayed, set VAP profile parameters based on the plan (set the service VLAN ID to 110 and set the forwarding mode to tunnel forwarding), and then click **Apply**.

# Bind the SSID profile to the VAP profile.



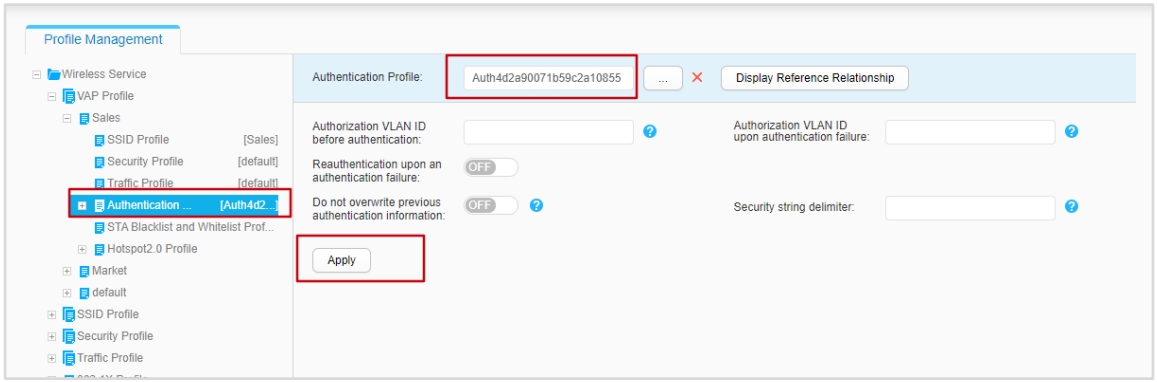

On the **Profile Management** tab page, select the created VAP profile, expand the profile, and click **SSID Profile**. On **SSID Profile** displayed on the right, select SSID profile **Sales** and click **Apply**.

# Configure a security profile.




On **VAP Profile**, click **Security Profile**. Set **Security Profile** to **default** and **Security policy** to **Open**.

# Bind the authentication profile to the VAP profile.



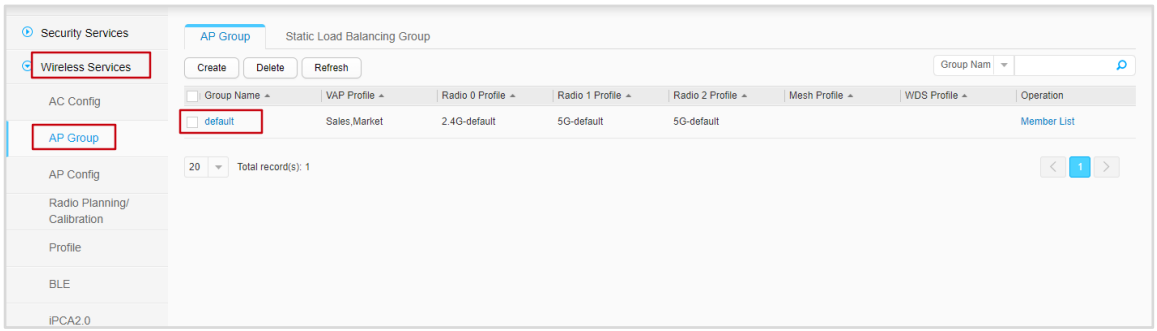
On **Profile Management**, select the created VAP profile, expand the profile, and click **Authentication Profile**. On the page displayed on the right, click **...** next to **Authentication Profile**. In the displayed dialog box, select the wireless authentication profile created for SSID: Sales on the controller.

The VAP profile of Sales is created. Use the same method to configure the VAP profile of Market.

Step 7 Bind a VAP profile to an AP Group.

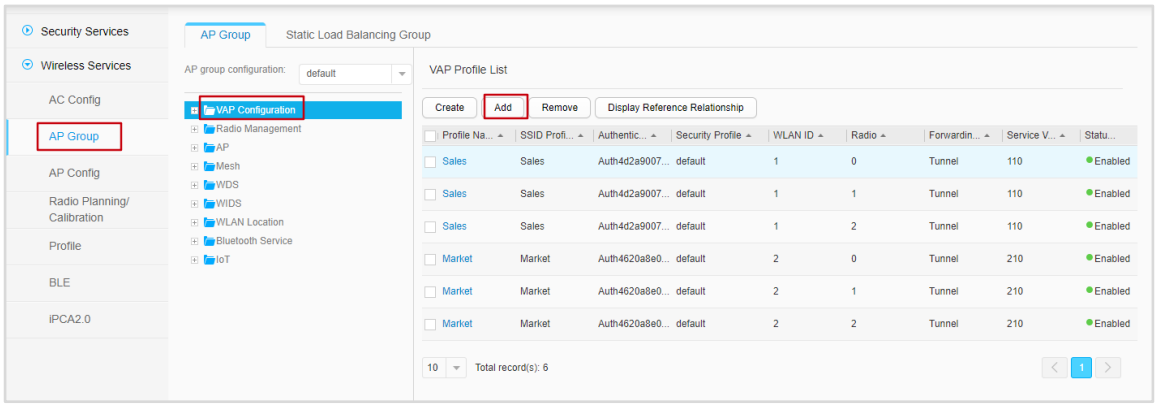
Bind the configured VAP profile to an AP group.

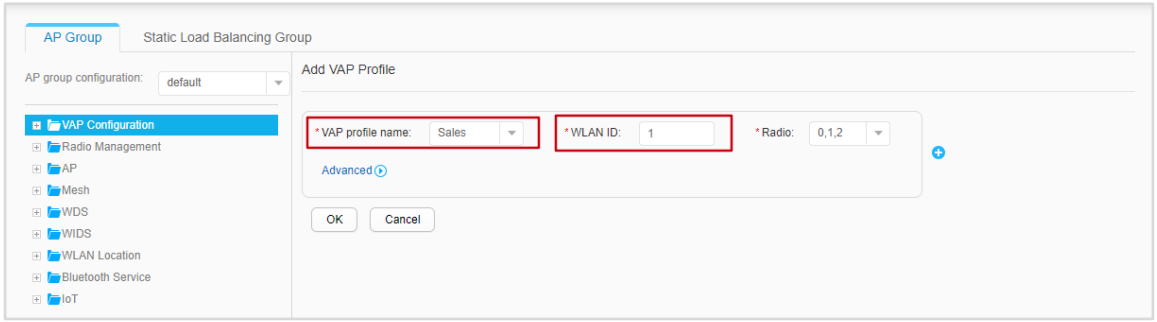
# Enter an AP group.



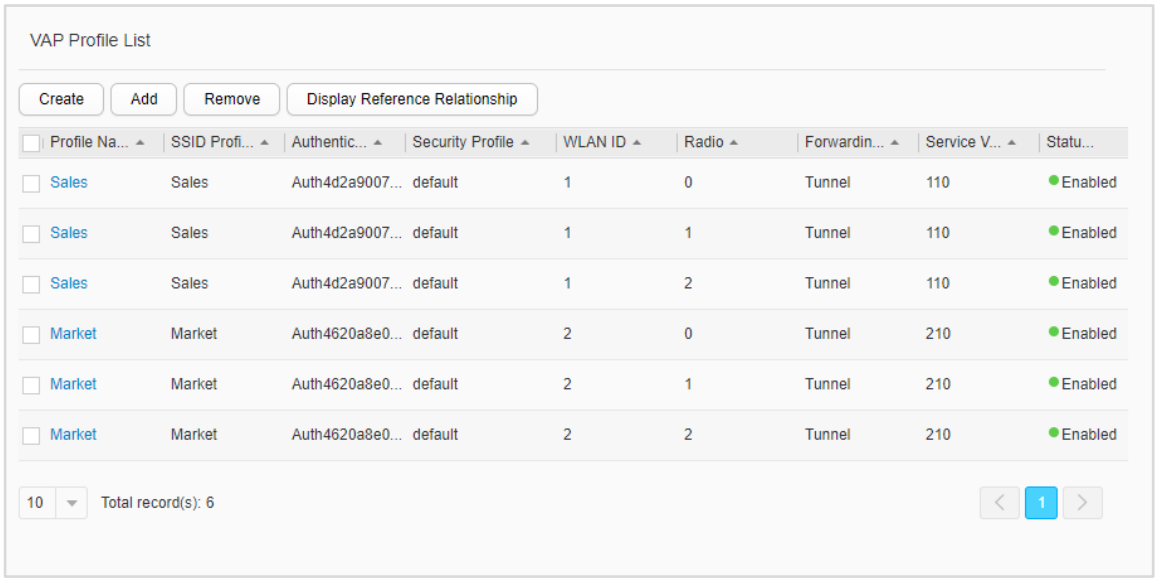

Select **default** on the **AP Group** tab.

# Add a VAP profile.




Add the VAP profile **Sales**, then add the VAP profile **Market**.




The VAP profile configuration is complete.

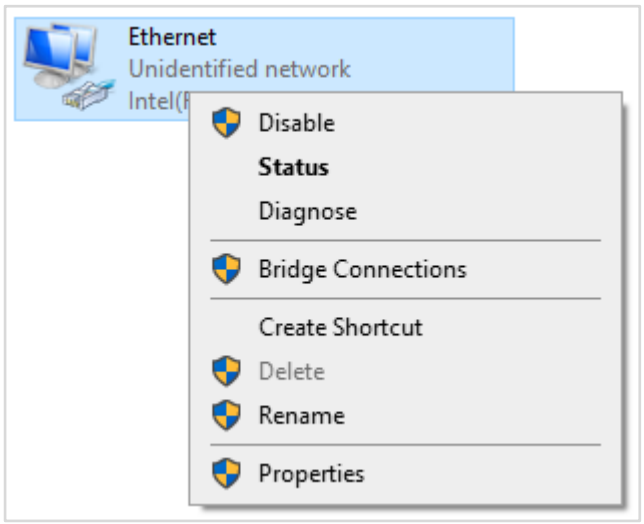
## 6.1.2.9 Verification

### 6.1.2.9.1 Verifying Access Authentication

Step 1 Verify 802.1X authentication.

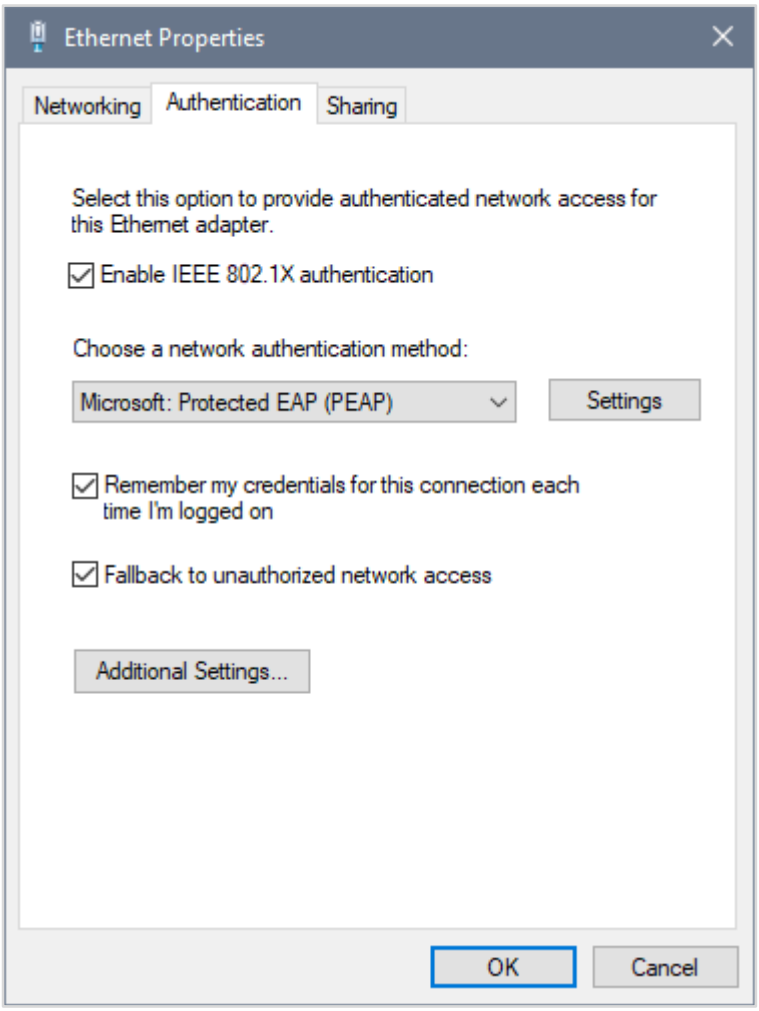
Use PC1 and PC2 for authentication. Test accounts **sales**, **market**, and **kris** respectively. Check the IP address and authorization result of the switch after passing authentication.

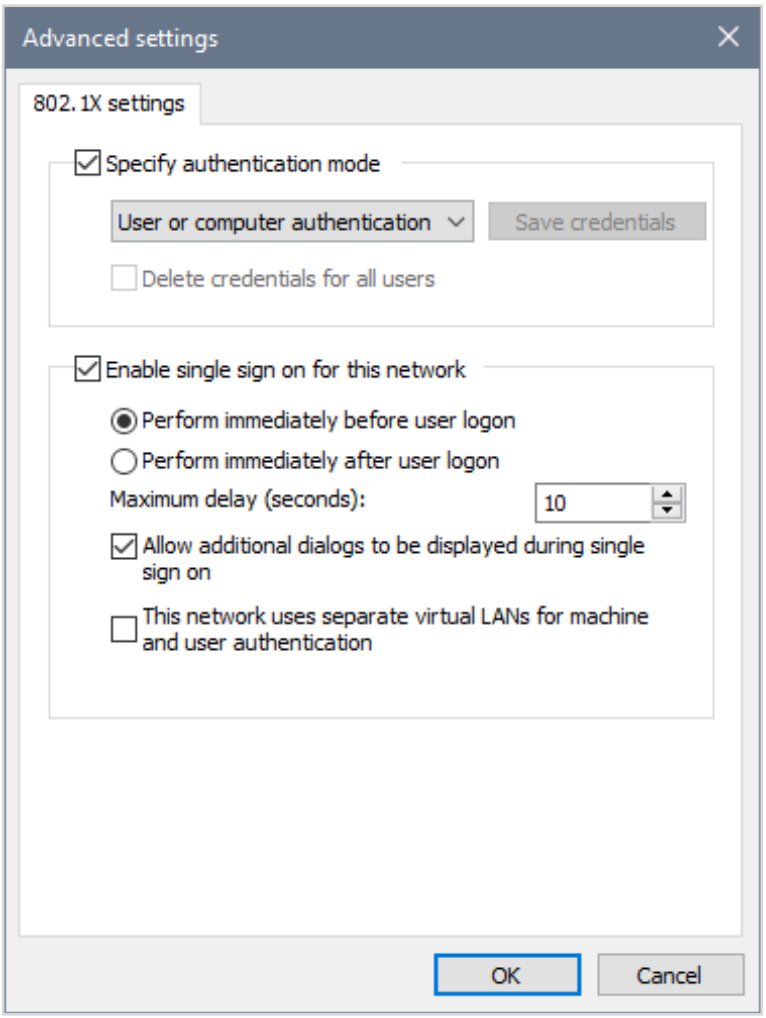
# Modify network adapter attributes.



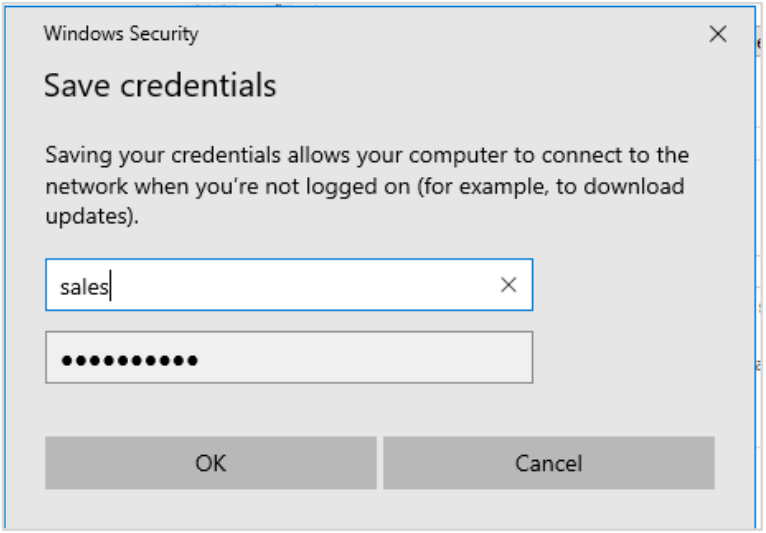

Take PC1 as an example. Choose **Control Panel** > **Network and Internet** > **Network and Sharing Center** > **Change adapter settings** in the operating system. Find the network adapter that connects the terminal to the access switch. Right-click the network adapter and select **Properties**.

# Configure authenticate identity.


Ethernet Properties
Networking
Authentication
Sharing
Select this option to provide authenticated network access for this Ethernet adapter.
Enable IEEE 802.1X authentication
Choose a network authentication method:
Microsoft: Protected EAP (PEAP)
Settings
Remember my credentials for this connection each time I'm logged on
Fallback to unauthorized network access
Additional Settings...
OK
Cancel




Click the **Authentication** tab page, and click **Additional Settings**. In the dialog box that is displayed, select **User authentication** and click **Replace credentials**.




In the dialog box that is displayed, enter the user name and password created on iMaster NCE. The following uses sales as an example.

# Log in to kris on PC1.

```
PS C:\Users\PC1\Desktop> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . . . . . :
   Link-local IPv6 Address . . . . . :  fe80::a46b:62f8:a696:b974%12
   IPv4 Address . . . . . :  172.17.30.225
   Subnet Mask . . . . . :  255.255.255.0
   Default Gateway . . . . . :  172.17.30.254
```

The IP address obtained by PC1 belongs to the 172.17.30.0/24 network segment, which meets the expectation.

# Log in to sales on PC2.

```
C:\Users\PC2>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet1:

   Connection-specific DNS Suffix . . . . . :
   Link-local IPv6 Address . . . . . :  fe80::407a:8e15:4cf:679%5
   IPv4 Address . . . . . :  172.17.10.138
   Subnet Mask  . . . . . :  255.255.255.0
   Default Gateway . . . . . :  172.17.10.254
```

The IP address obtained by PC2 belongs to the 172.17.10.0/24 network segment, which meets the expectation.

# Log in to market on PC2.

```
C:\Users\PC2>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet1:

   Connection-specific DNS Suffix . . . . . :
   Link-local IPv6 Address . . . . . :  fe80::407a:8e15:4cf:679%5
   IPv4 Address . . . . . :  172.17.20.167
   Subnet Mask  . . . . . :  255.255.255.0
   Default Gateway . . . . . :  172.17.20.254
```

The IP address obtained by PC2 belongs to the 172.17.20.0/24 network segment, which meets the expectation.

# View online users on Edge_1.

```
<Edge_1>display access-user
 ------------------------------------------------------------------------------------------------
 UserID   Username                  IP address                      MAC             Status
```

```
--------------------------------------------------------------------------------
49170    kris                 172.17.30.225                000c-29b3-efea Success
--------------------------------------------------------------------------------
Total: 1, printed: 1
```

Edge_1 is the authentication control point.

# View detailed information about the user **kris**.

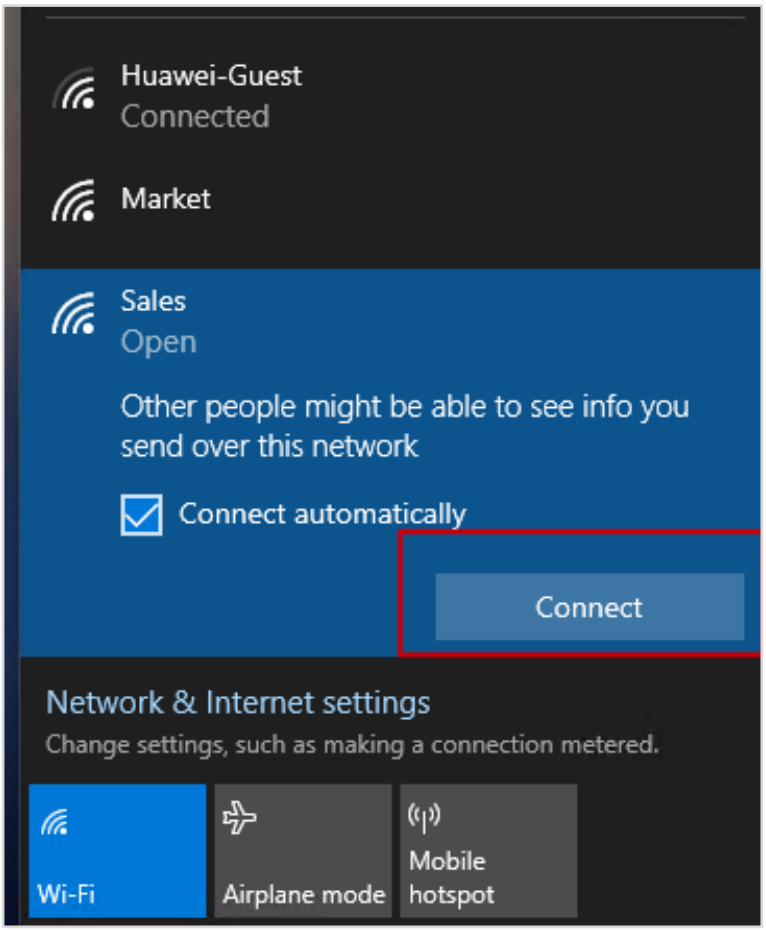
```
<Edge_1>display access-user username kris detail

Basic:
  User ID                          : 49170
  User name                        : kris
  Domain-name                      : default
  User MAC                         : 000c-29b3-efea
  User IP address                  : 172.17.30.225
  User vpn-instance                : RD
  User IPv6 address                : FE80::A46B:62F8:A696:B974
  User IPv6 link local address     : FE80::A46B:62F8:A696:B974
  User access Interface            : GigabitEthernet0/0/24
  User vlan event                  : Success
  QinQVlan/UserVlan                : 0/300
  User vlan source                 : server vlan
  User access time                 : ****/**/** **:**:**
  User accounting session ID       : Edge_10002400000030019****0300012
  User access type                 : 802.1x
  AS ID                            : 0
  AS name                          : ACC_1
  AS IP                            : 172.16.17.133
  AS MAC                           : 9400-b049-9d80
  AS Interface                     : GigabitEthernet0/0/24
  Terminal Device Type             : Data Terminal
  Dynamic VLAN ID                  : 300
  Dynamic group index(Effective): 5
  Dynamic group name(Effective) : RD
  Service Scheme Priority          : 0

AAA:
  User authentication type         : 802.1x authentication
  Current authentication method    : RADIUS
  Current authorization method     : -
  Current accounting method        : RADIUS

 --------------------------------------------------------------------------------
 Total: 1, printed: 1
```

In the authorization information, check the authorized VLAN ID and security groups.

Step 2 Verify Portal authentication.

Check whether wireless terminal PC3 can pass Portal authentication and obtain related authorization information.

# Connect to the SSID Sales.



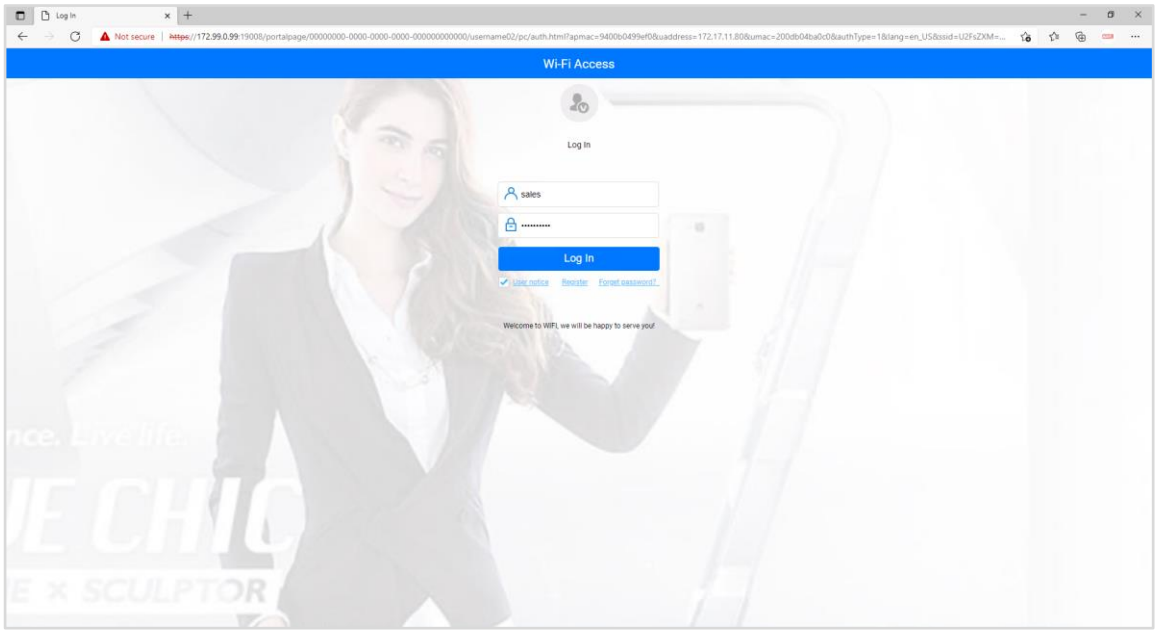

# Expand the Wi-Fi list, find the previously defined SSID **Sales**, and connect to the SSID.

# Check the IP address.

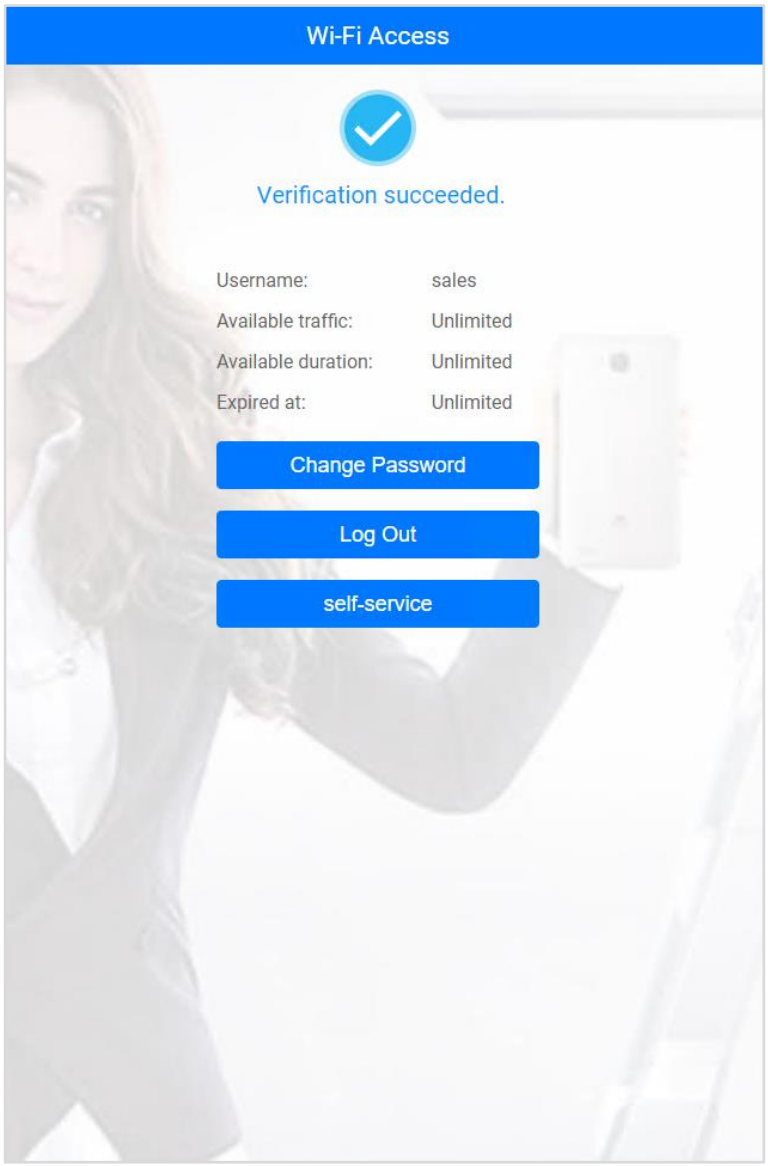
```
C:\Users\PC3>ipconfig

Windows IP Configuration

Wireless LAN adapter WLAN:

   Connecting specific DNS Suffix . . . . . :
   Local link IPv6 Address . . . . . :   fe80::407a:8e15:4cf:679%5
   IPv4 Address . . . . . :            172.17.11.70
   Subnet Mask . . . . . :             255.255.255.0
   Default Gateway . . . . . :             172.17.11.254
```

The obtained IP address belongs to network segment 172.17.11.0/24, which meets the expectation.

# Verify Portal authentication.

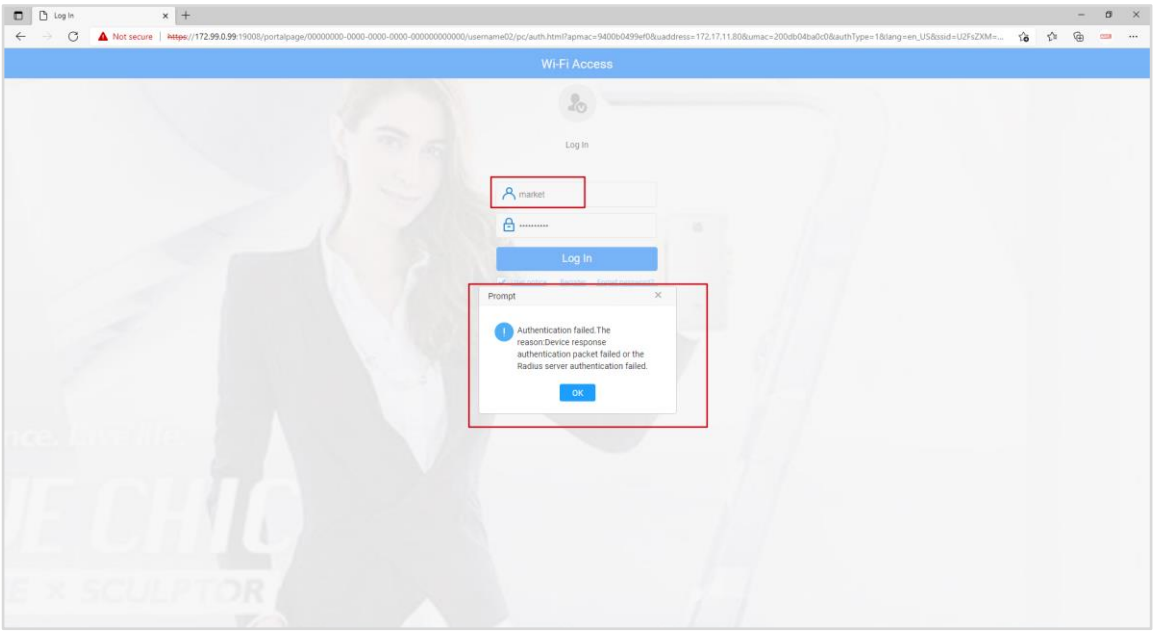



Users can open a browser and access any IP address (No DNS is deployed in this lab, so only an IP address can be entered). The authentication point (switch Border) redirects the user access to the Portal authentication page, because the user has not been authenticated.




Enter the user name **sales** and the password. The login is successful.

# Use the account **market** for authentication.




After connecting to the SSID Sales, use the account **market** for authentication. The authentication will fail, because the authorization rules on the controller cannot be matched. By matching SSID with user group information through authorization rules, the SSID can only be used for authentication for the accounts in the user group Sales.

# Check the authenticated user on Border.

```
<Border>display access-user
------------------------------------------------------------------------------------------------
UserID   Username                IP address                MAC                Status
------------------------------------------------------------------------------------------------
32790    sales                   172.17.11.70              200d-b04b-a0c0     Success
------------------------------------------------------------------------------------------------
Total: 1, printed: 1
```

Use account sales for login again on PC3, and then check the access user on Border.

# Check detailed information of the user sales on Border.

```
<Edge_1>display access-user username sales detail
Basic:
  User ID                          : 32790
  User name                        : sales
  Domain-name                      : aaa46d7bd04792c27bbdc4c
  User MAC                         : 200d-b04b-a0c0
  User IP address                  : 172.17.11.70
  User vpn-instance                : OA
  User IPv6 address                : -
  User access Interface            : Wlan-Dbss2282
  User vlan event                  : Success
  QinQVlan/UserVlan                : 0/110
  User vlan source                 : user request
  User access time                 :  ****/**/** **:**:**
  User accounting session ID       : Border00024000000110db****0200016
  User accounting mult session ID : F4DEAF36B580200DB04BA0C0613AC****C5CA321
```

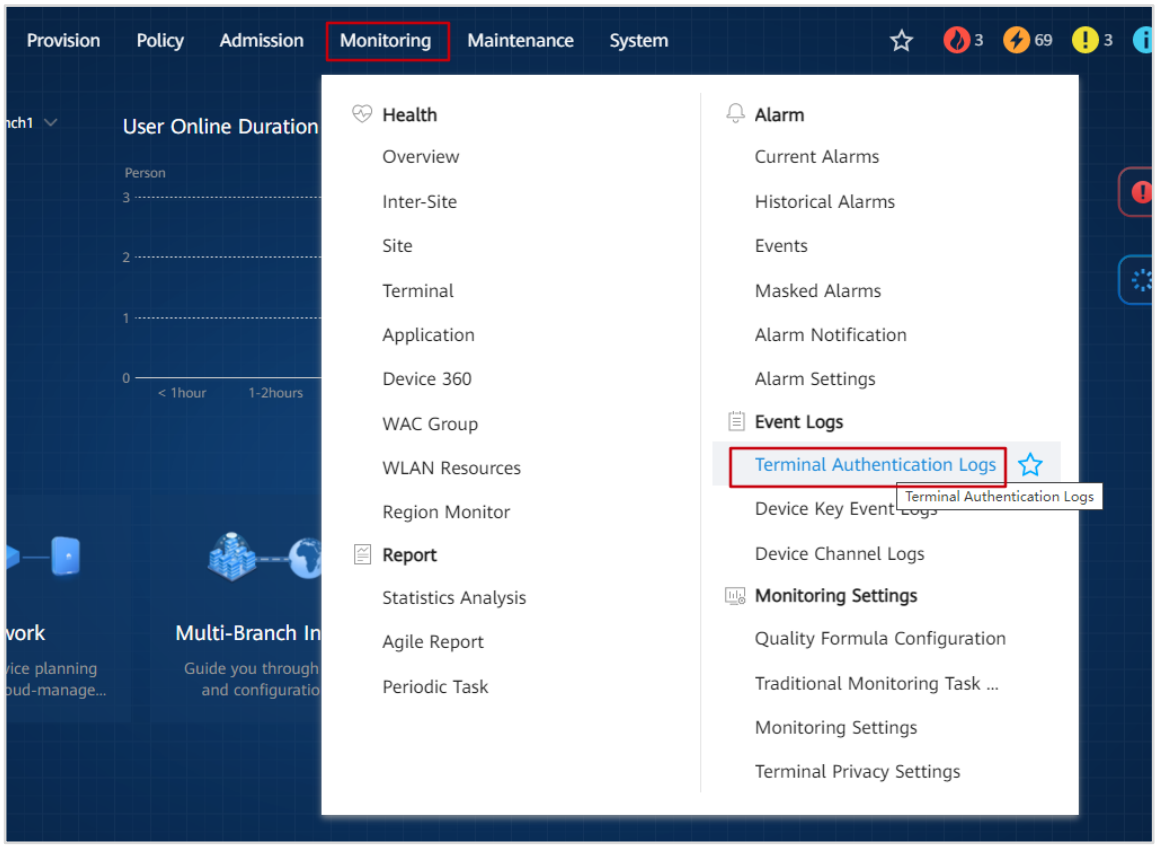
```
  User access type                    : WEB
  AP name                              : AP1
  Radio ID                             : 1
  AP MAC                               : f4de-af36-b580
  SSID                                 : Sales
  Online time                          : 132(s)
  Web-server IP address                : 172.99.0.99
  Dynamic group index(Effective): 2
  Dynamic group name(Effective): Sales_Wireless
  User inbound data flow(Packet): 365
  User inbound data flow(Byte)     : 33,637
  User outbound data flow(Packet): 40
  User outbound data flow(Byte)     : 33,768
  Service Scheme Priority              : 0

AAA:
  User authentication type            : WEB authentication
  Current authentication method    : RADIUS
  Current authorization method     : -
  Current accounting method        : RADIUS

 --------------------------------------------------------------------------------
 Total: 1, printed: 1
```

A security group is authorized.

Step 3 Check the authentication log.

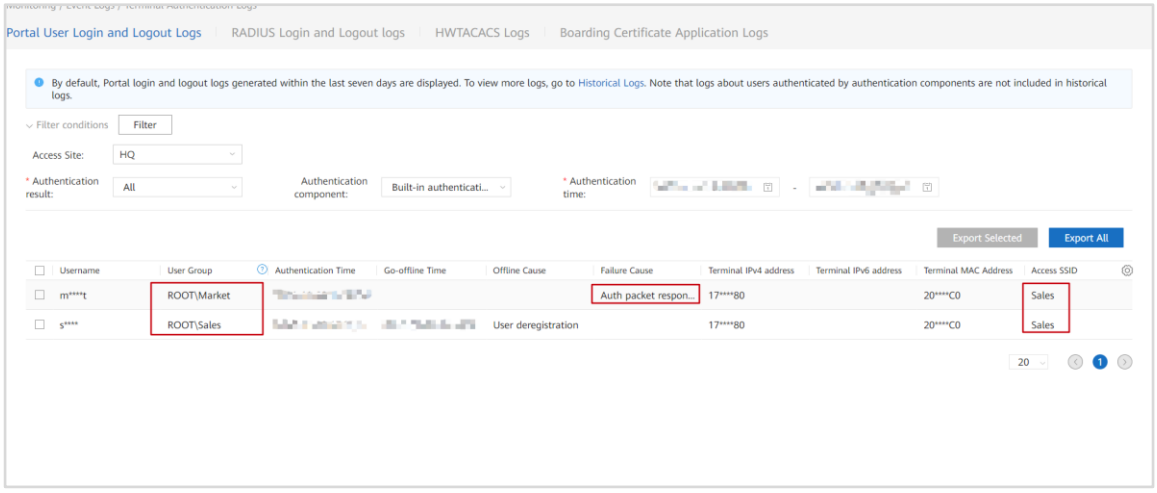
Check Portal logs and RADIUS logs on iMaster NCE.

# Switch between pages.



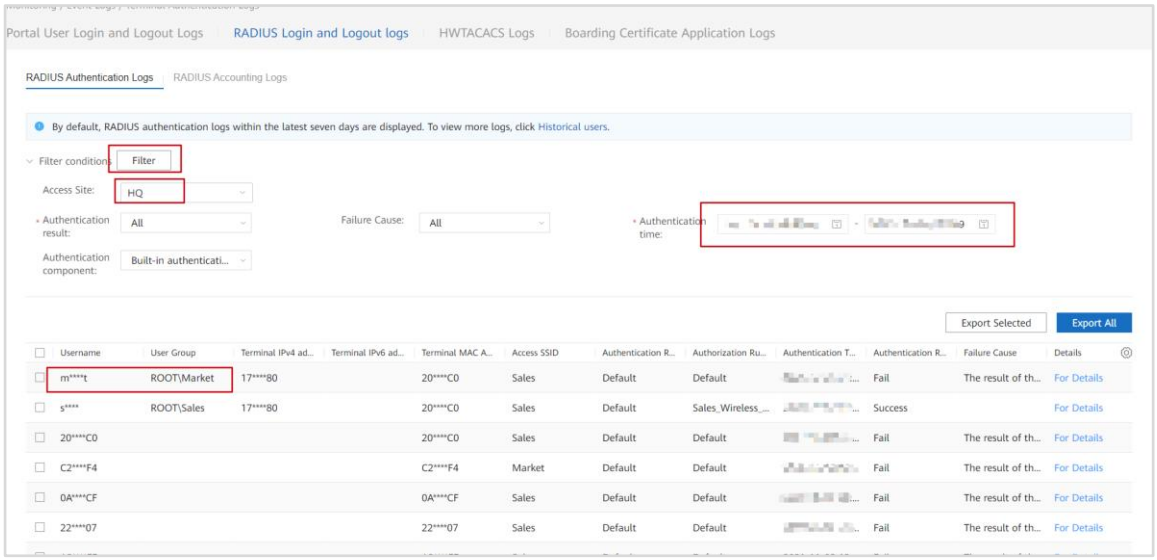

Choose **Monitoring** > **Terminal Authentication Logs** on iMaster NCE. The **Terminal Authentication Logs** page is displayed.

# Check Portal logs.




All Portal login and logout logs are displayed.

# Check RADIUS logs.




Filter RADIUS authentication logs by site and time. In the filtering result, you can check the failure logs of PC3 using account market for login when connected to the SSID Sales, as well as the failure reason.

### 6.1.2.9.2 Network Connectivity Verification

Verify whether free mobility takes effect through tests in the following scenarios.

1. Check whether PC1 can pass 802.1X authentication using the account kris. Check whether PC2 can pass 802.1X authentication using the account sales. Connect PC3 to the SSID Sales and use the account sales for authentication.
2. Check whether PC1 can pass 802.1X authentication using the account kris. Check whether PC2 can pass 802.1X authentication using the account market. Connect PC3 to the SSID Market and use the account market for authentication.

Step 1 Perform verification in scenario 1.

# Test whether PCs can access a resource group.

```
C:\Users\PC1>ping 172.17.3.3

Pinging 172.17.3.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 172.17.3.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

```
C:\Users\PC2>ping 172.17.3.3

Pinging 172.17.3.3 with 32 bytes of data:
Reply from 172.17.3.3: bytes=32 time<1ms TTL=253
Reply from 172.17.3.3: bytes=32 time<1ms TTL=253
```

```
Reply from 172.17.3.3: bytes=32 time<1ms TTL=253
Reply from 172.17.3.3: bytes=32 time<1ms TTL=253

Ping statistics for 172.17.3.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss);
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\Users\PC3>ping 172.17.3.3

Pinging 172.17.3.3 with 32 bytes of data:
Reply from 172.17.3.3: bytes=32 time=4ms TTL=254
Reply from 172.17.3.3: bytes=32 time=4ms TTL=254
Reply from 172.17.3.3: bytes=32 time=3ms TTL=254
Reply from 172.17.3.3: bytes=32 time=3ms TTL=254

Ping statistics for 172.17.3.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss);
Approximate round trip times in milli-seconds:
...Minimum = 3ms, Maximum = 4ms, Average = 3ms
```

Only PC1 cannot access the resource group E_mail in the test.

# Test the communication between wired and wireless sales users.

```
C:\Users\PC3>ipconfig

Windows IP Configuration

Wireless LAN adapter WLAN:

   Connecting specific DNS Suffix . . . . . :
   Local link IPv6 Address . . . . . : fe80::4c25:e93f:ee11:5ad1%16
   IPv4 Address . . . . . : 172.17.11.70
   Subnet Mask . . . . . : 255.255.255.0
   Default Gateway . . . . . : 172.17.11.254

C:\Users\PC3>ping 172.17.10.138 -n 1

Pinging 172.17.10.138 with 32 bytes of data:
Reply from 172.17.10.138: bytes=32 time=5ms TTL=126

Ping statistics for 172.17.10.138:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss);
Approximate round trip times in milli-seconds:
...Minimum = 5ms, Maximum = 5ms, Average = 5ms
```

```
C:\Users\PC2>ipconfig

Windows IP Configuration
```

```
Network Adapter Ethernet1:

    Connecting specific DNS Suffix . . . . . :
    Local link IPv6 Address . . . . . : fe80::407a:8e15:4cf:679%5
    IPv4 Address . . . . . : 172.17.10.138
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 172.17.10.254

C:\Users\PC2>ping 172.17.11.70 -n 1

Pinging 172.17.11.70 with 32 bytes of data:
Reply from 172.17.11.70: bytes=32 time=4ms TTL=126

Ping statistics for 172.17.11.70:
     Packets: Sent = 1, Received = 1, Lost = 0 (0% loss);
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 4ms, Average = 4ms
```

Login accounts on PC2 and PC3 both belong to the security group Sales and the VN OA. The policy matrix has no limitation for mutual access.

# Test whether wired and wireless sales users can access VN RD.

```
C:\Users\PC2>ping 172.17.30.225 -n 1

Pinging 172.17.30.225 with 32 bytes of data:
Request timed out.

Ping statistics for 172.17.30.225:
Packets: Sent = 1, Received = 0, Lost = 1 (100% loss)
```

```
C:\Users\PC3>ping 172.17.30.225 -n 1

Pinging 172.17.30.225 with 32 bytes of data:
Request timed out.

Ping statistics for 172.17.30.225:
Packets: Sent = 1, Received = 0, Lost = 1 (100% loss)
```

PC1 in the VN RD cannot be accessed.

Step 2 Perform verification in scenario 2.

# Test whether PC2 and PC3 can access resource groups.

```
C:\Users\PC2>ping 172.17.3.3 -n 1

Pinging 172.17.3.3 with 32 bytes of data:
Reply from 172.17.3.3: bytes=32 time<1ms TTL=253

Ping statistics for 172.17.3.3:
     Packets: Sent = 1, Received = 1, Lost = 0 (0% loss);
Approximate round trip times in milli-seconds:
```

```
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\Users\PC3>ping 172.17.3.3 -n 1

Pinging 172.17.3.3 with 32 bytes of data:
Reply from 172.17.3.3: bytes=32 time=4ms TTL=254

Ping statistics for 172.17.3.3:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss);
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 4ms, Average = 4ms
```

All PCs can access resource groups.

# Test whether PC2 and PC3 can access the VN RD.

```
C:\Users\PC2>ping 172.17.30.225 -n 1

Pinging 172.17.30.225 with 32 bytes of data:
Reply from 172.17.30.225: bytes=32 time<1ms TTL=125

Ping statistics for 172.17.30.225:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss);
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\Users\PC3>ping 172.17.30.225 -n 1

Pinging 172.17.30.225 with 32 bytes of data:
Request timed out.

Ping statistics for 172.17.30.225:
Packets: Sent = 1, Received = 0, Lost = 1 (100% loss)
```

As expected, PC2 can access the VN RD but PC3 cannot, which means the Market wired network can be accessed but the Market wireless network cannot.

## 6.1.3 (Optional) Clearing Configurations

When the lab begins, clear the configurations of iMaster NCE and network devices (Border, Edge_1, Edge_2, ACC_1, and ACC_2) made by the previous user and restore the nodes to factory settings.

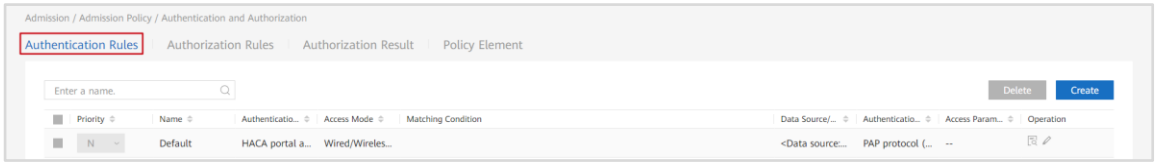
Node: Delete the configurations on iMaster NCE first, then on network devices.

### 6.1.3.1 Deleting Authentication Rules, Authorization Rules, and Authorization Results

Step 1 Choose Admission > Admission Policy > Authentication And Authorization on iMaster NCE to access the Authentication and Authorization page.

Step 2 Delete authentication rules.

As shown in the following figure, click **Authentication Rules** to access the configuration page of authentication rules.




Except the default authentication rule **Default**, if other customized authentication rules exist, select all the created rules and click **Delete**.

Step 3 Delete authorization rules.

Click **Authorization Rules** on the current page to access the configuration page of authorization rules. Except the default authorization rule **Default**, if other customized authorization rules exist, select all the created rules and click **Delete**.

Step 4 Delete authorization results.

Click **Authorization Result** on the current page to access the configuration page of authorization results. Except the default authorization results (**Permit Access** and **Deny Access**), if other customized authorization results exist, select all the created results and click **Delete**.

In the authorization results to be deleted, click to bind a site, select **HQ** and click **delete** to unbind the authorization results from the HQ site. The authorization result can be deleted only after this operation is performed. Perform the same operation to unbind all other items except default authorization results. After unbinding, select all the created authorization results and click **Delete**.

## 6.1.3.2 Disconnecting Users and Deleting User Accounts

Note: To ensure that users' accounts can be deleted, disconnect the users first.

Step 1 Choose **Admission** > **Admission Policy** > **Online User Control** on iMaster NCE to access the online user view. Select all users and choose **More** > **Force Offline**.

Step 2 Choose **Admission** > **Admission Resources** > **User Management** on iMaster NCE to access the user management page.

Step 3 Select all created user accounts and click **Delete**.

## 6.1.3.3 Deleting the Policy Matrix and Security Groups

Step 1 Choose **Admission** > **Policy Control** on iMaster NCE to access the policy control configuration page.

Step 2 Click icon to delete the created policy matrix.

Step 3 Choose **Admission** > **Security Group** on iMaster NCE to access the security group configuration page. Select all created security groups and click **Delete**.

### 6.1.3.4 Deleting VN Communication Policies and VNs

Step 1 Choose **Provision** > **Logical Network** to access the VN configuration page. Click **VN Interwork** to access the VN interworking policy configuration page. Then delete the created VN interworking policies.

Step 2 Click to delete VNs one by one in the **Operation** column of every created VN.

### 6.1.3.5 Deleting Access Management Configurations

Step 1 Choose **Provision** > **Fabric Management** to access the fabric management page. Click **Access management** to access the management configuration page.

Step 2 By default, choose **Edge_1** from **Authentication control point**, scroll to the bottom of the page, and click the **Reset** button to reset the access management configuration of Edge_1.

Step 3 Choose **Edge_2** from **Authentication control point**, scroll to the bottom of the page, and click the **Reset** button to reset the access management configuration of Edge_2.

### 6.1.3.6 Deleting Network Service Resources and External Networks

Step 1 Choose **Provision** > **Fabric Management** to access the fabric management page.

Step 2 Click in the column of **External Network**. On the configuration page of external networks, click next to the created external networks to delete all the external networks.

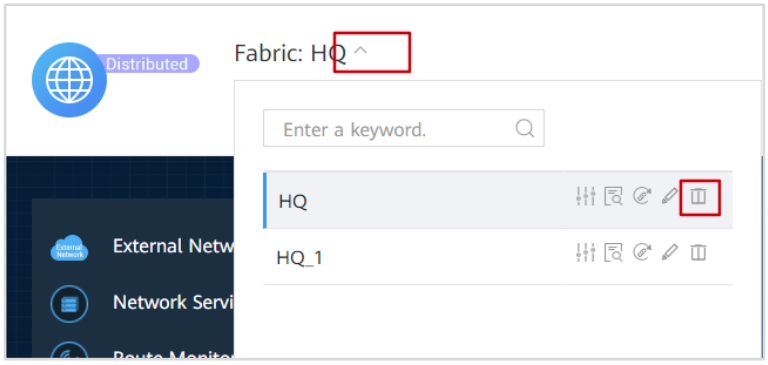
Step 3 Click in the column of **Network Service Resources** on the fabric management page. On the configuration page of network service resources, click next to the created network service resources to delete them.

### 6.1.3.7 Deleting Devices on the Fabric Page and the Fabric Network

Step 1 Choose **Provision** > **Fabric Management** to access the fabric management page.

Step 2 On the current page, click on the right side and select all the nodes. Then click **Delete Device**.

Step 3 Delete the current fabric network, as shown in the following figure.



### 6.1.3.8 Deleting the Fabric Global Resource Pool and Underlay Automation Resource Pool

Step 1 Choose **Design** > **Network Settings** to access the network settings page. Select all the created VLANs and click to delete them. Select all the created BDs and click to delete them. Select all the created VNIs and click to delete them.

Step 2 Click **Underlay Automation Resource Pool** on the current page to access the configuration page of underlay automation resource pool. Select all the created interconnection VLANs and click to delete them. Select all the created interconnection IP addresses and click to delete them. Select all the created loopback interface IP addresses and click to delete them.

### 6.1.3.9 Deleting Devices Managed by iMaster NCE

Choose **Design** > **Device Management** to access the device management page. Select all devices under the site HQ and click **Delete Device**.

### 6.1.3.10 Deleting Sites

Choose **Design** > **Site Management** to access the site management page. Select the site and click to delete sites.

### 6.1.3.11 Deleting Authentication Templates and Server Templates

Step 1 Delete authentication templates.

Choose **Design** > **Template Management** to access the template management page. Click **Authentication Template** from the navigation tree on the left, select all the created templates, and click **Delete**.

Step 2 Delete RADIUS server templates.

Click **RADIUS Server** on the left, select all the created RADIUS server templates, and click **Delete**.

Step 3 Delete Portal server templates.

Click **Portal Server** on the left, select all the created Portal server templates, and click **Delete**.

### 6.1.3.12 Deleting Configurations on Switches

Log in to Border, Edge_1, Edge_2, ACC_1, and ACC_2 respectively. Run the following commands to restore the devices to factory settings.

```
<Border>system-view
Enter system view, return user view with Ctrl+Z.
[Border]reset netconf db-configuration
Warning: This operation will clear the database configuration and saved configuration file and restart the device. Continue? [Y/N]:y
```

### 6.1.3.13 Deleting Configurations on AR3

To clear current configurations of AR3 and restore AR3 to the default status, run the following commands after logging in to the device.

```
<AR3>set factory-configuration from default
 Warning: The current factory configuration will be replaced, and it's irreversible. Are you sure to set the factory configuration?[Y/N]:y
 Info: Successfully set factory config!
<AR3>factory-configuration reset
Warning: It will clean the configuration which you have saved. If you have set the factory-configuration by hand, it will start from the modified factory-configuration, else it will start from the original one, when you restart the device. Continue? [y/n]:y
 Info: Successfully set factory config!
<AR3>reset saved-configuration
 Warning: This will delete the configuration in the flash memory.
 The device configurations will be erased to reconfigure.
 Are you sure? (y/n):y
Info: Clear the configuration in the device successfully.
<AR3>reboot fast
System will reboot! Continue? [y/n]:y
```

**----End**

## 6.1.4 Quiz

After a VN invokes network service resources configured on iMaster NCE, how do the underlying configurations implement route reachability between the VN and network service resources?

# 7 WAN Interconnection Network Deployment

## 7.1 WAN Interconnection Network Deployment

### 7.1.1 About This Lab

#### 7.1.1.1 Objectives

Upon completion of this task, you will be able to:

- Configure network templates.
- Design and configure sites.
- Implement DHCP-based deployment.
- Configure traffic policies.
- Configure security policies.

#### 7.1.1.2 Networking Description

**Figure 7-1 Topology of WAN interconnection network deployment**

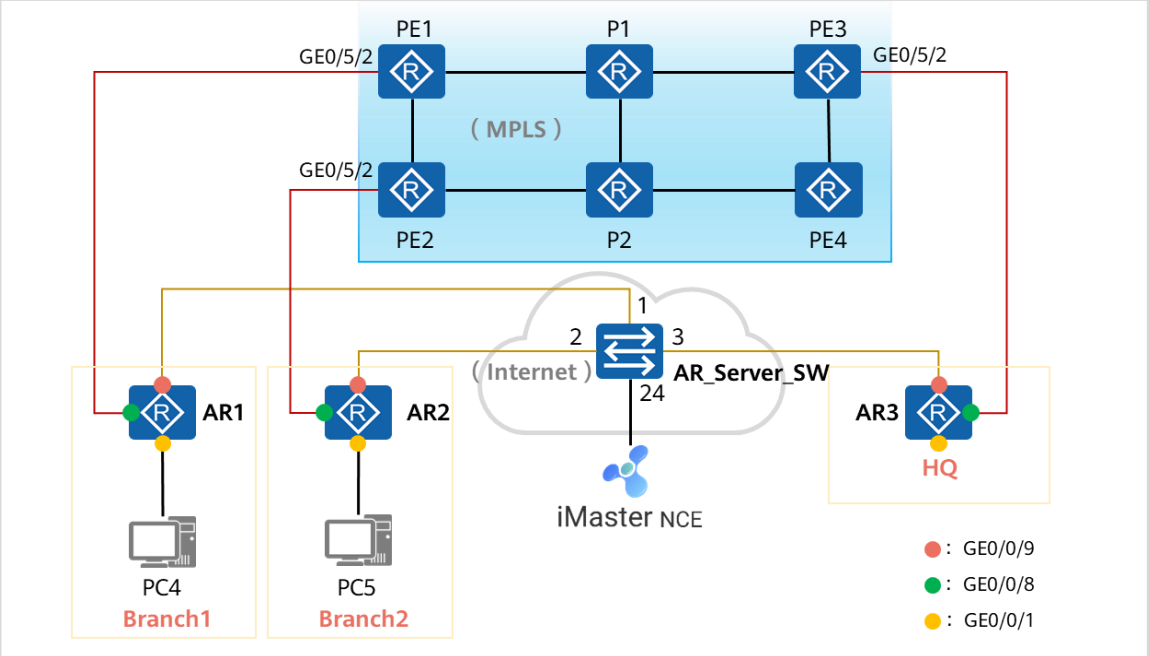
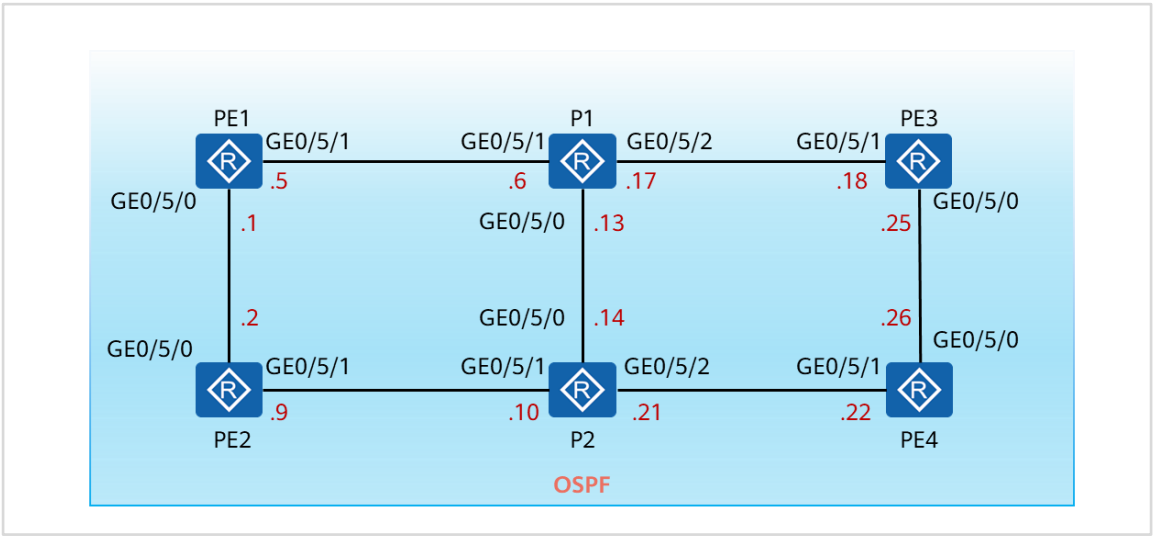


**Figure 7-2 Internal topology of the MPLS network**




As shown in the preceding figures, the topology consists of three parts:

1. HQ and branch networks
2. Internet
3. MPLS network

HQ and branch networks: The HQ network in this lab refers to the HQ network in 6 "VXLAN-based Virtualized Campus Network Deployment" and has been configured. The ARs on the two branch networks and HQ network will be onboarded on the controller in this lab and complete the SD-WAN configuration.

Internet: simulated by AR_Server_SW. All ARs connect to the Internet through GE0/0/9.

MPLS network: simulated by a network consisting of NEs. All devices run OSPF to implement network connectivity. Note that the MPLS network is a simulated network without the need to run MPLS. All ARs connect to the MPLS network through GE0/0/8.

In this lab, the ARs obtain the IP address of the controller through DHCP and then onboard.

## 7.1.1.3 Lab Plan

### 7.1.1.3.1 Virtual Network Parameter Plan

The following table describes the parameter plan for SD-WAN overlay networks, including the AS number of BGP EVPN and the global address pool of the overlay network.

**Table 7-1 Parameter plan for virtual networks**

| Parameter | Value |
|---|---|
| AS number | 65001 |
| IP pool | 100.0.0.0/24 |

Retain the default values of other parameters.

#### 7.1.1.3.2 Device Onboarding Plan

On iMaster NCE, you can add routers in this lab by device model or ESN to implement device management.

**Table 7-2 Device onboarding plan**

| Device | ESN | Model | Site | Device Type | Role |
|---|---|---|---|---|---|
| AR1 | 1002352RLG1980065037 | AR6120 | Branch1 | AR | Gateway |
| AR2 | 1002352RLG1980065092 | AR6120 | Branch2 | AR | Gateway |
| AR3 | 2102115641DMK8000908 | AR6120 | HQ | AR | Gateway+RR |

Note that AR3 is the egress router of the HQ network and functions as an RR in the SD-WAN scenario.

#### 7.1.1.3.3 ZTP

After the ARs obtain the controller's IP address through DHCP and register with the controller, configure ZTP on the controller by referring to the detailed parameter plan in the following table.

**Table 7-3 ZTP parameter plan for Branch1**

| Parameter | Value |
|---|---|
| ZTP mode | DHCP Option |
| Link name: Internet | |
| Transport network | Internet |
| Role | Active |
| Device | AR1 |
| Interface | GE0/0/9 |
| VN instance | underlay_1 |
| Interface protocol | IPoE |
| IP address access mode | DHCP |
| Southbound interface service | Public Default South Access |
| Uplink bandwidth (Mbit/s) | 1000 |
| Downlink bandwidth (Mbit/s) | 1000 |
| Link name: MPLS | |
| Transport network | MPLS |

| Parameter | Value |
| --- | --- |
| Role | Active |
| Device | AR1 |
| Interface | GE0/0/8 |
| VN instance | underlay_2 |
| Interface protocol | IPoE |
| IP address access mode | Static |
| IPv4 address | 10.0.0.29 |
| Subnet mask | 30 |
| IPv4 gateway | 10.0.0.30 |
| Southbound interface service | Public Default South Access |
| Uplink bandwidth (Mbit/s) | 1000 |
| Downlink bandwidth (Mbit/s) | 1000 |

**Table 7-4 ZTP parameter plan for Branch2**

| Parameter | Value |
| --- | --- |
| ZTP mode | DHCP Option |
| Link name: Internet | |
| Transport network | Internet |
| Role | Active |
| Device | AR2 |
| Interface | GE0/0/9 |
| VN instance | underlay_1 |
| Interface protocol | IPoE |
| IP address access mode | DHCP |
| Southbound interface service | Public Default South Access |
| Uplink bandwidth (Mbit/s) | 1000 |
| Downlink bandwidth (Mbit/s) | 1000 |
| Link name: MPLS | |

| Parameter | Value |
| --- | --- |
| Transport network | MPLS |
| Role | Active |
| Device | AR1 |
| Interface | GE0/0/8 |
| VN instance | underlay_2 |
| Interface protocol | IPoE |
| IP address access mode | Static |
| IPv4 address | 10.0.0.33 |
| Subnet mask | 30 |
| IPv4 gateway | 10.0.0.34 |
| Southbound interface service | Public Default South Access |
| Uplink bandwidth (Mbit/s) | 1000 |
| Downlink bandwidth (Mbit/s) | 1000 |

**Table 7-5 ZTP parameter plan for HQ**

| Parameter | Value |
| --- | --- |
| ZTP mode | DHCP Option |
| Link name: Internet | |
| Transport network | Internet |
| Role | Active |
| Device | AR3 |
| Interface | GE0/0/9 |
| VN instance | underlay_1 |
| Interface protocol | IPoE |
| IP address access mode | static |
| IPv4 address | 65.0.0.3 |
| Subnet mask | 24 |
| IPv4 gateway | 65.0.0.254 |

| Parameter | Value |
| --- | --- |
| Southbound interface service | Public Default South Access |
| Uplink bandwidth (Mbit/s) | 1000 |
| Downlink bandwidth (Mbit/s) | 1000 |
| Link name: MPLS | |
| Transport network | MPLS |
| Role | Active |
| Device | AR3 |
| Interface | GE0/0/8 |
| VN instance | underlay_2 |
| Interface protocol | IPoE |
| IP address access mode | Static |
| IPv4 address | 10.0.0.37 |
| Subnet mask | 30 |
| IPv4 gateway | 10.0.0.38 |
| Southbound interface service | Public Default South Access |
| Uplink bandwidth (Mbit/s) | 1000 |
| Downlink bandwidth (Mbit/s) | 1000 |

Note that the AR at the HQ site functions as an RR and its interface IP addresses must be static IP addresses.

#### 7.1.1.3.4 LAN-WAN Interconnection

The following tables describe the LAN-WAN interconnection parameter plan for the HQ site:

**Table 7-6 LAN-WAN interconnection parameter plan for the virtual network public**

| Parameter | Value |
| --- | --- |
| VN: public | |
| Interconnection interface 1 | |
| Gateway | AR3 |

| Parameter | Value |
| --- | --- |
| Gateway interface | L2 |
| VLAN ID | 10 |
| Physical interfaces | GE0/0/1 (untagged) |
| IP address | 172.16.10.254/24 |
| Advanced Settings | Configured |
| DHCP | Enabled |
| DHCP Option | Option 148 is configured. |
| Static | Configured |
| Interconnection interface 2 | |
| Gateway | AR3 |
| Gateway interface | L2 |
| VLAN ID | 150 |
| Physical interfaces | GE0/0/1 (tagged) |
| IP address | 192.168.150.1/24 |
| Interconnection route configuration | |
| Entry 1 | 172.17.10.0/24 192.168.150.2 |
| Entry 2 | 172.17.11.0/24 192.168.150.2 |
| Entry 3 | 172.17.20.0/24 192.168.150.2 |
| Entry 4 | 172.17.21.0/24 192.168.150.2 |
| Entry 5 | 172.17.30.0/24 192.168.150.2 |

**Table 7-7 LAN-WAN interconnection parameter plan for the virtual network OA**

| Parameter | Value |
| --- | --- |
| VN: OA | |
| Interconnection interface 1 | |
| Gateway | AR3 |
| Gateway interface | L2 |

| Parameter | Value |
| --- | --- |
| VLAN ID | 130 |
| Physical interfaces | GE0/0/1 (tagged) |
| IP address | 13.1.1.1/30 |
| Interconnection route configuration | |
| Entry 1 | 172.17.10.0/24 13.1.1.2 |
| Entry 2 | 172.17.11.0/24 13.1.1.2 |
| Entry 3 | 172.17.20.0/24 13.1.1.2 |
| Entry 4 | 172.17.21.0/24 13.1.1.2 |

**Table 7-8 LAN-WAN interconnection parameter plan for the virtual network RD**

| Parameter | Value |
| --- | --- |
| VN: RD | |
| Interconnection interface 1 | |
| Gateway | AR3 |
| Gateway interface | L2 |
| VLAN ID | 140 |
| Physical interfaces | GE0/0/1 (tagged) |
| IP address | 14.1.1.1/30 |
| Interconnection route configuration | |
| Entry 1 | 172.17.30.0/24 14.1.1.2 |

The following table describes the LAN-WAN interconnection parameter plan for Branch2.

**Table 7-9 LAN-WAN interconnection parameter plan for the virtual network OA**

| Parameter | Value |
| --- | --- |
| VN: OA | |
| Interconnection interface 1 | |

| Parameter | Value |
| --- | --- |
| Gateway | AR2 |
| Gateway interface | L2 |
| VLAN ID | 200 |
| Physical interfaces | GE0/0/1 (untagged) |
| IP address | 172.19.20.254/24 |
| Advanced Settings | Configured |
| DHCP | Enabled |

The following table describes the LAN-WAN interconnection parameter plan for Branch1.

**Table 7-10 LAN-WAN interconnection parameter plan for the virtual network RD**

| Parameter | Value |
| --- | --- |
| VN: RD | |
| Interconnection interface 1 | |
| Gateway | AR1 |
| Gateway interface | L2 |
| VLAN ID | 300 |
| Physical interfaces | GE0/0/1 (untagged) |
| IP address | 172.18.30.254/24 |
| Advanced Settings | Configured |
| DHCP | Enabled |

# 7.1.2 Lab Task

## 7.1.2.1 Configuration Roadmap

1. Pre-configure AR_Server_SW that simulates the Internet, including assigning VLANs for the interfaces connected to the ARs, configuring the DHCP address pool, and configuring DHCP options for AR onboarding.
2. Pre-configure the simulated MPLS network, including setting the IP addresses of interconnection interfaces, configuring OSPF, and setting the IP addresses of the interfaces connected to the ARs.

3. Create sites HQ, Branch1, and Branch2 and add AR3, AR1, and AR2 to the three sites respectively for device onboarding.
4. Configure SD-WAN deployment, including configuring ZTP for each site, Internet and MPLS links for each site, underlay routes, and the inter-site networking model.
5. Create a virtual network **public** and complete the following configurations on the virtual network **public** at the HQ site: Create VLANIF 10 for LAN-WAN interconnection (this interface assigns IP addresses to devices in the fabric), and configure return routes destined for user network segments to ensure network reachability for the devices in the fabric to onboard.
6. Create a virtual network **OA** that provides external network services for the logical network **OA** on the fabric of the HQ site.
7. Create a virtual network **RD** that provides external network services for the logical network **RD** on the fabric of the HQ site.
8. Configure Internet access for the sites, configure NAT for the underlay network, create a DHCP address pool for the fabric, and create a loopback interface that simulates the E-mail server.
9. Configure application identification and intelligent traffic steering. Configure intelligent traffic steering policies to preferentially direct FTP traffic of virtual network **RD** to Internet links and HTTP traffic of virtual network **OA** to MPLS links. Then simulate packet loss and check the link switchover result.

### 7.1.2.2 Initialization

AR3 is used as an example in this lab. In the previous lab, some configurations have been manually performed, including address allocation, NAT, and static route configurations. To enable the controller to manage AR3 in this lab, you need to clear the configurations of AR3. The commands configured in the previous lab will be re-delivered through the controller.

```
<AR3>set factory-configuration from default
  Warning: The current factory configuration will be replaced, and it's irreversible. Are you sure to set
the factory configuration?[Y/N]:y
  Info: Successfully set factory config!
<AR3>factory-configuration reset
Warning: It will clean the configuration which you have saved. If you have set the factory-
configuration by hand, it will start from the modified factory-configuration, else it will start from the
original one, when you restart the device. Continue? [y/n]:y
  Info: Successfully set factory config!
<AR3>reset saved-configuration
  Warning: This will delete the configuration in the flash memory.
  The device configurations will be erased to reconfigure.
  Are you sure? (y/n):y
Info: Clear the configuration in the device successfully.
<AR3>reboot fast
System will reboot! Continue? [y/n]:y
```

### 7.1.2.3 Pre-configuring AR_Server_SW

AR_Server_SW is used to implement interconnection between the ARs and the controller. In this lab, AR_Server_SW is used to simulate Internet links, and needs to be configured

to provide the DHCP service for the ARs to onboard. Therefore, some pre-configurations need to be performed on the switch.

# Create an interconnection VLAN and assign the interfaces connected to the ARs to the VLAN.

```
[AR_Server_SW]vlan 650
[AR_Server_SW-vlan650] quit
[AR_Server_SW]interface GigabitEthernet0/0/1
[AR_Server_SW-GigabitEthernet0/0/1] description TO_AR
[AR_Server_SW-GigabitEthernet0/0/1] port default vlan 650
[AR_Server_SW-GigabitEthernet0/0/1] quit
[AR_Server_SW]interface GigabitEthernet0/0/2
[AR_Server_SW-GigabitEthernet0/0/2] description TO_AR
[AR_Server_SW-GigabitEthernet0/0/2] port default vlan 650
[AR_Server_SW-GigabitEthernet0/0/2] quit
[AR_Server_SW]interface GigabitEthernet0/0/3
[AR_Server_SW-GigabitEthernet0/0/3] description TO_AR
[AR_Server_SW-GigabitEthernet0/0/3] port default vlan 650
[AR_Server_SW-GigabitEthernet0/0/3] quit
```

# Enable DHCP globally, create VLANIF 650, enable DHCP on the interface, and configure static IP address binding and Option 148.

```
[AR_Server_SW]dhcp enable
[AR_Server_SW]interface Vlanif650
[AR_Server_SW-Vlanif650] ip address 65.0.0.254 255.255.255.0
[AR_Server_SW-Vlanif650] dhcp select interface
[AR_Server_SW-Vlanif650] dhcp server static-bind ip-address 65.0.0.1 mac-address c8a7-7600-3e51
[AR_Server_SW-Vlanif650] dhcp server static-bind ip-address 65.0.0.2 mac-address c8a7-7600-3f64
[AR_Server_SW-Vlanif650] dhcp server static-bind ip-address 65.0.0.3 mac-address cc64-a651-7f1c
[AR_Server_SW-Vlanif650] dhcp server option 148 ascii agilemode=agile-cloud;agilemanage-
mode=ip;agilemanage-domain=172.99.0.99;agilemanage-port=10020;
```

To facilitate management, configure static IP address binding to statically assign IP addresses 65.0.0.1, 65.0.0.2, and 65.0.0.3 to AR1, AR2, and AR3 respectively.

In this lab, all ARs connect to AR_Server_SW through GE0/0/9. View the MAC addresses on the ARs and obtain the MAC addresses required in the static binding configuration.

# Create a VLAN and assign the interface connected to the controller to the VLAN.

```
[AR_Server_SW]vlan 99
[AR_Server_SW-vlan99] quit
[AR_Server_SW]interface Vlanif99
[AR_Server_SW-Vlanif99] ip address 172.99.0.254 255.255.255.0
```

Configure VLANIF 99 as the gateway of the controller.

## 7.1.2.4 Pre-configuring the MPLS Network

In this lab, a network consisting of six NEs is used to simulate an MPLS network. The network only simulates MPLS access links, and MPLS and MPLS VPN do not need to be configured. Therefore, only OSPF needs to be configured and enabled on all network-wide interconnection interfaces (including the interfaces connected to the ARs).

Step 1 Perform basic device configurations.

Configure the command validation mode as immediate validation and configure IP addresses for interconnection and Loopback0 interfaces. Loopback0 interface addresses need to be configured according to the following table.

**Table 7-11 Loopback0 IP addresses**

| Device ID | Loopback0 IP Address |
|---|---|
| PE1 | 10.0.1.1 |
| PE2 | 10.0.2.2 |
| PE3 | 10.0.3.3 |
| PE4 | 10.0.4.4 |
| P1 | 10.0.5.5 |
| P2 | 10.0.6.6 |

# Name the devices.

N/A

# Configure the command validation mode as immediate validation.

```
<PE1>system-view immediately
```

```
<PE2>system-view immediately
```

```
<PE3>system-view immediately
```

```
<PE4>system-view immediately
```

```
<P1>system-view immediately
```

```
<P2>system-view immediately
```

# Disable DCN globally on each device.

```
[PE1] undo dcn
Warning: This operation will disable DCN function. Continue? [Y/N]:y
```

Here, PE1 is used as an example. Repeat this operation for other devices.

By default, DCN is enabled on NE router interfaces. To facilitate the lab, disable DCN globally on all devices.

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE1.

```
[PE1]interface LoopBack0
[PE1-LoopBack0] ip address 10.0.1.1 32
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0]ip address 10.0.0.1 30
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
[PE1-GigabitEthernet0/5/1]ip address 10.0.0.5 30
[PE1-GigabitEthernet0/5/1] quit
[PE1]interface GigabitEthernet0/5/2
[PE1-GigabitEthernet0/5/2] ip address 10.0.0.30 30
[PE1-GigabitEthernet0/5/2] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE2.

```
[PE2]interface LoopBack0
[PE2-LoopBack0] ip address 10.0.2.2 32
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ip address 10.0.0.2 30
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ip address 10.0.0.9 30
[PE2-GigabitEthernet0/5/1] quit
[PE2]interface GigabitEthernet0/5/2
[PE2-GigabitEthernet0/5/2] ip address 10.0.0.34 255.255.255.252
[PE2-GigabitEthernet0/5/2] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE3.

```
[PE3]interface LoopBack0
[PE3-LoopBack0] ip address 10.0.3.3 32
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] ip address 10.0.0.25 30
[PE3-GigabitEthernet0/5/0] quit
[PE3]interface GigabitEthernet0/5/1
[PE3-GigabitEthernet0/5/1] ip address 10.0.0.18 30
[PE3-GigabitEthernet0/5/1] quit
[PE3]interface GigabitEthernet0/5/2
[PE3-GigabitEthernet0/5/2] ip address 10.0.0.38 255.255.255.252
[PE3-GigabitEthernet0/5/2] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE4.

```
[PE4]interface LoopBack0
[PE4-LoopBack0] ip address 10.0.4.4 32
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] ip address 10.0.0.26 30
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] ip address 10.0.0.22 30
[PE4-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on P1.

```
[P1]interface LoopBack0
[P1-LoopBack0] ip address 10.0.5.5 32
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] ip address 10.0.0.13 30
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] ip address 10.0.0.6 30
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] ip address 10.0.0.17 30
[P1-GigabitEthernet0/5/2] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on P2.

```
[P2]interface LoopBack0
[P2-LoopBack0] ip address 10.0.6.6 32
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ip address 10.0.0.14 30
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ip address 10.0.0.10 30
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] ip address 10.0.0.21 30
[P2-GigabitEthernet0/5/2] quit
```

# Test the connectivity between the interconnection interfaces on PE1, P2, and PE3.

```
[PE1]ping -c 1 10.0.0.6
  PING 10.0.0.6: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.6: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.6 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

[PE1]ping -c 1 10.0.0.2
  PING 10.0.0.2: 56  data bytes, press CTRL_C to break
```

```
    Reply from 10.0.0.2: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.2 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 1/1/1 ms


[P2]ping -c 1 10.0.0.9
  PING 10.0.0.9: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.9: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.9 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

[P2]ping -c 1 10.0.0.13
  PING 10.0.0.13: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.13: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.13 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

[PE3]ping -c 1 10.0.0.17
  PING 10.0.0.17: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.17: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.17 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

[PE3]ping -c 1 10.0.0.26
  PING 10.0.0.26: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.26: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.26 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

The interface connectivity test succeeds. (The interconnection interfaces connected to the ARs are not tested.)

Step 2 Configure OSPF.

Configure all devices in OSPF area 0, use the IP address of Loopback0 as the router ID, set the OSPF process ID to 1, and enable OSPF on all interconnection interfaces.

# Configure PE1.

```
[PE1]ospf 1 router-id 10.0.1.1
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
[PE1]interface LoopBack0
[PE1-LoopBack0] ospf enable 1 area 0
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0] ospf enable 1 area 0
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
[PE1-GigabitEthernet0/5/1] ospf enable 1 area 0
[PE1-GigabitEthernet0/5/1] quit
[PE1]interface GigabitEthernet0/5/2
[PE1-GigabitEthernet0/5/2] ospf enable 1 area 0
[PE1-GigabitEthernet0/5/2] quit
```

# Configure PE2.

```
[PE2]ospf 1 router-id 10.0.2.2
[PE2-ospf-1] area 0
[PE2-ospf-1-area-0.0.0.0] quit
[PE2-ospf-1] quit
[PE2]interface LoopBack0
[PE2-LoopBack0] ospf enable 1 area 0
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ospf enable 1 area 0
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ospf enable 1 area 0
[PE2-GigabitEthernet0/5/1] quit
[PE2]interface GigabitEthernet0/5/2
[PE2-GigabitEthernet0/5/2] ospf enable 1 area 0
[PE2-GigabitEthernet0/5/2] quit
```

# Configure PE3.

```
[PE3]ospf 1 router-id 10.0.3.3
[PE3-ospf-1] area 0
[PE3-ospf-1-area-0.0.0.0] quit
[PE3-ospf-1] quit
[PE3]interface LoopBack0
[PE3-LoopBack0] ospf enable 1 area 0
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] ospf enable 1 area 0
[PE3-GigabitEthernet0/5/0] quit
[PE3]interface GigabitEthernet0/5/1
[PE3-GigabitEthernet0/5/1] ospf enable 1 area 0
```

```
[PE3-GigabitEthernet0/5/1] quit
[PE3]interface GigabitEthernet0/5/2
[PE3-GigabitEthernet0/5/2] ospf enable 1 area 0
[PE3-GigabitEthernet0/5/2] quit
```

# Configure PE4.

```
[PE4]ospf 1 router-id 10.0.4.4
[PE4-ospf-1] area 0
[PE4-ospf-1-area-0.0.0.0] quit
[PE4-ospf-1] quit
[PE4]interface LoopBack0
[PE4-LoopBack0] ospf enable 1 area 0
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] ospf enable 1 area 0
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] ospf enable 1 area 0
[PE4-GigabitEthernet0/5/1] quit
```

# Configure P1.

```
[P1]ospf 1 router-id 10.0.5.5
[P1-ospf-1] area 0
[P1-ospf-1-area-0.0.0.0] quit
[P1-ospf-1] quit
[P1]interface LoopBack0
[P1-LoopBack0] ospf enable 1 area 0
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] ospf enable 1 area 0
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] ospf enable 1 area 0
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] ospf enable 1 area 0
[P1-GigabitEthernet0/5/2] quit
```

# Configure P2.

```
[P2]ospf 1 router-id 10.0.6.6
[P2-ospf-1] area 0
[P2-ospf-1-area-0.0.0.0] quit
[P2-ospf-1] quit
[P2]interface LoopBack0
[P2-LoopBack0] ospf enable 1 area 0
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ospf enable 1 area 0
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ospf enable 1 area 0
```

```
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] ospf enable 1 area 0
[P2-GigabitEthernet0/5/2] quit
```

# Check the OSPF neighbor relationships on P1, PE2, and PE4.

```
[P1]display ospf peer brief
(M) Indicates MADJ neighbor


             OSPF Process 1 with Router ID 10.0.5.5
                     Peer Statistic Information
Total number of peer(s): 3
Peer(s) in full state: 3
--------------------------------------------------------------------------------
Area Id          Interface                     Neighbor id          State
0.0.0.0          GE0/5/0                         10.0.6.6           Full
0.0.0.0          GE0/5/1                         10.0.1.1           Full
0.0.0.0          GE0/5/2                         10.0.3.3           Full
--------------------------------------------------------------------------------
```

```
[PE2]display ospf peer brief
(M) Indicates MADJ neighbor


             OSPF Process 1 with Router ID 10.0.2.2
                     Peer Statistic Information
Total number of peer(s): 2
Peer(s) in full state: 2
--------------------------------------------------------------------------------
Area Id          Interface                     Neighbor id          State
0.0.0.0          GE0/5/0                         10.0.1.1           Full
0.0.0.0          GE0/5/1                         10.0.6.6           Full
--------------------------------------------------------------------------------
```

```
[PE4]display ospf peer brief
(M) Indicates MADJ neighbor


             OSPF Process 1 with Router ID 10.0.4.4
                     Peer Statistic Information
Total number of peer(s): 2
Peer(s) in full state: 2
--------------------------------------------------------------------------------
Area Id          Interface                     Neighbor id          State
0.0.0.0          GE0/5/0                         10.0.3.3           Full
0.0.0.0          GE0/5/1                         10.0.6.6           Full
--------------------------------------------------------------------------------
```

The OSPF neighbor relationships have been established.

# Check OSPF routes on PE1, PE2, and PE3.

```
[PE1]display ospf routing
          OSPF Process 1 with Router ID 10.0.1.1
                Routing Tables

Routing for Network
Destination        Cost    Type       NextHop        AdvRouter      Area
10.0.0.0/30        1       Direct     10.0.0.1       10.0.1.1       0.0.0.0
10.0.0.4/30        1       Direct     10.0.0.5       10.0.1.1       0.0.0.0
10.0.0.8/30        2       Transit    10.0.0.2       10.0.2.2       0.0.0.0
10.0.0.12/30       2       Transit    10.0.0.6       10.0.6.6       0.0.0.0
10.0.0.16/30       2       Transit    10.0.0.6       10.0.3.3       0.0.0.0
10.0.0.20/30       3       Transit    10.0.0.6       10.0.4.4       0.0.0.0
10.0.0.20/30       3       Transit    10.0.0.2       10.0.4.4       0.0.0.0
10.0.0.24/30       3       Transit    10.0.0.6       10.0.4.4       0.0.0.0
10.0.0.28/30       1       Direct     10.0.0.30      10.0.1.1       0.0.0.0
10.0.0.32/30       2       Stub       10.0.0.2       10.0.2.2       0.0.0.0
10.0.0.36/30       3       Stub       10.0.0.6       10.0.3.3       0.0.0.0
10.0.1.1/32        0       Direct     10.0.1.1       10.0.1.1       0.0.0.0
10.0.2.2/32        1       Stub       10.0.0.2       10.0.2.2       0.0.0.0
10.0.4.4/32        3       Stub       10.0.0.6       10.0.4.4       0.0.0.0
10.0.4.4/32        3       Stub       10.0.0.2       10.0.4.4       0.0.0.0
10.0.5.5/32        1       Stub       10.0.0.6       10.0.5.5       0.0.0.0
10.0.6.6/32        2       Stub       10.0.0.6       10.0.6.6       0.0.0.0
10.0.6.6/32        2       Stub       10.0.0.2       10.0.6.6       0.0.0.0

Total Nets: 15
Intra Area: 15    Inter Area: 0    ASE: 0    NSSA: 0
```

PE1 has learned the routes destined for other branches.

```
[PE2]display ospf routing
          OSPF Process 1 with Router ID 10.0.2.2
                Routing Tables

Routing for Network
Destination        Cost    Type       NextHop        AdvRouter      Area
10.0.0.0/30        1       Direct     10.0.0.2       10.0.2.2       0.0.0.0
10.0.0.4/30        2       Transit    10.0.0.1       10.0.1.1       0.0.0.0
10.0.0.8/30        1       Direct     10.0.0.9       10.0.2.2       0.0.0.0
10.0.0.12/30       2       Transit    10.0.0.10      10.0.6.6       0.0.0.0
10.0.0.16/30       3       Transit    10.0.0.10      10.0.3.3       0.0.0.0
10.0.0.16/30       3       Transit    10.0.0.1       10.0.3.3       0.0.0.0
10.0.0.20/30       2       Transit    10.0.0.10      10.0.4.4       0.0.0.0
10.0.0.24/30       3       Transit    10.0.0.10      10.0.4.4       0.0.0.0
10.0.0.28/30       2       Stub       10.0.0.1       10.0.1.1       0.0.0.0
10.0.0.32/30       1       Direct     10.0.0.34      10.0.2.2       0.0.0.0
10.0.0.36/30       4       Stub       10.0.0.10      10.0.3.3       0.0.0.0
10.0.0.36/30       4       Stub       10.0.0.1       10.0.3.3       0.0.0.0
10.0.1.1/32        1       Stub       10.0.0.1       10.0.1.1       0.0.0.0
10.0.2.2/32        0       Direct     10.0.2.2       10.0.2.2       0.0.0.0
10.0.4.4/32        2       Stub       10.0.0.10      10.0.4.4       0.0.0.0
```

```
10.0.5.5/32         2         Stub      10.0.0.10       10.0.5.5        0.0.0.0
10.0.5.5/32         2         Stub      10.0.0.1        10.0.5.5        0.0.0.0
10.0.6.6/32         1         Stub      10.0.0.10       10.0.6.6        0.0.0.0

Total Nets: 15
Intra Area: 15  Inter Area: 0  ASE: 0  NSSA: 0
```

PE2 has learned the routes destined for other branches.

```
[PE3]display ospf routing
          OSPF Process 1 with Router ID 10.0.3.3
                Routing Tables

Routing for Network
Destination         Cost      Type      NextHop         AdvRouter       Area
10.0.0.0/30         3         Transit   10.0.0.17       10.0.1.1        0.0.0.0
10.0.0.4/30         2         Transit   10.0.0.17       10.0.1.1        0.0.0.0
10.0.0.8/30         3         Transit   10.0.0.17       10.0.2.2        0.0.0.0
10.0.0.8/30         3         Transit   10.0.0.26       10.0.2.2        0.0.0.0
10.0.0.12/30        2         Transit   10.0.0.17       10.0.6.6        0.0.0.0
10.0.0.16/30        1         Direct    10.0.0.18       10.0.3.3        0.0.0.0
10.0.0.20/30        2         Transit   10.0.0.26       10.0.4.4        0.0.0.0
10.0.0.24/30        1         Direct    10.0.0.25       10.0.3.3        0.0.0.0
10.0.0.28/30        3         Stub      10.0.0.17       10.0.1.1        0.0.0.0
10.0.0.32/30        4         Stub      10.0.0.17       10.0.2.2        0.0.0.0
10.0.0.32/30        4         Stub      10.0.0.26       10.0.2.2        0.0.0.0
10.0.0.36/30        1         Direct    10.0.0.38       10.0.3.3        0.0.0.0
10.0.1.1/32         2         Stub      10.0.0.17       10.0.1.1        0.0.0.0
10.0.2.2/32         3         Stub      10.0.0.17       10.0.2.2        0.0.0.0
10.0.2.2/32         3         Stub      10.0.0.26       10.0.2.2        0.0.0.0
10.0.4.4/32         1         Stub      10.0.0.26       10.0.4.4        0.0.0.0
10.0.5.5/32         1         Stub      10.0.0.17       10.0.5.5        0.0.0.0
10.0.6.6/32         2         Stub      10.0.0.17       10.0.6.6        0.0.0.0
10.0.6.6/32         2         Stub      10.0.0.26       10.0.6.6        0.0.0.0

Total Nets: 15
Intra Area: 15  Inter Area: 0  ASE: 0  NSSA: 0
```

PE3 has learned the routes destined for other branches.

## 7.1.2.5 Creating Sites and Onboarding Devices

Create sites Branch1 and Branch2. Site HQ has been created in the previous lab.

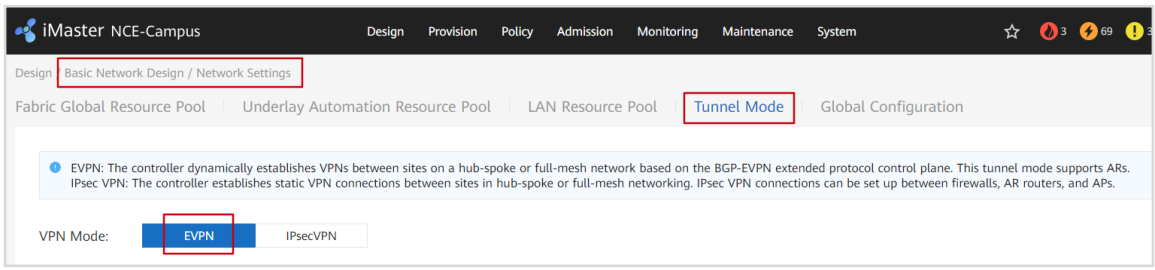
Add devices to the sites as planned. The ARs will obtain the controller's IP address from AR_Server_SW upon startup and register with the controller.

Note: Ensure that AR1, AR2, and AR3 start with factory defaults. For details about how to restore factory settings, refer to the previous lab. The devices need to register with the controller without any input on the console port.

Step 1 Change the tunnel mode.

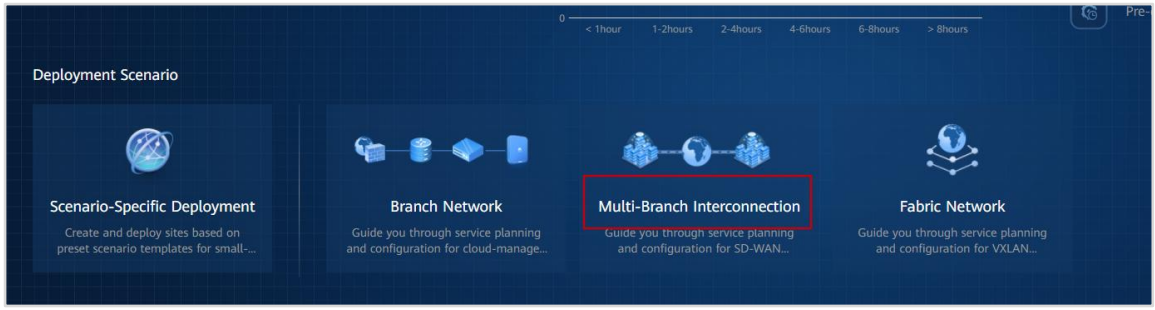
By default, the inter-site tunnel mode on the controller is **IPsec VPN**. To support SD-WAN configurations, switch the tunnel mode to **EVPN**.

# Change the mode.



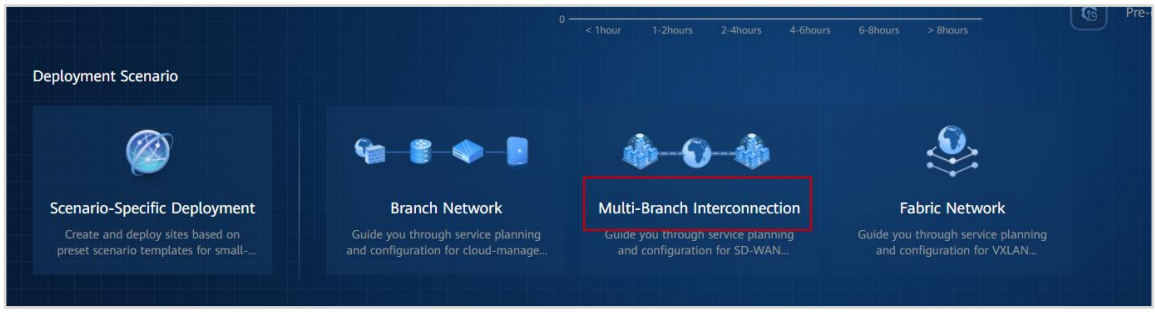

Choose **Design** > **Network Settings** from the main menu, click the **Tunnel Mode** tab, and switch the tunnel mode to **EVPN**.

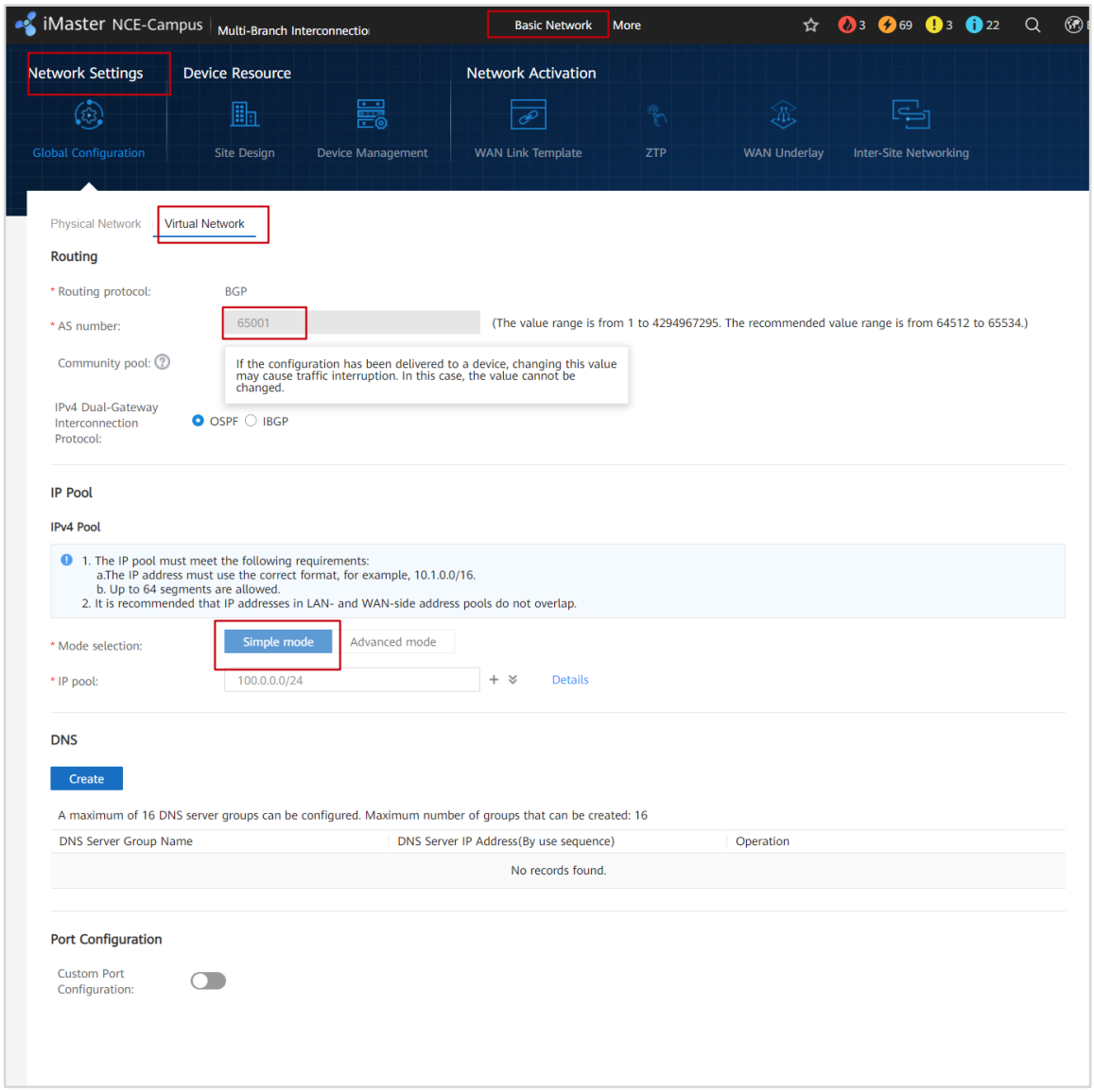
# View the home page.




After the tunnel mode is switched, **Multi-Branch Interconnection** is displayed in the **Deployment Scenario** area on the controller's home page. This lab is performed using this wizard mode.

Step 2 Configure virtual networks.






Access the configuration page and configure virtual networks as planned.

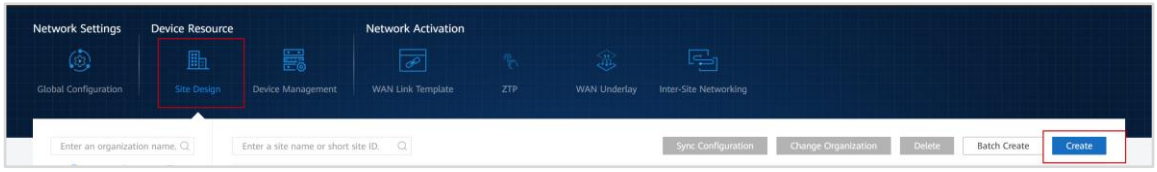
In the **IP pool** area, retain the default value **Simple mode** for **Mode selection**, enter an IP address and click the "+" button to add the entered IP address.

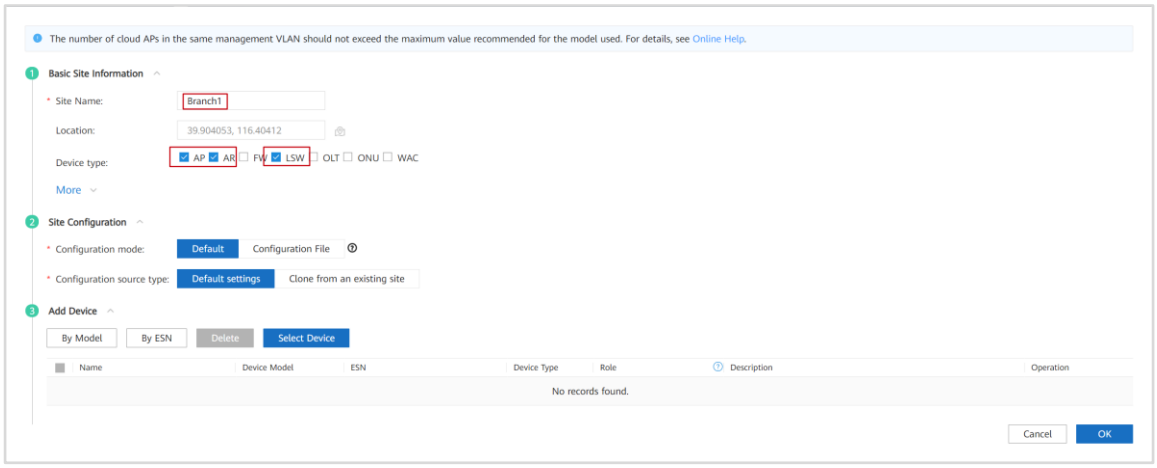
Then click **OK** at the bottom of the page.

Step 3 Design sites.

Click **Site Design** on the current navigation page and create sites Branch1 and Branch2.

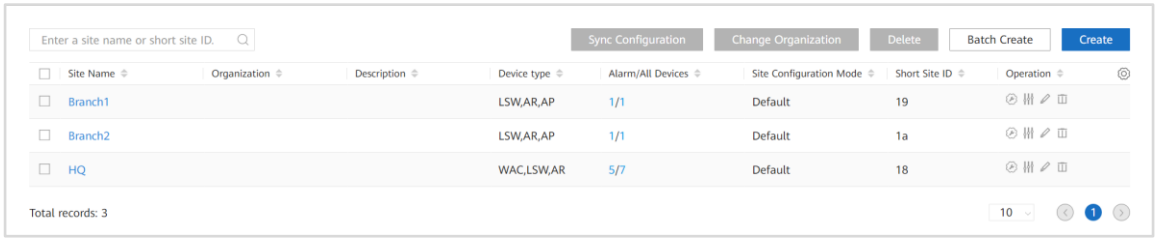
# Create the sites.






Click **Create** and create a site Branch1 (select **AP**, **AR** and **LSW** as device types).

Then repeat the preceding steps to create Branch2.

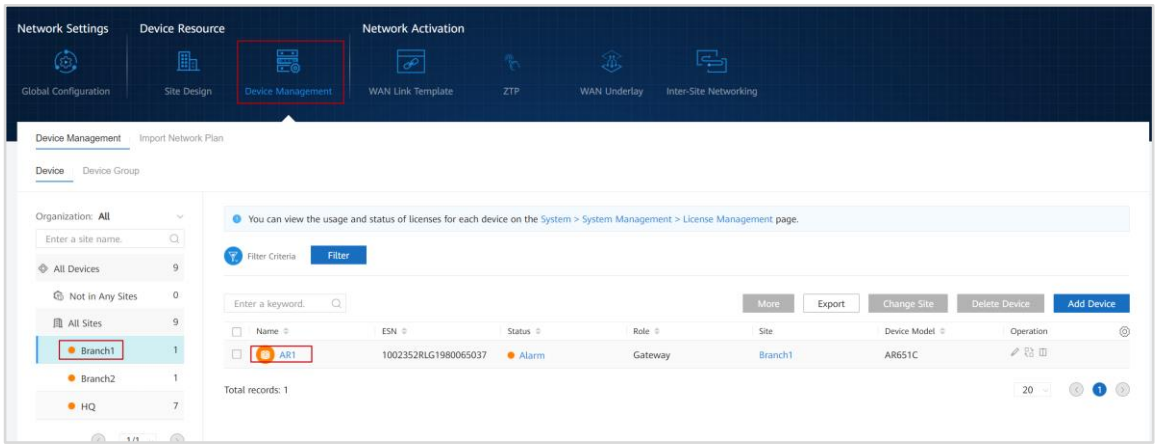



The sites are created successfully.

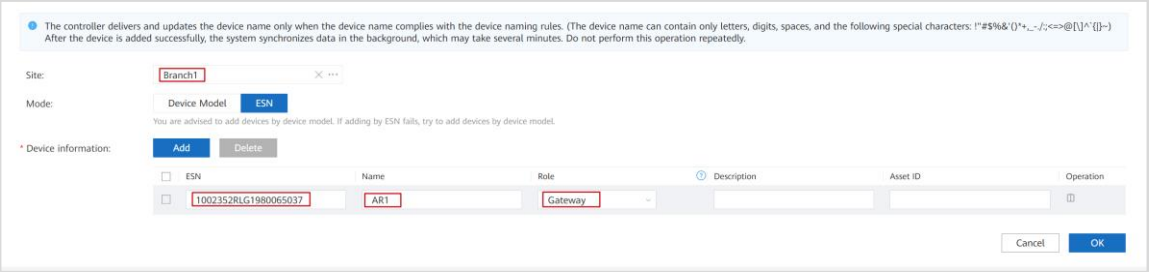
Step 4 Manage devices.

Add AR1 to Branch1, AR2 to Branch2, and AR3 to the existing site HQ.

# Manage the devices.

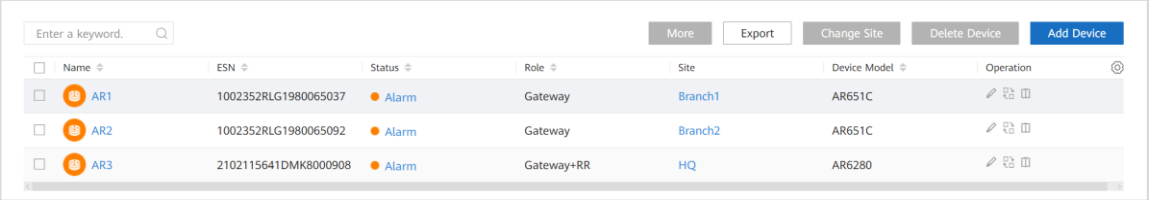



Click **Device Management** on the current navigation page and add the devices to the corresponding sites.




Add devices by ESN. Note that the roles of AR1 and AR2 are gateways, and the role of AR3 is **Gateway+RR**.

Add the three routers to the corresponding sites one by one.




The sites and devices are onboarded.

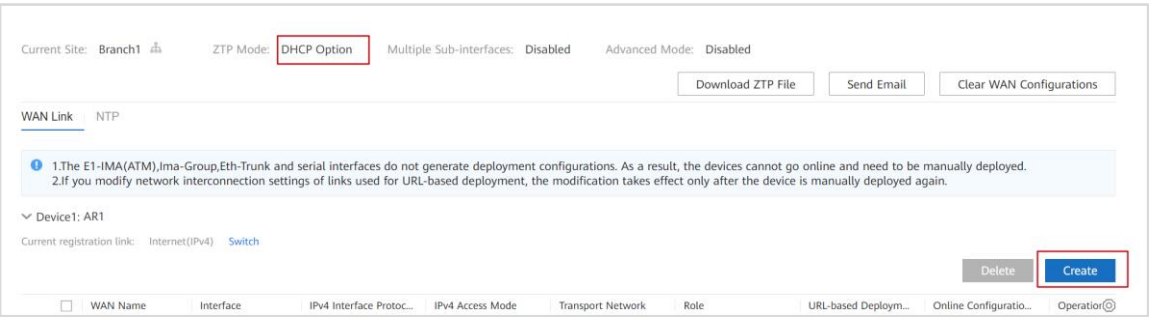
## 7.1.2.6 Deploying Devices

Complete SD-WAN deployment configurations, including ZTP, WAN-side underlay routes, and inter-site networking.

Step 1 Configure ZTP.

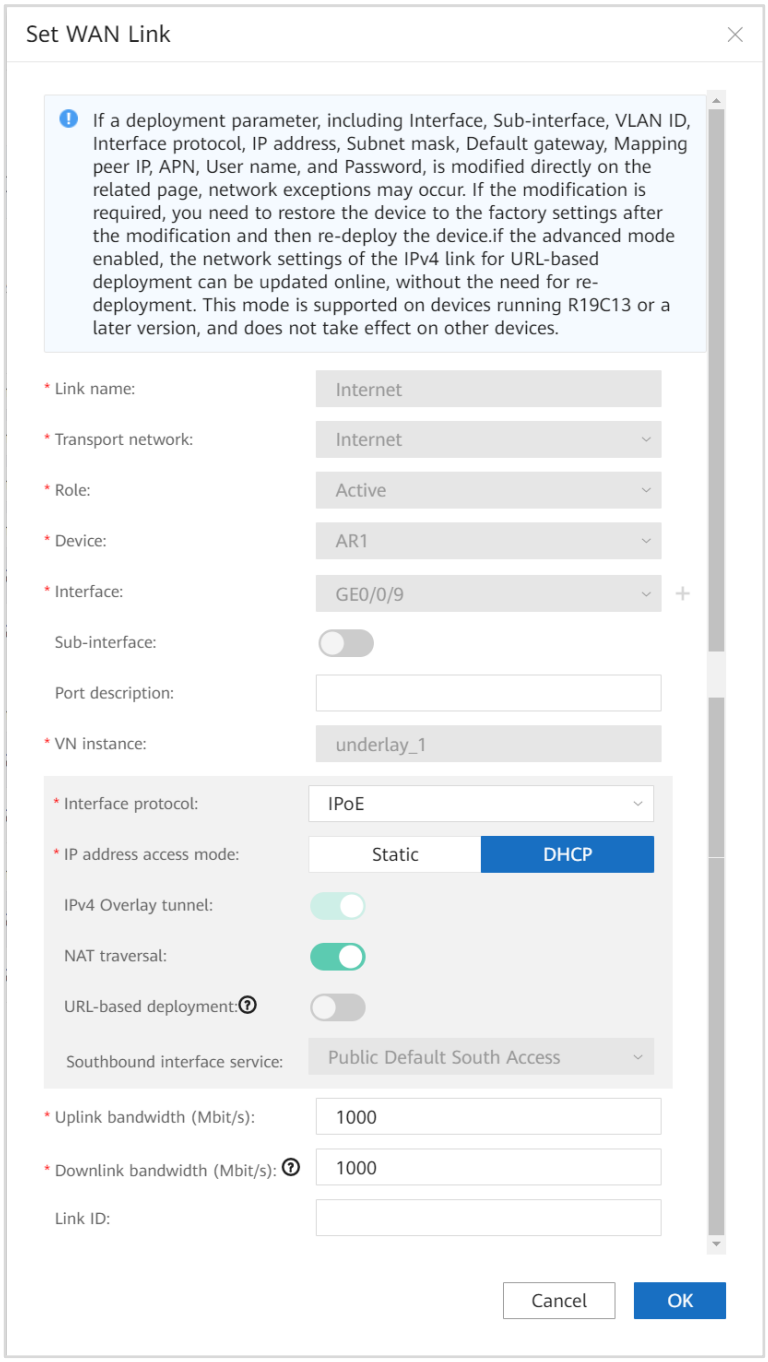
The ARs have obtained IP addresses through Internet links during startup and successfully registered with the controller for onboarding. After SD-WAN configurations are complete, the configurations will be re-delivered to Internet interfaces. In this case, you need to configure ZTP to enable the ARs to onboard again.

# Configure ZTP for AR1.




Click **ZTP** on the navigation page, select **Branch1** from the site list, and click **Click to deploy**.

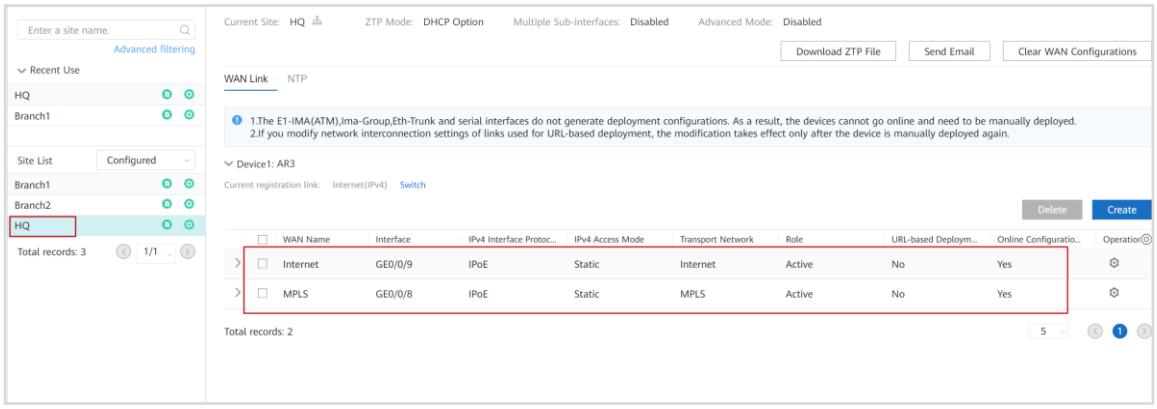
Select **DHCP Option** for the ZTP mode and click **Create**.




In the displayed dialog box, configure the Internet link and MPLS link for Branch1 as planned and then click **OK** in the lower right corner.

The configurations of other branches are similar and are not described here. Complete the configurations as planned.

# Check the ZTP configuration result.




Click the sites one by one to check the configuration.

# Check whether IP addresses are configured successfully.

```
<AR_Server_SW>ping -c 1 65.0.0.1
  PING 65.0.0.1: 56  data bytes, press CTRL_C to break
    Reply from 65.0.0.1: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 65.0.0.1 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms

<AR_Server_SW>ping -c 1 65.0.0.2
  PING 65.0.0.2: 56  data bytes, press CTRL_C to break
    Reply from 65.0.0.2: bytes=56 Sequence=1 ttl=255 time=32 ms

  --- 65.0.0.2 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 32/32/32 ms

<AR_Server_SW>ping -c 1 65.0.0.3
  PING 65.0.0.3: 56  data bytes, press CTRL_C to break
    Reply from 65.0.0.3: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 65.0.0.3 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 1/1/1 ms
```

```
[PE1]ping -c 1 10.0.0.29
  PING 10.0.0.29: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.29: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.29 ping statistics ---
    1 packet(s) transmitted
```

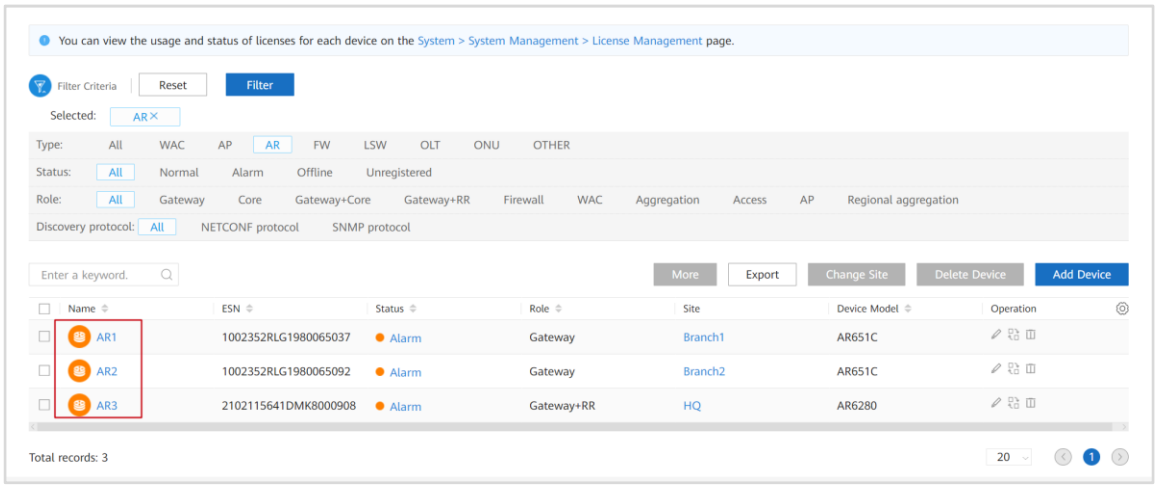
```
    1 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 1/1/1 ms
```

```
[PE2]ping -c 1 10.0.0.33
  PING 10.0.0.33: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.33: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.33 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 1/1/1 ms
```

```
[PE3]ping -c 1 10.0.0.37
  PING 10.0.0.37: 56  data bytes, press CTRL_C to break
    Reply from 10.0.0.37: bytes=56 Sequence=1 ttl=255 time=1 ms

  --- 10.0.0.37 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

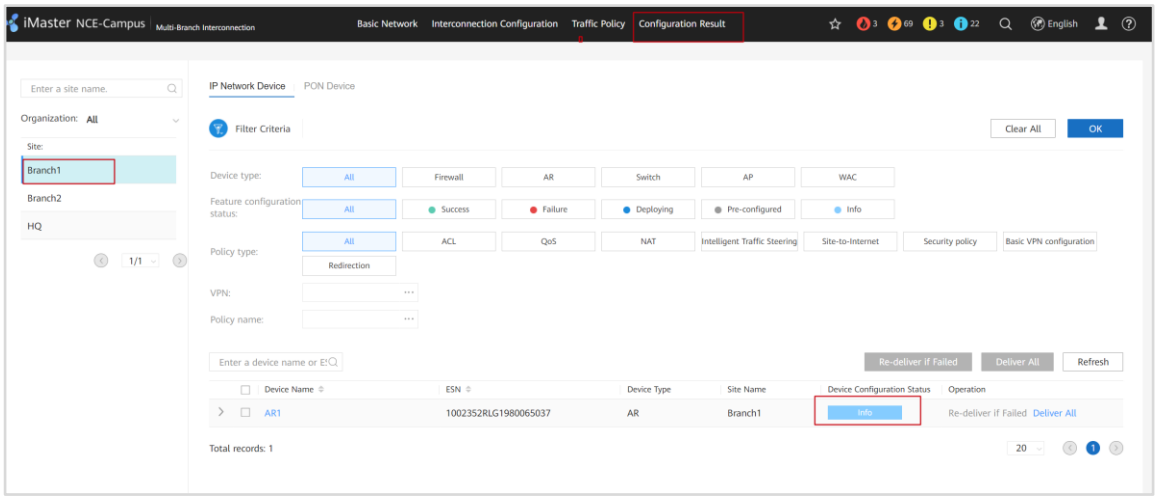
Check whether IP addresses of ARs are successfully configured on AR_Server_SW, PE1, PE2, and PE3, respectively.

# Check whether the ARs are online.




After ZTP is configured, the Internet and MPLS interfaces are bound to the corresponding VPN instances and ARs can be onboarded again.

# Verify configurations.




If the device configuration status is **Failure**, click **Re-deliver if Failed** in the **Operation** column to re-deliver the configuration.

If the deployment fails, you can restore the factory settings of the corresponding AR router. Log in to the CLI of the AR router and run the following commands in sequence. After each command is executed, the system prompts you to confirm the operation. Enter **y** to confirm the operation.

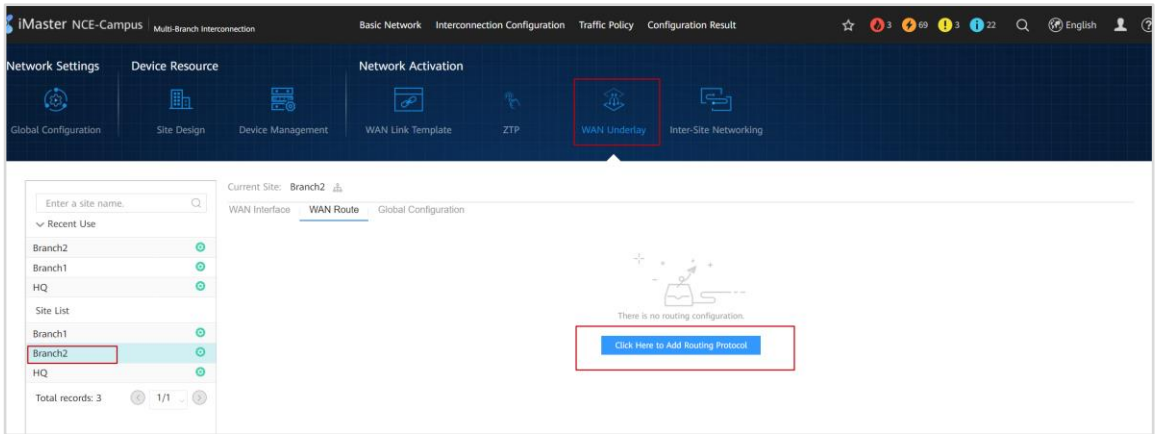
```
set factory-configuration from default
factory-configuration reset
delete /unreserved flash:/startup_v1.rdb
reset saved-configuration
reboot fast
```

After the commands are executed, the AR router will restart and restore to factory defaults. Then configure ZTP for the AR router again.

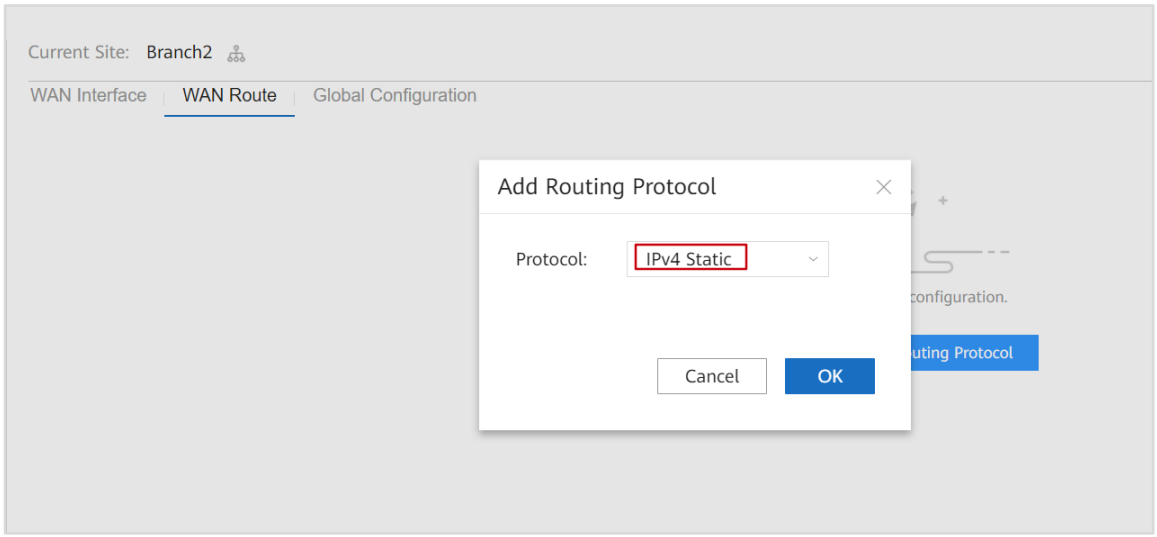
Step 2 Configure WAN-side underlay routes.

Configure WAN-side underlay routes so that traffic can be forwarded from the underlay network to external networks for underlay traffic forwarding through inter-site tunnels.

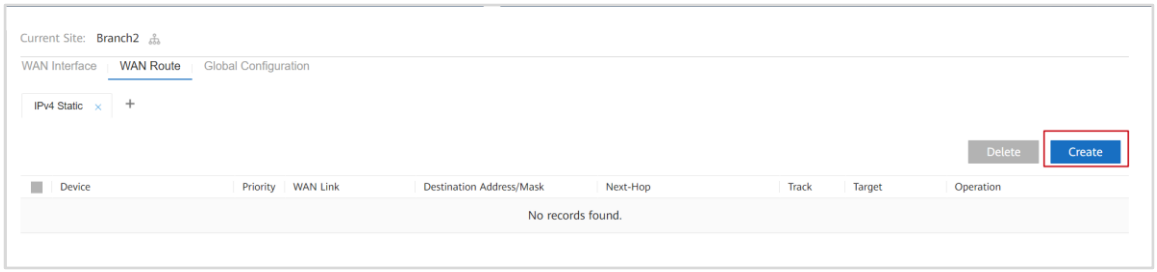
# Create WAN routes.

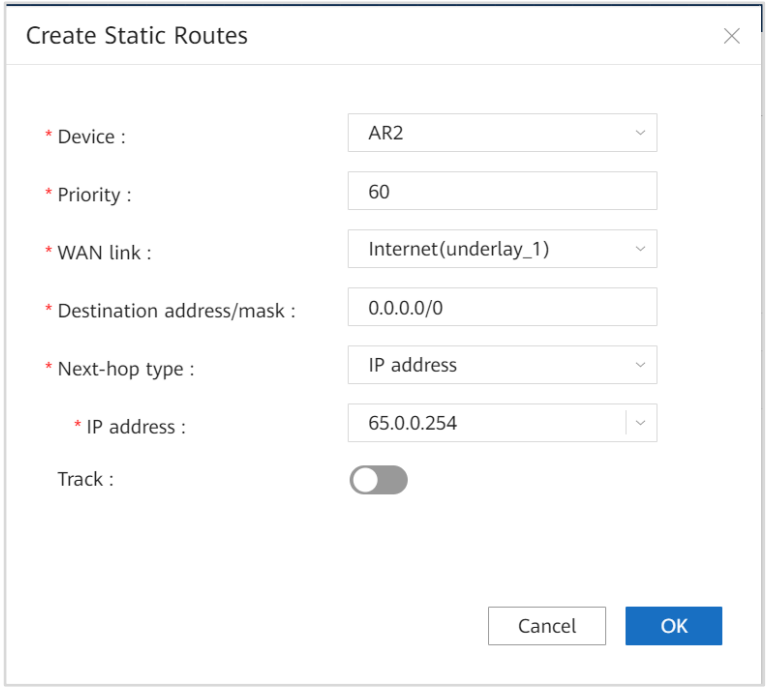



Click **WAN Underlay** on the navigation page, select **Branch2** from the site list, and add underlay routes.

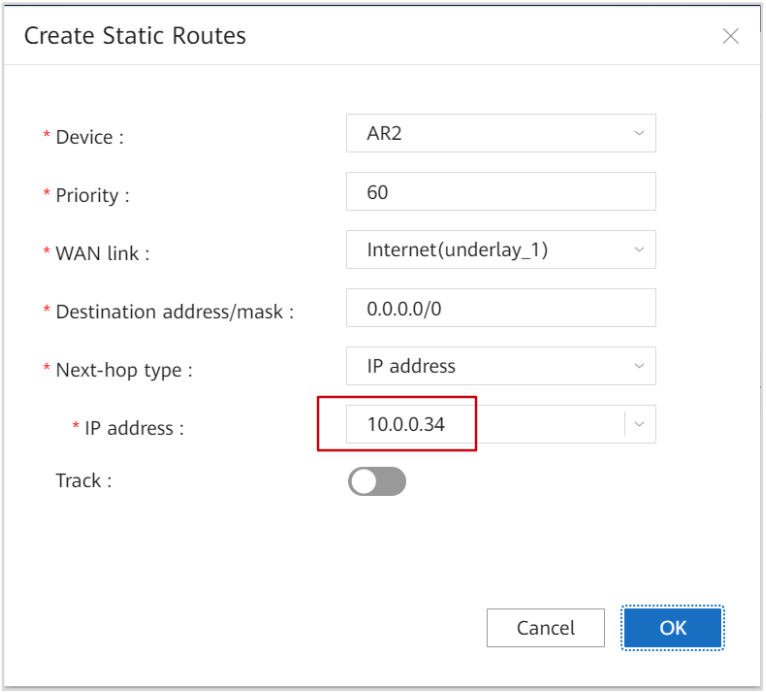



Select a static routing protocol.

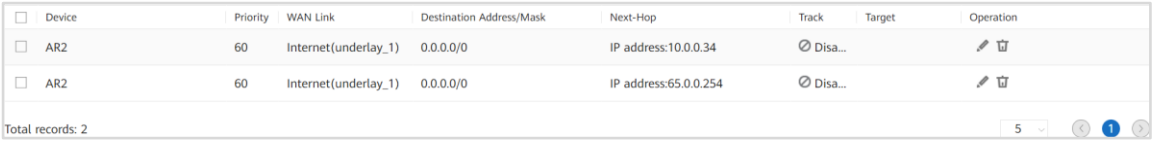



Configure a default route for the Internet link.




Configure a default route for the MPLS link.

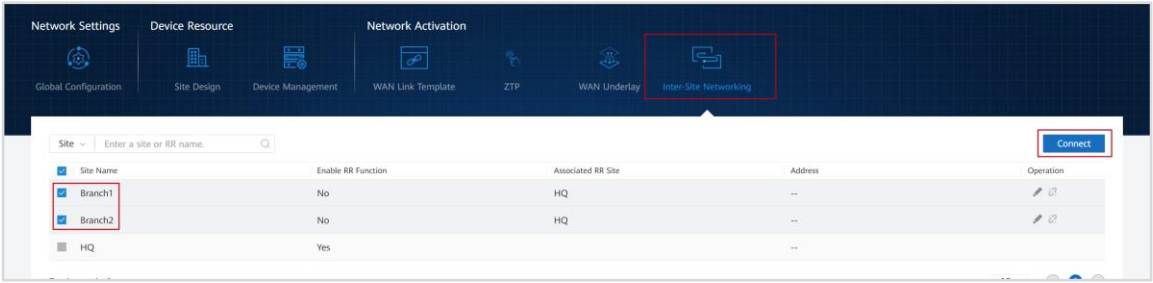


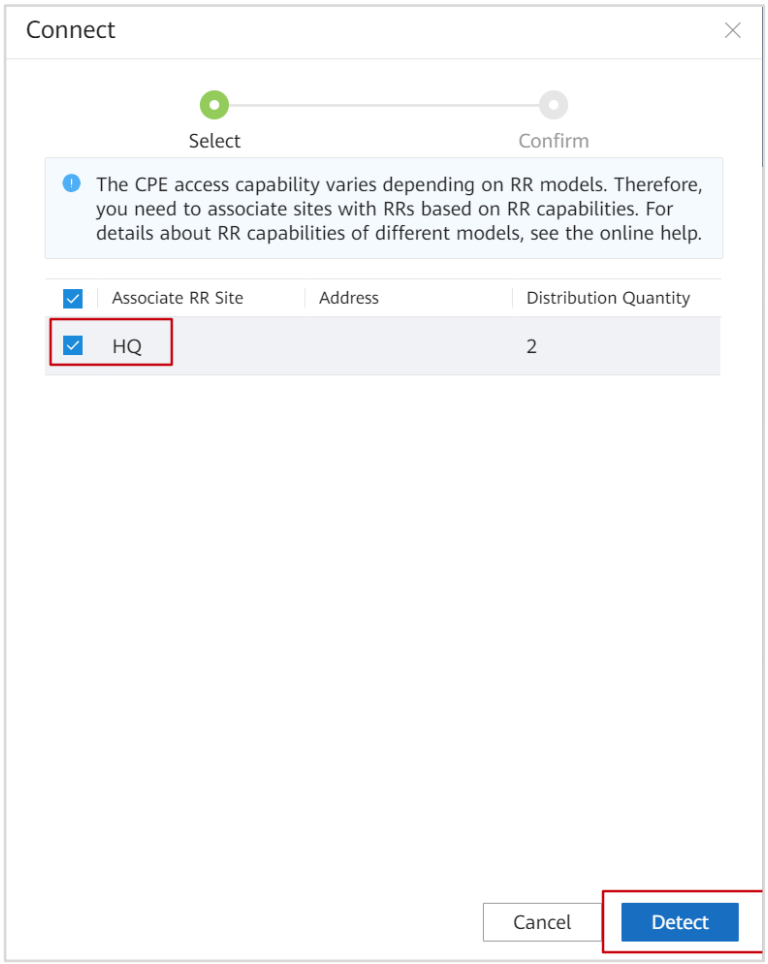

The WAN-side underlay routes of Branch2 have been configured. Configure WAN-side underlay routes for Branch1 and HQ in the same manner.

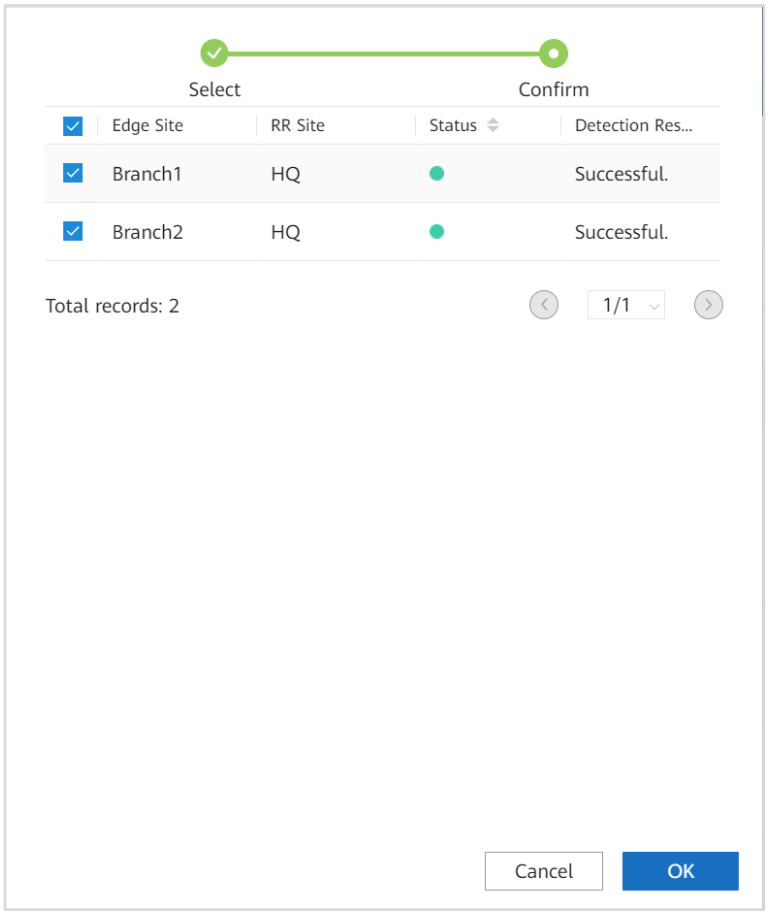
Step 3 Configure the inter-site networking.

Establish peer relationships between the RR (HQ) and Branch1 and Branch2.

# Connect branch sites to the RR.




Connect
Select
Confirm
The CPE access capability varies depending on RR models. Therefore, you need to associate sites with RRs based on RR capabilities. For details about RR capabilities of different models, see the online help.
Associate RR Site
Address
Distribution Quantity
HQ
2
Cancel
Detect




Click **Inter-Site Networking** on the navigation page to complete the configuration for connecting branch sites to the RR.

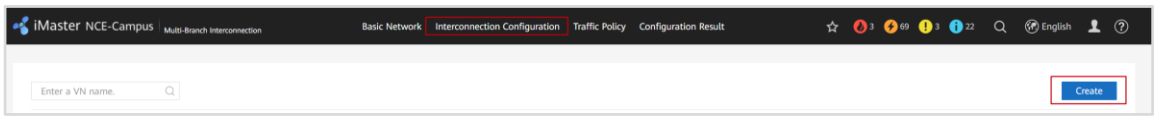
## 7.1.2.7 Creating Virtual Network public

Create a virtual network named **public**. This virtual network is used to implement some functions of AR3 described in 6 "VXLAN-based Virtualized Campus Network Deployment", including:

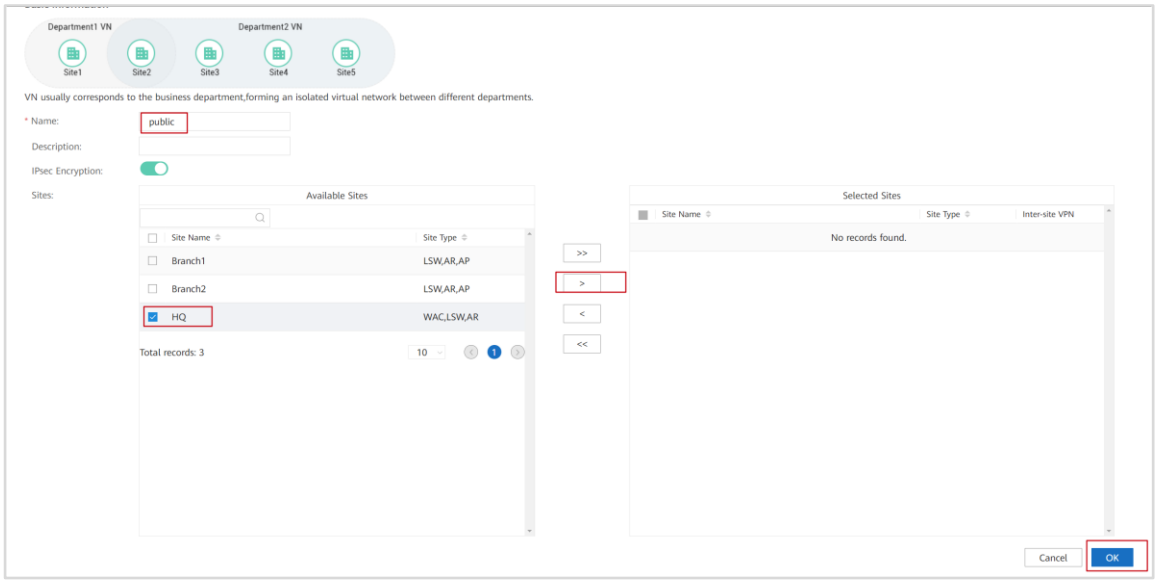
1. DHCP address pool and interface for device onboarding
2. DHCP address pool for terminal users
3. Source NAT for device onboarding
4. Destination NAT for Portal authentication
5. DHCP and E-mail services

Step 1 Create a virtual network.

# Create virtual network **public**.

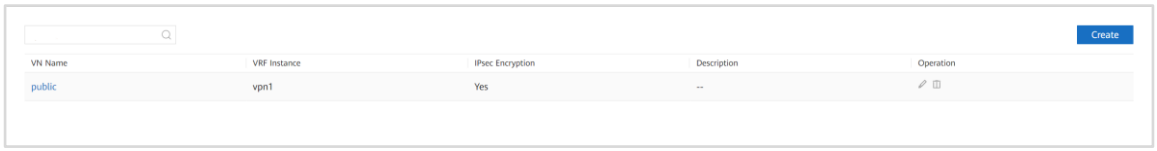



Click **Interconnection Configuration** on the top navigation bar. On the page that is displayed, click **Create** and create a VPN.




The VPN provides services only for the HQ site. Therefore, you only need to add the HQ site to virtual network **public**. Then click **OK**.

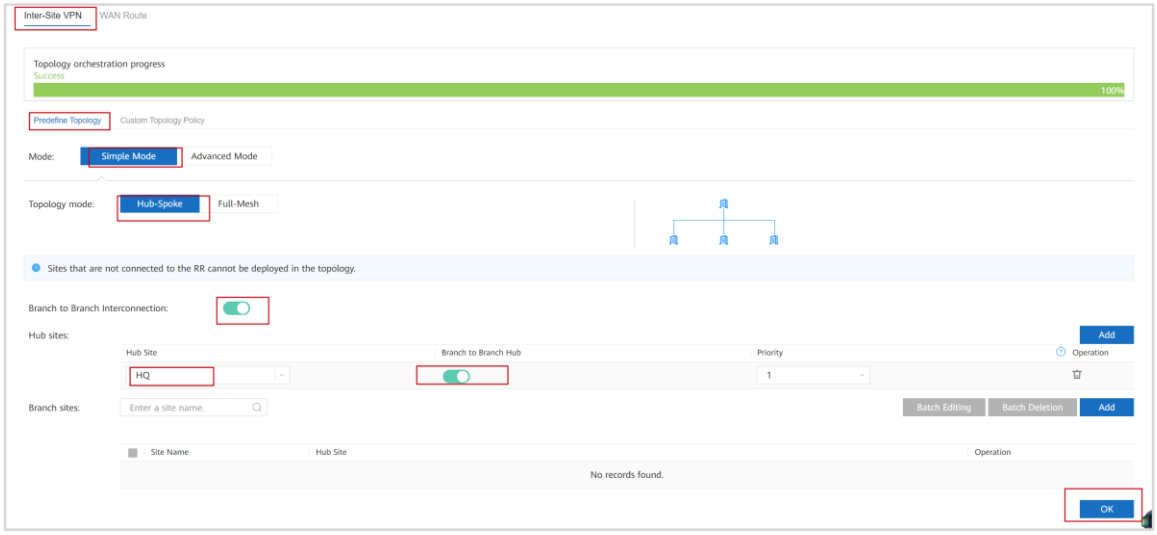
# Configure virtual network **public**.




Click the virtual network name to access the virtual network configuration page.

Step 2 Configure WAN services.

Configure WAN services for virtual network **public**, set the topology mode to **Hub-Spoke**, and set the HQ site as the hub site. This virtual network provides services only for the HQ site.

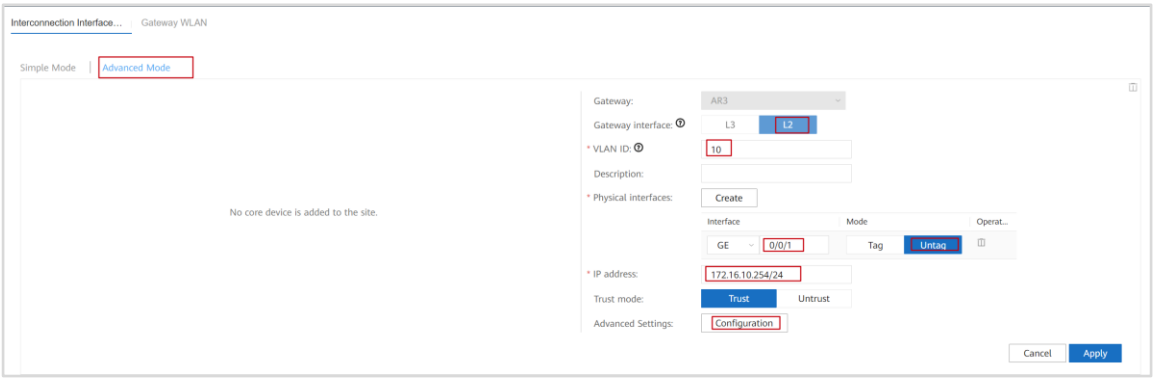


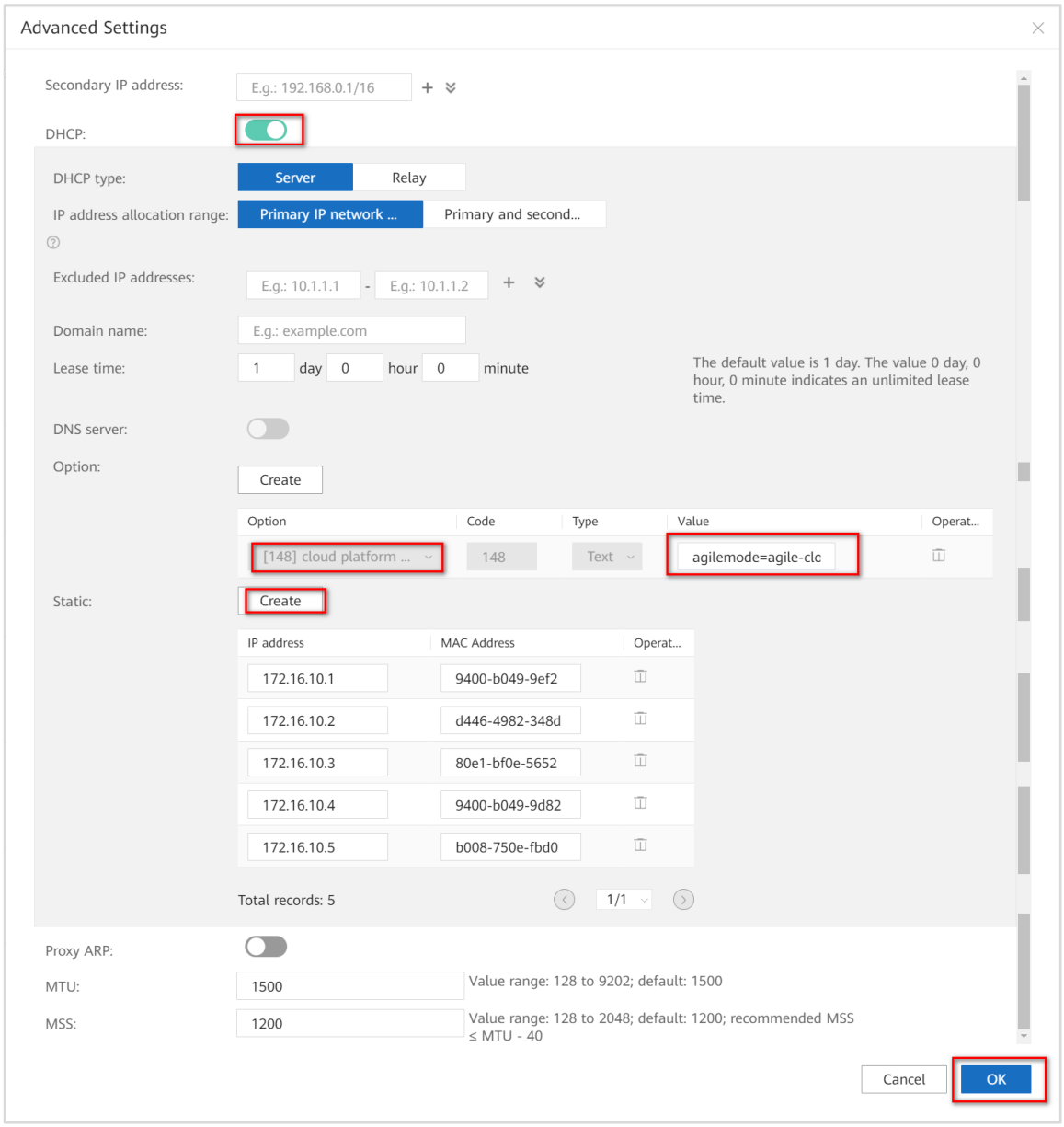

Click **WAN Service** on the virtual network configuration page and complete WAN service configurations.

Step 3 Configure LAN-WAN interconnection.

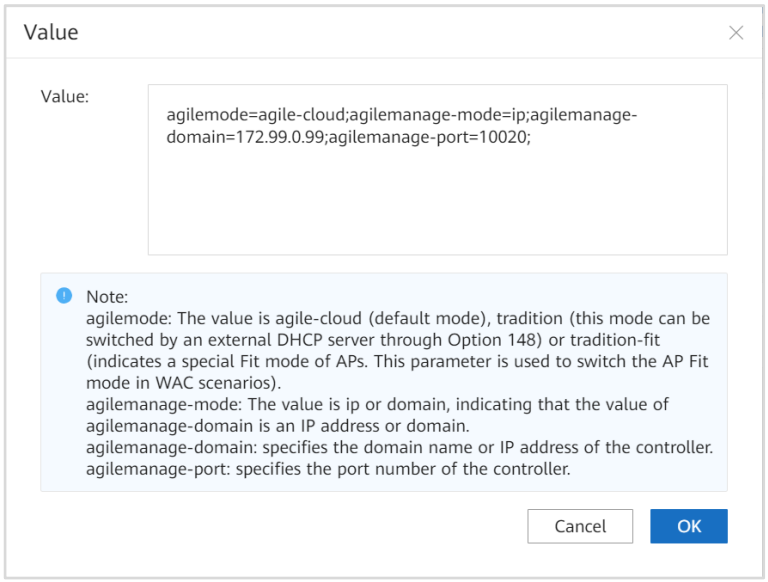

Click **LAN-WAN Interconnection** on the virtual network configuration page and configure the interconnection with Border on the fabric network. For details, see the AR3 pre-configuration and external network configuration in 6 "VXLAN-based Virtualized Campus Network Deployment."

Configure LAN-WAN interconnection for virtual network **public** of the HQ as planned.





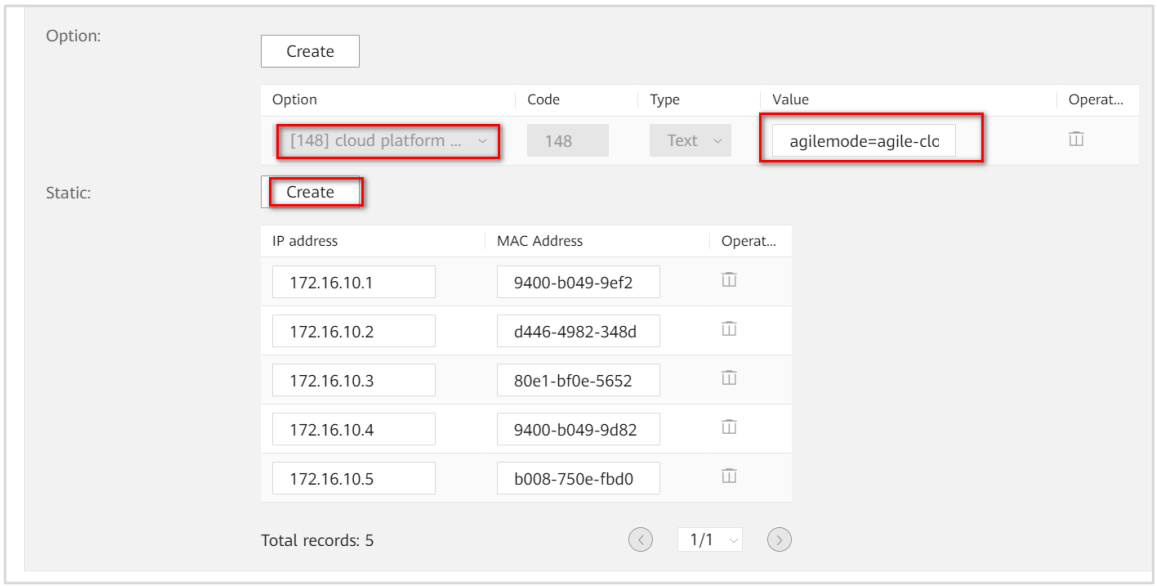

Enable DHCP, create Option 148, and click **Value** to configure the controller IP address.




The value of Option 148 is as follows:

```
agilemode=agile-cloud;agilemanage-mode=ip;agilemanage-domain=172.99.0.99;agilemanage-port=10020;
```

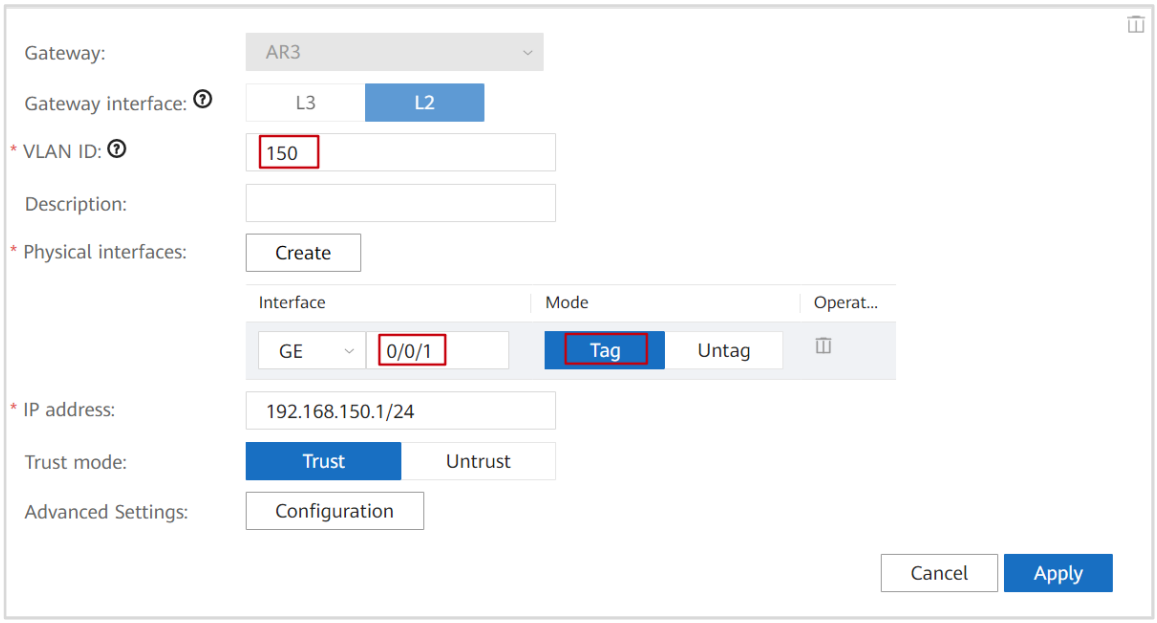
Configure MAC addresses for static binding. For details, see **Pre-configuration for AR3** > **Pre-configuration for the DHCP Server (Used for Device Plug and Play)** in 6 "VXLAN-based Virtualized Campus Network Deployment."




Click **OK**. The interconnection interface configuration is complete.

In this case, the interconnection interface for switches on the fabric network to be onboarded and DHCP service are configured.

# Configure another interconnection interface (used to provide DHCP and E-mail services for the fabric network).



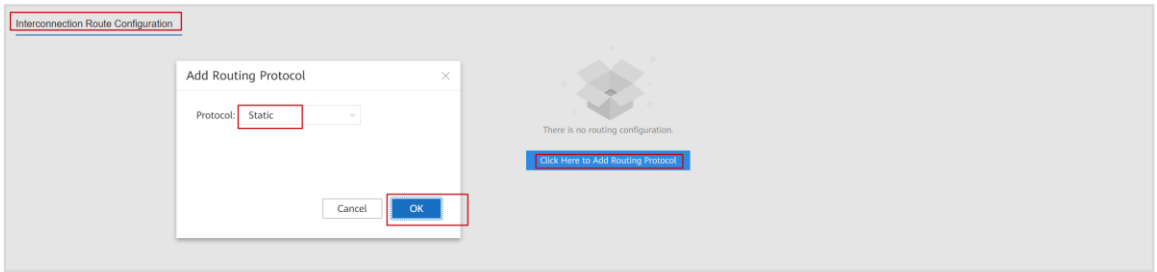

Click **OK**. The interconnection interface configuration is complete.

# Check whether interface configurations are successfully delivered.

```
[AR3]display current-configuration interface GigabitEthernet0/0/1
[V300R019C10SPC300]
#
interface GigabitEthernet0/0/1
 portswitch
 port hybrid pvid vlan 10
 undo port hybrid vlan 1
 port hybrid tagged vlan 150
 port hybrid untagged vlan 10
#
return
```

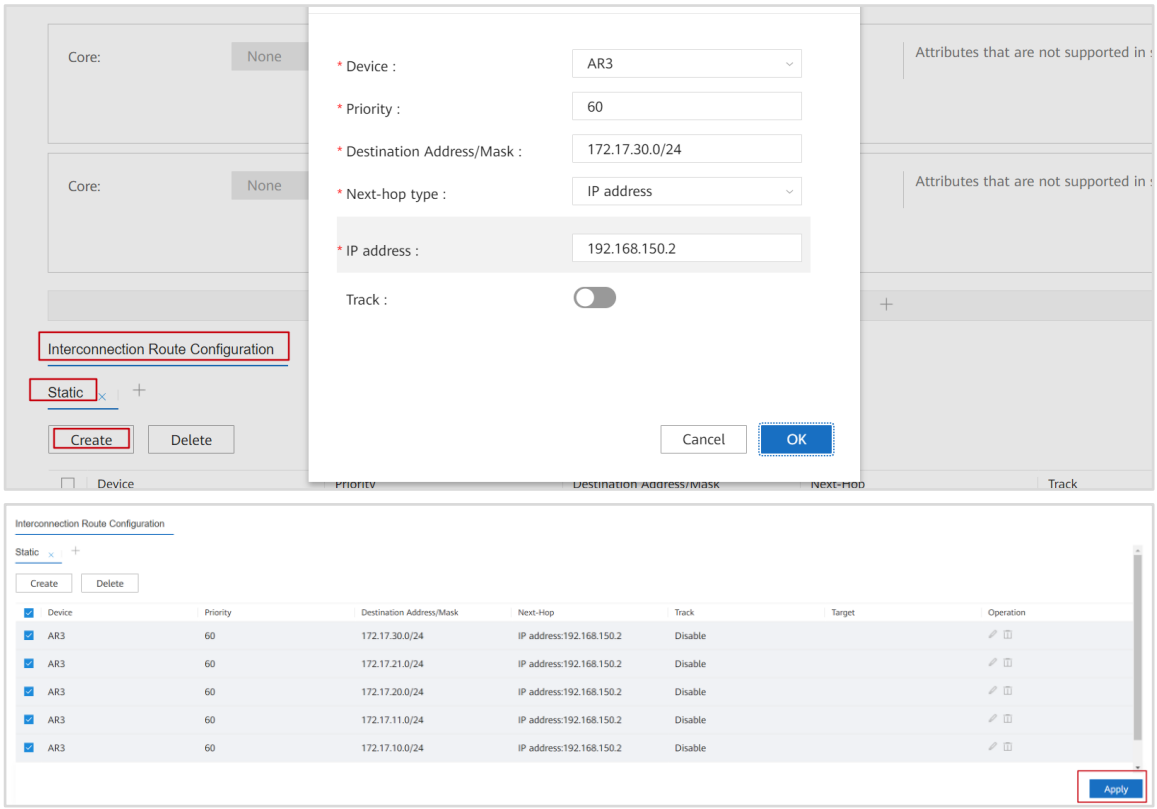
The interconnection VLAN has been delivered.

# Configure interconnection routes.




Configure interconnection routes in the **Interconnection Route Configuration** area.

Configure return routes to the user network segment on the fabric network. To respond to DHCP relay packets whose source IP address is the IP address of the distributed gateway interface from the fabric network, configure these return routes.




Configure return routes to user network segments 172.17.10.0/24, 172.17.11.0/24, 172.17.20.0/24, 172.17.21.0/24, and 172.17.30.0/24 on the fabric network.

Then click **Apply**.

Step 4 Configure a DHCP address pool.

AR3 provides DHCP services for users on the fabric network through an interconnection interface on virtual network **public**, and the interconnection interface is bound to a VPN instance. You need to create a DHCP address pool for allocating IP addresses to terminals and bind the DHCP address pool to the VPN instance.

# Check the VPN instance bound to VLANIF 150 on AR3.

```
[AR3]display current-configuration interface vlanif150
[V300R019C10SPC300]
#
interface Vlanif150
 ip binding vpn-instance vpn1
 tcp adjust-mss 1200
 ip address 192.168.150.1 255.255.255.0
 sa application-statistic enable
#
return
```

The command output shows that the name of the VPN instance corresponding to virtual network **public** on the controller GUI is **vpn1**.

# Create a DHCP address pool.

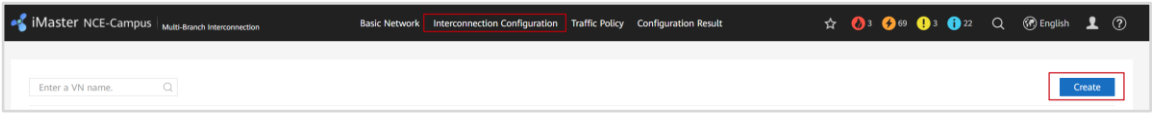
```
[AR3]ip pool Sales_Wired
[AR3-ip-pool-Sales_Wired] vpn-instance vpn1
[AR3-ip-pool-Sales_Wired] gateway-list 172.17.10.254
[AR3-ip-pool-Sales_Wired] network 172.17.10.0 mask 255.255.255.0
[AR3-ip-pool-Sales_Wired] quit
[AR3]ip pool Sales_Wireless
[AR3-ip-pool-Sales_Wireless] vpn-instance vpn1
[AR3-ip-pool-Sales_Wireless] gateway-list 172.17.11.254
[AR3-ip-pool-Sales_Wireless] network 172.17.11.0 mask 255.255.255.0
[AR3-ip-pool-Sales_Wireless] quit
[AR3]ip pool Market_Wired
[AR3-ip-pool-Market_Wired] vpn-instance vpn1
[AR3-ip-pool-Market_Wired] gateway-list 172.17.20.254
[AR3-ip-pool-Market_Wired] network 172.17.20.0 mask 255.255.255.0
[AR3-ip-pool-Market_Wired] quit
[AR3]ip pool Market_Wireless
[AR3-ip-pool-Market_Wireless] vpn-instance vpn1
[AR3-ip-pool-Market_Wireless] gateway-list 172.17.21.254
[AR3-ip-pool-Market_Wireless] network 172.17.21.0 mask 255.255.255.0
[AR3-ip-pool-Market_Wireless] quit
[AR3]ip pool RD
[AR3-ip-pool-RD] vpn-instance vpn1
[AR3-ip-pool-RD] gateway-list 172.17.30.254
[AR3-ip-pool-RD] network 172.17.30.0 mask 255.255.255.0
[AR3-ip-pool-RD] quit
```

## 7.1.2.8 Creating Virtual Network OA

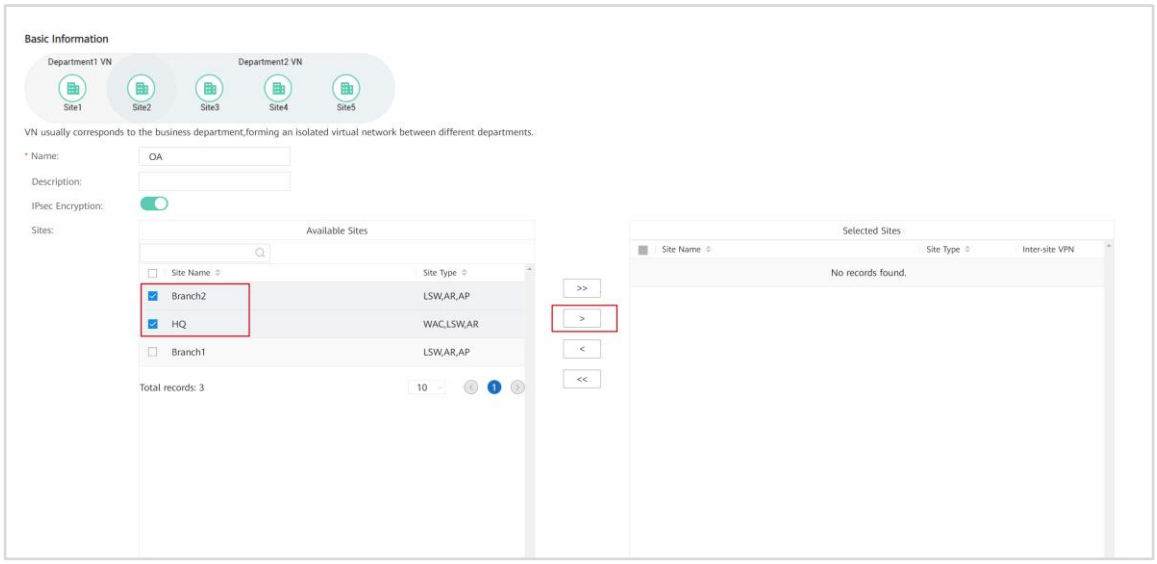
Create virtual network **OA** to interconnect with logical network **OA** on the fabric of the HQ site and connect logical network **OA** on the fabric of the HQ site to OA users at Branch2.

Step 1 Create a virtual network.

# Create virtual network **OA**.




Click **Interconnection Configuration** on the top navigation bar. On the page that is displayed, click **Create** and create a VPN.




The VPN provides services for the HQ site and Branch2. Add the HQ site and Branch2 to virtual network **OA**. Then click **OK**. The VPN configuration is complete.

# Configure virtual network **OA**.

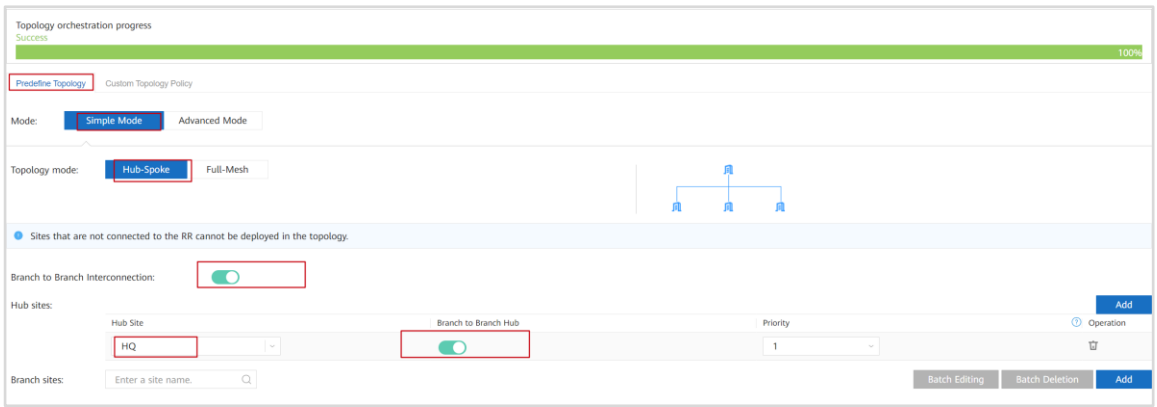

Click the virtual network name to access the virtual network configuration page.

Step 2 Configure WAN services.

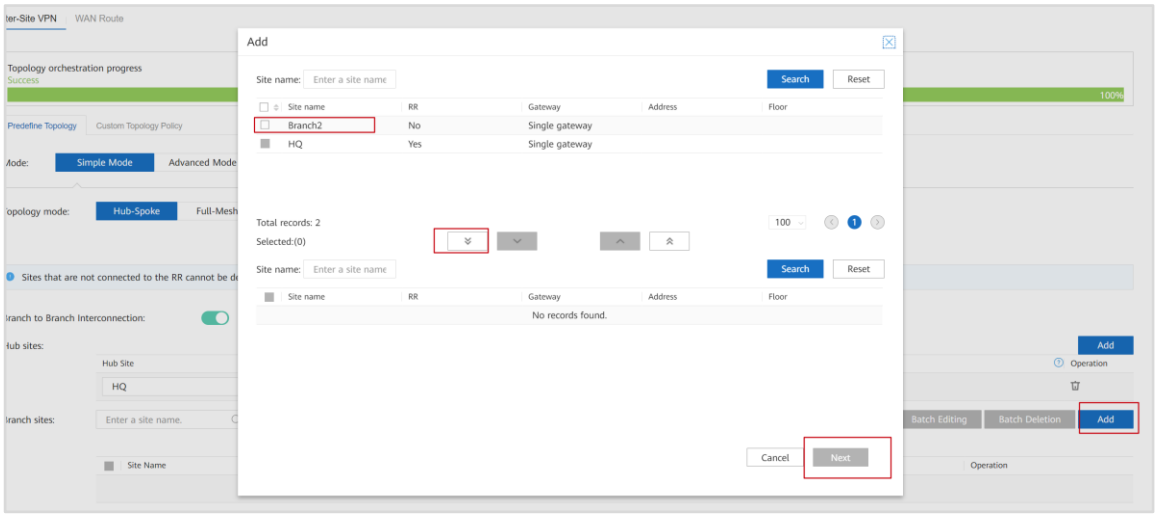
Configure WAN services for virtual network **OA**, set the topology mode to **Hub-Spoke**, and specify the HQ site as the hub site and Branch2 as a branch site.

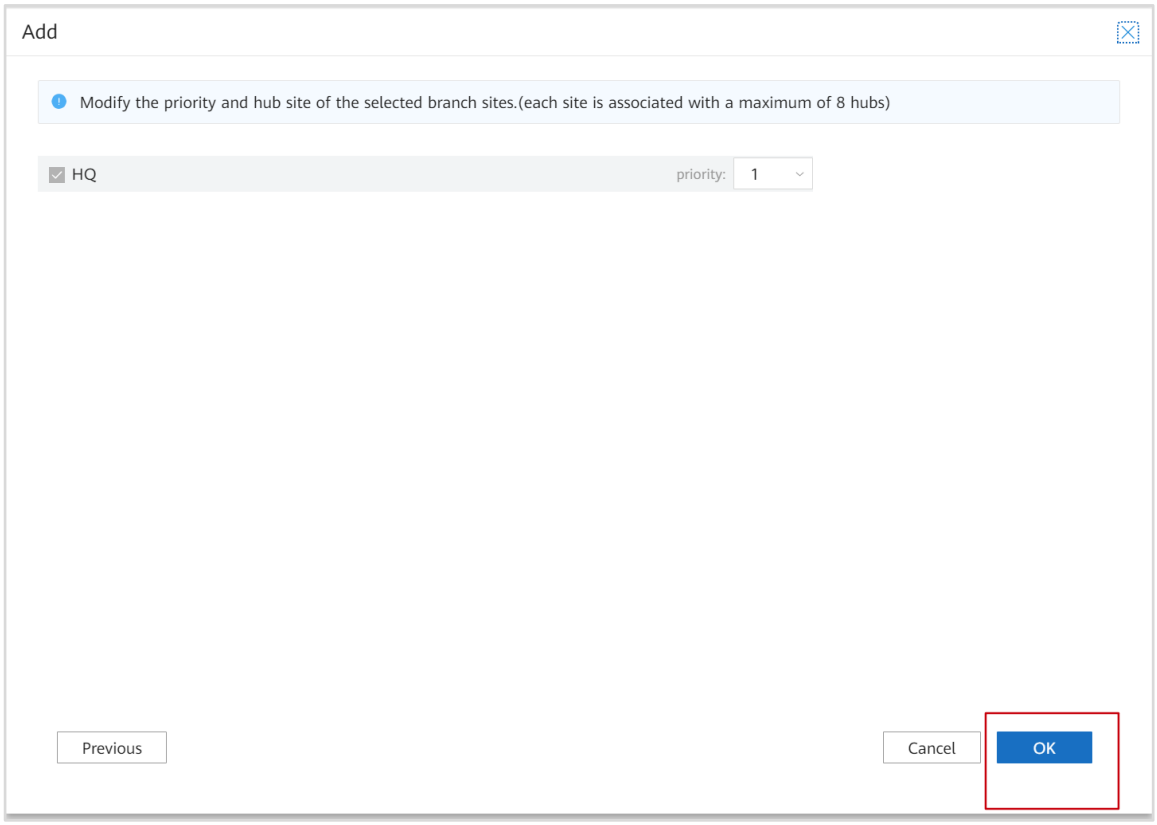
In this lab, you need to configure WAN route filtering to ensure that users at Branch2 can communicate with wired marketing users (172.17.20.0/24) on virtual network **OA** at the HQ.

# Select a hub site.



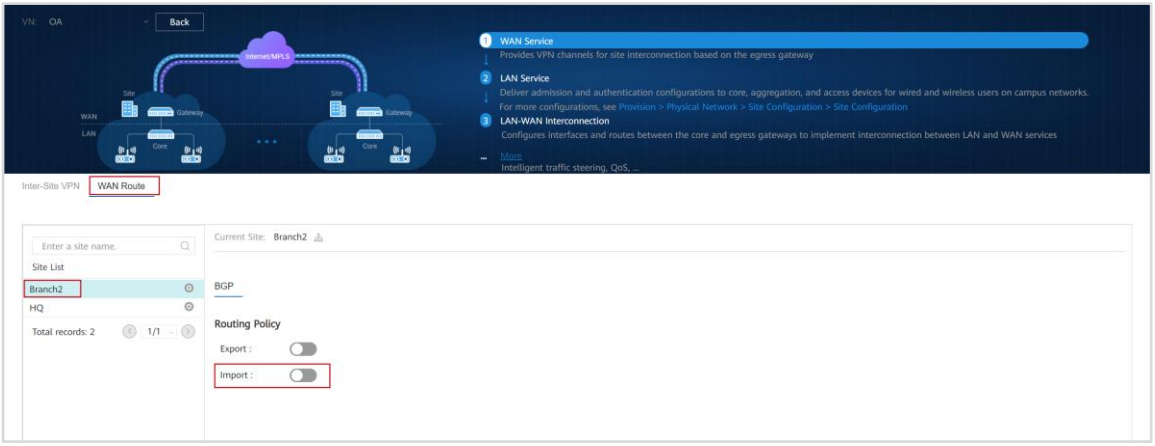

Specify the HQ as the hub site, and click **Add** in the lower right corner to specify **Branch2** as a branch site.

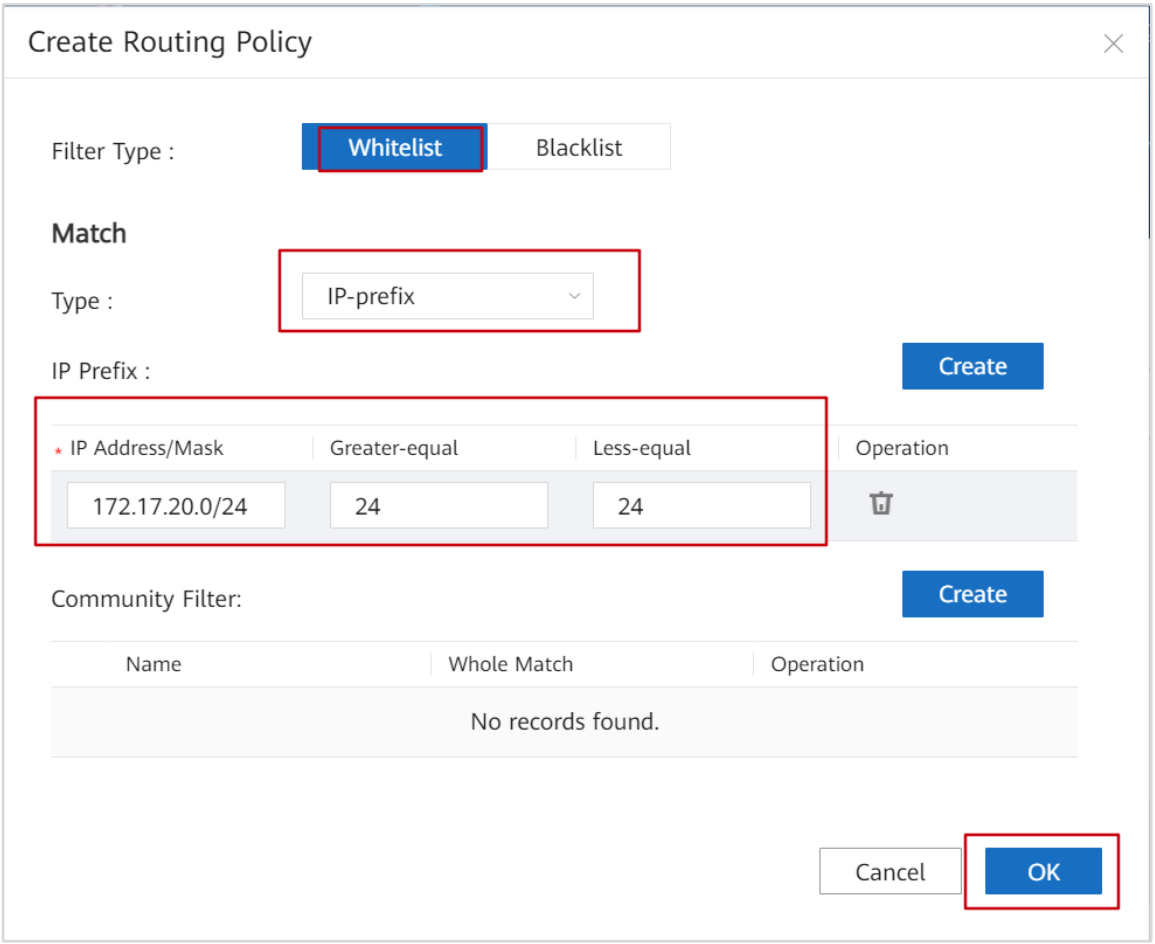



Then click **OK**.

# Configure WAN routing policies.



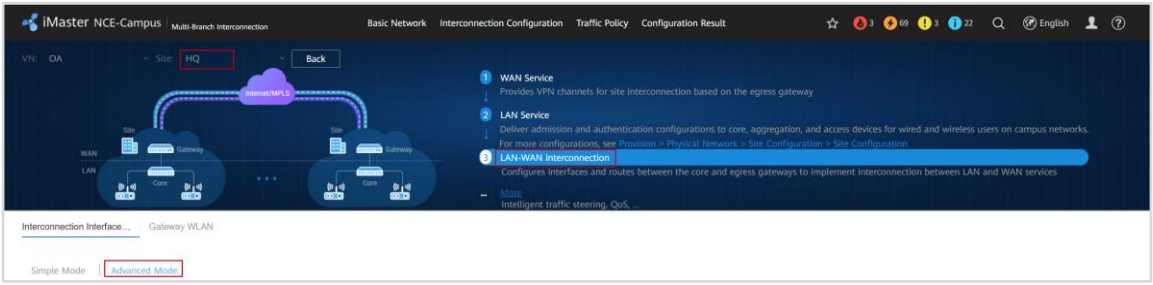

Click **WAN Service** on the virtual network configuration page. On the displayed page, click the **WAN Route** tab, and configure WAN routing policies as shown in the preceding figure.

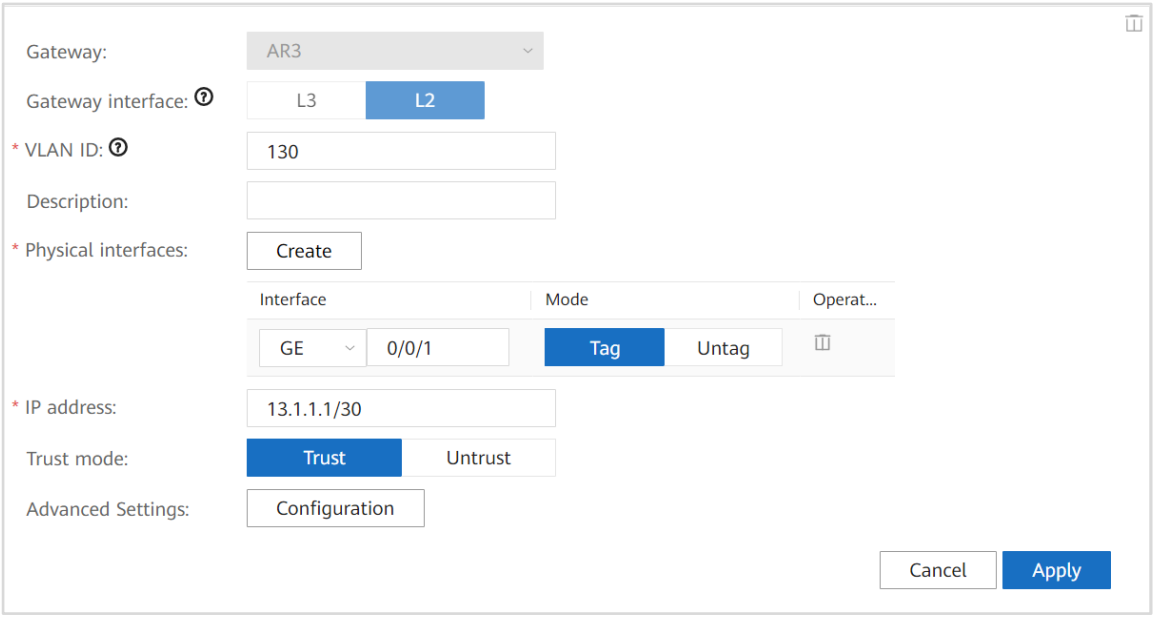
Configure an IP prefix list to match the network segment 172.17.20.0/24 so that Branch2 can learn only routes from wired marketing users at the HQ.

Step 3 Configure LAN-WAN interconnection.

Configure LAN-WAN interconnection as planned to ensure that traffic from PC5 can be transmitted to overlay virtual network **OA**.

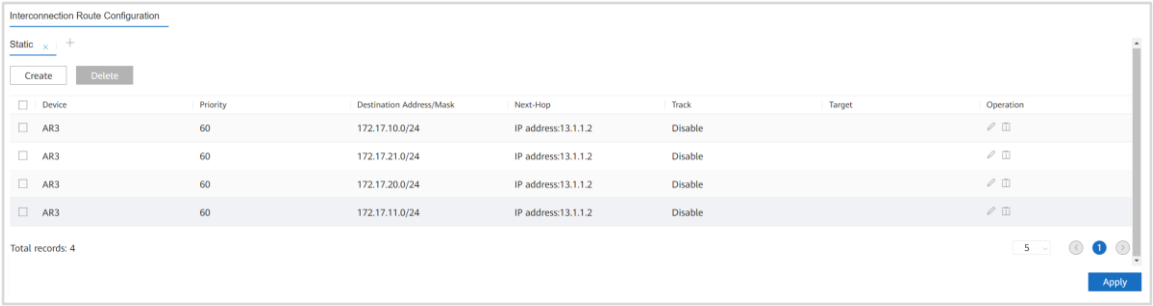
# Configure LAN-WAN interconnection for the HQ.



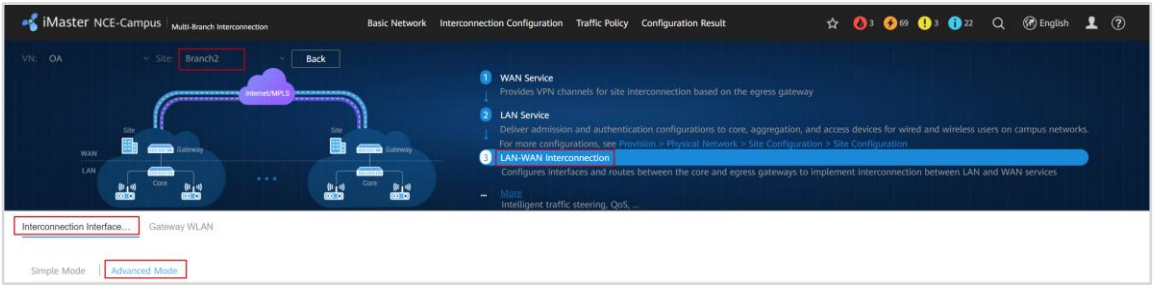

Click **LAN-WAN Interconnection** and configure an interconnection interface in advanced mode.

Configure LAN-WAN interconnection as planned for the HQ to connect to virtual network **OA** on the fabric network. Then click **Apply**.

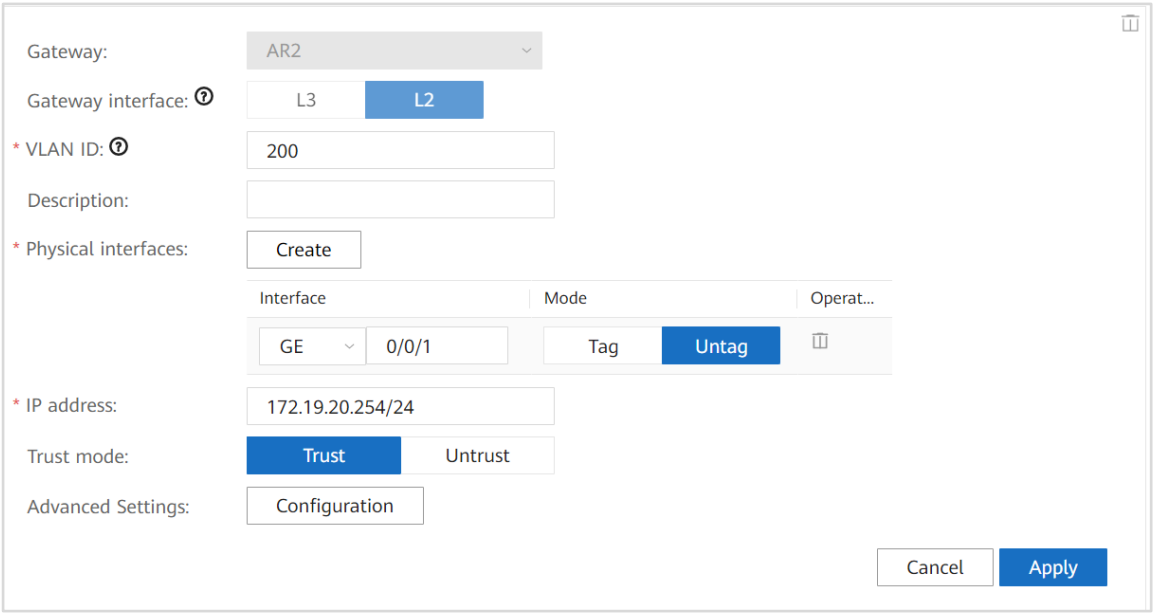



Configure return routes to the virtual network **OA** on the fabric of the HQ as planned. Then click **Apply**.

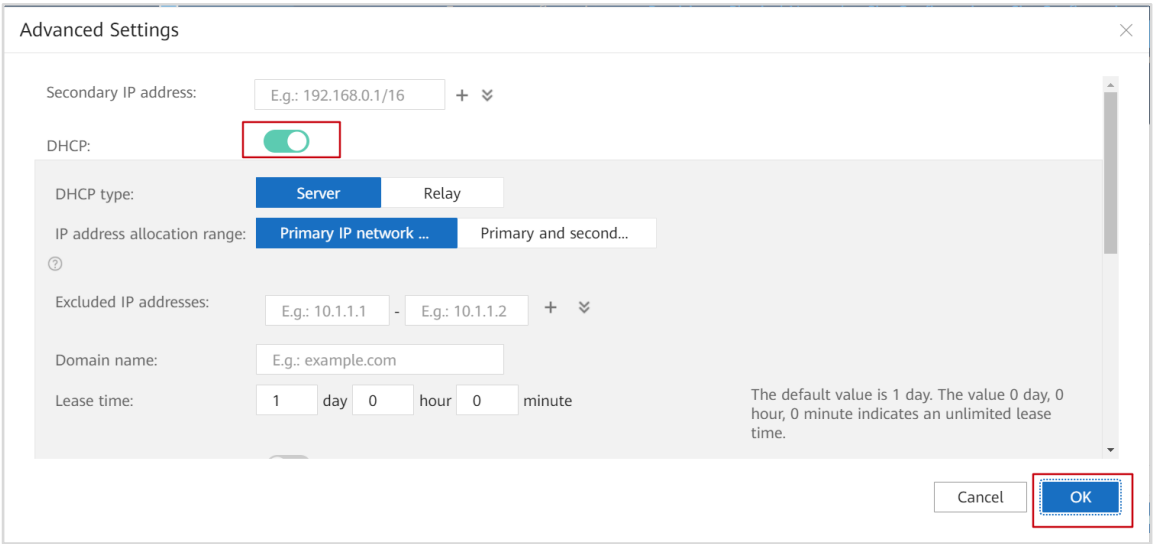
# Configure LAN-WAN interconnection for Branch2.




Switch from the HQ to Branch2.




Configure the LAN-WAN interconnection interface as planned and enable DHCP in the **Advanced Settings** area.

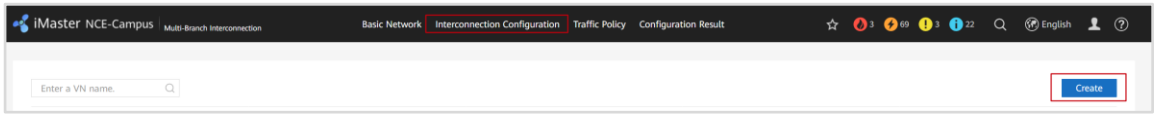



Allocate an IP address to PC5 through DHCP on the interface.

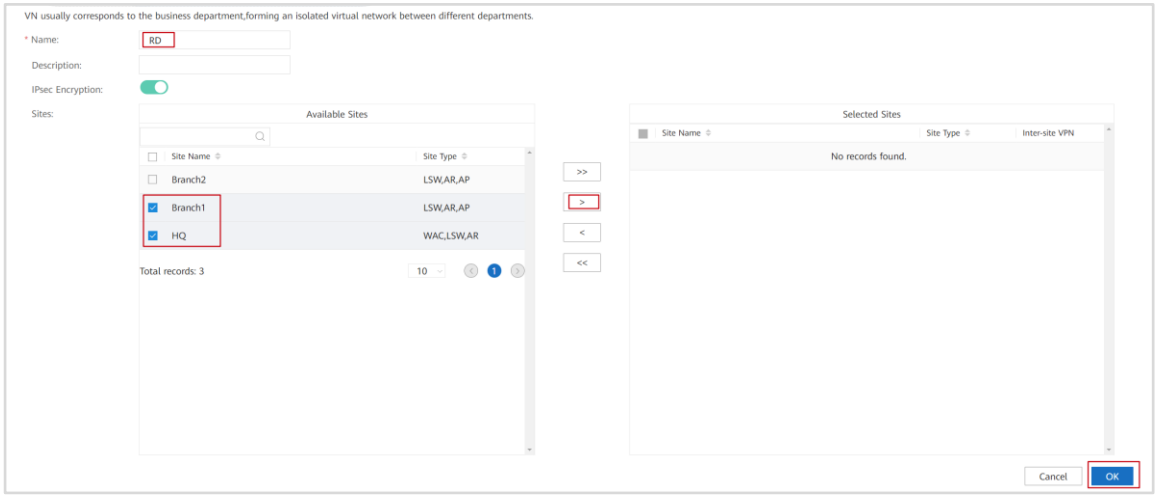
## 7.1.2.9 Creating Virtual Network RD

Create virtual network **RD** to interconnect with logical network **RD** on the fabric of the HQ site and connect logical network **RD** on the fabric of the HQ site to RD users at Branch1.

Step 1 Create a virtual network.




Click **Interconnection Configuration** on the top navigation bar. On the page that is displayed, click **Create** and create a VPN.

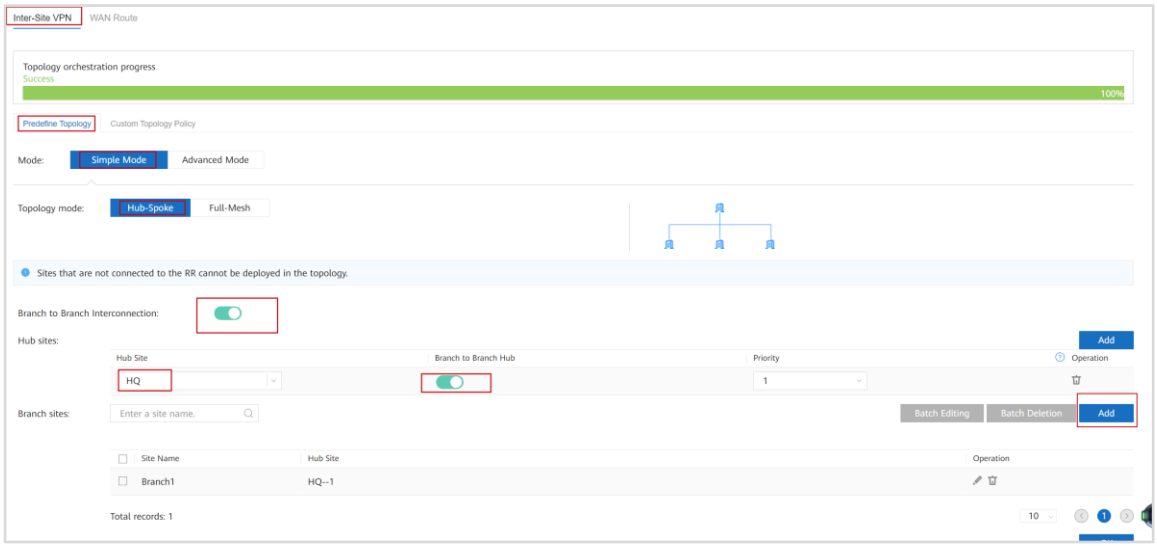



The VPN provides services for the HQ site and Branch1. Add the HQ site and Branch1 to virtual network **RD**. Then click **OK**.

Step 2 Configure WAN services.

Configure WAN services for virtual network **RD**, set the topology mode to **Hub-Spoke**, and specify the HQ site as the hub site and Branch1 as a branch site.

# Configure WAN services.

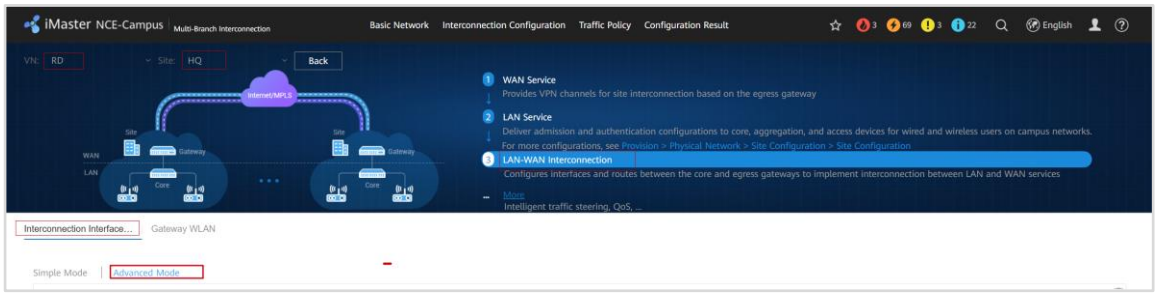


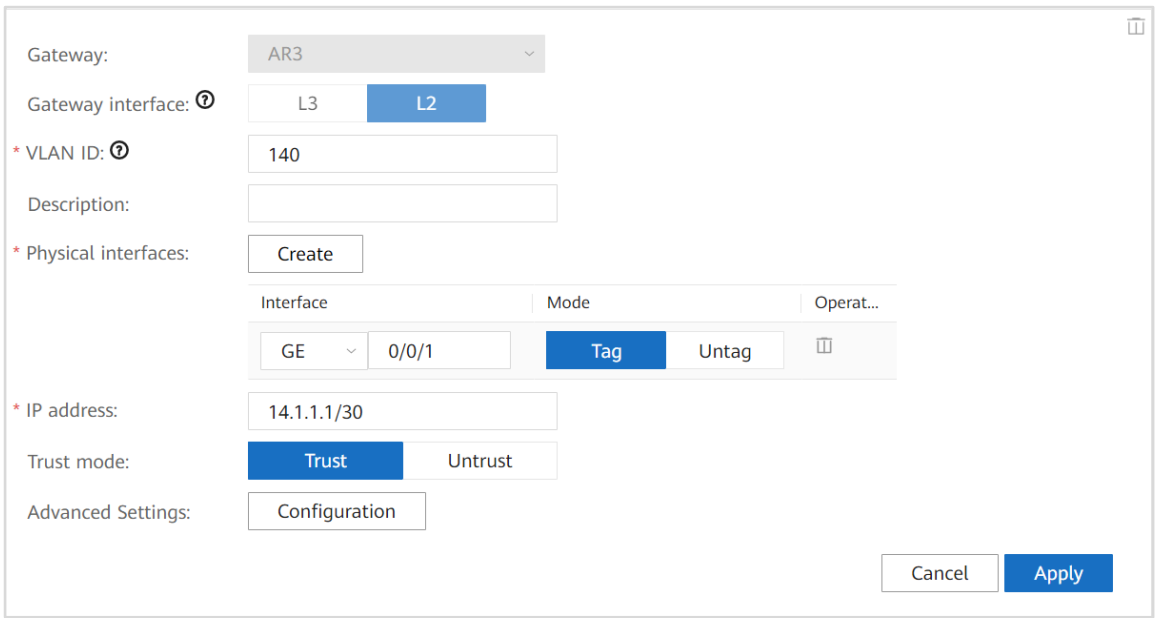

Click **WAN Service**, and specify the HQ as the hub site and Branch1 as the branch site.

## Step 3 Configure LAN-WAN interconnection.

Configure LAN-WAN interconnection as planned to ensure that traffic from PC4 can be transmitted to overlay VN **RD**.

# Configure LAN-WAN interconnection for the HQ.



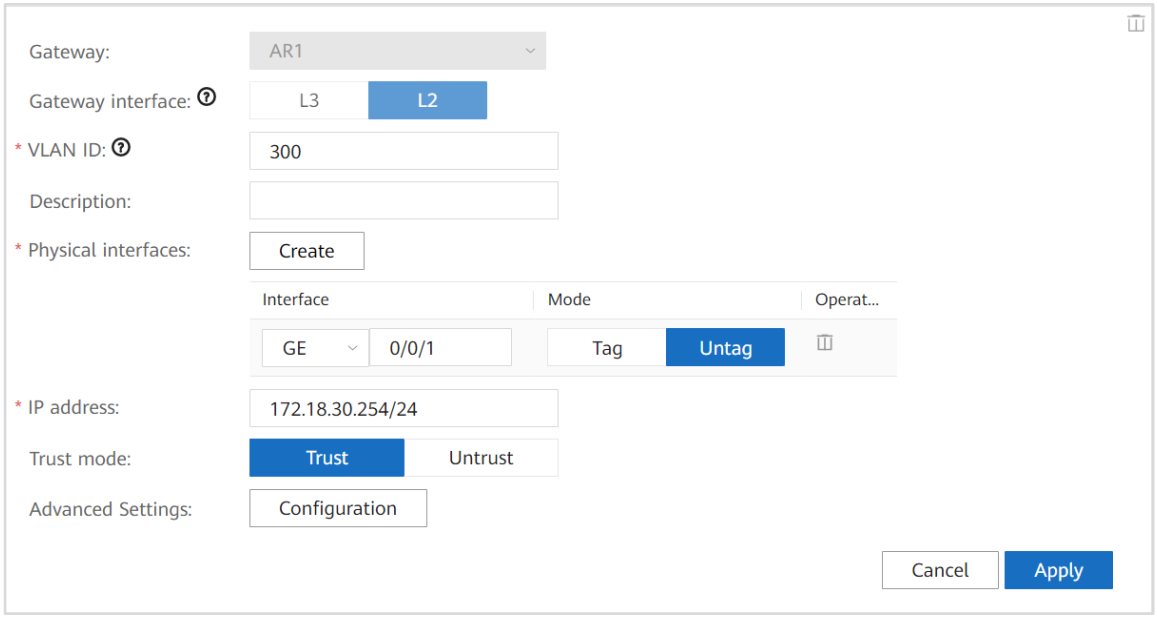



Configure LAN-WAN interconnection as planned for the HQ to connect to virtual network **RD** on the fabric network. Then click **Apply**.

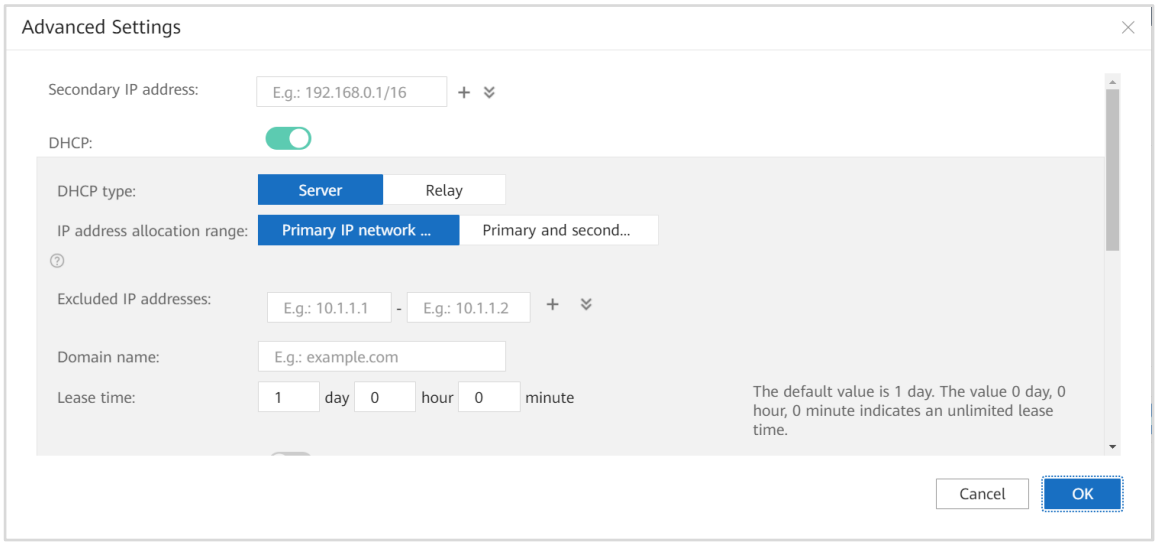

Configure return routes to the virtual network **RD** on the fabric of the HQ as planned. Then click **Apply**.

# Configure LAN-WAN interconnection for Branch1.


Switch from the HQ to Branch1.




Configure the LAN-WAN interconnection interface as planned and enable DHCP in the **Advanced Settings** area.




Allocate an IP address to PC4 through DHCP on the interface. Then click **Apply**.

## 7.1.2.10 Configuring Site-to-Internet Access and Fabric-related Functions

After local Internet access is configured, users can access the Internet through the local site, and switches at the HQ site can register with iMaster NCE for onboarding.

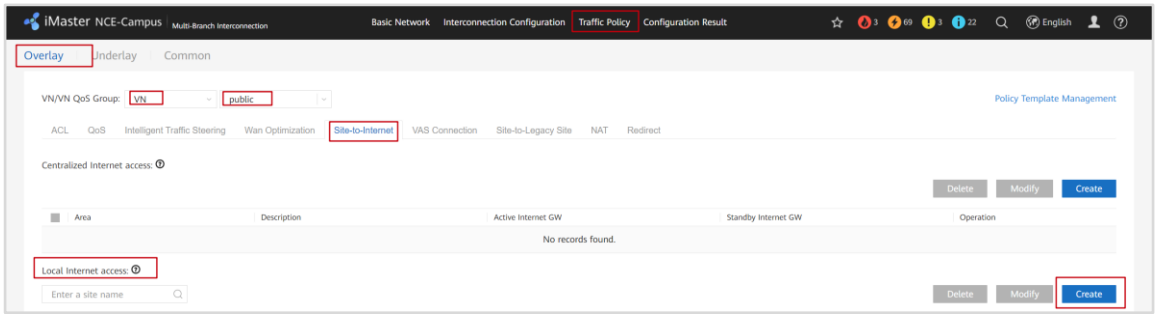
The loopback interface simulating an E-mail server has been created in the AR3 pre-configuration in 6 "VXLAN-based Virtualized Campus Network Deployment". In the SD-WAN scenario, you need to configure the loopback interface in virtual network **public**.

To implement Portal authentication for the fabric of the HQ site, configure NAT mapping for the underlay network.

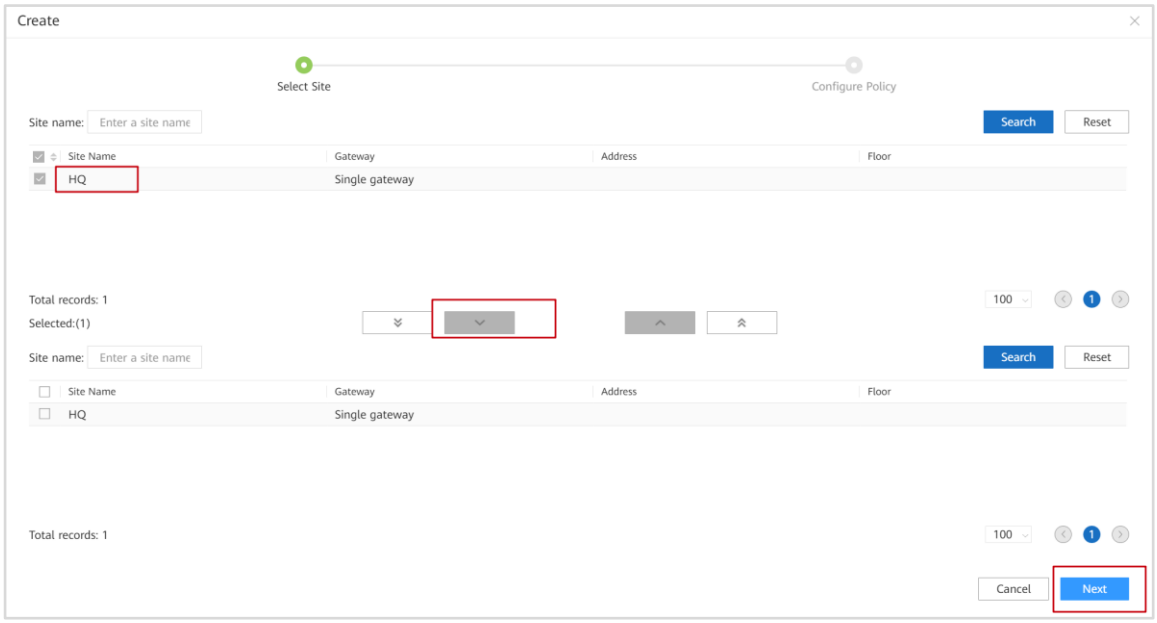
Step 1 Configure site-to-Internet services.

Configure site-to-Internet services for the HQ, Branch1, and Branch2 to access the Internet through the local Internet link.

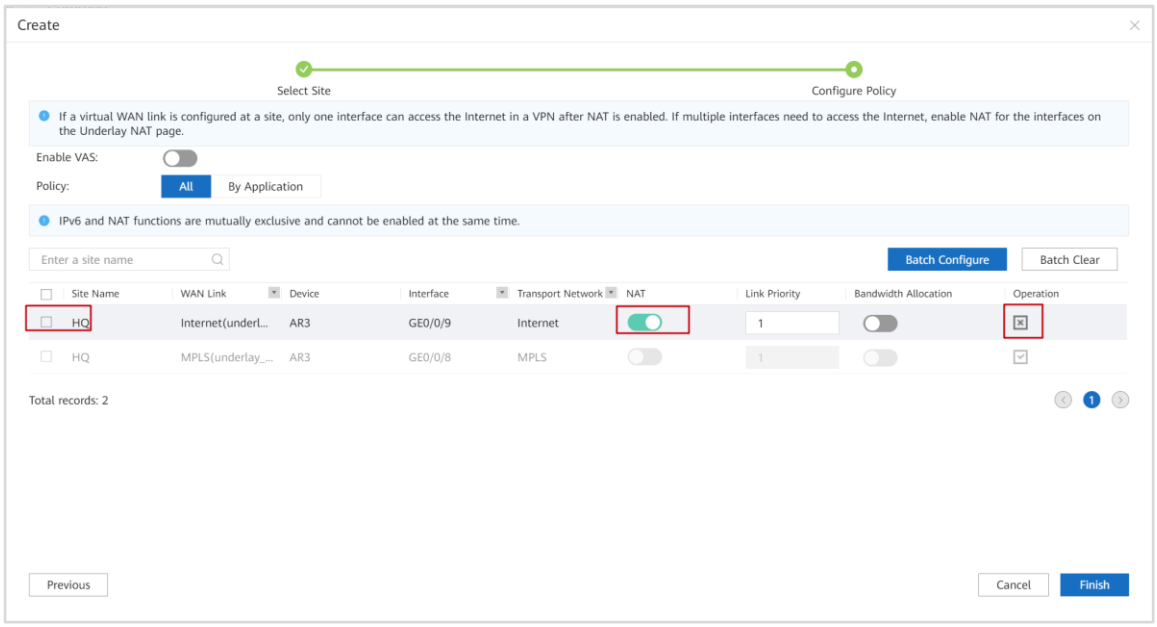
# Configure site-to-Internet services for virtual network **public**.




Choose **Multi-Branch Interconnection** > **Traffic Policy** > **Overlay** from the main menu, select virtual network **public**, click the **Site-to-Internet** tab, and click **Create** in the **Local Internet access** area.

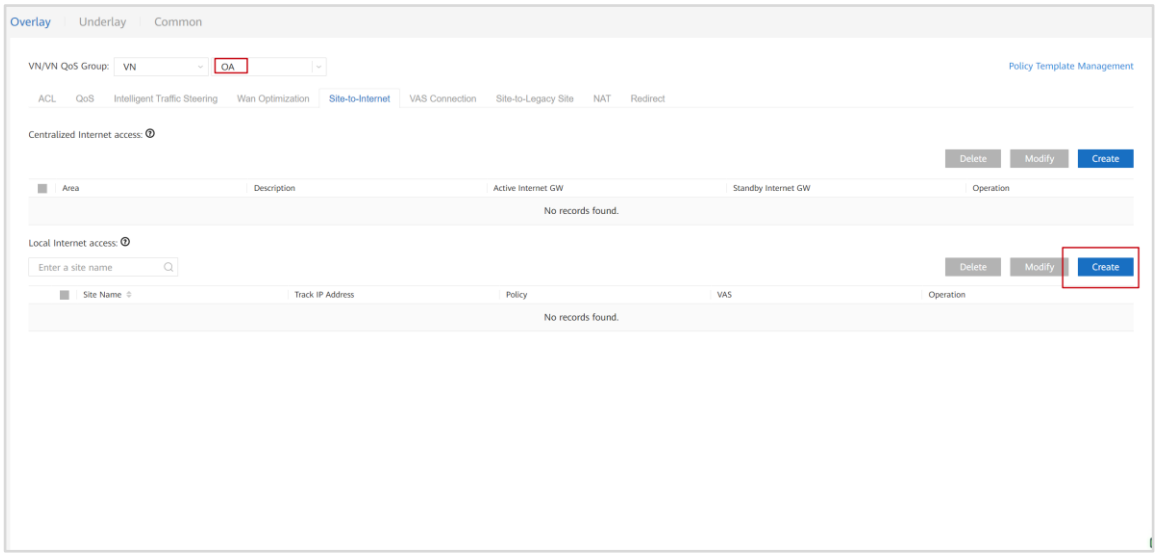


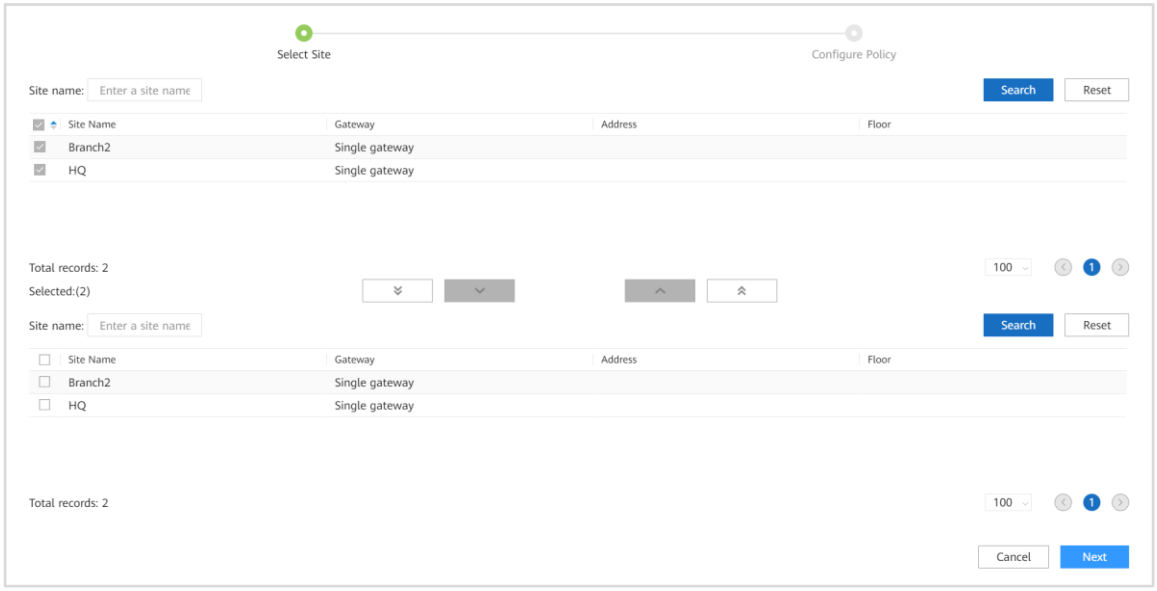

Select **HQ**.



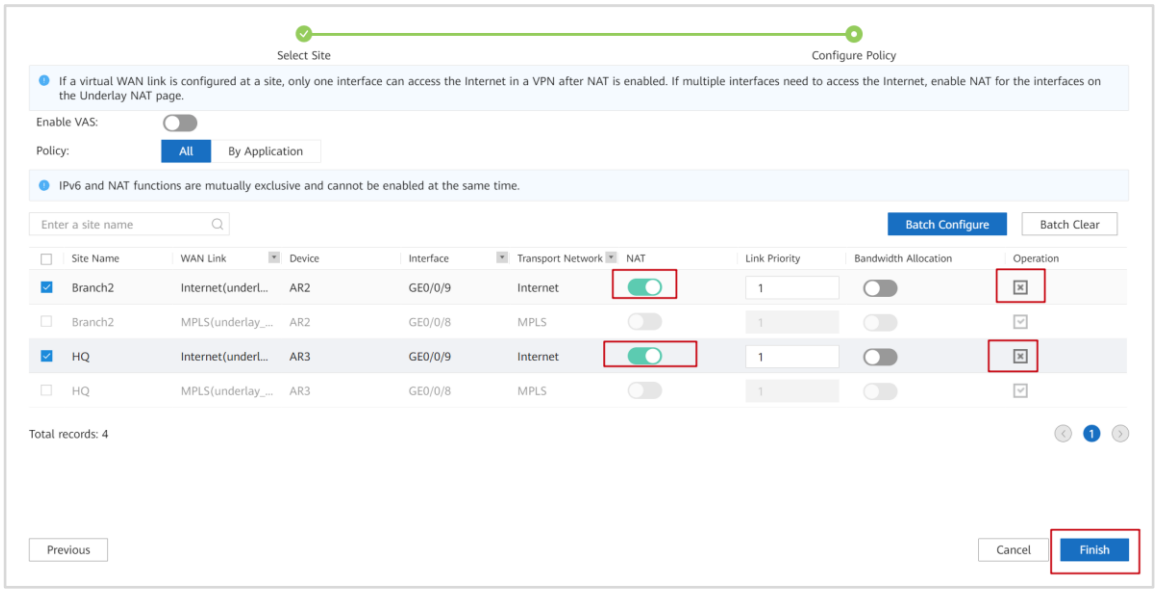

Configure local Internet access through the Internet link and enable NAT.

# Configure site-to-Internet services for virtual network **OA**.



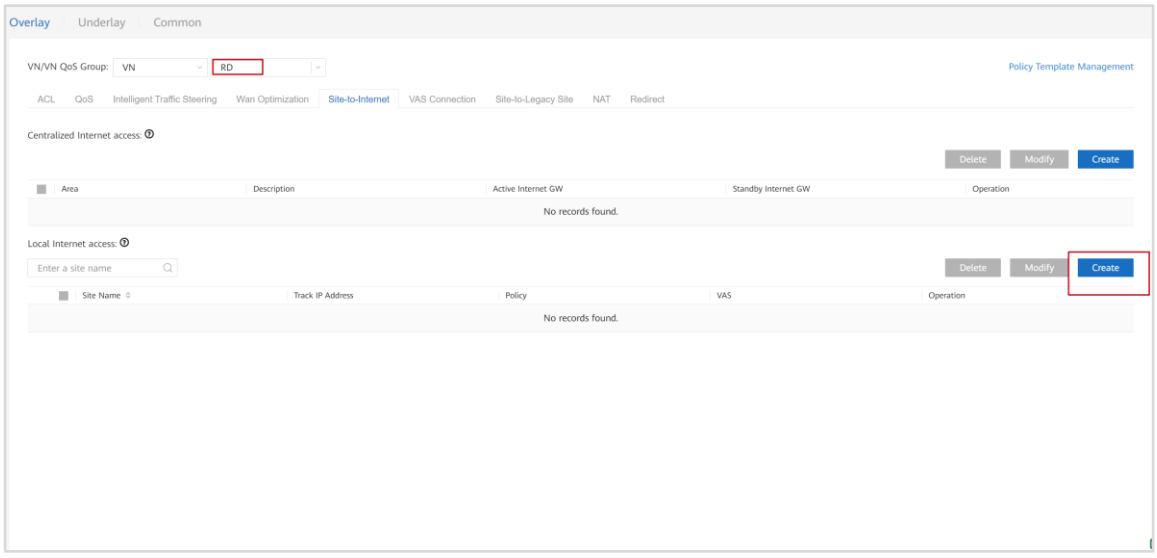



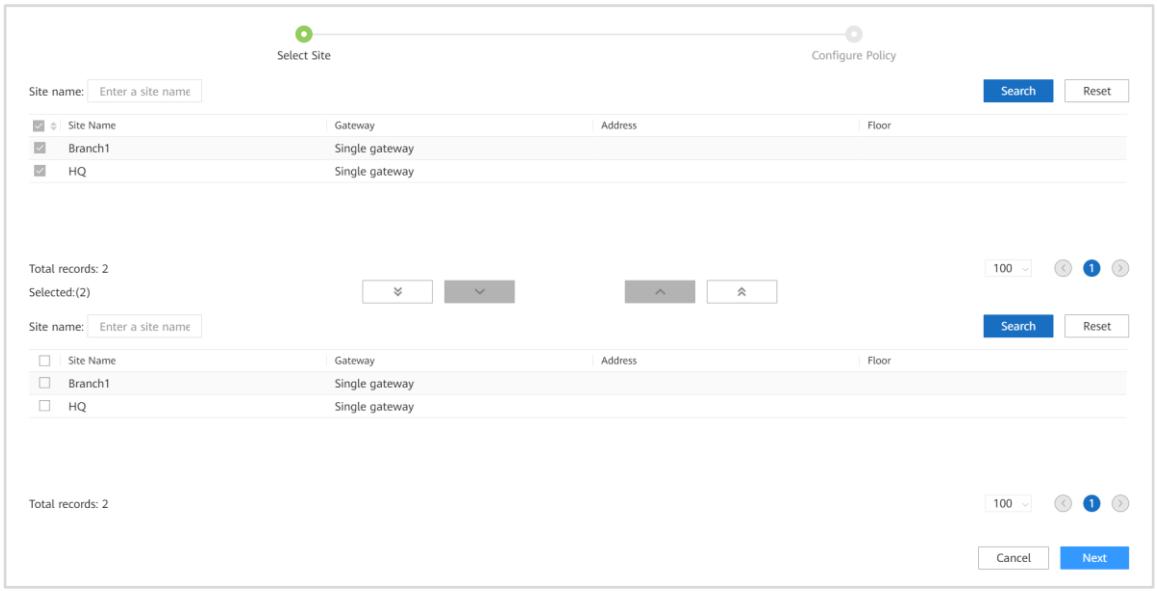

Select **HQ** and **Branch2**.



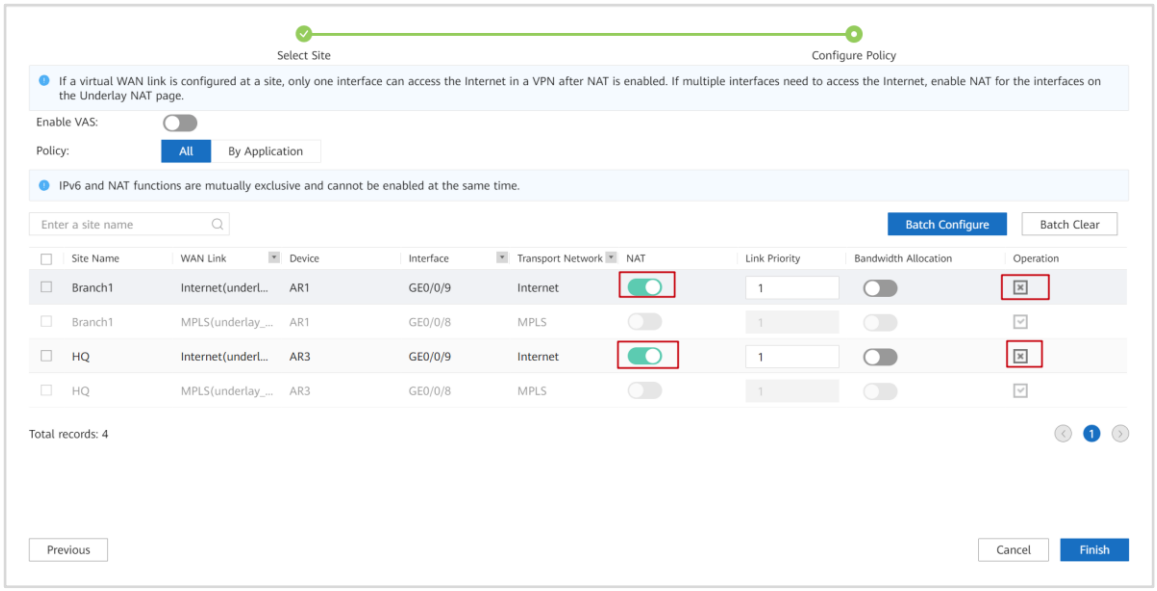

Configure site-to-Internet access through the Internet link.

# Configure site-to-Internet services for virtual network **RD**.



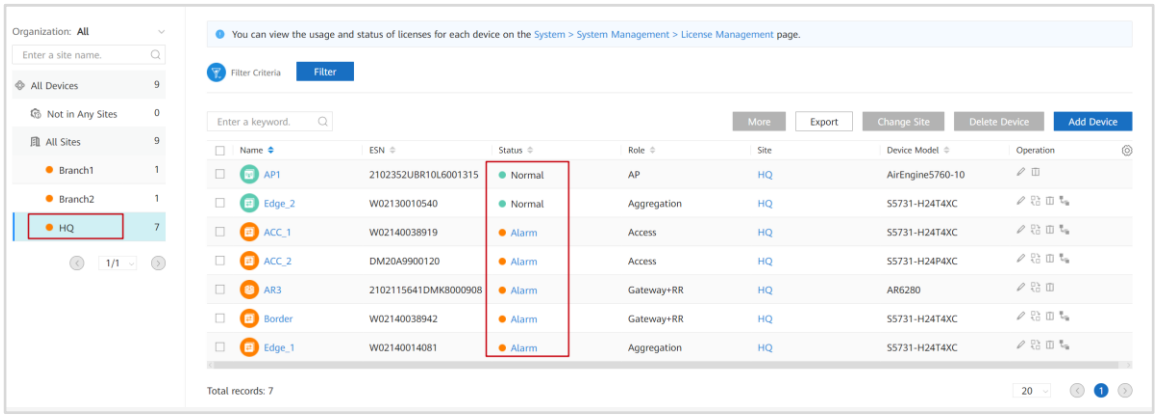



Select **HQ** and **Branch1**.




Configure local Internet access through the Internet link.

# Check whether devices at the HQ site are online.



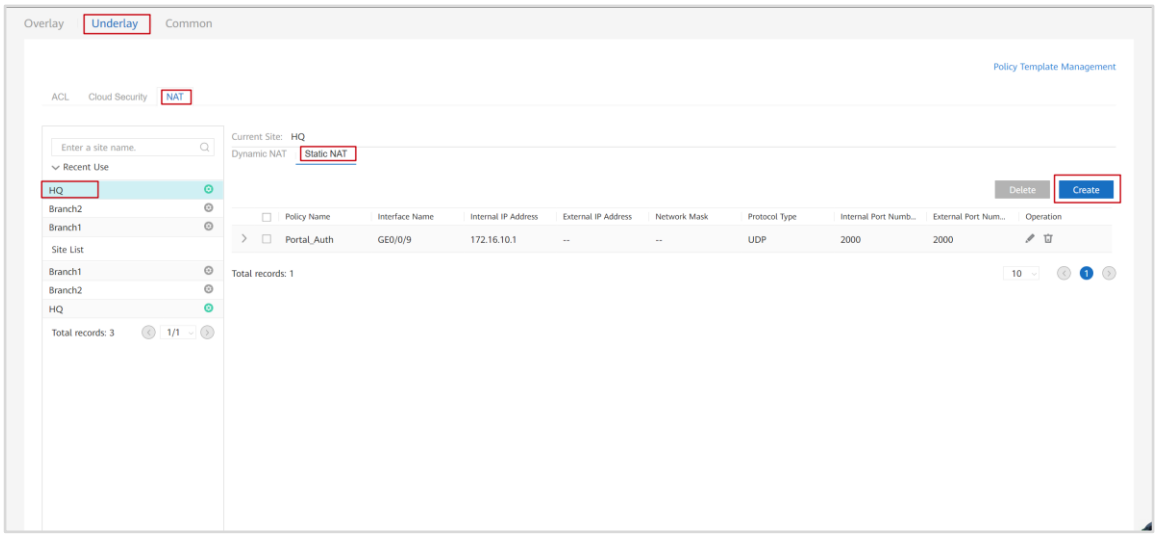
Choose **Design** > **Site Agile Deployment** > **Device Management** from the main menu and view the online status of devices at the HQ site.

In this case, all devices at the HQ site can access the Internet locally through virtual network **public**, and communicate with iMaster NCE through source NAT, implementing device registration and onboarding.

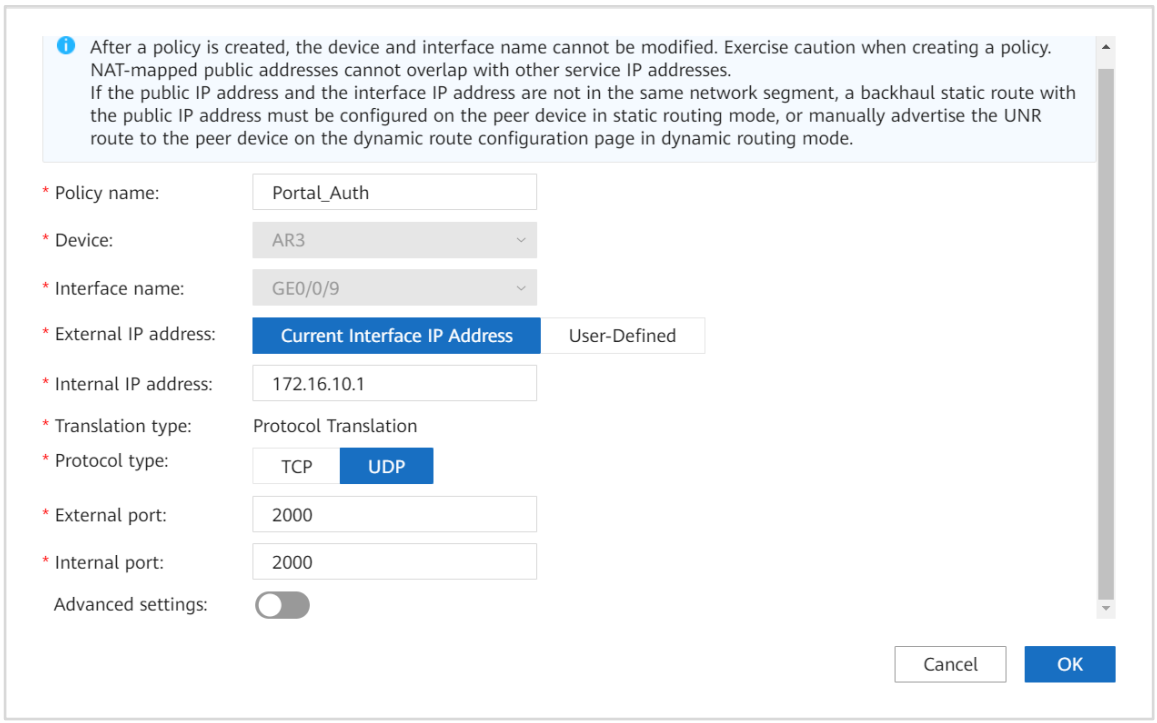
Step 2 Configure NAT.

To enable iMaster NCE to proactively communicate with Border at the HQ site through Portal, configure static NAT to map UDP port 2000 of Border to an external port.

# Configure NAT.




Choose **Multi-Branch Interconnection** > **Traffic Policy** > **Underlay** from the main menu, click the **NAT** tab, select **HQ**, and configure static NAT.



Map UDP port 2000 of Border to the IP address of GE0/0/9 on AR3. In LAN-WAN interconnection, static DHCP binding is configured for subsequent management and address mapping.

Step 3 Configure an E-mail server.

Create Loopback1 used for simulating an E-mail server and bind Loopback1 to VPN instance **vpn1**.

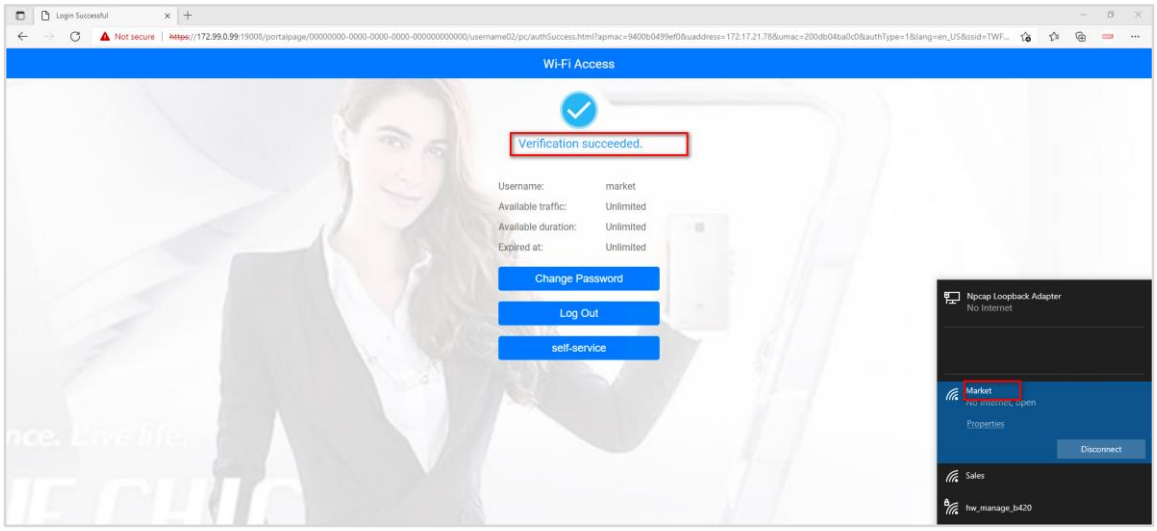
# Create Loopback1.

```
[AR3]interface LoopBack1
[AR3-LoopBack1] ip binding vpn-instance vpn1
[AR3-LoopBack1] ip address 172.17.3.3 32
[AR3-LoopBack1] quit
```

Step 4 Verify configurations.

Check whether users on the fabric network can obtain IP addresses and be authenticated. The verification method is the same as that in 6 "VXLAN-based Virtualized Campus Network Deployment", and only Portal authentication configuration is verified here.

# Perform Portal authentication on PC3.




Connect PC3 to SSID **Market** for Portal authentication.

Step 5 Perform a connectivity test.

Test the connectivity between terminals at Branch1 and Branch2 and the HQ site, respectively.

# Check the IP address of PC4.

```
C:\Users\PC4>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet1:

   Connection-specific DNS Suffix . . . . . :
```

```
   Link-local IPv6 Address . . . . . : fe80::6d76:b5c8:387c:7a14%12
   IPv4 Address . . . . . : 172.18.30.27
   Subnet Mask  . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . . . . : 172.18.30.254
```

# Check the IP address of PC5.

```
C:\Users\PC5>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet1:

   Connection-specific DNS Suffix . . . . . :
   Link-local IPv6 Address . . . . . : fe80::6d76:b5c8:387c:7a14%12
   IPv4 Address . . . . . : 172.19.20.234
   Subnet Mask  . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . . . . : 172.18.30.254
```

# Log in to PC1 using the account of the RD department and check the IP address of PC1.

```
C:\Users\PC1>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix . . . . . :
   Link-local IPv6 Address . . . . . : fe80::a46b:62f8:a696:b974%12
   IPv4 Address . . . . . : 172.17.30.225
   Subnet Mask  . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . . . . : 172.17.30.254
```

# Verify the connectivity between PC4 and PC1.

```
C:\Users\PC4>ping 172.17.30.225

Pinging 172.17.30.225 with 32 bytes of data:
Reply from 172.17.30.225: bytes=32 time=1ms TTL=124
Reply from 172.17.30.225: bytes=32 time=1ms TTL=124
Reply from 172.17.30.225: bytes=32 time=1ms TTL=124
Reply from 172.17.30.225: bytes=32 time=1ms TTL=124

Ping statistics for 172.17.30.225:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss);
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

The command output shows that PC4 can properly communicate with PC1.

# Log in to PC2 using the account of the marketing department and check the IP address of PC2.

```
C:\Users\PC2>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix . . . . . . . :
   Link-local IPv6 Address . . . . . . . . : fe80::a46b:62f8:a696:b974%12
   IPv4 Address . . . . . . . . . . . . . . : 172.17.20.167
   Subnet Mask    . . . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . . . .    : 172.17.30.254
```

# Verify the connectivity between PC5 and PC2.

```
C:\Users\PC5>ping 172.17.20.167

Pinging 172.17.20.167 with 32 bytes of data:
Reply from 172.17.20.167: bytes=32 time=11ms TTL=124
Reply from 172.17.20.167: bytes=32 time=1ms TTL=124
Reply from 172.17.20.167: bytes=32 time=1ms TTL=124
Reply from 172.17.20.167: bytes=32 time=2ms TTL=124

Ping statistics for 172.17.20.167:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss);
Approximate round trip times in milli-seconds:
...Minimum = 1ms, Maximum = 11ms, Average = 3ms
```

The command output shows that PC5 can properly communicate with PC2.

# Connect PC3 to SSID **Market** and check the IP address of PC3 after authentication.

```
C:\Users\PC3>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix . . . . . . . :
   Link-local IPv6 Address . . . . . . . . : fe80::a46b:62f8:a696:b974%12
   IPv4 Address . . . . . . . . . . . . . . : 172.17.21.81
   Subnet Mask    . . . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . . . .    : 172.17.30.254
```

# Verify the connectivity between PC5 and PC1.

```
C:\Users\PC5>ping 172.17.21.81

Pinging 172.17.21.81 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

```
Ping statistics for 172.17.21.81:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

The command output shows that PC5 cannot communicate with PC1.

# Log in to PC2 using the account of the sales department and check the IP address of PC2.

```
C:\Users\PC2>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix . . . . . :
   Link-local IPv6 Address . . . . . : fe80::a46b:62f8:a696:b974%12
   IPv4 Address . . . . . . . . . . . : 172.17.10.198
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . . . . : 172.17.30.254
```

# Verify the connectivity between PC5 and PC2.

```
C:\Users\PC5>ping 172.17.10.198

Pinging 172.17.10.198 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 172.17.10.198:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

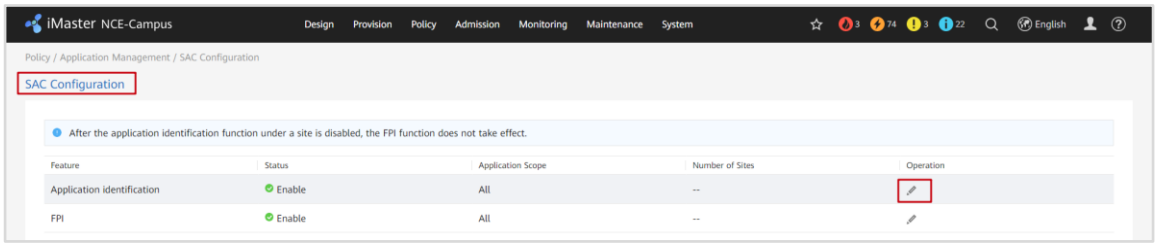
The command output shows that PC5 cannot communicate with PC2.

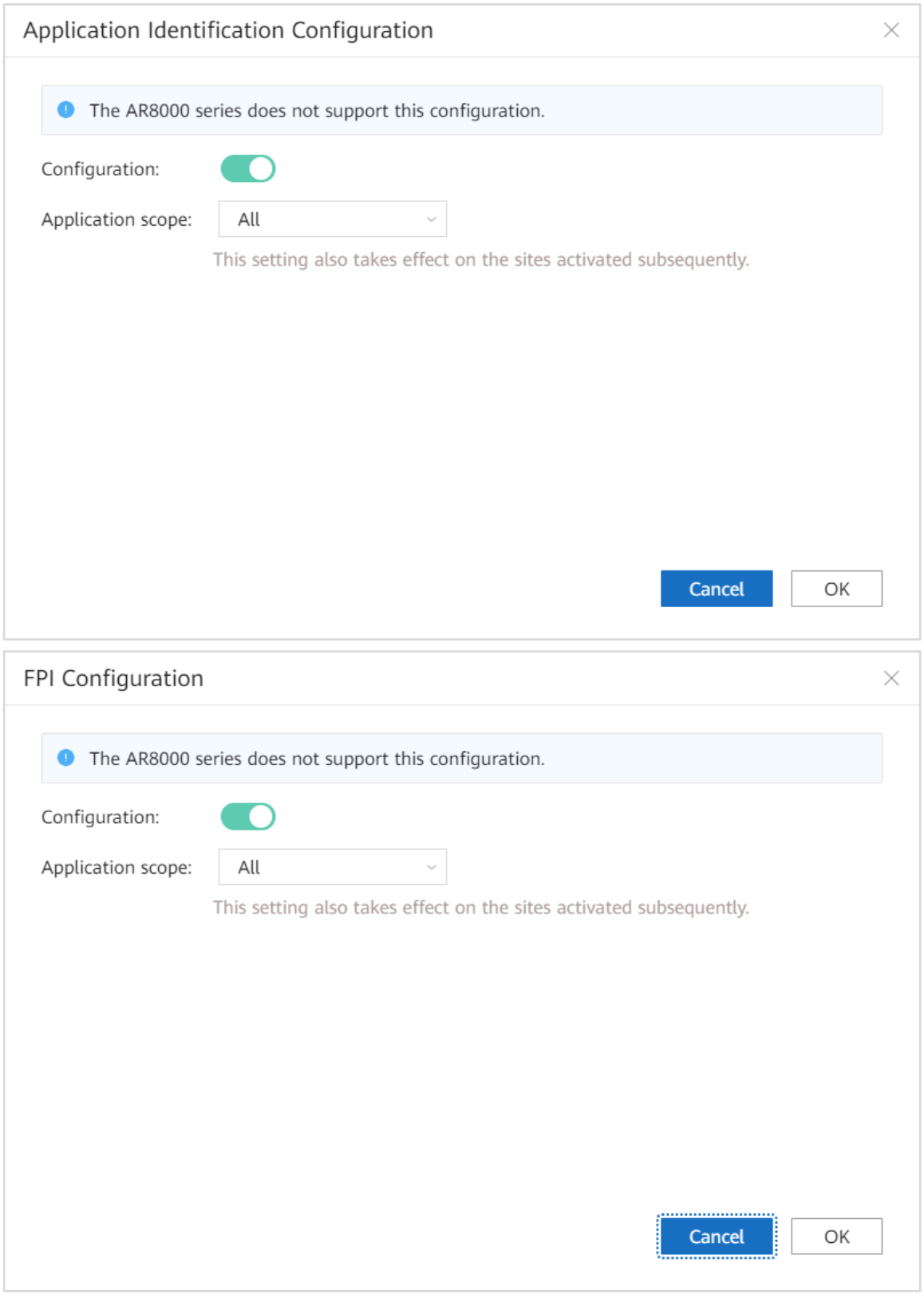
## 7.1.2.11 Configuring Application Identification and Intelligent Traffic Steering

Step 1 Enable application identification and data reporting.

Enable application identification on the controller and enable the data reporting function on AR routers.

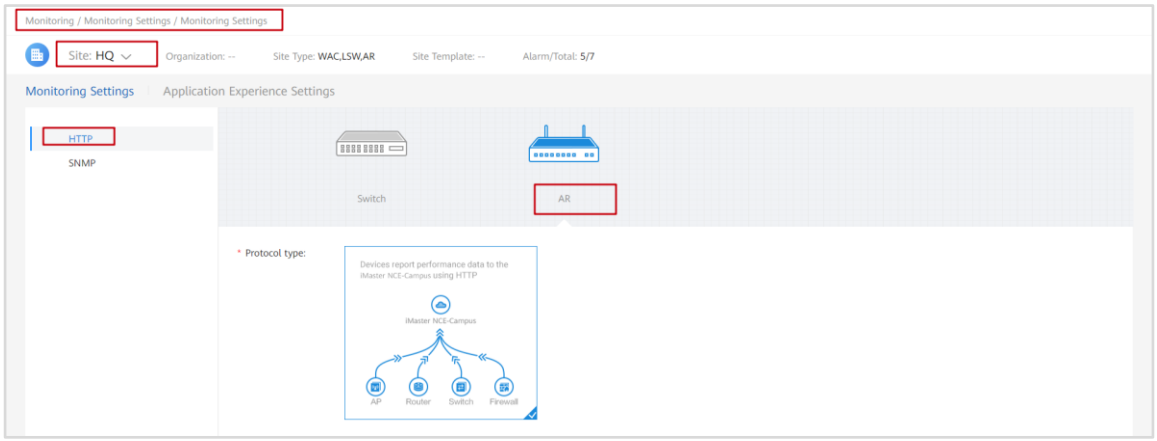
# Enable application identification.

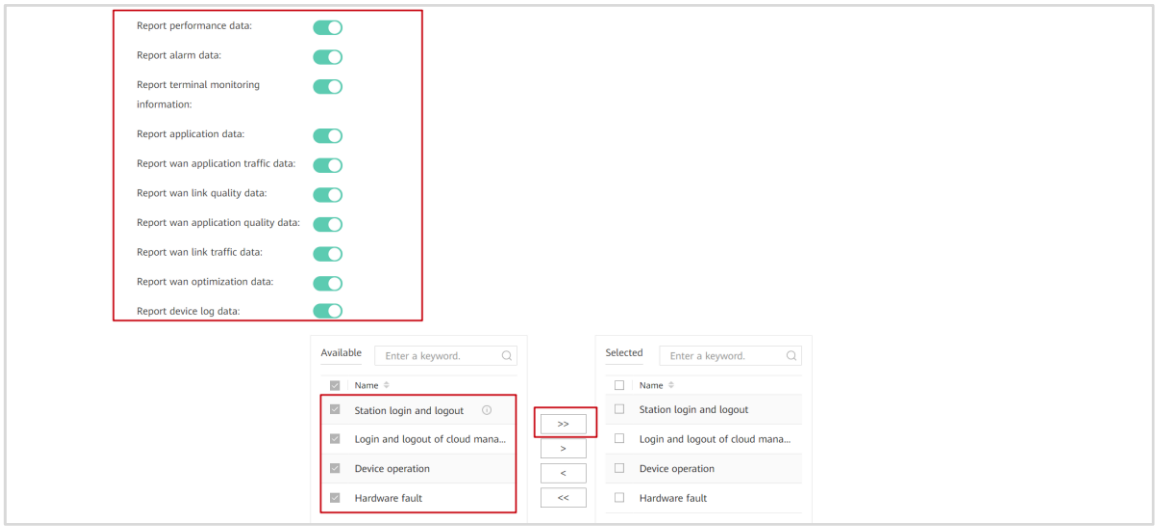





Choose **Policy** > **Application Management** > **SAC Configuration** from the main menu, and enable application identification and FPI.

# Configure AR routers to report monitoring data.




The HQ site is used as an example. Choose **Monitoring** > **Monitoring Settings** > **Monitoring Settings** from the main menu, select the HQ site and then **Router**. Enable the functions of reporting all information, and select **Station login and logout**, **Login and logout of cloud managed device**, **Device operation**, and **Hardware fault**.

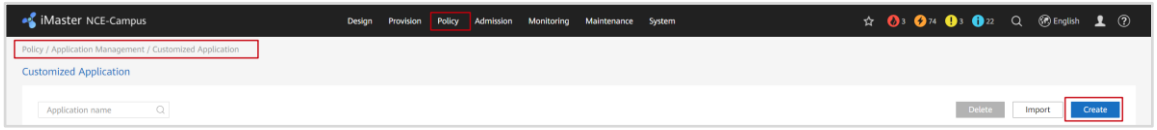
Perform the same operations for Branch1 and Branch2.

Step 2 Configure application groups.

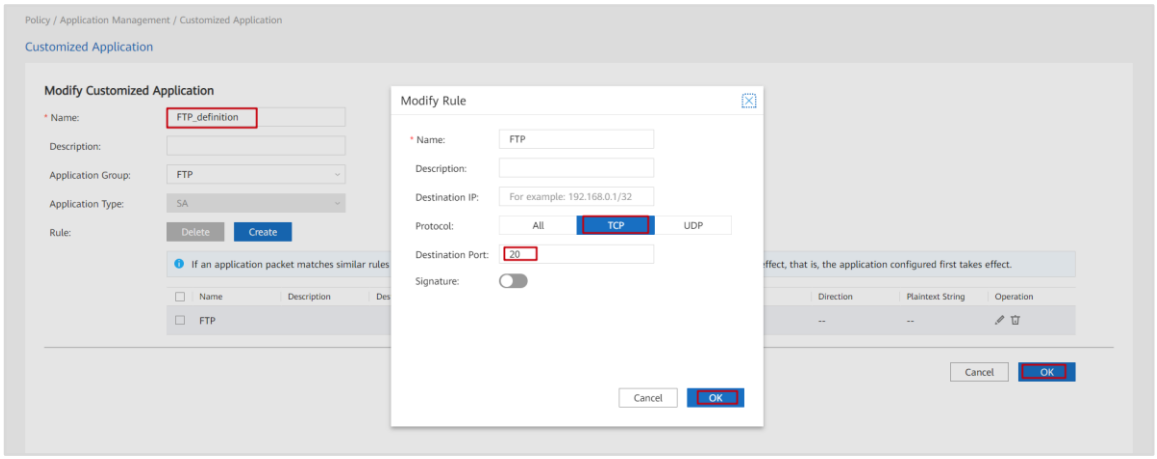
Create application groups to match FTP and HTTP traffic.

In this lab, iPerf3 is used to simulate service traffic. To ensure that the controller can identify the traffic, add customized applications to the application groups.

# Create customized applications.

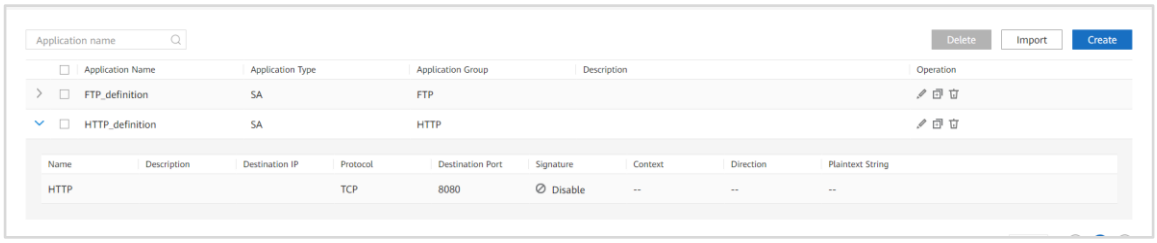



Choose **Policy** > **Application Management** > **Customized Application** from the main menu, click **Create**, and create customized applications **FTP_definition** and **HTTP_definition**.



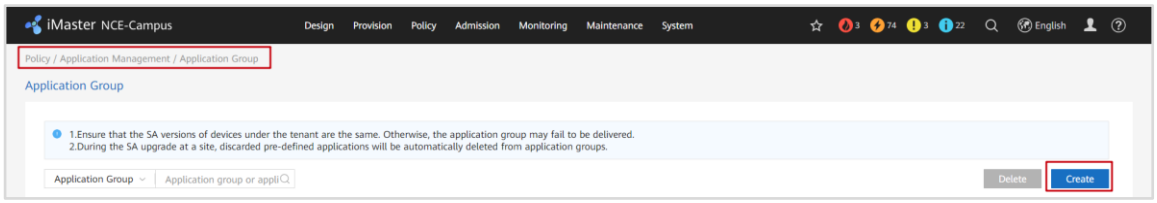

Perform traffic identification by matching the TCP port number.

The method of creating **FTP_definition** is similar to that of creating **HTTP_definition**.



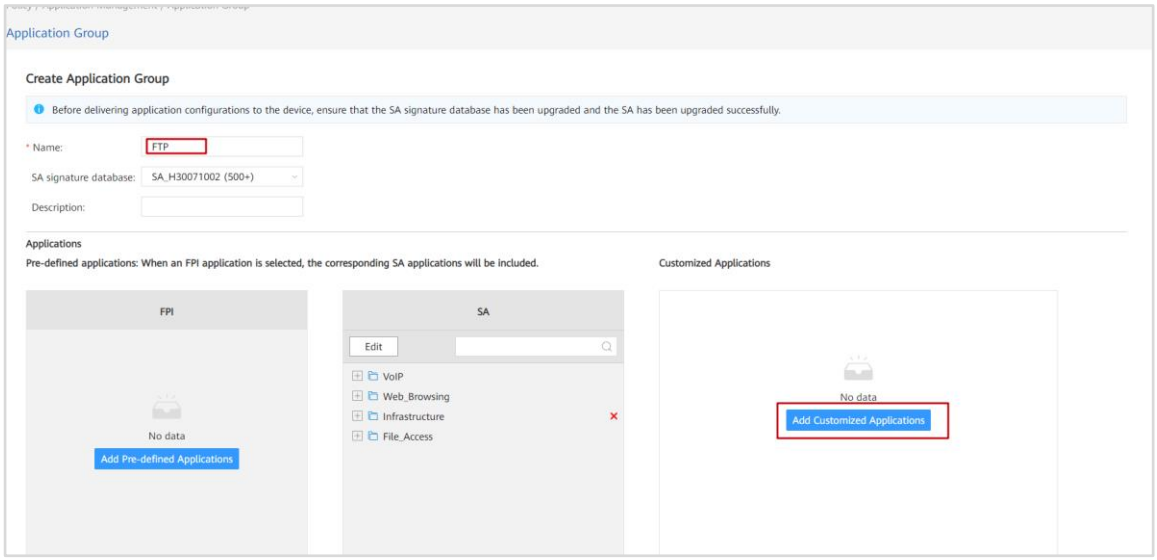

In the preceding figure, the customized application **HTTP_definition** matches destination TCP port 8080.

# Create application groups.

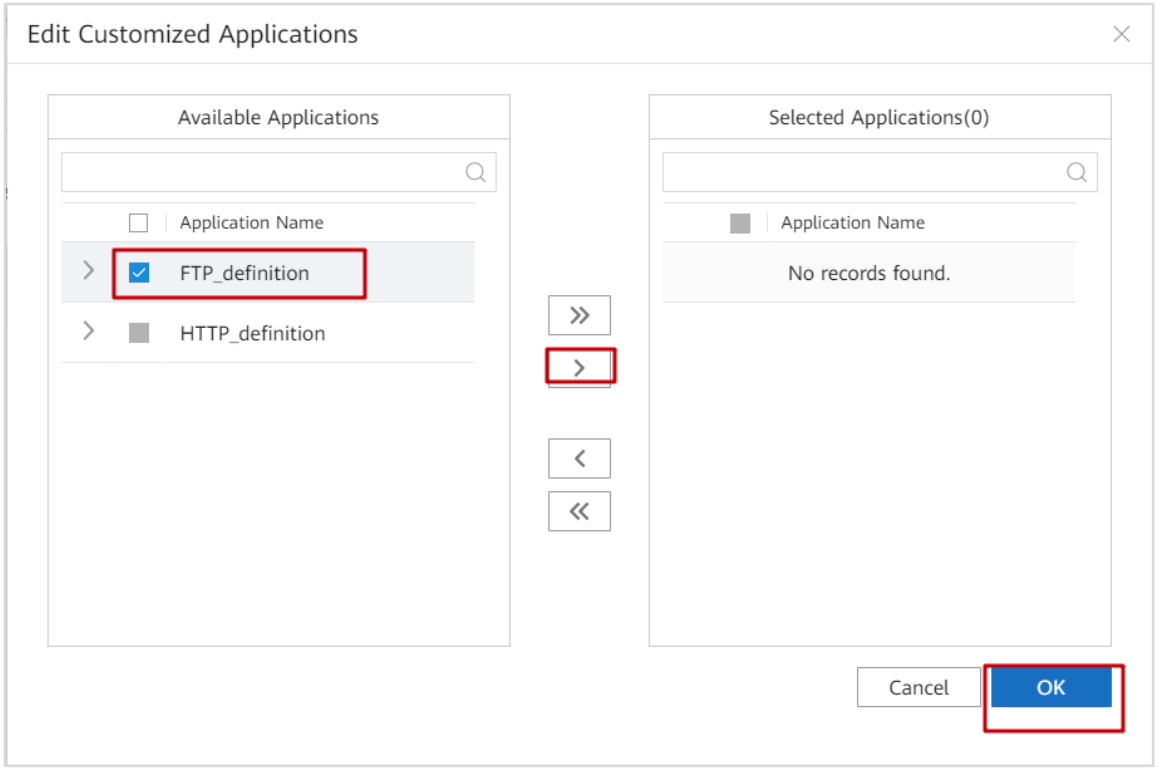



Choose **Policy** > **Application Management** > **Application Group** from the main menu, and create an application group.

Create application group **FTP**.



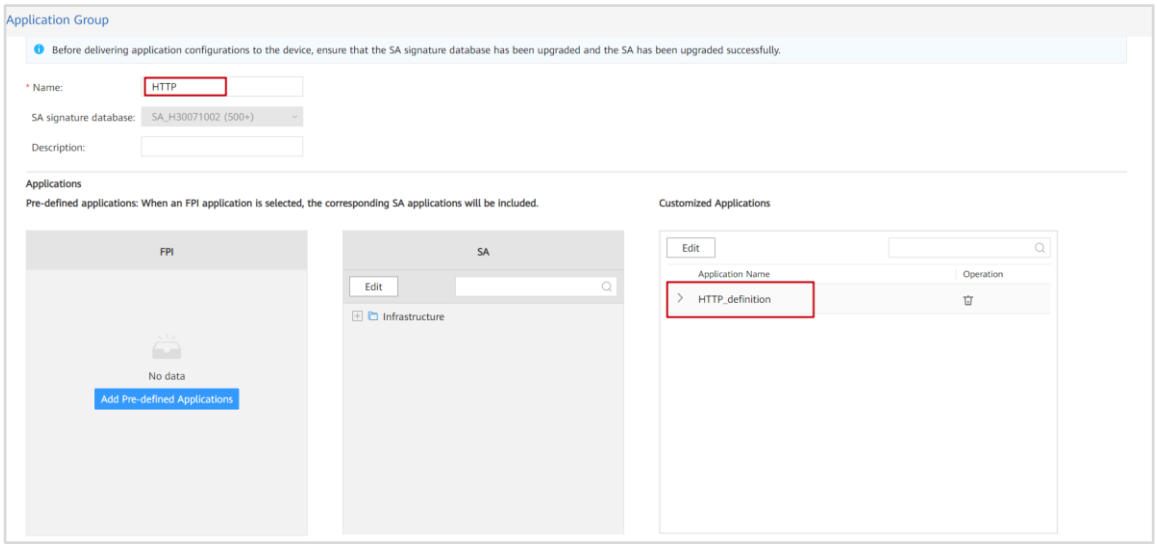

Search for and add pre-defined application **FTP**.




Select the customized application **FTP_definition**.

Then click **OK**. The application group **FTP** is created.

Use the same method to create application group **HTTP**.

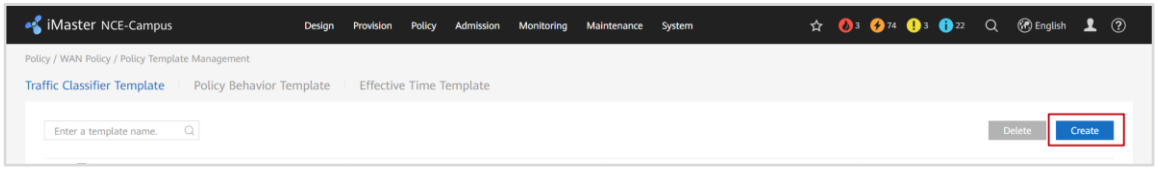



You can add pre-defined and customized applications to the application groups to ensure that both the test traffic and actual service traffic can be identified.

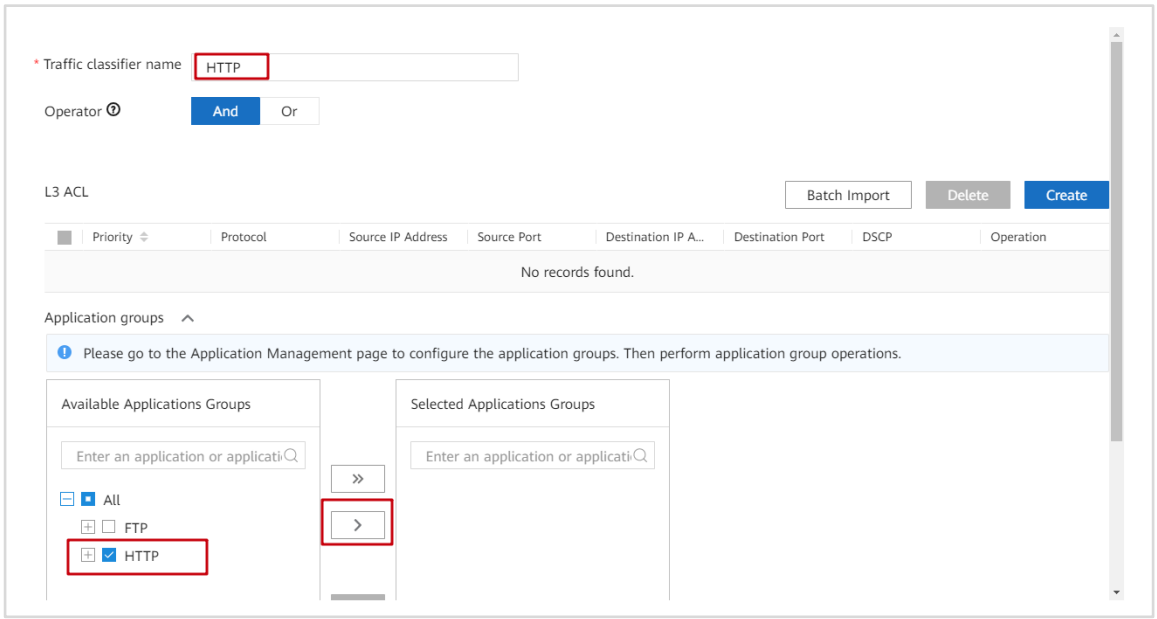
Step 3 Configure traffic classifier templates.

Create traffic classifier templates **FTP** and **HTTP** to match the created application groups.

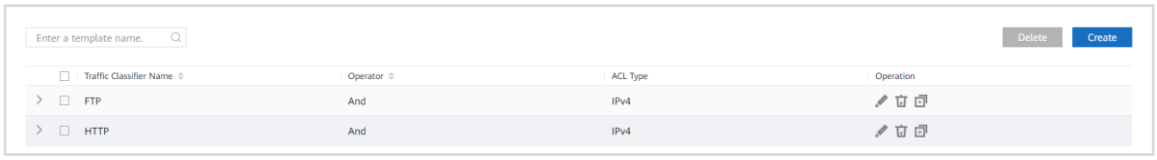
# Create a traffic classifier template.




Choose **Policy** > **WAN Policy** > **Policy Template Management** from the main menu and create a traffic classifier template named **HTTP**.



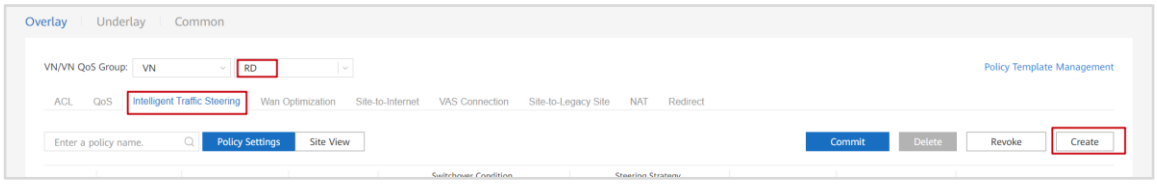

Invoke application group **HTTP**. Use the same method to create traffic classifier template **FTP** and invoke application group **FTP**.



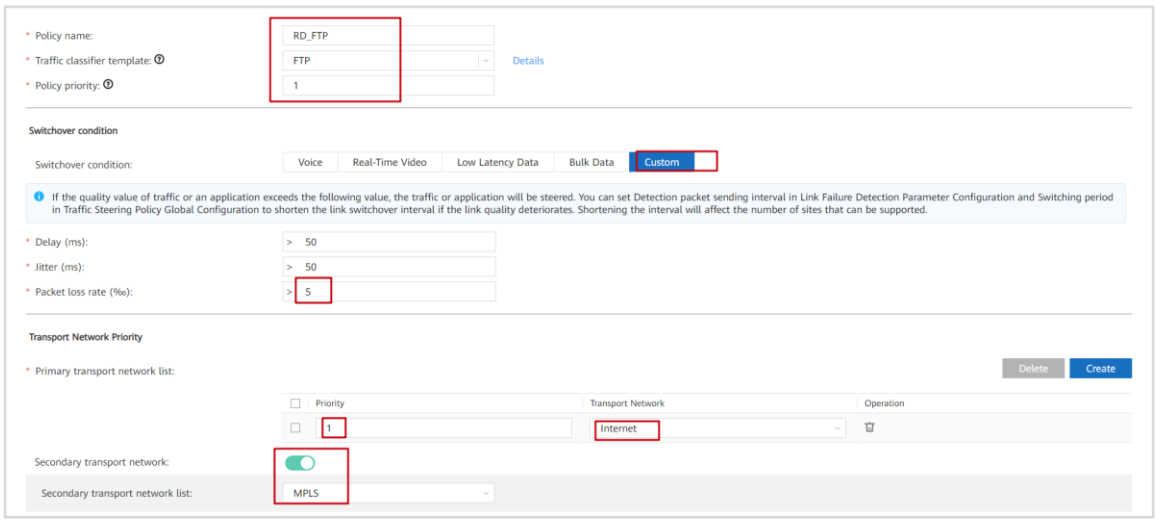

Step 4 Configure intelligent traffic steering policies.

Create intelligent traffic steering policies to preferentially transmit FTP traffic of virtual network **RD** over the Internet link and HTTP traffic of virtual network **OA** over the MPLS link.

# Create an intelligent traffic steering policy for FTP traffic.



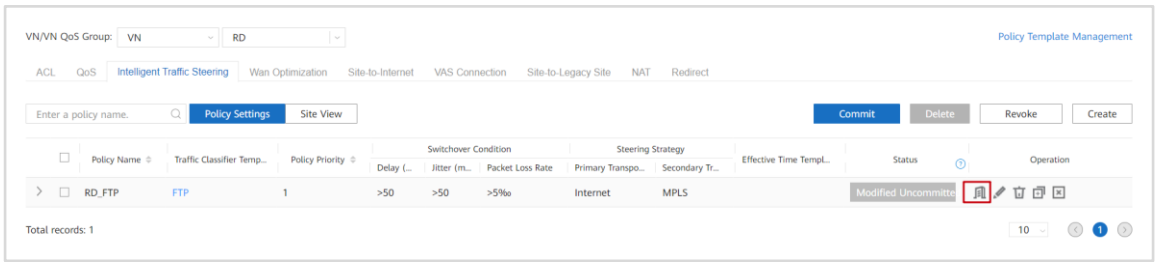

Choose **Policy** > **WAN Policy** > **Traffic Policy** from the main menu and create an intelligent traffic steering policy.

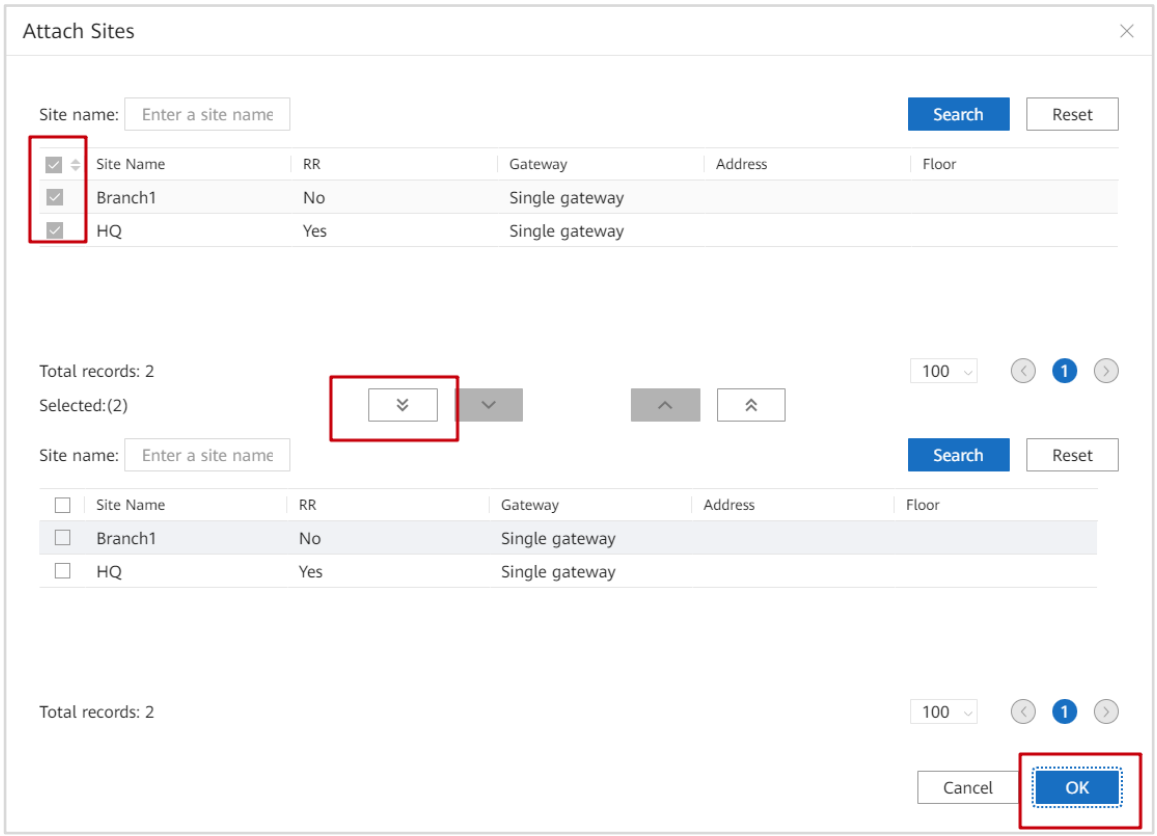



Set the policy name to **RD_FTP**, traffic classifier template to **FTP**, policy priority to **1**, switchover indicator to **Custom**, **Packet loss rate** to **> 5%**, **Primary transport network list** to **Internet**, and **Secondary transport network list** to **MPLS**, and keep other parameters unchanged.

Then click **OK**.

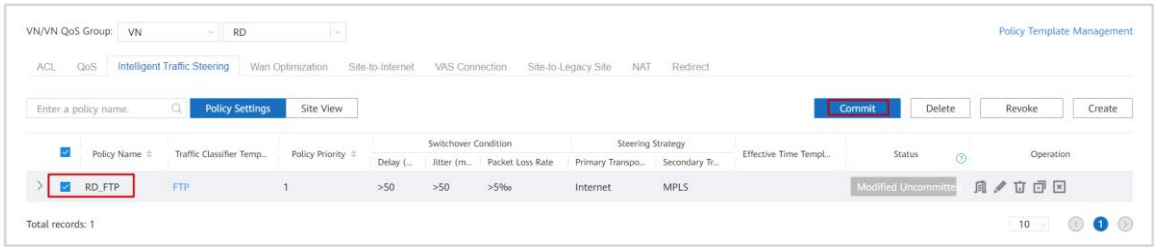
# Associate the intelligent traffic steering policy with sites.





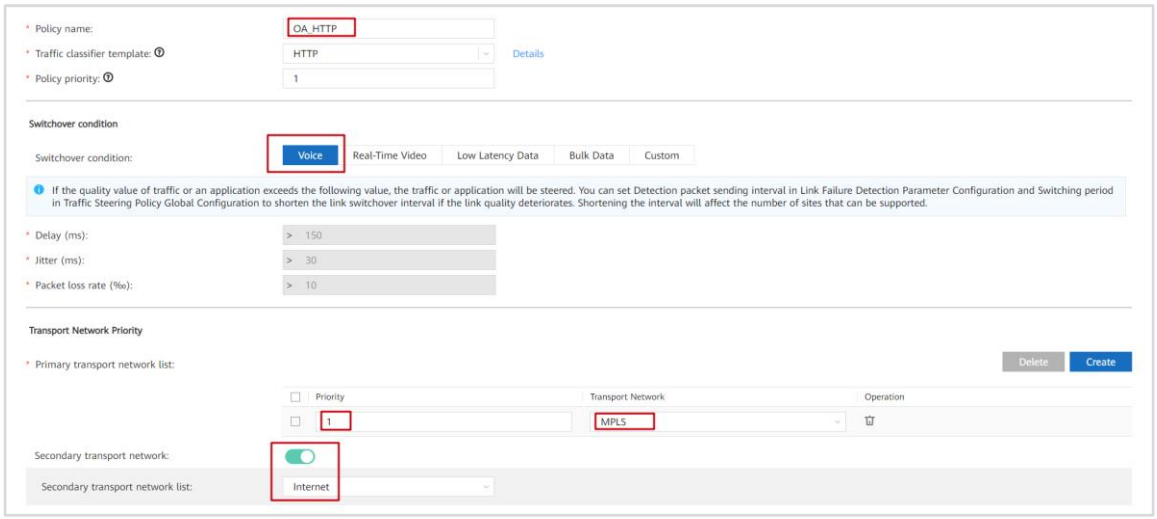

Associate the intelligent traffic steering policy with HQ and Branch1.

# Commit the policy.




Commit policy **RD_FTP**.

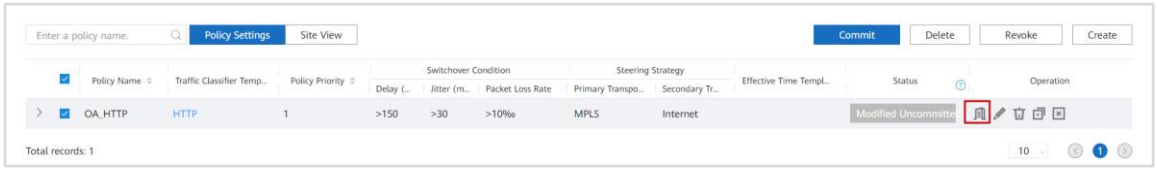
# Create an intelligent traffic steering policy for HTTP traffic.

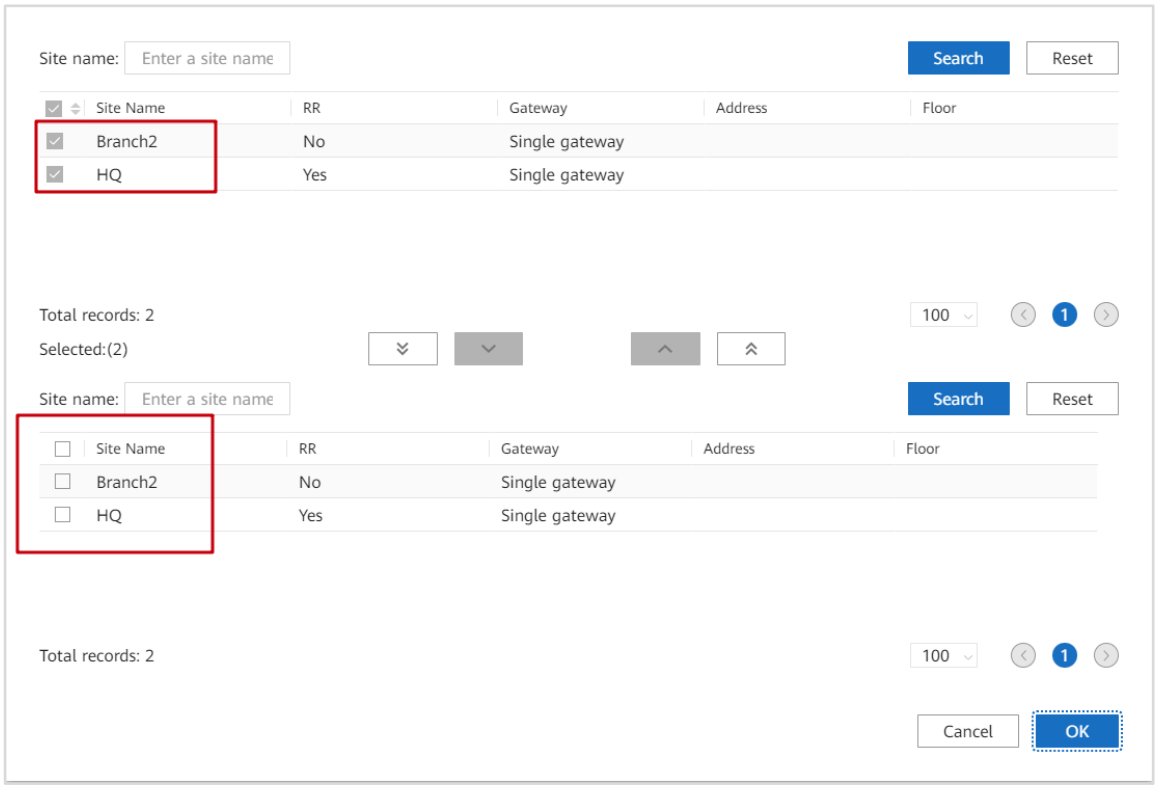



Set the policy name to **OA_HTTP**, traffic classifier template to **HTTP**, policy priority to **1**, **Primary transport network list** to **MPLS**, and **Secondary transport network list** to **Internet**, retain the default value of the switchover indicators, and keep other parameters unchanged.

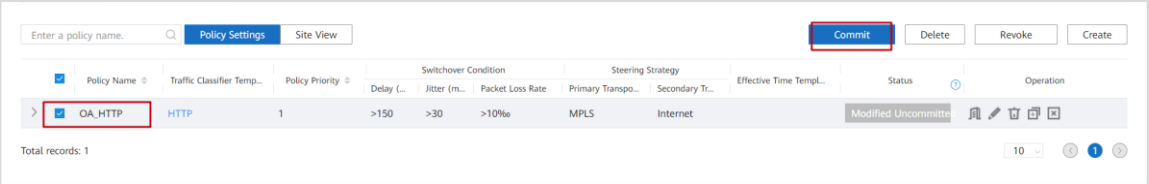
Then click **OK**.

# Associate the intelligent traffic steering policy with sites.




# Commit the policy.




Commit policy **OA_HTTP**.

Step 5 Simulate service traffic.

Use iPerf3 on PCs at the HQ and branch sites to simulate service traffic.

In this lab, the installation and usage guidance of iPerf3 are not provided. You can query the required information on the Internet or use actual service software and servers to generate HTTP and FTP traffic.

# Log in to PC1 at the HQ using the account of the RD department and check the IP address of PC1.

```
C:\Users\PC1>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix . . . . . :
   Link-local IPv6 Address . . . . . : fe80::a46b:62f8:a696:b974%12
   IPv4 Address . . . . . : 172.17.30.225
   Subnet Mask . . . . . : 255.255.255.0
   Default Gateway . . . . . : 172.17.30.254
```

# Check the IP address of PC4 at Branch2.

```
C:\Users\PC4>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix . . . . . :
   Link-local IPv6 Address . . . . . : fe80::a46b:62f8:a696:b974%12
   IPv4 Address . . . . . : 172.18.30.27
   Subnet Mask . . . . . : 255.255.255.0
   Default Gateway . . . . . : 172.18.30.254
```

# Start the iPerf3 server process on PC1 and set the listening port number to 20.

```
PS C:\Users\PC1\Desktop> iperf3 -s -p 20 -B 172.17.30.225 -i 2
-----------------------------------------------------------
Server listening on 20
-----------------------------------------------------------
```

Start iPerf3 in server mode, set the listening port number to 20, and bind the IP address of the port to simulate the FTP server.

# Start the iPerf3 client process on PC4 to connect to the server process on PC1.

```
C:\Users\PC4>iperf3 -c 172.17.30.225 -t 20 -p 20 -i 2
Connecting to host 172.17.30.225, port 20
[  4] local 172.18.30.27 port 53078 connected to 172.17.30.225 port 20
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-2.00   sec   180 MBytes   754 Mbits/sec
[  4]   2.00-4.00   sec   188 MBytes   787 Mbits/sec
[  4]   4.00-6.00   sec   188 MBytes   791 Mbits/sec
[  4]   6.00-8.00   sec   189 MBytes   793 Mbits/sec
[  4]   8.00-10.00  sec   189 MBytes   792 Mbits/sec
[  4]  10.00-12.00  sec   189 MBytes   794 Mbits/sec
[  4]  12.00-14.00  sec   189 MBytes   791 Mbits/sec
[  4]  14.00-16.00  sec   189 MBytes   794 Mbits/sec
[  4]  16.00-18.00  sec   188 MBytes   787 Mbits/sec
[  4]  18.00-20.00  sec   188 MBytes   788 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-20.00  sec  1.83 GBytes   787 Mbits/sec                  sender
[  4]   0.00-20.00  sec  1.83 GBytes   787 Mbits/sec                  receiver

iperf Done.
```

Simulate FTP traffic at the maximum rate for 20s.

# Log in to PC2 at the HQ using the account of the marketing department and check the IP address of PC2.

```
C:\Users\PC2>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix . . . . . :
   Link-local IPv6 Address . . . . . : fe80::a46b:62f8:a696:b974%12
   IPv4 Address . . . . . . . . . . . : 172.17.20.167
   Subnet Mask  . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . . . . : 172.17.20.254
```

# Check the IP address of PC5.

```
C:\Users\PC5>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix . . . . . :
   Link-local IPv6 Address . . . . . : fe80::a46b:62f8:a696:b974%12
```

```
   IPv4 Address . . . . . . . . . . . : 172.19.20.234
   Subnet Mask  . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . .  : 172.19.20.234
```

# Start the iPerf3 server process on PC2 and set the listening port number to 8080.

```
C:\Users\PC2>iperf3 -s -p 8080 -B 172.17.20.167 -i 2
-----------------------------------------------------------
Server listening on 8080
-----------------------------------------------------------
```

Start iPerf3 in server mode, set the listening port number to 8080, and bind the IP address of the port.

# Start the iPerf3 client process on PC5 to connect to the server process on PC2.

```
C:\Users\PC5>iperf3 -c 172.17.20.167 -t 20 -p 8080 -i 2
Connecting to host 172.17.20.167, port 8080
[  4] local 172.19.20.234 port 62603 connected to 172.17.20.167 port 8080
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-2.00   sec   158 MBytes   663 Mbits/sec
[  4]   2.00-4.00   sec   166 MBytes   697 Mbits/sec
[  4]   4.00-6.00   sec   166 MBytes   695 Mbits/sec
[  4]   6.00-8.00   sec   166 MBytes   697 Mbits/sec
[  4]   8.00-10.00  sec   166 MBytes   696 Mbits/sec
[  4]  10.00-12.00  sec   166 MBytes   696 Mbits/sec
[  4]  12.00-14.00  sec   166 MBytes   696 Mbits/sec
[  4]  14.00-16.00  sec   166 MBytes   697 Mbits/sec
[  4]  16.00-18.00  sec   167 MBytes   700 Mbits/sec
[  4]  18.00-20.00  sec   166 MBytes   698 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-20.00  sec  1.61 GBytes   694 Mbits/sec                  sender
[  4]   0.00-20.00  sec  1.61 GBytes   694 Mbits/sec                  receiver

iperf Done.
```

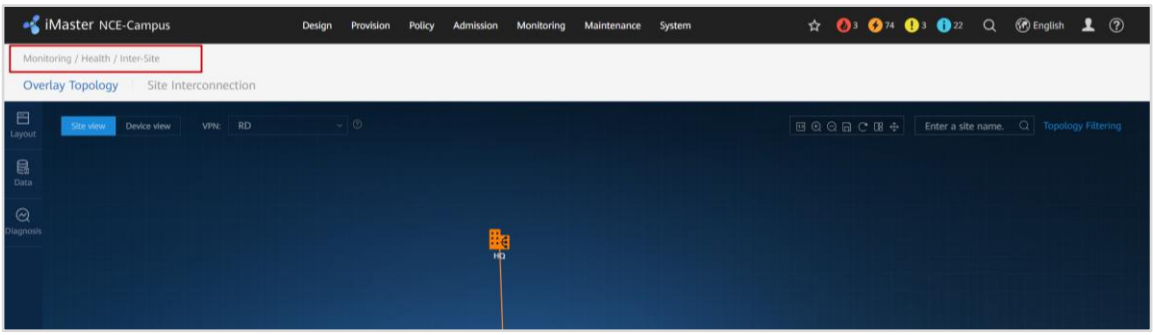
Simulate HTTP traffic at the maximum rate for 20s.

## Step 6 Verify configurations.

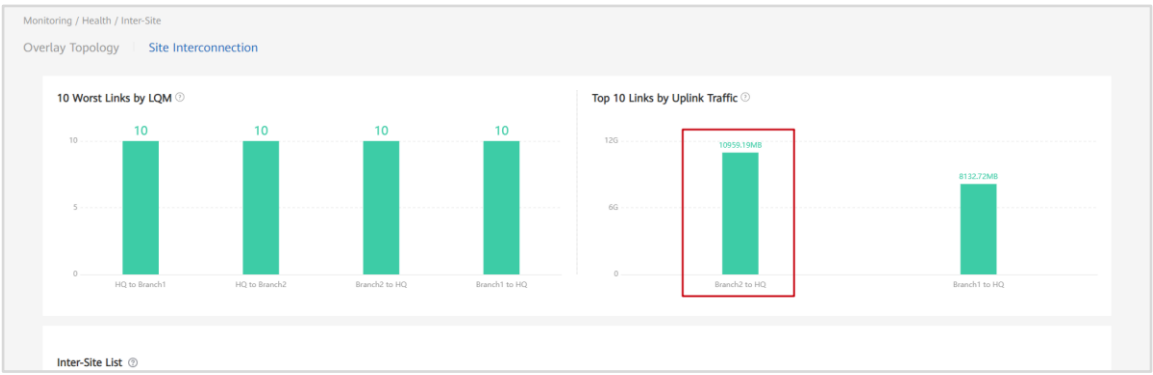
Check whether the service traffic direction on the controller is the same as expected.

It takes a period of time for devices to report traffic and the controller to analyze the traffic. After the traffic test is complete, you are advised to wait for several minutes before viewing the result.

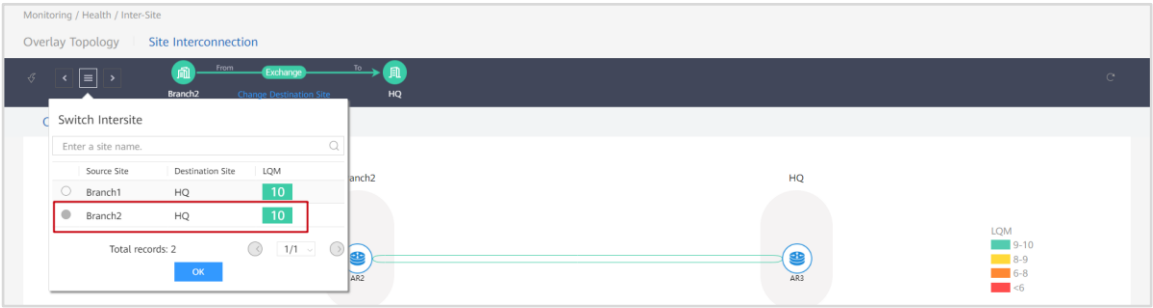
# Check the HTTP traffic direction on virtual network **OA**.



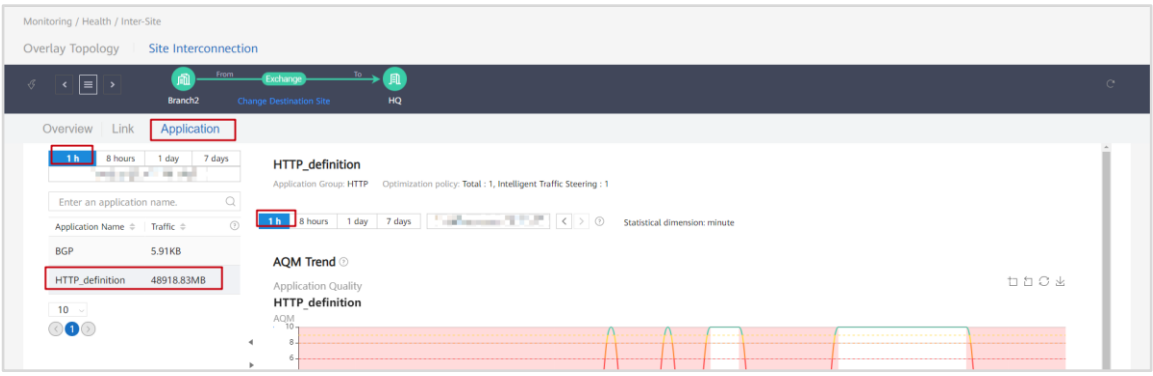

Choose **Monitoring** > **Health** > **Inter-Site** from the main menu and view the inter-site traffic.



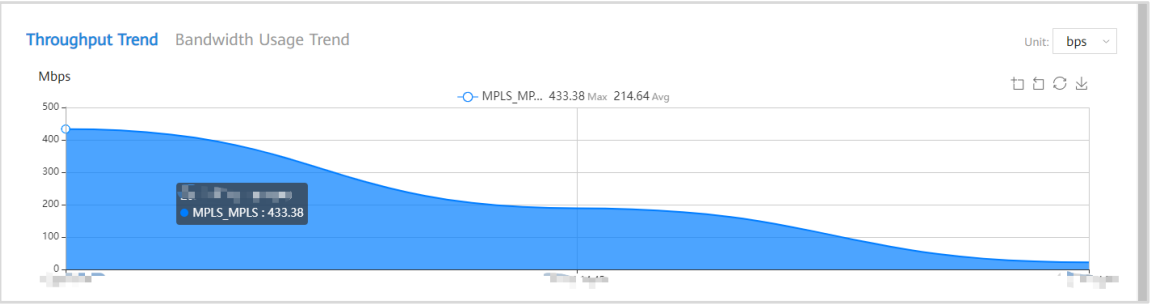

Click the **Site Interconnection** tab and click **Top 10 Links by Uplink Traffic**. The page for viewing inter-site traffic is displayed.




Click the icon in the upper left corner to switch to **Branch2-HQ** and view the HTTP traffic of OA users.

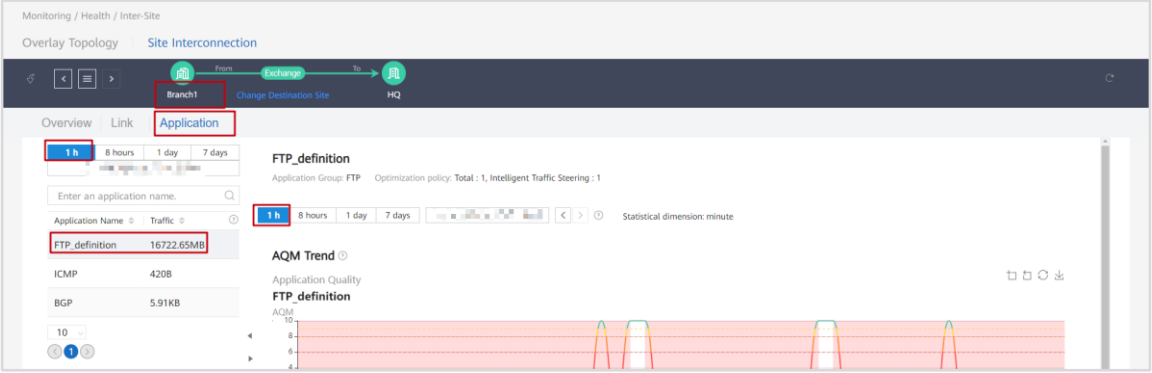



Click the **Application** tab, select a time range, click a specific application (such as **HTTP_definition**), and view **Throughput Trend** on the right of the page.

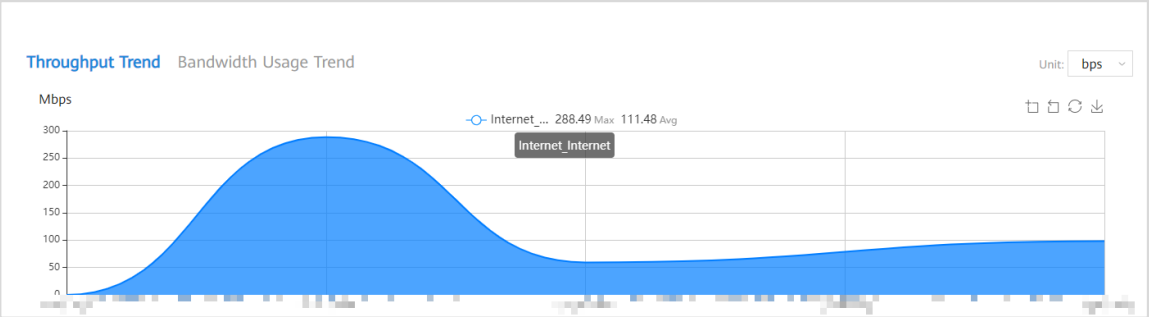



It shows that traffic from Branch2 to the HQ is forwarded through the MPLS link as expected.

# Check the FTP traffic direction on virtual network **RD**.




Perform the same operations to view the throughput trend of FTP traffic from Branch1 to the HQ. It shows that the traffic is forwarded through the Internet link as expected.




# Simulate a fault on the Internet link.

```
[AR_Server_SW]interface GigabitEthernet 0/0/1
[AR_Server_SW-GigabitEthernet0/0/1]qos lr inbound cir 64
```

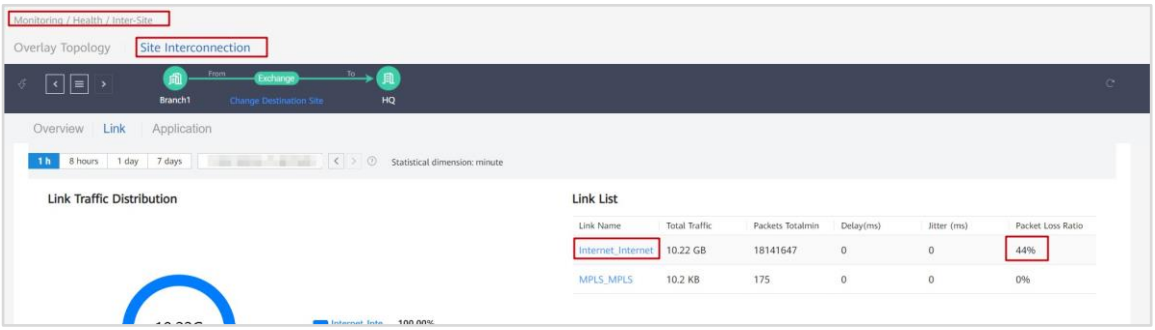
Limit the inbound traffic rate on the interface connecting AR_Server_SW to AR1 to the minimum value.

# Perform a traffic test again on PC4.

```
C:\Users\PC4>iperf3 -c 172.17.30.225 -t 20 -p 20 -i 2
Connecting to host 172.17.30.225, port 20
[  4] local 172.18.30.27 port 64173 connected to 172.17.30.225 port 20
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-2.01   sec   256 KBytes   1.04 Mbits/sec
[  4]   2.01-4.01   sec   0.00 Bytes   0.00 bits/sec
```

```
[  4]   4.01-6.01   sec  0.00 Bytes  0.00 bits/sec
[  4]   6.01-8.02   sec  0.00 Bytes  0.00 bits/sec
[  4]   8.02-10.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  10.01-12.00  sec  0.00 Bytes  0.00 bits/sec
[  4]  12.00-14.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  14.01-16.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  16.01-18.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  18.01-20.01  sec  0.00 Bytes  0.00 bits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-20.01  sec   256 KBytes   105 Kbits/sec                  sender
[  4]   0.00-20.01  sec  74.0 KBytes  30.3 Kbits/sec                  receiver

iperf Done.
```

Simulate traffic for multiple times. After a period of time, check the packet loss rate of the link.




Choose **Monitoring** > **Health** > **Inter-Site** from the main menu, click the **Link** tab, and view the link quality of **Branch1-HQ**. It shows that the packet loss ratio of the Internet link is 41%.

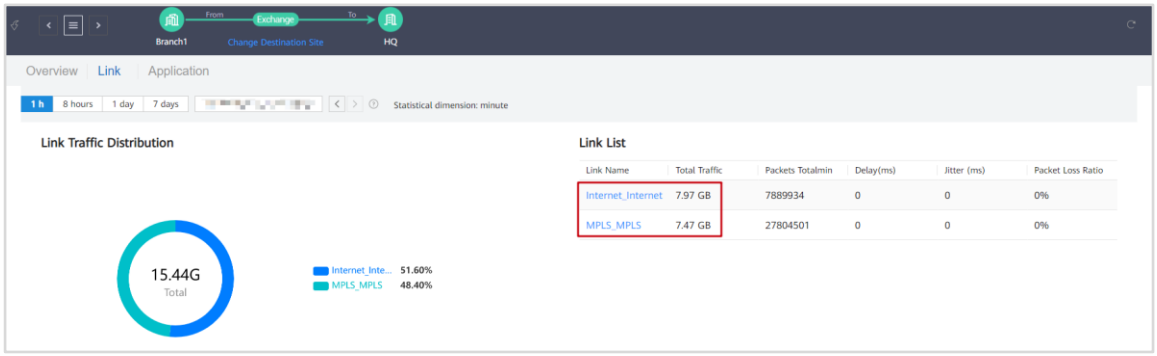
# Perform a traffic test on PC4 again.

```
C:\Users\PC4>iperf3 -c 172.17.30.225 -t 400 -p 20 -i 2
Connecting to host 172.17.30.225, port 20
[  4] local 172.18.30.27 port 64279 connected to 172.17.30.225 port 20
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-2.01   sec   256 KBytes  1.04 Mbits/sec
[  4]   2.01-4.01   sec  0.00 Bytes  0.00 bits/sec
[  4]   4.01-6.01   sec  0.00 Bytes  0.00 bits/sec
[  4]   6.01-8.01   sec  0.00 Bytes  0.00 bits/sec
[  4]   8.01-10.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  10.01-12.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  12.01-14.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  14.01-16.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  16.01-18.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  18.01-20.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  20.01-22.01  sec   128 KBytes   524 Kbits/sec
[  4]  22.01-24.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  24.01-26.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  26.01-28.01  sec  0.00 Bytes  0.00 bits/sec
[  4]  28.01-30.00  sec  93.8 MBytes   396 Mbits/sec
[  4]  30.00-32.81  sec   256 KBytes   747 Kbits/sec
[  4]  32.81-34.00  sec   100 MBytes   706 Mbits/sec
```

```
[  4]  34.00-36.00  sec   167 MBytes   701 Mbits/sec
[  4]  36.00-38.00  sec   168 MBytes   703 Mbits/sec
[  4]  38.00-40.00  sec   166 MBytes   698 Mbits/sec
[  4]  40.00-42.00  sec   166 MBytes   698 Mbits/sec
[  4]  42.00-44.00  sec   167 MBytes   702 Mbits/sec
[  4]  44.00-46.00  sec   168 MBytes   703 Mbits/sec
[  4]  46.00-48.00  sec   165 MBytes   694 Mbits/sec
[  4]  48.00-50.26  sec   256 KBytes   928 Kbits/sec
[  4]  50.26-52.00  sec   146 MBytes   703 Mbits/sec
[  4]  52.00-54.00  sec   167 MBytes   702 Mbits/sec
[  4]  54.00-56.00  sec   167 MBytes   701 Mbits/sec
[  4]  56.00-58.00  sec   167 MBytes   701 Mbits/sec
[  4]  58.00-59.30  sec   108 MBytes   701 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-59.30  sec  2.07 GBytes   300 Mbits/sec                  sender
[  4]   0.00-59.30  sec  0.00 Bytes  0.00 bits/sec                    receiver
iperf3: interrupt - the client has terminated
```

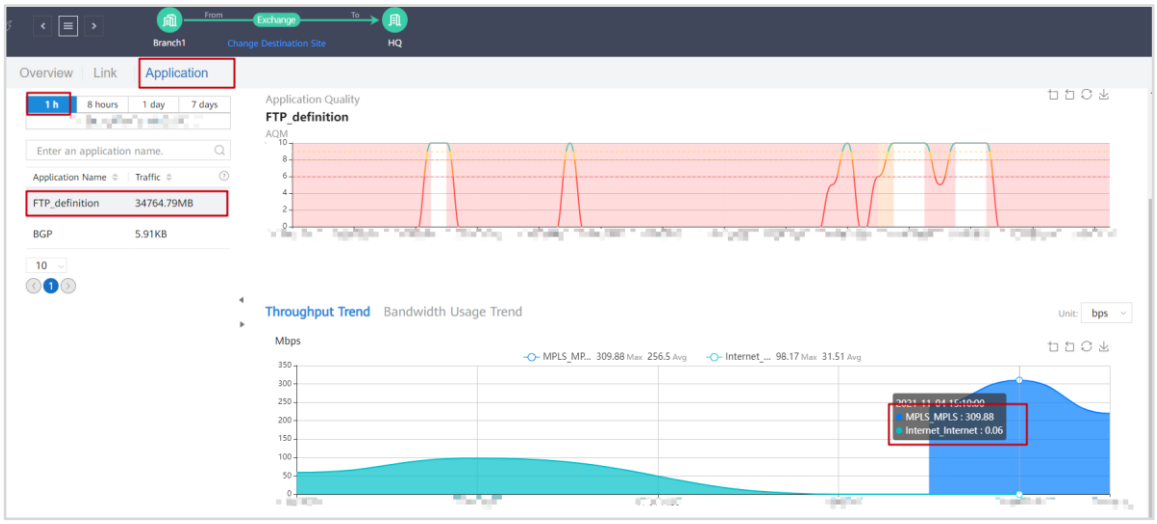
Simulate traffic transmission for a long time. It shows that the traffic rate becomes normal after packets are lost for a period of time, indicating that traffic has been switched to the MPLS link.

# Check the link traffic distribution.




It shows that traffic between Branch1 and HQ is transmitted through both Internet and MPLS links.

# Check the throughput trend of an application.




It shows that traffic is switched from the Internet link to the MPLS link after packet loss.

## 7.1.3 (Optional) Clearing Configurations

Before the lab, restore the factory settings of iMaster NCE and network devices (AR1, AR2, and AR3).

Note: Delete the configurations of iMaster NCE and then those of network devices.

### 7.1.3.1 Deleting Intelligent Traffic Steering Policies

Step 1 Choose **Multi-Branch Interconnection** > **Traffic Policy** from the main menu of iMaster NCE. The traffic policy page is displayed.

Step 2 Set **VN** to **RD**, click **Intelligent Traffic Steering**, select policy **RD_FTP**, and click  . The policy is in the **Delete Uncommitted** state.

Step 3 Select policy **RD_FTP** and click **Commit**. On the page that is displayed, click **OK**. The policy is deleted.

Step 4 Set **VN** to **OA** and perform the same operations to delete intelligent traffic steering policy **OA_HTTP**.

### 7.1.3.2 Deleting Application Groups and Traffic Classifier Templates

Step 1 Choose **Policy** > **Policy Template Management** > **Traffic Classifier Template** from the main menu of iMaster NCE. The policy template management page is displayed.

Step 2 Select all traffic classifier templates and click **Delete** in the upper right corner.

Step 3 Choose **Policy** > **Application Group** from the main menu. The application group page is displayed.

Step 4 Select all application groups and click **Delete** in the upper right corner.

Step 5 Choose **Policy** > **Customized Application** from the main menu. The customized application page is displayed.

Step 6 Select all customized applications and click **Delete** in the upper right corner.

### 7.1.3.3 Deleting Site-to-Internet Access and NAT Configurations

Step 1 Choose **Multi-Branch Interconnection** > **Traffic Policy** from the main menu of iMaster NCE. The traffic policy page is displayed.

Step 2 Select virtual network **public** and click **Site-to-Internet**. In the **Local Internet access** area, select site **HQ** and click  to delete the Internet access configuration of the site.

Step 3 Perform the same operations to delete the Internet access configurations of virtual networks **OA** and **RD**.

Step 4 Choose Multi-Branch Interconnection > Traffic Policy from the main menu and click the Underlay tab.

Step 5 Click the **NAT** tab, select site **HQ**, click **Static NAT**, select the NAT policy, and click  . The NAT policy is deleted.

### 7.1.3.4 Deleting Virtual Networks

Step 1 Choose **Multi-Branch Interconnection** > **Interconnection Configuration** from the main menu of iMaster NCE. The interconnection configuration page is displayed.

Step 2 Select virtual network **OA**, and click **WAN Service**. On the displayed page, delete **Branch sites** and **Hub sites**.

Step 3 In the **Branch sites** area, select sites and click  . The branch sites are deleted.

Step 4 In the **Hub sites** area, select sites and click  . The HQ sites are deleted.

Step 5 Click **OK** at the bottom of the page. The WAN services of virtual network **OA** are deleted.

Step 6 Click **LAN-WAN Interconnection**. On the displayed page, select **Branch2**. On the **Interconnection Interface Configuration** tab page, click **Advanced Mode**, and then click  . The LAN-WAN interconnection interface is deleted. Select site **HQ**. In the **Interconnection Route Configuration** area, select and delete all routes, and then click  next to **static**. The interconnection routing protocol is deleted. On the **Interconnection Interface Configuration** tab page, click **Advanced Mode**, and click  . The interconnection interface is deleted. Then click **Apply** at the bottom of the page.

Step 7 Click **Back** in the upper left corner of the page to go to the interconnection configuration page. Click  next to virtual network **OA**, select all sites under **Selected Sites**, click  to disassociate the sites from the virtual network, and click **OK** at the bottom of the page.

Step 8 Click  next to virtual network **OA**. The virtual network is then deleted.

Step 9 Perform the same operations to delete virtual networks **RD** and **public**.

### 7.1.3.5 Deleting the Inter-Site Networking

Step 1 Choose **Multi-Branch Interconnection** > **Basic Network** > **Inter-Site Networking** from the main menu of iMaster NCE. The inter-site networking page is displayed.

Step 2 Select Branch1 and Branch2 one by one and click  . The branch sites are disconnected from the RR site.

### 7.1.3.6 Deleting WAN Routes

Step 1 Choose **Multi-Branch Interconnection** > **Basic Network** > **WAN Underlay** from the main menu of iMaster NCE. The WAN underlay configuration page is displayed.

Step 2 Select site **HQ**, click the **WAN Route** tab, select all static routes, and click **Delete** to delete all static routes. Then click  next to **IPv4 Static** The WAN route of this type is deleted.

Step 3 Perform the same operations to delete WAN routes of Branch1 and Branch2.

### 7.1.3.7 Deleting ZTP Configurations

Step 1 Choose **Multi-Branch Interconnection** > **Basic Network** > **ZTP** from the main menu of iMaster NCE. The ZTP configuration page is displayed.

Step 2 Select site **HQ** on the left of the page and click **Clear WAN Configurations** on the right of the page. The ZTP configuration is deleted.

Step 3 Perform the same operations to delete ZTP configurations of Branch1 and Branch2.

### 7.1.3.8 Deleting Devices

Step 1 Choose **Multi-Branch Interconnection** > **Basic Network** > **Device Management** from the main menu of iMaster NCE. The device management page is displayed.

Step 2 Select all ARs (AR1, AR2, and AR3) and click **Delete Device**. The devices are then deleted.

### 7.1.3.9 Deleting a Site

Step 1 Choose **Multi-Branch Interconnection** > **Basic Network** > **Site Design** from the main menu of iMaster NCE. The site design page is displayed.

Step 2 Select the site to be deleted and click **Delete**. This lab is associated with VXLAN-based virtualized campus network deployment. Therefore, delete the HQ site as required.

### 7.1.3.10 Restoring Global Configurations

Step 1 Choose **Multi-Branch Interconnection** > **Basic Network** > **Global Configuration** from the main menu of iMaster NCE. The global configuration page is displayed.

Step 2 In the **IP pool** area, click  . In the displayed area, select an address pool and click  next to it. The address pool is deleted.

### 7.1.3.11 Deleting Configurations of AR Routers

Run the following commands to delete configurations of an AR router and restore the factory settings.

```
<AR3>set factory-configuration from default
 Warning: The current factory configuration will be replaced, and it's irreversible. Are you sure to set the factory configuration?[Y/N]:y
 Info: Successfully set factory config!
<AR3>factory-configuration reset
Warning: It will clean the configuration which you have saved. If you have set the factory-configuration by hand, it will start from the modified factory-configuration, else it will start from the original one, when you restart the device. Continue? [y/n]:y
 Info: Successfully set factory config!
<AR3>delete /unreserved flash:/startup_v1.rdb
Warning: The contents of file flash:/startup_v1.rdb cannot be recycled. Continue? [Y/N]:y
Info: Deleting file flash:/startup_v1.rdb...
Info: Deleting file permanently from flash will take a long time if needed...succeeded.
<AR3>reset saved-configuration
 Warning: This will delete the configuration in the flash memory.
 The device configurations will be erased to reconfigure.
 Are you sure? (y/n):y
Info: Clear the configuration in the device successfully.
<AR3>reboot fast
System will reboot! Continue? [y/n]:y
```

AR3 is used as an example. Restore other ARs to factory defaults in the same way.

### 7.1.3.12 Deleting Configurations of NE Routers

```
<P1>reset saved-configuration
Warning: The action will delete the saved configuration on the device.
The configuration will be erased to reconfigure.Continue? [Y/N]:y
Warning: Now clearing the configuration on the device.
Info: Succeeded in clearing the configuration on the device.
```

```
<P1>reboot
MPU 7:
Next startup system software: cfcard:/NetEngine8000-M6-V800R012C10SPC300.cc
Next startup saved-configuration file: NULL
Next startup paf file: default
Next startup patch package: NULL
The configuration information of any other MPU is the same as that of MPU 7.
Warning: Current configuration will be saved to the next startup saved-configuration file! Continue?
[Y/N]:n
System will reboot! Continue? [Y/N]:y
```

P1 is used as an example. Restore other NEs to factory defaults in the same way.

**----End**

## 7.1.4 Quiz

In this lab, we can control traffic transmission between sites by controlling BGP EVPN route transmission. In addition to this method, how can we control branch site access to a network segment at the HQ?

# 8 SR-MPLS

## 8.1 L3VPNv4 over SR-MPLS BE

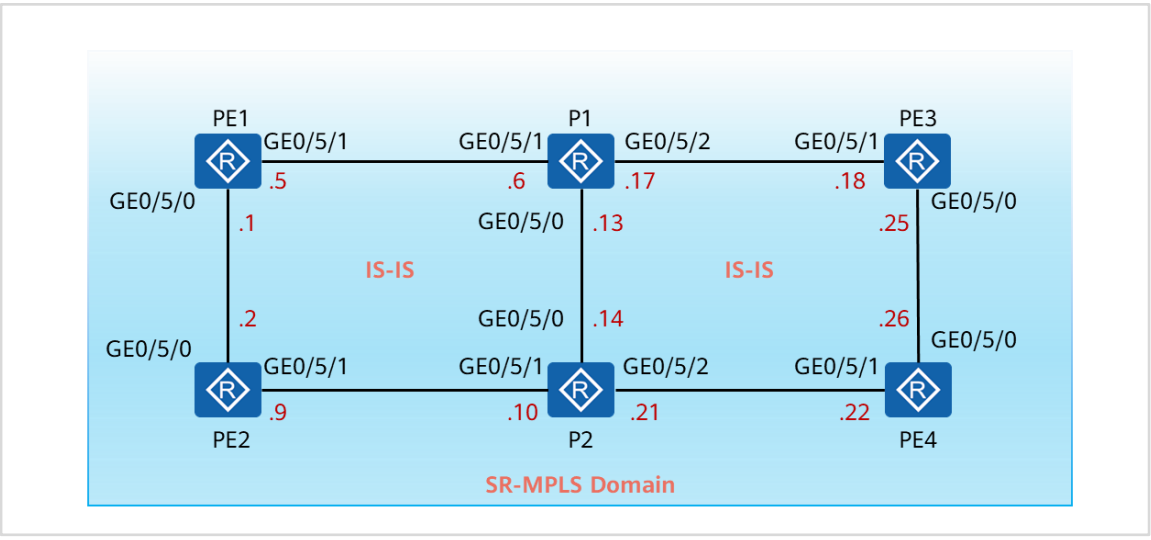
### 8.1.1 About This Lab

#### 8.1.1.1 Objectives

- Configure IS-IS to ensure that PEs are routable to each other.
- Configure SR-MPLS to establish SR LSPs.
- Recurse L3VPN tunnels used for communication between CEs to SR-MPLS BE tunnels.

#### 8.1.1.2 Networking Description

**Figure 8-1 L3VPNv4 over SR-MPLS BE topology**




The figure shows the device connection and IP address planning. The interface interconnection addresses are in the format of 10.0.0.*Y*/30, and the values represented by Y are shown in the figure. Loopback0 is created on all devices, and loopback interface IP addresses are in the format of 10.0.*X*.*X*. The values indicated by *X* are listed in the table of the corresponding step.

Loopback1 is created on PE1 and PE4, with addresses being 10.1.1.1/32 and 10.1.4.4/32, respectively, to simulate CE user access.

# 8.1.2 Lab Task

## 8.1.2.1 Configuration Roadmap

1. Configure basic IP addresses for devices.
2. Configure IS-IS in the SR domain. Specifically, enable IS-IS on interconnection and Loopback0 interfaces for communication in the SR domain.
3. Configure MPLS. Specifically, enable MPLS and set MPLS LSR IDs on devices.
4. Configure SR. Specifically, enable SR globally, enable IS-IS extensions for SR, and configure node SIDs.
5. Establish an MP-IBGP peer relationship between PE1 and P1, between PE1 and P2, between PE4 and P1, and between PE4 and P2. P1 and P2 function as RRs to reflect VPNv4 routes from PE1 and PE4.
6. Create a VPN instance named **vpna**, add Loopback1 to the VPN instance on PE1 and PE4, and import direct routes to the BGP instance.

## 8.1.2.2 Configuration Procedure

Step 1 Complete basic device configuration.

Configure the command validation mode as immediate validation and configure IP addresses for interconnection and Loopback0 interfaces. Loopback0 interface addresses need to be configured according to the following table.

**Table 8-1 Loopback0 IP addresses**

| Device ID | X value | Loopback0 IP Address |
|---|---|---|
| PE1 | 1 | 10.0.1.1 |
| PE2 | 2 | 10.0.2.2 |
| PE3 | 3 | 10.0.3.3 |
| PE4 | 4 | 10.0.4.4 |
| P1 | 5 | 10.0.5.5 |
| P2 | 6 | 10.0.6.6 |

# Name the devices.

N/A

# Disable unnecessary interfaces in this lab.

N/A

# Configure the command validation mode as immediate validation.

```
<PE1>system-view immediately
```

```
<PE2>system-view immediately
```

```
<PE3>system-view immediately
```

```
<PE4>system-view immediately
```

```
<P1>system-view immediately
```

```
<P2>system-view immediately
```

# Disable the DCN function globally on all devices.

```
[PE1] undo dcn
Warning: This operation will disable DCN function. Continue? [Y/N]:y
```

Here, PE1 is used as an example. Repeat the same operations for other devices.

By default, the DCN function is enabled on NE router interfaces. To facilitate this lab, disable the DCN function globally on all devices.

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE1.

```
[PE1]interface LoopBack0
[PE1-LoopBack0] ip address 10.0.1.1 32
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0]ip address 10.0.0.1 30
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
[PE1-GigabitEthernet0/5/1]ip address 10.0.0.5 30
[PE1-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE2.

```
[PE2]interface LoopBack0
[PE2-LoopBack0] ip address 10.0.2.2 32
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ip address 10.0.0.2 30
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ip address 10.0.0.9 30
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE3.

```
[PE3]interface LoopBack0
[PE3-LoopBack0] ip address 10.0.3.3 32
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] ip address 10.0.0.25 30
[PE3-GigabitEthernet0/5/0] quit
[PE3]interface GigabitEthernet0/5/1
[PE3-GigabitEthernet0/5/1] ip address 10.0.0.18 30
[PE3-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE4.

```
[PE4]interface LoopBack0
[PE4-LoopBack0] ip address 10.0.4.4 32
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] ip address 10.0.0.26 30
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] undo shutdown
[PE4-GigabitEthernet0/5/1] ip address 10.0.0.22 255.255.255.252
[PE4-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on P1.

```
[P1]interface LoopBack0
[P1-LoopBack0] ip address 10.0.5.5 32
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] ip address 10.0.0.13 30
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] ip address 10.0.0.6 30
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] ip address 10.0.0.17 30
[P1-GigabitEthernet0/5/2] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on P2.

```
[P2]interface LoopBack0
[P2-LoopBack0] ip address 10.0.6.6 32
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ip address 10.0.0.14 30
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ip address 10.0.0.10 30
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] undo shutdown
[P2-GigabitEthernet0/5/2] ip address 10.0.0.21 255.255.255.252
```

Step 2 Configure IS-IS.

The IS-IS area ID is 49.0001, the IS-IS process ID is 1, all devices are Level-2 devices, and the NET is **49.0001.000*X*.000*X*.000*X*.00**. The values indicated by *X* are listed in the table of Step 1. Enable IS-IS on Loopback0 and interconnection interfaces.

Note that you need to set **cost-style** to **wide** to support IS-IS extensions.

# Configure IS-IS on PE1.

```
[PE1]isis 1
[PE1-isis-1] is-level level-2
[PE1-isis-1] cost-style wide
[PE1-isis-1] network-entity 49.0001.0001.0001.0001.00
[PE1-isis-1] is-name PE1
[PE1]interface LoopBack0
[PE1-LoopBack0] isis enable 1
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0] isis enable 1
[PE1-GigabitEthernet0/5/0] isis circuit-type p2p
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
[PE1-GigabitEthernet0/5/1] isis enable 1
[PE1-GigabitEthernet0/5/1] isis circuit-type p2p
[PE1-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on PE2.

```
[PE2]isis 1
[PE2-isis-1] is-level level-2
[PE2-isis-1] cost-style wide
[PE2-isis-1] network-entity 49.0001.0002.0002.0002.00
[PE2-isis-1] is-name PE2
[PE2-isis-1]interface LoopBack0
[PE2-LoopBack0] isis enable 1
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] isis enable 1
[PE2-GigabitEthernet0/5/0] isis circuit-type p2p
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] isis enable 1
[PE2-GigabitEthernet0/5/1] isis circuit-type p2p
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on PE3.

```
[PE3]isis 1
[PE3-isis-1] is-level level-2
[PE3-isis-1] cost-style wide
[PE3-isis-1] network-entity 49.0001.0003.0003.0003.00
[PE3-isis-1] is-name PE3
[PE3-isis-1] quit
[PE3]interface LoopBack0
[PE3-LoopBack0] isis enable 1
[PE3-LoopBack0] quit
```

```
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] isis enable 1
[PE3-GigabitEthernet0/5/0] isis circuit-type p2p
[PE3-GigabitEthernet0/5/0] quit
[PE3]interface GigabitEthernet0/5/1
[PE3-GigabitEthernet0/5/1] isis enable 1
[PE3-GigabitEthernet0/5/1] isis circuit-type p2p
[PE3-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on PE4.

```
[PE4]isis 1
[PE4-isis-1] is-level level-2
[PE4-isis-1] cost-style wide
[PE4-isis-1] network-entity 49.0001.0004.0004.0004.00
[PE4-isis-1] is-name PE4
[PE4-isis-1] quit
[PE4]interface LoopBack0
[PE4-LoopBack0] isis enable 1
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] isis enable 1
[PE4-GigabitEthernet0/5/0] isis circuit-type p2p
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] isis enable 1
[PE4-GigabitEthernet0/5/1] isis circuit-type p2p
[PE4-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on P1.

```
[P1]isis 1
[P1-isis-1] is-level level-2
[P1-isis-1] cost-style wide
[P1-isis-1] network-entity 49.0001.0005.0005.0005.00
[P1-isis-1] is-name P1
[P1-isis-1] quit
[P1]interface LoopBack0
[P1-LoopBack0] isis enable 1
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] isis enable 1
[P1-GigabitEthernet0/5/0] isis circuit-type p2p
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] isis enable 1
[P1-GigabitEthernet0/5/1] isis circuit-type p2p
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] isis enable 1
[P1-GigabitEthernet0/5/2] isis circuit-type p2p
[P1-GigabitEthernet0/5/2] quit
```

# Configure IS-IS on P2.

```
[P2]isis 1
[P2-isis-1] is-level level-2
[P2-isis-1] cost-style wide
[P2-isis-1] network-entity 49.0001.0006.0006.0006.00
[P2-isis-1] is-name P2
[P2-isis-1] quit
[P2]interface LoopBack0
[P2-LoopBack0] isis enable 1
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] isis enable 1
[P2-GigabitEthernet0/5/0] isis circuit-type p2p
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] isis enable 1
[P2-GigabitEthernet0/5/1] isis circuit-type p2p
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] isis enable 1
[P2-GigabitEthernet0/5/2] isis circuit-type p2p
[P2-GigabitEthernet0/5/2] quit
```

# Check the IS-IS neighbor relationship on P1, PE2, and PE4.

```
[P1]display isis peer

                    Peer information for ISIS(1)

  System Id     Interface          Circuit Id          State HoldTime  Type     PRI
------------------------------------------------------------------------------
P2              GE0/5/0            0000000006          Up    26s       L2       --
PE1             GE0/5/1            0000000007          Up    25s       L2       --
PE3             GE0/5/2            0000000007          Up    27s       L2       --

Total Peer(s): 3
```

```
[PE2]display isis peer

                    Peer information for ISIS(1)

  System Id     Interface          Circuit Id          State HoldTime  Type     PRI
------------------------------------------------------------------------------
PE1             GE0/5/0            0000000006          Up    26s       L2       --
P2              GE0/5/1            0000000007          Up    25s       L2       --

Total Peer(s): 2
```

```
[PE4]display isis peer

                    Peer information for ISIS(1)
```

```
  System Id      Interface          Circuit Id              State HoldTime  Type       PRI
------------------------------------------------------------------------------
PE3              GE0/5/0            0000000006              Up    23s       L2         --
P2               GE0/5/1            0000000007              Up    23s       L2         --

Total Peer(s): 2
```

# Check IS-IS routes on P1.

```
[P1]display isis route

                    Routeinformation forISIS(1)
                    ---------------------------

                   ISIS(1) Level-2Forwarding Table
                   -------------------------------

IPV4Destination   IntCost    ExtCost ExitInterface     NextHop         Flags
-------------------------------------------------------------------------------
10.0.0.0/30       20         NULL    GE0/5/1           10.0.0.5        A/-/-/-
10.0.0.4/30       10         NULL    GE0/5/1           Direct          D/-/L/-
10.0.0.8/30       20         NULL    GE0/5/0           10.0.0.14       A/-/-/-
10.0.0.12/30      10         NULL    GE0/5/0           Direct          D/-/L/-
10.0.0.16/30      10         NULL    GE0/5/2           Direct          D/-/L/-
10.0.0.20/30      20         NULL    GE0/5/0           10.0.0.14       A/-/-/-
10.0.0.24/30      20         NULL    GE0/5/2           10.0.0.18       A/-/-/-
10.0.1.1/32       10         NULL    GE0/5/1           10.0.0.5        A/-/-/-
10.0.2.2/32       20         NULL    GE0/5/1           10.0.0.5        A/-/-/-
                                     GE0/5/0           10.0.0.14
10.0.3.3/32       10         NULL    GE0/5/2           10.0.0.18       A/-/-/-
10.0.4.4/32       20         NULL    GE0/5/2           10.0.0.18       A/-/-/-
                                     GE0/5/0           10.0.0.14
10.0.5.5/32       0          NULL    Loop0             Direct          D/-/L/-
10.0.6.6/32       10         NULL    GE0/5/0           10.0.0.14       A/-/-/-
    Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
           U-Up/Down Bit Set, LP-Local Prefix-Sid
    Protect Type: L-Link Protect, N-Node Protect
```

Route learning has completed.

Step 3 Configure MPLS.

Enable MPLS on all devices and configure MPLS LSR IDs (by using the IP addresses of Loopback0). MPLS does not need to be enabled on interfaces.

```
[PE1]mpls lsr-id 10.0.1.1
[PE1]mpls
```

```
[PE2]mpls lsr-id 10.0.2.2
[PE2]mpls
```

```
[PE3]mpls lsr-id 10.0.3.3
[PE3]mpls
```

```
[PE4]mpls lsr-id 10.0.4.4
[PE4]mpls
```

```
[P1]mpls lsr-id 10.0.5.5
[P1]mpls
```

```
[P2]mpls lsr-id 10.0.6.6
[P2]mpls
```

Step 4 Configure SR capabilities on devices.

Enable SR-MPLS globally, enable IS-IS extensions for SR, configure an SRGB for IS-IS, and set the SRGB range to 16000 to 17000 on all devices.

Configure a SID for Loopback0 and specify the relative label value *X* using the **index** parameter. The values indicated by *X* are listed in the table of Step 1.

# Configure PE1.

```
[PE1] segment-routing
[PE1-segment-routing] quit
[PE1] isis 1
[PE1-isis-1] segment-routing mpls
[PE1-isis-1] segment-routing global-block 16000 17000
[PE1-isis-1] quit
[PE1] interface LoopBack 0
[PE1-LoopBack0] isis prefix-sid index 1
[PE1-LoopBack0] quit
```

# Configure PE2.

```
[PE2] segment-routing
[PE2-segment-routing] quit
[PE2] isis 1
[PE2-isis-1] segment-routing mpls
[PE2-isis-1] segment-routing global-block 16000 17000
[PE2-isis-1] quit
[PE2] interface LoopBack 0
[PE2-LoopBack0] isis prefix-sid index 2
[PE2-LoopBack0] quit
```

# Configure PE3.

```
[PE3] segment-routing
[PE3-segment-routing] quit
[PE3] isis 1
[PE3-isis-1] segment-routing mpls
[PE3-isis-1] segment-routing global-block 16000 17000
[PE3-isis-1] quit
[PE3] interface LoopBack 0
[PE3-LoopBack0] isis prefix-sid index 3
[PE3-LoopBack0] quit
```

# Configure PE4.

```
[PE4] segment-routing
[PE4-segment-routing] quit
[PE4] isis 1
[PE4-isis-1] segment-routing mpls
[PE4-isis-1] segment-routing global-block 16000 17000
[PE4-isis-1] quit
[PE4] interface LoopBack 0
[PE4-LoopBack0] isis prefix-sid index 4
[PE4-LoopBack0] quit
```

# Configure P1.

```
[P1] segment-routing
[P1-segment-routing] quit
[P1] isis 1
[P1-isis-1] segment-routing mpls
[P1-isis-1] segment-routing global-block 16000 17000
[P1-isis-1] quit
[P1] interface LoopBack 0
[P1-LoopBack0] isis prefix-sid index 5
[P1-LoopBack0] quit
```

# Configure P2.

```
[P2] segment-routing
[P2-segment-routing] quit
[P2] isis 1
[P2-isis-1] segment-routing mpls
[P2-isis-1] segment-routing global-block 16000 17000
[P2-isis-1] quit
[P2] interface LoopBack 0
[P2-LoopBack0] isis prefix-sid index 6
[P2-LoopBack0] quit
```

# Run the **display tunnel-info all** command on PE1 to check SR LSP establishment.

```
[PE1]display tunnel-info all
Tunnel ID                Type             Destination      Status
--------------------------------------------------------------------------------
0x000000002900000004 srbe-lsp         10.0.2.2         UP
0x000000002900000005 srbe-lsp         10.0.3.3         UP
0x000000002900000006 srbe-lsp         10.0.4.4         UP
```

```
0x000000002900000007 srbe-lsp          10.0.5.5              UP
0x000000002900000008 srbe-lsp          10.0.6.6              UP
```

The SR LSP to PE4 has been established.

# Check the SR label forwarding table.

```
[PE1]display segment-routing prefix mpls forwarding
              Segment Routing Prefix MPLS Forwarding Information
          ------------------------------------------------------------
          Role: I-Ingress, T-Transit, E-Egress, I&T-Ingress And Transit

Prefix           Label     OutLabel   Interface  NextHop      Role MPLSMtu   Mtu     State
-------------------------------------------------------------------------------------------------------------------
10.0.1.1/32      16001     NULL       Loop0      127.0.0.1    E    ---       1500    Active
10.0.2.2/32      16002     16002      GE0/5/0    10.0.0.2     I&T  ---       1500    Active
10.0.3.3/32      16003     16003      GE0/5/1    10.0.0.6     I&T  ---       1500    Active
10.0.4.4/32      16004     16004      GE0/5/0    10.0.0.2     I&T  ---       1500    Active
10.0.4.4/32      16004     16004      GE0/5/1    10.0.0.6     I&T  ---       1500    Active
10.0.5.5/32      16005     16005      GE0/5/1    10.0.0.6     I&T  ---       1500    Active
10.0.6.6/32      16006     16006      GE0/5/0    10.0.0.2     I&T  ---       1500    Active
10.0.6.6/32      16006     16006      GE0/5/1    10.0.0.6     I&T  ---       1500    Active

Total information(s): 8
```

# Check the connectivity of the CR LSP from PE1 to PE4.

```
[PE1]ping lsp segment-routing ip 10.0.4.4 32 version draft2
  LSP PING FEC: SEGMENT ROUTING IPV4 PREFIX 10.0.4.4/32 : 100   data bytes, press CTRL_C to
break
    Reply from 10.0.4.4: bytes=100 Sequence=1 time=12 ms
    Reply from 10.0.4.4: bytes=100 Sequence=2 time=3 ms
    Reply from 10.0.4.4: bytes=100 Sequence=3 time=3 ms
    Reply from 10.0.4.4: bytes=100 Sequence=4 time=3 ms
    Reply from 10.0.4.4: bytes=100 Sequence=5 time=3 ms

  --- FEC: SEGMENT ROUTING IPV4 PREFIX 10.0.4.4/32 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 3/4/12 ms
```

The communication is normal.

## Step 5 Configure L3VPN.

On both PE1 and PE4, create a VPN instance named **vpna**, create Loopback1, add the interface to the VPN instance, and establish an MP-BGP EVPN peer relationship between PE1 and P1, between PE1 and P2, between PE4 and P1, and between PE4 and P2 (the AS number is 65100). P1 and P2 function as RRs. PE1 and PE4 function as RR clients, which advertise VPNv4 routes through P1 and P2.

# Create a VPN instance named **vpna**.

```
[PE1]ip vpn-instance vpna
```

```
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:10
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 100:1020 both
[PE1-vpn-instance-vpna-af-ipv4] quit
```

```
[PE4]ip vpn-instance vpna
[PE4-vpn-instance-vpna] ipv4-family
[PE4-vpn-instance-vpna-af-ipv4] route-distinguisher 100:40
[PE4-vpn-instance-vpna-af-ipv4] vpn-target 100:1020 both
[PE4-vpn-instance-vpna-af-ipv4] quit
```

# Create Loopback1, associate it with the VPN instance, and configure an IP address for the interface.

```
[PE1]interface LoopBack 1
[PE1-LoopBack1]ip binding vpn-instance vpna
Info: All IPv4 and IPv6 related configurations on this interface are removed.
[PE1-LoopBack1]ip address 10.1.1.1 32
```

```
[PE4]interface LoopBack 1
[PE4-LoopBack1]ip binding vpn-instance vpna
Info: All IPv4 and IPv6 related configurations on this interface are removed.
[PE4-LoopBack1]ip address 10.1.4.4 32
```

Note that you need to associate the interface with the VPN instance before configuring an IP address for the interface.

# Configure the MP-BGP VPNv4 peer relationships through Loopback0 and use the Loopback0 address as the router ID.

```
[PE1]bgp 65100
[PE1-bgp] router-id 10.0.1.1
[PE1-bgp] peer 10.0.5.5 as-number 65100
[PE1-bgp] peer 10.0.5.5 connect-interface LoopBack0
[PE1-bgp] peer 10.0.6.6 as-number 65100
[PE1-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE1-bgp] #
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 10.0.5.5 enable
[PE1-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[PE2]bgp 65100
[PE2-bgp] router-id 10.0.2.2
[PE2-bgp] peer 10.0.5.5 as-number 65100
[PE2-bgp] peer 10.0.5.5 connect-interface LoopBack0
[PE2-bgp] peer 10.0.6.6 as-number 65100
[PE2-bgp] peer 10.0.6.6 connect-interface LoopBack0
```

```
[PE2-bgp] #
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 10.0.5.5 enable
[PE2-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[PE3-bgp] router-id 10.0.3.3
[PE3-bgp] peer 10.0.5.5 as-number 65100
[PE3-bgp] peer 10.0.5.5 connect-interface LoopBack0
[PE3-bgp] peer 10.0.6.6 as-number 65100
[PE3-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE3-bgp] #
[PE3-bgp] ipv4-family vpnv4
[PE3-bgp-af-vpnv4] peer 10.0.5.5 enable
[PE3-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[PE4]bgp 65100
[PE4-bgp] router-id 10.0.4.4
[PE4-bgp] peer 10.0.5.5 as-number 65100
[PE4-bgp] peer 10.0.5.5 connect-interface LoopBack0
[PE4-bgp] peer 10.0.6.6 as-number 65100
[PE4-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE4-bgp] #
[PE4-bgp] ipv4-family vpnv4
[PE4-bgp-af-vpnv4] peer 10.0.5.5 enable
[PE4-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[P1]bgp 65100
[P1-bgp] router-id 10.0.5.5
[P1-bgp] peer 10.0.1.1 as-number 65100
[P1-bgp] peer 10.0.1.1 connect-interface LoopBack0
[P1-bgp] peer 10.0.2.2 as-number 65100
[P1-bgp] peer 10.0.2.2 connect-interface LoopBack0
[P1-bgp] peer 10.0.3.3 as-number 65100
[P1-bgp] peer 10.0.3.3 connect-interface LoopBack0
[P1-bgp] peer 10.0.4.4 as-number 65100
[P1-bgp] peer 10.0.4.4 connect-interface LoopBack0
[P1-bgp] #
[P1-bgp] ipv4-family vpnv4
[P1-bgp-af-vpnv4] undo policy vpn-target
[P1-bgp-af-vpnv4] peer 10.0.1.1 enable
[P1-bgp-af-vpnv4] peer 10.0.1.1 reflect-client
[P1-bgp-af-vpnv4] peer 10.0.2.2 enable
[P1-bgp-af-vpnv4] peer 10.0.2.2 reflect-client
[P1-bgp-af-vpnv4] peer 10.0.3.3 enable
[P1-bgp-af-vpnv4] peer 10.0.3.3 reflect-client
[P1-bgp-af-vpnv4] peer 10.0.4.4 enable
[P1-bgp-af-vpnv4] peer 10.0.4.4 reflect-client
```

When configuring an RR, disable the RT check on VPNv4 routes.

```
[P2]bgp 65100
[P2-bgp] router-id 10.0.6.6
[P2-bgp] peer 10.0.1.1 as-number 65100
[P2-bgp] peer 10.0.1.1 connect-interface LoopBack0
[P2-bgp] peer 10.0.2.2 as-number 65100
[P2-bgp] peer 10.0.2.2 connect-interface LoopBack0
[P2-bgp] peer 10.0.3.3 as-number 65100
[P2-bgp] peer 10.0.3.3 connect-interface LoopBack0
[P2-bgp] peer 10.0.4.4 as-number 65100
[P2-bgp] peer 10.0.4.4 connect-interface LoopBack0
[P2-bgp] #
[P2-bgp] ipv4-family vpnv4
[P2-bgp-af-vpnv4] undo policy vpn-target
[P2-bgp-af-vpnv4] peer 10.0.1.1 enable
[P2-bgp-af-vpnv4] peer 10.0.1.1 reflect-client
[P2-bgp-af-vpnv4] peer 10.0.2.2 enable
[P2-bgp-af-vpnv4] peer 10.0.2.2 reflect-client
[P2-bgp-af-vpnv4] peer 10.0.3.3 enable
[P2-bgp-af-vpnv4] peer 10.0.3.3 reflect-client
[P2-bgp-af-vpnv4] peer 10.0.4.4 enable
[P2-bgp-af-vpnv4] peer 10.0.4.4 reflect-client
```

# Check the VPNv4 peer relationship status on P1 and P2.

```
[P1]display bgp vpnv4 all peer
BGP local router ID : 10.0.5.5
Local AS number : 65100
Total number of peers : 4                 Peers in established state : 4

 Peer            V         AS  MsgRcvd  MsgSent  OutQ  Up/Down       State          PrefRcv
 10.0.1.1        4      65100        6        6     0  00:02:07   Established            0
 10.0.2.2        4      65100        6        6     0  00:02:05   Established            0
 10.0.3.3        4      65100        6        6     0  00:02:06   Established            0
 10.0.4.4        4      65100        6        6     0  00:02:04   Established            0
```

```
[P2]display bgp vpnv4 all peer
BGP local router ID : 10.0.6.6
Local AS number : 65100
Total number of peers : 4                 Peers in established state : 4

 Peer            V         AS  MsgRcvd  MsgSent  OutQ  Up/Down       State          PrefRcv
 10.0.1.1        4      65100        6        6     0  00:02:07   Established            0
 10.0.2.2        4      65100        6        6     0  00:02:05   Established            0
 10.0.3.3        4      65100        6        6     0  00:02:06   Established            0
 10.0.4.4        4      65100        6        6     0  00:02:04   Established            0
```

# Import the direct route of Loopback1 to BGP so that PE2 and PE4 can both learn the route of Loopback1 from the peer PE.

```
[PE1] bgp 65100
```

```
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-vpna] import-route direct
```

```
[PE4] bgp 65100
[PE4-bgp] ipv4-family vpn-instance vpna
[PE4-bgp-vpna] import-route direct
```

# Check VPNv4 routes on PE1.

```
[PE1]display bgp vpnv4 all routing-table | include 10.1.4.4
 BGP Local router ID is 10.0.1.1
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete
 RPKI validation codes: V - valid, I - invalid, N - not-found

 Total number of routes from all PE: 4
 Route Distinguisher: 100:10

        Network            NextHop          MED        LocPrf    PrefVal Path/Ogn

 Route Distinguisher: 100:40

        Network            NextHop          MED        LocPrf    PrefVal Path/Ogn
*>i     10.1.4.4/32        10.0.4.4           0          100          0         ?

 VPN-Instance vpna, Router ID 10.0.1.1:

 Total Number of Routes: 4
        Network            NextHop          MED        LocPrf    PrefVal Path/Ogn
*>i     10.1.4.4/32        10.0.4.4           0          100          0         ?
```

PE1 has learned the VPNv4 route from PE4 through MP-BGP.

# Check the IP routing table on PE1.

```
[PE1]display ip routing-table vpn-instance vpna
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
         Destinations : 4        Routes : 4

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        10.1.1.1/32  Direct  0    0           D   127.0.0.1       LoopBack1
        10.1.4.4/32  IBGP    255  0           RD  10.0.4.4        GigabitEthernet0/5/0
                     IBGP    255  0           RD  10.0.4.4        GigabitEthernet0/5/1
      127.0.0.0/8    Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

The equal-cost routes to the network segment of the peer CE have been loaded to the VPN instance routing table on PE1.

# Check route details.

```
[PE1]display ip routing-table vpn-instance vpna 10.1.4.4 verbose
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
Summary Count : 1

Destination:      10.1.4.4/32
       Protocol:   IBGP                 Process ID   : 0
    Preference:   255                        Cost    : 0
       NextHop:   10.0.4.4               Neighbour : 10.0.5.5
         State:   Active Adv Relied           Age : 00h04m31s
           Tag:   0                      Priority : low
         Label:   2141                   QoSInfo : 0x0
     IndirectID:  0x10000BA              Instance :
RelayNextHop:   10.0.0.2                Interface : GigabitEthernet0/5/0
      TunnelID:  0x000000002900000006    Flags : RD
RelayNextHop:   10.0.0.6                Interface : GigabitEthernet0/5/0
      TunnelID:  0x000000002900000006    Flags : RD
```

The tunnel ID can be found. Based on previous tunnel information, you can determine that the tunnel is an SR-MPLS BE tunnel.

# Check the connectivity between Loopback1 on PE1 and Loopback1 on PE4.

```
[PE1]ping -vpn-instance vpna -a 10.1.1.1 10.1.4.4
  PING 10.1.4.4: 56  data bytes, press CTRL_C to break
    Reply from 10.1.4.4: bytes=56 Sequence=1 ttl=253 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=2 ttl=253 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=3 ttl=253 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=4 ttl=253 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=5 ttl=253 time=1 ms

  --- 10.1.4.4 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

The communication is normal.

**----End**

## 8.1.3 Quiz

In an L3VPNv4 over SR-MPLS BE scenario, will the outer label change during packet forwarding?

# 8.2 L3VPNv4 over SR-MPLS TE

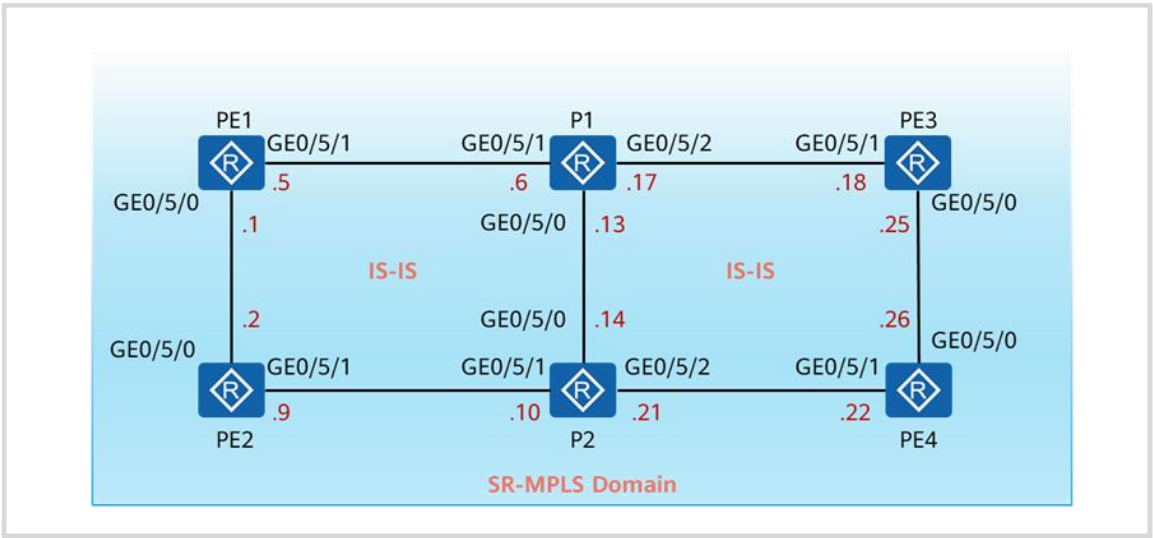
## 8.2.1 About This Lab

### 8.2.1.1 Objectives

- Configure IS-IS to ensure that PEs are routable to each other.
- Manually configure SR-MPLS TE tunnels.
- Recurse L3VPN tunnels used for communication between CEs to SR-MPLS TE tunnels.
- Observe label changes in packets forwarded through an SR-MPLS TE tunnel.

### 8.2.1.2 Networking Description

**Figure 8-2 L3VPNv4 over SR-MPLS TE topology**




The figure shows the device connection and IP address planning. The interface interconnection addresses are in the format of 10.0.0.*Y*/30, and the values represented by Y are shown in the figure. Loopback0 is created on all devices, and loopback interface IP addresses are in the format of 10.0.*X.X*. The values indicated by *X* are listed in the table of the corresponding step.

Loopback1 is created on PE1 and PE4, with addresses being 10.1.1.1/32 and 10.1.4.4/32, respectively, to simulate CE user access.

## 8.2.2 Lab Task

### 8.2.2.1 Configuration Roadmap

1. Configure basic IP addresses for devices.
2. Configure IS-IS in the SR domain. Specifically, enable IS-IS on interconnection and Loopback0 interfaces for communication in the SR domain.
3. Configure MPLS. Specifically, enable MPLS and MPLS TE and set MPLS LSR IDs on devices.
4. Configure SR. Specifically, enable SR globally, enable IS-IS extensions for SR and the traffic engineering capability, and configure node SIDs.

5. Configure explicit paths and TE tunnel interfaces on PE1 and PE4.
6. Configure a VPN instance, add Loopback1 on PE1 and PE4 to the instance, and establish a VPNv4 peer relationship between PE1 and P1, between PE1 and P2, between PE4 and P1, and between PE4 and P2.
7. Configure a tunnel selection policy to recurse VPN traffic to TE tunnels.

### 8.2.2.2 Configuration Procedure

Step 1 Complete basic device configuration.

Configure the command validation mode as immediate validation and configure IP addresses for interconnection and Loopback0 interfaces. Loopback0 interface addresses need to be configured according to the following table.

**Table 8-2 Loopback0 IP addresses**

| Device ID | X value | Loopback0 IP Address |
|---|---|---|
| PE1 | 1 | 10.0.1.1 |
| PE2 | 2 | 10.0.2.2 |
| PE3 | 3 | 10.0.3.3 |
| PE4 | 4 | 10.0.4.4 |
| P1 | 5 | 10.0.5.5 |
| P2 | 6 | 10.0.6.6 |

# Name the devices.

N/A

# Disable unnecessary interfaces in this lab.

N/A

# Configure the command validation mode as immediate validation.

```
<PE1>system-view immediately
```

```
<PE2>system-view immediately
```

```
<PE3>system-view immediately
```

```
<PE4>system-view immediately
```

```
<P1>system-view immediately
```

```
<P2>system-view immediately
```

# Disable the DCN function globally on all devices.

```
[PE1] undo dcn
Warning: This operation will disable DCN function. Continue? [Y/N]:y
```

Here, PE1 is used as an example. Repeat the same operations for other devices.

By default, the DCN function is enabled on NE router interfaces. To facilitate this lab, disable the DCN function globally on all devices.

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE1.

```
[PE1]interface LoopBack0
[PE1-LoopBack0] ip address 10.0.1.1 32
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0]ip address 10.0.0.1 30
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
[PE1-GigabitEthernet0/5/1]ip address 10.0.0.5 30
[PE1-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE2.

```
[PE2]interface LoopBack0
[PE2-LoopBack0] ip address 10.0.2.2 32
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ip address 10.0.0.2 30
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ip address 10.0.0.9 30
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE3.

```
[PE3]interface LoopBack0
[PE3-LoopBack0] ip address 10.0.3.3 32
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] ip address 10.0.0.25 30
[PE3-GigabitEthernet0/5/0] quit
[PE3]interface GigabitEthernet0/5/1
[PE3-GigabitEthernet0/5/1] ip address 10.0.0.18 30
[PE3-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE4.

```
[PE4]interface LoopBack0
[PE4-LoopBack0] ip address 10.0.4.4 32
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] ip address 10.0.0.26 30
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] undo shutdown
[PE4-GigabitEthernet0/5/1] ip address 10.0.0.22 255.255.255.252
[PE4-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on P1.

```
[P1]interface LoopBack0
[P1-LoopBack0] ip address 10.0.5.5 32
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] ip address 10.0.0.13 30
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] ip address 10.0.0.6 30
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] ip address 10.0.0.17 30
[P1-GigabitEthernet0/5/2] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on P2.

```
[P2]interface LoopBack0
[P2-LoopBack0] ip address 10.0.6.6 32
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ip address 10.0.0.14 30
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ip address 10.0.0.10 30
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] undo shutdown
[P2-GigabitEthernet0/5/2] ip address 10.0.0.21 255.255.255.252
```

## Step 2 Configure IS-IS.

The IS-IS area ID is 49.0001, the IS-IS process ID is 1, all devices are Level-2 devices, and the NET is **49.0001.000*X*.000*X*.000*X*.00**. The values indicated by *X* are listed in the table of Step 1. Enable IS-IS on Loopback0 and interconnection interfaces.

Note that you need to set **cost-style** to **wide** to support IS-IS extensions.

# Configure IS-IS on PE1.

```
[PE1]isis 1
[PE1-isis-1] is-level level-2
```

```
[PE1-isis-1] cost-style wide
[PE1-isis-1] network-entity 49.0001.0001.0001.0001.00
[PE1-isis-1] is-name PE1
[PE1]interface LoopBack0
[PE1-LoopBack0] isis enable 1
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0] isis enable 1
[PE1-GigabitEthernet0/5/0] isis circuit-type p2p
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
[PE1-GigabitEthernet0/5/1] isis enable 1
[PE1-GigabitEthernet0/5/1] isis circuit-type p2p
[PE1-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on PE2.

```
[PE2]isis 1
[PE2-isis-1] is-level level-2
[PE2-isis-1] cost-style wide
[PE2-isis-1] network-entity 49.0001.0002.0002.0002.00
[PE2-isis-1] is-name PE2
[PE2-isis-1]interface LoopBack0
[PE2-LoopBack0] isis enable 1
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] isis enable 1
[PE2-GigabitEthernet0/5/0] isis circuit-type p2p
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] isis enable 1
[PE2-GigabitEthernet0/5/1] isis circuit-type p2p
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on PE3.

```
[PE3]isis 1
[PE3-isis-1] is-level level-2
[PE3-isis-1] cost-style wide
[PE3-isis-1] network-entity 49.0001.0003.0003.0003.00
[PE3-isis-1] is-name PE3
[PE3-isis-1] quit
[PE3]interface LoopBack0
[PE3-LoopBack0] isis enable 1
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] isis enable 1
[PE3-GigabitEthernet0/5/0] isis circuit-type p2p
[PE3-GigabitEthernet0/5/0] quit
[PE3]interface GigabitEthernet0/5/1
[PE3-GigabitEthernet0/5/1] isis enable 1
[PE3-GigabitEthernet0/5/1] isis circuit-type p2p
[PE3-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on PE4.

```
[PE4]isis 1
[PE4-isis-1] is-level level-2
[PE4-isis-1] cost-style wide
[PE4-isis-1] network-entity 49.0001.0004.0004.0004.00
[PE4-isis-1] is-name PE4
[PE4-isis-1] quit
[PE4]interface LoopBack0
[PE4-LoopBack0] isis enable 1
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] isis enable 1
[PE4-GigabitEthernet0/5/0] isis circuit-type p2p
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] isis enable 1
[PE4-GigabitEthernet0/5/1] isis circuit-type p2p
[PE4-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on P1.

```
[P1]isis 1
[P1-isis-1] is-level level-2
[P1-isis-1] cost-style wide
[P1-isis-1] network-entity 49.0001.0005.0005.0005.00
[P1-isis-1] is-name P1
[P1-isis-1] quit
[P1]interface LoopBack0
[P1-LoopBack0] isis enable 1
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] isis enable 1
[P1-GigabitEthernet0/5/0] isis circuit-type p2p
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] isis enable 1
[P1-GigabitEthernet0/5/1] isis circuit-type p2p
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] isis enable 1
[P1-GigabitEthernet0/5/2] isis circuit-type p2p
[P1-GigabitEthernet0/5/2] quit
```

# Configure IS-IS on P2.

```
[P2]isis 1
[P2-isis-1] is-level level-2
[P2-isis-1] cost-style wide
[P2-isis-1] network-entity 49.0001.0006.0006.0006.00
[P2-isis-1] is-name P2
[P2-isis-1] quit
[P2]interface LoopBack0
[P2-LoopBack0] isis enable 1
```

```
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] isis enable 1
[P2-GigabitEthernet0/5/0] isis circuit-type p2p
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] isis enable 1
[P2-GigabitEthernet0/5/1] isis circuit-type p2p
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] isis enable 1
[P2-GigabitEthernet0/5/2] isis circuit-type p2p
[P2-GigabitEthernet0/5/2] quit
```

# Check the IS-IS neighbor relationship on P1, PE2, and PE4.

```
[P1]display isis peer

                    Peer information for ISIS(1)

  System Id     Interface          Circuit Id         State HoldTime  Type     PRI
------------------------------------------------------------------------------
P2              GE0/5/0            0000000006         Up    26s       L2       --
PE1             GE0/5/1            0000000007         Up    25s       L2       --
PE3             GE0/5/2            0000000007         Up    27s       L2       --

Total Peer(s): 3
```

```
[PE2]display isis peer

                    Peer information for ISIS(1)

  System Id     Interface          Circuit Id         State HoldTime  Type     PRI
------------------------------------------------------------------------------
PE1             GE0/5/0            0000000006         Up    26s       L2       --
P2              GE0/5/1            0000000007         Up    25s       L2       --

Total Peer(s): 2
```

```
[PE4]display isis peer

                    Peer information for ISIS(1)

  System Id     Interface          Circuit Id         State HoldTime  Type     PRI
------------------------------------------------------------------------------
PE3             GE0/5/0            0000000006         Up    23s       L2       --
P2              GE0/5/1            0000000007         Up    23s       L2       --

Total Peer(s): 2
```

# Check IS-IS routes on P1.

```
[P1]display isis route

                  Routeinformation forISIS(1)
                  ---------------------------

                 ISIS(1) Level-2Forwarding Table
                 -------------------------------

IPV4Destination   IntCost    ExtCost ExitInterface     NextHop         Flags
--------------------------------------------------------------------------------
10.0.0.0/30       20          NULL    GE0/5/1           10.0.0.5        A/-/-/-
10.0.0.4/30       10          NULL    GE0/5/1           Direct          D/-/L/-
10.0.0.8/30       20          NULL    GE0/5/0           10.0.0.14       A/-/-/-
10.0.0.12/30      10          NULL    GE0/5/0           Direct          D/-/L/-
10.0.0.16/30      10          NULL    GE0/5/2           Direct          D/-/L/-
10.0.0.20/30      20          NULL    GE0/5/0           10.0.0.14       A/-/-/-
10.0.0.24/30      20          NULL    GE0/5/2           10.0.0.18       A/-/-/-
10.0.1.1/32       10          NULL    GE0/5/1           10.0.0.5        A/-/-/-
10.0.2.2/32       20          NULL    GE0/5/1           10.0.0.5        A/-/-/-
                                      GE0/5/0           10.0.0.14
10.0.3.3/32       10          NULL    GE0/5/2           10.0.0.18       A/-/-/-
10.0.4.4/32       20          NULL    GE0/5/2           10.0.0.18       A/-/-/-
                                      GE0/5/0           10.0.0.14
10.0.5.5/32       0           NULL    Loop0             Direct          D/-/L/-
10.0.6.6/32       10          NULL    GE0/5/0           10.0.0.14       A/-/-/-
    Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
           U-Up/Down Bit Set, LP-Local Prefix-Sid
    Protect Type: L-Link Protect, N-Node Protect
```

Route learning has completed.

## Step 3 Configure MPLS.

Enable MPLS on all devices and configure MPLS LSR IDs (by using the IP addresses of Loopback0). MPLS does not need to be enabled on interfaces.

```
[PE1]mpls lsr-id 10.0.1.1
[PE1]mpls
```

```
[PE2]mpls lsr-id 10.0.2.2
[PE2]mpls
```

```
[PE3]mpls lsr-id 10.0.3.3
[PE3]mpls
```

```
[PE4]mpls lsr-id 10.0.4.4
```

```
[PE4]mpls
```

```
[P1]mpls lsr-id 10.0.5.5
[P1]mpls
```

```
[P2]mpls lsr-id 10.0.6.6
[P2]mpls
```

Step 4 Configure SR capabilities on devices.

Enable SR-MPLS globally, enable IS-IS extensions for SR, configure an SRGB for IS-IS, and set the SRGB range to 16000 to 17000 on all devices.

Configure a SID for Loopback0 and specify the relative label value *X* using the **index** parameter. The values indicated by *X* are listed in the table of Step 1.

# Configure PE1.

```
[PE1] segment-routing
[PE1-segment-routing] quit
[PE1] isis 1
[PE1-isis-1] segment-routing mpls
[PE1-isis-1] segment-routing global-block 16000 17000
[PE1-isis-1] quit
[PE1] interface LoopBack 0
[PE1-LoopBack0] isis prefix-sid index 1
[PE1-LoopBack0] quit
```

# Configure PE2.

```
[PE2] segment-routing
[PE2-segment-routing] quit
[PE2] isis 1
[PE2-isis-1] segment-routing mpls
[PE2-isis-1] segment-routing global-block 16000 17000
[PE2-isis-1] quit
[PE2] interface LoopBack 0
[PE2-LoopBack0] isis prefix-sid index 2
[PE2-LoopBack0] quit
```

# Configure PE3.

```
[PE3] segment-routing
[PE3-segment-routing] quit
[PE3] isis 1
[PE3-isis-1] segment-routing mpls
[PE3-isis-1] segment-routing global-block 16000 17000
[PE3-isis-1] quit
[PE3] interface LoopBack 0
[PE3-LoopBack0] isis prefix-sid index 3
```

```
[PE3-LoopBack0] quit
```

# Configure PE4.

```
[PE4] segment-routing
[PE4-segment-routing] quit
[PE4] isis 1
[PE4-isis-1] segment-routing mpls
[PE4-isis-1] segment-routing global-block 16000 17000
[PE4-isis-1] quit
[PE4] interface LoopBack 0
[PE4-LoopBack0] isis prefix-sid index 4
[PE4-LoopBack0] quit
```

# Configure P1.

```
[P1] segment-routing
[P1-segment-routing] quit
[P1] isis 1
[P1-isis-1] segment-routing mpls
[P1-isis-1] segment-routing global-block 16000 17000
[P1-isis-1] quit
[P1] interface LoopBack 0
[P1-LoopBack0] isis prefix-sid index 5
[P1-LoopBack0] quit
```

# Configure P2.

```
[P2] segment-routing
[P2-segment-routing] quit
[P2] isis 1
[P2-isis-1] segment-routing mpls
[P2-isis-1] segment-routing global-block 16000 17000
[P2-isis-1] quit
[P2] interface LoopBack 0
[P2-LoopBack0] isis prefix-sid index 6
[P2-LoopBack0] quit
```

# Manually configure adjacency SIDs on P1 and P2.

To ensure that the adjacency SIDs specified during explicit path configuration remain unchanged, you are advised to configure static adjacency SIDs. In this way, the SIDs remain unchanged after the device restarts.

The following table lists the static adjacency SIDs to be configured on P1 and P2.

**Table 8-3 Adjacency SIDs**

| Device ID | Local IP Address | Peer IP Address | SID |
|---|---|---|---|
| P1 | 10.0.0.6 | 10.0.0.5 | 142336 |
| P1 | 10.0.0.13 | 10.0.0.14 | 142337 |

| | | | |
|---|---|---|---|
| P1 | 10.0.0.17 | 10.0.0.18 | 142338 |
| P2 | 10.0.0.10 | 10.0.0.9 | 142336 |
| P2 | 10.0.0.14 | 10.0.0.13 | 142337 |
| P2 | 10.0.0.21 | 10.0.0.22 | 142338 |

```
[P1] segment-routing
[P1-segment-routing] ipv4 adjacency local-ip-addr 10.0.0.6 remote-ip-addr 10.0.0.5 sid 142336
[P1-segment-routing] ipv4 adjacency local-ip-addr 10.0.0.13 remote-ip-addr 10.0.0.14 sid 142337
[P1-segment-routing] ipv4 adjacency local-ip-addr 10.0.0.17 remote-ip-addr 10.0.0.18 sid 142338
```

```
[P2] segment-routing
[P2-segment-routing] ipv4 adjacency local-ip-addr 10.0.0.10 remote-ip-addr 10.0.0.9 sid 142336
[P2-segment-routing] ipv4 adjacency local-ip-addr 10.0.0.14 remote-ip-addr 10.0.0.13 sid 142337
[P2-segment-routing] ipv4 adjacency local-ip-addr 10.0.0.21 remote-ip-addr 10.0.0.22 sid 142338
```

Step 5 Configure SR-MPLS TE explicit paths and TE tunnel interfaces.

Configure explicit paths on PE1 and PE4, specify the nodes that the paths must pass through by specifying node and adjacency SIDs, create TE tunnel interfaces on PE1 and PE4, and associate the interfaces with the configured explicit paths.

This experiment is implemented through CLIs and does not involve any controller. In scenarios where a controller is used, the paths are typically computed by the controller.

# Create explicit paths and specify forcible forwarding paths.

```
[PE1] explicit-path PE1_PE4_Manual
[PE1-explicit-path-PE1_PE4_Manual] next sid label 16005 type prefix
[PE1-explicit-path-PE1_PE4_Manual] next sid label 142337 type adjacency
[PE1-explicit-path-PE1_PE4_Manual] next sid label 142338 type adjacency
```

Configure the explicit path PE1_PE4_Manual on PE1 and forcibly enable the path to pass through P1, GE0/5/0 on P1, and GE0/5/2 on P2.

```
[PE4] explicit-path PE4_PE1_Manual
[PE4-explicit-path-PE4_PE1_Manual] next sid label 16003 type prefix
[PE4-explicit-path-PE4_PE1_Manual] next sid label 16005 type prefix
[PE4-explicit-path-PE4_PE1_Manual] next sid label 142337 type adjacency
[PE4-explicit-path-PE4_PE1_Manual] next sid label 142336 type adjacency
[PE4-explicit-path-PE4_PE1_Manual] next sid label 16001 type prefix
```

Configure the explicit path PE4_PE1_Manual on PE4 and forcibly enable the path to pass through PE3, P1, GE0/5/0 on P1, GE0/5/1 on P2, and PE2.

# Enable MPLS TE globally.

```
[PE1]mpls
```

```
[PE1-mpls]mpls te
```

```
[PE4]mpls
[PE4-mpls]mpls te
```

# Create TE tunnel interfaces.

```
[PE1] interface Tunnel10
[PE1-Tunnel10] ip address unnumbered interface LoopBack0
[PE1-Tunnel10] tunnel-protocol mpls te
[PE1-Tunnel10] destination 10.0.4.4
[PE1-Tunnel10] mpls te signal-protocol segment-routing
[PE1-Tunnel10] mpls te tunnel-id 10
[PE1-Tunnel10] mpls te path explicit-path PE1_PE4_Manual
```

Create tunnel interface 10 on PE1, configure the interface to borrow the Loopback0 IP address, set the destination address to 10.0.4.4 (Loopback0 IP address of PE4), and associate the tunnel interface with the explicit path PE1_PE4_Manual.

```
[PE4] interface Tunnel10
[PE4-Tunnel10] ip address unnumbered interface LoopBack0
[PE4-Tunnel10] tunnel-protocol mpls te
[PE4-Tunnel10] destination 10.0.1.1
[PE4-Tunnel10] mpls te signal-protocol segment-routing
[PE4-Tunnel10] mpls te tunnel-id 10
[PE4-Tunnel10] mpls te path explicit-path PE4_PE1_Manual
```

Create tunnel interface 10 on PE4, configure the interface to borrow the Loopback0 IP address, set the destination address to 10.0.1.1 (Loopback0 IP address of PE4), and associate the tunnel interface with the explicit path PE4_PE1_Manual.

# Check SR-MPLS TE tunnel status.

```
[PE1]display tunnel-info all
Tunnel ID                Type             Destination                Status
--------------------------------------------------------------------------------
0x000000000300000001     sr-te            10.0.4.4                   UP
0x000000002900000004     srbe-lsp         10.0.2.2                   UP
0x000000002900000005     srbe-lsp         10.0.3.3                   UP
0x000000002900000006     srbe-lsp         10.0.4.4                   UP
0x000000002900000007     srbe-lsp         10.0.5.5                   UP
0x000000002900000008     srbe-lsp         10.0.6.6                   UP
```

```
[PE4]display tunnel-info all
Tunnel ID                Type             Destination                Status
--------------------------------------------------------------------------------
0x000000000300000001     sr-te            10.0.1.1                   UP
0x000000002900000003     srbe-lsp         10.0.1.1                   UP
0x000000002900000004     srbe-lsp         10.0.2.2                   UP
```

```
0x000000002900000005   srbe-lsp        10.0.4.4          UP
0x000000002900000007   srbe-lsp        10.0.5.5          UP
0x000000002900000008   srbe-lsp        10.0.6.6          UP
```

The tunnel status is normal on PE1 and PE4.

# Check the tunnel connectivity on PE1 and PE4.

```
[PE1]ping lsp segment-routing te Tunnel 10
  LSP PING FEC: SEGMENT ROUTING TE TUNNEL IPV4 SESSION QUERY Tunnel10 : 100   data bytes,
press CTRL_C to break
    Reply from 10.0.4.4: bytes=100 Sequence=1 time=8 ms
    Reply from 10.0.4.4: bytes=100 Sequence=2 time=3 ms
    Reply from 10.0.4.4: bytes=100 Sequence=3 time=3 ms
    Reply from 10.0.4.4: bytes=100 Sequence=4 time=2 ms
    Reply from 10.0.4.4: bytes=100 Sequence=5 time=3 ms

  --- FEC: SEGMENT ROUTING TE TUNNEL IPV4 SESSION QUERY Tunnel10 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 2/3/8 ms
```

```
[PE4]ping lsp segment-routing te Tunnel 10
  LSP PING FEC: SEGMENT ROUTING TE TUNNEL IPV4 SESSION QUERY Tunnel10 : 100   data bytes,
press CTRL_C to break
    Reply from 10.0.1.1: bytes=100 Sequence=1 time=8 ms
    Reply from 10.0.1.1: bytes=100 Sequence=2 time=3 ms
    Reply from 10.0.1.1: bytes=100 Sequence=3 time=3 ms
    Reply from 10.0.1.1: bytes=100 Sequence=4 time=3 ms
    Reply from 10.0.1.1: bytes=100 Sequence=5 time=3 ms

  --- FEC: SEGMENT ROUTING TE TUNNEL IPV4 SESSION QUERY Tunnel10 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 3/4/8 ms
```

The TE tunnel connectivity is normal.

Step 6 Configure L3VPN.

On both PE1 and PE4, create a VPN instance named **vpna**, create Loopback1, add the interface to the VPN instance, and establish an MP-BGP EVPN peer relationship between PE1 and P1, between PE1 and P2, between PE4 and P1, and between PE4 and P2 (the AS number is 65100). P1 and P2 function as RRs. PE1 and PE4 function as RR clients, which advertise VPNv4 routes through P1 and P2.

# Create a VPN instance named **vpna**.

```
[PE1]ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:10
```

```
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 100:1020 both
[PE1-vpn-instance-vpna-af-ipv4] quit
```

```
[PE4]ip vpn-instance vpna
[PE4-vpn-instance-vpna] ipv4-family
[PE4-vpn-instance-vpna-af-ipv4] route-distinguisher 100:40
[PE4-vpn-instance-vpna-af-ipv4] vpn-target 100:1020 both
[PE4-vpn-instance-vpna-af-ipv4] quit
```

# Create Loopback1, associate it with the VPN instance, and configure an IP address for the interface.

```
[PE1]interface LoopBack 1
[PE1-LoopBack1]ip binding vpn-instance vpna
Info: All IPv4 and IPv6 related configurations on this interface are removed.
[PE1-LoopBack1]ip address 10.1.1.1 32
```

```
[PE4]interface LoopBack 1
[PE4-LoopBack1]ip binding vpn-instance vpna
Info: All IPv4 and IPv6 related configurations on this interface are removed.
[PE4-LoopBack1]ip address 10.1.4.4 32
```

Note that you need to associate the interface with the VPN instance before configuring an IP address for the interface.

# Configure the MP-BGP VPNv4 peer relationships through Loopback0 and use the Loopback0 address as the router ID.

```
[PE1]bgp 65100
[PE1-bgp] router-id 10.0.1.1
[PE1-bgp] peer 10.0.5.5 as-number 65100
[PE1-bgp] peer 10.0.5.5 connect-interface LoopBack0
[PE1-bgp] peer 10.0.6.6 as-number 65100
[PE1-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 10.0.5.5 enable
[PE1-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[PE2]bgp 65100
[PE2-bgp] router-id 10.0.2.2
[PE2-bgp] peer 10.0.5.5 as-number 65100
[PE2-bgp] peer 10.0.5.5 connect-interface LoopBack0
[PE2-bgp] peer 10.0.6.6 as-number 65100
[PE2-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 10.0.5.5 enable
[PE2-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[PE3-bgp] router-id 10.0.3.3
[PE3-bgp] peer 10.0.5.5 as-number 65100
[PE3-bgp] peer 10.0.5.5 connect-interface LoopBack0
[PE3-bgp] peer 10.0.6.6 as-number 65100
[PE3-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE3-bgp] ipv4-family vpnv4
[PE3-bgp-af-vpnv4] peer 10.0.5.5 enable
[PE3-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[PE4]bgp 65100
[PE4-bgp] router-id 10.0.4.4
[PE4-bgp] peer 10.0.5.5 as-number 65100
[PE4-bgp] peer 10.0.5.5 connect-interface LoopBack0
[PE4-bgp] peer 10.0.6.6 as-number 65100
[PE4-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE4-bgp] ipv4-family vpnv4
[PE4-bgp-af-vpnv4] peer 10.0.5.5 enable
[PE4-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[P1]bgp 65100
[P1-bgp] router-id 10.0.5.5
[P1-bgp] peer 10.0.1.1 as-number 65100
[P1-bgp] peer 10.0.1.1 connect-interface LoopBack0
[P1-bgp] peer 10.0.2.2 as-number 65100
[P1-bgp] peer 10.0.2.2 connect-interface LoopBack0
[P1-bgp] peer 10.0.3.3 as-number 65100
[P1-bgp] peer 10.0.3.3 connect-interface LoopBack0
[P1-bgp] peer 10.0.4.4 as-number 65100
[P1-bgp] peer 10.0.4.4 connect-interface LoopBack0
[P1-bgp] ipv4-family vpnv4
[P1-bgp-af-vpnv4] undo policy vpn-target
[P1-bgp-af-vpnv4] peer 10.0.1.1 enable
[P1-bgp-af-vpnv4] peer 10.0.1.1 reflect-client
[P1-bgp-af-vpnv4] peer 10.0.2.2 enable
[P1-bgp-af-vpnv4] peer 10.0.2.2 reflect-client
[P1-bgp-af-vpnv4] peer 10.0.3.3 enable
[P1-bgp-af-vpnv4] peer 10.0.3.3 reflect-client
[P1-bgp-af-vpnv4] peer 10.0.4.4 enable
[P1-bgp-af-vpnv4] peer 10.0.4.4 reflect-client
```

When configuring an RR, disable the RT check on VPNv4 routes.

```
[P2]bgp 65100
[P2-bgp] router-id 10.0.6.6
[P2-bgp] peer 10.0.1.1 as-number 65100
[P2-bgp] peer 10.0.1.1 connect-interface LoopBack0
[P2-bgp] peer 10.0.2.2 as-number 65100
[P2-bgp] peer 10.0.2.2 connect-interface LoopBack0
```

```
[P2-bgp] peer 10.0.3.3 as-number 65100
[P2-bgp] peer 10.0.3.3 connect-interface LoopBack0
[P2-bgp] peer 10.0.4.4 as-number 65100
[P2-bgp] peer 10.0.4.4 connect-interface LoopBack0
[P2-bgp] ipv4-family vpnv4
[P2-bgp-af-vpnv4] undo policy vpn-target
[P2-bgp-af-vpnv4] peer 10.0.1.1 enable
[P2-bgp-af-vpnv4] peer 10.0.1.1 reflect-client
[P2-bgp-af-vpnv4] peer 10.0.2.2 enable
[P2-bgp-af-vpnv4] peer 10.0.2.2 reflect-client
[P2-bgp-af-vpnv4] peer 10.0.3.3 enable
[P2-bgp-af-vpnv4] peer 10.0.3.3 reflect-client
[P2-bgp-af-vpnv4] peer 10.0.4.4 enable
[P2-bgp-af-vpnv4] peer 10.0.4.4 reflect-client
```

# Check the VPNv4 peer relationship status on P1 and P2.

```
[P1]display bgp vpnv4 all peer
BGP local router ID : 10.0.5.5
Local AS number : 65100
Total number of peers : 4                 Peers in established state : 4

 Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State          PrefRcv
 10.0.1.1        4       65100         6        6     0  00:02:07   Established            0
 10.0.2.2        4       65100         6        6     0  00:02:05   Established            0
 10.0.3.3        4       65100         6        6     0  00:02:06   Established            0
 10.0.4.4        4       65100         6        6     0  00:02:04   Established            0
```

```
[P2]display bgp vpnv4 all peer
BGP local router ID : 10.0.6.6
Local AS number : 65100
Total number of peers : 4                 Peers in established state : 4

 Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State          PrefRcv
 10.0.1.1        4       65100         6        6     0  00:02:07   Established            0
 10.0.2.2        4       65100         6        6     0  00:02:05   Established            0
 10.0.3.3        4       65100         6        6     0  00:02:06   Established            0
 10.0.4.4        4       65100         6        6     0  00:02:04   Established            0
```

# Import the direct route of Loopback1 to BGP so that PE2 and PE4 can both learn the route of Loopback1 from the peer PE.

```
[PE1] bgp 65100
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-vpna] import-route direct
[PE4] bgp 65100
[PE4-bgp] ipv4-family vpn-instance vpna
[PE4-bgp-vpna] import-route direct
```

# Check VPNv4 routes on PE1.

```
[PE1]display bgp vpnv4 all routing-table | include 10.1.4.4
```

```
BGP Local router ID is 10.0.1.1
Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
              h - history,  i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V - valid, I - invalid, N - not-found

Total number of routes from all PE: 4
Route Distinguisher: 100:10


     Network            NextHop          MED        LocPrf    PrefVal Path/Ogn

Route Distinguisher: 100:40

     Network            NextHop          MED        LocPrf    PrefVal Path/Ogn
*>i  10.1.4.4/32        10.0.4.4         0          100       0       ?

VPN-Instance vpna, Router ID 10.0.1.1:

Total Number of Routes: 4
     Network            NextHop          MED        LocPrf    PrefVal Path/Ogn
*>i  10.1.4.4/32        10.0.4.4         0          100       0       ?
```

PE1 has learned the VPNv4 route from PE4 through MP-BGP.

# Check the VPN instance routing tables on PE1 and PE4.

```
[PE1]display ip routing-table vpn-instance vpna
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
         Destinations : 4        Routes : 4

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      10.1.1.1/32   Direct  0    0           D   127.0.0.1       LoopBack1
      10.1.4.4/32   IBGP    255  0           RD  10.0.4.4        GigabitEthernet0/5/0
                    IBGP    255  0           RD  10.0.4.4        GigabitEthernet0/5/1
     127.0.0.0/8    Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

```
[PE4]display ip routing-table vpn-instance vpna
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
         Destinations : 4        Routes : 4

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      10.1.1.1/32   IBGP    255  0           RD  10.0.1.1        GigabitEthernet0/5/0
                    IBGP    255  0           RD  10.0.1.1        GigabitEthernet0/5/1
      10.1.4.4/32   Direct  0    0           D   127.0.0.1       LoopBack1
     127.0.0.0/8    Direct  0    0           D   127.0.0.1       InLoopBack0
```

```
255.255.255.255/32  Direct  0      0            D    127.0.0.1       InLoopBack0
```

The route advertised by the peer end has been learned.

Step 7 Configure a tunnel selection policy.

Configure a tunnel selection policy to preferentially select SR-MPLS TE tunnels and associate the tunnel policy with the specified VPN instance.

# Configure a tunnel selection policy.

```
[PE1] tunnel-policy p1
[PE1-tunnel-policy-p1] tunnel select-seq sr-te load-balance-number 1
[PE1-tunnel-policy-p1] quit
```

```
[PE4] tunnel-policy p1
[PE4-tunnel-policy-p1] tunnel select-seq sr-te load-balance-number 1
[PE4-tunnel-policy-p1] quit
```

# Apply the tunnel policy to the VPN instance.

```
[PE1] ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] tnl-policy p1
```

```
[PE4] ip vpn-instance vpna
[PE4-vpn-instance-vpna] ipv4-family
[PE4-vpn-instance-vpna-af-ipv4] tnl-policy p1
```

# Check the VPN instance routing tables on PE1 and PE4.

```
[PE1]display ip routing-table vpn-instance vpna
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
         Destinations : 4        Routes : 4

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        10.1.1.1/32  Direct  0    0           D   127.0.0.1       LoopBack1
        10.1.4.4/32  IBGP    255  0           RD  10.0.4.4        Tunnel10
      127.0.0.0/8    Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32   Direct  0    0           D   127.0.0.1       InLoopBack0
```

```
[PE4]display ip routing-table vpn-instance vpna
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
```

```
         Destinations:4        Routes :4

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      10.1.1.1/32       IBGP    255  0           RD  10.0.1.1        Tunnel10
      10.1.4.4/32       Direct  0    0           D   127.0.0.1       LoopBack1
     127.0.0.0/8        Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

The routes from PE1 and PE4 to the network segment of the remote CE have recursed to SR-MPLS TE tunnels.

# Verify the connectivity.

```
[PE1]ping -vpn-instance vpna -a 10.1.1.1 10.1.4.4
  PING 10.1.4.4: 56  data bytes, press CTRL_C to break
    Reply from 10.1.4.4: bytes=56 Sequence=1 ttl=251 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=2 ttl=251 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=3 ttl=251 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=4 ttl=251 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=5 ttl=251 time=1 ms

  --- 10.1.4.4 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
The communication is normal.
```

**----End**

## 8.2.3 Quiz

In an SR-MPLS TE scenario, how can we forcibly forward packets through a specific interface on a specific device?

# 8.3 L3VPNv4 over SR-MPLS Policy

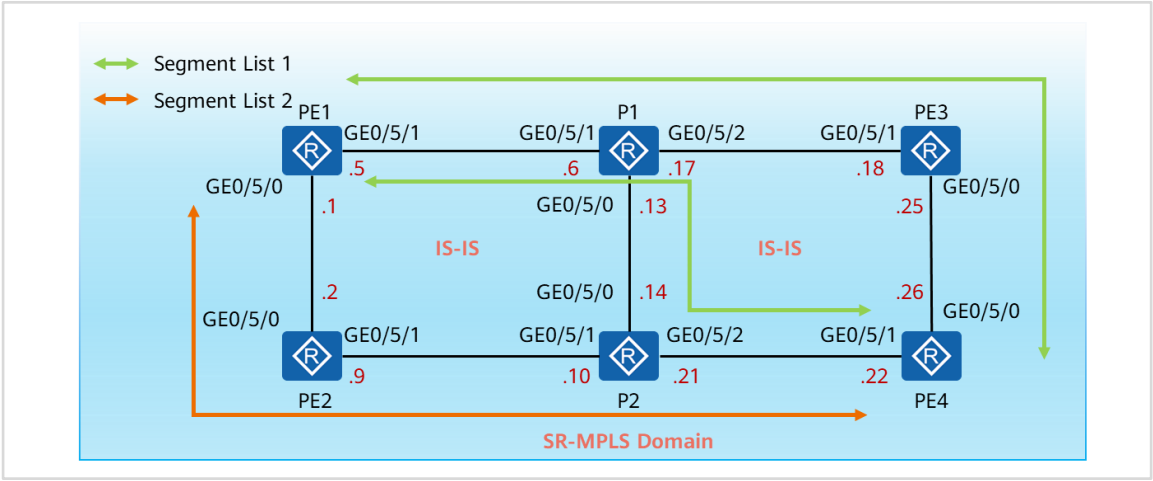
## 8.3.1 About This Lab

### 8.3.1.1 Objectives

- Configure IS-IS to ensure that PEs are routable to each other.
- Manually configure TE tunnels.
- Recurse L3VPN tunnels used for communication between CEs to specified static SR-MPLS Policies.
- Achieve high reliability of SR-MPLS Policies.

### 8.3.1.2 Networking Description

**Figure 8-3 L3VPNv4 over SR-MPLS Policy topology**




The figure shows the device connection and IP address planning. The interface interconnection addresses are in the format of 10.0.0.*Y*/30, and the values represented by Y are shown in the figure. Loopback0 is created on all devices, and loopback interface IP addresses are in the format of 10.0.*X*.*X*. The values indicated by *X* are listed in the table of the corresponding step.

Loopback1 is created on PE1 and PE4, with addresses being 10.1.1.1/32 and 10.1.4.4/32, respectively, to simulate CE user access.

L3VPNv4 traffic between PE1 and PE4 enters the WAN bearer network through SR-MPLS Policies. Multiple candidate paths are configured to ensure high service reliability. In addition, multiple segment lists are configured for candidate path 1 to ensure high path reliability.

## 8.3.2 Lab Task

### 8.3.2.1 Configuration Roadmap

1. Configure basic IP addresses for devices.
2. Configure IS-IS in the SR domain. Specifically, enable IS-IS on interconnection and Loopback0 interfaces for communication in the SR domain.

3. Configure MPLS. Specifically, enable MPLS and MPLS TE and set MPLS LSR IDs on devices.
4. Configure SR. Specifically, enable SR globally, enable IS-IS extensions for SR and the traffic engineering capability, and configure node SIDs.
5. Configure candidate paths on PE2 and PE4 and reference these candidate paths in SR-MPLS Policies.
6. Configure a VPN instance, add Loopback1 on PE1 and PE4 to the instance, and establish a VPNv4 peer relationship between PE1 and P1, between PE1 and P2, between PE4 and P1, and between PE4 and P2.
7. Configure a tunnel selection policy to recurse L3VPN traffic to specified SR-MPLS Policies.

## 8.3.2.2 Configuration Procedure

Step 1 Complete basic device configuration.

Configure the command validation mode as immediate validation and configure IP addresses for interconnection and Loopback0 interfaces. Loopback0 interface addresses need to be configured according to the following table.

**Table 8-4 Loopback0 IP addresses**

| Device ID | X value | Loopback0 IP Address |
|---|---|---|
| PE1 | 1 | 10.0.1.1 |
| PE2 | 2 | 10.0.2.2 |
| PE3 | 3 | 10.0.3.3 |
| PE4 | 4 | 10.0.4.4 |
| P1 | 5 | 10.0.5.5 |
| P2 | 6 | 10.0.6.6 |

# Name the devices.

N/A

# Disable unnecessary interfaces in this lab.

N/A

# Configure the command validation mode as immediate validation.

```
<PE1>system-view immediately
<PE2>system-view immediately
```

```
<PE3>system-view immediately
```

```
<PE4>system-view immediately
```

```
<P1>system-view immediately
```

```
<P2>system-view immediately
```

# Disable the DCN function globally on all devices.

```
[PE1] undo dcn
Warning: This operation will disable DCN function. Continue? [Y/N]:y
```

Here, PE1 is used as an example. Repeat the same operations for other devices.

By default, the DCN function is enabled on NE router interfaces. To facilitate this lab, disable the DCN function globally on all devices.

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE1.

```
[PE1]interface LoopBack0
[PE1-LoopBack0] ip address 10.0.1.1 32
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0]ip address 10.0.0.1 30
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
[PE1-GigabitEthernet0/5/1]ip address 10.0.0.5 30
[PE1-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE2.

```
[PE2]interface LoopBack0
[PE2-LoopBack0] ip address 10.0.2.2 32
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ip address 10.0.0.2 30
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ip address 10.0.0.9 30
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE3.

```
[PE3]interface LoopBack0
[PE3-LoopBack0] ip address 10.0.3.3 32
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] ip address 10.0.0.25 30
[PE3-GigabitEthernet0/5/0] quit
```

```
[PE3]interface GigabitEthernet0/5/1
[PE3-GigabitEthernet0/5/1] ip address 10.0.0.18 30
[PE3-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on PE4.

```
[PE4]interface LoopBack0
[PE4-LoopBack0] ip address 10.0.4.4 32
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] ip address 10.0.0.26 30
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] undo shutdown
[PE4-GigabitEthernet0/5/1] ip address 10.0.0.22 255.255.255.252
[PE4-GigabitEthernet0/5/1] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on P1.

```
[P1]interface LoopBack0
[P1-LoopBack0] ip address 10.0.5.5 32
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] ip address 10.0.0.13 30
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] ip address 10.0.0.6 30
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] ip address 10.0.0.17 30
[P1-GigabitEthernet0/5/2] quit
```

# Configure IP addresses for the interconnection and Loopback0 interfaces on P2.

```
[P2]interface LoopBack0
[P2-LoopBack0] ip address 10.0.6.6 32
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ip address 10.0.0.14 30
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ip address 10.0.0.10 30
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] undo shutdown
[P2-GigabitEthernet0/5/2] ip address 10.0.0.21 255.255.255.252
```

Step 2 Configure IS-IS.

The IS-IS area ID is 49.0001, the IS-IS process ID is 1, all devices are Level-2 devices, and the NET is **49.0001.000*X*.000*X*.000*X*.00**. The values indicated by *X* are listed in the table of Step 1. Enable IS-IS on Loopback0 and interconnection interfaces.

Note that you need to set **cost-style** to **wide** to support IS-IS extensions.

# Configure IS-IS on PE1.

```
[PE1]isis 1
[PE1-isis-1] is-level level-2
[PE1-isis-1] cost-style wide
[PE1-isis-1] network-entity 49.0001.0001.0001.0001.00
[PE1-isis-1] is-name PE1
[PE1]interface LoopBack0
[PE1-LoopBack0] isis enable 1
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0] isis enable 1
[PE1-GigabitEthernet0/5/0] isis circuit-type p2p
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
[PE1-GigabitEthernet0/5/1] isis enable 1
[PE1-GigabitEthernet0/5/1] isis circuit-type p2p
[PE1-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on PE2.

```
[PE2]isis 1
[PE2-isis-1] is-level level-2
[PE2-isis-1] cost-style wide
[PE2-isis-1] network-entity 49.0001.0002.0002.0002.00
[PE2-isis-1] is-name PE2
[PE2-isis-1]interface LoopBack0
[PE2-LoopBack0] isis enable 1
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] isis enable 1
[PE2-GigabitEthernet0/5/0] isis circuit-type p2p
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] isis enable 1
[PE2-GigabitEthernet0/5/1] isis circuit-type p2p
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on PE3.

```
[PE3]isis 1
[PE3-isis-1] is-level level-2
[PE3-isis-1] cost-style wide
[PE3-isis-1] network-entity 49.0001.0003.0003.0003.00
[PE3-isis-1] is-name PE3
[PE3-isis-1] quit
[PE3]interface LoopBack0
[PE3-LoopBack0] isis enable 1
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] isis enable 1
[PE3-GigabitEthernet0/5/0] isis circuit-type p2p
[PE3-GigabitEthernet0/5/0] quit
[PE3]interface GigabitEthernet0/5/1
```

```
[PE3-GigabitEthernet0/5/1] isis enable 1
[PE3-GigabitEthernet0/5/1] isis circuit-type p2p
[PE3-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on PE4.

```
[PE4]isis 1
[PE4-isis-1] is-level level-2
[PE4-isis-1] cost-style wide
[PE4-isis-1] network-entity 49.0001.0004.0004.0004.00
[PE4-isis-1] is-name PE4
[PE4-isis-1] quit
[PE4]interface LoopBack0
[PE4-LoopBack0] isis enable 1
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] isis enable 1
[PE4-GigabitEthernet0/5/0] isis circuit-type p2p
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] isis enable 1
[PE4-GigabitEthernet0/5/1] isis circuit-type p2p
[PE4-GigabitEthernet0/5/1] quit
```

# Configure IS-IS on P1.

```
[P1]isis 1
[P1-isis-1] is-level level-2
[P1-isis-1] cost-style wide
[P1-isis-1] network-entity 49.0001.0005.0005.0005.00
[P1-isis-1] is-name P1
[P1-isis-1] quit
[P1]interface LoopBack0
[P1-LoopBack0] isis enable 1
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] isis enable 1
[P1-GigabitEthernet0/5/0] isis circuit-type p2p
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] isis enable 1
[P1-GigabitEthernet0/5/1] isis circuit-type p2p
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] isis enable 1
[P1-GigabitEthernet0/5/2] isis circuit-type p2p
[P1-GigabitEthernet0/5/2] quit
```

# Configure IS-IS on P2.

```
[P2]isis 1
[P2-isis-1] is-level level-2
[P2-isis-1] cost-style wide
[P2-isis-1] network-entity 49.0001.0006.0006.0006.00
```

```
[P2-isis-1] is-name P2
[P2-isis-1] quit
[P2]interface LoopBack0
[P2-LoopBack0] isis enable 1
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] isis enable 1
[P2-GigabitEthernet0/5/0] isis circuit-type p2p
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] isis enable 1
[P2-GigabitEthernet0/5/1] isis circuit-type p2p
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] isis enable 1
[P2-GigabitEthernet0/5/2] isis circuit-type p2p
[P2-GigabitEthernet0/5/2] quit
```

# Check the IS-IS neighbor relationship on P1, PE2, and PE4.

```
[P1]display isis peer

                    Peer information for ISIS(1)

  System Id     Interface          Circuit Id          State HoldTime  Type     PRI
------------------------------------------------------------------------------
P2              GE0/5/0            0000000006          Up    26s       L2       --
PE1             GE0/5/1            0000000007          Up    25s       L2       --
PE3             GE0/5/2            0000000007          Up    27s       L2       --

Total Peer(s): 3
```

```
[PE2]display isis peer

                    Peer information for ISIS(1)

  System Id     Interface          Circuit Id          State HoldTime  Type     PRI
------------------------------------------------------------------------------
PE1             GE0/5/0            0000000006          Up    26s       L2       --
P2              GE0/5/1            0000000007          Up    25s       L2       --

Total Peer(s): 2
```

```
[PE4]display isis peer

                    Peer information for ISIS(1)

  System Id     Interface          Circuit Id          State HoldTime  Type     PRI
------------------------------------------------------------------------------
PE3             GE0/5/0            0000000006          Up    23s       L2       --
```

```
P2              GE0/5/1             0000000007          Up    23s         L2          --

Total Peer(s): 2
```

# Check IS-IS routes on P1.

```
[P1]display isis route

                  Routeinformation forISIS(1)
                  ---------------------------

                 ISIS(1) Level-2Forwarding Table
                 -------------------------------

IPV4Destination  IntCost   ExtCost ExitInterface     NextHop         Flags
--------------------------------------------------------------------------------
10.0.0.0/30      20        NULL    GE0/5/1          10.0.0.5        A/-/-/-
10.0.0.4/30      10        NULL    GE0/5/1          Direct          D/-/L/-
10.0.0.8/30      20        NULL    GE0/5/0          10.0.0.14       A/-/-/-
10.0.0.12/30     10        NULL    GE0/5/0          Direct          D/-/L/-
10.0.0.16/30     10        NULL    GE0/5/2          Direct          D/-/L/-
10.0.0.20/30     20        NULL    GE0/5/0          10.0.0.14       A/-/-/-
10.0.0.24/30     20        NULL    GE0/5/2          10.0.0.18       A/-/-/-
10.0.1.1/32      10        NULL    GE0/5/1          10.0.0.5        A/-/-/-
10.0.2.2/32      20        NULL    GE0/5/1          10.0.0.5        A/-/-/-
                                   GE0/5/0          10.0.0.14
10.0.3.3/32      10        NULL    GE0/5/2          10.0.0.18       A/-/-/-
10.0.4.4/32      20        NULL    GE0/5/2          10.0.0.18       A/-/-/-
                                   GE0/5/0          10.0.0.14
10.0.5.5/32      0         NULL    Loop0            Direct          D/-/L/-
10.0.6.6/32      10        NULL    GE0/5/0          10.0.0.14       A/-/-/-
    Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
           U-Up/Down Bit Set, LP-Local Prefix-Sid
    Protect Type: L-Link Protect, N-Node Protect
```

Route learning has completed.

## Step 3 Configure MPLS.

Enable MPLS on all devices and configure MPLS LSR IDs (by using the IP addresses of Loopback0). MPLS does not need to be enabled on interfaces.

```
[PE1]mpls lsr-id 10.0.1.1
[PE1]mpls
```

```
[PE2]mpls lsr-id 10.0.2.2
[PE2]mpls
[PE3]mpls lsr-id 10.0.3.3
[PE3]mpls
```

```
[PE4]mpls lsr-id 10.0.4.4
[PE4]mpls
```

```
[P1]mpls lsr-id 10.0.5.5
[P1]mpls
```

```
[P2]mpls lsr-id 10.0.6.6
[P2]mpls
```

Step 4 Configure SR capabilities on devices.

Enable SR-MPLS globally, enable IS-IS extensions for SR, configure an SRGB for IS-IS, and set the SRGB range to 16000 to 17000 on all devices.

Configure a SID for Loopback0 and specify the relative label value *X* using the **index** parameter. The values indicated by *X* are listed in the table of Step 1.

# Configure PE1.

```
[PE1] segment-routing
[PE1-segment-routing] quit
[PE1] isis 1
[PE1-isis-1] segment-routing mpls
[PE1-isis-1] segment-routing global-block 16000 17000
[PE1-isis-1] quit
[PE1] interface LoopBack 0
[PE1-LoopBack0] isis prefix-sid index 1
[PE1-LoopBack0] quit
```

# Configure PE2.

```
[PE2] segment-routing
[PE2-segment-routing] quit
[PE2] isis 1
[PE2-isis-1] segment-routing mpls
[PE2-isis-1] segment-routing global-block 16000 17000
[PE2-isis-1] quit
[PE2] interface LoopBack 0
[PE2-LoopBack0] isis prefix-sid index 2
[PE2-LoopBack0] quit
```

# Configure PE3.

```
[PE3] segment-routing
[PE3-segment-routing] quit
[PE3] isis 1
[PE3-isis-1] segment-routing mpls
[PE3-isis-1] segment-routing global-block 16000 17000
[PE3-isis-1] quit
[PE3] interface LoopBack 0
```

```
[PE3-LoopBack0] isis prefix-sid index 3
[PE3-LoopBack0] quit
```

# Configure PE4.

```
[PE4] segment-routing
[PE4-segment-routing] quit
[PE4] isis 1
[PE4-isis-1] segment-routing mpls
[PE4-isis-1] segment-routing global-block 16000 17000
[PE4-isis-1] quit
[PE4] interface LoopBack 0
[PE4-LoopBack0] isis prefix-sid index 4
[PE4-LoopBack0] quit
```

# Configure P1.

```
[P1] segment-routing
[P1-segment-routing] quit
[P1] isis 1
[P1-isis-1] segment-routing mpls
[P1-isis-1] segment-routing global-block 16000 17000
[P1-isis-1] quit
[P1] interface LoopBack 0
[P1-LoopBack0] isis prefix-sid index 5
[P1-LoopBack0] quit
```

# Configure P2.

```
[P2] segment-routing
[P2-segment-routing] quit
[P2] isis 1
[P2-isis-1] segment-routing mpls
[P2-isis-1] segment-routing global-block 16000 17000
[P2-isis-1] quit
[P2] interface LoopBack 0
[P2-LoopBack0] isis prefix-sid index 6
[P2-LoopBack0] quit
```

# Manually configure adjacency SIDs on P1 and P2.

To ensure that the adjacency SIDs specified during explicit path configuration remain unchanged, you are advised to configure static adjacency SIDs. In this way, the SIDs remain unchanged after the device restarts.

The following table lists the static adjacency SIDs to be configured on P1 and P2.

**Table 8-5 Adjacency SIDs**

| Device ID | Local IP Address | Peer IP Address | SID |
|---|---|---|---|
| P1 | 10.0.0.6 | 10.0.0.5 | 142336 |
| P1 | 10.0.0.13 | 10.0.0.14 | 142337 |

| | | | |
|---|---|---|---|
| P1 | 10.0.0.17 | 10.0.0.18 | 142338 |
| P2 | 10.0.0.10 | 10.0.0.9 | 142336 |
| P2 | 10.0.0.14 | 10.0.0.13 | 142337 |
| P2 | 10.0.0.21 | 10.0.0.22 | 142338 |

```
[P1] segment-routing
[P1-segment-routing] ipv4 adjacency local-ip-addr 10.0.0.6 remote-ip-addr 10.0.0.5 sid 142336
[P1-segment-routing] ipv4 adjacency local-ip-addr 10.0.0.13 remote-ip-addr 10.0.0.14 sid 142337
[P1-segment-routing] ipv4 adjacency local-ip-addr 10.0.0.17 remote-ip-addr 10.0.0.18 sid 142338
```

```
[P2] segment-routing
[P2-segment-routing] ipv4 adjacency local-ip-addr 10.0.0.10 remote-ip-addr 10.0.0.9 sid 142336
[P2-segment-routing] ipv4 adjacency local-ip-addr 10.0.0.14 remote-ip-addr 10.0.0.13 sid 142337
[P2-segment-routing] ipv4 adjacency local-ip-addr 10.0.0.21 remote-ip-addr 10.0.0.22 sid 142338
```

Step 5 Configure SR-MPLS Policies.

On both PE1 and PE4, configure two candidate paths and three segment lists as planned, and reference the candidate paths in the SR-MPLS Policies.

# Configure segment lists and candidate paths on PE1.

```
[PE1-segment-routing] segment-list PE1_PE4_1
[PE1-segment-routing-segment-list-PE1_PE4_1] index 10 sid label 16005
[PE1-segment-routing-segment-list-PE1_PE4_1] index 20 sid label 142338
[PE1-segment-routing-segment-list-PE1_PE4_1] index 30 sid label 16004
[PE1-segment-routing-segment-list-PE1_PE4_1] quit
[PE1-segment-routing] segment-list PE1_PE4_2
[PE1-segment-routing-segment-list-PE1_PE4_2] index 10 sid label 16005
[PE1-segment-routing-segment-list-PE1_PE4_2] index 20 sid label 142337
[PE1-segment-routing-segment-list-PE1_PE4_2] index 30 sid label 142338
[PE1-segment-routing-segment-list-PE1_PE4_2] quit
[PE1-segment-routing] segment-list PE1_PE4_3
[PE1-segment-routing-segment-list-PE1_PE4_3] index 10 sid label 16002
[PE1-segment-routing-segment-list-PE1_PE4_3] index 20 sid label 16006
[PE1-segment-routing-segment-list-PE1_PE4_3] index 30 sid label 142338
[PE1-segment-routing-segment-list-PE1_PE4_3] quit
[PE1-segment-routing] sr-te policy p1 endpoint 10.0.4.4 color 100
[PE1-segment-routing-te-policy-p1] candidate-path preference 100
[PE1-segment-routing-te-policy-p1-path] segment-list PE1_PE4_1
[PE1-segment-routing-te-policy-p1-path] segment-list PE1_PE4_2
[PE1-segment-routing-te-policy-p1-path] quit
[PE1-segment-routing-te-policy-p1] candidate-path preference 50
[PE1-segment-routing-te-policy-p1-path] segment-list PE1_PE4_3
```

To reach PE4, segment lists 1, 2, and 3 need to pass through P1 and PE3, P1 and P2, and PE2 and P2, respectively. Segment lists 1 and 2 form candidate path 1, and segment list 3 forms candidate path 2.

# Configure segment lists and candidate paths on PE4.

```
[PE4]segment-routing
[PE4-segment-routing] segment-list PE4_PE1_1
[PE4-segment-routing-segment-list-PE4_PE1_1] index 10 sid label 16003
[PE4-segment-routing-segment-list-PE4_PE1_1] index 20 sid label 16005
[PE4-segment-routing-segment-list-PE4_PE1_1] index 30 sid label 142336
[PE4-segment-routing-segment-list-PE4_PE1_1] quit
[PE4-segment-routing] segment-list PE4_PE1_2
[PE4-segment-routing-segment-list-PE4_PE1_2] index 10 sid label 16006
[PE4-segment-routing-segment-list-PE4_PE1_2] index 20 sid label 142337
[PE4-segment-routing-segment-list-PE4_PE1_2] index 30 sid label 142336
[PE4-segment-routing-segment-list-PE4_PE1_2] quit
[PE4-segment-routing] segment-list PE4_PE1_3
[PE4-segment-routing-segment-list-PE4_PE1_3] index 10 sid label 16006
[PE4-segment-routing-segment-list-PE4_PE1_3] index 20 sid label 142336
[PE4-segment-routing-segment-list-PE4_PE1_3] index 30 sid label 16001
[PE4-segment-routing-segment-list-PE4_PE1_3] quit
[PE4-segment-routing] sr-te policy p1 endpoint 10.0.1.1 color 100
[PE4-segment-routing-te-policy-p1] candidate-path preference 100
[PE4-segment-routing-te-policy-p1-path] segment-list PE4_PE1_1
[PE4-segment-routing-te-policy-p1-path] segment-list PE4_PE1_2
[PE4-segment-routing-te-policy-p1-path] quit
[PE4-segment-routing-te-policy-p1] candidate-path preference 50
[PE4-segment-routing-te-policy-p1-path] segment-list PE4_PE1_3
```

To reach PE1, segment lists 1, 2, and 3 need to pass through PE3 and P1, P2 and P1, and P2 and PE2, respectively. Segment lists 1 and 2 form candidate path 1, and segment list 3 forms candidate path 2.

Step 6 Configure L3VPN.

On both PE1 and PE4, create a VPN instance named **vpna**, create Loopback1, add the interface to the VPN instance, and establish an MP-BGP EVPN peer relationship between PE1 and P1, between PE1 and P2, between PE4 and P1, and between PE4 and P2 (the AS number is 65100). P1 and P2 function as RRs. PE1 and PE4 function as RR clients, which advertise VPNv4 routes through P1 and P2.

# Create a VPN instance named **vpna**.

```
[PE1]ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:10
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 100:1020 both
[PE1-vpn-instance-vpna-af-ipv4] quit
```

```
[PE4]ip vpn-instance vpna
[PE4-vpn-instance-vpna] ipv4-family
```

```
[PE4-vpn-instance-vpna-af-ipv4] route-distinguisher 100:40
[PE4-vpn-instance-vpna-af-ipv4] vpn-target 100:1020 both
[PE4-vpn-instance-vpna-af-ipv4] quit
```

# Create Loopback1, associate it with the VPN instance, and configure an IP address for the interface.

```
[PE1]interface LoopBack 1
[PE1-LoopBack1]ip binding vpn-instance vpna
Info: All IPv4 and IPv6 related configurations on this interface are removed.
[PE1-LoopBack1]ip address 10.1.1.1 32
```

```
[PE4]interface LoopBack 1
[PE4-LoopBack1]ip binding vpn-instance vpna
Info: All IPv4 and IPv6 related configurations on this interface are removed.
[PE4-LoopBack1]ip address 10.1.4.4 32
```

Note that you need to associate the interface with the VPN instance before configuring an IP address for the interface.

# Configure the MP-BGP VPNv4 peer relationships through Loopback0 and use the Loopback0 address as the router ID.

```
[PE1]bgp 65100
[PE1-bgp] router-id 10.0.1.1
[PE1-bgp] peer 10.0.5.5 as-number 65100
[PE1-bgp] peer 10.0.5.5 connect-interface LoopBack0
[PE1-bgp] peer 10.0.6.6 as-number 65100
[PE1-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE1-bgp] #
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpnv4] peer 10.0.5.5 enable
[PE1-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[PE2]bgp 65100
[PE2-bgp] router-id 10.0.2.2
[PE2-bgp] peer 10.0.5.5 as-number 65100
[PE2-bgp] peer 10.0.5.5 connect-interface LoopBack0
[PE2-bgp] peer 10.0.6.6 as-number 65100
[PE2-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE2-bgp] #
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 10.0.5.5 enable
[PE2-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[PE3-bgp] router-id 10.0.3.3
[PE3-bgp] peer 10.0.5.5 as-number 65100
```

```
[PE3-bgp] peer 10.0.5.5 connect-interface LoopBack0
[PE3-bgp] peer 10.0.6.6 as-number 65100
[PE3-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE3-bgp] #
[PE3-bgp] ipv4-family vpnv4
[PE3-bgp-af-vpnv4] peer 10.0.5.5 enable
[PE3-bgp-af-vpnv4] peer 10.0.6.6 enable
[PE4]bgp 65100
[PE4-bgp] router-id 10.0.4.4
[PE4-bgp] peer 10.0.5.5 as-number 65100
[PE4-bgp] peer 10.0.5.5 connect-interface LoopBack0
[PE4-bgp] peer 10.0.6.6 as-number 65100
[PE4-bgp] peer 10.0.6.6 connect-interface LoopBack0
[PE4-bgp] #
[PE4-bgp] ipv4-family vpnv4
[PE4-bgp-af-vpnv4] peer 10.0.5.5 enable
[PE4-bgp-af-vpnv4] peer 10.0.6.6 enable
```

```
[P1]bgp 65100
[P1-bgp] router-id 10.0.5.5
[P1-bgp] peer 10.0.1.1 as-number 65100
[P1-bgp] peer 10.0.1.1 connect-interface LoopBack0
[P1-bgp] peer 10.0.2.2 as-number 65100
[P1-bgp] peer 10.0.2.2 connect-interface LoopBack0
[P1-bgp] peer 10.0.3.3 as-number 65100
[P1-bgp] peer 10.0.3.3 connect-interface LoopBack0
[P1-bgp] peer 10.0.4.4 as-number 65100
[P1-bgp] peer 10.0.4.4 connect-interface LoopBack0
[P1-bgp] #
[P1-bgp] ipv4-family vpnv4
[P1-bgp-af-vpnv4] undo policy vpn-target
[P1-bgp-af-vpnv4] peer 10.0.1.1 enable
[P1-bgp-af-vpnv4] peer 10.0.1.1 reflect-client
[P1-bgp-af-vpnv4] peer 10.0.2.2 enable
[P1-bgp-af-vpnv4] peer 10.0.2.2 reflect-client
[P1-bgp-af-vpnv4] peer 10.0.3.3 enable
[P1-bgp-af-vpnv4] peer 10.0.3.3 reflect-client
[P1-bgp-af-vpnv4] peer 10.0.4.4 enable
[P1-bgp-af-vpnv4] peer 10.0.4.4 reflect-client
```

When configuring an RR, disable the RT check on VPNv4 routes.

```
[P2]bgp 65100
[P2-bgp] router-id 10.0.6.6
[P2-bgp] peer 10.0.1.1 as-number 65100
[P2-bgp] peer 10.0.1.1 connect-interface LoopBack0
[P2-bgp] peer 10.0.2.2 as-number 65100
[P2-bgp] peer 10.0.2.2 connect-interface LoopBack0
[P2-bgp] peer 10.0.3.3 as-number 65100
[P2-bgp] peer 10.0.3.3 connect-interface LoopBack0
[P2-bgp] peer 10.0.4.4 as-number 65100
[P2-bgp] peer 10.0.4.4 connect-interface LoopBack0
[P2-bgp] #
```

```
[P2-bgp] ipv4-family vpnv4
[P2-bgp-af-vpnv4] undo policy vpn-target
[P2-bgp-af-vpnv4] peer 10.0.1.1 enable
[P2-bgp-af-vpnv4] peer 10.0.1.1 reflect-client
[P2-bgp-af-vpnv4] peer 10.0.2.2 enable
[P2-bgp-af-vpnv4] peer 10.0.2.2 reflect-client
[P2-bgp-af-vpnv4] peer 10.0.3.3 enable
[P2-bgp-af-vpnv4] peer 10.0.3.3 reflect-client
[P2-bgp-af-vpnv4] peer 10.0.4.4 enable
[P2-bgp-af-vpnv4] peer 10.0.4.4 reflect-client
```

# Check the VPNv4 peer relationship status on P1 and P2.

```
[P1]display bgp vpnv4 all peer
BGP local router ID : 10.0.5.5
Local AS number : 65100
Total number of peers : 4                 Peers in established state : 4

 Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State          PrefRcv
 10.0.1.1        4       65100         6        6     0  00:02:07   Established            0
 10.0.2.2        4       65100         6        6     0  00:02:05   Established            0
 10.0.3.3        4       65100         6        6     0  00:02:06   Established            0
 10.0.4.4        4       65100         6        6     0  00:02:04   Established            0
```

```
[P2]display bgp vpnv4 all peer
BGP local router ID : 10.0.6.6
Local AS number : 65100
Total number of peers : 4                 Peers in established state : 4

 Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State          PrefRcv
 10.0.1.1        4       65100         6        6     0  00:02:07   Established            0
 10.0.2.2        4       65100         6        6     0  00:02:05   Established            0
 10.0.3.3        4       65100         6        6     0  00:02:06   Established            0
 10.0.4.4        4       65100         6        6     0  00:02:04   Established            0
```

# Import the direct route of Loopback1 to BGP so that PE2 and PE4 can both learn the route of Loopback1 from the peer PE.

```
[PE1] bgp 65100
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-vpna] import-route direct
```

```
[PE4] bgp 65100
[PE4-bgp] ipv4-family vpn-instance vpna
[PE4-bgp-vpna] import-route direct
```

# Check VPNv4 routes on PE1.

```
[PE1]display bgp vpnv4 all routing-table | include 10.1.4.4
```

```
BGP Local router ID is 10.0.1.1
Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
              h - history,  i - internal, s - suppressed, S - Stale
              Origin : i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V - valid, I - invalid, N - not-found

Total number of routes from all PE: 4
Route Distinguisher: 100:10


         Network            NextHop          MED        LocPrf    PrefVal Path/Ogn

Route Distinguisher: 100:40

         Network            NextHop          MED        LocPrf    PrefVal Path/Ogn
*>i      10.1.4.4/32        10.0.4.4           0          100          0       ?

VPN-Instance vpna, Router ID 10.0.1.1:

Total Number of Routes: 4
         Network            NextHop          MED        LocPrf    PrefVal Path/Ogn
*>i      10.1.4.4/32        10.0.4.4           0          100          0       ?
```

PE1 has learned the VPNv4 route from PE4 through MP-BGP.

# Check the VPN instance routing tables on PE1 and PE4.

```
[PE1]display ip routing-table vpn-instance vpna
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
         Destinations : 4        Routes : 4

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      10.1.1.1/32   Direct  0    0           D   127.0.0.1       LoopBack1
      10.1.4.4/32   IBGP    255  0           RD  10.0.4.4        GigabitEthernet0/5/0
                    IBGP    255  0           RD  10.0.4.4        GigabitEthernet0/5/1
     127.0.0.0/8    Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

```
[PE4]display ip routing-table vpn-instance vpna
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
         Destinations : 4        Routes : 4

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      10.1.1.1/32   IBGP    255  0           RD  10.0.1.1        GigabitEthernet0/5/0
                    IBGP    255  0           RD  10.0.1.1        GigabitEthernet0/5/1
      10.1.4.4/32   Direct  0    0           D   127.0.0.1       LoopBack1
     127.0.0.0/8    Direct  0    0           D   127.0.0.1       InLoopBack0
```

```
255.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

The route advertised by the peer end has been learned.

Step 7 Configure a tunnel selection policy.

# Configure a tunnel selection policy.

```
[PE1]tunnel-policy p1
[PE1-tunnel-policy-p1] tunnel select-seq sr-te-policy load-balance-number 1 unmix
```

```
[PE4] tunnel-policy p1
[PE4-tunnel-policy-p1] tunnel select-seq sr-te-policy load-balance-number 1 unmix
```

# Apply the tunnel selection policy to the VPN instance.

```
[PE1]ip vpn-instance vpna
[PE1-vpn-instance-vpna]tnl-policy p1
```

```
[PE4]ip vpn-instance vpna
[PE4-vpn-instance-vpna]tnl-policy p1
```

# Set the default color for the route in the VPN instance.

```
[PE1]ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] default-color 100
```

```
[PE4]ip vpn-instance vpna
[PE4-vpn-instance-vpna] ipv4-family
[PE4-vpn-instance-vpna-af-ipv4] default-color 100
```

To enable the route in the VPN instance to recurse to the corresponding SR-MPLS Policy, you can use a route-policy to set a color value for the route or set the default color value for the route in the local VPN instance. The second method is used in this example.

# Check the VPN instance IP routing tables on PE1 and PE4.

```
[PE1]display ip routing-table vpn-instance vpna
RouteFlags:R- relay, D - download tofib, T- tovpn-instance, B- black holeroute
------------------------------------------------------------------------------
Routing Table :vpna
         Destinations:4        Routes :4

Destination/Mask    Proto   Pre  Cost       Flags NextHop         Interface
```

```
     10.1.1.1/32        Direct  0    0           D   127.0.0.1       LoopBack1
     10.1.4.4/32        IBGP    255  0           RD  10.0.4.4        p1
    127.0.0.0/8         Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32      Direct  0    0           D   127.0.0.1       InLoopBack0
```

```
[PE4]display ip routing-table vpn-instance vpna
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
         Destinations : 4        Routes : 4

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      10.1.1.1/32   IBGP    255  0           RD  10.0.1.1        p1
      10.1.4.4/32   Direct  0    0           D   127.0.0.1       LoopBack1
     127.0.0.0/8    Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

The routes have recursed to the corresponding SR-MPLS Policy.

# Check the tunnel ID of the route on PE1.

```
[PE1]display ip routing-table vpn-instance vpna 10.1.4.4 verbose | include TunnelID
Info: It will take a long time if the content you search is too much or the string you input is too long,
you can press CTRL_C to break.
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
Summary Count : 1
        TunnelID: 0x000000003200000001      Flags: RD
```

# Check the tunnel type.

```
[PE1]display tunnel-info all | include 0x000000003200000001
Info: It will take a long time if the content you search is too much or the string you input is too long,
you can press CTRL_C to break.
Tunnel ID                Type                 Destination      Status
------------------------------------------------------------------------------
0x000000003200000001 srtepolicy             10.0.4.4         UP
```

The tunnel type is SR-MPLS Policy.

## Step 8 Perform a path switchover test.

Configure SBFD and test whether service traffic can be switched to different segment lists.

# Test the service connectivity on PE1.

```
[PE1]ping -vpn-instance vpna -a 10.1.1.1 10.1.4.4
  PING 10.1.4.4: 56  data bytes, press CTRL_C to break
    Reply from 10.1.4.4: bytes=56 Sequence=1 ttl=253 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=2 ttl=253 time=1 ms
```

```
    Reply from 10.1.4.4: bytes=56 Sequence=3 ttl=253 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=4 ttl=253 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=5 ttl=253 time=1 ms

  --- 10.1.4.4 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

The communication is normal.

# Configure SBFD.

```
[PE1]bfd
[PE1-bfd]quit
[PE1]sbfd
[PE1-sbfd]reflector discriminator 10.0.1.1
[PE1-sbfd]quit
[PE1]segment-routing
[PE1-segment-routing]sr-te-policy seamless-bfd enable
[PE1-segment-routing]quit
```

```
[PE4]bfd
[PE4-bfd]quit
[PE4]sbfd
[PE4-sbfd]reflector discriminator 10.0.4.4
[PE4-sbfd]quit
[PE4]segment-routing
[PE4-segment-routing]sr-te-policy seamless-bfd enable
[PE4-segment-routing]quit
```

Set the reflection address of SBFD to the local MPLS LSR ID.

# Configure hot standby.

```
[PE1]segment-routing
[PE1-segment-routing]sr-te-policy backup hot-standby enable
```

```
[PE4]segment-routing
[PE4-segment-routing]sr-te-policy backup hot-standby enable
```

# Check the SBFD status. PE1 is used as an example.

```
[PE1]display sr-te policy policy-name p1
PolicyName : p1
Endpoint                : 10.0.4.4                      Color                 : 100
TunnelId                : 1                              TunnelType            : SR-TE Policy
Binding SID             : -                              MTU                   : -
```

```
Policy State              : Up                          State Change Time      : 2021-08-26
03:35:22
Admin State               : UP                          Traffic Statistics     : Disable
BFD                       :   SBFD Enable                 Backup Hot-Standby     : Enable
DiffServ-Mode             : -
Candidate-path Count : 2

Candidate-path Preference: 100
Path State              : Active                  Path Type              : Primary
Protocol-Origin         : Configuration(30)       Originator             : 0, 0.0.0.0
Discriminator           : 100                      Binding SID            : -
GroupId                 : 8193                      Policy Name            : p1
Template ID             : -
Segment-List Count      : 2
 Segment-List           : PE1_PE4_1
  Segment-List ID       : 16385                    XcIndex                : 2016385
  List State            : Up                      BFD State              : UP
  EXP                   : -                        TTL                    : -
  DeleteTimerRemain     : -
  Label : 16005, 142338, 16004
 Segment-List           : PE1_PE4_2
  Segment-List ID       : 16386                    XcIndex                : 2016386
  List State            : Up                      BFD State              : UP
  EXP                   : -                        TTL                    : -
  DeleteTimerRemain     : -
  Label : 16005, 142337, 142338

Candidate-path Preference: 50
Path State              : Active                  Path Type              : Backup
Protocol-Origin         : Configuration(30)       Originator             : 0, 0.0.0.0
Discriminator           : 50                       Binding SID            : -
GroupId                 : 8194                      Policy Name            : p1
Template ID             : -
Segment-List Count      : 1
 Segment-List           : PE1_PE4_3
  Segment-List ID       : 16387                    XcIndex                : 2016387
  List State            : Up                      BFD State              : UP
  EXP                   : -                        TTL                    : -
  DeleteTimerRemain     : -
  Label : 16002, 16006, 142338
```

In this case, all candidate paths and segment lists are up, and candidate path 1 is the primary one.

# Shut down GE0/5/2 on P1.

```
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2]shutdown
[P1-GigabitEthernet0/5/2]quit
```

# Check the SBFD status on PE1 again.

```
[PE1]display sr-te policy policy-name p1
PolicyName : p1
Endpoint                : 10.0.4.4                Color                  : 100
```

```
TunnelId                : 1                          TunnelType              : SR-TE Policy
Binding SID             : -                          MTU                     : -
Policy State            : Up                         State Change Time       : 2021-08-26
03:35:22
Admin State             : UP                         Traffic Statistics      : Disable
BFD                     : SBFD Enable                Backup Hot-Standby      : Enable
DiffServ-Mode           : -
Candidate-path Count : 2

Candidate-path Preference: 100
Path State              : Active                     Path Type               : Primary
Protocol-Origin         : Configuration(30)          Originator              : 0, 0.0.0.0
Discriminator           : 100                        Binding SID             : -
GroupId                 : 8193                       Policy Name             : p1
Template ID             : -
Segment-List Count      : 2
 Segment-List           : PE1_PE4_1
  Segment-List ID       : 16385                      XcIndex                 : 2016385
  List State            : Down (BFD Down)            BFD State               : DOWN
  EXP                   : -                          TTL                     : -
  DeleteTimerRemain     : -
  Label : 16005, 142338, 16004
 Segment-List           : PE1_PE4_2
  Segment-List ID       : 16386                      XcIndex                 : 2016386
  List State            : Up                         BFD State               : UP
  EXP                   : -                          TTL                     : -
  DeleteTimerRemain     : -
  Label : 16005, 142337, 142338

Candidate-path Preference: 50
Path State              : Active                     Path Type               : Backup
Protocol-Origin         : Configuration(30)          Originator              : 0, 0.0.0.0
Discriminator           : 50                         Binding SID             : -
GroupId                 : 8194                       Policy Name             : p1
Template ID             : -
Segment-List Count      : 1
 Segment-List           : PE1_PE4_3
  Segment-List ID       : 16387                      XcIndex                 : 2016387
  List State            : Up                         BFD State               : UP
  EXP                   : -                          TTL                     : -
  DeleteTimerRemain     : -
  Label : 16002, 16006, 142338
```

Candidate path 1 is still the primary one, but the first segment list has gone down.

# Test the service connectivity.

```
[PE1]ping -vpn-instance vpna -a 10.1.1.1 10.1.4.4
  PING 10.1.4.4: 56  data bytes, press CTRL_C to break
    Reply from 10.1.4.4: bytes=56 Sequence=1 ttl=251 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=2 ttl=251 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=3 ttl=251 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=4 ttl=251 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=5 ttl=251 time=1 ms
```

```
  --- 10.1.4.4 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

The communication is normal.

# Shut down GE0/5/0 on P1.

```
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] shutdown
```

# Check the SBFD status on PE1 again.

```
[PE1]display sr-te policy policy-name p1
PolicyName : p1
Endpoint                 : 10.0.4.4                    Color                 : 100
TunnelId                 : 1                            TunnelType            : SR-TE Policy
Binding SID              : -                           MTU                   : -
Policy State             : Up                          State Change Time     : 2021-08-26
03:42:31
Admin State              : UP                           Traffic Statistics    : Disable
BFD                      : SBFD Enable                  Backup Hot-Standby    : Enable
DiffServ-Mode            : -
Candidate-path Count : 2

Candidate-path Preference: 100
Path State               : Inactive (BFD Down)         Path Type             : -
Protocol-Origin          : Configuration(30)           Originator            : 0, 0.0.0.0
Discriminator            : 100                          Binding SID           : -
GroupId                  : 8193                         Policy Name           : p1
Template ID              : -
Segment-List Count       : 2
 Segment-List            : PE1_PE4_1
  Segment-List ID        : 16385                        XcIndex               : 2016385
  List State             : Down (BFD Down)              BFD State             : DOWN
  EXP                    : -                           TTL                   : -
  DeleteTimerRemain      : -
  Label : 16005, 142338, 16004
 Segment-List            : PE1_PE4_2
  Segment-List ID        : 16386                        XcIndex               : 2016386
  List State             : Down (BFD Down)              BFD State             : DOWN
  EXP                    : -                           TTL                   : -
  DeleteTimerRemain      : -
  Label : 16005, 142337, 142338

Candidate-path Preference: 50
Path State               : Active                      Path Type             : Primary
Protocol-Origin          : Configuration(30)           Originator            : 0, 0.0.0.0
Discriminator            : 50                           Binding SID           : -
GroupId                  : 8194                         Policy Name           : p1
Template ID              : -
Segment-List Count       : 1
 Segment-List            : PE1_PE4_3
```

```
  Segment-List ID      : 16387                    XcIndex              : 2016387
  List State           : Up                       BFD State            : UP
  EXP                  : -                        TTL                  : -
  DeleteTimerRemain    : -
  Label : 16002, 16006, 142338
```

All the segment lists of candidate path 1 are faulty, and candidate path 2 becomes the primary one.

# Test the service connectivity.

```
[PE1]ping -vpn-instance vpna -a 10.1.1.1 10.1.4.4
  PING 10.1.4.4: 56  data bytes, press CTRL_C to break
    Reply from 10.1.4.4: bytes=56 Sequence=1 ttl=249 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=2 ttl=249 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=3 ttl=249 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=4 ttl=249 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=5 ttl=249 time=1 ms

  --- 10.1.4.4 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

The communication is still normal.

**----End**

## 8.3.3 Quiz

What is the 3-tuple used to uniquely identify an SR-MPLS Policy?

# 9 SRv6

## 9.1 L3VPNv4 over SRv6 BE

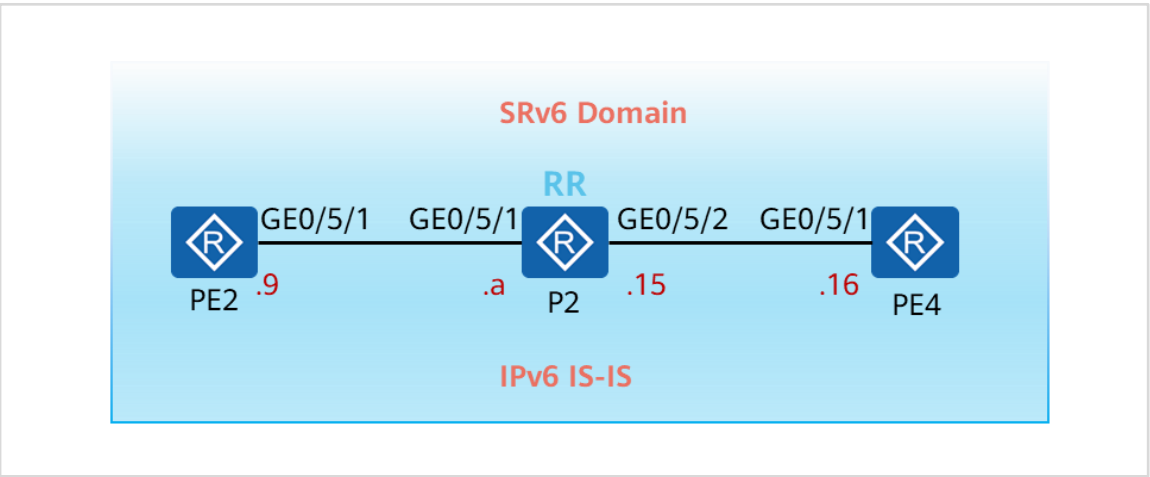
### 9.1.1 About This Lab

#### 9.1.1.1 Objectives

- Configure SRv6 locators for automatic allocation of SIDs to local VPN routes.
- Recurse L3VPN tunnels used for communication between CEs to SRv6 BE tunnels.
- Observe packet forwarding over an SRv6 BE tunnel.

#### 9.1.1.2 Networking Description

**Figure 9-1 L3VPNv4 over SRv6 BE topology**




The figure shows the device connection and IP address planning. The interface interconnection addresses are in the format of 2001::*Y*/126, and the values represented by Y are shown in the figure. Loopback0 is created on all devices, and loopback interface IP addresses are in the format of 2001::*X*:*X*/128. The values indicated by *X* are listed in the table of the corresponding step.

Loopback1 is created on PE2 and PE4, with IP addresses being 10.1.*X*.*X*/32 to simulate CE user access.

### 9.1.2 Lab Task

#### 9.1.2.1 Configuration Roadmap

1. Configure IPv6 addresses for devices.

2. Configure IS-IS in the SR domain. Specifically, enable IS-IS on interconnection and Loopback0 interfaces for communication in the SR domain.
3. Create a VPN instance named **vpna**, add Loopback1 to the VPN instance on PE2 and PE4, and import direct routes to the BGP instance.
4. Establish an MP-IBGP peer relationship between PE2 and P2 and another one between PE4 and P2. P2 functions as an RR to reflect VPNv4 routes from PE2 and PE4.
5. Configure SRv6. Specifically, enable SRv6 globally, enable IS-IS extensions for SR, configure the source addresses for SRv6 encapsulation and locators, and enable SID allocation to VPN instance routes as well as the function to add SIDs to routes to be advertised to BGP peers.

## 9.1.2.2 Configuration Procedure

Step 1 Complete basic connectivity configuration.

Configure the command validation mode as immediate validation and configure IP addresses for interconnection and Loopback0 interfaces. Loopback0 interface addresses need to be configured according to the following table.

**Table 9-1 Loopback0 IP addresses**

| Device ID | Value of X | Loopback0 IP Address |
|---|---|---|
| PE2 | 2 | 2001::2:2 |
| PE4 | 4 | 2001::4:4 |
| P2 | 6 | 2001::6:6 |

# Name the devices.

N/A

# Disable unnecessary interfaces in this lab.

N/A

# Configure the command validation mode as immediate validation.

```
<PE2>system-view immediately
```

```
<PE4>system-view immediately
```

```
<P2>system-view immediately
```

# Disable the DCN function globally on all devices.

```
[PE2] undo dcn
Warning: This operation will disable DCN function. Continue? [Y/N]:y
```

Here, PE2 is used as an example. Repeat this operation for other devices.

By default, the DCN function is enabled on NE router interfaces. To facilitate this lab, disable the DCN function globally on all devices.

# Configure IPv6 addresses for the interconnection and Loopback0 interfaces on PE2.

```
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ipv6 enable
[PE2-GigabitEthernet0/5/1] ipv6 address 2001::9/126
[PE2-GigabitEthernet0/5/1] quit
[PE2]interface LoopBack0
[PE2-LoopBack0] ipv6 enable
[PE2-LoopBack0] ipv6 address 2001::2:2/128
[PE2-LoopBack0] quit
```

# Configure IPv6 addresses for the interconnection and Loopback0 interfaces on P2.

```
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ipv6 enable
[P2-GigabitEthernet0/5/1] ipv6 address 2001::a/126
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] ipv6 enable
[P2-GigabitEthernet0/5/2] ipv6 address 2001::15/126
[P2-GigabitEthernet0/5/2] quit
[P2]interface LoopBack0
[P2-LoopBack0] ipv6 enable
[P2-LoopBack0] ipv6 address 2001::6:6/128
[P2-LoopBack0] quit
```

# Configure IPv6 addresses for the interconnection and Loopback0 interfaces on PE4.

```
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] ipv6 enable
[PE4-GigabitEthernet0/5/1] ipv6 address 2001::16/126
[PE4-GigabitEthernet0/5/1] quit
[PE4]interface LoopBack0
[PE4-LoopBack0] ipv6 enable
[PE4-LoopBack0] ipv6 address 2001::4:4/128
[PE4-LoopBack0] quit
```

# Test the connectivity of the IPv6 addresses of the interconnection interfaces on P2.

```
[P2]ping ipv6 -c 1 2001::9
  PING 2001::9 : 56  data bytes, press CTRL_C to break
    Reply from 2001::9
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::9 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
```

```
    0.00% packet loss
    round-trip min/avg/max=1/1/1 ms

[P2]ping ipv6 -c 1 2001::16
  PING 2001::16 : 56  data bytes, press CTRL_C to break
    Reply from 2001::16
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::16 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
round-trip min/avg/max=1/1/1 ms
```

Step 2 Configure IS-IS.

Deploy IS-IS between PE2 and P2 and between P2 and PE4, and set the process ID to 1. All IS-IS routers are Level-2 ones. Set the cost type to **wide** (to support IS-IS extensions) and the area ID of the NET address to 49.0001. Configure the system ID as 49.0001.000*X*.000*X*.000*X*, where *X* indicates the device ID. For details, see the table in Step 1. The IS-IS hostname must be the same as the device name.

Enable IS-IS for interconnection and Loopback0 interfaces on PE2 and PE4, and change the interface type to P2P.

# Configure IS-IS on PE2.

```
[PE2]isis 1
[PE2-isis-1] is-level level-2
[PE2-isis-1] cost-style wide
[PE2-isis-1] network-entity 49.0001.0002.0002.0002.00
[PE2-isis-1] is-name PE2
[PE2-isis-1] ipv6 enable topology ipv6
[PE2-isis-1] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] isis ipv6 enable 1
[PE2-GigabitEthernet0/5/1] isis circuit-type p2p
[PE2-GigabitEthernet0/5/1] quit
[PE2]interface LoopBack 0
[PE2-LoopBack0] isis ipv6 enable 1
```

# Configure IS-IS on P2.

```
[P2]isis 1
[P2-isis-1] is-level level-2
[P2-isis-1] cost-style wide
[P2-isis-1] network-entity 49.0001.0006.0006.0006.00
[P2-isis-1] is-name P2
[P2-isis-1] ipv6 enable topology ipv6
[P2-isis-1] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] isis ipv6 enable 1
[P2-GigabitEthernet0/5/1] isis circuit-type p2p
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
```

```
[P2-GigabitEthernet0/5/2] isis ipv6 enable 1
[P2-GigabitEthernet0/5/2] isis circuit-type p2p
[P2-GigabitEthernet0/5/2] quit
[P2]interface LoopBack 0
[P2-LoopBack0] isis ipv6 enable 1
[P2-LoopBack0] quit
```

# Configure IS-IS on PE4.

```
[PE4]isis 1
[PE4-isis-1] is-level level-2
[PE4-isis-1] cost-style wide
[PE4-isis-1] network-entity 49.0001.0004.0004.0004.00
[PE4-isis-1] is-name PE4
[PE4-isis-1] ipv6 enable topology ipv6
[PE4-isis-1] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] isis ipv6 enable 1
[PE4-GigabitEthernet0/5/1] isis circuit-type p2p
[PE4-GigabitEthernet0/5/1] quit
[PE4]interface LoopBack 0
[PE4-LoopBack0] isis ipv6 enable 1
```

# Check the IS-IS neighbor status on P2.

```
[P2]display isis peer

                    Peer information for ISIS(1)

  System Id     Interface          Circuit Id         State HoldTime  Type     PRI
------------------------------------------------------------------------------
PE2*            GE0/5/1            0000000007         Up    7s        L2       64
PE4*            GE0/5/2            0000000007         Up    9s        L2       64

Total Peer(s): 2
```

The neighbor relationship has been normally established.

# Check IS-IS routes on P2.

```
[P2]display isis route

                  Routeinformation forISIS(1)
                  ---------------------------

                 ISIS(1) Level-2Forwarding Table
                 -------------------------------

IPV6 Dest.    ExitInterface     NextHop                      Cost    Flags
------------------------------------------------------------------------------
2001::8/126   GE0/5/1          Direct                       10      D/-/L/-
2001::14/126  GE0/5/2          Direct                       10      D/-/L/-
2001::2:2/128 GE0/5/1          FE80::F29B:B8FF:FECC:743B    10      A/-/-/-
2001::4:4/128 GE0/5/2          FE80::F29B:B8FF:FECC:77B3    10      A/-/-/-
2001::6:6/128 Loop0            Direct                       0       D/-/L/-
```

```
    Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
           U-Up/Down Bit Set, LP-Local Prefix-Sid
    Protect Type: L-Link Protect, N-Node Protect
```

The routes to the Loopback0 interface on PE2 and PE4 have been properly learned.

Step 3 Configure L3VPN.

On both PE2 and PE4, create a VPN instance named **vpna**, create Loopback1, add the interface to the VPN instance, and establish an MP-BGP VPNv4 peer relationship between PE2 and P2 and another one between PE4 and P2 (the AS number is 65100). P2 functions as an RR. PE2 and PE4 function as RR clients, which advertise VPNv4 routes through P2.

# Create a VPN instance named **vpna**.

```
[PE2]ip vpn-instance vpna
[PE2-vpn-instance-vpna] ipv4-family
[PE2-vpn-instance-vpna-af-ipv4] route-distinguisher 100:20
[PE2-vpn-instance-vpna-af-ipv4] vpn-target 100:1020 both
```

```
[PE4]ip vpn-instance vpna
[PE4-vpn-instance-vpna] ipv4-family
[PE4-vpn-instance-vpna-af-ipv4] route-distinguisher 100:40
[PE4-vpn-instance-vpna-af-ipv4] vpn-target 100:1020 both
```

# Create Loopback1, associate it with the VPN instance, and configure an IP address for it according to the table in Step 1.

```
[PE2]interface LoopBack 1
[PE2-LoopBack1] ip binding vpn-instance vpna
Info: All IPv4 and IPv6 related configurations on this interface are removed.
[PE2-LoopBack1] ip address 10.1.2.2 32
```

```
[PE4]interface LoopBack 1
[PE4-LoopBack1] ip binding vpn-instance vpna
Info: All IPv4 and IPv6 related configurations on this interface are removed.
[PE4-LoopBack1] ip address 10.1.4.4 32
```

Note that you need to associate the interface with the VPN instance before configuring an IP address for the interface.

# Establish MP-BGP VPNv4 peer relationships through Loopback0 and set the router ID to 10.0.*X*.*X*. The values indicated by *X* are listed in the table of Step 1.

```
[PE2]bgp 65100
[PE2-bgp] router-id 10.0.2.2
[PE2-bgp] peer 2001::6:6 as-number 65100
[PE2-bgp] peer 2001::6:6 connect-interface LoopBack 0
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 2001::6:6 enable
```

```
[PE4]bgp 65100
[PE4-bgp]router-id 10.0.4.4
[PE4-bgp]peer 2001::6:6 as-number 65100
[PE4-bgp]peer 2001::6:6 connect-interface LoopBack 0
[PE4-bgp]ipv4-family vpnv4
[PE4-bgp-af-vpnv4]peer 2001::6:6 enable
```

```
[P2]bgp 65100
[P2-bgp] router-id 10.0.6.6
[P2-bgp] peer 2001::2:2 as-number 65100
[P2-bgp] peer 2001::2:2 connect-interface LoopBack 0
[P2-bgp] peer 2001::4:4 as-number 65100
[P2-bgp] peer 2001::4:4 connect-interface LoopBack 0
[P2-bgp] ipv4-family vpnv4
[P2-bgp-af-vpnv4] undo policy vpn-target
[P2-bgp-af-vpnv4] peer 2001::2:2 enable
[P2-bgp-af-vpnv4] peer 2001::2:2 reflect-client
[P2-bgp-af-vpnv4] peer 2001::4:4 enable
[P2-bgp-af-vpnv4] peer 2001::4:4 reflect-client
```

When configuring an RR, disable the RT check on VPNv4 routes.

# Check the VPNv4 peer relationship status.

```
[P2]display bgp vpnv4 all peer

 BGP local router ID : 10.0.6.6
 Local AS number : 65100
 Total number of peers : 2                 Peers in established state : 2

  Peer            V         AS  MsgRcvd  MsgSent  OutQ  Up/Down       State       PrefRcv
  2001::2:2       4      65100         4        4     0  00:00:48    Established         0
  2001::4:4       4      65100         4        5     0  00:00:49    Established         0
```

# Import the direct routes of Loopback1 to BGP.

```
[PE2] bgp 65100
[PE2-bgp] ipv4-family vpn-instance vpna
[PE2-bgp-vpna] import-route direct
```

```
[PE4] bgp 65100
[PE4-bgp] ipv4-family vpn-instance vpna
[PE4-bgp-vpna] import-route direct
```

# Check VPNv4 routes on PE2.

```
 [PE2]display bgp vpnv4 all routing-table
 BGP Local router ID is 10.0.2.2
 Status codes: * - valid, > - best, d - damped, x - best external, a - add path,
               h - history,  i - internal, s - suppressed, S - Stale
```

```
                    Origin : i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V - valid, I - invalid, N - not-found


Total number of routes from all PE: 3
Route Distinguisher: 100:20

         Network            NextHop                MED        LocPrf    PrefVal Path/Ogn

*>       10.1.2.2/32        0.0.0.0                0                    0       ?
*>       127.0.0.0/8        0.0.0.0                0                    0       ?
Route Distinguisher: 100:40

         Network            NextHop                MED        LocPrf    PrefVal Path/Ogn

*>i      10.1.4.4/32        2001::4:4              0          100       0       ?

VPN-Instance   vpna, Router ID 10.0.2.2:

Total Number of Routes: 3
         Network            NextHop                MED        LocPrf    PrefVal Path/Ogn

*>       10.1.2.2/32        0.0.0.0                0                    0       ?
  i      10.1.4.4/32        2001::4:4              0          100       0       ?
*>       127.0.0.0/8        0.0.0.0                0                    0       ?
```

PE2 has learned the VPNv4 route from PE4 through MP-BGP.

Step 4 Configure SRv6 BE.

On PE2 and PE4, enable SRv6 globally, configure the Loopback0 IPv6 addresses as the source addresses for SRv6 encapsulation, configure locators, enable automatic SRv6 SID allocation for VPN routes in the BGP VPN instance, enable the function to add SRv6 SIDs to VPN routes to be advertised in the BGP VPNv4 view, and enable IS-IS to advertise SRv6 locators.

Configure SRv6 locators as planned in the following table.

**Table 9-2 SRv6 locator planning**

| Device | IPv6 Prefix | MASK | Static Segment Length |
|---|---|---|---|
| PE2 | 2001:2:: | 96 | 16 |
| PE4 | 2001:4:: | 96 | 16 |

# Enable SR globally, and configure the source addresses for SR encapsulation and locators.

```
[PE2]segment-routing ipv6
[PE2-segment-routing-ipv6] encapsulation source-address 2001::2:2
[PE2-segment-routing-ipv6] locator SRv6 ipv6-prefix 2001:2:: 96 static 16
```

```
[PE4]segment-routing ipv6
[PE4-segment-routing-ipv6] encapsulation source-address 2001::4:4
[PE4-segment-routing-ipv6] locator SRv6 ipv6-prefix 2001:4:: 96 static 16
```

# Enable the function to add SIDs to VPN routes to be advertised to BGP peers.

```
[PE2]bgp 65100
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpnv4] peer 2001::6:6 prefix-sid
```

```
[PE4]bgp 65100
[PE4-bgp] ipv4-family vpnv4
[PE4-bgp-af-vpnv4] peer 2001::6:6 prefix-sid
```

```
[P2]bgp 65100
[P2-bgp] ipv4-family vpnv4
[P2-bgp-af-vpnv4] peer 2001::2:2 prefix-sid
[P2-bgp-af-vpnv4] peer 2001::4:4 prefix-sid
```

# Enable the function to add SIDs to VPN routes in the BGP VPN instance and specify the previously created SRv6 locator as the locator for allocated SIDs.

```
[PE2]bgp 65100
[PE2-bgp] ipv4-family vpn-instance vpna
[PE2-bgp-vpna] segment-routing ipv6 best-effort
[PE2-bgp-vpna] segment-routing ipv6 locator SRv6
```

```
[PE4]bgp 65100
[PE4-bgp] ipv4-family vpn-instance vpna
[PE4-bgp-vpna] segment-routing ipv6 best-effort
[PE4-bgp-vpna] segment-routing ipv6 locator SRv6
```

# Enable IS-IS to advertise SRv6 locators.

```
[PE2]isis 1
[PE2-isis-1]segment-routing ipv6 locator SRv6
```

```
[PE4]isis 1
[PE4-isis-1]segment-routing ipv6 locator SRv6
```

# Check IS-IS IPv6 routes on P2.

```
[P2]display isis route ipv6
                    Routeinformation forISIS(1)
                    ---------------------------

                   ISIS(1) Level-2Forwarding Table
                   -------------------------------

IPV6 Dest.     ExitInterface      NextHop                    Cost     Flags
--------------------------------------------------------------------------------
2001::8/126    GE0/5/1            Direct                     10       D/-/L/-
2001::14/126   GE0/5/2            Direct                     10       D/-/L/-
2001::2:2/128  GE0/5/1            FE80::F29B:B8FF:FECC:743B  10       A/-/-/-
2001::4:4/128  GE0/5/2            FE80::F29B:B8FF:FECC:77B3  10       A/-/-/-
2001::6:6/128  Loop0              Direct                     0        D/-/L/-
2001:2::/96    GE0/5/1            FE80::F29B:B8FF:FECC:743B  10       A/-/-/-
2001:4::/96    GE0/5/2            FE80::F29B:B8FF:FECC:77B3  10       A/-/-/-
    Flags:D-Direct, A-Addedto URT, L-Advertised in LSPs,S-IGP Shortcut,
          U-Up/Down Bit Set,LP-Local Prefix-Sid
    Protect Type: L-Link Protect, N-Node Protect
```

In the IS-IS IPv6 routing table, you can find routes generated based on locators on PE2 and PE4. Reachability to SIDs generated based on these locators is ensured through these routes.

# Check the SIDs (VPN labels) generated by SRv6 for VPN routes.

```
[PE2]display segment-routing ipv6 local-sid end-dt4 forwarding
                        My Local-SID End.DT4 Forwarding Table
                        -------------------------------------

SID             : 2001:2::1:0/128                    FuncType     : End.DT4
VPN Name        : vpna                                 VPN ID     : 3
LocatorName     : SRv6                                 LocatorID  : 1

Total SID(s): 1
```

PE2 generates the SID 2001:2::1:0 for VPN routes in the VPN instance named **vpna** and sends the SID to PE4 through a BGP Update message.

# On PE4, check detailed information about the VPNv4 route (10.1.2.2) from PE2.

```
[PE4]display bgp vpnv4 all routing-table 10.1.2.2

 BGP local router ID : 10.0.4.4
 Local AS number : 65100

 Total routes of Route Distinguisher(100:20): 1
 BGP routing table entry information of 10.1.2.2/32:
 Label information (Received/Applied): 3/NULL
 From: 2001::6:6 (10.0.6.6)
 Route Duration: 0d00h06m17s
 Relay IP Nexthop: FE80::F29B:B8FF:FECC:740C
 Relay IP Out-Interface: GigabitEthernet0/5/1
 Relay Tunnel Out-Interface:
```

```
  Original nexthop: 2001::2:2
  Qos information : 0x0
  Ext-Community: RT <100 : 1020>
  Prefix-sid: 2001:2::1:0
  AS-path Nil, origin incomplete, MED 0, localpref 100, pref-val 0, valid, internal, best, select, pre 255,
IGP cost 20
  Originator: 10.0.2.2
  Cluster list: 10.0.6.6
  Not advertised to any peer yet

  VPN-Instance vpna, Router ID 10.0.4.4:

  Total Number of Routes: 1
  BGP routing table entry information of 10.1.2.2/32:
  Route Distinguisher: 100:20
  Remote-Cross route
  Label information (Received/Applied): 3/NULL
  From: 2001::6:6 (10.0.6.6)
  Route Duration: 0d00h05m57s
  Relay IP Nexthop: FE80::F29B:B8FF:FECC:740C
  Relay IP Out-Interface: GigabitEthernet0/5/1
  Relay Tunnel Out-Interface:
  Original nexthop: 2001::2:2
  Qos information : 0x0
  Ext-Community: RT <100 : 1020>
  Prefix-sid: 2001:2::1:0
  AS-path Nil, origin incomplete, MED 0, localpref 100, pref-val 0, valid, internal, best, select, pre 255,
IGP cost 20
  Originator: 10.0.2.2
  Cluster list: 10.0.6.6
  Not advertised to any peer yet
```

The command output shows that the BGP VPNv4 route carries the SID.

# Check the VPN instance IP routing table on PE4.

```
[PE4]display  ip  routing-table vpn-instance vpna
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
         Destinations : 4        Routes : 4

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      10.1.2.2/32   IBGP    255  0           RD 2001:2::1:0      SRv6 BE
      10.1.4.4/32   Direct  0    0           D   127.0.0.1       LoopBack1
     127.0.0.0/8    Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

The command output shows that the next hop of the route from PE4 to 10.1.2.2 is 2001:2::1:0, which is the SID allocated by PE2 to VPN routes in the VPN instance.

When the CE (10.1.4.4) connected to PE4 accesses the CE (10.1.2.2) connected to PE2, the destination IPv6 address carried in the outer packet header is this address. After receiving

the packet, PE2 can determine to which CE the inner packet should be sent according to the destination IPv6 address.

# Test the Loopback1 interface connectivity between PE2 and PE4.

```
[PE4]ping -vpn-instance vpna -a 10.1.4.4 10.1.2.2
  PING 10.1.2.2: 56  data bytes, press CTRL_C to break
    Reply from 10.1.2.2: bytes=56 Sequence=1 ttl=255 time=1 ms
    Reply from 10.1.2.2: bytes=56 Sequence=2 ttl=255 time=1 ms
    Reply from 10.1.2.2: bytes=56 Sequence=3 ttl=255 time=1 ms
    Reply from 10.1.2.2: bytes=56 Sequence=4 ttl=255 time=1 ms
    Reply from 10.1.2.2: bytes=56 Sequence=5 ttl=255 time=1 ms

  --- 10.1.2.2 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 1/1/1 ms
```

The communication is normal.

**----End**

## 9.1.3 Quiz

In an L3VPNv6 over SRv6 BE scenario, which type of SID do BGP routes in a VPN instance carry?

# 9.2 EVPN L3VPN over SRv6 Policy

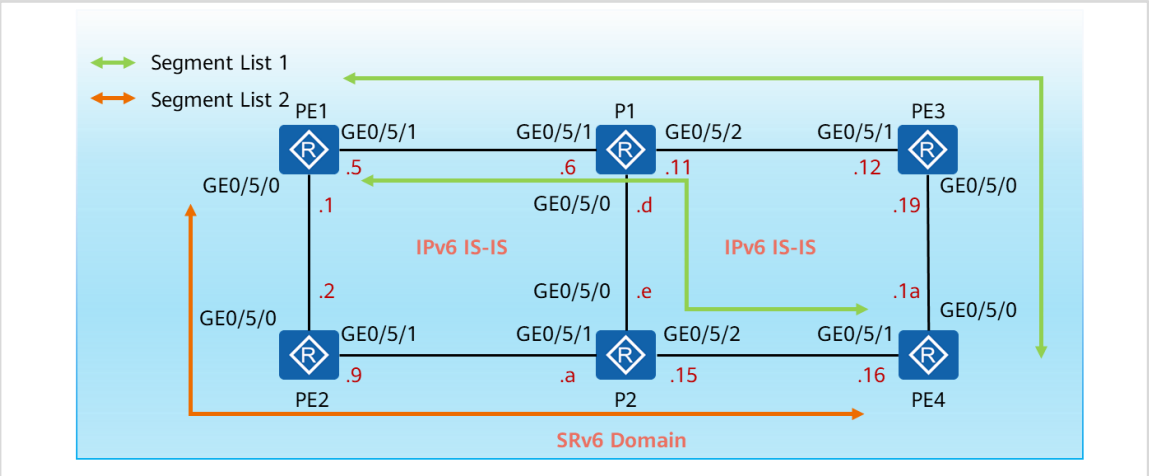
## 9.2.1 About This Lab

### 9.2.1.1 Objectives

- Manually allocate SIDs to VPN routes.
- Recurse L3VPN tunnels used for communication between CEs to SRv6 Policies.
- Configure multiple segment lists to achieve high reliability of the bearer network.
- Configure multiple candidate paths to achieve high reliability of the bearer network.

### 9.2.1.2 Networking Description

**Figure 9-2 EVPN L3VPNv4 over SRv6 Policy topology**




The figure shows the device connection and IP address planning. The interface interconnection addresses are in the format of 2001::*Y*/126, and the values represented by Y are shown in the figure. Loopback0 is created on all devices, and loopback interface IP addresses are in the format of 2001::*X*:*X*/128. The values indicated by *X* are listed in the table of the corresponding step.

Loopback1 is created on PE1 and PE4, with IP addresses being 10.1.*X*.*X*/32 to simulate CE user access. The values indicated by *X* are listed in the table of the corresponding step.

Loopback2 is created on PE1 and PE4, with IPv6 addresses being 2002::*X*:*X*/128 to simulate CE user access. The values indicated by *X* are listed in the table of the corresponding step.

L3VPN traffic between PE1 and PE4 enters the WAN bearer network through SRv6 Policies. Multiple candidate paths are configured to ensure high service reliability. In addition, multiple segment lists are configured for candidate path 1 to ensure high path reliability.

## 9.2.2 Lab Task

### 9.2.2.1 Configuration Roadmap

1. Configure basic IP addresses for devices.

2. Configure IS-IS in the SR domain. Specifically, enable IS-IS on interconnection and Loopback0 interfaces for communication in the SR domain.
3. Create a VPN instance named **vpna**, add Loopback1 to the VPN instance on PE1 and PE4, and import direct routes to the BGP instance. Then, create a VPN instance named **vpna6**, add Loopback2 to the VPN instance on PE1 and PE4, and import direct routes to the BGP instance.
4. Establish an IBGP peer relationship between PE1 and P1, between PE1 and P2, between PE4 and P1, and between PE4 and P2. P1 and P2 function as RRs to reflect VPNv4 routes from PE1 and PE4.
5. Configure a route-policy to allow PE1 and PE4 to add a color value to VPNv4 routes to be advertised to each other.
6. Configure SRv6. Specifically, enable SRv6 globally, enable IS-IS extensions for SR, configure the source addresses for SRv6 encapsulation and locators, manually allocate SIDs to VPN instance routes and SIDs used for device identification to devices, and enable the function to add SIDs to routes to be advertised to BGP peers.
7. Configure a tunnel policy to recurse VPN routes to SRv6 Policies.

## 9.2.2.2 Configuration Procedure

Step 1 Complete basic connectivity configuration.

Configure the command validation mode as immediate validation and configure IP addresses for interconnection and Loopback0 interfaces. Loopback0 interface addresses need to be configured according to the following table.

**Table 9-3 Loopback0 IP addresses**

| Device ID | Value of X | Loopback0 IP Address |
|---|---|---|
| PE1 | 1 | 2001::1:1 |
| PE2 | 2 | 2001::2:2 |
| PE3 | 3 | 2001::3:3 |
| PE4 | 4 | 2001::4:4 |
| P1 | 5 | 2001::5:5 |
| P2 | 6 | 2001::6:6 |

# Name the devices.

N/A

# Disable unnecessary interfaces in this lab.

N/A

# Configure the command validation mode as immediate validation.

```
<PE1>system-view immediately
```

Here, PE1 is used as an example. Repeat the same operations for other devices.

# Disable the DCN function globally on all devices.

```
[PE1] undo dcn
Warning: This operation will disable DCN function. Continue? [Y/N]:y
```

Here, PE1 is used as an example. Repeat the same operations for other devices.

By default, the DCN function is enabled on NE router interfaces. To facilitate this lab, disable the DCN function globally on all devices.

# Configure IPv6 addresses for the interconnection and Loopback0 interfaces on PE1.

```
[PE1]interface LoopBack0
[PE1-LoopBack0] ipv6 enable
[PE1-LoopBack0] ipv6 address 2001::1:1/128
[PE1-LoopBack0] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0] ipv6 enable
[PE1-GigabitEthernet0/5/0] ipv6 address 2001::1/126
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
[PE1-GigabitEthernet0/5/1] ipv6 enable
[PE1-GigabitEthernet0/5/1] ipv6 address 2001::5/126
[PE1-GigabitEthernet0/5/1] quit
```

# Configure IPv6 addresses for the interconnection and Loopback0 interfaces on PE2.

```
[PE2]interface LoopBack0
[PE2-LoopBack0] ipv6 enable
[PE2-LoopBack0] ipv6 address 2001::2:2/128
[PE2-LoopBack0] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] ipv6 enable
[PE2-GigabitEthernet0/5/0] ipv6 address 2001::2/126
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] ipv6 enable
[PE2-GigabitEthernet0/5/1] ipv6 address 2001::9/126
[PE2-GigabitEthernet0/5/1] quit
```

# Configure IPv6 addresses for the interconnection and Loopback0 interfaces on PE3.

```
[PE3]interface LoopBack0
[PE3-LoopBack0] ipv6 enable
[PE3-LoopBack0] ipv6 address 2001::3:3/128
[PE3-LoopBack0] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] ipv6 enable
[PE3-GigabitEthernet0/5/0] ipv6 address 2001::19/126
[PE3-GigabitEthernet0/5/0] quit
[PE3]interface GigabitEthernet0/5/1
[PE3-GigabitEthernet0/5/1] ipv6 enable
[PE3-GigabitEthernet0/5/1] ipv6 address 2001::12/126
```

```
[PE3-GigabitEthernet0/5/1] quit
```

# Configure IPv6 addresses for the interconnection and Loopback0 interfaces on PE4.

```
[PE4]interface LoopBack0
[PE4-LoopBack0] ipv6 enable
[PE4-LoopBack0] ipv6 address 2001::4:4/128
[PE4-LoopBack0] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] ipv6 enable
[PE4-GigabitEthernet0/5/0] ipv6 address 2001::1a/126
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] ipv6 enable
[PE4-GigabitEthernet0/5/1] ipv6 address 2001::16/126
[PE4-GigabitEthernet0/5/1] quit
```

# Configure IPv6 addresses for the interconnection and Loopback0 interfaces on P1.

```
[P1]interface LoopBack0
[P1-LoopBack0] ipv6 enable
[P1-LoopBack0] ipv6 address 2001::5:5/128
[P1-LoopBack0] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] ipv6 enable
[P1-GigabitEthernet0/5/0] ipv6 address 2001::d/126
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] ipv6 enable
[P1-GigabitEthernet0/5/1] ipv6 address 2001::6/126
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] ipv6 enable
[P1-GigabitEthernet0/5/2] ipv6 address 2001::11/126
[P1-GigabitEthernet0/5/2] quit
```

# Configure IPv6 addresses for the interconnection and Loopback0 interfaces on P2.

```
[P2]interface LoopBack0
[P2-LoopBack0] ipv6 enable
[P2-LoopBack0] ipv6 address 2001::6:6/128
[P2-LoopBack0] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] ipv6 enable
[P2-GigabitEthernet0/5/0] ipv6 address 2001::e/126
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] ipv6 enable
[P2-GigabitEthernet0/5/1] ipv6 address 2001::a/126
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] ipv6 enable
[P2-GigabitEthernet0/5/2] ipv6 address 2001::15/126
[P2-GigabitEthernet0/5/2] quit
```

# Test interconnection interface connectivity on P1, PE2, and PE4.

```
[P1]ping ipv6 -c 1 2001::5
  PING 2001::5 : 56  data bytes, press CTRL_C to break
    Reply from 2001::5
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::5 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=1/1/1 ms

[P1]ping ipv6 -c 1 2001::12
  PING 2001::12 : 56  data bytes, press CTRL_C to break
    Reply from 2001::12
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::12 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=1/1/1 ms

[P1]ping ipv6 -c 1 2001::e
  PING 2001::E : 56  data bytes, press CTRL_C to break
    Reply from 2001::E
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::E ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
round-trip min/avg/max=1/1/1 ms
```

```
[PE2]ping ipv6 -c 1 2001::1
  PING 2001::1 : 56  data bytes, press CTRL_C to break
    Reply from 2001::1
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::1 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=1/1/1 ms

[PE2]ping ipv6 -c 1 2001::a
  PING 2001::A : 56  data bytes, press CTRL_C to break
    Reply from 2001::A
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::A ping statistics---
    1 packet(s) transmitted
```

```
    1 packet(s) received
    0.00% packet loss
round-trip min/avg/max=1/1/1 ms
```

```
[PE4]ping ipv6 -c 1 2001::15
  PING 2001::15 : 56  data bytes, press CTRL_C to break
    Reply from 2001::15
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::15 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=1/1/1 ms

[PE4]ping ipv6 -c 1 2001::19
  PING 2001::19 : 56  data bytes, press CTRL_C to break
    Reply from 2001::19
    bytes=56 Sequence=1 hop limit=64 time=1 ms

  --- 2001::19 ping statistics---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
round-trip min/avg/max=1/1/1 ms
```

The connectivity is normal.

Step 2 Configure IS-IS.

Deploy IS-IS in the SRv6 domain and set the process ID to 1. All IS-IS routers are Level-2 ones. Set the cost type to **wide** (to support IS-IS extensions) and the area ID of the NET address to 49.0001. Configure the system ID as 49.0001.000*X*.000*X*.000*X*, where *X* indicates the device ID. For details, see the table in Step 1. The IS-IS hostname must be the same as the device name.

Enable IS-IS for interconnection and Loopback0 interfaces, and change the interface type to P2P.

# Configure IS-IS on PE1.

```
[PE1]isis 1
[PE1-isis-1] is-level level-2
[PE1-isis-1] cost-style wide
[PE1-isis-1] network-entity 49.0001.0001.0001.0001.00
[PE1-isis-1] is-name PE1
[PE1-isis-1] ipv6 enable topology ipv6
[PE1-isis-1] quit
[PE1]interface GigabitEthernet0/5/0
[PE1-GigabitEthernet0/5/0] isis ipv6 enable 1
[PE1-GigabitEthernet0/5/0] isis circuit-type p2p
[PE1-GigabitEthernet0/5/0] quit
[PE1]interface GigabitEthernet0/5/1
```

```
[PE1-GigabitEthernet0/5/1] isis ipv6 enable 1
[PE1-GigabitEthernet0/5/1] isis circuit-type p2p
[PE1-GigabitEthernet0/5/1] quit
[PE1]interface LoopBack 0
[PE1-LoopBack0] isis ipv6 enable 1
```

# Configure IS-IS on PE2.

```
[PE2]isis 1
[PE2-isis-1] is-level level-2
[PE2-isis-1] cost-style wide
[PE2-isis-1] network-entity 49.0001.0002.0002.0002.00
[PE2-isis-1] is-name PE2
[PE2-isis-1] ipv6 enable topology ipv6
[PE2-isis-1] quit
[PE2]interface GigabitEthernet0/5/0
[PE2-GigabitEthernet0/5/0] isis ipv6 enable 1
[PE2-GigabitEthernet0/5/0] isis circuit-type p2p
[PE2-GigabitEthernet0/5/0] quit
[PE2]interface GigabitEthernet0/5/1
[PE2-GigabitEthernet0/5/1] isis ipv6 enable 1
[PE2-GigabitEthernet0/5/1] isis circuit-type p2p
[PE2-GigabitEthernet0/5/1] quit
[PE2]interface LoopBack 0
[PE2-LoopBack0] isis ipv6 enable 1
```

# Configure IS-IS on PE3.

```
[PE3]isis 1
[PE3-isis-1] is-level level-2
[PE3-isis-1] cost-style wide
[PE3-isis-1] network-entity 49.0001.0003.0003.0003.00
[PE3-isis-1] is-name PE3
[PE3-isis-1] ipv6 enable topology ipv6
[PE3-isis-1] quit
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] isis ipv6 enable 1
[PE3-GigabitEthernet0/5/0] isis circuit-type p2p
[PE3-GigabitEthernet0/5/0] quit
[PE3]interface GigabitEthernet0/5/1
[PE3-GigabitEthernet0/5/1] isis ipv6 enable 1
[PE3-GigabitEthernet0/5/1] isis circuit-type p2p
[PE3-GigabitEthernet0/5/1] quit
[PE3]interface LoopBack 0
[PE3-LoopBack0] isis ipv6 enable 1
```

# Configure IS-IS on PE4.

```
[PE4]isis 1
[PE4-isis-1] is-level level-2
[PE4-isis-1] cost-style wide
[PE4-isis-1] network-entity 49.0001.0004.0004.0004.00
[PE4-isis-1] is-name PE4
[PE4-isis-1] ipv6 enable topology ipv6
```

```
[PE4-isis-1] quit
[PE4]interface GigabitEthernet0/5/0
[PE4-GigabitEthernet0/5/0] isis ipv6 enable 1
[PE4-GigabitEthernet0/5/0] isis circuit-type p2p
[PE4-GigabitEthernet0/5/0] quit
[PE4]interface GigabitEthernet0/5/1
[PE4-GigabitEthernet0/5/1] isis ipv6 enable 1
[PE4-GigabitEthernet0/5/1] isis circuit-type p2p
[PE4-GigabitEthernet0/5/1] quit
[PE4]interface LoopBack 0
[PE4-LoopBack0] isis ipv6 enable 1
```

# Configure IS-IS on P1.

```
[P1]isis 1
[P1-isis-1] is-level level-2
[P1-isis-1] cost-style wide
[P1-isis-1] network-entity 49.0001.0005.0005.0005.00
[P1-isis-1] is-name P1
[P1-isis-1] ipv6 enable topology ipv6
[P1-isis-1] quit
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] isis ipv6 enable 1
[P1-GigabitEthernet0/5/0] isis circuit-type p2p
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] isis ipv6 enable 1
[P1-GigabitEthernet0/5/1] isis circuit-type p2p
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] isis ipv6 enable 1
[P1-GigabitEthernet0/5/2] isis circuit-type p2p
[P1-GigabitEthernet0/5/2] quit
[P1]interface LoopBack 0
[P1-LoopBack0] isis ipv6 enable 1
```

# Configure IS-IS on P2.

```
[P2]isis 1
[P2-isis-1] is-level level-2
[P2-isis-1] cost-style wide
[P2-isis-1] network-entity 49.0001.0006.0006.0006.00
[P2-isis-1] is-name P2
[P2-isis-1] ipv6 enable topology ipv6
[P2-isis-1] quit
[P2]interface GigabitEthernet0/5/0
[P2-GigabitEthernet0/5/0] isis ipv6 enable 1
[P2-GigabitEthernet0/5/0] isis circuit-type p2p
[P2-GigabitEthernet0/5/0] quit
[P2]interface GigabitEthernet0/5/1
[P2-GigabitEthernet0/5/1] isis ipv6 enable 1
[P2-GigabitEthernet0/5/1] isis circuit-type p2p
[P2-GigabitEthernet0/5/1] quit
[P2]interface GigabitEthernet0/5/2
[P2-GigabitEthernet0/5/2] isis ipv6 enable 1
```

```
[P2-GigabitEthernet0/5/2] isis circuit-type p2p
[P2-GigabitEthernet0/5/2] quit
[P2]interface LoopBack 0
[P2-LoopBack0] isis ipv6 enable 1
[P2-LoopBack0] quit
```

# Check the IS-IS neighbor relationship on P1, PE2, and PE4.

```
[P1]display isis peer

                    Peer information for ISIS(1)

  System Id     Interface          Circuit Id          State HoldTime  Type     PRI
------------------------------------------------------------------------------
P2*             GE0/5/0            0000000006          Up    26s       L2       --
PE1*            GE0/5/1            0000000007          Up    25s       L2       --
PE3*            GE0/5/2            0000000007          Up    27s       L2       --

Total Peer(s): 3
```

```
[PE2]display isis peer

                    Peer information for ISIS(1)

  System Id     Interface          Circuit Id          State HoldTime  Type     PRI
------------------------------------------------------------------------------
PE1*            GE0/5/0            0000000006          Up    26s       L2       --
P2*             GE0/5/1            0000000007          Up    25s       L2       --

Total Peer(s): 2
```

```
[PE4]display isis peer

                    Peer information for ISIS(1)

  System Id     Interface          Circuit Id          State HoldTime  Type     PRI
------------------------------------------------------------------------------
PE3*            GE0/5/0            0000000006          Up    23s       L2       --
P2*             GE0/5/1            0000000007          Up    23s       L2       --

Total Peer(s): 2
```

# Check IS-IS routes on P1.

```
[P1]display isis route
                  Routeinformation forISIS(1)
                  ---------------------------

                 ISIS(1) Level-2Forwarding Table
                 -------------------------------
```

```
IPV6 Dest.     ExitInterface      NextHop                    Cost    Flags
------------------------------------------------------------------------------
2001::/126     GE0/5/1            FE80::F29B:B8FF:FECC:7453  20      A/-/-/-
2001::4/126    GE0/5/1            Direct                     10      D/-/L/-
2001::8/126    GE0/5/0            FE80::F29B:B8FF:FECC:740A  20      A/-/-/-
2001::C/126    GE0/5/0            Direct                     10      D/-/L/-
2001::10/126   GE0/5/2            Direct                     10      D/-/L/-
2001::14/126   GE0/5/0            FE80::F29B:B8FF:FECC:740A  20      A/-/-/-
2001::18/126   GE0/5/2            FE80::F29B:B8FF:FECC:70C3  20      A/-/-/-
2001::1:1/128  GE0/5/1            FE80::F29B:B8FF:FECC:7453  10      A/-/-/-
2001::2:2/128  GE0/5/1            FE80::F29B:B8FF:FECC:7453  20      A/-/-/-
               GE0/5/0            FE80::F29B:B8FF:FECC:740A
2001::3:3/128  GE0/5/2            FE80::F29B:B8FF:FECC:70C3  10      A/-/-/-
2001::4:4/128  GE0/5/2            FE80::F29B:B8FF:FECC:70C3  20      A/-/-/-
               GE0/5/0            FE80::F29B:B8FF:FECC:740A
2001::5:5/128  Loop0              Direct                     0       D/-/L/-
2001::6:6/128  GE0/5/0            FE80::F29B:B8FF:FECC:740A  10      A/-/-/-
  Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
         U-Up/Down Bit Set, LP-Local Prefix-Sid
  Protect Type: L-Link Protect, N-Node Protect
```

All routes have been properly learned.

Step 3 Configure L3VPN.

On both PE1 and PE4, create a VPN instance named **vpna** and the Loopback1 interface, and add the interface to the VPN instance.

On both PE1 and PE4, create a VPN instance named **vpna6**, add Loopback2 to the VPN instance, and import direct routes to the BGP instance.

Establish an MP-BGP EVPN peer relationship between PE1 and P1, between PE1 and P2, between PE4 and P1, and between PE4 and P2 (the AS number is 65100). P1 and P2 function as RRs. PE1 and PE4 function as RR clients, which advertise EVPN routes through P1 and P2.

# Create a VPN instance named **vpna**.

```
[PE1]ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] route-distinguisher 100:10
[PE1-vpn-instance-vpna-af-ipv4] vpn-target 100:1020 evpn
[PE1-vpn-instance-vpna-af-ipv4] evpn mpls routing-enable
```

```
[PE4]ip vpn-instance vpna
[PE4-vpn-instance-vpna] ipv4-family
[PE4-vpn-instance-vpna-af-ipv4] route-distinguisher 100:40
[PE4-vpn-instance-vpna-af-ipv4] vpn-target 100:1020 evpn
[PE4-vpn-instance-vpna-af-ipv4] evpn mpls routing-enable
```

You only need to configure EVPN RTs. In addition, enable EVPN to generate and advertise IP prefix routes and IRB routes.

# Create Loopback1, associate it with the VPN instance, and configure an IP address for it according to the table in Step 1.

```
[PE1]interface LoopBack 1
[PE1-LoopBack1]ip binding vpn-instance vpna
Info: All IPv4 and IPv6 related configurations on this interface are removed.
[PE1-LoopBack1]ip address 10.1.1.1 32
```

```
[PE4]interface LoopBack 1
[PE4-LoopBack1]ip binding vpn-instance vpna
Info: All IPv4 and IPv6 related configurations on this interface are removed.
[PE4-LoopBack1]ip address 10.1.4.4 32
```

Note that you need to associate the interface with the VPN instance before configuring an IP address for the interface.

# Create a VPN instance named **vpna6**.

```
[PE1]ip vpn-instance vpna6
[PE1-vpn-instance-vpna6] ipv6-family
[PE1-vpn-instance-vpna6-af-ipv6] route-distinguisher 100:11
[PE1-vpn-instance-vpna6-af-ipv6] vpn-target 100:1122 evpn
[PE1-vpn-instance-vpna6-af-ipv6] evpn mpls routing-enable
```

```
[PE4]ip vpn-instance vpna6
[PE4-vpn-instance-vpna6] ipv6-family
[PE4-vpn-instance-vpna6-af-ipv6] route-distinguisher 100:44
[PE4-vpn-instance-vpna6-af-ipv6] vpn-target 100:1122 evpn
[PE4-vpn-instance-vpna6-af-ipv6] evpn mpls routing-enable
```

You only need to configure EVPN RTs. In addition, enable EVPN to generate and advertise IP prefix routes and IRB routes.

# Create Loopback2, associate it with the VPN instance, and configure an IP address for it according to the table in Step 1.

```
[PE1]interface LoopBack 2
[PE1-LoopBack2]ip binding vpn-instance vpna6
Info: All IPv4 and IPv6 related configurations on this interface are removed.
[PE1-LoopBack2]ipv6 enable
[PE1-LoopBack2]ipv6 address 2002::1:1 128
```

```
[PE4]interface LoopBack 2
[PE4-LoopBack2]ip binding vpn-instance vpna6
Info: All IPv4 and IPv6 related configurations on this interface are removed.
[PE4-LoopBack2]ipv6 enable
[PE4-LoopBack2]ipv6 address 2002::4:4 128
```

Note that you need to associate the interface with the VPN instance before configuring an IP address for the interface.

# Establish MP-BGP EVPN peer relationships through Loopback0 and set the router ID to 10.0.*X.X*. The values indicated by *X* are listed in the table of Step 1.

```
[PE1]bgp 65100
[PE1-bgp] router-id 10.0.1.1
[PE1-bgp] peer 2001::5:5 as-number 65100
[PE1-bgp] peer 2001::5:5 connect-interface LoopBack 0
[PE1-bgp] peer 2001::6:6 as-number 65100
[PE1-bgp] peer 2001::6:6 connect-interface LoopBack 0
[PE1-bgp] l2vpn-family evpn
[PE1-bgp-af-evpn] peer 2001::5:5 enable
[PE1-bgp-af-evpn] peer 2001::5:5 advertise encap-type srv6
[PE1-bgp-af-evpn] peer 2001::6:6 enable
[PE1-bgp-af-evpn] peer 2001::6:6 advertise encap-type srv6
```

By default, EVPN routes advertised by a local device to its peers carry the MPLS encapsulation attribute, which cannot be used for SRv6 forwarding. To enable EVPN routes to recurse to SRv6 tunnels, run the **advertise encap-type srv6** command.

```
[PE4]bgp 65100
[PE4-bgp] router-id 10.0.4.4
[PE4-bgp] peer 2001::5:5 as-number 65100
[PE4-bgp] peer 2001::5:5 connect-interface LoopBack0
[PE4-bgp] peer 2001::6:6 as-number 65100
[PE4-bgp] peer 2001::6:6 connect-interface LoopBack0
[PE4-bgp] l2vpn-family evpn
[PE4-bgp-af-evpn] peer 2001::5:5 enable
[PE4-bgp-af-evpn] peer 2001::5:5 advertise encap-type srv6
[PE4-bgp-af-evpn] peer 2001::6:6 enable
[PE4-bgp-af-evpn] peer 2001::6:6 advertise encap-type srv6
```

```
[P1]bgp 65100
[P1-bgp] router-id 10.0.5.5
[P1-bgp] peer 2001::1:1 as-number 65100
[P1-bgp] peer 2001::1:1 connect-interface LoopBack0
[P1-bgp] peer 2001::4:4 as-number 65100
[P1-bgp] peer 2001::4:4 connect-interface LoopBack0
[P1-bgp] l2vpn-family evpn
[P1-bgp-af-evpn] undo policy vpn-target
[P1-bgp-af-evpn] peer 2001::1:1 enable
[P1-bgp-af-evpn] peer 2001::1:1 advertise encap-type srv6
[P1-bgp-af-evpn] peer 2001::1:1 reflect-client
[P1-bgp-af-evpn] peer 2001::4:4 enable
[P1-bgp-af-evpn] peer 2001::4:4 advertise encap-type srv6
[P1-bgp-af-evpn] peer 2001::4:4 reflect-client
```

```
[P2]bgp 65100
```

```
[P2-bgp] router-id 10.0.6.6
[P2-bgp] peer 2001::1:1 as-number 65100
[P2-bgp] peer 2001::1:1 connect-interface LoopBack0
[P2-bgp] peer 2001::4:4 as-number 65100
[P2-bgp] peer 2001::4:4 connect-interface LoopBack0
[P2-bgp] l2vpn-family evpn
[P2-bgp-af-evpn] undo policy vpn-target
[P2-bgp-af-evpn] peer 2001::1:1 enable
[P2-bgp-af-evpn] peer 2001::1:1 advertise encap-type srv6
[P2-bgp-af-evpn] peer 2001::1:1 reflect-client
[P2-bgp-af-evpn] peer 2001::4:4 enable
[P2-bgp-af-evpn] peer 2001::4:4 advertise encap-type srv6
[P2-bgp-af-evpn] peer 2001::4:4 reflect-client
```

# Check the MP-BGP peer relationships on P1 and P2.

```
[P1]display bgp evpn peer

 BGP local router ID : 10.0.5.5
 Local AS number : 65100
 Total number of peers : 2                 Peers in established state : 2

  Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State  PrefRcv
  2001::1:1       4       65100        6        6     0  00:00:48 Established        0
  2001::4:4       4       65100        6        6     0  00:00:49 Established        0
```

```
[P2]display bgp evpn peer

 BGP local router ID : 10.0.6.6
 Local AS number : 65100
 Total number of peers : 2                 Peers in established state : 2

  Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State  PrefRcv
  2001::1:1       4       65100        6        6     0  00:00:48 Established        0
  2001::4:4       4       65100        6        6     0  00:00:49 Established        0
```

# Import the direct routes of Loopback1 to BGP.

```
[PE1]bgp 65100
[PE1-bgp]ipv4-family vpn-instance vpna
[PE1-bgp-vpna] import-route direct
[PE1-bgp-vpna] advertise l2vpn evpn
```

```
[PE4]bgp 65100
[PE4-bgp]ipv4-family vpn-instance vpna
[PE4-bgp-vpna] import-route direct
[PE4-bgp-vpna] advertise l2vpn evpn
```

Note that you need to enable the VPN instance to advertise IP routes to the EVPN instance.

Because SRv6 has not been configured, EVPN routes do not carry any SRv6 parameter. As such, information about the advertised EVPN routes cannot be displayed at present.

# Import the direct routes of Loopback2 to BGP.

```
[PE1] bgp 65100
[PE1-bgp] ipv6-family vpn-instance vpna6
[PE1-bgp-6-vpna6] import-route direct
[PE1-bgp-6-vpna6] advertise l2vpn evpn
```

```
[PE4] bgp 65100
[PE4-bgp] ipv6-family vpn-instance vpna6
[PE4-bgp-6-vpna6] import-route direct
[PE4-bgp-6-vpna6] advertise l2vpn evpn
```

Step 4 Configure SRv6.

Enable SRv6 globally on all devices, configure Loopback0 IPv6 addresses as source addresses for SRv6 encapsulation, configure locators, and manually allocate SIDs used for device identification to devices and SIDs to VPN routes in the VPN instance. Enable the function to add SRv6 SIDs to VPN routes to be advertised in the BGP L2VPN view and the function to advertise SRv6 locators through IS-IS.

Configure SRv6 locators as planned in the following table.

**Table 9-4 SRv6 locator planning**

| Device | IPv6 Prefix | MASK | Static Segment Length |
|---|---|---|---|
| PE1 | 2001:1:: | 96 | 16 |
| PE2 | 2001:2:: | 96 | 16 |
| PE3 | 2001:3:: | 96 | 16 |
| PE4 | 2001:4:: | 96 | 16 |
| P1 | 2001:5:: | 96 | 16 |
| P2 | 2001:6:: | 96 | 16 |

# Enable SR globally, and configure the source addresses for SR encapsulation and locators.

```
[PE1]segment-routing ipv6
[PE1-segment-routing-ipv6] encapsulation source-address 2001::1:1
[PE1-segment-routing-ipv6] locator SRv6 ipv6-prefix 2001:1:: 96 static 16
[PE1-segment-routing-ipv6-locator] opcode ::1 end
[PE1-segment-routing-ipv6-locator] opcode ::11 end-dt4 vpn-instance vpna evpn
```

```
[PE1-segment-routing-ipv6-locator] opcode ::61 end-dt6 vpn-instance vpna6 evpn
```

Opcodes of the End type are configured, and End-DT4 and End-DT6 SIDs are manually allocated to VPN instances on PE1 and PE4.

```
[PE2]segment-routing ipv6
[PE2-segment-routing-ipv6] encapsulation source-address 2001::2:2
[PE2-segment-routing-ipv6] locator SRv6 ipv6-prefix 2001:2:: 96 static 16
[PE2-segment-routing-ipv6-locator] opcode ::1 end
```

```
[PE3]segment-routing ipv6
[PE3-segment-routing-ipv6] encapsulation source-address 2001::3:3
[PE3-segment-routing-ipv6] locator SRv6 ipv6-prefix 2001:3:: 96 static 16
[PE3-segment-routing-ipv6-locator] opcode ::1 end
```

```
[PE4]segment-routing ipv6
[PE4-segment-routing-ipv6] encapsulation source-address 2001::4:4
[PE4-segment-routing-ipv6] locator SRv6 ipv6-prefix 2001:4:: 96 static 16
[PE4-segment-routing-ipv6-locator] opcode ::1 end
[PE4-segment-routing-ipv6-locator] opcode ::44 end-dt4 vpn-instance vpna evpn
[PE4-segment-routing-ipv6-locator] opcode ::64 end-dt6 vpn-instance vpna6 evpn
```

```
[P1]segment-routing ipv6
[P1-segment-routing-ipv6] encapsulation source-address 2001::5:5
[P1-segment-routing-ipv6] locator SRv6 ipv6-prefix 2001:5:: 96 static 16
[P1-segment-routing-ipv6-locator] opcode ::1 end
```

```
[P2]segment-routing ipv6
[P2-segment-routing-ipv6] encapsulation source-address 2001::6:6
[P2-segment-routing-ipv6] locator SRv6 ipv6-prefix 2001:6:: 96 static 16
[P2-segment-routing-ipv6-locator] opcode ::1 end
```

# In the BGP VPN instance, enable the function to recurse services to an SRv6 Policy.

```
[PE1]bgp 65100
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-vpna] segment-routing ipv6 locator SRv6 evpn
[PE1-bgp-vpna] segment-routing ipv6 traffic-engineer best-effort evpn
[PE1-bgp-vpna] quit
[PE1-bgp] ipv6-family vpn-instance vpna6
[PE1-bgp-6-vpna6] segment-routing ipv6 locator SRv6 evpn
[PE1-bgp-6-vpna6] segment-routing ipv6 traffic-engineer best-effort evpn
```

```
[PE4-bgp] ipv4-family vpn-instance vpna
[PE4-bgp-vpna] segment-routing ipv6 locator SRv6 evpn
[PE4-bgp-vpna] segment-routing ipv6 traffic-engineer best-effort evpn
[PE4-bgp-vpna] quit
[PE4-bgp] ipv6-family vpn-instance vpna6
[PE4-bgp-6-vpna6] segment-routing ipv6 locator SRv6 evpn
[PE4-bgp-6-vpna6] segment-routing ipv6 traffic-engineer best-effort evpn
```

Run the **segment-routing ipv6 traffic-engineer best-effort evpn** command to enable the function to recurse EVPN routes in a VPN instance to an SRv6 Policy and use an SRv6 BE tunnel as a backup best-effort path.

# Enable IS-IS to advertise SRv6 locators.

```
[PE1]isis 1
[PE1-isis-1] segment-routing ipv6 locator SRv6 auto-sid-disable
```

PE1 is used as an example. The operations on other devices are similar to those on PE1. Because End SIDs are manually allocated, the automatic allocation function is disabled here.

# Check the End SID. P1 is used as an example.

```
[P1]display segment-routing ipv6 local-sid end forwarding

                    My Local-SID End Forwarding Table
                    ---------------------------------

SID           : 2001:5::1/128                          FuncType : End
Flavor        : PSP
LocatorName: SRv6                                        LocatorID: 1

Total SID(s): 1
```

End SIDs will be used to configure forwarding paths in SRv6 Policies.

# Check the End.DT4 SID on PE1.

```
[PE1]display segment-routing ipv6 local-sid end-dt4 forwarding
                    My Local-SID End.DT4 Forwarding Table
                    -------------------------------------

SID              : 2001:1::11/128                        FuncType : End.DT4
VPN Name     : vpna                                      VPN ID     : 3
LocatorName : SRv6                                       LocatorID: 1

Total SID(s): 1
```

# Check the SID carried in the EVPN route 0:10.1.1.1:32 of PE4.

```
[PE4]display bgp evpn all routing-table prefix-route 0:10.1.1.1:32 | include Prefix
 BGP local router ID : 10.0.4.4
 Local AS number : 65100
 Prefix-sid: 2001:1::11
 Route Type: 5 (Ip Prefix Route)
```

```
 Ethernet Tag ID: 0, IP Prefix/Len: 10.1.1.1/32, ESI: 0000.0000.0000.0000.0000, GW IP Address: 0.0.0.0
 Prefix-sid: 2001:1::11
 Route Type: 5 (Ip Prefix Route)
 Ethernet Tag ID: 0, IP Prefix/Len: 10.1.1.1/32, ESI: 0000.0000.0000.0000.0000, GW IP Address: 0.0.0.0
```

The prefix SID carried in the EVPN route received by PE4 is the same as the End.DT4 SID allocated by PE1, which meets the expectation.

# Check the End.DT6 SID on PE1.

```
[PE1-isis-1]display segment-routing ipv6 local-sid end-dt6 forwarding
                        My Local-SID End.DT6 Forwarding Table
                        ------------------------------------

SID                : 2001:1::61/128                          FuncType : End.DT6
VPN Name     : vpna6                                         VPN ID     : 3
LocatorName: SRv6                                          LocatorID: 1

Total SID(s): 1
```

# Check the SID carried in the EVPN route 0:[2002::1:1]:128 of PE4.

[PE4]display bgp evpn all routing-table prefix-route 0:[2002::1:1]:128 | include Prefix

```
 BGP local router ID : 10.0.4.4
 Local AS number : 65100
 Prefix-sid: 2001:1::61
 Route Type: 5 (Ip Prefix Route)
 Ethernet Tag ID: 0, IPv6 Prefix/Len: 2002::1:1/128, ESI: 0000.0000.0000.0000.0000, GW IPv6
Address: ::
 Prefix-sid: 2001:1::61
 Route Type: 5 (Ip Prefix Route)
 Ethernet Tag ID: 0, IPv6 Prefix/Len: 2002::1:1/128, ESI: 0000.0000.0000.0000.0000, GW IPv6
Address: ::
```

The prefix SID carried in the EVPN route received by PE4 is the same as the End.DT6 SID allocated by PE1, which meets the expectation.

Step 5 Configure SRv6 Policies and a tunnel policy.

On both PE1 and PE4, configure two candidate paths and three segment lists as planned, and reference the candidate paths in the SRv6 Policies.

# Configure candidate paths on PE1.

```
[PE1]segment-routing ipv6
[PE1-segment-routing-ipv6] segment-list PE1_PE4_VPNA_1
[PE1-segment-routing-ipv6-segment-list-PE1_PE4_VPNA_1] index 5 sid ipv6 2001:5::1
[PE1-segment-routing-ipv6-segment-list-PE1_PE4_VPNA_1] index 10 sid ipv6 2001:3::1
[PE1-segment-routing-ipv6-segment-list-PE1_PE4_VPNA_1] index 15 sid ipv6 2001:4::1
[PE1-segment-routing-ipv6-segment-list-PE1_PE4_VPNA_1] quit
[PE1-segment-routing-ipv6] segment-list PE1_PE4_VPNA_2
[PE1-segment-routing-ipv6-segment-list-PE1_PE4_VPNA_2] index 5 sid ipv6 2001:5::1
[PE1-segment-routing-ipv6-segment-list-PE1_PE4_VPNA_2] index 10 sid ipv6 2001:6::1
[PE1-segment-routing-ipv6-segment-list-PE1_PE4_VPNA_2] index 15 sid ipv6 2001:4::1
```

```
[PE1-segment-routing-ipv6-segment-list-PE1_PE4_VPNA_2] quit
[PE1-segment-routing-ipv6] segment-list PE1_PE4_VPNA_3
[PE1-segment-routing-ipv6-segment-list-PE1_PE4_VPNA_3] index 5 sid ipv6 2001:2::1
[PE1-segment-routing-ipv6-segment-list-PE1_PE4_VPNA_3] index 10 sid ipv6 2001:6::1
[PE1-segment-routing-ipv6-segment-list-PE1_PE4_VPNA_3] index 15 sid ipv6 2001:4::1
[PE1-segment-routing-ipv6-segment-list-PE1_PE4_VPNA_3] quit
[PE1-segment-routing-ipv6] srv6-te policy p1 endpoint 2001::4:4 color 100
[PE1-segment-routing-ipv6-policy-p1] candidate-path preference 100
[PE1-segment-routing-ipv6-policy-p1-path] segment-list PE1_PE4_VPNA_1 weight 2
[PE1-segment-routing-ipv6-policy-p1-path] segment-list PE1_PE4_VPNA_2 weight 1
[PE1-segment-routing-ipv6-policy-p1-path] quit
[PE1-segment-routing-ipv6-policy-p1] candidate-path preference 50
[PE1-segment-routing-ipv6-policy-p1-path] segment-list PE1_PE4_VPNA_3
[PE1-segment-routing-ipv6-policy-p1-path] quit
```

To reach PE4, segment lists 1, 2, and 3 need to pass through P1 and PE3, P1 and P2, and PE2 and P2, respectively. Segment lists 1 and 2 form candidate path 1, and segment list 3 forms candidate path 2.

# Configure candidate paths on PE4.

```
[PE4]segment-routing ipv6
[PE4-segment-routing-ipv6] segment-list PE4_PE1_VPNA_1
[PE4-segment-routing-ipv6-segment-list-PE4_PE1_VPNA_1] index 5 sid ipv6 2001:3::1
[PE4-segment-routing-ipv6-segment-list-PE4_PE1_VPNA_1] index 10 sid ipv6 2001:5::1
[PE4-segment-routing-ipv6-segment-list-PE4_PE1_VPNA_1] index 15 sid ipv6 2001:1::1
[PE4-segment-routing-ipv6-segment-list-PE4_PE1_VPNA_1] quit
[PE4-segment-routing-ipv6] segment-list PE4_PE1_VPNA_2
[PE4-segment-routing-ipv6-segment-list-PE4_PE1_VPNA_2] index 5 sid ipv6 2001:6::1
[PE4-segment-routing-ipv6-segment-list-PE4_PE1_VPNA_2] index 10 sid ipv6 2001:5::1
[PE4-segment-routing-ipv6-segment-list-PE4_PE1_VPNA_2] index 15 sid ipv6 2001:1::1
[PE4-segment-routing-ipv6-segment-list-PE4_PE1_VPNA_2] quit
[PE4-segment-routing-ipv6] segment-list PE4_PE1_VPNA_3
[PE4-segment-routing-ipv6-segment-list-PE4_PE1_VPNA_3] index 5 sid ipv6 2001:6::1
[PE4-segment-routing-ipv6-segment-list-PE4_PE1_VPNA_3] index 10 sid ipv6 2001:2::1
[PE4-segment-routing-ipv6-segment-list-PE4_PE1_VPNA_3] index 15 sid ipv6 2001:1::1
[PE4-segment-routing-ipv6-segment-list-PE4_PE1_VPNA_3] quit
[PE4-segment-routing-ipv6] srv6-te policy p1 endpoint 2001::1:1 color 100
[PE4-segment-routing-ipv6-policy-p1] candidate-path preference 100
[PE4-segment-routing-ipv6-policy-p1-path] segment-list PE4_PE1_VPNA_1 weight 2
[PE4-segment-routing-ipv6-policy-p1-path] segment-list PE4_PE1_VPNA_2 weight 1
[PE4-segment-routing-ipv6-policy-p1-path] quit
[PE4-segment-routing-ipv6-policy-p1] candidate-path preference 50
[PE4-segment-routing-ipv6-policy-p1-path] segment-list PE4_PE1_VPNA_3
[PE4-segment-routing-ipv6-policy-p1-path] quit
```

To reach PE1, segment lists 1, 2, and 3 need to pass through PE3 and P1, P2 and P1, and P2 and PE2, respectively. Segment lists 1 and 2 form candidate path 1, and segment list 3 forms candidate path 2.

# Configure a tunnel selection policy.

```
[PE1]tunnel-policy p1
[PE1-tunnel-policy-p1] tunnel select-seq ipv6 srv6-te-policy load-balance-number 1
```

```
[PE4]tunnel-policy p1
[PE4-tunnel-policy-p1] tunnel select-seq ipv6 srv6-te-policy load-balance-number 1
```

# Apply the tunnel selection policy to the VPN instance.

```
[PE1]ip vpn-instance vpna
[PE1-vpn-instance-vpna]tnl-policy p1 evpn
```

```
[PE4]ip vpn-instance vpna
[PE4-vpn-instance-vpna]tnl-policy p1 evpn
```

```
[PE1] ip vpn-instance vpna6
[PE1-vpn-instance-vpna6] ipv6-family
[PE1-vpn-instance-vpna6-af-ipv6] tnl-policy p1 evpn
```

```
[PE4] ip vpn-instance vpna6
[PE4-vpn-instance-vpna6] ipv6-family
[PE4-vpn-instance-vpna6-af-ipv6] tnl-policy p1 evpn
```

# Set the default color for the route in the VPN instance.

```
[PE1]ip vpn-instance vpna
[PE1-vpn-instance-vpna] ipv4-family
[PE1-vpn-instance-vpna-af-ipv4] default-color 100 evpn
```

```
[PE4]ip vpn-instance vpna
[PE4-vpn-instance-vpna] ipv4-family
[PE4-vpn-instance-vpna-af-ipv4] default-color 100 evpn
```

To enable the route in the VPN instance to recurse to the corresponding SRv6 Policy, you can use a route-policy to set a color value for the route or set the default color value for the route in the local VPN instance. The second method is used in this example.

```
[PE1] ip vpn-instance vpna6
[PE1-vpn-instance-vpna6] ipv6-family
[PE1-vpn-instance-vpna6-af-ipv6] default-color 100 evpn
```

```
[PE4] ip vpn-instance vpna6
[PE4-vpn-instance-vpna6] ipv6-family
[PE4-vpn-instance-vpna6-af-ipv6] default-color 100 evpn
```

# Check the VPN instance IP routing tables on PE1 and PE4.

```
[PE1]display ip routing-table vpn-instance  vpna
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
         Destinations : 4        Routes : 4

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        10.1.1.1/32  Direct  0    0           D   127.0.0.1       LoopBack1
        10.1.4.4/32  IBGP    255  0           RD  2001::4:4       p1
      127.0.0.0/8    Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32   Direct  0    0           D   127.0.0.1       InLoopBack0
```

The route has recursed to a logical interface (based on the tunnel policy).

```
[PE4]display ip routing-table vpn-instance vpna
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
         Destinations : 4        Routes : 4

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        10.1.1.1/32  IBGP    255  0           RD  2001::1:1       p1
        10.1.4.4/32  Direct  0    0           D   127.0.0.1       LoopBack1
      127.0.0.0/8    Direct  0    0           D   127.0.0.1       InLoopBack0
255.255.255.255/32   Direct  0    0           D   127.0.0.1       InLoopBack0
```

# Check detailed information about the VPN instance IPv6 routes on PE1 and PE4.

```
[PE1]display ipv6 routing-table vpn-instance vpna6 2002::4:4 verbose
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna6
Summary Count : 1

Destination     : 2002::4:4                    PrefixLength   : 128
NextHop         : 2001::4:4                    Preference     : 255
Neighbour       : 2001::5:5                    ProcessID      : 0
Label           : NULL                         Protocol       : IBGP
State           : Active Adv Relied            Cost           : 0
Entry ID        : 0                            EntryFlags     : 0x00000000
Reference Cnt   : 0                            Tag            : 0
Priority        : low                          Age            : 105sec
IndirectID      : 0x10000C5                    Instance       :
RelayNextHop    : ::                           TunnelID       : 0x000000003400000001
Interface       : p1                           Flags          : RD
```

```
[PE4]display ipv6 routing-table vpn-instance vpna6 2002::1:1 verbose
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
```

```
------------------------------------------------------------------------------
Routing Table : vpna6
Summary Count : 1

Destination      : 2002::1:1                    PrefixLength   : 128
NextHop          : 2001::1:1                    Preference     : 255
Neighbour        : 2001::5:5                    ProcessID      : 0
Label            : NULL                         Protocol       : IBGP
State            : Active Adv Relied            Cost           : 0
Entry ID         : 0                            EntryFlags     : 0x00000000
Reference Cnt    : 0                            Tag            : 0
Priority         : low                          Age            : 188sec
IndirectID       : 0x10000C4                    Instance       :
RelayNextHop : ::                               TunnelID       : 0x000000003400000001
Interface        : p1                           Flags          : RD
```

The IPv6 route has recursed to a logical interface (based on the tunnel policy).

# Check detailed information about the route 10.1.1.1 on PE4.

```
[PE4]display ip routing-table vpn-instance vpna  10.1.1.1  verbose
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
------------------------------------------------------------------------------
Routing Table : vpna
Summary Count : 1

Destination: 10.1.1.1/32
      Protocol   :  IBGP                 Process ID     : 0
    Preference   : 255                      Cost       : 0
       NextHop : 2001::1:1              Neighbour      : 2001::5:5
         State  : Active Adv Relied          Age        : 00h07m40s
           Tag  : 0                        Priority     : low
         Label  : NULL                     QoSInfo    : 0x0
    IndirectID  : 0x100010C                Instance   :
 RelayNextHop : ::                       Interface : p1
      TunnelID  : 0x000000003400000001     Flags : RD
```

The route has recursed to a tunnel with the tunnel ID of 0x000000003400000001.

# Check tunnel information on PE4.

```
[PE4]display tunnel-info all | include 0x000000003400000001
Info: It will take a long time if the content you search is too much or the string you input is too long,
you can press CTRL_C to break.
Tunnel ID                Type                 Destination          Status
------------------------------------------------------------------------------
0x000000003400000001  srv6tepolicy          2001::1:1            UP
```

The tunnel is an SRv6 Policy.

Step 6 Perform a path switchover test.

Enable headend-based fault detection and test whether service traffic can be switched to different segment lists.

# Test the L3VPNv6 service connectivity on PE1.

```
[PE1]ping ipv6 vpn-instance vpna6 -a 2002::1:1 2002::4:4
  PING 2002::4:4 : 56  data bytes, press CTRL_C to break
    Reply from 2002::4:4
    bytes=56 Sequence=1 hop limit=64 time=11 ms
    Reply from 2002::4:4
    bytes=56 Sequence=2 hop limit=64 time=3 ms
    Reply from 2002::4:4
    bytes=56 Sequence=3 hop limit=64 time=3 ms
    Reply from 2002::4:4
    bytes=56 Sequence=4 hop limit=64 time=3 ms
    Reply from 2002::4:4
    bytes=56 Sequence=5 hop limit=64 time=2 ms

  --- 2002::4:4 ping statistics---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=2/4/11 ms
```

IPv6 traffic is forwarded properly.

# Test the L3VPNv4 service connectivity on PE1.

```
[PE1]ping -vpn-instance vpna -a 10.1.1.1 10.1.4.4
  PING 10.1.4.4: 56  data bytes, press CTRL_C to break
    Reply from 10.1.4.4: bytes=56 Sequence=1 ttl=255 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=2 ttl=255 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=3 ttl=255 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=4 ttl=255 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=5 ttl=255 time=1 ms

  --- 10.1.4.4 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 1/1/1 ms
```

Traffic is forwarded properly.

# Configure End.OP opcodes on the two headends to test the corresponding forwarding paths.

```
[PE1]segment-routing ipv6
[PE1-segment-routing-ipv6] locator SRv6
[PE1-segment-routing-ipv6-locator] opcode ::2 end-op
```

```
[PE4]segment-routing ipv6
[PE4-segment-routing-ipv6] locator SRv6
[PE4-segment-routing-ipv6-locator] opcode ::2 end-op
```

# Run the **tracert** command on PE1 to test the SRv6 Policy connectivity.

```
[PE1] tracert  srv6-tepolicy  policy-name  p1end-op2001:4::2
```

```
 Trace Routesrv6-te policy:100   data bytes,pressCTRL_C tobreak
 srv6-te policy'ssegmentlist:
 Preference: 100;Path Type: primary; Protocol-Origin:local; Originator:0, 0.0.0.0; Discriminator:100;
Segment-ListID:1; Xcindex: 1; end-op: 2001:4::2
 TTL   Replier            Time    Type      SRH
 0                                Ingress   [SRH:2001:5::1,2001:3::1,2001:4::1,2001:4::2,SL=3]
 1     2001::6            8ms     Transit   [SRH:2001:5::1,2001:3::1,2001:4::1,2001:4::2,SL=3]
 2     2001::12           3ms     Transit   [SRH:2001:5::1,2001:3::1,2001:4::1,2001:4::2,SL=2]
 3     2001:4::2          3ms     Egress    [SRH:2001:5::1,2001:3::1,2001:4::1,2001:4::2,SL=1]
 srv6-te policy'ssegmentlist:
 Preference: 100;Path Type: primary; Protocol-Origin:local; Originator:0, 0.0.0.0; Discriminator:100;
Segment-ListID:4; Xcindex: 3; end-op: 2001:4::2
 TTL   Replier            Time    Type      SRH
 0                                Ingress   [SRH:2001:5::1,2001:6::1,2001:4::1,2001:4::2,SL=3]
 1     2001::6            6ms     Transit   [SRH:2001:5::1,2001:6::1,2001:4::1,2001:4::2,SL=3]
 2     2001::E            2ms     Transit   [SRH:2001:5::1,2001:6::1,2001:4::1,2001:4::2,SL=2]
 3     2001:4::2          3ms     Egress    [SRH:2001:5::1,2001:6::1,2001:4::1,2001:4::2,SL=1]
```

Traffic reaches PE4 along segment lists 1 and 2.

# Enable headend-based fault detection.

```
[PE1]segment-routing ipv6
[PE1-segment-routing-ipv6] srv6-te-policy path verification enable
[PE1-segment-routing-ipv6] srv6-te policy p1
[PE1-segment-routing-ipv6-policy-p1] path verification enable
```

```
[PE4]segment-routing ipv6
[PE4-segment-routing-ipv6] srv6-te-policy path verification enable
[PE4-segment-routing-ipv6] srv6-te policy p1
[PE4-segment-routing-ipv6-policy-p1] path verification enable
```

# Shut down all interconnection interfaces on PE3 to simulate a node fault.

```
[PE3]interface GigabitEthernet0/5/0
[PE3-GigabitEthernet0/5/0] shutdown
[PE3-GigabitEthernet0/5/0] quit
[PE3]interface GigabitEthernet0/5/1
[PE3-GigabitEthernet0/5/1] shutdown
[PE3-GigabitEthernet0/5/1] quit
```

# Test the service connectivity on PE1.

```
[PE1]ping -vpn-instance vpna -a 10.1.1.1 10.1.4.4
  PING 10.1.4.4: 56   data bytes, press CTRL_C to break
    Reply from 10.1.4.4: bytes=56 Sequence=1 ttl=255 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=2 ttl=255 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=3 ttl=255 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=4 ttl=255 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=5 ttl=255 time=1 ms

  --- 10.1.4.4 ping statistics ---
```

```
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 1/1/1 ms
```

Service traffic is forwarded properly.

```
[PE1]ping ipv6 vpn-instance vpna6 -a 2002::1:1 2002::4:4
  PING 2002::4:4 : 56  data bytes, press CTRL_C to break
    Reply from 2002::4:4
    bytes=56 Sequence=1 hop limit=64 time=8 ms
    Reply from 2002::4:4
    bytes=56 Sequence=2 hop limit=64 time=2 ms
    Reply from 2002::4:4
    bytes=56 Sequence=3 hop limit=64 time=3 ms
    Reply from 2002::4:4
    bytes=56 Sequence=4 hop limit=64 time=2 ms
    Reply from 2002::4:4
    bytes=56 Sequence=5 hop limit=64 time=2 ms

  --- 2002::4:4 ping statistics---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=2/3/8 ms
```

IPv6 service traffic is forwarded properly.

# Run the **tracert** command on PE1 to test the SRv6 Policy connectivity again.

```
[PE1]tracert srv6-te policy policy-name p1 end-op 2001:4::2 -m 4
Trace Routesrv6-te policy:100  data bytes,pressCTRL_C tobreak
 srv6-te policy'ssegmentlist:
 Preference: 100;Path Type: primary; Protocol-Origin:local; Originator:0, 0.0.0.0; Discriminator:100;
Segment-ListID:1; Xcindex: 0; end-op: 2001:4::2
 TTL   Replier          Time   Type      SRH
 0                             Ingress   [SRH:2001:5::1,2001:3::1,2001:4::1,2001:4::2,SL=3]
 1     ::               *      Transit
 2     ::               *      Transit
 3     ::               *      Transit
 4     ::               *      Transit
 srv6-te policy'ssegmentlist:
 Preference: 100;Path Type: primary; Protocol-Origin:local; Originator:0, 0.0.0.0; Discriminator:100;
Segment-ListID:4; Xcindex: 3; end-op: 2001:4::2
 TTL   Replier          Time   Type      SRH
 0                             Ingress   [SRH:2001:5::1,2001:6::1,2001:4::1,2001:4::2,SL=3]
 1     2001::6          8ms    Transit   [SRH:2001:5::1,2001:6::1,2001:4::1,2001:4::2,SL=3]
 2     2001::E          3ms    Transit   [SRH:2001:5::1,2001:6::1,2001:4::1,2001:4::2,SL=2]
 3     2001:4::2        2ms    Egress    [SRH:2001:5::1,2001:6::1,2001:4::1,2001:4::2,SL=1]
```

Use the **-m** parameter to specify the maximum hop limit supported by tracert.

The command output shows that segment list 1 cannot work properly.

# Shut down all interfaces on P1 to simulate a node fault.

```
[P1]interface GigabitEthernet0/5/0
[P1-GigabitEthernet0/5/0] shutdown
[P1-GigabitEthernet0/5/0] quit
[P1]interface GigabitEthernet0/5/1
[P1-GigabitEthernet0/5/1] shutdown
[P1-GigabitEthernet0/5/1] quit
[P1]interface GigabitEthernet0/5/2
[P1-GigabitEthernet0/5/2] shutdown
[P1-GigabitEthernet0/5/2] quit
```

# Test the service connectivity on PE1.

[PE1]ping -vpn-instance vpna -a 10.1.1.1 10.1.4.4

```
  PING 10.1.4.4: 56  data bytes, press CTRL_C to break
    Reply from 10.1.4.4: bytes=56 Sequence=1 ttl=255 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=2 ttl=255 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=3 ttl=255 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=4 ttl=255 time=1 ms
    Reply from 10.1.4.4: bytes=56 Sequence=5 ttl=255 time=1 ms

  --- 10.1.4.4 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 1/1/1 ms
```

Service traffic is still forwarded properly.

```
[PE1]ping ipv6 vpn-instance vpna6 -a 2002::1:1 2002::4:4
  PING 2002::4:4 : 56  data bytes, press CTRL_C to break
    Reply from 2002::4:4
    bytes=56 Sequence=1 hop limit=64 time=8 ms
    Reply from 2002::4:4
    bytes=56 Sequence=2 hop limit=64 time=2 ms
    Reply from 2002::4:4
    bytes=56 Sequence=3 hop limit=64 time=3 ms
    Reply from 2002::4:4
    bytes=56 Sequence=4 hop limit=64 time=2 ms
    Reply from 2002::4:4
    bytes=56 Sequence=5 hop limit=64 time=2 ms

  --- 2002::4:4 ping statistics---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max=2/3/8 ms
```

IPv6 service traffic is forwarded properly.

# Run the **tracert** command on PE1 to test the SRv6 Policy connectivity.

```
[PE1]tracert srv6-te policy policy-name p1 end-op 2001:4::2 -m 4
Trace Routesrv6-te policy:100  data bytes,pressCTRL_C tobreak
 srv6-te policy'ssegmentlist:
```

```
 Preference: 50; Path Type: primary; Protocol-Origin: local; Originator: 0, 0.0.0.0; Discriminator: 50;
Segment-ListID: 3; Xcindex: 65; end-op: 2001:4::2
 TTL   Replier          Time    Type      SRH
 0                              Ingress   [SRH: 2001:2::1,2001:6::1,2001:4::1,2001:4::2,SL=3]
 1     2001::2          10 ms   Transit   [SRH: 2001:2::1,2001:6::1,2001:4::1,2001:4::2,SL=3]
 2     2001::A          2 ms    Transit   [SRH: 2001:2::1,2001:6::1,2001:4::1,2001:4::2,SL=2]
 3     2001:4::2        2 ms    Egress    [SRH: 2001:2::1,2001:6::1,2001:4::1,2001:4::2,SL=1]
```

The command output shows that the traffic has been switched to the segment list in the second candidate path for forwarding.

**----End**

## 9.2.3 Quiz

In addition to headend-based fault detection, what methods can be used to check whether a segment list works properly?

# 10 Open Network Programmability

## 10.1 SSH Lab

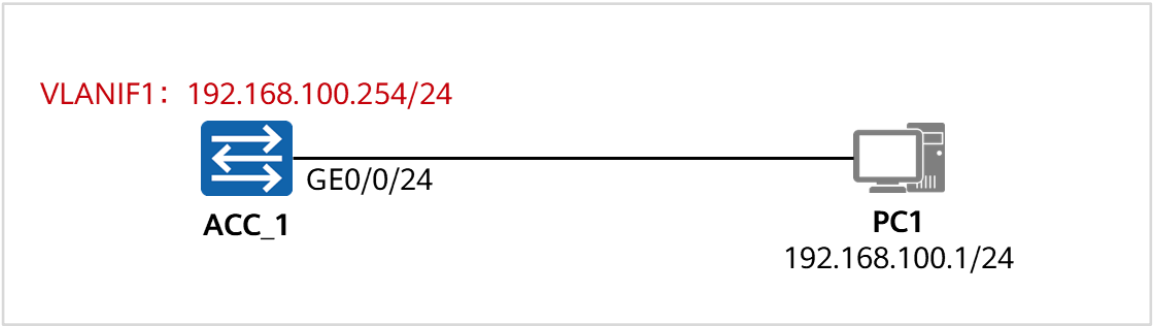
### 10.1.1 About This Lab

#### 10.1.1.1 Objectives

- Log in to the device using Paramiko module by SSH.
- Log in to the device using Paramiko, check the configurations, and transfer files.

#### 10.1.1.2 Networking Description

**Figure 10-1 SSH lab topology**




ACC_1 and PC1 are used in this lab. The IP address of VLANIF1 on ACC_1 is 192.168.100.254/24, and the IP address of PC1 is 192.168.100.1/24. Install the Anaconda compilation environment. Write a script to obtain some output of the device and upload and download files.

For details about how to install the Anaconda lab environment, see *HCIP-Datacom-Python Programming Basics Lab Guide*.

### 10.1.2 Lab Task

#### 10.1.2.1 Configuration Roadmap

1. Perform basic configuration on ACC_1.
2. Configure SSH for ACC_1, create an SSH user, generate the public and private keys of PC1, upload the keys to ACC_1, and configure SSH for public key login.
3. Compile and execute the SSH login script.
4. Parse the script.
5. Configure SFTP for ACC_1.

6. Compile the SFTP code.
7. Execute and interpret the script.

## 10.1.2.2 Configuration Procedure

Step 1 Perform basic configuration on ACC_1.

Configure interconnection IP addresses for ACC_1.

# Name the devices.

N/A

# Configure an IP address for VLANIF 1.

```
[ACC_1]interface Vlanif1
[ACC_1-Vlanif1] ip address 192.168.100.254 255.255.255.0
[ACC_1-Vlanif1] quit
```

# Check the connectivity between ACC_1 and PC1.

```
[ACC_1]ping 192.168.100.1
  PING 192.168.100.1: 56  data bytes, press CTRL_C to break
    Reply from 192.168.100.1: bytes=56 Sequence=1 ttl=128 time=1 ms
    Reply from 192.168.100.1: bytes=56 Sequence=2 ttl=128 time=1 ms
    Reply from 192.168.100.1: bytes=56 Sequence=3 ttl=128 time=1 ms
    Reply from 192.168.100.1: bytes=56 Sequence=4 ttl=128 time=1 ms
    Reply from 192.168.100.1: bytes=56 Sequence=5 ttl=128 time=1 ms

  --- 192.168.100.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

ACC_1 can communicate with PC1.

Step 2 Configure SSH.

Configure SSH on ACC_1, create SSH user **python**, use software on PC1 to generate public and private keys, upload the public key to ACC_1, and configure the **python** user to use keys for authentication.

# Enable STelnet on the server and configure the VTY.

```
[ACC_1] stelnet server enable
[ACC_1] ssh server-source all-interface
[ACC_1] user-interface vty 0 4
[ACC_1-ui-vty0-4] authentication-mode aaa
[ACC_1-ui-vty0-4] protocol inbound ssh
[ACC_1-ui-vty0-4] user privilege level 15
[ACC_1-ui-vty0-4] quit
```

# Create a local user **python** and set the user service type to SSH.

```
[ACC_1] aaa
```

```
[ACC_1-aaa] local-user python password irreversible-cipher Huawei@123
[ACC_1-aaa] local-user python privilege level 15
[ACC_1-aaa] local-user python service-type ssh
[ACC_1-aaa] quit
```

# Create an SSH user, set the authentication mode to RSA key, and set the service mode to STelnet.

```
[ACC_1] ssh user python
Info: Succeeded in adding a new SSH user.
[ACC_1] ssh user python authentication-type rsa
[ACC_1] ssh user python service-type stelnet
```

# Generate a key pair using OpenSSH.

```
C:\Windows\System32\OpenSSH> .\ssh-keygen.exe -m pem -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\example_user/.ssh/id_rsa):
Created directory 'C:\Users\example_user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\example_user/.ssh/id_rsa.
Your public key has been saved in C:\Users\example_user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:y4abapa2DtcLTr/Z8esxAydcOqcZeyNaafmfu7ULTQc china\example_user@example_userA
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|                 |
|            .  E |
|         . o  .  |
|          S o  . |
|       . o #  o . |
|  . +.o # * . o  |
|    ==o @ * = + .|
|    ==oO...=o=oo.|
+----[SHA256]-----+
```

Generate an RSA key pair using OpenSSH, with the key length being 2048 bits.

The methods for downloading and using OpenSSH are not provided here. OpenSSH has been integrated in mainstream Windows and Linux versions and can be directly used. For example, OpenSSH built in Windows 10 is used in this lab.

Alternatively, you can use the key pair provided in this lab. Note that the key pair can be used only in labs and cannot be used in actual environments.

Private key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAw6y+0DNNXIq62YBDlLELXrGxpuZ0npbPpdDp2keqASmCjDfE
rxKLGvHSoBEAj5mBOe8SPsirmuNJIfGtQikXKBTUzjCD0b0ZDWq0oYZNUaC45+H6
3gGu0C/VH8KKbasM/7mPmqjSYqPN1E4IYgI9I4D6R3HogkAra8EZo1TM4NGjHNvj
e48f8jV+1iKJc3eZ7mW+HB33XyrF+YAoG/mny4ro8mbLdHg3kFUCR9rOHAE5lJKG
YQzHoAs+ygSWHxcXU0xRxOw0yDRkE2ZzlYHldSaj4vMA/3LfQbzezBIgpqIkpMv/
```

```
tVZZxTGBhlXcaAaHhdfWk8c1/Lu9T0HijphGIQIDAQABAoIBABwXs/luNCAiJrtN
R+aUGH7K4ieFFJ8kJcG3oatd+JFcJUr+QrWj4ubGayfh6QO01TG9GhB7fO+qy/gc
m3RRM+rkOr0zh+lXzb34YOCDyJv8iC96aSrwcOmgxGdf5cHi+eXI/U4GCNZyj/14
CooPqjlrrgMN3oS8s78vpP+nEBcCoYvjdwaMjB83sLork4uMMFc1INf8CxIaY4Za
z5j1UdjH/YpJlkvCNikiog4QdqnB4fvx+Ef4PSs6boc5CRcqtUWhD/hrK0bleVDN
kCJsPYz6dntxuF/epprUR+v0ayfROZIr78Lcd93vgKWbS196vCjsfzA7S129t7e2
OLcAEDECgYEA8EVFwv9gn3S9QnHVhTHHwIAI6LA6kfRfZXf4GoeVavPgdwm7fmsR
PEQYmGAK2lSB93RVyfMZUeat7pqHV3ZiXj8ar4bk0SPBWiNcUweZXPu2d0buKDRu
xxX1dLbDPHHyoIJBe05NpaHK060JapekKCPVu77Avdyiuh4mXtAvZX0CgYEA0HwV
7NgMJa0dsHAL43C8nOjlCRYaw07W3m56Cs+LQlzoY4kVgUs7V5rmYHj65NYJ3OPB
UH89u+F63iOVgXnw3i5wxGcHCzbxqLdJeBU0FyWqFmkGkN3C+QgxqxqavoskaxfA
vpsaljLRZnO7lvx8SM86/mFE8yG6V+fkdhRPtHUCgYEAo9WYjC63JAPmSbh1cC1R
l+G2MXR5e6y2n+n+M5jgOd2x4Xmxb7JyPjXwz3Yt8FJPjPy2ws4RJde2Ik0cnHcM
/FbrS2UgAbvvzBbwjW1RFiZNdWfYXXHjV72RcdRtrHnGh8xJ+lqad6arN0t2cEiW
lwhYZXzc0JnyHBcS8vPiXcECgYEAiAILQ6GF+yY+khXmi4/GXA7LK+xRtUw/fIhR
8a+Hx9lYeWa6sshjhDk3RYRnGNZRRNsIB/2aRnFIeJZ6OCdW7XsaceZMgBJuAITh
1wbuHpOhrFEDyYoCYffofxkyDlAzh/HM+guxgn7QgKfLnypD4jPe5oiKNJqyOBq9
vhJuFOkCgYEAzN1iwCJs5p9+Y+/12rjv3NyCgM6DL1+Ldz6AnQ3r8ioJ1q4erBXE
JwapPq9f+zAglyo0DGeY2bGAnjnOrCcvoED7eU8qFmuq+EA/0EhiBsXJwwotp9BI
JTOQQJUmgU62mtpp+l1lyJ9KtnZPEtSpCgujSrAqHk5rjrP1TskgiMc=
-----END RSA PRIVATE KEY-----
```

Public key:

```
-----BEGIN RSA PUBLIC KEY-----
MIIBCgKCAQEAw6y+0DNNXIq62YBDlLELXrGxpuZ0npbPpdDp2keqASmCjDfErxKL
GvHSoBEAj5mBOe8SPsirmuNJIfGtQikXKBTUzjCD0b0ZDWq0oYZNUaC45+H63gGu
0C/VH8KKbasM/7mPmqjSYqPN1E4IYgI9I4D6R3HogkAra8EZo1TM4NGjHNvje48f
8jV+1iKJc3eZ7mW+HB33XyrF+YAoG/mny4ro8mbLdHg3kFUCR9rOHAE5lJKGYQzH
oAs+ygSWHxcXU0xRxOw0yDRkE2ZzlYHldSaj4vMA/3LfQbzezBIgpqIkpMv/tVZZ
xTGBhlXcaAaHhdfWk8c1/Lu9T0HijphGIQIDAQAB
-----END RSA PUBLIC KEY-----
```

Copy the contents of the public key and private key to the TXT file, and name the public key **id_rsa.pub** and the private key **id_rsa.pem** to obtain the public key and private key.

Note that you need to change the filename extension of the .txt file.

S5731-H24T4XC supports only hexadecimal public keys. Therefore, you need to convert the base64-encoded public key file using the third-party software.

You can convert the base64 code to the hexadecimal format using a conversion tool and typeset the conversion result in the following format.

The following is an example of the converted public key:

```
[root@localhost usr]# ssh-keygen -e -m pem -f /usr/id_rsa.pub | egrep -v "BEGIN|END" | base64 -d |
od -t x1 -An -w4 | tr 'a-f' 'A-F' | tr -d ' ' | fmt -w 48
3082010A 02820101 00C3ACBE D0334D5C 8ABAD980
4394B10B 5EB1B1A6 E6749E96 CFA5D0E9 DA47AA01
29828C37 C4AF128B 1AF1D2A0 11008F99 8139EF12
3EC8AB9A E34921F1 AD422917 2814D4CE 3083D1BD
190D6AB4 A1864D51 A0B8E7E1 FADE01AE D02FD51F
C28A6DAB 0CFFB98F 9AA8D262 A3CDD44E 0862023D
2380FA47 71E88240 2B6BC119 A354CCE0 D1A31CDB
E37B8F1F F2357ED6 22897377 99EE65BE 1C1DF75F
```

```
2AC5F980 281BF9A7 CB8AE8F2 66CB7478 37905502
47DACE1C 01399492 86610CC7 A00B3ECA 04961F17
17534C51 C4EC34C8 34641366 739581E5 7526A3E2
F300FF72 DF41BCDE CC1220A6 A224A4CB FFB55659
C5318186 55DC6806 8785D7D6 93C735FC BBBD4F41
E28E9846 21020301 0001
```

Convert the public key. If the Linux environment is available, convert the original **id_rsa.pub** file based on the format of the preceding information. Note that the public key is not the one provided above because the public key provided above has been converted using the **ssh-keygen -e -m pem -f** command.

# Add a public key to the SSH server and assign the public key to the user.

```
[ACC_1]rsa peer-public-key ssh
Enter "RSA public key" view, return system view with "peer-public-key end".
[ACC_1-rsa-public-key]public-key-code begin
Enter "RSA key code" view, return last view with "public-key-code end".
[ACC_1-rsa-key-code]3082010A 02820101 00C3ACBE D0334D5C 8ABAD980
[ACC_1-rsa-key-code]4394B10B 5EB1B1A6 E6749E96 CFA5D0E9 DA47AA01
[ACC_1-rsa-key-code]29828C37 C4AF128B 1AF1D2A0 11008F99 8139EF12
[ACC_1-rsa-key-code]3EC8AB9A E34921F1 AD422917 2814D4CE 3083D1BD
[ACC_1-rsa-key-code]190D6AB4 A1864D51 A0B8E7E1 FADE01AE D02FD51F
[ACC_1-rsa-key-code]C28A6DAB 0CFFB98F 9AA8D262 A3CDD44E 0862023D
[ACC_1-rsa-key-code]2380FA47 71E88240 2B6BC119 A354CCE0 D1A31CDB
[ACC_1-rsa-key-code]E37B8F1F F2357ED6 22897377 99EE65BE 1C1DF75F
[ACC_1-rsa-key-code]2AC5F980 281BF9A7 CB8AE8F2 66CB7478 37905502
[ACC_1-rsa-key-code]47DACE1C 01399492 86610CC7 A00B3ECA 04961F17
[ACC_1-rsa-key-code]17534C51 C4EC34C8 34641366 739581E5 7526A3E2
[ACC_1-rsa-key-code]F300FF72 DF41BCDE CC1220A6 A224A4CB FFB55659
[ACC_1-rsa-key-code]C5318186 55DC6806 8785D7D6 93C735FC BBBD4F41
[ACC_1-rsa-key-code]E28E9846 21020301 0001
[ACC_1-rsa-key-code]public-key-code end
[ACC_1-rsa-public-key]peer-public-key end
[ACC_1]ssh user python assign rsa-key ssh
```

Step 3 Log in to the device using Paramiko.

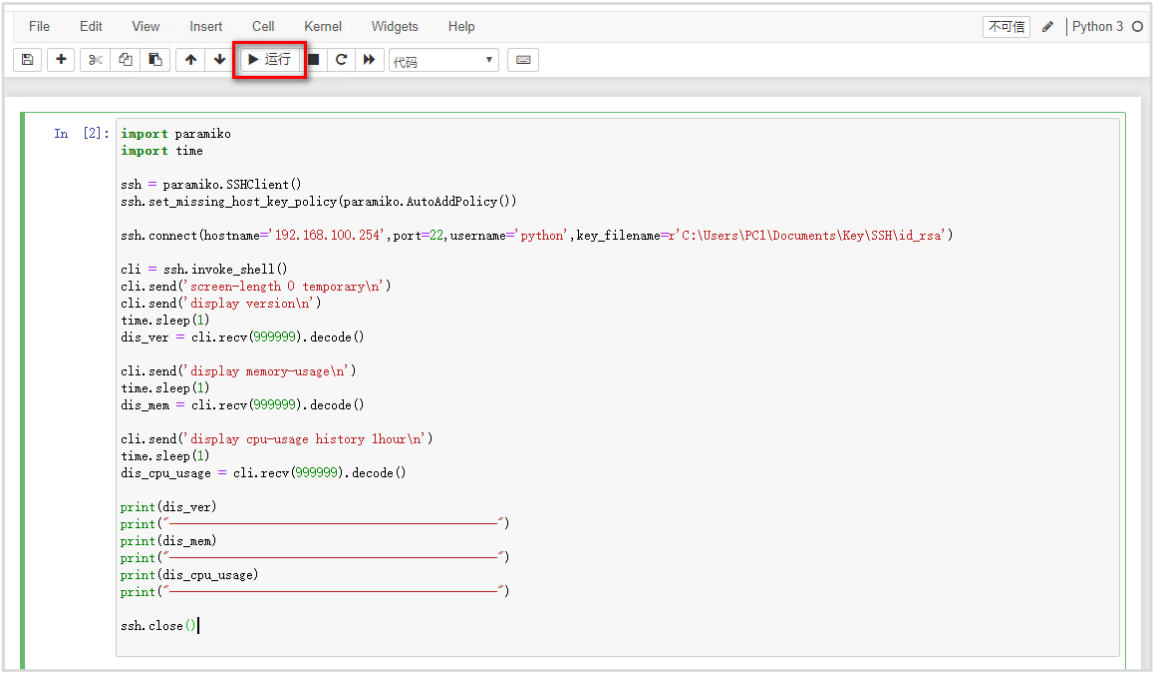
Log in to ACC_1 using Paramiko based on the Python script, run the **display version**, **display memory-usage**, and **display cpu-usage history 1hour** commands, and check the command output.

# Configure the full code.

```
import paramiko
import time

ssh = paramiko.SSHClient()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())

ssh.connect(hostname='192.168.100.254',port=22,username='python',key_filename=r'C:\Users\PC1\Documents\Key\SSH\id_rsa')

cli = ssh.invoke_shell()
cli.send('screen-length 0 temporary\n')
```

```
cli.send('display version\n')
time.sleep(1)
dis_ver = cli.recv(999999).decode()

cli.send('display memory-usage\n')
time.sleep(1)
dis_mem = cli.recv(999999).decode()

cli.send('display cpu-usage history 1hour\n')
time.sleep(1)
dis_cpu_usage = cli.recv(999999).decode()

print(dis_ver)
print("-----------------------------------------------")
print(dis_mem)
print("-----------------------------------------------")
print(dis_cpu_usage)
print("-----------------------------------------------")

ssh.close()
```

# Execute the code in a compiler.




# Output the result.

```
Info: The max number of VTY users is 10, and the number
        of current VTY users on line is 1.
        The current login time is 2021-09-01 10:17:02+08:00.
Info: Smart-upgrade is currently disabled. Enable Smart-upgrade to get recommended version
information.
<ACC_1>screen-length 0 temporary
Info: The configuration takes effect on the current user terminal interface only.
<ACC_1>display version
Huawei Versatile Routing Platform Software
```

```
VRP (R) software, Version 5.170 (S5731 V200R020C10SPC500)
Copyright (C) 2000-2020 HUAWEI TECH Co., Ltd.
HUAWEI S5731-H24T4XC Routing Switch uptime is 0 week, 1 day, 0 hour, 38 minutes

ES5D2T28C007 0(Master)   : uptime is 0 week, 1 day, 0 hour, 37 minutes
DDR                Memory Size : 4096   M bytes
FLASH Total        Memory Size : 1024   M bytes
FLASH Available Memory Size : 739    M bytes
Pcb             Version    : VER.B
BootROM          Version    : 0000.0511
BootLoad         Version    : 0214.0000
CPLD             Version    : 0106
Software         Version    : VRP (R) Software, Version 5.170 (V200R020C10SPC500)
FLASH            Version    : 0000.0000
PWR1 information
Pcb             Version    : PWR VER.D
FAN1 information
Pcb             Version    : NA
FAN2 information
Pcb             Version    : NA
<ACC_1>
-----------------------------------------------
display memory-usage
  Memory utilization statistics at 2021-09-01 10:17:03+08:00
  System Total Memory Is: 2134900736 bytes
  Total Memory Used Is: 339317372 bytes
  Memory Using Percentage Is: 15%
<ACC_1>
-----------------------------------------------
display cpu-usage history 1hour

100%|
 95%|
 90%|
 85%|
 80%|
 75%|
 70%|
 65%|
 60%|
 55%|
 50%|
 45%|
 40%|
 35%|
 30%|
 25%|
 20%|
 15%|
 10%|
  5%|
     ---------+---------+---------+---------+---------+---------+>
              10         20         30         40         50         60
              System cpu-usage last 60 minutes(Per Min)
```

```
<ACC_1>
```

Step 4 Parse the script.

The code is parsed as follows:

# Import the Paramiko module.

```
import paramiko
import time
```

Import the paramiko and time modules. If a module is not installed, you can install it by running the **pip install paramiko** command.

This section describes common classes and methods used by Paramiko as a client, for example, the SSHClient class and its AutoAddPolicy, connect, invoke_shell, and close methods. For more Paramiko methods, visit **http://docs.paramiko.org/**.

By default, Python executes all code in sequence without intervals. When you use Paramiko to send configuration commands to a switch, SSH may not respond in a timely manner or the command output may be incomplete. In this case, you can use the sleep method in the time module to manually pause the program.

# Instantiate an SSH object.

```
ssh = paramiko.SSHClient()
```

Instantiate the SSH object using **Paramiko SSHClient()** and assign a value to SSH.

# Allow unknown hosts to be connected.

```
ssh.set_missing_host_key_policy(paramiko.client.AutoAddPolicy())
```

# Establish an SSH session.

```
ssh.connect(hostname='192.168.100.254',port=22,username='python',key_filename=r'C:\Users\PC1\Documents\Key\SSH\id_rsa')
```

Set up an SSH session. The destination SSH server is 192.168.100.254, the user name is **python**, and **key_filename** specifies the local private key file (id_rsa) of the client. The user is authenticated using a key.

# Open an interactive session.

```
cli = ssh.invoke_shell()
```

Set **cli** to **invoke_shell()**. **invoke_shell()** is used to open an interactive shell session. The session is a logical channel and is established over the SSH session connection.

# Run the following commands:

```
cli.send('screen-length 0 temporary\n')
cli.send('display version\n')
```

Run the **Screen-length 0 temporary** command to cancel the screen splitting, and then check the device version number.

# Set the sleep time.

```
time.sleep(1)
```

Configure the sleep time to 1 second and wait for the **display version** command output.

# Obtain the command output of the channel.

```
dis_ver = cli.recv(999999).decode()
```

**invoke_shell()** has created a channel. All previous input and output process information is stored in this channel. You can get all the information in this channel and display it to the Python compiler.

Invoke **cli.recv()**, decode it using **decode()**, and assign a value to **dis_ver**.

**recv(999999)** is used to receive data from a channel. The maximum data size is 999999 bytes.

The **decode()** method is used to decode the bytes object in the specified encoding format. The default encoding format is utf-8.

During decoding, the result is presented to the interface in a new line, facilitating your reading.

# Display the memory and CPU usage of the device.

```
cli.send('display memory-usage\n')
time.sleep(1)
dis_mem = cli.recv(999999).decode()

cli.send('display cpu-usage history 1hour\n')
time.sleep(1)
dis_cpu_usage = cli.recv(999999).decode()
```

The implementation is the same as that of the preceding code.

# Print the command execution result.

```
print(dis_ver)
print("----------------------------------------------")
print(dis_mem)
print("----------------------------------------------")
print(dis_cpu_usage)
print("----------------------------------------------")
```

Print the previously stored output using **print**.

# Close the SSH session.

```
ssh.close()
```

The session is closed by invoking **close()**. The number of VTY connections on the device is limited. Therefore, you need to close the SSH session after running the script.

Step 5 Configure SFTP.

Configure SFTP for ACC_1.

# Enable the SFTP server function

```
[ACC_1]sftp server enable
```

# Add an SFTP user type and configure the server directory.

```
[ACC_1]ssh user python service-type all
[ACC_1]ssh user python sftp-directory flash:/
```

Step 6 Log in to the device using Paramiko and perform file operations.

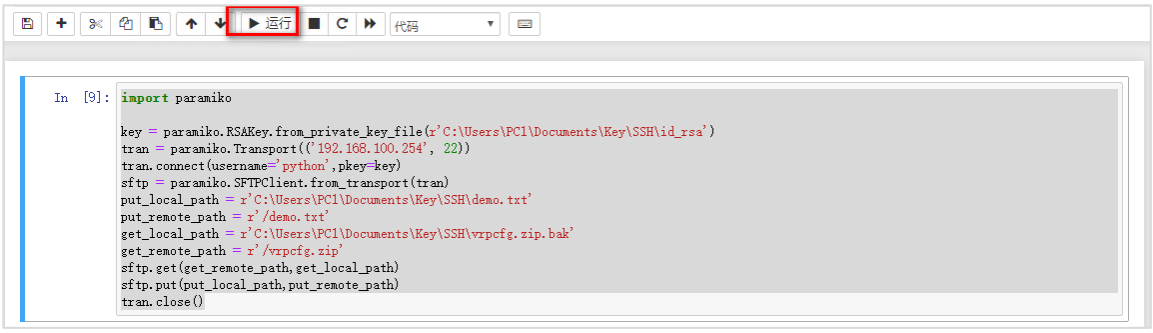
The Python script invokes the paramiko module to log in to ACC_1, downloads the device file **vrpcfg.cfg** configuration file, and uploads the **test.cfg** configuration file.

# Complete script.

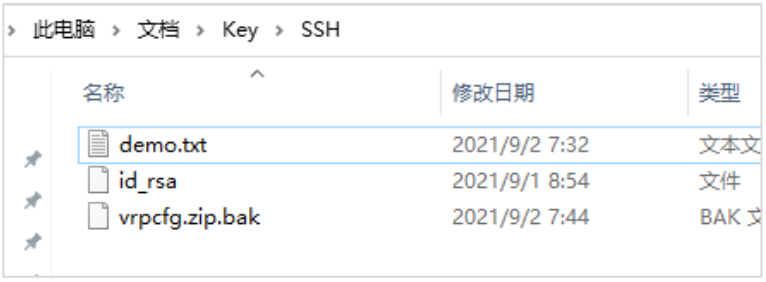
```
import paramiko

key = paramiko.RSAKey.from_private_key_file(r'C:\Users\PC1\Documents\Key\SSH\id_rsa')
tran = paramiko.Transport(('192.168.100.254', 22))
tran.connect(username='python',pkey=key)
sftp = paramiko.SFTPClient.from_transport(tran)
put_local_path = r'C:\Users\PC1\Documents\Key\SSH\demo.txt'
put_remote_path = r'/demo.txt'
get_local_path = r'C:\Users\PC1\Documents\Key\SSH\vrpcfg.zip.bak'
get_remote_path = r'/vrpcfg.zip'
sftp.get(get_remote_path,get_local_path)
sftp.put(put_local_path,put_remote_path)
tran.close()
```

Create a file named **demo.txt** in the **C:\Users\PC1\Documents\Key\SSH** directory on your local PC.

# Execute the code in a compiler.




# Verify the configuration.




The **vrpcfg.zip.bak** file exists in the local PC.

```
<ACC_1>dir
Directory of flash:/

  Idx  Attr      Size(Byte)  Date         Time          FileName
    0  -rw-               0  Aug 09 2021 17:57:55   last_startup_software_info.txt
    1  drw-               -  Apr 28 2021 12:32:21   dhcp
    2  drw-               -  Apr 28 2021 12:31:17   user
    3  -rw-          13,432  Apr 28 2021 12:32:23   default_ca.cer
    4  -rw-              36  Aug 09 2021 17:52:14   $_patchstate_reboot
    5  -rw-             224  Aug 09 2021 17:57:55   current_startup_software_info.txt
    6  drw-               -  Sep 01 2021 10:16:19   sessionlog
    7  drw-               -  Apr 28 2021 12:31:34   security
    8  -rw-           7,779  Aug 31 2021 09:40:19   root.cer
    9  -rw-           3,684  Aug 09 2021 17:52:14   $_patch_history
   10  -rw-           3,933  Aug 09 2021 18:05:15   crl.bin
   11  -rw-           1,407  Apr 28 2021 12:32:32   default_local.cer
   12  -rw-     154,627,368  Aug 09 2021 17:51:12   s5731-h-v200r020c10spc500.cc
   13  drw-               -  Aug 19 2021 23:20:11   logfile
   14  -rw-             120  Aug 31 2021 09:36:12   vrpcfg.zip
   15  -rw-     141,558,148  Nov 21 2019 03:53:41   s5731-h-v200r019c00spc500.cc
   16  drw-               -  Apr 28 2021 12:31:11   $_user
   17  -rw-              15  Sep 01 2021 11:53:48   demo.txt
   18  -rw-          65,540  Aug 30 2007 23:12:30   logfile.txt
   19  -rw-         933,116  Aug 30 2007 23:08:21   s5731-h-v200r019sph007.pat
   20  drw-               -  Apr 28 2021 12:31:17   pmdata
   21  -rw-              92  Aug 31 2021 10:09:52   radio_info.txt
   22  drw-               -  Aug 09 2021 17:52:16   $_install_mod
   23  -rw-             836  Aug 31 2021 09:40:15   rr.bak
   24  -rw-             836  Aug 31 2021 09:40:15   rr.dat
   25  -rw-           1,182  Aug 31 2021 09:39:55   private-data.txt
   26  drw-               -  Aug 16 2021 12:04:47   localuser
   27  drw-               -  Aug 31 2021 09:39:34   default-sdb
   28  drw-               -  Apr 28 2021 12:31:41   unimng
   29  drw-               -  Aug 31 2021 09:39:35   update
   30  drw-               -  Aug 09 2021 13:54:38   $_backup
   31  -rw-           3,281  Aug 31 2021 09:40:19   device.pem
   32  drw-               -  Aug 31 2021 09:39:49   sys_apinfo
   33  -rw-               4  Aug 09 2021 17:53:00   snmpnotilog.txt
   34  -rw-             200  Apr 28 2021 12:32:32   ca_config.ini
   35  -rw-       2,403,095  Aug 09 2021 18:05:24   help.web

756,952 KB total (459,084 KB free)
```

The **demo.txt** file exists in **flash:/** of ACC_1.

Step 7 Parse the SFTP script.

The SFTP script is parsed as follows:

# Import the Paramiko module.

```
import paramiko
```

Import the Paramiko module.

SFTP involves the Transport class, Key handling class, and SFTPClient class.

The Transport class is used to instantiate a session channel and establish a session connection.

The Key handling class is used to instantiate a key object.

The SFTPClient class is used to create an SFTP session connection and perform remote file operations.

# Create an RSA key object.

```
key=paramiko.RSAKey.from_private_key_file(r'C:\Users\PC1\Documents\Key\SSH\id_rsa')
```

Read the local RSA private key file on the client and copy it to the key.

# Instantiate the session channel. The destination SSH server is 192.168.100.254, and the port is 22.

```
tran = paramiko.Transport(('192.168.100.254', 22))
```

# Establish an SSH session.

```
tran.connect(username='python', pkey=key)
```

**python** is the user name, and **pkey** is the key object. The user is authenticated using a key.

# Establish an SFTP channel.

```
sftp = paramiko.SFTPClient.from_transport(tran)
```

Create an SFTP channel from an open session connection and assign a value to **sftp**.

# Set the local and remote paths for uploading and downloading.

```
put_local_path = r'C:\Users\PC1\Documents\Key\SSH\demo.txt'
put_remote_path = r'/demo.txt'
get_local_path = r'C:\Users\PC1\Documents\Key\SSH\vrpcfg.zip.bak'
get_remote_path = r'/vrpcfg.zip'
```

Upload the local **demo.txt** file to the device and download the **vrpcfg.zip** file to the local PC and change the file name to **vrpcfg.zip.bak**.

# Transfer the file.

```
sftp.get(get_remote_path,get_local_path)
sftp.put(put_local_path,put_remote_path)
```

# Close the session.

```
tran.close()
```

**----End**

## 10.1.3 Quiz

How can we automatically run a series of commands after logging in to a device without manually running them one by one?

# 10.2 NETCONF Lab

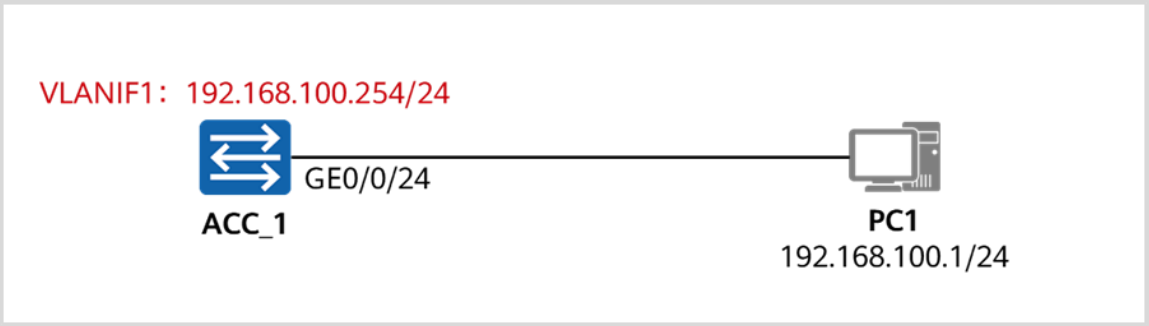
## 10.2.1 About This Lab

### 10.2.1.1 Objectives

- Deliver configurations through ncclient.
- Log in to the device using Paramiko.

### 10.2.1.2 Networking Description

**Figure 10-2 NETCONF Configuration Lab**




In this lab, ACC_1 and PC1 are used. The IP address of VLANIF 1 on ACC_1 is 192.168.100.254/24, and the IP address of PC1 is 192.168.100.1/24. Write a script on PC1 to deliver configurations through NETCONF.

## 10.2.2 Lab Task

### 10.2.2.1 Configuration Roadmap

1. Perform basic configuration on ACC_1.
2. Configure SSH for ACC_1.
3. Write a script. The script enables NETCONF on the device and then delivers configurations through NETCONF.
4. Parse the script.

### 10.2.2.2 Configuration Procedure

Step 1 Perform basic configuration on ACC_1.

Configure interconnection IP addresses for ACC_1.

# Name the devices.

N/A

# Configure an IP address for VLANIF 1.

```
[ACC_1]interface Vlanif1
[ACC_1-Vlanif1] ip address 192.168.100.254 255.255.255.0
[ACC_1-Vlanif1] quit
```

# Check the connectivity between ACC_1 and PC1.

```
[ACC_1]ping 192.168.100.1
  PING 192.168.100.1: 56  data bytes, press CTRL_C to break
    Reply from 192.168.100.1: bytes=56 Sequence=1 ttl=128 time=1 ms
    Reply from 192.168.100.1: bytes=56 Sequence=2 ttl=128 time=1 ms
    Reply from 192.168.100.1: bytes=56 Sequence=3 ttl=128 time=1 ms
    Reply from 192.168.100.1: bytes=56 Sequence=4 ttl=128 time=1 ms
    Reply from 192.168.100.1: bytes=56 Sequence=5 ttl=128 time=1 ms

  --- 192.168.100.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

ACC_1 can communicate with PC1.

Step 2 Configure SSH.

Perform basic SSH configuration on ACC_1.

# Enable STelnet on the server and configure the VTY.

```
[ACC_1] stelnet server enable
[ACC_1] ssh server-source all-interface
[ACC_1] user-interface vty 0 4
[ACC_1-ui-vty0-4] authentication-mode aaa
[ACC_1-ui-vty0-4] protocol inbound ssh
[ACC_1-ui-vty0-4] user privilege level 15
[ACC_1-ui-vty0-4] quit
```

# Create a local user **python** and set the user service type to SSH.

```
[ACC_1] aaa
[ACC_1-aaa] local-user python password irreversible-cipher Huawei@123
[ACC_1-aaa] local-user python privilege level 15
[ACC_1-aaa] local-user python service-type ssh
[ACC_1-aaa] local-aaa-user password policy administrator
[ACC_1-aaa-lupp-admin] undo password alert original
[ACC_1-aaa] quit
```

Disable the function of prompting you to change the password upon the first login.

# Create an SSH user, set the authentication mode to password-based authentication, and set the service mode to STelnet.

```
[ACC_1] ssh user python
Info: Succeeded in adding a new SSH user.
[ACC_1] ssh user python authentication-type password
[ACC_1] ssh user python service-type stelnet
```

Step 3 Write a script.

Use Paramiko to log in to the device, invoke the local configuration script to enable NETCONF on the device, and use NETCONF to set the description of GE0/0/2 to **Config by NETCONF** through ncclient.

# Configure the local script.

```
system-view
 aaa
 local-user netconf password irreversible-cipher Huawei@123
 local-user netconf privilege level 15
 local-user netconf service-type api
 quit
 netconf
 source ip interface Vlanif 1 port 830
 quit
```

Create a .txt file named **NETCONF** and invoke it in the script.

# Install ncclient.

```
[C:\~]$ pip3 install ncclient
```

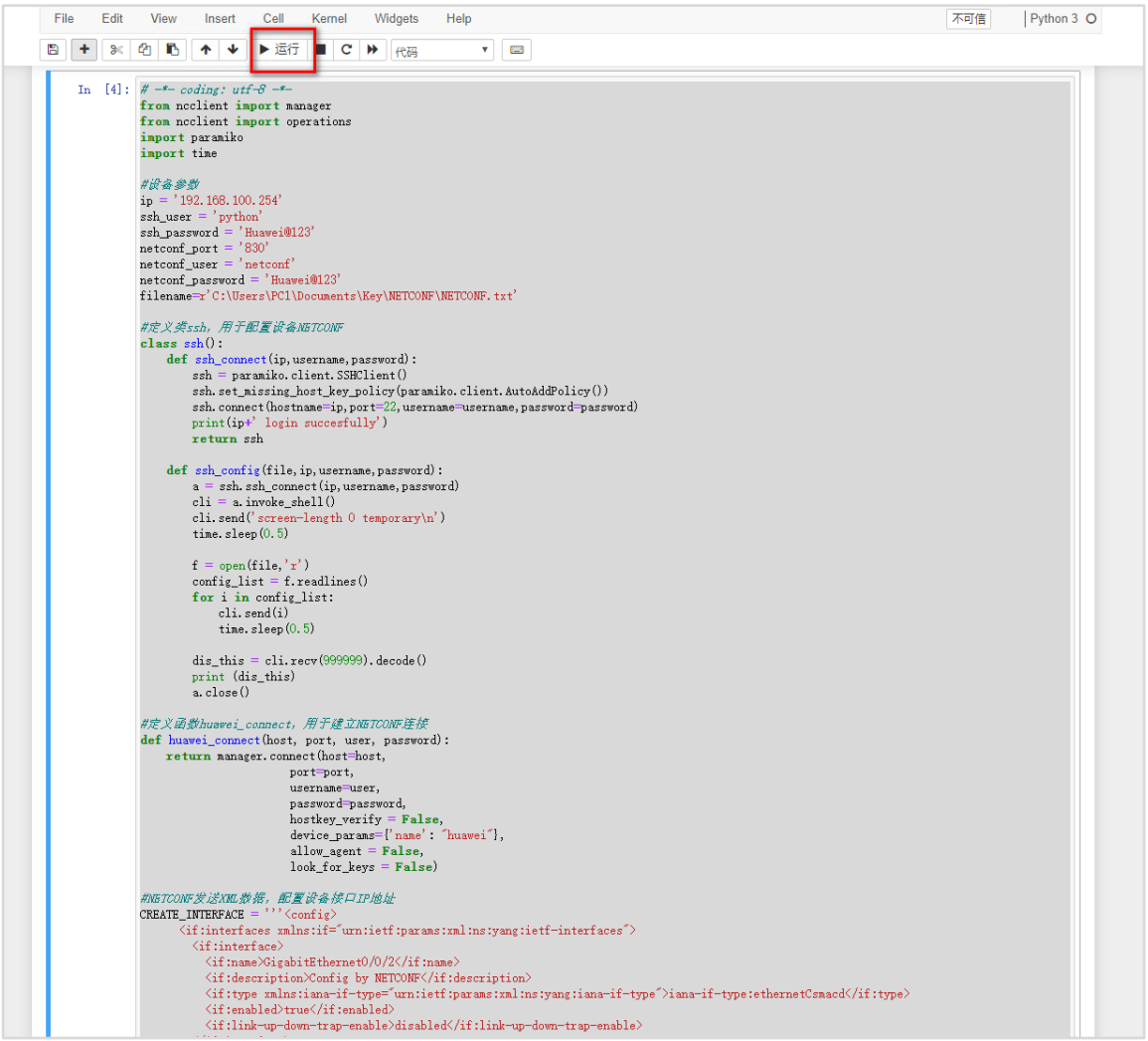
Install ncclient on the CLI using pip3.

# Configure the full code.

```
# -*- coding: utf-8 -*-
from ncclient import manager
from ncclient import operations
import paramiko
import time

# Configure the device parameters.
ip = '192.168.100.254'
ssh_user = 'python'
ssh_password = 'Huawei@123'
netconf_port = '830'
netconf_user = 'netconf'
netconf_password = 'Huawei@123'
filename=r'C:\Users\PC1\Documents\Key\NETCONF\NETCONF.txt'

# Define the SSH class for configuring NETCONF.
class ssh():
    def ssh_connect(ip,username,password):
        ssh = paramiko.client.SSHClient()
        ssh.set_missing_host_key_policy(paramiko.client.AutoAddPolicy())
        ssh.connect(hostname=ip,port=22,username=username,password=password)
        print(ip+' login succesfully')
        return ssh

    def ssh_config(file,ip,username,password):
        a = ssh.ssh_connect(ip,username,password)
        cli = a.invoke_shell()
        cli.send('screen-length 0 temporary\n')
        time.sleep(0.5)

        f = open(file,'r')
        config_list = f.readlines()
        for i in config_list:
```

```
                cli.send(i)
                time.sleep(0.5)

        dis_this = cli.recv(999999).decode()
        print (dis_this)
        a.close()

# Define huawei_connect to establish a NETCONF connection.
def huawei_connect(host, port, user, password):
    return manager.connect(host=host,
                           port=port,
                           username=user,
                           password=password,
                           hostkey_verify = False,
                           device_params={'name': "huawei"},
                           allow_agent = False,
                           look_for_keys = False)

# Send XML data using NETCONF to configure IP addresses for the interfaces.
CREATE_INTERFACE = '''<config>
      <if:interfaces xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <if:interface>
          <if:name>GigabitEthernet0/0/2</if:name>
          <if:description>Config by NETCONF</if:description>
          <if:type xmlns:iana-if-type="urn:ietf:params:xml:ns:yang:iana-if-type">iana-if-
type:ethernetCsmacd</if:type>
          <if:enabled>true</if:enabled>
          <if:link-up-down-trap-enable>disabled</if:link-up-down-trap-enable>
        </if:interface>
      </if:interfaces>
    </config>'''

#Execute the main function, in which the statements are executed in sequence.
if __name__ == '__main__':
    ssh.ssh_config(filename,ip,ssh_user,ssh_password)
    m = huawei_connect(ip,netconf_port,netconf_user,netconf_password)
    m.edit_config(target='running',config=CREATE_INTERFACE)
```

# Execute the code in a compiler.




# Output the result.

```
192.168.100.254 login succesfully

Info: The max number of VTY users is 10, and the number
      of current VTY users on line is 2.
      The current login time is 2021-09-01 16:16:45+08:00.
Info: Lastest accessed IP: 192.168.100.1   Time: 2021-09-01 16:16:05+08:00   Failed: 0   Password will
expire in: -

Info: Smart-upgrade is currently disabled. Enable Smart-upgrade to get recommended version
information.

<ACC_1>screen-length 0 temporary
Info: The configuration takes effect on the current user terminal interface only.
<ACC_1>system-view
Enter system view, return user view with Ctrl+Z.
[ACC_1] aaa
[ACC_1-aaa] local-user netconf password irreversible-cipher Huawei@123
Info: Add a new user.
[ACC_1-aaa] local-user netconf privilege level 15
```

```
Info: After changing the rights (including the password, access type, FTP directory, HTTP directory, and level) of a local user, the rights of users already online do not change. The change takes effect to users who are onboarded after the change.
[ACC_1-aaa] local-user netconf service-type api
[ACC_1-aaa] quit
[ACC_1] netconf
[ACC_1-netconf] source ip interface Vlanif 1 port 830
[ACC_1-netconf] quit
```

# Log in to the device and check information about GE0/0/2 on ACC_1.

```
[ACC_1]interface GigabitEthernet 0/0/2
[ACC_1-GigabitEthernet0/0/2]dis this
#
interface GigabitEthernet0/0/2
 description Config by NETCONF
 undo enable snmp trap updown
```

The interface description has been configured.

The NETCONF configuration is successfully delivered.

Step 4 Parse the script.

The script is parsed as follows:

# Import the Paramiko module.

```
# -*- coding: utf-8 -*-
from ncclient import manager
from ncclient import operations
import paramiko
import time
```

By default, Anaconda does not contain ncclient. You need to manually install ncclient. When Anaconda is installed, pip3 is automatically installed. Then ncclient can be installed through the CLI on Windows.

```
[C:\~]$ pip3 install ncclient
Collecting ncclient
  Downloading ncclient-0.6.12.tar.gz (106 kB)
Requirement already satisfied: setuptools>0.6 in c:\users\pc1\anaconda3\lib\site-packages (from ncclient) (52.0.0.post20210125)
Requirement already satisfied: paramiko>=1.15.0 in c:\users\pc1\anaconda3\lib\site-packages (from ncclient) (2.7.2)
Requirement already satisfied: lxml>=3.3.0 in c:\users\pc1\anaconda3\lib\site-packages (from ncclient) (4.6.3)
Requirement already satisfied: six in c:\users\pc1\anaconda3\lib\site-packages (from ncclient) (1.15.0)
Requirement already satisfied: bcrypt>=3.1.3 in c:\users\pc1\anaconda3\lib\site-packages (from paramiko>=1.15.0->ncclient) (3.2.0)
Requirement already satisfied: cryptography>=2.5 in c:\users\pc1\anaconda3\lib\site-packages (from paramiko>=1.15.0->ncclient) (3.4.7)
Requirement already satisfied: pynacl>=1.0.1 in c:\users\pc1\anaconda3\lib\site-packages (from paramiko>=1.15.0->ncclient) (1.4.0)
```

```
Requirement already satisfied: cffi>=1.1 in c:\users\pc1\anaconda3\lib\site-packages (from
bcrypt>=3.1.3->paramiko>=1.15.0->ncclient) (1.14.5)
Requirement already satisfied: pycparser in c:\users\pc1\anaconda3\lib\site-packages (from
cffi>=1.1->bcrypt>=3.1.3->paramiko>=1.15.0->ncclient) (2.20)
Building wheels for collected packages: ncclient
  Building wheel for ncclient (setup.py): started
  Building wheel for ncclient (setup.py): finished with status 'done'
  Created wheel for ncclient: filename=ncclient-0.6.12-py2.py3-none-any.whl size=83781
sha256=61d2fe8cc5de8a4698902851204759a5c5debc5f01cde3c85e0b47bebe5dfabb
  Stored in directory:
c:\users\pc1\appdata\local\pip\cache\wheels\b2\ea\95\9bdf798bf88d883a5ee9536d8a483042030d182
ab252953b9d
Successfully built ncclient
Installing collected packages: ncclient
Successfully installed ncclient-0.6.12
```

# Define the device parameter variables.

```
ip = '192.168.100.254'
ssh_user = 'python'
ssh_password = 'Huawei@123'
netconf_port = '830'
netconf_user = 'netconf'
netconf_password = 'Huawei@123'
filename=r'C:\Users\PC1\Documents\Key\NETCONF\NETCONF.txt'
```

Define variables to set parameters for the device. Set the host IP address, SSH user name, SSH password, NETCONF port, NETCONF user name, NETCONF password, and local file name.

# Declare a class named **ssh**.

```
class ssh():
```

The class contains **ssh_connect()** and **ssh_config()**. **ssh_connect()** is used to establish an SSH connection and **ssh_config()** is used to deliver the SSH configuration.

# Define ssh_connect.

```
    def ssh_connect(ip,username,password):
        ssh = paramiko.client.SSHClient()
        ssh.set_missing_host_key_policy(paramiko.client.AutoAddPolicy())
        ssh.connect(hostname=ip,port=22,username=username,password=password)
        print(ip+' login succesfully')
        return ssh
```

Define **ssh_connect(ip,username,password)** in the SSH class. Enter the SSH IP address, user name, and password. This function encapsulates the paramiko method to create an SSH session. For details, see the preceding sections.

# Define **ssh_config**.

```
    def ssh_config(file,ip,username,password):
        a = ssh.ssh_connect(ip,username,password)
        cli = a.invoke_shell()
```

```
            cli.send('screen-length 0 temporary\n')
            time.sleep(0.5)

            f = open(file,'r')
            config_list = f.readlines()
            for i in config_list:
                  cli.send(i)
                  time.sleep(0.5)

            dis_this = cli.recv(999999).decode()
            print (dis_this)
close()
```

Define **ssh_config(file,ip,username,password)** in the SSH class. Define the configuration file path, SSH IP address, user name, and password.

**ssh_config()** connects to the device by invoking **ssh_connect()** and then sends configuration commands. Use the **open** function to open the local **NETCONF.txt** file and write the file to the SSH channel line by line. Check the interaction with the device and close the session.

# Define huawei_connect.

```
def huawei_connect(host, port, user, password):
      return manager.connect(host=host,
                             port=port,
                             username=user,
                             password=password,
                             hostkey_verify = False,
                             device_params={'name': "huawei"},
                             allow_agent = False,
                             look_for_keys = False)
```

Define the **huawei_connect(host, port, user, password)** function. Set the IP address of the NETCONF host, port, NETCONF user name, and NETCONF password for the function. The return result of the function is the **manager.connect** method of ncclient.

**manager.connect** is used to establish a NETCONF connection. The parameter is defined in RFC 4741. Two options are available to **device_params** on Huawei devices: **huawei** and **huaweiyang**, which indicate the IETF YANG model and Huawei YANG model, respectively.

# Build an XML file.

```
CREATE_INTERFACE = '''<config>
        <if:interfaces xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces">
          <if:interface>
            <if:name>GigabitEthernet0/0/2</if:name>
            <if:description>Config by NETCONF</if:description>
            <if:type xmlns:iana-if-type="urn:ietf:params:xml:ns:yang:iana-if-type">iana-if-
type:ethernetCsmacd</if:type>
            <if:enabled>true</if:enabled>
            <if:link-up-down-trap-enable>disabled</if:link-up-down-trap-enable>
          </if:interface>
        </if:interfaces>
```

```
</config>'''
```

NETCONF transfers configuration information through XML files. XML is a commonly used text format that allows you to nest and expand data. A complete NETCONF session includes a transport layer, a message layer, an operation layer, and a content layer. Only the information at the operation and content layers is included in the current XML configuration file.

Basic NETCONF operations include **get-config**, **get**, **edit-config**, **copy-config**, **delete-config**, **lock**, **unlock**, **close-session**, and **kill session**. For example, **edit-config** is used for the information at the operation layer in this example, and the corresponding operation attribute is **merge**, which modifies existing data (if the target data exists) or creates data (if the target data does not exist).

The NETCONF content layer is used to edit specific parameters. In this example, configure **description** of GE0/0/2. For details about the XML format, see **NETCONF YANG API Reference** in the product documentation.

# Run the main function.

```
if __name__ == '__main__':
    ssh.ssh_config(filename,ip,ssh_user,ssh_password)
    m = huawei_connect(ip,netconf_port,netconf_user,netconf_password)
m.edit_config(target='running',config=CREATE_INTERFACE)
```

Run the main function.

Run **ssh.ssh_config(filename,ip,ssh_user,ssh_password)** to invoke **ssh_config** of the SSH class. The input parameters are the variables defined in step 2.

Run **huawei_connect(ip,netconf_port,netconf_user,netconf_password)** to assign a value to *m*. Set NETCONF parameters to establish a NETCONF connection.

Finally, run **m.edit_config(target='running',config=CREATE_INTERFACE)**. Send the constructed XML configuration file to the running configuration file of the device through **edit_config**.

## 10.2.3 Quiz

What are the differences between NETCONF- and SNMP-based device information query?

# 10.3 OPS Lab

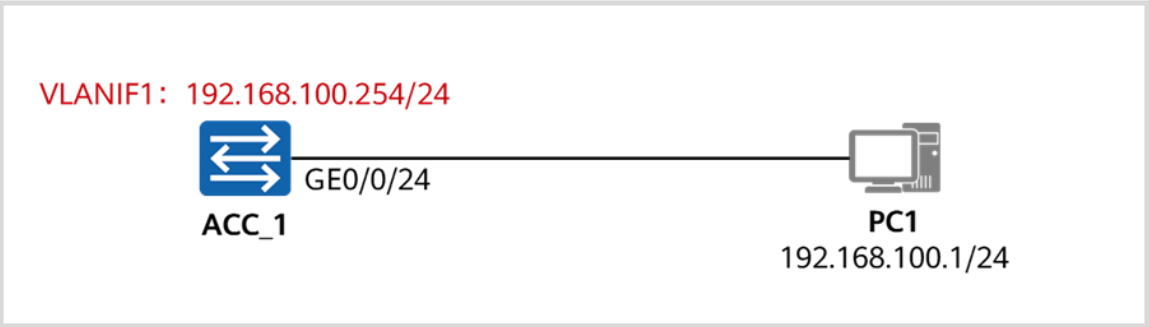
## 10.3.1 About This Lab

### 10.3.1.1 Objectives

- Configure automatic configuration creation and deletion for the device through OPS.
- Describe the OPS workflow.
- Describe common OPS APIs.

### 10.3.1.2 Networking Description

**Figure 10-3 OPS lab topology**




In this lab, ACC_1 and PC1 are used. The IP address of VLANIF 1 on ACC_1 is 192.168.100.254/24, and the IP address of PC1 is 192.168.100.1/24. Upload the script to ACC_1 through FTP. Configure the OPS function of ACC_1 to run the Python script.

The script in this lab can detect the status of GE0/0/23 on ACC_1. When GE0/0/23 is Up, VLAN 10 can be automatically created. Add GE0/0/23 to VLAN 10, create VLANIF 10, configure an IP address, and configure a static route. When detecting that the interface is Down, the system automatically deletes the corresponding configuration.

## 10.3.2 Lab Task

### 10.3.2.1 Configuration Roadmap

1. Perform basic configuration on ACC_1.
2. Write the script on PC1 and upload it to ACC_1.
3. Install the Python script on ACC_1.
4. Execute and verify the script.
5. Parse the script.

### 10.3.2.2 Configuration Procedure

Step 1 Perform basic configuration on ACC_1.

Complete basic connectivity and configure FTP on the device.

# Name the devices.

N/A

# Configure an IP address for VLANIF 1.

```
[ACC_1]interface Vlanif1
[ACC_1-Vlanif1] ip address 192.168.100.254 255.255.255.0
[ACC_1-Vlanif1] quit
```

# Check the connectivity between ACC_1 and PC1.

```
[ACC_1]ping 192.168.100.1
  PING 192.168.100.1: 56  data bytes, press CTRL_C to break
    Reply from 192.168.100.1: bytes=56 Sequence=1 ttl=128 time=1 ms
    Reply from 192.168.100.1: bytes=56 Sequence=2 ttl=128 time=1 ms
    Reply from 192.168.100.1: bytes=56 Sequence=3 ttl=128 time=1 ms
    Reply from 192.168.100.1: bytes=56 Sequence=4 ttl=128 time=1 ms
    Reply from 192.168.100.1: bytes=56 Sequence=5 ttl=128 time=1 ms

  --- 192.168.100.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 1/1/1 ms
```

ACC_1 can communicate with PC1.

# Enable FTP and set the source interface.

```
[ACC_1]FTP server enable
Info: The FTP server is already enabled.
[ACC_1]FTP server-source all-interface
```

Set the source interface to any interface.

# Configure the **admin** user.

```
[ACC_1]aaa
[ACC_1-aaa] local-user admin password irreversible-cipher Huawei@123
[ACC_1-aaa] local-user admin privilege level 15
[ACC_1-aaa] local-user admin ftp-directory flash:
[ACC_1-aaa] local-user admin service-type telnet terminal ssh ftp http
```

## Step 2 Compile and upload scripts.

Compile the Python script and upload it to ACC_1 through FTP.

# Compile the **linkup.py** script.

```
# -*- coding: utf-8 -*-
import ops              # Import the ops module.
import sys              # Import the sys module.
import os               # Import the os module.

# Subscription processing function.
def ops_condition (ops):
    # Subscribe to the trap generated when GigabitEthernet0/0/23 goes Up.
    value1, descri_str1 = ops.trap.subscribe(
```

```
    "GE23_log",
    "IFNET",
    1,
    "IF_LINKUP",
    "Interface 29 turned into UP
state.(AdminStatus=1,OperStatus=1,InterfaceName=GigabitEthernet0/0/23)",
    "1.3.6.1.6.3.1.1.5.4"
    )

    return 0

# Work processing function.
def ops_execute (ops):

    # Print logs to notify users of the route change.
    value1, descri_str1 = ops.syslog("Route 1.1.1.1/32 has configured", "warning", "syslog")

    # Add a route.
    handle, descri_str = ops.cli.open()
    result, n11, n21 = ops.cli.execute(handle,"system-view")
    result, n11, n21 = ops.cli.execute(handle,"vlan 10")
    result, n11, n21 = ops.cli.execute(handle,"port GigabitEthernet 0/0/23")
    result, n11, n21 = ops.cli.execute(handle,"interface vlanif 10")
    result, n11, n21 = ops.cli.execute(handle,"ip address 10.0.0.1 24")
    result, n11, n21 = ops.cli.execute(handle,"quit")
    result, n11, n21 = ops.cli.execute(handle,"ip route-static 1.1.1.1 32 10.0.0.2 ")
    result = ops.cli.close(handle)

return 0
```

# Compile the **linkdown.py** script.

```
# -*- coding: utf-8 -*-
import ops              # Import the ops module.
import sys              # Import the sys module.
import os               # Import the os module.

# Subscription processing function.
def ops_condition (ops):
    # Subscribe to the trap generated when GigabitEthernet0/0/23 goes Down.
    value1, descri_str1 = ops.trap.subscribe(
    "GE23_log",
    "IFNET",
    1,
    "IF_LINKDOWN",
    "Interface 29 turned into DOWN
state.(AdminStatus=1,OperStatus=2,InterfaceName=GigabitEthernet0/0/23)",
    "1.3.6.1.6.3.1.1.5.3"
    )

    return 0

# Work processing function.
def ops_execute (ops):
```

```
    # Print logs to notify users of the route change.
    value1, descri_str1 = ops.syslog("Route 1.1.1.1/32 has been removed", "warning", "syslog")

    # Delete the route.
    handle, descri_str = ops.cli.open()
    result, n11, n21 = ops.cli.execute(handle,"system-view")
    result, n11, n21 = ops.cli.execute(handle,"undo ip route-static 1.1.1.1 32 10.0.0.2")
    result, n11, n21 = ops.cli.execute(handle,"undo interface Vlanif 10 ")
    result, n11, n21 = ops.cli.execute(handle,"vlan 10")
    result, n11, n21 = ops.cli.execute(handle,"undo port GigabitEthernet 0/0/23")
    result, n11, n21 = ops.cli.execute(handle,"quit")
    result, n11, n21 = ops.cli.execute(handle,"undo vlan 10")
    result = ops.cli.close(handle)

return 0
```

# Log in to ACC_1 using the FTP function of the CMD and upload the script.

```
PS C:\Users\PC1\Documents\Key\OPS> ftp
ftp> open 192.168.100.254
Connected to 192.168.100.254.
220 FTP service ready.
530 Please login with USER and PASS.
User (192.168.100.254:(none)): admin
331 Password required for admin.
Password:
230 User logged in.
ftp> put linkdown.py
200 Port command okay.
150 Opening ASCII mode data connection for directory list.
226 Transfer complete.
ftp: 1328 byte(s) received in 0.13 second(s) 10.62 Kbyte(s)/sec.
ftp> put linkup.py
200 Port command okay.
150 Opening ASCII mode data connection for directory list.
226 Transfer complete.
ftp: 1316   byte(s) received in 0.20 second(s) 6.48 Kbyte(s)/sec.
ftp>
```

Here, uploading files through FTP of the terminal on Windows is used as an example. The FTP operations vary depending on the FTP software used.

Step 3 Install the script.

Install the script and load it on the OPS.

# Install the Python script.

```
<ACC_1>ops install file linkup.py
<ACC_1>ops install file linkdown.py
```

# Configure the Python script assistant.

```
<ACC_1>system-view
Enter system view, return user view with Ctrl+Z.
```

```
[ACC_1]ops
[ACC_1-ops]script-assistant python linkup.py
[ACC_1-ops]script-assistant python linkdown.py
```

# Display the OPS configuration result.

```
[ACC_1]display ops assistant current
----------------------------------------------------------------Assistant                                State
Condition----------------------------------------------------------------linkup.py
ready               traplinkdown.py                          ready               trap--------------------------------
---------------------------------
```

The scripts are in **Ready** state, and the conditions of all the OPS subscription are **Trap**.

Step 4 Execute and verify the script.

Enable GE0/0/23 to go Up by running the **Loopback** command and check the OPS execution result.

# Enable GE0/0/23 forcibly.

```
[ACC_1]interface GigabitEthernet 0/0/23
[ACC_1-GigabitEthernet0/0/23]loopback internal
```

# Display the OPS execution result.

```
Sep  2 2021 15:01:16+08:00 ACC_1 %%01OPSA/4/SCRIPT_LOG(l)[143]:OPS: Route 1.1.1.1/32 has
configured (user="linkup.py", session=2751994760)
```

The log information indicates that the script has been successfully executed.

# Display the route configuration.

```
[ACC_1]display current-configuration | in route
ip route-static 1.1.1.1 255.255.255.255 10.0.0.2
```

# Display the configurations of VLANs, interfaces, and VLANIF interfaces.

```
[ACC_1]display vlan 10
--------------------------------------------------------------------------------
U: Up;              D: Down;              TG: Tagged;              UT: Untagged;
MP: Vlan-mapping;                         ST: Vlan-stacking;
#: ProtocolTransparent-vlan;       *: Management-vlan;
--------------------------------------------------------------------------------

VID   Type      Ports
--------------------------------------------------------------------------------
10    common    UT:GE0/0/23(U)

VID   Status    Property         MAC-LRN Statistics Description
--------------------------------------------------------------------------------
10    enable    default          enable   disable      VLAN 0010
```

```
[ACC_1]interface GigabitEthernet 0/0/23
[ACC_1-GigabitEthernet0/0/23]display this
#
interface GigabitEthernet0/0/23
 loopback internal
 port default vlan 10
```

```
[ACC_1]display ip   interface brief   | in 10
*down: administratively down
^down: standby
(l): loopback
(s): spoofing
(E): E-Trunk down
The number of interface that is UP in Physical is 3
The number of interface that is DOWN in Physical is 1
The number of interface that is UP in Protocol is 3
The number of interface that is DOWN in Protocol is 1

Interface                         IP Address/Mask      Physical   Protocol
Vlanif1                           192.168.100.254/24   up         up
Vlanif10                          10.0.0.1/24          up         up
```

All commands in **linkup.py** have been executed.

# Disable GE0/0/23.

```
[ACC_1]interface GigabitEthernet 0/0/23
[ACC_1-GigabitEthernet0/0/23] undo loopback
```

The process for checking the execution of **linkdown.py** is omitted.

Step 5 Parse the script.

The script is parsed as follows. Here, **linkup.py** is used as an example.

# Import the Paramiko module.

```
# -*- coding: utf-8 -*-
import ops              # Import the ops module.
import sys              # Import the sys module.
import os               # Import the os module.
```

Import modules. The ops module is mandatory. Other modules are imported based on the modules to be used in the script.

# Subscription processing function.

```
def ops_condition (ops):
    # Subscribe to the log generated when GigabitEthernet0/0/23 goes Up.
    value1, descri_str1 = ops.trap.subscribe(
    "GE23_log",
    "IFNET",
    1,
```

```
    "IF_LINKUP",
    "Interface 29 turned into UP
state.(AdminStatus=1,OperStatus=1,InterfaceName=GigabitEthernet0/0/23)",
    "1.3.6.1.6.3.1.1.5.4"
    )

    return 0
```

On the S5731, the OPS needs to execute the subscription processing function and then the execution processing function.

Here, you can use the **trap.subscribe** interface of OPS to monitor the **trap** logs of the device.

For details about OPS APIs, see the configuration document of the device. Take the S5731 as an example. You can choose **Configuration** > **Device Management Configuration** > **OPS Configuration** > **OPS API List** to query the information.

# Work processing function.

```
def ops_execute (ops):

    # Print logs to notify users of the route change.
    value1, descri_str1 = ops.syslog("Route 1.1.1.1/32 has configured", "warning", "syslog")

    # Add a route.
    handle, descri_str = ops.cli.open()
    result, n11, n21 = ops.cli.execute(handle,"system-view")
    result, n11, n21 = ops.cli.execute(handle,"vlan 10")
    result, n11, n21 = ops.cli.execute(handle,"port GigabitEthernet 0/0/23")
    result, n11, n21 = ops.cli.execute(handle,"interface vlanif 10")
    result, n11, n21 = ops.cli.execute(handle,"ip address 10.0.0.1 24")
    result, n11, n21 = ops.cli.execute(handle,"quit")
    result, n11, n21 = ops.cli.execute(handle,"ip route-static 1.1.1.1 32 10.0.0.2 ")
    result = ops.cli.close(handle)

    return 0
```

Print syslogs for users through the OPS syslog interface.

Open the CLI by running **ops.cli.open()**, run **ops.cli.execute()**, and close the CLI.

**----End**

## 10.3.3 Quiz

How to notify users when they enter high-risk commands using OPS?

# Reference Answers to Quiz

Answers to the quizzes in **IPv4 Routing**.

1. The OSPF inter-area route advertisement depends on Type 3 LSAs. In OSPF, an ABR imports inter-area routes destined for other areas into the local area through Type 3 LSAs. Users can run the **filter** command in the ABR area view to set filtering conditions for incoming and outgoing ABR Type 3 LSAs. Only the routes permitted by the filtering policy can be advertised or received. Note that the **filter** command in the area view can be run only on ABRs and takes effect only for Type 3 LSAs. OSPF intra-area route calculation depends on Type 1 LSAs and Type 2 LSAs. Therefore, the **filter** command cannot filter these LSAs.
2. Deploy a routing policy in the inbound direction of the local device or the outbound direction of the peer device.

Answer to the quiz in **IPv6 Routing**.

1. The Nexthop attribute and NLRI no longer exist in BGP4+. The MP_REACH_NLRI attribute is added in BGP4+ to carry the next hop and routing information of IPv6 routes.

Answer to the quiz in **MPLS VPN**.

1. The tag value is generated when BGP routes are imported into OSPF on PE2. By default, no specific configurations are implemented. The value consists of a fixed prefix (0XD000) and an AS number. Therefore, by converting 0XD000 into a binary number, 1101000000000000 is obtained. Similarly, by converting AS number 65100 into a binary number, 1111111001001100 is obtained. After merging the two numbers and converting the result number into decimal, 3489726028 is obtained.

Answer to the quiz in **EVPN**.

1. The information is advertised by Type 2 routes.

Answers to the quizzes in **VXLAN Lab**.

1. The broadcast packets will be flooded in the local BD and the packets will be duplicated and sent to the remote VTEP in the ingress replication list.
2. After receiving the VXLAN packets, Edge_2 checks the destination MAC address in the outer data frame and determines that it needs to terminate the data frame locally. Then Edge_2 checks the destination IP address of the outer IP header and finds that the destination IP address is the IP address of its Loopback interface. Subsequently, Edge_2 further decapsulates the packets and determines the corresponding BD based on the VNI in the VXLAN header. After that, Edge_2 checks the destination MAC address of the inner data frame, finding that the address is not the MAC address of a local interface. Finally, Edge_2 queries the destination MAC address in the corresponding BD. According to the result, Edge_2 externally forwards the data frame through a local sub-interface.
3. Layer 3 forwarding between VTEPs can be implemented by enabling ARP proxy on the VBDIF interfaces on the distributed gateways. In this case, during communication in the same network segment, the peer ARPs learned by the terminals are gateway MAC addresses (the MAC addresses of the VBDIF interfaces). When the VTEP receives the packets, it implements Layer 3 forwarding and matches the 32-bit host routes generated based on ARP to implement route forwarding between VTEPs.

Answer to the quiz in **VXLAN-based Virtualized Campus Network Deployment**.

1. The VN is displayed as an IP VPN instance on the CLI and the network service resources are bound to the corresponding VPN instance. When Border advertises EVPN routes, the VNs on Edge_1 and Edge_2 can learn the routes corresponding to network service resources through the planned RT, implementing route reachability.

Answer to the quiz in **WAN Interconnection Network Deployment**.

1. Configure a policy matrix of the HQ site to deny access from the security group to which users belong to external networks (any security group) and response packets from the security group of the HQ site to external networks. In this way, the HQ and branch sites cannot communicate with each other.

Answers to the quizzes in **SR-MPLS**:

1. Unlike in MPLS forwarding, the outer label remains unchanged in a BE scenario.
2. Configure an explicit path and specify first the node SID and then the adjacency SID of the device for the explicit path.
3. An SR-MPLS Policy is identified by <headend, color, endpoint>.

Answers to the quizzes in **SRv6**:

1. End.DT6 SID.
2. Deploy SBFD for SRv6 Policy on the headend to implement unidirectional BFD.

Answers to the quizzes in **Open Network Programmability**.

1. You can place the commands to be executed in the list. You can read the commands in the list in a loop, and run the commands in a loop each time.
2. The NETCONF query object is YANG, and the SNMP query object is MIB.
3. Monitor the user command line input in real time through ops.cli.subscribe, and output to the terminal through ops.terminal.write.