VLAN SETUP

CCNA 200-301

—- 10.11.2024

NIKHIL KUMAR

Types of VLANs

1. **Default VLAN** 🏡
    - **VLAN ID**: 1
    - **Purpose**: The default VLAN where all ports initially reside. All devices are automatically part of VLAN 1 unless assigned to another VLAN.
    - **Example**: Think of VLAN 1 as a "common area" in a building where everyone starts by default.

2. **Native VLAN** 🚧
    - **VLAN ID**: 1 (by default, but can be changed)
    - **Purpose**: Handles untagged traffic on a **trunk link**. Trunk links are used to send data for multiple VLANs over a single cable.
    - **Example**: Imagine this as a dedicated lane for guests without ID badges. Untagged packets are directed to the Native VLAN, usually VLAN 1.
    - **Technical Note**: On trunk links, all packets have a tag indicating their VLAN except those assigned to the Native VLAN. This VLAN ensures backward compatibility with older devices that don't support tagging.

3. **Management VLAN** 🛠️
    - **VLAN ID**: 1 (by default, but can be customized)
    - **Purpose**: This VLAN is used specifically for network management purposes, like accessing switch configurations.
    - **Example**: Think of it as a "control room" accessible only to network admins.
    - **Technical Note**: Using a separate VLAN (other than VLAN 1) for management is recommended for security, as VLAN 1 is widely known.

4. **Voice VLAN** 🎤
    - **Dedicated for Voice Traffic** (typically using **UDP** protocol)
    - **Purpose**: A separate VLAN for voice data (e.g., VoIP) to ensure high quality, low latency, and priority over other types of traffic.
    - **Example**: Imagine it as a soundproof room where voice traffic can flow freely without interruption.
    - **Technical Note**: Voice VLANs are configured with **Quality of Service (QoS)** settings to prioritize voice packets and reduce delays.

VLAN Ranges 🌐

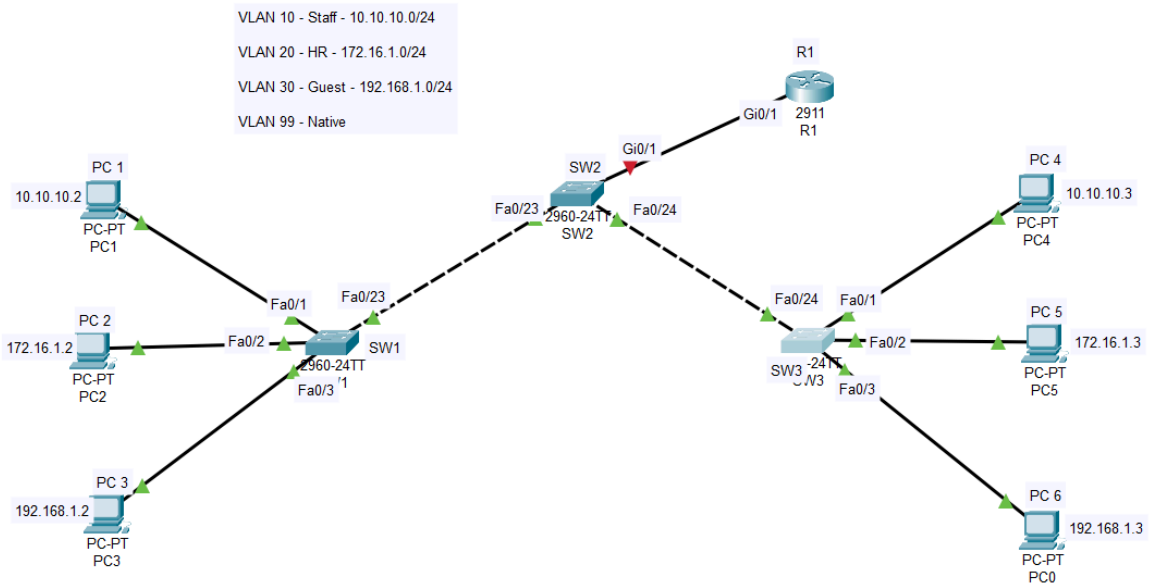
1. **Normal Range VLANs** (IDs 1-1005) 📂
    - **ID Range**: `1-1005`
    - **Special VLAN IDs**: 1002-1005 are reserved for older technologies like **Token Ring** and **FDDI** and are automatically created by Cisco switches.
    - **Storage**: VLAN configurations for this range are saved in the `vlan.dat` file in **flash memory**.
    - **Example**: Think of this range as the "main neighborhoods" with full services, supporting all VLAN features.

2. **Extended Range VLANs** (IDs 1006-4094) 🔄
    - **ID Range**: `1006-4094`
    - **Purpose**: Used in large networks with a need for more VLANs.
    - **Storage**: Configuration for these VLANs is stored in the **running-config** file, not in `vlan.dat`.
    - **Technical Note**: Extended VLANs support fewer features, as they are often used in specialized scenarios like service provider networks.
    - **Example**: This range is like "expansion areas" of the network, providing extra space but with fewer capabilities.

Key Points to Remember 📝

- **All ports** are assigned to **VLAN 1** by default.
- **Native VLAN** is VLAN 1 by default but can be changed to reduce potential security risks.
- **Management VLAN** is also VLAN 1 by default; however, using a different VLAN for management is recommended for security.
- **VLAN 1** and VLANs **1002-1005** cannot be renamed or deleted, as they are essential for Cisco device operations.

Configuration/Setup Switch VLAN 🛠️



Creating VLANs

In this setup, we are using VLANs to segment our network into different departments. Here's how you can create and assign VLANs, as well as verify the configuration on each switch.

VLAN Setup Summary (based on the image)

- **VLAN 10 - Staff**: IP range 10.10.10.0/24 (assigned to PCs like PC1 and PC4)
- **VLAN 20 - HR**: IP range 172.16.1.0/24 (assigned to PCs like PC2 and PC5)
- **VLAN 30 - Guest**: IP range 192.168.1.0/24 (assigned to PCs like PC3 and PC6)
- **VLAN 99 - Native VLAN**: Used for untagged traffic on trunk links

---

Commands to Create and Name VLANs

To create VLANs and assign names based on departments (e.g., Staff, HR), use the following commands in **Global Configuration Mode**:

Enter Configuration Mode:

```
Switch# configure terminal
```

```
SW3#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SW3(config)#
SW3(config)#
SW3(config)#vlan
% Incomplete command.
SW3(config)#vlan 10
SW3(config-vlan)#name Staff
SW3(config-vlan)#vlan 20
SW3(config-vlan)#name HR
SW3(config-vlan)#vlan 30
SW3(config-vlan)#name Guest
SW3(config-vlan)#vlan 99
SW3(config-vlan)#name Native
SW3(config-vlan)#exit
```

Create VLANs and Assign Names:(All Switches SW1,SW2 and SW3)

```
Switch(config)# vlan 10
Switch(config-vlan)# name Staff
Switch(config-vlan)# exit

Switch(config)# vlan 20
Switch(config-vlan)# name HR
Switch(config-vlan)# exit
```

```
Switch(config)# vlan 30
Switch(config-vlan)# name Guest
Switch(config-vlan)# exit

Switch(config)# vlan 99
Switch(config-vlan)# name Native
Switch(config-vlan)# exit
```

```
SW3#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
10   Staff                            active
20   HR                               active
30   Guest                            active
99   Native                           active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

This will create VLANs 10, 20, 30, and 99 with specific names for each department.

Assigning VLANs to Ports (SW1 & SW3)

Each device (PC) connected to a switch is assigned to a specific VLAN based on the port they're connected to.

For example:

- **Port Fa0/1** on **Switch SW1** for VLAN 10 (Staff)
- **Port Fa0/2** on **Switch SW1** for VLAN 20 (HR)
- **Port Fa0/3** on **Switch SW1** for VLAN 30 (Guest)

```
SW3#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SW3(config)#interface fast
SW3(config)#interface fastEthernet 0/1
SW3(config-if)#switchport mode access
SW3(config-if)#switchport
SW3(config-if)#switchport access vlan
% Incomplete command.
SW3(config-if)#switchport access vlan 10
SW3(config-if)#switchport no
SW3(config-if)#switchport nonegoti
SW3(config-if)#switchport nonegotiate
SW3(config-if)#exit
SW3(config)#interface fast
SW3(config)#interface fastEthernet 0/2
SW3(config-if)#switchpo
SW3(config-if)#switchport mode access
SW3(config-if)#switchport access vlan 20
SW3(config-if)#switchport no
SW3(config-if)#switchport nonegotiation
                                    ^
% Invalid input detected at '^' marker.

SW3(config-if)#switchport none
SW3(config-if)#switchport nonegotiate
SW3(config-if)#exit
SW3(config)#
SW3(config)#interface fast
SW3(config)#interface fastEthernet 0/3
SW3(config-if)#sw
SW3(config-if)#switchport mode access
SW3(config-if)#switchport access vlan 30
SW3(config-if)#switchport no
SW3(config-if)#switchport nonego
SW3(config-if)#switchport nonegotiate
SW3(config-if)#exit
SW3(config)#
SW3(config)#
SW3(config)#exit
```

To assign these VLANs to specific ports:

```
Switch(config)# interface fa0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
Switch(config-if)# switchport no
Switch(config-if)# switchport nonegotiate
Switch(config-if)# exit
```

Verifying VLANs

To view all configured VLANs and their status on the switch:

```
Switch# show vlan brief
```

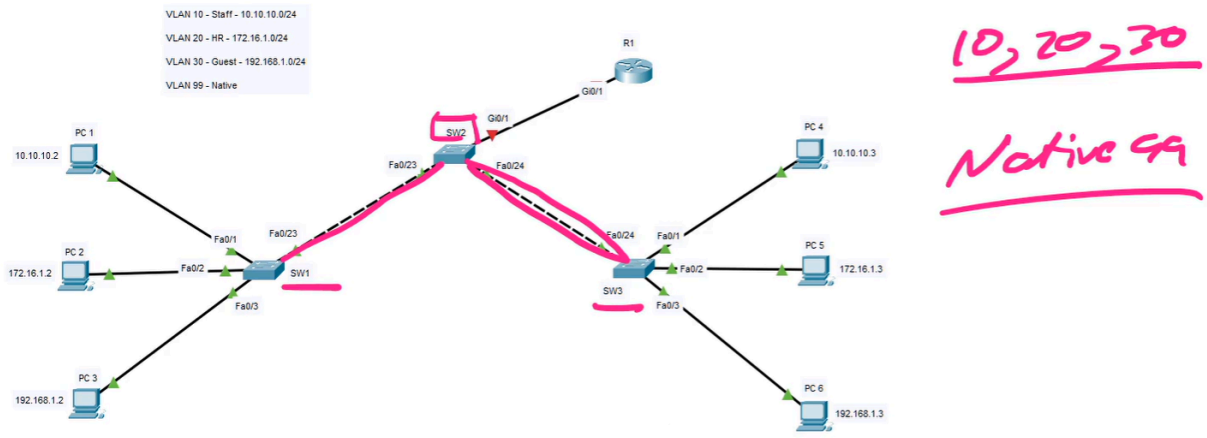
```
SW3#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/4, Fa0/5, Fa0/6, Fa0/7
                                                Fa0/8, Fa0/9, Fa0/10, Fa0/11
                                                Fa0/12, Fa0/13, Fa0/14, Fa0/15
                                                Fa0/16, Fa0/17, Fa0/18, Fa0/19
                                                Fa0/20, Fa0/21, Fa0/22, Fa0/23
                                                Fa0/24, Gig0/1, Gig0/2
10   Staff                            active    Fa0/1
20   HR                               active    Fa0/2
30   Guest                            active    Fa0/3
99   Native                           active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
SW3#
```

This command will display the VLAN IDs, names, and assigned ports, helping verify that the correct VLANs have been set up.

🌐 Setting Up VLANs and Trunking for Inter-Switch Data Transfer

This guide will walk you through creating and configuring VLANs, setting up trunk ports to allow inter-switch VLAN traffic, and troubleshooting common trunking issues. We'll also explore how to verify connectivity across VLANs and resolve errors related to native VLAN mismatches.



🚀 Preliminary Setup: Enable Synchronous Logging

Before diving into VLAN and trunk configurations, enable **logging synchronous** to prevent system messages from interrupting command input. This keeps your terminal output clean and avoids disruptions while configuring.

Commands:

```
Switch# enable
Switch# configure terminal
Switch(config)# line console 0
Switch(config-line)# logging synchronous
Switch(config-line)# exit
```

Why Do This?
When configuring native VLANs, mismatches between switches can lead to errors. For example, setting VLAN 99 as the native VLAN on one switch and leaving the default VLAN 1 as native on another switch will create a mismatch error.

🌐 Step 1: Configuring Native VLAN and Trunk on Switch 1 (SW1)

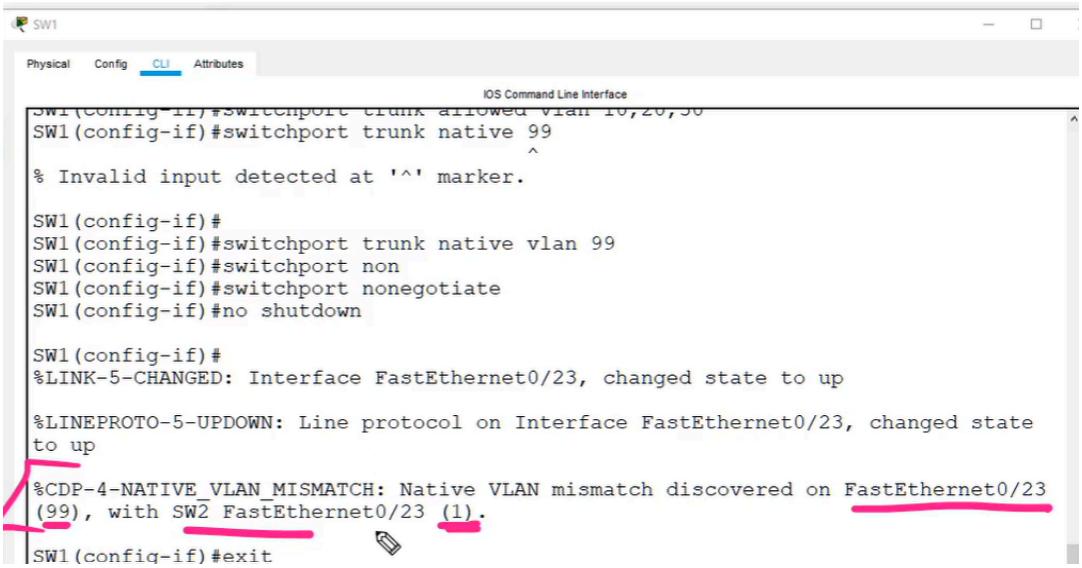
Configure VLAN 99 as Native VLAN on Switch 1:

```
Switch(config)# vlan 99
Switch(config-vlan)# name Native
Switch(config-vlan)# exit
```

```
SW1(config)#line console 0
SW1(config-line)#
SW1(config-line)#loggin
SW1(config-line)#logging sy
SW1(config-line)#logging synchronous
SW1(config-line)#
SW1(config-line)#exit
SW1(config)#
SW1(config)#
SW1(config)#
SW1(config)#interface fa 0/23
SW1(config-if)#shutdown
```

Configure FastEthernet 0/23 as a Trunk Interface:

```
Switch(config)# interface fastEthernet 0/23
Switch(config-if)# shutdown          (Temporarily disable interface)
Switch(config-if)# switchport mode trunk        (Set mode to trunk)
Switch(config-if)# switchport trunk allowed vlan 10,20,30
(Allow VLANs 10, 20, 30)
Switch(config-if)# switchport trunk native vlan 99  (Set VLAN 99)
as native
Switch(config-if)# switchport nonegotiate
(Prevent dynamic trunk negotiation)
```




```
Switch(config-if)# no shutdown                    (Enable the interface)
Switch(config-if)# exit
```

Verify Trunk Configuration:

```
Switch# show interfaces trunk
```

```
SW1#show interface trunk
Port        Mode         Encapsulation  Status        Native vlan
Fa0/23      on           802.1q         trunking      99

Port        Vlans allowed on trunk
Fa0/23      10,20,30

Port        Vlans allowed and active in management domain
Fa0/23      10,20,30

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/23      10,20,30
```

After this setup, you may encounter a **native VLAN mismatch error** if Switch 2 still has VLAN 1 as its native VLAN. Let's address that by configuring Switch 2 accordingly.

🔧 Step 2: Resolving Native VLAN Mismatch on Switch 2 (SW2)

Enable Logging Synchronous on Switch 2:
bash
Copy code

```
Switch# configure terminal
Switch(config)# line console 0
Switch(config-line)# logging synchronous
Switch(config-line)# exit
```

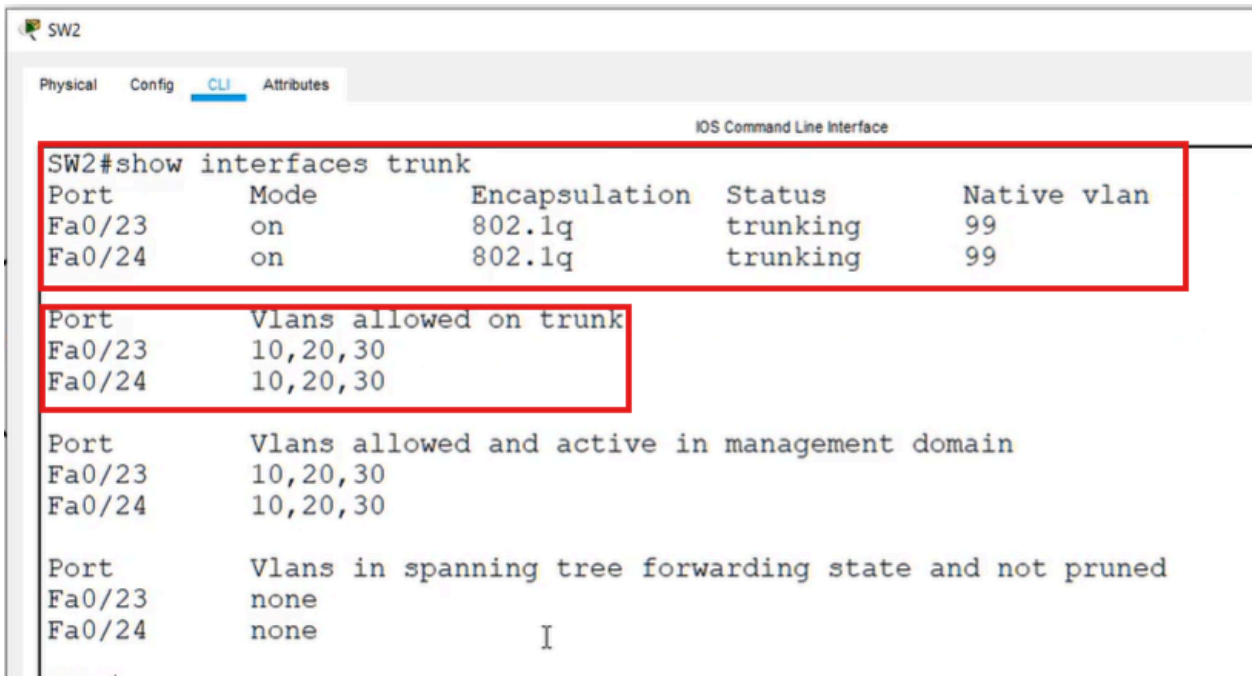
```
SW2(config-if-range)#
SW2(config-if-range)#switchport trunk allowed vlan 10,20,30
SW2(config-if-range)#switch trunk native vlan 99
SW2(config-if-range)#swi
SW2(config-if-range)#switchport none
SW2(config-if-range)#switchport nonegotiate
SW2(config-if-range)#exit
SW2(config)#interface range fastEthernet 0/23 - fastEthernet 0/24
SW2(config-if-range)#
SW2(config-if-range)#no shutdown
```

Configure Trunk for Multiple Ports Using Range Command: Since FastEthernet 0/23 and 0/24 will share the same configuration, we can apply settings to both at once.

```
Switch(config)# interface range fastEthernet 0/23 - 0/24
Switch(config-if-range)# switchport mode trunk
Switch(config-if-range)# switchport trunk allowed vlan 10,20,30
(Allow VLANs 10, 20, 30)
Switch(config-if-range)# switchport trunk native vlan 99
(Set VLAN 99 as native)
Switch(config-if-range)# switchport nonegotiate
(Prevent negotiation)
Switch(config-if-range)# no shutdown
(Enable the interfaces)
Switch(config-if-range)# exit
```

Verify Trunk Configuration:

```
Switch# show interfaces trunk
```



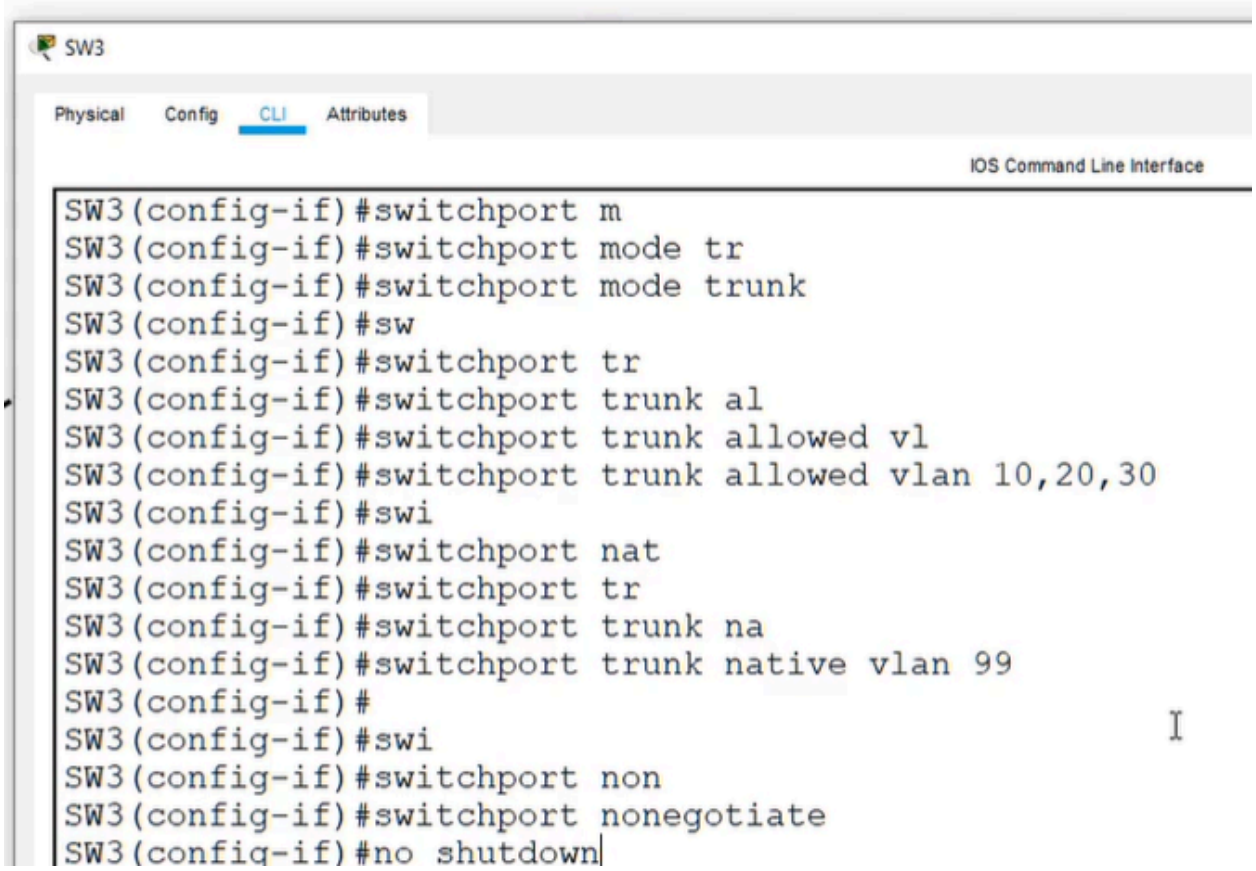
If the mismatch error persists, proceed to configure Switch 3 in the same manner.

🔗 Step 3: Configuring Trunk on Switch 3 (SW3)

Enable Logging Synchronous on Switch 3:

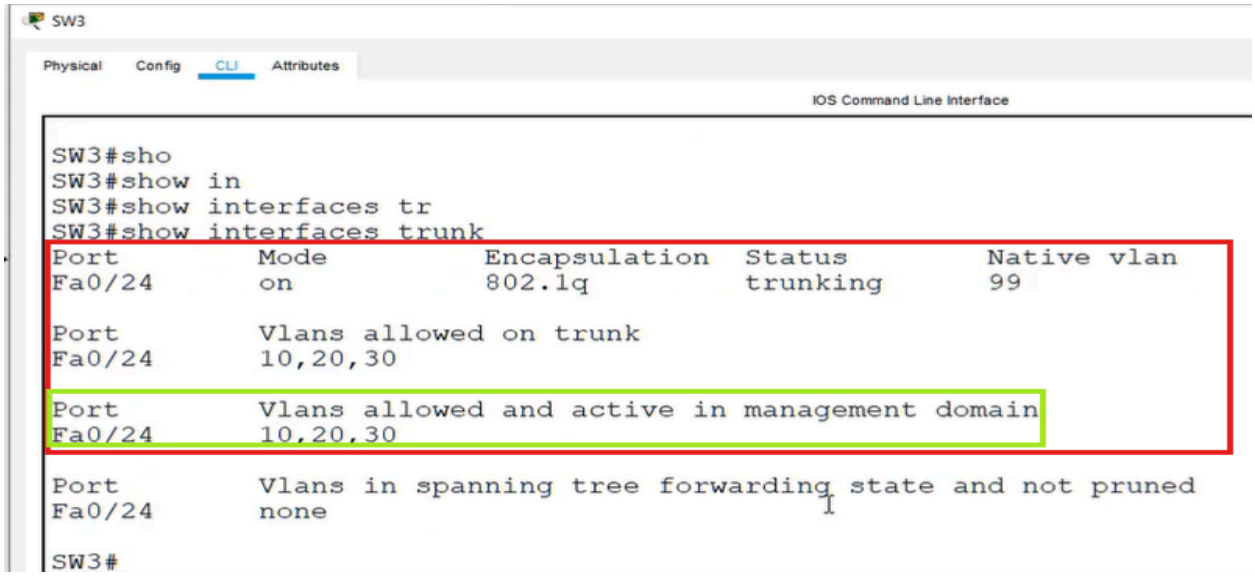
```
Switch# configure terminal
Switch(config)# line console 0
Switch(config-line)# logging synchronous
Switch(config-line)# exit
```

Configure FastEthernet 0/24 as a Trunk Interface:




```
Switch(config)# interface fastEthernet 0/24
Switch(config-if)# shutdown     ( Temporarily disable interface)
Switch(config-if)# switchport mode trunk      (Set mode to trunk)
Switch(config-if)# switchport trunk allowed vlan 10,20,30
(Allow VLANs 10, 20, 30)
Switch(config-if)# switchport trunk native vlan 99
(Set VLAN 99 as native)
Switch(config-if)# switchport nonegotiate
 (Prevent dynamic trunk negotiation)
Switch(config-if)# no shutdown         (Enable the interface)
Switch(config-if)# exit
```

Verify Trunk Configuration:
`Switch# show interfaces trunk`

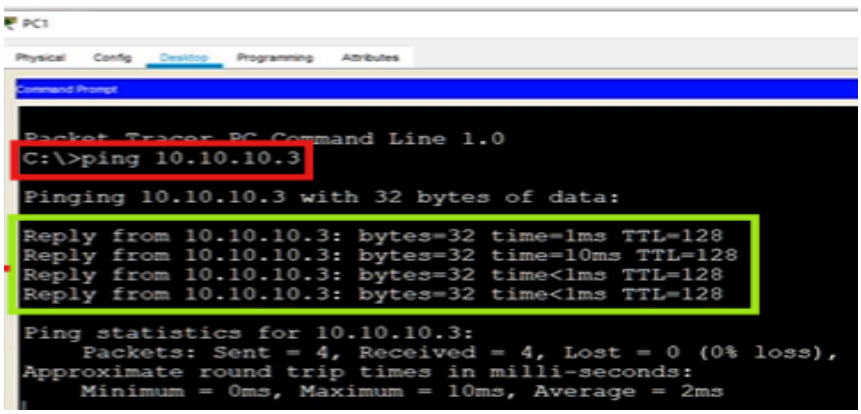


🔍 Verifying Connectivity Across VLANs

To check if VLANs are configured and working correctly, attempt to **ping devices within the same VLAN** and observe whether the packets are routed correctly.

1. **Ping from PC1 to PC4** (VLAN 10):

   From PC1's command prompt:
   `ping 10.10.10.3`



- **Expected Result**: *Successful ping. Both PCs are in VLAN 10, so communication should work.*

2. **Ping from PC2 to PC5** (VLAN 20):

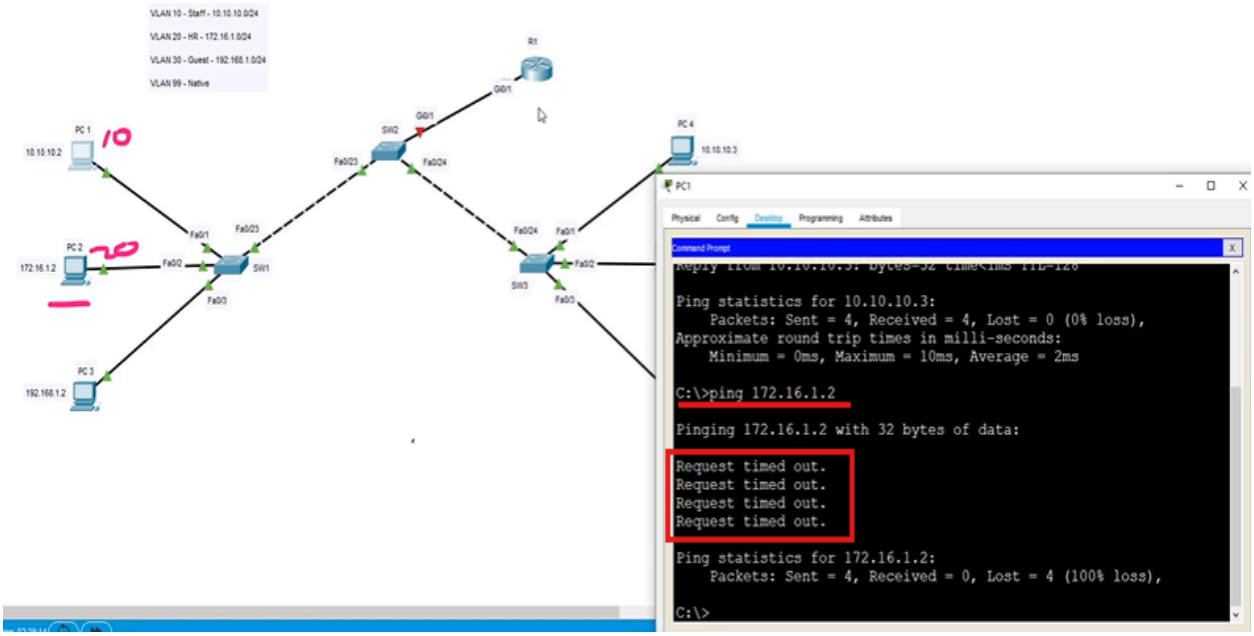
   From PC2's command prompt:
   `ping 172.16.1.3`

   - **Expected Result**: *Successful ping. Both PCs are in VLAN 20, allowing communication within the same VLAN.*

3. **Ping from PC1 (VLAN 10) to PC2 (VLAN 20)**:

   From PC1's command prompt:
   `ping 172.16.1.2`

   - **Expected Result**: *Request timed out. Devices on different VLANs cannot communicate without additional configuration, such as* ***inter-VLAN routing***.



⚙️ Troubleshooting Trunk Links

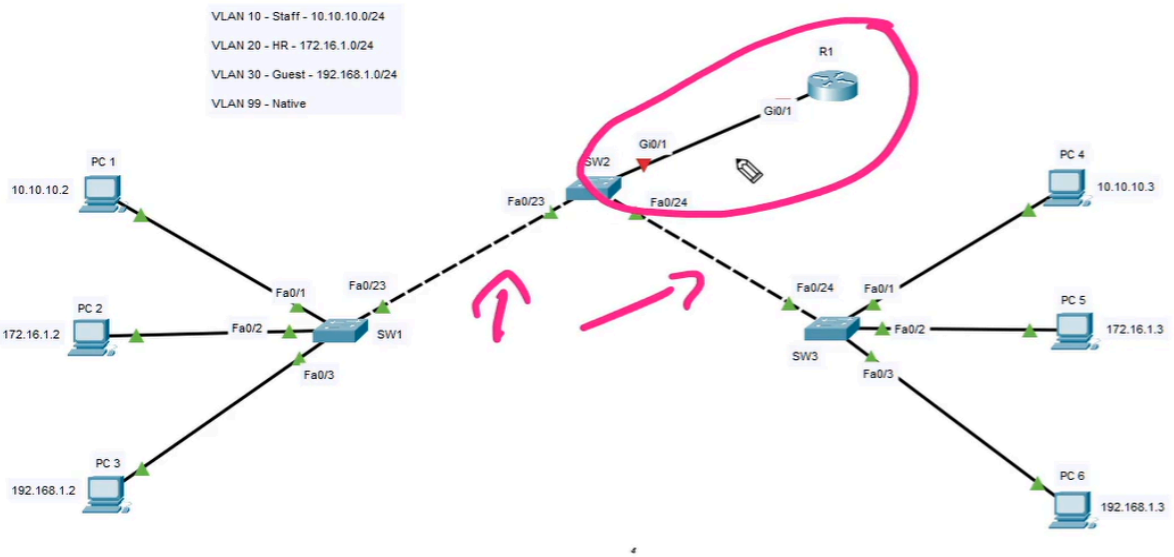
If issues arise with trunk links, check the following:

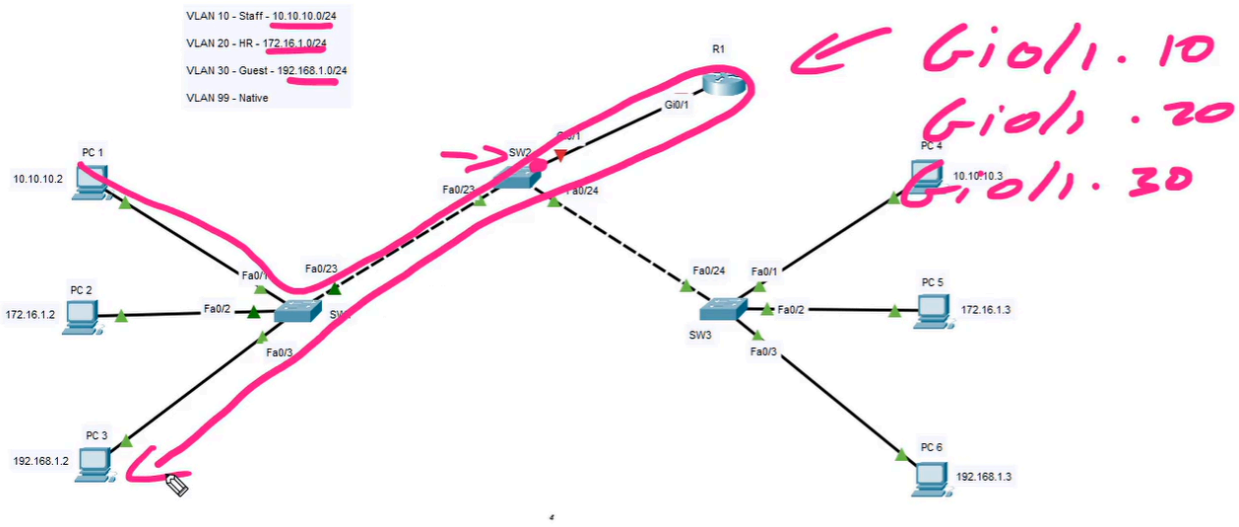
1. **Native VLAN Mismatch** 🔄
   Ensure that the native VLAN is consistent on both sides of the trunk link.
2. **Allowed VLANs on the Trunk Interface** 🚫
   Verify that all necessary VLANs are included in the allowed VLANs list.
3. **Mode Mismatch** 🔧
   Ensure that both ends of the link are set to **trunk mode**. If one side is in **access mode**, communication will fail.

🌐 Configuring Inter-VLAN Routing

When devices in different VLANs need to communicate, inter-VLAN routing is required. In this setup, we use a router (or a Layer 3 switch) to enable communication between VLANs, allowing devices in different segments to exchange data.

To achieve this, we configure **sub-interfaces** on the router. Each sub-interface is assigned to a specific VLAN and acts as the router's virtual interface within that VLAN.





📜 Steps for Configuring Inter-VLAN Routing

1. **Ensure the Switch Port is in Trunk Mode**
   - To allow multiple VLANs to pass through, the switch port connected to the router must be set to **trunk mode**. This ensures the router can receive and transmit tagged VLAN traffic.

```
SW2>
SW2>enable
SW2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SW2(config)#
SW2(config)#
SW2(config)#interface gigabit
SW2(config)#interface gigabitEthernet 0/1
SW2(config-if)#switchport mode trunk
SW2(config-if)#no shu
SW2(config-if)#no shutdown
SW2(config-if)#exit
SW2(config)#
SW2(config)#
```

```
Switch(config)# interface fastEthernet 0/24
Switch(config-if)# switchport mode trunk
```

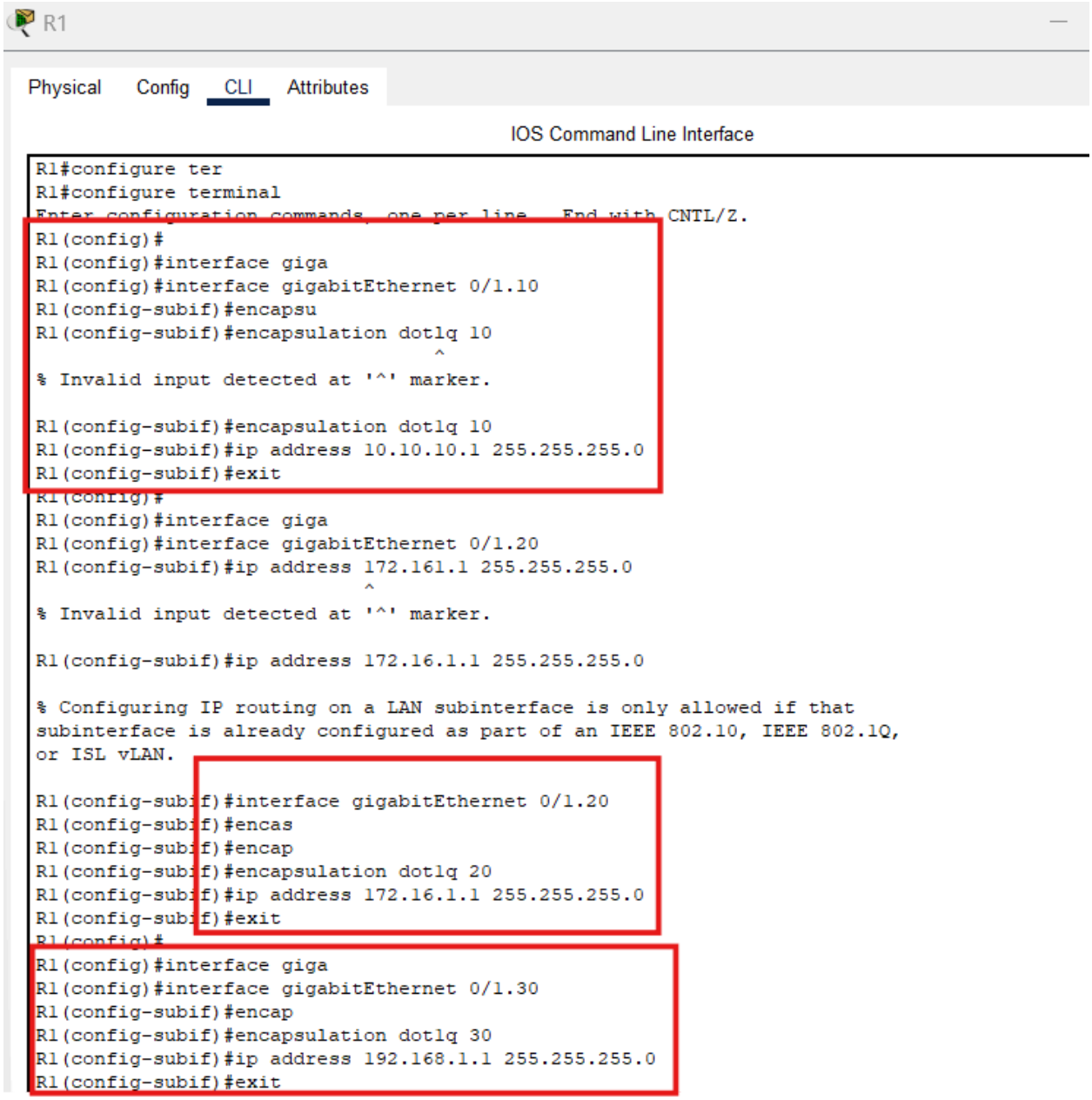
2. **Create Sub-Interfaces on the Router**
    - Sub-interfaces on the router allow it to act as a gateway for each VLAN. For each VLAN, create a sub-interface under the router's physical interface connected to the switch.

Example: Configuring sub-interfaces for VLANs 10, 20, and 30 on interface `GigabitEthernet 0/1`.

```
Router(config)# interface gigabitEthernet 0/1.10
Router(config-subif)# encapsulation dot1Q 10
 # Assign VLAN 10
Router(config-subif)# ip address 10.10.10.1 255.255.255.0
# IP for VLAN 10
Router(config-subif)# exit

Router(config)# interface gigabitEthernet 0/1.20
Router(config-subif)# encapsulation dot1Q 20
 # Assign VLAN 20
Router(config-subif)# ip address 172.16.1.1 255.255.255.0
# IP for VLAN 20
Router(config-subif)# exit

Router(config)# interface gigabitEthernet 0/1.30
Router(config-subif)# encapsulation dot1Q 30
# Assign VLAN 30
Router(config-subif)# ip address 192.168.1.1 255.255.255.0
 # IP for VLAN 30
Router(config-subif)# exit
```


R1
Physical
Config
CLI
Attributes
IOS Command Line Interface
R1#configure ter
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#
R1(config)#interface giga
R1(config)#interface gigabitEthernet 0/1.10
R1(config-subif)#encapsu
R1(config-subif)#encapsulation dotlq 10
^
% Invalid input detected at '^' marker.
R1(config-subif)#encapsulation dotlq 10
R1(config-subif)#ip address 10.10.10.1 255.255.255.0
R1(config-subif)#exit
R1(config)#
R1(config)#interface giga
R1(config)#interface gigabitEthernet 0/1.20
R1(config-subif)#ip address 172.161.1 255.255.255.0
^
% Invalid input detected at '^' marker.
R1(config-subif)#ip address 172.16.1.1 255.255.255.0
% Configuring IP routing on a LAN subinterface is only allowed if that
subinterface is already configured as part of an IEEE 802.10, IEEE 802.1Q,
or ISL vLAN.
R1(config-subif)#interface gigabitEthernet 0/1.20
R1(config-subif)#encas
R1(config-subif)#encap
R1(config-subif)#encapsulation dotlq 20
R1(config-subif)#ip address 172.16.1.1 255.255.255.0
R1(config-subif)#exit
R1(config)#
R1(config)#interface giga
R1(config)#interface gigabitEthernet 0/1.30
R1(config-subif)#encap
R1(config-subif)#encapsulation dotlq 30
R1(config-subif)#ip address 192.168.1.1 255.255.255.0
R1(config-subif)#exit

```
R1#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     unassigned      YES unset  administratively down down
GigabitEthernet0/1     unassigned      YES unset  up                    up
GigabitEthernet0/1.10  10.10.10.1      YES manual up                    up
GigabitEthernet0/1.20  172.16.1.1      YES manual up                    up
GigabitEthernet0/1.30  192.168.1.1     YES manual up                    up
GigabitEthernet0/2     unassigned      YES unset  administratively down down
Vlan1                  unassigned      YES unset  administratively down down
R1#
```

Copy

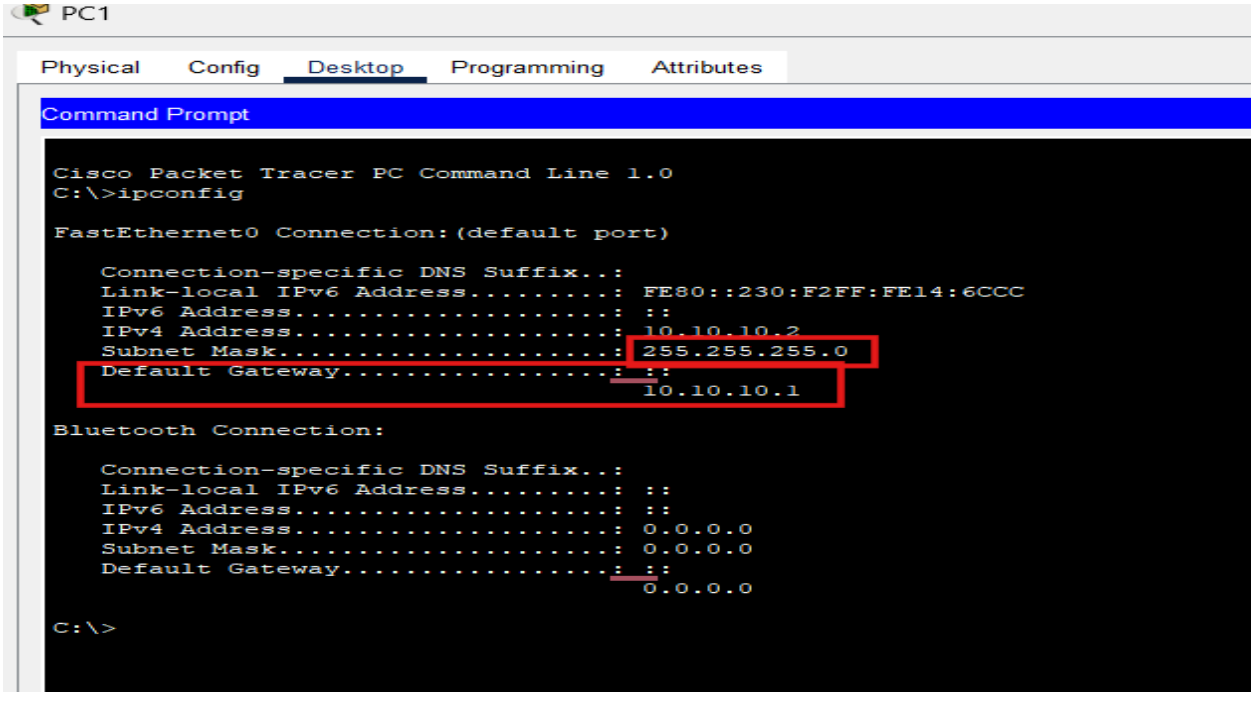
3. **Turn On the Physical Interface of the Router**
   - Ensure the main interface (e.g., `GigabitEthernet 0/1`) is **enabled** to allow traffic flow.

```
Router(config)# interface gigabitEthernet 0/1
Router(config-if)# no shutdown
```

```
R1(config)#
R1(config)#inter
R1(config)#interface giga
R1(config)#interface gigabitEthernet 0/1
R1(config-if)#no shut
R1(config-if)#no shutdown
```

With these configurations, the router can now route traffic between VLANs. Each VLAN's devices use the router's sub-interface IP as their **default gateway**.

- **Now testing connection on the pc1 to check inter vlan connection**.



So we can see that inter vlan setup success !!

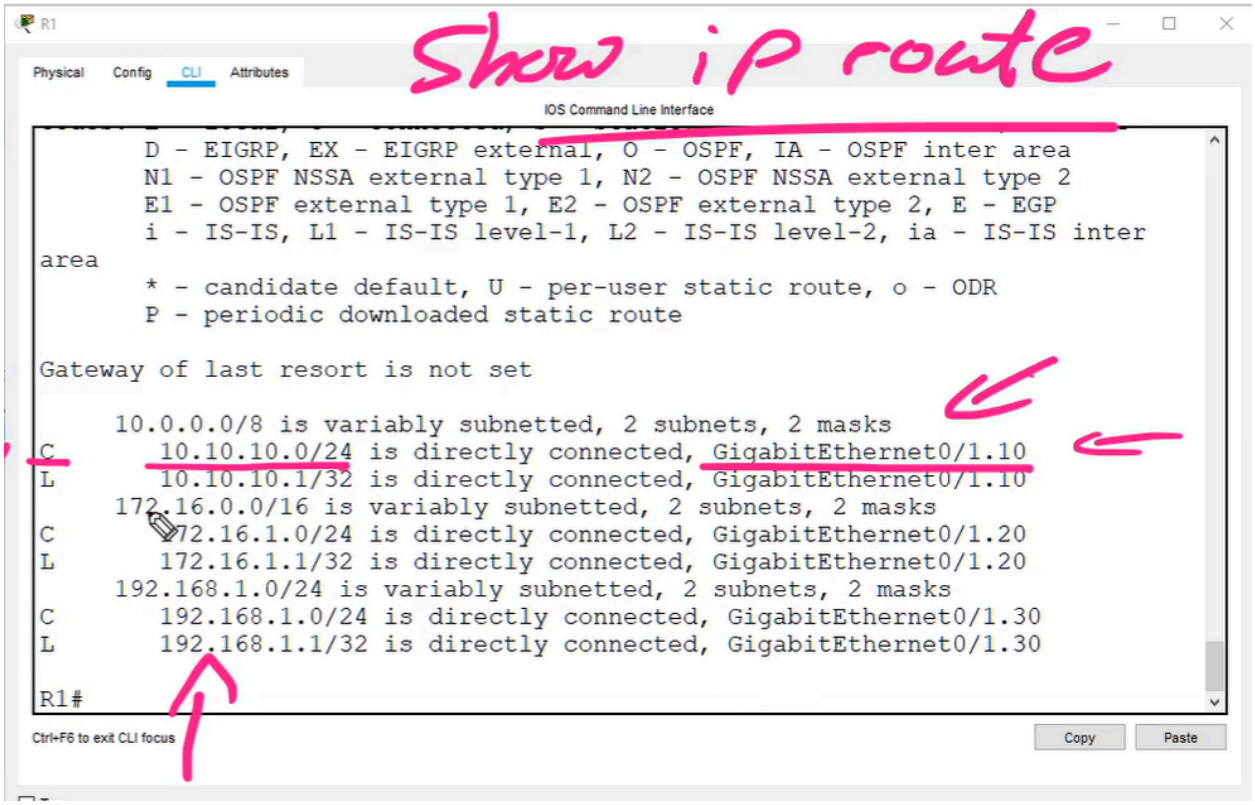
```
Pinging 172.16.1.2 with 32 bytes of data:

Reply from 172.16.1.2: bytes=32 time<1ms TTL=127
Reply from 172.16.1.2: bytes=32 time<1ms TTL=127
Reply from 172.16.1.2: bytes=32 time<1ms TTL=127
Reply from 172.16.1.2: bytes=32 time<1ms TTL=127

Ping statistics for 172.16.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

- **Now we can see the routing table with the shown below command .**



🛠️ Troubleshooting Guidelines for Inter-VLAN Routing

If devices in different VLANs still cannot communicate, consider these troubleshooting steps:

1. **Check IP Address Configurations** 📍
   Ensure each device has the correct IP address, subnet mask, and default gateway corresponding to its VLAN.
2. **Perform Ping and Traceroute Tests** 🔍
   - Test connectivity between VLANs using `ping` or `traceroute`. For example, from a PC in VLAN 10, try pinging a PC in VLAN 20.
3. **Verify Switch Port Assignments** 🔄
   Ensure all switch ports are correctly assigned to the intended VLANs, and that trunk ports are properly configured to allow VLAN traffic.
4. **Check for Native VLAN Mismatch** ⚠️
   Confirm that the native VLAN setting is consistent on both the router and the switch to avoid VLAN mismatch errors.
5. **Check Allowed VLANs on the Trunk Interface** ✅
   Verify that the trunk port allows the required VLANs. Use the `switchport trunk allowed vlan` command to confirm.
6. **Ensure Trunk Mode Consistency** 🔧
   Confirm that trunk mode is enabled on all required interfaces. Mode mismatch between switch and router may block traffic.

By following these guidelines, inter-VLAN routing should be successfully set up, allowing seamless communication between devices in different VLANs!