



1. Configure the switch interfaces connected to the pcs as access ports in the correct vlan

```
sw1#int range f0/3,f0/4
sw1#switch mode access
sw1#switch access vlan30
sw1#int range f0/2,f0/1
sw1#switch mode access
sw1#switch access vlan 10
sw1#do sh vlan br
```

VLAN	Name	Status	Ports
1	default	active	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 Gig0/2
10	VLAN0010	active	Fa0/1, Fa0/2
30	VLAN0030	active	Fa0/3, Fa0/4

2. Configure the connection between sw1 and sw2 as a trunk only allowing necessary vlans configure the native vlan

```
sw1#int g0/1
sw1#sw mode trunk
sw1#sw trunk allowed vlan 10,30
sw1#sw trunk native vlan 1001
```

```
Sw2#int g0/1
```

```

Sw2#sw mode trunk
Sw2#sw trunk allowed vlan 10,30
Sw2#sw trunk native vlan 1001
Sw2#do sh int trunk

```

remember we don't include vlan 20 in the trunk because it's going to go directly to the router to be retranslated into the new vlan it's going to

```
Sw2#vlan 30
```

```

Sw2#int g0/2
Sw2#sw mode trunk
Sw2#sw trunk allowed vlan 10,20,30
Sw2#sw trunk native vlan 1001

```

```

SW2(config)#do sh int trunk
Port      Mode      Encapsulation  Status      Native vlan
Gig0/1    on        802.1q         trunking    1001
Gig0/2    on        802.1q         trunking    1001

Port      Vlans allowed on trunk
Gig0/1    10,30
Gig0/2    10,20,30

Port      Vlans allowed and active in management domain
Gig0/1    10,30
Gig0/2    10,20,30

Port      Vlans in spanning tree forwarding state and not pruned
Gig0/1    10,30
Gig0/2    10,20,30

```

```
Sw2#do sh vlan br
```

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
10	VLAN0010	active	Fa0/2, Fa0/3
20	VLAN0020	active	Fa0/1
30	VLAN0030	active	
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

3. Connect sw2 and r1 using router on a stick technology

```
R1#int g0/0.10
R1#encap dot1q 10
R1#ip address 10.0.0.62 255.255.255.192
R1#int g0/0.20
R1#encap dot1q 20
R1#ip address 10.0.0.126 255.255.255.192
R1#int g0/0.30
R1#encap dot1q 30
R1#ip address 10.0.0.190 255.255.255.192
R1#int g0/0
R1#no shut
```

4. Test the connectivity by pining between pcs

Command Prompt

```
C:\>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 10.0.0.65

Pinging 10.0.0.65 with 32 bytes of data:

Request timed out.
Reply from 10.0.0.65: bytes=32 time<1ms TTL=127
Reply from 10.0.0.65: bytes=32 time<1ms TTL=127
Reply from 10.0.0.65: bytes=32 time<1ms TTL=127

Ping statistics for 10.0.0.65:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 10.0.0.129

Pinging 10.0.0.129 with 32 bytes of data:

Request timed out.
Reply from 10.0.0.129: bytes=32 time<1ms TTL=127
Reply from 10.0.0.129: bytes=32 time<1ms TTL=127
Reply from 10.0.0.129: bytes=32 time<1ms TTL=127

Ping statistics for 10.0.0.129:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```