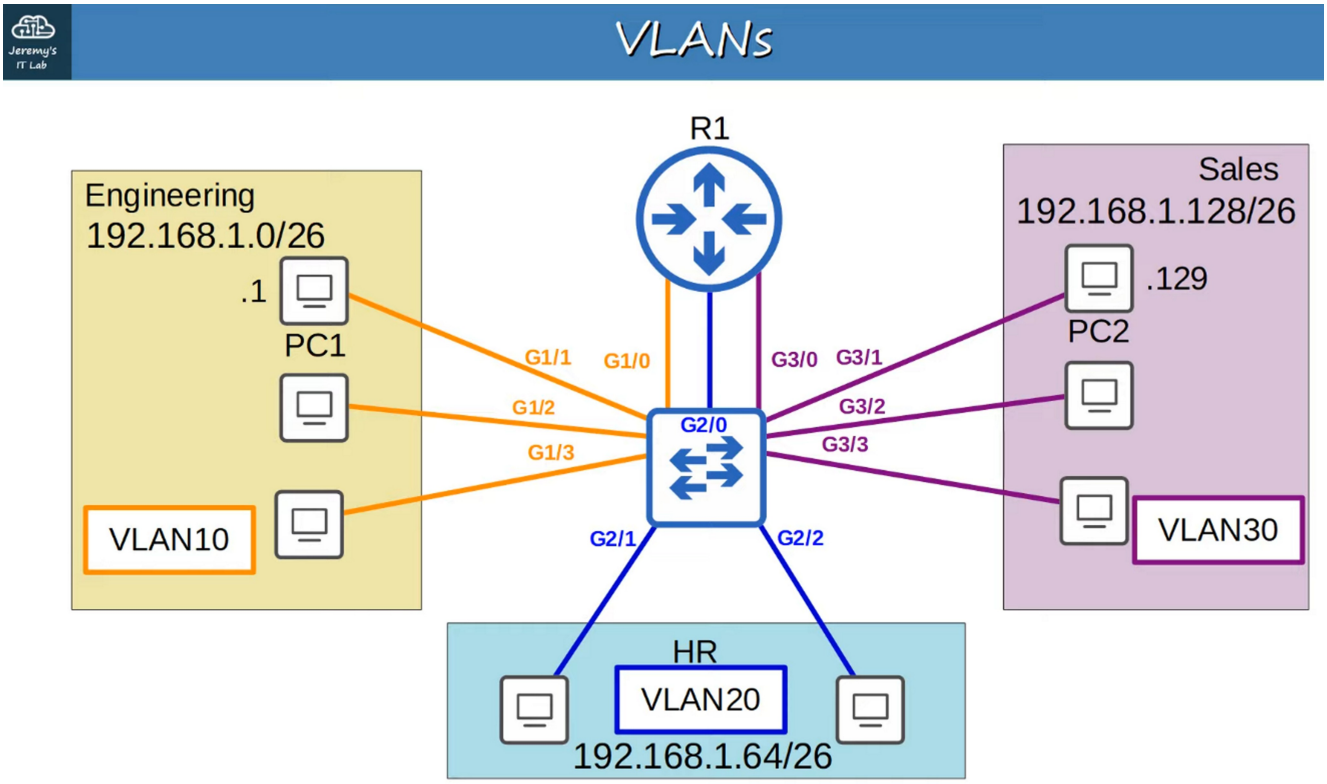
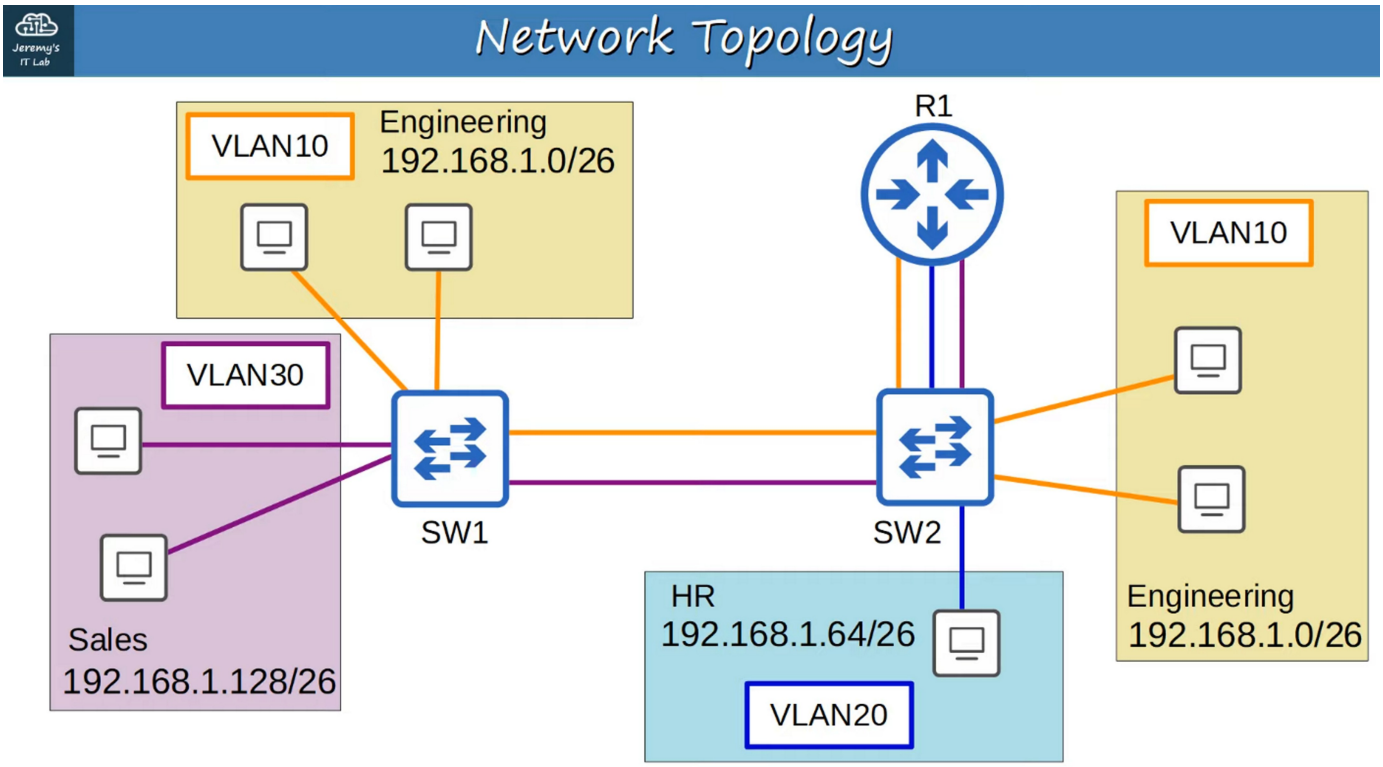
Thursday, July 31, 2025 4:12 PM

LANS : PART 2

Basic VLAN topology from PART 1





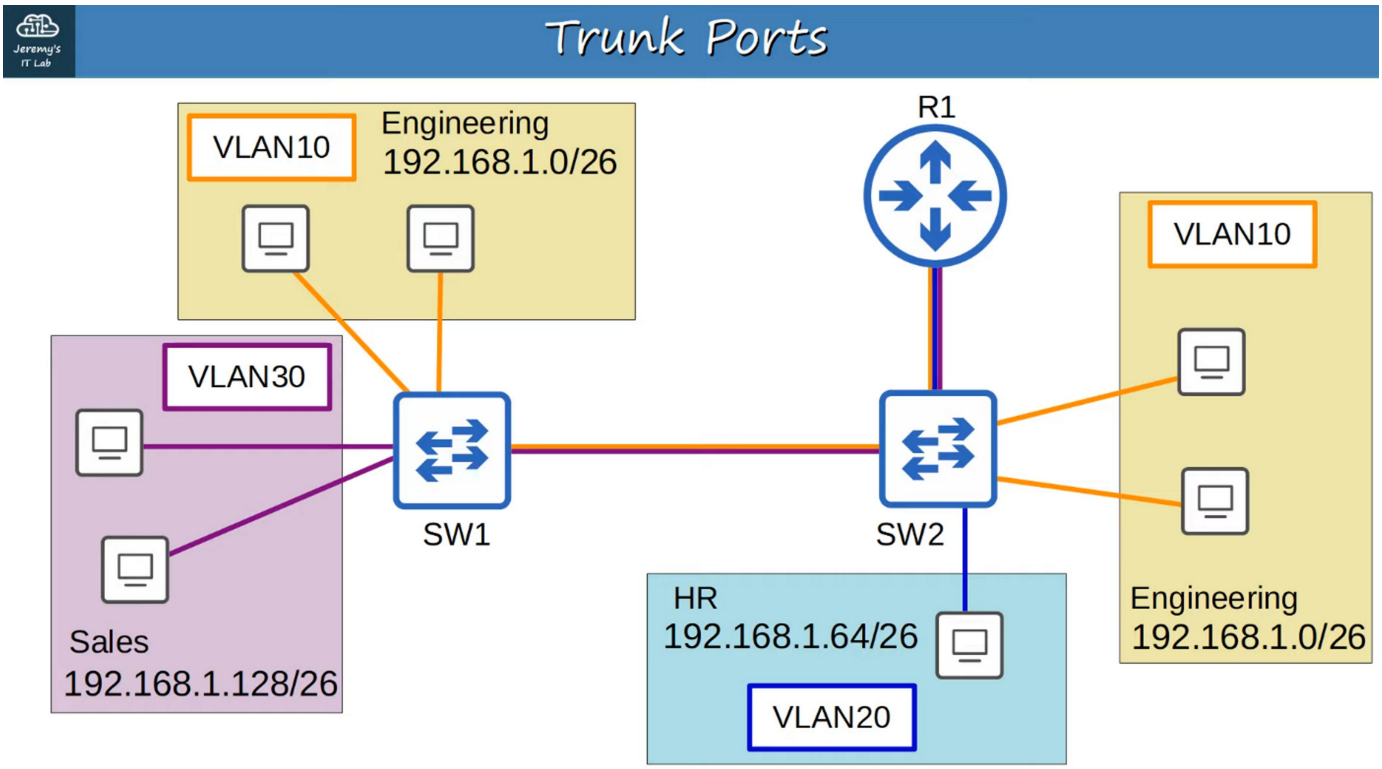
Notice this one has TWO Switches (SW1 and SW2) and ENGINEERING (VLAN 10) has two

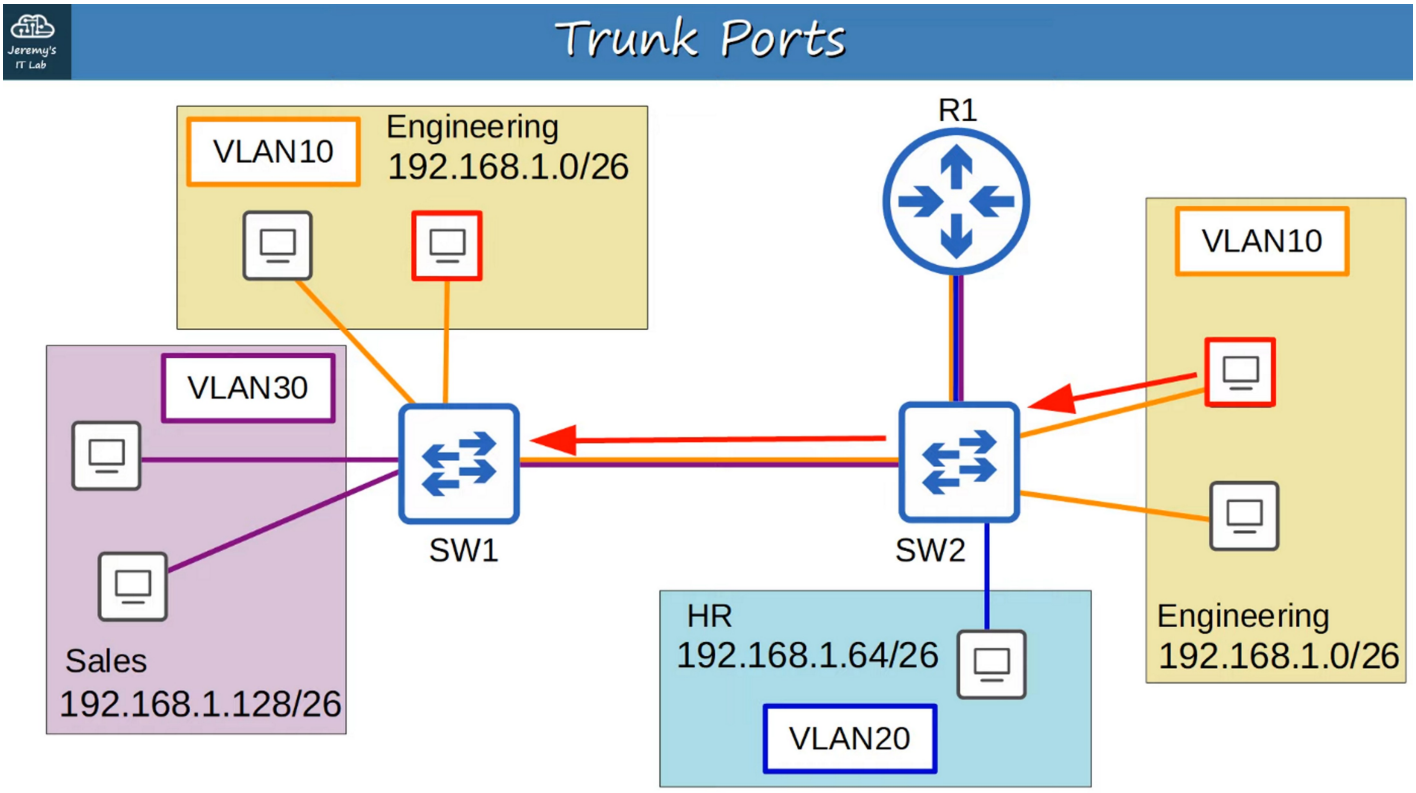
separate locations on the network.

TRUNK PORTS

- In a small network with few VLANS, it's possible to use a separate interface for EACH VLAN when connecting SWITCHES to SWITCHES, and SWITCHES to ROUTERS
- HOWEVER, when the number of VLANS increases, this is not viable. It will result in wasted interfaces, and often ROUTERS won't have enough INTERFACES for each VLAN
- You can use TRUNK PORTS to carry traffic from multiple VLANS over a single interface

A TRUNK PORT carrying multiple VLAN connections over single interface





How does a packet know WHICH VLAN to send traffic to over the TRUNK PORT ?

VLAN TAGS !

SWITCHES will “tag” all frames that they send over a TRUNK LINK. This allows the receiving SWITCH to know which VLAN the frame belongs to.

TRUNK PORT = “Tagged” ports

ACCESS PORT = “Untagged” ports

VLAN TAGGING

- There are TWO main TRUNK protocols:
  - ISL (Inter-Switch Link)
  - IEEE 802.1Q (also known as “dot1q”)

ISL is an old Cisco proprietary protocol created before industry standard IEEE 802.1Q

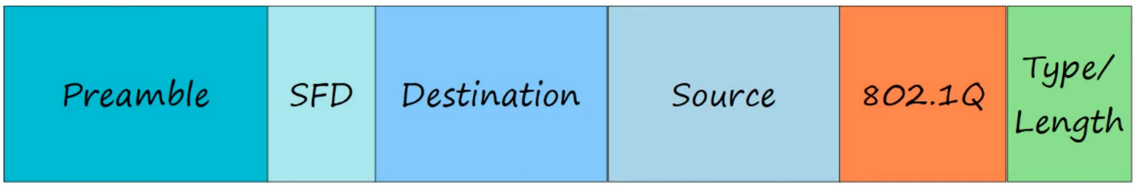
IEEE 802.1Q is an industry standard protocol created by the IEEE (Institute of Electrical and Electronics Engineers)

You will probably NEVER use ISL in the real world; even modern Cisco equipment doesn’t use it.

For the CCNA, you will only need to learn 802.1Q

ETHERNET HEADER with 802.1Q

Ethernet Header w/ 802.1Q Tag



- The 802.1Q TAG Is inserted between the SOURCE and TYPE/LENGTH fields in the ETHERNET FRAME
- The TAG is 4 bytes (32 bits) in length
- The TAG consists of TWO main fields:
  - Tag Protocol Identifier (TPID)
  - Tag Control Information (TCI)
    - TCI consists of THREE sub-fields:

802.1Q Tag

802.1Q tag format

16 bits	3 bits	1 bit	12 bits
TPID	TCI		
	PCP	DEI	VID

TPID (TAG Protocol Identifier) :

- 16 bits (2 bytes) in length
- Always set to a value of 0x8100. This indicates that the frame is 802.1Q TAG

TCI / PCP (Priority Code Point) :

- 3 bits in length
- Used for Class of Service (CoS), which prioritizes important traffic in congested networks

TCI / DEI (Drop Eligible Indicator) :

- 1 bit in length
- Used to indicated frames that can be dropped if the network is congested

TCI / VID (VLAN ID) :

- 12 bits in length
- Identifies the VLAN the frame belongs to

- 12 bits in length = 4096 total VLANS (2^12), range of 0 - 4095
- VLANs 0 and 4095 are reserved and can't be used
- Therefore, the actual range of VLANs is 1 - 4094

NOTE : Cisco's ISL also had a VLAN range of 1 - 4094

VLAN RANGES

- The range of VLANs (1 – 4094) is divided into two sections:
  Normal VLANs: 1 – 1005
  Extended VLANs: 1006 – 4094

- Some older devices cannot use the extended VLAN range, however it's safe to expect that modern switches will support the extended VLAN range.

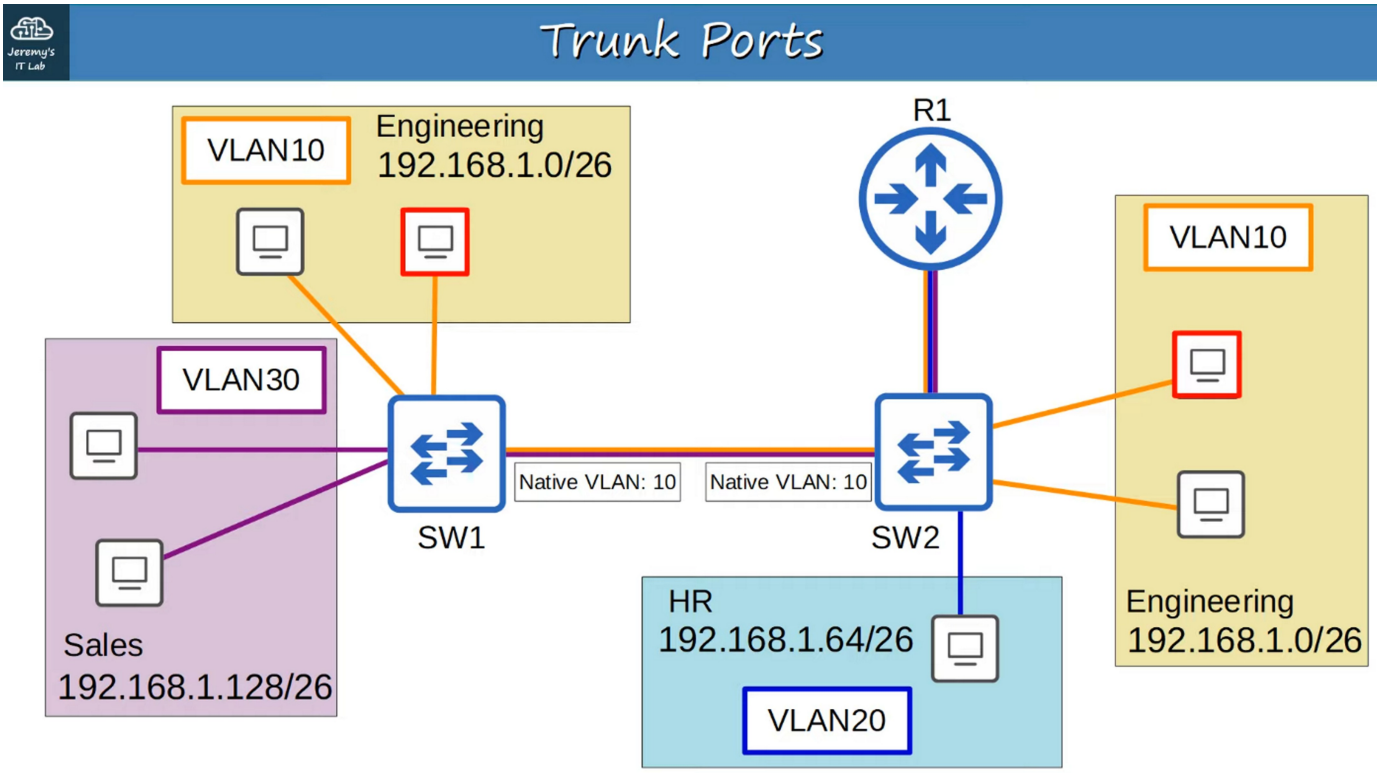
NATIVE VLAN

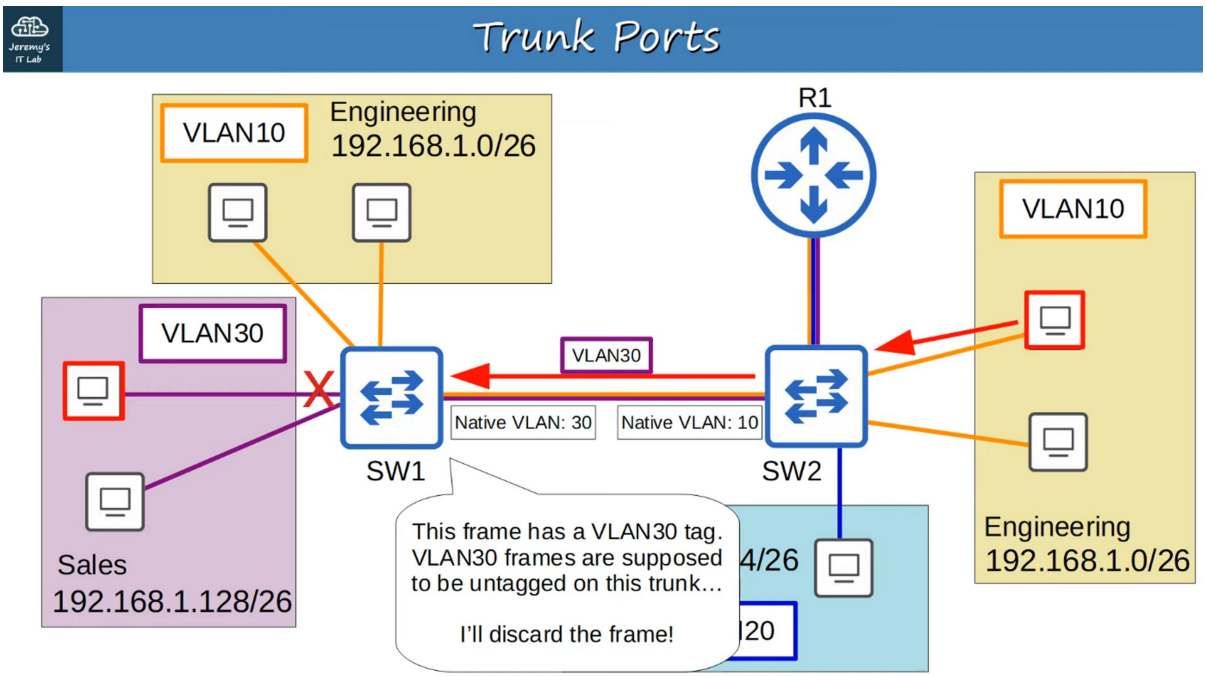
- 802.1Q has a feature called the **native VLAN**.
  (ISL does not have this feature)

- The native VLAN is VLAN 1 by default on all trunk ports, however this can be manually configured on each trunk port.

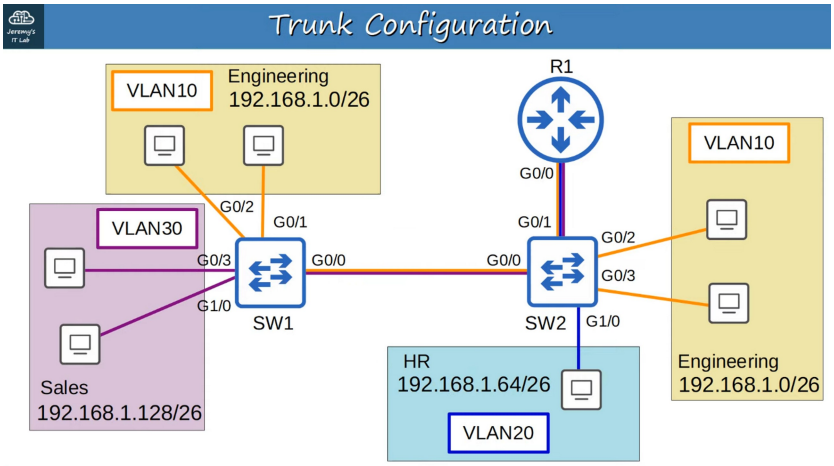
- The switch does not add an 802.1Q tag to frames in the native VLAN.

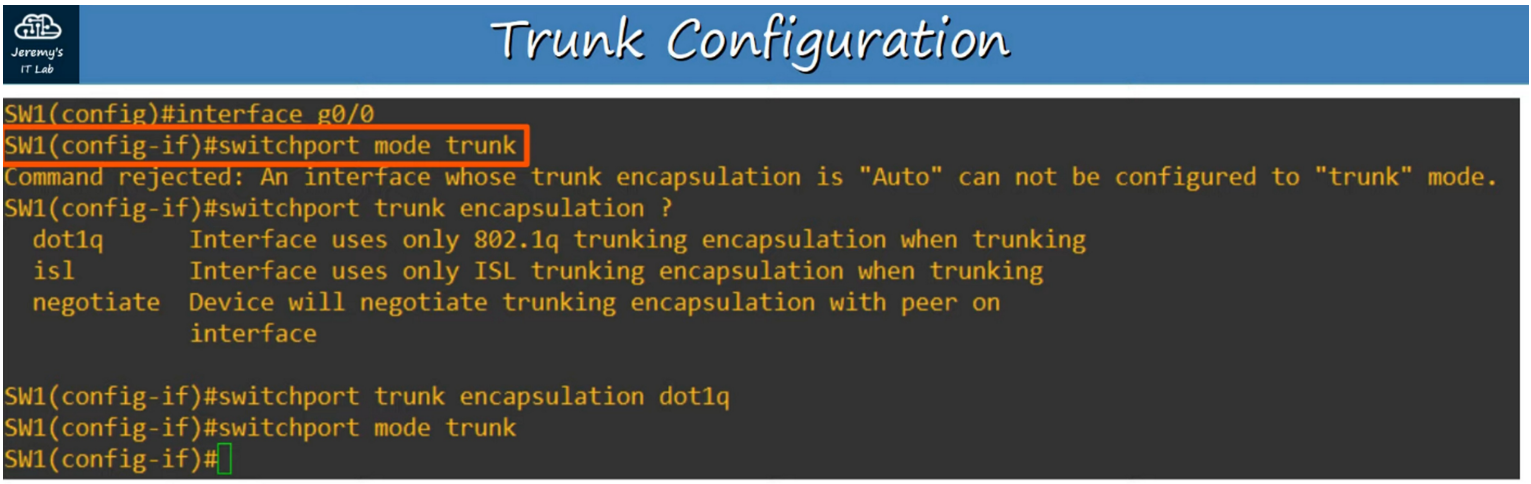
- When a switch receives an untagged frame on a trunk port, it assumes the frame belongs to the native VLAN.
  It's very important that the native VLAN matches!





TRUNK CONFIGURATION





Many modern switches do not support Cisco's ISL at all. They only support 802.1Q (dot1q)

However, SWITCHES that do support both (like the one I am using in this example) have a TRUNK encapsulation of "AUTO" by default

To MANUALLY configure the INTERFACE as a TRUNK PORT, you must first set the encapsulation to "802.1Q" or "ISL". On SWITCHES that only support 802.1Q, this is not necessary

After you set the encapsulation type, you can then configure the interface as a TRUNK

1. Select the interface to configure
2. Use "#switchport trunk encapsulation dot1q" to set the encapsulation mode to 802.1Q
3. Use "#switchport mode trunk" to manually configure the interface to TRUNK

Trunk Configuration

```
SW1#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Gi0/0       1-4094

Port        Vlans allowed and active in management domain
Gi0/0       1,10,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       1,10,30
SW1#
```

Use the “#show interfaces trunk” command to confirm INTERFACES on TRUNK

Trunk Configuration

```
SW1#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Gi0/0       1-4094

Port        Vlans allowed and active in management domain
Gi0/0       1,10,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       1,10,30
SW1#
```

```
SW1#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Gi1/1, Gi1/2, Gi1/3, Gi2/0
                                                Gi2/1, Gi2/2, Gi2/3, Gi3/0
                                                Gi3/1, Gi3/2, Gi3/3
10   ENGINEERING                      active    Gi0/1, Gi0/2
30   SALES                            active    Gi0/3, Gi1/0
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
SW1#
```

Commands to allow a VLAN on a given TRUNK

Trunk Configuration

```
SW1(config)#int g0/0
SW1(config-if)#
SW1(config-if)#switchport trunk allowed vlan ?
  WORD    VLAN IDs of the allowed VLANs when this port is in trunking mode
  add     add VLANs to the current list
  all     all VLANs
  except  all VLANs except the following
  none    no VLANs
  remove  remove VLANs from the current list

SW1(config-if)#switchport trunk allowed vlan
```

Trunk Configuration

```
SW1(config-if)#switchport trunk allowed vlan 10,30
SW1(config-if)#do show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Gi0/0       10,30

Port        Vlans allowed and active in management domain
Gi0/0       10,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10,30
SW1(config-if)#
```

For security purposes, it is best to change the native VLAN to an **unused VLAN**.
(network security will be explained more in-depth later in the course)
Make sure the native VLAN matches on between switches

Command to change the NATIVE VLAN

Trunk Configuration

```
SW1(config-if)#switchport trunk native vlan 1001
SW1(config-if)#do show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1001

Port        Vlans allowed on trunk
Gi0/0       10,30

Port        Vlans allowed and active in management domain
Gi0/0       10,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10,30
SW1(config-if)#
```

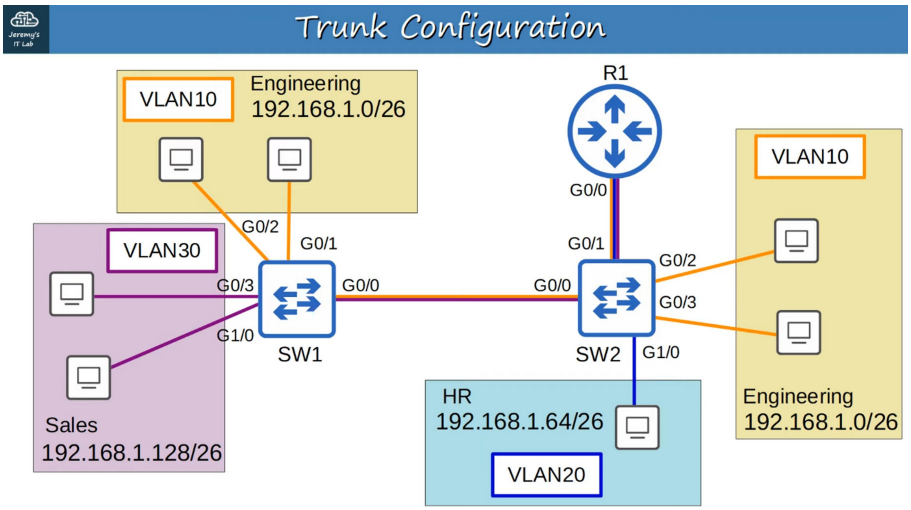
Trunk Configuration

```
SW1#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Gi1/1, Gi1/2, Gi1/3, Gi2/0
                                                Gi2/1, Gi2/2, Gi2/3, Gi3/0
                                                Gi3/1, Gi3/2, Gi3/3
10   ENGINEERING                      active    Gi0/1, Gi0/2
30   SALES                            active    Gi0/3, Gi1/0
1002 fddi-default                     act/unsup
```

The **show vlan brief** command shows the access ports assigned to each VLAN, NOT the trunk ports that allow each VLAN.

Use the **show interfaces trunk** command instead to confirm trunk ports.

Setting up our TRUNKS for this Network

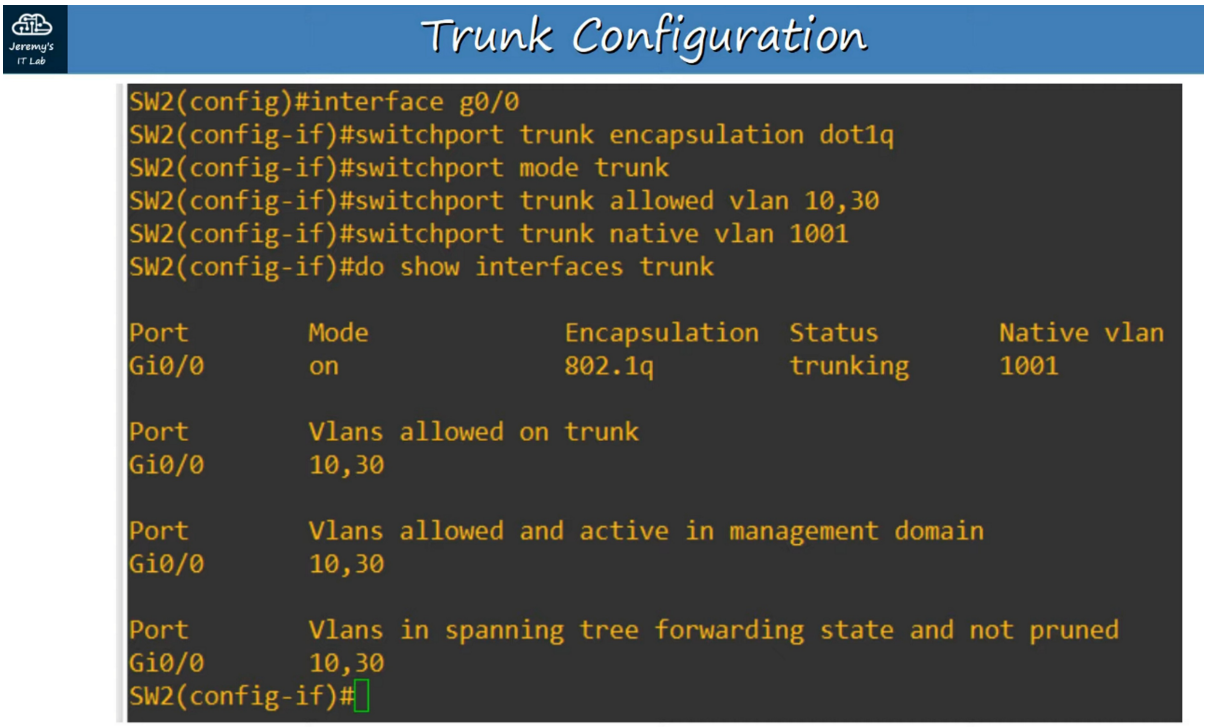


We will need to configure :

SW1 : g0/0 interface (already configure above this section)

SW2: g0/0, and g0/1 interface

SW2 g0/0

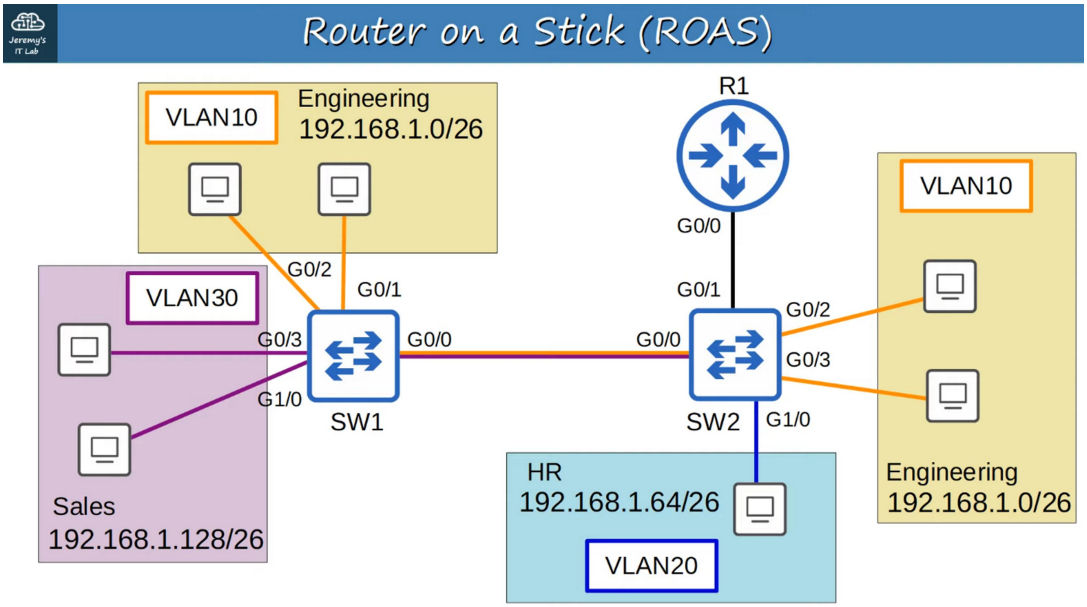


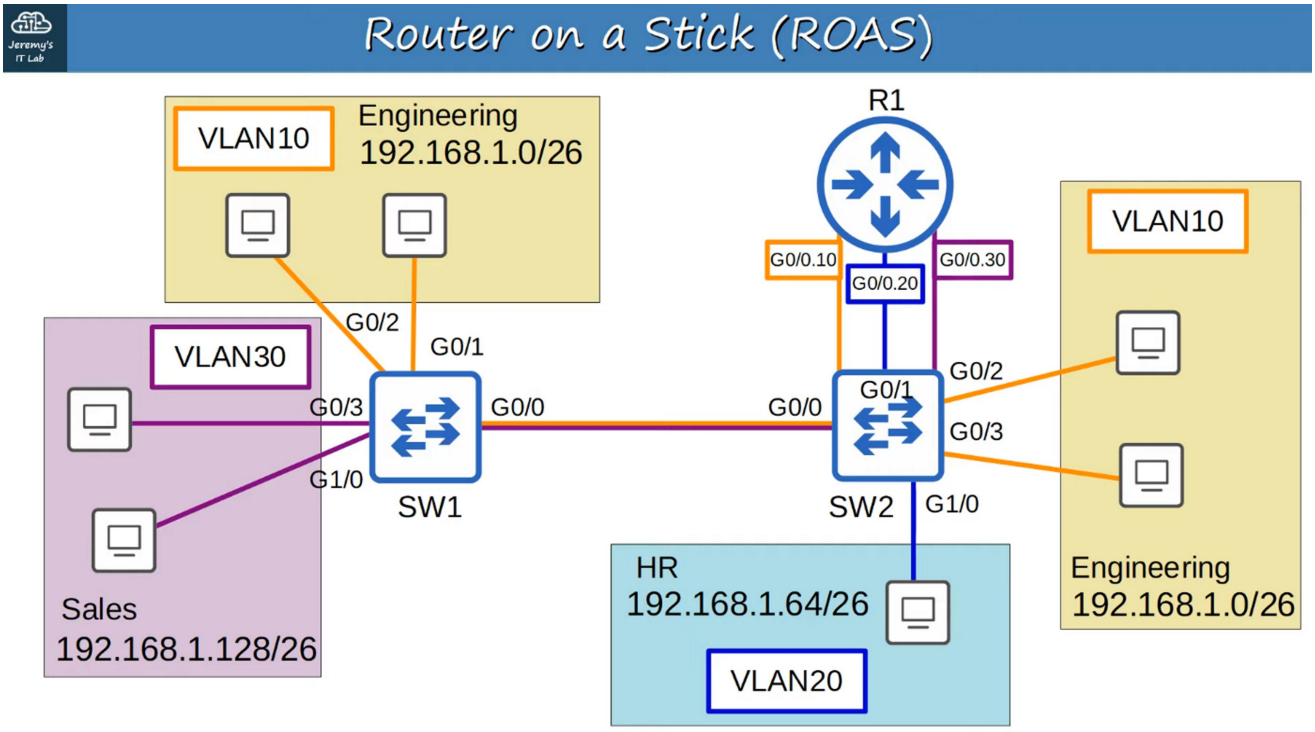
SW2 g0/1

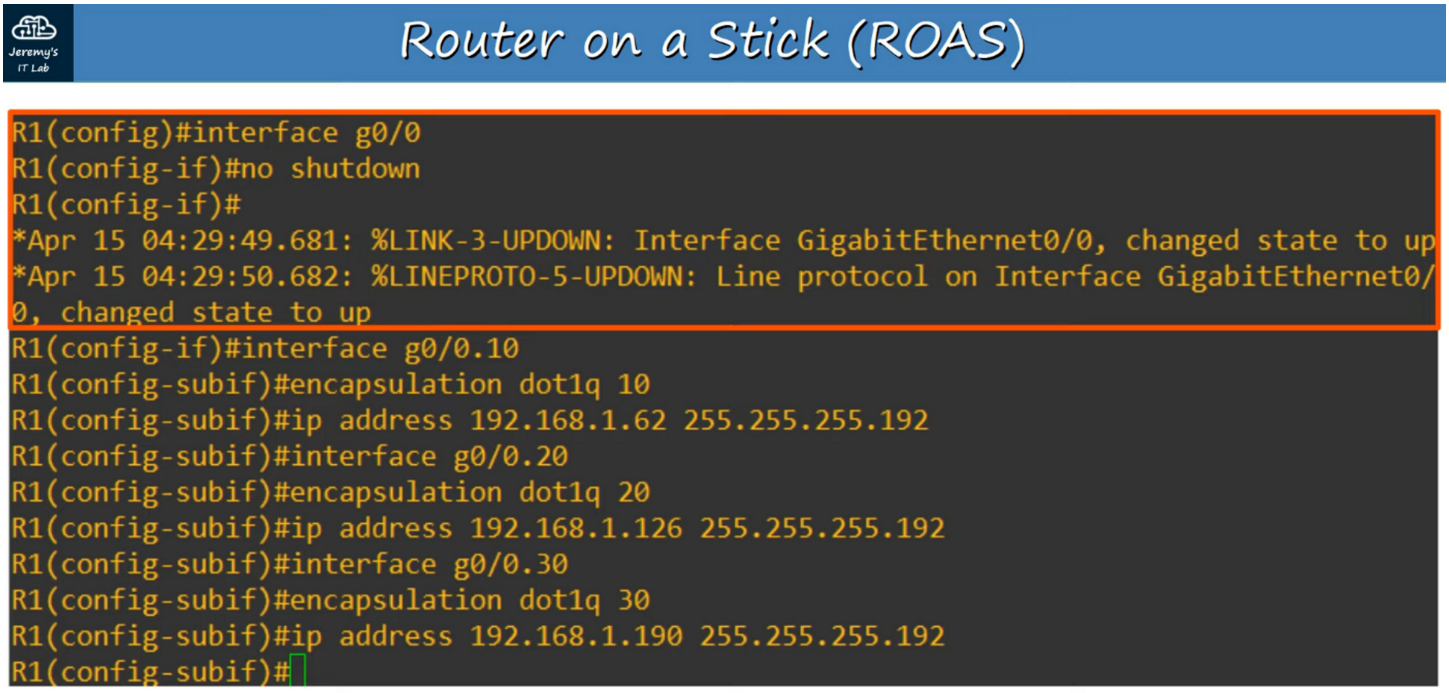
Trunk Configuration

```
SW2(config)#interface g0/1
SW2(config-if)#switchport trunk encapsulation dot1q
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk allowed vlan 10,20,30
SW2(config-if)#switchport trunk native vlan 1001
SW2(config-if)#do show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1001
Gi0/1       on               802.1q         trunking      1001

Port        Vlans allowed on trunk
Gi0/0       10,30
Gi0/1       10,20,30

Port        Vlans allowed and active in management domain
Gi0/0       10,30
Gi0/1       10,20,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10,30
Gi0/1       none
SW2(config-if)#
```

ROUTER ON A STICK (ROAS)







NOTE the Sub-Interface names (like the network diagram) of 0.10, 0.20 and 0.30

You assign them IP addresses identically like you would a regular interface (using the last usable IP address of a given VLAN subnet)

Sub-interfaces will appear with the “show ip interface brief” command


```
R1#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         unassigned      YES NVRAM  up                    up
GigabitEthernet0/0.10      192.168.1.62    YES manual up                    up
GigabitEthernet0/0.20      192.168.1.126   YES manual up                    up
GigabitEthernet0/0.30      192.168.1.190   YES manual up                    up
GigabitEthernet0/1         unassigned      YES NVRAM  administratively down down
GigabitEthernet0/2         unassigned      YES NVRAM  administratively down down
GigabitEthernet0/3         unassigned      YES NVRAM  administratively down down
```

They also appear in the “show ip route” command (Route Table)


```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 6 subnets, 2 masks
C        192.168.1.0/26 is directly connected, GigabitEthernet0/0.10
L        192.168.1.62/32 is directly connected, GigabitEthernet0/0.10
C        192.168.1.64/26 is directly connected, GigabitEthernet0/0.20
L        192.168.1.126/32 is directly connected, GigabitEthernet0/0.20
C        192.168.1.128/26 is directly connected, GigabitEthernet0/0.30
L        192.168.1.190/32 is directly connected, GigabitEthernet0/0.30
R1#
```

set arp.spoof.targets 192.168.1.5Vlan

ROAS is used to route between multiple VLANs using a SINGLE interface on a ROUTER and SWITCH

The SWITCH interface is configured as a regular TRUNK

The ROUTER interface is configured using SUB-INTERFACES. You configure the VLAN tag and IP address on EACH SUB-INTERFACE

The ROUTER will behave as if frames arriving with a certain VLAN tag have arrived on the SUB-INTERFACE configured with that VLAN tag

The ROUTER will TAG frames sent out of EACH SUB-INTERFACE with the VLAN TAG configured on the SUB-INTERFACE