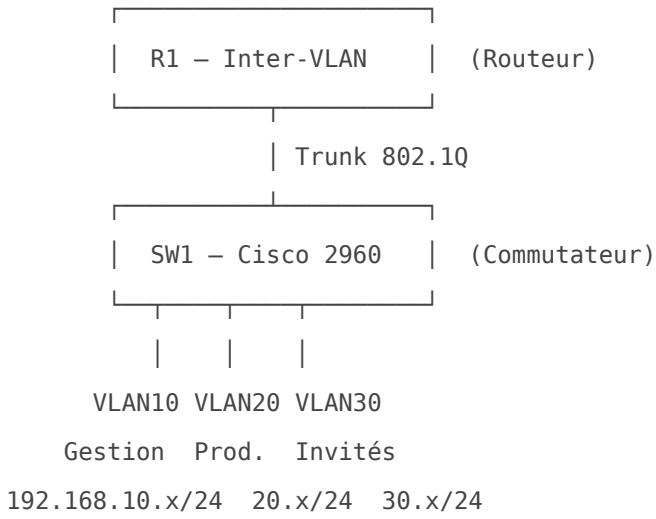
VLAN

- Créer et faire fonctionner un VLAN

Créer et faire fonctionner un VLAN

> **Standard :** IEEE 802.1Q · **Plateformes :** Cisco IOS, Linux · **Niveau :** Intermédiaire---

Topologie de référence



---

01 — Comprendre les VLANs

Un **VLAN (Virtual Local Area Network)** est une segmentation logique d'un réseau physique. Il permet d'isoler des groupes de machines comme s'ils étaient sur des réseaux séparés, même s'ils partagent le même commutateur.

Avantages

Aspect	Description
Sécurité	Les machines de VLANs différents ne communiquent pas sans routage explicite

Aspect	Description
⚡ **Performance**	Réduit les domaines de broadcast, limitant les tempêtes et le trafic inutile
◫ **Flexibilité**	Regroupez des machines par fonction (RH, IT, Invités) indépendamment de leur localisation physique
◫ **Standard**	IEEE 802.1Q ajoute un tag de 4 octets dans la trame Ethernet — jusqu'à **4094 VLANs** distincts

Types de ports

Type	Description	Usage typique
Access	Appartient à un seul VLAN. Connecte les hôtes finaux (PC, serveurs)	Ports vers postes de travail
Trunk	Transporte plusieurs VLANs avec tags 802.1Q. Connecte switches entre eux ou un routeur	Liaison switch-switch, switch-routeur
Native	Trafic non tagué sur un trunk. VLAN 1 par défaut (à changer !)	VLAN de gestion dédié

> **i Note :** Le tag 802.1Q contient un **VLAN ID (VID)** de 12 bits, permettant jusqu'à **4094 VLANs** distincts (1–4094, 0 et 4095 étant réservés).

---

02 — Configuration sur Cisco IOS

Créer les VLANs

```
! Entrer en mode configuration globale
SW1# enable
SW1# configure terminal

! Créer le VLAN 10 — Gestion
SW1(config)# vlan 10
SW1(config-vlan)# name Gestion
SW1(config-vlan)# exit
```

```
! Créer le VLAN 20 – Production
SW1(config)# vlan 20
SW1(config-vlan)# name Production
SW1(config-vlan)# exit

! Créer le VLAN 30 – Invités
SW1(config)# vlan 30
SW1(config-vlan)# name Invites
SW1(config-vlan)# exit
```

Assigner des ports Access

```
! Assigner Fa0/1 et Fa0/2 au VLAN 10 (Gestion)
SW1(config)# interface range FastEthernet0/1-2
SW1(config-if-range)# switchport mode access
SW1(config-if-range)# switchport access vlan 10
SW1(config-if-range)# exit

! Assigner Fa0/3 et Fa0/4 au VLAN 20 (Production)
SW1(config)# interface range FastEthernet0/3-4
SW1(config-if-range)# switchport mode access
SW1(config-if-range)# switchport access vlan 20
SW1(config-if-range)# exit

! Assigner Fa0/5 au VLAN 30 (Invités)
SW1(config)# interface FastEthernet0/5
SW1(config-if)# switchport mode access
SW1(config-if)# switchport access vlan 30
SW1(config-if)# exit
```

Configurer un port Trunk

```
! Configurer Fa0/24 comme trunk vers le routeur
SW1(config)# interface FastEthernet0/24
SW1(config-if)# switchport trunk encapsulation dot1q
SW1(config-if)# switchport mode trunk

! Autoriser uniquement nos VLANs (bonne pratique)
SW1(config-if)# switchport trunk allowed vlan 10,20,30
```

```
! Changer le VLAN natif (ne jamais laisser le VLAN 1)
SW1(config-if)# switchport trunk native vlan 99
SW1(config-if)# exit

! Sauvegarder la configuration
SW1# write memory
```

> ⚠ **Attention :** Toujours changer le **VLAN natif** depuis le VLAN 1 pour éviter les attaques de type VLAN hopping.

Commandes de vérification

```
! Voir tous les VLANs et leurs ports
SW1# show vlan brief

! Voir le statut des trunks
SW1# show interfaces trunk

! Voir la config d'une interface
SW1# show interfaces FastEthernet0/1 switchport

! Voir les VLANs en base de données
SW1# show vlan id 10
```

---

03 — Configuration sur Linux

Sur Linux, les VLANs sont créés via le module **8021q** du noyau.

ip / iproute2 (temporaire)

```
# Charger le module VLAN
sudo modprobe 8021q

# Créer les interfaces VLAN sur eth0
```

```
sudo ip link add link eth0 name eth0.10 type vlan id 10
sudo ip link add link eth0 name eth0.20 type vlan id 20
sudo ip link add link eth0 name eth0.30 type vlan id 30

# Activer les interfaces
sudo ip link set eth0.10 up
sudo ip link set eth0.20 up
sudo ip link set eth0.30 up

# Assigner les adresses IP
sudo ip addr add 192.168.10.1/24 dev eth0.10
sudo ip addr add 192.168.20.1/24 dev eth0.20
sudo ip addr add 192.168.30.1/24 dev eth0.30

# Vérifier
ip -d link show eth0.10
ip addr show
```

Netplan (Ubuntu — persistant)

Créer le fichier `/etc/netplan/01-vlan.yaml` :

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      dhcp4: false
  vlans:
    eth0.10:
      id: 10
      link: eth0
      addresses: [192.168.10.1/24]
    eth0.20:
      id: 20
      link: eth0
      addresses: [192.168.20.1/24]
    eth0.30:
      id: 30
      link: eth0
```

```
      addresses: [192.168.30.1/24]
```

Puis appliquer :

```
sudo netplan apply
```

NetworkManager / nmcli

```
# Créer les connexions VLAN
nmcli connection add type vlan con-name vlan10 \
  dev eth0.10 vlan.parent eth0 vlan.id 10 \
  ipv4.addresses 192.168.10.1/24 ipv4.method manual

nmcli connection add type vlan con-name vlan20 \
  dev eth0.20 vlan.parent eth0 vlan.id 20 \
  ipv4.addresses 192.168.20.1/24 ipv4.method manual

# Activer
nmcli connection up vlan10
nmcli connection up vlan20
```

---

04 — Routage inter-VLAN (Router-on-a-Stick)

Par défaut, les VLANs sont isolés. Pour leur permettre de communiquer, on utilise un routeur avec des **sous-interfaces** — technique dite "router-on-a-stick".

```
         R1
          |  (GigabitEthernet0/0 — trunk)
         SW1
        / | \
    V10  V20  V30
```

Configuration du routeur R1

```
! Activer l'interface physique sans IP
R1(config)# interface GigabitEthernet0/0
R1(config-if)# no shutdown
R1(config-if)# exit

! Sous-interface pour VLAN 10
R1(config)# interface GigabitEthernet0/0.10
R1(config-subif)# encapsulation dot1Q 10
R1(config-subif)# ip address 192.168.10.254 255.255.255.0
R1(config-subif)# exit

! Sous-interface pour VLAN 20
R1(config)# interface GigabitEthernet0/0.20
R1(config-subif)# encapsulation dot1Q 20
R1(config-subif)# ip address 192.168.20.254 255.255.255.0
R1(config-subif)# exit

! Sous-interface pour VLAN 30
R1(config)# interface GigabitEthernet0/0.30
R1(config-subif)# encapsulation dot1Q 30
R1(config-subif)# ip address 192.168.30.254 255.255.255.0
R1(config-subif)# exit

! Vérifier le routage
R1# show ip route
```

> □□ **Astuce :** Configurez les **passerelles par défaut** de vos hôtes vers l'IP du routeur sur leur VLAN (ex: `192.168.10.254` pour les hôtes du VLAN 10).

---

05 — Bonnes pratiques & Sécurité

- □ **Désactiver le VLAN 1** pour le trafic utilisateur — réserver le VLAN 1 uniquement au trafic interne du switch.
- □ **Changer le VLAN natif** sur tous les trunks vers un VLAN dédié (ex: VLAN 99) pour prévenir le VLAN hopping.
- □ **Limiter les VLANs autorisés** sur les trunks avec `switchport trunk allowed vlan` — ne jamais laisser "all".

- 🔒 **Utiliser Port Security** (`switchport port-security`) sur les ports access pour limiter les adresses MAC.
- ⚠ **Documenter votre plan d'adressage** VLAN dès le départ — un tableau ID/Nom/Sous-réseau/Rôle évite les conflits.
- 🌳 **Activer STP (Spanning Tree)** pour éviter les boucles en cas de liens redondants entre switches.
- 💾 **Sauvegarder la config** après chaque modification avec `write memory` ou `copy run start` sur Cisco.

> ⚠ **VLAN Hopping :** Si le VLAN natif d'un trunk correspond au VLAN d'un port access, un attaquant peut injecter du trafic double-tagué pour sauter entre VLANs. Toujours utiliser un VLAN natif dédié et non utilisé.

---

06 — Dépannage rapide

Symptôme	Cause probable	Solution
Deux hôtes du même VLAN ne se pingent pas	Port mal assigné au VLAN	`show vlan brief` → vérifier l'assignation du port
Trafic inter-VLAN impossible	Sous-interface manquante ou trunk mal configuré	Vérifier `show interfaces trunk` et les subinterfaces
VLAN absent du trunk	VLAN non autorisé sur le trunk	`switchport trunk allowed vlan add X`
Interface VLAN Linux inactive	Module 8021q non chargé	`sudo modprobe 8021q`
VLAN créé mais non actif	VLAN en état "suspended"	`show vlan id X` → vérifier l'état

---

IEEE 802.1Q · Cisco IOS / Linux · NET/LAB