RAPPORT DU PROJET_M1_CNS_SR_2025

Groupe 5
Membre du groupe :
DIALLO Boubacar
ZHIPENG WANG

Encadrant :
Monsieur Mehdi Denou

Architecture du projet :

1- Etude du besoin
2-Analyse de l'existant
3.Cahier de charges
4.Conception & choix du matériel
5-Maquette & Tests
6-Installation & Tests
7-Recette & Validation

1-ETUDE DU BESOIN

L'objectif de ce projet est de mettre en place un réseau fonctionnel avec :

- Isolation entre une machine 1 et une autre machine 2
- Créer un Accès Sécurisé au serveur NFS
- Mise en place d'une Architecture de Tolerance de Panne sur les commutateurs et le serveur NFS

Contraintes techniques :

- Utilisation de ansible recommandé pour l'automatisation (optionnel),
- Git pour le versionning des configurations
- Soutenance orale avec slides à prevoir
- utilisé une machine virtuelle (Vmware ou virtualbox)
- deux Commutateurs de niveau 2 et un autre du niveau 3 (Cumulus linux)

2-ANALYSE DE L'EXISTANT

État initial du réseau :

- 3 Commutateurs en Cumulus linux
- Machine 1 et Machine 2 potentiellement qui communiquent
- Un serveur NFS accessible par les deux machines.
- Pas de Redondance en cas de panne d'une interface .

Problème à résoudre :

- Isolation du réseau : Séparer la machine 1 et Machine 2
- Tolérance de panne : Si une interface des interface du serveur tombe en panne qu'on soit capable de maintenir la communication .
- Chaque machine doit pouvoir lire et écrire des fichiers sur son point de montage NFS
- Le trafic (ping/ssh) ne doit pas être possible entre machine1 et machine2.

3- CAHIER DES CHARGES

Architecture Réseau :

- Utilisation de VLANs pour l'isolation machine 1 et machine 2.
- Bonding (LACP) sur le serveur NFS pour la rédondance

Configuration des équipements :

- Equipement	Configuration requise
Switch1 & Switch2:	VLANs
Switch3:	Routage
Machine 1 & Machine 2	VLAN respectif
Serveur NFS	Bonding sur les deux interfaces Réseau

4- CONCEPTION & CHOIX DU MATÉRIEL

Topologie Réseau:

Nous allons créer 3 VLANs:
VLAN 10: Machine 1
VLAN 20: Machine 2
VLAN 30: Serveur NFS

Configuration des Commutateurs

* Switch1 et Switch2 (Niveau 2)

  . Port access pour la machine 1 et Machine 2
  . Trunk vers Switch3
* Switch 3 (niveau 3)

* Serveur NFS:
  . Bonding (LACP)
  . NFSV4

4- MAQUETTE & TEST

- **1 Machine Admin: qui contient ansible et les playbook :**
  interface d'admin 192.168.10.1

- **Machine 1:**

  interface Nat: pour la connexion
  192.168.10.2: interface d'admin
  Interface LAN SEGMENT 1:192.168.20.10/24

- **Machine 2:**

  interface Nat pour la connexion
  192.168.10.3 interface admin
  interface LAN SEGMENT2 192.168.30.10/24

- **Serveur NFS:**
  interface admin: 192.168.10.4
  interface NAT
  interface LAN SEGMENT:3 192.168.20.100/24
  interface LAN SEGMENT4: 192.168.30.100/24

- **Switch1:**

  interface admin 192.168.10.5
  interface sw1 LAN SEGMENT1 vers machine 1
  interface LAN SEGMENT3 vers Serveur NFS
  interface LAN SEGMENT4 vers Serveur NFS
  interface LAN SEGMENT 5 vers Switch3

  Switch3:
  interface admin: 192.168.10.6
  interface LAN SEGMENT5 vers Switch1
  interface LAN SEGMENT6 vers Switch2

  Switch2:
  interface admin: 192.168.10.7
  interface LAN SEGMENT6 vers Switch3
  interface LAN SEGMENT2 vers machine 2

Toutes les machines sont administrées par une seule machine (Machine A) et sont renommées dans le fichier hosts afin de faciliter la configuration en utilisant uniquement les noms des machines.

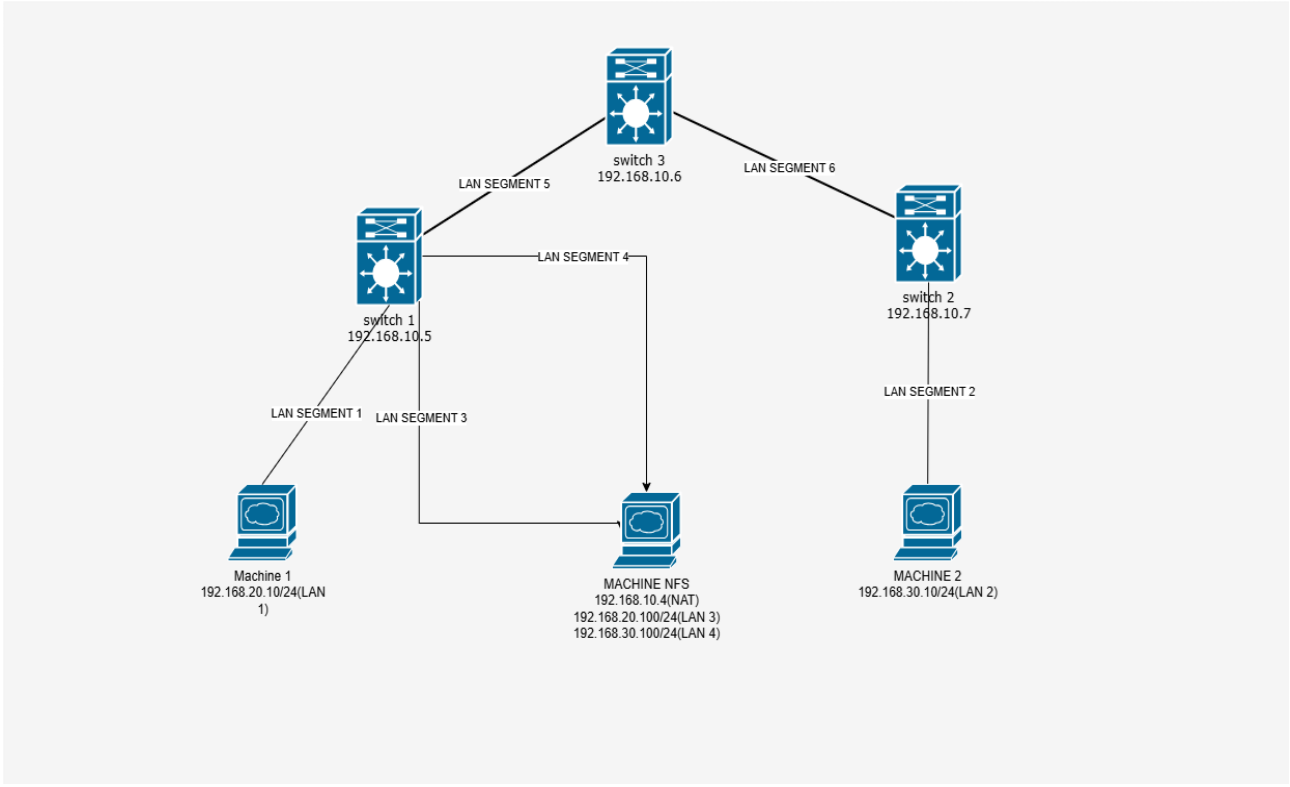
```
  GNU nano 7.2
127.0.0.1       localhost
127.0.1.1       debian

192.168.10.2 machineB
192.168.10.3 machineC
192.168.10.4 nfs
192.168.10.5 sw1
192.168.10.6 sw2
192.168.10.7 sw3
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Test Ansible

```
root@debian:~# ansible -m ping machineB
machineB | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
root@debian:~# ansible -m ping machineC
machineC | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
root@debian:~#
```

Mise en place du Partage réseau NFS

Topologie Réseau



Configuration des VLANs et Segmentation

Équipement	VLAN	Sous-réseau	Rôle
Machine1	20	192.168.20.0/24	Client NFS (VLAN20).
Machine2	30	192.168.30.0/24	Client NFS (VLAN30).

Serveur NFS 20,30 192.168.20.100/24, 192.168.30.100/24 Partage NFS multi-VLAN.

Nous allons utilisé VLAN 20 et VLAN 30 pour isoler la communication entre les machines et le serveur NFS, assurant ainsi une séparation stricte du trafic et renforçant la sécurité du réseau.

Explication : du choix de deux vlan

Machine 1: /etc/network/interface

```
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug enp0s3


#interface connexion internet
auto ens33
iface ens33 inet dhcp

#interface lan segment
auto ens35
iface ens35 inet static
address 192.168.20.10/24
```

Machine 2 : /etc/network/interface

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug enp0s3


#interface connexion internet
auto ens33
iface ens33 inet dhcp

#interface lan segment
auto ens35
iface ens35 inet static
address 192.168.20.10/24
#gateway 192.168.20.254
```

Configuration du Serveur NFS :

/etc/network/interface

```
auto ens36
iface ens36 inet manual
mtu 1500
link-duplex full

# Interface physique 2
auto ens37
iface ens37 inet manual
mtu 1500
link-duplex full

# Interface de bonding
auto bond0
iface bond0 inet manual
    bond-mode 802.3ad
    bond-slaves ens36 ens37
    bond-miimon 100
    bond-lacp-rate fast
```

```
# Sous-interface VLAN 30 pour Machine 2
auto bond0.30
iface bond0.30 inet static
    address 192.168.30.100/24
   # gateway 192.168.30.254
    vlan-raw-device bond0
    mtu 1500
```

Configuration des Switches

Switch1

Terminal | Terminal

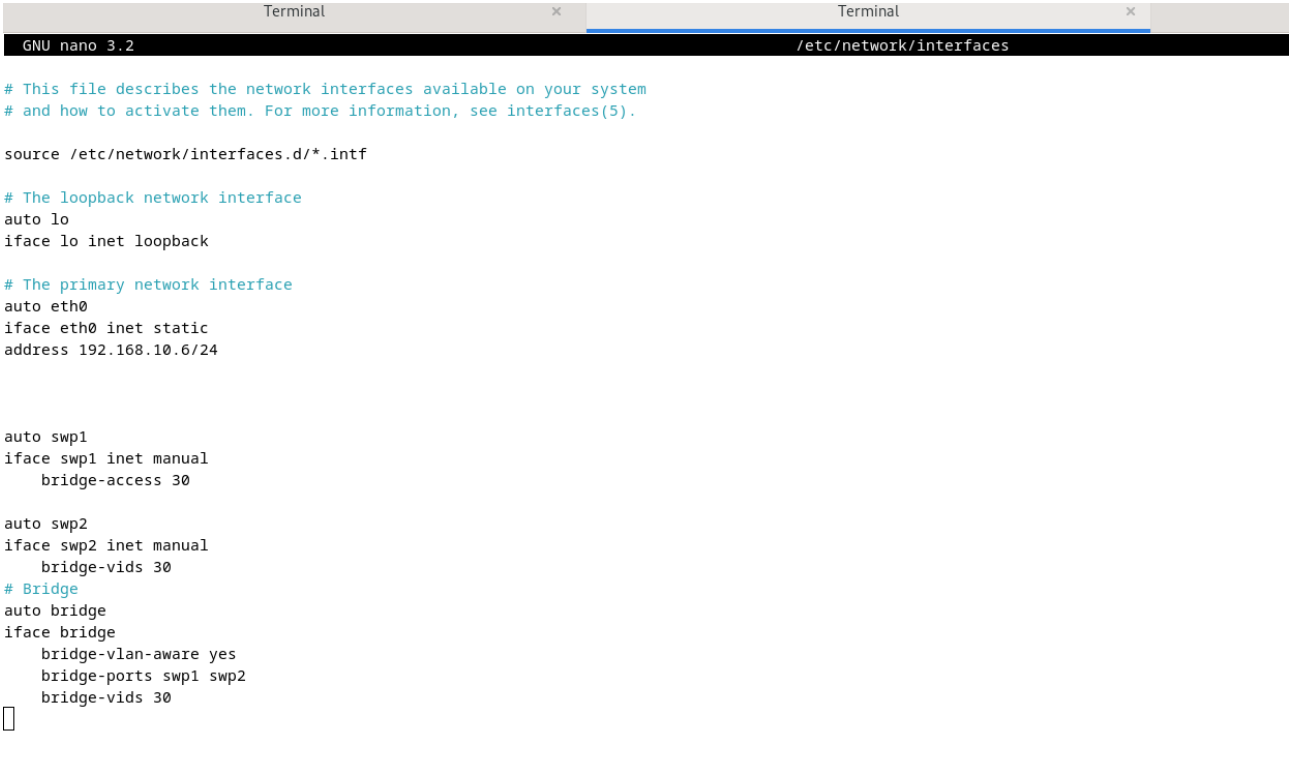
```
 GNU nano 3.2                                  /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*.intf

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.10.5/24

#port vers la machine1
auto swp1
iface swp1
    bridge-access 20
    mtu 1500
auto swp2
iface swp2
    bridge-access 20

auto swp3
iface swp3
    bridge-access 30

auto swp4
iface swp4
    bridge-vids 20 30
    mtu 1500
```

bonding

Terminal | Terminal

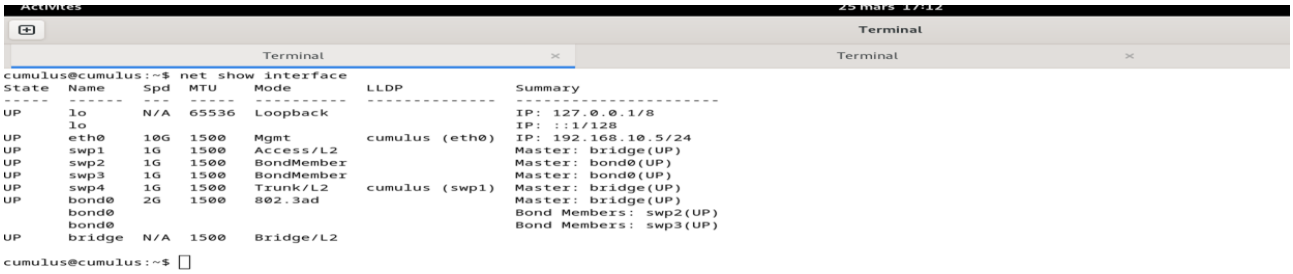
```
 GNU nano 3.2                                  /etc/network/interfaces

auto bond0
iface bond0
    bond-mode 802.3ad
    bond-slaves swp2 swp3
    bond-miimon 100
    bond-lacp-rate fast
    mtu 1500
    bond-xmit-hash-policy layer3+4

# Bridge
auto bridge
iface bridge
    bridge-vlan-aware yes
    bridge-ports swp1 swp4 bond0
    bridge-vids 20 30
    mtu 1500
    link-speed 1000
```

switch2 :



switch3

net show interface :



Details de la configuration

1. Machine1 (192.168.20.10)

VLAN20 : Interface ens35 configurée en 192.168.20.10/24 sans passerelle
Isolation : Aucune route vers VLAN30, trafic limité au VLAN20 via Switch1 et Switch3.

2. Machine2 (192.168.30.10)
VLAN30 : Interface ens36 en 192.168.30.10/24 avec passerelle 192.168.30.254 (Switch3).
Isolation : Routage via Switch3 pour accéder au NFS, blocage des paquets vers VLAN20.

3. Serveur NFS (192.168.20.100, 192.168.30.100)
Bond LACP : Agrégation bond0 (mode 802.3ad) sur ens36/ens37 pour redondance.
VLANs : Sous-interfaces bond0.20 (VLAN20) et bond0.30 (VLAN30) pour accès multi-sous-réseaux.
4. Switch1
Ports : swp1 (VLAN20), swp2 (VLAN20), swp3 (VLAN30), bond0 (sw2 et swp3), swp4 (trunk VLAN20/30 vers Switch3).

LACP : Bond 802.3ad avec bond-miimon 100 pour détection de liens.
5. Switch2
Ports : swp1 (VLAN30), swp2 (trunk VLAN30 vers Switch3).
6. Switch3
Routage L3 : vlan30 (192.168.30.254).

Test :
Ping de MACHINE 1 VERS MACHINE2 (pas de communication entre les deux)

```
root@debian:~# ping 192.168.30.100
PING 192.168.30.100 (192.168.30.100) 56(84) bytes of data.
From 192.168.30.1 icmp_seq=3 Destination Host Unreachable
^C
--- 192.168.30.100 ping statistics ---
5 packets transmitted, 0 received, +1 errors, 100% packet loss, time 4065ms

root@debian:~#
```

Explication :

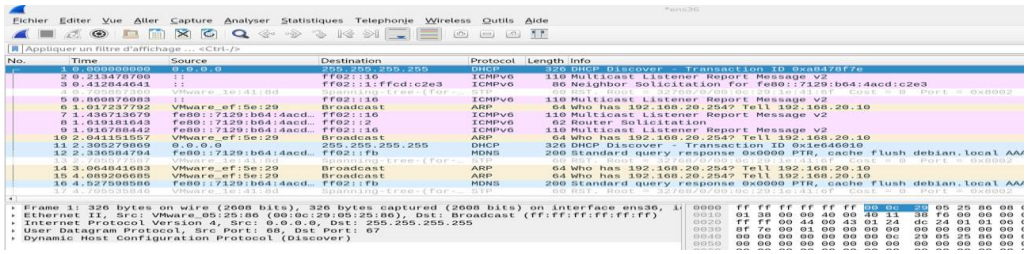
Étape 1 :

Machine1 envoie une requête ICMP (ping) vers 192.168.30.10.
Décision de routage :
La destination 192.168.30.10 ne se trouve pas dans le sous-réseau local (192.168.20.0/24).

Étape 2 :

Le paquet traverse Switch1 (VLAN20) et arrive sur Switch3 via le trunk (VLAN20).
Switch3 (routeur L3) :
Vérifie la table de routage.
Aucune route n'est configurée entre VLAN20 et VLAN30 (isolation stricte).
Action : Le paquet est jeté

on ne voit pas le paquet response



Ping de Machine 1 vers serveurs NFS

```
root@debian:~# ping 192.168.20.100
PING 192.168.20.100 (192.168.20.100) 56(84) bytes of data.
64 bytes from 192.168.20.100: icmp_seq=1 ttl=64 time=0.919 ms
64 bytes from 192.168.20.100: icmp_seq=2 ttl=64 time=1.79 ms
64 bytes from 192.168.20.100: icmp_seq=3 ttl=64 time=2.03 ms
^C
--- 192.168.20.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.919/1.580/2.030/0.477 ms
root@debian:~#
```

Machine1 :

Le paquet est envoyé depuis ens35 (VLAN20).
La passerelle par défaut n'est pas nécessaire car la destination est dans le même sous-réseau (192.168.20.0/24).
Switch1 :
Port swp1 : Reçoit le paquet en tant que port d'accès VLAN20.
Table de commutation :
Le VLAN20 est autorisé sur swp1, bond0, et le trunk swp4.
Décision :
Le paquet est transmis vers bond0 (interface du serveur NFS dans VLAN20).
Serveur NFS :
Interface bond0.20 : Reçoit le paquet via le bond LACP (ens36 ou ens37).
Réponse : Le serveur renvoie le paquet via bond0.20 vers Machine1.

Ping de Machine 2 vers serveurs NFS :

```
root@debian:~# ping 192.168.30.100
PING 192.168.30.100 (192.168.30.100) 56(84) bytes of data.
64 bytes from 192.168.30.100: icmp_seq=1 ttl=64 time=1.62 ms
64 bytes from 192.168.30.100: icmp_seq=2 ttl=64 time=3.90 ms
64 bytes from 192.168.30.100: icmp_seq=3 ttl=64 time=4.38 ms
^C
--- 192.168.30.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 1.621/3.299/4.383/1.203 ms
root@debian:~#
```

Étape 1 : Machine2 → Switch2

Interface source : ens36 (adresse IP 192.168.30.10).
Destination : 192.168.30.100 (interface bond0.30 du serveur NFS).

Machine2 :
Le paquet est envoyé depuis ens36 (VLAN30).
Passerelle : 192.168.30.254 (interface VLAN30 de Switch3).

Switch2 :
Port swp1 : Reçoit le paquet en tant que port d'accès VLAN30.
Table de commutation :
Le VLAN30 est autorisé sur swp1 et le trunk swp2.

Décision :

Le paquet est transmis via le trunk swp2 vers Switch3.
Switch3 (Routeur L3) :
Interface VLAN30 (192.168.30.254) :
Reçoit le paquet via le trunk swp2 (VLAN30).
Routage :

La destination 192.168.30.100 est dans le même sous-réseau.
Le paquet est renvoyé via le trunk swp1 vers Switch1.

Switch1 :
Trunk swp4 : Reçoit le paquet taggué VLAN30.
Table de commutation :
Le VLAN30 est autorisé sur bond0 et swp4.
Décision :
Le paquet est transmis vers bond0 (interface du serveur NFS dans VLAN30).
Serveur NFS :
Interface bond0.30 : Reçoit le paquet via le bond LACP (ens36 ou ens37).
Réponse : Le serveur renvoie le paquet via bond0.30 vers Machine2.

Partie 1-2 NFS : CONFIGURATION

1 Installation du Serveur NFS

```
sudo apt update && sudo apt install nfs-kernel-server
```

2. Créer les dossiers sur le Serveur NFS

```
sudo mkdir -p /machine1 /machine2
```

2- dans le fichier /etc/exports du serveur

autorisé mes deux machines à monté les deux dossiers

```
GNU nano 7.2                                                    /etc/exports
/etc/exports: the access control list for filesystems which may be exported
              to NFS clients.  See exports(5).

Example for NFSv2 and NFSv3:
/srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)

Example for NFSv4:
/srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
/srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)

/srv/nfs1      192.168.20.10(rw,sync,no_subtree_check)
/srv/nfs1       192.168.30.10(rw,sync,no_subtree_check)
Fichier /etc/exports sur le serveur NFS
machine1 192.168.20.10/24(rw,sync,no_subtree_check)
machine2 192.168.30.10/24(rw,sync,no_subtree_check)
```

Nous allons appliqué les changement

sudo exportfs -ra

nous allons verifer avec la commande

showmount -e

3-dans chaque machine nous allons monté les fichiers respectifs

dans le fichier /etc/fstab
machine 1

```
GNU nano 7.2                                                        /etc/fstab
/etc/fstab: static file system information.

Use 'blkid' to print the universally unique identifier for a
device; this may be used with UUID= as a more robust way to name devices
that works even if disks are added and removed. See fstab(5).

systemd generates mount units based on this file, see systemd.mount(5).
Please run 'systemctl daemon-reload' after making changes here.

<file system> <mount point>   <type>  <options>       <dump>  <pass>
/ was on /dev/sda1 during installation
JID=22755f57-78a6-4801-aba2-0e79c8d76bbf /               ext4    errors=remount-ro 0       1
swap was on /dev/sda5 during installation
JID=d394fba2-28bd-4546-8276-427ab9887fbd none            swap    sw              0       0
dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
92.168.20.100:/machine1  /mnt/machine1  nfs  vers=3,rw,soft,intr  0  0
```

pour machine 2 pareil

```
 GNU nano 7.2                                                                /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=22755f57-78a6-4801-aba2-0e79c8d76bbf /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=d394fba2-28bd-4546-8276-427ab9887fbd none            swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
192.168.30.100:/machine2  /mnt/machine2  nfs  vers=3,rw,soft,intr  0  0
```

verfions avec la command df -h pour voir si le montage c'est bien passé
machine1 :

Nous allons créer un fichier, y écrire un texte, puis vérifier son contenu à la fois sur la machine et sur le serveur.

```
root@debian:~# df -h
Sys. de fichiers         Taille Utilisé Dispo Uti% Monté sur
udev                       953M       0  953M   0% /dev
tmpfs                      197M    1,3M  196M   1% /run
/dev/sda1                   19G    5,4G   13G  31% /
tmpfs                      984M       0  984M   0% /dev/shm
tmpfs                      5,0M       0  5,0M   0% /run/lock
192.168.20.100:/machine1    19G    5,5G   13G  31% /machine1partage
tmpfs                      197M     68K  197M   1% /run/user/0
192.168.20.100:/machine1    19G    5,5G   13G  31% /mnt/machine1
root@debian:~# nano /etc/fstab
root@debian:~# ls /mnt/machine1
root@debian:~# touch /mnt/machine2/test.txt
touch: impossible de faire un touch '/mnt/machine2/test.txt': Aucun fichier ou dossier de ce type
root@debian:~# touch /mnt/machine1/test1.txt
root@debian:~# nano /mnt/machine1/test1.txt
root@debian:~# cat /mnt/machine1/test1.txt
Bonjour je suis machine1
root@debian:~# mount -a
root@debian:~# nano /etc/fstab
root@debian:~#
```

machine2

```
root@debian:~# df -h
Sys. de fichiers         Taille Utilisé Dispo Uti% Monté sur
udev                       953M       0  953M   0% /dev
tmpfs                      197M    1,3M  196M   1% /run
/dev/sda1                   19G    5,3G   13G  30% /
tmpfs                      984M       0  984M   0% /dev/shm
tmpfs                      5,0M       0  5,0M   0% /run/lock
tmpfs                      197M     68K  197M   1% /run/user/0
192.168.30.100:/machine2    19G    5,5G   13G  31% /mnt/machine2
root@debian:~# nano /etc/fstab
root@debian:~# nano /etc/fstab
root@debian:~# ls /mnt/machine1
ls: impossible d'accéder à '/mnt/machine1': Aucun fichier ou dossier de ce type
root@debian:~# ls /mnt/machine2/
root@debian:~# touch /mnt/machine2/text2.txt
root@debian:~# nano /mnt/machine2/text2.txt
root@debian:~# ls /mnt/machine2/
text2.txt
root@debian:~# cat /mnt/machine2/text2.txt
Bonjour je suis machine 2
root@debian:~# mount -a
root@debian:~# nano /etc/fstab
```

dans le serveur

```
root@debian:~# cd /m
machine1/ machine2/ media/    mnt/
root@debian:~# cd /machine1
root@debian:/machine1# ls
test1.txt
root@debian:/machine1# cat test1.txt
Bonjour je suis machine1
root@debian:/machine1# cd #
root@debian:~# cd /machine2/
root@debian:/machine2# ls
text2.txt
root@debian:/machine2# cat text2.txt
Bonjour je suis machine 2
root@debian:/machine2#
```

Partie II
2) Tolérance de panne, couche « access

2.1 Introduction

Cette partie présente une solution technique pour répondre aux exigences de tolérance de panne au niveau de la couche « access » d'un réseau de type campus hiérarchique, tout en optimisant les performances. L'objectif est d'assurer une continuité de service entre deux machines (Machine 1 et Machine 2) en cas de défaillance d'un commutateur d'accès (Switch 1 ou Switch 2), tout en respectant la contrainte de non-redondance du commutateur de niveau 3 (Switch 3). Les configurations fournies sont analysées, expliquées ligne par ligne avec les motifs des choix techniques, et validées par des tests de panne.

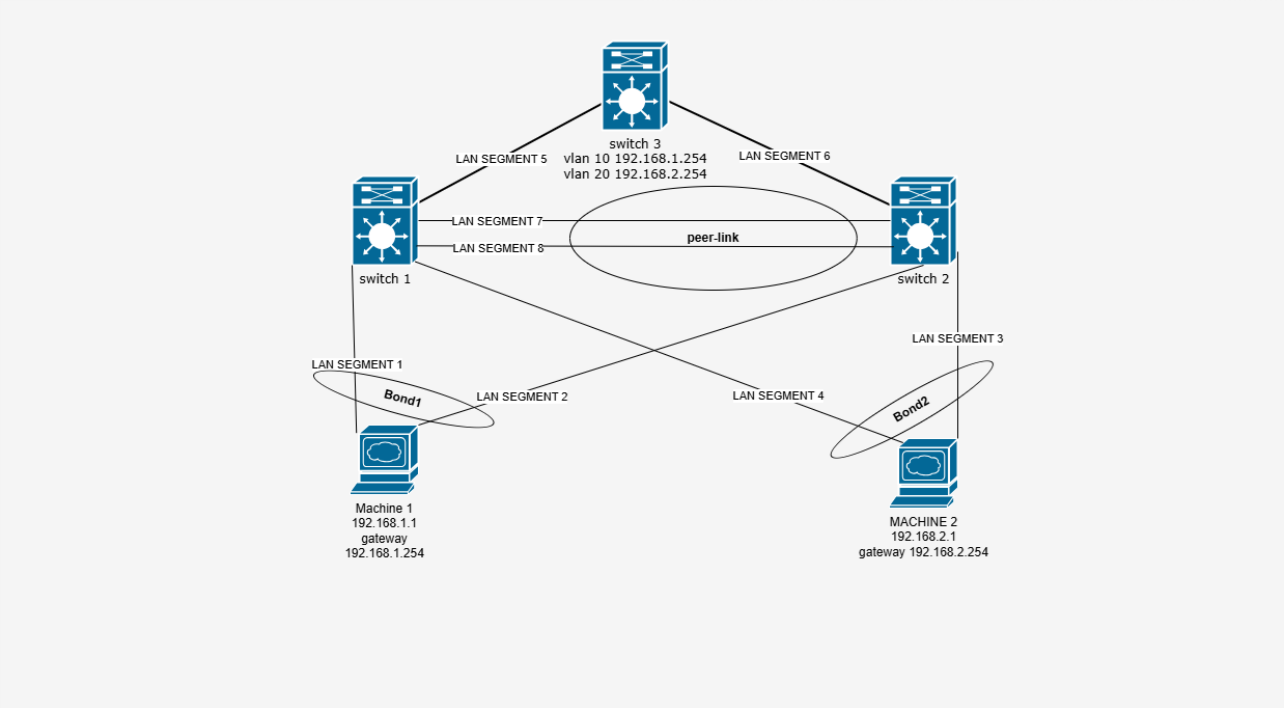
2.2. Contexte et Analyse des Besoins

L'architecture réseau actuelle est une topologie hiérarchique de type campus avec :
Couche Access : Commutateurs 1 et 2 (niveau 2).
Couche Distribution/Core : Commutateur 3 (multi-niveaux, routage statique).
Machines :
Machine 1 (192.168.1.1/24) connectée à Switch 1,
Machine 2 (192.168.2.1/24) connectée à Switch 2.

2.3 Besoins exprimés :

Tolérance de panne des commutateurs 1 et 2 : aucune interruption de trafic entre Machine 1 et Machine 2 en cas de panne de l'un des deux.
Maximisation des performances.
Tolérance à la non-redondance de Switch 3.
Contraintes : utilisation de Cumulus Linux, pas d'ajout de commutateurs, pas de connexion directe entre Machine 1/Machine 2 et Switch 3.

2.4 Schéma réseau



3. Configurations et Explications Détaillées

Machine 1 :

```
auto ens35
iface ens35 inet manual
     bond-master bond0

auto ens33
iface ens33 inet manual
     bond-master bond0

auto bond0
iface bond0 inet static
     address 192.168.1.1/24
     gateway 192.168.1.254
     bond-mode 802.3ad
     bond-slaves ens35 ens33
     bond-miimon 100
     bond-updelay 100
     bond-xmit-hash-policy layer3+4
     bond-lacp-rate 1
```

Nous avons agrégé nos interfaces en deux interfaces groupées, chaque interface étant liée à un switch distinct. Cela garantit qu'en cas de panne d'un switch, la liaison entre Machine1 et Machine2 ne sera pas interrompue nous avons plus de performance, d'où le choix d'agréger les interfaces. Comme on peut le constater, le mode utilisé est LACP.

bond0 : Agrégation LACP avec IP 192.168.1.1/24 et passerelle 192.168.1.254 (Switch 3)

Bond-mode 802.3ad (LACP) : Assure redondance et performance.

miimon 100 : Vérification des liens toutes les 100 ms pour une détection rapide des pannes.

updelay 100 : Introduit un délai pour éviter les basculements intempestifs.

xmit-hash-policy layer3+4 : Équilibrage du trafic basé sur les adresses IP et les ports pour maximiser la bande passante.

Sortie de cat /proc/net/bonding/bond0 : permet de vérifier que la machine est bien synchronisée avec les deux switches.

```
root@debian:~# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v6.1.0-12-amd64

Bonding Mode: IEEE 802.3ad Dynamic link aggregation
Transmit Hash Policy: layer3+4 (1)
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 100
Down Delay (ms): 0
Peer Notification Delay (ms): 0

802.3ad info
LACP active: on
LACP rate: fast
Min links: 0
Aggregator selection policy (ad_select): stable
System priority: 65535
System MAC address: 00:0c:29:a6:cf:97
Active Aggregator Info:
        Aggregator ID: 2
        Number of ports: 2
        Actor Key: 9
        Partner Key: 9
        Partner Mac Address: 44:38:39:ff:00:aa

Slave Interface: ens33
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 1
Permanent HW addr: 00:0c:29:a6:cf:97
Slave queue ID: 0
Aggregator ID: 2
Actor Churn State: none
Partner Churn State: none
Actor Churned Count: 1
```

00:0c:29:a6:cf:97 (System MAC address, ens33 Permanent HW addr)
c'est l'identifiant matériel (MAC) de l'interface ens33, également utilisée comme adresse système pour l'agrégation LACP (Actor). Elle est générée par la carte réseau physique de Machine 1 (typiquement une VM VMware, d'où le préfixe 00:0c:29). Dans LACP, l'Actor est l'entité locale qui négocie l'agrégation avec le Partner (Switch 1/2).
Rôle : Identifie Machine 1 dans les trames LACP et le trafic réseau. Comme ens33 est la première interface esclave, sa MAC est adoptée pour bond0.

00:0c:29:a6:cf:a1 (ens35 Permanent HW addr)
c'est l'Adresse MAC physique de l'interface ens35. Les deux interfaces conserve leur mac, mais elle n'est pas utilisée comme adresse source dans le trafic agrégé (bond0 utilise la MAC d'ens33). Elle apparaît dans les PDU LACP pour identifier le port spécifique ici c'est le port number: 2.

Rôle : Permet au partenaire (Switch 1/2) de distinguer les deux liens physiques dans l'agrégation, essentiel pour la détection de panne et l'équilibrage.

44:38:39:ff:00:aa (Partner Mac Address)
c'est l'Adresse MAC virtuelle partagée par Switch 1 et Switch 2, qui est configurée via clagd-sys-mac dans MLAG. Elle représente le Partner dans la négociation LACP, c'est-à-dire l'entité logique formée par les deux commutateurs d'accès. Cette MAC est fixe et identique sur les deux switches pour simuler un seul dispositif face à Machine 1.
Rôle : Assure que Machine 1 voit une entité cohérente, même en cas de panne d'un switch, grâce à la synchronisation MLAG.
Motif : Stabilité et transparence de la redondance active-active.

Machine2 aura la même configuration et detail que Machine1.

```
auto ens35
iface ens35 inet manual
    bond-master bond0

auto ens33
iface ens33 inet manual
    bond-master bond0

auto bond0
iface bond0 inet static
    address 192.168.2.1/24
    gateway 192.168.2.254
    bond-mode 802.3ad
    bond-slaves ens35 ens33
    bond-miimon 100
    bond-updelay 100
    bond-xmit-hash-policy layer3+4
    bond-lacp-rate 1
```

Les switches

switch1 :

```
auto eth0
iface eth0 inet static
    address 192.168.10.11/24
#liens vers machine 1 acces

auto swp1
iface swp1
    bond-master bond1

auto swp2
iface swp2
    bridge-vids 10 20
    bridge-pvid 1

auto swp3
iface swp3

auto swp4
iface swp4

auto swp5
iface swp5
    bond-master bond2

auto bond1
iface bond1
    bond-slaves swp1
    bond-mode 802.3ad
    clag-id 1
    bridge-access 10
    bond-miimon 100
```

```
auto bond2
iface bond2
    bond-slaves swp5
    bond-mode 802.3ad
    clag-id 2
    bridge-access 20
    bond-miimon 100

#peerlink Mlag vers Switch2
auto peerlink
iface peerlink
    bond-slaves swp3 swp4
    bond-mode 802.3ad
    bond-miimon 100
    bond-lacp-rate fast

#Controle Mlag
auto peerlink.4094
iface peerlink.4094
    address 192.168.10.11/24
    clagd-peer-ip 192.168.10.12
    clagd-backup-ip 192.168.10.12
    clagd-sys-mac 44:38:39:FF:00:AA
    clagd-priority 100

auto bridge
iface bridge
    bridge-vlan-aware yes
    bridge-ports bond1 bond2 swp2 peerlink
    bridge-vids 10 20
    bridge-pvid 1
```

NB:l'interface eth0 est l'interface d'administration

swp2:C'est l'interface qui vas relié mon switch2 au switch3 cette interface laisse passé toutes les trames taggées avec VID 10 ou 20 pour acheminer le trafic des deux VLANs vers Switch 3, qui effectue le routage inter-VLAN.

Nous avons utilisés des interfaces logiques pour lié chacun des liens physique à une machine
Bond1 : Connecte Machine 1 (VLAN 10) via swp1 sur chaque switch.
Bond2 : Connecte Machine 2 (VLAN 20) via swp5 (Switch 1) et swp3 (Switch 2).

Pour assurer la continuité du trafic entre les deux machines même en cas de panne d'un commutateur, il est nécessaire de regrouper les deux switches physiques en un switch logique à l'aide de MLAG (Multi-Chassis Link Aggregation).
Configuration MLAG :
Agrégation LACP entre Switch 1 (swp3, swp4) et Switch 2 (swp4, swp5).
Synchronisation MLAG via un VLAN dédié (VLAN 4094), transportant les états (tables MAC) et relayant le trafic si un lien direct échoue.
Sous-interface VLAN sur le peerlink avec une adresse IP attribuée à chaque switch :
Switch 1 → 192.168.10.11/24
Switch 2 → 192.168.10.12/24
Adresse MAC virtuelle (clagd-sys-mac) : Permet aux machines de voir un seul partenaire LACP, simulant un commutateur unique.

Priorité MLAG (clagd-priority 100) : Switch 1 est maître, Switch 2 est secondaire (priorité 200).
Mécanisme de fonctionnement :
En fonctionnement normal : Les switches échangent des messages via peerlink.4094 pour synchroniser leurs états.
En cas de panne (ex. swp1 défaillant sur Switch 1) :
Switch 2 détecte l'anomalie via clagd.

Il relaie le trafic via le peerlink pour assurer la continuité du réseau.
Et à la fin nous allons créer un pont : Le bridge pour connecter bond1, bond2, swp2 et peerlink, gérant les VLANs 10 et 20 pour commuter le trafic entre les machines et Switch 3.
Mode VLAN-aware, ports spécifiques, VLANs autorisés (10, 20),

Switch 2 : disposera d'une configuration identique, à l'exception des adresses IP qui diffèrent.

```
iface eth0 inet static
   address 192.168.10.12/24


auto swp1
iface swp1
    bond-master bond1

auto swp2
iface swp2
    bridge-vids 10 20
    bridge-pvid 1


auto swp3
iface swp3
    bond-master bond2

auto swp4
iface swp4

auto swp5
iface swp5

auto bond1
iface bond1
    bond-slaves swp1
    bridge-access 10
    bond-mode 802.3ad
    clag-id 1
    bond-miimon 100
    bond-lacp-rate fast
```

```
auto bond2
iface bond2
    bond-slaves swp3
    bond-mode 802.3ad
    bridge-access 20
    clag-id 2
    bond-miimon 100
    bond-lacp-rate fast


#peerlink MLAG vers Switch1
auto peerlink
iface peerlink
    bond-slaves swp4 swp5
    bond-mode 802.3ad
    bond-miimon 100
    bond-lacp-rate fast

#controle MLAG
auto peerlink.4094
iface peerlink.4094
    address 192.168.10.12/24
    clagd-peer-ip 192.168.10.11
    clagd-backup-ip 192.168.10.11
    clagd-sys-mac 44:38:39:FF:00:AA
    clagd-priority 200

auto bridge
iface bridge
    bridge-vlan-aware yes
    bridge-ports bond1 bond2 swp2 peerlink
    bridge-vids 10 20
```

```
source /etc/network/interfaces.d/*.intf

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.10.13/24

auto swp1
iface swp1

auto swp2
iface swp2

auto vlan1
iface vlan1 inet static
     address 192.168.1.254/24
     vlan-id 10
     vlan-raw-device bridge

auto vlan2
iface vlan2 inet static
     address 192.168.2.254/24
     vlan-id 20
      vlan-raw-device bridge

auto bridge
iface bridge
     bridge-vlan-aware yes
     bridge-ports swp1 swp2
     bridge-vids 10 20
     bridge-pvid 1
```

Switch3 Il assure

Commutation L2 : Le bridge transporte les trames des VLANs 10 et 20 entre swp1 et swp2, en apprenant les adresses MAC (ex. Machine 1 via Switch 1, Machine 2 via Switch 2).

Routage L3 : Les interfaces vlan1 et vlan2 routent les paquets entre les sous-réseaux :
Exemple : Machine 1 (192.168.1.1) envoie un paquet à Machine 2 (192.168.2.1) :
Le paquet arrive sur vlan1 (192.168.1.254).

Switch 3 consulte sa table de routage statique (implicite ici) et envoie le paquet vers vlan2 (192.168.2.254).

La trame sort via swp1 ou swp2 vers Machine 2.
Passerelle : Les machines utilisent les adresses .254 comme next-hop pour tout trafic hors de leur sous-réseau.

Tests et Vérifications

l'interface switch1

l'interface switch2

```
cumulus@cumulus:~$ net show interface
State  Name           Spd  MTU    Mode        LLDP           Summary
-----  -------------  ---  -----  ----------  -------------  ---------------------
UP     lo             N/A  65536  Loopback                   IP: 127.0.0.1/8
       lo                                                    IP: ::1/128
UP     eth0           10G  1500   Mgmt        cumulus (eth0) IP: 192.168.10.11/24
UP     swp1           1G   9216   BondMember                 Master: bond1(UP)
UP     swp2           1G   9216   Trunk/L2    cumulus (swp1) Master: bridge(UP)
UP     swp3           1G   9216   BondMember  cumulus (swp4) Master: peerlink(UP)
UP     swp4           1G   9216   BondMember  cumulus (swp5) Master: peerlink(UP)
UP     swp5           1G   9216   BondMember                 Master: bond2(UP)
UP     bond1          1G   9216   802.3ad                    Master: bridge(UP)
       bond1                                                 Bond Members: swp1(UP)
UP     bond2          1G   9216   802.3ad                    Master: bridge(UP)
       bond2                                                 Bond Members: swp5(UP)
UP     bridge         N/A  9216   Bridge/L2
UP     peerlink       2G   9216   802.3ad                    Master: bridge(UP)
       peerlink                                              Bond Members: swp3(UP)
       peerlink                                              Bond Members: swp4(UP)
UP     peerlink.4094  2G   9216   SubInt/L3                  IP: 192.168.10.11/24

cumulus@cumulus:~$
```

```
cumulus@cumulus:~$ net show interface
State  Name           Spd  MTU    Mode        LLDP           Summary
-----  -------------  ---  -----  ----------  -------------  ---------------------
UP     lo             N/A  65536  Loopback                   IP: 127.0.0.1/8
       lo                                                    IP: ::1/128
UP     eth0           1G   1500   Mgmt        cumulus (eth0) IP: 192.168.10.12/24
UP     swp1           1G   9216   BondMember                 Master: bond1(UP)
UP     swp2           1G   9216   Trunk/L2    cumulus (swp2) Master: bridge(UP)
UP     swp3           1G   9216   BondMember                 Master: bond2(UP)
UP     swp4           1G   9216   BondMember  cumulus (swp3) Master: peerlink(UP)
UP     swp5           1G   9216   BondMember  cumulus (swp4) Master: peerlink(UP)
UP     bond1          1G   9216   802.3ad                    Master: bridge(UP)
       bond1                                                 Bond Members: swp1(UP)
UP     bond2          1G   9216   802.3ad                    Master: bridge(UP)
       bond2                                                 Bond Members: swp3(UP)
UP     bridge         N/A  9216   Bridge/L2
UP     peerlink       2G   9216   802.3ad                    Master: bridge(UP)
       peerlink                                              Bond Members: swp4(UP)
       peerlink                                              Bond Members: swp5(UP)
UP     peerlink.4094  2G   9216   SubInt/L3                  IP: 192.168.10.12/24

cumulus@cumulus:~$
```

switch3 :

```
cumulus@cumulus:~$ net show interface
State  Name    Spd  MTU    Mode         LLDP           Summary
-----  ------  ---  -----  -----------  -------------  --------------------
UP     lo      N/A  65536  Loopback                    IP: 127.0.0.1/8
       lo                                              IP: ::1/128
UP     eth0    1G   1500   Mgmt         cumulus (eth0) IP: 192.168.10.13/24
UP     swp1    1G   9216   Trunk/L2     cumulus (swp2) Master: bridge(UP)
UP     swp2    1G   9216   Trunk/L2     cumulus (swp2) Master: bridge(UP)
UP     bridge  N/A  9216   Bridge/L2
UP     vlan1   N/A  9216   Interface/L3                IP: 192.168.1.254/24
UP     vlan2   N/A  9216   Interface/L3                IP: 192.168.2.254/24

cumulus@cumulus:~$
```

Ce switch de niveau 3 possède deux interfaces trunk (swp1, swp2) connectées à un bridge L2 pour la commutation interne.
Deux interfaces VLAN (vlan1 et vlan2) sont configurées en L3 avec les IP 192.168.1.254/24 et 192.168.2.254/24, servant de passerelles par sous-réseau.
L'interface eth0 sert à la gestion avec l'IP 192.168.10.13/24.

Explication swtich1
1. Interfaces physiques

Interface	Description
eth0	Interface de management
swp1	Port physique utilisé dans bond1 avec le vlan10 vers machine1
swp2	Port trunk connecté au bridge Connecte les VLANs à la couche L2
swp3/swp4	Ports physiques membres de peerlink UP Lien MLAG avec un switch partenaire
swp5	Port physique dans bond2 Agrégé dans bond2

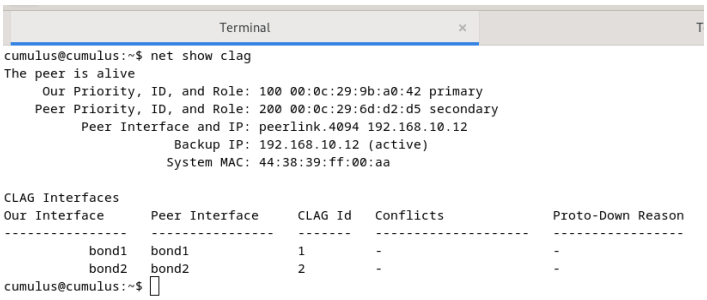
les bonds :
bond1 et bond2 connectent les deux swits au deux machines .
peerlink est le lien critique pour la redondance MLAG.

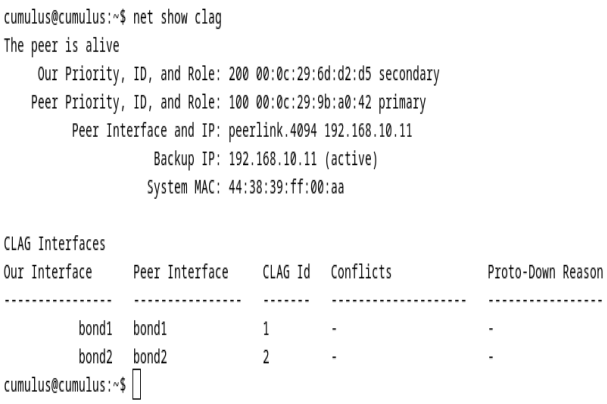
Le bridge : regroupant les interfaces physiques et logiques pour permettre la commutation au niveau 2. Il transporte aussi les VLANs, via swp2 (Trunk).

Subinterface L3 (peerlink.4094) :
Cette interface estutilisée pour la communication MLAG, la synchronisation des bases de données ARP/MAC entre les deux switchs L2 partenaires.

même logique pour le switch2

Vérification de la redondance et de la synchronisation entre les switchs
swtich1 switch2

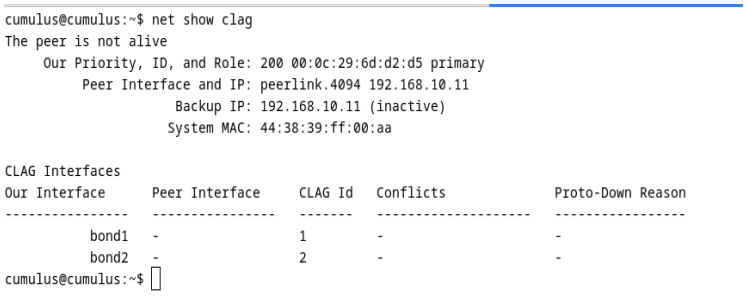




Ici Nous constatons que le MLAG (CLAG) entre les deux switchs fonctionne : le peer est actif.
Le switch local est primaire (priorité 100) et le switch2 est secondaire (priorité 200).
La communication MLAG passe par l'interface peerlink.4094 avec l'IP du peer 192.168.10.12.
Les interfaces bond1 et bond2 sont bien appariées avec le switch secondaire sans conflit.
Aucune erreur ou désactivation n'est détectée, la redondance est pleinement opérationnelle.

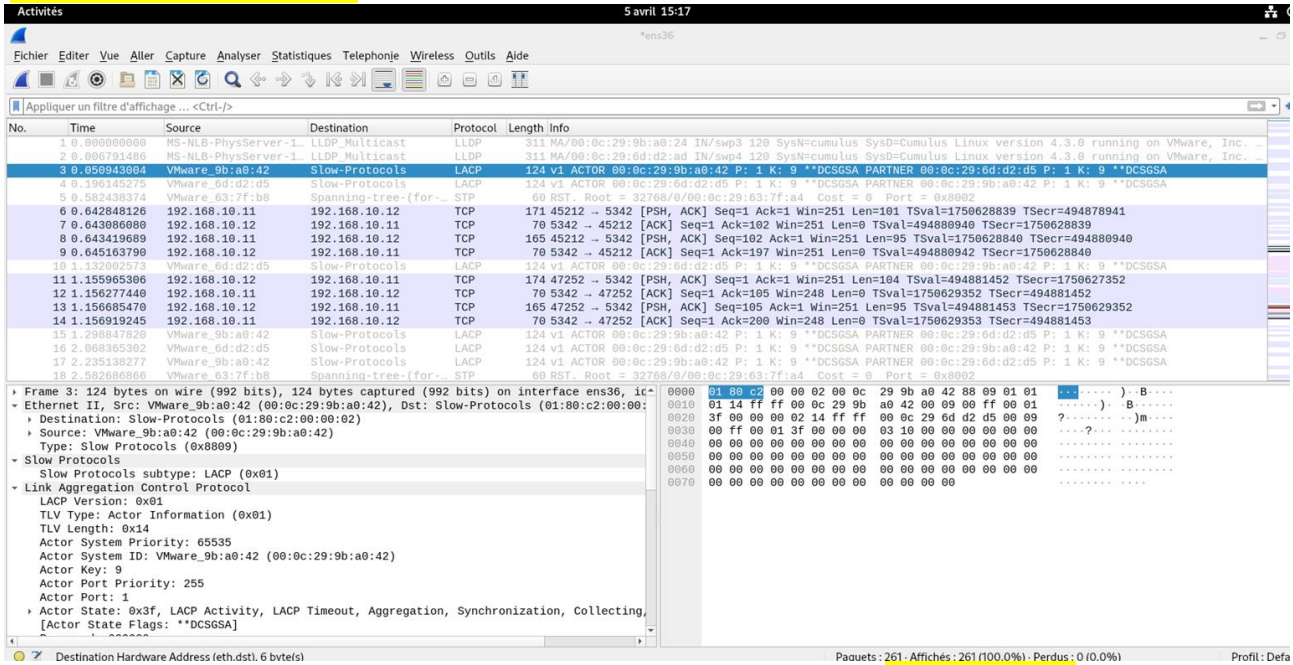
Test de la Rédondance après coupure du switch1(primaire)

status du switch2 change en primary maintenant et sont partenaire pas inactive



Fonctionnement de la redondance(explication avec une capture wireshark en ecoutant l'une des interface membre du peerlink

les deux switchs fonctionnent



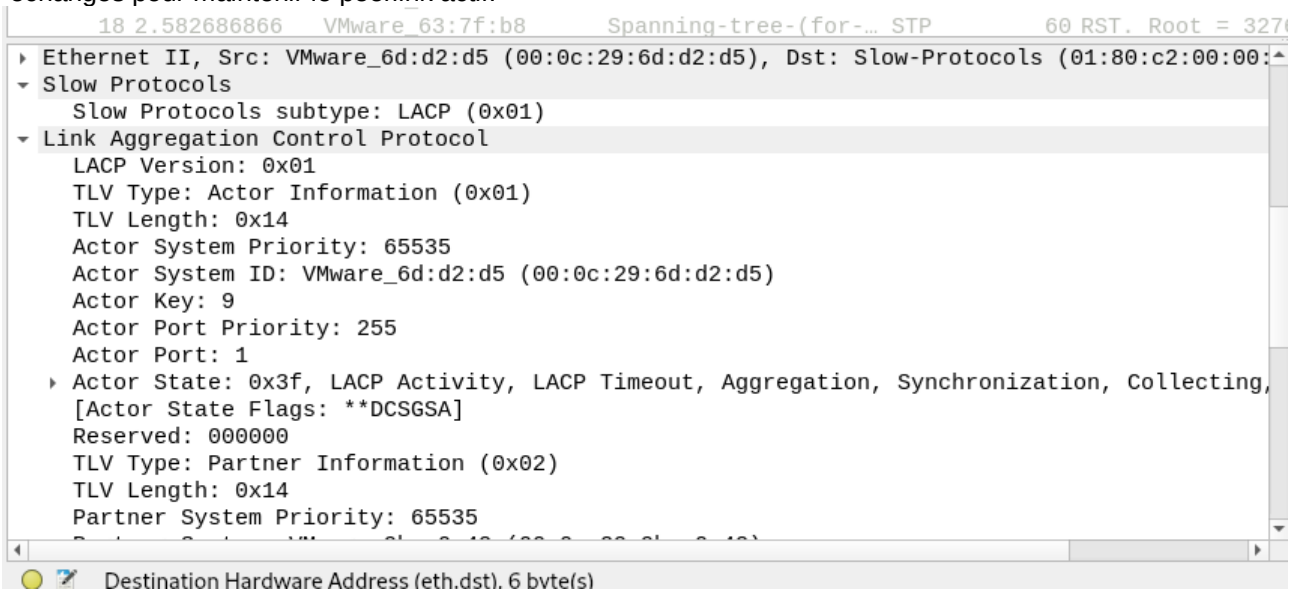
La prémière de chose que nous pouvons constaté c'est le protocol utilisé (LACP (802.3ad)):
il gère le peerlink pour Confirme la santé des ports (swp3-swp4, swp4-swp5).

CLAG :

Synchronise MLAG (MAC, bonds)
nous voyons les paquets tcp c'est eux qui permet de faire le keepalives toutes 1s pour verifier l'etat de sant du partenaire et c'est vraiment dans les deux sens Coordonne Switch1 et Switch2 via peerlink.4094

Fonctionnement

LACP: PDU paquet spécial envoyé pour négocier, maintenir, et surveiller les liens agrégés
échangés pour maintenir le peerlink actif.



Détails de la PDU

Ethernet : Src 44:38:39:FF:00:AA (MLAG), Dst 01:80:c2:00:00:02 (multicast LACP).

LACP : Version 1, Actor (Switch1) : System ID 44:38:39:FF:00:AA, Port swp3, State 0x3f (actif, synchronisé).

Partner (Switch2) : System ID attendu 44:38:39:FF:00:AA, Port swp4.

Fréquence : 1 PDU/s (bond-lacp-rate fast).

Note : Capture réelle (MAC 00:0c:29:6d:d2:d5) suggère un échange Machine1 → Switch1/Switch2.
PDU LACP maintient le peerlink actif. Si elles cessent (panne Switch1), Switch2 bascule en ~3s, gérant bond1 et bond2 via MLAG.

Test avec un ping :

Workflow d'un ping machine 1 vers machine 2
1. Machine1 :
Paquet : IP {Src: 192.168.1.1, Dst: 192.168.2.1}, ICMP Echo Request.
Décision L3 : 192.168.2.1 n'est pas dans le sous-réseau (192.168.1.0/24), donc envoi à la passerelle 192.168.1.254 (Switch3).
ARP : Machine1 résout la MAC de 192.168.1.254 → 00:0c:29:f1:f5:0e (Switch3).
LACP (bond0) :
bond-xmit-hash-policy layer3+4 : Hash sur IP src (192.168.1.1) et IP dst (192.168.2.1).
Résultat : Choix ens35 → Switch1 (swp1) dépend du hash.
Paquet envoyé : Ethernet {Src: 00:11:22:33:44:55 (Machine1), Dst: 00:0c:29:f1:f5:0e (Switch3)}, IP, ICMP.
2. Switch1 : Réception et commutation L2

Réception : swp1 (bond1, VLAN10).
Table MAC : 00:11:22:33:44:55 → bond1, VLAN10 (apprise).
MLAG : Synchronise la table MAC avec Switch2 via peerlink (swp3-swp4).
Décision L2 : MAC_switch3 (00:0c:29:f1:f5:0e) inconnue → Inondation sur swp2 (Switch3) et peerlink.
Sortie : swp2 (vers Switch3 swp1), taggé VLAN10.

3. Switch3 : Routage L3
Réception : swp1.
Table MAC : 00:11:22:33:44:55 → swp1, VLAN10.
Routage : 192.168.2.1 → VLAN20 (interface vlan2, 192.168.2.254).
ARP : 192.168.2.1 → 00:11:22:33:44:66 (Machine2).
Sortie : swp2 (vers Switch2), taggé VLAN20.
Paquet modifié : Ethernet {Src: 00:0c:29:f1:f5:0e, Dst: 00:11:22:33:44:66}, IP, ICMP.

4. Switch2 : Commutation L2

Réception : swp2.
Table MAC : 00:0c:29:f1:f5:0e → swp2, VLAN20.
Décision L2 : 00:11:22:33:44:66 → bond2, VLAN20 (apprise via trafic ou ARP).
VLAN : bridge-access 20 → Tag VLAN20 retiré.
Sortie : swp3 (bond2) vers Machine2.
LACP (Machine2 bond0) : Hash choisit ens33 .
5. Machine2 : Réception et réponse

Réception : bond0 (ens33).
Réponse : IP {Src: 192.168.2.1, Dst: 192.168.1.1}, ICMP Echo Reply.
LACP (bond0) : Hash (IP src: 192.168.2.1, IP dst: 192.168.1.1) → ens33 → Switch2 (swp3).
6. Reply : Machine2 → Switch2 → Switch3 → Switch2 → Machine1
Switch2 : swp3 (bond2) → swp2 (Switch3), taggé VLAN20.
Switch3 : swp2 → Routage → swp2 (Switch2), taggé VLAN10.
Switch2 : swp2 → swp1 (bond1, VLAN10, tag retiré).
Machine1 : Reçoit via ens33 (hash LACP).

Partie 3 - Redondance des switches de niveau 3 (Switch3 et Switch4)

1. Étude du besoin
L'objectif est d'assurer une redondance au niveau 3 (L3) entre Switch3 et Switch4 pour garantir la haute disponibilité des passerelles (192.168.1.254 pour VLAN10, 192.168.2.254 pour VLAN20). Cela permet de maintenir la connectivité inter-VLAN (Machine1 → Machine2) en cas de panne d'un switch L3, avec un basculement rapide et transparent pour les clients.

2. Analyse de l'existant
Topologie actuelle :

Switch3 : Routeur pour VLAN10 (192.168.1.254) et VLAN20 (192.168.2.254).

Switch1 et Switch2 : Redondance L2 via MLAG (bond1 pour Machine1, bond2 pour Machine2).
Machine1 (192.168.1.1, VLAN10) et Machine2 (192.168.2.1, VLAN20) : Connectées via bonds LACP.
Problèmes initiaux :
Pas de redondance L3 : Switch3 est un point de défaillance unique.
Duplicatas (DUP!) lors des pings : Résolus après configuration correcte de VRRP (un seul Master répond).
Performance :
Ping Machine1 → Machine2 : Fonctionnel, temps de réponse ~1ms, plus de (DUP!).

3. Cahier des charges

Objectifs :
Implémenter une redondance L3 entre Switch3 et Switch4.
Assurer un basculement en moins de 5s.
Maintenir la compatibilité avec MLAG (Switch1/Switch2).
Exigences :
Utiliser VRRP pour partager une IP virtuelle (VIP) : 192.168.1.254 (VLAN10), 192.168.2.254 (VLAN20).
Switch3 : Master (priorité 200), Switch4 : Backup (priorité 100).

Contraintes :
Switch4 doit avoir une configuration similaire à Switch3.
Minimiser les interruptions lors de l'ajout de Switch4.

4. Conception & choix du matériel

Choix de VRRP :
VRRP est simple, standard (IETF), et adapté à une maquette de petite taille.

Alternative : OSPF (trop complexe pour ce besoin).
Matériel :
Switch4 : Même modèle que Switch3 (Cumulus Linux).
Ports : swp1 (Switch1), swp2 (Switch2), VLAN10 (192.168.1.253), VLAN20 (192.168.2.253).
Configuration VRRP :
VIP : 192.168.1.254 (VLAN10), 192.168.2.254 (VLAN20).
Switch3 : Priorité 200 (Master).
Switch4 : Priorité 100 (Backup).
Intervalle VRRP : 1s, timeout ~3s.

5. Maquette & Tests

switch3

```
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.10.13/24

auto swp1
iface swp1

auto swp2
iface swp2

auto vlan10
iface vlan10 inet static
    address 192.168.1.252/24
    vlan-id 10
    vlan-raw-device bridge
    vrrp 10 192.168.1.254
    priority 200
    vrrp-preempt

auto vlan20
iface vlan20 inet static
    address 192.168.2.252/24
    vlan-id 20
    vlan-raw-device bridge
    vrrp 20 192.168.2.254
    priority 200
    vrrp-preempt

auto bridge
iface bridge
    bridge-vlan-aware yes
    bridge-ports swp1 swp2
    bridge-vids 10 20
    bridge-stp off
cumulus@cumulus:~$
```

switch4

```
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.10.5/24

auto swp1
iface swp1

auto swp2
iface swp2

auto vlan10
iface vlan10 inet static
    address 192.168.1.253/24
    vlan-id 10
    vlan-raw-device bridge
    vrrp 10 192.168.1.254
    priority 100
    vrrp-preempt

auto vlan20
iface vlan20 inet static
    address 192.168.2.253/24
    vlan-id 20
    vlan-raw-device bridge
    vrrp 20 192.168.2.254
    priority 100
    vrrp-preempt

auto bridge
iface bridge
    bridge-vlan-aware yes
    bridge-ports swp1 swp2
    bridge-vids 10 20

cumulus@cumulus:~$
```

1-Interface de gestion (eth0)

Switch3 : address 192.168.10.13/24 – IP de gestion unique.
Switch4 : address 192.168.10.5/24 – IP de gestion distincte.

Rôle : Permet l'accès administratif aux switches.

2. Interfaces physiques (swp1, swp2)

swp1, swp2 : Ports connectés à Switch1 (swp1) et Switch2 (swp2) sur les deux switches.
Rôle : Transportent le trafic VLAN10 et VLAN20 via le bridge.

3. Interfaces VLAN (vlan10, vlan20)

vlan10 :
Switch3 : address 192.168.1.252/24, Switch4 : 192.168.1.253/24.
VIP VRRP : 192.168.1.254 (passerelle partagée pour VLAN10).
vlan20 :
Switch3 : address 192.168.2.252/24, Switch4 : 192.168.2.253/24.
VIP VRRP : 192.168.2.254 (passerelle partagée pour VLAN20).
vlan-id : 10 et 20, associés au bridge (vlan-raw-device bridge).

4. VRRP (Redondance L3)

vrrp 10 192.168.1.254 (VLAN10), vrrp 20 192.168.2.254 (VLAN20) : Définit les VIP partagées.
priority : Switch3 : 200 (Master), Switch4 : 100 (Backup).
Switch3 est préféré, Switch4 prend le relais si Switch3 tombe.
vrrp-preempt : Si Switch3 revient, il reprend son rôle de Master.

5. Bridge

bridge-vlan-aware yes : Active la gestion des VLANs.
bridge-ports swp1 swp2 : Agrège les ports pour le trafic.
bridge-vids 10 20 : Autorise VLAN10 et VLAN20.
bridge-stp off (Switch3) : Désactive STP (MLAG gère les boucles).

Configuration de VRRP via FRR

Choix de la configuration dans /etc/frr/frr.conf

Pour assurer la redondance L3 entre Switch3 et Switch4, nous configurons VRRP (Virtual Router Redundancy Protocol) dans le fichier /etc/frr/frr.conf sur les deux switches. Ce fichier est choisi car FRR (Free Range Routing) est une suite de routage puissante et standard sur Cumulus Linux, permettant une gestion centralisée des protocoles de routage comme VRRP. Les paramètres VRRP (VIP, priorité, etc.) sont identiques à ceux définis dans /etc/network/interfaces pour garantir une cohérence, mais FRR offre une gestion plus fine (logs, timers ajustables).

```
cumulus@cumulus:~$ cat /etc/frr/frr.conf
cat: /etc/frr/frr.conf: Permission denied
cumulus@cumulus:~$ sudo cat /etc/frr/frr.conf
[sudo] password for cumulus:
# default to using syslog. /etc/rsyslog.d/45-frr.conf places the log
# in /var/log/frr/frr.log
log syslog informational

interface vlan10
 vrrp 10
 vrrp 10 advertissement-interval 5000
 vrrp 10 priority 200
 vrrp 10 ip 192.168.1.254

interface vlan20
 vrrp 20
 vrrp 20 advertissement-interval 5000
 vrrp 20 priority 200
 vrrp 20 ip 192.168.2.254
cumulus@cumulus:~$
```

Les logs qui montre le switching

```
2025-04-05T17:32:16.380977+00:00 cumulus vrrpd[969]: [CORE] [VRID 20] [IPv4] Master_Down_Timer expired
2025-04-05T17:32:16.383000+00:00 cumulus vrrpd[969]: [CORE] [VRID 20] [IPv4] Backup -> Master
2025-04-05T17:32:16.386842+00:00 cumulus zebra[928]: Setting interface vrrp4-8-20 (9): protodown off
2025-04-05T17:32:16.387896+00:00 cumulus vrrpd[969]: [CORE] [VRID 20] [IPv4] Refusing to start Virtual Router: Already running
```

Activation du démon vrrpd

Dans le fichier
etc/frr/daemons, nous activons le démon vrrpd en modifiant la ligne correspondante à vrrpd=yes. Ce choix est nécessaire pour que FRR puisse exécuter le protocole VRRP, permettant à Switch3 (Master, priorité 200) et Switch4 (Backup, priorité 100) de partager les VIP (192.168.1.254 pour VLAN10, 192.168.2.254 pour VLAN20) et d'assurer un basculement automatique en cas de panne.

Status des deux switchs :

switch3

```
cumulus@cumulus:~$ net show vrrp
Virtual Router ID                       10
 Protocol Version                       3
 Autoconfigured                         No
 Shutdown                               No
 Interface                              vlan10
 VRRP interface (v4)                    vrrp4-6-10
 VRRP interface (v6)                    None
 Primary IP (v4)                        192.168.1.252
 Primary IP (v6)                        ::
 Virtual MAC (v4)                       00:00:5e:00:01:0a
 Virtual MAC (v6)                       00:00:5e:00:02:0a
 Status (v4)                            Master
 Status (v6)                            Initialize
 Priority                               200
 Effective Priority (v4)                200
 Effective Priority (v6)                200
 Preempt Mode                           Yes
 Accept Mode                            Yes
 Advertisement Interval                 1000 ms
 Master Advertisement Interval (v4)     1000 ms
 Master Advertisement Interval (v6)     0 ms
 Advertisements Tx (v4)                 3248
 Advertisements Tx (v6)                 0
 Advertisements Rx (v4)                 1
 Advertisements Rx (v6)                 0
 Gratuitous ARP Tx (v4)                 1
 Neigh. Adverts Tx (v6)                 0
 State transitions (v4)                 2
 State transitions (v6)                 0
 Skew Time (v4)                         210 ms
 Skew Time (v6)                         0 ms
 Master Down Interval (v4)              3210 ms
 Master Down Interval (v6)              0 ms
 IPv4 Addresses                         1
```

switch4

```
cumulus@cumulus:~$ net show vrrp
Virtual Router ID                       10
 Protocol Version                       3
 Autoconfigured                         No
 Shutdown                               No
 Interface                              vlan10
 VRRP interface (v4)                    vrrp4-6-10
 VRRP interface (v6)                    None
 Primary IP (v4)
 Primary IP (v6)                        ::
 Virtual MAC (v4)                       00:00:5e:00:01:0a
 Virtual MAC (v6)                       00:00:5e:00:02:0a
 Status (v4)                            Backup
 Status (v6)                            Initialize
 Priority                               100
 Effective Priority (v4)                100
 Effective Priority (v6)                100
 Preempt Mode                           Yes
 Accept Mode                            Yes
 Advertisement Interval                 1000 ms
 Master Advertisement Interval (v4)     1000 ms
 Master Advertisement Interval (v6)     0 ms
 Advertisements Tx (v4)                 132
 Advertisements Tx (v6)                 0
 Advertisements Rx (v4)                 3627
 Advertisements Rx (v6)                 0
 Gratuitous ARP Tx (v4)                 3
 Neigh. Adverts Tx (v6)                 0
 State transitions (v4)                 7
 State transitions (v6)                 0
 Skew Time (v4)                         600 ms
 Skew Time (v6)                         0 ms
 Master Down Interval (v4)              3600 ms
 Master Down Interval (v6)              0 ms
 IPv4 Addresses                         1
```

switch3 (hs) switch4 devient master

```
cumulus@cumulus:~$ net show vrrp
Virtual Router ID                       10
 Protocol Version                       3
 Autoconfigured                         No
 Shutdown                               No
 Interface                              vlan10
 VRRP interface (v4)                    vrrp4-6-10
 VRRP interface (v6)                    None
 Primary IP (v4)                        192.168.1.253
 Primary IP (v6)                        ::
 Virtual MAC (v4)                       00:00:5e:00:01:0a
 Virtual MAC (v6)                       00:00:5e:00:02:0a
 Status (v4)                            Master
 Status (v6)                            Initialize
 Priority                               100
 Effective Priority (v4)                100
 Effective Priority (v6)                100
 Preempt Mode                           Yes
 Accept Mode                            Yes
 Advertisement Interval                 1000 ms
 Master Advertisement Interval (v4)     1000 ms
 Master Advertisement Interval (v6)     0 ms
 Advertisements Tx (v4)                 174
 Advertisements Tx (v6)                 0
 Advertisements Rx (v4)                 4242
 Advertisements Rx (v6)                 0
 Gratuitous ARP Tx (v4)                 4
 Neigh. Adverts Tx (v6)                 0
 State transitions (v4)                 8
 State transitions (v6)                 0
 Skew Time (v4)                         600 ms
 Skew Time (v6)                         0 ms
 Master Down Interval (v4)              3600 ms
 Master Down Interval (v6)              0 ms
 IPv4 Addresses                         1
```

Virtual Router ID (10 et 20) : Identifie les groupes VRRP par VLAN (10 pour 192.168.1.0/24, 20 pour 192.168.2.0/24).
Priorité (200 vs 100) : Switch3 est Master (priorité élevée), Switch4 est Backup. En cas de panne, Switch4 devient Master après le Master Down Interval (3.6s).
Adresse IP Virtuelle : 192.168.1.254 (VLAN10) et 192.168.2.254 (VLAN20) servent de passerelles uniques pour les machines, même après basculement.
Multicast VRRP (224.0.0.18) : Les paquets VRRPv3 sont envoyés périodiquement (1s) pour vérifier l'état des routeurs.

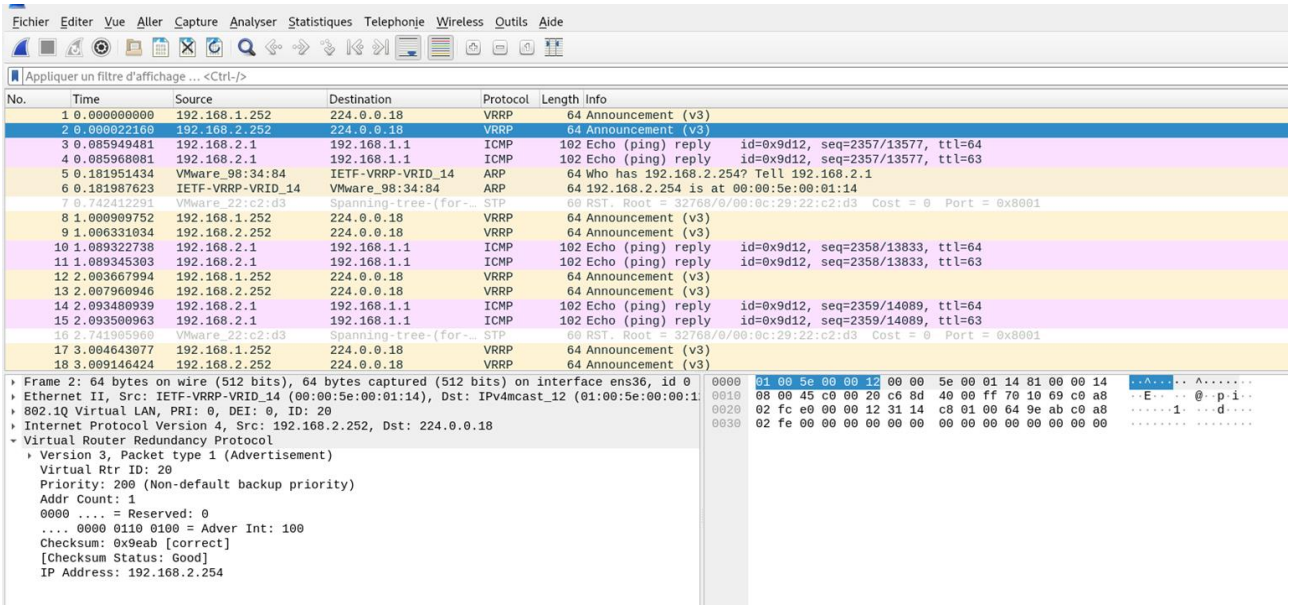
MAC Virtuelle (00:00:5e:00:01:0a) : Permet aux machines de communiquer avec la passerelle sans reconfiguration, même après basculement.

Test 1 : Ping normal :

```
root@debian:~# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=63 time=2.66 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=63 time=3.09 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=63 time=2.65 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=63 time=2.86 ms
^C
--- 192.168.2.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 2.653/2.816/3.093/0.179 ms
root@debian:~#
```

Échange des Annonces VRRP entre Routeurs

Paquets Multicast VRRPv3 : Les routeurs Switch3 (Master) et Switch4 (Backup) échangent périodiquement (toutes les 1 000 ms) des annonces via l'adresse multicast 224.0.0.18, contenant leur priorité et état. Contrôle de l'État : Switch3 (priorité 200) envoie des annonces pour affirmer son rôle Master. Si Switch4 (priorité 100) ne reçoit plus ces annonces pendant 3 600 ms (Master Down Interval), il devient Master.
Gratuitous ARP : Lors d'un basculement, le nouveau Master envoie une trame ARP non sollicitée pour mettre à jour la table MAC des commutateurs avec l'adresse virtuelle (00:00:5e:00:01:0a).
Transparence Réseau : Les machines utilisent toujours l'IP virtuelle (192.168.1.254) comme passerelle, ignorant le basculement physique entre Switch3 et Switch4.
Preuve via Wireshark : Les annonces VRRP sont visibles sur l'interface vlan10 avec le filtre vrrp, montrant les priority et Advertisement Interval.



Conclusion

Ce projet nous a permis de renforcer nos compétences en administration réseau et de collaborer efficacement autour d'une infrastructure complète. Nous avons appris à configurer des services critiques comme NFS, à mettre en place des VLANs, du routage statique et des mécanismes de redondance (VRRP, agrégation de liens). Les tests de panne nous ont permis de mieux comprendre la haute disponibilité et la résilience réseau. Travailler à deux nous a aidés à partager les tâches, à apprendre l'un de l'autre et à structurer notre travail. Cette expérience nous a préparés concrètement aux défis réels du métier d'administrateur système et réseau.