

Mission 1 : Installation et Configuration des systèmes

PRE-REQUIS

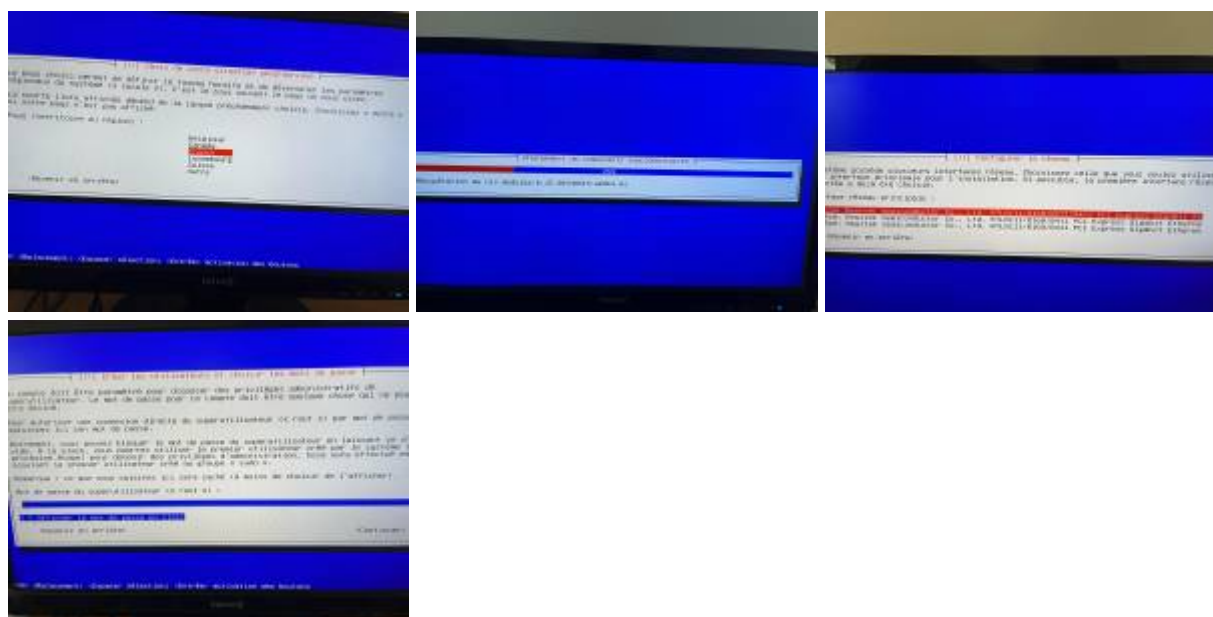
Dans cette nouvelle année , nous allons fonctionner sur nos machines (Routeur et Serveur) en utilisant **Debian 13** pour le routeur et **Proxmox** pour le serveur .
Proxmox est un outil qui nous permettra de créer des VM et des Conteneurs . L'année 2025-2026 , nous allons bosser pour la Zone **Océanie**

I) CONFIGURATION DU ROUTEUR

Un routeur est un équipement qui permet d'interconnecter plusieurs réseaux , elle est connue pour avoir la possibilités d'avoir plusieurs interfaces réseau .

Pour l'installation de notre routeur nous allons utiliser l'OS **Debian 13**

1.1 Procédure d'Installation



1.2 Configuration du Routeur dans le fichier /etc/network/interfaces

```
auto lo
iface lo inet loopback

# The primary network interface
auto enp2s0
iface enp2s0 inet static
    address 172.31.208.254
```

```
netmask 255.255.0.0
gateway 172.31.0.1
dns-nameservers 8.8.8.8

auto enp4s0
iface enp4s0 inet static
address 10.31.211.254
netmask 255.255.252.0
dns-nameservers 8.8.8.8

auto enp5s0
iface enp5s0 inet static
address 10.31.219.254
netmask 255.255.252.0
dns-nameservers 8.8.8.8
```

1.3. Configuration dans le /etc/rc.local

```
#!/bin/sh -e

echo "nameserver 8.8.8.8" > /etc/resolv.conf
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -j MASQUERADE
```

```
apt install net-tools # Installe un ensemble d'outils réseau classiques,
dont ifconfig et netstat
```

NB:

Après avoir terminer les configurations nous devons , premièrement télécharger **Iptables** avec la commande :

```
apt install iptables
```

Et attribuer les droits au fichiers **/etc/rc.local** afin qu'il puisse s'exécuter au démarrage .

```
chmod +x /etc/rc.local
```

II) CONFIGURATION DU SERVEUR

Un serveur est un ordinateur puissant possédant les mêmes composants qu'un poste classique, mais avec une grande capacité de stockage. Il permet de stocker et partager des fichiers ou applications afin que plusieurs utilisateurs puissent accéder aux mêmes données de manière centralisée.

Notre réseau a besoin d'un routeur, d'un switch et d'un serveur pour fonctionner et être accessible à distance. Nous disposons de l'adresse IP 10.31.208.0/22.

L'adresse IP du serveur est 10.31.208.1

Masque : 255.255.252.0

Passerelle par défaut : 10.31.211.254 (routeur)

Le serveur utilise également un DNS pour la résolution de noms de domaine : 8.8.8.8 (Google).

Installation des outils réseaux

Sur Proxmox (basé sur Debian), certains outils comme ifconfig ne sont pas présents par défaut. Nous les avons installés avec :

```
apt install net-tools # Installe un ensemble d'outils réseau classiques, dont ifconfig et netstat
```

Configuration avec /etc/rc.local

Nous avons testé une configuration réseau rapide via le fichier /etc/rc.local :

```
ifconfig ens2f0 10.31.208.1/22 up
ifconfig ens2f1 10.31.216.1/22 up

route add default gw 10.31.211.254
echo "nameserver 8.8.8.8" >> /etc/resolv.conf
exit 0
```

Cette méthode n'était pas fonctionnelle. Pour une configuration permanente, nous utilisons /etc/network/interfaces.

Configuration persistante avec /etc/network/interfaces

Voici la configuration réseau propre pour Proxmox :

```
auto lo iface lo inet loopback
Interface principale du serveur

auto ens2f0
iface ens2f0 inet static
address 10.31.208.1
netmask 255.255.252.0
gateway 10.31.211.254
dns-nameservers 8.8.8.8

Interface secondaire (réseau secondaire)

auto ens2f1
iface ens2f1 inet static
address 10.31.216.1
netmask 255.255.252.0

Interfaces non utilisées

iface ens2f2 inet manual
```

```
iface ens2f3 inet manual
iface eno1 inet manual
iface eno2 inet manual
```

Chargement des autres configurations réseau

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

Vérification

Après sauvegarde et redémarrage du service réseau ou du serveur, on peut vérifier la configuration avec :

```
ping 10.31.211.254
ping 8.8.8.8
```

Résultat :

ens2f0 IP principale : 10.31.208.1/22

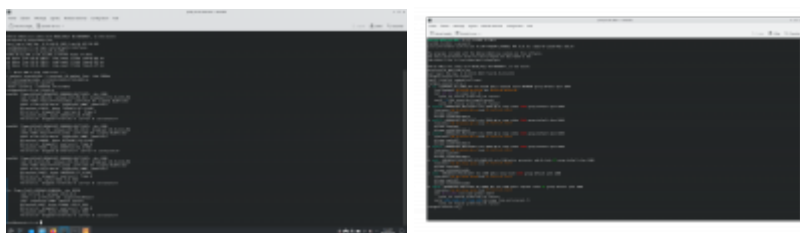
ens2f1 IP secondaire : 10.31.216.1/22

DNS opérationnel (Google 8.8.8.8)

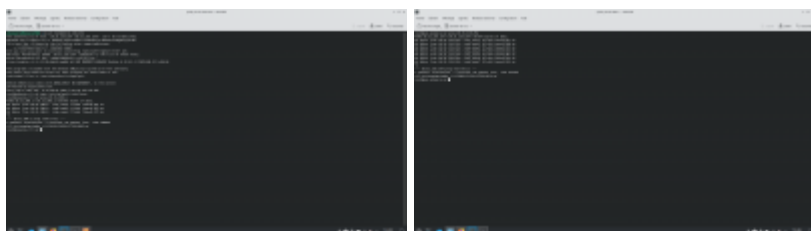
Configuration stable et persistante sur Proxmox

III) PRODUCTIONS ATTENDUES

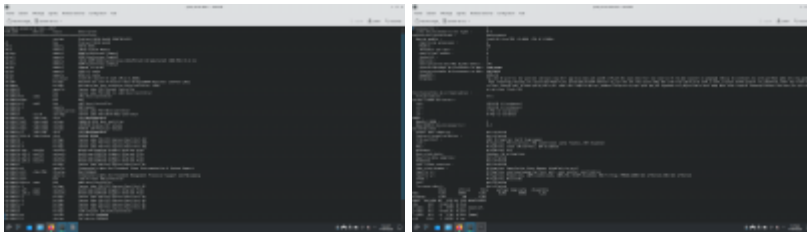
1 Configurations IP des deux machines



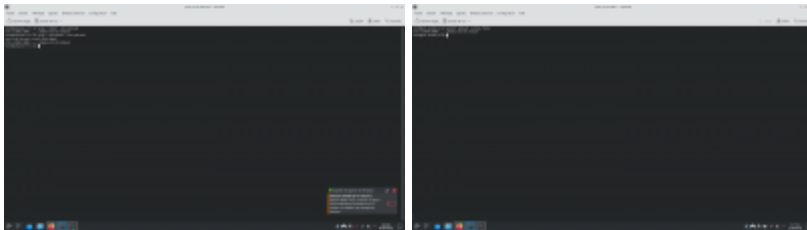
2 test de ping entre serveur et routeur



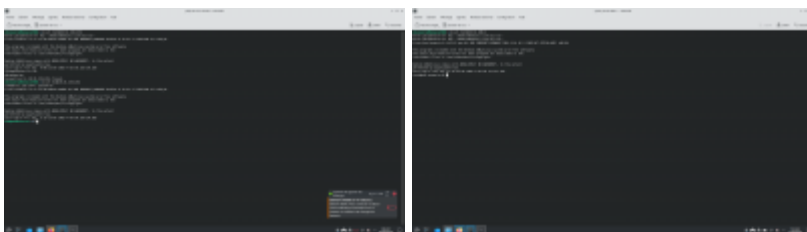
3- Caractéristiques des machines



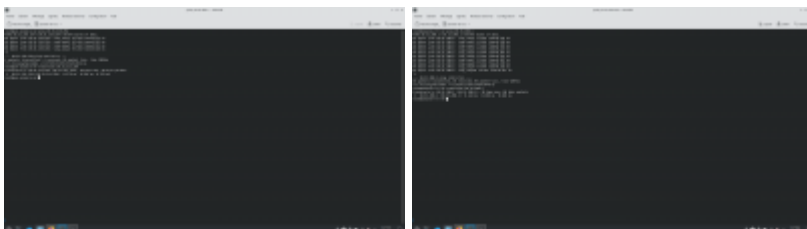
4- Compte des machines créées des machines



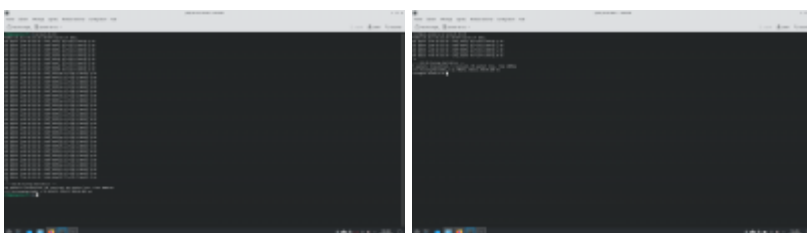
5- test aux machines par ssh



6- communication ip entre clients et serveur



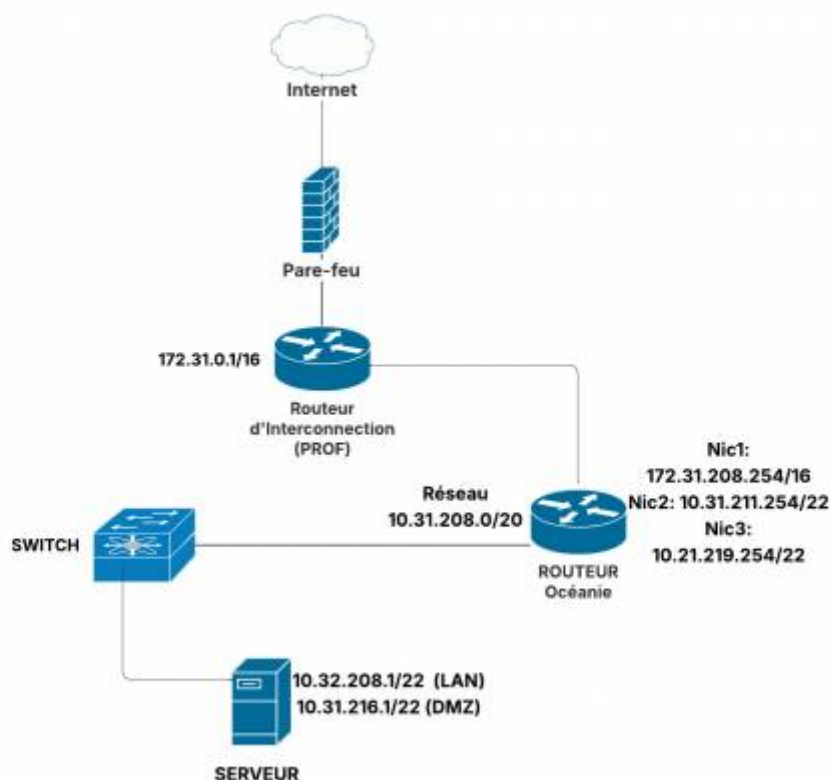
7- preuve connectivité internet



IV) MAQUETTE REPRESENTATIVE DE NOTRE RESEAU

Explication

Concernant notre réseau Océanie , Notre routeur est connecté au réseau SIO via un câble RJ45 , ensuite le Switch est connecté au routeur via 2 câbles Ethernet et le Serveur est connecté au Switch via 2 câbles .



V) RECOMMANDATIONS DE L'ANSSI

1.1 Mise en Place

Les recommandations de l'ANSSI sont très importantes pour sécuriser nos appareils , postes , machines et de les préserver de toutes attaques malveillantes .

M1.Désactiver les comptes utilisateurs non utilisés

Tous nos utilisateurs sont utilisés .Avec la commande nous pouvons vérifier tous les users :

```
cat /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
nobody:x:65534:65534:/var/lib/nobody:/usr/sbin/nologin
```

M2. Utiliser des Mots de passe Robuste

Nous allons modifier les mots de passe de nos utilisateurs , en utilisant la commande suivante :

```
passwd username
```



```
#!/bin/bash
# Script de mise à jour automatique
LOGFILE="update_log_$(date +%d-%m-%Y).log"
# Met à jour la liste des paquets
apt-get update -y

# Met à jour les paquets installés
apt-get upgrade -y

# Nettoie les paquets inutiles
apt-get autoremove -y
// Après nous attribuons les droits avec chmod +x /etc/update.sh
```

```
nano /etc/crontab
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR
sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
0 8 */2 * * /etc/update.sh >> /var/log/update-system.log 2>&
```

M8. Protéger les mots de passe Stockés

Modifions le fichier **/etc/pam.d/common-password** :

```
password required pam_unix.so obscure yescrypt rounds=11 # Directive ajoutée
```

```
# here are the per-package modules (the "Primary" block)
password [success=1 default=ignore] pam_unix.so obscure yescrypt
# here's the fallback if no module succeeds
password requisite pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password required pam_permit.so
# and here are more per-package modules (the "Additional" block)
# end of pam-auth-update config
password required pam_unix.so obscure yescrypt rounds=11
```

M9. Réduire la surface d'attaque des services réseau

La commande suivante permet de lister l'ensemble des services en écoute sur le réseau :

```
netstat -nautp
```

```
root@oceanie-rtr:~# netstat -nautp
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale Adresse distante Etat PID/Program name
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 795/sshd: /usr/sbin
tcp 0 196 172.31.208.254:22 18.187.20.102:57527 ESTABLISHED 2347/sshd-session:
tcp 0 0 18.31.219.254:22 18.187.20.156:60152 ESTABLISHED 2313/sshd-session:
tcp6 0 0 :::22 :::* LISTEN 795/sshd: /usr/sbin
root@oceanie-rtr:~#
```


From:

<https://sisr2.beaupeyrat.com/> - **Documentations SIO2 option SISR**

Permanent link:

https://sisr2.beaupeyrat.com/doku.php?id=sisr2-oceanie:installation_et_configuration_des_systemes

Last update: **2025/09/25 20:47**

