**Machines Virtuelle Linux:**

## **Réglages:**

- Réseaux : NAT

- Coeur : 1

- RAM : 2

- DD : 20Go

**Installer le ssh:** sudo apt install openssh → sudo systemctl status ssh →

sudo systemctl enable ssh → sudo systemctl start ssh.

**Activer le copier/coller et dossier échangés:** Activer bidirection dans le réglage Devices (VirtualBox)→ Insérer l’image CD Additions guest→ Aller dans la VM→

Cliquer sur le disque dans la barres de favoris→ Clique droit dans la fenêtre pour ouvrir le terminale→ ls→ sudo ./VBoxLinuxAdd…..→ sudo reboot

**Activer un dossier partagé:** Sélectionner la VM concerner→ Partage de dossier→ Ajoute un document→ Montage Automatique→ Aller dans la VM→ Document→ Sélectionner le dossier qui apparaît en SF\_….→ Ouvrir un terminale→ sudo usermod -a -G vboxsf nom du User→ (optionnel)sudo mount -t vboxsf NomDuDossier /media/sf\_Nom du dossier

**Manipulation:**

- Simulation remplissage de disk : dd if= /dev/zero of=~ /Téléchargement/un-fichier

- Redémarrer la machine: Touche hôte + suppr

- Accéder aux options avancées au redémarrage = Maintenir esc

- Ctrl+alt+T = Ouvre un terminale

**Créer un disque supplémentaire:** Sélectionner la VM→ Configuration→ Stockage→ Contrôleur SATA→ Ajouter un disk dur→ Créer (ou sélectionner un existant) → Type→ Taille fixe→ … → Insérer le disque.

- **Le /tmp** = Contient les fichiers temporaire effacé à chaque reboot; Rajouter /tmp en fin de ligne un dossier à déplacer dedans: cp -R 1 copy\_1 /tmp

**Réglages réseaux:**  VBox→ Config→ Activer carte réseaux→ Câbles branchées→ Réseaux interne→ paramètres→ Réseaux→ Paramètres→ IPV4→ Manuel→ 192.168.1.10 (exemple)→ Appliquer→ Reboot.

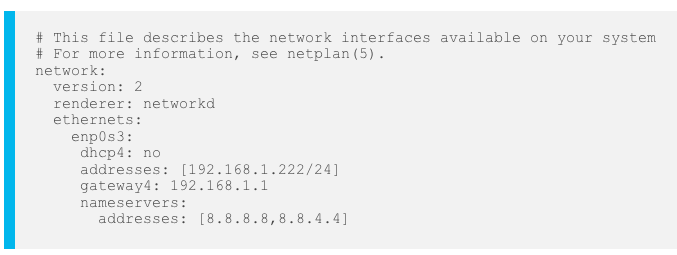
**Réglages réseaux en ligne de commandes:** En root→ nano /etc/network/interface (ou /etc/NetworkManager/system-connection/Connexion filaire)→ ENPOS3→ ip -a→ Commenter les enpos3→ Interface Enpos3 inet static→ address 192.168.1.10 (exemple)→ masque à renseigner.

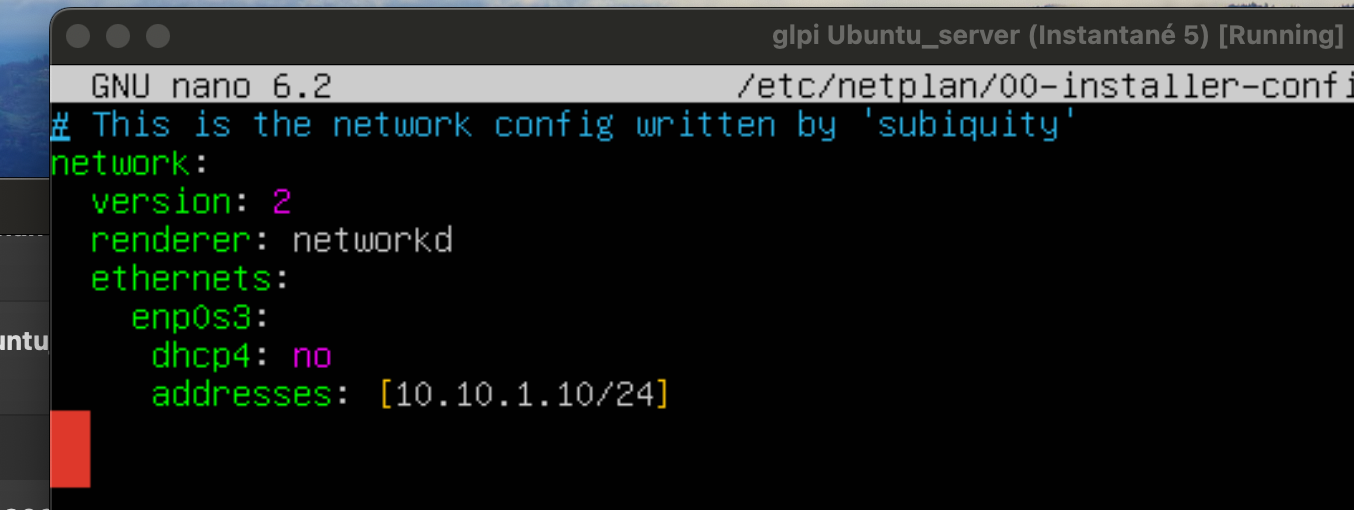
**Attribuer une IPV4 en ligne de commande**

sudo ip addr add 192.168.0.150/24 dev enp0s3.

**IP Linux si /interfaces est vide**. Editer /etc/netplan/\*.yaml (Cf group support Linux\_DHCP)

Sous l’interfaces rentrer les infos:





Save & quit puis taper: sudo netplan apply

**Ajout d’une IPV6:**

**sudo ip -6 addr add 2001:db8::1/64 dev enp0s3** enp0s3 suivant interface réseaux

Pour contrôler l’IPV6:

ip -6 addr show enp0s3 | grep inet6

**Configuration routeur Linux:**

Pour fixer une IPV4 et IPV6:

auto ens4

17iface ens4 inet static

18 address 10.0.0.10

19 netmask 255.255.255.0

20 gateway 10.0.0.1

21

22iface ens4 inet6 static

23 address fdd3:9430:138e::10

24 netmask 64

25 gateway fdd3:9430:138e::1

**Activer l’IP Forwarding:**

* sysctl net.ipv4.ip\_forward : consultation de l'état du routage IPv4

Activer de façon temporaire:

sysctl -w net.ipv4.ip\_forward=1

ipv6:

sysctl -w net.ipv6.conf.all.forwarding=1

Activation permanente:

Modifier le fichier /etc/sysctl.conf et désactiver la ligne:

net.ipv4.ip\_forward = 1

idem pour l’IPV6:

net.ipv6.conf.all.forwarding=1

Recharger la config:

sysctl -p /etc/sysctl.conf

**Fixer une route:**

(Façon temporaire)En ligne de commande:

route add -net 172.17.250.0/24 gw 172.17.10.141 dev eth0

gw = passerelle par défaut

(Façon permanente) Editer le fichier /etc/network/interfaces

en bas du fichier rajouter:

# static route

up route add -net 172.17.250.0/24 gw 172.17.10.141 dev eth0

* ip route : affiche table de routage ipv4
* ip -6 route : affiche table de routage ipv6
* ip route add 192.168.128.0/17 via 10.0.0.2 : ajout d'une route
* ip route add default via 10.0.0.254 : ajout d'une passerelle par défaut

Pour une configuration persistante => /etc/network/interfaces

**Connection ssh:**

*;* ***Cf support group “Config Debian”***

**Pour l’exemple deux machine:**

**-** Ubuntu toto et mot de passe 12345 ip:10.1.1.1

- Ubuntu zorro et mot de passe azerty ip:10.1.1.2

Les machine sont en carte réseaux interne et une en NAT

Régler les adresse ip (/etc/network/interfaces)→ systemctl restart networking (ou en GUI)

- Sur la machine toto, Si besoin installer le paquet openssh-server

- nano /etc/ssh/sshd\_config→changer port par défaut (55885 ex) → rajouter ligne en dessous UsePAM yes AllowUsers toto; deuxième ligne PermitrootLogin no→ Save & quit→ sudo systemcyl restart sshd.

- Sur la machine zorro, Si besoin installer le paquet openssh-server

- nano /etc/ssh/sshd\_config→changer port par défaut (55885 ex) → rajouter ligne en dessous UsePAM yes AllowUsers zorro; deuxième ligne PermitRootLogin no→ Save & quit→ sudo systemctl restart sshd.

- Pour se connecter à zorro depuis toto:

ssh zorro@10.1.1.2→ azerty→ su - (pour se connecter en admin)

- Pour se connecter à toto depuis zorro:

ssh toto@10.1.1.1→ 12345→ su - (pour se connecter en admin)

**Avec Clé publique sans mot de passe:**

Zorro est devenu le serveur et toto le client. Le client initie la communication ssh et devra générer une clé cryptée et la partager à zorro

- **Sur le client:** ssh-keygen→ ssh-copy-id zorro@10.1.1.2

- **Sur le serveur:** Editer /etc/ssh/sshd\_config→ Décommenter la ligne PubKeyAuthentication yes→ PasswordAuthentication no

Avec cette configuration, toto n’a plus besoin de s’authentifier avec le mot de passe.

**Serveur DHCP:**

- Installation: sudo apt install isc-dhcp-server

- Choix de l’interface du DHCP request: sudo nano /etc/default/isc-dhcp-server (le copier si problème en .ori) → dans la ligne: INTERFACES=”enp0s3” (si plusieurs eth1,eth2,etc...)→ fermer et enregistrer.

- Configurer une adresse static au server. CLI ou GUI possible

- Configuration: sudo nano /etc/dhcp/dhcpd.conf→ dans le premier paragraphe rentrer le nom de domaine si besoin→ a la fin du fichier écrire: # Notre configuration pour le réseau 172.20.0.0 (exemple)

subnet 172.20.0.0 netmask 255.255.255.0 {

range 172.18.0.2 172.18.0.100;

option domain-name-servers 8.8.8.8;( pas nécessaire en réseaux interne)

option domain-name "reseau.lan";(Pas nécessaire en réseaux interne)

option routers 172.18.0.1; (Pas nécessaire en réseaux interne)

option broadcast-address 172.20.255.255;

default-lease-time 600;

max-lease-time 7200;

}

-Reboot la machine ou sudo systemctl start isc-dhcp-server.service

$ sudo systemctl enable isc-dhcp-server.service

- Configurer une adresse static à un client: Configuration: sudo nano /etc/dhcp/dhcpd.conf→ en dessous du réseaux précédemment inscrit→

host client1\_ubuntu {

hardware ethernet 08:00:27:bb:40:6b; (@MAC du poste client)

fixed-address 172.20.0.10; (exemple d’@IP)

}

Enregistrer et quitter le fichier

- Reboot ou sudo systemctl restart isc-dhcp-server.service

- Permettre le DHCP au port 67: sudo ufw allow 67/udp

- **Configurer les client DHCP**: sudo nano /etc/network/interfaces→

auto enp0s3

iface enp0s3 inet dhcp

Sauvegarder et quitter le fichier

Reboot ou sudo systemctl restart networking

Il est possible de le faire en utilisant le GUI: Paramètres→ Réseaux→ Onglet IPV4→ Cocher Automatique (DHCP)→ Appliquer.

- Pour tester:

sudo dhclient -r -v enp0s3 (pour casser le bail)

sudo dhclient -v enp0s3 (Pour demander une adresse IP)

**Service DNS:**

**En local:**

Exemple:

IP du serveur: 192.168.1.10

Nom de Domaine: labo.lan

Réseau privé: 192.168.0.0/24

Mettre à jour l’OS→ sudo apt install -y bind9 bind9utils bind9-doc dnsutils.

Editer le fichier /etc/bind/named.conf.options:

Rajouter au dessus de option:

acl internal-network {

192.168.0.0/24;

};

Dans option rajouter:

options {

directory "/var/cache/bind";

allow-query { localhost; internal-network; };

allow-transfer { localhost; };

option forwarders { 8.8.8.8; };

recursion yes;

Editer le fichier /etc/bind/named.conf.local

Rajouter en dessous de //include……:

zone "labo.lan" IN {

type master;

file "/etc/bind/forward.labo.lan";

allow-update { none; };

};

zone "0.168.192.in-addr.arpa" IN {

type master;

file "/etc/bind/reverse.labo.lan";

allow-update { none; };

};

Save &quit.

Dans /etc/bind copier le fichier db.local en forward.labo.lan

Editer le nouveaux fichier.

cf photo en bas!

Renseigner la ligne:

$TTL 604800

@ IN SOA primary.labo.lan. root.primary.labo.lan. (

Ensuite renseigner les infos d’adresse du dns et du NS:

;Name Server Information

@ IN NS labo.lan.

;IP address of Domain Name Server(DNS)

Primary IN A 192.168.1.10

@ IN AAAA ::1

Créer les nom d’hôtes:

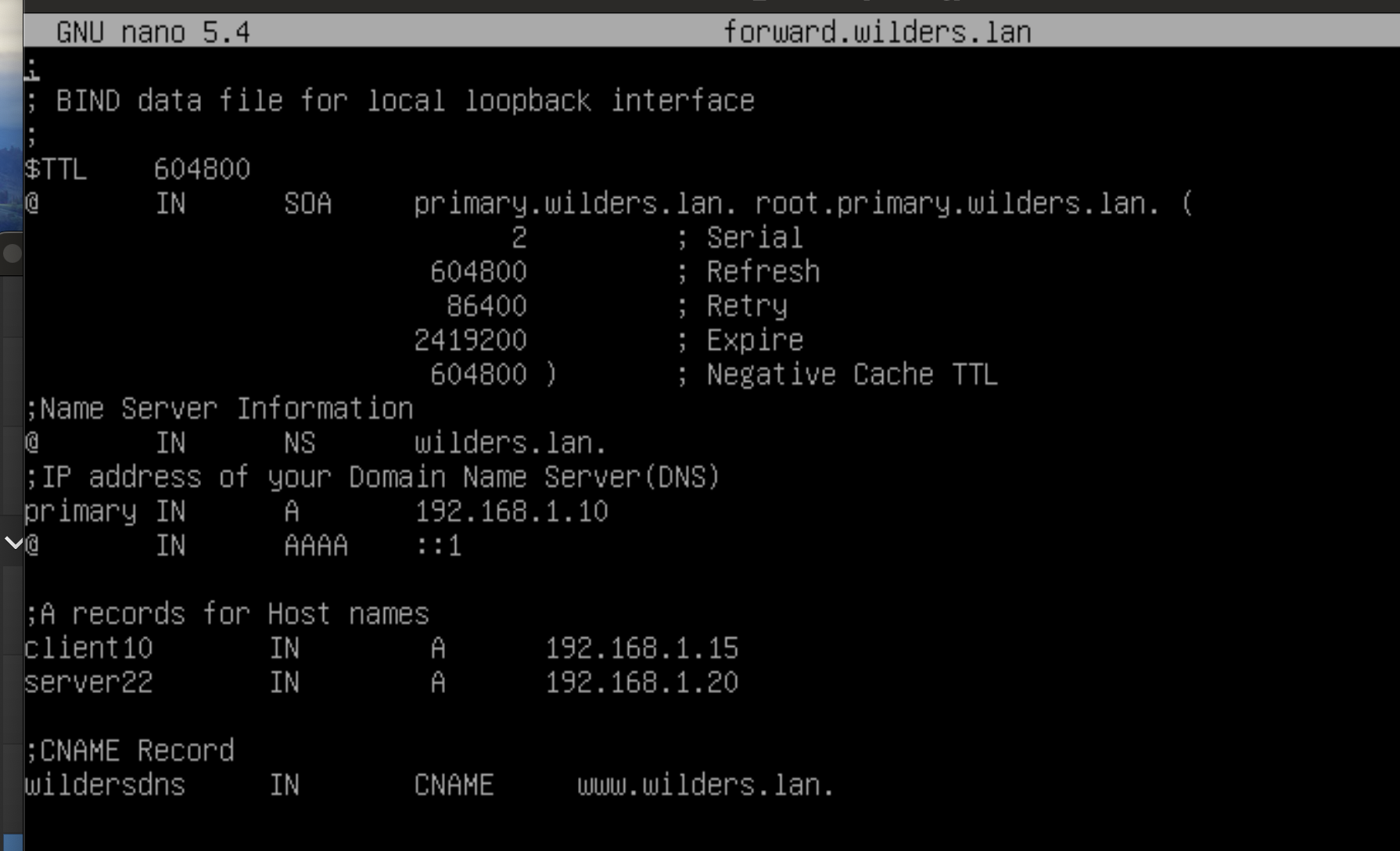
;A Record for Host names

client10 IN A 192.168.1.15

server22 IN A 192.168.1.20

Créer le CNAME:

;CNAME Record

labodns IN CNAME [www.labo.lan](http://www.labo.lan)

save&quit

**Pour le reverse**, copier le fichier db.127 dans /etc/bind en reverse.labo.lan

Editer le nouveaux fichier et renseigner:

@ IN SOA labo.lan. root.labo.lan.

Ainsi que:

@ IN NS primary.labo.lan.

primary IN A 192.168.1.10

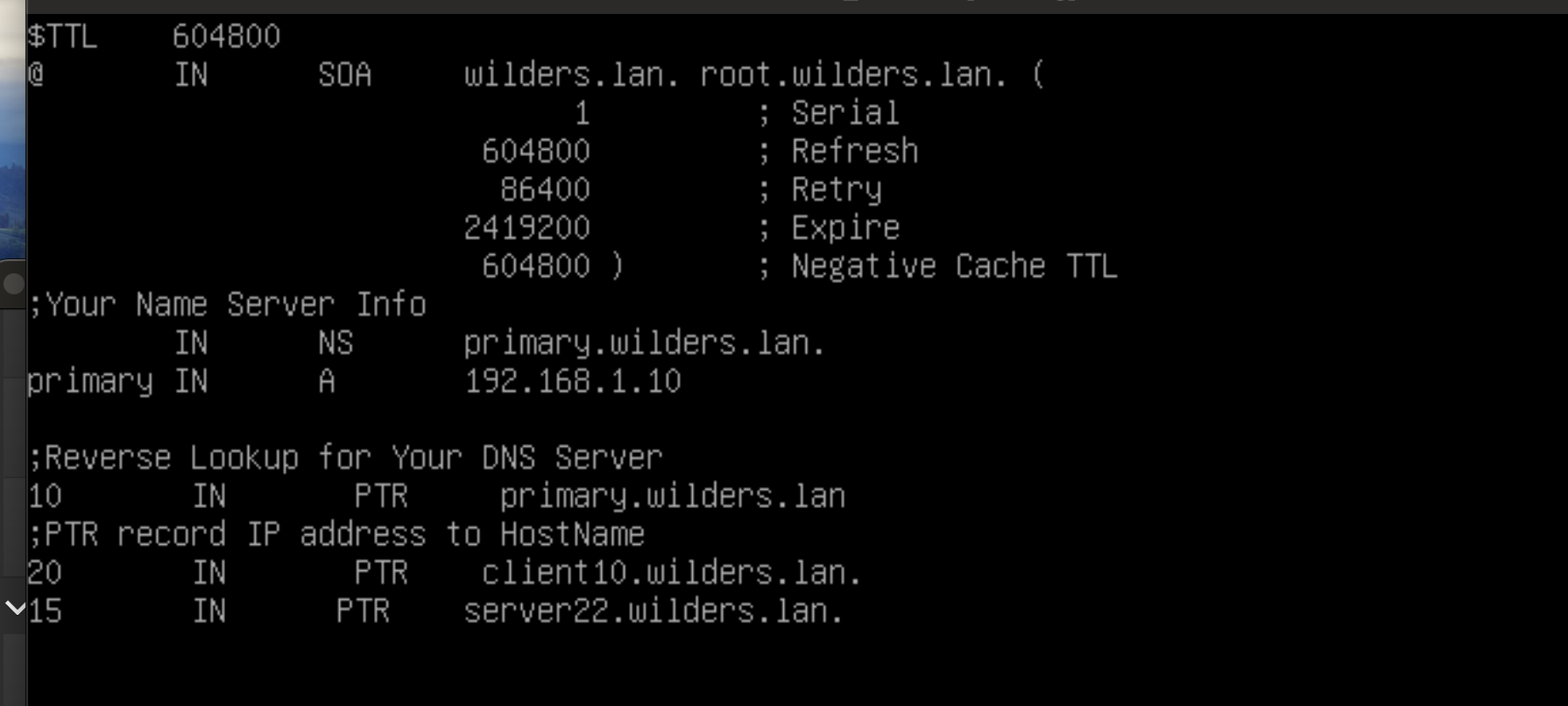
;Reverse lookup for DNS server

10 IN PTR primary.labo.lan.

;PTR record IP address to HostName

20 IN PTR client10.labo.lan

15 IN PTR server22.labo.lan

(Client 10 et server22 sont des clients du réseaux du labo)

Save & quit

**Pour que le DNS écoute en IPV4**, éditer le fichier /etc/default/named et compléter:

OPTIONS="-u bind -4"

Save & quit

**Pour démarrer et permettre le démarrage à chaque allumage de la machine:**

sudo systemctl start named

sudo systemctl enable named

**Pour contrôler le statut**:

sudo systemctl status named

**Pour autoriser le port 53 s’il y a un pare-feu**:

sudo ufw allow 53

**Pour valider la syntaxe des fichiers:**

sudo named-checkconf /etc/bind/named.conf.local

Si tout est bon, le shell revient à l’invite de commande sans afficher d’erreur.

**Pour vérifier la syntaxe des fichiers forward & reverse:**

sudo named-checkzone labo.local /etc/bind/forward.wilders.lan

sudo named-checkzone labo.local /etc/bind/forward.wilders.lan

**Renseigner les clients de l’adresse du DNS.**

GUI ou éditer le fichier /etc/resolv.conf→nameserver 192.168.0.10

**Appeler le dns depuis le client pour contrôler sa présence:**

dig primary.labo.lan

En reverse:

dig -x 192.168.0.10

Avec nslookup

nslookup primary.labo.lan

nslookup client10.labo.lan

nslookup labodns.labo.lan

nslookup 192.168.1.10

**Installer PowerShell**

wget<https://packages.microsoft.com/config/ubuntu/22.04/packages-microsoft-prod.deb> -O packages-microsoft-prod.deb

**dpkg** -i packages-microsoft-prod.deb

**sudo** **apt-get update**

**sudo** **apt-get install** -y powershell

Pour lancer PowerShell:

pwsh

**Serveur ZIMBRA:**

Git hub projet3 sem15

**Serveur Zabbix:**

Github projet3 sem13

**Stockage RAID:**

Exercices SEM12

**Commandes:**

- sudo cfdisk = Affiche la partition disk

- df = Affiche taux d’utilisation des partitions

- df -h = Affiche en unités lisible

- watch = Voir en temps réel

- watch df -ht ext4 = Vision en temps réel de ext4

- watch -h = Taille en unités lisible (human readables)

- watch df -t ext = Affiche partition de type ext4

- dd if of = Copier une info dont la source est if dont la source est of

- tar -xvzf NomDuFichier.tar = Décompresser un fichier tar

Ajouter une adresse ip sur linux = ip addr add <@IP/ mask> dev nom\_d’interface

Ajouter route par défaut = ip route add default via 192.168.1.100

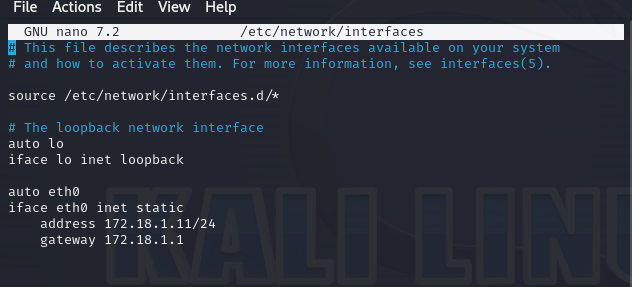
Enregistrer dans le fichier /etc/interface

**Debian:**

Version = iso 12.00netsink

Réglages = Install→ Français→ Grub: oui→ miroir avec (france)→ deb.debian.org

**@static** = En root→ nano /etc/network/interfaces→ auto enp0s3 iface enp0s3 inet static address 192.0.0.0/24→ enregistrer→ systemctl restart networking→ ip a



**Activer une interface réseau:**

ifup <nomdeL’interface>

**Dhcp ipv6 Debian**

apt radvd

créer fichier /etc/radvd.conf\*

l’éditer

interface enp0s3;

{

AdvSendAdvert on;

Prefix 2001:db8::/64

{

AdvOnLink on;

AdvAutonomous on;

};

};

Sav & redémarrer systemctl restart radvd

**RAID 1 sur le système:**

Github projet3 S10