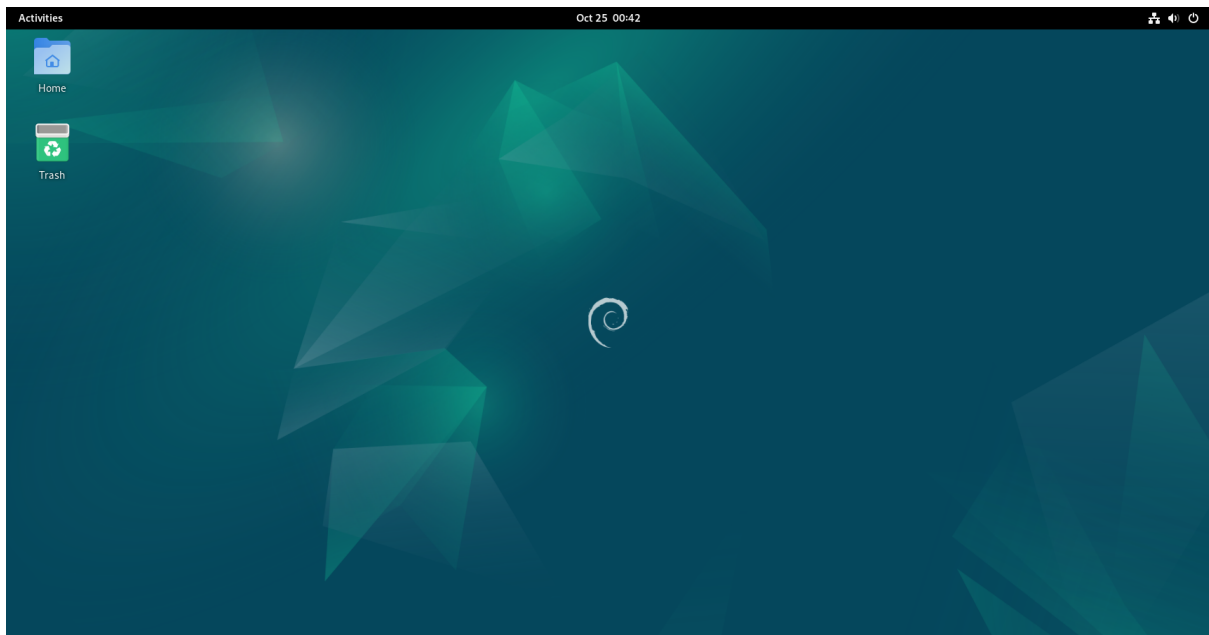


DDWS

JOB 01 : Installer debian avec Interface Graphique



JOB 02 : Installer Apache2 et le lancer

Pour l'installer on va utiliser la commande suivant : `sudo apt install apache2`

```
root@debian:/home/goldroger# sudo apt install apache2
```

Ensuite nous allons installer `ufw` (**Un**complicated **F**irewall) un programme qui va nous permettre de gérer et configurer notre pare-feu.

Pour l'installer on va utiliser la commande suivant : `sudo apt install ufw`

```
root@debian:/home/goldroger# apt install ufw
```

Après l'avoir installé nous allons maintenant autoriser nos port tcp 80 et 443 dans notre pare-feu, pour le faire on va utiliser les suivants :

`sudo ufw allow 80/tcp` et `sudo ufw allow 443/tcp`

```
root@debian:/home/goldroger# sudo ufw allow 80/tcp
Rules updated
Rules updated (v6)
root@debian:/home/goldroger# sudo ufw allow 443/tcp
Rules updated
Rules updated (v6)
root@debian:/home/goldroger#
```

Ensuite nous on va vérifier si `apache2` est bien en cours de processus, pour cela on va utiliser la commande suivante : `sudo systemctl status apache2`

```
root@debian:/home/goldroger# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2023-10-25 00:50:28 CEST; 7min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 30238 (apache2)
      Tasks: 55 (limit: 9452)
     Memory: 18.9M
        CPU: 126ms
    CGroup: /system.slice/apache2.service
            └─30238 /usr/sbin/apache2 -k start
              └─30239 /usr/sbin/apache2 -k start
                └─30240 /usr/sbin/apache2 -k start

Oct 25 00:50:28 debian systemd[1]: Starting apache2.service - The Apache HTTP Server...
Oct 25 00:50:28 debian apachectl[30237]: AH00558: apache2: Could not reliably determine the server's fully
Oct 25 00:50:28 debian systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-16/16 (END)
```

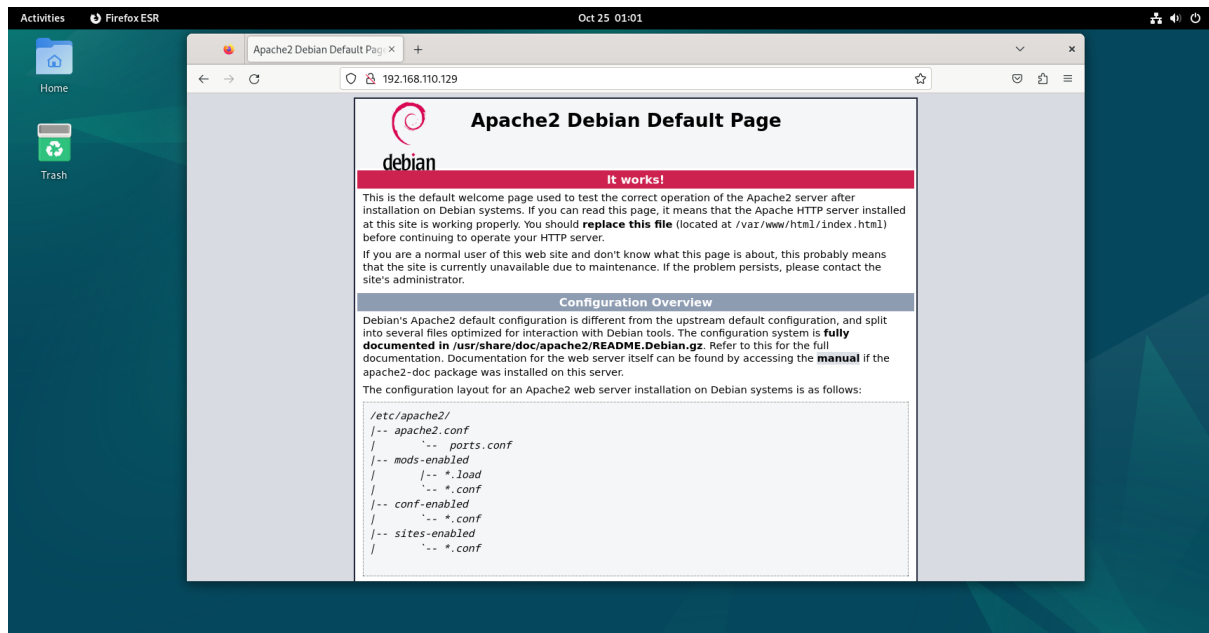
On voit ici qu'il est en `active(running)` qui veut dire qu'il est bien en cours d'exécution

Maintenant, nous allons chercher le IP du notre Serveur, pour cela on va utiliser la commande suivante pour l'afficher : `hostname -I`

```
root@debian:/home/goldroger# hostname -I
192.168.110.129
root@debian:/home/goldroger#
```

Comme on peut le voir ici l'adresse de notre serveur est : `192.168.110.129`

Enfin, maintenant on va le copier et le saisir sur notre navigateur pour y accéder



JOB 03 : Avantages et inconvénients des différents serveurs web

Il existe plusieurs serveurs web populaires, chacun ayant ses propres avantages et inconvénients. Voici une liste de quelques-uns d'entre eux :

1. Apache HTTP Server :

Apache est l'un des serveurs web les plus anciens et les plus utilisés. Il est open source et fonctionne sur la plupart des systèmes d'exploitation, y compris Linux, Unix, Windows, et plus encore.

Avantages : Stabilité, robustesse, modularité grâce à des modules tiers, une grande communauté de soutien.

Inconvénients : Peut être un peu plus lourd en termes de consommation de ressources comparé à certains autres serveurs web.

2. Nginx :

Nginx est un serveur web open source et un serveur proxy inverse. Il est connu pour sa légèreté et sa capacité à gérer de nombreux utilisateurs simultanément.

Avantages : Hautement performant, capable de gérer de nombreuses connexions simultanées, excellent pour servir du contenu statique, supporte les technologies modernes comme WebSocket

Inconvénients : Moins flexible pour le traitement de contenu dynamique par rapport à Apache.

3. Microsoft Internet Information Services (IIS) : IIS est le serveur web de Microsoft, principalement utilisé sur les systèmes d'exploitation Windows Server.

Avantages : Intégration étroite avec d'autres produits Microsoft, support natif pour les technologies Microsoft comme ASP.NET.

Inconvénients : Moins couramment utilisé en dehors des environnements Windows, peut ne pas être aussi performant que certains serveurs web open source.

4. LiteSpeed Web Server : LiteSpeed est un serveur web commercial qui se vante d'être un remplaçant direct d'Apache, offrant une meilleure performance.

Avantages : Haute performance, compatibilité avec les configurations Apache, consommation de ressources relativement faible.

Inconvénients : Licence commerciale avec une version gratuite limitée.

5. Caddy : Caddy est un serveur web open source qui se distingue par sa configuration automatique, la gestion automatique du certificat SSL et son utilisation facile.

Avantages : Configuration automatique, gestion facile des certificats SSL, performances solides.

Inconvénients : Peut ne pas être aussi adapté pour des scénarios complexes ou des configurations très spécifiques.

6. Cherokee : Cherokee est un serveur web open source qui se concentre sur la facilité d'utilisation et la performance.

Avantages : Interface utilisateur graphique conviviale, performances élevées, support pour de nombreuses technologies.

Inconvénients : Moins populaire, donc une communauté de soutien plus petite.

JOB 04 : Mise en place d'un DNS

Pour faire correspondre l'IP de notre serveur a un nom de domaine, tout d'abord on va utiliser la commande suivante pour installer bind9 (logiciel qui fournit des services de serveur DNS) : `sudo apt install bind9`

```
goldroger@debian:~$ sudo apt install bind9
```

Ensuite nous allons procéder à la configuration de notre serveur DNS.

Tout d'abord nous allons editez le fichier de configuration du serveur Bind avec la commande nano : `sudo nano /etc/bind/named.conf.local`

```
goldroger@debian:~$ sudo nano /etc/bind/named.conf.local
```

Dans le fichier de configuration nous allons ajouter la zone de notre domaine, on enregistre puis on quitte :



```
goldroger@debian: ~
GNU nano 7.2 /etc/bind/named.conf.local
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "dnsproject.prepa.com" {
    type master;
    file "/etc/bind/db.dnsproject.prepa.com";
};

[ Read 12 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Nous allons maintenant créer ensuite éditer notre fichier zone avec la commande **nano** suivi du répertoire et le fichier : **sudo nano /etc/bind/db.dnsproject.prepa.com**

```
goldroger@debian:~$ sudo nano /etc/bind/db.dnsproject.prepa.com
```

Ensuite dans notre fichier zone nous on va ajouter les enregistrements de zone, on enregistre puis on quitte.

voir :



```
goldroger@debian: ~
GNU nano 7.2 /etc/bind/db.dnsproject.prepa.com *
$TTL      604800
@         IN      SOA      dnsproject.prepa.com. admin.dnsproject.prepa.com. (
                                2023102501 ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@         IN      NS       dnsproject.prepa.com.
@         IN      A        192.168.110.129

```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line

Maintenant on va redemarrer le service bind9 pour appliquer les modifications avec la commande : **sudo systemctl restart bind9**

```
goldroger@debian:~$ sudo systemctl restart bind9
```

On va maintenant ajouter l'adresse IP du serveur dns dans le fichier /etc/resolv.conf avec la commande nano : **sudo nano /etc/resolv.conf**

```
goldroger@debian:~$ sudo nano /etc/resolv.conf
```

Dans le fichier on va ajouter la ligne **nameserver + ipduserveur** , on enregistre puis on quitte.
nameserver 192.168.110.129

Maintenant essayons de ping sur le nom de domaine : dnsproject.prepa.com

```
goldroger@debian:~$ ping dnsproject.prepa.com
PING dnsproject.prepa.com (192.168.110.129) 56(84) bytes of data.
64 bytes from debian (192.168.110.129): icmp_seq=1 ttl=64 time=0.053 ms
64 bytes from debian (192.168.110.129): icmp_seq=2 ttl=64 time=0.083 ms
64 bytes from debian (192.168.110.129): icmp_seq=3 ttl=64 time=0.063 ms
64 bytes from debian (192.168.110.129): icmp_seq=4 ttl=64 time=0.052 ms
64 bytes from debian (192.168.110.129): icmp_seq=5 ttl=64 time=0.044 ms
64 bytes from debian (192.168.110.129): icmp_seq=6 ttl=64 time=0.065 ms
64 bytes from debian (192.168.110.129): icmp_seq=7 ttl=64 time=0.027 ms
64 bytes from debian (192.168.110.129): icmp_seq=8 ttl=64 time=0.066 ms
64 bytes from debian (192.168.110.129): icmp_seq=9 ttl=64 time=0.033 ms
64 bytes from debian (192.168.110.129): icmp_seq=10 ttl=64 time=0.053 ms
64 bytes from debian (192.168.110.129): icmp_seq=11 ttl=64 time=0.068 ms
^C
--- dnsproject.prepa.com ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10076ms
rtt min/avg/max/mdev = 0.027/0.055/0.083/0.015 ms
goldroger@debian:~$
```

Ça fonctionne, ca ping sur notre IP !!

JOB 05 : Comment obtient-on un nom de domaine public ?

Pour réserver un nom de domaine, le processus implique plusieurs étapes clés qui mettent en jeu différents acteurs du domaine de l'enregistrement des noms de domaine. Voici une procédure détaillée :

1. Sélection du Nom de Domaine : Commencez par choisir un nom de domaine pertinent et représentatif de votre site ou entreprise. Assurez-vous qu'il est facilement mémorisable et qu'il correspond à vos objectifs.

2. Choix du Registraire : Un registraire est un intermédiaire entre le propriétaire du nom de domaine (le registrant) et l'organisme de gestion des noms de domaine (le registre). Choisissez un registraire de confiance, réputé pour ses services et sa conformité aux normes.

3. Vérification de la Disponibilité : Utilisez l'interface en ligne du registraire pour vérifier la disponibilité du nom de domaine souhaité. Respectez les règles de syntaxe et de droits spécifiques à chaque extension.

4. Fourniture des Informations et Documents : Entamez le processus d'enregistrement en fournissant les informations requises. Celles-ci incluent généralement les coordonnées du registrant (vous, en tant que propriétaire du domaine) et du contact administratif. Certains registrars peuvent également demander des documents supplémentaires pour des extensions spécifiques ou pour des raisons de vérification.

5. Paiement de la Transaction : Une fois les détails fournis, procédez au paiement de la transaction. Les registrars acceptent généralement diverses méthodes de paiement. Assurez-vous de suivre les protocoles de sécurité pour protéger vos informations financières.

6. Options de Services Complémentaires : Certains registrars proposent des services complémentaires tels que l'hébergement web, la protection de la vie privée du domaine, des certificats SSL, ou encore la création d'adresses email associées au domaine. Explorez ces options selon vos besoins.

7. Configuration des Paramètres DNS : Accédez à votre compte chez le registraire pour configurer les paramètres DNS de votre domaine. Cela peut inclure la gestion des enregistrements MX, CNAME, et d'autres configurations nécessaires à votre site.

8. Renouvellement Périodique : Les noms de domaine sont généralement enregistrés pour une période déterminée (par exemple, un an). Assurez-vous de renouveler votre enregistrement avant l'expiration pour éviter de perdre la propriété de votre domaine.

Quelles sont les spécificités que l'on peut avoir sur certaines extensions de nom de domaine ?

L'extension d'un nom de domaine constitue la dernière partie de celui-ci et joue un rôle crucial tant pour les visiteurs que pour les moteurs de recherche. Voici les points à considérer, classés par ordre d'importance :

1. Confiance des visiteurs :

- Les extensions courantes comme ".com", ".net" et ".org" renforcent la confiance des visiteurs.
- Les extensions moins connues telles que ".zip", ".biz", ou ".info" peuvent susciter la méfiance en raison de leur utilisation fréquente par des spammeurs et des cybercriminels.

2. Particularités géographiques :

- Les extensions reflètent parfois la zone géographique, comme ".fr" pour la France ou ".be" pour la Belgique.
- Les extensions géographiques peuvent favoriser le trafic localisé, augmentant ainsi les chances de conversion en prospects ou clients.

3. Score de confiance élevé :

- Pour les institutions telles que les universités ou les organismes gouvernementaux, l'utilisation d'extensions comme ".edu" ou ".org" peut améliorer le score de confiance auprès des moteurs de recherche.
- Cependant, il est crucial de ne pas abuser de ces extensions pour éviter de tromper les internautes et les moteurs de recherche.

4. Types d'extensions :

- **Extensions génériques** : Comme ".com", ".org", ou ".net", largement utilisées sans exigences particulières.
- **Extensions géographiques** : Représentant des pays, villes ou régions spécifiques, telles que ".fr" ou ".paris".
- **Extensions spécifiques** : Réservées à des secteurs ou entreprises spécifiques, comme ".news" pour l'information, ".blog" pour les blogs, ou ".tech" pour des activités particulières.
- **Extensions restreintes** : Dédiées à des entités spécifiques, comme ".gouv" pour les branches gouvernementales ou ".edu" pour les institutions éducatives.
- **Extensions de marque** : La propriété d'une entreprise ou d'une marque, comme ".google", ".apple" ou ".amazon", avec des conditions d'enregistrement strictes.

En conclusion, le choix de l'extension de domaine est crucial, influençant la confiance des visiteurs, la pertinence géographique, le score de confiance, et s'alignant avec la nature spécifique d'une entité, qu'elle soit une institution, une entreprise sectorielle, ou une marque.

JOB 06 : Connectez notre hôte au nom de domaine local

Pour faire correspondre l'IP de notre serveur à un nom de domaine, nous allons tout d'abord éditer et ajouter l'adresse IP et notre nom de domaine dans le fichier host, pour le faire on utilise la commande : **sudo nano /etc/host** et ajoute notre IP et le nom de domaine :

```
GNU nano 7.2 /etc/hosts *
127.0.0.1 localhost
127.0.1.1 debian
192.168.110.129 dnsproject.prepa.com
```

Et on redémarre notre Service DNS avec commande : **sudo systemctl restart networking**

```
goldroger@debian:~$ sudo systemctl restart networking
```

Maintenant on va configurer Apache sur notre serveur, pour cela on va se rendre dans le répertoire de configuration d'apache avec cette commande : **cd /etc/apache2/sites-available**

```
goldroger@debian:~$ cd /etc/apache2/sites-available
goldroger@debian:/etc/apache2/sites-available$ ls
000-default.conf default-ssl.conf
```

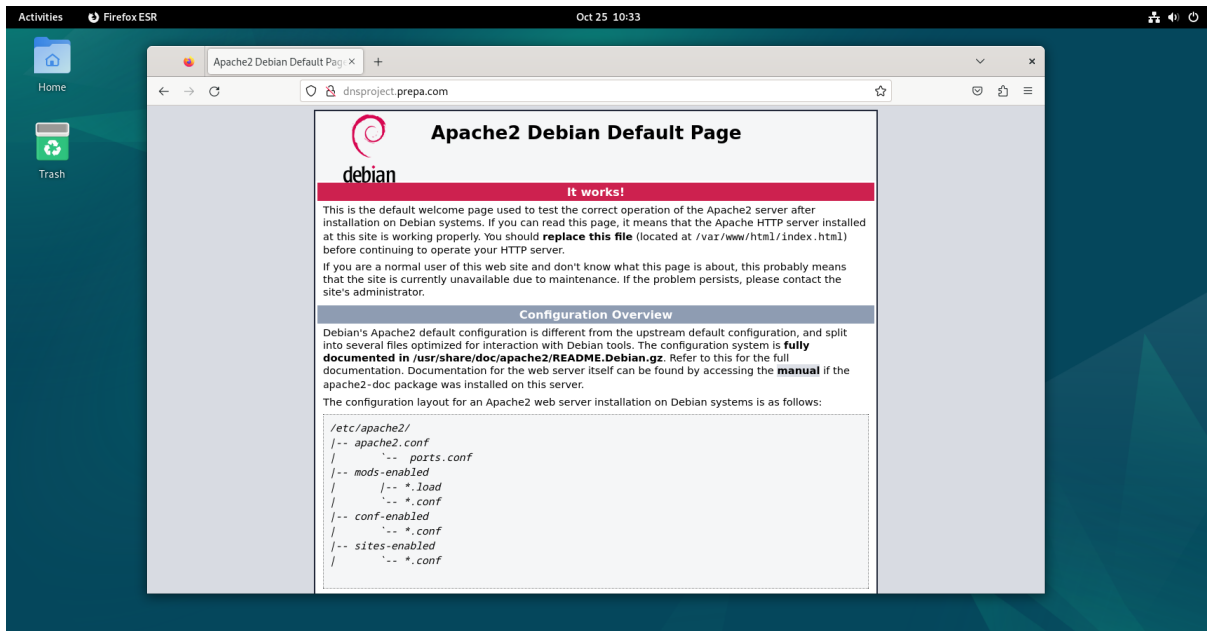
On va créer et éditer un fichier virtual host pour notre serveur avec nano et ajouter une configuration de base : **sudo nano blabla.com.conf**

```
goldroger@debian:/etc/apache2/sites-available
GNU nano 7.2 dnsproject.prepa.com.conf
<VirtualHost *:80>
    ServerAdmin webmaster@dnsproject.prepa.com
    ServerName dnsproject.prepa.com
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Maintenant on va activer le site et redémarrer Apache, pour le faire on va utiliser les commandes suivantes : **sudo a2ensite dnsproject.prepa.com.conf & sudo systemctl restart apache2**

```
goldroger@debian:/etc/apache2/sites-available$ sudo a2ensite dnsproject.prepa.com.conf
Enabling site dnsproject.prepa.com.
To activate the new configuration, you need to run:
    systemctl reload apache2
goldroger@debian:/etc/apache2/sites-available$ sudo systemctl restart apache2
goldroger@debian:/etc/apache2/sites-available$
```

Maintenant testons pour voir si notre configuration a bien été appliquée, pour cela on va se rendre sur notre nom de domaine : <http://dnsproject.prepa.com>



CA FONCTIONNE !!

JOB 07 : Mettre en place un pare-feu sur notre serveur pour qu'il ne puisse plus ping

Tout d'abord faudra installer ufw, mais comme on l'a déjà installé précédemment et autoriser les ports.

On va cette fois bloquer les ping pour cela on va se rendre dans le répertoire `/etc/ufw/` et modifier le fichier `before.rules` nano avec la commande `sudo nano before.rules`

Ensuite on scroll en bas jusqu'au paramètre **INPUT** , la 4eme ligne echo-request est en **ACCEPT**

```
# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT
```

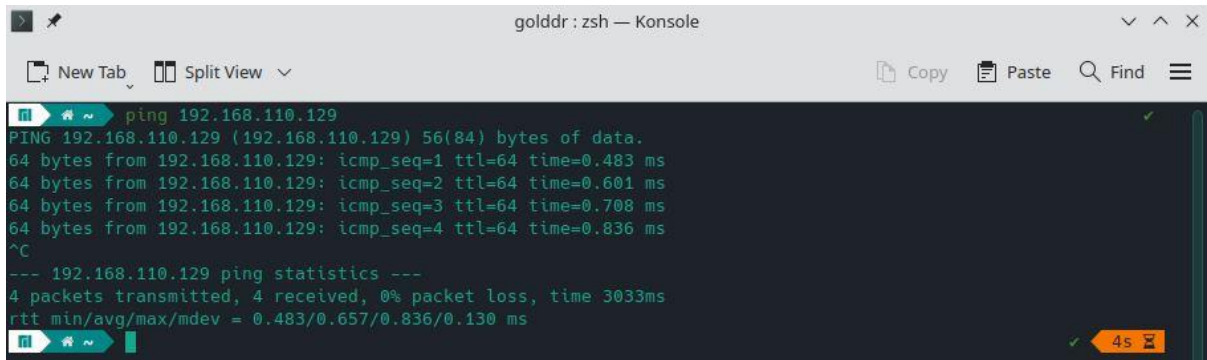
On le modifie en le mettant en **DROP**

```
# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-request -j DROP
```

Ensuite on enregistre, ensuite truc important pour appliquer notre changement on va reload notre pare-feu avec la commande : **sudo ufw reload**

```
root@debian:/etc/ufw# sudo ufw reload
Firewall reloaded
```

AVANT BLOCAGE - PING A PARTIR D'UNE MACHINE EXTERNE



A terminal window titled 'golddr : zsh — Konsole' showing a successful ping command. The output displays four successful pings with varying times (0.483 ms to 0.836 ms) and a summary: '4 packets transmitted, 4 received, 0% packet loss, time 3033ms'.

APRES BLOCAGE - RIEN NE PASSE



A terminal window titled 'golddr : zsh — Konsole' showing a failed ping command. The output displays '11 packets transmitted, 0 received, 100% packet loss, time 10129ms'. A red error indicator is visible in the bottom right corner of the terminal window.

JOB 08 : Mise en place d'un dossier partagé en réseau.

Ici pour mettre en place notre dossier partagé, on va utiliser **Samba**, c'est une suite d'applications pour partager des fichiers et des imprimantes entre systèmes d'exploitation. Pour l'installation on va utiliser la commande suivante : **sudo apt install samba**

```
goldroger@debian:~$ sudo apt install samba
```

Maintenant nous allons créer un repertoire que nous allons nommer "**shared_folder**" et c'est ce dossier que nous allons partager en réseau.

Pour créer notre répertoire nous allons utiliser la commande : **mkdir shared_folder**

```
goldroger@debian:~$ mkdir shared_folder
goldroger@debian:~$ ls
Desktop    Downloads  Music      Public      shell      Templates
Documents  le_reseau Pictures    shared_folder shell-exe   Videos
```

Maintenant nous allons changer les permissions de notre répertoire pour mettre au membre de notre réseau de pouvoir lire et écrire dans notre répertoire, pour ce faire nous allons utiliser la commande : **chmod -R 777 shared_folder**

```
goldroger@debian:~$ chmod -R 777 shared_folder
goldroger@debian:~$ ls -l
total 48
drwxr-xr-x  2 goldroger goldroger 4096 Oct 25 16:53 Desktop
drwxr-xr-x  2 goldroger goldroger 4096 Sep 20 12:34 Documents
drwxr-xr-x  3 goldroger goldroger 4096 Oct 21 08:25 Downloads
drwxr-xr-x  3 goldroger goldroger 4096 Oct 21 08:26 le_reseau
drwxr-xr-x  2 goldroger goldroger 4096 Sep 20 12:34 Music
drwxr-xr-x  3 goldroger goldroger 4096 Oct 25 00:41 Pictures
drwxr-xr-x  2 goldroger goldroger 4096 Sep 20 12:34 Public
drwxrwxrwx  2 goldroger goldroger 4096 Oct 25 22:08 shared_folder
drwxr-xr-x  4 root      root      4096 Sep 25 13:09 shell
drwxr-xr-x 12 goldroger goldroger 4096 Sep 29 15:49 shell-exe
drwxr-xr-x  2 goldroger goldroger 4096 Sep 20 12:34 Templates
drwxr-xr-x  2 goldroger goldroger 4096 Sep 20 12:34 Videos
goldroger@debian:~$
```

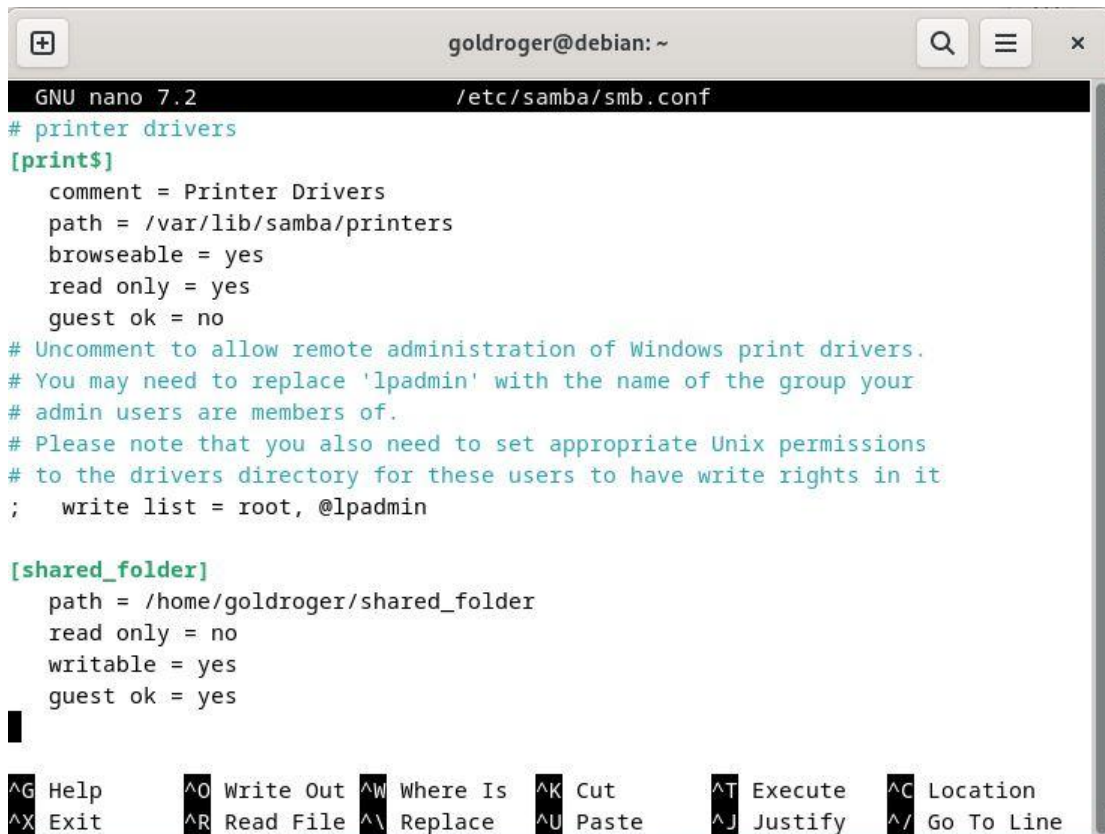
Ensuite nous allons ajouter notre **user** dans la database de **samba** et définir un mot de passe pour cela on va utiliser la commande : **sudo smbpasswd -a + user**

```
goldroger@debian:~$ sudo smbpasswd -a goldroger
[sudo] password for goldroger:
New SMB password:
Retype new SMB password:
goldroger@debian:~$
```

Maintenant nous allons modifier et ajouter dans le fichier de configuration de samba notre configuration et pouvoir enfin partager notre répertoire, le fichier se trouve dans le rep **/etc/samba/** nommer **smb.conf** , pour le modifier on va utiliser la commande **nano** : **sudo nano /etc/samba/smb.conf**

```
goldroger@debian:~$ sudo nano /etc/samba/smb.conf
```


Ensuite on va scroller tout en bas et ajouter notre configuration



```
goldroger@debian: ~
GNU nano 7.2 /etc/samba/smb.conf
# printer drivers
[print$]
    comment = Printer Drivers
    path = /var/lib/samba/printers
    browseable = yes
    read only = yes
    guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
; write list = root, @lpadmin

[shared_folder]
    path = /home/goldroger/shared_folder
    read only = no
    writable = yes
    guest ok = yes
```

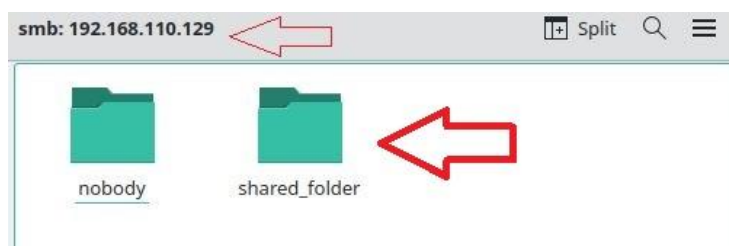
Help Write Out Where Is Cut Execute Location
Exit Read File Replace Paste Justify Go To Line

Maintenant on va enregistrer notre fichier et relancer le service samba pour appliquer notre configuration, pour relancer le service on va utiliser la commande : **sudo service smb restart**

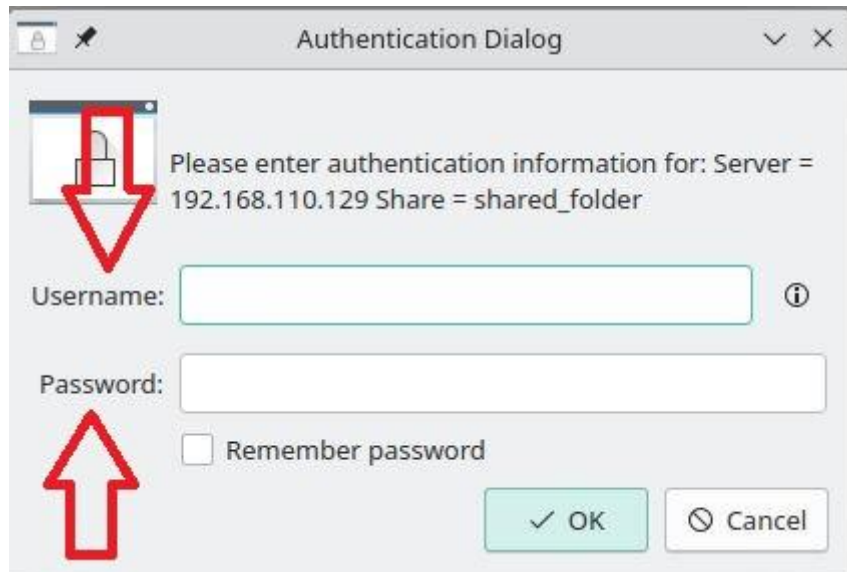
Maintenant allons tester si tout fonctionne , pour accéder au dossier partagé , on peut soit utiliser le gestionnaire de fichiers de notre environnement de bureau graphique pour accéder ou bien en utilisant l'adresse IP du serveur dans l'explorateur de fichiers : **smb://192.168.110.129**

sous Manjaro :

On est bien a l'adresse IP de serveur et on peut voir notre dossier qu'on a partager



On nous demande de nous authentifier



Ça fonctionne !!!

Pour aller plus loin...

Pour installer un certificat à notre serveur et activer le HTTPS, on va commencer par installer **OpenSSL** avec la commande : `sudo apt install openssl`

```
goldroger@debian:~$ sudo apt install openssl
[sudo] password for goldroger:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssl is already the newest version (3.0.11-1~deb12u2).
openssl set to manually installed.
```

Maintenant nous allons générer un certificat auto-signé avec cette commande : `sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/dnsproject.prepa.com.key -out /etc/ssl/certs/dnsproject.prepa.com.crt`


```
goldroger@debian:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/dnsproject.prepa.com.key -out /etc/ssl/certs/dnsproject.prepa.com.crt
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:FR
Locality Name (eg, city) []:PARIS
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Plateformeurs
Organizational Unit Name (eg, section) []:Akatsuki
Common Name (e.g. server FQDN or YOUR name) []:Gol D Roger
Email Address []:pirate@king.ocean
goldroger@debian:~$
```

Nous allons maintenant dans `/etc/apache2/` dans le dossier `sites-available` modifier et ajouter a notre fichier `.conf` de notre site ces lignes :

SSL Engine on

SSLCertificateFile /etc/ssl/certs/dnsproject.prepa.com.crt

SSLCertificateKeyFile /etc/ssl/private/dnsproject.prepa.com.key

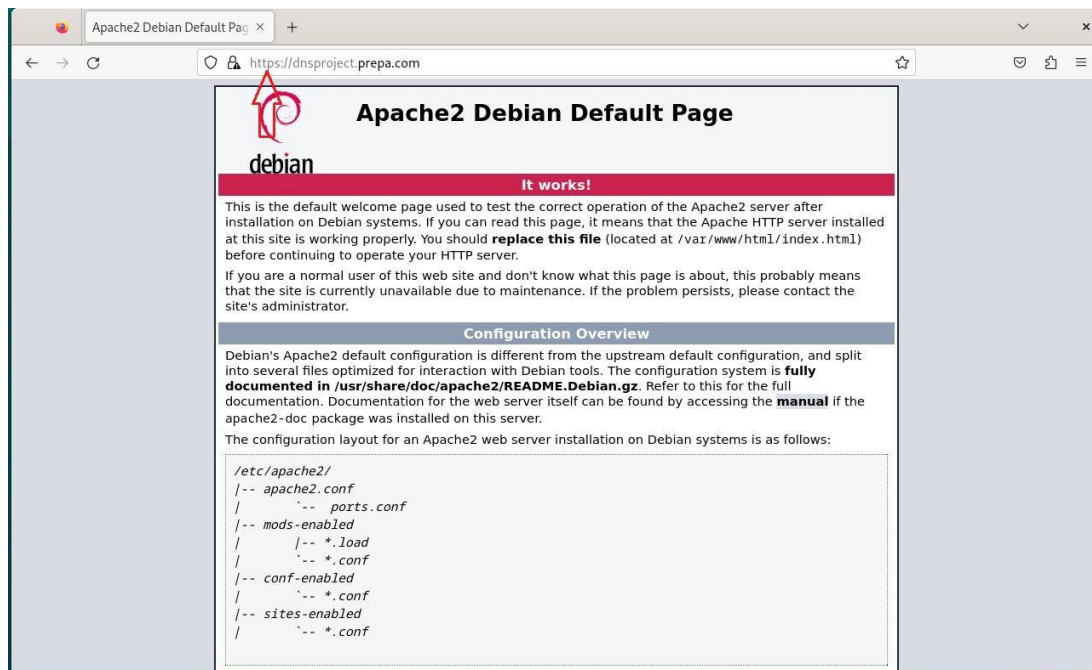
```
GNU nano 7.2 /etc/apache2/sites-available/dnsproject.prepa.com.conf
<VirtualHost *:443>
    ServerAdmin webmaster@dnsproject.prepa.com
    ServerName dnsproject.prepa.com
    DocumentRoot /var/www/html

    SSL Engine on
    SSLCertificateFile /etc/ssl/certs/dnsproject.prepa.com.crt
    SSLCertificateKeyFile /etc/ssl/private/dnsproject.prepa.com.key

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Enfin nous allons activer le module SSL avec cette commande : `sudo a2enmod ssl` et redémarrez Apache pour appliquer les modifications avec cette commande : `sudo systemctl restart apache2`

```
goldroger@debian:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
goldroger@debian:~$ sudo systemctl restart apache2
goldroger@debian:~$
```



La différence entre les certificats SSL donnés par des organismes extérieurs et le vôtre auto-signé ?

Les certificats SSL, qu'ils soient délivrés par des organismes de certification externes (**Certification Authorities - CA**) ou auto-signés, sont utilisés pour sécuriser les communications sur Internet en établissant des connexions chiffrées entre les utilisateurs et les serveurs. Cependant, il y a des différences importantes entre ces deux types de certificats.

Certificats SSL délivrés par des organismes de certification externes :

- 1. Confiance universelle :** Les certificats SSL émis par des CA externes sont généralement inclus dans les listes de confiance des navigateurs. Cela signifie que lorsque vous utilisez un certificat émis par une CA, la plupart des navigateurs font confiance à ce certificat par défaut.
- 2. Validation approfondie :** Les CAs effectuent des vérifications approfondies pour s'assurer que le demandeur du certificat est légitime. Ces vérifications peuvent inclure des processus tels que la vérification du domaine, la vérification de l'organisation et d'autres procédures.
- 3. Coût :** Les certificats SSL émis par des CAs externes sont payants. Le coût peut varier en fonction du type de certificat et du niveau de validation effectué.
- 4. Durée de validité :** Les certificats SSL délivrés par des CAs ont une durée de validité limitée (généralement un an ou plus). Ils doivent être renouvelés régulièrement.

Certificats auto-signés :

- 1. Confiance limitée :** Les certificats auto-signés ne sont pas inclus dans les listes de confiance des navigateurs par défaut. Lorsqu'un utilisateur accède à un site utilisant un certificat auto-signé, le navigateur génère généralement un avertissement, indiquant que le certificat n'est pas de confiance.
- 2. Validation limitée :** Les certificats auto-signés ne subissent pas le même processus de validation approfondie que les certificats délivrés par des CAs externes. Le processus de génération d'un certificat auto-signé peut être réalisé par n'importe qui sans aucune vérification externe.
- 3. Coût :** Les certificats auto-signés sont gratuits à générer. Cela les rend attrayants pour des environnements de développement ou des tests.
- 4. Durée de validité :** Les certificats auto-signés peuvent avoir une durée de validité définie par l'utilisateur, mais cela dépend du logiciel utilisé pour les générer.

Recommandations :

- Production vs Développement :** En production, il est fortement recommandé d'utiliser des certificats émis par des CAs externes pour garantir la confiance des utilisateurs. Pour le développement ou des environnements locaux, les certificats auto-signés peuvent être utilisés.
- Confidentialité et Sécurité :** Les certificats délivrés par des CAs offrent un niveau plus élevé de confiance et de sécurité, ce qui les rend appropriés pour les sites web nécessitant une transmission sécurisée d'informations sensibles.

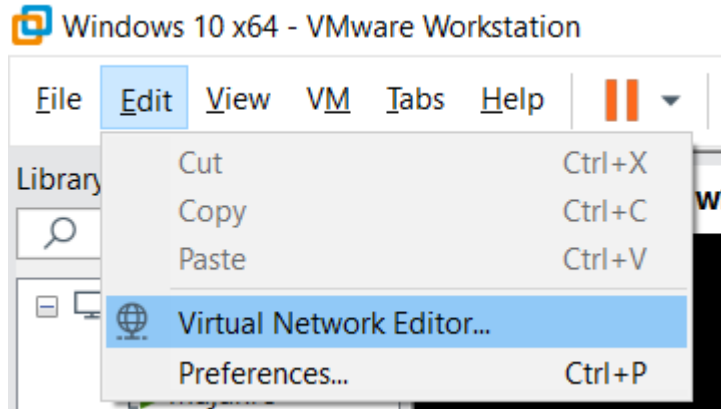
Pourquoi votre certificat apparaît-il comme non sécurisé dans votre navigateur ?

Un certificat auto-signé apparaît comme non sécurisé dans un navigateur principalement parce qu'il n'a pas été émis par une autorité de certification reconnue, ce qui conduit à un manque de confiance.

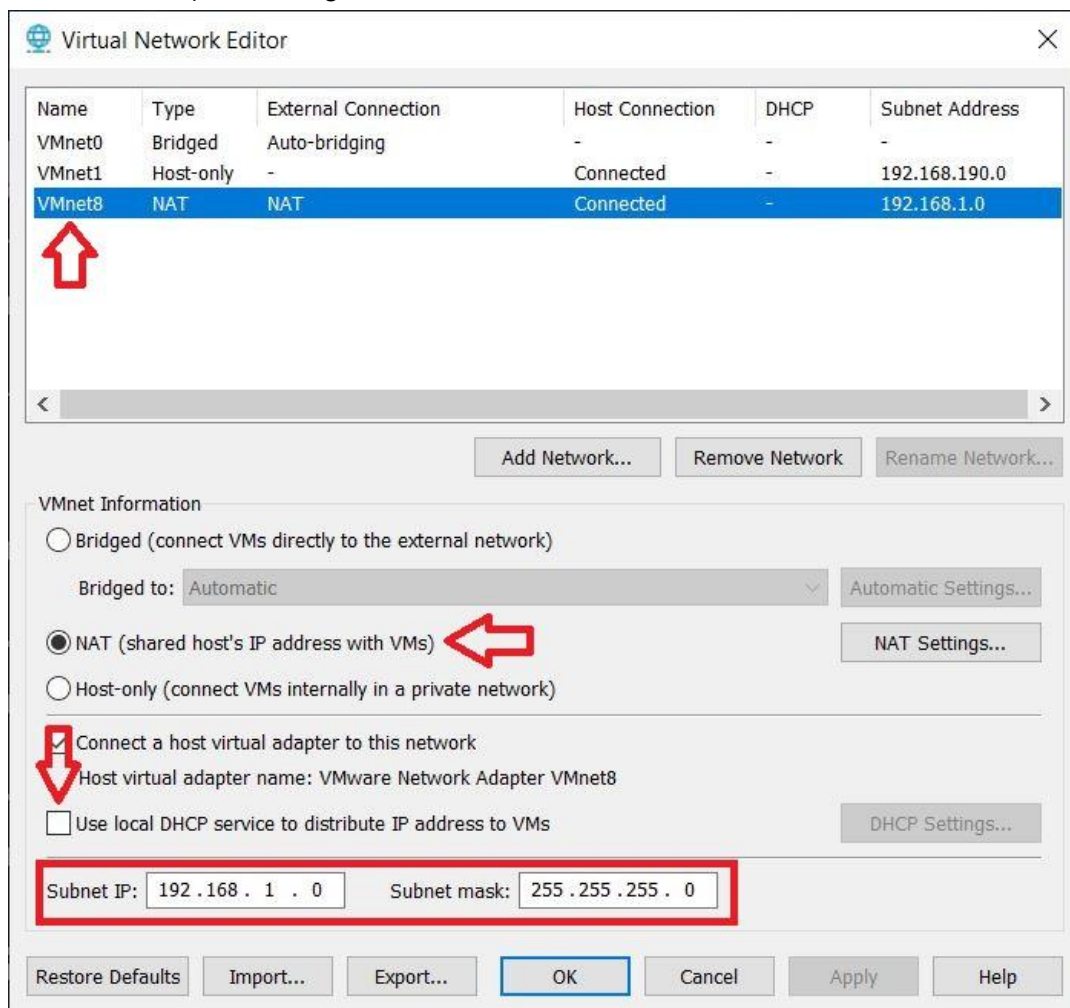
Pour aller encore plus loin ...

Installer un DHCP en dehors de celui de VMWare.

Tout d'abord nous allons désactiver le DHCP de notre VM pour le faire on va dans Edit > Virtual Network Editor



Ensuite on sélectionne notre réseau > VMnet Information (on reste en NAT) > On décoche "Use local DHCP..." > Enfin on définit le réseau et son masque (ici on au 192.168.1.0 255.255.255.0). On enregistre le tout et on commence !



Maintenant sous linux pour pouvoir mettre en place notre DHCP, nous aurons besoin d'installer **isc-dhcp-server** avec la commande : **sudo apt install isc-dhcp-server**

```
goldroger@debian:~$ sudo apt install isc-dhcp-server
[sudo] password for goldroger:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  linux-image-6.1.0-10-amd64
Use 'sudo apt autoremove' to remove it.
```

Ensuite, on va changer notre réseau IPv4 de auto en manuel, et configurer notre serveur en lui assignant l'adresse IP et son masque **192.168.1.1 255.255.255.0**

The screenshot shows the 'Wired' network configuration window. The 'IPv4 Method' is set to 'Manual'. The 'Addresses' table has one entry: Address '192.168.1.1', Netmask '255.255.255.0', and Gateway is empty. The 'DNS' section is set to 'Automatic' with the address '192.168.1.1'. The 'Routes' section is also set to 'Automatic'.

Address	Netmask	Gateway
192.168.1.1	255.255.255.0	

Address	Netmask	Gateway	Metric

On va maintenant se déplacer dans le repertoire **/etc/default/** et chercher le fichier **isc-dhcp-server**

```
root@debian:/home/goldroger# cd /etc/default/
root@debian:/etc/default# ls
anacron          console-setup  grub           im-config      locale         saned
apache-htcacheclean  cron          grub.d         intel-microcode  named          ufw
avahi-daemon        dbus          grub.ucf-dist  isc-dhcp-server  networking     useradd
bluetooth          google-chrome hwclock        keyboard        nss
```

On va ensuite ouvrir et modifier le fichier **isc-dhcp-server** avec nano : **nano isc-dhcp-server**

```
root@debian:/etc/default# nano isc-dhcp-server
```


On va venir décommenter la 4e ligne (**DHCPDv4_CONF...**) et tout en bas on va venir indiquer l'interface réseau sur laquelle écoute le serveur (**INTERFACESv4**)

```
GNU nano 7.2 /etc/default/isc-dhcp-server *
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="ens192"
INTERFACESv6=""
```

Ensuite on enregistre et on se déplace dans vers le répertoire **/etc/dhcp/** et chercher le fichier **dhcpd.conf**

```
root@debian:/etc# cd dhcp
root@debian:/etc/dhcp# ls
debug  dhclient.conf  dhclient-enter-hooks.d  dhclient-exit-hooks.d  dhcpd6.conf  dhcpd.conf
```

On va ensuite ouvrir et modifier le fichier **dhcpd.conf** avec nano : **nano dhcpd.conf** et venir commenter la 7e et 8e ligne (**option domain-name et domain-name-server...**)

```
GNU nano 7.2 dhcpd.conf *
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# option definitions common to all supported networks...
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;
```

Et tout en bas on va venir décommenter la ligne "**A slightly different configuration for an internal subnet**" et configurer notre réseau :

```
goldroger@debian:/etc/dhcp
GNU nano 7.2 dhcpd.conf

# A slightly different configuration for an internal subnet.
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.10 192.168.1.100;
    option domain-name-servers 192.168.1.1;
# option domain-name "dnsproject.prepa.com";
    option routers 192.168.1.1;
    option broadcast-address 192.168.1.255;
    default-lease-time 600;
    max-lease-time 7200;
}
```

Ensuite on enregistre et on quitte

Maintenant on va mettre en marche notre serveur pour cela on le restart et l'activer :

La commande pour restart : **sudo systemctl restart isc-dhcp-server**

La commande pour restart : **sudo systemctl enable isc-dhcp-server**

```
goldroger@debian:/$ sudo systemctl restart isc-dhcp-server
goldroger@debian:/$ sudo systemctl enable isc-dhcp-server
isc-dhcp-server.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable isc-dhcp-server
goldroger@debian:/$
```

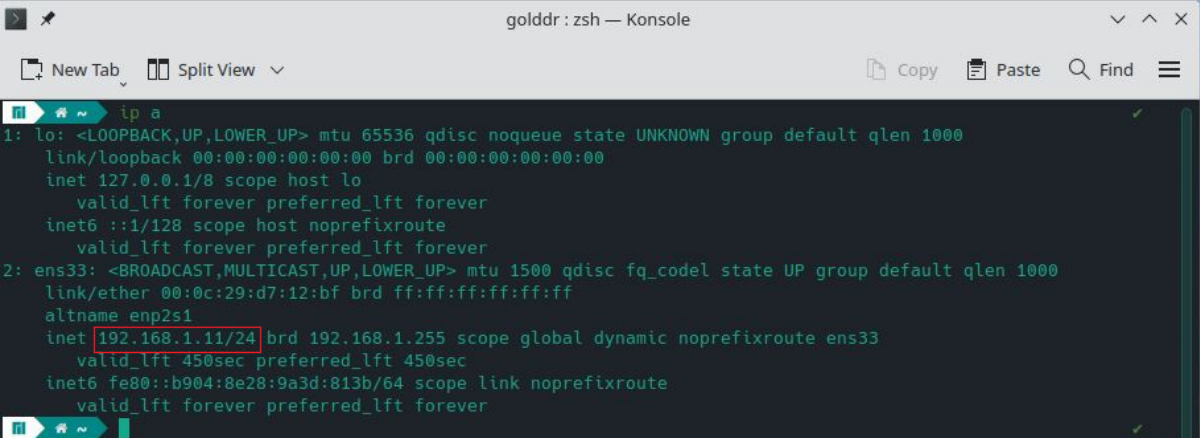
On va vérifier qu'il bien actif : **sudo systemctl status isc-dhcp-server**

```
goldroger@debian:/$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
   Active: active (running) since Thu 2023-10-26 14:01:30 CEST; 1min 14s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 1 (limit: 9452)
   Memory: 4.3M
      CPU: 68ms
   CGroup: /system.slice/isc-dhcp-server.service
           └─4601 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf ens192

Oct 26 14:01:28 debian systemd[1]: Starting isc-dhcp-server.service - LSB: DHCP server...
Oct 26 14:01:28 debian isc-dhcp-server[4588]: Launching IPv4 server only.
Oct 26 14:01:28 debian dhcpd[4601]: Wrote 3 leases to leases file.
Oct 26 14:01:28 debian dhcpd[4601]: Server starting service.
Oct 26 14:01:30 debian isc-dhcp-server[4588]: Starting ISC DHCPv4 server: dhcpd.
Oct 26 14:01:30 debian systemd[1]: Started isc-dhcp-server.service - LSB: DHCP server.
goldroger@debian:/$
```

On a finit maintenant testons si ca fonctionne, sur nos autres VM

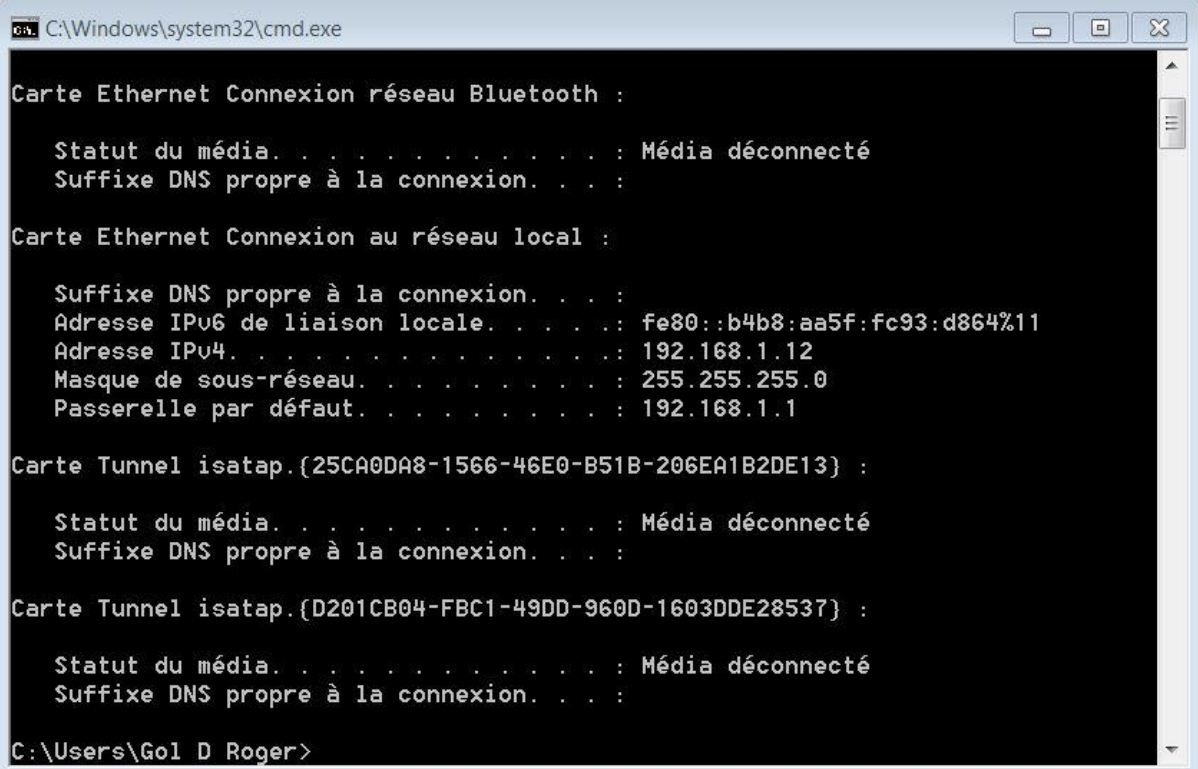
CAPTURE D'ÉCRAN MANJARO



```
goldr: zsh — Konsole
New Tab Split View Copy Paste Find
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 00:0c:29:d7:12:bf brd ff:ff:ff:ff:ff:ff
   altname enp2s1
   inet 192.168.1.11/24 brd 192.168.1.255 scope global dynamic noprefixroute ens33
       valid_lft 450sec preferred_lft 450sec
   inet6 fe80::b904:8e28:9a3d:813b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

On voit qu'on lui a bien assigner une IP : **192.168.1.11**

CAPTURE D'ÉCRAN WINDOWS 7



```
C:\Windows\system32\cmd.exe

Carte Ethernet Connexion réseau Bluetooth :

    Statut du média. . . . . : Média déconnecté
    Suffixe DNS propre à la connexion. . . :

Carte Ethernet Connexion au réseau local :

    Suffixe DNS propre à la connexion. . . :
    Adresse IPv6 de liaison locale. . . . : fe80::b4b8:aa5f:fc93:d864%11
    Adresse IPv4. . . . . : 192.168.1.12
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : 192.168.1.1

Carte Tunnel isatap.{25CA0DA8-1566-46E0-B51B-206EA1B2DE13} :

    Statut du média. . . . . : Média déconnecté
    Suffixe DNS propre à la connexion. . . :

Carte Tunnel isatap.{D201CB04-FBC1-49DD-960D-1603DDE28537} :

    Statut du média. . . . . : Média déconnecté
    Suffixe DNS propre à la connexion. . . :

C:\Users\Gol D Roger>
```

On voit qu'on lui aussi assigner une IP : 192.168.1.12

NOTRE DHCP FONCTIONNE !!