

Holodek

Jordan Reinaldo

Bachelor

03/09/2024



SOMMAIRE

Présentation du sujet page 1

Configuration VM Clients et serveur page 4

Installation et configuration du DHCP page 10

Configuration DNS page 13

Configuration Nginx page 19

Certificats auto-signés pour HTTPS page 22

Installation et configuration PHP8 et PHP7 page 24

Nginx dernière version page 30

Installation MariaDB page 32

Installation phpMyAdmin page 35

Configuration SFTP page 37

Installation Idap page 39

Installation Zabbix et UFW page 46



Présentation du sujet

Objectif :

L'objectif du projet Holodek est d'avoir deux machines virtuelles, une réservée à l'hébergement de serveurs et la seconde en tant que client pour tester les différentes fonctionnalités.

Configuration VM serveur et fonctionnalités attendues :

Debian 12 **sans** interface graphique/ 2Go RAM/ 2vpcu/ 32 Go stockage/ 2 cartes reseaux (une WAN et une LAN)

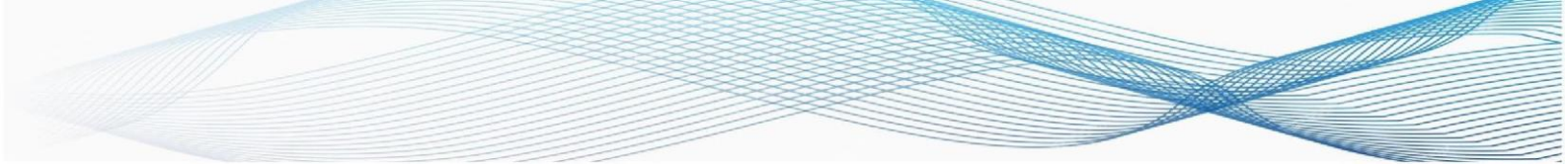
Serveur DHCP/DNS(lan)/ SFTP/ Serveur Web, Serveur de base de données SQL et serveur LDAP

contraintes :

- Pas de compte Sudo
- Pare-feu autorisant uniquement les ports des services utilisés
- Serveur Web Nginx et HTTPS
- Dernière version pour PHP, MariaDB et Nginx
- Cohabitation entre PhP version 7.x et 8.x

Mise en place :

- DHCP/DNS** : domaine starfleet.lan
- Web** : www8.starfleet.lan ⇒ site web en php8/ www7.starfleet.lan ⇒ site web en php7/ php.starfleet.lan ⇒ phpMyAdmin/ admin.starfleet.lan ⇒ administration de la VM
- Serveur FTP** : (en SSL/TLS) pour copier les fichiers du serveur Web (chrooté sur le dossier web).



-Création d'un certificat SSL qui servira pour le serveur Web et le Serveur FTP.

Pour authentifier les Utilisateurs, le serveur Web utilisera un annuaire LDAP.

Configuration VM Client et fonctionnalités attendues :

Debian 12 **avec** interface graphique/ 2Go RAM/ 2vcpu/ 16 Go Stockage/ Carte réseaux (LAN).

La VM client devra pouvoir accéder à tous les serveurs de la VM serveur en passant par le réseau LAN et un navigateur web.

Configuration VM Client

1) Installation et configuration de l'OS

Pour la configuration hardware nous suivons les spécifications précisées ci-dessus et nous mettons en Lan segment pour avoir uniquement une carte réseau en LAN :

The screenshot shows the VMware Workstation 'Hardware' configuration window. The 'Hardware' tab is active, displaying a list of devices on the left and the 'Memory' configuration on the right.

Device	Summary
Memory	2 GB
Processors	2
Hard Disk (SCSI)	20 GB
CD/DVD (SATA)	Using file D:\La plateforme\iso...
Network Adapter 2	LAN Segment
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: MB

Memory slider scale (from bottom to top): 4 MB, 8 MB, 16 MB, 32 MB, 64 MB, 128 MB, 256 MB, 512 MB, 1 GB, 2 GB, 4 GB, 8 GB, 16 GB, 32 GB, 64 GB, 128 GB.

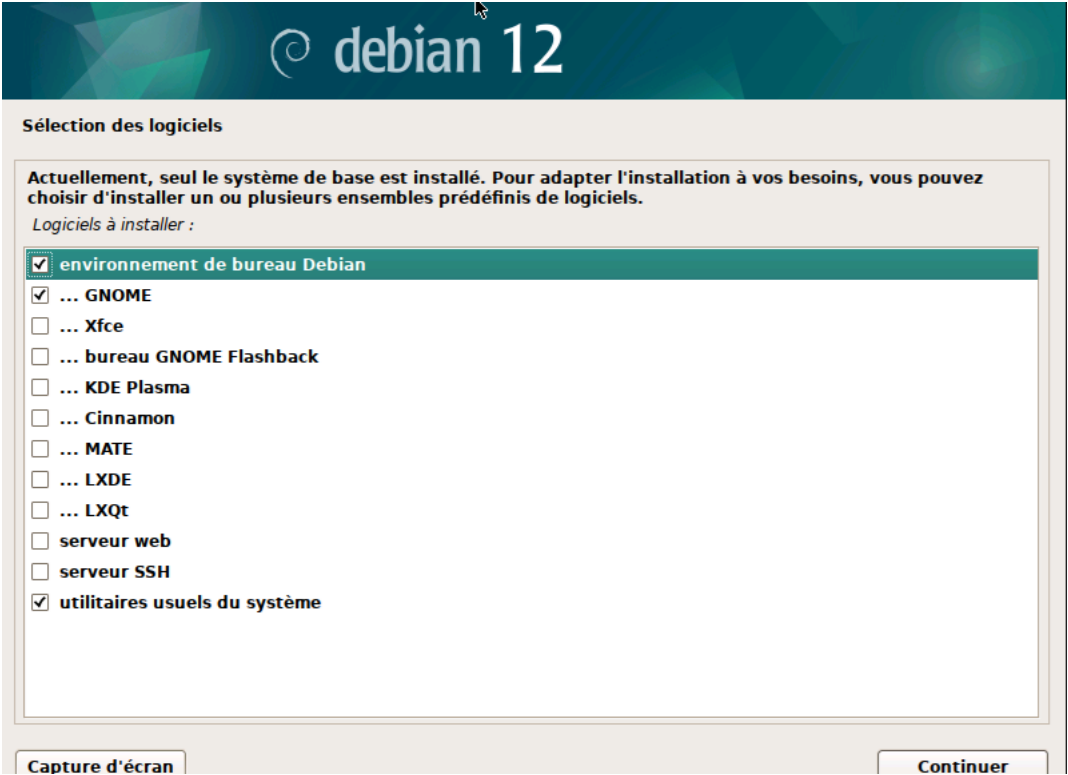
Legend:

- Maximum recommended memory (Memory swapping may occur beyond this size.): 26.0 GB
- Recommended memory: 2 GB
- Guest OS recommended minimum: 1 GB

Tableau récapitulatif réseau :

Mode d'accès	Communication				
	VM à VM	VM vers hôte	Hôte vers VM	VM vers LAN	LAN vers VM
NAT	+	+	Redirection de port	+	Redirection de port
Bridged	+	+	+	+	+
Host-Only	+	+	+	-	-
LAN Segments	+	-	-	-	-
Aucune connexion	-	-	-	-	-

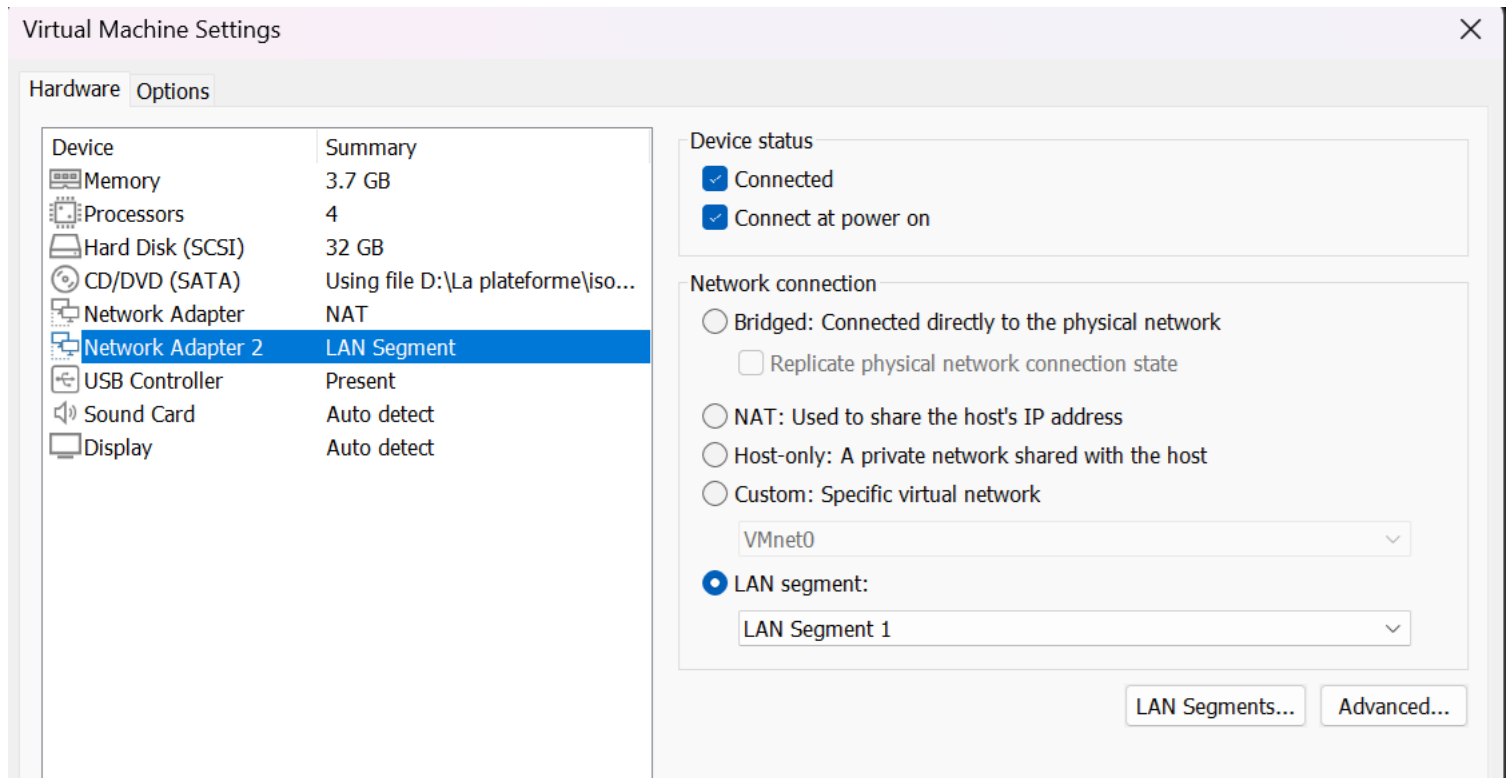
On reprend exactement le même processus que pour la VM serveur lors de l'installation de l'OS sauf que cette fois-ci, nous choisissons d'avoir une interface graphique :



Configuration VM Serveur

1) Installation et configuration de l'OS

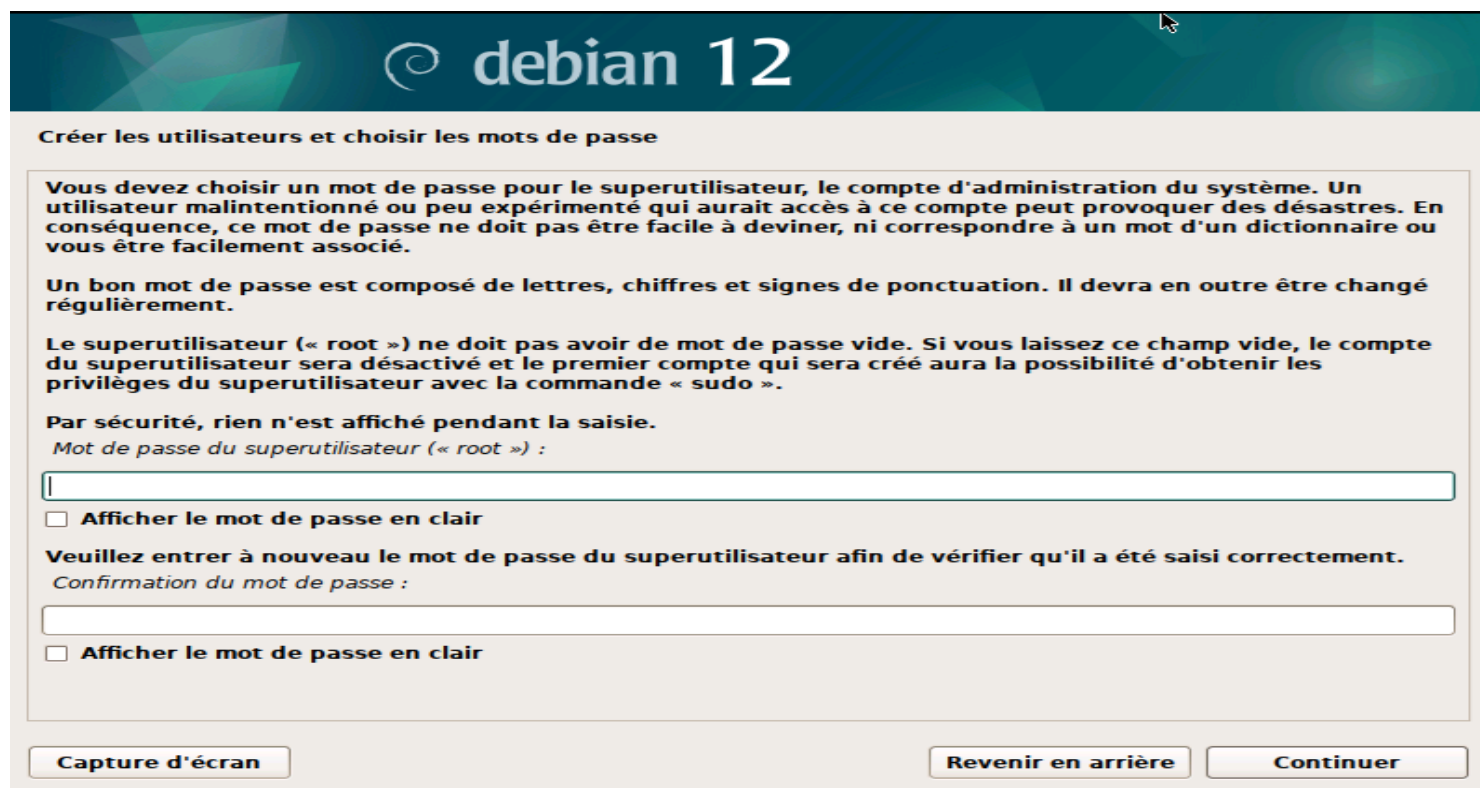
Configuration Hardware :



Pour la configuration, hardware, nous devons utiliser deux cartes réseaux, une en NAT(WAN) pour avoir accès à internet et une autre pour le réseau local. Concernant cette dernière, nous avons deux possibilités, le host only et le LAN segment. Le souci avec le host only est que l'utilisateur n'a pas accès à internet, ce que nous souhaitons mais à accès à des communications avec notre machine hôte. Ce n'est pas le cas en Lan Segment, pour des raisons de sécurité, nous avons donc pris cette dernière option. A noter également que les différentes machines du réseau devront être sur le même LAN segment.

Pour le reste de la configuration, nous avons décidé de mettre 4 cœurs pour de meilleures performances, les 32 Go de stockage qui étaient nécessaires, 4 Go de RAM et l'ISO de notre OS Debian 12.

On installe Debian 12, ici on configure le mot de passe de notre utilisateur root :



The screenshot shows the 'debian 12' installer window with the title 'Créer les utilisateurs et choisir les mots de passe'. It contains instructions for choosing a strong password for the root user, a password input field, a checkbox for 'Afficher le mot de passe en clair', a confirmation field, and another checkbox. At the bottom are buttons for 'Capture d'écran', 'Revenir en arrière', and 'Continuer'.

© debian 12

Créer les utilisateurs et choisir les mots de passe

Vous devez choisir un mot de passe pour le superutilisateur, le compte d'administration du système. Un utilisateur malintentionné ou peu expérimenté qui aurait accès à ce compte peut provoquer des désastres. En conséquence, ce mot de passe ne doit pas être facile à deviner, ni correspondre à un mot d'un dictionnaire ou vous être facilement associé.

Un bon mot de passe est composé de lettres, chiffres et signes de ponctuation. Il devra en outre être changé régulièrement.

Le superutilisateur (« root ») ne doit pas avoir de mot de passe vide. Si vous laissez ce champ vide, le compte du superutilisateur sera désactivé et le premier compte qui sera créé aura la possibilité d'obtenir les privilèges du superutilisateur avec la commande « sudo ».

Par sécurité, rien n'est affiché pendant la saisie.

Mot de passe du superutilisateur (« root ») :

☐ Afficher le mot de passe en clair

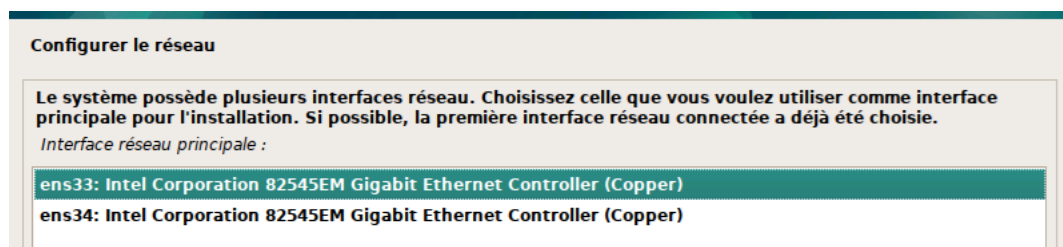
Veuillez entrer à nouveau le mot de passe du superutilisateur afin de vérifier qu'il a été saisi correctement.

Confirmation du mot de passe :

☐ Afficher le mot de passe en clair

Capture d'écran Revenir en arrière Continuer

On configure notre réseau NAT qui nous permettra d'avoir accès au monde extérieur :



The screenshot shows the 'Configurer le réseau' window. It contains instructions for selecting a network interface, a label for the selected interface, and a list of available interfaces with 'ens33' selected.

Configurer le réseau

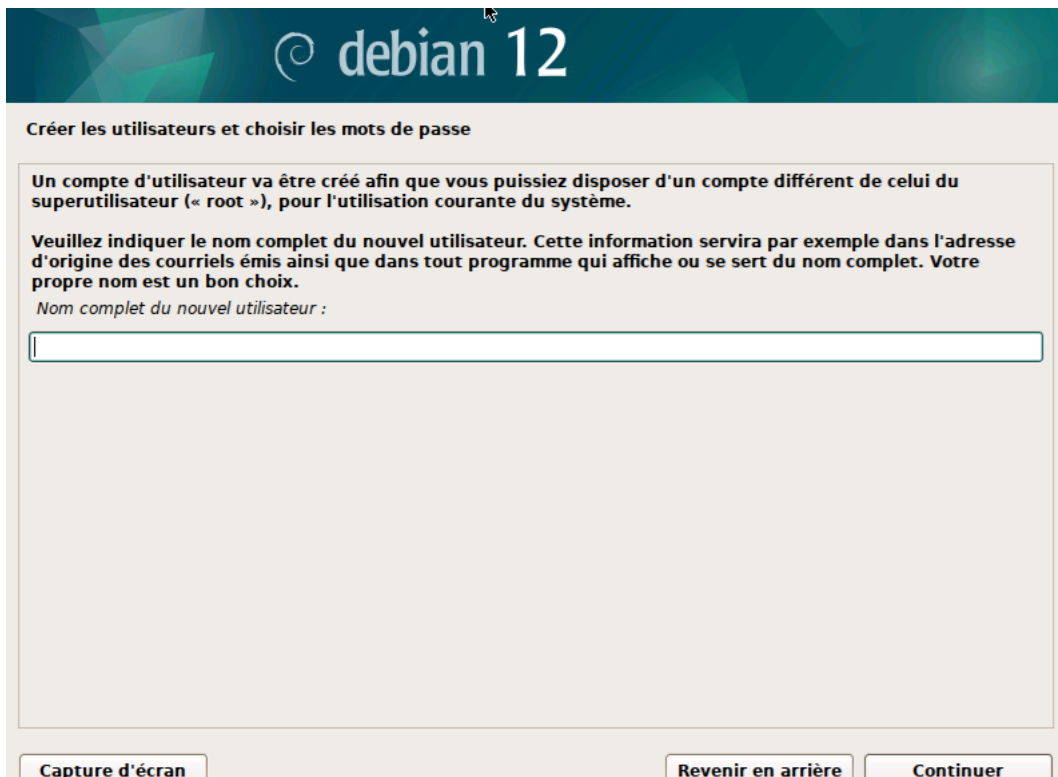
Le système possède plusieurs interfaces réseau. Choisissez celle que vous voulez utiliser comme interface principale pour l'installation. Si possible, la première interface réseau connectée a déjà été choisie.

Interface réseau principale :

ens33: Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)

ens34: Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)

Ensuite on crée notre utilisateur classique :



The screenshot shows the 'debian 12' logo at the top. Below it, the title 'Créer les utilisateurs et choisir les mots de passe' is displayed. The main text explains that a user account will be created for daily system use, distinct from the root superuser. It instructs the user to provide the full name of the new user, which will be used in email addresses and system programs. A text input field is provided for the full name. At the bottom, there are three buttons: 'Capture d'écran', 'Revenir en arrière', and 'Continuer'.

© debian 12

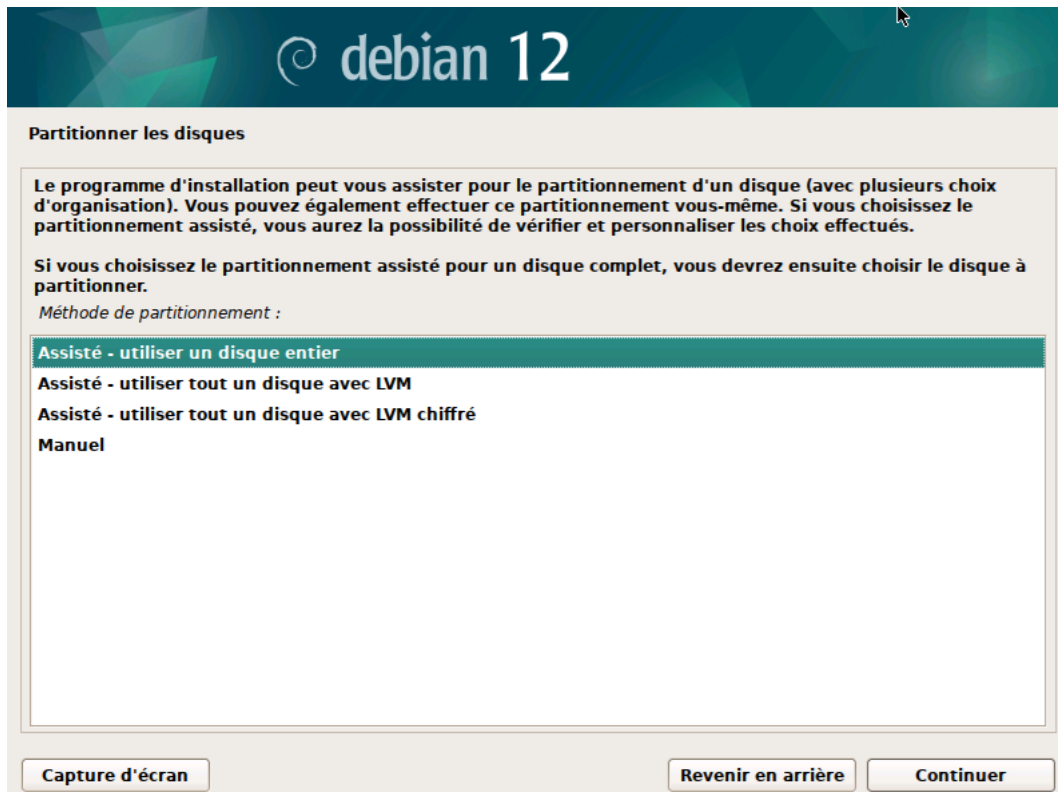
Créer les utilisateurs et choisir les mots de passe

Un compte d'utilisateur va être créé afin que vous puissiez disposer d'un compte différent de celui du superutilisateur (« root »), pour l'utilisation courante du système.

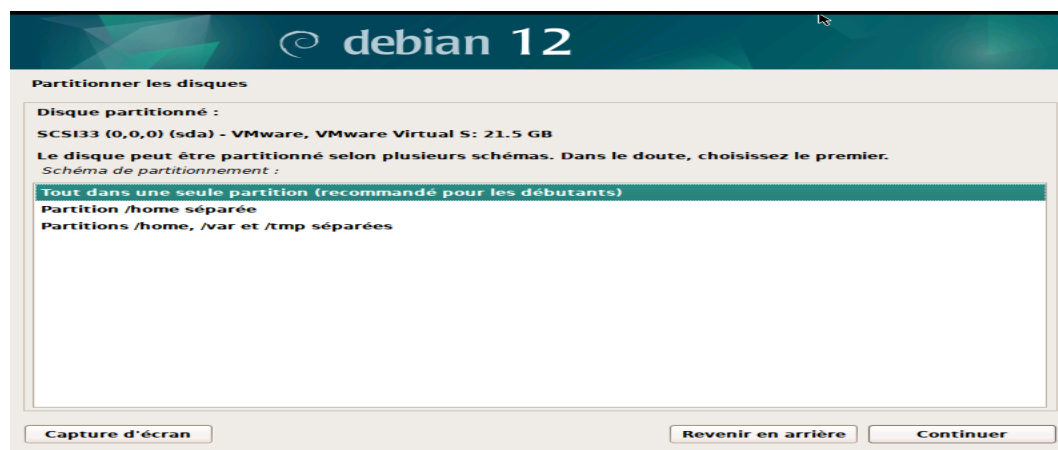
Veuillez indiquer le nom complet du nouvel utilisateur. Cette information servira par exemple dans l'adresse d'origine des courriels émis ainsi que dans tout programme qui affiche ou se sert du nom complet. Votre propre nom est un bon choix.

Nom complet du nouvel utilisateur :

Capture d'écran Revenir en arrière Continuer

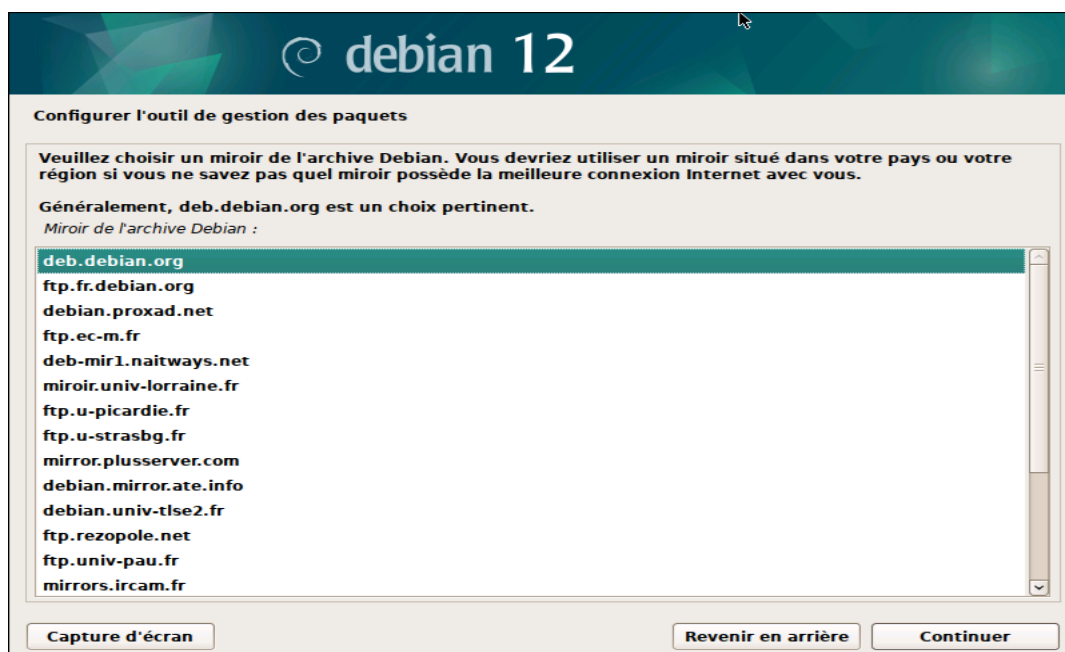


On choisit l'installation avec un disque entier :





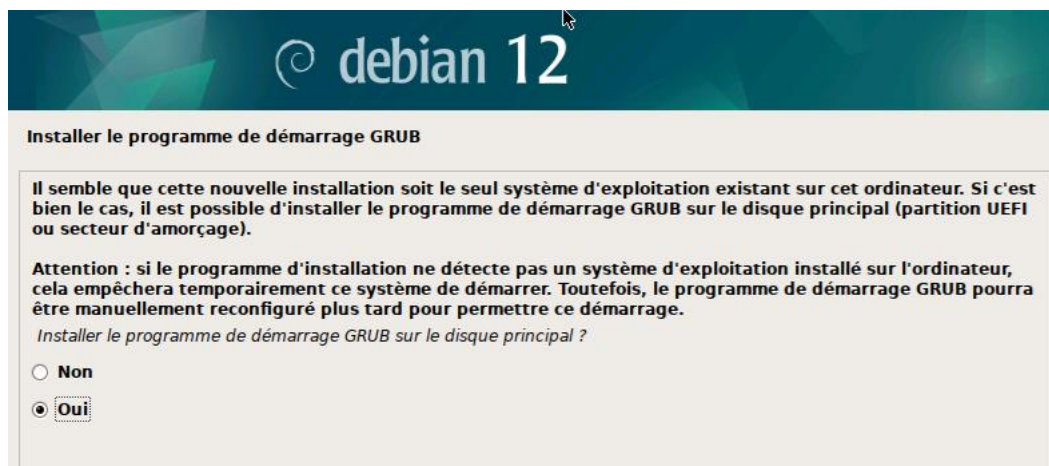
Gestion des paquets :

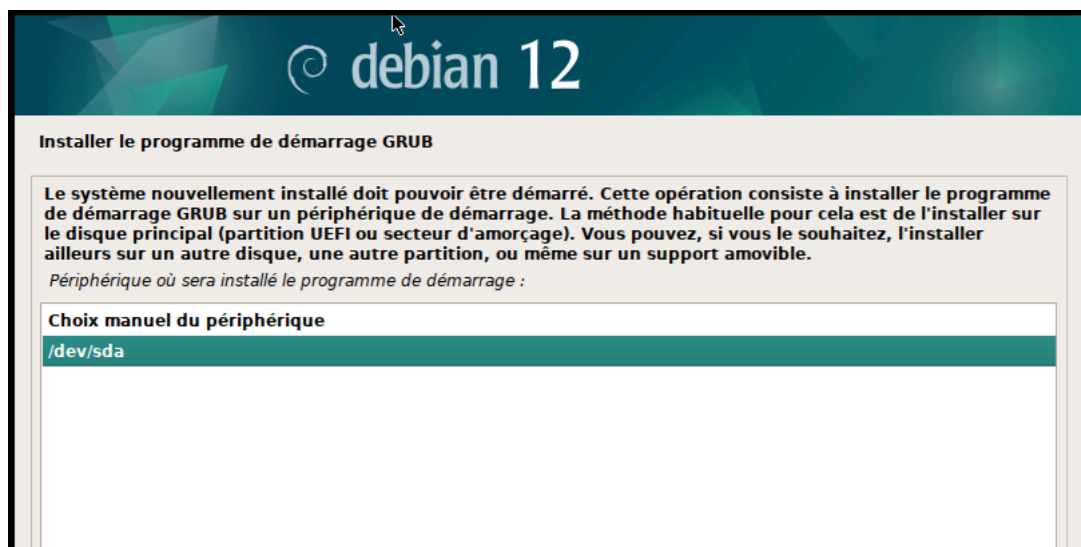


On choisit les différents logiciels, nous ne voulons pas d'interfaces graphiques :



On installe le grub :





2) Installation et configuration du DHCP

Explication : Dans notre projet, l'objectif est de configurer une machine virtuelle qui servira de serveur pour un réseau local (LAN), nous allons donc configurer le DHCP dans ce sens.

On utilise la commande **apt install isc-dhcp-server**

On configure maintenant les différents fichiers pour le bon fonctionnement de notre DHCP.

Nano etc/default/isc-dhcp-server

```
GNU nano 7.2                                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="ens34"
INTERFACESv6=""
```


INTERFACESv4 : Cette ligne spécifie l'interface réseau sur laquelle le serveur DHCP doit répondre aux requêtes DHCP. Ici, elle est définie sur ens34, qui correspond à une interface réseau de la machine virtuelle, plus précisément notre LAN Segment.

INTERFACESv6 : Ce champ est vide, ce qui indique que la configuration pour IPv6 n'est pas utilisée dans ce cas.

Ensuite, on va dans **Nano etc/network/interfaces**

```
GNU nano 7.2
# 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/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens33
iface ens33 inet dhcp

auto ens34
iface ens34 inet static
address 192.168.209.5
netmask 255.255.255.0
```

ens33 (Interface principale) : Cette interface utilise le DHCP pour obtenir son adresse IP (iface ens33 inet dhcp). Cela signifie qu'elle va récupérer une adresse IP d'un serveur DHCP externe.

ens34 (Deuxième interface) : Cette interface est configurée statiquement avec une adresse IP fixe : 192.168.76.5 et un masque de sous-réseau de 255.255.255.0. Cette adresse IP statique est utilisée pour que la machine virtuelle puisse fonctionner comme serveur DHCP pour le réseau local.

Enfin on va modifier le dhcpd.conf **nano etc/dhcp/dhcpd.conf**

```
subnet 192.168.209.0 netmask 255.255.255.0 {  
    range 192.168.209.50 192.168.209.100;  
    option subnet-mask 255.255.255.0;  
    option routers 192.169.205.5;  
    option broadcast-address 192.168.255.255;  
    default-lease-time 600;  
    max-lease-time 600;  
}  
  
ddns-update-style none;  
authoritative;
```

subnet : Cette section définit le sous-réseau pour lequel le serveur DHCP fournira des adresses IP. Dans ce cas, il s'agit du réseau 192.168.209.0 avec un masque de sous-réseau de 255.255.255.0.

range : Le serveur DHCP distribuera des adresses IP entre 192.168.209.50 et 192.168.209.100. Ce sont les adresses que les clients du réseau pourront recevoir.

option routers : Cela spécifie la passerelle par défaut pour les clients du sous-réseau, ici définie comme 192.169.205.5.

lease-time : Le default-lease-time et le max-lease-time sont tous deux fixés à 600 secondes (10 minutes). Ces paramètres contrôlent la durée pendant laquelle une adresse IP est attribuée à un client avant qu'il doive la renouveler.

ddns-update-style none : Cela désactive la mise à jour dynamique des enregistrements DNS, c'est-à-dire que le serveur DHCP ne tentera pas de mettre à jour les enregistrements DNS automatiquement.

authoritative : Cela indique que ce serveur DHCP est l'autorité principale pour ce sous-réseau, ce qui lui permet de répondre sans ambiguïté à toutes les requêtes DHCP.

3) Configuration du DNS

Pour commencer on utilise la commande suivante :

apt install bind9 bind9utils bind9-doc

BIND9 est une version moderne et largement utilisée du serveur DNS qui permet de gérer la résolution des noms de domaine, d'héberger des zones DNS, et bien plus encore.

bind9utils : Ce paquet contient des outils utilitaires pour BIND9. Il comprend des programmes comme rndc (pour contrôler le serveur DNS), dig (pour interroger des serveurs DNS), et named-checkconf (pour vérifier la syntaxe du fichier de configuration de BIND). Ces outils sont essentiels pour diagnostiquer et gérer le serveur DNS BIND.

Modification du fichier dhcpd.conf :

```
GNU nano 7.2
subnet 192.168.209.0 netmask 255.255.255.0 {
    range 192.168.209.50 192.168.209.100;
    option subnet-mask 255.255.255.0;
    option routers 192.169.205.5;
    option broadcast-address 192.168.255.255;
    option domain-name-servers 192.168.209.5;
    option domain-name "starfleet.lan";
    default-lease-time 600;
    max-lease-time 600;
}

ddns-update-style none;
authoritative;
```

On ajoute les lignes entourées en rouge :

option domain-name-servers 192.168.209.5; : Cette ligne indique au serveur DHCP de fournir l'adresse IP 192.168.209.5 comme serveur DNS aux clients DHCP.

option domain-name "starfleet.lan"; : Cette ligne spécifie le nom de domaine starfleet.lan pour les clients DHCP.

Configuration named.conf.local :

```
GNU nano 7.2                                named.conf.local *
//
// Do any local configuration here
//

zone "starfleet.lan" {
    type master;
    file "/etc/bind/db.starfleet.lan";
};

zone "209.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192.168.209";
};

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
```

Ce fichier configure les zones DNS sur le serveur Bind9. Il spécifie deux zones

Zone directe pour starfleet.lan :

zone "starfleet.lan" : Cela définit la zone DNS pour le domaine starfleet.lan.

type master : Cela indique que ce serveur est le serveur maître pour cette zone, c'est-à-dire qu'il est responsable des enregistrements DNS principaux pour ce domaine.

file "/etc/bind/db.starfleet.lan" : Cela indique que les enregistrements DNS pour la zone starfleet.lan se trouvent dans le fichier /etc/bind/db.starfleet.lan.

Ce fichier contient les enregistrements pour résoudre des noms de domaine comme www8.starfleet.lan vers des adresses IP.

Zone inverse pour 209.168.192.in-addr.arpa :

zone "209.168.192.in-addr.arpa" : Cela définit la zone DNS pour la résolution inverse du sous-réseau 192.168.209.x. Cette zone permet de mapper les adresses IP vers des noms d'hôte.

file **"/etc/bind/db.192.168.209"** : Cela indique que les enregistrements DNS pour la zone inverse se trouvent dans le fichier /etc/bind/db.192.168.209.

Ce fichier contient des enregistrements PTR (Pointer Records) qui permettent de résoudre une adresse IP en un nom d'hôte.

Résumé : Ce fichier configure les zones DNS que le serveur gère :

- La zone **directe** pour **starfleet.lan** (résolution de **nom de domaine vers IP**).
- La zone **inverse** pour **192.168.209.x** (résolution **d'IP vers nom de domaine**).

Pourquoi utiliser la résolution inverse ?

La résolution inverse est utilisée dans plusieurs scénarios, comme :

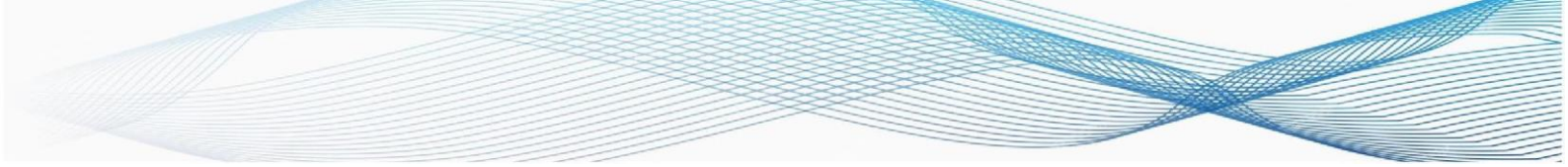
Diagnostic réseau : Lorsqu'on effectue un traceroute ou d'autres tests réseau, la résolution inverse est souvent utilisée pour associer une adresse IP à un nom de domaine pour mieux comprendre les résultats.

Journalisation et sécurité : Certains serveurs (comme les serveurs web ou de messagerie) peuvent utiliser la résolution inverse pour associer des adresses IP aux noms d'hôtes dans leurs journaux afin de faciliter l'analyse.

Configuration db.starfleet.lan :

```
GNU nano 7.2 db.starfleet.lan *
;
; Fichier de zone pour starfleet.lan
;
$TTL 604800
@      IN      SOA      ns1.starfleet.lan. admin.starfleet.lan. (
                                2      ; Numéro de série
                                604800 ; Rafraichir
                                86400  ; Réessayer
                                2419200 ; Expirer
                                604800 ); TTL cache négatif
;
@      IN      NS       ns1starfleet.lan.
ns1     IN      A        192.168.209.5 ; IP du serveur DNS

; Sous-domaines
www8    IN      A        192.169.209.5 ; Site web en PHP8
www7    IN      A        192.168.209.5 ; Site web en PHP7
php     IN      A        192.168.209.5 ; phpmyAdmin
admin   IN      A        192.168.209.5 ; interface admin_
```

db.starfleet.lan : est la zone directe pour le domaine starfleet.lan. Ce fichier définit la façon dont les noms de domaine sont résolus en adresses IP sur le serveur DNS.

@ : C'est un raccourci qui représente le domaine racine de la zone, ici starfleet.lan.

IN SOA : Signifie que c'est l'enregistrement Start of Authority (SOA). Il contient des informations sur le serveur DNS principal pour la zone et d'autres paramètres de la zone.

ns1.starfleet.lan. : C'est le serveur de noms principal (ou "autoritaire") pour cette zone.

admin.starfleet.lan. : Cela représente l'e-mail de l'administrateur du domaine (où le @ est remplacé par un point).

ns1 IN A 192.168.209.5 : Cet enregistrement a associé le nom de domaine ns1.starfleet.lan à l'adresse IP 192.168.209.5. Cela signifie que les requêtes pour ns1.starfleet.lan résoudront cette adresse IP, qui est celle de ton serveur DNS.

www8 IN A 192.168.209.5 : Associe www8.starfleet.lan à l'IP 192.168.209.5. Ce sous-domaine est configuré pour un site web utilisant PHP8.

Résolution de noms de domaine : L'enregistrement **A** permet à un client DNS (comme un navigateur web) de traduire un nom de domaine en une adresse IP qu'il peut utiliser pour se connecter au serveur. Les ordinateurs ne comprennent pas les noms de domaine comme www8.starfleet.lan, ils utilisent des adresses IP comme 192.168.209.5 pour communiquer.

Exemple :

Imaginons que l'on entre `www8.starfleet.lan` dans un navigateur :

Le navigateur demande au serveur DNS la résolution du nom `www8.starfleet.lan`.

Le serveur DNS consulte son enregistrement A et répond : `192.168.209.5`.

Le navigateur utilise cette adresse IP pour se connecter au serveur et charger le site web hébergé sur `192.168.209.5`.

Configuration `db.192.168.209` :

```
GNU nano 7.2                                     db.192.168.209
; Fichier de zone inverse pour 192.168.209.x
;
$TTL      604800
@         IN      SOA      ns1.starfleet.lan. admin.starfleet.lan. (
                                1          ; Numéro de série
                                604800     ; Rafraichir
                                86400      ; Réessayer
                                2419200    ; Expirer
                                604800     ; TTL cache négatif
;
@         IN      NS       ns1.starfleet.lan.
5         IN      PTR      ns1.starfleet.lan.
5         IN      PTR      www8.starfleet.lan.
5         IN      PTR      www7.starfleet.lan.
5         IN      PTR      php.starfleet.lan.
5         IN      PTR      admin.starfleet.lan.
```

Ce fichier est l'équivalent du fichier précédent mais en résolution inverse.

Type de résolution : L'enregistrement PTR est utilisé dans le processus de résolution DNS inverse (IP vers nom de domaine), contrairement à un enregistrement A, qui est utilisé pour la résolution directe (nom de domaine vers adresse IP).

Vérification :

```
root@debian-server:/etc/bind# named-checkzone 209.168.192.in-addr /etc/bind/db.192.168.209
zone 209.168.192.in-addr/IN: loaded serial 1
OK
root@debian-server:/etc/bind# named-checkzone starfleet.lan /etc/bind/db.starfleet.lan
zone starfleet.lan/IN: loaded serial 2
OK
```

Systemctl restart bind9 pour redémarrer notre DNS

On vérifie également que tout est bon côté client :

Commande: nslookup 192.168.209.5 ou **nslookup starfleet.lan**

```
jordan@debian:/etc$ nslookup 192.168.209.5
5.209.168.192.in-addr.arpa      name = www8.starfleet.lan.
5.209.168.192.in-addr.arpa      name = php.starfleet.lan.
5.209.168.192.in-addr.arpa      name = ns1.starfleet.lan.
5.209.168.192.in-addr.arpa      name = www7.starfleet.lan.
5.209.168.192.in-addr.arpa      name = admin.starfleet.lan.
```

```
jordan@debian:/etc$ nslookup www8.starfleet.lan
Server:      192.168.209.5
Address:     192.168.209.5#53
```

```
Name:   www8.starfleet.lan
Address: 192.168.209.5
```

4) Configuration Nginx

Pour installer nginx, on effectue les commandes suivantes :

Apt install nginx

Systemctl start nginx

Systemctl enable nginx

On vérifie maintenant que Nginx a été correctement installé :

Systemctl status nginx

```
root@starfleet:~# systemctl start nginx
root@starfleet:~# systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nginx
root@starfleet:~# systemctl status nginx
• nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
  Active: active (running) since Thu 2024-09-05 16:07:49 CEST; 51s ago
    Docs: man:nginx(8)
  Main PID: 1957 (nginx)
    Tasks: 5 (limit: 4281)
  Memory: 3.6M
    CPU: 215ms
  CGroup: /system.slice/nginx.service
          └─1957 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─1960 "nginx: worker process"
                └─1961 "nginx: worker process"
                   └─1962 "nginx: worker process"
                      └─1963 "nginx: worker process"
```

Vérification VM client :

<http://192.168.209.5> (ip du serveur)

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

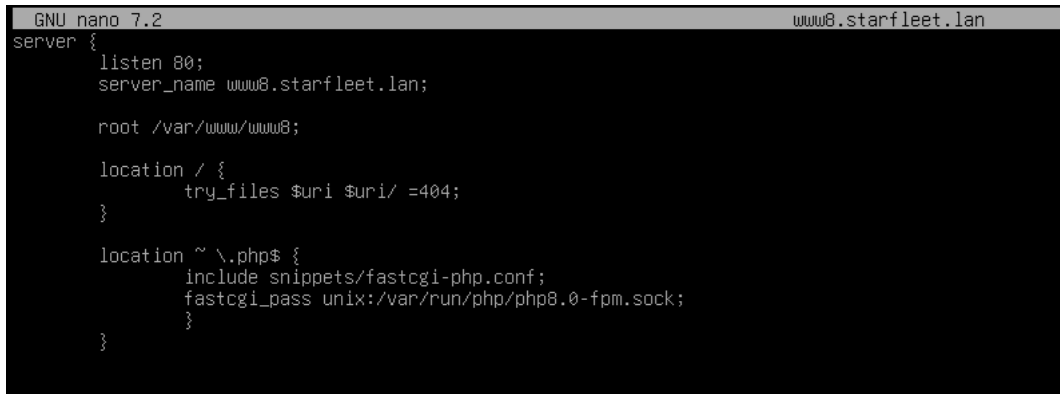
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Pour configurer Nginx pour servir `www8.starfleet.lan` via un hôte virtuel (Virtual Host), voici les étapes à suivre :

Créer le fichier de configuration pour `www8.starfleet.lan` :

nano /etc/nginx/sites-available/www8.starfleet.lan

A screenshot of the GNU nano 7.2 text editor. The title bar shows 'GNU nano 7.2' on the left and 'www8.starfleet.lan' on the right. The editor contains the following configuration for an Nginx server block:

```
server {  
    listen 80;  
    server_name www8.starfleet.lan;  
  
    root /var/www/www8;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
  
    location ~ \.php$ {  
        include snippets/fastcgi-php.conf;  
        fastcgi_pass unix:/var/run/php/php8.0-fpm.sock;  
    }  
}
```

listen 80 : Nginx écoute sur le port 80 (HTTP) pour les requêtes vers ce site.

server_name www8.starfleet.lan : Cette directive indique à Nginx que ce bloc de configuration doit être utilisé pour les requêtes dirigées vers `www8.starfleet.lan`.

root /var/www/www8 : Le répertoire où sont stockés les fichiers pour ce sous-domaine. Tu devras créer ce répertoire si ce n'est pas déjà fait.

location ~ \.php\$: Permet d'exécuter des fichiers PHP pour ce sous-domaine. Assure-toi que PHP et PHP-FPM sont correctement configurés.

Créer le répertoire pour `www8.starfleet.lan` :

Mkdir -p /var/www/www8

-p = création répertoire parent si non existant

On ajoute un fichier test pour le moment :

```
echo "Bienvenue sur www8.starfleet.lan" | tee  
/var/www/www8/index.html
```

tee : Reçoit la sortie et la duplique. Elle affiche le résultat à la fois dans le terminal (comme avec une commande normale) et l'enregistre dans un fichier spécifié.

On peut maintenant créer le lien symbolique :

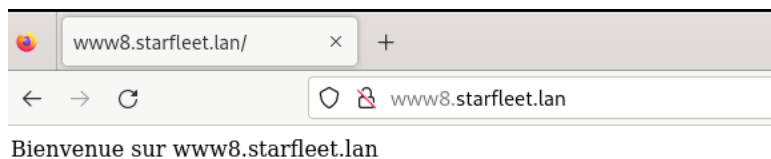
```
ln -s /etc/nginx/sites-available/www8.starfleet.lan  
/etc/nginx/sites-enabled/
```

Vérification du bon fonctionnement :

```
nginx -t
```

Si tout fonctionne, on redémarre le système : **systemctl restart nginx**

On regarde maintenant si tout fonctionne sur la machine client :



Utiliser les certificats auto-signés pour HTTPS :

On exécute la commande suivante pour générer le certificat auto-signé et la clé privée :

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
/etc/nginx/selfsigned.key -out /etc/nginx/selfsigned.crt
```

On répond à ces différentes question :

Country Name : FR

State or Province Name : PACA

Locality Name : Marseille

Organization Name : Starfleet

Common Name : www8.starfleet.lan.

On fait un fichier de paramètres Diffie-Hellman pour une sécurité renforcée :

```
openssl dhparam -out /etc/nginx/dhparam.pem 2048
```

Ensuite on modifie le fichier etc/nginx/sites-available/www8.starfleet.lan comme ceci :

```
GNU nano 7.2                               www8.starfleet.lan
server {
    listen 443 ssl;
    server_name www8.starfleet.lan;

    ssl_certificate /etc/nginx/selfsigned.crt;
    ssl_certificate_key /etc/nginx/selfsigned.key;
    ssl_dhparam /etc/nginx/dhparam.pem;

    root /var/www/www8;
    index index.html index.php;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.0-fpm.sock;
    }
}

server {
    listen 80;
    server_name www8.starfleet.lan;
    return 301 https://$host$request_uri;
}
```

Vérifie que la configuration Nginx est correcte :

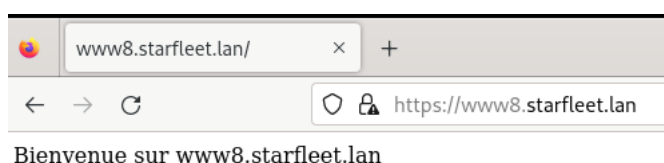
```
nginx -t
```

si jamais nous avons un soucis, on refait le lien symbolique vers enabled en supprimant le fichier précédent auparavant.

On redémarre Nginx pour appliquer les changements, une fois que la configuration nginx est correcte :

systemctl reload nginx

On test sur la machine client que tout fonctionne :



Installation et configuration PHP8 et PHP7 :

Ajouter le dépôt Sury à la liste des sources car il contient des versions récentes de PHP :

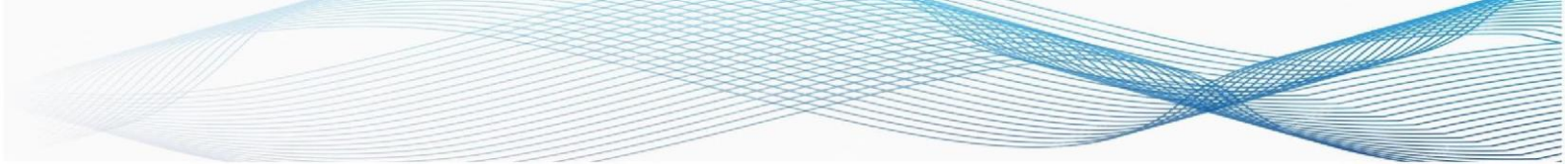
La commande ci-dessous ouvre un fichier appelé php.list dans le répertoire /etc/apt/sources.list.d/ pour que nous puissions ajouter le dépôt Sury.

nano /etc/apt/sources.list.d/php.list

Ensuite, nous y ajoutons la ligne suivante :

deb https://packages.sury.org/php/ bookworm main

Cela permet à APT (l'outil de gestion des paquets) de télécharger et installer les versions récentes de PHP depuis ce dépôt.



Ajouter la clé GPG pour le dépôt, cette clé est nécessaire pour authentifier les paquets téléchargés depuis le dépôt. Sans cette clé, Debian refusera d'installer les paquets. :

```
wget -O /etc/apt/trusted.gpg.d/sury-keyring.gpg  
https://packages.sury.org/php/apt.gpg
```

apt update

On installe PHP7 et PHP8 :

```
apt install php7.4 php7.4-fpm php8.0 php8.0-fpm
```

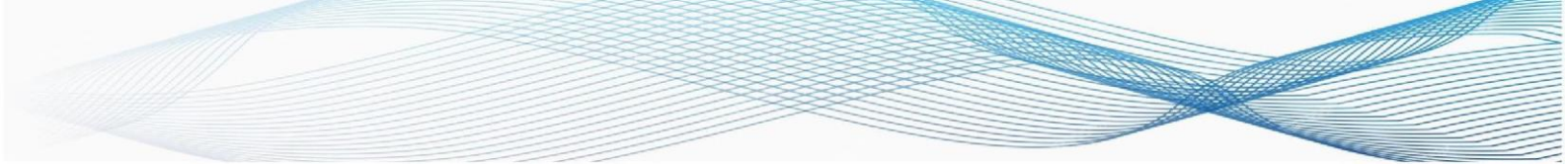
la commande ci-dessus installe :

php7.4 et php7.4-fpm : PHP 7.4 et son gestionnaire FastCGI Process Manager (FPM) qui est utilisé pour exécuter les scripts PHP dans Nginx.

php8.0 et php8.0-fpm : PHP 8.0 et son gestionnaire FPM

On vérifie que tout a été bien installé :

```
php7.4 -v  
php8.0 -v
```



```
root@starfleet:~# php7.4 -v
PHP 7.4.33 (cli) (built: Aug  2 2024 16:10:33) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.33, Copyright (c), by Zend Technologies
root@starfleet:~# php8.0 -v
PHP 8.0.30 (cli) (built: Aug  2 2024 16:09:56) ( NTS )
Copyright (c) The PHP Group
Zend Engine v4.0.30, Copyright (c) Zend Technologies
    with Zend OPcache v8.0.30, Copyright (c), by Zend Technologies
root@starfleet:~#
```

Ajouter un fichier phpinfo() : Pour vérifier que PHP fonctionne correctement :

nano /var/www/www8/phpinfo.php

```
<?php
phpinfo();
?>
```

Puis on test avec la machine client : `www8.starfleet.lan/phpinfo.php`

Il est important de noter qu'il est mieux de retirer ce fichier une fois le test effectué car cela affiche des informations sur le serveur et peut constituer une faille de sécurité.

Configuration pour `www7.starfleet.lan` :

mkdir -p /var/www/www7

chown -R www-data:www-data /var/www/www7

chmod -R 755 /var/www/www7

nano /etc/nginx/sites-available/www7.starfleet.lan

```
GNU nano 7.2                               www7.starfleet.lan
server {
    listen 80;
    server_name www7.starfleet.lan;

    # Redirection automatique de HTTP vers HTTPS
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name www7.starfleet.lan;

    ssl_certificate /etc/nginx/selfsigned.crt;
    ssl_certificate_key /etc/nginx/selfsigned.key;
    ssl_dhparam /etc/nginx/dhparam.pem;

    root /var/www/www7;
    index index.html index.php;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ /\.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    }

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers on;
}
```

**ln -s /etc/nginx/sites-available/www7.starfleet.lan
/etc/nginx/sites-enabled/**

nano /var/www/www7/index.html

```
GNU nano 7.2                               index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Test www7.starfleet.lan</title>
</head>
<body>
    <h1>Bienvenue sur www7.starfleet.lan</h1>
    <p>Ce site est servi avec Nginx et PHP 7.4.</p>
</body>
</html>
```



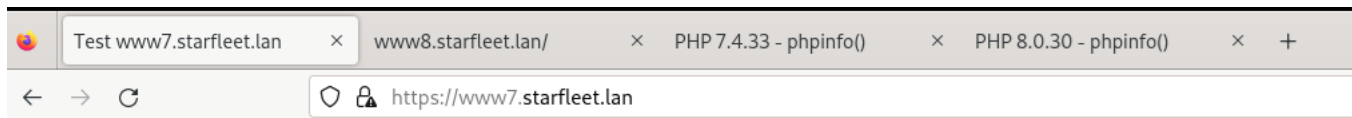
```
nano /var/www/www7/phpinfo.php
```

```
<?php  
phpinfo();  
?>
```

```
systemctl restart nginx
```

On vérifie maintenant avec la vm cliente que tout est bon pour
www7.starfleet.lan :


- la redirection
- https de base
- et la version php qui doit être en php7



Bienvenue sur www7.starfleet.lan

Ce site est servi avec Nginx et PHP 7.4.

PHP Version 7.4.33




System	Linux starfleet.lan 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64
Build Date	Aug 2 2024 16:10:33
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/fpm
Loaded Configuration File	/etc/php/7.4/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/fpm/conf.d
Additional .ini files parsed	/etc/php/7.4/fpm/conf.d/10-opcache.ini, /etc/php/7.4/fpm/conf.d/10-pdo.ini, /etc/php/7.4/fpm/conf.d/20-calendar.ini, /etc/php/7.4/fpm/conf.d/20-ctype.ini, /etc/php/7.4/fpm/conf.d/20-exif.ini, /etc/php/7.4/fpm/conf.d/20-fileinfo.ini, /etc/php/7.4/fpm/conf.d/20-ftp.ini, /etc/php/7.4/fpm/conf.d/20-gettext.ini, /etc/php/7.4/fpm/conf.d/20-iconv.ini, /etc/php/7.4/fpm/conf.d/20-json.ini, /etc/php/7.4/fpm/conf.d/20-phar.ini, /etc/php/7.4/fpm/conf.d/20-posix.ini, /etc/php/7.4/fpm/conf.d/20-readline.ini, /etc/php/7.4/fpm/conf.d/20-shmop.ini, /etc/php/7.4/fpm/conf.d/20-sockets.ini, /etc/php/7.4/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-sysvshm.ini, /etc/php/7.4/fpm/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902.NTS
PHP Extension Build	API20190902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:

Zend Engine v3.4.0, Copyright (c) Zend Technologies

with Zend OPcache v7.4.33, Copyright (c), by Zend Technologies



Configuration
calendar

Dans le cas où il y aurait un souci avec www7.starfleet.lan qui redirige vers www8.starfleet.lan, on vide le cache et l'historique de navigateur, cela devrait régler le problème.



Pour avoir la dernière version de Nginx :

Télécharger et ajouter la clé GPG correcte pour Nginx :

```
curl -fsSL https://nginx.org/keys/nginx_signing.key | gpg --dearmor -o  
/usr/share/keyrings/nginx-archive-keyring.gpg
```

Vérifier que le dépôt Nginx est bien ajouté à la liste des sources : Ouvrez le fichier de liste des dépôts :

```
nano /etc/apt/sources.list.d/nginx.list
```

```
deb [signed-by=/usr/share/keyrings/nginx-archive-keyring.gpg]  
http://nginx.org/packages/debian/ bookworm nginx
```

```
apt update
```

```
apt upgrade nginx
```

```
nginx -v
```

Modification du fichier de configuration nginx.conf :

On ouvre et modifie le fichier de configuration Nginx pour mettre à jour le bon utilisateur (www-data) et les chemins :

```
nano /etc/nginx/nginx.conf
```



```
GNU nano 7.2 nginx.conf
user www-data;
worker_processes auto;

error_log /var/log/nginx/error.log notice;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    #tcp_nopush on;

    keepalive_timeout 65;

    #gzip on;

    include /etc/nginx/sites-available/*;
}
```

Systemctl reload nginx

Systemctl restart nginx

Configurer MariaDB (base de données SQL) :

apt install mariadb-server mariadb-client

On vérifie que MariaDB s'est correctement installé :

systemctl status mariadb

```
● mariadb.service - MariaDB 10.11.6 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-09-10 12:47:02 CEST; 36s ago
     Docs: man:mariabdb(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 2244 (mariabdb)
  Status: "Taking your SQL requests now..."
    Tasks: 12 (limit: 4281)
  Memory: 207.6M
     CPU: 1.594s
   CGroup: /system.slice/mariadb.service
           └─2244 /usr/sbin/mariabdb

sept. 10 12:47:02 starfleet.lan mariabdb[2244]: 2024-09-10 12:47:02 0 [Note] Plugin 'FEEDBACK' is disabled.
sept. 10 12:47:02 starfleet.lan mariabdb[2244]: 2024-09-10 12:47:02 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
sept. 10 12:47:02 starfleet.lan mariabdb[2244]: 2024-09-10 12:47:02 0 [Warning] You need to use --log-bin to make --expire-logs-days or --binlog-expire-logs-seconds work.
sept. 10 12:47:02 starfleet.lan mariabdb[2244]: 2024-09-10 12:47:02 0 [Note] Server socket created on IP: '127.0.0.1'.
sept. 10 12:47:02 starfleet.lan mariabdb[2244]: 2024-09-10 12:47:02 0 [Note] InnoDB: Buffer pool(s) load completed at 240910 12:47:02
sept. 10 12:47:02 starfleet.lan mariabdb[2244]: 2024-09-10 12:47:02 0 [Note] /usr/sbin/mariabdb: ready for connections.
sept. 10 12:47:02 starfleet.lan mariabdb[2244]: Version: '10.11.6-MariaDB-0+deb12u1' socket: '/run/mysqld/mysqld.sock' port: 3306 Debian 12
sept. 10 12:47:02 starfleet.lan systemd[1]: Started mariadb.service - MariaDB 10.11.6 database server.
sept. 10 12:47:02 starfleet.lan /etc/mysql/debian-start[2259]: Upgrading MySQL tables if necessary.
sept. 10 12:47:03 starfleet.lan /etc/mysql/debian-start[2270]: Checking for insecure root accounts.
~
```

Pour se connecter à mysql :

mysql -u root

ensuite on crée une base de donnée :

```
CREATE DATABASE starfleet_db;
CREATE USER 'starfleet_user'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON starfleet_db.* TO
'starfleet_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Pour voir les utilisateurs MariaDB :

```
MariaDB [(none)]> SELECT User, Host FROM mysql.user;
```

User	Host
mariadb.sys	localhost
mysql	localhost
root	localhost
starfleet_user	localhost

```
4 rows in set (0,023 sec)
```

Pour voir les bases de données :

```
MariaDB [(none)]> SHOW DATABASES;
```

Database
information_schema
mysql
performance_schema
starfleet_db
sys

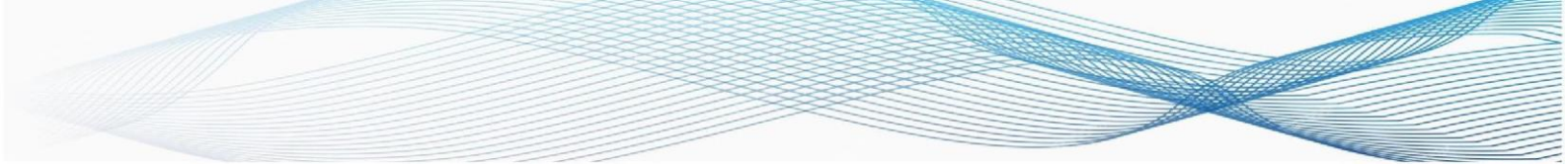
```
5 rows in set (0,010 sec)
```

```
MariaDB [(none)]> |
```

Pour voir les droits d'un utilisateur :

```
MariaDB [(none)]> SHOW GRANTS FOR 'starfleet_user'@'localhost';
```

Grants for starfleet_user@localhost
GRANT USAGE ON *.* TO 'starfleet_user'@'localhost' IDENTIFIED BY PASSWORD '*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19'
GRANT ALL PRIVILEGES ON 'starfleet_db'.* TO 'starfleet_user'@'localhost'



Si l'on veut sécuriser plus notre base de données, on fait la commande suivante :

mysql_secure_installation

Pour avoir la dernière version de MariaDB :

```
curl -Ls https://mariadb.org/mariadb_release_signing_key.asc |  
gpg --dearmor | tee /usr/share/keyrings/mariadb-keyring.gpg >  
/dev/null
```

nano /etc/apt/sources.list.d/mariadb.list

```
deb [signed-by=/usr/share/keyrings/mariadb-keyring.gpg]  
https://mirrors.xtom.com/mariadb/repo/10.11/debian bookworm main
```

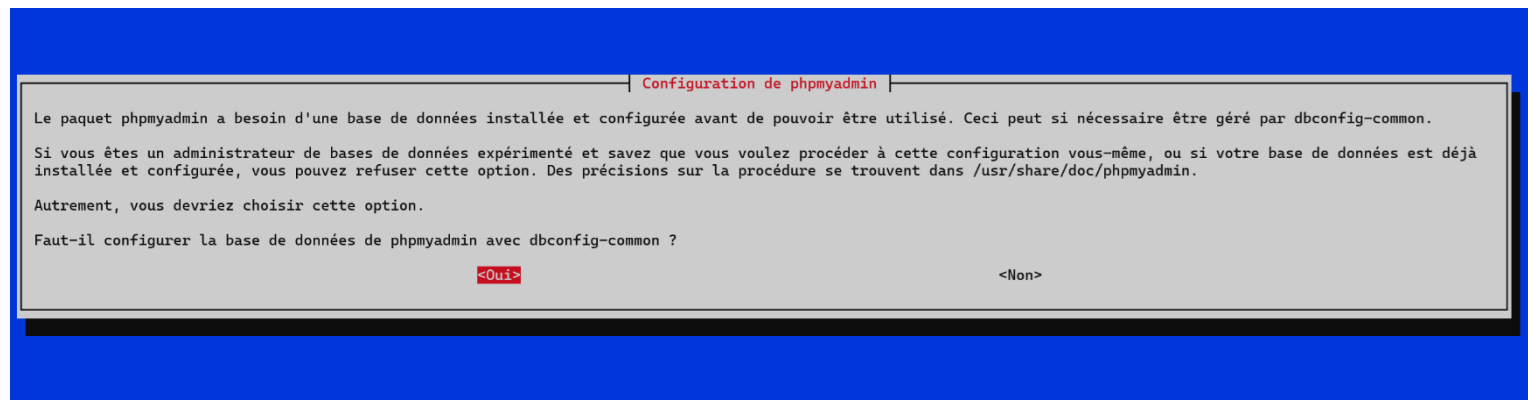
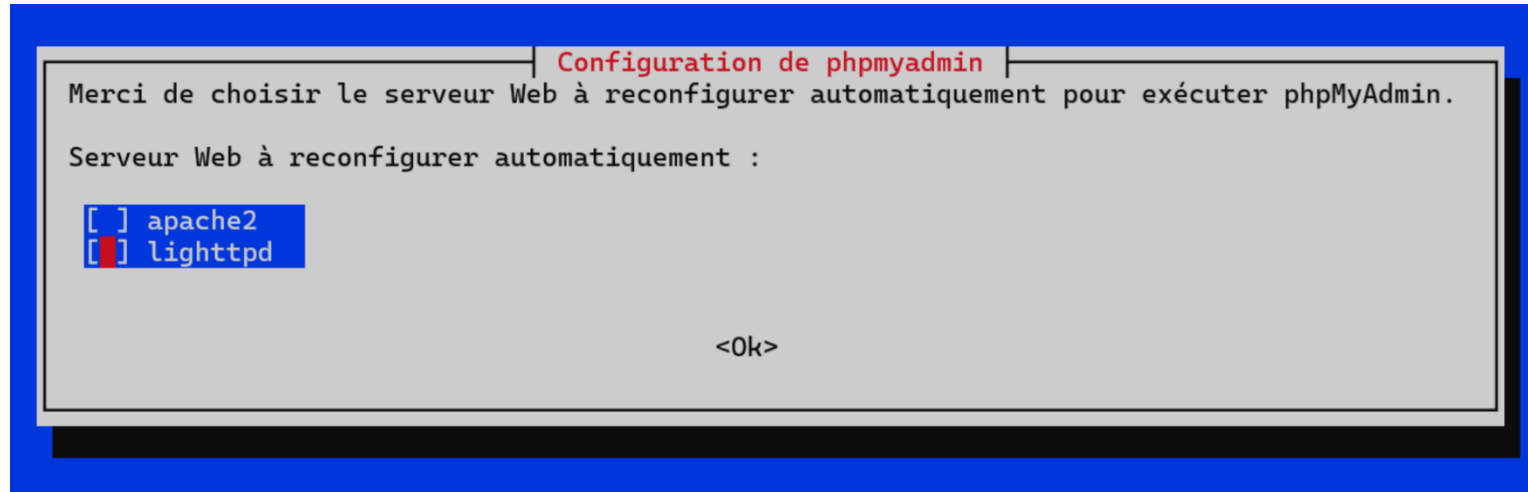
apt update

apt upgrade mariadb-server

mariadb --version

Installation PhPMyAdmin :

apt install phpmyadmin



Ensuite, nous avons dû créer des liens symboliques pour que Nginx puisse accéder aux fichiers de phpMyAdmin, car phpMyAdmin est installé par défaut dans /usr/share/phpmyadmin, et Nginx sert les fichiers à partir de /var/www :

ln -s /usr/share/phpmyadmin /var/www/phpmyadmin

nano /etc/nginx/sites-available/php.starfleet.lan

```
GNU nano 7.2 /etc/nginx/sites-available/php.starfleet.lan
server {
    listen 80;
    server_name php.starfleet.lan;

    # Redirection automatique vers HTTPS
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name php.starfleet.lan;

    # Certificats SSL (auto-signés ou émis par Let's Encrypt)
    ssl_certificate /etc/nginx/selfsigned.crt;
    ssl_certificate_key /etc/nginx/selfsigned.key;
    ssl_dhparam /etc/nginx/dhparam.pem;

    root /usr/share/phpmyadmin;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
    }
}
```

Une fois le fichier créé, nous avons activé ce site en créant un lien symbolique dans le dossier sites-enabled :

ln -s /etc/nginx/sites-available/php.starfleet.lan /etc/nginx/sites-enabled/

apt install php8.1 php8.1-fpm

apt install php8.1-mbstring php8.1-xml php8.1-zip php8.1-mysql

nginx -t

systemctl restart nginx

Configuration SFTP:

Création du groupe d'utilisateurs pour SFTP

On commence par créer un groupe d'utilisateurs spécifique qui sera dédié au service SFTP.

groupadd sftpusers

Cette commande crée un groupe nommé sftpusers dans lequel tous les utilisateurs SFTP seront ajoutés.

Modification du fichier de configuration SSH pour activer SFTP :

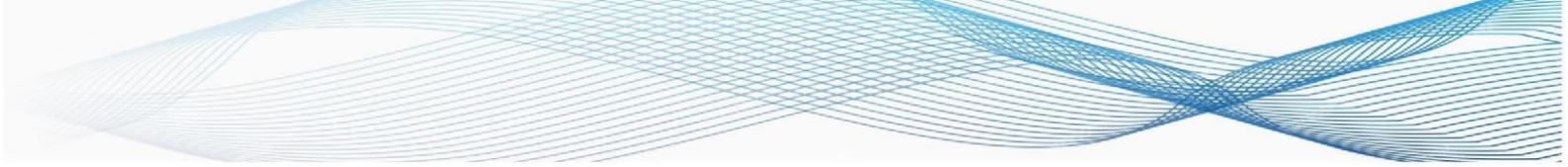
Ensuite, on doit modifier le fichier de configuration SSH pour spécifier que les utilisateurs du groupe sftpusers doivent utiliser le sous-système SFTP et non pas SSH classique.

nano /etc/ssh/sshd_config

```
# override default of no subsystems
#Subsystem      sftp      /usr/lib/openssh/sftp-server
Subsystem       sftp      internal-sftp
# Example of overriding settings on a per-user basis
Match Group sftpusers
    ChrootDirectory %h
    X11Forwarding no
    AllowTcpForwarding no
    ForceCommand internal-sftp
    GatewayPorts no
```

Subsystem sftp internal-sftp : Définit internal-sftp comme sous-système pour SFTP au lieu de l'ancienne méthode.

Match Group sftpusers : Applique les règles suivantes uniquement aux utilisateurs du groupe sftpusers.



ChrootDirectory %h : Chroot, c'est-à-dire que les utilisateurs seront enfermés dans leur répertoire personnel (%h correspond au répertoire de l'utilisateur).

X11Forwarding no et AllowTcpForwarding no : Ces options désactivent le forwarding pour des raisons de sécurité. ForceCommand internal-sftp : Force l'utilisation de SFTP uniquement, et non d'une session SSH classique.

Ajout d'un utilisateur au groupe SFTP :

useradd -G sftpusers -g www-data -s /usr/sbin/nologin -c "admin" -d /var/www/ admin

-G sftpusers : Ajoute l'utilisateur au groupe sftpusers.

-g www-data : Définit le groupe primaire de l'utilisateur comme étant www-data.

-s /usr/sbin/nologin : Empêche l'utilisateur de se connecter en SSH, il pourra seulement utiliser SFTP.

-d /var/www/ : Définit le répertoire personnel (home directory) de l'utilisateur comme étant /var/www/.

admin : Le nom de l'utilisateur à créer.

Définir le mot de passe de l'utilisateur :

passwd admin

Définir les droits d'accès au répertoire :

chown -R admin:www-data /var/www/html

systemctl restart ssh

Tester la connexion SFTP :

sftp admin@localhost

```
root@starfleet:~# sftp admin@localhost
admin@localhost's password:
Connected to localhost.
sftp> exit
root@starfleet:~# |
```

Installation de LDAP (slapd) et des paquets nécessaires :

apt update

apt install slapd ldap-utils

Lors de l'installation pour configurer slapd. Si l'on est invité à supprimer la base de données lors de la purge, "Non".

dpkg-reconfigure slapd

On rentre notre domaine LDAP : starfleet.lan.

On définit ensuite un mot de passe administrateur pour le compte cn=admin.

Configuration de la base de données LDAP :

cd /tmp



nano ou_people.ldif

dn: ou=people,dc=starfleet,dc=lan
objectClass: organizationalUnit
ou: people

Ajoutez de cette unité organisationnelle à la base LDAP :

ldapadd -x -D "cn=admin,dc=starfleet,dc=lan" -W -f ou_people.ldif

Création d'un utilisateur dans LDAP :

On génère un mot de passe crypté pour l'utilisateur : **slappasswd**

Création d'un fichier new_user.ldif dans /tmp pour l'utilisateur LDAP:

nano /tmp/new_user.ldif

dn: uid=jdoe,ou=people,dc=starfleet,dc=lan
objectClass: inetOrgPerson
cn: John Doe
sn: Doe
uid: jdoe
userPassword: {SSHA}MotDePasseCrypté

On remplace **motdepassecrypté** par le mot de passe que l'ont a eu précédemment avec : **slappasswd**

Ajoutez l'utilisateur dans la base LDAP :

ldapadd -x -D "cn=admin,dc=starfleet,dc=lan" -W -f /tmp/new_user.ldif

Vérifiez que l'utilisateur a bien été ajouté :

ldapsearch -x -LLL -b "dc=starfleet,dc=lan" "uid=jdoe"



Vérifiez que l'utilisateur peut s'authentifier :

```
ldapwhoami -x -D "uid=jdoe,ou=people,dc=starfleet,dc=lan" -W
```

Téléchargement et compilation de Nginx avec le module LDAP :

Installation des prérequis pour Nginx avec LDAP :

```
apt update
```

```
apt install -y build-essential libpcre3 libpcre3-dev zlib1g zlib1g-dev  
libssl-dev libldap2-dev libssl-dev git
```

Ensuite :

```
cd /usr/local/src
```

```
git clone https://github.com/kvspb/nginx-auth-ldap.git
```

```
wget https://nginx.org/download/nginx-1.26.2.tar.gz
```

```
tar zxvf nginx-1.26.2.tar.gz
```

```
cd nginx-1.26.2
```

Configurez et compilez Nginx avec le module LDAP :

```
./configure --with-http_ssl_module --add-  
module=/usr/local/src/nginx-auth-ldap
```

```
Make
```

```
make install
```



Remplacez l'ancienne version de Nginx :

```
mv /usr/sbin/nginx /usr/sbin/nginx.old
```

```
cp /usr/local/nginx/sbin/nginx /usr/sbin/nginx
```

```
nginx -v
```

Configuration de Nginx pour LDAP :

```
nano /etc/nginx/nginx.conf
```

Ajout le serveur LDAP dans la section http :

```
http {  
    ldap_server starfleet_ldap {  
        url  
ldap://localhost:389/ou=people,dc=starfleet,dc=lan?uid?sub?(objectClass=inetOrgPerson);  
        binddn "cn=admin,dc=starfleet,dc=lan";  
        binddn_passwd ton_mot_de_passe_admin;  
        group_attribute memberUid;  
        group_attribute_is_dn off;  
        require valid_user;  
    }  
  
    # Autres configurations...  
    include /etc/nginx/sites-available/*;  
}
```



```
user www-data;
worker_processes auto;

error_log /var/log/nginx/error.log notice;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    # Définir le serveur LDAP ici, dans la section http
    ldap_server starfleet_ldap {
        url ldap://localhost:389/ou=people,dc=starfleet,dc=lan?uid?sub?(objectClass=inetOrgPerson);
        binddn "cn=admin,dc=starfleet,dc=lan";
        binddn_passwd admin;
        group_attribute memberUid;
        group_attribute_is_dn off;
        require valid_user;
    }

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    #tcp_nopush on;

    keepalive_timeout 65;

    #gzip on;

    include /etc/nginx/sites-available/*;
}
```



Modifiez la configuration du site web (par exemple pour `www8.starfleet.lan`) :

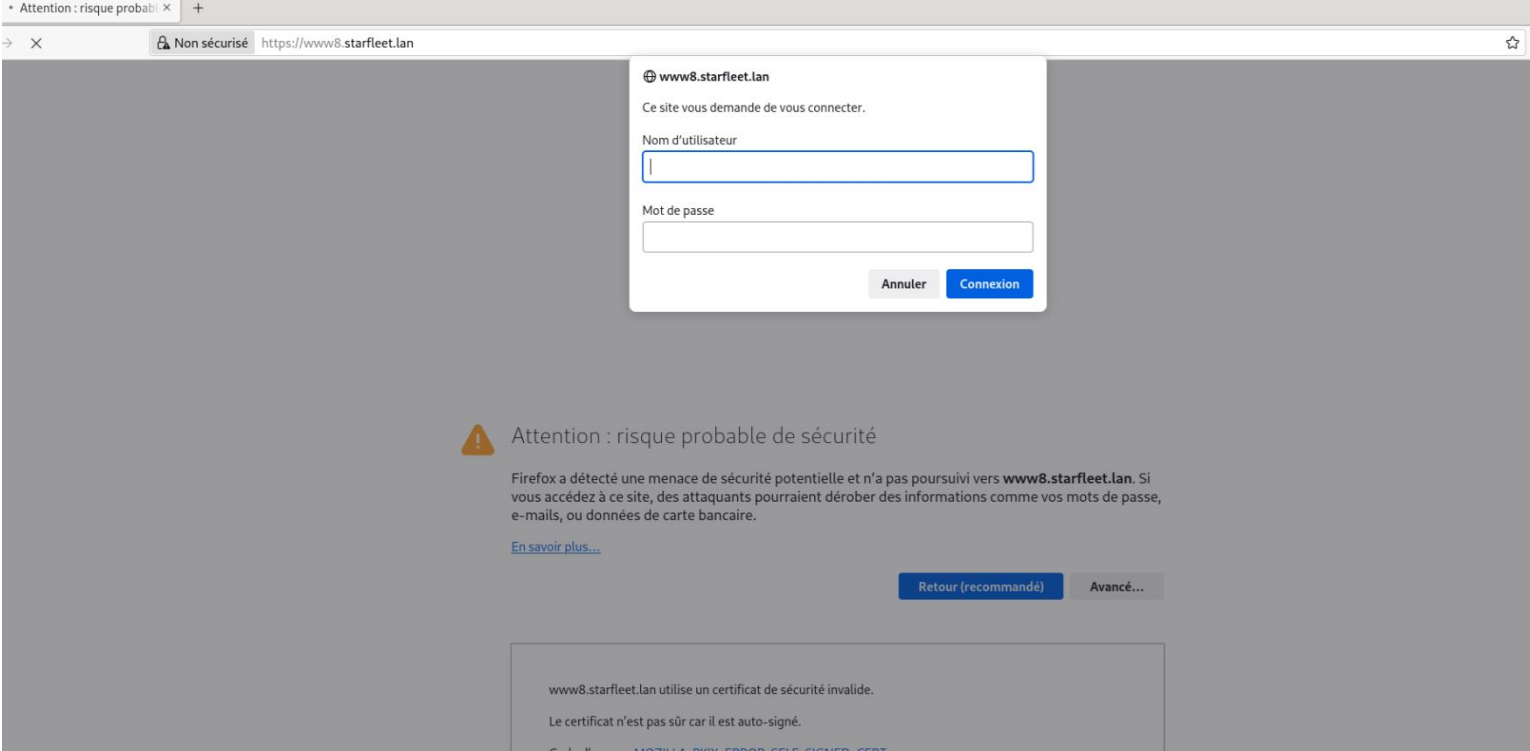
`nano /etc/nginx/sites-available/www8.starfleet.lan`

```
server {  
    listen 443 ssl;  
    server_name www8.starfleet.lan;  
  
    ssl_certificate /etc/nginx/selfsigned.crt;  
    ssl_certificate_key /etc/nginx/selfsigned.key;  
    ssl_dhparam /etc/nginx/dhparam.pem;  
  
    root /var/www/www8;  
    index index.html index.php;  
  
    location / {  
        auth_ldap "LDAP Authentication";  
        auth_ldap_servers starfleet_ldap;  
        try_files $uri $uri/ =404;  
    }  
  
    location ~ \.php$ {  
        include snippets/fastcgi-php.conf;  
        fastcgi_pass unix:/var/run/php/php8.0-fpm.sock;  
    }  
}  
  
server {  
    listen 80;  
    server_name www8.starfleet.lan;  
    return 301 https://$host$request_uri;  
}
```

nginx -t

systemctl restart nginx

Vérification de l'authentification LDAP via le site :





Problèmes potentiels et vérification des logs :

Si jamais quelque chose ne fonctionne pas, il faut vérifier les logs d'erreurs de Nginx :

```
tail -f /var/log/nginx/error.log
```

Installation Zabbix :

Nous avons suivi à la lettre ce tutoriel fonctionnel à partir du step 4 :

<https://technologyrss.com/how-to-install-zabbix-7-0-on-debian-12/>

Installation de UFW :

```
apt update
```

```
apt install ufw
```

```
ufw status
```

Autorisation des services par nom avec UFW :

```
ufw allow ssh
```

```
ufw allow http
```

```
ufw allow https
```

```
ufw allow dns
```

```
ufw allow ldap
```

```
ufw allow ldap
```




ufw allow ftp

ufw allow mysql

ufw allow zabbix-agent

ufw allow in on eth1

Autorisation des services par port avec UFW :

ufw allow 10051/tcp (Zabbix)

ufw allow 67/udp (DHCP)

root@starfleet:~# dnsmasq status verbose

Status: active

Logging: on (low)

Default: deny (incoming), allow (outgoing), disabled (routed)

New profiles: skip

To	Action	From
--	-----	----
22/tcp	ALLOW IN	Anywhere
80/tcp	ALLOW IN	Anywhere
443	ALLOW IN	Anywhere
53 (DNS)	ALLOW IN	Anywhere
389	ALLOW IN	Anywhere
21/tcp	ALLOW IN	Anywhere
3306/tcp	ALLOW IN	Anywhere
10050/tcp	ALLOW IN	Anywhere
Anywhere on eth1	ALLOW IN	Anywhere
67/udp	ALLOW IN	Anywhere
68/udp	ALLOW IN	Anywhere
10051/tcp	ALLOW IN	Anywhere
22/tcp (v6)	ALLOW IN	Anywhere (v6)
80/tcp (v6)	ALLOW IN	Anywhere (v6)
443 (v6)	ALLOW IN	Anywhere (v6)
53 (DNS (v6))	ALLOW IN	Anywhere (v6)
389 (v6)	ALLOW IN	Anywhere (v6)
21/tcp (v6)	ALLOW IN	Anywhere (v6)
3306/tcp (v6)	ALLOW IN	Anywhere (v6)
10050/tcp (v6)	ALLOW IN	Anywhere (v6)
Anywhere (v6) on eth1	ALLOW IN	Anywhere (v6)
67/udp (v6)	ALLOW IN	Anywhere (v6)
68/udp (v6)	ALLOW IN	Anywhere (v6)
10051/tcp (v6)	ALLOW IN	Anywhere (v6)

root@starfleet:~#

```
root@starfleet:~# ufw status
Status: active
```

To	Action	From
--	-----	----
22/tcp	ALLOW	Anywhere
80/tcp	ALLOW	Anywhere
443	ALLOW	Anywhere
DNS	ALLOW	Anywhere
389	ALLOW	Anywhere
21/tcp	ALLOW	Anywhere
3306/tcp	ALLOW	Anywhere
10050/tcp	ALLOW	Anywhere
Anywhere on eth1	ALLOW	Anywhere
67/udp	ALLOW	Anywhere
68/udp	ALLOW	Anywhere
10051/tcp	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)
80/tcp (v6)	ALLOW	Anywhere (v6)
443 (v6)	ALLOW	Anywhere (v6)
DNS (v6)	ALLOW	Anywhere (v6)
389 (v6)	ALLOW	Anywhere (v6)
21/tcp (v6)	ALLOW	Anywhere (v6)
3306/tcp (v6)	ALLOW	Anywhere (v6)
10050/tcp (v6)	ALLOW	Anywhere (v6)
Anywhere (v6) on eth1	ALLOW	Anywhere (v6)
67/udp (v6)	ALLOW	Anywhere (v6)
68/udp (v6)	ALLOW	Anywhere (v6)
10051/tcp (v6)	ALLOW	Anywhere (v6)