

Holodeck

```
dome@debian12-Sl: ~  
GNU nano 7.2 /etc/issue *  
Debian GNU/Linux 12 \n \l  
echo "IPv4 4\"
```

Première partie installations des paquets nécessaires

Tout d'abord le dhcp, dns et FTP

`apt update && sudo apt install -y isc-dhcp-server bind9 proftpd`

Installation de nginx, php et mariadb à la version demandé

1. Nginx 1.26 sur Debian:

Install the prerequisites:

`sudo apt install curl gnupg2 ca-certificates lsb-release debian-archive-keyring`

Import an official nginx signing key so apt could verify the packages authenticity.
Fetch the key:

**`curl https://nginx.org/keys/nginx_signing.key | gpg --dearmor \
| sudo tee /usr/share/keyrings/nginx-archive-keyring.gpg >/dev/null`**

Verify that the downloaded file contains the proper key:

**`gpg --dry-run --quiet --no-keyring --import --import-options import-show
/usr/share/keyrings/nginx-archive-keyring.gpg`**

The output should contain the full fingerprint

573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62 as follows:

```
pub  rsa2048 2011-08-19 [SC] [expires: 2027-05-24]  
     573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62  
uid           nginx signing key <signing-key@nginx.com>
```

Note that the output can contain other keys used to sign the packages.

To set up the apt repository for stable nginx packages, run the following command:

```
echo "deb [signed-by=/usr/share/keyrings/nginx-archive-keyring.gpg] \
http://nginx.org/packages/debian `lsb_release -cs` nginx" \
| sudo tee /etc/apt/sources.list.d/nginx.list
```

If you would like to use mainline nginx packages, run the following command instead:

```
echo "deb [signed-by=/usr/share/keyrings/nginx-archive-keyring.gpg] \
http://nginx.org/packages/mainline/debian `lsb_release -cs` nginx" \
| sudo tee /etc/apt/sources.list.d/nginx.list
```

Set up repository pinning to prefer our packages over distribution-provided ones:

```
echo -e "Package: *\nPin: origin nginx.org\nPin: release o=nginx\nPin-Priority:
900\n" \
| tee /etc/apt/preferences.d/99nginx
```

To install nginx, run the following commands:

```
apt update
apt install nginx
```

2. Installation Mariadb

il suffit d'aller sur le site officiel et installer la bonne version:

<https://mariadb.org/download/?t=repo-config&d=Debian+12+%22Bookworm%22&v=11.4&rm=icam>

3. Installation Php:

Par défaut, nous installons en tant qu'utilisateur root, si vous avez un utilisateur régulier, utilisez sudo.

```
install sury repository
curl -sSL https://packages.sury.org/php/README.txt | bash -x
apt update
```

```
show available PHP 7.4 packages
apt-cache search php7.4
```

```
apt install PHP 7.4 fpm && apt install php8.4-fpm
```

4. Installation Cockpit

Cockpit has been available in Debian since version 10 (Buster).

To get the latest version, we recommend to enable the [backports repository](#) (as root):
. /etc/os-release

```
echo "deb http://deb.debian.org/debian ${VERSION_CODENAME}-backports main" > \
/etc/apt/sources.list.d/backports.list
apt update
```

puis se rendre sur cette page:

<https://backports.debian.org/Instructions/>

Install or update the package:

```
apt install -t ${VERSION_CODENAME}-backports cockpit
```

5. Installation Firewall

Il est important d'installer le pare feu avant de couper internet

```
apt install ufw
```

6. Installation Cockpit

7. VSCode

Téléchargez la dernière version de VS Code Server

```
wget
```

[https://github.com/coder/code-server/releases/download/v4.16.1/code-server_4.16.1_amd64](https://github.com/coder/code-server/releases/download/v4.16.1/code-server_4.16.1_amd64.deb)
[.deb](#)

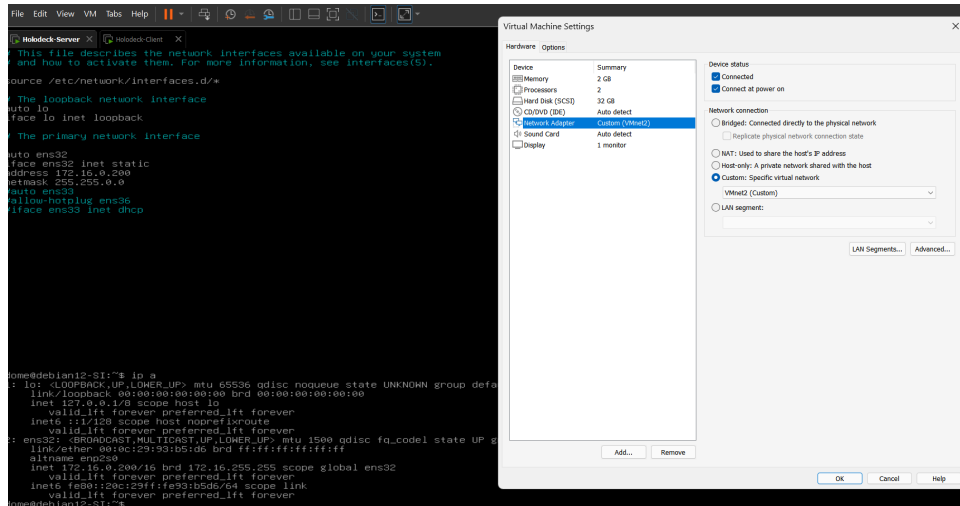
(Vérifiez la dernière version sur <https://github.com/coder/code-server/releases>)

Téléchargez la dernière version de VS Code Server

Deuxième partie configuration du réseau interne

Configuration DHCP

1. Définir un IP statique



Après avoir installé tous les paquets nécessaires, nous pouvons basculer la carte réseau de NAT Custom de la VMs Serveur et lui affecter un **ip static**

2. Configurer le DHCP sur la VM Server.

Éditez le fichier `/etc/resolv.conf` :

```
domain starfleet.lan
search starfleet.lan
nameserver 172.16.0.200
```

Après installation place à la configuration, je modifie le fichier `/etc/dhcp/dhcpd.conf` comme suit:

- j'ajoute le nom de domaine:
`option domain-name "starfleet.lan";`
`option domain-name-servers ns1.starfleet.lan, ns2.starfleet.lan;`
- active authoritative

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

default-lease-time 600;
max-lease-time 7200;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;
```

- *definie la zone dns*

```
GNU nano 7.2
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}

# This is a very basic subnet declaration.

subnet 172.16.0.0 netmask 255.255.0.0 {
    range 172.16.0.10 172.16.0.100;
    option routers 172.16.0.254;
    option domain-name "starfleet.lan";
    option domain-name-servers 172.16.0.200;
    option broadcast-address 172.16.0.255;
}
```

3. Spécifier l'interface réseau pour le DHCP

`nano /etc/default/isc-dhcp-server`

Ajouter la ligne suivante (remplacer eth1 par votre interface LAN) :

`INTERFACESv4="ens32"`

```
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="ens32"
INTERFACESv6=""
```

4. Redémarrer le service

`systemctl restart isc-dhcp-server`

`systemctl enable isc-dhcp-server`

5. Vérifier le statut du service

`systemctl status isc-dhcp-server`

`systemctl restart networking`

Dans la capture d'écran suivante une machine client après avoir monté une carte réseau custom et je modifié le fichier `/etc/network/interfaces` on peut constater qu'elle récupère bien un IP du server DHCP

```
GNU nano 7.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto ens33
iface ens33 inet dhcp
```

```
dome@debian12AI:~$ ip a
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:29:04:a2:a2 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 172.16.0.10/16 brd 172.16.0.255 scope global dynamic ens33
        valid_lft 485sec preferred_lft 485sec
    inet6 fe80::20c:29ff:fe04:a2a2/64 scope link
        valid_lft forever preferred_lft forever
dome@debian12AI:~$ ping 172.16.0.200
PING 172.16.0.200 (172.16.0.200) 56(84) bytes of data:
64 bytes from 172.16.0.200: icmp_seq=1 ttl=64 time=0.442 ms
64 bytes from 172.16.0.200: icmp_seq=2 ttl=64 time=0.685 ms
^C
--- 172.16.0.200 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1009ms
rtt min/avg/max/mdev = 0.442/0.563/0.685/0.121 ms
```

Configuration DNS

Configuration des fichiers principaux coté Serveur :

named.conf.local : pour définir les zones

Nous configurons le domaine "starfleet.lan" comme spécifié.

```
nano /etc/bind/named.conf.local
```

Ajouter les zones suivantes :

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

zone "16.172.in-addr.arpa" {
```

```
type master;

file "/etc/bind/db.172.16";

};
```

named.conf.options : pour les options globales

nano /etc/bind/named.conf.options

Ajouter:

```
options {

    directory "/var/cache/bind";

    listen-on { 127.0.0.1; 172.16.0.200; };

    allow-query { localhost; 172.16.0.0/24; };

    forwarders {

        8.8.8.8;

        8.8.4.4;

    };

};
```

!! "Nous utilisons 172.16.0.200 comme adresse IP du serveur DNS, correspondant à l'interface LAN" !!

Explication :

1. listen-on { 127.0.0.1; 172.16.0.200; };
 - 127.0.0.1 : C'est l'adresse de bouclage (localhost) du serveur.
 - 172.16.0.200 : C'est l'adresse IP réelle de votre serveur DNS dans le sous-réseau 172.16.0.0/24.
2. allow-query { localhost; 172.16.0.0/24; };
 - localhost : Permet les requêtes du serveur lui-même.
 - 172.16.0.0/24 : Autorise les requêtes de tout le sous-réseau 172.16.0.0.

Cette configuration est correcte pour votre réseau où le serveur a l'adresse 172.16.0.200 et fait partie du sous-réseau 172.16.0.0/24.

Création des fichiers de zone pour chaque domaine géré

Création du fichier de zone directe

`nano /etc/bind/db.starfleet.lan`

Contenu du fichier :

```
$TTL 604800
@ IN SOA ns1.starfleet.lan. admin.starfleet.lan. (
    2      ; Serial
    604800 ; Refresh
    86400  ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@ IN NS ns1.starfleet.lan.
ns1 IN A 172.16.0.200
www8 IN A 172.16.0.200
www7 IN A 172.16.0.200
php IN A 172.16.0.200
admin IN A 172.16.0.200
```

Nous avons ajouté les sous-domaines spécifiés (www8, www7, php, admin) pointant tous vers 172.16.0.200.

Assurez-vous que les noms de domaine dans les fichiers de zone se terminent par un point (.) pour indiquer qu'il s'agit de noms de domaine complets (FQDN).

Les enregistrements CNAME permettent de rediriger plusieurs sous-domaines vers la même adresse IP.

Création du fichier de zone inverse

`nano /etc/bind/db.172.16`

Contenu du fichier :

```
$TTL 604800
@ IN SOA ns1.starfleet.lan. admin.starfleet.lan. (
    1      ; Serial
    604800 ; Refresh
    86400  ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@ IN NS ns1.starfleet.lan.
1.0 IN PTR ns1.starfleet.lan.
200.0 IN PTR ns1.starfleet.lan.
```


Explication :

La zone "16.172.in-addr.arpa" que vous avez définie est en fait pour la résolution DNS inverse (reverse DNS). Expliquons pourquoi c'est important et comment cela fonctionne :

1. Résolution DNS normale (directe) :
 - Convertit un nom de domaine en adresse IP
 - Ex: www.starfleet.lan -> 172.16.0.200
2. Résolution DNS inverse :
 - Convertit une adresse IP en nom de domaine
 - Ex: 172.16.0.200 -> ns1.starfleet.lan

La zone "16.172.in-addr.arpa" est spécifiquement pour la résolution inverse des adresses IP dans le sous-réseau 172.16.0.0/16.

Pourquoi c'est important :

1. Vérification d'identité : Certains services utilisent la résolution inverse pour vérifier l'authenticité d'une connexion.
2. Logs et débogage : Les logs avec des noms de domaine sont souvent plus lisibles que ceux avec des adresses IP.
3. Configuration de serveurs de messagerie : Beaucoup de serveurs de messagerie vérifient la résolution inverse pour lutter contre le spam.
4. Complétude de la configuration DNS : Une configuration DNS complète inclut généralement à la fois la résolution directe et inverse.

Vérifiez que votre serveur DNS est correctement configuré comme serveur DNS sur votre système :

[nano /etc/resolv.conf](#)

Ajouter:

```
domain starfleet.lan
search starfleet.lan
nameserver 127.0.0.200
```

Concernant l'ajout de nameserver 172.16.0.1 dans /etc/resolv.conf :

- Il n'est généralement pas nécessaire d'ajouter 172.16.0.1 comme nameserver supplémentaire si 172.16.0.200 est votre serveur DNS principal et fonctionne correctement.
- Cependant, si 172.16.0.1 est également configuré comme un serveur DNS secondaire ou de secours dans votre réseau.

Vérifiez chaque zone

`named-checkzone starfleet.lan /etc/bind/db.starfleet.lan`

`named-checkzone 16.172.in-addr.arpa /etc/bind/db.172.16`

Démarrage et activation du service DNS

Redémarrer le service BIND9

`systemctl restart bind9`

Vérifier le statut du service

`systemctl status bind9`

Vérification Fonctionnement Server Dns:

Pour vérifier si votre serveur DNS fonctionne correctement, vous pouvez utiliser plusieurs méthodes. Voici quelques étapes pour tester votre configuration DNS :

1. Vérifiez d'abord que le service BIND9 est en cours d'exécution :

```
systemctl status named
```

Assurez-vous que le statut est "**active (running)**".

2. Utilisez la commande **dig** pour interroger votre serveur DNS :

```
dig @localhost starfleet.lan
```

```
dig @localhost -x 172.16.0.1
```

Remplacez "localhost" par l'adresse IP de votre serveur DNS si vous testez depuis une autre machine

3. Testez la résolution des sous-domaines que vous avez configurés :

```
dig @localhost www8.starfleet.lan
```

```
dig @localhost www7.starfleet.lan
```

```
dig @localhost php.starfleet.lan
```

```
dig @localhost admin.starfleet.lan
```

4. Vérifiez les logs DNS pour voir s'il y a des requêtes et des réponses :

```
sudo tail -f /var/log/syslog | grep named
```

5. Utilisez **nslookup** pour tester la résolution de noms :

```
nslookup starfleet.lan localhost
```

```
nslookup 172.16.0.200 localhost
```

6. Utilisez l'outil **host** pour vérifier les enregistrements DNS :

```
delv @localhost starfleet.lan
```

Si ces tests fonctionnent correctement, cela signifie que votre serveur DNS est opérationnel. Vous devriez voir les réponses appropriées pour chaque requête, correspondant aux enregistrements que vous avez configurés dans vos fichiers de zone.

Troisième partie configuration des services

Configurer le serveur web Nginx pour héberger les différents sites demandés

Création des dossiers pour chaque site

Nous créons un dossier distinct pour chaque site dans `/var/www/` si Dépôt Debian
`/usr/share/nginx/`.

Cela permet une séparation claire du contenu de chaque site.

`sudo mkdir -p /var/www/www8.starfleet.lan` Dépôt Debian

`mkdir -p /usr/share/nginx/www8.starfleet.lan` Dépôt Nginx demandé dans la consigne

`sudo mkdir -p /var/www/www7.starfleet.lan` Dépôt Debian

`mkdir -p /usr/share/nginx/www7.starfleet.lan` Dépôt Nginx demandé dans la consigne

`sudo mkdir -p /var/www/php.starfleet.lan` Dépôt Debian

`mkdir -p /usr/share/nginx/php.starfleet.lan` Dépôt Nginx demandé dans la consigne

`sudo mkdir -p /var/www/admin.starfleet.lan` Dépôt Debian

`mkdir -p /usr/share/nginx/admin.starfleet.lan` Dépôt Nginx demandé dans la consigne

Permissions

Nous attribuons la propriété des dossiers à l'utilisateur `www-data`, qui est l'utilisateur sous lequel Nginx s'exécute généralement.

`sudo chown -R www-data:www-data /var/www` Dépôt Debian

`chown -R www-data:www-data /usr/share/nginx/`

`sudo chmod -R 755 /var/www` Dépôt Debian

`chmod -R 755 /usr/share/nginx/`

Vous n'êtes pas obligé d'utiliser l'utilisateur `www-data` avec Nginx. C'est une pratique courante, mais il existe d'autres options. Voici quelques points importants à ce sujet :

- *Utilisateur par défaut :*

- Sur de nombreuses distributions Linux, `www-data` est l'utilisateur par défaut pour Nginx (et Apache).
- Cela vient de la configuration par défaut et de l'histoire de ces serveurs web.
- *Raisons d'utiliser `www-data` :*
 - *Sécurité : Cet utilisateur a généralement des privilèges limités, ce qui est bon pour la sécurité.*
 - *Compatibilité : Beaucoup de guides et de configurations supposent l'utilisation de `www-data`.*
- *Alternatives :*
 - *Vous pouvez créer un utilisateur spécifique pour Nginx.*
 - *Certains préfèrent utiliser l'utilisateur `nginx`.*
- *Comment changer l'utilisateur :*
 - *Modifiez le fichier de configuration principal de Nginx (`/etc/nginx/nginx.conf`)*

:user votre_utilisateur;
 - *Assurez-vous que cet utilisateur a les permissions nécessaires sur les fichiers et dossiers web.*
- *Considérations de sécurité : Quel que soit l'utilisateur choisi, il doit avoir des privilèges limités.*
- *Évitez d'utiliser `root` comme utilisateur Nginx pour des raisons de sécurité.*
- *Permissions : Si vous changez l'utilisateur, n'oubliez pas de mettre à jour les permissions des fichiers et dossiers web :*

sudo chown -R votre_utilisateur:votre_groupe /var/www
- *Cohérence : Si vous décidez de ne pas utiliser `www-data`, assurez-vous d'utiliser le même utilisateur de manière cohérente dans toute votre configuration.*
- *En résumé, bien que `www-data` soit couramment utilisé, vous avez la flexibilité de choisir ou de créer un autre utilisateur pour Nginx. L'important est de maintenir une bonne sécurité et une gestion cohérente des permissions.*

Création de l'utilisateur

1. Créons un utilisateur système nommé `starfleet_web` sans répertoire home et sans possibilité de connexion.

```
sudo adduser --system --no-create-home --disabled-login --group
starfleet_web
```

Modification de la configuration Nginx :

2. Changeons l'utilisateur dans le fichier de configuration principal de Nginx.

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

```
user starfleet_web;
```

Changement de propriété :

3. Attribuons la propriété des fichiers web au nouvel utilisateur.

```
sudo chown -R starfleet_web:starfleet_web /var/www
```

Ajustement des permissions :

4. Nous nous assurons que les permissions sont correctes pour le fonctionnement de Nginx.

```
sudo find /var/www -type d -exec chmod 755 {} \;
```

```
sudo find /var/www -type f -exec chmod 644 {} \;
```

Vérification et redémarrage :

5. Vérifions la configuration et redémarrons Nginx pour appliquer les changements.

```
sudo nginx -t
```

```
sudo systemctl restart nginx
```

Vérification du processus :

6. Nous confirmons que Nginx fonctionne bien sous le nouvel utilisateur.

```
ps aux | grep nginx
```

Configuration PHP-FPM :

7. Si vous utilisez PHP, il faut également mettre à jour sa configuration pour utiliser le nouvel utilisateur.

```
sudo nano /etc/php/7.4/fpm/pool.d/www.conf
```

```
user = starfleet_web
```

```
group = starfleet_web
```

```
sudo systemctl restart php7.4-fpm
```

Vérification des logs :

Nous surveillons les logs pour détecter d'éventuels problèmes.

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

Points importants à noter :

- Cette modification peut avoir des implications sur la sécurité et le fonctionnement de votre serveur.
- Assurez-vous de tester minutieusement après ces changements.
- Vous devrez peut-être ajuster d'autres configurations ou scripts qui supposent l'utilisation de www-data.

Les permissions 755 permettent à Nginx de lire et exécuter les fichiers, tout en maintenant une sécurité adéquate.

Pages d'index

Nous créons une page d'index simple pour chaque site.

```
echo "<h1>Welcome to www8.starfleet.lan</h1>" | sudo tee
/var/www/www8.starfleet.lan/index.html Dépôt Debian
```

```
echo "<h1>Welcome to www8.starfleet.lan</h1>" | sudo tee
/usr/share/nginx/www8.starfleet.lan/index.html Dépôt Nginx demandé dans la consigne
```

```
echo "<h1>Welcome to www7.starfleet.lan</h1>" | sudo tee
/var/www/www8.starfleet.lan/index.html Dépôt Debian
```

```
echo "<h1>Welcome to www7.starfleet.lan</h1>" | sudo tee
/usr/share/nginx/www7.starfleet.lan/index.html Dépôt Nginx demandé dans la consigne
```

```
echo "<?php phpinfo(); ?>" | sudo tee /var/www/php.starfleet.lan/index.php Dépôt Debian
```

```
echo "<?php phpinfo(); ?>" | sudo tee /usr/share/nginx/php.starfleet.lan/index.php Dépôt
Nginx demandé dans la consigne
```

```
echo "<h1>Welcome to admin.starfleet.lan</h1>" | sudo tee
/var/www/admin.starfleet.lan/index.html Dépôt Debian
```

```
echo "<h1>Welcome to admin.starfleet.lan</h1>" | sudo tee
/usr/share/nginx/admin.starfleet.lan/index.html Dépôt Nginx demandé dans la consigne
```

Pour le site PHP, nous utilisons une page PHP basique qui affiche les informations de configuration PHP.

/

Modification du fichier nginx.conf

Cet étape est surtout importante pour la version *Dépôt Nginx* et non pas pour le dépôt *Debian*.

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

Modifier :

user www-data; car le dépôt Nginx prend en compte le user=Nginx à la place

include /etc/nginx/sites-enabled/.conf* à différence du dépôt Debian elle n'est pas par défaut

Configuration des virtual hosts

Chaque site aura son propre fichier de configuration dans */etc/nginx/sites-available/*.

```
mkdir -p /etc/nginx/sites-available/www8.starfleet.lan.conf
```

```
mkdir -p /etc/nginx/sites-available/www7.starfleet.lan.conf
```

```
mkdir -p /etc/nginx/sites-available/php.starfleet.lan.conf
```

```
mkdir -p /etc/nginx/sites-available/admin.starfleet.lan.conf
```

Ces fichiers définissent comment Nginx doit gérer les requêtes pour chaque domaine.

Activation des sites

Créer les liens symboliques dans */etc/nginx/sites-enabled/* Activation des sites

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

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

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

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

```
dome@debian12-SI:/etc/nginx/sites-available$ ls  
admin.starfleet.lan.conf  php.starfleet.lan.conf  www7.starfleet.lan.conf  www8.starfleet.lan.conf  
dome@debian12-SI:/etc/nginx/sites-available$ cd /etc/nginx/sites-enabled/  
dome@debian12-SI:/etc/nginx/sites-enabled$ ls  
admin.starfleet.lan.conf  php.starfleet.lan.conf  www7.starfleet.lan.conf  www8.starfleet.lan.conf  
dome@debian12-SI:/etc/nginx/sites-enabled$
```


Vérification et démarrage

Nous vérifions la syntaxe de la configuration Nginx avant de redémarrer le service

Vérification de la configuration Nginx

nginx -t

Redémarrage de Nginx

systemctl restart nginx

```
dome@debian12-SI:/etc/nginx/sites-available$ sudo mv admin.starfleet.lan admin.starfleet.lan.conf
dome@debian12-SI:/etc/nginx/sites-available$ sudo mv php.starfleet.lan php.starfleet.lan.conf
dome@debian12-SI:/etc/nginx/sites-available$ sydo mv www7.starfleet.lan www7.starfleet.lan.conf
-bash: sydo : commande introuvable
dome@debian12-SI:/etc/nginx/sites-available$ sudo mv www7.starfleet.lan www7.starfleet.lan.conf
dome@debian12-SI:/etc/nginx/sites-available$ sudo mv www8.starfleet.lan www8.starfleet.lan.conf
dome@debian12-SI:/etc/nginx/sites-available$ ls
admin.starfleet.lan.conf  php.starfleet.lan.conf  www7.starfleet.lan.conf  www8.starfleet.lan.conf
dome@debian12-SI:/etc/nginx/sites-available$ cd ..
dome@debian12-SI:/etc/nginx$ cd sites-enabled/
dome@debian12-SI:/etc/nginx/sites-enabled$ ls
dome@debian12-SI:/etc/nginx/sites-enabled$ sudo ln -s /etc/nginx/sites-available/admin.starfleet.lan.conf /etc/nginx/sites-enabled/
dome@debian12-SI:/etc/nginx/sites-enabled$ sudo ln -s /etc/nginx/sites-available/php.starfleet.lan.conf /etc/nginx/sites-enabled/
dome@debian12-SI:/etc/nginx/sites-enabled$ sudo ln -s /etc/nginx/sites-available/www7.starfleet.lan.conf /etc/nginx/sites-enabled/
dome@debian12-SI:/etc/nginx/sites-enabled$ sudo ln -s /etc/nginx/sites-available/www8.starfleet.lan.conf /etc/nginx/sites-enabled/
dome@debian12-SI:/etc/nginx/sites-enabled$ ls -l /etc/nginx/sites-enabled/
total 0
lrwxrwxrwx 1 root root 51  9 sept. 15:58 admin.starfleet.lan.conf -> /etc/nginx/sites-available/admin.starfleet.lan.conf
lrwxrwxrwx 1 root root 49  9 sept. 15:58 php.starfleet.lan.conf -> /etc/nginx/sites-available/php.starfleet.lan.conf
lrwxrwxrwx 1 root root 50  9 sept. 15:58 www7.starfleet.lan.conf -> /etc/nginx/sites-available/www7.starfleet.lan.conf
lrwxrwxrwx 1 root root 50  9 sept. 15:58 www8.starfleet.lan.conf -> /etc/nginx/sites-available/www8.starfleet.lan.conf
dome@debian12-SI:/etc/nginx/sites-enabled$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
dome@debian12-SI:/etc/nginx/sites-enabled$ ss
```

Création certificat auto-signé

Pour faire cela j'ai suivi le tuto suivant:

<https://www.tremplin-numerique.org/comment-creer-et-utiliser-un-ssl-auto-signe-dans-nginx-cloudsavvy-it>

Configuration Certificat SSL et restriction HTTPS

D'abord, assurons-nous que votre certificat auto-signé est bien en place. Généralement, ils sont stockés dans `/etc/ssl/certs/` pour le certificat public et `/etc/ssl/private/` pour la clé privée. Une fois que nous avons confirmé l'emplacement, nous allons modifier chaque fichier de configuration de site pour activer HTTPS.

Vérifier les droits de lecture écriture seulement pour root pour clé privé.et certificat tous les autre en lecture

Voici un exemple de configuration pour un site :

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

Répétez cette configuration pour chaque site (www7, php, admin), en ajustant le `server_name` et le `root` appropriés.

```
server {  
    listen 80;  
  
    server_name www8.starfleet.lan;  
  
    return 301 https://$server_name$request_uri;  
}  
  
server {  
    listen 443 ssl;  
  
    server_name www8.starfleet.lan;  
  
    ssl_certificate /etc/ssl/certs/nginx.crt;  
    ssl_certificate_key /etc/ssl/private/nginx.key;  
  
    root /usr/share/nginx/www8.starfleet.lan;  
    index index.html index.htm index.php;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
  
    location ~ \.php$ {  
        root /usr/share/nginx/www8.starfleet.lan;  
  
        fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;  
        fastcgi_index index.php;  
        include fastcgi_params;  
    }  
}
```

```
}
```

Après avoir modifié tous les fichiers, vérifiez la configuration Nginx :

```
nginx -t
```

Si tout est correct, rechargez Nginx :

```
systemctl reload nginx
```

Quelques points importants :

- Assurez-vous que le chemin vers le socket PHP (`fastcgi_pass`) correspond à la version de PHP que vous utilisez.
- Si vous avez des configurations spécifiques pour certains sites (comme des règles de réécriture), assurez-vous de les inclure dans la nouvelle configuration HTTPS.
- Le block `server` écoutant sur le port 80 redirige tout le trafic HTTP vers HTTPS.

Après avoir effectué ces modifications, vos sites devraient être accessibles via HTTPS. N'oubliez pas que, comme il s'agit d'un certificat auto-signé, les navigateurs afficheront un avertissement de sécurité. C'est normal pour un environnement de test ou de développement.

Pare feu

Activez UFW si ce n'est pas déjà fait :

```
ufw enable
```

Nous allons ajouter les règles manuellement pour ouvrir les ports nécessaires pour Nginx.

```
ufw allow 80/tcp
```

```
ufw allow 443/tcp
```

Au passage ouvrir les ports nécessaires avec UFW : Pour DNS (port 53, UDP et TCP) :

```
ufw allow 53/udp
```

```
ufw allow 53/tcp
```

Pour DHCP (ports 67 et 68, UDP) :

```
ufw allow 67/udp
```

```
ufw allow 68/udp
```

Vérifier le statut de UFW après ces ajouts :

`ufw status numbered`

Vérifiez que les services écoutent sur les bons ports :

`ss -tulnp | grep named`

`ss -tulnp | grep dhcp`

Si vous avez modifié la configuration du pare-feu, redémarrez les services pour s'assurer qu'ils utilisent les nouvelles règles :

`systemctl restart named`

`systemctl restart isc-dhcp-server`

Assurez-vous que Nginx peut lire la clé :

`chown root:www-data /etc/ssl/private/nginx.key`

Vérifiez le statut de UFW pour vous assurer que les règles ont été ajoutées correctement :

`ufw status`

```
dome@debian12-SI:~$ sudo ufw status
Status: active

To Action From
--
80/tcp ALLOW Anywhere
443/tcp ALLOW Anywhere
22/tcp ALLOW Anywhere
53/tcp ALLOW Anywhere
67/udp ALLOW Anywhere
68/udp ALLOW Anywhere
53/udp ALLOW Anywhere
68/udp (v6) ALLOW Anywhere (v6)
53/udp (v6) ALLOW Anywhere (v6)

dome@debian12-SI:~$ S
```

Si vous rencontrez des problèmes, vous pouvez toujours désactiver temporairement le pare-feu pour dépanner :

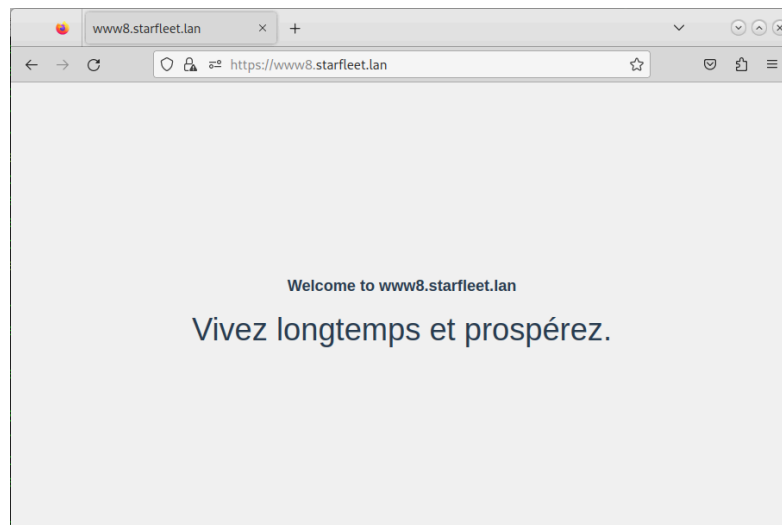
`ufw disable`

N'oubliez pas de sauvegarder votre configuration UFW :

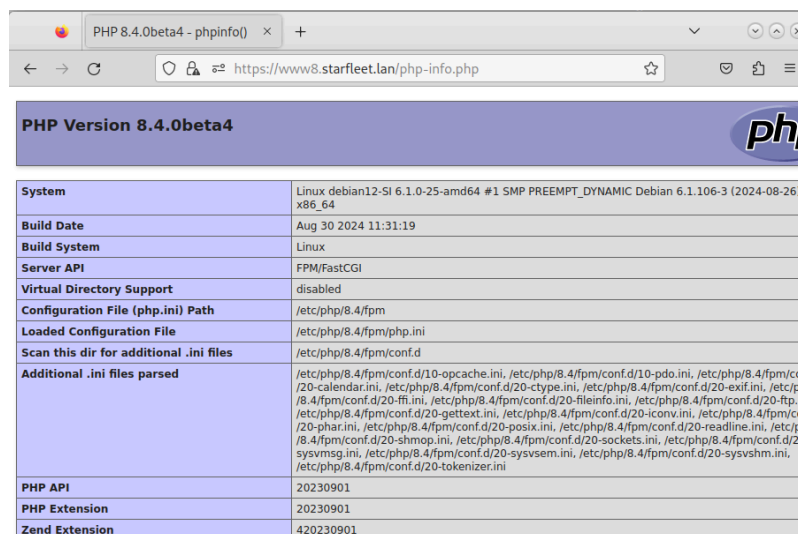
`cp /etc/ufw/user.rules /etc/ufw/user.rules.backup`

A ce stade on doit pouvoir ouvrir la page html de chaque site dans un terminal

exemple <https://www8.starfleet.lan>



et la page php exemple <https://www8.starfleet.lan/php-info.php>



PHP Version 8.4.0beta4	
System	Linux debian12-SI 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64
Build Date	Aug 30 2024 11:31:19
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.4/fpm
Loaded Configuration File	/etc/php/8.4/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.4/fpm/conf.d
Additional .ini files parsed	/etc/php/8.4/fpm/conf.d/10-opcache.ini, /etc/php/8.4/fpm/conf.d/10-pdo.ini, /etc/php/8.4/fpm/conf.d/20-calendar.ini, /etc/php/8.4/fpm/conf.d/20-ctype.ini, /etc/php/8.4/fpm/conf.d/20-exif.ini, /etc/php/8.4/fpm/conf.d/20-ffi.ini, /etc/php/8.4/fpm/conf.d/20-fileinfo.ini, /etc/php/8.4/fpm/conf.d/20-ftp.ini, /etc/php/8.4/fpm/conf.d/20-gettext.ini, /etc/php/8.4/fpm/conf.d/20-iconv.ini, /etc/php/8.4/fpm/conf.d/20-phar.ini, /etc/php/8.4/fpm/conf.d/20-posix.ini, /etc/php/8.4/fpm/conf.d/20-readline.ini, /etc/php/8.4/fpm/conf.d/20-shmop.ini, /etc/php/8.4/fpm/conf.d/20-sockets.ini, /etc/php/8.4/fpm/conf.d/20-sysvmsg.ini, /etc/php/8.4/fpm/conf.d/20-sysvsem.ini, /etc/php/8.4/fpm/conf.d/20-sysvshm.ini, /etc/php/8.4/fpm/conf.d/20-tokenizer.ini
PHP API	20230901
PHP Extension	20230901
Zend Extension	420230901

Commandes importantes à retenir de ufw :

tail -f /var/log/ufw.log pour voir les log

ufw status numbered pour lister le port offerts dans une liste chiffrées

ufw delete 14 pour supprimer en par exemple le n 14 de la liste

nginx -T | grep root Vérifiez le répertoire racine actuel de Nginx

FTP

Modifier le fichier de config [/etc/proftpd/proftpd.conf](#)

Assurez-vous que les lignes suivantes sont présentes et non commentées :

[UseIPv6 off](#)

[ServerName "Debian FTP server"](#)

[DefaultRoot ~](#)

[RequireValidShell off](#)

Ces lignes signifient :

- [ServerName](#) : Le nom de votre serveur FTP
- [DefaultRoot ~](#) : Limite les utilisateurs à leur répertoire personnel
- [RequireValidShell off](#) : Permet aux utilisateurs sans shell valide de se connecter

Et ajouter les paramètres concernant l'User

[# Activer l'accès pour développeur](#)

[<Directory /usr/share/nginx>](#)

[# Interdire l'écriture dans ce répertoire](#)

[<Limit WRITE>](#)

[AllowAll](#)

[</Limit>](#)

[# Autoriser la lecture dans ce répertoire](#)

[<Limit READ>](#)

[AllowAll](#)

[</Limit>](#)

[</Directory>](#)

Restreindre toto à son répertoire (chroot)

[DefaultRoot /usr/share/nginx](#)

Configurer les ports passifs pour les connexions FTP

[PassivePorts 60000 65535](#)

Chroot : Le chroot est correctement configuré avec la ligne :

[DefaultRoot /usr/share/nginx](#)

Le support SSL/TLS n'est pas activé dans cette configuration. La ligne suivante est commentée :

[include /etc/proftpd/tls.conf](#)

SSL/TLS :

Configurer correctement le fichier [tls.conf](#).

Éditez le fichier `/etc/proftpd/tls.conf` et assurez-vous qu'il contient :

```
<IfModule mod_tls.c>
    TLSEngine                on
    TLSLog                   /var/log/proftpd/tls.log
    TLSProtocol               TLSv1.2 TLSv1.3
    TLSRSACertificateFile     /etc/ssl/certs/nginx.crt j'ai utilisé le même certificat de nginx
    TLSRSACertificateKeyFile  /etc/ssl/private/nginx.key j'ai utilisé le même certificat de nginx
    TLSVerifyClient           off
    TLSOptions                NoSessionReuseRequired
    RequireValidShell         off
</IfModule>
```

Générez un certificat SSL si ce n'est pas déjà fait :

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/proftpd.key
-out /etc/ssl/certs/proftpd.crt
```

Redémarrez ProFTPD :

```
systemctl restart proftpd
```

Vérifiez que le service est en cours d'exécution :

```
systemctl status proftpd
```

Configuration de l'authentification

Par défaut, ProFTPD utilise l'authentification du système. Si vous voulez créer des utilisateurs FTP spécifiques :

```
adduser developpeur
```

Remplacez "ftpuser" par le nom d'utilisateur souhaité. Dans notre cas developpeur

Créez un répertoire pour les fichiers FTP et définissez les permissions :

```
mkdir /home/developpeur/ftp
chown nobody:nogroup /home/developpeur/ftp
chmod a-w /home/developpeur/ftp
```

Créez un sous-répertoire pour le téléchargement des fichiers:

```
mkdir /home/developpeur/ftp/files
chown developpeur:developpeur /home/developpeur/ftp/files
```

Redémarrez le service ProFTPD et vérifiez que le service fonctionne correctement :

```
systemctl restart proftpd
systemctl status proftpd
```

Testez la connexion depuis un autre ordinateur en utilisant un client FTP ou la commande ftp :

ftp [votre_adresse_ip](#) nous [172.16.0.200](#)

Entrez le nom d'utilisateur et le mot de passe que vous avez créés.

MariaDB

Sécurisation de l'installation

[mysql_secure_installation](#)

Sécurisation de l'installation

[mysql_secure_installation](#)

Suivez les instructions à l'écran. Vous devrez :

- Définir un mot de passe root pour MariaDB
- Supprimer les utilisateurs anonymes
- Désactiver la connexion root à distance
- Supprimer la base de test
- Recharger les privilèges

Connectez-vous à MariaDB

[mysql](#) ou [use mysql](#)

Création d'un utilisateur et d'une base de données

```
CREATE DATABASE mabase;  
CREATE USER 'monutilisateur'@'localhost' IDENTIFIED BY 'monmotdepasse';  
GRANT ALL ON mabase.* TO 'developpeur'@'localhost' IDENTIFIED BY 'developpeur06'  
WITH GRANT OPTION;  
GRANT ALL PRIVILEGES ON mabase.* TO 'developpeur'@'localhost';  
FLUSH PRIVILEGES;  
EXIT;
```


Configuration de MariaDB : Editez le fichier de configuration

```
nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Ajustez les paramètres suivants selon vos besoins

```
[mysqld]  
bind-address = 127.0.0.1 # Changez en 0.0.0.0 pour permettre les connexions distantes  
max_connections = 100  
key_buffer_size = 16M  
max_allowed_packet = 16M
```

Redémarrez MariaDB

```
systemctl restart mariadb
```

Vérifiez que MariaDB est en cours d'exécution

```
systemctl status mariadb
```

Vérification de l'existence de l'utilisateur : Connectez-vous à MySQL en tant qu'utilisateur root :

```
mysql
```

Une fois dans l'invite MySQL, exécutez la commande suivante :

```
SELECT User, Host FROM mysql.user WHERE User = 'developpeur';
```

Si l'utilisateur n'existe pas, créez-le :

```
CREATE USER 'developpeur'@'localhost' IDENTIFIED BY 'developpeurbd06';
```

Accordez les privilèges nécessaires à l'utilisateur :

```
GRANT ALL PRIVILEGES ON *.* TO 'developpeurbd'@'localhost' WITH GRANT OPTION;  
FLUSH PRIVILEGES;
```

Vérifiez que les privilèges ont été correctement accordés :

```
SHOW GRANTS FOR 'developpeurbd'@'localhost';
```

Sortez de l'invite MySQL :

```
EXIT;
```

Testez la connexion avec le nouvel utilisateur :

```
mysql -u developpeurbd -p
```

Entrez le mot de passe 'developpeurbd06' quand demandé.

Pour changer le mot de passe de l'utilisateur

Une fois dans l'invite MySQL, utilisez la commande suivante pour changer le mot de passe :

```
ALTER USER 'developpeur'@'localhost' IDENTIFIED BY 'nouveau_mot_de_passe';
```

Remplacez 'nouveau_mot_de_passe' par le mot de passe que vous souhaitez utiliser.

Actualisez les privilèges :

```
FLUSH PRIVILEGES;
```

Vérifiez que le changement a bien été pris en compte :

```
SELECT User, Host, authentication_string FROM mysql.user WHERE User = 'developpeur';
```

Vous devriez voir une entrée pour 'developpeur' avec une chaîne d'authentification mise à jour.

Quittez l'invite MySQL/

`EXIT;`

Testez la connexion avec le nouvel utilisateur et le nouveau mot de passe :

`mysql -u developpeurbd -p`

Entrez le nouveau mot de passe quand demandé.

Si vous utilisez **phpMyAdmin**, vous devrez peut-être mettre à jour sa configuration si le mot de passe est stocké dans un fichier de configuration :(procédure pas conseillé)

`nano /etc/phpmyadmin/config.inc.php`

Cherchez une ligne qui ressemble à :

`$cfg['Servers'][$i]['password'] = 'ancien_mot_de_passe';`

Et mettez-la à jour avec le nouveau mot de passe.

Redémarrez les services web après toute modification :

`systemctl restart nginx`

`systemctl restart php7.4-fpm # ou la version de PHP que vous utilisez`

PhpMyAdmin

Configurer Nginx pour phpMyAdmin :

Créez un nouveau fichier de configuration Nginx

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

Ajoutez le contenu suivant

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

```
server {  
    listen 443 ssl http2;  
    server_name php.starfleet.lan;  
  
    ssl_certificate /etc/ssl/certs/nginx.crt;  
    ssl_certificate_key /etc/ssl/private/nginx.key;  
  
    root /usr/share/phpmyadmin;  
    index index.php index.html index.htm;  
  
    location / {
```

```

        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
# Assurez-vous que la version de PHP est correcte
        fastcgi_index index.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }

    location ~ /\.ht {
        deny all;
    }
}
domcd

```

Activer la configuration

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

Vérifier la configuration Nginx

```
nginx -t
```

Si la vérification est réussie, redémarrez Nginx

```
systemctl restart nginx
```

Modifier le fichier *config.inc.php*

Dans notre cas de figure le chemin est `opt/phpMyadmin/config.inc.php` ajouter le token a la ligne(attention aux 32 caracteres)

```
$cfg['blowfish_secret'] = 'dirnforltnDIRlpdrofnsibdorlfonsi'; /* YOU MUST FILL >
```

Assurez-vous que PHP-FPM est installé et en cours d'exécution

```
apt install php-fpm
```

```
systemctl status php7.4-fpm # Remplacez 7.4 par votre version de PHP si différente
```

Accédez à phpMyAdmin via <https://php.starfleet.lan>

Puis configurez la résolution DNS pour que `php.starfleet.lan` pointe vers l'adresse IP de votre serveur.

Cockpit

Installez le paquet

```
dpkg -i code-server_4.16.1_amd64.deb
```

Démarrez le service code-server

```
systemctl enable --now code-server@$USER
```

Configurez code-server : Créez ou modifiez le fichier de configuration

```
nano ~/.config/code-server/config.yaml
```

Ajoutez ou modifiez ces lignes

```
bind-addr: 0.0.0.0:8080
auth: password
password: votre_mot_de_passe_secret
cert: false
```

Redémarrez le service

```
systemctl restart code-server@$USER
```

Configurez Nginx comme reverse proxy : Créez un nouveau fichier de configuration Nginx

```
nano /etc/nginx/sites-available/vscode
```

Ajoutez cette configuration

```
server {
    listen 80;
    server_name vscore.starfleet.lan;

    location / {
        proxy_pass http://localhost:8080;
        proxy_set_header Host $host;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection upgrade;
        proxy_set_header Accept-Encoding gzip;
    }
}
```

Activez le site

```
ln -s /etc/nginx/sites-available/vscode /etc/nginx/sites-enabled/
```

Vérifiez la configuration Nginx

`nginx -t`

Redémarrez Nginx

`systemctl restart nginx`

Configurez le pare-feu si nécessaire

`ufw allow 80/tcp`

Configurez la résolution DNS pour que `vscore.starfleet.lan` pointe vers l'adresse IP de votre serveur.

Accédez à VS Code Server via votre navigateur à l'adresse <http://vscore.starfleet.lan>

VsCore

Installez le paquet :

`dpkg -i code-server_4.16.1_amd64.deb`

Démarrez le service code-server :

`systemctl enable --now code-server@$USER`

Configurez code-server : Créez ou modifiez le fichier de configuration :

`nano ~/.config/code-server/config.yaml`

Ajoutez ou modifiez ces lignes :

`bind-addr: 0.0.0.0:8080`

`auth: password`

`password: votre_mot_de_passe_secret`

`cert: false`

Redémarrez le service :

`systemctl restart code-server@$USER`

Nous allons créer une nouvelle configuration pour VS Code Server tout en préservant celle d'admin. Créez un nouveau fichier :

`nano /etc/nginx/sites-available/vscore.starfleet.lan.conf`

Ajoutez le contenu suivant pour VS Code Server :

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

```
server {  
    listen 443 ssl http2;
```

```

server_name vscore.starfleet.lan;

ssl_certificate /etc/ssl/certs/nginx.crt;
ssl_certificate_key /etc/ssl/private/nginx.key;

# Ajoutez ici vos paramètres SSL si nécessaire

location / {
    proxy_pass http://localhost:8080;
    proxy_set_header Host $host;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection upgrade;
    proxy_set_header Accept-Encoding gzip;
}

```

}

Activez la nouvelle configuration pour VS Code Server :

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

Vérifiez la configuration Nginx

```
nginx -t
```

Si tout est correct, redémarrez Nginx :

```
systemctl restart nginx
```

Assurez-vous que VS Code Server est en cours d'exécution :

```
systemctl status code-server@$USER
```

Si ce n'est pas le cas, démarrez-le :

```
systemctl start code-server@$USER
```

N'oubliez pas de :

1. Mettre à jour votre configuration DNS ou les fichiers hosts des clients pour que vscore.starfleet.lan pointe vers l'adresse IP de votre serveur.
2. Vérifier que le pare-feu autorise le trafic sur les ports 80 et 443.

Configuration d'une Machine Client pour le Réseau starfleet.lan

La machine client doit avoir

1. 1Configuration Réseau

Configurez l'interface réseau pour utiliser une adresse IP statique dans la plage 172.16.0.0/24 (par exemple, 172.16.0.10).

Éditez le fichier `/etc/network/interfaces` :

```
auto eth0
iface eth0 inet dhcp
```

2. Configuration DNS(optionnelle)

Dans notre cas de figure nous n'avons pas besoin de limiter la connection de notre client seulement à notre serveur, le but est de pouvoir utiliser cette machine aussi sur d'autres reseau

Éditez le fichier `/etc/resolv.conf` :

```
domain starfleet.lan
search starfleet.lan
nameserver 172.16.0.200
```

Comparons les deux scénarios pour comprendre l'impact de ces directives sur la résolution des noms de domaine.

Scénario 1: Avec la configuration

domain starfleet.lan

search starfleet.lan

Noms de domaine potentiels pour atteindre le serveur :

1. Nom complet (FQDN) :
 - www.starfleet.lan
 - www8.starfleet.lan
 - www7.starfleet.lan
 - php.starfleet.lan
 - admin.starfleet.lan
2. Noms courts (grâce aux directives domain et search) :
 - www
 - www8
 - www7
 - php
 - admin
3. Adresse IP directe :
 - 172.16.0.200

Dans ce scénario, l'utilisateur peut utiliser soit le nom complet, soit le nom court, et le système ajoutera automatiquement ".starfleet.lan" si nécessaire.

Scénario 2: Sans les directives domain et search

Noms de domaine potentiels pour atteindre le serveur :

1. Nom complet (FQDN) uniquement :
 - `www.starfleet.lan`
 - `www8.starfleet.lan`
 - `www7.starfleet.lan`
 - `php.starfleet.lan`
 - `admin.starfleet.lan`
2. Adresse IP directe :
 - `172.16.0.200`

Dans ce scénario, l'utilisateur doit toujours utiliser le nom de domaine complet. Les noms courts ne fonctionnent pas automatiquement.

Différences clés :

3. Convivialité :
 - a. Avec les directives : Les utilisateurs peuvent utiliser des noms courts, ce qui est plus pratique.
 - b. Sans les directives : Les utilisateurs doivent toujours taper le nom de domaine complet.
4. Résolution de noms :
 - a. Avec les directives : Le système tente automatiquement d'ajouter le suffixe `".starfleet.lan"`.
 - b. Sans les directives : Aucune tentative automatique d'ajout de suffixe.
5. Ambiguïté potentielle :
 - a. Avec les directives : Il pourrait y avoir une ambiguïté si des noms courts identiques existent dans différents domaines.
 - b. Sans les directives : Pas d'ambiguïté, mais moins de flexibilité.
6. Performance :
 - a. Avec les directives : Légèrement plus lent si le nom court n'existe pas, car le système essaiera d'ajouter le suffixe.
 - b. Sans les directives : Potentiellement plus rapide pour les noms non existants, car pas de tentative supplémentaire.

En conclusion, la présence des directives `domain` et `search` offre plus de flexibilité et de convivialité aux utilisateurs, mais nécessite une gestion plus attentive pour éviter les conflits de noms. L'absence de ces directives rend le système plus prévisible mais moins convivial pour les utilisateurs.

7. Configurer Network Manager gestionnaire de réseau

Pour rendre ces changements permanents du dossier `/etc/resolv.conf`, utilisez le gestionnaire de réseau de votre distribution

```
sudo nano /etc/NetworkManager/NetworkManager.conf
```


Ajouter `dns=none` à `[main]` comme suit

```
[main]
plugins=ifupdown,keyfile
dns=none
```

```
[ifupdown]
managed=false
```

8. Configuration DNS optionnelle gestionnaire de réseau

Pour rendre ces changements permanents, utilisez le gestionnaire de réseau de votre distribution ou configurez `/etc/resolvconf/resolv.conf.d/base`:

Ajoutez vos configurations, par exemple :

```
nameserver 172.16.0.200
domain starfleet.lan
search starfleet.lan
```

Appliquer les changements : Après avoir modifié les fichiers, vous devez mettre à jour `resolvconf` :

```
resolvconf -u
```

Rappel :

1. L'utilisation de `resolvconf` est particulièrement utile dans les environnements où la configuration réseau peut changer fréquemment (par exemple, sur des ordinateurs portables qui se connectent à différents réseaux).
2. Dans un environnement de serveur stable comme votre configuration `starfleet.lan`, la configuration directe de `/etc/resolv.conf` ou l'utilisation du DHCP pour fournir les informations DNS peuvent être suffisantes.

10. Configuration DNS optionnelle dans `/etc/dhcp/dhclient.conf`

Voici une explication plus détaillée :

- Normalement, lorsqu'un client utilise DHCP, il reçoit automatiquement les paramètres DNS du serveur DHCP. Cela inclut l'adresse du serveur DNS et potentiellement le domaine de recherche.

- Cependant, dans certains cas, vous pourriez vouloir que le client utilise toujours des paramètres DNS spécifiques, même si le serveur DHCP fournit des informations différentes.
- La configuration dans `/etc/dhcp/dhclient.conf` permet de "remplacer" (supersede) les informations fournies par le serveur DHCP avec vos propres paramètres.

```
supersede domain-name "starfleet.lan";  
supersede domain-search "starfleet.lan";  
supersede domain-name-servers 172.16.0.200;
```

Ces lignes disent au client DHCP :

- Toujours utiliser "starfleet.lan" comme nom de domaine, même si le DHCP suggère autre chose.
- Toujours utiliser "starfleet.lan" pour la recherche de domaine.
- Toujours utiliser 172.16.0.200 comme serveur DNS, même si le DHCP fournit une adresse différente.

Cette configuration est utile si vous voulez vous assurer que vos clients utilisent toujours votre serveur DNS interne, même s'ils se connectent à d'autres réseaux ou si le serveur DHCP est mal configuré.

C'est une mesure de sécurité et de contrôle supplémentaire, mais elle n'est pas toujours nécessaire si votre serveur DHCP est correctement configuré pour votre réseau starfleet.lan.

9. Vérification

Redémarrez le service réseau :

```
systemctl restart networking
```

Testez la connectivité :

```
ping 172.16.0.200
```

Vérifiez la résolution DNS :

`nslookup www8.starfleet.lan`

10. Sécurité

Configurez le pare-feu pour autoriser uniquement le trafic nécessaire.
Assurez-vous que les mises à jour de sécurité sont activées et appliquées régulièrement.

11. Dépannage

- Si la résolution DNS échoue, vérifiez la connectivité au serveur DNS (172.16.0.200).
- Utilisez `dig` ou `nslookup` pour tester spécifiquement la résolution DNS.
- Vérifiez les logs système pour tout message d'erreur lié au réseau ou au DNS.