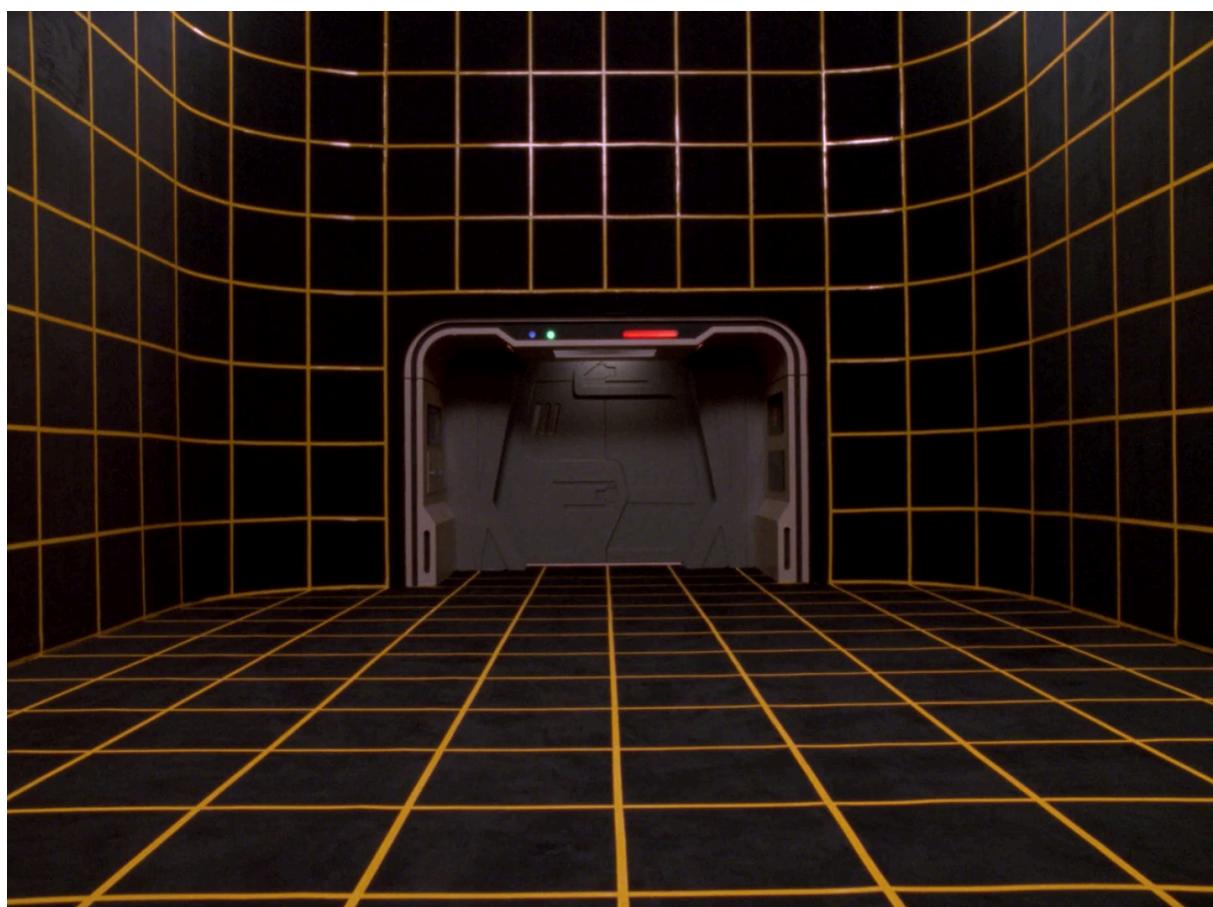


# Holodeck(VMs)



projet réalisé par Youcef

## Résumé-projet Holodeck(VMs):

### Notice d'installation et d'utilisation :

#### 1. Création et Configuration de la VM Serveur.

##### 1.1 Création de la VM sous Debian sans interface graphique.

1.1.1 Configuration matérielle : 2 Go RAM, 2 vCPU, 32 Go de disque.

1.1.2 Configuration des cartes réseau : WAN et LAN.

##### 1.2 Installation du système d'exploitation Debian

##### 1.3 Installation des services de base

1.3.1 Installation et configuration du serveur DHCP

1.3.2 Installation et configuration du serveur DNS

1.3.2 Installation de Nginx (dernière version)

1.3.3 Installation de PHP (versions 7.x et 8.x)

1.3.4 Installation de MariaDB (dernière version)

1.3.5 Installation et configuration du serveur FTP (en SSL/TLS)

1.3.6 Installation et configuration de CockPit

1.3.7 Installation et configuration de UFW et autorisations des services

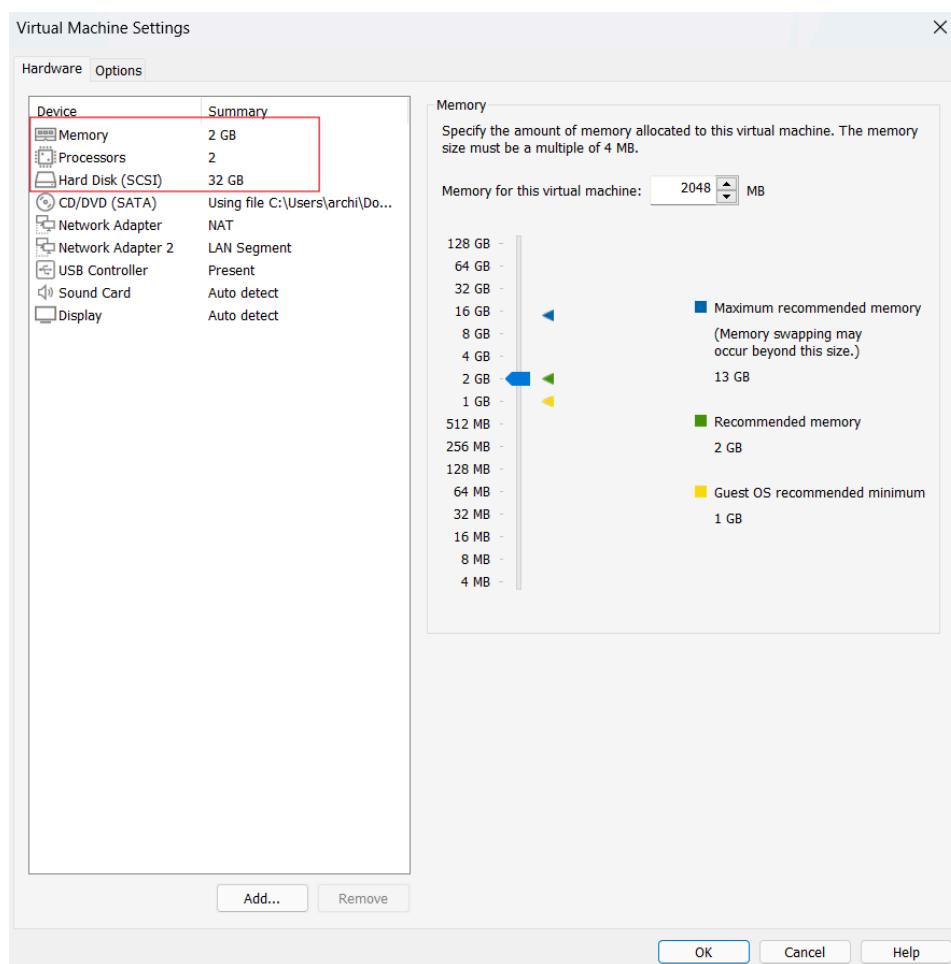
# PROJET

## 1. Création et configuration de la VM serveur.

### 1.1 Création de la VM sous Debian sans interface graphique.

#### 1.1.1 Configuration matérielle : 2 Go RAM, 2 vCPU, 32 Go de disque.

On commence par faire la configuration matérielle(Hardware) de notre VM Serveur ; en lui attribuant 2 Go de RAM, 2 vCPU et un disque de 32 Go."



#### 1.1.2 Configuration des cartes réseau : WAN et LAN.

On utilise 2 cartes réseaux, une en NAT (WAN) ce qui nous donne accès à Internet et une autre en LAN soit Host Only ou Lan Segment . Pour mon cas, j'ai préféré utiliser Lan Segment.

# La Plateforme

## La différence entre Host-Only et Lan Segment ?

### “Host-Only”

**Connexion restreinte** : En mode “Host-Only”, la machine virtuelle est connectée à un réseau virtuel privé qui inclut uniquement l’hôte (la machine physique qui exécute les machines virtuelles) et les autres machines virtuelles configurées en “Host-Only”. Il n'y a aucune connexion directe avec le réseau externe (Internet ou réseau local physique).

**Accès à l’hôte** : La machine virtuelle peut communiquer directement avec l’hôte (et vice versa), mais elle ne peut pas accéder à Internet ou au réseau local physique.

**Usage typique** : Ce mode est souvent utilisé pour les tests ou le développement, où vous souhaitez isoler les machines virtuelles du reste du réseau mais leur permettre d’interagir avec l’hôte.

### “Lan Segment”

**Isolation complète** : En mode "LAN Segment", les machines virtuelles sont connectées à un réseau privé séparé, complètement isolé, qui ne comprend pas l’hôte ni le réseau physique. Les machines virtuelles sur le même LAN Segment peuvent se voir entre elles, mais pas avec l’hôte ou le réseau externe.

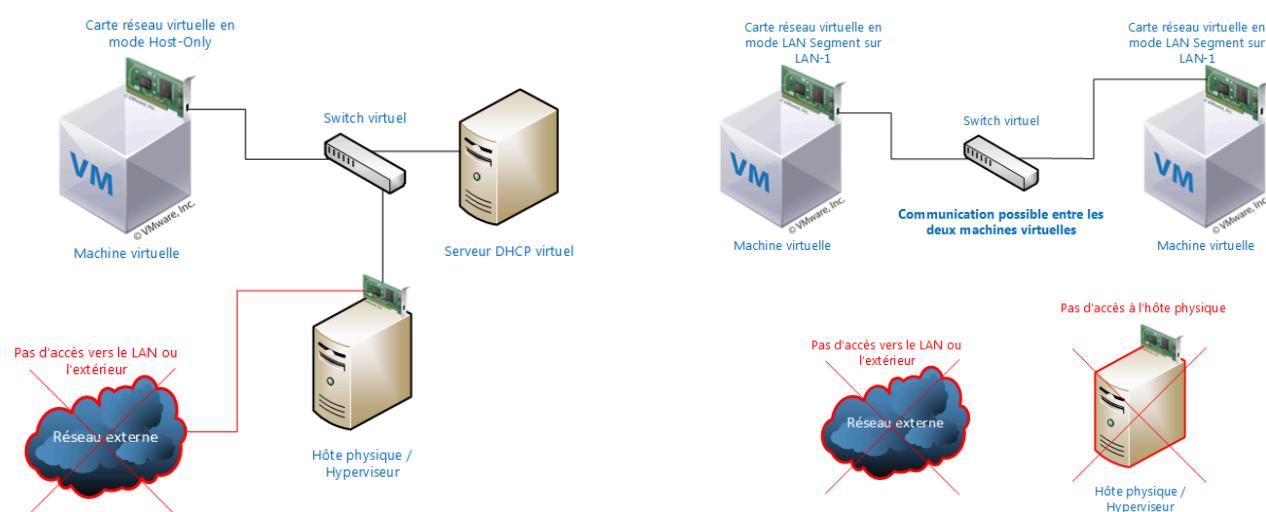
**Pas d'accès à l'hôte ou à Internet** : Il n'y a aucune communication possible avec l'hôte ou le réseau physique, ni d'accès à Internet, sauf si vous ajoutez manuellement une autre interface réseau configurée différemment.

**Usage typique** : Ce mode est utilisé pour simuler un réseau isolé, par exemple pour tester des scénarios de réseau spécifique sans aucune interférence de l’extérieur.

### Pour Résumer:

**Host-Only** : Permet la communication entre les machines virtuelles et l’hôte, mais pas avec le réseau physique ou Internet.

**LAN Segment** : Permet uniquement la communication entre les machines virtuelles dans le même segment, sans aucune interaction avec l’hôte ou le réseau physique.



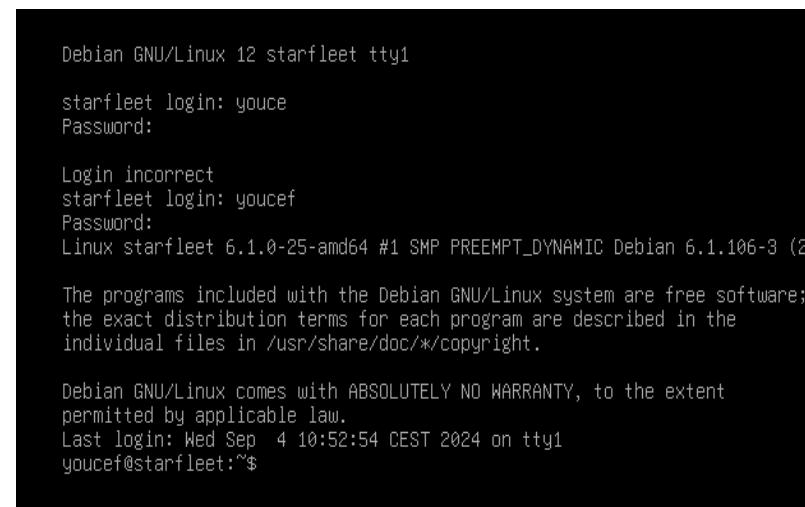
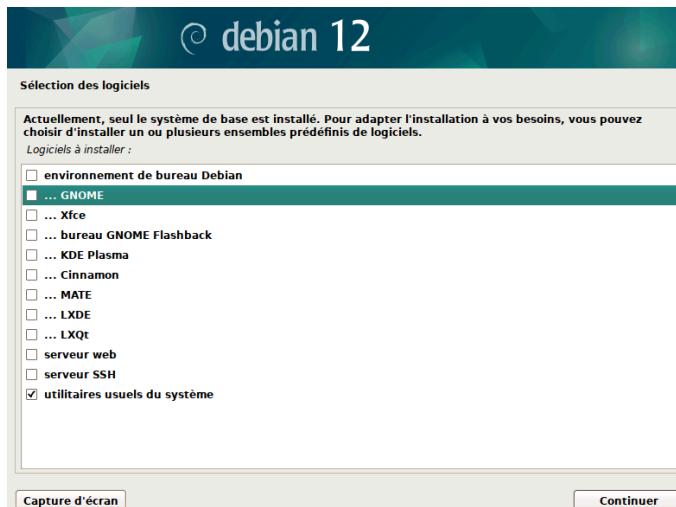
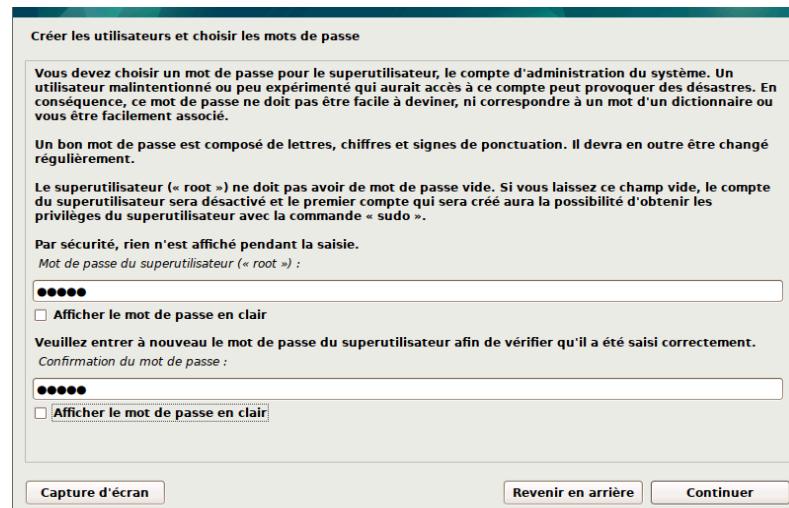
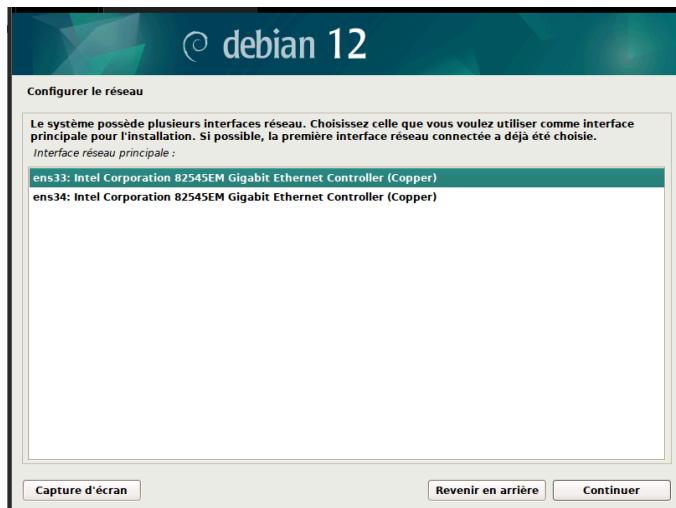
“Host-Only”

“Lan Segment”



## 1.2 Installation du système d'exploitation Debian:

Après avoir configuré le matériel de la machine virtuelle, nous passons à l'installation du système d'exploitation Debian. Cette étape comprend le démarrage de l'installation, la sélection des options de partitionnement, le choix de l'interface réseau connecté à internet, utile pour les mises à jour durant l'installation et la configuration des paramètres de base, comme le fuseau horaire et les informations de compte utilisateur."



### 1.3.1 Installation et configuration du serveur DHCP

Après l'installation du système d'exploitation, nous procédons à l'installation et à la configuration du serveur DHCP/DNS. Cette étape inclut l'installation des paquets nécessaires, la configuration du serveur DHCP.

Pour Installer le serveur DHCP on utilise la commande :

```
root@starfleet:~# apt install isc-dhcp-server
```

Sur Debian, il y a une petite spécificité, il faut indiquer dans /etc/default/isc-dhcp-server sur quelles interfaces va écouter le service DHCP:

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

On fait un backup du fichier:

```
root@starfleet:/etc/default# ls
amd64-microcode  console-setup  cron  dbus  grub  grub.d  hwclock  isc-dhcp-server  isc-dhcp-server.backup
```

Dans le fichier “isc-dhcp-server”, mettre les interfaces d’écoutes qui correspondent à la seconde carte réseau:

```
GNU nano 7.2                                         /etc/default/isc-dhcp-server *
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)
#
# Path to dhcpcd's config file (default: /etc/dhcp/dhcpcd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpcd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpcd6.conf

# Path to dhcpcd's PID file (default: /var/run/dhcpcd.pid).
#DHCPDv4_PID=/var/run/dhcpcd.pid
#DHCPDv6_PID=/var/run/dhcpcd6.pid

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

# On what interfaces should the DHCP server (dhcpcd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="ens34"
INTERFACESv6=""
```

Configuration du fichier “interfaces” situé dans dans le répertoire “/etc/network”:

```
GNU nano 7.2                                         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

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

auto ens34
iface ens34 inet static
    address 192.168.187.10
    netmask 255.255.255.0
```

# La Plateforme

Configuration du fichier “dhcpd.conf” situé dans le répertoire “/etc/dhcp”:

```
GNU nano 7.2                                         dhcpd.conf
#
# Sample configuration file for ISC dhcpcd
#
# option definitions common to all supported networks...
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;
#
#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;
#
#Domaine
#option domain-name "starfleet.lan";
#
#Déclaration d'un réseau
subnet 192.168.187.10 netmask 255.255.255.0 {
    range 192.168.187.50 192.168.187.60;
    option subnet-mask 255.255.255.0;
    #option domain-name-servers 192.168.1.10; #DNS
    option routers 192.168.87.2;
    option broadcast-address 192.168.255.255;
    default-lease-time 600;
    max-lease-time 7200;
}
#
# 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;
#
# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.
#
#subnet 10.152.187.0 netmask 255.255.255.0 {
#}
```

Redémarrer networking et isc-dhcp-server:

```
$ sudo systemctl restart networking
sudo systemctl restart isc-dhcp-server_
```

L'adresse IP de mon client correspond à bien à une adresse IP de mon serveur DHCP:

```
youcef@Client:~$ 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
        valid_lft forever preferred_lft forever
2: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:07:f2:a5 brd ff:ff:ff:ff:ff:ff
    altname enp2s5
    inet 192.168.187.50/24 brd 192.168.187.255 scope global dynamic noprefixroute
        valid_lft 592sec preferred_lft 592sec
    inet6 fe80::20c:29ff:fe07:f2a5/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
youcef@Client:~$
```

# La Plateforme

## 1.3.2 Installation et configuration du serveur DNS

On commence par installer le paquet Bind9 qui est un serveur DNS populaire.

```
root@starfleet:/etc# apt install bind9
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
bind9 est déjà la version la plus récente (1:9.18.28-1~deb12u2).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@starfleet:/etc#
```

**Bind9 :** BIND9 (Berkeley Internet Name Domain, version 9) est un logiciel open-source qui implémente le protocole DNS (Domain Name System). Il permet la résolution des noms de domaine en adresses IP, une fonction essentielle pour la communication sur les réseaux, qu'il s'agisse d'Internet ou de réseaux privés.

- Après l'installation de Bind9, il est essentiel de vérifier et, si nécessaire, de modifier le nom d'hôte de votre machine. Ce nom est utilisé par le système pour identifier votre serveur sur le réseau, et il est important que ce nom corresponde à celui configuré dans vos fichiers de zone DNS.

Le fichier `/etc/hostname` contient le nom d'hôte de la machine.

```
GNU nano 7.2                                         hostname
starfleet.lan
```

- Puis on se rend dans le fichier hosts pour ajouter l'adresse ipv4 de notre serveur ainsi que son FQDN qui signifie textuellement : « Fully Qualified Domain Name », que l'on pourrait traduire par « Nom d'hôte pleinement nommé »

```
GNU nano 7.2                                         hosts
127.0.0.1      localhost
127.0.1.1      starfleet.lan
192.168.187.10 starfleet.lan starfleet

# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
```

# La Plateforme

- On retourne configurer notre dhcp.conf pour qu'il distribue DNS aux machines clientes.

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

#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;

#Domaine
#option domain-name "starfleet.lan";

#Déclaration d'un réseau
subnet 192.168.187.0 netmask 255.255.255.0 {
    range 192.168.187.50 192.168.187.60;
    option subnet-mask 255.255.255.0;
    #option domain-name-servers 192.168.1.10; #DNS
    option broadcast-address 192.168.255.255;
    option domain-name-servers 192.168.187.10;
    option domain-name "starfleet.lan";
    default-lease-time 600;
    max-lease-time 7200;
}

# 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;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
```

**option domain-servers 192.168.187.10:** Cela spécifie l'adresse IP du serveur DNS (Domain Name Server) que les clients recevront via DHCP. Dans ce cas, 192.168.187.10 est l'adresse IP du serveur DNS (probablement Bind9) qui sera utilisé pour résoudre les noms de domaine pour les machines sur le réseau.

**option domain-name "starfleet.lan":** Cette option définit le nom de domaine par défaut pour le réseau. Tous les clients qui recevront cette configuration via DHCP seront considérés comme faisant partie du domaine "starfleet.lan". Cela permet aux machines du réseau de résoudre les noms d'hôtes sans avoir à spécifier le domaine complet (par exemple, en tapant juste machine au lieu de machine.starfleet.lan).



# La Plateforme

Ensuite on ouvre le fichier de configuration pour définir une nouvelle zone :

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

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

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

zone "187.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192.168.187";
};
```

**Zone "starfleet.lan"** : Définit le nom de la zone DNS.

**Type master** : Indique que ce serveur est l'autorité principale pour cette zone.

**File "/etc/bind/db.starfleet.lan"** : Spécifie le chemin vers le fichier qui contient les enregistrements DNS pour cette zone.

**Zone "187.168.192.in-addr.arpa"** : C'est la zone inverse qui permet de faire correspondre des adresses IP à des noms de domaine.



# La Plateforme

## Création du fichier de zone directe ([db.starfleet.lan](#))

Ensuite, nous devons créer le fichier de zone directe qui contient les enregistrements DNS pour la zone **starfleet.lan**. Nous partons du fichier modèle [db.local](#) pour créer [db.starfleet.lan](#).

```
GNU nano 7.2                                     /etc/bind/db.starfleet.lan
;
; Fichier de zone pour starfleet.lan
;
$TTL    604800
@      IN      SOA    ns1.starfleet.lan. admin.starfleet.lan. (
                      2           ;Numéro de série
                      604800    ;Rafrachir
                      86400     ; Réessayer
                     2419200   ; Expirer
                     604800 ) ; TTL cache négatif
;
@      IN      NS     ns1.starfleet.lan.
ns1    IN      A      192.168.187.10 ; IP du serveur DNS
;
; Sous-domaines
www8   IN      A      192.168.187.10 ; Site web en PHP8
www7   IN      A      192.168.187.10 ; Site web en PHP7
php    IN      A      192.168.187.10 ; phpmyAdmin
admin  IN      A      192.168.187.10 ; interface admin
-
```

**SOA (Start of Authority)** : Indique que le serveur DNS est l'autorité principale pour cette zone.

**NS (Name Server)** : Spécifie le serveur DNS pour la zone.

**A (Address Record)** : Associe le nom ns.starfleet.lan et starfleet.lan à l'adresse IP 192.168.187.10.



# La Plateforme

## . Création du fichier de zone inverse (**db.192.168.187**)

La zone inverse permet de faire la correspondance entre une adresse IP et un nom de domaine. Nous devons également créer ce fichier, en partant du modèle db.127 pour créer **db.192.168.187**.

```
GNU nano 7.2                                         /etc/bind/db.192.168.187
;
; Fichier de zone invers pour 192.168.187.10
;
$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.
10    IN      PTR     ns1.starfleet.lan.
10    IN      PTR     www8.starfleet.lan.
10    IN      PTR     www7.starfleet.lan.
10    IN      PTR     php.star.fleet.lan.
10    IN      PTR     admin.starfleet.lan.
```

**PTR (Pointer Record)** : Associe l'adresse IP 192.168.187.10 au nom d'hôte ns.starfleet.lan.

Commande pour vérifier la configuration générale de Bind9 :

On utilise **named-checkzone** pour vérifier l'ensemble des fichiers de configuration de Bind9, y compris les références aux fichiers de zone.

```
root@starfleet:/etc/bind# named-checkzone 187.168.192.Bonjour /etc/bind/db.192.168.187
zone 187.168.192.Bonjour/IN: loaded serial 1
OK
root@starfleet:/etc/bind# named-checkzone starfleet.lan /etc/bind/db.starfleet.lan
zone starfleet.lan/IN: loaded serial 2
OK
```

# La Plateforme

Vérification du côté client avec la commande nslookup :

```
youcef@Client:/etc$ nslookup www8.starfleet.lan
Server:      192.168.187.10
Address:     192.168.187.10#53

Name:   www8.starfleet.lan
Address: 192.168.187.10

youcef@Client:/etc$ nslookup 192.168.187.10
10.187.168.192.in-addr.arpa      name = admin.starfleet.lan.
10.187.168.192.in-addr.arpa      name = www8.starfleet.lan.
10.187.168.192.in-addr.arpa      name = php.star.fleet.lan.
10.187.168.192.in-addr.arpa      name = www7.starfleet.lan.
10.187.168.192.in-addr.arpa      name = ns1.starfleet.lan.

youcef@Client:/etc$ █
```



### 1.3.2 Installation de Nginx (dernière version)

Pour commencer, nous allons installer le paquet Nginx. Avant cela, mettons à jour le cache des paquets sur notre machine en exécutant la commande suivante :

```
root@starfleet:/etc/bind# apt update -y
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Récception de :2 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Récception de :3 http://security.debian.org/debian-security bookworm-security InRelease [47,9 kB]
Récception de :4 http://security.debian.org/debian-security bookworm-security/main Sources [109 kB]
Récception de :5 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [179 kB]
392 ko réceptionnés en 2s (189 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
2 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
```

Ensuite, nous procédons à l'installation de Nginx. Cette étape est simple, car le paquet est disponible dans les dépôts officiels :

```
root@starfleet:/etc/bind# apt install nginx -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  nginx-common
Paquets suggérés :
  fcgiwrap nginx-doc ssl-cert
Les NOUVEAUX paquets suivants seront installés :
  nginx nginx-common
0 mis à jour, 2 nouvellement installés, 0 à enlever et 2 non mis à jour.
Il est nécessaire de prendre 640 ko dans les archives.
Après cette opération, 1 696 ko d'espace disque supplémentaires seront utilisés.
Récception de :1 http://deb.debian.org/debian bookworm/main amd64 nginx-common all 1.22.1-9 [112 kB]
Récception de :2 http://deb.debian.org/debian bookworm/main amd64 nginx amd64 1.22.1-9 [527 kB]
640 ko réceptionnés en 1s (674 ko/s)
Préconfiguration des paquets...
Sélection du paquet nginx-common précédemment désélectionné.
(Lecture de la base de données... 36629 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../nginx-common_1.22.1-9_all.deb ...
Dépaquetage de nginx-common (1.22.1-9) ...
Sélection du paquet nginx précédemment désélectionné.
Préparation du dépaquetage de .../nginx_1.22.1-9_amd64.deb ...
Dépaquetage de nginx (1.22.1-9) ...
Paramétrage de nginx-common (1.22.1-9) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /lib/systemd/system/nginx.service.
Paramétrage de nginx (1.22.1-9) ...
Upgrading binary: nginx.
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
```

Une fois l'installation terminée, vous pouvez vérifier la version de Nginx installée en exécutant cette commande (similaire à celles utilisées pour Apache ou d'autres logiciels) :

```
root@starfleet:/etc/bind# nginx -v
nginx version: nginx/1.22.1
```

# La Plateforme

- Après l'installation, le serveur Nginx est automatiquement démarré. Vous pouvez vérifier son statut pour confirmer qu'il fonctionne correctement avec la commande suivante :

```
root@starfleet:/etc/bind# 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 Mon 2024-09-09 11:31:24 CEST; 2min 44s ago
    Docs: man:nginx(8)
 Process: 2263 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Process: 2264 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 2288 (nginx)
   Tasks: 3 (limit: 2273)
    Memory: 2.4M
      CPU: 130ms
     CGroup: /system.slice/nginx.service
             ├─2288 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─2291 "nginx: worker process"
             └─2292 "nginx: worker process"

sept. 09 11:31:24 starfleet.lan systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
sept. 09 11:31:24 starfleet.lan systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
```

- Pour que notre serveur Web Nginx démarre automatiquement lorsque la machine Linux démarre ou redémarre, on doit exécuter la commande suivante :

```
root@starfleet:/etc/bind# 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
```

- On peut vérifier du côté de notre VM client :



## 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](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

- Pour configurer Nginx afin de servir le domaine `www8.starfleet.lan` en utilisant un hôte virtuel (Virtual Host):

- On crée le fichier de configuration pour "`www8.starfleet.lan`" avec la commande :

```
root@starfleet:/etc# nano /etc/nginx/sites-available/www8.starfleet.lan
```

# La Plateforme

```
GNU nano 7.2                                         www8.starfleet.lan
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 traiter les requêtes destinées à ce site.

**server\_name www8.starfleet.lan :** Cette directive indique à Nginx d'utiliser ce bloc de configuration pour toutes les requêtes adressées au domaine www8.starfleet.lan.

**root /var/www/www8 :** Spécifie le répertoire où sont situés les fichiers de ce sous-domaine. Si ce dossier n'existe pas, il faudra le créer.

**location ~ .php\$ :** Cette section permet d'exécuter les fichiers PHP associés à ce sous-domaine. Il est nécessaire de s'assurer que PHP et PHP-FPM sont correctement configurés

- Ensuite on crée le répertoire pour www8.starfleet.lan :

```
root@starfleet:~# mkdir -p /var/www/www8|
```

**-p :** Ce paramètre indique à `mkdir` de créer également tous les répertoires parents qui n'existent pas déjà. Par exemple, si `/var/www` n'existe pas, l'option `-p` s'assure que ces répertoires seront créés avant de créer le répertoire final `/www8`. Sans ce flag, si les répertoires parents n'existent pas, la commande échouerait.

- Créons un fichier de test :

```
root@starfleet:~# echo "Welcome to www8.starfleet.lan" | tee /var/www/www8/index.html
```

- Nous pouvons à présent créer le lien symbolique:

```
root@starfleet:~# ln -s /etc/nginx/sites-available/www8.starfleet.lan /etc/nginx/sites-enabled/
```

# La Plateforme

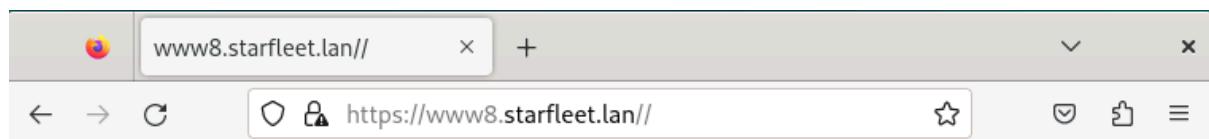
On vérifie que ça fonctionne bien :

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

- On redémarre ensuite le système :

```
root@starfleet:~# systemctl restart nginx
```

- On retourne sur le client pour voir si tout fonctionne :



## Utilisation des certificats auto-signés pour HTTPS :

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

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

On répond aux différentes questions :

```
root@starfleet:/etc/ssl/private# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt
-----+
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:PACA
Locality Name (eg, city) []:MARSEILLE
Organization Name (eg, company) [Internet Widgits Pty Ltd]:LA PLATEFORME
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:starfleet.lan
Email Address []:youcef.ouarhleb@laplateforme.io
```

# La Plateforme

- On fait un fichier de paramètres Diffie-Hellman :

```
root@starfleet:~# openssl dhparam -out /etc/nginx/dhparam.pem 2048
```

**openssl** : C'est l'outil en ligne de commande utilisé pour gérer les certificats, les clés et divers protocoles de sécurité comme SSL/TLS.

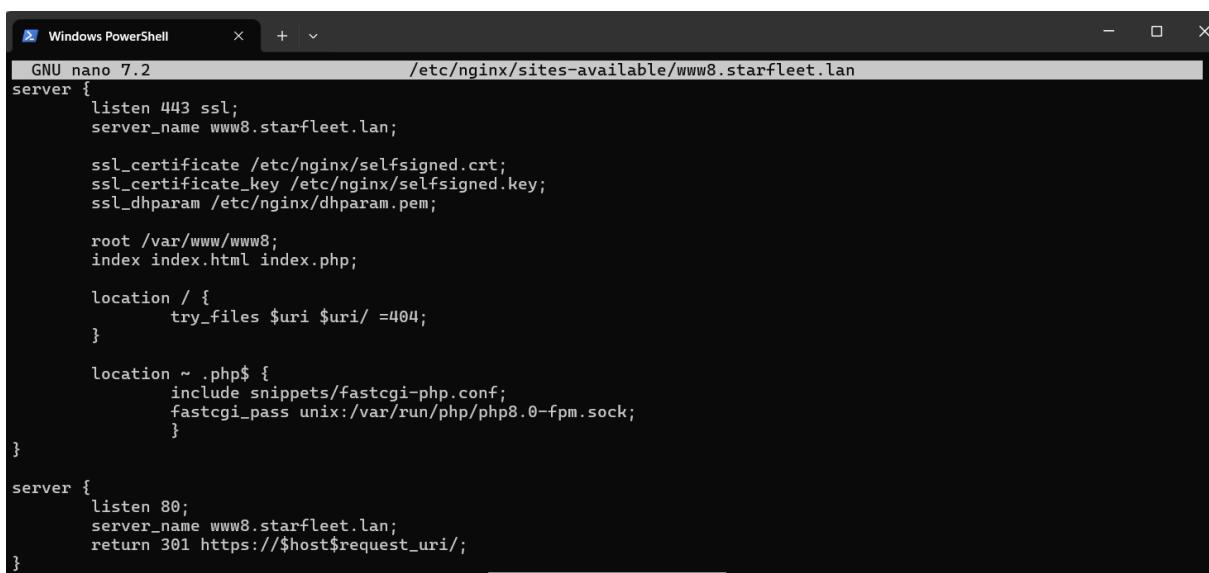
**dhparam** : Indique que l'on génère des paramètres Diffie-Hellman. DH est un algorithme utilisé pour échanger des clés de manière sécurisée entre un client et un serveur lors d'une connexion chiffrée.

**-out /etc/nginx/dhparam.pem** : Cette option spécifie le chemin et le nom du fichier de sortie. Ici, le fichier sera enregistré sous **/etc/nginx/dhparam.pem**, ce fichier contiendra les paramètres DH.

**2048** : C'est la taille (en bits) des paramètres générés. 2048 bits est un choix courant, car il offre un bon équilibre entre sécurité et performance. Plus la taille est grande, plus la sécurité est élevée, mais cela demande plus de ressources de calcul.

- Maintenant on va modifier le fichier **www8.starfleet.lan** avec la commande :

```
root@starfleet:~# nano /etc/nginx/sites-available/www8.starfleet.lan
```



```
Windows PowerShell
GNU nano 7.2
/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 / {
        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/;
```

On vérifie que la configuration nginx fonctionne toujours bien :

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

# La Plateforme

- On redémarre Nginx pour appliquer les changements :

```
root@starfleet:~# systemctl reload nginx
```

On retourne la VM Client pour voir si ça fonctionne :



### 1.3.3 Installation de PHP (versions 7.x et 8.x)

- On commence par ouvrir un fichier nommé `php.list` dans le répertoire `/etc/apt/sources.list.d/`, afin d'ajouter le dépôt Sury qui propose des versions récentes de PHP.
- On utilise la commande :

```
root@starfleet:/# root@starfleet:/# nano /etc/apt/sources.list.d/php.list
```

```
GNU nano 7.2                               /etc/apt/sources.list.d/php.list
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.

- Ensuite ,on ajoute 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 :

```
root@starfleet:~# wget -O /etc/apt/trusted.gpg.d/sury-keyring.gpg https://packages.sury.org/php/apt.gpg
--2024-09-10 15:55:34--  https://packages.sury.org/php/apt.gpg
Résolution de packages.sury.org (packages.sury.org)... 143.244.56.51, 2400:52e0:1e02::1072:1
Connexion à packages.sury.org (packages.sury.org)|143.244.56.51|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 1769 (1,7K) [application/octet-stream]
Sauvegarde en : « /etc/apt/trusted.gpg.d/sury-keyring.gpg »

/etc/apt/trusted.gpg.d/sury-k 100%[=====] 1,73K --.-KB/s   ds 0s
2024-09-10 15:55:35 (42,9 MB/s) - « /etc/apt/trusted.gpg.d/sury-keyring.gpg » sauvagardé [1769/1769]
```

- On remet à jour nos paquets :

```
root@starfleet:~# apt update
Réception de :1 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
Atteint :2 http://deb.debian.org/debian bookworm InRelease
Réception de :3 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Réception de :4 https://packages.sury.org/php bookworm InRelease [7 551 B]
Réception de :5 http://security.debian.org/debian-security bookworm-security/main Sources [109 kB]
Réception de :6 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [179 kB]
Réception de :7 https://packages.sury.org/php bookworm/main amd64 Packages [242 kB]
641 ko réceptionnés en 1s (812 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
2 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
```

# La Plateforme

- Maintenant on installe PhP7 et PhP8 :

- J'ai décidé d'installer **PHP 7.4** car c'est la dernière version stable de la série 7.x qui reçoit des mises à jour de sécurité.

```
root@starfleet:~# apt-get install php7.4 php7.4-cli php7.4-cgi php7.4-fpm php7.4-gd php7.4-mysql php7.4-imap php7.4-curl  
php7.4-intl php7.4-pspell php7.4-sqlite3 php7.4-tidy php7.4-xmlrpc php7.4-xsl php7.4-zip php7.4-mbstring php7.4-soap ph  
p7.4-opcache libonig5 php7.4-common php7.4-json php7.4-readline php7.4-xml  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Les paquets supplémentaires suivants seront installés :  
  debssuryorg-archive-keyring fontconfig-config fonts-dejavu-core libabssl20220623 libaom3 libavif15 libc-client2007e  
  libcurl4 libdav1d6 libde265-0 libdeflate0 libfontconfig1 libfribidi0 libgavl-1 libgd3 libomp1 libgraphite2-3  
  libharfbuzz0b libheif1 libimagequant0 libjbig0 libjpeg62-turbo liblrc4 libnuma1 libraqm0 librav1e0 libsvtav1enc1
```

**php7.4** : Installe le moteur PHP 7.4 de base.

**php7.4-cli** : Installe l'interface en ligne de commande (CLI) de PHP 7.4.

**php7.4-cgi** : Installe PHP pour l'utiliser avec les serveurs CGI.

**php7.4-fpm** : PHP-FPM (FastCGI Process Manager) pour un traitement rapide des requêtes PHP, souvent utilisé avec Nginx.

**php7.4-gd** : Permet de manipuler des images dans PHP.

**php7.4-mysql** : Ajoute la prise en charge des bases de données MySQL ou MariaDB.

**php7.4-imap** : Fournit des fonctions pour interagir avec des serveurs de messagerie via IMAP.

**php7.4-curl** : Installe cURL pour effectuer des requêtes HTTP ou FTP via PHP.

**php7.4-intl** : Ajoute des fonctionnalités pour la gestion des formats internationaux (dates, nombres, etc.).

**php7.4-pspell** : Permet la vérification orthographique dans PHP.

**php7.4-sqlite3** : Ajoute la gestion des bases de données SQLite3.

**php7.4-tidy** : Permet de nettoyer et corriger les fichiers HTML et XML.

**php7.4-xmlrpc** : Active le support XML-RPC, un protocole de communication à distance basé sur XML.

**php7.4-xsl** : Ajoute le support des transformations de documents XML via XSLT.

**php7.4-zip** : Permet la gestion des fichiers ZIP.

**php7.4-mbstring** : Fournit des fonctions pour gérer des chaînes de caractères multioctets (utiles pour les langues non latines).

**php7.4-soap** : Active le support du protocole SOAP, utilisé pour interagir avec des services web SOAP.

**php7.4-opcache** : Active le mécanisme de cache OPcache pour améliorer les performances PHP en stockant des scripts pré-compilés.

**libonig5** : Bibliothèque requise pour la gestion des expressions régulières, utilisée principalement avec l'extension **mbstring**.

**php7.4-common** : Contient des fichiers communs à toutes les installations de PHP 7.4.

**php7.4-json** : Ajoute des fonctionnalités pour gérer le format JSON (JavaScript Object Notation).



# La Plateforme

**php7.4-readline** : Permet l'édition de ligne de commande dans PHP (utilisé pour la CLI).

**php7.4-xml** : Installe des bibliothèques pour le traitement XML dans PHP.

- Et **PHP 8.2**, qui est l'une des dernières versions stables avec des améliorations significatives en termes de performance et de fonctionnalités.

```
root@starfleet:~# apt-get install php8.2 php8.2-cli php8.2-cgi php8.2-fpm php8.2-gd php8.2-imap php8.2-curl
php8.2-intl php8.2-pspell php8.2-sqlite3 php8.2-tidy php8.2-xsl php8.2-zip php8.2-mbstring php8.2-soap php8.2-opcache libonig5 php8.2-common php8.2-readline php8.2-xml
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
libonig5 est déjà la version la plus récente (6.9.8-1).
Paquets suggérés :
  php-pear
Les NOUVEAUX paquets suivants seront installés :
  php8.2-cgi php8.2-cli php8.2-common php8.2-curl php8.2-gd php8.2-imap php8.2-intl php8.2-mbstring
  php8.2-mysql php8.2-opcache php8.2-pspell php8.2-readline php8.2-soap php8.2-sqlite3 php8.2-tidy php8.2-xml
  php8.2-xsl php8.2-zip
0 mis à jour, 20 nouvellement installés, 0 à enlever et 2 non mis à jour.
Il est nécessaire de prendre 7 417 Ko dans les archives.
Après cette opération, 36,5 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 https://packages.sury.org/php bookworm/main amd64 php8.2-common amd64 8.2.23-1+0~20240830.60+debian12~1.gbp2d65da [683 kB]
Réception de :2 https://packages.sury.org/php bookworm/main amd64 php8.2-opcache amd64 8.2.23-1+0~20240830.60+debian12~1.gbp2d65da [346 kB]
Réception de :3 https://packages.sury.org/php bookworm/main amd64 php8.2-readline amd64 8.2.23-1+0~20240830.60+debian12~1.gbp2d65da [12,5 kB]
Réception de :4 https://packages.sury.org/php bookworm/main amd64 php8.2-cli amd64 8.2.23-1+0~20240830.60+debian12~1.gbp2d65da [1 737 kB]
Réception de :5 https://packages.sury.org/php bookworm/main amd64 php8.2-fpm amd64 8.2.23-1+0~20240830.60+debian12~1.gbp
```

**php8.2** : Installe PHP 8.2 de base, c'est-à-dire le moteur PHP lui-même.

**php8.2-cli** : Installe l'interface en ligne de commande (CLI) de PHP 8.2, ce qui permet d'exécuter des scripts PHP depuis le terminal.

**php8.2-cgi** : Installe le binaire PHP pour exécuter des scripts PHP avec le serveur CGI.

**php8.2-fpm** : PHP-FPM (FastCGI Process Manager), qui est une alternative plus performante à l'usage du PHP via CGI. Il est souvent utilisé avec Nginx.

**php8.2-gd** : Installe l'extension GD pour manipuler des images (création, édition, etc.) dans PHP.

**php8.2-mysql** : Installe l'extension permettant à PHP de se connecter à une base de données MySQL ou MariaDB.

**php8.2-imap** : Installe l'extension IMAP pour interagir avec des serveurs de messagerie via IMAP.

**php8.2-curl** : Permet à PHP de faire des requêtes HTTP et FTP en utilisant cURL.

**php8.2-intl** : Fournit des fonctions pour la gestion des formats internationaux (dates, nombres, etc.).

**php8.2-pspell** : Ajoute le support pour la vérification orthographique dans PHP.

**php8.2-sqlite3** : Permet à PHP de travailler avec des bases de données SQLite3.

**php8.2-tidy** : Fournit des outils pour nettoyer et corriger les fichiers HTML et XML.

**php8.2-xsl** : Ajoute le support de la transformation de documents XML via XSLT.

**php8.2-zip** : Permet de gérer les fichiers compressés au format ZIP dans PHP.

**php8.2-mbstring** : Ajoute des fonctions pour gérer des chaînes de caractères en multibyte, utile pour les langues nécessitant plusieurs octets par caractère (ex : japonais, chinois).



# La Plateforme

**php8.2-soap** : Active le protocole SOAP pour permettre à PHP de communiquer avec des services web SOAP.

**php8.2-opcache** : Active le mécanisme de cache OPcache qui améliore les performances des scripts PHP en stockant les résultats compilés.

**libonig5** : C'est une bibliothèque nécessaire pour gérer les expressions régulières dans PHP, utilisée principalement avec l'extension **mbstring**.

**php8.2-common** : Contient les fichiers communs à toutes les versions de PHP 8.2.

**php8.2-readline** : Ajoute un support pour l'édition en ligne de commande avec l'extension **readline**, ce qui facilite l'utilisation de PHP en mode CLI.

**php8.2-xml** : Installe les fonctions de traitement XML (DOM, SimpleXML, XMLReader, etc.) dans PHP.

- On vérifie que tout est bien installé pour les deux :

```
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:~# php8.2 -v
PHP 8.2.23 (cli) (built: Aug 30 2024 11:34:15) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.23, Copyright (c) Zend Technologies
    with Zend OPcache v8.2.23, Copyright (c), by Zend Technologies
```

- On ajoute un fichier **phpinfo()** : Pour vérifier que PHP fonctionne correctement :

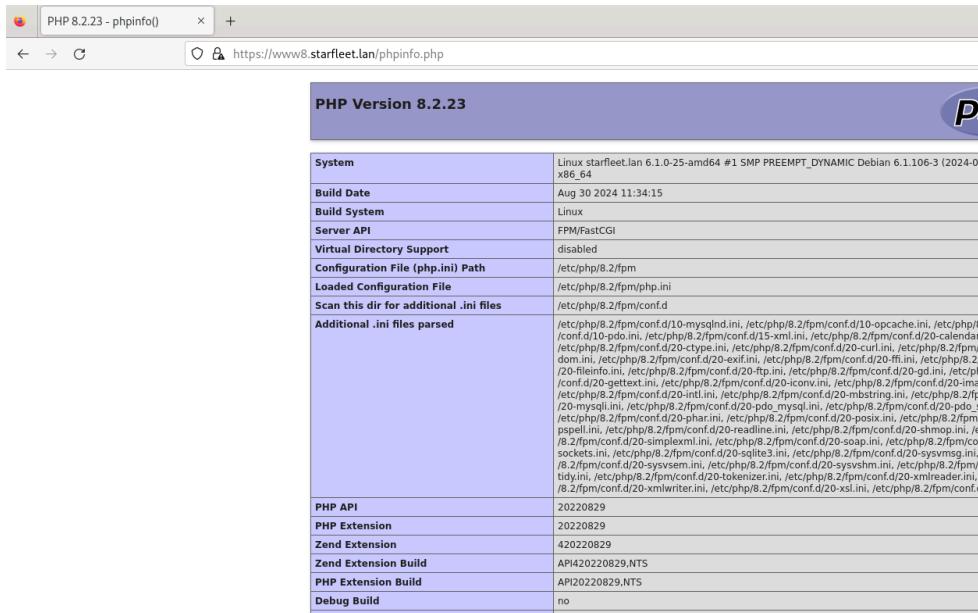
- On crée le fichier **phpinfo.php**

```
root@starfleet:/var/www/www8# root@starfleet:/var/www/www8# nano /var/www/www8/phpinfo.php
```

```
GNU nano 7.2                                         /var/www/www8/phpinfo.php
<?php
phpinfo();
?>
```

# La Plateforme

- On va ensuite sur notre machine Client pour vérifier :

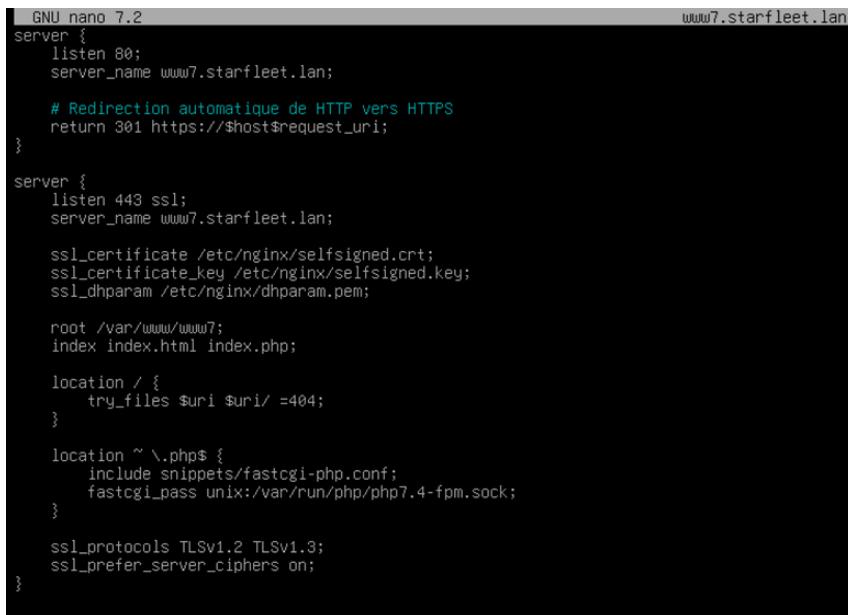


The screenshot shows the PHPinfo output for PHP Version 8.2.23. Key sections include:

- System:** Linux starfleet.lan 6.1.0-25-amd64 #1 SMP PREEMPT\_DYNAMIC Debian 6.1.106-3 (2024-08-x86\_64)
- Build Date:** Aug 30 2024 11:34:15
- Build System:** Linux
- Server API:** FPM/FastCGI
- Virtual Directory Support:** disabled
- Configuration File (php.ini) Path:** /etc/php/8.2/fpm
- Loaded Configuration File:** /etc/php/8.2/fpm/php.ini
- Scan this dir for additional .ini files:** /etc/php/8.2/fpm/conf.d
- Additional .ini files parsed:** /etc/php/8.2/fpm/conf.d/10-mysqlind.ini, /etc/php/8.2/fpm/conf.d/10-opcache.ini, /etc/php/8.2/fpm/conf.d/10-pdo.ini, /etc/php/8.2/fpm/conf.d/15-xml.ini, /etc/php/8.2/fpm/conf.d/20-calendar.ini, /etc/php/8.2/fpm/conf.d/20-ctype.ini, /etc/php/8.2/fpm/conf.d/20-curini.ini, /etc/php/8.2/fpm/conf.d/20-dom.ini, /etc/php/8.2/fpm/conf.d/20-exif.ini, /etc/php/8.2/fpm/conf.d/20-fi.ini, /etc/php/8.2/fpm/conf.d/20-fileinfo.ini, /etc/php/8.2/fpm/conf.d/20-ftp.ini, /etc/php/8.2/fpm/conf.d/20-gd.ini, /etc/php/8.2/fpm/conf.d/20-gettext.ini, /etc/php/8.2/fpm/conf.d/20-iconv.ini, /etc/php/8.2/fpm/conf.d/20-imap.ini, /etc/php/8.2/fpm/conf.d/20-mbstring.ini, /etc/php/8.2/fpm/conf.d/20-mysqli.ini, /etc/php/8.2/fpm/conf.d/20-pdo\_mysql.ini, /etc/php/8.2/fpm/conf.d/20-pdo\_sq...
- PHP API:** 20220829
- PHP Extension:** 20220829
- Zend Extension:** 420220829
- Zend Extension Build:** API420220829.NTS
- PHP Extension Build:** API20220829.NTS
- Debug Build:** no

- Pour PhP 7 on reprend la logique qu'on a utilisé avec Php8 :

- 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
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/

# La Plateforme

- nano /var/www/www7/index.html

```
GNU nano 7.2 index.html
<h1> Welcome to www7.starfleet.lan</h1>
```

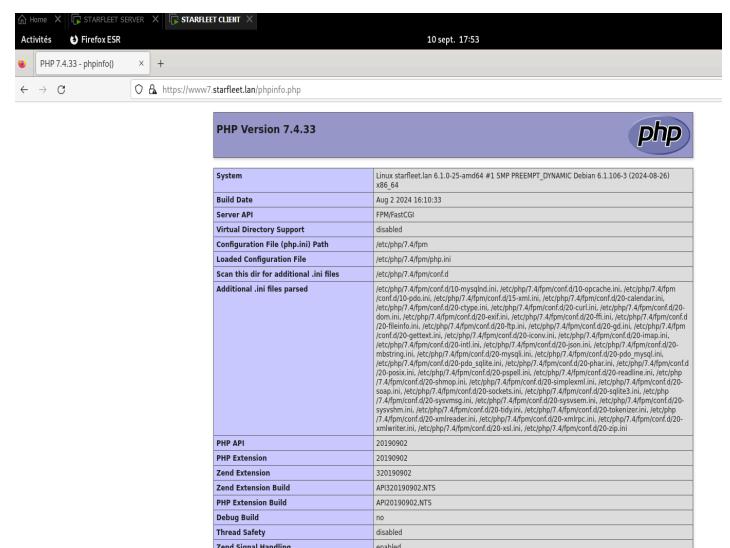
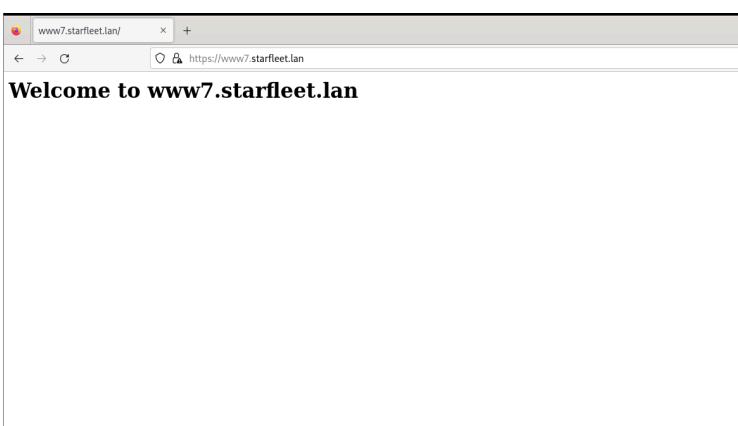
- nano /var/www/www7/phpinfo.php

```
GNU nano 7.2                               /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



### 1.3.4 Installation de MariaDB (dernière version)

- On configure et installe MariaDB (base de données SQL) :

```
root@starfleet:/# apt install mariadb-server mariadb-client
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  galera-4 gawk libcgi-fast-perl libcgi-pm-perl libclone-perl libconfig-inifiles-perl libdaxctl1 libdbd-mariadb-perl
  libdbi-perl libencode-locale-perl libfcgi-bin libfcgi-perl libfcgioldbl libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl liblzo2-2
  libmariadb3 libmpfr6 libncurses6 libndctl6 libpmem1 librexp-ipv6-perl libsigsegv2 libsnappy1v5
  libterm-readkey-perl libtimedate-perl liburi-perl liburing2 mariadb-client-core mariadb-common
  mariadb-plugin-provider-bzip2 mariadb-plugin-provider-lz4 mariadb-plugin-provider-lzma mariadb-plugin-provider-lzo
  mariadb-plugin-provider-snappy mariadb-server-core mysql-common pv rsync socat
```

- On vérifie le statut pour voir si ça fonctionne correctement :

```
root@starfleet:/# 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 Wed 2024-09-11 11:09:45 CEST; 2min 21s ago
    Docs: man:mariadb(8)
          https://mariadb.com/kb/en/library/systemd/
   Main PID: 2575 (mariadb)
     Status: "Taking your SQL requests now..."
       Tasks: 9 (limit: 2273)
      Memory: 202.8M
        CPU: 567ms
      CGroup: /system.slice/mariadb.service
              └─2575 /usr/sbin/mariadb

sept. 11 11:09:45 starfleet.lan mariadb[2575]: 2024-09-11 11:09:45 0 [Note] Plugin 'FEEDBACK' is disabled.
sept. 11 11:09:45 starfleet.lan mariadb[2575]: 2024-09-11 11:09:45 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql
sept. 11 11:09:45 starfleet.lan mariadb[2575]: 2024-09-11 11:09:45 0 [Warning] You need to use --log-bin to make --exp
sept. 11 11:09:45 starfleet.lan mariadb[2575]: 2024-09-11 11:09:45 0 [Note] Server socket created on IP: '127.0.0.1'.
sept. 11 11:09:45 starfleet.lan mariadb[2575]: 2024-09-11 11:09:45 0 [Note] InnoDB: Buffer pool(s) load completed at 2
sept. 11 11:09:45 starfleet.lan mariadb[2575]: 2024-09-11 11:09:45 0 [Note] /usr/sbin/mariadb: ready for connections.
sept. 11 11:09:45 starfleet.lan mariadb[2575]: Version: '10.11.6-MariaDB-0+deb12u1' socket: '/run/mysqld/mysqld.sock'
sept. 11 11:09:45 starfleet.lan systemd[1]: Started mariadb.service - MariaDB 10.11.6 database server.
sept. 11 11:09:45 starfleet.lan /etc/mysql/debian-start[2602]: Checking for insecure root accounts.
sept. 11 11:09:45 starfleet.lan /etc/mysql/debian-start[2607]: Triggering myisam-recover for all MyISAM tables and aria
| Lines 1-23/23 (END)
```

- Ensuite on se connecte à mysql :

```
root@starfleet:/# mysql -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> |
```

# La Plateforme

- Maintenant on va créer notre base de données :

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE starfleet_db;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> CREATE USER 'starfleet_user'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON starfleet_db.* TO 'starfleet_user'@'localhost';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,000 sec)

MariaDB [(none)]> EXIT;
Bye
```

- Une fois que c'est fait on va créer une base de données :

```
MariaDB [(none)]> SELECT User, Host FROM mysql.user;
+-----+-----+
| User      | Host     |
+-----+-----+
| mariadb.sys | localhost |
| mysql       | localhost |
| root        | localhost |
| starfleet_user | localhost |
+-----+-----+
4 rows in set (0,001 sec)

MariaDB [(none)]>
```

- Pour voir les bases de données :

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| starfleet_db   |
| sys            |
+-----+
5 rows in set (0,001 sec)
```



# La Plateforme

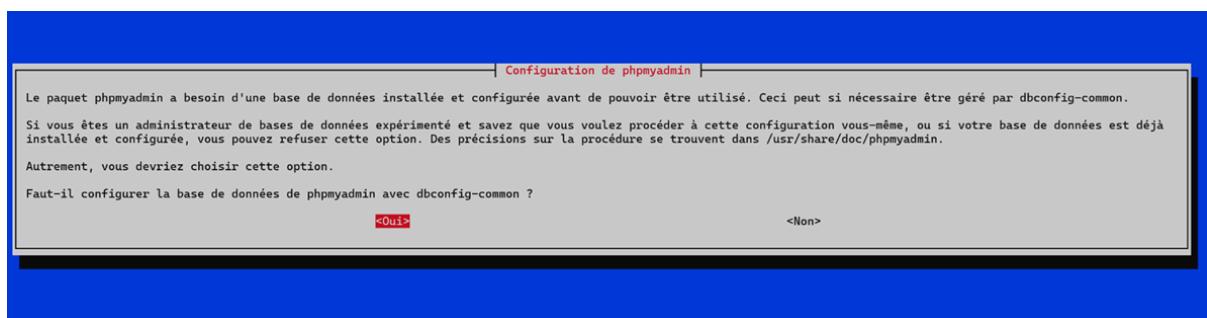
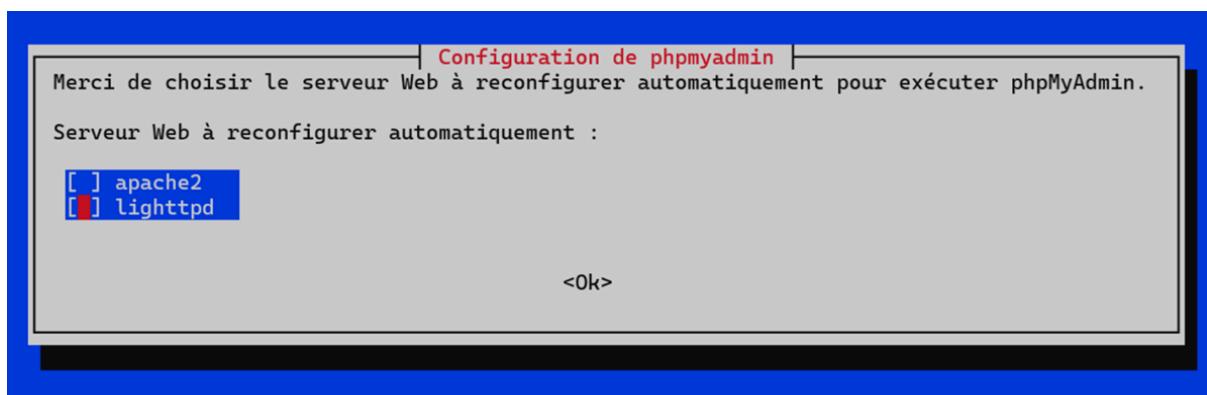
- Pour voir les droits d'un utilisateur :

```
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`
+-----+
2 rows in set (0,000 sec)
```

- Pour renforcer la sécurité de notre base de données on peut utiliser :  
`mysql_secure_installation`

### 1.3.5 Installation et configuration de PhPmyAdmin

- On utilise la commande suivante pour installer PhPmyAdmin :  
`apt install phpmyadmin`



# La Plateforme

- 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
```

- On se rend ensuite dans le fichier "php.starfleet.lan" :  
`nano /etc/nginx/sites-available/php.starfleet.lan`

```
GNU nano 7.2
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;

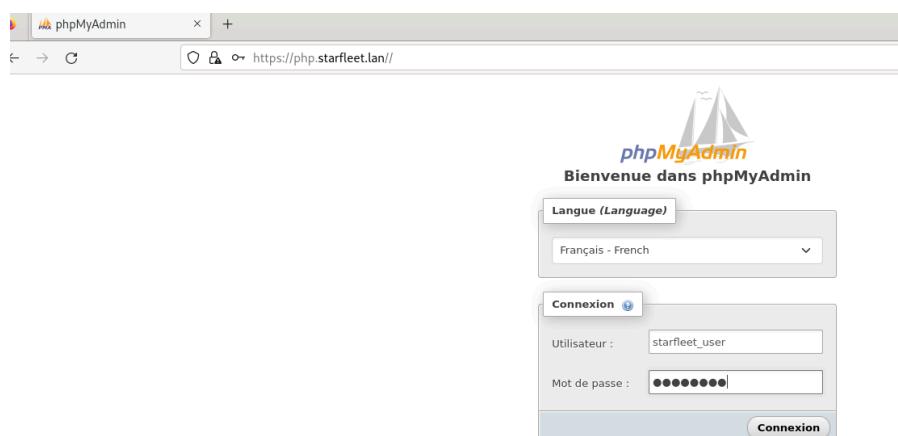
    location / {
        try_files $uri $uri/ =404;
    }

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

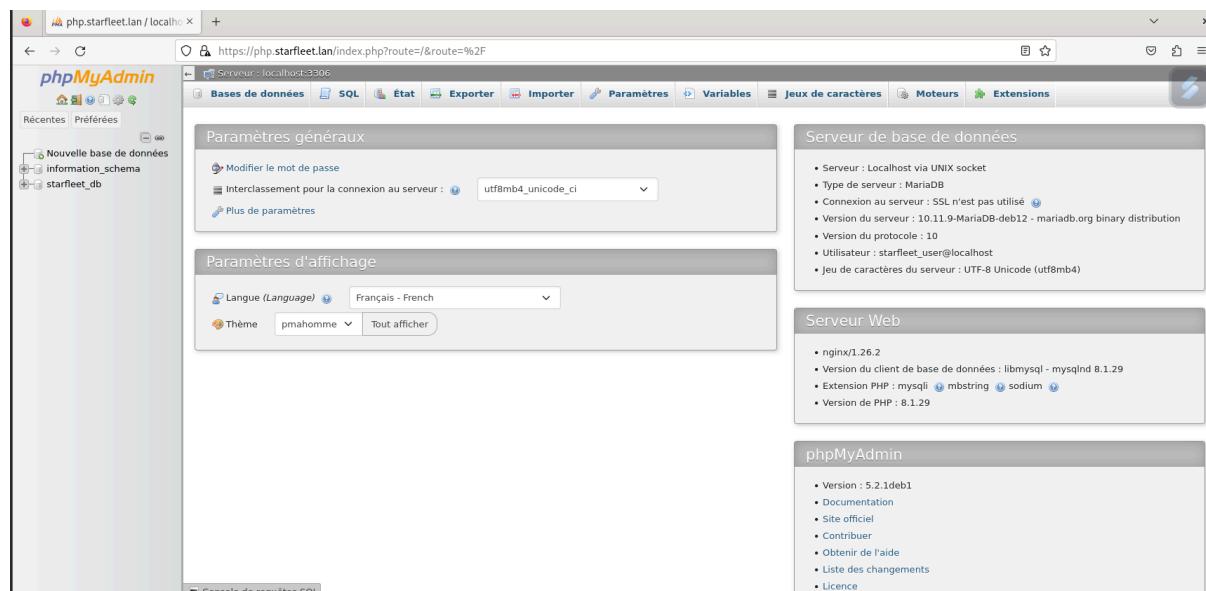
- Une fois le fichier créé, nous avons activé ce site en créant un lien symbolique dans le dossier sites-enabled :

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

- On retourne sur notre Client pour vérifier que ça marche :



# La Plateforme



## 1.3.5 Installation et configuration de SFTP

- On crée notre groupe d'utilisateurs qui sera dédié à SFTP

```
root@starfleet:~# groupadd sfptusers
```

- Ensuite on va dans le fichier de configuration de 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 sfptusers
    ChrootDirectory %h
    X11Forwarding no
    AllowTcpForwarding no
    ForceCommand internal-sftp
    GatewayPorts no
```

On définit des règles spécifiques pour un groupe d'utilisateurs (groupe que nous avons préalablement créé).

**Match Group sfptusers :** Cette instruction indique que les directives suivantes s'appliquent uniquement aux membres du groupe **sfptusers**. Si l'on souhaite limiter l'accès de certains utilisateurs à SFTP uniquement, il faut les ajouter à ce groupe.

# La Plateforme

On ajoute notre utilisateur à notre groupe :

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

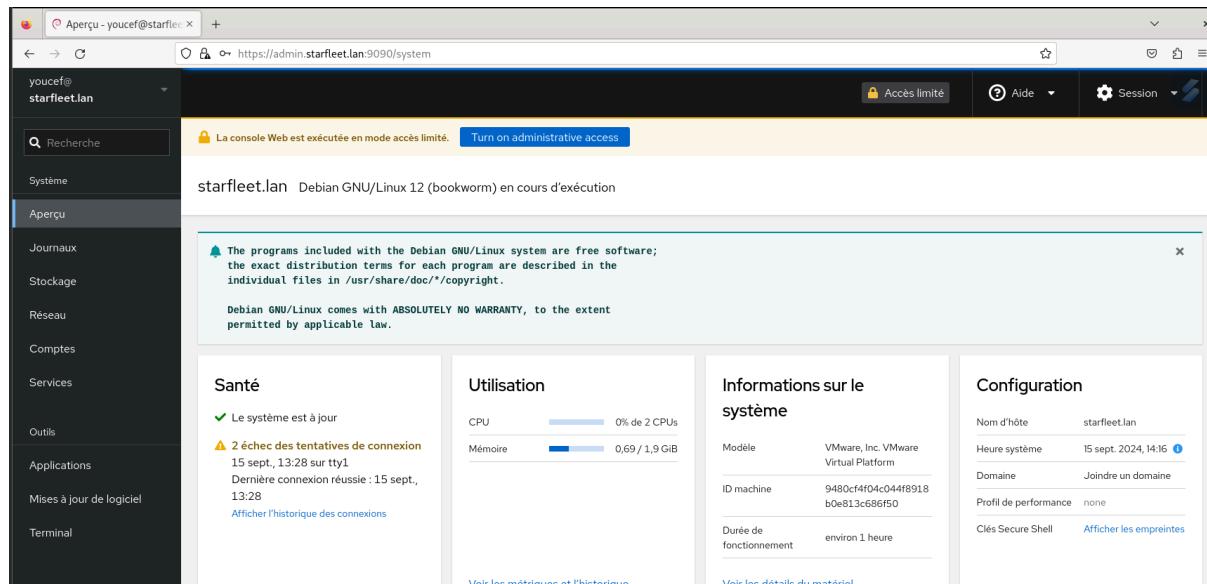
```
root@starfleet:~# passwd admin
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
```

```
root@starfleet:~# chown -R admin:www-data /var/www/html
```

```
root@starfleet:~# sftp admin@localhost
The authenticity of host 'localhost (::1)' can't be established.
ED25519 key fingerprint is SHA256:0g67y0HPnAFuYrHtVRcWoj8eaMcZbGaLru77vUjuafk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
admin@localhost's password:
Connected to localhost.
sftp>
```

## 1.3.6 Installation et configuration de CockPit

- On commence par installer le paquet "cockpit" avant ça on met à jour le cache des paquets.



# La Plateforme

## 1.3.7 Installation et configuration de UFW et autorisations des services

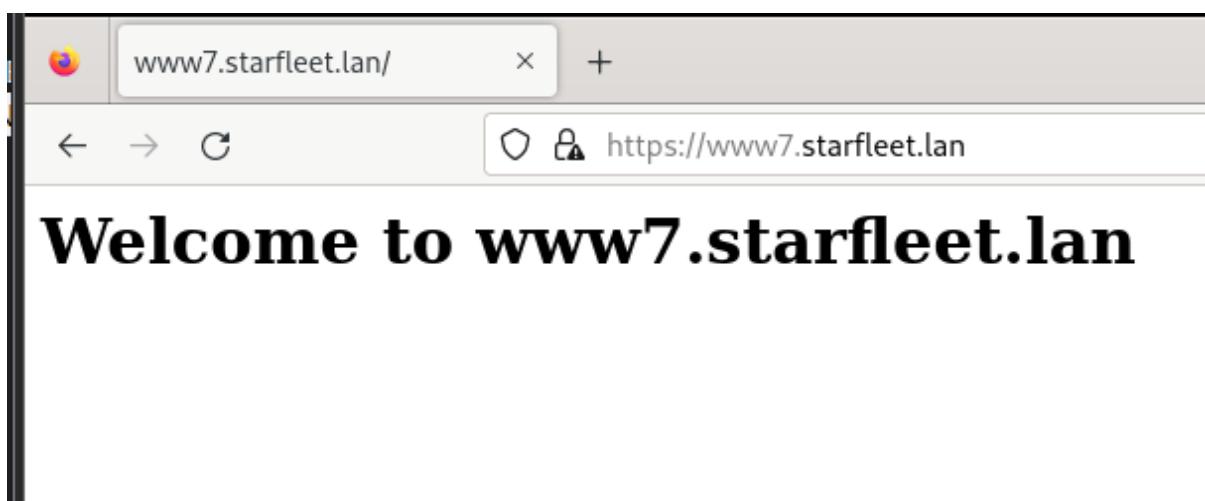
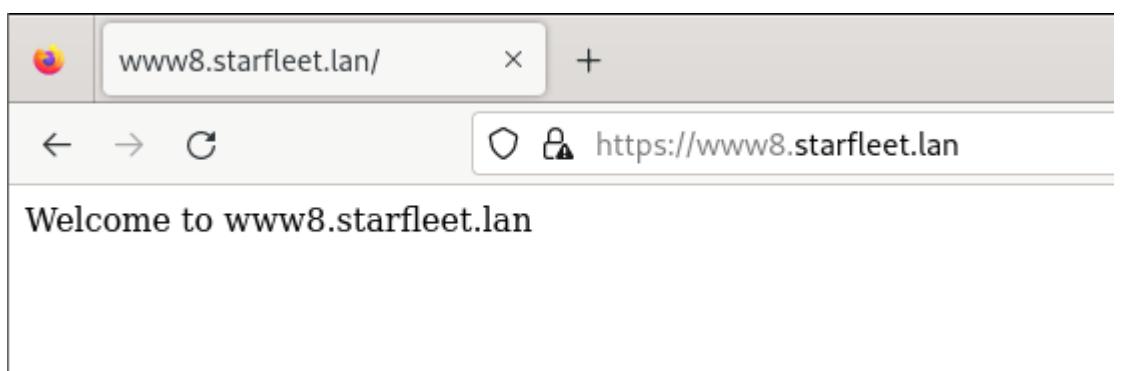
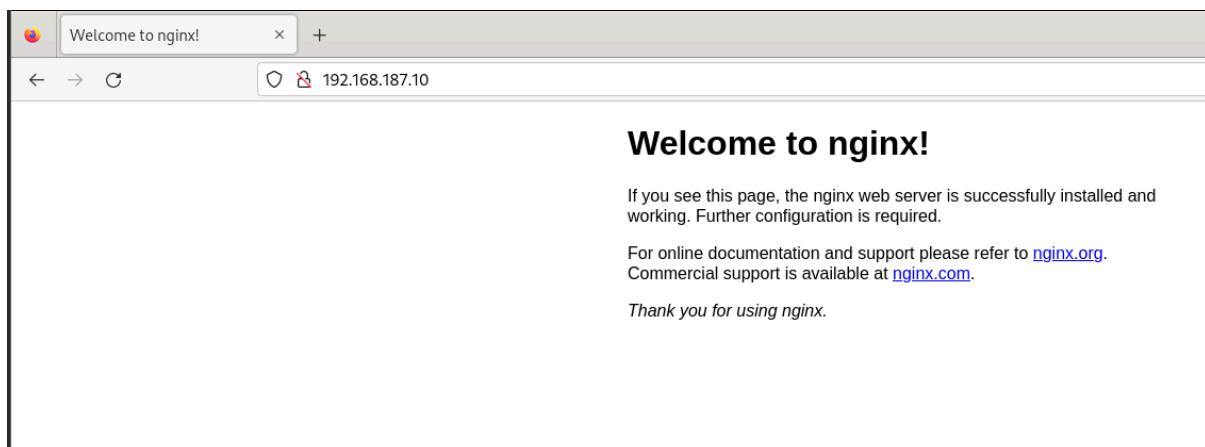
- apt update
- apt install ufw
- ufw status
- ufw allow ftp

Autorisation des services par port avec UFW :

```
ufw allow mysql  
ufw allow 67/udp (DHCP)
```

To	Action	From
--	-----	-----
22/tcp	ALLOW	Anywhere
80/tcp	ALLOW	Anywhere
443	ALLOW	Anywhere
DNS	ALLOW	Anywhere
21/tcp	ALLOW	Anywhere
3306/tcp	ALLOW	Anywhere
67/udp	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)
80/tcp (v6)	ALLOW	Anywhere (v6)
443 (v6)	ALLOW	Anywhere (v6)
DNS (v6)	ALLOW	Anywhere (v6)
21/tcp (v6)	ALLOW	Anywhere (v6)
3306/tcp (v6)	ALLOW	Anywhere (v6)
67/udp (v6)	ALLOW	Anywhere (v6)





# La Plateforme

PHP 7.4.33 - phpinfo()

https://www7.starfleet.lan/phpinfo.php

**PHP Version 7.4.33**

<b>System</b>	Linux starfleet.lan 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64
<b>Build Date</b>	Aug 2 2024 16:10:33
<b>Server API</b>	FPM/FastCGI
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php/7.4/fpm
<b>Loaded Configuration File</b>	/etc/php/7.4/fpm/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php/7.4/fpm/conf.d
<b>Additional .ini files parsed</b>	/etc/php/7.4/fpm/conf.d/10-mysqlind.ini, /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/15-xml.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-curl.ini, /etc/php/7.4/fpm/conf.d/20-dom.ini, /etc/php/7.4/fpm/conf.d/20-exif.ini, /etc/php/7.4/fpm/conf.d/20-ffi.ini, /etc/php/7.4/fpm/conf.d/20-fileinfo.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-imap.ini, /etc/php/7.4/fpm/conf.d/20-intl.ini, /etc/php/7.4/fpm/conf.d/20-json.ini, /etc/php/7.4/fpm/conf.d/20-mbstring.ini, /etc/php/7.4/fpm/conf.d/20-mysqli.ini, /etc/php/7.4/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.4/fpm/conf.d/20-pdo_sqlite.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-pspell.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-simplexml.ini, /etc/php/7.4/fpm/conf.d/20-soap.ini, /etc/php/7.4/fpm/conf.d/20-sockets.ini, /etc/php/7.4/fpm/conf.d/20-sqlite3.ini, /etc/php/7.4/fpm/conf.d/20-sysmsg.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-tidy.ini, /etc/php/7.4/fpm/conf.d/20-tokenizer.ini, /etc/php/7.4/fpm/conf.d/20-xmlreader.ini, /etc/php/7.4/fpm/conf.d/20-xmlrpc.ini, /etc/php/7.4/fpm/conf.d/20-xsl.ini, /etc/oh/o/7.4/fpm/conf.d/20-zip.ini

PHP 8.2.23 - phpinfo()

https://www8.starfleet.lan/phpinfo.php

**PHP Version 8.2.23**

<b>System</b>	Linux starfleet.lan 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64
<b>Build Date</b>	Aug 30 2024 11:34:15
<b>Build System</b>	Linux
<b>Server API</b>	FPM/FastCGI
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php/8.2/fpm
<b>Loaded Configuration File</b>	/etc/php/8.2/fpm/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php/8.2/fpm/conf.d
<b>Additional .ini files parsed</b>	/etc/php/8.2/fpm/conf.d/10-mysqlind.ini, /etc/php/8.2/fpm/conf.d/10-opcache.ini, /etc/php/8.2/fpm/conf.d/10-pdo.ini, /etc/php/8.2/fpm/conf.d/15-xml.ini, /etc/php/8.2/fpm/conf.d/20-calendar.ini, /etc/php/8.2/fpm/conf.d/20-ctype.ini, /etc/php/8.2/fpm/conf.d/20-curl.ini, /etc/php/8.2/fpm/conf.d/20-dom.ini, /etc/php/8.2/fpm/conf.d/20-exif.ini, /etc/php/8.2/fpm/conf.d/20-ffi.ini, /etc/php/8.2/fpm/conf.d/20-fileinfo.ini, /etc/php/8.2/fpm/conf.d/20-gettext.ini, /etc/php/8.2/fpm/conf.d/20-iconv.ini, /etc/php/8.2/fpm/conf.d/20-imap.ini, /etc/php/8.2/fpm/conf.d/20-intl.ini, /etc/php/8.2/fpm/conf.d/20-json.ini, /etc/php/8.2/fpm/conf.d/20-mbstring.ini, /etc/php/8.2/fpm/conf.d/20-mysqli.ini, /etc/php/8.2/fpm/conf.d/20-pdo_mysql.ini, /etc/php/8.2/fpm/conf.d/20-pdo_sqlite.ini, /etc/php/8.2/fpm/conf.d/20-phar.ini, /etc/php/8.2/fpm/conf.d/20-posix.ini, /etc/php/8.2/fpm/conf.d/20-pspell.ini, /etc/php/8.2/fpm/conf.d/20-readline.ini, /etc/php/8.2/fpm/conf.d/20-shmop.ini, /etc/php/8.2/fpm/conf.d/20-simplexml.ini, /etc/php/8.2/fpm/conf.d/20-soap.ini, /etc/php/8.2/fpm/conf.d/20-sockets.ini, /etc/php/8.2/fpm/conf.d/20-sqlite3.ini, /etc/php/8.2/fpm/conf.d/20-sysmsg.ini, /etc/php/8.2/fpm/conf.d/20-sysvsem.ini, /etc/php/8.2/fpm/conf.d/20-sysvshm.ini, /etc/php/8.2/fpm/conf.d/20-tidy.ini, /etc/php/8.2/fpm/conf.d/20-tokenizer.ini, /etc/php/8.2/fpm/conf.d/20-xmlreader.ini, /etc/php/8.2/fpm/conf.d/20-xsl.ini, /etc/php/8.2/fpm/conf.d/20-zip.ini
<b>PHP API</b>	20220829
<b>PHP Extension</b>	20220829
<b>Zend Extension</b>	420220829