

# DDWS

DHCP DNS Web server Samba

BEAUSSAERT Valentin

# Installation d'un serveur Web Apache 2

## Sur une VM Linux Debian 12

- Avant de commencer le paramétrage et l'installation, il est préférable de mettre à jour les paquets avec « sudo apt update » et « sudo apt upgrade » pour installer les derniers paquets.
- Une fois les paquets mis à jour on peut installer Apache2 via la commande « sudo apt install apache2 »

```
valuxadmi@debian:~$ sudo apt install apache2
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 :
  apache2-data apache2-utils
Paquets suggérés :
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Les NOUVEAUX paquets suivants seront installés :
  apache2 apache2-data apache2-utils
0 mis à jour, 3 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 574 ko dans les archives.
Après cette opération, 2 320 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] 0
```

- Après l'installation nous pouvons démarrer et activer le service Apache2 pour qu'il démarre automatiquement au démarrage de la machine grâce aux commandes « sudo systemctl start apache2 » et « sudo systemctl enable apache2 »
- Nous devons prendre connaissance de l'adresse IP de notre VM afin de pouvoir tester le service Apache2. On entre donc la commande « ip addr show ».

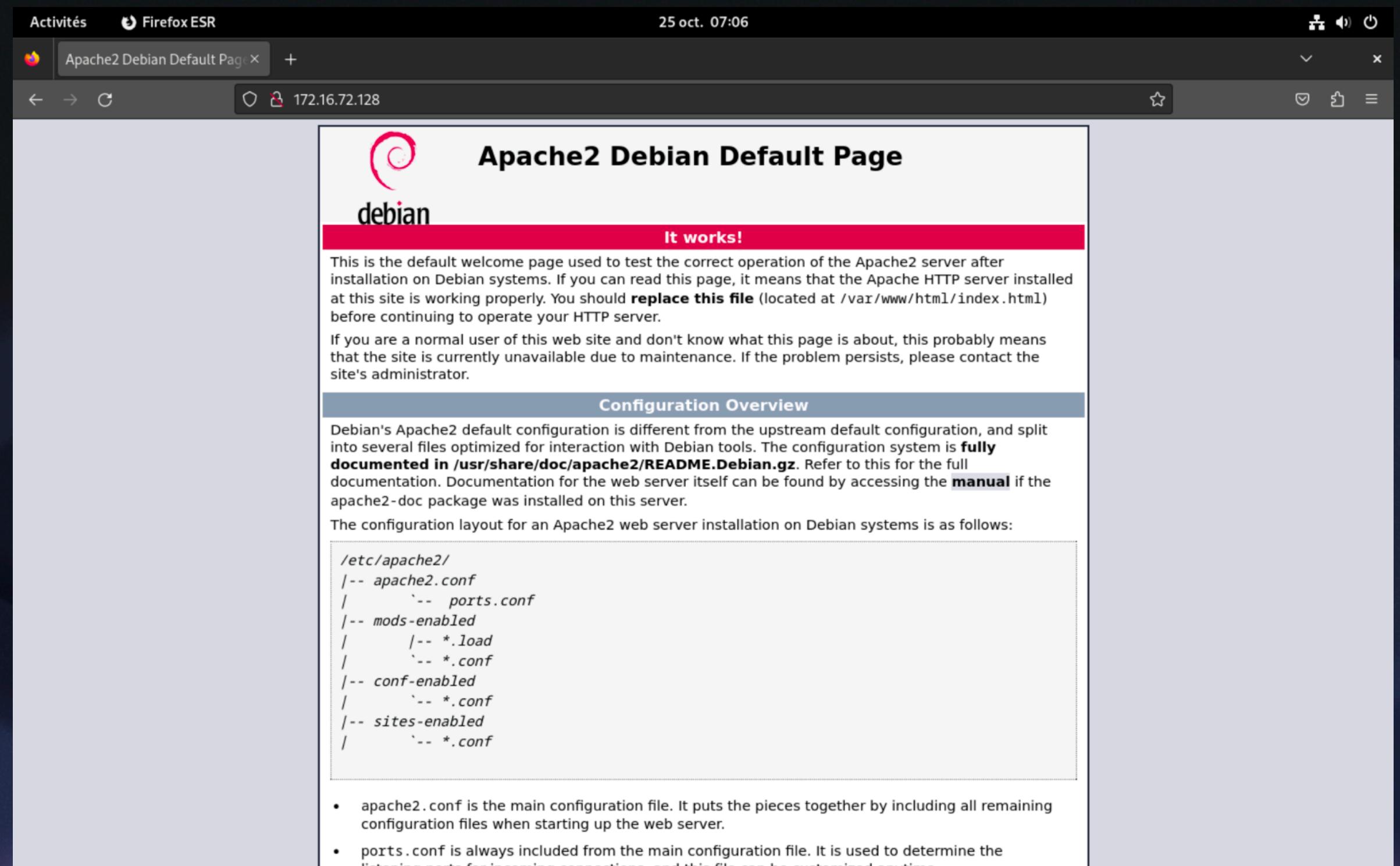
```
valuxadmi@debian:~$ sudo systemctl start apache2
valuxadmi@debian:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
valuxadmi@debian:~$ ip addr show
1: 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: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:83:d5:56 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 172.16.72.128/24 brd 172.16.72.255 scope global dynamic noprefixroute ens160
```

- Avant de rentrer notre adresse IP dans l'URL pour tester le serveur, nous allons configurer le pare feu afin d'autoriser le trafic HTTP et HTTPS grâce aux commandes « sudo ufw allow http » et « sudo ufw allow https ».

- On entre donc les commandes de la slide précédentes et nous avons :

```
valuxadmi@debian:~$ sudo ufw allow http
Rule added
Rule added (v6)
valuxadmi@debian:~$ sudo ufw allow https
Rule added
Rule added (v6)
```

- Une fois le pare feu configuré on redémarre le service Apache2 en entrant la commande « sudo systemctl restart apache2 »
- Nous pouvons à présent nous rendre dans une URL et rentrer notre adresse IP. Si tout à fonctionner correctement nous devrions avoir une page nous indiquant que Apache2 est bien en train de fonctionner.



# Les différents serveurs Web

Leurs avantages et leurs inconvénients :

## **Apache HTTP Server (ou Apache)**

Avantages:

- Vaste communauté et de nombreuses années de développement.
- Hautement configurable et extensible avec des modules.
- Compatible avec de nombreux systèmes d'exploitation.

Inconvénients:

- Peut consommer plus de ressources que certains concurrents plus récents.
- Moins performant sous des charges très lourdes.

# Nginx

## Avantages:

- Haute performance et faible consommation de ressources.
- Excellente gestion des connexions simultanées.
- Peut également être utilisé comme un serveur proxy inverse.

## Inconvénients:

- Configuration légèrement plus complexe pour certaines tâches avancées par rapport à Apache.
- Bien que la documentation soit complète, elle peut être moins accessible pour les débutants.

## LiteSpeed

### Avantages:

- Conçu pour la vitesse et la performance.
- Interface d'administration web conviviale.
- Compatible avec les fichiers .htaccess d'Apache.

### Inconvénients:

- La version entreprise est payante.
- Moins populaire, donc moins de ressources et de communauté que Apache ou Nginx.

## **Microsoft Internet Information Services (IIS)**

### Avantages:

- Intégration étroite avec le système d'exploitation Windows et les applications Microsoft.
- Interface graphique pour la gestion et la configuration.

### Inconvénients:

- Limité à la plateforme Windows.
- Historiquement, il a eu des problèmes de sécurité.

## **Tomcat**

### Avantages:

- Serveur d'applications Java, idéal pour les applications web Java.
- Fait partie de la fondation Apache, avec une solide base de support.

### Inconvénients:

- Spécifique aux applications Java.
- Moins performant comme serveur web statique par rapport à des solutions comme Nginx ou Apache.

## Caddy

### Avantages:

- Configuration automatique des certificats SSL grâce à Let's Encrypt.
- Simple à configurer et à utiliser.

### Inconvénients:

- Moins connu et moins utilisé que Apache ou Nginx.
- Certains aspects de la licence peuvent ne pas convenir à tous les utilisateurs ou cas d'utilisation.

Chaque serveur a ses propres forces en fonction de l'utilisation prévue. Le choix dépendra de nos besoins spécifiques, de notre environnement d'exploitation, de notre niveau d'expertise et d'autres facteurs.

# Mise en place d'un serveur DNS

Dans une VM Linux Debian

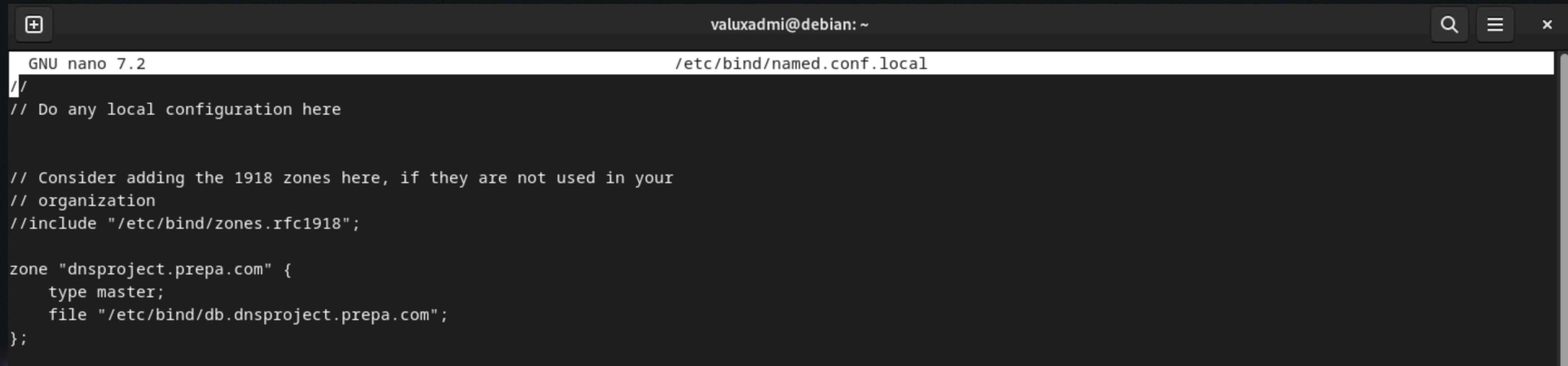
- Afin de pouvoir connecter plusieurs machines de notre réseau local vers "[dnsproject.prepa.com](http://dnsproject.prepa.com)" nous allons configurer un serveur DNS local avec Bind9.

Pour commencer, nous allons installer bind9 via la commande « sudo apt-get install bind9 ».

```
valuxadmi@debian:~$ sudo apt-get install bind9
[sudo] Mot de passe de valuxadmi :
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 :
  bind9-utils
Paquets suggérés :
  bind-doc resolvconf
Les NOUVEAUX paquets suivants seront installés :
  bind9 bind9-utils
0 mis à jour, 2 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 874 ko dans les archives.
Après cette opération, 2 962 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
```

Nous allons à présent ouvrir le fichier de configuration de bind pour ajouter une nouvelle zone. Pour cela nous entrons la commande « sudo nano /etc/bind/named.conf.local » et nous tapons les informations de notre zone dans le fichier que nous venons de créer.

Lorsque vous configurez un serveur DNS, vous devez définir des "zones". Une zone est essentiellement une portion de l'espace de noms DNS



```
valuxadmi@debian: ~
GNU nano 7.2
/etc/bind/named.conf.local

// Do any local configuration here

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

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

Ceci indique à BIND que pour le domaine "dnsproject.prepa.com", il doit agir en tant que serveur principal (ou "master") et chercher les détails du domaine dans le fichier /etc/bind/db.dnsproject.prepa.com.

Nous pouvons enregistrer et fermer le nano afin d'en créer un nouveau qui contiendra les informations de la zone que l'on vient de créer.

On entre donc la commande « sudo nano /etc/bind/db.dnsproject.prepa.com » et on rempli le fichier de zone.



```
GNU nano 7.2                               valuxadmi@debian: ~
; BIND data file for dnsproject.prepa.com
;
$TTL 604800
@ IN SOA dnsproject.prepa.com. root.dnsproject.prepa.com. (
        2           ; Serial
    604800       ; Refresh
    86400        ; Retry
   2419200      ; Expire
   604800 )     ; Negative Cache TTL
;
@ IN NS dnsproject.prepa.com.
@ IN A 172.16.72.128
```

- \$TTL 604800 définit le Time To Live par défaut pour le domaine.
- SOA est le Start of Authority record. Il contient des informations générales sur le domaine, telles que l'adresse e-mail de l'administrateur (ici, "root@dnsproject.prepa.com", mais le "@" est remplacé par un point) et d'autres paramètres tels que les intervalles de rafraîchissement.
- NS est le Name Server record. Il indique quel serveur est responsable du domaine.
- A est le Address record. Il associe le nom de domaine à une adresse IP, dans notre cas 127.0.0.1
- Après avoir ajouté ces informations, sauvegardez le fichier et fermez l'éditeur.

Il ne nous reste plus qu'à mettre à jour le résolveur de notre système pour utiliser le serveur DNS.

En effet le système d'exploitation consulte généralement le fichier /etc/resolv.conf pour savoir quels serveurs DNS utiliser pour la résolution des noms. Pour que notre système utilise notre serveur BIND9 comme serveur DNS, nous devons mettre à jour ce fichier.

Pour cela on entre la commande « sudo nano /etc/resolv.conf » et nous allons modifier la ligne comportant une adresse IP pour la remplacer par celle de notre serveur. Nous redémarrerons par la même occasion bind9 afin d'appliquer les changements en entrant la commande « sudo systemctl restart bind9 ».

Nous allons à présent ping « [dnsproject.prepa.com](http://dnsproject.prepa.com) » si les packets sont reçus, c'est que tout est bien configuré.

```
valuxadmi@debian:~$ ping dnsproject.prepa.com
PING dnsproject.prepa.com (172.16.72.128) 56(84) bytes of data.
64 bytes from debian (172.16.72.128): icmp_seq=1 ttl=64 time=0.024 ms
64 bytes from debian (172.16.72.128): icmp_seq=2 ttl=64 time=0.077 ms
64 bytes from debian (172.16.72.128): icmp_seq=3 ttl=64 time=0.151 ms
64 bytes from debian (172.16.72.128): icmp_seq=4 ttl=64 time=0.108 ms
^C
--- dnsproject.prepa.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 0.024/0.090/0.151/0.046 ms
```

Tout indique que notre DNS fait correspondre l'adresse IP de notre serveur au nom de domaine « [dnsproject.prepa.com](http://dnsproject.prepa.com) »

# Comment obtient-on un nom de domaine public ?

## Choisir un nom de domaine :

- Réfléchir au nom de domaine que l'on. Cela peut être le nom de notre entreprise, de notre marque, ou tout autre nom pertinent pour notre site web ou projet.

## Vérifier la disponibilité :

- Avant de pouvoir enregistrer un nom de domaine, nous devons vérifier qu'il est disponible. De nombreux registraires de domaines offrent des outils de recherche pour vérifier la disponibilité d'un nom de domaine

## Sélectionner un registraire de domaines :

- Un registraire de domaines est une entreprise accréditée pour enregistrer et gérer les noms de domaine pour le compte des propriétaires de domaines.

## Enregistrer le nom de domaine :

- Une fois que nous avons vérifié que notre nom de domaine est disponible et que nous avons choisi un registraire, nous pouvons procéder à l'enregistrement. Cela implique généralement de remplir des détails sur le propriétaire du domaine et de payer des frais d'enregistrement.
- La plupart des noms de domaine sont enregistrés pour une durée d'un an, on peut généralement choisir d'enregistrer un domaine pour plusieurs années si on le souhaitez.

## Configurer les enregistrements DNS :

- Après avoir enregistré notre nom de domaine, nous pouvons configurer nos enregistrements DNS pour diriger le trafic vers l'adresse IP de votre serveur web, définir des serveurs de messagerie, etc. Ces paramètres se font généralement via le tableau de bord ou le panneau de gestion fourni par votre registraire

## Renouveler le nom de domaine :

- Les noms de domaine ne sont pas achetés de manière permanente; ils sont loués pendant une certaine durée. Vous devrez renouveler votre nom de domaine avant qu'il n'expire pour continuer à l'utiliser. De nombreux registraires offrent la possibilité de renouvellement automatique pour éviter que votre domaine n'expire accidentellement.

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

Les extensions de noms de domaine, également appelées Top Level Domains (TLDs), peuvent avoir des spécificités ou des restrictions basées sur leur nature et leur objectif.

### Géographiques (ccTLDs) :

- Exemples : .fr (France), .ca (Canada), .uk (Royaume-Uni)
- Spécificités : Beaucoup de ccTLDs ont des restrictions basées sur la résidence ou la présence commerciale dans le pays ou la région concerné. Par exemple, pour enregistrer un .fr, il se peut que vous ayez besoin d'une adresse en France.

### Généralistes (gTLDs) :

- Exemples : .com, .net, .org
- Spécificités : Ces TLDs sont généralement ouverts à tous sans restrictions spécifiques, bien que l'origine historique de certains, comme .org, était destinée aux organisations à but non lucratif.

## Sectoriels (sTLDs) :

- Exemples : .edu (éducation), .gov (gouvernement des États-Unis), .mil (militaire des États-Unis)
- Spécificités : Ces TLDs ont des restrictions basées sur l'entité ou l'industrie. Par exemple, .edu est souvent réservé aux établissements d'enseignement accrédités.

## Marques :

- Exemple : .apple (pour la société Apple Inc.)
- Spécificités : Ces TLDs sont détenus par des marques ou des entreprises et ne sont généralement pas disponibles pour l'enregistrement public. Ils sont utilisés pour des initiatives de branding ou de sécurité.

## Communautaires :

- Exemple : .coop (coopératives)
- Spécificités : Pour des groupes ou des communautés spécifiques. Par exemple, .coop est réservé aux coopératives légitimes.

## Nouveaux gTLDs :

- Exemples : .app, .blog, .guru
- Spécificités : Ces TLDs ont été introduits récemment et couvrent un large éventail de sujets, d'intérêts et d'industries. Les restrictions varient; certains sont ouverts à tous, tandis que d'autres peuvent avoir des critères d'éligibilité.

## Extensions avec des spécificités techniques :

- Exemple : .onion (pour les sites accessibles via le réseau Tor)
- Spécificités : Ces TLDs ont des exigences techniques particulières pour leur utilisation.

## Extensions avec des engagements :

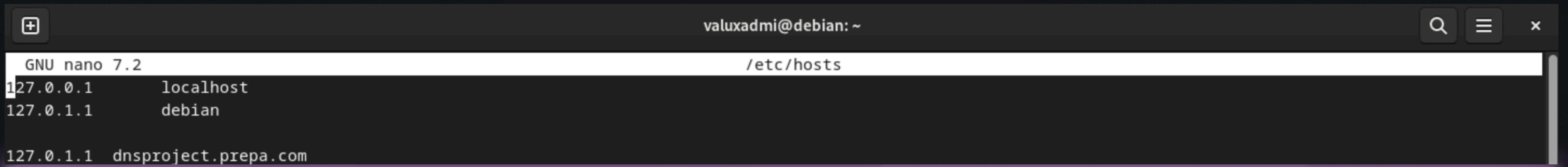
- Exemple : .bank
- Spécificités : L'enregistrement peut nécessiter que l'entité respecte certaines normes ou réglementations, comme des exigences de sécurité renforcées pour .bank.

# Connexion de l'hôte au domaine local

Accessible via le même nom de domaine.

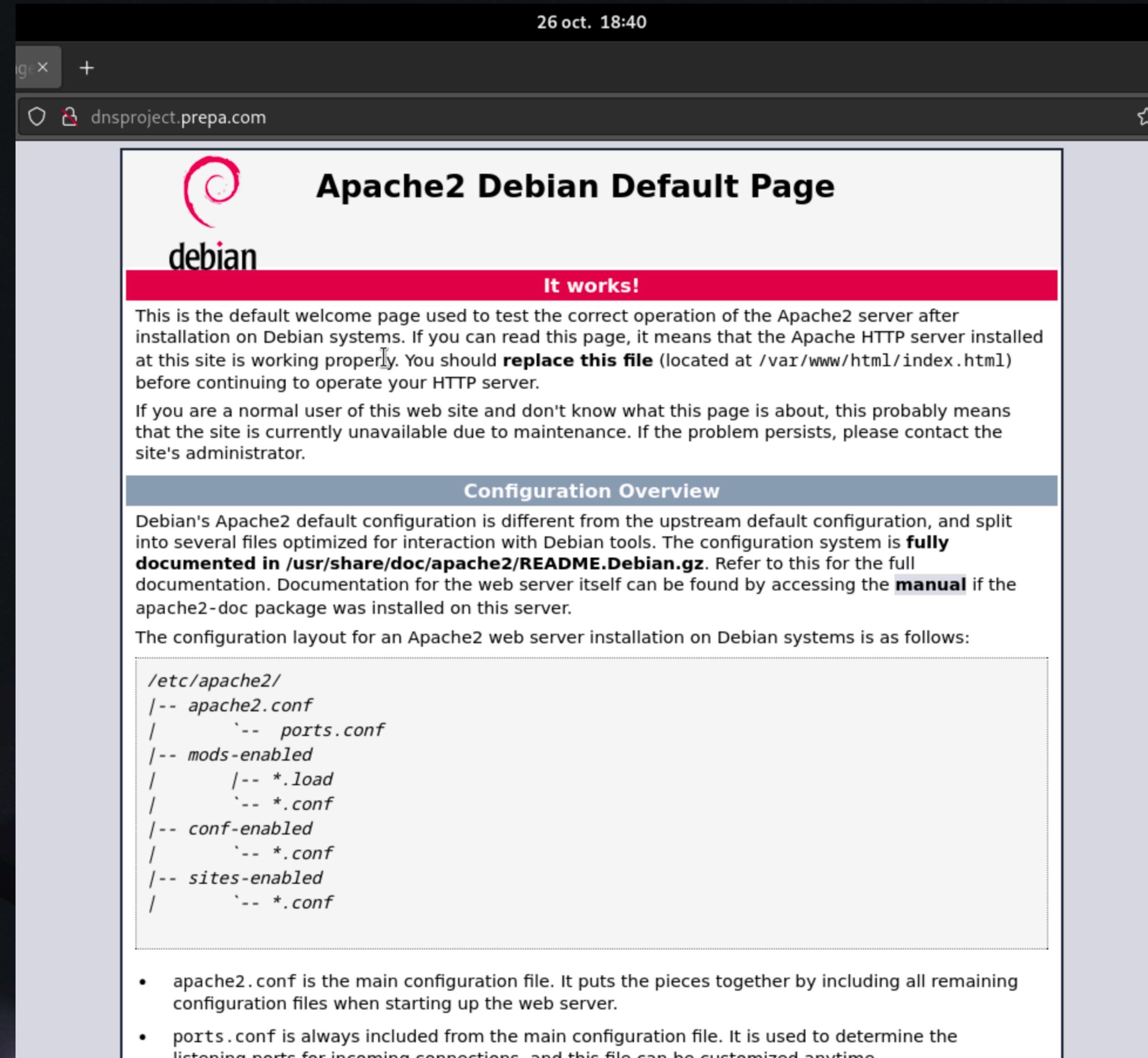
Afin de pouvoir se connecter sur Apache2 depuis dnsproject.prepa.com dans l'URL, il est nécessaire dans un premier temps d'ouvrir le fichier « hosts ».

Pour cela, on rentre la commande « sudo nano /etc/hosts » et on ajoute à la fin de notre fichier : « 127.0.0.1 dnsproject.prepa.com » La suite de chiffre étant notre adresse IP.



```
valuxadmi@debian: ~
GNU nano 7.2
/etc/hosts
127.0.0.1      localhost
127.0.1.1      debian
127.0.1.1  dnsproject.prepa.com
```

Cette manipulation nous permet d'ajouter une entrée au fichier « hosts » de notre hôte, ce qui nous permet de nous connecter à un nom de domaine local.



# Mise en place d'un pare-feu UFW

Empêcher de ping mais accéder à la page par défaut

Tout d'abord, nous allons commencer par vérifier et que nous avons bien autoriser le trafic HTTP et HTTPS via les commandes « sudo ufw allow HTTP » et « sudo ufw allow HTTPS »

```
valuxadmi@debian:~$ sudo ufw allow http
Skipping adding existing rule
Skipping adding existing rule (v6)
valuxadmi@debian:~$ sudo ufw allow https
Skipping adding existing rule
Skipping adding existing rule (v6)
```

Puis nous allons ouvrir le fichier se trouvant dans « /etc/ufw/ before.rules » avec la commande nano pour y ajouter une règle stipulant le « drop » des pings entrant.

```
valuxadmi@debian: ~
/etc/ufw/before.rules
GNU nano 7.2
+
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#
# Don't delete these required lines, otherwise there will be errors
*filter
:ufw-before-input - [0:0]
:ufw-before-output - [0:0]
:ufw-before-forward - [0:0]
:ufw-not-local - [0:0]
# End required lines
#
# Bloquer le ping (ICMP echo-request)
-A ufw-before-input -p icmp --icmp-type echo-request -j DROP
```

Avant d'activer le ufw on vérifie que l'on a autoriser les connexions SSH.

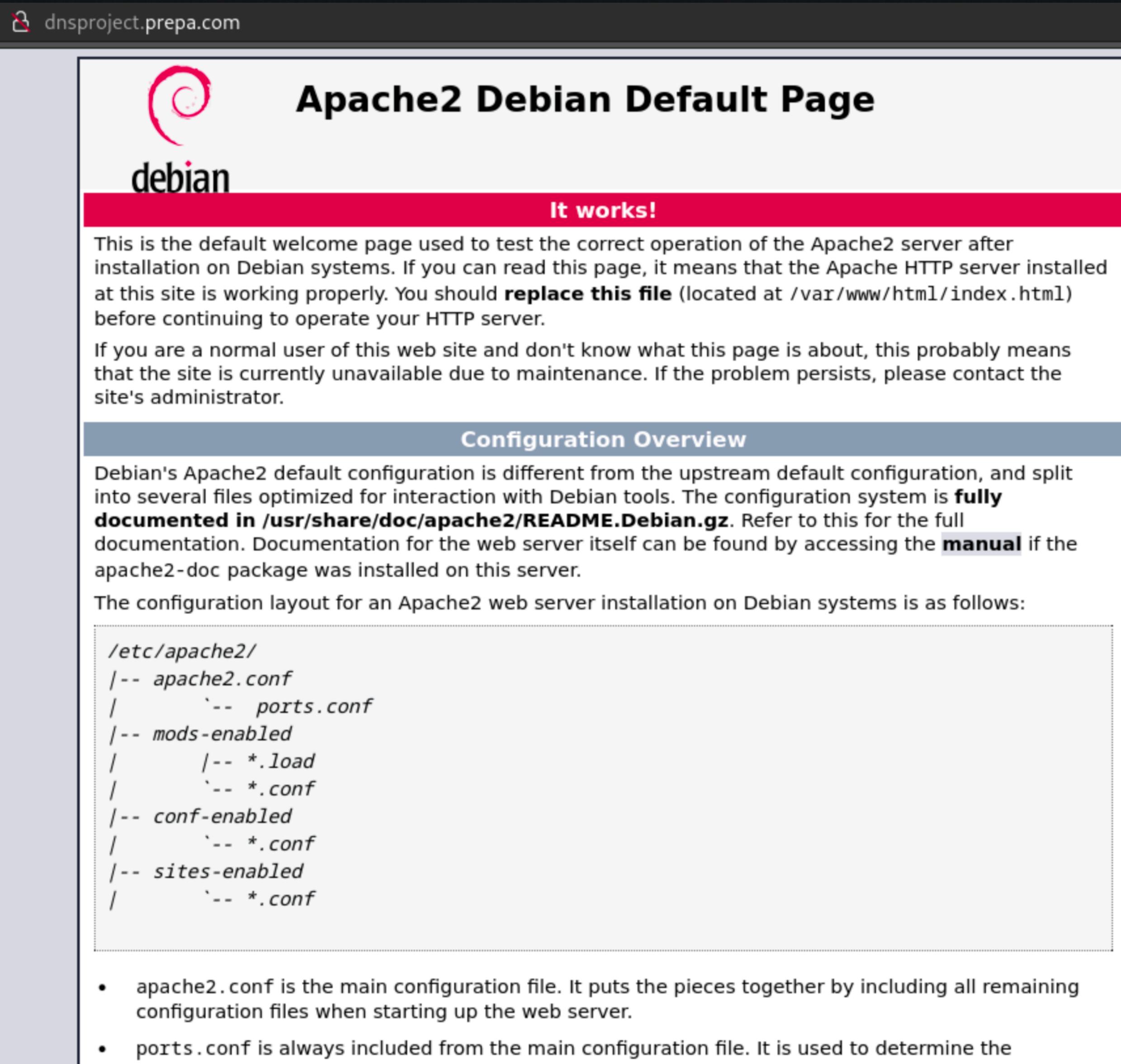
```
valuxadmi@debian:~$ sudo ufw allow ssh  
Rules updated  
Rules updated (v6)
```

Nous pouvons maintenant activer le ufw en entrant la commande « sudo ufw enable » et lancer un est de ping.

```
valuxadmi@debian:~$ ping dnsproject.prepa.com  
PING dnsproject.prepa.com (127.0.1.1) 56(84) bytes of data.  
^C  
--- dnsproject.prepa.com ping statistics ---  
8 packets transmitted, 0 received, 100% packet loss, time 7164ms  
  
valuxadmi@debian:~$ ping 176.0.0.1  
PING 176.0.0.1 (176.0.0.1) 56(84) bytes of data.  
^C  
--- 176.0.0.1 ping statistics ---  
3 packets transmitted, 0 received, 100% packet loss, time 2045ms
```



On voit ici que tous les packets envoyés sont perdus, on peut également continuer de se connecter à la page par défaut de Apache.



The screenshot shows a web browser displaying the Apache2 Debian Default Page. The URL in the address bar is `dnsproject.prepa.com`. The page itself has a red header with the text "Apache2 Debian Default Page" and the Debian logo. Below the header, a red banner contains the text "It works!". The main content area contains text explaining that this is the default welcome page for testing the Apache2 server on Debian systems. It also instructs users to replace the `/var/www/html/index.html` file if they want to change the content. A "Configuration Overview" section provides a tree view of the Apache2 configuration directory structure:

```
/etc/apache2/  
|-- apache2.conf  
|   '-- ports.conf  
|-- mods-enabled  
|   '-- *.load  
|   '-- *.conf  
|-- conf-enabled  
|   '-- *.conf  
|-- sites-enabled  
|   '-- *.conf
```

Below this, a list of bullet points explains the configuration files:

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

# Dossier partagé

## Membres réseau

- Pour mettre donner accès au dossier partagé il est nécessaire dans un premier temps d'installer samba en entrant la commande « sudo apt install samba sambaclient ».
- Il faut ensuite créer le dossier que l'on souhaite partager en utilisant mkdir. Et lui donner les permissions nécessaires.
- Nous pouvons ensuite ouvrir le fichier de configuration de Samba en entrant la commande : « sudo nano /etc/samba/smb.conf » et y ajouter les lignes représentant le dossier partagé.

```
[shared]
comment = partage de fichiers
path = /home/valuxadmi/shared
read only = no
valid users = valentinbsrt
write list = valentinbsrt
browseable = yes
```

Les mentions « Valid users » et « write list » serviront plus tard lorsque nous allons configurer les utilisateurs pouvant se connecter au serveur pour accéder au dossier partagé.

On pensera également à désactiver le pare-feu avec « sudo ufw disable » le temps de paramétrier la machine hôte pour faciliter la connexion.

Nous allons maintenant créer un utilisateur référence ayant toutes les autorisations sur le dossier partagé après s'être identifier sur la page de connexion.

On entre donc dans un premier temps la commande « sudo adduser valentinbsrt » (il est préférable de mettre le même nom d'utilisateur que celui de la machine que l'on veut connecter au serveur). Nous lui attribuons à présent à mot de passe avec la commande « sudo smbpasswd -a valentinbsrt »

La prochaine étape est de s'assurer que l'utilisateur à bien les permissions du dossier partagé en entrant les commandes suivantes :

« sudo chown -R monmacuser:monmacuser /path/vers/mon/dossier/partagé »

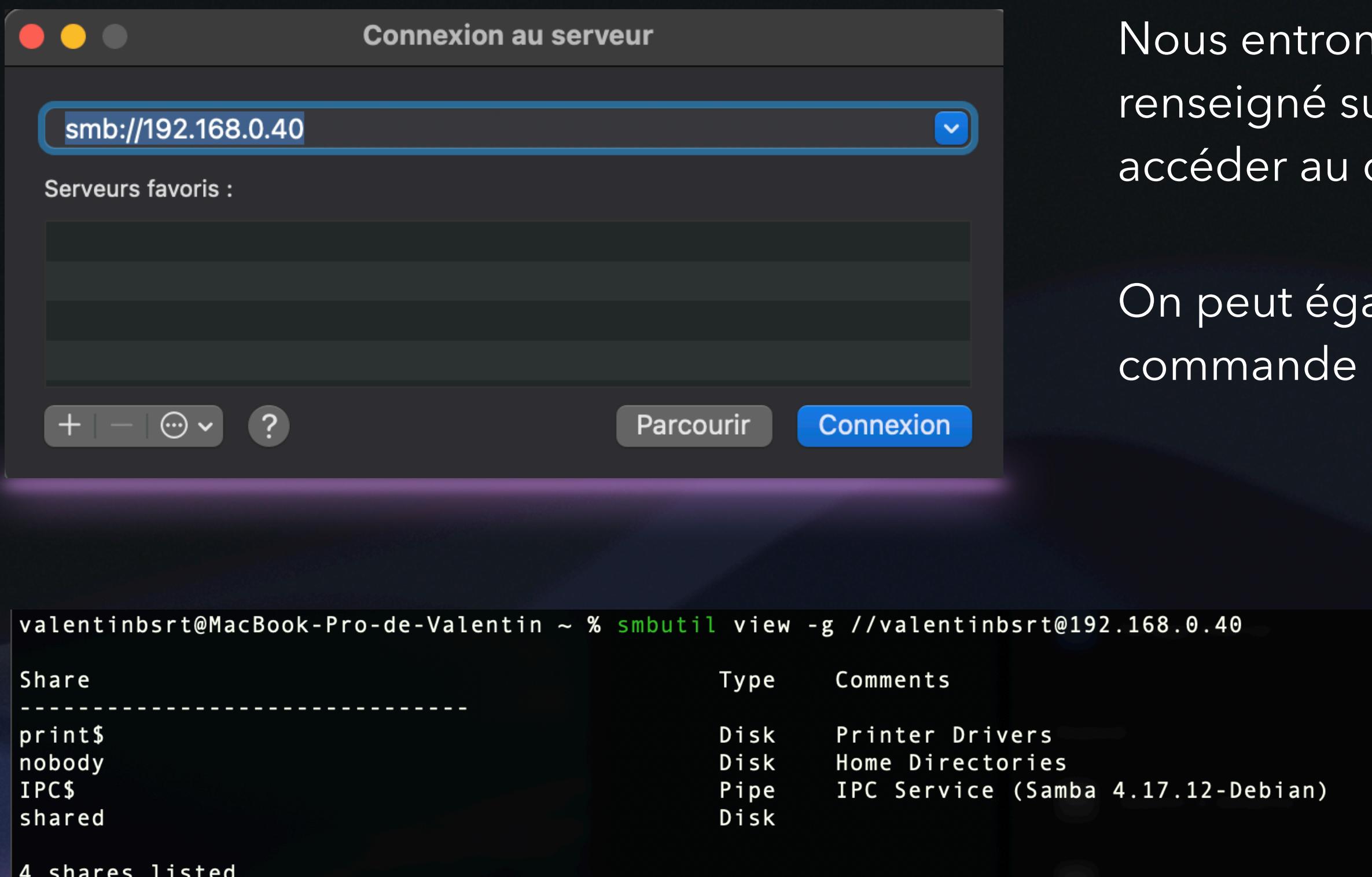
« sudo chmod -R 0755 /path/vers/mon/dossier/partagé »

Nous pouvons à présent redémarrer le service samba en entrant « sudo service smbd restart » afin d'appliquer les changements.

Etant sur MacOs les manipulations seront différentes que sur Windows.

Nous ouvrons dans un premier temps le Finder, puis nous allons sur l'onglet « Aller » puis « Se connecter au serveur... »

Pour pouvoir accéder au serveur nous allons écrire « smb://192.168.0.40 qui est l'adresse de notre serveur. Nous pourrions également ajouter le nom du dossier partagé.



Nous entrons le nom d'utilisateur et le mot de passe que nous avons renseigné sur Linux et nous nous devrions pouvoir nous connecter et accéder au dossier partagé.

On peut également vérifier que notre machine hôte est connecté via une commande dans le terminal :

```
valuxadmi@debian:~ $ systemctl status smbd
● smbd.service - Samba SMB Daemon
  Loaded: loaded (/lib/systemd/system/smbd.service; enabled; preset: enabled)
  Active: active (running) since Fri 2023-10-27 21:37:58 CEST; 1h 57min left
    Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
   Process: 1068 ExecCondition=/usr/share/samba/is-configured smb (code=exited, status=0/SUCCE
   Process: 1071 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, sta
 Main PID: 1081 (smbd)
   Status: "smbd: ready to serve connections..."
      Tasks: 4 (limit: 7579)
     Memory: 13.5M
        CPU: 156ms
      CGroup: /system.slice/smbd.service
              └─1081 /usr/sbin/smbd --foreground --no-process-group
                  ├─1089 /usr/sbin/smbd --foreground --no-process-group
                  ├─1090 /usr/sbin/smbd --foreground --no-process-group
                  ├─2433 /usr/sbin/smbd --foreground --no-process-group
```

Fin