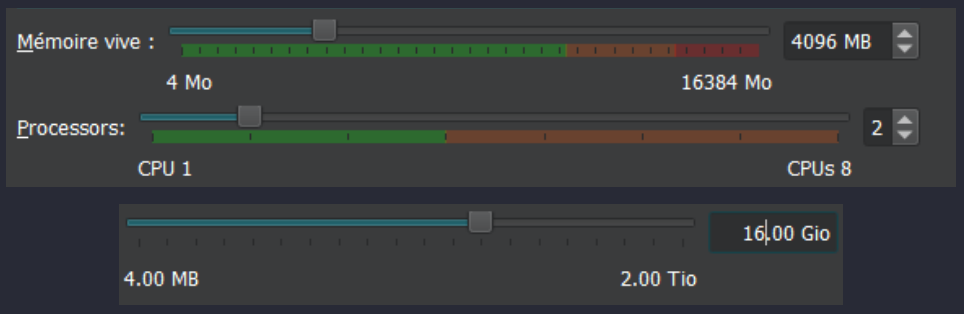
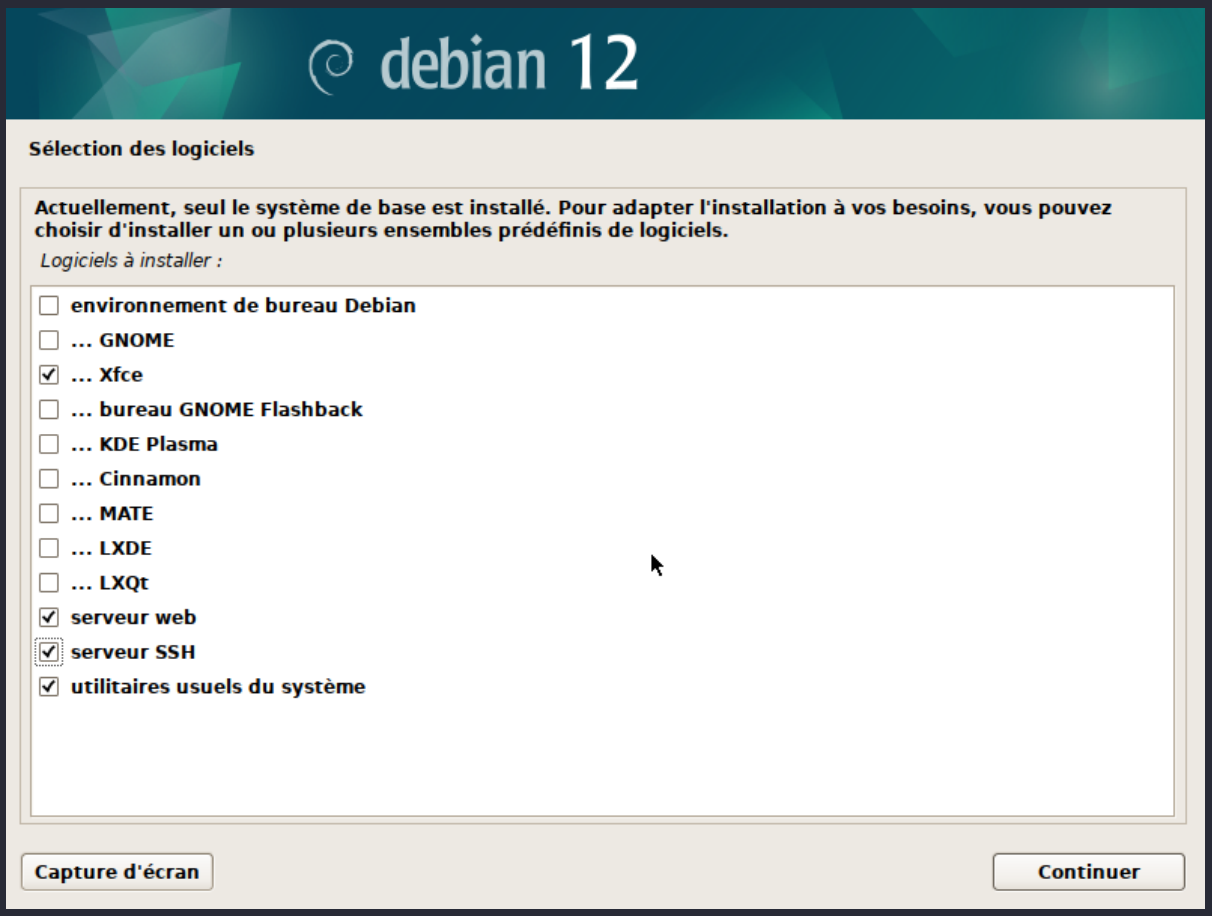
## Configuration et installation VM Debian

Puisque nous n'allons pas utiliser beaucoup d'applications et logiciels graphiques dans la VM, nous pouvons nous contenter de 4 Go de RAM et 2 processeurs virtuels.
Pour le disque dur virtuel, 16 Gio suffira largement.
Nous allons aussi nous mettre en mode NAT, afin qu'on puisse installer Debian et quelques paquets.




J'utilise Virtualbox, mais c'est le même processus que sur VMWare.

La seule chose particulière que nous allons faire durant l'installation de Debian, c'est que sur l'écran de "sélection des logiciels", nous allons sélectionner "serveur web", "serveur ssh", "utilitaires usuels" et un environnement de bureau, je recommande XFCE qui est léger mais qui possèdent quand même les fonctionnalités d'un environnement de bureau moderne.



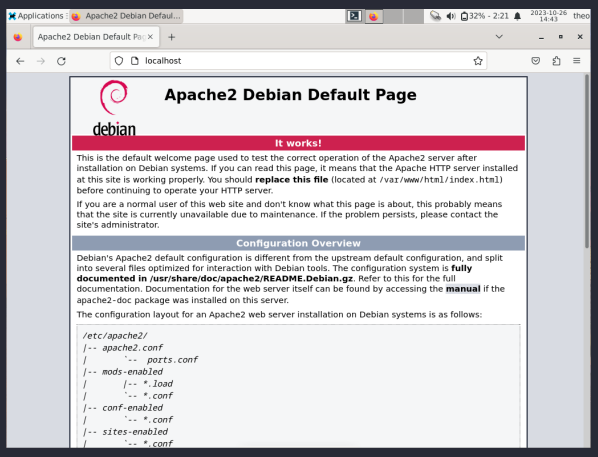
## Logiciels de serveur web et Apache HTTP

Il existe une multitude de logiciels de serveur web, chacun avec ses avantages et inconvénients, voici les trois plus utilisés:

Apache est un logiciel de serveur web libre disponible sur Windows, Linux et MacOS, sorti en 1995, c'était le plus utilisé durant plusieurs années, donc on peut dire qu'il y a une très grande communauté autour de ce logiciel qui aide à le maintenir et le documenter, l'inconvénient est la méthode de gestion de connexion, c'est-à-dire un processus pour une connexion, ce qui peut-être inefficace avec des sites plus large.

Nginx est également un logiciel libre disponible sur Windows Linux et MacOS qui devient de plus en plus populaire récemment, ce logiciel fût conçu afin de répondre au problème d'Apache en gérant les connexions de façon plus efficace, cependant Nginx ne peut pas gérer les contenus dynamiques par lui-même, et dépend de programmes externes.

Microsoft IIS est un logiciel propriétaire qui n'est que disponible pour Windows, cependant puisque c'est conçu spécifiquement pour ce système d'exploitation, ça s'intègre très bien dans un serveur Windows.




Nous utiliserons Apache, car c'est ce qui est installé lorsque nous avons sélectionné "serveur Web" durant l'installation.
Par défaut, Apache créée une page qu'on peut accéder en allant dans un navigateur et en tapant dans la barre d'adresse "localhost".

## Installation de paquets requis et config réseau.

Nous aurons besoin de ces paquets pour les prochaines étapes car on ne pourra plus se connecter à Internet.

```
theo@debian-ddws:~$ sudo apt install bind9 bind9utils dnsutils ufw samba isc-dhcp-server
```

Maintenant, nous allons nous mettre en réseau Bridge dans notre hyperviseur, ensuite nous allons redémarrer networking et NetworkManager.

```
theo@debian-ddws:~$ sudo systemctl restart networking
theo@debian-ddws:~$ sudo systemctl restart NetworkManager
```

Nous avons besoin aussi de notre adresse IP.

```
theo@debian-ddws:~$ hostname -I
10.10.8.89
```

Ça pourrait afficher deux adresses IP, juste au cas où, ça peut être une bonne idée de noter l'adresse avant de changer de mode de réseau.

## Configuration DNS avec BIND

Nous allons travailler dans '/etc/bind'

```
theo@debian-ddws:/etc/bind$ ls
total 56K
drwxr-sr-x   2 root bind 4,0K 26 oct.  09:52 .
drwxr-xr-x 121 root root 4,0K 26 oct.  10:15 ..
-rw-r--r--   1 root root 2,4K 21 sept. 19:33 bind.keys
-rw-r--r--   1 root root  255 21 sept. 19:33 db.0
-rw-r--r--   1 root root  271 21 sept. 19:33 db.127
-rw-r--r--   1 root root  237 21 sept. 19:33 db.255
-rw-r--r--   1 root root  353 21 sept. 19:33 db.empty
-rw-r--r--   1 root root  270 21 sept. 19:33 db.local
-rw-r--r--   1 root bind  458 21 sept. 19:33 named.conf
-rw-r--r--   1 root bind  498 21 sept. 19:33 named.conf.default-zones
-rw-r--r--   1 root bind  165 21 sept. 19:33 named.conf.local
-rw-r--r--   1 root bind  846 21 sept. 19:33 named.conf.options
-rw-r-----   1 bind bind  100 26 oct.  09:52 rndc.key
-rw-r--r--   1 root root 1,3K 21 sept. 19:33 zones.rfc1918
```

Il ne faudra pas oublier de faire `sudo` pour quasiment tout, puisqu'on est en dehors de notre '/home'

Afin de configurer le DNS, nous allons en premier prendre 'db.local' comme modèle, afin de créer deux fichiers de zone, 'db.prepa.com' pour associer un nom de domaine à une adresse, et 'db.prepa.com.inv 'pour associer une adresse à un nom de domaine

```
theo@debian-ddws:/etc/bind$ cp db.local db.prepa.com

theo@debian-ddws:/etc/bind$ cp db.local db.prepa.com.inv
```

Un fichier de zone consiste de plusieurs paramètres, tels que:

En premier, “$TTL” (Time To Live) qui permet de définir, en secondes, la durée de mise en cache des paramètres DNS avant une mise à jour automatique.
En général, c’est mieux de mettre une durée courte si nous comptons faire des modifications.
Dans notre cas, ce n’est pas important.

Ensuite, un tableau SOA (Start of Authority):

Le premier élément est la classe du tableau, dans ce cas-là IN qui veut dire Internet, il y a d'autres classes comme CH pour ChaosNet mais IN est quasiment la seule qui est utilisée de nos jours.

Les deux prochains arguments sont les FQDN (Fully Qualified Domain Name), qui spécifie où se trouve un nom dans la hiérarchie DNS, le premier désigne le nom de domaine du serveur.

Les autres arguments n’affectent que les serveurs DNS secondaires.

Enfin:

“NS” qui attribue l’hostname.
“A” qui associe un nom de domaine à une adresse IP
“PTR” qui associe une adresse IP à un nom de domaine.

Nous configurons ces deux fichiers comme ceci:

```
; db.prepa.com
$TTL    604800
@       IN      SOA     prepa.com. dnsproject.prepa.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      dnsproject.prepa.com.
dnsproject      IN      A       10.10.8.89
```

```
; db.prepa.com.inv
$TTL    604800
@       IN      SOA     prepa.com. dnsproject.prepa.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      dnsproject.prepa.com.
dnsproject      IN      A       10.10.8.89
89      IN      PTR     dnsproject.prepa.com.
```

Nous allons dans ‘named.conf.local’ et nous mettons ceci:

```
zone "prepa.com" IN {
        type master;
        file"/etc/bind/db.dnsproject.prepa.com";
};
zone "2.8.10.in-addr-arpa" IN {
        type master;
        file"/etc/bind/db.dnsproject.prepa.com.inv";
}
```

Puis nous allons dans '/etc/resolv.conf', et nous remplaçons le contenu avec:

```
search prepa.com
nameserver 10.10.8.2
```

Pour que ce changement soit permanent, nous créons le fichier '/etc/NetworkManager/conf.d/90-dns-none.conf', et on écrit ceci à l'intérieur.

```
[main]
dns=none
```

On fait ceci pour empêcher NetworkManager de reconfigurer à chaque fois l'adresse du DNS.

Nous démarrons le service bind9

```
theo@debian-ddws:/etc/bind$ sudo systemctl restart bind9
```

Ensuite, on utilise dig pour voir le status du DNS.
Nous pouvons maintenant aller dans le navigateur, taper le nom du domaine, et nous allons obtenir la page Apache

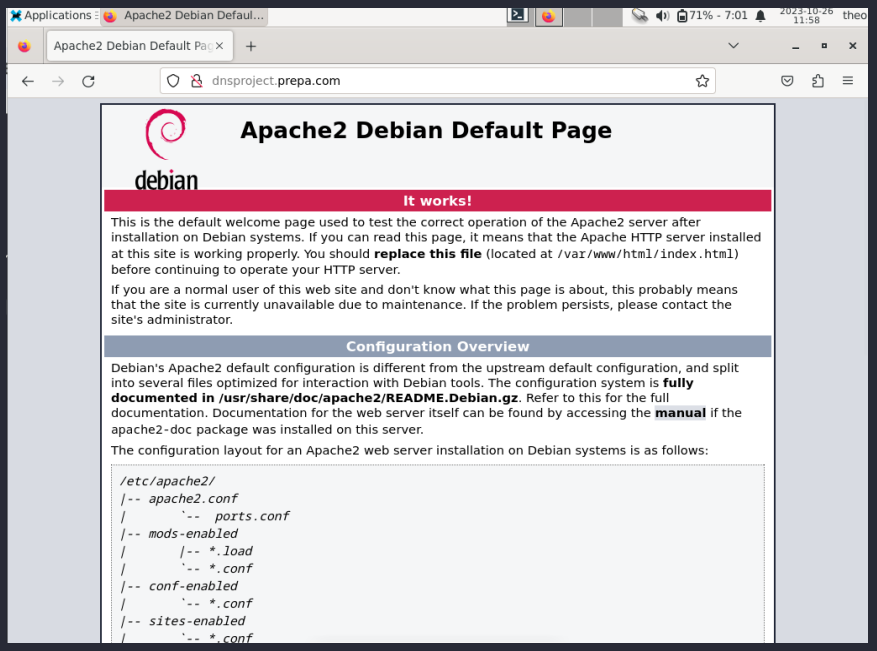
```
theo@debian-ddws:/etc/bind$ dig prepa.com

; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> prepa.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28650
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 4efdad2722d159a601000000653b933c9a594b20ae60b511 (good)
;; QUESTION SECTION:
;prepa.com.                     IN      A

;; AUTHORITY SECTION:
prepa.com.              604800  IN      SOA     prepa.com. dnsproject.prepa.com.
 2 604800 86400 2419200 604800

;; Query time: 4 msec
;; SERVER: 10.10.8.89#53(10.10.8.89) (UDP)
;; WHEN: Fri Oct 27 12:38:52 CEST 2023
;; MSG SIZE  rcvd: 113
```



## Les nom de domaine publique

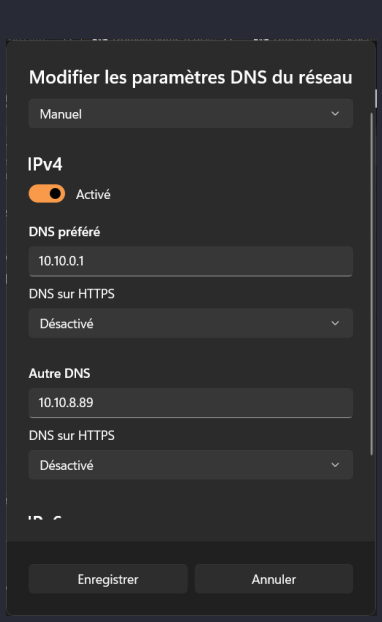
Afin d'obtenir un nom de domaine public, il faut la réserver auprès d'un registrar, une société qui gère les noms de domaine.

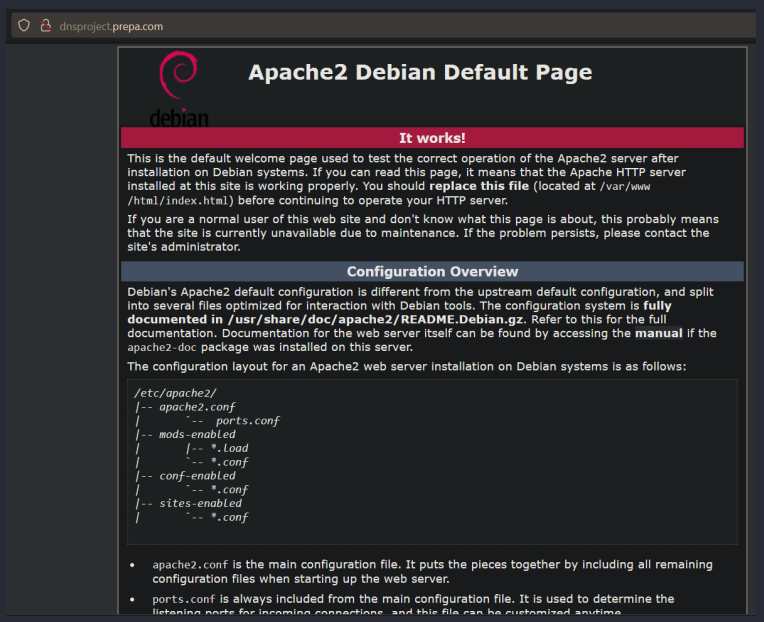
Les extensions de nom de domaine qui peuvent être disponibles dépendent des circonstances de celui qui souhaite en réserver.
Il y a plusieurs types d'extensions, voici les trois types les plus connus.

| | |
|---|---|
| gTLD<br>(Extension générique) | .com, .net, .org |
| grTLD<br>(Extension générique restreinte) | .edu, .gov, .mil |
| ccTLD<br>(Extension de pays) | .fr, .uk, .jp |

## Accéder à la page Apache à partir de la machine hôte

Il faut aller dans les paramètres du réseau où est connecté la machine, et régler manuellement les DNS, il ne faut pas oublier l'adresse DNS principale sinon on risque d'avoir beaucoup de mal à naviguer sur internet.



## ufw, **u**ncomplicated **f**ire**w**all

ufw est un programme qui permet de gérer les pare-feu.

```
theo@debian-ddws:/etc/bind$ sudo ufw status
Status: inactive
theo@debian-ddws:/etc/bind$ sudo ufw enable
Firewall is active and enabled on system startup
theo@debian-ddws:/etc/bind$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
```

On vérifie le statut d'ufw, ensuite on l'active, puis on vérifie encore son statut, cette fois en détail.

Ensuite, on va refuser tous les trafics d'entrée et de sortie.

```
theo@debian-ddws:~$ sudo ufw default deny outgoing
Default outgoing policy changed to 'deny'
(be sure to update your rules accordingly)
theo@debian-ddws:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
```

```
theo@debian-ddws:~$ sudo ufw allow 80/tcp
Rule added
Rule added (v6)
theo@debian-ddws:~$ sudo ufw allow samba
Rule added
Rule added (v6)
```

Maintenant, nous allons créer des exceptions à cette règle.
On autorise tout trafic qui concerne Apache et pour samba.

Puis, on va dans les fichiers de configuration d'ufw car on ne peut pas exclure directement les pings via le command-line.

```
-A ufw-before-output -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-forward -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

# drop INVALID packets (logs these in loglevel medium and higher)
-A ufw-before-input -m conntrack --ctstate INVALID -j ufw-logging-deny
-A ufw-before-input -m conntrack --ctstate INVALID -j DROP

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

# ok icmp code for FORWARD
-A ufw-before-forward -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type echo-request -j ACCEPT

# allow dhcp client to work
-A ufw-before-input -p udp --sport 67 --dport 68 -j ACCEPT

#
4 substitutions sur 4 lignes                                33,1
```

```
-A ufw-before-output -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-forward -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

# drop INVALID packets (logs these in loglevel medium and higher)
-A ufw-before-input -m conntrack --ctstate INVALID -j ufw-logging-deny
-A ufw-before-input -m conntrack --ctstate INVALID -j DROP

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

# ok icmp code for FORWARD
-A ufw-before-forward -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type echo-request -j ACCEPT

# allow dhcp client to work
-A ufw-before-input -p udp --sport 67 --dport 68 -j ACCEPT

#
"/etc/ufw/before.rules" 76L, 2530B écrit(s)                 33,1          47%
```

On va dans /etc/ufw/before.rules, On trouve la section "ok icmp codes for INPUT" et on remplace "ACCEPT" avec "DROP"

On vérifie les règles qu'on ajouté, et on redémarre ufw.

```
theo@debian-ddws:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), deny (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW IN    Anywhere
137,138/udp (Samba)        ALLOW IN    Anywhere
139,445/tcp (Samba)        ALLOW IN    Anywhere
80/tcp (v6)                ALLOW IN    Anywhere (v6)
137,138/udp (Samba (v6))   ALLOW IN    Anywhere (v6)
139,445/tcp (Samba (v6))   ALLOW IN    Anywhere (v6)

theo@debian-ddws:~$ sudo ufw reload
Firewall reloaded
```

## Samba et dossiers partagés

Parlé de configuration compte

`binary/service`
'/directory'
"Misc. Highlight"