

Job 1

Installation de la vm Debian: ne pas oublier, lors de l'installation des logiciels, de bien cocher les cases "serveur SSH" et "serveur WEB" qui pourront nous servir par la suite.

Job 2

Pour ce job nous devons installer un serveur web apache 2, pour ce faire nous allons entrer la commande suivante.

```
root@debian:~# apt install apache2
```

Notre serveur est installé. Nous devons ensuite connaître notre adresse IP. Pour ce faire nous allons entrer la commande: ip a dans le terminal. La suivante va s'afficher:

```
192.168.31.128/24
```

Enfin nous allons ouvrir une nouvelle page internet et dans la barre de recherche web nous allons y ajouter l'adresse ip trouvée ci dessus : 192.168.31.128 ce qui va nous afficher à l'écran la page suivante!



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was **installed** on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

Job 3

Il existe de nombreux serveurs parmi les plus connus que nous pouvons citer, avec chacun leurs avantages et leurs inconvénients.

Apache HTTP Server :

- *Avantages :*
 - Très populaire et largement utilisé.
 - Dispose d'une vaste communauté d'utilisateurs et de développeurs.
 - Peut être configuré pour répondre à un large éventail de besoins.
- *Inconvénients :*
 - Peut être gourmand en ressources dans certaines configurations.
 - La configuration peut être complexe pour les débutants.

Nginx :

- *Avantages :*
 - Conçu pour être léger et efficace en termes de ressources.
 - Excellente gestion de la concurrence et de la mise en cache.
 - Idéal pour servir des fichiers statiques ou en tant que proxy inverse.
- *Inconvénients :*
 - Peut être moins flexible pour le traitement de requêtes complexes comparé à Apache.

Microsoft Internet Information Services (IIS) :

- *Avantages :*
 - Intégré à Windows Server, ce qui facilite l'administration pour les utilisateurs de cette plateforme.
 - Excellente intégration avec les technologies Microsoft, telles que ASP.NET.
- *Inconvénients :*
 - Moins courant sur les systèmes non-Windows.
 - Peut nécessiter plus de ressources que certains concurrents.

LiteSpeed :

- *Avantages :*
 - Connu pour ses performances élevées et sa faible utilisation de ressources.
 - Supporte de nombreuses technologies modernes, telles que HTTP/2 et QUIC.
 - Propose un accélérateur de cache intégré.

- *Inconvénients:*
 - La version gratuite a des limitations par rapport à la version payante.

Job 4

Pour le job 4, il est demandé d'installer un DNS.

Etape 1

Pour se faire nous allons ouvrir le terminal de notre vm et y rentrer la commande suivante : `apt -y install bind9 bind9utils dnsutils`

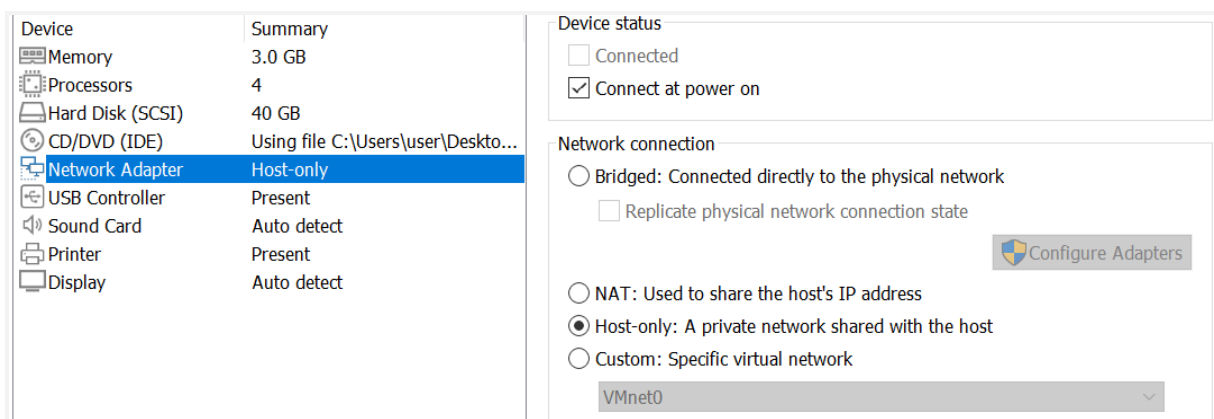
```
root@debian:~# apt -y install bind9 bind9utils dnsutils
```

BIND9 est le serveur de noms de domaine (DNS) le plus largement utilisé sur Internet. Il associe les noms de domaine (par exemple, `www.example.com`) aux adresses IP correspondantes (par exemple, `192.168.1.1`).

Etape 2

Ensuite il va falloir changer le mode de réseau de notre vm debian: nous allons le paramétrer en mode host only

Les changements se font via le menu de démarrage de vmware clic droit → settings → network adapter → cocher host only



Etape 3

Ensuite nous allons nous déplacer dans le fichier de configuration de Bind9 `cd /etc/bind` et l'éditer.

```
root@debian:~# cd /etc/bind
root@debian:/etc/bind# nano named.conf.local
```

Voilà ce que nous allons lui ajouter :

```
GNU nano 5.4 named.conf.local
zone "dnsproject.prepa.com" {
    type master;
    file "/etc/bind/zones/db.dnsproject.prepa.com";
};
```

Etape 4

Nous allons maintenant modifier les fichiers de configuration pour associer l'adresse ip au nom de serveur dnsproject ainsi qu'au nom de domaine prepa.com.

Nous allons ouvrir le fichier `db.local` et l'éditer → `nano db.local`

voilà ce qui s'affiche :

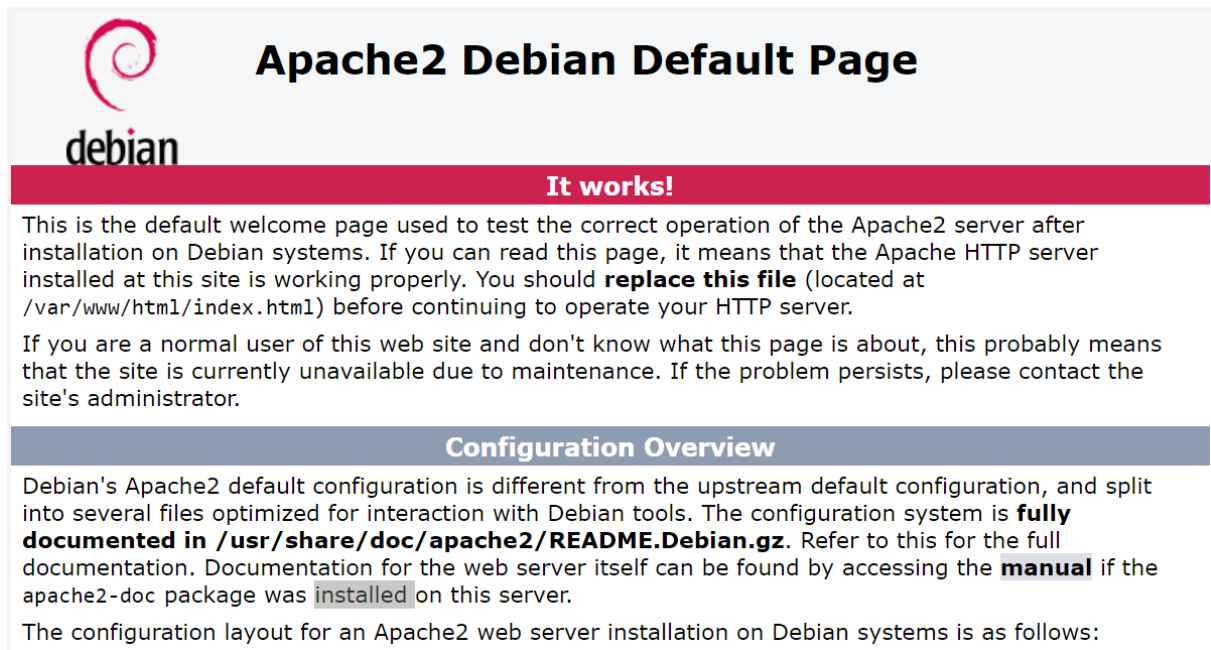
```
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                        2      ; Serial
                        604800  ; Refresh
                        86400   ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       localhost.
@         IN      A        127.0.0.1
@         IN      AAAA     ::1
```

A ce fichier nous allons y ajouter quelques lignes supplémentaires :

```
@         IN      NS       dnsproject.prepa.com
dnsproject IN      A        192.168.31.128
www        IN      CNAME    dnsproject.prepa.com
```

Etape 5

Lorsque nous entrons notre adresse ip dans la barre de recherche web va s'afficher :



The screenshot shows the Apache2 Debian Default Page. At the top left is the Debian logo. The title is "Apache2 Debian Default Page". Below the title is a red banner with the text "It works!". The main content area contains a paragraph explaining that this is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. It states that if you can read this page, it means that the Apache HTTP server is installed at this site and is working properly. It advises to replace the file located at /var/www/html/index.html before continuing to operate your HTTP server. Below this is another paragraph stating that if you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. It suggests contacting the site's administrator if the problem persists. A section titled "Configuration Overview" follows, explaining that Debian's Apache2 default configuration is different from the upstream default configuration and is split into several files optimized for interaction with Debian tools. It mentions that the configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz and refers to this for the full documentation. It also notes that documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server. The final paragraph states that the configuration layout for an Apache2 web server installation on Debian systems is as follows:

etape 1 installer bind9

```
erika@debian:~$ sudo apt install bind9
[sudo] Mot de passe de erika :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
```

etape 2 configurer bind avec la commande sudo nano /etc/bind/named.conf.local et y ajouter :

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

enregistrer et fermer.

etape 3 créer le dossier zones : mkdir zones

se déplacer à l'intérieur de zones et créer le fichier dnsproject.prepa.com avec la commande sudo nano /etc/bind/zones/db.dnsproject.prepa.com dans lequel nous allons éditer la zone de texte suivante :

```
$TTL      604800
@         IN      SOA      dnsproject.prepa.com. admin.dnsproject.prepa.com. (
        2023102501 ; Serial
        604800    ; Refresh
        86400     ; Retry
        2419200   ; Expire
        604800 )   ; Negative Cache TTL

@         IN      NS       dnsproject.prepa.com.
dnsproject.prepa.com. IN A 192.168.214.129
```

étape 4 pour appliquer les changements nous allons entrer la commande `sudo systemctl restart bind9`

```
erika@debian:~$ sudo systemctl start bind9
```

étape 5 nous pouvons enfin ping notre nom de domaine

```
erika@debian:~$ ping dnsproject.prepa.com
PING dnsproject.prepa.com (192.168.214.129) 56(84) bytes of data.
64 bytes from debian (192.168.214.129): icmp_seq=1 ttl=64 time=0.011 ms
64 bytes from debian (192.168.214.129): icmp_seq=2 ttl=64 time=0.066 ms
64 bytes from debian (192.168.214.129): icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from debian (192.168.214.129): icmp_seq=4 ttl=64 time=0.063 ms
64 bytes from debian (192.168.214.129): icmp_seq=5 ttl=64 time=0.035 ms
```

Job 5

Faites des recherches sur comment obtient-on un nom de domaine public ?

Tout d'abord il faut choisir un bureau d'enregistrement, ensuite il faut vérifier la disponibilité et sélectionner un nom de domaine. Ensuite il est nécessaire de fournir des informations tels que l'email, le nom etc.. On sélectionne la durée d'enregistrement puis on passe au paiement. Enfin il faudra configurer les serveurs de noms pour rediriger le trafic vers notre site puis attendre l'enregistrement pour que le nom de domaine soit enregistré et actif sur internet.

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

Chaque TLD peut avoir ses propres spécificités en termes d'éligibilité, d'utilisation et de règles de domaine. Voici quelques-unes des

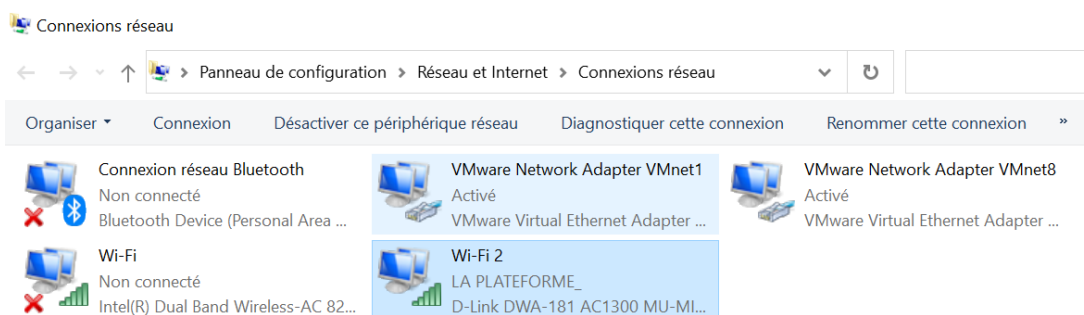
spécificités que vous pouvez rencontrer sur certaines extensions de nom de domaine :

- TLD Génériques : .com ; .org ; .net
- TLD Géographique : .us ; .uk ; .ca
- TLD Génériques Restreints : .gov ; .mil ; .int

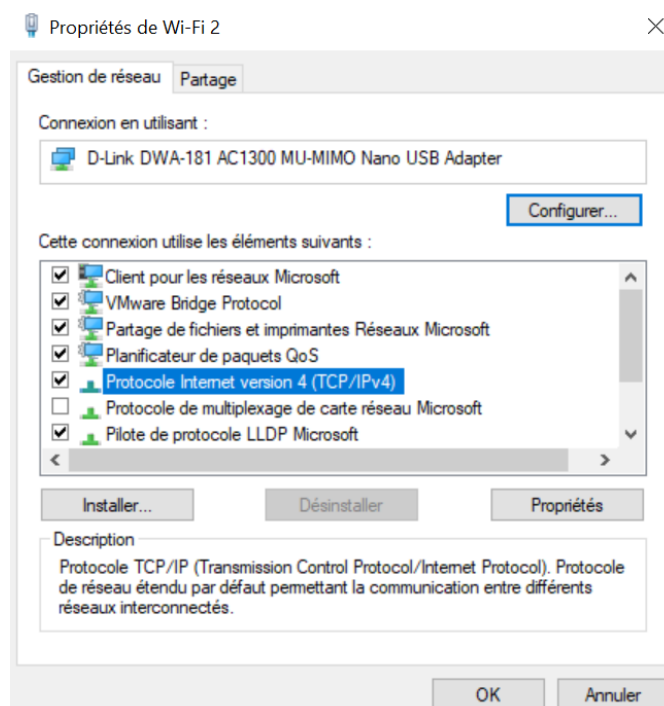
Job 6

Pour ce job il nous est demandé de connecter l' hôte au nom de domaine local de votre serveur, pour que la page apache soit accessible via ce même nom de domaine soit `dnsproject.prepa.com`

Etape 1 : Nous allons dans les paramètres de notre réseau sur la machine hôte



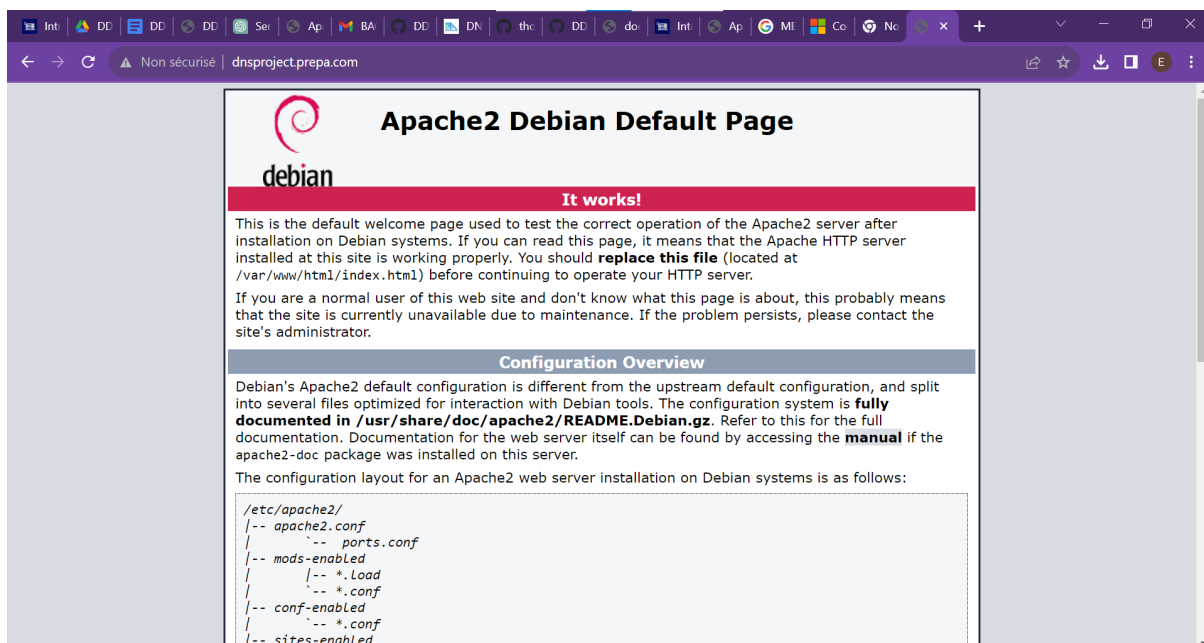
Etape 2 : Il faut ensuite aller dans les paramètres du wifi et configurer le TCP/IPv4





Etape 3 : Nous allons entrer l'adresse IP de notre vm pour activer le dns.

Le nom de domaine est maintenant accessible sur notre machine hôte.



Job 7

Pour ce job nous devons installer un pare-feu ufw de notre serveur apache sur la vm de manière que la machine hôte puisse accéder à la page apache, mais qu'il ne puisse plus ping.

Etape 1 : installation de UFW avec la commande `sudo apt-get install ufw`

```
erika@debian:~$ sudo apt-get install ufw
[sudo] Mot de passe de erika :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  ufw
```

Etape 2 : Avec la commande `sudo ufw default deny incoming` nous avons passé toutes les connexions entrantes en mode refus.

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

Etape 3 : Nous allons en faire de même pour les connexions sortantes

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

Etape 4 : Avec les commandes `sudo ufw allow 80/tcp` et `sudo ufw allow 443/tcp` j'autorise les ports 80 http et 443 https à échanger.

```
erika@debian:~$ sudo ufw allow 80/tcp
Rule added
Rule added (v6)
erika@debian:~$ sudo ufw allow 443/tcp
Rule added
Rule added (v6)
erika@debian:~$
```

Etape 5 : Ensuite je vais aller modifier le fichier `before.rules` pour DROP le protocole icmp qui correspond au ping .

```
# 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 ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-request -j DROP
```

Etape 6 : Enfin je vais rentrer les commandes `sudo ufw disable` et `sudo ufw enable` pour activer le pare-feu.

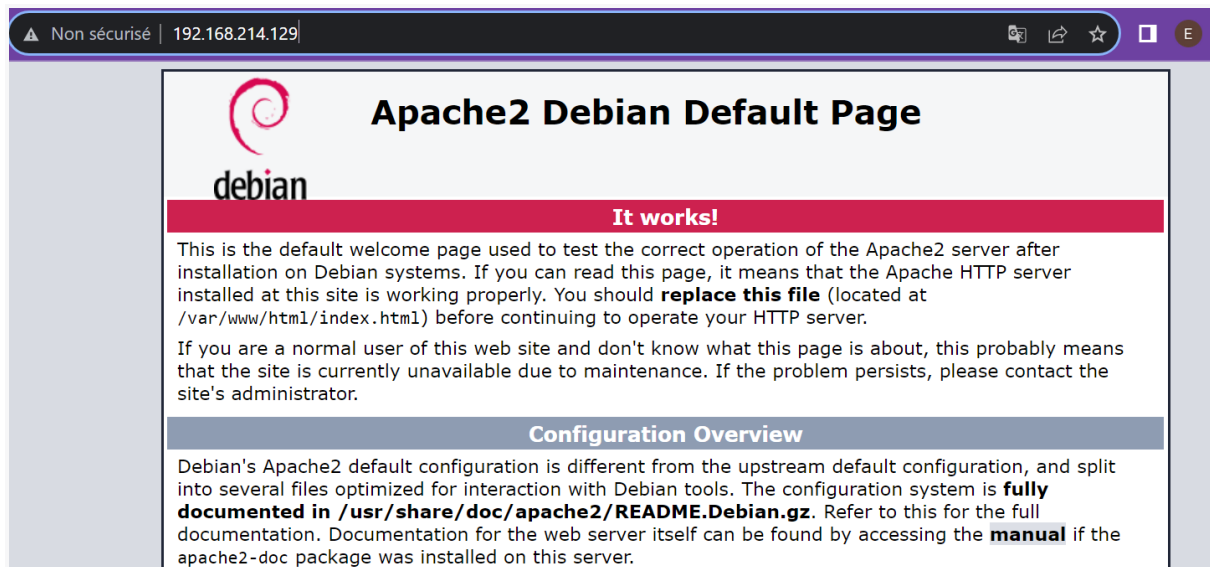
```
erika@debian:/etc/ufw$ sudo ufw disable
Firewall stopped and disabled on system startup
erika@debian:/etc/ufw$ sudo ufw enable
Firewall is active and enabled on system startup
erika@debian:/etc/ufw$
```

Nous allons maintenant tester le ping, celui ci est desactivé.

```
Envoi d'une requête 'Ping' 192.168.214.129 avec 32 octets de données :
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.
```

```
Statistiques Ping pour 192.168.214.129:
  Paquets : envoyés = 4, reçus = 0, perdus = 4 (perte 100%),
```

Et notre serveur reste accessible depuis notre machine hôte.



Job 8

Etape 1 : Pour ce job nous allons tout d'abord installer samba avec la commande `sudo apt-get install samba`

```
erika@debian:~$ sudo apt-get install samba
[sudo] Mot de passe de erika :
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 :
  attr ibverbs-providers libcephfs2 libgfs2 libgfrpc0 lib
  python3-dnspython python3-gpg python3-markdown python3-py
  samba-dsdb-modules samba-vfs-modules tdb-tools
```

Ne pas oublier de désactiver le pare-feu avec la commande `sudo ufw disable` pour cet exercice.

Avant de passer à la suite, nous allons activer le démarrage automatique de samba : `systemctl enable smb`

```
erika@debian:~$ sudo systemctl enable smb
Synchronizing state of smb.service with SysV service script with /lib/systemd/s
ystemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable smb
```

Etape 2 : Maintenant nous allons créer le dossier partage : `sudo mkdir partage`

Etape 3 : Maintenant nous allons éditer le fichier de configuration de samba en faisant : `sudo nano /etc/samba/smb.conf` puis nous allons y ajouter ces lignes pour indiquer notre partage.

```
[partage]
comment = Partage de données
path = /srv/partage
guest ok = no
read only = no
browseable = yes
valid users = @partage
```

Etape 4 : sauvegardez et redémarrez samba avec la commande : `systemctl restart samba`

```
erika@debian:~$ systemctl restart smb
```

Etape 5 : Nous allons maintenant créer un utilisateur Samba : `sudo smbpasswd -a erika` et lui attribuer un mot de passe.

```
erika@debian:~$ sudo smbpasswd -a erika
New SMB password:
Retype new SMB password:
Added user erika.
erika@debian:~$
```

Etape 6 : nous allons maintenant donner tous les droits à ce dossier

```
erika@debian:/home$ sudo chmod -R 777 /home/partage
```

Une fois cela fait, l'utilisateur que nous avons créé précédemment devrait pouvoir accéder au dossier partage avec les autorisations appropriées.

