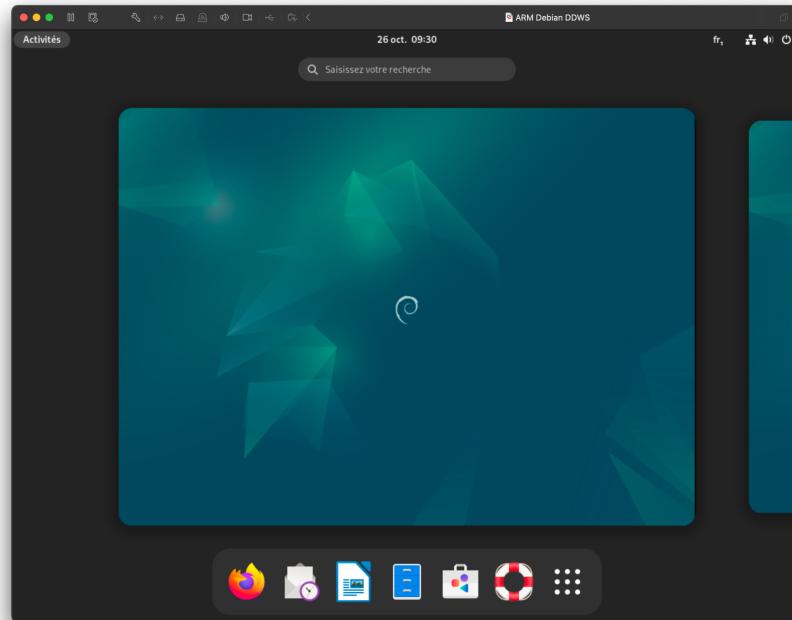


DDWS

Job 01 : Installation de la VM.



Job 02 : Installation du serveur Web Apache2.

Commande : `sudo apt update && sudo apt install apache2`

Interface VM :

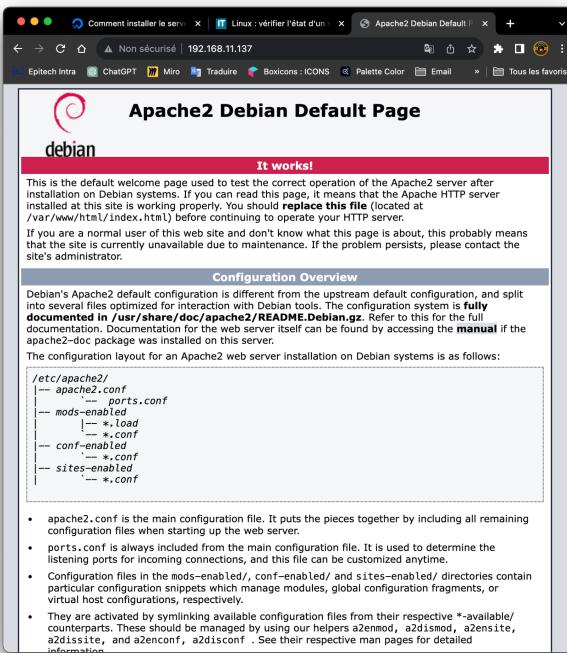
The screenshot shows a Firefox browser window displaying the Apache2 Debian Default Page. The page title is "Apache2 Debian Default Page" and it features a "It works!" message. Below this, there is a detailed configuration overview and a file tree diagram of the Apache2 configuration directory structure:

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

Below the file tree, there is a bulleted list of configuration details:

- `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.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain

Interface Hôte :



Job 03 : Les différents serveurs Web existants.

Il existe de nombreux serveurs web disponibles, chacun ayant ses propres avantages et inconvénients. Voici une liste de certains des serveurs web les plus couramment utilisés, ainsi que leurs caractéristiques principales :

1. Apache HTTP Server :

- Avantages :

- Très populaire et largement utilisé.
- Excellente documentation et une grande communauté de support.
- Possibilité d'extension via des modules tiers.
- Multiplateforme (compatible avec de nombreux systèmes d'exploitation).

- Inconvénients :

- Peut être moins performant que certains concurrents dans certaines situations.
- La configuration peut sembler complexe pour les débutants.

2. Nginx :

- Avantages :

- Extrêmement performant, surtout pour le traitement de nombreuses requêtes simultanées.
- Utilisation efficace des ressources système.
- Peut servir de proxy inverse pour équilibrer la charge.

- Inconvénients :

- La configuration peut sembler plus complexe que celle d'Apache pour les novices.
- Les modules tiers sont moins nombreux que pour Apache.

3. Microsoft Internet Information Services (IIS) :

- Avantages :

- Intégré à Windows Server, ce qui le rend facile à déployer pour les environnements Windows.
- Prise en charge des technologies Microsoft, telles que .NET.
- Interface utilisateur graphique conviviale pour la configuration.

- Inconvénients :

- Principalement adapté aux environnements Windows, ce qui limite sa portabilité.
- Peut être moins performant que certains serveurs web open source.

4. LiteSpeed Web Server :

- Avantages :

- Très performant et efficace en matière de gestion des ressources.
- Possède un équilibrage de charge intégré.
- Compatible avec Apache, ce qui facilite la migration.

- Inconvénients :

- La version gratuite a des fonctionnalités limitées par rapport à la version payante.
- Moins répandu que certains concurrents.

5. Caddy :

- Avantages :

- Facile à configurer avec une interface utilisateur simple.
- Prend en charge automatiquement le chiffrement HTTPS.
- Génère automatiquement des fichiers de configuration.

- Inconvénients :

- Peut ne pas être aussi performant que Nginx ou Apache dans certaines situations.
- Certains utilisateurs peuvent préférer une configuration manuelle.

6. Lighttpd :

- Avantages :

- Léger et conçu pour une utilisation efficace des ressources.
- Bon pour les serveurs intégrés, les objets connectés, etc.
- Possède une architecture modulaire.

- Inconvénients :

- Peut ne pas être aussi polyvalent que les serveurs web plus populaires.
- Moins de documentation et de support que certains concurrents.

Chaque serveur web a ses propres forces et faiblesses, et le choix dépendra de vos besoins spécifiques, de vos compétences en configuration, de votre plateforme et de vos préférences personnelles. Il est recommandé de faire des tests et de consulter la documentation pertinente pour prendre une décision éclairée.

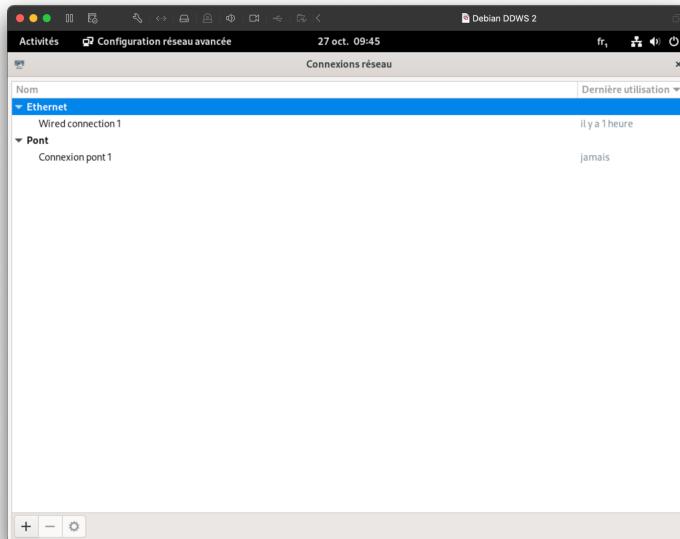
Job 04 : Mise en place du DNS.

➤ **Première étape :** Installation de bind9 et les paquets nécessaires pour la configuration du serveur DNS.

■ **Commande :** `apt -y install bind9 bind9utils dnsutils`

➤ **Deuxième étape :** Changement du mode d'accès réseau de la VM en mode Pont (Bridge).

■ **Image Représentative :**



➤ **Troisième étape :** Utilisation de la commande hostname pour connaître mon IP.

■ **Image Représentative :**

```
root@debian:/home/trystan# hostname -I  
192.168.11.139
```

➤ **Quatrième étape** : Aller dans le répertoire de configuration bind.

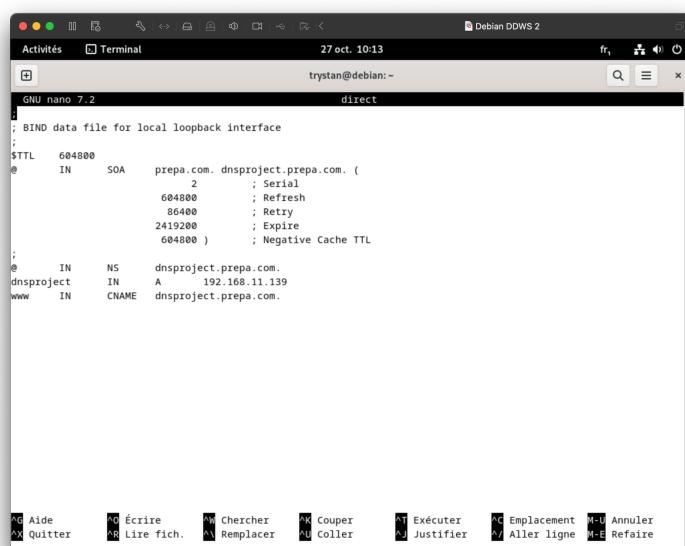
■ **Image Représentative :**

```
root@debian:/home/trystan# cd /etc/bind
```

➤ **Cinquième étape** : Modification des fichiers de configuration DNS pour associer le nom de domaine '*prepa.com*' à une adresse IP spécifique pour le serveur '*dnsproject*'.

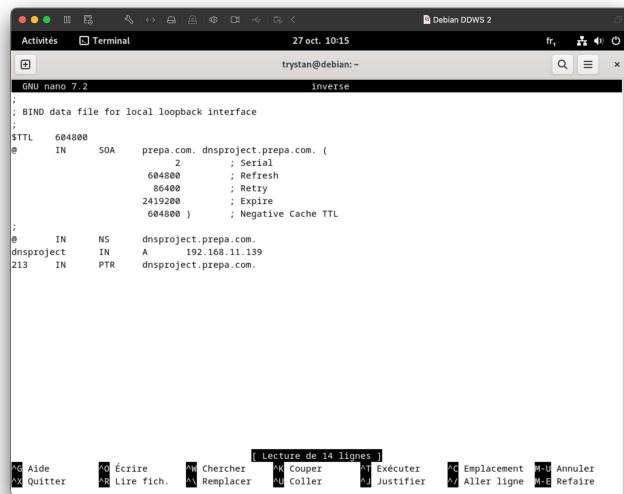
■ **Image Représentative :**

```
root@debian:/etc/bind# cp db.local direct  
root@debian:/etc/bind# nano direct
```



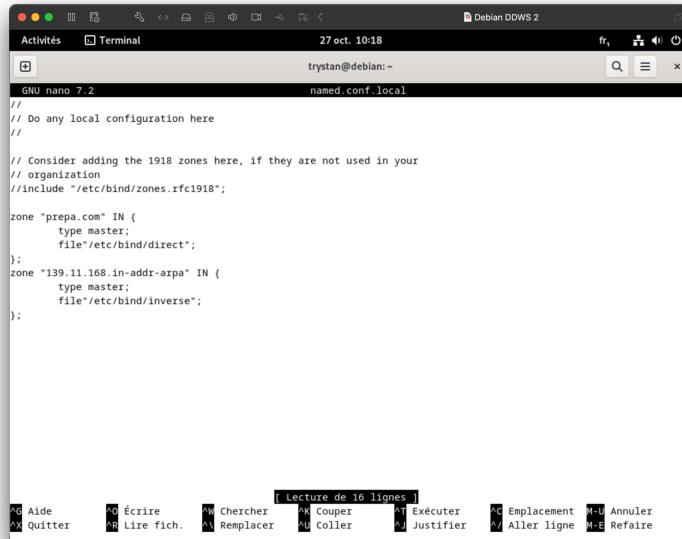
```
; BIND data file for local loopback interface  
;  
$TTL 604800  
@ IN SOA prepacom. dnsproject.prepacom. (  
    2 ; Serial  
    604800 ; Refresh  
    86400 ; Retry  
    2419200 ; Expire  
    604800 ) ; Negative Cache TTL  
;  
@ IN NS dnsproject.prepacom.  
dnsproject IN A 192.168.11.139  
www IN CNAME dnsproject.prepacom.
```

```
root@debian:/etc/bind# cp direct inverse  
root@debian:/etc/bind# nano inverse
```



```
; BIND data file for local loopback interface  
;  
$TTL 604800  
@ IN SOA prepacom. dnsproject.prepacom. (  
    2 ; Serial  
    604800 ; Refresh  
    86400 ; Retry  
    2419200 ; Expire  
    604800 ) ; Negative Cache TTL  
;  
@ IN NS dnsproject.prepacom.  
dnsproject IN A 192.168.11.139  
213 IN PTR dnsproject.prepacom.
```

```
root@debian:/etc/bind# nano 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 "prepa.com" IN {
    type master;
    file"/etc/bind/direct";
};

zone "139.11.168.in-addr-arpa" IN {
    type master;
    file"/etc/bind/inverse";
};
```

```
root@debian:/etc/bind# nano /etc/resolv.conf
```



```
# Generated by NetworkManager
search prepa.com
nameserver 192.168.11.139
```

On redémarre le service bind9 pour appliquer les changements.

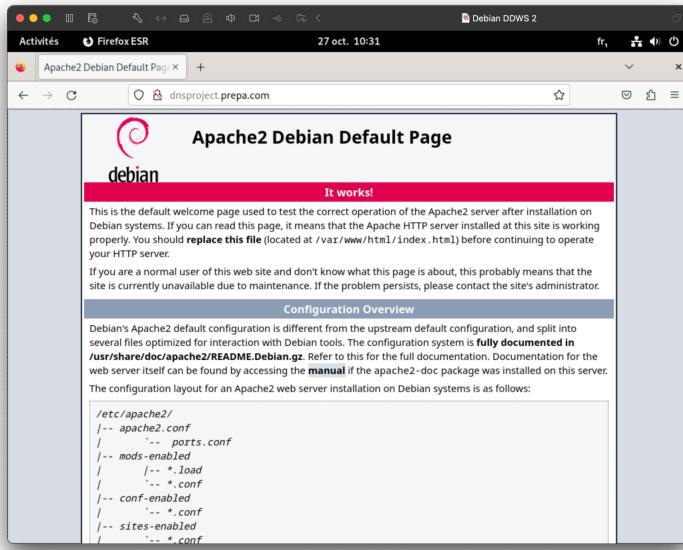
```
root@debian:/etc/bind# systemctl restart bind9
```

Je ping le nom de domaine.

```
root@debian:/etc/bind# ping dnsproject.prepa.com
```

```
PING dnsproject.prepa.com (192.168.11.139) 56(84) bytes of data.
64 bytes from debian (192.168.11.139): icmp_seq=1 ttl=64 time=0.012 ms
64 bytes from debian (192.168.11.139): icmp_seq=2 ttl=64 time=0.069 ms
64 bytes from debian (192.168.11.139): icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from debian (192.168.11.139): icmp_seq=4 ttl=64 time=0.056 ms
```

Apache est bien accessible via le nom de domaine dans la VM.



Job 05 : Nom de domaine public.

Pour obtenir un nom de domaine public, vous devez passer par un registraire de noms de domaine accrédité. Voici comment cela fonctionne :

1. Choix du nom de domaine : Tout d'abord, vous devez choisir un nom de domaine qui soit unique et qui ne soit pas déjà enregistré par quelqu'un d'autre. Vous devez également décider de l'extension de domaine que vous souhaitez (par exemple, .com, .net, .org, .fr, .io, etc.).

2. Vérification de la disponibilité : Vous pouvez utiliser les services de recherche de disponibilité de noms de domaine fournis par de nombreux registres pour vérifier si le nom de domaine que vous avez choisi est disponible.

3. Inscription auprès d'un registraire : Une fois que vous avez choisi un nom de domaine disponible, vous devez vous inscrire auprès d'un registraire de noms de domaine. Les registraires sont des entreprises qui sont autorisées à enregistrer des noms de domaine au nom des clients.

4. Fournir des informations : Lors de l'inscription, le registraire vous demandera de fournir certaines informations, telles que vos coordonnées et les informations de contact, ainsi que les serveurs DNS que vous souhaitez associer à votre nom de domaine. Ces serveurs DNS permettront de diriger les visiteurs de votre site web vers le bon emplacement.

5. Paiement : Vous devrez payer les frais d'enregistrement du nom de domaine. Les coûts varient en fonction du registraire et de l'extension de domaine choisie. La plupart des noms de domaine sont enregistrés pour une période d'un an, mais vous pouvez généralement renouveler votre enregistrement chaque année.

6. Enregistrement : Une fois que vous avez fourni toutes les informations requises et effectué le paiement, le registraire enregistre le nom de domaine pour vous. Vous devenez le titulaire officiel du nom de domaine.

7. Configuration DNS : Après avoir obtenu votre nom de domaine, vous devrez configurer les enregistrements DNS, comme les enregistrements A, MX, CNAME, etc., pour diriger le trafic vers les serveurs appropriés, gérer les e-mails, etc.

En ce qui concerne les spécificités des extensions de nom de domaine, chaque extension peut avoir ses propres règles, politiques et restrictions. Voici quelques exemples de spécificités courantes :

1. .com, .net, .org : Ce sont des extensions génériques couramment utilisées. Elles sont généralement disponibles pour tout le monde et n'ont pas de restrictions particulières.

2. .gov, .edu, .mil : Ce sont des extensions réservées à des entités spécifiques, comme les gouvernements, les établissements éducatifs et les organismes militaires.

3. .fr, .uk, .de : Ce sont des extensions nationales (ccTLDs) réservées à des pays spécifiques. Elles peuvent avoir des restrictions sur qui peut les enregistrer. Par exemple, .fr est réservé aux entités en France.

4. .io, .ai, .ly : Ce sont des extensions de domaine de pays (ccTLDs) qui ont été commercialisées pour des utilisations internationales. Elles sont souvent populaires pour les startups et les entreprises technologiques.

5. .app, .blog, .guru : Ce sont des exemples d'extensions de domaine de premier niveau (gTLDs) spécifiques qui ont été introduites pour des secteurs ou des intérêts particuliers.

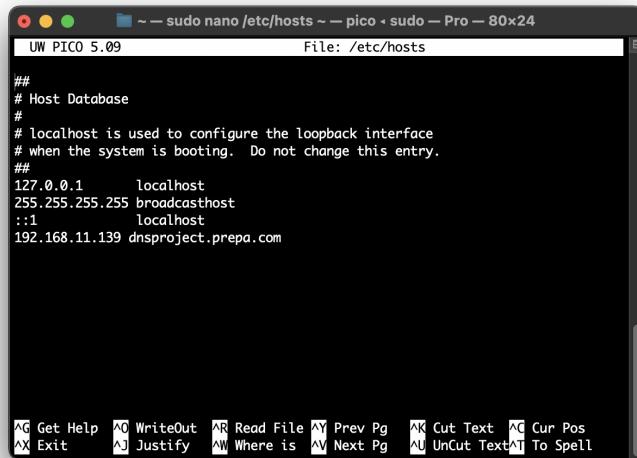
Les spécificités peuvent varier d'une extension à l'autre, et il est important de vérifier les règles d'enregistrement spécifiques à l'extension de domaine que vous envisagez d'utiliser.

Job 06 : Connexion de l'hôte au nom de domaine local du serveur.

Première étape : Commande pour configurer ces hôtes.

```
▷ ~ sudo nano /etc/hosts
```

Deuxième étape : On ajoute l'IP local du serveur qu'on relie au nom de domaine. (Dernière ligne)

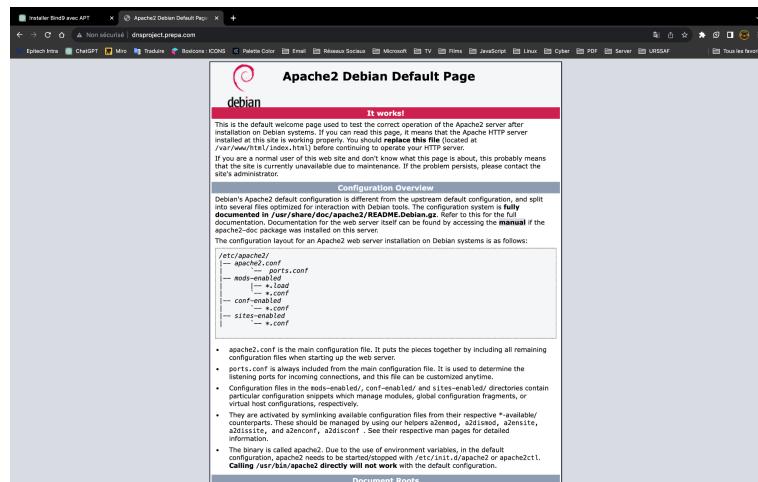


```
##  
# Host Database  
#  
# localhost is used to configure the loopback interface  
# when the system is booting. Do not change this entry.  
##  
127.0.0.1      localhost  
255.255.255.255 broadcasthost  
::1             localhost  
192.168.11.139 dnsproject.prepa.com
```

Troisième étape : Rechargement / Redémarrage du serveur Apache.

```
▷ ~ sudo apachectl restart
```

Quatrième étape : Notre page Apache sur la machine hôte via son nom de domaine.



Job 07 : Mise en place du pare-feu via UFW.

Voici une liste d'étapes pour installer, activer et configurer un pare-feu de base en utilisant UFW (Uncomplicated Firewall) sur un serveur Linux :

1. Installation du pare-feu UFW :

- Commande : `sudo apt install ufw`
- Cette commande installe UFW sur votre système. Vous devez exécuter cette commande en tant qu'administrateur (utilisez `sudo`).

2. Activation du pare-feu UFW :

- Commande : `sudo ufw enable`
- Cette commande active UFW pour qu'il soit opérationnel. Lorsque vous activez UFW, il utilisera la politique par défaut de blocage de toutes les connexions entrantes.

3. Autorisation du trafic entrant pour les ports HTTP (80) et HTTPS (443) :

- Commande : `sudo ufw allow 80/tcp && sudo ufw allow 443/tcp`
- Ces commandes autorisent le trafic entrant sur les ports 80 (HTTP) et 443 (HTTPS) de manière à ce que votre serveur puisse servir des pages web. Il est important d'autoriser ces ports si vous exécutez un serveur web comme Apache ou Nginx.

4. Liste des règles autorisées :

- Commande : `sudo ufw status`
- Cette commande affiche la liste des règles autorisées actuellement. Assurez-vous que les règles pour les ports 80 et 443 apparaissent dans la liste.

5. Personnalisation des règles de pare-feu (si nécessaire) :

- Vous pouvez personnaliser davantage les règles de pare-feu en fonction de vos besoins spécifiques. Par exemple, si vous exécutez d'autres services sur votre serveur, vous devrez peut-être autoriser d'autres ports.

6. Redémarrage d'UFW (si des règles personnalisées ont été ajoutées) :

- Commande : `sudo ufw reload`
- Après avoir ajouté ou modifié des règles de pare-feu, rechargez UFW pour que les nouvelles règles prennent effet.

Ces étapes vous permettent de mettre en place un pare-feu de base avec UFW, d'activer le trafic HTTP et HTTPS, et de personnaliser les règles en fonction de vos besoins spécifiques. Assurez-vous de comprendre les implications de chaque règle que vous ajoutez, car cela peut avoir un impact sur la sécurité et la connectivité de votre serveur.

Job 08 : Mise en place d'un dossier partagé.

Première étape : Installation de Samba :

Commande : `sudo apt install samba`

Deuxième étape : Activer le démarrage automatique de *smbd* (Samba) :

Commande : `systemctl enable smbd`

Troisième étape : Éditer le fichier de configuration de Samba est "/etc/samba/smb.conf" :

Commande : `nano /etc/samba/smb.conf`

Quatrième étape : Ajoutez ensuite les lignes suivantes pour déclarer votre partage :

Code : `

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

`

Cinquième étape : La configuration étant terminée, sauvegardez le fichier et redémarrez le service *smbd* :

Commande : `systemctl restart smbd`

Sixième étape : Créez l'utilisateur "it-connect" et définissez son mot de passe :

Commande : `useradd usershare`

Septième étape : Voici la commande pour ajouter l'utilisateur "it-connect" :

Commande : `smbpasswd -a usershare`

Huitième étape : L'utilisateur étant prêt, nous allons créer le groupe "*partage*" :

Commande : `groupadd partage`

Neuvième étape : Avec **gpasswd** ou **usermod**, ajoutez l'utilisateur "*it-connect*" au groupe "*partage*" :

Commande : `gpasswd -a it-connect partage`

Dixième étape : Le partage va être hébergé à l'emplacement "/srv/partage" de notre serveur. Commençons par créer le dossier :

Commande : `mkdir /srv/partage`

Onzième étape : Ensuite, on va attribuer le groupe "*partage*" comme groupe propriétaire de ce dossier :

Commande : `chgrp -R partage /srv/partage`

Douzième étape : Puis, nous allons ajouter les droits de lecture/écriture à ce groupe sur ce dossier :

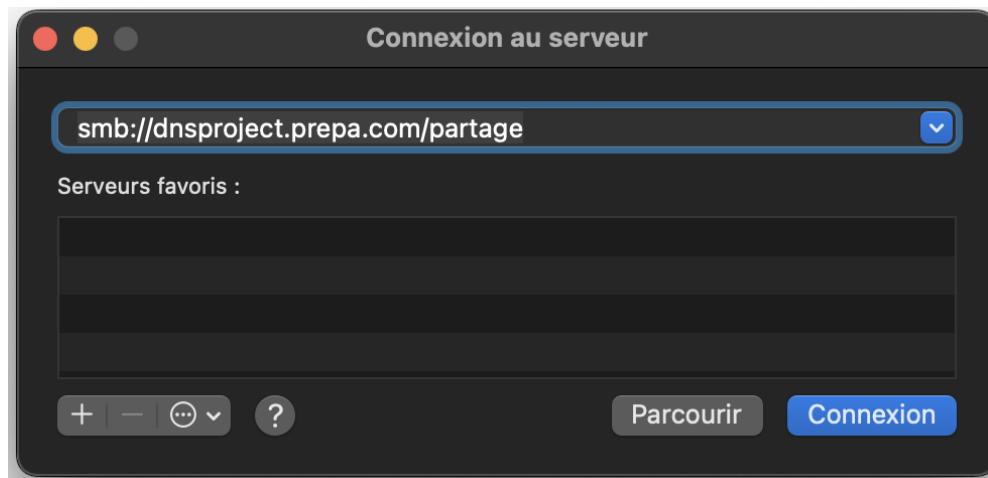
Commande : `chmod -R g+rwx /srv/partage/`

Treizième étape : On peut vérifier la configuration des droits avec la commande suivante :

Commande : `ls -l /srv/`

Quatorzième étape : Connexion au serveur sur la machine hôte :

Image de représentation :



Quinzième étape : Une fois connecté on accès au dossier partager puis on peut en créer de nouveau dans le dossier racine du serveur qui est "partage" :

Image de représentation :

