

# DDWS

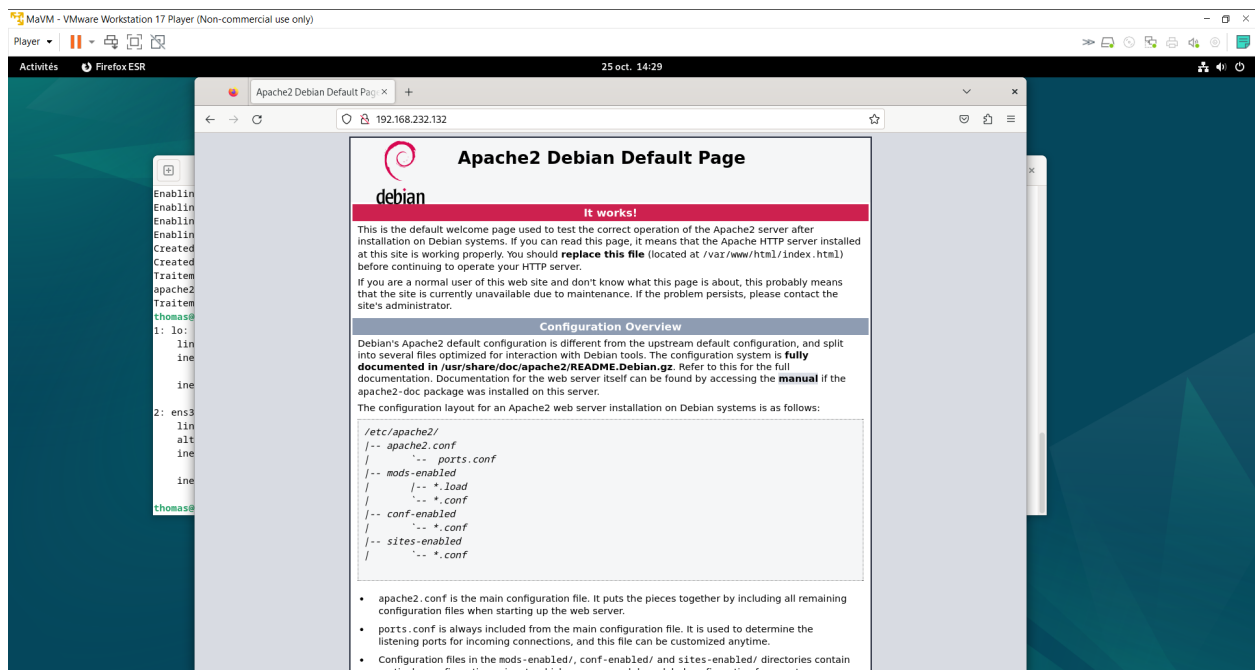
## Job 1:

Dans un premier temps nous devons installer une VM sur debian avec interface graphique, mais dans mon cas la VM était déjà créé car on avait eu recours à cette VM sur de précédent projet.

## Job 2:

Nous avons besoin d'installer le serveur apache2

```
toto@debian:~$ sudo apt install apache2 -y
```



## Job 3:

Les différents serveurs web sont :

### Apache HTTP Server (Apache) :

- Avantages :
  - Très populaire et largement utilisé dans le monde entier.
  - Open source et gratuit.
  - Une grande communauté de support et de documentation.
  - Possibilité de personnalisation via des modules.
- Inconvénients :
  - La configuration peut être complexe pour les utilisateurs novices.
  - Peut être plus gourmand en ressources que certains concurrents.

### Nginx :

- Avantages :
  - Conçu pour gérer de nombreuses connexions simultanées efficacement.
  - Excellente performance en termes de vitesse et d'évolutivité.
  - Possibilité de gérer des charges de travail HTTP et HTTPS.
  - Bonnes fonctionnalités de répartition de charge.
- Inconvénients :
  - La configuration peut être moins intuitive pour les débutants.
  - Pas aussi flexible que Apache en termes de modules tiers.

### Microsoft Internet Information Services (IIS) :

- Avantages :
  - Intégré à Windows Server, ce qui le rend facile à déployer pour les utilisateurs Windows.
  - Bonne intégration avec d'autres produits Microsoft.
  - Dispose de fonctionnalités de sécurité avancées.
- Inconvénients :
  - Moins courant sur les serveurs web non-Windows.
  - Peut nécessiter des licences, selon la version utilisée.

### LiteSpeed :

- Avantages :
  - Conçu pour être performant et efficace.
  - Peut s'intégrer avec Apache, ce qui facilite la transition.
  - Possède des fonctionnalités de mise en cache avancées.
  - Convivial pour les débutants.

- Inconvénients :
  - La version gratuite a des limitations de fonctionnalités.

#### Caddy :

- Avantages :
  - Facilité d'utilisation avec une configuration en langage naturel.
  - Prise en charge automatique du HTTPS via Let's Encrypt.
  - Possède des fonctionnalités de serveur de fichiers, de proxy inversé, etc.
  - Open source avec une option de licence commerciale.
- Inconvénients :
  - Peut ne pas être aussi rapide que Nginx dans des scénarios très chargés.

#### Cherokee :

- Avantages :
  - Interface utilisateur graphique conviviale pour la configuration.
  - Possibilité de gérer plusieurs sites web avec facilité.
  - Bonnes fonctionnalités de répartition de charge.
- Inconvénients :
  - Moins courant que les serveurs web plus populaires, ce qui peut entraîner une communauté de support plus petite.

#### OpenLiteSpeed :

- Avantages :
  - Version open source de LiteSpeed.
  - Performant et adapté aux environnements partagés.
  - Configuration facile à comprendre.
- Inconvénients :
  - Peut ne pas être aussi rapide que LiteSpeed dans certains cas.

## Job 4:

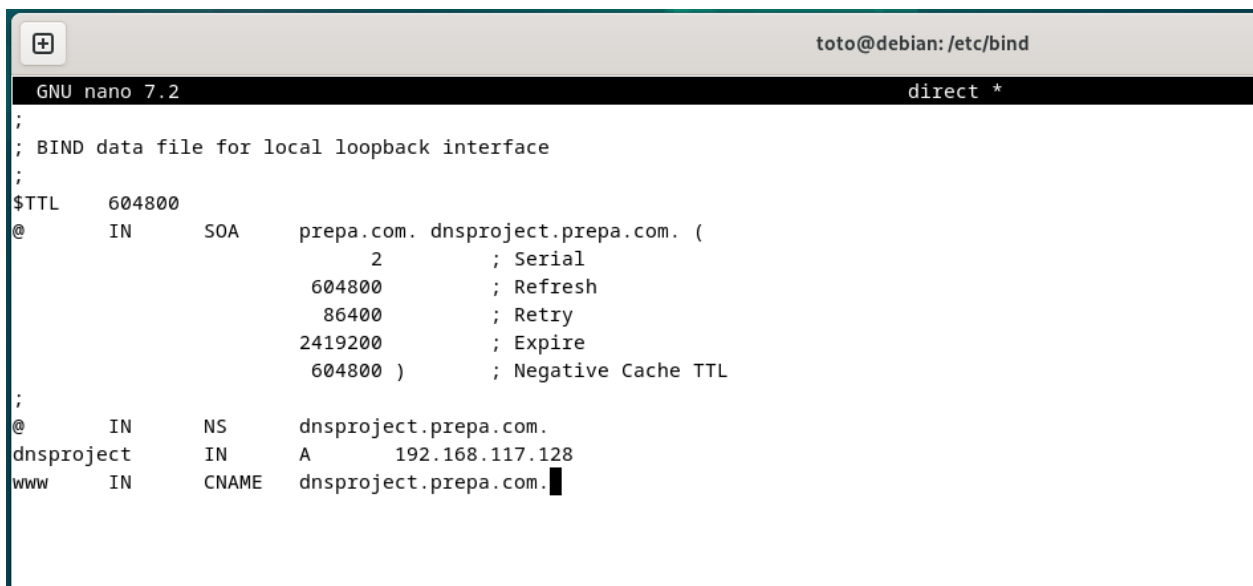
Nous allons installer les paquets:

```
toto@debian:~$ sudo apt install bind9 bind9utils dnsutils
```

Création du fichier direct avec les informations du db.local.

Pour cela nous allons utiliser la commande `cp db.local direct`.

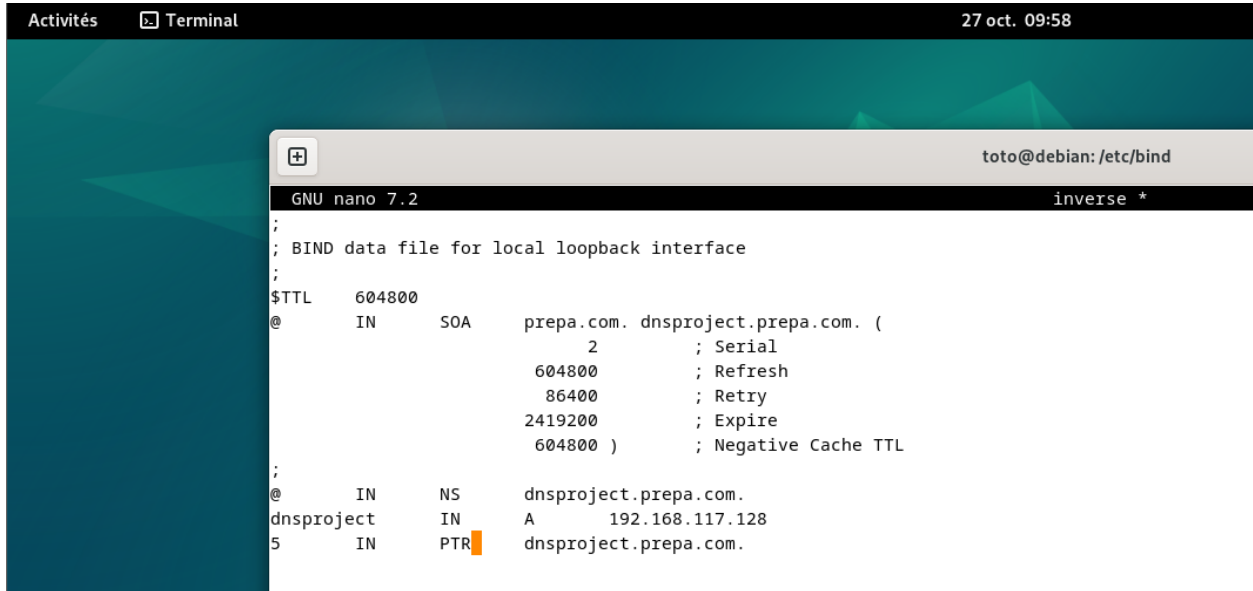
Puis nous allons modifier le fichier direct, voici la version après modification



```
GNU nano 7.2                                toto@debian: /etc/bind
direct *
;
; BIND data file for local loopback interface
;
$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        192.168.117.128
www       IN      CNAME    dnsproject.prepa.com.
```

Nous allons ensuite créer un fichier direct avec les informations du db.local, avec la commande `cp direct invers`.

Nous allons modifier le fichier inverse, voici le fichier entièrement modifié.

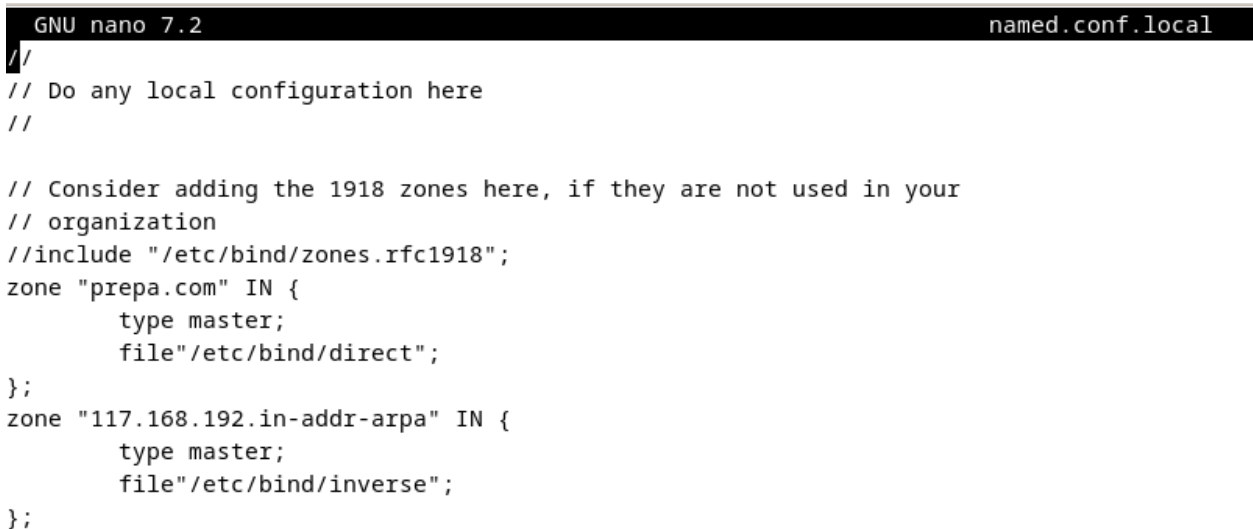


```

Activités  Terminal  27 oct. 09:58
toto@debian: /etc/bind
GNU nano 7.2  inverse *
;
; BIND data file for local loopback interface
;
$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       192.168.117.128
5         IN      PTR      dnsproject.prepa.com.

```

On modifie le fichier named.conf.local, voici la version finale



```

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 "prepa.com" IN {
    type master;
    file "/etc/bind/direct";
};
zone "117.168.192.in-addr-arpa" IN {
    type master;
    file "/etc/bind/inverse";
};

```

On restart bin9 puis on ping notre adresse

```
toto@debian: /etc/bind$ systemctl restart bind9
```

```
toto@debian:/etc/bind$ ping dnsproject.prepa.com
PING dnsproject.prepa.com (192.168.117.128) 56(84) bytes of data.
64 bytes from debian (192.168.117.128): icmp_seq=1 ttl=64 time=0.014 ms
64 bytes from debian (192.168.117.128): icmp_seq=2 ttl=64 time=0.056 ms
64 bytes from debian (192.168.117.128): icmp_seq=3 ttl=64 time=0.066 ms
64 bytes from debian (192.168.117.128): icmp_seq=4 ttl=64 time=0.061 ms
^C
--- dnsproject.prepa.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3018ms
rtt min/avg/max/mdev = 0.014/0.049/0.066/0.020 ms
```

## Job 5:

### Comment obtient-on un nom de domaine public?

Il faut vérifier si le nom de domaine est disponible, on sélectionne un registrar de domaine, s'inscrire sur le site du registrar, on configure les paramètres DNS, on paye et on le renouvelle régulièrement.

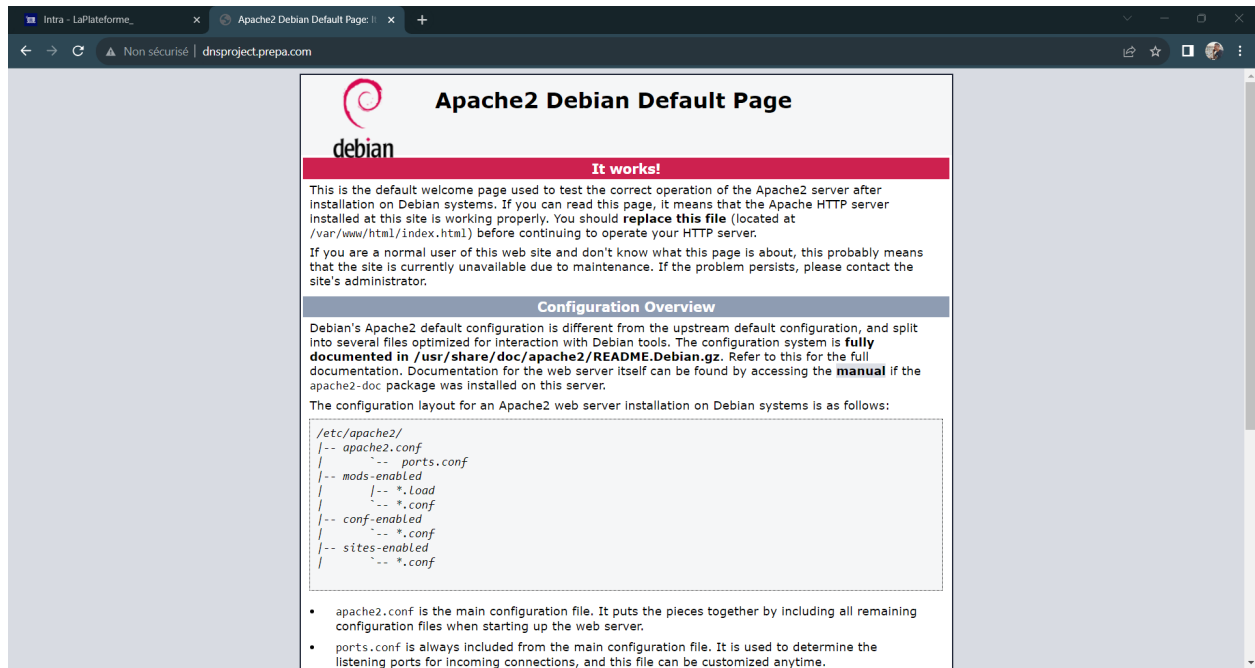
### Quelles sont les spécificités que l'on peut avoir sur certaines extensions de nom de domaine?

Voici des exemples de spécificités que l'on peut trouver sur certaines extensions de nom de domaine.

<b>Extension génériques</b>	.com, .net, .org, .info, exemple.com
<b>Extensions de Pays</b>	.fr (France), .de (Allemagne), .co.uk (Royaume-Uni), .jp (Japon), exemple.fr
<b>Extension thématiques</b>	.app, .io, .blog, .guru, exemple.app
<b>Extensions restreintes</b>	.gov (entités gouvernementales aux États-Unis), .edu (institutions éducatives), .mil (organisations militaires), exemple.gov
<b>Extensions premium</b>	.luxury, .invest, .property, exemple.luxury
<b>Extensions géographique &amp; culturelles</b>	s & culturelles .paris (pour des entreprises ou sites liés à Paris), .bzh (pour la Bretagne, en France), .nyc (pour New York City), exemple.paris

## Job 6:

On modifie les paramètres de DNS réseau pour le mettre en manuel,  
puis on essaye de se connecter à notre page apache directement sur la machine hôte.



## Job 7:

Mise en place du pare feu

Nous allons télécharger le paquet ufw

```
toto@debian:~$ sudo apt install ufw
```

On modifie le fichier before.rules

```

toto@debian: ~
GNU nano 7.2 /etc/ufw/before.rules
-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

```

On relance le ufw (pare feu) avec la commande `sudo ufw reload`.

On effectue un ping de l'ip sur la machine hôte.

```

PS C:\Users\Thomas> ping 192.168.117.128

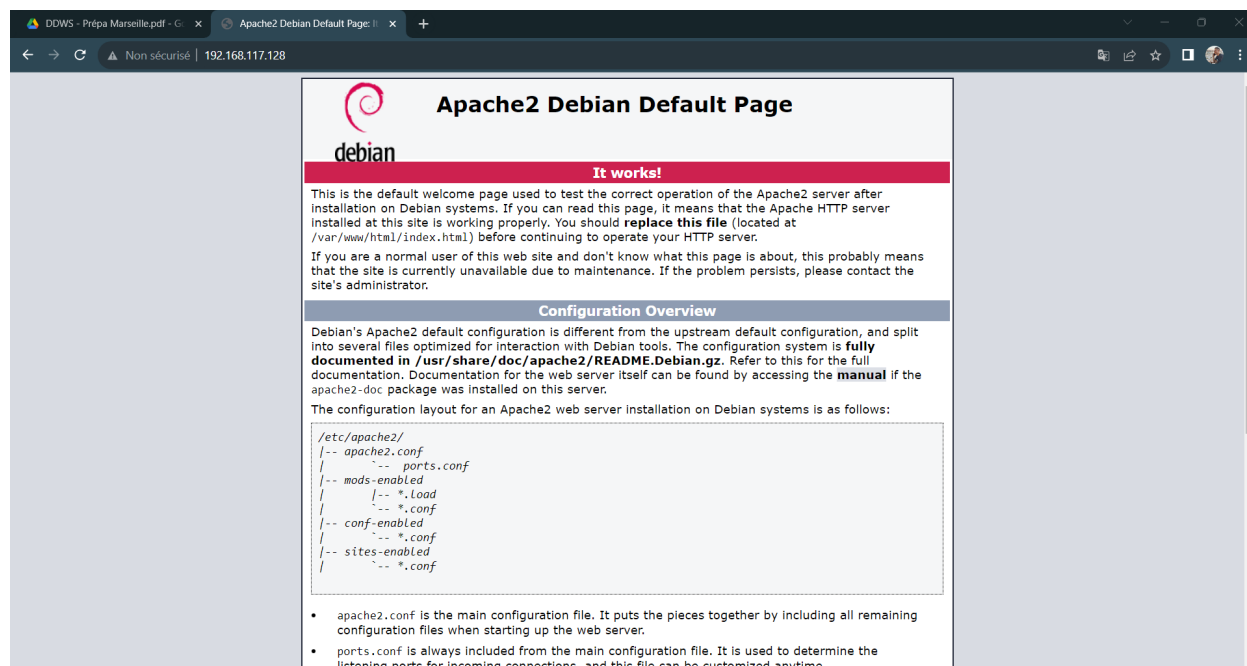
Envoi d'une requête 'Ping' 192.168.117.128 avec 32 octets de données :
Délai d'attente de la demande dépassé.

Statistiques Ping pour 192.168.117.128:
    Paquets : envoyés = 1, reçus = 0, perdus = 1 (perte 100%),
Ctrl+C
PS C:\Users\Thomas>

```

Puis on peut voir qu'on peut toujours accéder au site





## Job 8:

Nous allons télécharger le paquet samba

```
toto@debian:~$ sudo apt install samba
```

On créer le dossier partagé avec la commande `sudo mkdir Partage`,

On donne les droits au dossier avec la commande `sudo chmod 777 Partage`.

Ensuite nous devons redémarrer le service samba avec la commande `sudo service smbd restart`.

Pour finir nous pouvons voir si notre machine hôte peut accéder au dossier avec `\\notre-ip\Partage`.

