



Guillaume BOMBEN

Documentation DDWS



Job 1

Nous allons installer Debian sur une VM. Pour cela télécharger une iso de la dernière version de Debian. Ensuite ouvrez VMware et cliquez sur "Create new virtual machine". Sélectionnez "Installer disc image file" et cliquez sur Browse pour sélectionner l'iso de Debian précédemment crée. Sélectionnez l'endroit ou vous voulez installer votre VM puis déterminer une taille de stockage pour la VM. Cliquez enfin sur Finish pour terminer l'installation de la VM.

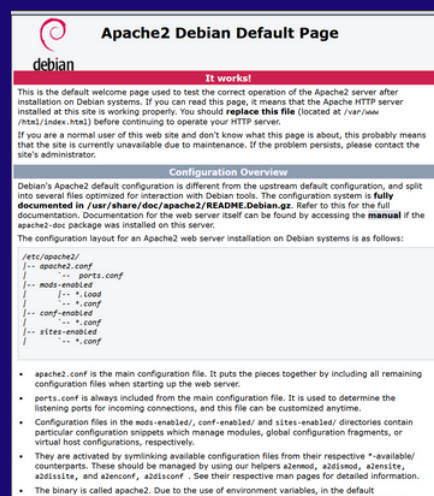
Lancez maintenant votre VM et sélectionnez "Graphical Install". Faites normalement l'installation de Debian puis lorsque vous arrivez sur la page de sélection des logiciels sélectionnez "serveur SSH" et "serveur Web" de sélectionnez. Terminer ensuite l'installation de Debian.

Job 2

Pour installer apache2 il faut faire la commande `sudo apt-get install apache2`. Avec apache2 nous avons accès au commande suivante :

- `sudo systemctl stop apache2` pour arrêter apache2.
- `sudo systemctl start apache2` pour lancer apache2.
- `sudo systemctl restart apache2` pour restart apache2.
- `sudo systemctl disable apache2` pour empêcher apache2 de se lancer automatiquement.
- `sudo systemctl enable apache2` pour permettre à apache2 de se lancer automatiquement.

faite ensuite la commande `ip a` pour trouvez l'IP de la machine puis la copier sur le navigateur hôte. Si tout est bon on devrait arriver sur la page ci-contre.



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 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:

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

- `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 particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2ensite`, `a2enssl`, and `a2enconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stoppped with `apachectl`, `apachectl` or `apachectl`.

Job 3

Nom du Serveur	Avantage	Inconvéniant
Apache HTTP Server	<ul style="list-style-type: none">• Très populaire et largement utilisé dans le monde entier.• Une grande communauté de soutien et de nombreuses ressources disponibles en ligne.• Modulaire et extensible grâce à des modules tiers.• Fiable et stable.	<ul style="list-style-type: none">• Peut être moins performant que certains serveurs web plus récents dans certaines situations.• La configuration peut être complexe pour les utilisateurs débutants.
Nginx	<ul style="list-style-type: none">• Excellente performance, particulièrement en ce qui concerne la gestion des connexions simultanées.• Léger en termes de ressources système.• Peut être utilisé comme serveur proxy inverse pour améliorer la sécurité et les performances.	<ul style="list-style-type: none">• La configuration peut sembler complexe pour les nouveaux utilisateurs.• Moins adapté pour le traitement des applications CGI comparé à Apache.
Microsoft Internet Information Services	<ul style="list-style-type: none">• Intégré à Windows Server et bien pris en charge pour les environnements Windows.• Facile à administrer via l'interface utilisateur graphique.• Soutien pour ASP.NET et d'autres technologies Microsoft.	<ul style="list-style-type: none">• Moins couramment utilisé sur les serveurs non-Windows.• Peut nécessiter des licences coûteuses dans certaines configurations.

<p>LiteSpeed Web Server</p>	<ul style="list-style-type: none"> • Hautes performances grâce à une technologie de cache efficace. • Une faible utilisation des ressources système. • Une version gratuite est disponible. 	<p>La version gratuite a des limitations en termes de fonctionnalités par rapport à la version payante.</p>
<p>Caddy</p>	<ul style="list-style-type: none"> • Facile à configurer avec une interface utilisateur conviviale. • Prise en charge automatique du chiffrement SSL grâce à Let's Encrypt. • Possède des fonctionnalités de serveur proxy inversé intégrées. 	<p>Moins répandu que certains autres serveurs web, ce qui peut signifier une communauté de soutien plus petite.</p>
<p>Cherokee</p>	<ul style="list-style-type: none"> • Interface web conviviale pour la configuration. • Hautes performances. • Possède des fonctionnalités avancées telles que la répartition de charge. 	<p>Moins populaire que d'autres serveurs web, ce qui peut limiter le support communautaire.</p>
<p>Tomcat</p>	<ul style="list-style-type: none"> • Conçu pour exécuter des applications Java, telles que les applications Web Java EE. • Intégration facile avec d'autres technologies Java. 	<p>Spécialisé dans l'exécution d'applications Java et n'est pas un serveur web générique.</p>

Job 4

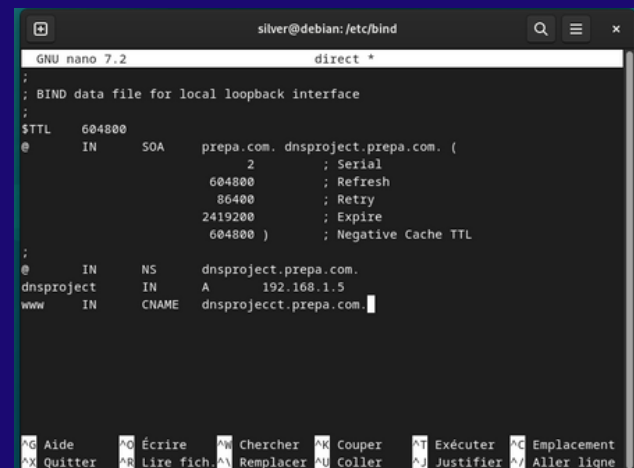
Tout d'abord nous allons installer bind9 et quelque autre paquet avec la commande :

```
sudo apt -y install bind9 bind9utils dnsutils
```

On vas aussi installer ufw et samba pour les Jobs 7 et 8 avec les commande `sudo apt-get install ufw` et `sudo apt-get install samba`

Ensuite changer le mode d'accès au réseau par bridge. Utilisait la commande `hostname -I` pour récupérer votre IP. Ici mon IP est 192.168.1.5

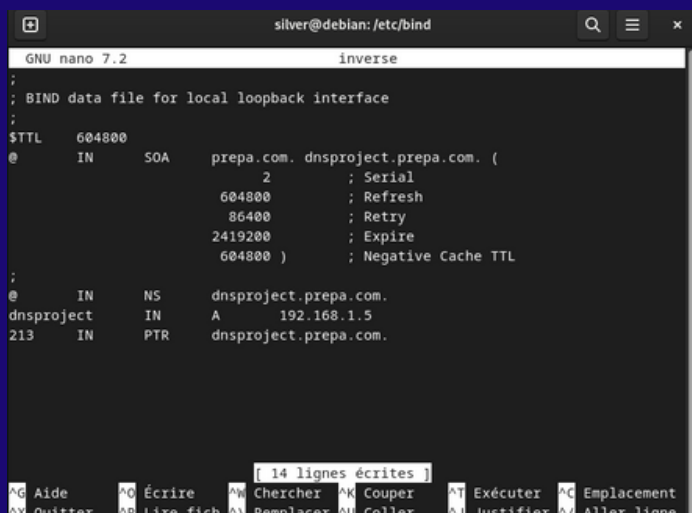
Se déplacer ensuite dans le dossier config de bind9 avec la commande `cd /etc/bind/` puis entrer les commandes `sudo cp db.local direct` puis ouvrir le fichier avec `sudo nano direct` .Modifier le fichier pour obtenir le même résultat que ci-contre (en remplaçant mon IP par la votre).



```
GNU nano 7.2      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        192.168.1.5
www       IN      CNAME    dnsprojecct.prepa.com.
```

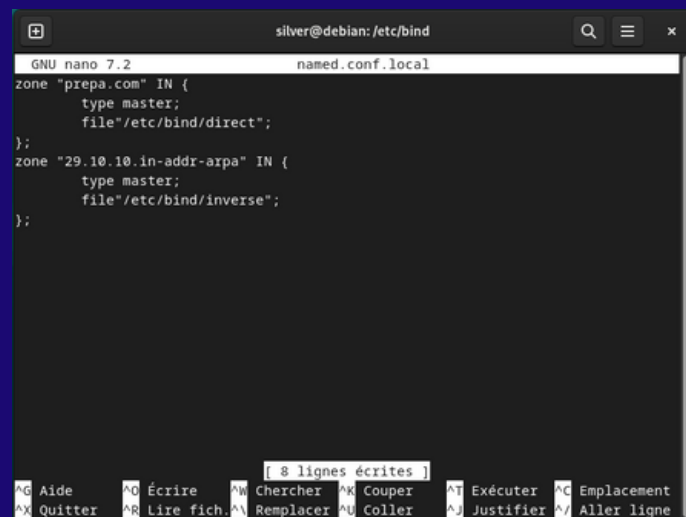
Après cela faite les commande `sudo cp direct inverse` puis ouvrir le fichier avec `sudo nano inverse` .Modifier le fichier pour obtenir le même résultat que ci-dessous (1).

Ouvrer ensuite le fichier `named.conf.local` avec `sudo nano named.conf.local` et modifier le pour qu'il soit comme sur l'image ci-dessous (2).



```
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        192.168.1.5
213       IN      PTR      dnsproject.prepa.com.
```

1



```
GNU nano 7.2      named.conf.local
zone "prepa.com" IN {
    type master;
    file "/etc/bind/direct";
};
zone "29.10.10.in-addr-arpa" IN {
    type master;
    file "/etc/bind/inverse";
};
```

2

Entrer la commande `sudo nano /etc/resolv.conf` et modifier le fichier comme ci-contre. redémarrer ensuite bind9 avec `sudo systemctl restart bind9` .Normalement si tout a été fait correctement vous pouvez accéder à la page internet apache sur votre VM en utilisant l'adresse `dnsproject.prepa.com` .

```

silver@debian:/etc/bind
GNU nano 7.2 /etc/resolv.conf
# Generated by NetworkManager
search prepa.com
nameserver 192.168.1.3
  
```

Job 5

Il y a plusieurs étape pour obtenir un nom de domaine public :

- Tout d'abord il faut choisir un nom de domaine qui est disponible. Pour cela on peut vérifier la disponibilité d'un nom de domaine auprès d'un registrar qui est une entreprise qui est habilité par l'ICANN (Internet Corporation for Assigned Names and Numbers) pour vendre des noms de domaine.
- Il faudra ensuite acheter le nom de domaine auprès du registrar choisi. Les noms de domaine sont généralement enregistrés pour une période d'un an, il faut donc s'assurer de renouveler le nom de domaine a temps avant de le perdre.

Il existe plusieurs type d'extension de nom de domaine :

Extensions géographiques (ccTLDs)	Les ccTLDs sont associés à des pays ou à des territoires spécifiques, et ils peuvent avoir des restrictions d'enregistrement. Par exemple, certains ccTLDs exigent que le titulaire d'un nom de domaine ait une présence physique ou une adresse dans le pays correspondant. D'autres ccTLDs sont ouverts à tous, sans restriction.
Extensions restreintes	Certaines extensions de domaine, comme .museum, .gov, .edu, .mil, etc., sont réservées à des types d'entités spécifiques, comme les musées, les entités gouvernementales, les établissements d'enseignement, les institutions militaires, etc.
Extensions thématiques	Il existe des extensions de domaine spécifiquement conçues pour des industries ou des communautés particulières. Par exemple, .blog, .app, .music, .guru, .bank, .insurance, etc. Ces extensions peuvent parfois avoir des restrictions d'enregistrement pour garantir que les titulaires sont liés à la thématique.

Extensions à caractère régional	Certaines extensions, telles que .paris, .nyc, .berlin, etc., sont associées à des villes ou des régions spécifiques. Elles peuvent exiger que les titulaires aient une adresse ou une présence dans cette région.
Extensions de marque	Les entreprises peuvent enregistrer des extensions de domaine correspondant à leur propre marque (par exemple, .apple, .microsoft). Ces extensions peuvent être utilisées exclusivement par la société ou ses affiliés.
Extensions IDN	Les extensions de domaine internationalisées (IDN) prennent en charge des caractères non latins, ce qui permet d'utiliser des alphabets, des scripts et des caractères spécifiques à d'autres langues.

Job 6

Pour avoir accès à la page apache depuis la machine hôte il faut configurer le dns de la machine hôte. Pour cela aller dans Paramètre puis Réseau et Internet .Sélectionner ensuite votre connexion (WIFI ou Ethernet) et sur la ligne Attribution du serveur dns cliquez sur modifier. Changer le paramètre Automatique (DHCP) en manuel et activer l'IPv4 et rentrer l'adresse IP utiliser dans le Job 4. Vous pouvez maintenant accéder à la page web apache en utilisant le même nom de domaine que dans le Job 4.

Job 7

Après avoir installer ufw dans le Job 4 nous allons tout d'abord configurer les politique de connexion in et out par défaut en refus avec les commandes `sudo ufw default incoming (in)` et `sudo ufw default outgoing (out)`.

Nous allons maintenant ouvrir leport 80 avec la commande `sudo ufw allow 80/tcp` pour autoriser le Trafic tcp sur ce port.

Ouvrer maintenant le fichier `before.rules` avec la commande `sudo nano /etc/ufw/before.rules` et modifier les `ENABLE` dans la section `# ok icmp codes for INPUT` en `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
```

Lancez ensuite ufw avec la commande `sudo ufw enable` .Pour voir si ufw est lancez vous pouvez faire la commande `sudo ufw statut` .

Job 8

Tout d'abord créer un fichier partage a l'endroit de votre choix. Ouvrer ensuite les port 139 et 445 avec les commandes `sudo ufw allow 139/tcp` et `sudo ufw allow 445/tcp` . Ouvrez ensuite le fichier `smb.conf` avec `sudo nano /etc/samba/smb.conf` et ajouter a la fin de celui ci le contenu ci-contre en changeant le path par votre propre path.

```
[patage]
comment = partage
path = /home/silver/patage
valid users = @users
force group = users
create mask = 0600
directory mask = 0771
writable = yes
```

Il faut ensuite créer un utilisateur smb et lui donnez un mot de passe avec la commande `sudo smbpasswd -a username` en remplaçant username par votre nom d'utilisateur. Changez ensuite les droit du dossier partage avec `sudo chmod -R 777 path` en remplaçant path par le chemin de votre dossier partage.

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

Recharger ensuite smb grâce à la commande `sudo service smbd restart` .Vous pouvez maintenant accéder au dossier partage sur la machine hôte en utilisant le chemin `\\IP\partage` en remplaçant IP par l'IP de votre serveur .