

JOB 01:

Installation d'une VM Debian, avec interface graphique.

JOB 02:

En premier lieu j'ai installé apache2 grâce à cette commande: "`sudo apt-get install apache2`".

Ensuite, j'ai utilisé la commande: "`sudo systemctl status apache2`" pour vérifier l'état du service apache. Puis, je suis allé dans la destination: "`/var/www/html`" cependant, j'ai taper "`firefox`" puis j'ai cliquer sur le lien "`/index.html`" qui ma permis d'ouvrir la page de lancement d'Apache sur firefox.


Debian 11.x 64-bit - VMware Workstation 17 Player (Non-commercial use only)

Player ▾ | |

Activités Firefox ESR

Apache2 Debian Default Page × + ▾

← → ↻ 127.0.0.1/index.html



Apache2 Debian Default Page

debian

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 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.

JOB 03:

Les différents serveurs Web existants: + les avantages et **inconvénients**.

Il existe de nombreux serveurs Web disponibles, chacun ayant ses propres avantages et inconvénients en fonction des besoins spécifiques.

1. **Apache HTTP Server** : Il peut être utilisé sur divers systèmes d'exploitation. Par compte, **il peut être moins performant que d'autres serveurs web pour des charges très élevées.**
2. **Nginx** : Très performant, optimisé pour servir de grandes quantités de contenu statique. mais, **il est moins adapté pour les applications nécessitant des modules complexes.**
3. **Cherokee** : il possède une interface web pour la gestion. Cependant, **la base d'utilisateurs est plus limitée par rapport à Apache ou Nginx.**
4. **Caddy** : Facilité d'installation et de configuration. Tandis qu'il peut **ne pas être aussi performant que Nginx ou LiteSpeed pour des charges élevées.**
5. **Microsoft Internet Information Services (IIS)** : Il prend en charge des technologies Microsoft telles que .NET. **Néanmoins, il est moins commun sur les systèmes non-Windows.**
6. Enfin, **LiteSpeed** : Il a une configuration conviviale. Mais, **les versions commerciales peuvent être coûteuses.**

Il est important de noter que le choix d'un serveur web dépendra des besoins spécifiques de votre projet, de votre expérience en matière de gestion de serveurs, et de la plateforme sur laquelle vous prévoyez de l'exécuter.

JOB 04:

D'abord, j'ai utilisé cette commande : **"sudo apt update"** pour mettre à jour la liste des paquets disponibles dans les dépôts de logiciels du système Linux. Ensuite, j'ai installé bind9 avec la commande: **"sudo apt install bind9"** puis, j'ai taper : **"sudo nano /etc/bind/named.conf.options"** pour configurer un forwarder. une fois rentré dans cette commande j'ai mis:

- **"forwarders {
10.0.2.15;
8.8.8.8;
};"**
- On utilise d'abord **10.0.2.15**, un serveur local, puis **8.8.8.8**, le serveur DNS public de Google. Ensuite dans le terminal j'ai saisi la commande: **"nano /etc/bind/named.conf.local"** et une fois dans le fichier nano j'ai inscrit dedans:

```
- "zone "local.lan" IN {
    type master;
    file "/etc/bind/dnsproject.prepa.com";
};"
```

ensuite dans le terminal j'ai utilisé cette commande: **"sudo cp /etc/bind/db.local /etc/bind/dnsproject.prepa.com"** elle permet de copier un modèle de configuration de zone DNS ("db.local") vers un fichier ("dnsproject.prepa.com"). Ensuite, j'ai utilisé la commande suivante: **"nano /etc/bind/dnsproject.prepa.com"** il sert à éditer le fichier de configuration de zone pour le domaine "dnsproject.prepa.com" et d'exécuté les modifications, j'ai saisi cette commande: **"nano /etc/bind/named.conf.options"**. Cette commande permet de changer à l'intérieur **"any;"** en **"::1;"**.

J'ai redémarré Bind en tapant : **"sudo service bind9 restart"** dans le terminal.

Enfin j'ai tapé la commande suivante: **"ping dnsproject.prepa.com"** pour tester.

JOB 05 :

Comment obtient-on un nom de domaine public ?

- ❖ D'abord, choisissez un registre : Un registre de noms de domaine est une entreprise accréditée par l'ICANN (Internet Corporation for Assigned Names and Numbers) pour enregistrer des noms de domaine. Vous pouvez en choisir un parmi de nombreuses options disponibles, comme **GoDaddy**, **Namecheap**, **Google Domains**, etc. Comparez les prix, les services et les fonctionnalités offerts par différents registres avant de faire votre choix. Puis, enregistrez le nom de domaine choisi. Ensuite, configurez les enregistrements DNS : Après l'enregistrement, configurez les enregistrements DNS (Domain Name System) de votre nom de domaine pour diriger le trafic vers le serveur où votre site web est hébergé. Vous pouvez généralement le faire à partir du panneau de contrôle de votre registre et pour éviter que votre nom de domaine ne soit détourné ou volé, activez la protection contre le vol de votre registrar et assurez-vous que les informations de contact associées à votre nom de domaine sont à jour. Enfin, créez votre site web et gérez les renouvellements.

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

- ❖ Les extensions de nom de domaine, également appelées domaines de premier niveau (TLD), peuvent avoir des spécificités différentes en fonction de l'extension. par exemple: Les **gTLDs** sont des extensions plus courantes comme **".com"**, **".org"**, **".net"**, etc. Il existe des extensions avec des restrictions géographiques, comme celles qui sont associées à des pays. Par exemple, le domaine **".fr"** est réservé aux entités liées à la France. Certains TLDs sont

spécifiques à un secteur ou à une industrie particulière. Par exemple, ".aero" est destiné à l'industrie aéronautique, ".museum" est destiné aux musées, ".coop" est réservé aux coopératives, ".gov" pour les entités gouvernementales aux Etat-Unis, ou ".edu" pour les institutions éducatives etc.

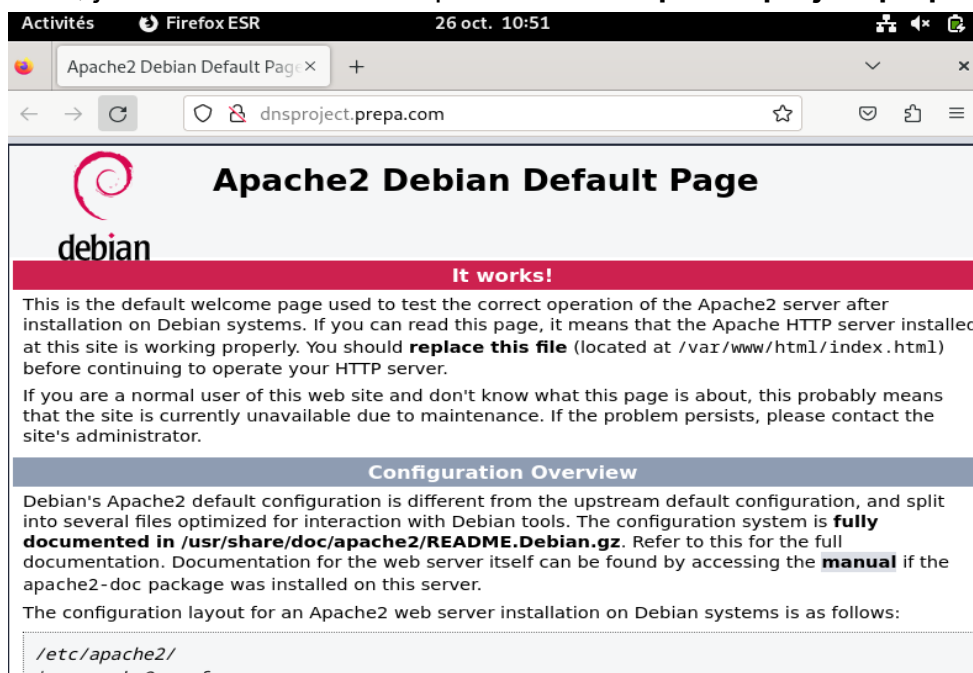
Chaque registre de domaine peut avoir ses propres politiques, il est donc recommandé de consulter les informations fournies par le registre spécifique pour le TLD qui vous intéresse.

JOB 06:

en premier lieu j'ai utilisé cette commande: "**sudo nano /etc/apache2/sites-available/dnsproject.conf**" et j'ai inscrit à l'intérieur:

```
- "<VirtualHost *:80>
    ServerName dnsproject.prepa.com
    DocumentRoot /var/www/html
</VirtualHost>"
```

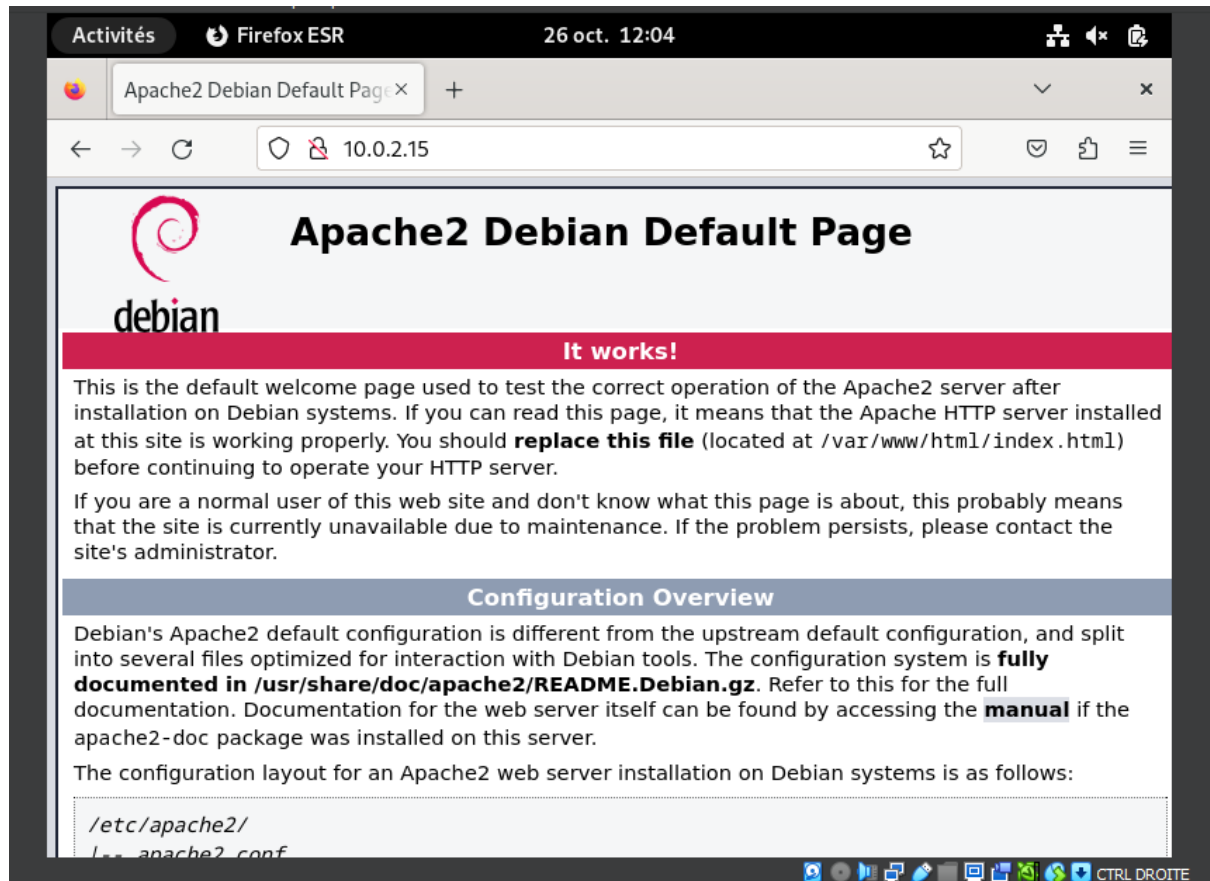
Ensuite j'ai activé apache avec la commande: "**sudo a2ensite dnsproject.conf**" et pour redémarrer j'ai utilisé la commande: "**sudo systemctl restart apache2**". Ensuite, j'ai utilisé la commande: "**sudo nano /etc/hosts**" et j'ai ajouté dedans; "**10.0.2.15 dnsproject.prepa.com**" puis je l'ai enregistré. Enfin, j'ai utilisé cette adresse pour tester: "**http://dnsproject.prepa.com**".



JOB 07:

J'ai utilisé la commande: "**sudo apt-get update**" ensuite la commande : "**sudo apt-get install ufw**" pour installer UFW. après j'ai commencé la configuration avec cette commande: "**sudo ufw allow 80/tcp**". puis j'ai bloqué le trafic ICMP (ping) avec la commande suivante: "**sudo ufw deny 5353/udp**". et pour activer UFW j'ai

utilisé la commande: **"sudo ufw enable"** ensuite pour tester la configuration j'ai saisi la commande: **"sudo ufw status"**. Pour voir si ça marche, on va taper dans le terminal : **"ping 10.0.2.15"** alors si le ping ne marche pas c'est que c'est bon. et voir également si la page Apache s'affiche sur internet : **"http://10.0.2.15"**.



JOB 08:

En premier lieu j'ai installé Samba avec la commande: **"sudo apt-get install samba"** ensuite j'ai créé un dossier dans le répertoire personnel, que j'ai nommé "Partage", et je l'ai attribué les permissions de lecture/écriture à tous les utilisateurs avec la commande: **"chmod -R 777 ~/Partage"**. et pour configurer samba voici la commande que j'ai utilisée: **"sudo nano /etc/samba/smb.conf"**.

Ensuite j'ai enregistré dedans:

```
"[Partage]
comment = Dossier partagé
path = /home/azerty/Partage
browseable = yes
read only = no
create mask = 0777
directory mask = 0777"
```

Après j'ai créé un utilisateur Samba qui aura accès au dossier partagé, avec la commande: **"sudo smbpasswd -a azerty"** et pour redémarrer voici la commande **"sudo service smb restart"**.

Pour finir on va essayer de voir si ce qu'on a fait marche, avec la commande : **"sudo service smb status"**, si un message indiquant que Samba est actif (running) apparaît, c'est que ce qu'on a fait marche, ensuite on va dans nos **fichiers sur Linux > Autres emplacements > saisir l'adresse du serveur**, à ce moment-la, on rentrera : **"smb://10.0.2.15/"**, si le dossier **"Partage"** s'affiche c'est que tout marche normalement.

