

DDWS

JOB 1

Pour commencer le projet, nous avons besoin d'utiliser une machine virtuelle. Pour cela, nous allons utiliser l'application Oracle VM VirtualBox.

Une fois installée nous allons lui transmettre une image de la distribution Linux, debian.iso.

L'installation de Debian est faite, il faut désormais configurer SSH.

Pour cela, il faut se rendre dans le terminal de la vm debian et rentrer les instructions suivantes :

- **sudo apt-get update** (Permet de mettre à jour les paquets)
- **sudo apt install openssh-server** ou **apt install openssh-server** (si mode root)

```
root@debian:~# apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  openssh-sftp-server runit-helper
Suggested packages:
  molly-guard monkeysphere ssh-askpass ufw
The following NEW packages will be installed:
  openssh-server openssh-sftp-server runit-helper
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 528 kB of archives.
After this operation, 2,214 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bookworm/main amd64 openssh-sftp-server amd64 1:9.2p1-2+deb12u1 [65.8 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 runit-helper all 2.15.2 [6,520 B]
Get:3 http://deb.debian.org/debian bookworm/main amd64 openssh-server amd64 1:9.2p1-2+deb12u1 [455 kB]
Fetched 528 kB in 0s (3,202 kB/s)
Preconfiguring packages ...
Selecting previously unselected package openssh-sftp-server.
(Reading database ... 169193 files and directories currently installed.)
```

On peut vérifier l'état du service SSH avec la commande suivante :

- **systemctl status ssh** (en mode root)

```
root@debian:~# systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Thu 2023-10-26 14:26:09 CEST; 19min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 3238 (sshd)
     Tasks: 1 (limit: 9455)
    Memory: 1.5M
       CPU: 22ms
    CGroup: /system.slice/ssh.service
           └─3238 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Oct 26 14:26:09 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server.>
Oct 26 14:26:09 debian sshd[3238]: Server listening on 0.0.0.0 port 22.
Oct 26 14:26:09 debian sshd[3238]: Server listening on :: port 22.
Oct 26 14:26:09 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
lines 1-16/16 (END)...skipping...
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Thu 2023-10-26 14:26:09 CEST; 19min ago
```

Ainsi pour démarrer le serveur SSH ou le désactiver procédez comme suit :

Pour arrêter le serveur hôte SSH

- **sudo service ssh stop**

```
debian@debian:~$ sudo service ssh stop
[sudo] password for debian:
debian@debian:~$ sudo systemctl status ssh
○ ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: inactive (dead)
     Docs: man:sshd(8)
           man:sshd_config(5)
```

Pour activer le serveur hôte SSH

- **sudo service ssh start**

```
debian@debian:~$ sudo service ssh start
debian@debian:~$ sudo systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: active (running) since Thu 2023-10-26 16:10:00 CEST; 4s ago
```

JOB 2

Pour la suite du projet, nous devons installer apache2, pour cela utiliser les commandes suivante :

- **sudo apt update**

- **sudo apt install apache2 -y**

```
debian@debian:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.57-2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Pour déterminer l'état d'apache2 on utilise la commande suivante :

- **sudo systemctl status apache2**

```
debian@debian:~$ sudo systemctl status apache2
[sudo] password for debian:
○ apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: inactive (dead) since Fri 2023-10-27 12:00:09 CEST; 27min ago
   Duration: 19h 35min 13.203s
   Docs: https://httpd.apache.org/docs/2.4/
   Process: 9290 ExecStop=/usr/sbin/apachectl graceful-stop (code=exited, status=0/SUCCESS)
   Main PID: 5183 (code=exited, status=0/SUCCESS)
   CPU: 1.206s
```

Pour activer et désactiver apache2 utiliser les commandes suivantes :

- **sudo systemctl start apache2**

```
debian@debian:~$ sudo systemctl start apache2
debian@debian:~$ sudo systemctl status apache2
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Fri 2023-10-27 12:34:10 CEST; 4s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 9424 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 9428 (apache2)
    Tasks: 55 (limit: 9455)
   Memory: 8.7M
      CPU: 27ms
   CGroup: /system.slice/apache2.service
           └─9428 /usr/sbin/apache2 -k start
             └─9429 /usr/sbin/apache2 -k start
               └─9430 /usr/sbin/apache2 -k start
```

- **sudo systemctl stop apache2**

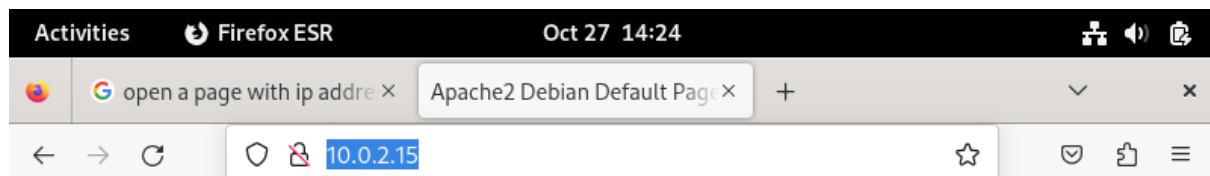
```
debian@debian:~$ sudo systemctl stop apache2
debian@debian:~$ sudo systemctl status apache2
○ apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: inactive (dead) since Fri 2023-10-27 12:36:11 CEST; 2s ago
 Duration: 2min 469ms
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 9424 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Process: 9495 ExecStop=/usr/sbin/apachectl graceful-stop (code=exited, status=0/SUCCESS)
  Main PID: 9428 (code=exited, status=0/SUCCESS)
      CPU: 55ms
```

Une fois apache2 configuré, on doit trouver son adresse IPv4. Pour cela utiliser la commande suivante :

- **nmcli -p device show**

```
debian@debian:~$ nmcli -p device show
=====
                        Device details (enp0s3)
=====
GENERAL.DEVICE:                enp0s3
-----
GENERAL.TYPE:                   ethernet
-----
GENERAL.HWADDR:                 08:00:27:5C:A6:AF
-----
GENERAL.MTU:                    1500
-----
GENERAL.STATE:                 100 (connected)
-----
GENERAL.CONNECTION:            Wired connection 1
-----
GENERAL.CON-PATH:              /org/freedesktop/NetworkManager/ActiveConnections/1
-----
WIRED-PROPERTIES.CARRIER:     on
-----
IP4.ADDRESS[1]:                10.0.2.15/24
IP4.GATEWAY:                   10.0.2.2
IP4.ROUTE[1]:                  dst = 0.0.0.0/0, nh = 10.0.2.2, mt = 100
IP4.ROUTE[2]:                  dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[3]:                  dst = 169.254.0.0/16, nh = 0.0.0.0, mt = 1000
IP4.DNS[1]:                    10.10.0.1
```

Ouvrir son navigateur web sur Debian puis insérer dans l'URL l'adresse obtenue, dans ce cas là : **10.0.2.15**



On obtient ensuite la page “Apache2 Debian Default Page”

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:

```
./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`, `a2enconf`, `a2ensite`, `a2dissite`, and `a2disconf`. 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/stopped with `/etc/init.d/apache2` or `apache2ctl`. Calling `/usr/bin/apache2` directly will **not work** with the default configuration.

Document Roots

By default, Debian does not allow access through the web browser to any file apart of those located in /var/www, public_html directories (when enabled) and /usr/share (for web applications). If your site is using a web document root located elsewhere (such as in /srv) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Debian document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the [reportbug](#) tool to report bugs in the Apache2 package with Debian. However, check [existing bug reports](#) before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

JOB 3

Un serveur web est essentiel pour héberger des sites web et des applications web.

Il permet de distribuer du contenu web aux utilisateurs finaux, que ce soit pour des sites web, des applications web, des fichiers, des vidéos, etc.

Maintenance : Les serveurs web nécessitent une maintenance régulière, notamment pour les mises à jour de sécurité, les correctifs et la gestion des performances.

Sécurité : Les serveurs web peuvent être vulnérables aux attaques, et la sécurité est une préoccupation constante.

On recense plusieurs serveurs web, on parlera ici de 3 principaux serveurs.

Serveur web	AVANTAGES	INCONVÉNIENTS
Apache HTTP Server	Logiciel open source et gratuit, grande communauté	Performance, configuration avancée
Nginx	Performant, excellente gestion des traitements de nombreuses connexions simultanées	Configuration complexe
Microsoft (IIS)	Gestion avancé des appli web, performant	Uniquement Windows, non open source, évolutivité

JOB 4

Pour ce job, nous aurons besoin de mettre en place un serveur DNS. Pour cela, on installera BIND 9 (pour Berkeley Internet Name Domain) c'est le serveur de référence.

Sur le terminal on aura besoin d'installer les éléments suivants :

- bind9
- bind9utils
- dnsutils
- ufw (pare-feu)

```
root@debian:/etc# apt install bind9
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
bind9 est déjà la version la plus récente (1:9.18.19-1~deb12u1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 17 non mis à jour.
root@debian:/etc# apt install bind9utils
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
bind9utils est déjà la version la plus récente (1:9.18.19-1~deb12u1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 17 non mis à jour.

root@debian:/etc# apt install dnsutils
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
dnsutils est déjà la version la plus récente (1:9.18.19-1~deb12u1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@debian:/etc# apt install ufw
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
ufw est déjà la version la plus récente (0.36.2-1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
```

Ensuite, on doit accéder aux fichiers suivant :

- /etc/bind/named.conf
- /etc/bind/named.conf.options
- /etc/bind/named.conf.local
- /etc/bind/db.local (/etc/bind/prepa.com)

On va apporter des modifications.

On se rend donc dans le fichier "named.conf.options", et on y met l'adresse IP 8.8.8.8 située dans le forwarders.

Cette adresse correspond à celle de Google.

```
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
    };
};
```

Se diriger ensuite dans le fichier "named.conf.local", et on va déclarer une zone nommée prepa.com.

```
GNU nano 7.2 /etc/bind/named.conf.local
//
// Do any local configuration here
//
zone "prepa.com" IN {
    type master;
    file "/etc/bind/local.lan";
};
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
```


On va copier le fichier `/etc/bind/db.local` qui servira de référence. Pour cela, on utilisera la commande suivante : `sudo cp /etc/bind/db.local /etc/bind/prepa.com`.

Se rendre dans le fichier `prepa.com` et apporter des modifications dans la zone SOA (Start Of Authority), NS (Name server) et A (type). Ce dernier permet la correspondance entre un nom DNS et une adresse IP.

```
GNU nano 7.2 /etc/bind/prepa.com *
;
; 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        192.168.1.1
www       IN      CNAME    dnsproject.prepa.com.
```

Pour éviter que les tests soient faussés on va modifier la zone `listen-on-v6` du fichier `named.conf.options`.

```
//=====
// If BIND logs error messages about the root key being expired,
// you will need to update your keys.  See https://www.isc.org/bind-keys
//=====
dnssec-validation auto;

listen-on-v6 { any; }; listen-on-v6 { ::1; };
```

Après ça, je restart mon bind9 avec la commande : `service bind9 restart`

Enfin, j'effectue un ping de mon dns :

```
root@debian:/# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=63 time=6.62 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=63 time=8.21 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=63 time=6.49 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=63 time=17.3 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=63 time=6.20 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=63 time=6.04 ms
```

La transmission est bien effectuée.

JOB 5

Comment obtient-on un nom de domaine public ?

Pour obtenir un nom de domaine public, il faut choisir un nom qui représente notre site et vérifier sa disponibilité en ligne.

Ensuite, il faut un registrar tel que GoDaddy ou Namecheap, et suivre les instructions pour enregistrer le domaine en fournissant les informations requises et en effectuant le paiement.

De plus, il faudra configurer les enregistrements DNS pour associer le domaine à l'hébergement web.

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

L'objectif initial des extensions de domaine était de classer les sites web par type.

Quelques exemples de spécificités que l'on peut trouver sur certaines extensions de nom de domaine :

- **.gov** : Réservée aux entités gouvernementales.
- **.org** : Historiquement associée aux organisations à but non lucratif, mais désormais accessible à un large public.
- **.io** : Populaire dans le secteur de la technologie, mais également utilisée pour des projets liés à l'innovation.
- **.net** : Associer aux entreprises liées aux réseaux.
- **.fr** : Associer aux entité françaises

JOB 6

Pour connecter le nom de domaine local de mon serveur dns afin que ma page apache soit accessible via ce même nom de domaine, il faut procéder comme suit :

- Ajouter le nom suivi du nom de domaine dans le fichier:

/etc/apache2/sites-available/000-default.conf

```
GNU nano 7.2                                000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerName prepa.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
```

Ajouter l'adresse url (nom + nom de domaine) dans le fichier /etc/hosts, dans la zone du prepa.com.

```
GNU nano 7.2                                /etc/hosts
127.0.0.1      localhost
127.0.1.1      debian.devops    debian
10.0.2.15      dnsproject.prepa.com  debian  debian.devops
192.168.1.1    prepa.com
```

Pour finir effectuer un apt update, pour mettre à jour les modifications.



JOB 7

Comme vu précédemment, le pare-feu s'installe avec la commande **apt install ufw**.

Il va permettre de filtrer les connexions entrantes et sortantes de la machine hôte.

Afin de bloquer le ping de notre serveur dns vers notre machine il faudra utiliser les commandes suivantes :

- Activer son pare-feu avec la commande, **ufw enable**
- **iptables -A INPUT -p icmp -j REJECT**

```
root@debian:/etc/bind# iptables -A INPUT -p icmp -j REJECT
```

- **iptables -A ou -p icmp -j REJECT**

```
root@debian:/etc/bind# iptables -A OUTPUT -p icmp -j REJECT
root@debian:/etc/bind# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
From 10.0.2.15 icmp_seq=1 Destination Port Unreachable
ping: sendmsg: Opération non permise
From 10.0.2.15 icmp_seq=2 Destination Port Unreachable
ping: sendmsg: Opération non permise
From 10.0.2.15 icmp_seq=3 Destination Port Unreachable
```

Suivi d'un ping, on s'aperçoit que la transmission est mise en échec.

JOB 8

Pour mettre en place un dossier partagé sur mon serveur dns afin que les autres membres du réseau puissent y accéder, il faudra utiliser Samba. Pour cela nous devons l'installer avec la commande, `apt install samba`.

Dans un second temps on doit créer un répertoire nommé Dossier_partagé avec la commande `mkdir` dans `/home/debian/Documents`.

Ensuite on doit se rendre dans le fichier `/etc/samba/smb.conf`. A la fin du fichier ajoutera le dossier de partage.

```
GNU nano 7.2 /etc/samba/smb.conf *
[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
; write list = root, @lpadmin
```

```
[Partage]
comment = Dossier partagé
path = /home/debian/Documents/dossier_partagé
browseable = yes
read only = no
guest ok = yes
```

On doit donner les droits aux utilisateurs pour cela j'utilise les commandes suivantes:

```
root@debian:/home/debian/Documents# chown nobody:nogroup Dossier_partagé
root@debian:/home/debian/Documents# chmod 1777 Dossier_partagé
```

On restart Samba, avec la commande suivante, **systemctl restart smbd**.

Enfin, le dossier “Dossier_partagé” est disponible dans l'interface graphique.

