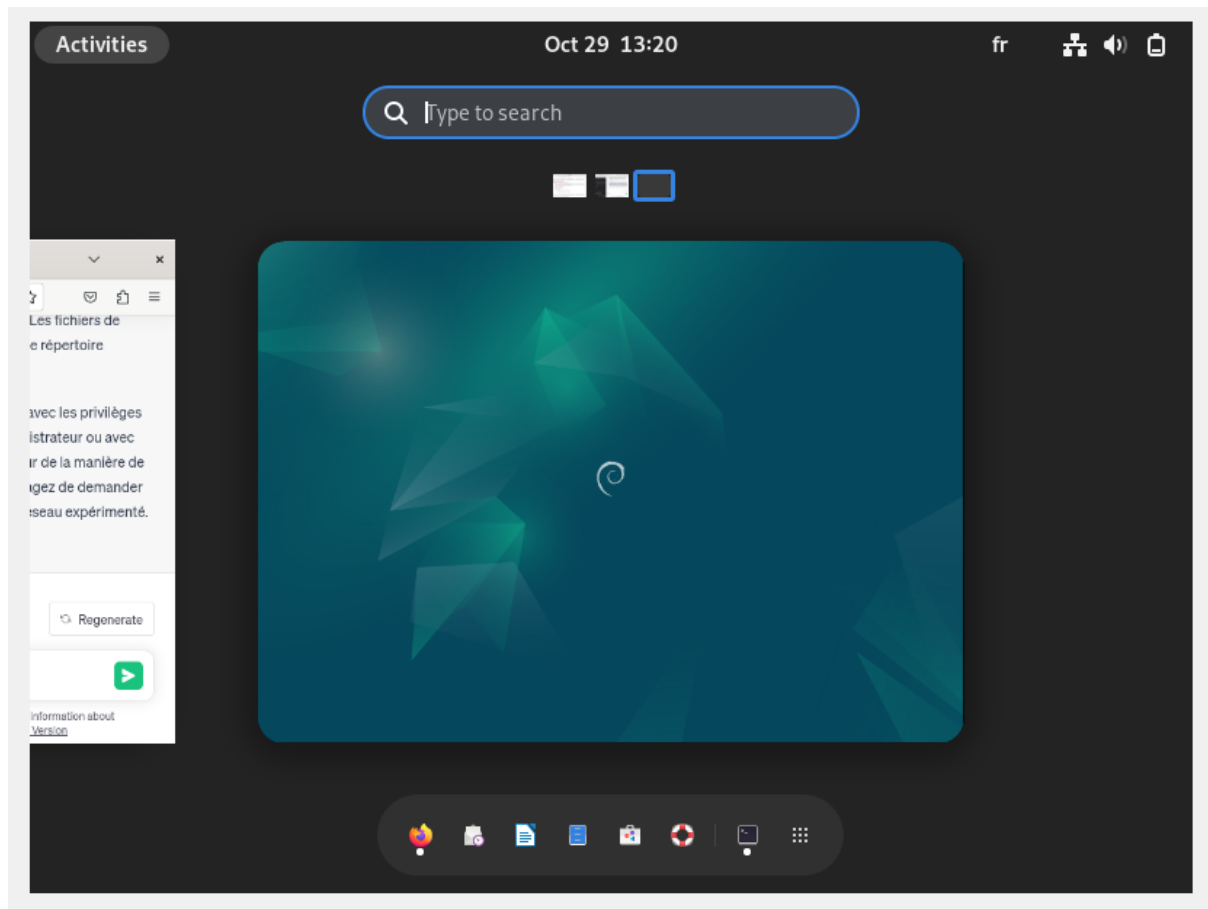


## Job 01

### Installation d'une VM Debian



**Puis installation du serveur OpenSSH avec la commande “*sudo apt-get install openssh-server*”**

```
root@debian:~# sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.2p1-2+deb12u1).
0 upgraded, 0 newly installed, 0 to remove and 112 not upgraded.
```

**Démarrage du service SSH avec la commande “*sudo service ssh start*”**

```
root@debian:~# sudo service ssh start
```

**Démarrage automatique du serveur SSH à chaque redémarrage de la VM avec la commande “*sudo systemctl enable ssh*”**

```
root@debian:~# sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
```

## Job 02

### **Installation d'Apache avec la commande “sudo apt-get install apache2”**

```
root@debian:~# sudo apt-get install apache2
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 112 not upgraded.
```

### **Démarrage du service Apache2 avec la commande “sudo systemctl start apache2”**

```
root@debian:~# sudo systemctl start apache2
```

### **Démarrage automatique du service Apache2 à chaque redémarrage de la VM avec la commande “sudo systemctl enable apache2”**

```
root@debian:~# sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
```

### **Vérification du service Apache pour savoir s'il est en cours d'exécution avec la commande “sudo systemctl status apache2”**

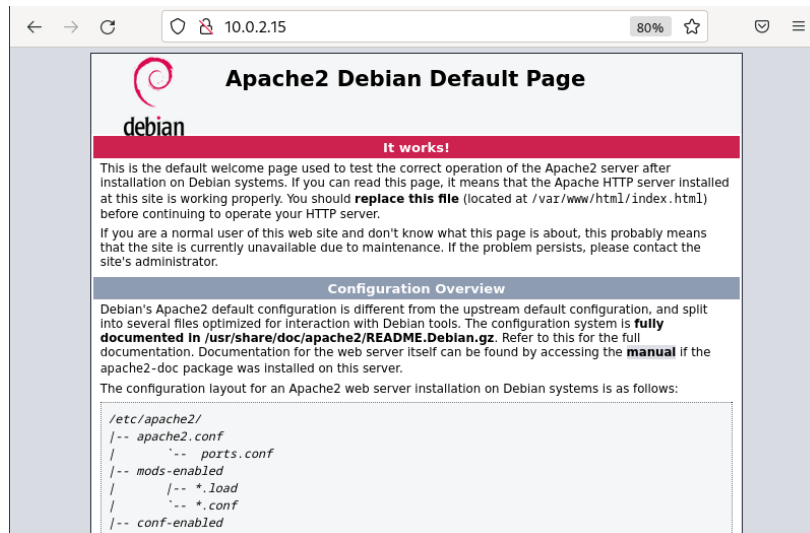
```
root@debian:~# sudo systemctl status apache2
• apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
  Active: active (running) since Sun 2023-10-29 16:35:42 CET; 18h ago
    Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 28189 (apache2)
    Tasks: 55 (limit: 2294)
  Memory: 8.2M
    CPU: 968ms
  CGroup: /system.slice/apache2.service
          └─28189 /usr/sbin/apache2 -k start
             └─28191 /usr/sbin/apache2 -k start
                └─28192 /usr/sbin/apache2 -k start
```

```
Oct 29 16:35:42 debian systemd[1]: Starting apache2.service - The Apache HTTP Server...
Oct 29 16:35:42 debian systemd[1]: Started apache2.service - The Apache HTTP Server.
```

## S'informer de l'adresse IP de la VM avec la commande "id addr"

```
root@debian:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:0d:84:ed brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 82830sec preferred_lft 82830sec
    inet6 fe80::a00:27ff:fe0d:84ed/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

## Tapez l'adresse IP de la VM dans un navigateur "http://10.0.2.15"



## Job 03

### Serveurs Web

Les serveurs web sont **essentiels** pour **héberger** des **sites web et des applications en ligne**. Chacun a ses propres **avantages** et **inconvénients** en **fonction** des **besoins** et des exigences spécifiques. Voici une comparaison des serveurs web les plus couramment utilisés :

#### 1. Apache HTTP Server (Apache)

##### Avantages :

- Prise en charge de **modules et d'extensions** pour une **personnalisation étendue**.
- **Compatible** avec de nombreux **systèmes d'exploitation**.
- Configuration **flexible** via des fichiers .htaccess.

#### **Inconvénients :**

- **Moins performant** que certains concurrents, notamment pour le **traitement** de nombreuses **connexions simultanées**.
- Peut être **complexe à configurer** pour optimiser les performances.

## **2. Nginx**

#### **Avantages :**

- **Hautes performances**, **idéal** pour gérer de **nombreuses connexions** simultanées.
- **Faible utilisation** de la **mémoire**.
- Gestion **efficace** de la **mise en cache**.
- Prise en charge du **proxy inverse** et de **l'équilibrage de charge**.

#### **Inconvénients :**

- La **configuration** peut sembler **moins intuitive pour les débutants**.
- **Moins de modules** disponibles par rapport à Apache.

## **3. Lighttpd (Lighty)**

#### **Avantages :**

- Extrêmement **léger et efficace** en matière de ressources.
- **Adapté aux serveurs** avec des **ressources limitées**.
- **Gestion efficace** des requêtes statiques.

#### **Inconvénients :**

- **Moins de support et de documentation** que les autres serveurs web.
- N'est **pas la meilleure** option pour des **applications web complexes**.

## **4. Microsoft Internet Information Services (IIS)**

#### **Avantages :**

- Intégré à Windows Server.
- **Prise en charge** des technologies Microsoft, telles que **ASP.NET**.

- Intégration étroite avec d'autres services Windows.

### **Inconvénients :**

- Principalement **adapté à l'écosystème Windows**, ce qui peut limiter la portabilité.
- Moins couramment **utilisé** pour les **serveurs web publics basés sur Linux**.

**Performance** : **Nginx** est souvent considéré comme le **champion** en matière de **performances** pour le **traitement** de **nombreuses connexions simultanées**, tandis qu'**Apache** peut être **moins performant** dans ces scénarios.

**Fonctionnalités** : **Apache** est connu pour sa **flexibilité** grâce à de nombreux modules, tandis que **Nginx** se distingue par ses **performances et sa gestion efficace de la mise en cache**.

**Sécurité** : La sécurité **dépendra** principalement de la **configuration**, mais tous ces **serveurs web peuvent être sécurisés** si configurés correctement.

**Conclusion** : Le choix du serveur web dépend du **besoin spécifique** de chacun. Si besoin de **hautes performances**, **Nginx** est un **excellent** choix. Si vous recherchez la **flexibilité et la personnalisation**, **Apache est solide**. **Lighttpd** est **idéal** pour les **environnements légers**, tandis que **IIS est la solution naturelle** pour les **systèmes Windows**.

## **Job 04**

**Installation des paquets bind9, bind9utils, dnsutils avec les commandes:**  
**"sudo apt update" "sudo apt install bind9 bind9utils dnsutils"**

```
root@debian:~# sudo apt install bind9
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bind9 is already the newest version (1:9.18.19-1~deb12u1).
0 upgraded, 0 newly installed, 0 to remove and 112 not upgraded.
```

```

root@debian:~# sudo apt install bind9utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bind9utils is already the newest version (1:9.18.19-1~deb12u1).
0 upgraded, 0 newly installed, 0 to remove and 112 not upgraded.

root@debian:~# sudo apt install dnsutils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
dnsutils is already the newest version (1:9.18.19-1~deb12u1).
0 upgraded, 0 newly installed, 0 to remove and 112 not upgraded.

```

*On se déplace dans le répertoire **/etc/bind***

```

root@debian:~# cd /etc/bind
root@debian:/etc/bind#

```

*Copier le contenu de la zone de base de données (db.local) dans un fichier nommé **"direct"**, (cela permet de faire correspondre des noms de nom de domaine avec leurs adresses IP).*

```

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

```

*Dans le répertoire **/etc/bind**, ouvrir l'éditeur nano pour le fichier **"direct"** avec la commande : **"nano direct"***

```

root@debian:/etc/bind# nano direct

```

*On modifie le contenu du fichier **"direct"** en ajoutant mon adresse IP et mon nom de domaine :*

```

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        10.0.2.15
www       IN      CNAME    dnsproject.prepa.com.

```

**Sauvegardez et fermez le fichier avec les commandes : “Ctrl+C “Ctrl+X”**

**Copier le contenu du fichier “direct” dans un fichier appelé “inverse”**

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

**Ouvrir l’éditeur de texte nano du fichier “inverse”**

```
|root@debian:/etc/bind# nano inverse█
```

**Modification du fichier “inverse” par “213 PTR dnsproject.prepa.com”  
(cela permet d’associer l’adresse IP 213 au domaine  
“dnsproject.prepa.com” pour une résolution inverse)**

```
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        10.0.2.15
213       IN      PTR      dnsproject.prepa.com.
```

**Sauvegardez et fermez le fichier avec les commandes : “Ctrl+C “Ctrl+X”**

**Ouvrir l’éditeur de texte nano du fichier “named.conf.local”**

```
|root@debian:/etc/bind# nano named.conf.local█
```

**Modification du fichier “named.conf.local” en incluant mon adresse IP à  
l’envers et mon domaine (permet de configurer le serveur DNS pour qu’il  
reconnaisse et réponde aux requêtes liées à mon domaine)**

```

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 "2.0.10.in-addr-arpa" In {
    type master;
    file"/etc/bind/inverse";
};

```

**Sauvegardez et fermez le fichier avec les commandes : “Ctrl+C “Ctrl+X”**

**Configurer les résolution DNS dans le fichier “*resolv.conf*” (permet à ma VM d’utiliser correctement le serveur DNS pour résoudre les noms de domaine)**

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

**Modifier le fichier en mettant mon adresse IP et le nom du domaine “*prepa.com*”**

```

GNU nano 7.2
# Generated by NetworkManager
search prepa.com
nameserver 10.0.2.15

```

**Sauvegardez et fermez le fichier avec les commandes : “Ctrl+C “Ctrl+X”**

**Redémarrage de Bind9 avec la commande : “*sudo systemctl restart bind9*”**

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

**Test de la résolution DNS avec la commande : “*ping dnsproject.prepa.com*”**

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

## **Job 05**

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



Pour obtenir un nom de **domaine public**, on peut utiliser des **registrars de domaine** tels que **GoDaddy**, **Namecheap**, ou **Google Domains**. Choisir un nom, vérifier sa disponibilité, créer un compte chez un registrar de domaine, ajouter le domaine à votre panier, passer commande et configurer les paramètres DNS. Gérez et protégez votre domaine pour éviter son expiration et préserver votre vie privée en ligne.

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

Les extensions de domaine (**TLDs**) peuvent avoir des **spécificités**, les extensions de domaine nationales (**ccTLDs**) sont associées à des **pays spécifiques**, et les règles d'enregistrement peuvent varier considérablement d'un pays à l'autre certaines extensions, telles que .gov, .edu et .mil, sont réservées à des entités spécifiques, telles que **les gouvernements**, **les établissements d'enseignement**, **comprendre ces spécifiques**. Le choix se fera en fonction de vos besoins, donc comprendre les particularités de chaque extension est primordial.

<b>Type d'extension</b>	<b>Nom de domaine</b>
Extensions génériques (gTLDs)	.org .net .com .info exemple.net
Extensions restreintes (rTLD)	.edu .gov .int .post .mil
Extensions thématiques (nTLDs)	.io .art .glob .app .tech
Extensions de pays (ccTLDs)	.fr(France) .cn(Chinois) .us(USA)
Extensions premium	.invest .property

### Job 06

**Mettre à jour les paramètres DNS de la machine hôte (10.10.0.1 est le DNS de La Plateforme)**

Cancel

Wired

Apply

Details

Identity

IPv4

IPv6

Security

IPv4 Method

☐ Automatic (DHCP)
☐ Link-Local Only
☒ Manual
☐ Disable
☐ Shared to other computers

Addresses

Address	Netmask	Gateway	
10.10.0.1	255.255.255.0	10.10.0.1	✕
			✕

DNS

Automatic ☐

10.0.2.15

Separate IP addresses with commas

Routes

Automatic ☒



## Job 07

**Installer le par feu ufw avec la commande “*sudo apt-get install ufw*”**

```
root@debian:~# sudo apt-get install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-1).
0 upgraded, 0 newly installed, 0 to remove and 112 not upgraded.
```

**Configurer les politiques de refus entrant avec la commande “*sudo ufw default deny incoming*” (cette politique signifie que par défaut, toutes les connexions entrantes sont rejetées, sauf celles qu’on a configurer pour autoriser le trafic entrant )**

```
root@debian:~# sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
```

**Configurer les politiques de refus sortant avec la commande “*sudo ufw default deny outgoing*” (cette politique signifie que par défaut, toutes les connexions sortantes sont rejetées, sauf celles qu’on configure avec une autorisation pour le trafic sortant )**

```
root@debian:~# sudo ufw default deny outgoing
Default outgoing policy changed to 'deny'
(be sure to update your rules accordingly)
```

**Autoriser le trafic TCP sur le port 80 avec la commande “*ufw allow 80/tcp*” (Permet aux utilisateurs d’accéder à des sites web par le trafic HTTP )**

```
root@debian:~# sudo ufw allow 80/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
```

**Autoriser le trafic TCP sur les port 139 et 445 avec la commande “*ufw allow 139/tcp*” “*ufw allow 445/tcp*”(Associé au protocole SMB, permet le partage de fichiers et d'imprimantes, autorisant le trafic sur ses ports )**

```
root@debian:~# sudo ufw allow 139/tcp
Rule added
Rule added (v6)
```

```
root@debian:~# sudo ufw allow 445/tcp
Rule added
Rule added (v6)
```

Modification du fichier *before.rules* situé dans */etc/ufw* avec la commande “*nano /etc/ufw/before.rules*”

Au commentaire *# ok icmp codes for INPUT*, on remplace les *ACCEPT* par *DROP* (Cela rejette par défaut les paquets ICMP entrants, renforçant la sécurité par limitation des informations via ces paquets )

```
# 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
```

Sauvegardez et fermez le fichier avec les commandes : “*Ctrl+C* “*Ctrl+X*”

Activer le par feu *ufw* avec la commande “*sudo ufw enable*”

```
root@debian:~# sudo ufw enable
Firewall is active and enabled on system startup
```

## Job 08

Installation de *Samba* avec la commande avec la commande “*apt-get install samba*”

```
root@debian:~# apt-get install samba
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
samba is already the newest version (2:4.17.12+dfsg-0+deb12u1).
0 upgraded, 0 newly installed, 0 to remove and 112 not upgraded.
```

Création du dossier partage entre le serveur et ma machine hôte ou autre machine avec la commande “*sudo mkdir /home/Dossier/PartageServerMachine*”

```
root@debian:~# sudo mkdir /home/DossierPartageServerMachine
```

Ouvrir le fichier *samba* avec la commande “*sudo nano /etc/samba/smb.conf*”

```
root@debian:~# sudo nano /etc/samba/smb.conf
```

**Ajout du contenu ci-dessous dans l'éditeur (Permet de configurer le dossier de partage afin qu'il soit accessible depuis ma machine, facilitant le partage de fichier entre le serveur et ma machine hôte ou autre machine)**

```
[Partage]
comment = Partage
path = /home/Partage
valid users = @users
force group = users
create mask = 0660
directory mask = 0771
writable = yes
```

**Sauvegardez et fermez le fichier avec les commandes : “Ctrl+C” “Ctrl+X”**

**Mettre en place un utilisateur avec son son MDP sur SMB pour permettre l'accès au dossier avec la commande “sudo smbpasswd -a debian”**

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

**Accorder les droits à l'utilisateur pour qu'il puisse interagir avec le fichier avec la commande “sudo chmod -R 777 /home/DossierPartageServerMachine”**

```
root@debian:~# sudo chmod -R 777 /home/DossierPartageServerMachine
```

**Redémarrage du service pour la prise en compte des modifications avec la commande “sudo service smbd restart”**

```
root@debian:~# sudo service smbd restart
```

**Conclusion :** Après réalisation de ces tâches l'utilisateur créé devrait pouvoir accéder au dossier “DossierPartageServerMachine” avec les autorisations adéquates.