

# Job 01

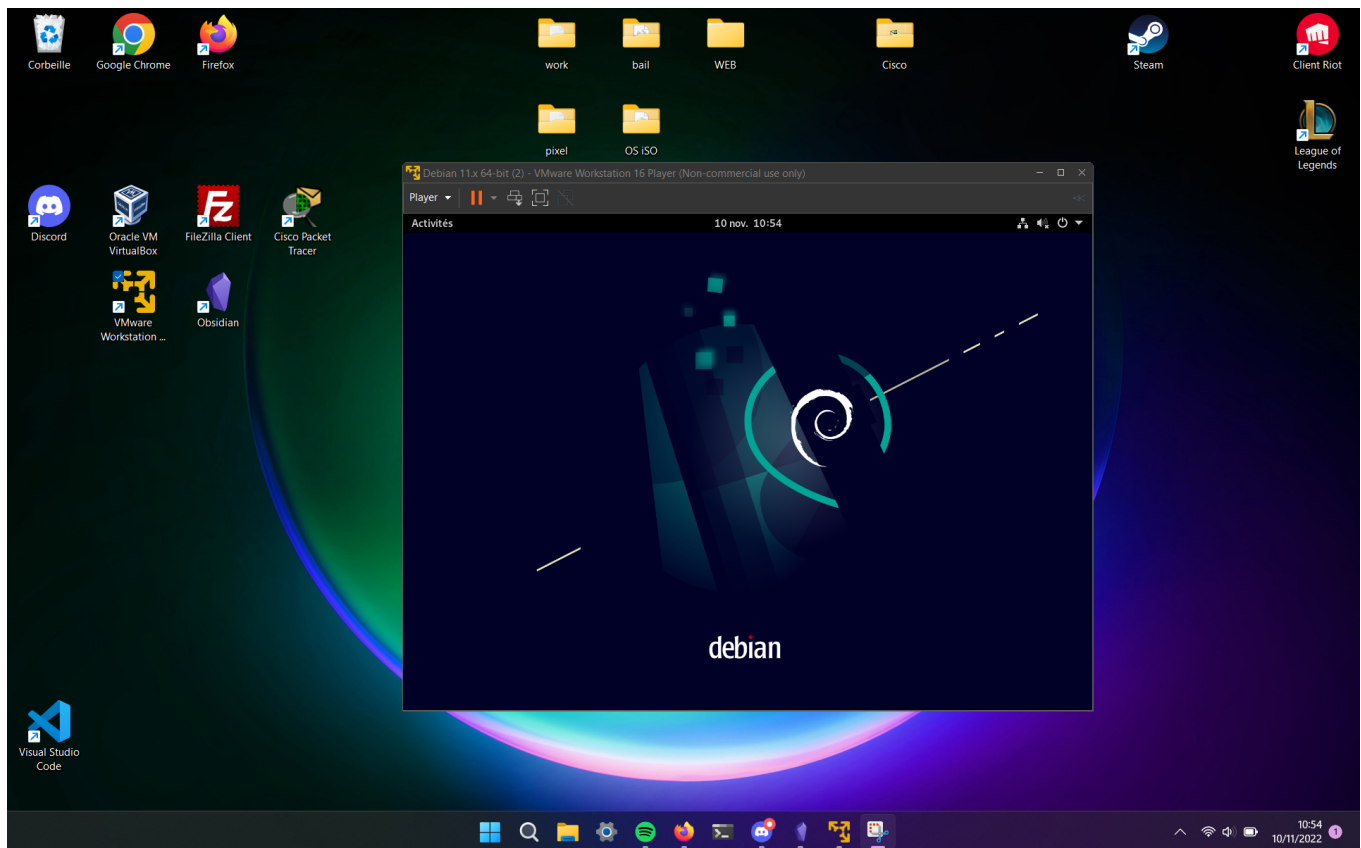
## Consigne

Pour ce sujet, il vous faudra installer une VM Debian, avec interface graphique.

Si vous souhaitez faciliter l'utilisation de votre serveur depuis votre hôte, vous pouvez très bien lui configurer SSH.

## Travail

Pour ce faire j'installe une VM [Debian](#) sur [VMware](#)



# Job 02

## Consigne

Pour commencer votre serveur, vous allez d'ores et déjà installer un serveur Web comme Apache2.

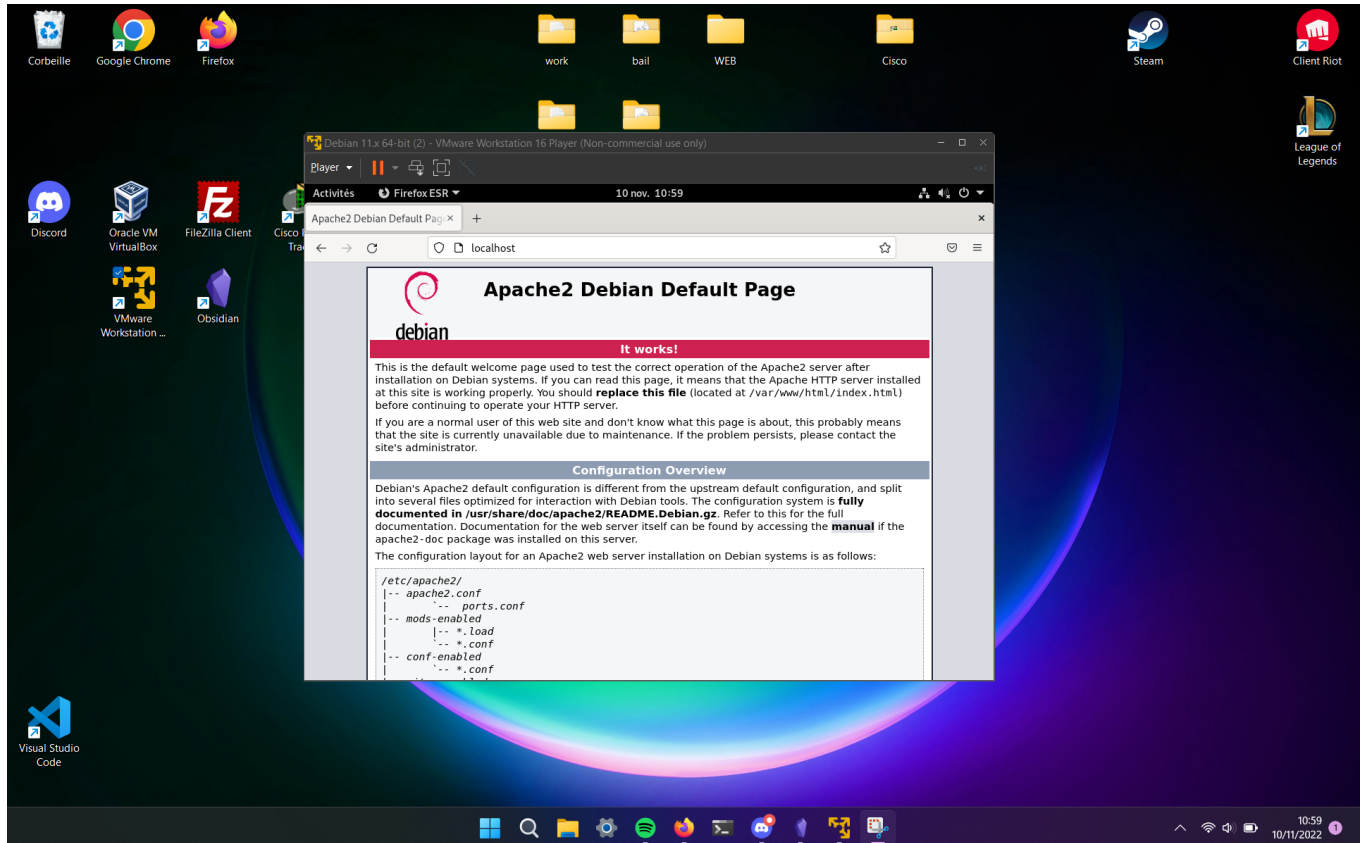
Votre serveur web devra être atteignable depuis votre hôte.

# Travail

On commence par mettre à jour les paquets et on installe [Apache2](#)

```
sudo apt-get upgrade && update  
sudo apt-get install apache2
```

Si l'installation est réussie on aura accès à la page ci-dessous en renseignant son IP dans n'importe quel browser.



## Job 03

### Consigne

Renseignez-vous sur les différents serveurs Web existants et produisez une documentation qui contiendra votre recherche ainsi que les avantages et inconvénients de chacun des serveurs.

# Travail

## [Apache HTTP Server](#)

Le serveur HTTP Apache souvent appelé httpd, ou simplement Apache a été lancé en 1995 et a célébré son 20<sup>ème</sup> anniversaire en Février 2015. 52% des sites web sont hébergés sur un serveur qui utilise Apache.

Alors que Apache httpd est plus souvent utilisé sur Linux, vous pouvez également le déployer sur OS X et Windows. Apache est, sans surprise, sous licence Apache 2. Le serveur Web lui-même utilise une architecture modulaire, dans lequel les modules supplémentaires peuvent être chargés d'étendre ses fonctionnalités. Par exemple, le chargement du mod\_proxy permettra un proxy / passerelle sur votre serveur, et mod\_proxy\_balancer permettra l'équilibrage de charge pour tous les protocoles pris en charge. Depuis la version 2.4, Apache prend également en charge HTTP / 2 grâce à un nouveau module, mod\_http2.

## NGINX

Igor Sysoev a commencé à développer NGINX en 2002, avec sa première version publique en 2004. NGINX a été développé comme une réponse à la soi-disant le problème C10K, qui est un raccourci pour «comment concevez un serveur web qui peut gérer 10 000 connexions simultanées ? » Nginx est deuxième sur une liste de serveurs web open source par l'usage, utilisé sur un peu plus de 30% des serveur.

NGINX repose sur une architecture événementielle asynchrone pour aider à alimenter son objectif de gérer des sessions simultanées massives. Il est devenu un serveur web très populaire en raison de son utilisation des ressources de sa rapidité et de sa capacité à évoluer facilement.

Nginx est publié sous une licence BAD-vie, il peut être déployé non seulement en tant que serveur Web, mais aussi en tant que serveur proxy ou équilibreur de charge.

## Apache Tomcat

Apache Tomcat Java est un conteneur de servlets open source qui fonctionne comme un serveur Web. Une servlet Java est un programme Java qui étend les capacités d'un serveur. Bien que les servlets peuvent répondre à tous les types de demandes, ils mettent en œuvre le plus souvent des applications hébergées sur des serveurs Web. Ces servlets Web sont la contrepartie Java à d'autres technologies de contenu web dynamique tels que PHP et ASP.NET. la base du code de Tomcat a été donné par Sun Microsystems à l'Apache Software Foundation en 1999, et est devenu un projet Apache haut niveau en 2005. Il est utilisé actuellement un peu moins de 1% de tous les sites.

Apache Tomcat, publié sous la version Apache License 2, est généralement utilisé pour exécuter des applications Java. Il peut, cependant, être étendue avec Coyote, pour effectuer également le rôle d'un serveur web normal qui sert des fichiers locaux en tant que documents HTTP. Plus d'informations peuvent être trouvées sur le site web du projet.

## Node.js

Node.js est un environnement JavaScript côté serveur pour les applications de réseau tels que les serveurs Web. Avec une part de marché plus petite, Node.js est utilisé sur 0,2% des sites. Il a été écrit en 2009 par Ryan Dahl. Le projet Node.js, régie par la Fondation Node.js, est facilitée par le programme des projets de collaboration de la Fondation Linux.

La différence entre Node.js et autres serveurs web populaires est qu'il est avant tout un environnement d'exécution multi-plateforme. Node.js applique une architecture événementielle capable de asynchrones I / O. Ces choix de conception d'optimiser le débit et l'évolutivité des applications Web permettant d'exécuter la communication et navigateur jeux en temps réel. Node.js met également en évidence la différence dans les piles de développement web, où Node.js fait clairement partie de l'HTML, CSS et JavaScript pile, par opposition à Apache ou Nginx qui font partie de nombreuses piles de logiciels différents.

Node.js est publié sous un mélange de licences; plus d'informations sont disponibles sur le site web du projet.

## Lighttpd

Lighttpd est apparu en Mars 2003. Il est actuellement utilisé sur environ 0,1% des sites Web et distribué sous licence BSD.

Lighttpd se distingue par sa faible utilisation mémoire, faible charge CPU, et des optimisations de vitesse. Il utilise une architecture événementielle, il est optimisé pour un grand nombre de connexions parallèles, et il prend en charge FastCGI, SCGI, Auth, Output-compression, URL-réécriture et de nombreuses autres fonctionnalités. Lighttpd est un serveur Web populaire pour Catalyst et Ruby on Rails framework web.

# Job 04

## Consigne

Mettez en place un DNS sur votre serveur Linux qui fera correspondre l'adresse IP de votre serveur au nom de domaine local suivant : "dnsproject.prepa.com"  
votre serveur devra donc pouvoir se ping via le nom de domaine

## Travail

Par définition le DNS me permet de faire le lien entre une IP et un nom de domaine c'est donc ce que je fait en modifiant le fichier "hosts"

```
sudo nano /etc/hosts
```

dans lequel j'ajoute cette ligne

```
10.0.0.1 dnsproject1.prepa.com
```

Pour effectuer nos tests, si le nom de domaine est bien configuré on sera en capacité de ping le nom de domaine comme si c'était l'adresse IP.

## Job 05

### Consigne

Faites des recherches sur comment obtient-on un nom de domaine public ?

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

### Travail

Pour acquérir le nom de domaine désiré, il faut s'adresser aux nombreux prestataires agréés. Il existe deux types de nom de domaine dits de "premier niveau" :

Les domaines génériques (gTLD), certains peuvent être libres (.com, .net, .info, .org...) et d'autres réservées à certains organismes (.int, .edu, .gov...).

Les codes pays (ccTLD) qui sont composés de 2 lettres conformément à la norme ISO 3166 (.fr, .es, .it, .uk, .de...). On compte actuellement 244 ccTLD.

Le choix de l'extension du nom de domaine dépend de la stratégie commerciale et du public-cible : Pour un positionnement à l'échelle mondiale, l'extension de domaine .com sera la plus adaptée. Bien que conçue initialement pour les organisations commerciales elle est aujourd'hui l'extension la plus utilisée. Elle est conventionnelle quand on échange une adresse internet.

Le choix d'une ccTLD réduira nécessairement la portée du site mais aura l'avantage d'illustrer l'ancrage des activités présentées dans l'aire géographique choisie et de rassurer ping dnsproject.prepa.com les visiteurs du pays en question

## Job 06

### Consigne

Connectez votre hôte au nom de domaine local de votre serveur, pour que votre page apache soit accessible via ce même nom de domaine.

## Travail

L'objectif de cette exercice est de pouvoir accéder a ma page apache depuis mon hôte c'est à dire dans mon réseau local.

Pour ce Job on doit utiliser un DNS, j'ai donc choisi d'utiliser bind9 que j'installe donc sur ma VM ainsi que les utilitaires qui vont avec .

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

Place maintenant à la configuration premièrement avec le server Apache

```
cd /etc/apache2
cp /sites-available/000-default.conf /sites-available/dnsproject1.prepa.com.conf
sudo nano /sites_available/dnsproject1.prepa.com.conf
```

```
/etc/apache2/sites-available/dnsproject1.prepa.com.conf *
```

```
<VirtualHost *:80>
```

```
    ServerName dnsproject1.prepa.com
    ServerAlias www.dnsproject1.prepa.com

    DocumentRoot /var/www/html

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
</VirtualHost>
```

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

```
cd
```

Ensuite on configure notre DNS "bind9"

```
cd /etc/bind/
sudo cp db.empty dnsproject1.prepa.com
sudo nano dnsproject1.prepa.com
```

```

GNU nano 5.4 /etc/bind/dnsproject1.prepa.com *
; BIND reverse data file for empty rfc1918 zone
;
; DO NOT EDIT THIS FILE - it is used for multiple zones.
; Instead, copy it, edit named.conf, and use that copy.
;
$TTL      86400
@         IN      SOA      dnsproject1.prepa.com. root.dnsproject1.prepa.com.
                        100      ; Serial
                        604800   ; Refresh
                        86400    ; Retry
                        2419200  ; Expire
                        86400 )  ; Negative Cache TTL
;
@         IN      NS       ns1.dnsproject1.prepa.com.
@         IN      A        10.0.0.1
ns1       IN      A        10.0.0.1
www       IN      A        10.0.0.1

```

```
sudo nano named.conf.local
```

```

GNU nano 5.4 /etc/bind/named.conf.local *
//
// Do any local configuration here
//

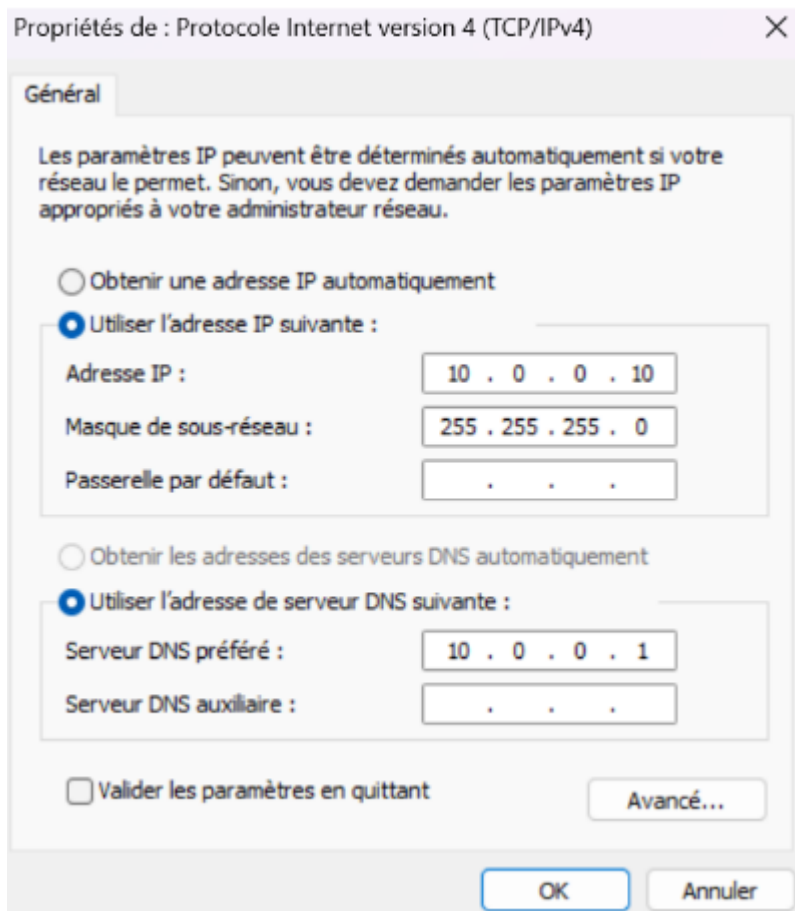
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "dnsproject1.prepa.com"{
    type master;
    file "/etc/bind/dnsproject1.prepa.com";
};

```

```
sudo systemctl restart apache2 && sudo systemctl restart bind9
```

On configure également au niveau de l'hôte



## Job 07

### Consigne

Maintenant que vous avez un serveur fonctionnel, il faudrait qu'il puisse distribuer des adresses IP aux autres machines virtuelles qui seraient connectées à lui.

Pour cela, mettez en place un serveur DHCP sur le serveur principal qui aura pour but d'attribuer des adresses IP aux machines virtuelles présentes sur son réseau local.

Vos autres machines devront aussi avoir accès à votre page web via le nom de domaine

### Travail

On commence par installer le paquet de DHCP server

```
sudo apt-get -y install isc-dhcp-server
```

On fait une backup du fichier de configuration

```
sudo cp /etc/dhcp/dhcpd.conf /etc/dhcp/backup_dhcpd.conf  
sudo nano /etc/resolv.conf
```



```
GNU nano 5.4 /etc/resolv.conf *
# Generated by NetworkManager
search dnsproject1.prepa.com
nameserver 10.0.0.1
```

```
sudo nano /etc/dhcp/dhcpd.conf
```

```
GNU nano 5.4 /etc/dhcp/dhcpd.conf

# A slightly different configuration for an internal subnet.
subnet 10.0.0.0 netmask 255.255.255.0 {
    range 10.0.0.20 10.0.0.40;
    option domain-name-servers 10.0.0.1;
    option domain-name "dnsproject1.prepa.com";
    option routers 10.0.0.1;
    option broadcast-address 10.0.0.255;
    default-lease-time 600;
    max-lease-time 7200;
}
```

Avant de finir cette configuration il est important de configurer l'interface avec :

```
sudo nano /etc/default/isc-dhcp-server
```

et en ajoutant cette ligne dans notre cas

```
INTERFACESv4="enp2s1"
```

On restart le tout et verifions ensuite l'état du server.

```
sudo systemctl restart isc-dhcp-server && sudo systemctl enable isc-dhcp-server
systemctl status isc-dhcp-server
```

## Job 08

### Consigne

Faites en sorte que votre serveur principal serve de Gateway à vos autres machines virtuelles. Les autres machines ne pourront avoir internet qu'en passant par votre serveur principal.

# Travail

Ce job est déjà réalisé par nos configuration précédente.

## Job 09

### Consigne

Mettez en place un pare-feu en utilisant ufw sur votre serveur principale de manière que votre hôte puisse accéder à la page apache par défaut, mais qu'il ne puisse plus ping votre serveur

# Travail

On commence comme toujours par l'installation du paquet

```
sudo apt-get -y install ufw
```

Et on active le pare-feu car ce n'est pas le cas par défaut

```
sudo ufw enable
```

Pour bloquer les pings on doit modifier les règles de notre pare-feu, ce qui se fait dans :

```
sudo nano /etc/ufw/before.rules
```

Pour bloquer le ping on doit commenter la ligne :

```
-A ufw-before-input -p icmp -icmp-type echo-request -j ACCEPT
```

Maintenant pour accéder au site nous devons autoriser le port sur lequel il se situe c'est à dire le port 80.

```
sudo ufw allow 80  
sudo ufw reload
```

On peut désormais accéder au site mais les pings ne sont pas autorisées.

# Job 09

## Consigne

Mettez en place sur votre serveur un dossier partagé avec les autres membres de votre réseau (soit la où les autres machines virtuelles). Ils pourront partager des fichiers dans ce dossier, ainsi que récupérer des fichiers depuis ce dossier.

Ce dossier doit être accessible dans votre gestionnaire de fichier en interface graphique.

## Travail

On commence par installer samba

```
sudo apt-get -y install samba
```

ainsi que l'activer

```
sudo systemctl -now smbd
```

Et on modifier le fichier de configuration tel que :

```
sudo nano /etc/samba.smbd.conf
```

dans lequel on ajoute

[Public]

path = /home/clement/Public

browsable = yes

writable = yes

read only = no

force create mode = 0666

force directory mode = 0777

```
sudo systemctl restart smbd
```

On crée un user "partagetest" pour tester la configuration auquel on donne les droits suivants:

```
sudo smbpasswd -a partagetest  
sudo smbpasswd -e partagetest
```

## Pour aller plus loin ...

### Consigne

Faites l'installation d'un certificat pour votre serveur web, pour activer le HTTPS sur votre serveur web Apache

Vous devrez donc pouvoir utiliser votre serveur web de manière sécurisée.

N'hésitez pas à utiliser openssl pour générer votre certificat.

Renseignez-vous aussi sur la différence entre les certificats SSL donnés par des organismes extérieurs et le vôtre auto-signé ?

Pourquoi votre certificat apparaît-il comme non sécurisé dans votre navigateur ?

Répondez à ces questions dans votre documentation

### Travail