Fiche Infra

I/ Installations

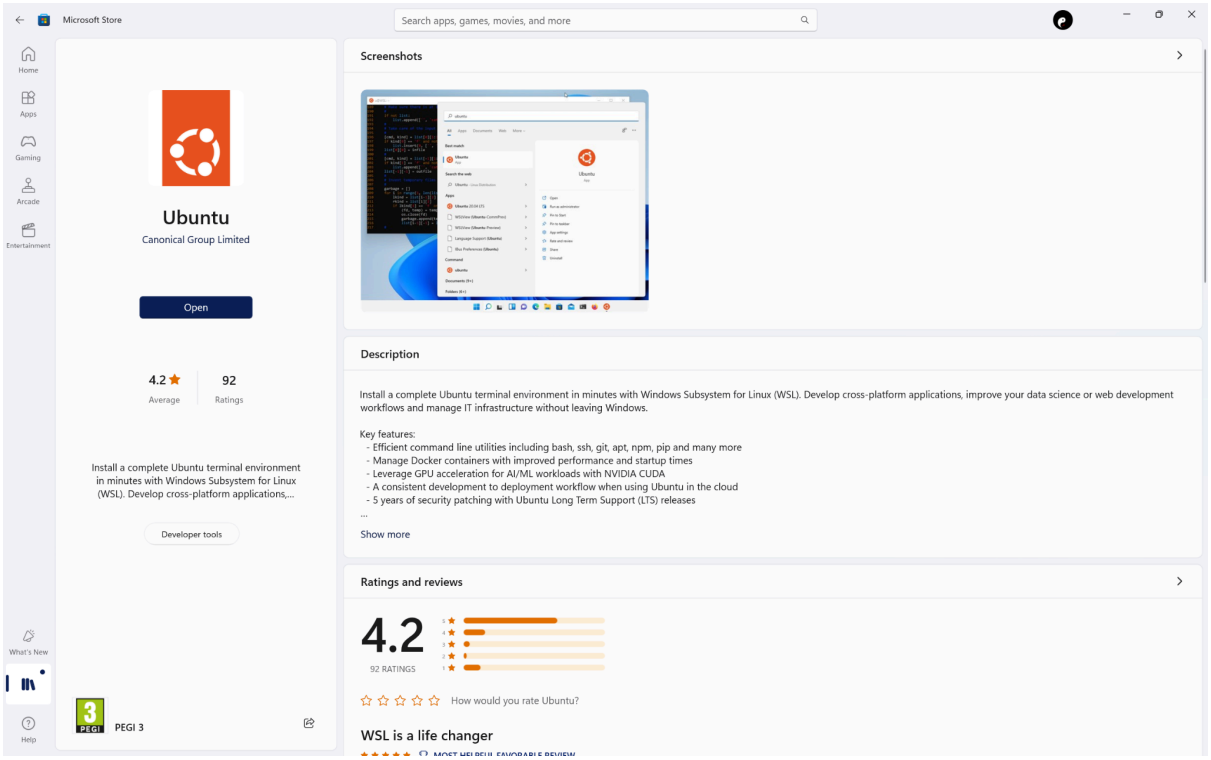
Les prérequis sont bien évidemment Linux et une distribution d'images (J'ai opté pour Ubuntu).

Tout d'abord j'ai téléchargé Linux directement sur mon pc à la place d'utiliser une VM, ce qui signifie que j'ai pratiquement que manipuler sur mon cmd.

Premièrement j'ai tapé " wsl –install " afin d'installer Linux sur mon système windows:

```
C:\Windows\System32>wsl --install
Ubuntu is already installed.
Launching Ubuntu...
```

Deuxièmement, téléchargez Ubuntu sur votre Microsoft Store comme ci dessous:



II/ Configuration d'un Serveur Web

Tout d'abord, nous allons installer Apache 2 avec la commande suivante (Verifiez bien que vous soyez en mode root):

```
jlao@JacktopV2:~$ sudo apt install apache2
[sudo] password for jlao:
```

Ensuite nous devons modifier ce fichier /etc/apache2/apache2.conf avec un éditeur de texte (dans notre cas ça sera avec Vim):

```
jlao@JacktopV2:~$ sudo apt install vim
```

1) Mise en place d'un projet local:

Nous allons créer une configuration pour notre hôte virtuel pour notre projet:

```
root@JacktopV2:~# vi /etc/apache2/sites-available/reservation-flights.local.conf
```

Et écrire ceci:

```
<VirtualHost *:80>
    ServerName reservation-flights.local
    DocumentRoot /var/www/reservation-flights

    <Directory /var/www/reservation-flights>
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Nous allons ensuite activer le site et redémarrer Apache:

```
root@JacktopV2:~# sudo a2ensite reservation-flights.local.conf
Site reservation-flights.local already enabled
root@JacktopV2:~# sudo systemctl reload apache2
```

Ensuite nous allons mettre à jour le fichier /etc/hosts et ajouter cette ligne

```
127.0.0.1       reservation-flights.local
```

2) Certificat SSL

Afin de créer un certificat SSL, il faudra taper cette commande

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/selfsigned.key -out /etc/ssl/certs/selfsigned.crt

Le certificat est désormais créé, maintenant il suffit de remplir les informations suivantes:

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Paris
Locality Name (eg, city) []:Paris
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Reservation Flights Co
Organizational Unit Name (eg, section) []:RSC
Common Name (e.g. server FQDN or YOUR name) []:Jacques
Email Address []:jacqueslao03@gmail.com
```

Après avoir remplis les informations nécessaires nous allons configurer Apache afin d'utiliser le certificat SSL par le biais de cette commande;

```
root@JacktopV2:~# sudo vi /etc/apache2/sites-available/reservation-flights.local-ssl.conf
```

Et y mettre ceci:

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
    ServerName reservation-flights.local
    DocumentRoot /var/www/reservation-flights

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/selfsigned.key

    <Directory /var/www/reservation-flights>
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
</IfModule>
```

Pour la suite, nous allons activer le site de configuration SSL, les modules requis et redémarrer Apache avec ces commandes

```
root@JacktopV2:~# sudo a2ensite reservation-flights.local-ssl.conf
Site reservation-flights.local-ssl already enabled
root@JacktopV2:~# sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
root@JacktopV2:~# sudo systemctl reload apache2
```

Maintenant si tout fonctionne on devrait être capable d'accéder au site HTTPS

https://reservation-flights.local dans notre cas là.

Nous pouvons parfois rencontrer un avertissement au sujet du certificat. Nous pouvons procéder au site sans nous soucier puisque nous sommes dans un environnement local.

III/ DNS & DHCP

Configuration du DNS:

Tout d'abord nous allons installer bind9 en faisant;

```
sudo apt install bind9
```

Par la suite nous allons configurer le fichier bind9

```
sudo vi /etc/bind/named.conf.local
```

Et le modifier en ça

```
zone "reservation-flights.local" {
    type master;
    file "/etc/bind/db.reservation-flights.local";
};
```

Dans la foulée, nous allons après nous créer un fichier zone pour notre domaine

```
sudo vi /etc/bind/db.reservation-flights.local
```

Et y mettre ceci dedans

```
;
;BIND data file for reservation-flights.local
;
$TTL    604800
@       IN      SOA     ns.reservation-flights.local. admin.reservation-flights.local. (
                              3         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.reservation-flights.local.
ns      IN      A       127.0.0.1
@       IN      A       127.0.0.1
```

Maintenant faire un petit sudo systemctl restart bind9 et mettre à jour les machines clientes en modifiant le fichier /etc/resolv.conf

```
nameserver 127.0.0.1
```

Configuration du DHCP:

Dans cette partie, nous allons utiliser un server DHCP local (ISC DHCP sur Linux).

Afin de l'installer, il suffit tout simplement de faire un petit sudo apt update suivi par cette commande

```
root@JacktopV2:~# sudo apt install isc-dhcp-server
```

Après l'installation du paquet nous allons par la suite modifier ce fichier /etc/dhcp/dhcpd.conf et mettre en point cette configuration:

```
default-lease-time 600;
max-lease-time 7200;
authoritative;

subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.100 192.168.0.200;
    option routers 192.168.0.1;
    option subnet-mask 255.255.255.0;
    option domain-name-servers 192.168.0.1, 8.8.8.8;
    option domain-name "localdomain";
}
```

Après cela, nous allons conséquemment redémarrer le service DHCP

```
root@JacktopV2:~# sudo systemctl restart isc-dhcp-server
```

Et nous assurer que le service démarre dès le boot de la machine:

```
root@JacktopV2:~# sudo systemctl enable isc-dhcp-server
Synchronizing state of isc-dhcp-server.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable isc-dhcp-server
```

IV/ Mise en place du pare feu

Tout d'abord, il faut installer l'outil comme pour chaque procédure. Cette fois-ci, ça sera ufw (Uncomplicated Firewall).

```
root@JacktopV2:~# sudo apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.1-4ubuntu0.1).
0 upgraded, 0 newly installed, 0 to remove and 47 not upgraded.
```

Après l'installation nous allons activer ufw avec ceci

```
root@JacktopV2:~# sudo ufw enable
Firewall is active and enabled on system startup
```

Et ensuite donner/interdire accès à certains ports;

```
root@JacktopV2:~# sudo ufw allow ssh
Rule added
Rule added (v6)
root@JacktopV2:~# sudo ufw allow http
Skipping adding existing rule
Skipping adding existing rule (v6)
root@JacktopV2:~# sudo ufw allow https
Rule added
Rule added (v6)
```

Sur l'image ci-dessus, nous avons donc donné accès au port 22 (SSH), port 80 (HTTP) et au port 443 (HTTPS).

Par la suite nous allons bien vérifier le statut et les règles du pare feu dans notre machine avec sudo ufw status verbose:

```
root@JacktopV2:~# sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), deny (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW IN    Anywhere
22/tcp                     ALLOW IN    Anywhere
443                        ALLOW IN    Anywhere
80/tcp (v6)                ALLOW IN    Anywhere (v6)
22/tcp (v6)                ALLOW IN    Anywhere (v6)
443 (v6)                   ALLOW IN    Anywhere (v6)
```

Constat:

Nous avons constater que pour que l'url https fonctionne il est obligatoire de changer quelques réglages de l'Apache d'XAMPP plus précisément dans httpd-vhosts.conf or httpd-ssl.conf cependant lorsque nous le modifions avec les informations reliées au chemin absolu de notre dossier; apache cesse de fonctionner.

```
<VirtualHost *:80>
    DocumentRoot "C:\xampp\htdocs\xampp\Reservation_Plane-Infra-Dev"
    ServerName reservation-flights.local
    <Directory "C:/xampp/htdocs/reservation-flights">
        Options Indexes FollowSymLinks Includes ExecCGI
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>

<VirtualHost *:443>
```

```
    DocumentRoot "C:\xampp\htdocs\xampp\Reservation_Plane-Infra-Dev"
    ServerName reservation-flights.local

    SSLEngine on
    SSLCertificateFile "C:/xampp/apache/conf/ssl.crt/selfsigned.crt"
    SSLCertificateKeyFile "C:/xampp/apache/conf/ssl.key/selfsigned.key"

    <Directory "C:/xampp/htdocs/reservation-flights">
        Options Indexes FollowSymLinks Includes ExecCGI
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```