

## Projet E5 – Mise en place d'un intranet avec certificat auto-signé

### I. Préparation de l'environnement

#### 1) Création de la VM

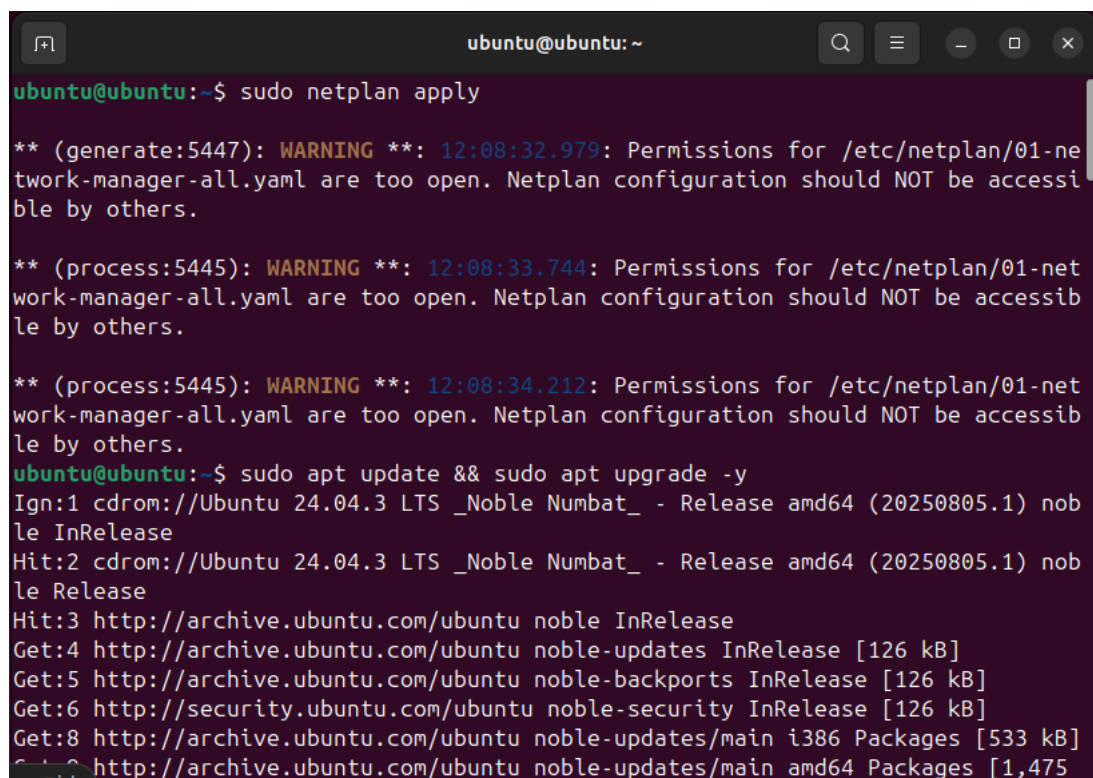
Voici la configuration qu'on va utiliser :

- 2 vCPU, 4 Go RAM, 20 Go disque dur
- OS : Ubuntu Server 22.04

#### 2) Installation du système

Avec VMware Station, on a installé l'ISO Ubuntu avec comme nom de machine « intranet »

Avec la commande « **sudo netplan apply** » et **sudo apt update && sudo apt upgrade -y** », on va se mettre en mode administrateur puis mettre à jour le système.



```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ sudo netplan apply  
** (generate:5447): WARNING **: 12:08:32.979: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.  
  
** (process:5445): WARNING **: 12:08:33.744: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.  
  
** (process:5445): WARNING **: 12:08:34.212: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.  
ubuntu@ubuntu:~$ sudo apt update && sudo apt upgrade -y  
Ign:1 cdrom://Ubuntu 24.04.3 LTS _Noble Numbat_ - Release amd64 (20250805.1) noble InRelease  
Hit:2 cdrom://Ubuntu 24.04.3 LTS _Noble Numbat_ - Release amd64 (20250805.1) noble Release  
Hit:3 http://archive.ubuntu.com/ubuntu noble InRelease  
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]  
Get:6 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:8 http://archive.ubuntu.com/ubuntu noble-updates/main i386 Packages [533 kB]  
Get:9 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,475
```

```
ubuntu@ubuntu: ~  
libgstreamer-plugins-base1.0-0 libgstreamer-plugins-good1.0-0 libgtk-4-1  
libgtk-4-bin libgtk-4-common libgtk-4-media-gstreamer libipa-hbac0t64  
libjavascriptcoregtk-4.1-0 libjavascriptcoregtk-6.0-1 libldb2  
libmalcontent-0-0 libnss-sss libopenjp2-7 libpam-modules libpam-modules-bin  
libpam-runtime libpam-sss libpam0g libpoppler-cpp0t64 libpoppler-glib8t64  
libpoppler134 libpython3.12-minimal libpython3.12-stdlib libpython3.12t64  
libsmblclient0 libsqlite3-0 libssl3t64 libsss-certmap0 libsss-idmap0  
libsss-nss-idmap0 libtiff6 libudisks2-0 libwbclient0 libwebkit2gtk-4.1-0  
libwebkitgtk-6.0-4 libxatracker2 libxml2 linux-firmware  
linux-generic-hwe-24.04 linux-headers-generic-hwe-24.04  
linux-image-generic-hwe-24.04 linux-libc-dev linux-tools-common locales  
mesa-libgallium mesa-vulkan-drivers openssh-client openssl openvpn  
poppler-utils powermgmt-base python3-software-properties python3-sss  
python3.12 python3.12-minimal samba-ls simple-scan  
software-properties-common software-properties-gtk sssd sssd-ad  
sssd-ad-common sssd-common sssd-ipa sssd-krb5 sssd-krb5-common sssd-ldap  
sssd-proxy systemd-hwe-hwdb tecla ubuntu-drivers-common udisks2 vim-common  
vim-tiny xserver-xorg-video-nouveau xserver-xorg-video-vesa xxd  
134 upgraded, 8 newly installed, 0 to remove and 5 not upgraded.  
68 standard LTS security updates  
Need to get 947 MB of archives.  
After this operation, 442 MB of additional disk space will be used.  
E: You don't have enough free space in /var/cache/apt/archives/.  
ubuntu@ubuntu: ~$
```

## II. Installation du serveur web

### Installer Nginx

Avec la commande « **sudo apt update && sudo apt install nginx -y** », nous allons installer Nginx pour installer le serveur web.

Pour savoir si le service tourne, on va utiliser la commande « **systemctl status nginx** »

```
ubuntu@ubuntu: ~$ sudo apt update && sudo apt install nginx -y  
Ign:1 cdrom://Ubuntu 24.04.3 LTS _Noble Numbat_ - Release amd64 (20250805.1) noble InRelease  
Hit:2 cdrom://Ubuntu 24.04.3 LTS _Noble Numbat_ - Release amd64 (20250805.1) noble Release  
Hit:3 http://archive.ubuntu.com/ubuntu noble InRelease  
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease  
Hit:5 http://archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:7 http://archive.ubuntu.com/ubuntu noble-backports InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
139 packages can be upgraded. Run 'apt list --upgradable' to see them.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  nginx-common  
Suggested packages:  
  fcgiwrap nginx-doc  
The following NEW packages will be installed:  
  nginx nginx-common  
0 upgraded, 2 newly installed, 0 to remove and 139 not upgraded.  
Need to get 564 kB of archives.  
After this operation, 1,596 kB of additional disk space will be used.  
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.5 [43.4 kB]  
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.5 [520 kB]
```

```

ubuntu@ubuntu:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: en>
   Active: active (running) since Wed 2025-10-01 12:13:38 UTC; 16s ago
     Docs: man:nginx(8)
   Process: 7008 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_proce>
   Process: 7009 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (c>
  Main PID: 7039 (nginx)
    Tasks: 3 (limit: 2197)
   Memory: 4.0M (peak: 6.7M)
      CPU: 180ms
   CGroup: /system.slice/nginx.service
           └─7039 "nginx: master process /usr/sbin/nginx -g daemon on; master>
             └─7041 "nginx: worker process"
               └─7042 "nginx: worker process"

Oct 01 12:13:38 ubuntu systemd[1]: Starting nginx.service - A high performance >
Oct 01 12:13:38 ubuntu systemd[1]: Started nginx.service - A high performance w>
lines 1-17/17 (END)

```

### III. Création du site intranet

#### 1) Créer le dossier web

Après avoir installer Nginx, on va devoir créer le dossier web avec la commande :

```
sudo mkdir -p /var/www/intranet
```

```
sudo chown -R www-data:www-data /var/www/intranet
```

```

ubuntu@ubuntu:~$ sudo mkdir -p /var/www/intranet
ubuntu@ubuntu:~$ sudo chown -R www-data:www-data /var/www/intranet
ubuntu@ubuntu:~$

```

#### 2) Création de la page d'accueil

On va également créer la page d'accueil avec cette commande :

```
echo "<h1>Bienvenue sur l'intranet (Nginx)</h1>" | sudo tee
/var/www/intranet/index.html
```

```

ubuntu@ubuntu:~$ echo "<h1> Bienvenue sur l'intranet Nginx</h1>" | sudo tee /var/www/intranet/index.html
<h1> Bienvenue sur l'intranet Nginx</h1>
ubuntu@ubuntu:~$

```

#### 3) Créer un bloc serveur http

Pour créer un bloc serveur http, on va devoir utiliser cette commande : `sudo nano /etc/nginx/sites-available/intranet`

```

<h1> Bienvenue sur l'intranet Nginx</h1>
ubuntu@ubuntu:~$ sudo nano /etc/nginx/sites-available/intranet

```

Cela ramener à un « bloc note » dans lequel on va devoir écrire la commande ci-dessous :

```

server {
    listen 80;

    server_name intranet.local;

```

```
root /var/www/intranet;
```

```
index index.html;
```

```
location / {
```

```
    try_files $uri $uri/ =404;
```

```
}
```

```
}
```

```
GNU nano 7.2 /etc/nginx/sites-available/intranet *
server {
    listen 80;
    server_name intranet.local;

    root /var/www/intranet;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Un
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^\_ Go To Line M-E Re

Contrôle X pour quitter et sauvegarder avec « Y » puis entrée, on retourne sur bash

On va activer le site avec la commande : `sudo ln -s /etc/nginx/sites-available/intranet /etc/nginx/sites-enabled/`

`sudo nginx -t`

`sudo systemctl reload nginx`

```
ubuntu@ubuntu:~$ sudo ln -s /etc/nginx/sites-available/intranet /etc/nginx/sites-enabled/
ubuntu@ubuntu:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ubuntu@ubuntu:~$ sudo systemctl reload nginx
```

## IV. Sécurisation avec HTTPS

Afin qu'il soit auto-certifié, on va créer un dossier pour le certificat pour qu'il soit généré avec la commande :

```
sudo mkdir /etc/ssl/intranet
```

puis

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
```

```
-keyout /etc/ssl/intranet/intranet.key \
```

```
-out /etc/ssl/intranet/intranet.crt
```

[illegible]

Common Name (CN) : intranet.local

## V. Création d'un bloc serveur HTTPS

Dans bash, nous allons mettre cette commande :

```
sudo nano /etc/nginx/sites-available/intranet-ssl
```

```
ubuntu@ubuntu:~$ sudo nano /etc/nginx/sites-available/intranet-ssl
```

Puis, on va écrire ceci :

```
server {
```

```
listen 443 ssl;
```

```
server_name intranet.local;
```

```
root /var/www/intranet;
```

```
index index.html;
```

```
ssl_certificate /etc/ssl/intranet/intranet.crt;
```

```
ssl_certificate_key /etc/ssl/intranet/intranet.key;
```

```
location / {
```

```
    try_files $uri $uri/ =404;
```

```
}
```

```
}
```

```
GNU nano 7.2 /etc/nginx/sites-available/intranet-ssl *
server {
    listen 443 ssl;
    server_name intranet.local;

    root /var/www/intranet;
    index index.html

    ssl_certificate /etc/ssl/intranet/intranet.crt;
    ssl_certificate_key /etc/ssl/intranet/intranet.key;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Puis, on va l'activer avec cette commande : `sudo ln -s /etc/nginx/sites-available/intranet-ssl /etc/nginx/sites-enabled/`

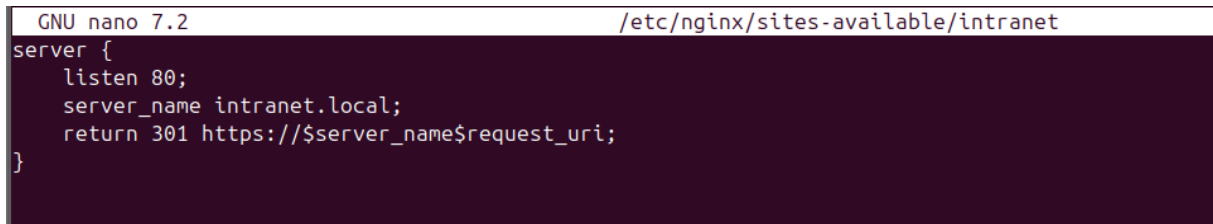
```
sudo nginx -t
```

```
sudo systemctl reload nginx
```

```
ubuntu@ubuntu:~$ sudo ln -s /etc/nginx/sites-available/intranet-ssl /etc/nginx/sites-enabled/
ln: failed to create symbolic link '/etc/nginx/sites-enabled/intranet-ssl': File exists
ubuntu@ubuntu:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ubuntu@ubuntu:~$ sudo systemctl reload nginx
ubuntu@ubuntu:~$
```

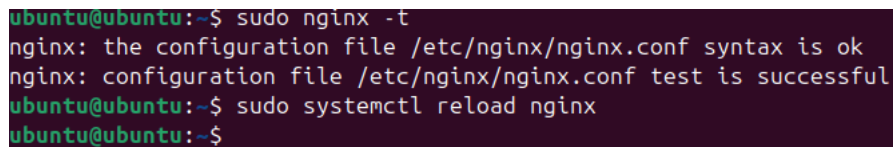
On va ensuite rediriger le HTTP en HTTPS dans bash en modifiant le fichier intranet :

```
server {  
    listen 80;  
    server_name intranet.local;  
    return 301 https://$server_name$request_uri;  
}
```

A screenshot of a terminal window showing the GNU nano 7.2 text editor. The editor is open to the file /etc/nginx/sites-available/intranet. The content of the file is a server block configuration: server {, listen 80;, server\_name intranet.local;, return 301 https://\$server\_name\$request\_uri;}. The cursor is at the end of the first line.

```
GNU nano 7.2 /etc/nginx/sites-available/intranet  
server {  
    listen 80;  
    server_name intranet.local;  
    return 301 https://$server_name$request_uri;  
}
```

On recharge Nginx :

A screenshot of a terminal window showing the execution of several commands to test and reload Nginx. The commands and their outputs are: 'sudo nginx -t' which returns 'nginx: the configuration file /etc/nginx/nginx.conf syntax is ok' and 'nginx: configuration file /etc/nginx/nginx.conf test is successful'; 'sudo systemctl reload nginx' which returns no output; and a final prompt 'ubuntu@ubuntu:~\$'.

```
ubuntu@ubuntu:~$ sudo nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful  
ubuntu@ubuntu:~$ sudo systemctl reload nginx  
ubuntu@ubuntu:~$
```

### Test du site

On va accéder à : <https://intranet.local> et une alerte de sécurité apparaît (normal avec auto-signé). Après validation, la page intranet s'affiche en HTTP

