Ce projet vise à mettre en place un serveur web avec NGINX sur Ubuntu server + la sécurité SSL :

1. **Présentation de Nginx**

Nginx (prononcé "engine x") est un serveur web open source très performant qui agit également comme proxy inverse, serveur de messagerie et load balancer. Développé par Igor Sysoev et lancé en 2004, il est réputé pour sa stabilité, ses riches fonctionnalités, sa faible consommation de ressources et sa capacité à gérer un grand nombre de connexions simultanées.

2. **Installation Ubuntu Server :**

Nous avions décidé d'évoluer avec le serveur Ubuntu comme indiquer sur le projet de ce fait nous devons d'abord être en mode super utilisateur pour commencer la configuration.

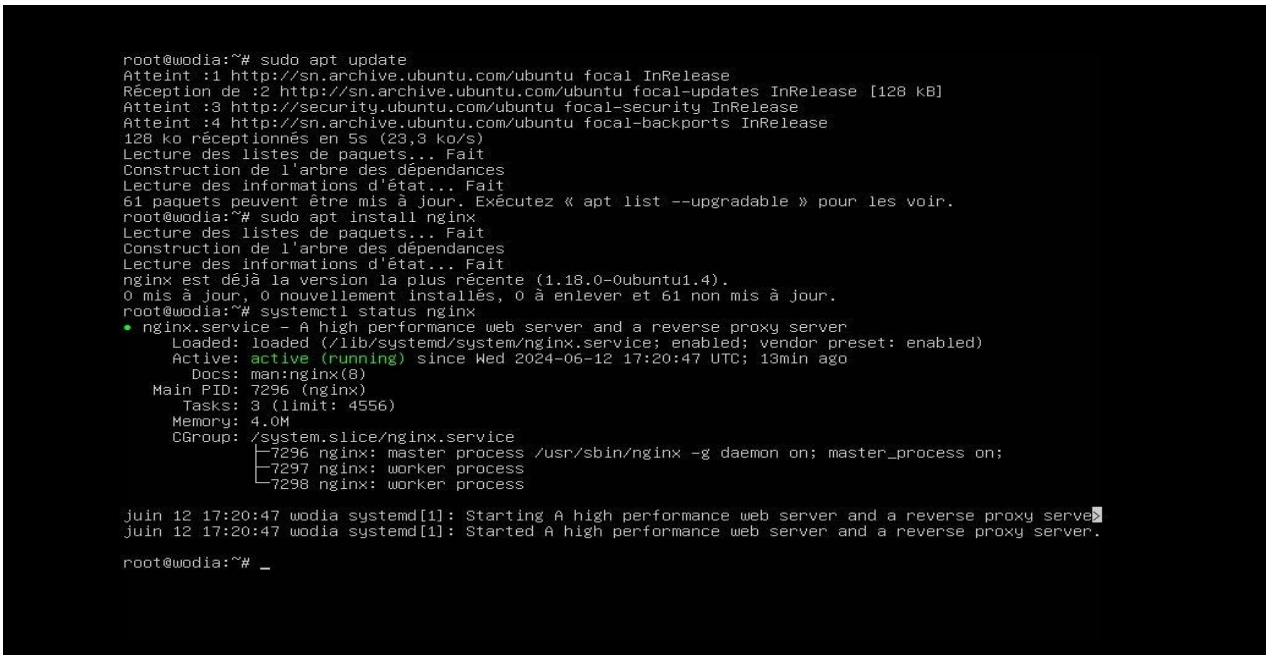
J'ai utilisé la commande : (sudo su) pour me mettre super utilisateur puis mettre mon mot de passe

3. **Installation de NGINX :**

On a commencé à faire la mise à jour des packages avec la commande : sudo apt update

Puis l'installation de nginx avec la commande : sudo apt Install nginx

Et enfin vérifier que Nginx est bien installé et en cours d'exécution avec la commande : systemctl status nginx. Vous voyez le message indiquant que le service est actif(running).



4. **<u>Configuration de NGINX :</u>** Commençons par déterminer les répertoires de configurations.

```
root@wodia:~# cd /etc/nginx/
root@wodia:/etc/nginx# ls
conf.d          koi-utf     modules-available  proxy_params     sites-enabled  uwsgi_params
fastcgi.conf    koi-win     modules-enabled    scgi_params      snippets       win-utf
fastcgi_params  mime.types  nginx.conf         sites-available  ssl
root@wodia:/etc/nginx#
```

Voici en quelques sortes les fichiers et répertoires de configurations se trouvant dans nginx, Les principaux fichiers de configuration de Nginx se trouvent dans le répertoire `/etc/nginx/`. Les fichiers importants incluent :

- /etc/nginx/ : Répertoire principal de configuration
- /etc/nginx/nginx.conf : Fichier de configuration principal
- /etc/nginx/sites-available/ : Répertoire contenant les fichiers de configurations des sites disponibles.
- /etc/nginx/sites-enabled/ : Répertoire contenant les configurations des sites activés

4.1 **Création d'un Nouveau site Web** : Créez un fichier de configuration pour votre site dans /etc/nginx/sites-availables/ :

Avec la commande : sudo nano /etc/nginx/sites-availables/example.com Puis on ajoute la configuration de base

```
  GNU nano 4.8                                    example.com
  server {
    listen 80;
    server_name wodia.com www.wodia.com;

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

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

dans ce fichier.

Ensuite, créé un répertoire racine du site et ajouter un fichier index.html dans /var/www/, pour mon site d'où avec l'arborescence on remarque le cheminement tout en affichant le contenu suivant à index.html.

```
root@wodia:~# cd /var/www/
root@wodia:/var/www# ls
html  mon_site
root@wodia:/var/www# cd mon_site/
root@wodia:/var/www/mon_site# ls
html
root@wodia:/var/www/mon_site# cd html/
root@wodia:/var/www/mon_site/html# ls
index.html
root@wodia:/var/www/mon_site/html# cat index.html
  <!DOCTYPE html>
  <html>
  <head>
        <title>Bienvenue sur Mon site!</title>
  </head>
  <body>
        <h1>Succes! Le serveur Nginx  fonctionne!</h1>
  </body>
  </html>
root@wodia:/var/www/mon_site/html#
```

Puis valider la configuration en créant un lien symbolique dans /etc/nginx/sites-available/

a
/etc/nginx/sites-enabled/ avec la commande :

Sudo ln -s /etc/nginx/sites-available/example.com /etc/nginx/sites-enabled/

Après cela, Nginx commencera à servir le site example.com selon les configurations spécifiées dans
/etc/nginx/sites-enabled/

```
root@wodia:/etc/nginx/sites-available# cat example.com
   server {
     listen 80;
     server_name wodia.com www.wodia.com;


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

            location / {
               try_files $uri $uri/ =404;
        }
}
root@wodia:/etc/nginx/sites-available# sudo ln -s /etc/nginx/sites-available/example.com /etc/nginx/
sites-enabled/
root@wodia:/etc/nginx/sites-available# cat /etc/nginx/sites-enabled/example.com
   server {
     listen 80;
     server_name wodia.com www.wodia.com;


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

            location / {
               try_files $uri $uri/ =404;
        }
}
root@wodia:/etc/nginx/sites-available#
```

Après on utilisera cette commande nginx (sudo systemctl restart) qui indique à systemd de recharger la configuration de Nginx sans arrêter le service. Cela permet à Nginx de prendre en compte les modifications apportées à ses fichiers de configuration, y compris les nouveaux sites activés.

```
root@wodia:/etc/nginx/sites-available# sudo systemctl restart nginx
root@wodia:/etc/nginx/sites-available#
```

5. Passons à la vérification du site chez le client Ubuntu :

Commençons par leur mettre dans un même réseau Lan : en quelques sortes on a essayé de leur mettre dans le même réseau le Serveur Ubuntu et le système Ubuntu dans un réseau : 192.168.20.0 /24. Puis vérifier s'il communique et voilà qu'ils communiquent.

```
root@wodia:~# cd /etc/netplan/
root@wodia:/etc/netplan# ls
00-installer-config.yaml  00-installer-config.yaml.nat
root@wodia:/etc/netplan# mv 00-installer-config.yaml 00-installer-config.yaml.lan
root@wodia:/etc/netplan# mv 00-installer-config.yaml.nat 00-installer-config.yaml
root@wodia:/etc/netplan# ls
00-installer-config.yaml  00-installer-config.yaml.lan
root@wodia:/etc/netplan# cat 00-installer-config.yaml
 network:
  version:  2
  ethernets:
    ens33:
      dhcp4: no
      addresses:
        - 192.168.20.101/24
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4

root@wodia:/etc/netplan# netplan apply
root@wodia:/etc/netplan# [12162.368922] e1000 0000:02:01.0 ens33: Reset adapter

root@wodia:/etc/netplan# netplan apply
root@wodia:/etc/netplan# ping 192.168.20.200
PING 192.168.20.200 (192.168.20.200) 56(84) bytes of data.
64 bytes from 192.168.20.200: icmp_seq=1 ttl=64 time=2.10 ms
64 bytes from 192.168.20.200: icmp_seq=2 ttl=64 time=2.13 ms
64 bytes from 192.168.20.200: icmp_seq=3 ttl=64 time=2.05 ms
64 bytes from 192.168.20.200: icmp_seq=4 ttl=64 time=2.11 ms
^C
--- 192.168.20.200 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 2.052/2.096/2.129/0.028 ms
root@wodia:/etc/netplan#
```

5.1 **Le Client Ubuntu** : Juste après ça on va aller sur le site du client vérifier si le site d'essai qu'on a créé peut-être visible dans le navigateur du client : Pour se faire on ira dans son navigateur ou moi j'ai le Firefox pour entrer l'adresse du serveur qui est 192.168.20.101

On peut aussi passer par une autre méthode tout en restant sur le serveur avec la commande CURL.

Pour ce cas, si vous n'avez pas cette commande dans votre package vous pouvez l'installez avec la commande sudo apt Install curl. Pour l'utiliser on tape : curl http://192.168.20.101 Voici le résultat sur la photo.

```
root@wodia:~# curl http://192.168.20.101
 <!DOCTYPE html>
 <html>
 <head>
    <tilte>Bienvenue sur ma page du Projet en NGINX</tile>
 </head>
 <body>
    <h1> Voila en quelques sortes a quoi ressemble le contenu de la page pour juste un debut d'essai
e!</h1>
 </body>
 </html>
root@wodia:~#
```

De cette configuration, on remarque sur le navigateur que le cadenas est marquer ou barrer avec une croix rouge navigateur qui indique tout simplement un avertissement de sécurité, vous incitant à être conscient des risques potentiels liés à l'utilisation d'une connexion non sécurisée. Pour assurer la sécurité de vos données.

Pour éviter cela, il sera préférable d'accéder aux sites web via HTTPS d'où l'objet de l'installation OPENSSL.

6. INSTALLATION OPENSSL :

OpenSSL étant une bibliothèque puissante qui fournit des outils pour la mise en œuvre du protocole SSL/TLS pour sécuriser les communications. Voici comment installer OpenSSL sur un serveur Ubuntu tout en vérifiant la version de notre openssl :

```
root@wodia:~# sudo apt install openssl
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
openssl est déjà la version la plus récente (1.1.1f-1ubuntu2.22).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 61 non mis à jour.
root@wodia:~# openssl version
OpenSSL 1.1.1f  31 Mar 2020
root@wodia:~# _
```

6.1 Etapes d'affichage d'un site en HTTPS avec OPENSSL : Une fois l'installation

terminée, on peut générer une clé privée et un certificat auto signes à l'aide de la commande OpenSSL.

1- Commençons par générer une clé privée et du certificat de demande (CSR)

```
root@wodia:/etc/nginx/ssl# openssl genrsa -out private.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.........+++++
.....................................................................................+++++
e is 65537 (0x010001)
root@wodia:/etc/nginx/ssl# openssl req -new -x509 -key private.key -out certificate.crt -days 365
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]:SN
State or Province Name (full name) [Some-State]:DAKAR
Locality Name (eg, city) []:DAKAR
Organization Name (eg, company) [Internet Widgits Pty Ltd]:wodia.company
Organizational Unit Name (eg, section) []:company
Common Name (e.g. server FQDN or YOUR name) []:wodia.com
Email Address []:wodya2000@gmail.com
root@wodia:/etc/nginx/ssl# ls
certificate.crt  private.key
root@wodia:/etc/nginx/ssl#
```

Dans ce cas de figure on a générer une clé privée avec la commande : openssl genrsa -out private.key 2048

2 - Puis générer un certificat auto signe avec la commande : openssl req -new -x509 -key private.key - out certificate.crt -days 365. Sa génère directement des fichiers de configuration avec certificate.crt et private.key.

```
root@wodia:/etc/nginx/ssl# ls
certificate.crt  private.key
root@wodia:/etc/nginx/ssl# cat certificate.crt
-----BEGIN CERTIFICATE-----
MIIEATCCAumgAwIBAgIUUalt/y/8LygyIjlrk8kidJ7HwPMwDQYJKoZIhvcNAQEL
BQAwgY8xCzAJBgNVBAYTAlNOMQ4wDAYDVQQIDAVEQUtBUjEOMAwGA1UEBwwFREFL
QVIxFjAUBgNVBAoMDXdvZGlhLmNvbXBhbnkxEDAOBgNVBAsMB2NvbXBhbnkxEjAQ
BgNVBAMMCXdvZGlhLmNvbTEiMCAGCSqGSIb3DQEJARYTd29keWEyMDAwQGdtYWls
LmNvbTAeFw0yNDA2MTIyMDU0MTlaFw0yNTA2MTIyMDU0MTlaMIGPMQswCQYDVQQG
EwJTTjEOMAwGA1UECAwFREFLQVIxDjAMBgNVBAcMBURBSOFSMRYwFAYDVQQKDA13
b2RpYS5jb21wYW55MRAwDgYDVQQLDAdjb21wYW55MRIwEAYDVQQDDA13b2RpYS5j
b20xIjAgBgkqhkiG9w0BCQEWE3dvZHlhMjAwMEBnbWFpbC5jb20wggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQDLLSAb/dG61Waj+XKUVNd1mi3HKDQZ+z7I
b8Rs3L8b6wpqjvjp9bN7zOkeSNqkcv2DlVSMdu3UKVM8PwI8uUeeAGAACdrkaeOX
DmcRJZA7+AMcEk7RRUe8t41lsgIzPDFDouNamUJCIQF8OW5nJsPKgNuBDtlV3xAd
dHQkxDnGEg6TPSamJHl9a8JlYs5qqUuYVx3oy3sFNEtz1+1H2/33dfXYim2hPS5d
yP2qjPu3WJJ8eOAMuaaxMGZknOtfty7yyGcdokEeJJpUh1V1mmKOPTljzPxYVg+I
MntlaKnQSFOHthdetb74uny7gdT2+YBkhPZoNv3T6zL6alVXBgSPAgMBAAGjUzBR
MB0GA1UdDgQWBBQq+PyO2GS/R1/5rMqCtJJFSYQciDAfBgNVHSMEGDAWgBQq+PyO
2GS/R1/5rMqCtJJFSYQciDAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUA
A4IBAQB3E4Mum092Rf27ntJH1mjLrarcgMcMJaLhpDMSF8RzaOqHZWUoQxkmMOy+
obF9HuWX2nvmTAkYXXbrtPPNgaKTgO2Fo2jL8oSyOjUK7TvW3ZsqT2VkIOripPYK
yAVAcON4unesE2avgOGM1ElPTL1N+ngYBxO1okqW1fVBcrgqdCLf/93WkAz/bUgs
jzOToKpu/ag7lvB/wNfTM6NgvtIf8LXCZPIOiM8rNA+SUkM+tJdDUS3jvY7BWzY+
u5PsLHyB5wTl9FxIygQONnWtuZjLFIXWBSrhfNKdUzYDVYBYAXch+bKoOLf8dzIR
gV3+HUwkBXDCKctU/qyYqCm7Zdh/
-----END CERTIFICATE-----
root@wodia:/etc/nginx/ssl# _
```

```
root@wodia:/etc/nginx/ssl# ls
certificate.crt  private.key
root@wodia:/etc/nginx/ssl# cat private.key
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAyyOgG/3RupVmo/lylFTXd2otxygOGfs+yG/EbNy/G+skao74
6fWze8zpHkjapHL9g5VUjHbt1ClTPD8CPL1HngBgAAna5Gnjlw5nESWQO/gDHBJO
OUVHvLeNZbICMzwxQ6LjWp1CQiEBfNFu2ybDyoDbgQ7ZVd8QHXROJMQ5xhIOkzOm
piR5fWvCZWL0aqlLmFcd6Mt7BTRLc9ftR9v993X12IptoTouXcj3qoz7t1iSfHtA
DMGmsTBmZJzrX7cu8shnHaJBH1SaVIdVd2pljjO5Y8z8WFYPiDJ7ZWip0EhdB7YX
XrW++Lp8y4HU9vmA2IT2aDb90+sy+mpVVuYEjwIDAQABAoIBAA8em44ZtFrLNWks
bf91lObWgZOXtVDdpo1a69ciEkkOmgDWBOX8N9AVrp9uUoX8XktHGJSfYRFxxdTs
53C3ZQH4CyzVjumCi8I4+yiGgBZGXYzgO+dM3YnIqggQWCOR7iSXYY3ja8/Z/u9X
8nxQoS9XqX4QKdI/GEbo3J/DznzTxxal7feniF7MXZU5eWKUjTfQsPkg4ni7yFfi
tNpnrOEnSERJOY5ykDhHrWJUNdxop/1yzMHdtdLicLFVsFoQ/x1U2VDOhfz1ULsg
uN6qhEoQJvl2pFMpN3XjjdD8mycA1xEorBKn8qLp7LrwrAfnWpu1P8gmaVFtEm4L
bZ38klECgYEA+a4f/02yKBMR8N3NjgTRTwBEcKqFyR+yaiXGnKrKzl78jp6anuSz
Cz5sU5ZPixqxa9RWIILCpwuWmxmGyGroO4Z4px965SLerjtFhmCMCWk7AKbwYe+T
DcNWYPkxRsH5oTNm1f5hM/6CCDqCX+dl7iSCGdo5WTaByETuka130ecCgYEAOFGq
OiwPANLtZ2nsJSbo87QpGlsZSDL4nIGciRab1VGuE7w/LskfTwNxJ1KsZYD2fzRB
Wkt2dRdV5IA9eA64hxBZRPimjnPvRoST9TkY5rxRYGOKrOWBTsSmKFtln9x+ZGD/
+zsPcvGVHvNAE1gm1cAmX1eMFG4LTrmVOVdCsxkCgYAHP/4fywskTMF1nAsu94bY
Y2q2VupB19EHcgddXzw6V9m/jm1w9bUZO3vLxY6TvHewaN8DXKioTxbVdB1MzoXl
zJbOS4S3CJcsuf3f1Afc8hhLgY1zWpCLksN8V2AN7O4Bt0tiJVvPNbGy6XgFruNv
OdGRTJvjskh8dzvyMPZMuwKBgCAQ8gSSTRzrWu/YNilh2+nniYwtCa87aVIt+hKA
EGy9rkTbbT5dZvhap0hOSYs2Vrst6AFmJ9cGSIprOwpv+aeQtKOROVIkNmi8b6oH
p/WcdOpHCEmk6kqdqpX7u6Dn5+y7oauiY72lyXaCKxLPNusgMMfsSfu7FLKBjrYy
oGnBAoGAaEx+EvuYgFzws4c/zfSFY+sBP/SL9nm3CewHtRS3FONAFj4tTgtBnGYY
SM3kDqxOoz+jltjrQ2P1MFxPdpVxxV7qR3/C1yVYzBI4LJgf2TmVcFVdGerctf3n
NOc96I4s41PouzrAIaVZkOu7CN2ezwrf55XO2/7xKP2cVzDffJO=
-----END RSA PRIVATE KEY-----
root@wodia:/etc/nginx/ssl#
```

3 - Passons au renouvellement de notre configuration NGINX pour utiliser SSL/TLS pour sécuriser les connexions : Dans cette partie, partie j'ai reconfigurer tout en Modifiant le fichier de configuration Nginx de mon site web pour activer HTTPS et utiliser le certificat SSL/TLS que je viens de créer.

```
root@wodia:/etc/nginx# cd sites-enabled/
root@wodia:/etc/nginx/sites-enabled# ls
default  example.com
root@wodia:/etc/nginx/sites-enabled# cat example.com
   server {
     listen 443 ssl;
     server_name wodia.com www.wodia.com;

     ssl_certificate /etc/nginx/ssl/certificate.crt;
     ssl_certificate_key /etc/nginx/ssl/private.key;

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

           location / {
              try_files $uri $uri/ =404;
        }
}
   server {
         listen 80;
         server_name wodia.com www.wodia.com;
         return 301 https://www.wodia.com$host$request_uri;
}
root@wodia:/etc/nginx/sites-enabled#
```

```
root@wodia:/etc/nginx# cd sites-available/
root@wodia:/etc/nginx/sites-available# ls
default  example.com
root@wodia:/etc/nginx/sites-available# cat example.com
   server {
     listen 443 ssl;
     server_name wodia.com www.wodia.com;

     ssl_certificate /etc/nginx/ssl/certificate.crt;
     ssl_certificate_key /etc/nginx/ssl/private.key;

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

           location / {
              try_files $uri $uri/ =404;
        }
}
   server {
         listen 80;
         server_name wodia.com www.wodia.com;
         return 301 https://www.wodia.com$host$request_uri;
}
root@wodia:/etc/nginx/sites-available#
```

4 - Revérifions dans la machine cliente qui est Ubuntu :

Dans cette photo, nous remarquons tout une différence par rapport au cadenas, d'où la présence de la sécurisations https. Le cadenas avec un point d'exclamation dans notre navigateur indique un avertissement de sécurité concernant des problèmes potentiels avec la connexion HTTPS vers un site web avec mon certificat auto-signer.

Conclusion :

Avantages de Nginx

- Haute performance, - Polyvalence en tant que serveur web, proxy inverse et serveur de messagerie - Faible utilisation des ressources
- Grande capacité de gestion des connexions simultanées

Recommandations :

- Configurer des sauvegardes régulières des configurations
- Utiliser des certificats SSL pour sécuriser les connexions