Mise en place d’une infrastructure WEB sécurisé

Sommaire :

I. Configuration des serveurs

1. Serveur Web Debian 11

➔ Configuration du réseau

Un serveur tel qu'il soit doit disposer d'une adresse IP fixe pour ne pas avoir de problème d'accès par la suite, on va commencé par mettre une adresse IP fixe, pour ceci on se rend sur ce répertoire :

```
nano /etc/network/interfaces
```

Avec l'outils 'nano' on peut ouvrir ce fichier texte afin de pouvoir le modifier comme ceci :

```
iface enp18 inet static
address 172.16.159.10 # Adresse IP que vous souhaitez attribuer
netmask 255.255.0.0 # Masque de sous-réseau
gateway 172.16.0.1 # Passerelle par défaut
dns-nameservers 172.16.0.100 # Serveurs DNS
```

2. Serveur certificats Debian 11

➔ Configuration du réseau

Pareil que pour le serveur Web mais pas avec la même adresse IP

```
nano /etc/network/interfaces
```

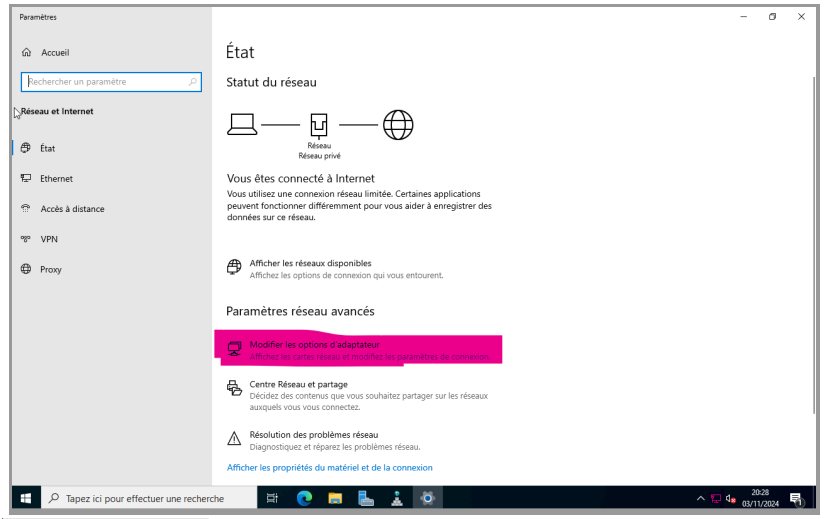
et :

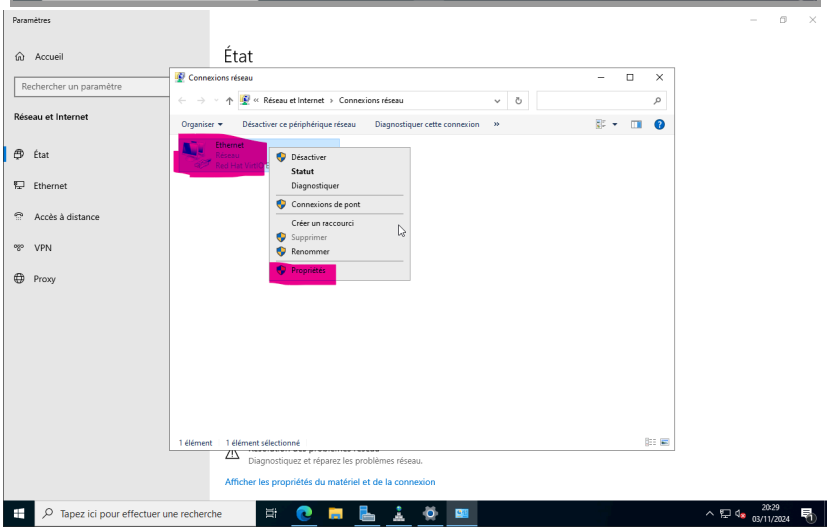
```
iface enp18 inet static
address 172.16.159.200 # Adresse IP
netmask 255.255.0.0 # Masque de sous-réseau
gateway 172.16.0.1 # Passerelle par défaut
dns-nameservers 172.16.0.100 # Serveurs DNS
```

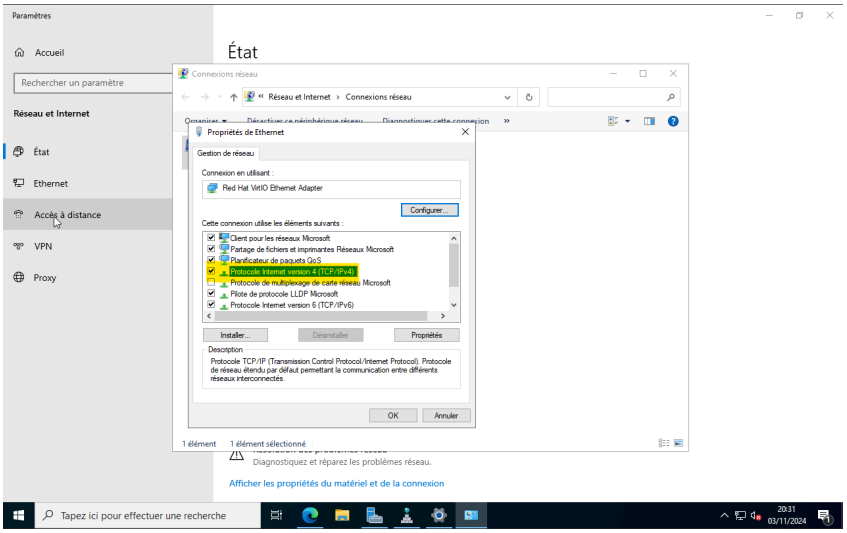
3. Serveur DNS Windows Server 2022

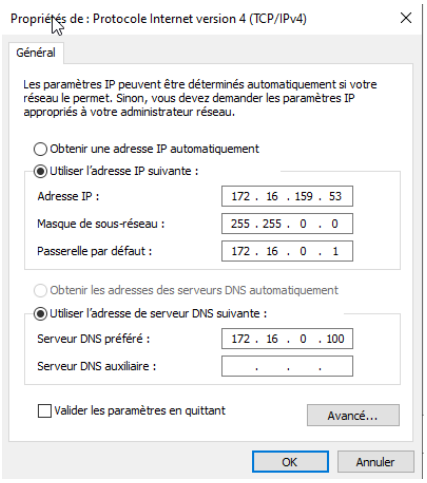
➔ Configuration du réseau

Pour windows server il faut aller chercher dans les sous menu en faisant clic droit sur l'icône réseau puis :







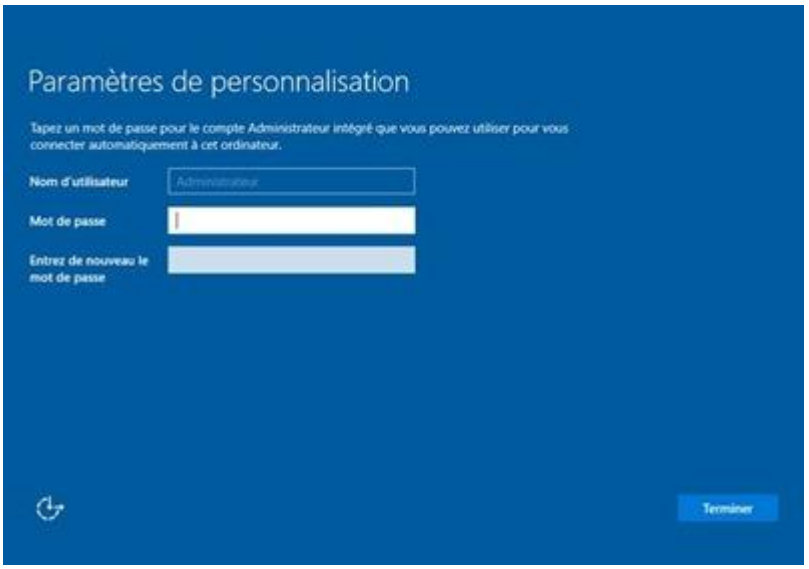


Après avoir fait la configuration réseau de toutes nos machines on peut donc commencer à installer les services qui vont leurs être chacune associées.

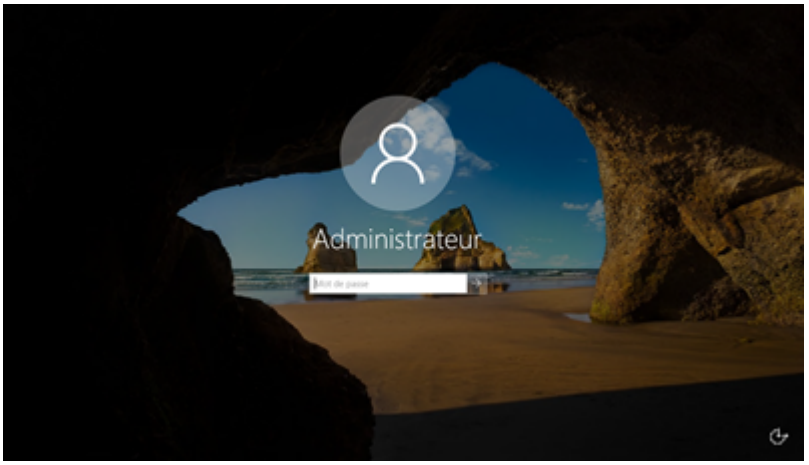
A la fin de l'installation, vous devez définir le mot de passe du compte Administrateur, avec un minimum de complexité :

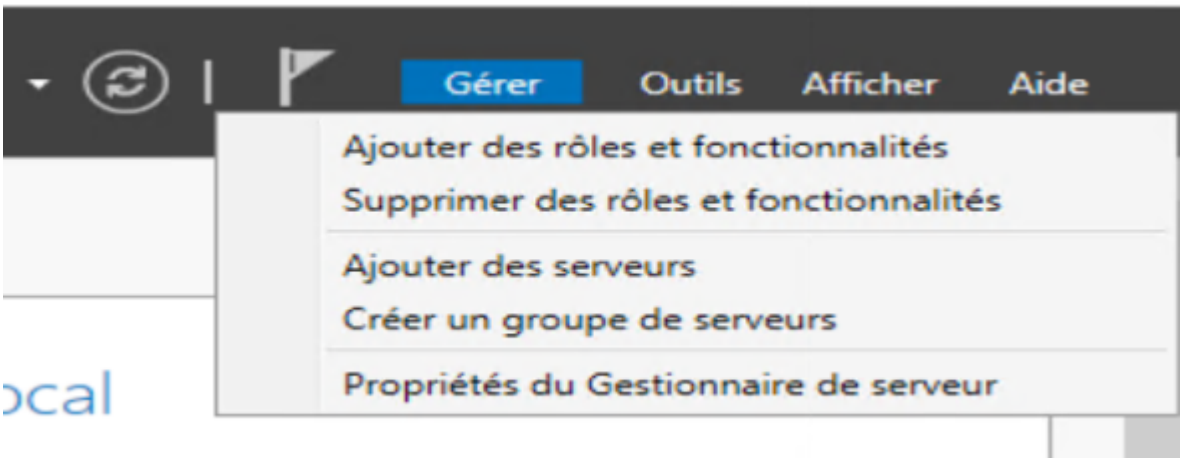
II. Installation des services

1. Installation du service DNS

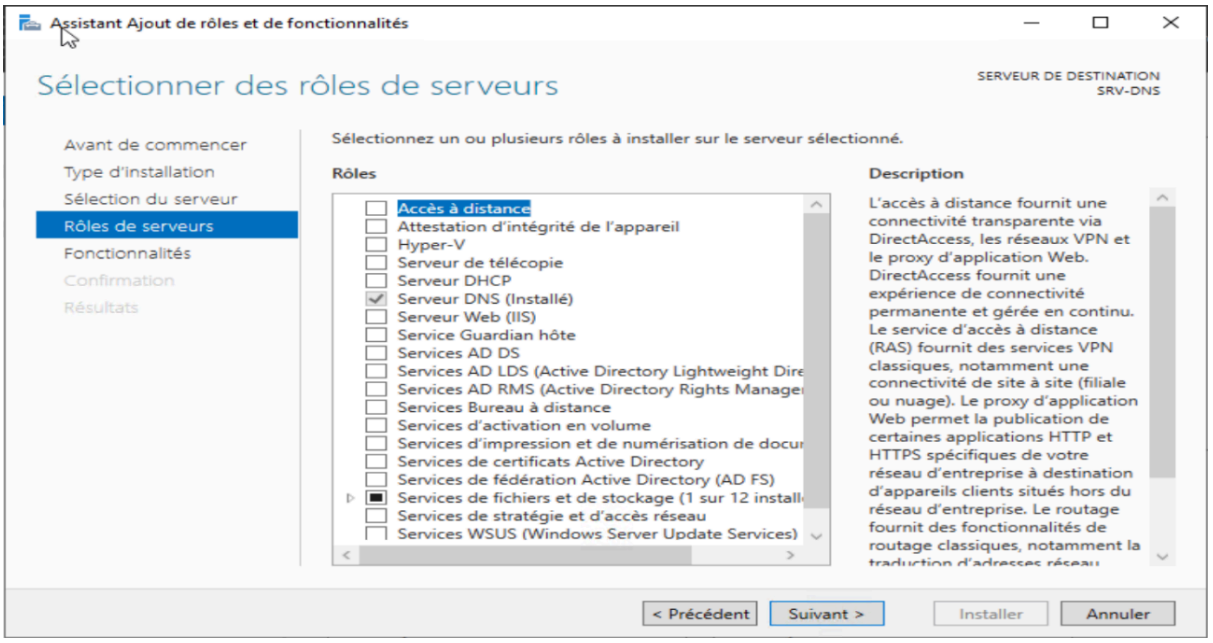


Vous pouvez ouvrir la session en saisissant le mot de passe :

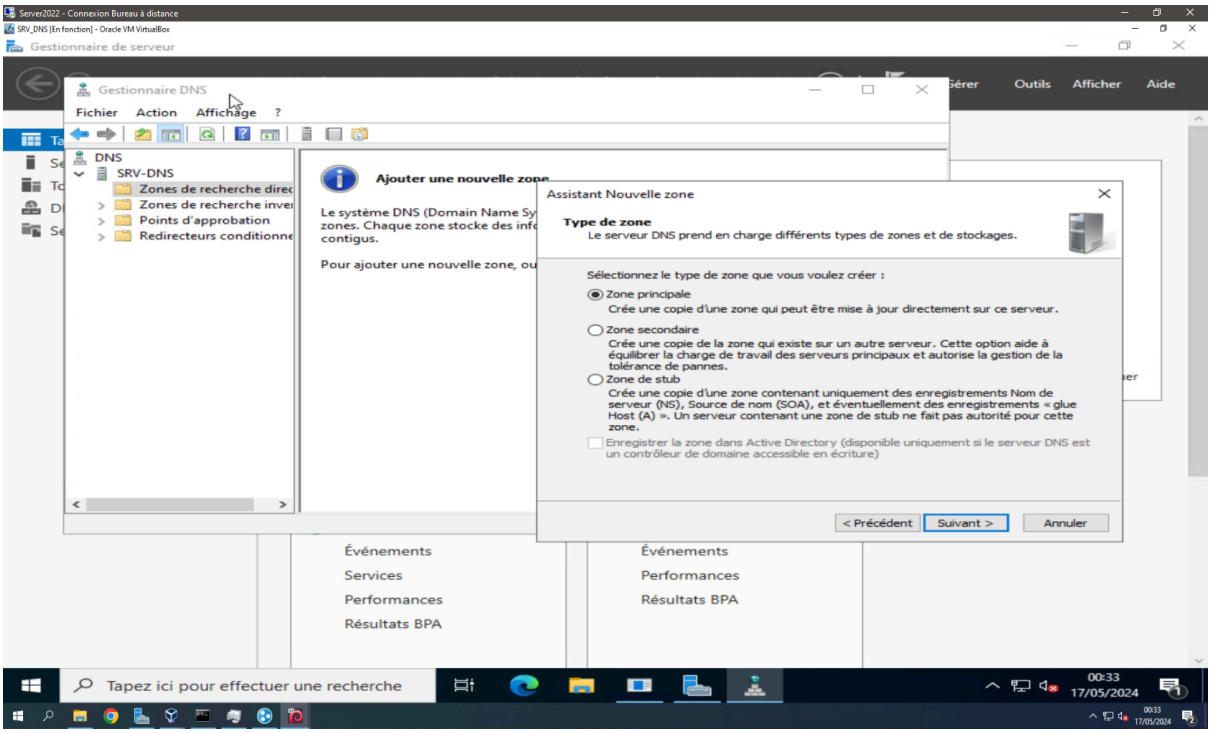




On sélectionne Serveur DNS et on clique sur suivant puis installer :

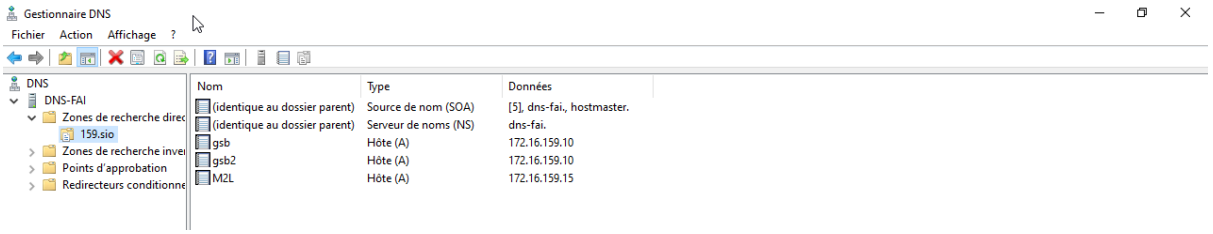


Quand le serveur DNS a finit de s'installer on ouvre son gestionnaire et on créer une nouvelle zone principale de recherche directe :



On y donne le nom que l'on veut, ici nous l'appelons *159.sio*

Et on créer 2 enregistrements DNS pour les deux sites : GSB et GSB2



2. Installation des services serveur web debian 11

A. Service de prise en main à distance | SSH

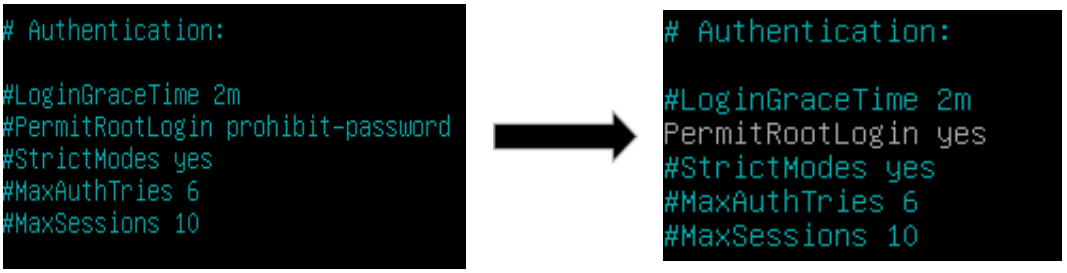
Pour pouvoir prendre en main plus facilement notre serveur et pouvoir faire des copier coller par exemple on va utiliser le protocole SSH afin de prendre la main de notre serveur à distance, pour ceci on va installer openssh :

```
apt-get install openssh-server
```

On va maintenant configurer openssh :

```
nano /etc/ssh/sshd_config
```

Une fois dans le fichier de configuration on va modifier la ligne suivante:



B. Service WEB | Apache

Pour pouvoir héberger des sites web nous avons besoin de apache qui va se charger de mettre en ligne des pages web, pour pouvoir l'installer on exécute la commande :

```
apt-get install apache2
```

On va maintenant créer les dossiers de nos 2 sites web grâce aux commandes:

On va dans le dossier www qui se trouve dans /var/www

```
cd /var/www
```

on y met nos deux dossiers nommée gsb et gsb2

```
mkdir gsb
```

```
mkdir gsb2
```

Par la suite on importera nos pages web via le service de partage de fichiers SFTP.

C. Service SFTP | Service partage de fichier

Pour installer ProFTPD, exécutez la commande suivante :

```
apt-get install proftpd
```

Création des groupes et utilisateurs

Une fois l'installation terminée, nous allons configurer les groupes et utilisateurs pour chaque site :

- **Groupes** :
    - `gsb` pour le site `gsb`
    - `gsb2` pour le site `gsb2`
- **Utilisateurs** :
    - `devgsb` pour le site `gsb`
    - `devgsb2` pour le site `gsb2`

Créez les utilisateurs avec les commandes suivantes :

```
adduser devgsb
```

```
adduser devgsb2
```

Puis, créez les groupes correspondants :

```
addgroup gsb
```

```
addgroup gsb2
```

Ajoutez ensuite les utilisateurs à leurs groupes respectifs :

```
adduser devgsb gsb
```

```
adduser devgsb2 gsb2
```

Installation et configuration d'OpenSSL

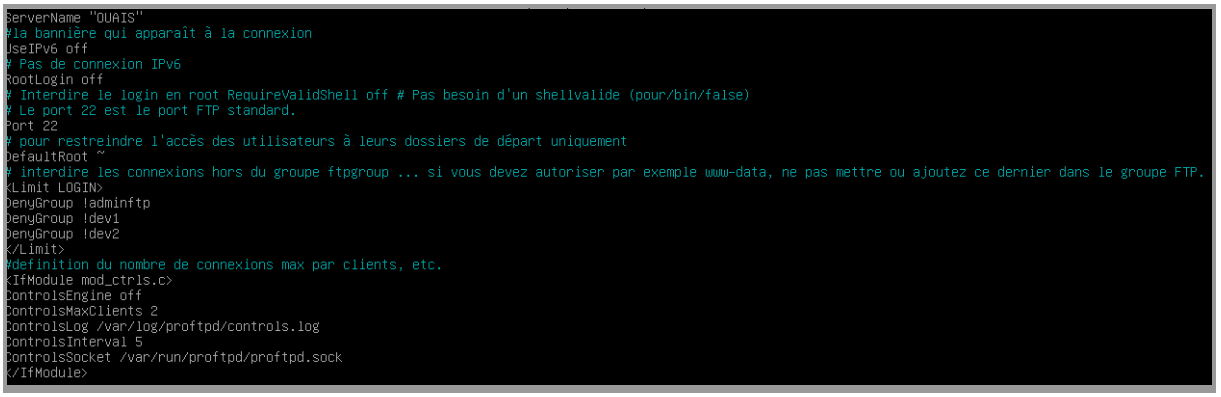
Pour sécuriser les connexions FTP, nous allons utiliser **OpenSSL**, une bibliothèque open-source qui chiffre les données et gère les certificats SSL/TLS.

Pour l'installer, exécutez :

```
apt-get install openssl
```

Ensuite, ajoutez un fichier de configuration pour SFTP :

```
nano /etc/proftpd/conf.d/personnalises.conf
```



A la ligne 9 *'Port'* on remplace “Port 21” par “Port 22” afin de bloquer toutes les tentatives de connection hors SFTP

Copiez la configuration suivante dans le fichier. Enregistrez et quittez **CTRL + X**.

Configurer ProFTPD TLS

Avant de configurer et d'exécuter TLS, il est nécessaire de générer un certificat.

```
mkdir /etc/proftpd/ssl
```

Veuillez exécuter la commande ci-dessous pour générer un certificat auto-signé. Assurez-vous de fournir les informations requises lorsqu'elles sont demandées.

```
openssl req -new -x509 -keyout /etc/proftpd/ssl/proftpd.key.pem -days 365 -nodes -out /etc/proftpd/ssl/proftpd.cert.pem
```

Generating a RSA private key.....+++++.....+++++
writing new private key to '/etc/proftpd/ssl/proftpd.key.pem'

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]:x
State or Province Name (full name) [Some-State]:x
Locality Name (eg, city) []:x
Organization Name (eg, company) [Internet Widgits Pty Ltd]:x
Organizational Unit Name (eg, section) []:x
Common Name (e.g. server FQDN or YOUR name) []:x
Email Address []:x

Ensuite, nous attribuons les permissions aux fichiers de clés. Le fichier "proftpd.key" doit être rendu lisible par le seul utilisateur "root".
Pour sécuriser l'environnement, veuillez exécuter la commande ci-dessous :

```
chmod 600 /etc/proftpd/ssl/proftpd.*
```

Enfin, veuillez ajouter le fichier suivant :

```
vi /etc/proftpd/conf.d/tls.conf
```

Ensuite, veuillez copier et coller le contenu suivant :

```
<IfModule mod_tls.c>

[::]

TLSEngine on

TLSLog /var/log/proftpd/tls.log

TLSProtocol SSLv23

TLSOptions NoCertRequest
AllowClientRenegotiations

TLSRSACertificateFile /etc/proftpd/ssl/proftpd.cert.pem

TLSRSACertificateKeyFile /etc/proftpd/ssl/proftpd.key.pem

TLSVerifyClient off

TLSRequired on

RequireValidShell no

TLSOptions NoSessionReuseRequired

</IfModule>
```

On relance ensuite ProFTP afin d’appliquer les modification :

```
service proftpd restart
```

Configuration du dossier root par défaut de chaque groupe

Chaque groupe doit tomber sur leurs dossier respectif, les développeur de GSB ne doivent pas tomber sur les dossier de GSB2 par exemple et vice versa, pour y parvenir on tappe la commande qui nous servait à configurer le serveur SSH (SFTP fonctionne à l'aide du protocole SSH).

```
nano /ect/ssh/sshd_config
```

```
# override default of no subsystems
Subsystem sftp internal-sftp

Match Group gsb
    ChrootDirectory /var/www/html/gsb
    ForceCommand internal-sftp
    AllowTcpForwarding no
    X11Forwarding no
    PermitTunnel no

Match Group gsb2
    ChrootDirectory /var/www/html/gsb2
    ForceCommand internal-sftp
    AllowTcpForwarding no
    X11Forwarding no
    PermitTunnel no
```

Ces commandes servent à ce que le groupe GSB tombe sur leurs répertoire GSB et interdit tout autre accès autre que leurs dossier GSB et également pour GSB2

D. Service PHP | Interpréteur PHP

Installation du package PHP

Pour pouvoir prendre en charge les pages en .PHP nous avons besoin d'installer le package PHP afin d'installer l'interpréteur PHP. Pour ceci nous allons procéder de la sorte :

```
sudo apt install php
```

Après avoir installé PHP, vous devez redémarrer Apache pour qu'il prenne en compte les changements :

```
sudo systemctl restart apache2
```

E. Service Base de données | SGBD MariaDB

Installation du package MariaDB

Pour installer le package MariaDB on suit ces commandes ci :

```
sudo apt install mariadb-server mariadb-client -y
```

Ensuite nous allons activer le service Mariadb et le démarrer

```
sudo systemctl enable mariadb
sudo systemctl start mariadb
```

Puis nous vérifions que le service est bien démarré et actif :

```
sudo systemctl status mariadb
```

```
● mariadb.service - MariaDB 10.3.34 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2022-04-26 07:58:16 CEST; 2h 28min ago
       Docs: man:mysqld(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 1307 (mysqld)
     Status: "Taking your SQL requests now..."
      Tasks: 30 (limit: 9361)
     Memory: 80.0M
     CGroup: /system.slice/mariadb.service
             └─1307 /usr/sbin/mysqld
```

Configuration de MariaDB

Une fois que ces étapes sont réalisées, nous pouvons configurer Mariadb :

```
mysql_secure_installation
```

Enter current password for root (enter for none):
Change the root password? [Y/n] Y
New password: votre_mdp
Re-enter new password: votre_mdp
Remove anonymous users? [Y/n] Y
Disallow root login remotely? [Y/n] Y
Remove test database and access to it? [Y/n] Y
Reload privilege tables now? [Y/n] Y

Ce message est une série de questions posées par MariaDB lors de la configuration initiale du serveur. Voici ce que chaque question signifie et les réponses recommandées :

1. **Enter current password for root (enter for none) :** Cette question demande le mot de passe actuel du compte root de MariaDB. Si vous venez de configurer MariaDB pour la première fois, il n'y a pas encore de mot de passe défini, donc appuyez simplement sur Entrée pour continuer sans saisir de mot de passe.

2. **Change the root password? [Y/n]** Cette question vous demande si vous voulez changer le mot de passe du compte root de MariaDB. Si vous venez d'installer MariaDB ou si vous n'avez pas encore défini de mot de passe pour le compte root, vous devrez répondre "Y" (oui). Sinon, si vous avez déjà un mot de passe configuré et que vous souhaitez le garder tel quel, répondez "n" (non).

3. **New password** : Si vous avez choisi de changer le mot de passe root, cette question vous demande de saisir le nouveau mot de passe.

4. **Re-enter new password** : Cette question vous demande de saisir à nouveau le nouveau mot de passe pour confirmer.

5. **Remove anonymous users? [Y/n] :** Cette question vous demande si vous souhaitez supprimer les utilisateurs anonymes. Les utilisateurs anonymes n'ont pas besoin d'identifiant ni de mot de passe pour se connecter à la base de données. Il est recommandé de supprimer les utilisateurs anonymes pour des raisons de sécurité. Répondez "Y" (oui).

6. **Disallow root login remotely? [Y/n**] : Cette question vous demande si vous souhaitez empêcher la connexion à distance du compte root. Il est généralement recommandé de désactiver la connexion à distance pour le compte root pour des raisons de sécurité. Répondez "Y" (oui).

7. **Remove test database and access to it? [Y/n]** : Cette question vous demande si vous souhaitez supprimer la base de données de test par défaut. Il est recommandé de supprimer la base de données de test pour des raisons de sécurité. Répondez "Y" (oui).

8. **Reload privilege tables now? [Y/n] :** Cette question vous demande si vous souhaitez recharger les tables de privilèges pour que les modifications apportées prennent effet immédiatement. Répondez "Y" (oui).

En répondant à ces questions, vous configurez initialement la sécurité et les paramètres de votre serveur MariaDB. Assurez-vous de répondre en fonction de vos besoins spécifiques et des bonnes pratiques de sécurité.

F. Service PHPMyAdmin | Application de gestion du SGBD

L'installation de PhpMyAdmin ne s'effectue pas comme un paquet classique, mais plutôt sur le même principe qu'une application web. Il faut que l'on télécharge les sources à partir du site officiel, directement dans le dossier "/tmp" (ou ailleurs) :

```
cd /tmp
```

```
wget https://files.phpmyadmin.net/phpMyAdmin/5.1.3/phpMyAdmin-5.1.3-all-languages.zip
```

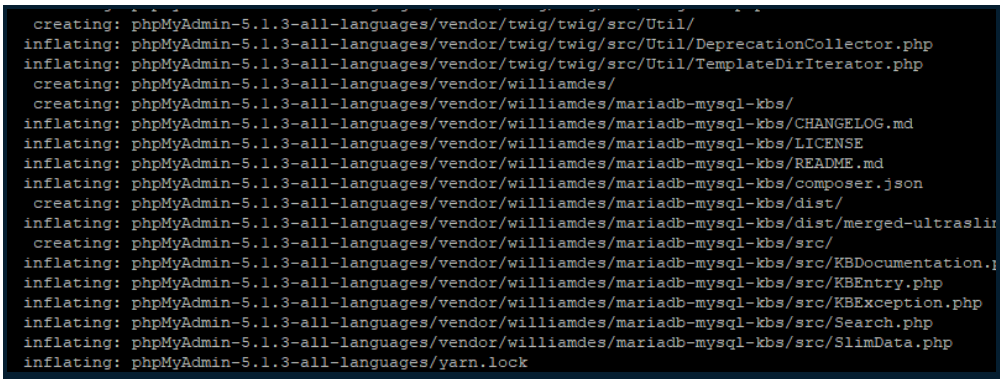
Ensuite, nous devons extraire le contenu de cette archive ZIP avec la commande "unzip". Elle n'est pas installée par défaut sur Debian 11. Nous devons l'installer avec cette commande :

```
sudo apt-get update
```

```
sudo apt-get install unzip
```

Ensuite, on décompresse l'archive ZIP dans le répertoire courant :

```
unzip phpMyAdmin-5.1.3-all-languages.zip
```



On va déplacer le dossier complet vers "/usr/share" dans un nouveau dossier nommé "phpmyadmin". Ce qui donne :

```
sudo mv phpMyAdmin-5.1.3-all-languages /usr/share/phpmyadmin
```

Ensuite, on crée un dossier distinct pour les fichiers temporaires :

```
sudo mkdir -p /var/lib/phpmyadmin/tmp
```

Puis, on attribue les droits sur le dossier racine "*phpmyadmin*" à l'utilisateur associé à Apache (www-data) afin qu'il soit propriétaire.

Nous préciserons le chemin vers le dossier "tmp" dans la configuration de PhpMyAdmin.

```
sudo chown -R www-data:www-data /var/lib/phpmyadmin/
```

PhpMyAdmin est fourni avec un template pour le fichier de configuration, alors on va créer une copie de ce template pour ne pas partir de zéro :

```
cp /usr/share/phpmyadmin/config.sample.inc.php
/usr/share/phpmyadmin/config.inc.php
```

Afin d'utiliser le mode d'authentification basé sur les cookies, nous devons générer une chaîne aléatoire qui est une sorte de passphrase au sein du fichier de configuration. Il doit s'agir d'une chaîne de 32 caractères.

Un cookie permanent stockera l'identifiant sur votre machine tandis que le mot de passe est géré par un cookie temporaire.

On peut générer cette chaîne aléatoire avec la commande suivante :

```
openssl rand -base64 32
```

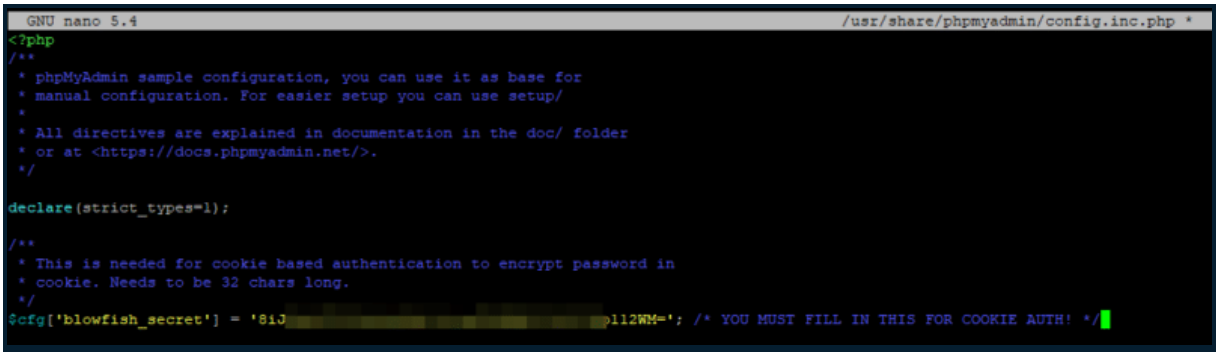
Copiez la valeur retournée en sortie. Nous allons l'insérer dans le fichier de configuration de PhpMyAdmin.

Ouvrez le fichier avec nano :

```
nano /usr/share/phpmyadmin/config.inc.php
```

Collez la valeur au niveau de l'option "*blowfish_secret*", comme ceci :

```
$cfg['blowfish_secret'] = 'deJ8reLGV1cXPyd32454/um/EGWRef/14Jo7tg112WM=';
```



Ensuite, il faut définir un user et un mot de passe que PhpMyAdmin va utiliser pour se connecter à sa base de données et stocker ses données.

Pour cela, il y a deux options à décommenter et modifier pour éviter d'avoir les valeurs par défaut :

```
$cfg['Servers'][$i]['controluser'] = 'clem';
$cfg['Servers'][$i]['controlpass'] = 'Azerty31';
```

Décommentez les autres options, comme sur l'image ci-dessous.

```
/**
 * phpMyAdmin configuration storage settings.
 */

/* User used to manipulate with storage */
// $cfg['Servers'][$i]['controlhost'] = '';
// $cfg['Servers'][$i]['controlport'] = '';
$cfg['Servers'][$i]['controluser'] = 'pma2022';
$cfg['Servers'][$i]['controlpass'] = 'MotDePasseComplexe';

/* Storage database and tables */
$cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
$cfg['Servers'][$i]['bookmarktable'] = 'pma__bookmark';
$cfg['Servers'][$i]['relation'] = 'pma__relation';
$cfg['Servers'][$i]['table_info'] = 'pma__table_info';
$cfg['Servers'][$i]['table_coords'] = 'pma__table_coords';
$cfg['Servers'][$i]['pdf_pages'] = 'pma__pdf_pages';
$cfg['Servers'][$i]['column_info'] = 'pma__column_info';
$cfg['Servers'][$i]['history'] = 'pma__history';
$cfg['Servers'][$i]['table_uiprefs'] = 'pma__table_uiprefs';
$cfg['Servers'][$i]['tracking'] = 'pma__tracking';
$cfg['Servers'][$i]['userconfig'] = 'pma__userconfig';
$cfg['Servers'][$i]['recent'] = 'pma__recent';
$cfg['Servers'][$i]['favorite'] = 'pma__favorite';
$cfg['Servers'][$i]['users'] = 'pma__users';
$cfg['Servers'][$i]['usergroups'] = 'pma__usergroups';
$cfg['Servers'][$i]['navigationhiding'] = 'pma__navigationhiding';
$cfg['Servers'][$i]['savedsearches'] = 'pma__savedsearches';
$cfg['Servers'][$i]['central_columns'] = 'pma__central_columns';
$cfg['Servers'][$i]['designer_settings'] = 'pma__designer_settings';
$cfg['Servers'][$i]['export_templates'] = 'pma__export_templates';
```

Enfin, ajoutez cette directive pour déclarer le répertoire temporaire (créé précédemment) :

```
$cfg['TempDir'] = '/var/lib/phpmyadmin/tmp';
```

```
/**
 * Directories for saving/loading files from server
 */
$cfg['UploadDir'] = '';
$cfg['SaveDir'] = '';
$cfg['TempDir'] = '/var/lib/phpmyadmin/tmp';
/**
```

Sauvegardez et fermez le fichier.

Avant de créer notre propre compte "admin" distinct pour administrer PhpMyAdmin, on va créer la base de données de l'outil. Pour cela, on va utiliser le script fournit :

```
mysql -u root -p < /usr/share/phpmyadmin/sql/create_tables.sql
```

Ensuite, on va se connecter à l'instance MySQL/MariaDB pour donner les droits sur cette base de données à l'utilisateur "clem" :

```
mysql -u root -p
```

Une fois connecté avec le prompt "mysql>" à l'écran, exécutez les requêtes SQL suivantes :

```
CREATE USER 'clem'@'localhost' IDENTIFIED BY 'Azerty31';
GRANT ALL PRIVILEGES ON phpmyadmin.* TO 'clem'@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Les informations (utilisateur et mot de passe) doivent correspondre aux valeurs définies dans le fichier de configuration.

Création d'un admin pour PhpMyAdmin

Nous allons profiter d'être connecté à la console MySQL pour créer un nouveau compte administrateur qui aura la main sur l'ensemble des bases de données. Nous utiliserons ce compte pour se connecter à PhpMyAdmin.

Voici les requêtes SQL à exécuter pour créer un utilisateur nommé "*adminclem*" avec le mot de passe "*Azerty31*".

```
CREATE USER 'adminclem'@'localhost' IDENTIFIED BY 'Azerty31';

GRANT ALL PRIVILEGES ON *.* TO 'adminclem'@'localhost' WITH GRANT OPTION;

FLUSH PRIVILEGES;

EXIT;
```

Contrairement à l'utilisateur précédent, celui-ci a les droits sur toutes les BDD de l'instant MySQL, d'où le "*.*" dans la requête GRANT.

G. Intégration de PhpMyAdmin à Apache

PhpMyAdmin est installé et configuré mais il nous manque une étape cruciale : la publication de l'application via Apache afin de pouvoir y accéder avec un navigateur.

Nous allons créer un fichier de configuration propre à PhpMyAdmin :

```
sudo nano /etc/apache2/conf-available/phpmyadmin.conf
```

Voici le contenu à intégrer au fichier de configuration (peut-être adapté) :

```
Alias /sgbd /usr/share/phpmyadmin

<Directory /usr/share/phpmyadmin>
  Options SymLinksIfOwnerMatch
  DirectoryIndex index.php

  # Autoriser accès depuis certaines adresses IP / sous-réseau
  Order deny,allow
  Deny from all
  Allow from 172.16.0.0/16

  <IfModule mod_php.c>
    <IfModule mod_mime.c>
      AddType application/x-httpd-php .php
    </IfModule>
    <FilesMatch ".+\.php$">
      SetHandler application/x-httpd-php
    </FilesMatch>

    php_value include_path .
    php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp

    php_admin_value open_basedir
/usr/share/phpmyadmin/:/etc/phpmyadmin/:/var/lib/phpmyadmin/:/usr/share/php/
php-gettext/:/usr/share/php/php-php-gettext/:/usr/share/javascript/:/usr/share/php/
tcpdf/:/usr/share/doc/phpmyadmin/:/usr/share/php/phpseclib/

    php_admin_value mbstring.func_overload 0
  </IfModule>

</Directory>

# Désactiver accès web sur certains dossiers
<Directory /usr/share/phpmyadmin/templates>
  Require all denied
</Directory>
<Directory /usr/share/phpmyadmin/libraries>
  Require all denied
</Directory>
<Directory /usr/share/phpmyadmin/setup/lib>
  Require all denied
</Directory>
```

Quelques explications :

- Le fait d'indiquer "*Alias /sgbd /usr/share/phpmyadmin*" cela signifie qu'il faudra préciser "/sgbd" à la fin de l'URL pour accéder à PhpMyAdmin. Vous pouvez mettre autre chose, mais seulement nous vous recommandons de ne pas mettre "phpmyadmin" afin que ce ne soit pas trop évident.

  Cela est d'autant plus important si votre PhpMyAdmin est accessible en mode public car les cyber-robots scans le web à la recherche d'interface PhpMyAdmin.

- On bloque l'accès aux dossiers "templates", "libraries" et "setup/lib".
- On autorise seulement l'accès à PhpMyAdmin à partir des hôtes connectés au LAN "172.16.0.0" car dans cet exemple, PhpMyAdmin n'est pas exposé publiquement.

Enregistrez le fichier et activez ce fichier de config (qui s'appuie sur le VirtualHost par défaut, mais on pourrait créer un vhost distinct) :

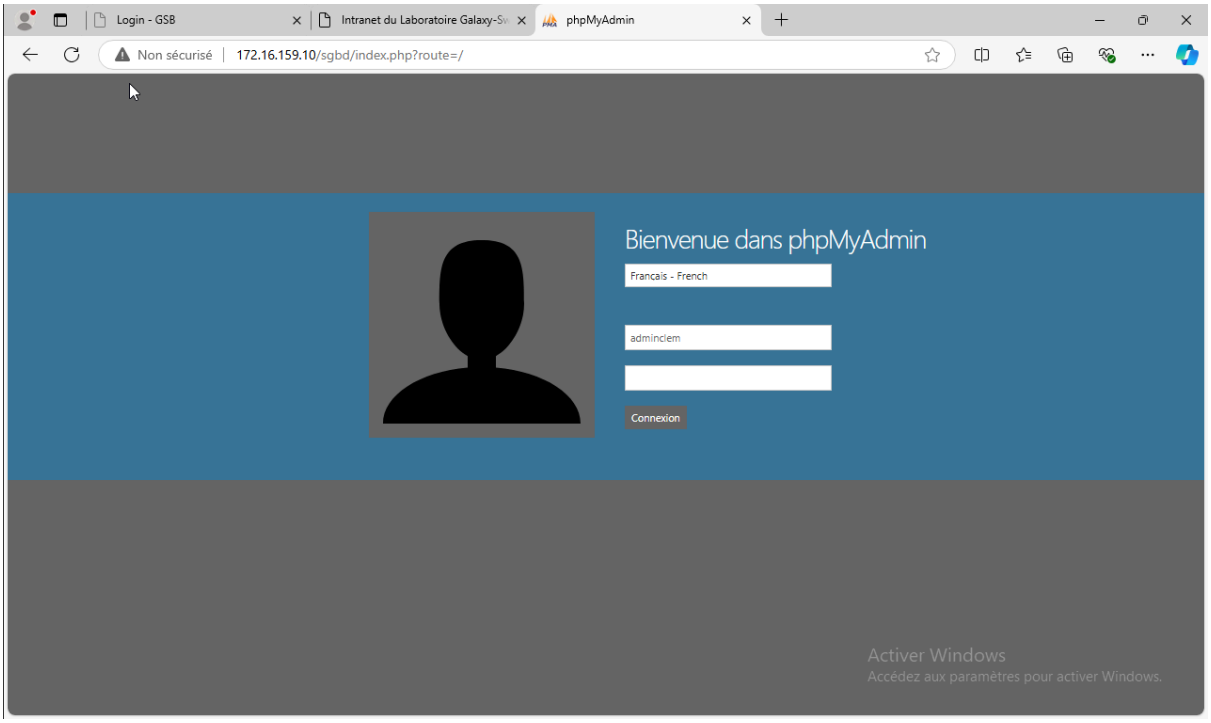
```
sudo a2enconf phpmyadmin.conf
```

Validez la configuration, et si c'est OK rechargez Apache :

```
sudo apachectl configtest
sudo systemctl reload apache2
```

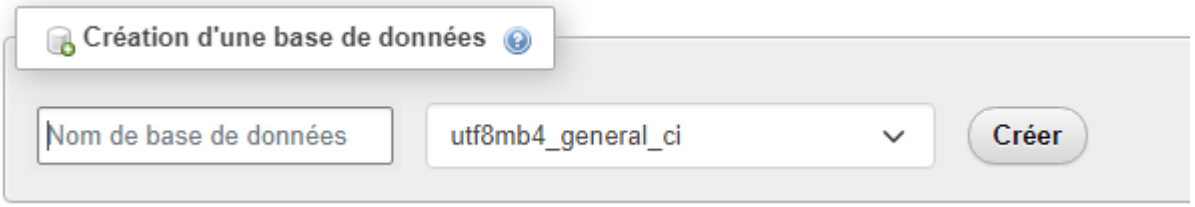
H. Importation des bases de données

On tape l'adresse IP de notre serveur WEB en suivant /sgbd comme l'on a indiqué sur la configuration juste avant :



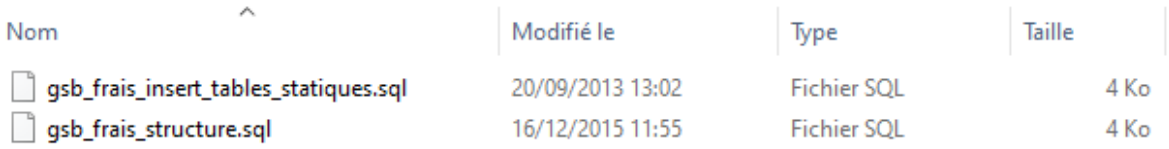
Une fois sur la page d'accueil on va y ajouter une nouvelle base de données

On applique un nom à la base de donnée que l'on doit créer (GSB_frais)



Une fois dans notre nouvelle base on va cliquer sur importer :

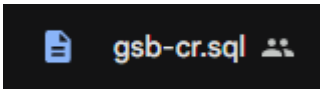
Choisir un fichier, on met les deux ci dessous en commençant par la structure puis les insert table :



Nom	Modifié le	Type	Taille
gsb_frais_insert_tables_statiques.sql	20/09/2013 13:02	Fichier SQL	4 Ko
gsb_frais_structure.sql	16/12/2015 11:55	Fichier SQL	4 Ko

Le principe de commencer par le fichier structure est de d'abord créer la structure de la base pour par la suite ajouter les données dessus.

Pour le premier site GSB du projet de l'an dernier on va utiliser un script tout en un qui va créer la base et en même temps ajouter les données



La dans ce cas là on a pas besoin de créer la base de donnée, on importe juste ce script et tout se génère tout seul.

3. Installation des services serveur autorité de certification Debian 11

A. Installation Logiciel

Installation OpenSSL

Pour pouvoir créer des certificats nous avons besoin de la boîte à outils OpenSSL qui va nous servire à créer des clés et des certificats pour nos site web :

```
apt-get install openssl
```

Par la suite on peut observer que nos sites ne sont pas sécurisés puisque l'on y accède via le port 80 (HTTP).

Ce que l'on va faire c'est créer deux certificats et deux clefs de cryptage pour nos deux site GSB, pour s'y faire on reste sur notre serveur de certificats et on va créer un répertoire que l'on va nommer 'Certificat'

```
mkdir ~/certificat
```

Dedans on va créer deux répertoire, un pour notre site GSB et un autre pour notre site GSB2

```
mkdir /gsb.159.sio
```

```
mkdir /gsb2.159.sio
```

Leur nom correspond également au nom de domaine, c'est plus facile pour nous y repérer.

Génération Clé privé

Pour générer notre clé privé on se rend dans un des deux répertoire créé précédemment et l'on tape cette commande :

```
openssl genpkey -algorithm RSA -out gsb.159.sio.key -pkeyopt
rsa_keygen_bits:4096
```

Créer une Demande de Signature de Certificat (CSR)

La demande de signature de certificat (CSR) contient des informations sur notre domaine et sera utilisée pour générer le certificat.

```
openssl req -new -key gsb.159.sio.key -out gsb.159.sio.csr
```

On vous demandera de fournir des informations, comme :

- **Country Name** : Code du pays (ex. : FR pour la France)
- **State or Province Name** : État ou province
- **Locality Name** : Ville
- **Organization Name** : Nom de votre organisation
- **Organizational Unit Name** : Département (facultatif)
- **Common Name** : **Nom de domaine** de votre site web
- **Email Address** : Adresse email pour les notifications SSL

Générer le Certificat Auto-Signé

Pour générer un certificat auto-signé valable un an (365 jours) :

```
openssl x509 -req -days 365 -in gsb.159.sio.csr -signkey
gsb.159.sio.key -out gsb.159.sio.crt
```

Transférer le Certificat et la Clé sur le Serveur Web

Sur le serveur d'autorité, utilisez **scp** pour transférer le certificat et la clé vers le Serveur Web :

```
scp ~/certificat/gsb.159.sio/gsb.159.sio.crt
root@172.16.159.10:/etc/ssl/certs/
```

```
scp ~/certificat/gsb.159.sio/gsb.159.sio.key
root@172.16.159.10:/etc/ssl/private/
```

Assurez-vous de remplacer `root@172.16.159.10` par l'utilisateur et l'adresse IP ou le nom de domaine de votre Serveur Web.

Configurer le Serveur Web pour Utiliser le Certificat SSL

Sur le **Serveur Web**, configurez votre serveur web Apache pour utiliser le certificat SSL pour les deux sites.

Configuration Apache

1. Ouvrez le fichier de configuration du site pour chaque domaine :

```
sudo nano /etc/apache2/sites-available/gsb.conf
```

Configurer le tel quel :

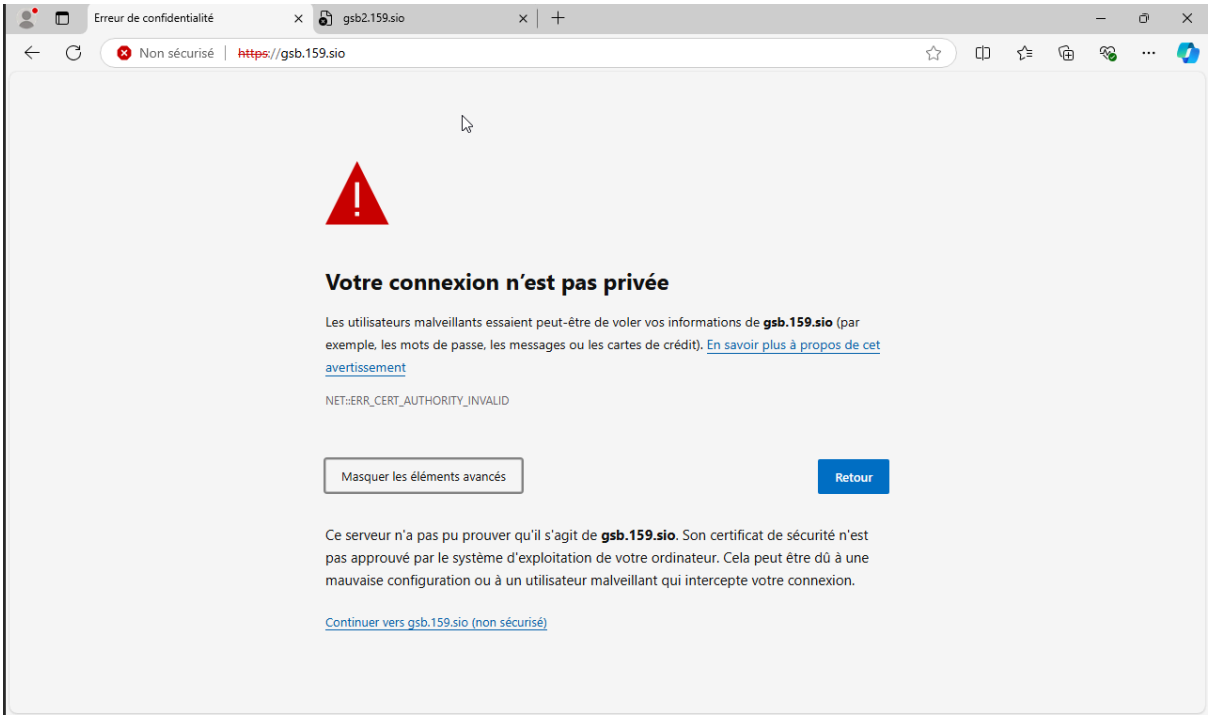
```
<VirtualHost *:443>
    ServerAdmin admin@gsb.159.sio
    ServerName gsb.159.sio

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/gsb.159.sio.crt
    SSLCertificateKeyFile /etc/ssl/private/gsb.159.sio.key

    DocumentRoot /var/www/html/gsb/
    <Directory /var/www/html/gsb/>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
<VirtualHost *:80>
    ServerName gsb.159.sio
    Redirect / https://gsb.159.sio/
<VirtualHost>_
```

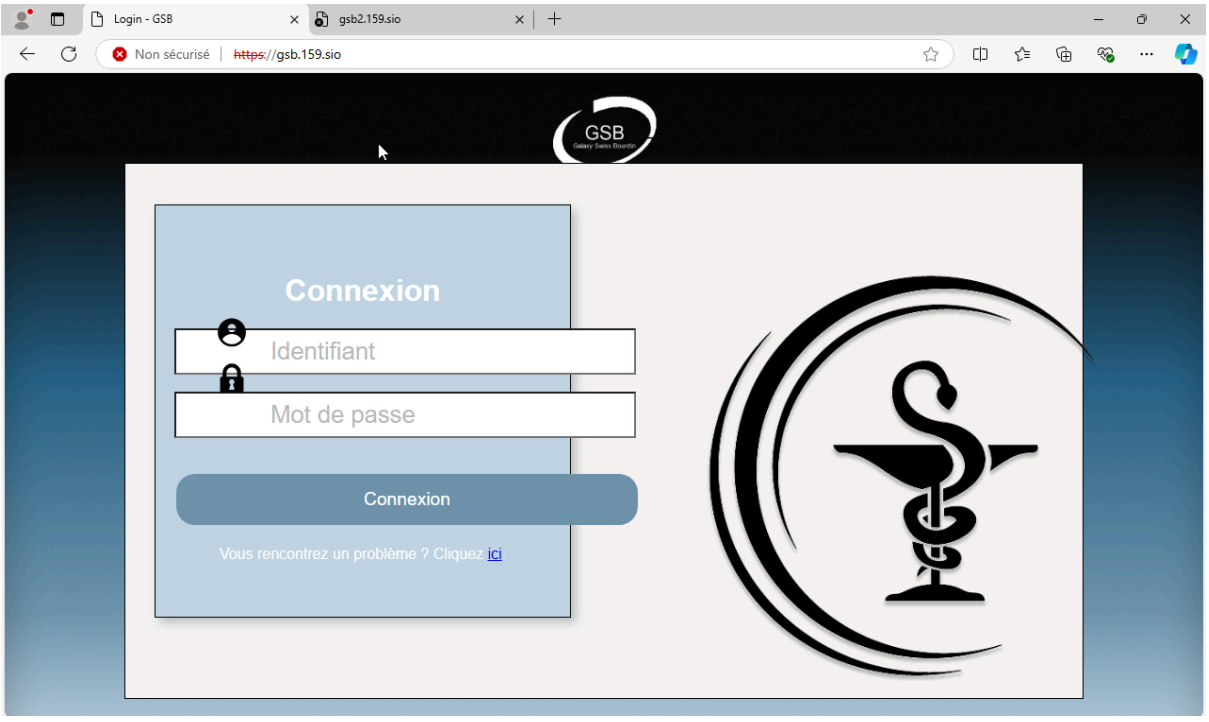
Le premier Bloc <VirutalHost *:443> sert à utiliser le certificats SSL en cas de connection sur le port 443 (HTTPS), en cas de connection vers le port 80 (HTTP) le deuxième bloc <VirutalHost *:80> prend le relais et lui redirige automatiquement vers l'URL du site avec la syntaxe HTTPS afin de bloquer les connexions HTTP.

On répète le même processus pour le deuxième site en changeant les informations nécessaires.

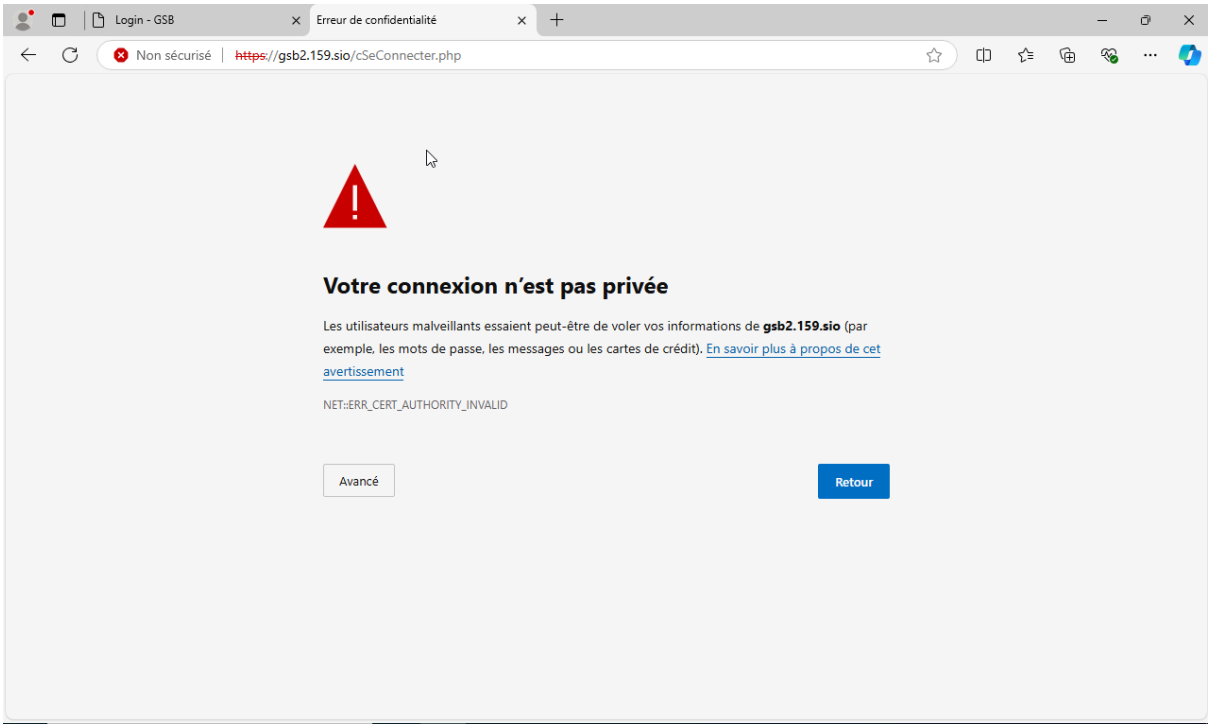
B. Test Certification + DNS

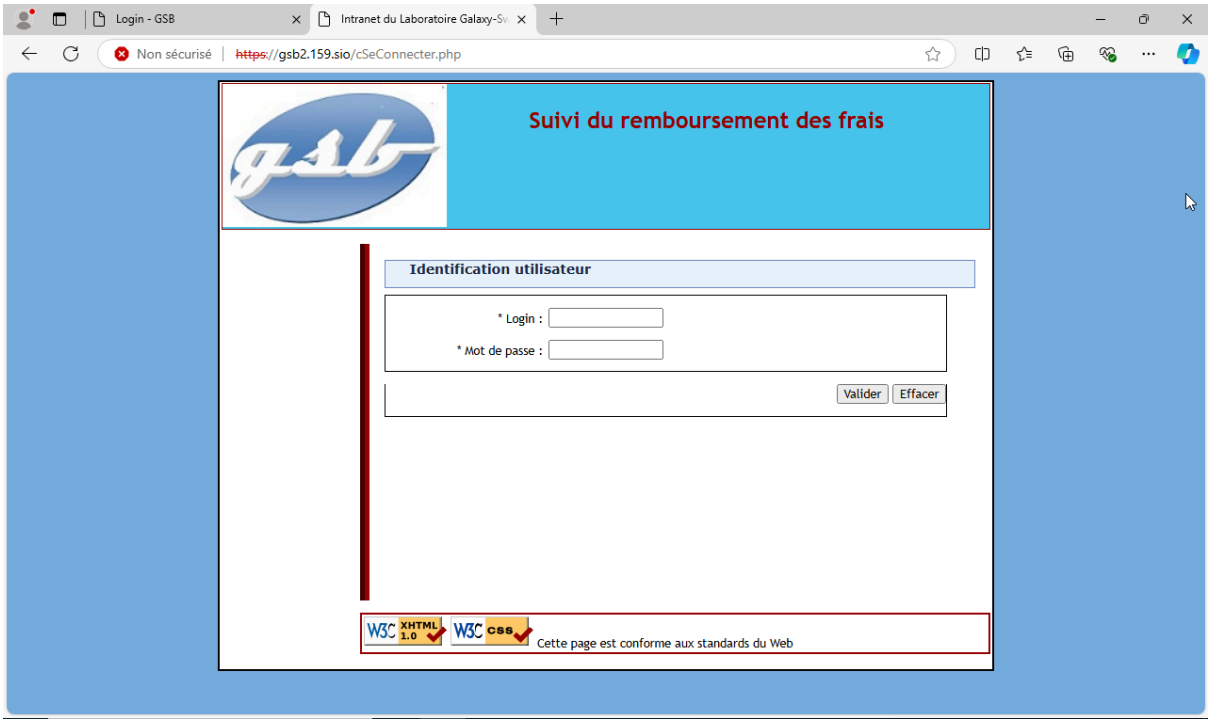
Site GSB





Site GSB2





On voit que nos deux site son bien sécurisé en HTTPS grâce à nos certificats générer via notre autorité de certifications