

Configuration d'un serveur Web et d'une autorité de certification.



L'objectif

L'objectif de ce TP est de configurer un serveur Web sécurisé en mettant en place une autorité de certification locale pour la gestion des certificats SSL/TLS

Résultat final : Un site web avec un certificat auto signée par l'autorité de certification

La réalisation



2 Serveurs debian
(web et autorité de
certification)



Windows Server

1 Machine Windows
Server (DNS)

Configuration d'un serveur Web et d'une autorité de certification.

1. **Configuration de Debian**	
1.1 Configuration réseau Debian	2
1.2 Installation des mises à jour	3
1.3 Création d'un utilisateur Admin	4
2. **Installation des services**	
2.1 Installation de OpenSSH	5
2.2 Installation et configuration de Apache2	5
2.3 Installation et configuration de ProFTPD	6
2.4 Installation et configuration de OpenSSL	7
2.5 Installation de PHP et MariaDB	10
2.6 Configuration de MariaDB	10
2.7 Installation et configuration de PhpMyAdmin	11
3. **Installation et configuration du DNS**	
3.1 Configuration des Virtual Hosts	20
3.2 Mise en place des bases de données	21
4. **Installation et configuration du serveur d'autorité de certification**	
4.1 Installation du serveur d'autorité de certification	22
4.2 Préparation d'un répertoire d'infrastructure à clés publiques	23
4.3 Création d'une autorité de certification	23
4.4 Distribution du certificat de l'autorité de certification	25

1. Configuration de Debian

Dans un premier temps il faut se connecter à la machine avec les identifiants suivants :

Login : **root**

Mot de passe : **Azerty31**

```
Debian GNU/Linux 11 debian11 tty1
debian11 login: root
Password:
Linux debian11 5.10.0-25-amd64 #1 SMP Debian 5.10.191-1 (2023-08-16) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec  4 17:29:56 CET 2023 on tty1
root@debian11:~# _
```

1.1 Configuration réseau Debian

Pour réaliser la configuration du réseau de la machine il va falloir se rendre dans le dossier de ou se trouve le fichier de configuration appelé interfaces qui se trouve dans le dossier network.

Pour y accéder on va taper la commande :

```
cd /etc/network
```

Puis on va aller modifier le fichier interfaces qui se trouve dedans grâce à la commande suivante :

```
nano interfaces
```

Une fois dans le fichier nous allons coller la configuration suivante :

c

Une fois réalisé on enregistre en faisant CTRL + S puis CTRL + X. On y ajoute dans ce fichier nos configurations réseau tel que notre adresse IP, son masque, la passerelle et son DNS. Attention à bien modifier le mode de fonctionnement de la carte en mettant bien static et non DHCP dans la ligne "iface ens18 inet static"

1.2 Installations des mises à jour

Maintenant que notre machine a accès à internet nous pouvons effectuer les mises à jours grâce à la commande suivante :

apt-get update

```
root@debian11:~# apt-get update
Réception de :1 http://deb.debian.org/debian bullseye InRelease [116 kB]
Réception de :2 http://deb.debian.org/debian bullseye-updates InRelease [44,1 kB]
Réception de :3 http://deb.debian.org/debian-security bullseye-security InRelease [27,2 kB]
Réception de :4 http://deb.debian.org/debian bullseye/main Sources [8 500 kB]
Réception de :5 http://deb.debian.org/debian bullseye/contrib Sources [43,2 kB]
Réception de :6 http://deb.debian.org/debian bullseye/main amd64 Packages [8 066 kB]
Réception de :7 http://deb.debian.org/debian bullseye/main Translation-fr [2 433 kB]
Réception de :8 http://deb.debian.org/debian bullseye/main Translation-en [6 235 kB]
Réception de :9 http://deb.debian.org/debian bullseye/contrib amd64 Packages [50,4 kB]
Réception de :10 http://deb.debian.org/debian bullseye/contrib Translation-en [46,9 kB]
Réception de :11 http://deb.debian.org/debian bullseye-updates/main Sources [7 908 B]
Réception de :12 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [18,8 kB]
Réception de :13 http://deb.debian.org/debian bullseye-updates/main Translation-en [10,9 kB]
Réception de :14 http://deb.debian.org/debian-security bullseye-security/main Sources [185 kB]
Réception de :15 http://deb.debian.org/debian-security bullseye-security/main amd64 Packages [294 kB]
Réception de :16 http://deb.debian.org/debian-security bullseye-security/main Translation-en [188 kB]
26,3 Mo réceptionnés en 6s (4 199 ko/s)
Lecture des listes de paquets... Fait
root@debian11:~# _
```

1.3 Création d'un utilisateur Admin

Pour une sécurité optimale dans le système, on va privilégier la création d'un compte administrateur et laisser le compte root de côté.

On va y ajouter l'utilisateur Admintom:

adduser Admintom

On ajoute notre nouvel utilisateur dans le groupe sudo ce qui lui donne les mêmes droits que le compte root:

```
usermod -aG sudo Admintom
```

On va se connecter en tant qu'Admintom grâce à la commande :

```
su -Admintom
```

2. Installation des services

Nous allons maintenant procéder à l'installation des différents services.

2.1 Installation de openssh

Nous allons installer le paquet openssh qui va nous permettre de nous connecter en ssh via Putty ou encore termius ce qui va nous donner l'option copier-coller pour nous faciliter la tâche.

Pour installer le paquet openssh la commande est la suivante :

```
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 33:

```
# Authentication:
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```



```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Une fois réalisé on enregistre en faisant CTRL + S puis CTRL + X

2.2 Installation et configuration de Apache2

Pour installer le paquet apache2 on va entrer 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 gsb1 et gsb2

```
mkdir gsb1
```

```
mkdir gsb2
```

Nous réaliserons la suite de la configuration plus tard dans le tutoriel

2.3 Installation et configuration de proftpd

Nous allons procéder à l'installation de proftpd qui nous servira à transférer les fichiers de nos sites web de notre machine Windows vers le serveur. ProFTPD est un serveur FTP open-source pour transférer des fichiers de manière sécurisée et flexible.

Pour installer le paquet on va taper la commande:

```
apt-get install proftpd
```

Une fois l'installation réalisée nous allons ajouter les groupes et les utilisateurs qui travaillent sur chaque site. Il y aura 2 groupes: dev1 qui sera pour le site gsb1 et dev2 qui sera pour le site gsb2. Pour les utilisateurs il y aura: devgsb1 pour le site gsb1 et devgsb2 pour le site gsb2.

```
adduser devgsb1
```

```
adduser devgsb2
```

Maintenant on crée les groupes :

```
addgroup dev1
```

```
addgroup dev2
```

On intègre nos utilisateurs dans leurs groupes respectif:

```
adduser devgsb1 dev1
```

```
adduser devgsb2 dev2
```

On va maintenant sécuriser notre ftp en mettant en place le sftp.
(Nous mettrons en place les restrictions de groupes plus tard)

2.4 Installation et configuration de openssl

Nous allons sécuriser notre ftp grâce à openssl. OpenSSL est une bibliothèque open-source qui assure la sécurité des communications en chiffrant les données sur les réseaux, utilisée pour le protocole HTTPS et la gestion des certificats SSL/TLS.

Pour l'installer on tape la commande suivante :

```
apt-get install openssl
```

On va ajouter notre fichier de configuration que l'on appelle sftp.conf :

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

Une fois dans le fichier on va copier-coller la configuration suivante:

```
ServerName "SFTP Tom"
UseIPv6 off
RootLogin off
Port 22
DefaultRoot ~
<Limit LOGIN>
DenyGroup !dev1
DenyGroup !dev2
</Limit>
<IfModule mod_ctrls.c>
ControlsEngine off
ControlsMaxClients 2
ControlsLog /var/log/proftpd/controls.log
ControlsInterval 5
ControlsSocket /var/run/proftpd/proftpd.sock
</IfModule>
```

Une fois réalisé on enregistre en faisant CTRL + S puis CTRL + X (Les lignes Denygroup avec un ! avant le nom des groupes servent à dire que seuls les users dans les groupes dev 1 et 2 ont accès au sftp)

On va désormais créer un dossier où sera stocké la clé de certification du sftp :

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

On va maintenant créer notre clé avec la commande :

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

On suit les étapes et on rentre les informations suivantes :

```
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:France
Locality Name (eg, city) []:Toulouse
Organization Name (eg, company) [Internet Widgits Pty Ltd]:GSB
Organizational Unit Name (eg, section) []:GSB1
Common Name (e.g. server FQDN or YOUR name) []:gsb1.160.sio
Email Address []:admingb@gsb.fr
```

On met les droits sur le fichier que l'on vient de créer :

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

On configure maintenant le TLS :

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

On va y coller la configuration suivante :

```
<IfModule mod_tls.c>
[::]
TLSEngine on
TLSLog /var/log/proftpd/tls.log
TLSProtocol SSLv23
TLSOptions NoCertRequest
AllowClientRenegotiations
TLSRSACertificationFile /etc/proftpd/ssl/proftpd.cert.pem
TLSRSACertificateKeyFile /etc/proftpd/ssl/proftpd.key.pem
TLSVerifyClient off
TLSRequired on
RequireValidShell no
TLSOptions NoSessionsReuseRequired
</IfModule>
```

Une fois réalisé on enregistre en faisant CTRL + S puis CTRL + X

On reboot la machine grâce à la commande :

reboot

Maintenant on va faire en sorte que les développeurs puissent avoir accès qu'à leurs dossiers et ne puissent pas remonter vers les autres dossiers. Pour ceci on va modifier la configuration sshd :

nano /etc/ssh/sshd_config

A la fin du fichier on va y coller la configuration suivante :

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

MatchGroup dev1
    ChrootDirectory /var/www/html/dev1
    ForceCommand internal -sftp
    AllowTcpForwarding no
    X11Forwarding no
    PermitTunnel no

MatchGroup dev2
    ChrootDirectory /var/www/html/dev2
    ForceCommand internal -sftp
    AllowTcpForwarding no
    X11Forwarding no
    PermitTunnel no_
```

On redémarre la machine avec la commande :

reboot

2.5 Installation de php et mariadb

On va installer dans un premier temps le paquet php. Le paquet PHP permet d'exécuter du code PHP sur un serveur pour créer des sites web dynamiques et des applications web.

apt-get install php

On redémarre le service apache 2 :

systemctl restart apache2

On va installer mariadb. Le paquet mariadb installe le serveur MariaDB, une base de données open-source permettant de stocker et gérer des données pour des applications et sites web.

`apt-get install mariadb-server`

```
Après cette opération, 164 Mo d'espace disque supplémentaires seront utilisés.  
Souhaitez-vous continuer ? [O/n]
```

On appuie sur O

2.6 Configuration de mariadb

On va réaliser la configuration de maria db :

`mysql_secure_installation`

Enter current password : press entrer car pas de MDP

Change the root password = On ajoute un mot de passe

Remove anonymous users =

Disallow root user = on va créer un compte admin et pour ajouter de la sécurité à notre système il vaut mieux le désactiver

Remove test database = pas besoin de base de test dans notre cas

Reload privilege tables now =

2.7 Installation et Configuration de PHPmyadmin

Nous allons procéder maintenant à l'installation et la configuration de PHPmyadmin. Le paquet *phpMyAdmin* fournit une interface web pour gérer facilement les bases de données MySQL ou MariaDB.

L'installation de PHPmyadmin est différente des installations classiques.

on va aller dans le dossier ou nous allons mettre notre installation temporaire :

```
cd /tmp
```

On va installer le paquet depuis le lien phpmyadmin :

```
wget https://files.phpmyadmin.net/phpMyAdmin/5.2.1/phpMyAdmin-5.2.1-all-languages.zip
```

Le dossier que l'on vient d'installer est en extension zip, il va falloir unzip le dossier.

On va installer l'extension unzip :

```
apt-get install unzip
```

On peut désormais unzip notre dossier

```
unzip phpMyAdmin-5.2.1-all-languages.zip
```

On va déplacer notre dossier pour le sortir des fichiers temporaires :

```
mv phpMyAdmin-5.2.1-all-languages.zip /usr/share/phpmyadmin
```

On va créer un dossier pour y mettre nos fichiers temporaires :

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

Nous allons utiliser le modèle de configuration de base fourni dans les fichiers de phpmyadmin :

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

Maintenant, nous allons générer une clé qui va servir à l'authentification dans notre base de données :

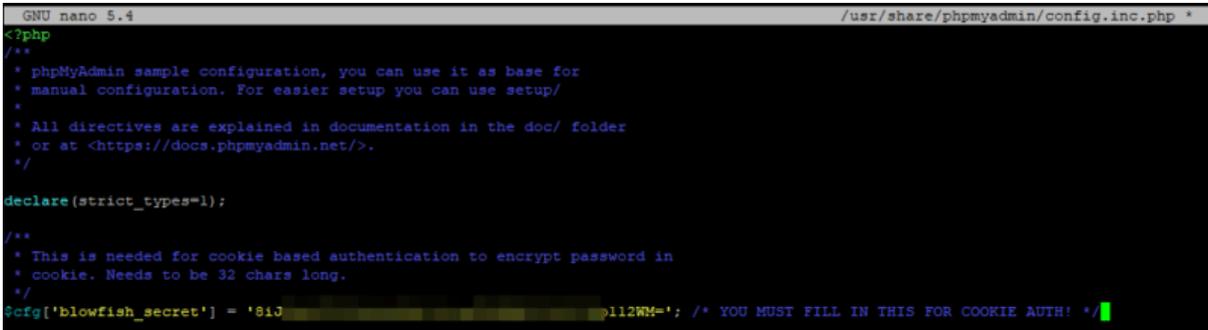
```
openssl rand -base64 32
```

```
root@debian11:~# openssl rand -base64 32  
4257hSpPVKihQHRg1GdWNMBgICirx09CoAVrGoihCpw=  
.....
```

On va copier la clé pour la mettre dans le fichier suivant :

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

On copie la clé au niveau du ['blowfish_secret'] :



```
GNU nano 5.4 /usr/share/phpmyadmin/config.inc.php *  
<?php  
/**  
 * phpMyAdmin sample configuration, you can use it as base for  
 * manual configuration. For easier setup you can use setup/  
 *  
 * All directives are explained in documentation in the doc/ folder  
 * or at <https://docs.phpmyadmin.net/>.  
 */  
  
declare(strict_types=1);  
  
/**  
 * This is needed for cookie based authentication to encrypt password in  
 * cookie. Needs to be 32 chars long.  
 */  
$cfg['blowfish_secret'] = '81J.....112WN='; /* YOU MUST FILL IN THIS FOR COOKIE AUTH! */
```

Pour se connecter à phpmyadmin on va modifier les valeurs de connexions par défaut donc on va modifier les 2 lignes suivantes :

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

```
/**  
 * 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';
```

On va ajouter la ligne suivante qui va déclarer le répertoire temporaire que l'on a 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';  
/**
```

Une fois réalisé on enregistre en faisant CTRL + S puis CTRL + X

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

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

On va désormais créer notre compte Admin:

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

On entre le mot de passe

```
root@debian11:/usr# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.5.26-MariaDB-0+deb11u2 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

On va taper la commande suivante :

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

On va maintenant intégrer phpmyadmin à notre apache

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

Une fois dans le fichier on y colle la configuration suivante :

```
Alias /PHPmyadmin /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-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>
```

Une fois réalisé on enregistre en faisant CTRL + S puis CTRL + X

On active la configuration grâce à la commande :

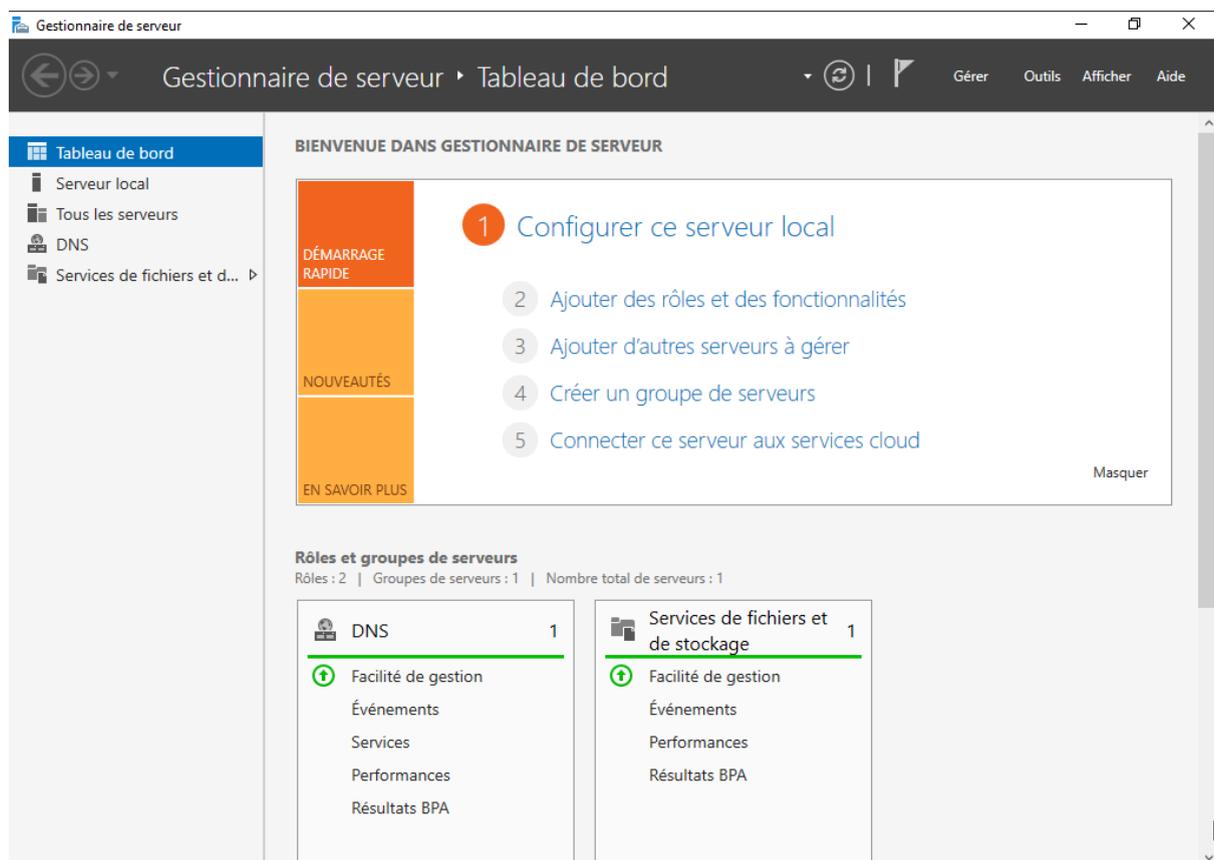
```
sudo a2enconf phpmyadmin.conf
```

On redémarre apache 2 :

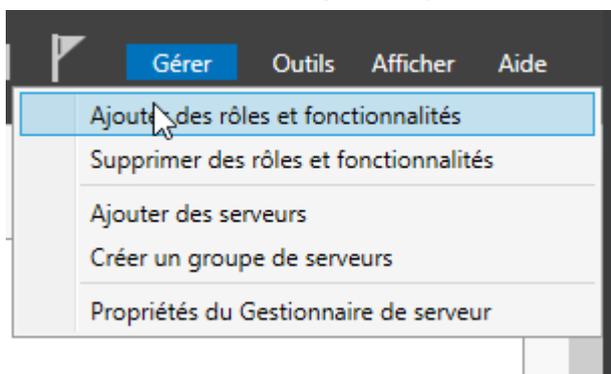
```
systemctl reload apache2
```

3. Installation et configuration du DNS

Pour mettre en place notre DNS il va nous falloir une machine qui tourne sous le système d'exploitation Windows Server 2022. Une fois connecté sur la machine on va accéder au gestionnaire de serveur



On va dans Gérer puis ajouter des rôles et fonctionnalités



On va y ajouter le rôle de serveur "Serveur DNS"

Sélectionner des rôles de serveurs

SERVEUR DE DESTINATION
WIN-9AKK3GNNQ01

Avant de commencer

Type d'installation

Sélection du serveur

Rôles de serveurs

Fonctionnalités

Serveur DNS

Confirmation

Résultats

Sélectionnez un ou plusieurs rôles à installer sur le serveur sélectionné.

Rôles

- Attestation d'intégrité de l'appareil
- Hyper-V
- Serveur de télécopie
- Serveur DHCP
- Serveur DNS**
- Serveur Web (IIS)
- Service Guardian hôte
- Services AD DS
- Services AD LDS (Active Directory Lightweight Dire
- Services AD RMS (Active Directory Rights Manage
- Services Bureau à distance
- Services d'activation en volume
- Services d'impression et de numérisation de docu
- Services de certificats Active Directory
- Services de fédération Active Directory (AD FS)
- ▾ Services de fichiers et de stockage (1 sur 12 install
- Services de stratégie et d'accès réseau
- Services WSUS (Windows Server Update Services)
- Windows Deployment Services

Description

Le serveur DNS (Domain Name System) permet la résolution de noms sur les réseaux TCP/IP. Le serveur DNS est plus facile à gérer lorsqu'il est installé sur le même serveur que les services de domaine Active Directory. Si vous sélectionnez le rôle Services de domaine Active Directory, vous pouvez installer et configurer le serveur DNS et les services de domaine Active Directory pour les faire fonctionner conjointement.

< Précédent

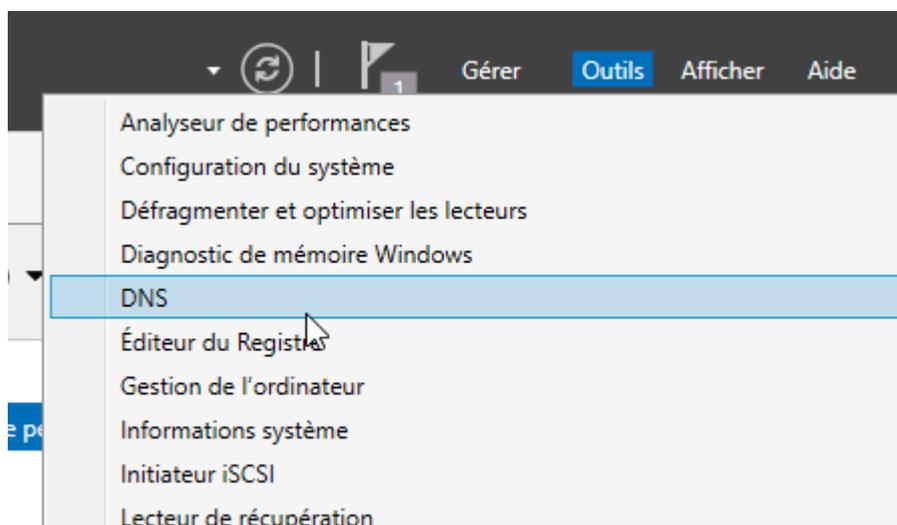
Suivant >

Installer

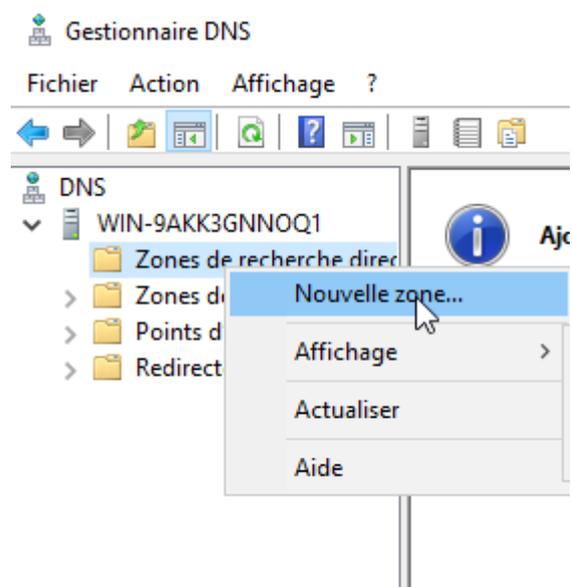
Annuler

On clique sur suivant jusqu'à que l'installation soit terminée.

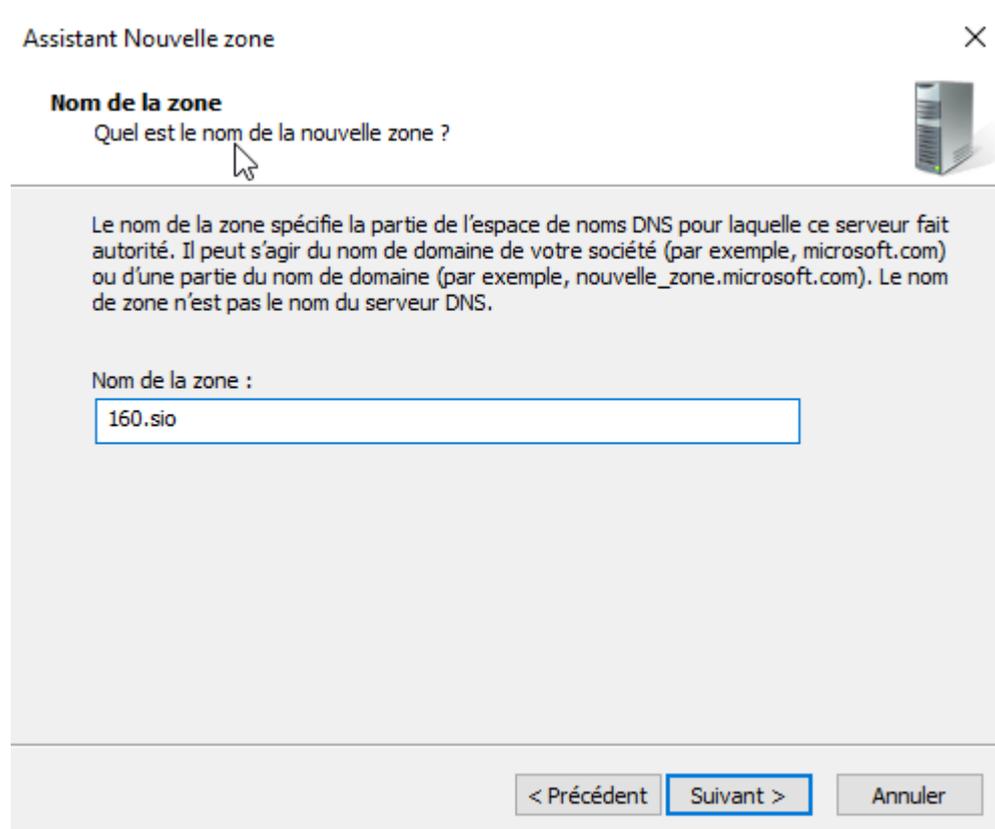
on va retourner sur le tableau de bord puis aller dans l'onglet "outils" en haut à droite puis sélectionner "DNS"



Un onglet nommé Gestionnaire DNS va s'ouvrir. Sur le dossier Zones de recherche directe nous allons effectuer un clic droit puis nouvelle zone



On va nommer notre nouvelle zone avec notre octet et .sio à la fin



On appuie sur suivant à chaque fois.

Dans notre nouvelle zone on y ajoute deux hôtes:

 gsb	Hôte (A)	172.16.160.1
 gsb2	Hôte (A)	172.16.160.1

3.1 Configuration des Virtuals Hosts

Pour configurer les Virtuals Hosts de notre site pour que notre DNS fonctionne on va créer deux nouveaux fichiers de conf :

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

Une fois dans notre nouveau fichier on va y coller la configuration suivante :

```
<VirtualHost *:80>
    ServerAdmin tom@gmail.com
    ServerName gsb.160.sio
    DirectoryIndex index.php

    DocumentRoot /var/www/html/gsb/
    <Directory /var/www/html/gsb/>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Maintenant on va faire de même pour notre deuxième site:

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

Une fois dans notre nouveau fichier on va y coller la configuration suivante :

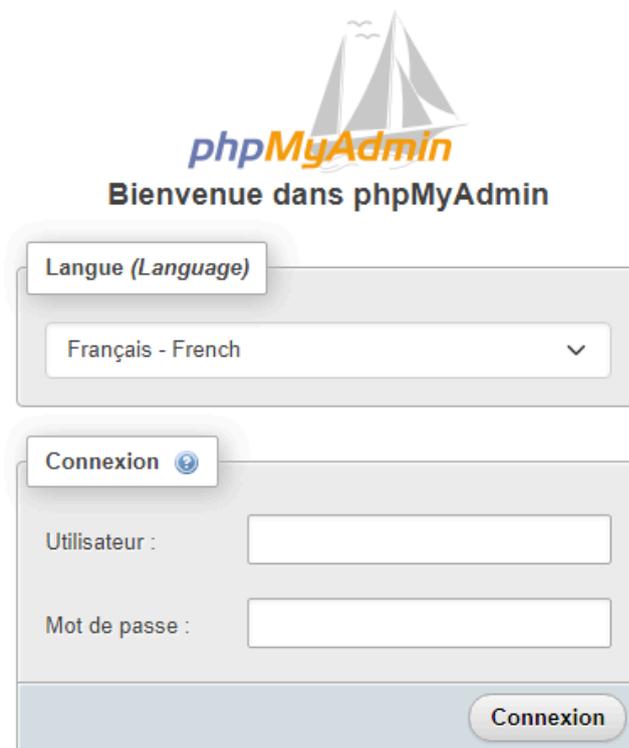
```
<VirtualHost *:80>
    ServerName gsb2.160.sio

    DocumentRoot /var/www/html/gsb2.160.sio
    <Directory /var/www/html/gsb2.160.sio>
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

3.2 Mise en place des bases de données

On va se connecter grâce à un navigateur à notre compte phpmyadmin

gsb.160.sio/PHPmyadmin/



phpMyAdmin
Bienvenue dans phpMyAdmin

Langue (*Language*)

Français - French

Connexion ⓘ

Utilisateur :

Mot de passe :

Connexion

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



On applique un nom aux deux bases de données à créer (gsb1 et gsb2)



Création d'une base de données

Nom de base de données

utf8mb4_general_ci

Créer

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



Choisir un fichier, on met les deux ci dessous :

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

Nos deux bases sont désormais prêtes.

4. Installation et configuration du serveur d'autorité de certification

Sur une nouvelle machine, (Voir le point numéro 1 pour configurer la nouvelle machine) On va y installer le service easy-RSA. Easy-RSA est un ensemble de scripts utilisés pour créer une infrastructure à clés publiques (PKI) pour générer et gérer des certificats TLS/SSL .

4.1 Installation du serveur d'autorité de certification

On va installer le paquet sur notre nouvelle machine :

```
apt-get install easy-rsa
```

4.2 Préparation d'un répertoire d'infrastructures à clés publique

On va préparer un répertoire "d'infrastructure à clé publiques" sur notre machine :

```
mkdir ~/easy-rsa
```

On crée maintenant un lien symboliques:

```
ln -s /usr/share/easy-rsa/* ~/easy-rsa/
```

On s'assure d'être le seul propriétaire du dossier :

```
chmod 700 /home/tom/easy-rsa
```

On initialise maintenant notre configuration de l'ICP :

```
cd ~/easy-rsa
```

```
./easyrsa init-pki
```

Le résultat devrait être le suivant si la procédure à été respectée.

```
init-pki complete; you may now create a CA or requests.  
Your newly created PKI dir is: /home/sammy/easy-rsa/pki
```

4.3 Création d'une autorité de certification

Avant de pouvoir créer la clé et le certificat privés de notre autorité de certification, on va d'abord créer et alimenter un fichier nommé vars avec quelques valeurs par défaut. On va créer un fichier vars :

```
cd ~/easy-rsa
```

```
nano vars
```

Une fois dans le fichier on va coller la configuration suivante :

```
set_var EASYRSA_REQ_COUNTRY "FR"
```

```
set_var EASYRSA_REQ_PROVINCE "Occitanie"
```

```
set_var EASYRSA_REQ_CITY "Toulouse"
```

```
set_var EASYRSA_REQ_ORG "GSB"
```

```
set_var EASYRSA_REQ_EMAIL "admin@example.com"
```

```
set_var EASYRSA_REQ_OU "Community"
```

```
set_var EASYRSA_ALGO "ec"
```

```
set_var EASYRSA_DIGEST "sha512"
```

Une fois réalisé on enregistre en faisant CTRL + S puis CTRL + X

On va créer la paire de clés root public et privé pour notre autorité de certification, on exécute à nouveau la commande `./easy-rsa`, cette fois-ci avec l'option `build-ca` :

```
./easyrsa build-ca
```

```
. . .
Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
. . .
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/home/sammy/easy-rsa/pki/ca.crt
```

On entre une phrase qui va servir de mot de passe pour sécuriser notre autorité de certification.

On entre ensuite le nom de notre serveur, ici dans notre cas `gsb.160.sio`.

4.4 Distribution du certificat de notre autorité de certification

Sur notre serveur on va réaliser la commande suivante:

```
cat ~/easy-rsa/pki/ca.crt
```

```
-----BEGIN CERTIFICATE-----  
MIIDSzCCAjOgAwIBAgIUcR9Crsv3FBEujrPZnZnU4nSb5TMwDQYJKoZIhvcNAQEL  
BQAwFjEUMBIGA1UEAwwLRWFzeS1SU0EgQ0EwHhcNMjAwMzE4MDMxNjI2WhcNMzAw  
.  
.  
.  
-----END CERTIFICATE-----
```

On copie toutes les lignes.

Sur notre serveur Web on va ouvrir un fichier nommé /tmp/ca.crt :

```
nano /tmp/ca.crt
```

Maintenant on va réaliser les commandes suivantes :

```
cp /tmp/ca.crt /usr/local/share/ca-certificates/
```

```
update-ca-certificates
```

Et voilà, notre autorité de certification à été mise en place.