# Compte-rendu déploiement de l'application GSBFRAIS

Date de début : 28/09/2022
Date de fin : 20/10/2022

Destinataire : Membre de la direction de la DSI

*Les membres présents :*

| | |
|---|---|
| Équipe techniciens réseaux :<br><br>Chef de projet : Maxence Picherie<br><br>Nassim Miri<br>Wesley Meri | Équipe développeurs web :<br><br>Chef de projet : Maxim Dailly Zagrebelnyy<br><br>Axel Wilquin<br>Aurélien Boitez<br>Mattew Venet |

Objet : Compte-rendu du déploiement du site interne GSBFrais

Galaxy Swiss Bourdin

# Sommaire

# Introduction

Les visiteurs médicaux démarchent les professionnels de santé susceptibles de prescrire aux patients les produits du laboratoire. L'objectif d'une visite n'est pas de vendre mais d'actualiser la connaissance de ces professionnels sur les produits de l'entreprise. Les déplacements qu'ils mènent engendrent des frais qui doivent être pris en charge par la comptabilité. Une gestion forfaitaire des principaux frais permet de limiter les justificatifs. Le remboursement est fait après le retour de ces pièces.

Le développement de l'application GSBFrais de gestion des demandes de remboursement des frais des visiteurs médicaux est terminé. Notre objectif est maintenant de déployer sur un serveur de production (machine virtuelle debian) l'application.

# La note de service pour GSBFrais demandé par le groupe de développeur

Pour le bon fonctionnement de notre application, certaines ressources sont nécessaires. L'équipe de développement a recherché les différentes ressources et ont trouvé la liste de ressources pour recouvrir à leurs besoins. Vous pouvez voir à la page d'après la demande des développeurs.

Après discussion avec le groupe des techniciens réseaux, une des ressources n'a pas été nécessaire. Le serveur FTP peut être remplacé par une simple liaison SSH.

L'équipe techniciens réseaux ont proposé d'utiliser un serveur DNS afin de faciliter l'accès au site depuis le navigateur.

*Le 27/09/2022*

**Emetteur** - Groupe 2 Développement Web

**Destinataire** - Groupe techniciens réseau

**Objet** : Déploiement de l'application web GSBFrais

Bonjour,

Voici la liste des différentes fonctionnalités dont nous avons besoin pour le bon déploiement du projet GSBFrais.

- Un serveur de Base De Données MariaDB 10.4.24 minimum
- PHP en version 8.1 minimum
- Un serveur web Apache 2.4
- Un serveur FTP sécurisé (pour le transfert du code de l'application)
- Création d'un certificat HTTPS
- Créer un virtualhost dans Apache pour accéder à l'application en HTTPS
- Nous devons accéder à votre serveur à l'aide de putty (par cela vous devez sécuriser les accès à votre serveur, protocole)
- Créer des utilisateurs pour chaque développeurs pour accéder au serveur MariaDB

Pour les utilisateurs de la BDD GSBFrais

- Créer un compte root avec un mot de passe fort sur le serveur MariaDB.
- Créer un compte USER_GSBFRAIS pour l'application sur la BDD ayant les privilèges (SELECT, INSERT, DELETE, UPDATE)

Vous remerciant d'avance pour la rapidité de vos services,

Cordialement,

Le groupe de développement web 2 composé de (Mattew Venet, Axel Wilquin, Maxim Dailly Zagrebelnyy, Aurélien Boitez)

# Récapitulatif de la mise en production

| Fonction | Nom | Adresse IP | Passerelle |
|---|---|---|---|
| Serveur Web - DNS | srv-ap-manawe | 10.1.128.2/16 | 10.1.2.5 |
| Serveur base de données | srv-bdd-manawe | 10.1.128.4/16 | 10.1.2.5 |

# Mise en place des serveurs

## Le serveur Web - Apache2 et php8.1

membre assigné : Wesley MERI

Le groupe des techniciens réseaux ont choisi d'utiliser le système d'exploitation Ubuntu 22.04, car les développeurs ont besoin de la version 8.1 de php et la version 10.4.24 de MariaDB. Ces versions sont disponibles nativement sur la version 22.04 de Ubuntu serveur. Un site internet fonctionne avec un dispositif client/serveur. Dans notre contexte, nous allons créer un serveur web qui utilisera apache2 ainsi que php8.1.

### Installation de apache2 :

Tout d'abord, il faut installer apache2 avec la commande suivante:

```
sudo apt install apache2
```

Puis par la suite installer php8.1 en effectuant la commande ci-dessous:

```
sudo apt install php8.1
```

On installe aussi phpmysql ainsi que libapache2-mod-php qui sont des extensions, conçue pour réaliser des applications PHP

```
sudo apt install phpmysql
sudo apt install libapache2-mod-php
```

On crée un utilisateur avec cette commande:

```
sudo adduser (nom d'utilisateur)
```

Puis on crée un groupe nommé developpeur

```
sudo addgroup developpeur
```

On ajoute chaque utilisateur au groupe developpeur

```
sudo usermod -aG developpeur (nom d'utilisateur)
```

On définit la propriété groupe des fichier du serveur web sur le groupe developpeur :

```
sudo chgrp -R  developpeur /var/www/
```

On définit les droit du groupe des fichiers du serveur web sur lecture et écriture :

```
sudo chmod g+rw /var/www
```

### Mise en place du protocole SSL:

Pour que le protocoles SSL  puisse fonctionner avec le serveur Web, il faut activer le module ssl

```
sudo a2enmod ssl
```

Puis il faut relancer apache pour bien prendre en compte les modifications sur le protocole SSL

```
sudo systemctl restart apache2
```

On peut créer un fichier de clés et de certificats SSL avec la openssl

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

Création d'un dossier gs2.coop:

```
sudo mkdir /var/www/gsb2.coop
```

Ajout d'un fichier gbs2.coop.conf dans le répertoire /etc/apache2/sites-available/

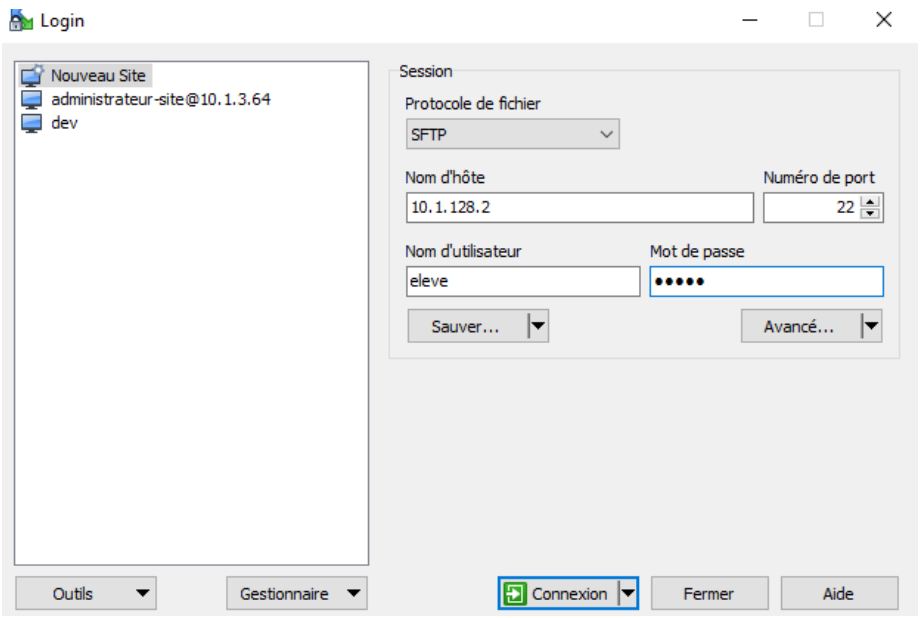
```
sudo nano /etc/apache2/sites-available/gsb2.coop.conf
```

Nous avons deux VirtualHost, un qui redirige vers le port 80 et l'autre sur le port 443.
Sur le VirtualHost du port 80 on indique le nom du serveur et on lui fait un redirect qui va permettre de faire une connexion rediriger vers en https.
Sur le VirtualHost du port 443, on lui indique toujours le nom du serveur puis on lui indique où se situe le dossier où tous le code vas se situer

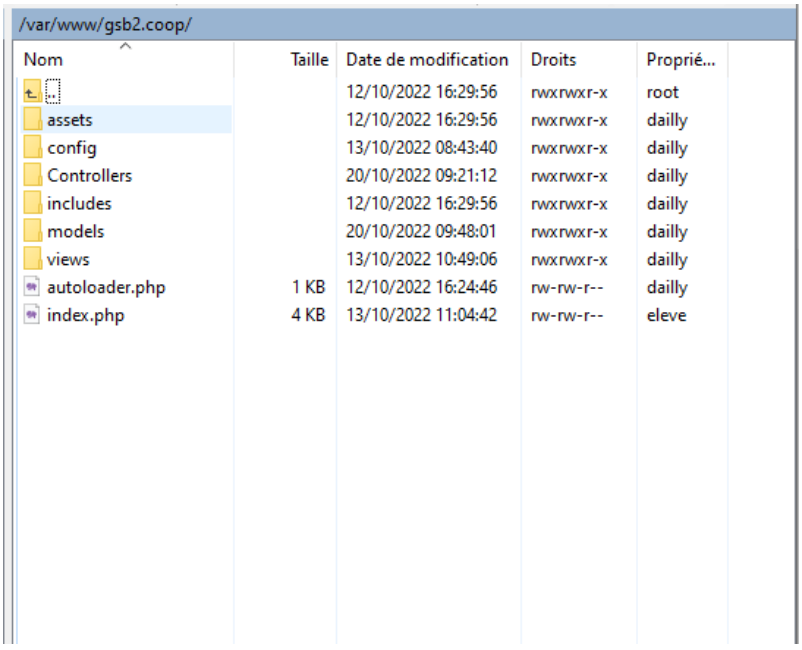
```
<VirtualHost *:80>
        ServerName gsb2.coop
        Redirect / https://gsb2.coop/
</VirtualHost>

<VirtualHost *:443>
   ServerName gsb2.coop
   DocumentRoot /var/www/gsb2.coop

   SSLEngine on
   SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
   SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>
```

### Mis en place des fichiers du serveur web :

Pour les transfert des fichiers nous utiliserons ssh en passant par un programme qui utilise le protocole scp ou sftp comme WinSCP ou FileZilla.




Une fois connecter un simple copier/coller des fichiers du serveur dans le répertoire d'apache /var/www/ suffit.



## Le serveur de Base de données - MariaDB

membre assigné : Nassim MIRI

MariaDB est un système de gestion de base de données relationnelles. Après le rachat de MySQL par Sun Microsystems, puis de Sun par Oracle Corporation, son fondateur (Michael Widenius) démissionne pour lancer une version alternative, sous licence GPL et 100% compatible avec MySQL. MariaDB se base sur le code source de MySQL 5.1.
Il s'agit donc d'un fork plus communautaire et ouvert, et 100% compatible MySQL. Il s'avère aussi plus performant selon certaines études.

### Installation de MariaDB :

Pour installer MariaDB sur la version 22.04 d' Ubuntu, il faut utiliser la commande:

```
sudo apt install mariadb-server
```

La commande apt install permet d'installer un paquet. Un paquet est une archive qui contient des fichiers informatiques.
Sur la version 22.04 d'ubuntu, la version de MariaDB est choisie  par défaut par le système sur la 10.6.7 quand on choisi le paquet mariadb-server

### Configuration de MariaDB :

Une fois MariaDB installé, pour y accéder:

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

La commande permet d'accéder à la base de donnée mysql, en utilisant l'utilisateur root et en précisant que ce dernier à un mot de passe.
-u défini l'utilisateur.
-p permet de préciser que cet utilisateur à un mot de passe et de pouvoir le saisir.

Créer un utilisateur pour accéder à la base de donnée:

```
CREATE USER 'user'@'%' IDENTIFIED BY 'password';
```

L'instruction CREATE USER permet de créer un utilisateur dans la base de donnée, cet utilisateur est identifié par un login et un mot de passe.
% sert à pouvoir  se connecter avec cet utilisateurs a partir de n'importe quel poste .
IDENTIFIED BY permet de créer le mot de passe de l'utilisateur.

Attribuer les privilèges demander:

```
GRANT SELECT, INSERT, DELETE, UPDATE PRIVILEGES ON * . * TO 'user'@'%';
```

SELECT permet de lire les données des tables existantes dans la base de données par exemple les utilisateurs.
INSERT INTO permet d'ajouter des données à une table.
UPDATE permet d'effectuer des modifications sur des données existantes.
DELETE permet de supprimer une ou plusieurs donnée(s) d'une table.

Appliquer les privilèges:

```
FLUSH PRIVILEGES;
```

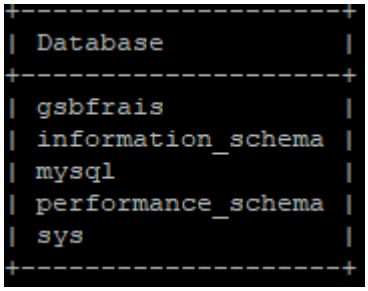
Création de la base de donnée GSBFRAIS avec la commande CREATE DATABASE:

```
CREATE DATABASE "name_of_the_base";
```

L'instruction CREATE DATABASE permet de créer la base de données identifiée par un nom.

Commande pour voir les bases de données créer sur MariaDB:

```
SHOW DATABASES;
```




L'instruction SHOW DATABASES permet de voir les bases de données qui ont été créées dans MariaDB.

Utiliser la Base de donnée mysql pour changer le mot de passe de root:

```
USE mysql;
```

L'instruction USE permet d'utiliser une base de données spécifique en mettant son nom.

Modification du mot de passe root avec un mot de passe fort:

```
SET PASSWORD FOR 'root'@'localhost' = PASSWORD ('NEW_PASSWORD');
```

SET PASSWORD FOR permet de modifier un mot de passe en précisant le nom d'utilisateur concerné et en y indiquant le nouveau mot de passe.

Ensuite nous allons devoir créer un compte  pour l'application, nous allons avoir besoin de changer de BDD:

```
USE GSB FRAIS;
```

Et pour finir, pour créer le compte d'application, il faut réutiliser la méthode utilisée ci-dessus dans la BDD:

```
CREATE USER 'user'@'%' IDENTIFIED BY 'password';
GRANT SELECT, INSERT, DELETE, UPDATE PRIVILEGES ON * . * TO 'user'@'%';
FLUSH PRIVILEGES;
```

Chaque développeurs à son compte utilisateurs, pour les voir, utiliser la commande:

```
SELECT user FROM mysql.user;
```

```
+---------------+
| User          |
+---------------+
| USER_GSBFRAIS |
| boitez        |
| temp          |
| venet         |
| wilquin       |
| zagrebelnyy   |
| mariadb.sys   |
| mysql         |
| root          |
+---------------+
```

SELECT user FROM permet de voir les utilisateurs de la table user de la base de donnée mysql.

## Le serveur DNS - Bind 9

membre assigné : Maxence PICHERIE

Pour le serveur DNS nous utiliserons un service nommé BIND9.
Le service DNS (Domain Name System) est un service TCP/IP permettant la correspondance entre un nom de domaine qualifié (FQDN : Fully Qualified Domain Name) et une adresse IP, par exemple www.gsb2.coop = 10.1.128.2. Ainsi, grâce au DNS, il n'est pas nécessaire de se souvenir des adresses IP.
Un serveur qui héberge le service DNS est appelé "serveur de noms". Ubuntu est livré par défaut avec BIND (Berkley Internet Naming Daemon), le serveur DNS le plus utilisé sur Internet.

### Installation de Bind9 :

Pour installer, il suffit d'utiliser la commande apt install, car le service bind9 est disponible dans le dépôt d'Ubuntu :

```
sudo apt install bind9
```

### Configuration de Bind9 :

Une fois BIND9 installé, il faut procéder à sa configuration. Le fichier de configuration principale est : */etc/bind/named.conf.local*

Dans notre cas nous utiliserons un serveur primaire, il va donc falloir modifier ce fichier afin de créer une zone :

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

Notre fichier de configuration sera le suivant :

```
zone "gsb2.coop" {            (déclaration du nom de zone)
      type master;            (déclaration du type de zone)
      file "/etc/bind/gsb2.coop";   (déclaration du fichier de
configuration de la zone)
};

logging {         (activation des logs)
      category default { default_syslog; default_debug; };
      category unmatched { null; };
};
```

Après cela il faut créer le fichier de configuration de la zone gsb2.coop et la configurer :

```
sudo nano /etc/bind/gsb2.coop
```

Notre configuration :

```
$TTL  86400
;
@     IN    SOA   srv-ap-manawe.gsb2.coop. maxence.picherie.gmail.com (
                        2022100610        ; serial, numéro de série, doit
augmenter à chaque modification
                        8H                ; refresh, secondes
                        2H                ; retry, secondes
                        1W                ; expire, secondes
                        1D )              ; minimum, secondes

@                IN               NS     srv-ap-manawe.gsb2.coop.
srv-ap-manawe    IN               A      10.1.128.2
www              CNAME            srv-ap-manawe
```

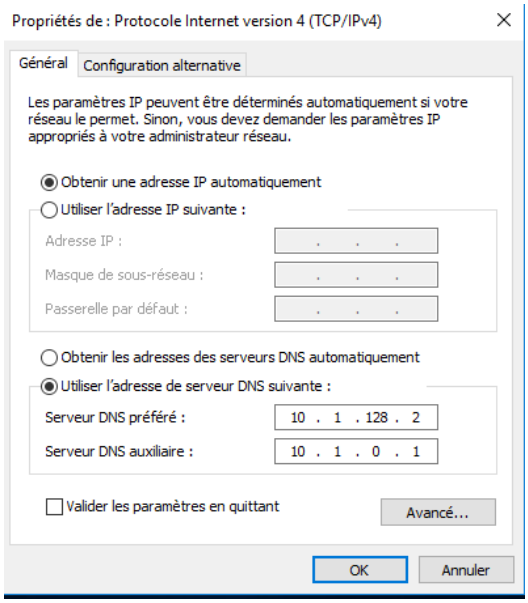
Puis il faut redémarrer le service bind9 pour bien mettre en place les modifications:

```
sudo service bind9 restart
```

### Test du serveur DNS :

Pour que le navigateur puisse utiliser notre serveur DNS, il va falloir le configurer dans les paramètre réseau du système d'exploitation:

Sur Windows :
Il faut aller dans les paramètres de la carte réseau, puis dans les propriétés IPv4 et modifier les paramètres DNS.




Puis en utilisant le raccourci windows + R et en tapant cmd dans la fenêtre un terminal de commande windows devrait s'ouvrir. Une fois ceci fait il suffit de taper la commande suivante pour tester le serveur DNS :

```
nslookup www.gsb2.coop
```

La commande nous renvoi :

```
C:\Users\eleve>nslookup www.gsb2.coop
Serveur :   UnKnown
Address:  10.1.128.2

Nom :    srv-ap-manawe.gsb2.coop
Address:  10.1.128.2
Aliases:  www.gsb2.coop
```

On y voit que le nom de domaine www.gsb2.coop correspond bien a l'adresse IP de notre serveur web. Ce qui veut dire que le serveur fait bien l'association nom de domaine/IP.

Sur Linux :
Il faut modifier le fichier : /etc/netplan/*.yaml et y ajouter notre DNS avec la ligne : nameservers

```
network:
  ethernets:
    enp3s0f0:
      addresses: [10.1.128.2/16]
      gateway4: 10.1.2.5
      nameservers:
        addresses: [10.1.128.2, 8.8.8.8]
    enp3s0f1:
      dhcp4: true
```

```
    enp4s0f0:
      dhcp4: true
    enp4s0f1:
      dhcp4: true
  version: 2
```

Puis faire un :

```
sudo netplan apply
```

Pour tester notre serveur DNS il suffit d'utiliser nslookup dans la console windows ou linux :

```
nslookup www.gsb2.coop
```

Il nous renvoie :

```
Address: 10.1.128.2
```

On voit que www.gsb2.coop correspond bien à notre serveur web 10.1.128.2 ce qui veut dire que le serveur DNS fonctionne bien.

## Sécurisation de l'accès ssh des serveurs - DenyHost

Membre assigné : Wesley MERI et Nassim MIRI

Denyhost est un daemon (un programme informatique qui s'exécute en arrière-plan) qui analyse en permanence le fichier de log /var/log/auth.log et qui au bout de plusieurs tentatives de connexion infructueuses
en ssh (protocole qui permet aux utilisateurs de se connecter à distance à des systèmes hôte de serveurs) blacklist l'IP en cause dans le fichier /etc/hosts.deny.

### Installation et configuration de denyhost :

La commande permet d'installer le client openssh elle doit être utilisée sur la machine client :

```
sudo apt install openssh-client
```

La commande suivante permet d'accéder au fichier wgetrc. Ce fichier permet de modifier le proxy par défaut pour la commande Wget et utiliser un proxy en https,http et ftp :

```
sudo nano /etc/wgetrc
```

```
# You can set the default proxies for Wget to use for http, https, and ftp.
# They will override the value in the environment.
https_proxy = http://10.1.2.5:8080/
http_proxy = http://10.1.2.5:8080/
ftp_proxy = http://10.1.2.5:8080/
```

L'url suivant permet l'installation de denyhost :

```
wget https://downloads.sourceforge.net/project/denyhost/denyhost-2.8/denyhosts-2.8.tar.gz
```

```
anawe@srv-ap-manawe:~$ wget https://downloads.sourceforge.net/project/denyhost/denyhost-2.8/denyhosts-2.8.tar.gz
-2022-10-13 09:09:02--  https://downloads.sourceforge.net/project/denyhost/denyhost-2.8/denyhosts-2.8.tar.gz
onnecting to 10.1.2.5:8080... connected.
roxy request sent, awaiting response... 302 Found
ocation: https://master.dl.sourceforge.net/project/denyhost/denyhost-2.8/denyhosts-2.8.tar.gz?viasf=1 [following]
-2022-10-13 09:09:03--  https://master.dl.sourceforge.net/project/denyhost/denyhost-2.8/denyhosts-2.8.tar.gz?viasf=1
onnecting to 10.1.2.5:8080... connected.
roxy request sent, awaiting response... 200 OK
ength: 48345 (47K) [application/x-gzip]
aving to: 'denyhosts-2.8.tar.gz'

enyhosts-2.8.tar.gz         100%[=======================================>]  47,21K   147KB/s    in 0,3s

022-10-13 09:09:04 (147 KB/s) - 'denyhosts-2.8.tar.gz' saved [48345/48345]
```

La commande tar permet de décompresser le fichier deny host dans l'emplacement élève :

```
tar xzf denyhosts-2.8.tar.gz
```

Le x permet d'extraire certains fichiers d'une archive
Le z permet de décompacter l'archive avec l'utilitaire gzip
Le f permet d'extraire un fichier donné

La commande suivante permet d'accéder au dossier denyhost :

```
cd DenyHosts-2.8
```

Pour l'installation de denyhost il nous faut python, la première versions de python n'étant plus présent dans la bibliothèque nous installons sa deuxième versions:

```
sudo apt install python2
```

Ci-joint la commande permettant d'installer le programme setup.py qui est en python*:*

```
sudo python2 setup.py install
```

```
build          daemon-control-dist  denyhosts.8     denyhosts.py  Makefile     PKG-INFO  README.txt  setup.py
CHANGELOG.txt  DenyHosts            denyhosts.conf  LICENSE.txt   MANIFEST.in  plugins   scripts     TODO
```

Après avoir exécuté les commandes ci-dessus, DenyHosts sera installé sur notre serveur mais pas entièrement configuré. Nous devons configurer manuellement l'outil et copier le fichier daemon dans le répertoire /etc/init.d/ :

```
sudo cp /usr/local/bin/daemon-control-dist /etc/init.d/denyhosts
```

Ensuite, ouvrir le script et effectuez le changement :

```
sudo nano /etc/init.d/denyhosts
```

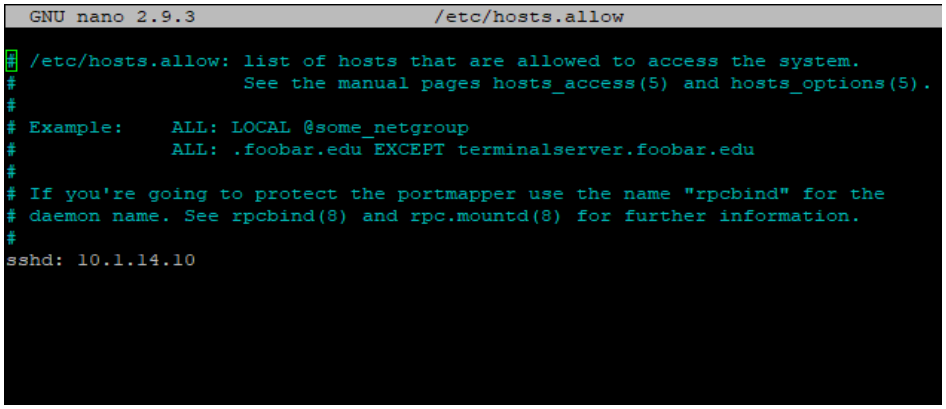
Modifier les fonctions suivantes :

- DENYHOSTS_BIN = "/usr/local/bin/denyhosts.py"
- DENYHOSTS_LOCK = "/run/denyhosts.pid"
- DENYHOSTS_CFG = "/etc/denyhosts.conf"
- PYTHON_BIN = "/usr/bin/env python2"

Pour ajouter une adresse IP à la liste blanche (moyen qu'ils ne soient pas bloqués ou bannis par erreur), il faut ouvrir le fichier hosts.allow dans le répertoire /etc/ et ajouter une adresse IP que nous utilisons pour nous connecter au serveur. La commande suivante permet de réaliser tout le problème :

```
sudo nano /etc/hosts.allow
```

*Voici un exemple :*

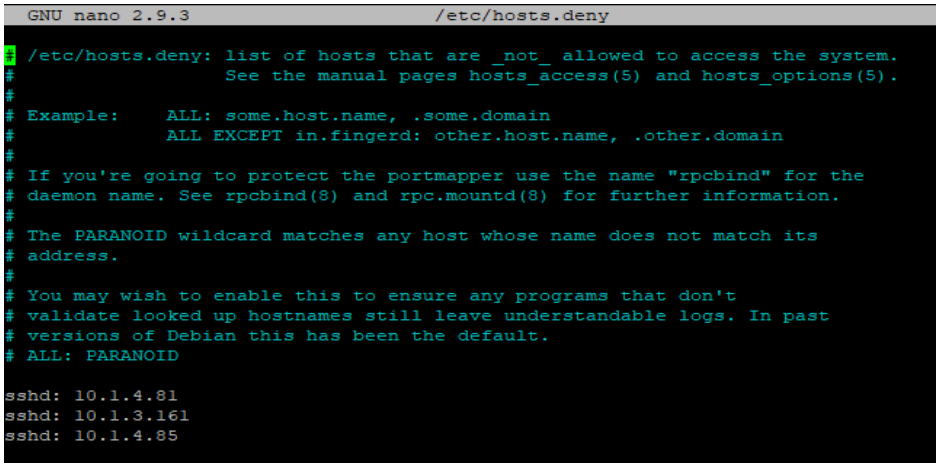



*Les commandes suivantes permet de démarrer denyhost :*

```
sudo /etc/init.d/denyhosts start
```

```
sudo systemctl start denyhosts
```

La blacklist de toutes les adresses IP sont présentées ci-joint dans le dossier hosts.deny (en anglais deny signifie rejeter ). Voici un exemple:




Pour bloquer ces adresses IP deny host va vérifier dans le fichier /var/log/auth.log puis regarder tout ce qui est en rapport avec une connexion ssh puis si il voit plusieurs tentative de connexion il va mettre adresse IP dans la blacklist ci-dessus

### Pour dé-bannir un utilisateur :

Il faut arrêter le service denyhost

```
sudo systemctl stop denyhost
```

Puis aller sur le fichier /etc/hosts.deny puis supprimer l'adresse IP

```
sudo nano /etc/hosts.deny
```

Et il faut vérifier la table de routage avec la commande:

```
sudo iptables -L -n -v
```

Nous pouvons voir l'adresse IP étant bannie, pour "débannir" cette adresse, on effectue cette commande pour l'enlever de la liste noir:

```
sudo iptables -F
```

Puis on peut relancer le service denyhost

```
sudo systemctl start denyhost
```

# Récapitulatif des problèmes rencontrés

## Modification du paramétrage php

- La ligne **“**display_Error -> *Off”* dans le fichier php.ini à dû être modifier en activant sur **“**On**”** les erreurs. Pour cela on a utilisé la commande :

```
sudo nano /etc/php/8.1/php.ini
```

- Ensuite faire la modification du dossier controllers en le renommant avec une majuscule : Controllers.
- Donner l'accès à la base pour le serveur Apache, écrivant l’adresse **“**0.0.0.0**”** dans le fichier mysql.conf avec la commande :

```
etc/mariadb/cdb/mysql.conf bind adresse 0.0.0.0
```

Cette adresse utilisée permet à d’autres personnes et d’autres machines de se connecter à distance.

- Ensuite dans le code de l’application, le fichier MYPDO dans le dossier models, on a dû retirer en supprimant la ligne 18 en la supprimant. Cette ligne concerne l’encodage utf-8, qui se trouve être la norme d'écriture.

## Installation de denyhost

Denyhost est un programme informatique qui analyse en permanence le fichier de log/var/log/auth.log. Au bout de plusieurs tentatives de connexion infructueuses, l’adresse IP est blacklistée dans le fichier /etc/hosts.deny. Il n’est plus disponible sur la version 22.04 d' Ubuntu maintenant, on utilise une Ubuntu 18.04 et on fait des tests en utilisant différentes machines virtuelles. On essaye des faux mots de passe pour voir si l’adresse IP est blacklistée. Nassim installe Denyhost sur sa machine virtuelle, les commandes d’installation normales ne marchent pas donc il installe Denyhost depuis le site internet avec la commande wget. Deny host est installé, Wesley tente de se connecter avec des faux mots de passe mais le service deny host ne blackliste pas son adresse IP.

## Changement dans les codes PHP de l’application

### Accès aux données la base

L’utilisation de serveurs Linux demande beaucoup de précision suite à la grande sensibilité à la casse. Chaque commande et requête utilisant des tables et des colonnes de la base

gsbfrais devaient être écrites au caractère près (ne pas mettre de majuscule). Nous avons dû vérifier chaque commande accédant à la base pour vérifier si des majuscules ont été tapées. Ce problème n'était pas envisagé étant donné qu'à chaque utilisation de notre application dans notre serveur local windows, aucun problème n'a été détecté.

### Problème de fonctions

Ayant utilisé un script pour remplir la base de données lors de la période de test en local, l'application n'était pas prête à fonctionner sans données.Certaines pages ne voulaient pas s'ouvrir (affichage de l'erreur 500). En effet, certaines fonctions ne possédaient pas de solution en cas d'aucune données sur la base. Une modification de certaines fonctions ont été nécessaires au bon fonctionnement de l'application.

### Petit point instable lors des modifications des fichiers

Lors de chaque modification d'un fichier, nous avions besoin de supprimer l'ancien fichier et de transférer le nouveau comportant les nouvelles modifications. Cette méthode était lente et agaçante pour l'utilisateur. Nous avons appris qu'il existait une méthode pour modifier directement les fichiers sur le serveur de base de données, mais notre stade d'avancement était déjà très avancés, nous avons donc continué à utiliser cette méthode.