Documentation
Serveur seulement

Documentation pour différent éléments

James Benone

v. 2025.08.21 - 15:54

Thursday 11 June 2026

Table des matières

1. Création et Configuration d'une LAMP

1.1 Prérequis

Un serveur sous linux ou une VM sous linux

1.2 Création d'une connexion sécurisée

1.2.1 Création du nom de domaine en local

Pour modifier le fichier hosts, sur windows, c'est `C:/Windows/System32/drivers/etc/hosts` et sur linux, c'est `/etc/hosts`

Ajouter une ligne avec :

Text Only

```
<ip-lamp> lamp.mshome.net
```

1.2.2 Création de la configuration

On va créer un fichier de config

Text Only

```
nano config
```

Et on va ajouté le contenu suivant :

Text Only

```
Host lamp
    HostName lamp.mshome.net
    User <utilisateur>
    Port 22
```

1.2.3 Générer une clé SSH

Génerer une clé SSH sur windows ou linux via la commande suivante :

Bash

```
ssh-keygen -t ed25519
```

Vous pouvez juste faire enter jusqu'à la création du fichier.

Récupérer la clé public que vous venez de créer en ouvrant le fichier sur windows ou via la commande suivante sous linux :

Bash

```
cat id_ed25519.pub
```

Et copier la clé

1.2.4 Mettre la clé sur la machine

Ensuite, connectez-vous une première fois sur la machine de manière physique ou via le shell (si vous utilisez une VM)

aller dans `~/.ssh/` et modifier le fichier `authorized_keys` ou créer le si il n'existe pas via VI ou NANO

Bash

```
nano authorized_keys
```

et ajouter la clé public que vous aviez fait juste avant

1.2.5 Votre première connexion via SSH

Ouvrez votre terminal et faite la commande suivante

Bash

```
ssh lamp
```

1.3 Installation et configuration des éléments nécessaire

1.3.1 Apache2

Pour installer apache2, il suffit de faire un :

Bash

```
sudo apt install apache2 -y
```

1.3.2 MariaDB

Pour installer MariaDB, il suffit de faire un :

Bash

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

Puis de le configurer avec :

Bash

```
mariadb_secure_installation
```

Si il vous met des trucs, mettez oui à chaque fois sauf dans le cas d'une configuration parcticulière

1.3.3 PHP

Pour installer PHP, on va uniquement installer les paquets nécessaire via :

Bash

```
sudo apt install php php-dev php-mysql -y
```

Test

Vous allez dans `/var/www/html`

Et vous allez créer le fichier `info.php` avec le contenu suivant :

PHP

```
<?php
    // Affiche les informations de PHP
    phpinfo();
?>
```

Et vous allez dans votre url lamp.mshome.net/info.php

Vous devriez avoir la page PHP

1.3.4 XDebug

Dans un premier temps, on va installer XDebug avec l'installeur officiel de PHP via la commande suivante :

Bash

```
sudo pecl install xdebug
```

Ensuite vous allez modifier le fichier de configuration PHP dans :

Bash

```
# Remplacer <version> par votre version de PHP (dans ce cas 8.3)
sudo nano /etc/php/<version>/php.ini
```

Et vous allez toute à la fin de la page puis vous ajoutez :

INI

```
; blabla du dessus

[xdebug]
zend_extension=/usr/lib/php/20230831/xdebug.so
xdebug.start_with_request=yes
xdebug.mode=debug
```

Toujours dans php.ini, il faut aussi chercher les éléments `display_errors` et `display_startup_errors`, les valeurs par défaut sont `Off`, remplacés les par `On`.

Pour en finir, il suffit de relancer apache2 avec la commande suivante :

Bash

```
sudo service apache2 restart
```

1.3.5 ZSH et OhMyZsh

Pour installer ZSH, il suffit de faire :

Text Only

```
sudo apt install zsh -y
```

Et installer OhMyZsh :

Text Only

```
sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

Si il vous propose : `Do you want to change your default shell to zsh? [Y/n]`, vous faite simplement la touche entrer.

Il suffit de changer le thême via :

Bash

```
nano .zshrc
```

Et de modifier `ZSH_THEME` par `bira` puis de mettre à jour avec :

Bash

```
source .zshrc
```

1.3.6 Configurer les permissions du dossier html

Premièrement, on va aller dans le dossier `/var/www`

Deuxièmement, on va donner les droits du dossier à Apache via la commande suivante :

Bash

```
sudo chown -R www-data:www-data html
```

Et maintenant, on va donner les droits au groupe de modifier le dossier et son contenu :

Bash

```
sudo chmod -R g+w html
```

Il suffit maintenant d'ajouter l'utilisateur au groupe `www-data` avec :

Bash

```
sudo usermod -aG www-data <utilisateur>
```

Pour finir, il suffit de sortir avec la commande `exit` et de revenir de la même manière dont vous êtes venu juste avant

1.4 Ajout d'un projet école

1.4.1 Ajout du fichier config

Créer le fichier ou modifier le via la commande :

Text Only

```
nano config
```

Et ajouter le contenu suivant :

Text Only

```
Host gitlab.ictge.ch
    HostName gitlab.ictge.ch
    User <utilisateur>
    Port 22002
```

1.4.2 Configurer git

Il faut pour cela modifier son nom d'utilisateur et son email de façon global via la commande :

Bash

```
git config --global user.name james-bnn
git config --global user.email james.bnn@eduge.ch
```

1.4.3 Création d'une clé SSH sur le serveur

Pour cela, vous pouvez simplement faire la commande :

Bash

```
ssh-keygen -t ed25519
```

Et faire entrer jusqu'à la création de la clé

Maintenant, il faut récupérer la clé via :

Bash

```
cat id_ed25519.pub
```

Copier son contenu.

1.4.4 Ajout de la clé sur GitLab

Aller sur votre instance GitLab et connectez-vous.

Ensuite, faite `Profile/Edit Profile/SSH Keys` et cliquer sur `Add new key`

Mettez ensuite la clé et faite `Add key`

1.4.5 Importer un projet

Sur votre instance GitLab, aller sur le projet que vous souhaitez ajouter et faite `Code->Clone with SSH`

Retourner sur le terminal SSH et aller dans `/var/www/html` et faite la commande suivante pour importer le projet :

Bash

```
git clone git@gitlab.ictge.ch:edu-bonvinp/webeng_es1.git
```

1.4.6 Ouvrir VSCode sur votre serveur

Ouvrez votre Visual Studio Code, cliquer sur la petite îcone en bas à droite des 2 flèches qui se croise puis faite `Connect to Host->lamp`

Une fois cela fait, faite `Open folder` et aller dans `/var/www/html/<votre-projet>`

Et voilà, vous êtes prêt à travailler.

15 octobre 2025

2. Protéger votre VPS/VDS/Serveur

2.1 Pare-feu

Dans un premier temps, on va installer un pare-feu, sur linux, on peut installer ufw via la commande suivante :

Bash

```
sudo apt install ufw -y
```

Il faut le configurer et mettre les ports que vous souhaitez exposer, rappelez-vous que moi de ports sont ouverts, mieux s'est :

Bash

```
sudo ufw allow 22 # Port SSH
sudo ufw allow 80 # Port HTTP
sudo ufw allow 443 # Port HTTPS
sudo ufw allow 3306 # Port MySQL (non recommandé)
```

Il faut absolument que vous laissiez le port SSH ouvert au risque de ne plus avoir accès à votre serveur.

Il ne vous reste plus qu'à l'activer

Bash

```
sudo ufw enable
```

Le message suivant va apparaitre, il suffit de mettre "y" :

Text Only

```
Command may disrupt existing ssh connections. Proceed with operation (y|n)?
```

2.2 Configurer SSH

IMPORTANT : Il faut avoir fait les sections "Générer une clé SSH" et "Mettre la clé sur la machine".

On va aller toucher au fichier de configuration, et pour cela, il faut aller dans :

Bash

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

Et vous allez chercher la première ligne à changer : `PermitRootLogin` et vous devez la décommenté et mettre no :

Text Only

```
PermitRootLogin no
```

Vous allez aussi chercher : `PasswordAuthentication`

Text Only

```
PasswordAuthentication no
```

Il ne vous reste plus qu'à redémarrer ssh :

Bash

```
sudo service ssh restart
```

Maintenant, essayer de vous connectez en SSH via le mot de passe et vous devriez avoir le message suivant :

Bash

```
martin@192.168.1.144: Permission denied (publickey).
```

2.2.1 Si le mot de passe est toujours fonctionnel

Dans le cas ou il vous propose toujours de vous connectez par mot de passe, c'est qu'il y a un fichier config dans le dossier `/etc/ssh/sshd_config.d/` qui contient une instruction opposée. Vous devez donc inspecter les différents fichiers pour modifier la valeur.

Dans mon cas, ce fichier était `50-cloud-init.conf` .

2.3 Fail2Ban

Le fail2ban est un outil qui s'occupe permet d'empêcher le brute-force

Bash

```
sudo apt install fail2ban
```

Si vous souhaitez configurer le fail2ban, vous pouvez en modifiant le fichier de configuration :

Bash

```
sudo nano /etc/fail2ban/jail.d/defaults-debian.conf
```

Mettez le contenu suivant et modifier selon vos envies

INI

```
[DEFAULT]
banaction = nftables
banaction_allports = nftables[type=allports]
backend = systemd
ignoreip = 127.0.0.1
bantime = 6000
findtime = 600
maxretry = 3

[sshd]
enabled = true
port = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s
```

Et il vous suffit de redémarrer le fail2ban

Bash

```
sudo service fail2ban restart
```


15 octobre 2025

3. Mettre la double authentification sur SSH

3.1 Installer le paquet

On va installer le packet de Google Authenticator via la commande suivante :

Bash

```
sudo apt install -y libpam-google-authenticator
```

3.2 Configurer

Dans un premier temps, on va activer la double authentification par SSH en allant modifier le fichier suivant :

Bash

```
sudo nano /etc/pam.d/sshd
```

Puis ajouter la ligne suivante :

Bash

```
auth required pam_google_authenticator.so
```

Mais il faut également commenter une ligne :

Bash

```
# @include common-auth
```

Puis redémarrer le service via la commande :

Bash

```
sudo service ssh restart
```

Maintenant, il faut modifier la configuration SSH :

Bash

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

Premièrement, ajouter une ligne dans le fichier :

INI

```
AuthenticationMethods publickey,keyboard-interactive
```

Et il faut modifier les lignes :

INI

```
KbdInteractiveAuthentication yes # SET TO YES
ChallengeResponseAuthentication yes # SET TO YES
```

3.3 Activer sur un compte

Pour cela, il faut éxecuter la commande de Google :

Bash

```
google-authenticator
```

Il va vous demandé si vous voulez un token basé sur le temps

Text Only

```
Do you want authentication tokens to be time-based (y/n) y
```

Vous allez avoir un QRCode a scanné avec votre natel et rentrez le code.

Il va vous donnez des codes d'urgences, sauvegardez-les quelques parts de sécurisé et sur une autre plateforme.

Ensuite, il va vous posez quelques questions, il va falloir les configurer :

Text Only

```
Do you want me to update your "/home/<user>/.google_authenticator" file? (y/n) y

Do you want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it increases
your chances to notice or even prevent man-in-the-middle attacks (y/n) y

By default, a new token is generated every 30 seconds by the mobile app.
In order to compensate for possible time-skew between the client and the server,
we allow an extra token before and after the current time. This allows for a
time skew of up to 30 seconds between authentication server and client. If you
experience problems with poor time synchronization, you can increase the window
from its default size of 3 permitted codes (one previous code, the current
code, the next code) to 17 permitted codes (the 8 previous codes, the current
code, and the 8 next codes). This will permit for a time skew of up to 4 minutes
between client and server.
Do you want to do so? (y/n) n

If the computer that you are logging into isn't hardened against brute-force
login attempts, you can enable rate-limiting for the authentication module.
By default, this limits attackers to no more than 3 login attempts every 30s.
Do you want to enable rate-limiting? (y/n) y
```

Vous avez terminé la configuration de votre appareil.

15 octobre 2025

4. Configurer votre Pare-feu

4.1 Situation

4.1.1 Description

Pour la configuration de notre Pare-feu, on va, dans le cas présent, partir du principe que vous avez un serveur avec l'OS et le pare-feu suivant : - OS : Ubuntu 24.04 LTS - Pare-feu : ufw 0.36.2

4.1.2 Informations complémentaires

Services actifs sur le serveur:

- SSH : port 2222
- Nginx : port 8080
- MySQL : port 3306

Informations réseau pour les scénarios:

- Réseau interne de l'entreprise : 10.0.0.0/8
- Réseau DMZ : 172.16.0.0/16
- Serveur d'administration : 192.168.1.100
- Serveur de backup : 192.168.1.50
- Développeurs : 10.10.10.0/24

4.2 Installation

Si vous n'avez pas UFW, voici la commande pour l'installer :

Bash

```
sudo apt install ufw -y
```

4.3 Configurer

Dans un premier temps, vous allez passer en root :

Bash

```
sudo -i
```

4.3.1 Bloquer tout le traffic

On va autoriser le traffic sortant et bloquer le traffic entrant via la commande suivante :

Bash

```
ufw default deny incoming
ufw default allow outgoing
```

4.3.2 Autoriser qu'une ip à se connecter

On souhaite autoriser la connexion par SSH uniquement au serveur d'administration, pour se faire, on va faire la commande suivante :

Bash

```
ufw allow from 192.168.1.100 to any port 2222
```

4.3.3 Autoriser une plage d'adresse IP

On souhaite que le réseau DMZ accède au serveur MySQL.

Pour cela, il suffit de faire la commande suivante :

Bash

```
ufw allow from 172.16.0.0/16 to any port 3306
```

Dans ce cas, tous les IP commençant par 172.16 pourront accdéer au port 3306.

Pour les IP a de classe A, on serait sur du 10.0.0.0/8 et de classe C, 192.168.1.0/24.

Si on décide par exemple de n'autoriser que 10.10.10.0/8, Cela veut dire que seul les IP commençant par 10.10.10 peuvent utiliser le port.

4.3.4 Bloquer une IP ou une plage d'adresse IP

On veut par exemple, empêcher au réseau interne de l'entreprise d'accéder au port MySQL sauf pour les personnes autorisées.

Bash

```
ufw deny from 192.168.1.0/24 to any port 3306
```

Personne avec une IP commençant par l'IP 192.168.1 ne pourra accéder au port SSH sauf les personnes autorisées spécifiquement comme l'IP `192.168.1.100`.

4.3.5 Limiter un port

Si on veut éviter le bruteforce d'un port, par exemple, SSH, on peut le faire via la commande suivante :

Bash

```
ufw limit 2222
```

4.3.6 Supprimer une rêgle

Supposons que vous ayez par erreur mit l'IP `192.168.10.100` au lieu du `192.168.1.100` et que vous voulez supprimer la rêgle.

Pour se faire, on peut supprimer une rêgle via la commande suivante :

Bash

```
ufw delete allow from 192.168.10.100 to any port 2222
```

4.3.7 Mettre en place des logs

Vous voulez mettre en place des logs pour surveiller le traffic avec ufw, cela est possible.

Puis on va définir son niveau de logs, il y en a 3 - low - medium - high

Dans notre cas, on va le définir sur medium :

Bash

```
ufw logging medium
```

4.4 Sauvegarder la configuration UFW

Pour se faire, vous allez dans un premier temps sauvegarder la liste des rêgles :

Bash

```
ufw status > ufw-rules-backup.txt
```

Dans un secon temps, vous allez sauvegarder votre configuration complète :

Bash

```
tar -cvzf ufw-backup.tar.gz /etc/ufw
```

4.5 Restaurer la configuration UFW

Dans un premier temps, désactiver UFW le temps de la restauration :

Bash

```
ufw disable
```

Dans un deuxième temps, vous allez extraire les fichiers de configuration :

Bash

```
tar -xvzf ufw-backup.tar.gz -C /etc/ufw
```

Dans un troisième temps, vous allez définir les permissions sur les fichiers :

Bash

```
chown -R root:root /etc/ufw
chmod -R 644 /etc/ufw
```

Dans un quatrième temps, vous allez réactiver votre pare-feu :

Bash

```
ufw enable
```

Dans un cinquième temps, vous allez voir si les rêgles ont été chargés :

Bash

```
ufw status
```

Dans un denier temps, vous allez tester votre configuration et vérifier si tout fonctionne.

28 octobre 2025

Tables des figures