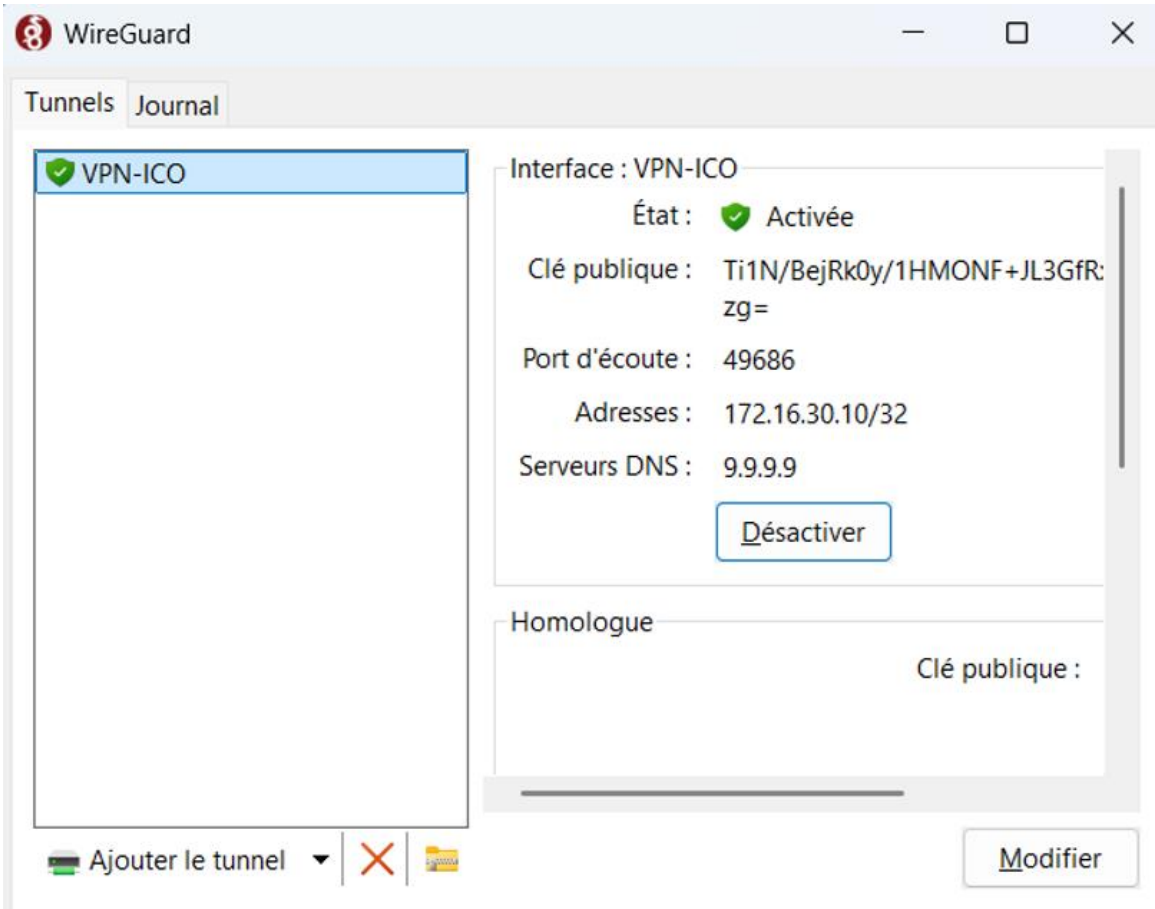
PROCÉDURE : Déploiement d’un site Laravel sur un serveur distant (Proxmox)

1. Connexion à l’infrastructure

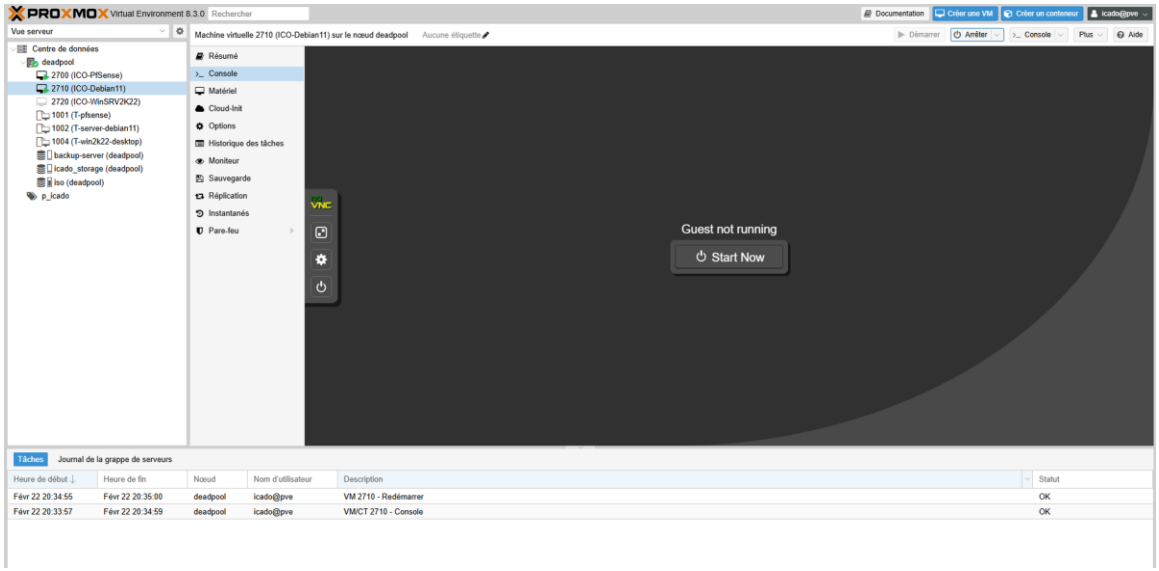
• Se connecter au VPN Wireguard pour accéder au réseau privé.



• Accéder à l’interface Proxmox via : http://10.0.0.1:8006

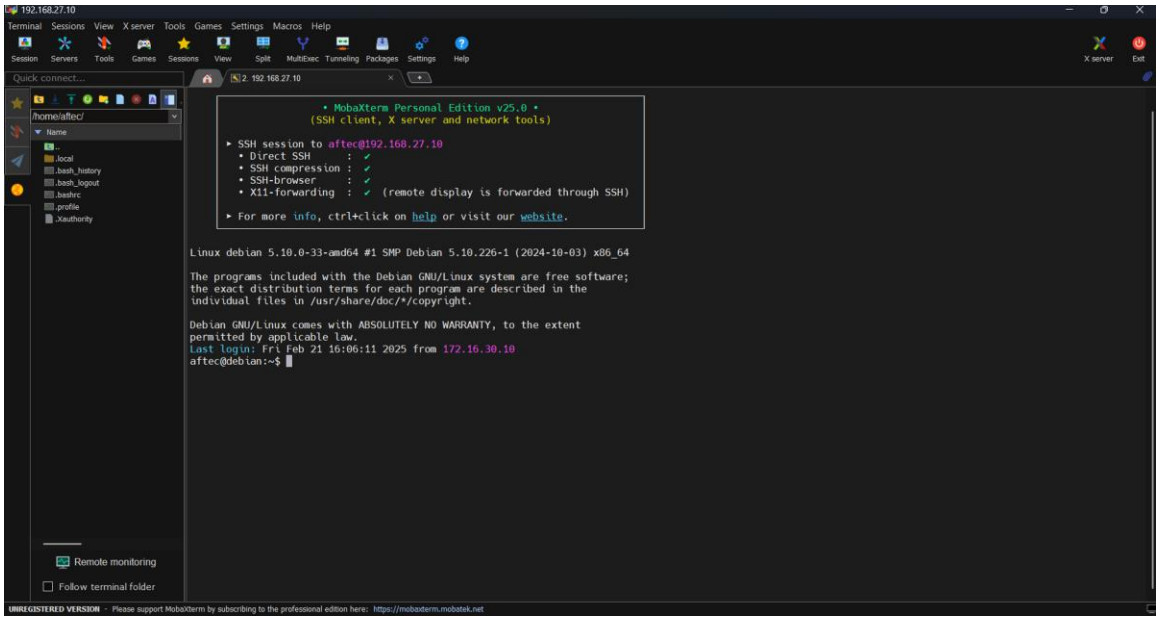
• Connexion avec les identifiants fournis

• Démarrer la machine Ubuntu déjà créée.



2. Préparation de la machine Ubuntu

• Connexion SSH avec MobaXterm ou terminal : ssh aftec@192.168.27.10



• Se connecter au compte root avec : su

• Mise à jour de la machine :

sudo apt update && sudo apt upgrade -y

• Installation des paquets nécessaires :

apt install -y nginx mariadb-server mariadb-client php-fpm php-mysql unzip curl git

3. Sécuriser MariaDB

• Exécuter la commande : mysql_secure_installation

• Suivre les étapes : définir un mot de passe root, supprimer les utilisateurs anonymes, interdire root à distance, supprimer la base test, recharger les privilèges.

```
root@debian:~# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
 ... Success!


You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!


By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.
```

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@debian:~#
```

4. Base de données pour Laravel

• mysql -u root -p

• Créer la base : CREATE DATABASE mangaverse CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

• Créer l'utilisateur : CREATE USER 'adminmanga'@'localhost' IDENTIFIED BY 'naruto';

• Donner les droits : GRANT ALL PRIVILEGES ON mangaverse.* TO 'adminmanga'@'localhost';

• FLUSH PRIVILEGES; puis EXIT

```
root@debian:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 40
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)]> CREATE DATABASE mangaverse CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> CREATE USER 'adminmanga'@'localhost' IDENTIFIED BY 'naruto';
Query OK, 0 rows affected (0,006 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON mangaverse.* TO 'adminmanga'@'localhost';
Query OK, 0 rows affected (0,016 sec)

MariaDB [(none)]>
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,003 sec)

MariaDB [(none)]> EXIT;
Bye
root@debian:~#
```

5. Installation PHP 8.2 (recommandé)

• Ajouter le dépôt Sury :

apt install -y ca-certificates apt-transport-https software-properties-common wget curl lsb-release

curl -sSL https://packages.sury.org/php/README.txt | bash -x

apt update

• Installer PHP 8.2 avec les extensions :

apt install -y php8.2 php8.2-fpm php8.2-cli php8.2-mysql php8.2-xml php8.2-mbstring php8.2-curl php8.2-zip php8.2-bcmath php8.2-tokenizer php8.2-gd unzip

• systemctl restart php8.2-fpm

• systemctl enable php8.2-fpm

```
root@debian:/var/www/mangaverse2.0# php -v
PHP 8.2.27 (cli) (built: Dec 24 2024 06:15:50) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.27, Copyright (c) Zend Technologies
    with Zend OPcache v8.2.27, Copyright (c), by Zend Technologies
root@debian:/var/www/mangaverse2.0#
```

6. Configuration de Nginx

• Créer le dossier Laravel : mkdir -p /var/www/laravel

• Créer fichier de conf : nano /etc/nginx/sites-available/laravel

• Ajouter la configuration adaptée à Laravel (avec root vers public/)

• Activer le site (lien symbolique) : ln -s /etc/nginx/sites-available/laravel /etc/nginx/sites-enabled/

• Redémarrer nginx : systemctl restart nginx

• Si Apache est actif : systemctl stop apache2 && systemctl disable apache2

```
aftec@debian:/etc/nginx/sites-available$ cat mangaverse2.0
server {
    listen 80;
    server_name mangaverse.blog www.mangaverse.blog;

    # Redirige tout le trafic HTTP vers HTTPS
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name mangaverse.blog www.mangaverse.blog;

    root /var/www/mangaverse2.0/public;
    index index.php index.html;

    ssl_certificate /etc/letsencrypt/live/mangaverse.blog/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/mangaverse.blog/privkey.pem;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.2-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\.ht {
        deny all;
    }
}
```

(screen ultérieur au déploiement, redirection https et dns available)

7. Clonage du projet Laravel

• cd /var/www

• git clone https://github.com/IlanCado/mangaverse2.0.git

• Vérifier les permissions : chown -R www-data:www-data /var/www/mangaverse2.0

```
root@debian:/var/www# ls
html
root@debian:/var/www# git clone https://github.com/IlanCado/mangaverse2.0.git
Clonage dans 'mangaverse2.0'...
remote: Enumerating objects: 594, done.
remote: Counting objects: 100% (594/594), done.
remote: Compressing objects: 100% (367/367), done.
remote: Total 594 (delta 257), reused 532 (delta 200), pack-reused 0 (from 0)
Réception d'objets: 100% (594/594), 4.03 Mio | 14.59 Mio/s, fait.
Résolution des deltas: 100% (257/257), fait.
root@debian:/var/www# ls
html  mangaverse2.0
root@debian:/var/www#
```

8. Configuration de Laravel

• cp .env.example .env

• Modifier le fichier : nano .env (adapter nom BDD, utilisateur, mot de passe)

• Générer la clé d'application : php artisan key:generate

• Appliquer les migrations : php artisan migrate

9. Installation de Composer

• apt install composer -y

• Installer manuellement composer 2.2 si nécessaire

```
root@debian:/var/www/mangaverse2.0# composer --version
Do not run Composer as root/super user! See https://getcomposer.org/root for details
Continue as root/super user [yes]? yes
Composer version 2.8.5 2025-01-21 15:23:40
PHP version 8.2.27 (/usr/bin/php8.2)
Run the "diagnose" command to get more detailed diagnostics output.
root@debian:/var/www/mangaverse2.0#
```

10. Fichiers statiques et permissions

• chown -R www-data:www-data /var/www/mangaverse2.0

• chmod -R 775 /var/www/mangaverse2.0/storage

• chmod -R 775 /var/www/mangaverse2.0/bootstrap/cache

• php artisan storage:link

11. Importer une base de données (bonus)

• Exporter en SQL depuis PhpMyAdmin

• scp fichier.sql root@192.168.27.10:/var/www/

• mysql -u adminmanga -p

DROP DATABASE mangaverse; CREATE DATABASE mangaverse;

mysql -u adminmanga -p mangaverse < /var/www/mangaverse.sql

12. Vérification

Redémarrer les services php, nginx

Tester l'URL http://192.168.27.10