

Bureau des petites annonces : Mode opératoire de la mise en place de l'environnement de recette/production

Le 15 février 2024

Objet du document

L'objectif de ce document est décrit les étapes réalisées lors de la mise en place de l'environnement de recette/production.
--

Niveau de confidentialité du document			
--	--	--	--

0 – Public	1 – Interne	2 – Restreint	3 – Confidentiel
	X		

Fiche de révision			
--------------------------	--	--	--

Version	Date	Auteur	Nature
0.1	04/12/2023	BPA	Création du document et rédaction du document
0.2	22/12/2023	BPA	Finalisation du document
0.3	29/01/2024	BPA	Modification de la partie du clone git
0.4	06/02/2024	BPA	Ajustement du document la version production
0.5	15/02/2024	BPA	Ajustement => recette devient la production

L'incrémentation de la version s'effectue :
 en version (0.x) pour les versions de travail, non diffusées à l'extérieur,
 en version majeure (1.0) pour les modifications fonctionnelles,
 en version mineure (x.1) pour les modifications de forme.

PREAMBULE

Ce document sera structuré de la manière suivante :

Prérequis de l'environnement de la recette.

Initialisation des comptes, répertoires et droits utilisateurs.

Installation et configuration des prérequis pour mettre en place la recette.

Clone du projet avec GitBlit.

Migrations de la base de données complet en recette.

Tests de la recette.

Ce document est destiné aux membres du projet BPA.

TABLE DES MATIERES

1	PREREQUIS DE L'ENVIRONNEMENT.....	5
2	COMPTES, REPERTOIRES ET DROITS	6
3	GIT CLONE DU PROJET.....	8
4	INSTALLATION ET CONFIGURATION DE LA RECETTE	10
4.1	vendor, node modules et .env.....	10
4.2	Mise en place de la BDD	12
4.3	Wine	17
5	MIGRATION DE LA BASE DE DONNEES.....	19
6	TESTS DE L'APPLICATION	21

1 PREREQUIS DE L'ENVIRONNEMENT

Pour commencer à configurer notre recette, il faut avoir :

Une machine virtuelle sur Debian installé sur un serveur distant.

Avoir FileZilla d'installer sur le poste.

Avoir accès à la VM avec un lien ssh fourni et un compte existant à notre nom fourni également avec un mot de passe temporaire.

- En ligne de commande terminal (peu importe l'emplacement), entrer cette commande : `ssh <nom_utilisateur>@192.168.2.9`

```
C:\Users\arthur.foucault>ssh arthur@192.168.2.9
arthur@192.168.2.9's password: █
```

- Entrer le mot de passe temporaire.

A ce stade vous êtes connectés à la VM avec votre compte donc vous devriez avoir quelque chose qui ressemble à ça :

```
C:\Users\arthur.foucault>ssh arthur@192.168.2.9
arthur@192.168.2.9's password:
Linux s-mtb-deb-bpa 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) 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: Thu Nov 30 13:50:49 2023 from 192.168.8.23
arthur@s-mtb-deb-bpa:~$ █
```

Vous serez connecté sur votre répertoire dans « `/home/<nom_utilisateur>` » :

```
arthur@s-mtb-deb-bpa:~$ ls /home
alexandre alexis antoine arthur bpa evan fgadmin furkan guillaume marceau mtbadmin nastassja sina tony
arthur@s-mtb-deb-bpa:~$
```

Modifier le mot de passe temporaire par un nouveau mot de passe sur votre compte avec la commande “`passwd`” et suivez les instructions.

Vous devez posséder les droits nécessaires à la manipulation de l'environnement “`sudo`” si non demander à un administrateur les droits.

A ce stade, vous avez les prérequis nécessaires à la manipulation de la recette.

2 COMPTES, REPERTOIRES ET DROITS

Pour créer un nouveau compte sur la recette :

- Entrer cette commande “sudo adduser nouveau_utilisateur”.
- Ecrire votre nom complet.
- Ecrire votre mot de passe.
- Ignorer les informations complémentaires en tapant “Entrée”.
- Créer des comptes pour chaque membre du projet.

Les commandes “exit” et “logout” permettent de se déconnecter du compte actuel.

Par défaut les nouveaux utilisateurs n'ont pas les droits administrateurs pour faire des commandes “sudo”.

Pour donner les droits administrateurs à un nouvel utilisateur :

- Entrer cette commande “sudo usermod -aG sudo nom_utilisateur”, il ajoute un utilisateur au groupe “sudo”.
- Vérifier en visualisant la liste des groupes avec soit la commande “getent group” ou soit la commande “cat /etc/group”.

Un fois les droits accordés, on peut commencer à mettre en place la configuration de l'environnement.

Il faut créer le répertoire du projet “/home/recette”:

- Faire la commande “sudo mkdir /home/recette” pour créer le répertoire.
- Si jamais vous créer le répertoire au mauvais endroit alors voici la commande pour supprimer un répertoire : `sudo rm -r nom_du_répertoire`

Il faut configurer les droits d'utilisateurs sur ce chemin “/home/recette” pour qu'on puisse avoir l'autorisation de lancer des commandes git :

- Ajouter un nouveau groupe “recette” : `sudo addgroup recette`
- Donner les droits sur le répertoire “/home/recette” à ce groupe : `sudo chown -R root:recette /home/recette`
- Donner également les droits de lecture, d'écriture et d'exécution à ce groupe sur ce répertoire : `sudo chmod -R g+rwX /home/recette`
- Installer « acl » qui permet d'appliquer les mêmes du groupe sur les nouveaux dossiers et fichiers créer dans un répertoire en particulier : `sudo apt-get install acl`
- Entrer cette commande qui applique les mêmes droits que le groupe « recette » par défaut aux nouveaux dossiers et fichiers du répertoire « /home/recette » : `sudo setfacl -R -m default:group:recette:rwX /home/recette/`
- Vous pouvez vérifier les droits qu'à la recette maintenant avec cette commande : `getfacl /home/recette`

```
arthur@s-mtb-deb-bpa:/home/recette/PetitesAnnonces/storage/logs$ getfacl /home/recette
getfacl : suppression du premier « / » des noms de chemins absolus
# file: home/recette
# owner: root
# group: recette
# flags: -s-
user::rwx
group::rwx
other::rwx
default:user::rwx
default:group::rwx
default:group:recette:rwx
default:mask::rwx
default:other::rwx
```

Si vous ne voyez pas les changements s'appliqués alors déconnectez-vous et reconnectez-vous pour voir les changements.

- Ajouter tous les utilisateurs au groupe "recette" avec cette commande :
sudo adduser <nom_utilisateur> recette
- Si jamais pour supprimer un utilisateur du groupe : sudo deluser <nom_utilisateur> recette

Maintenant que les droits et répertoire sont mise en place. On va voir pour mettre en place l'environnement avec un clone du projet et mysql.

3 GIT CLONE DU PROJET

Pour faire un clone du repository GitBlit de “PetitesAnnonces” en recette, il faut :

Installer git si ce n’est pas déjà fait avec cette commande “sudo apt-get install git”.

Vérifier avec la commande “git --version” si Git s’est bien installer.

Dû à des modifications de serveur pour les dépôts nous utiliserons une configuration avec http au lieu du SSH.

Si vous devez modifier le lien du dépôt sans refaire le clone (S’il existe déjà) alors suivez ces étapes :

- Entrer cette commande dans le repertoire du projet : `sudo git remote set-url origin http://192.168.2.121:8080/mtb/padawans/PetitesAnnonces.git`
- Vérifié le remote (ce qui permettra de configurer le compte GitLab qui sera utilisé sur le lien) : `sudo git remote show origin` :

```

arthur@s-mtb-deb-bpa:/home/recette/PetitesAnnonces$ git remote show origin
Username for 'http://192.168.2.121:8080': afo@mtb111.fr
Password for 'http://afo@mtb111.fr@192.168.2.121:8080':
* distant origin
  URL de rapatriement : http://192.168.2.121:8080/mtb/padawans/PetitesAnnonces.git
  URL push : http://192.168.2.121:8080/mtb/padawans/PetitesAnnonces.git
  Branche HEAD : main
  Branche distante :
    main suivi
  Branches locales configurées pour 'git pull' :
    main fusionne avec la distante main
    master fusionne avec la distante master
  Référence locale configurée pour 'git push' :
    main pousse vers main (le local n'est pas à jour)
arthur@s-mtb-deb-bpa:/home/recette/PetitesAnnonces$

```

La branch main se remet toute seule donc vous pouvez directement effectué des fetch, pull...

De principe, vous avez les droits nécessaires sur le répertoire “/home/recette” pour effectuer les étapes suivantes :

Placez-vous dans “/home/recette”: `cd /home/recette`

Lancer cette commande : `sudo git clone http://192.168.2.121:8080/mtb/padawans/PetitesAnnonces.git`

- On vous demandera de vous authentifiez à votre compte GitLab la première fois puis pour toute action, on ne vous demandera rien :

Une fois le clone fait, il aura déjà fait une récupération (pull) du dépôt distant donc pas besoin de faire un pull dessus la première fois donc placez-vous dans le répertoire du repository du clone qui devrait être “/home/recette/PetitesAnnonces”.

Vérifier les branches disponibles avec cette commande : `sudo git branch -a`

```

arthur@s-mtb-deb-bpa:/home/recette/PetitesAnnonces$ sudo git branch -a
* main
remotes/origin/main

```


- Là où est placé le petit « * » veut dire sur quel branch vous êtes actuellement.
- **Important** : Attention au tiret du “a” car copier-coller la commande directement peut donner un tiret différent en console. Donc vaut mieux écrire à la main

Voici la commande qui permet de changer de branch (si besoin) : `sudo git checkout <nom_de_la_branch>`

S’assurer que le clone est à jour avec cette commande : `sudo git pull origin <nom_de_la_branch>`

- “origin” est le remote qui contient le lien http du dépôt.

Nous avons notre clone du projet qui est prêt dès à présent.

4 INSTALLATION ET CONFIGURATION DE LA RECETTE

4.1 VENDOR, NODE MODULES ET .ENV

Une fois le projet récupéré mettez-vous à la racine du projet.

Mettez à jour les bibliothèques “vendor” avec la commande :

```
sudo composer install
```

Attention : Si des fichiers vendor spécifiques ont été modifier manuellement et qu’ils ne sont pas dans les configurations laravel alors faire un backup du vendor pour procéder à une intégration manuelle pour remettre ces fichiers dans les vendor après la commande.

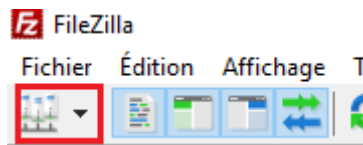
Il faut ensuite faire pareil avec Node en faisant :

```
sudo npm install
```

Pour le fichier “.env”, on utilise le logiciel FileZilla pour transférer le fichier qui est non déposable sur le dépôt :

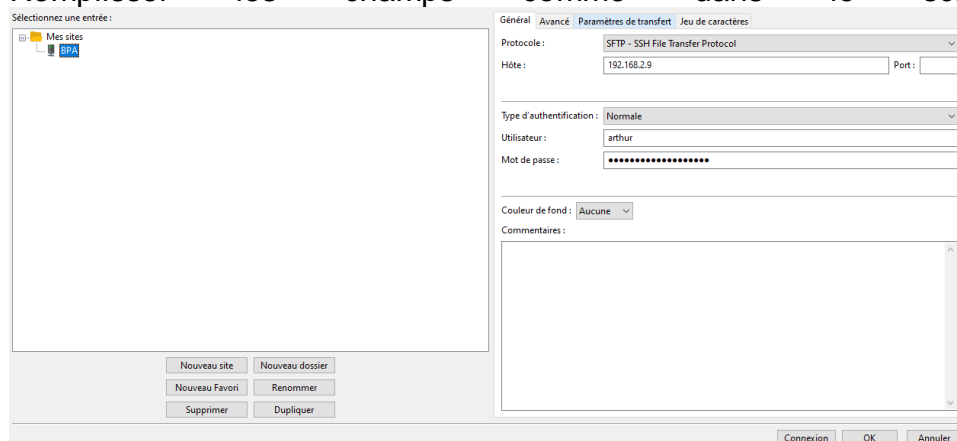
Ouvrer FileZilla.

Aller dans le gestionnaire de sites :



Faite “Nouveau site” et nommer le “BPA”.

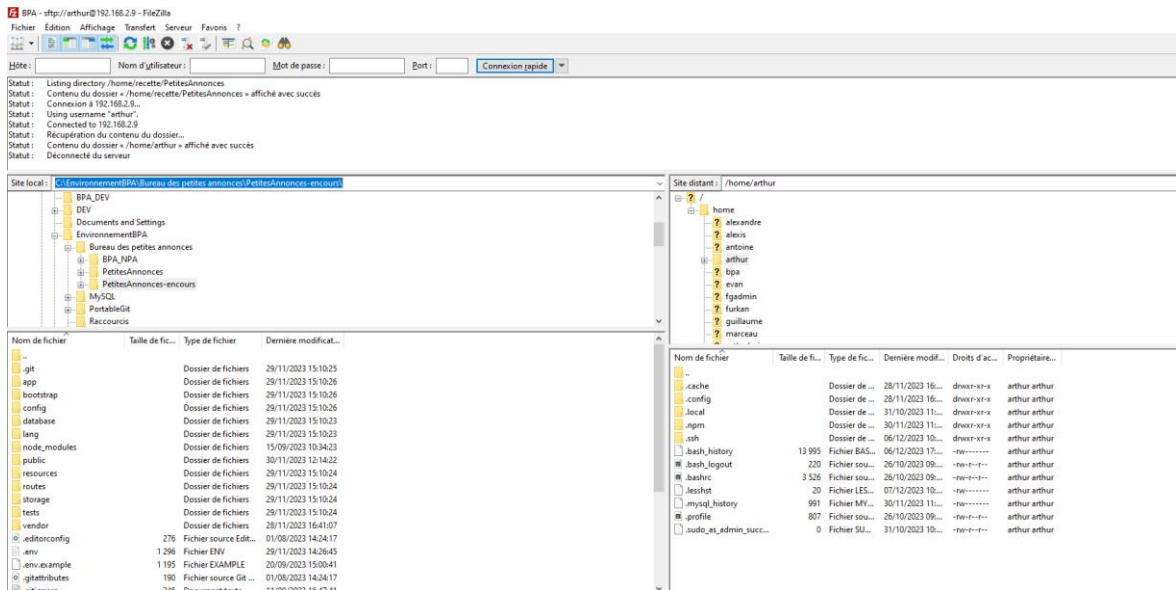
Remplir les champs comme dans le screen ci-dessous :



- Assurez-vous que le protocole est bien SFTP.

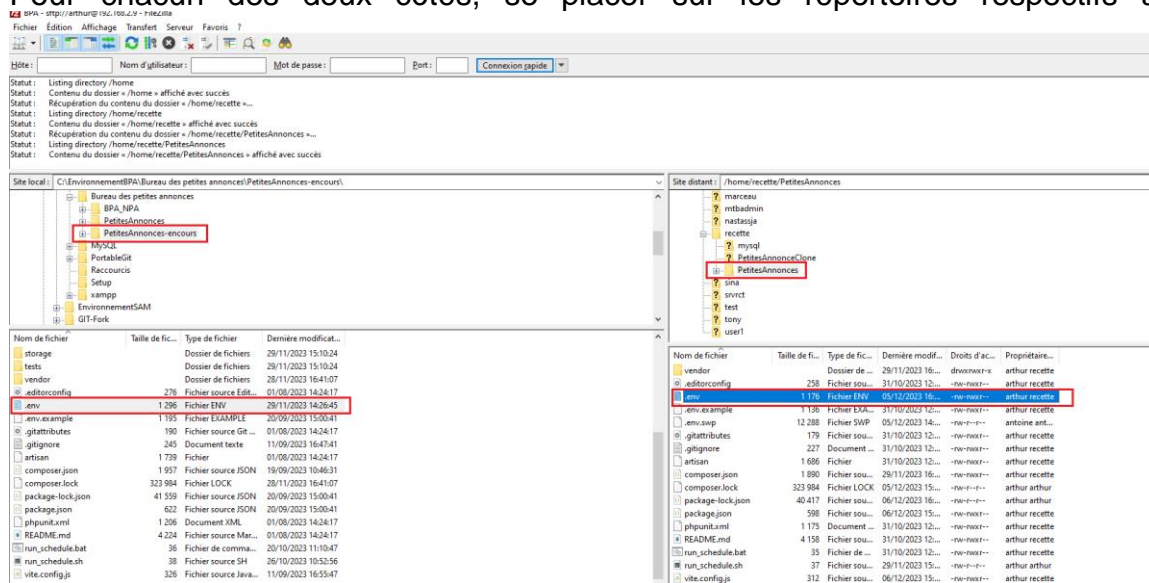
Puis faite connexion.

Normalement vous verrez l’interface suivant :



- A gauche : votre PC en local
- A droite : la machine virtuelle

Pour chacun des deux côtés, se placer sur les répertoires respectifs au projet :



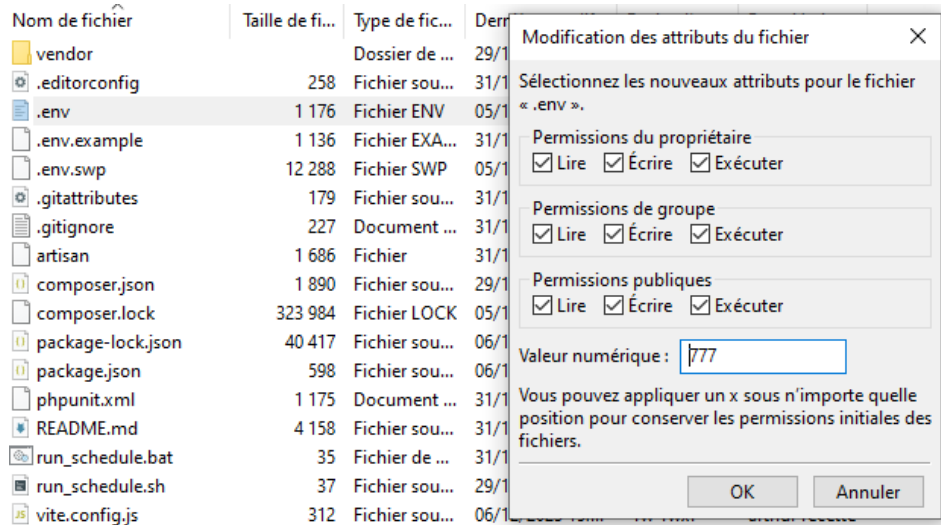
– Pour transférer le fichier du PC local sur la VM, faite un clic droit sur le fichier “.env” et faite “Téléverser”.

- Si le transfert échoue à cause d’une autorisation refusée alors suivre ces étapes avant de refaire un téléversement de fichier :
 - En terminal de la VM sur lequel vous êtes actuellement connecté, aller dans le répertoire du fichier qu’on veut téléverser.
 - Lancer cette commande qui donnera tous les droits sur ce répertoire : `sudo chmod 777 ".env"`

On ne peut lancer un sudo via FileZilla.

- Dans FileZilla, relancer une nouvelle connexion au site de la VM pour actualiser les changements.

- Du côté du site, aller dans le répertoire du fichier « .env » et clic droit dessus puis faite « Droits d'accès au fichier... »
- Dans le champ « valeur numérique », mettez la valeur « 777 » et faite ok :



- Refaite un téléversement du fichier qui devrait fonctionner cette fois.

Faire le téléversement des fichiers suivants :

- vendor\laravel\framework\src\Illuminate\Notifications\resources\views\email.blade.php
- vendor\laravel\framework\src\Illuminate\Auth\Notifications\ResetPassword.php
- vendor\laravel\framework\src\Illuminate\Auth\Notifications\VerifyEmail.php

4.2 MISE EN PLACE DE LA BDD

Il faut ensuite pouvoir créer une BDD.

Pour ce faire il faut installer : MySQL, Apache, phpMyAdmin et ufw.

Pour « MySQL » :

- `sudo apt update`
- `sudo apt install mysql-server`

Pour « Apache » :

- `sudo apt update`
- `sudo apt install apache2`

Pour “phpMyAdmin”:

- `sudo apt update`
- `sudo apt install phpmyadmin`
- `sudo systemctl restart apache2`

Pour “ufw”:

- `sudo apt-get update`
- `sudo apt-get install ufw`

- `sudo ufw enable`
- Verifier les installations:
 - MySQL: `sudo mysql --version`
 - Apache: `/usr/sbin/apache2 -v`
 - phpMyAdmin:
 - Verifier si le paquet est installé: `dpkg -l | grep phpMyAdmin`
 - Vérifier le répertoire d'installation : `ls /etc/phpMyAdmin`

Des commandes utiles :

- On peut lancer apache avec la commande : `sudo service apache2 start`
- Vous pouvez visualiser le statut de apache2 : `sudo service apache2 status`
- Pour arrêter apache : `sudo service apache2 stop`
- Pour redémarrer apache : `sudo service apache2 restart`
- On peut lancer MySQL avec la commande : `sudo service mysql start`
- Vous pouvez visualiser le statut de MySQL : `sudo service mysql status`
- Pour arrêter MySQL: `sudo service mysql stop`
- Pour redémarrer MySQL : `sudo service mysql restart`
- Avec ces commandes lancer apache2 et mysql.

Important : Parfois certaines modifications d'apache et MySQL demandent à ce qu'on relance la connexion ssh avec votre compte pour voir les modifications s'appliqués.

Il faut configurer le service MySQL avec phpMyAdmin :

- Lancer la commande « `cd /etc/apache2/sites-available` ».
- Vérifier que le fichier « `000-default.conf` » existe avec un « `ls` ». (De principe il existe quand on installe apache)
- Faire un « `sudo nano 000-default.conf` ».
- Remplacer tout le contenu de ce fichier par celui-ci et sauvegarder le fichier :

```
<VirtualHost *:80>
```

```
    ServerAdmin webmaster@localhost
```

```
    DocumentRoot /usr/share/phpmyadmin
```

```
<Directory /usr/share/phpmyadmin>
```

```
    Options FollowSymLinks
```

```
    DirectoryIndex index.php
```

```
    AllowOverride All
```

```

    Require all granted
</Directory>
ErrorLog ${APACHE_LOG_DIR}/phpmyadmin_error.log
CustomLog ${APACHE_LOG_DIR}/phpmyadmin_access.log combined
</VirtualHost>

```

```

<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /usr/share/phpmyadmin
    <Directory /usr/share/phpmyadmin>
        Options FollowSymlinks
        DirectoryIndex index.php
        AllowOverride All
        Require all granted
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/phpmyadmin_error.log
    CustomLog ${APACHE_LOG_DIR}/phpmyadmin_access.log combined
</VirtualHost>

```

- Aller dans ce répertoire « cd /etc/apache2/sites-enabled » puis faire un « ls » :

```

arthur@s-mtb-deb-bpa:/etc/apache2/sites-enabled$ ls
000-default.conf  default-ssl.conf

```

- Si le fichier n'est pas dans ce répertoire en couleur cyan alors faire cette commande qui active le site apache : `sudo a2ensite default-ssl.conf`
 - Commande pour la désactivation d'un site apache : `sudo a2dissite default-ssl.conf`
- Assurez-vous aussi que le module ssl est activé dans ce répertoire avec la commande : `sudo a2enmod ssl`
- Entrer les commandes suivantes pour autoriser les ports du serveur laravel et celui d'apache :
 - `sudo ufw allow 8000`
 - `sudo ufw allow 80`
- Vérifier les ports autorisés avec « `sudo ufw status` ».
- Relancer apache.

Il se peut que le mot de passe de root ne se paramètre pas correctement donc suivez les étapes suivantes :

- Arrêter le serveur mysql « `sudo service mysql stop` »
- Puis entrer les commandes suivantes :
 - Mode safe de mysql: `sudo mysqld_safe --skip-grant-tables &`
 - `mysql -u root`
 - `USE mysql ;`

- SET PASSWORD FOR 'root'@'localhost' = PASSWORD('nouveau-mot-de-passe');
 - FLUSH PRIVILEGES;
 - exit ;
- Arrêter le mode safe de mysql et relancer le :
- sudo service mysql stop
 - sudo service mysql start
- Si jamais l'arrêt ne se fait alors tenter ces commandes :
- ps aux | grep mysqld

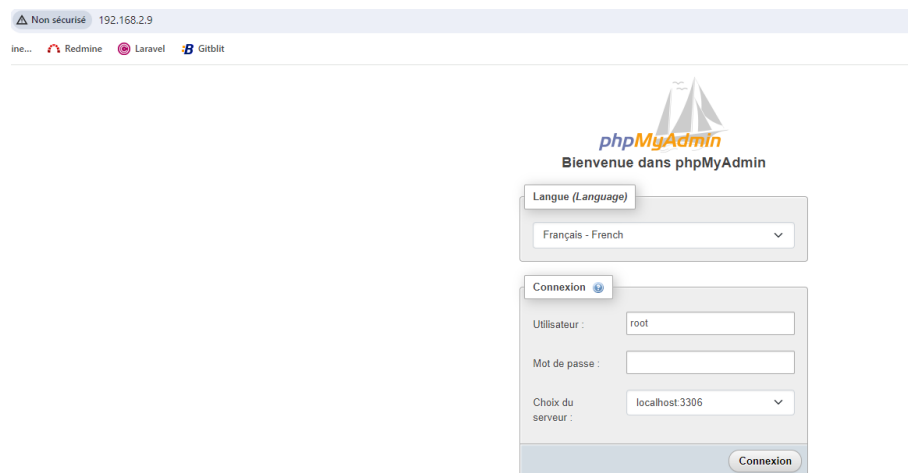
```
arthur@s-mtb-deb-bpa:/etc/phpmyadmin$ ps aux | grep mysqld
arthur      50102  0.0  0.0  6352  2056 pts/5    S+   16:45   0:00 grep mysqld
```

- « sudo kill <PID> » où <PID> est ici « 50102 » sur le screen ci-dessus.

Une fois réaliser on peut contrôler si cela fonction tapant <http://192.168.2.9> (IP de la VM). Cela nous amène sur « phpMyAdmin ».

Si on arrive sur cette page c'est que ça fonctionne.

Connectez-vous à « phpMyAdmin » avec root en localhost :



- Demander le mot de passe de root à un administrateur.

Ensuite, on va créer un nouvel utilisateur mysql (nom d'utilisateur « PetitesAnnonces ») car on ne veut pas se servir de root donc pour ça :

- Se connecter à MariaDB avec root avec la commande « mysql -u root -p »
- Entrer les commandes suivantes :
 - USE PetitesAnnonces;
 - CREATE USER 'PetitesAnnonces' identified by '<mettre le mot de passe de la base de données>';
 - GRANT ALL PRIVILEGES ON PetitesAnnonces.* TO 'PetitesAnnonces'@'localhost';

- FLUSH PRIVILEGES;
- Vérifier que l'utilisateur a bien été créé :
 - Faire les commandes suivantes comme sur les screens :

```
MariaDB [(none)]> SELECT User, Host FROM mysql.user;
```

User	Host
PetitesAnnonces	localhost
mariadb.sys	localhost
mysql	localhost
phpmyadmin	localhost
root	localhost

```
5 rows in set (0,002 sec)
```

```
MariaDB [(none)]> SHOW GRANTS FOR PetitesAnnonces@localhost;
```

Grants for PetitesAnnonces@localhost
GRANT USAGE ON *.* TO `PetitesAnnonces`@`localhost` IDENTIFIED BY PASSWORD '*22EBDA36B57044D028008AA0E1B73124D2FEBBEF'
GRANT ALL PRIVILEGES ON `PetitesAnnonces`.* TO `PetitesAnnonces`@`localhost`

```
2 rows in set (0,001 sec)
```

- Tester l'utilisateur en vous connectant avec le nouvel utilisateur sur MariaDB et faire la commande « USE PetitesAnnonces ».
- Effectuer la commande suivante comme sur le screen :

```
MariaDB [PetitesAnnonces]> SELECT * FROM users;
```

id	name	email	role_id	password	updated_at	email_verified_at	remember_token
1	NBA	nicolas.bacoup@mtb111.fr	1	\$2y\$10\$qKNvK4TQRFWUJrn3yqnLg0Suo0LPzigd28RM4BSWIY21UtlXZa5bS	2023-11-30 14:24:23	2023-11-30 14:24:53	Vxze0EcaZw0bGrEbmq69e6wRybbcf8qL0m03SswE9ZLqwMmXAbpGAQI2Feh7
2	BOSC	fbo@mtb111.fr	1	\$2y\$10\$KIdiS827Jinx8/UPK0f3PuGFe6tmzU.cy/CnAtpPVZjA0ZPVhFC8C	2023-11-30 14:24:39	2023-11-30 14:25:54	MdI4MNbykXEvhPtAwQcgSDWVGdVibc9QjjLu06v17IDDaLNAY0q6Ap1xPYwJ
4	Serge	SERGEauMIEL@mtb111.fr	3	\$2y\$10\$KMum2Rg7ANn7.s4bxKbfEutCQFPguUL4j5mnzTKiZ08UgqPgNJzVm	2023-12-05 16:18:20	2023-12-05 16:18:20	NULL
5	Guillaume Samson	guillaume.samson@mtb111.fr	3	\$2y\$10\$KM3oFSSPSvTLHz5GHGEY7uEhDwC0gEjLHHBUK/TC.Q4ThxeNlivDC	2023-12-05 16:19:21	2023-12-05 16:19:21	NULL
6	Tony	tony.otto@mtb111.fr	3	\$2y\$10\$QU0LhaQhBvFmD/e6Fgj65uhN/11ZZ3p7J5ReouEXeMxdRMVRyeXxe	2023-12-06 16:48:45	2023-12-06 16:48:45	NULL

```
5 rows in set (0,001 sec)
```

Si vous voyez des résultats comme sur le screen, c'est que ça fonctionne correctement.

- Relancer le serveur mysql.

Aller ensuite dans le « .env » du projet :

- Aller dans le répertoire du clone à la racine du projet et faire « ls -a » car le « .env » est caché si on fait seulement un « ls »
- Ouvrir le fichier avec un « sudo nano .env ».
- Modifier ces lignes pour mettre ces informations :

APP_URL=http://192.168.2.9

DB_HOST=localhost

DB_DATABASE=PetitesAnnonces

DB_USERNAME= PetitesAnnonces

DB_PASSWORD= <mettre le mot de passe de la base de données>

La configuration de la base de données en recette est maintenant terminée.

Il faut tester avec une migration de la BDD en recette donc aller vers la section suivante pour ça.

4.3 TMUX

Tmux (Terminal Multiplexer) est un utilitaire en ligne de commande qui permet de gérer plusieurs sessions et fenêtres de terminal dans un seul terminal. Pour l'installer :

- Mettre à jour le gestionnaire de paquets : `sudo apt update`
- Installer tmux : `sudo apt install tmux`
- Vérifier la version : `tmux -V`

4.4 WINE

Wine est un outil linux/Debian qui permet d'exécuter des fichiers « .exe » (applications windows). Pour installer Wine sur une machine virtuelle Linux Debian, vous pouvez suivre ces étapes dans la VM Debian:

- Mettez à jour la liste des paquets en exécutant la commande suivante : `sudo apt update`
- Installez Wine en utilisant la commande suivante : `sudo apt install wine`
- Vous serez invité à confirmer l'installation en appuyant sur "O" (Oui).
- Une fois l'installation terminée, Wine sera installé sur votre machine virtuelle Debian. Vous pouvez vérifier la version de Wine installée en exécutant la commande suivante : `wine --version`

Quand vous affichez la version de Wine, on vous proposera d'installer « wine32 » qui est nécessaire :

```
arthur@mtb-deb-bpa:~$ wine --version
it looks like wine32 is missing, you should install it.
multiarch needs to be enabled first. as root, please
execute "dpkg --add-architecture i386 && apt-get update &&
apt-get install wine32:i386"
wine-8.0 (Debian 8.0~repack-4)
```

Donc suivez aussi ces étapes :

- Entrer les commandes sur la copie d'écran ci-dessus séparément :
 - `sudo dpkg --add-architecture i386`

- sudo apt-get update
 - sudo apt-get install wine32:i386 (cette commande va prendre du temps à s'installer)
- Vérifier la version de Wine : wine --version

Vous devriez avoir ce résultat à la fin :

```
arthur@s-mtb-deb-bpa:/home/recette/PetitesAnnonces/public/batchs$ wine --version
wine-8.0 (Debian 8.0~repack-4)
```

5 MIGRATION DE LA BASE DE DONNEES

Pour la migration de la base de données « PetitesAnnonces » en recette, il faut :

Avoir tous les fichiers de migrations pour la BDD dans le répertoire de l'application « PetitesAnnonces\database\migrations » :

```
> app
> bootstrap
> config
✓ database
  > factories
  ✓ migrations
    2014_10_12_000000_create_users_table.php
    2014_10_12_100000_create_password_resets_table.php
    2019_08_19_000000_create_failed_jobs_table.php
    2019_12_14_000001_create_personal_access_tokens_table.php
    2023_09_12_072626_create_messages_table.php
    2023_11_26_151158_create_parametre_table.php
    2023_11_26_151223_create_photo_annonce_table.php
    2023_11_26_151258_create_type_annonce_table.php
    2023_11_27_104717_create_role_table.php
    2023_11_27_104810_create_annonces_table.php
    2023_11_27_104829_create_contacts_table.php
    2023_11_27_104851_create_favoris_table.php
    2023_11_27_104921_create_filtres_utilisateur_table.php
    2023_11_28_151244_create_signalements_table.php
  > seeders
  .gitignore
> lang
> node_modules
```

Vérifier bien que tous soient bien écrit.

Vous envoyez toutes les modifications sur le dépôt distant avec un commit et vous récupérez sur le clone de la VM le commit.

Une fois les fichiers correctement récupérer en recette, aller sur l'interface de phpmyadmin et créer une base de données vide du nom de « PetitesAnnonces » :

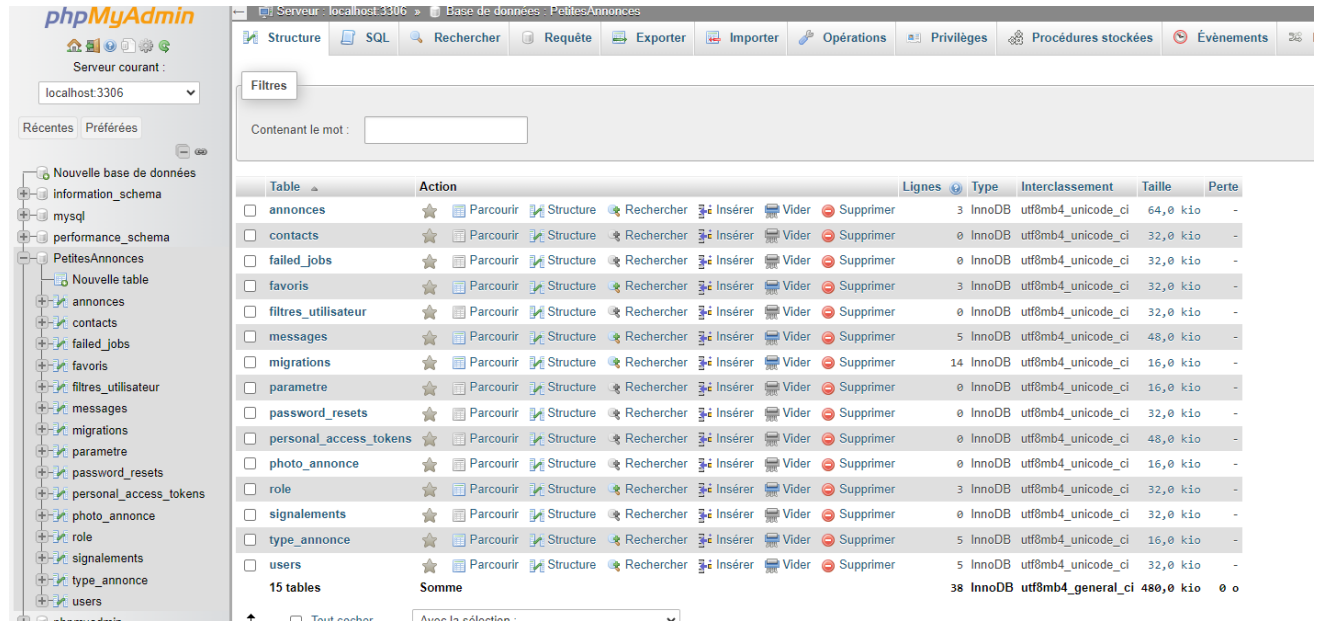


Rappel : la configuration de la connexion à la BDD « PetitesAnnonces » est déjà faite.

Dans le terminal en recette, se placer à la racine de l'application « /home/recette/PetitesAnnonces » et lancer une commande : `sudo php artisan migrate`

Vérifier qu'il n'y ait pas d'erreur. Si oui les corrigés et recommencé à partir de la création de la BDD.

Vous devriez avoir la base de données qui s'est construit :



The screenshot shows the phpMyAdmin interface for the 'PetitesAnnonces' database. The left sidebar lists the database structure, including tables like 'annonces', 'contacts', 'failed_jobs', 'favoris', 'filtres_utilisateur', 'messages', 'migrations', 'parametre', 'password_resets', 'personal_access_tokens', 'photo_annonce', 'role', 'signalements', 'type_annonce', and 'users'. The main panel displays a table of database tables with columns: Table, Action, Lignes, Type, Interclassement, Taille, and Perte. The table '15 tables' is highlighted, showing a total of 38 lines, InnoDB type, utf8mb4_general_ci collation, 480,0 kio size, and 0 o (other) space.

Table	Action	Lignes	Type	Interclassement	Taille	Perte
annonces	Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8mb4_unicode_ci	64,0 kio	-
contacts	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
failed_jobs	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
favoris	Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
filtres_utilisateur	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
messages	Parcourir Structure Rechercher Insérer Vider Supprimer	5	InnoDB	utf8mb4_unicode_ci	48,0 kio	-
migrations	Parcourir Structure Rechercher Insérer Vider Supprimer	14	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
parametre	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
password_resets	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
personal_access_tokens	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	48,0 kio	-
photo_annonce	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
role	Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
signalements	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
type_annonce	Parcourir Structure Rechercher Insérer Vider Supprimer	5	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
users	Parcourir Structure Rechercher Insérer Vider Supprimer	5	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
15 tables	Somme	38	InnoDB	utf8mb4_general_ci	480,0 kio	0 o

Lancer une commande : `sudo php artisan db:seed`

Cela devrait alimenter en données sur certaines tables nécessaires tel que « role », « type_annonce » et « parametre »

Maintenant il faut mettre la tâche automatisé "crontab" en place pour le fichier "run_schedule.sh" :

- Lancer la commande "`sudo crontab -e`", on vous demandera probablement de choisir un éditeur parmi une liste donc suivez les instructions puis relancer la commande derrière si l'éditeur ne s'ouvre pas directement. La commande se lance avec `sudo` donc pour définir cette tâche automatisée en tant qu'utilisateur « root ». (Pour des raisons de droits sur l'exécution des fichiers on préfère utiliser root)
- Ajouter cette ligne "`<minute> <heure> <jour> <mois> <jour_de_la_semaine> cd /home/recette/PetitesAnnonces && /home/recette/PetitesAnnonces/run_schedule.sh`" : ajuster les paramètres "<>" selon quand vous voulez que la tâche se lance. Exemple :

```
arthur@s-mtb-deb-bpa:/home/recette/PetitesAnnonces$ sudo crontab -l
0 9 * * 1 cd /home/recette/PetitesAnnonces && /home/recette/PetitesAnnonces/run_schedule.sh
```

Ici, la tâche se lance tous les lundis (valeur « 1 » sur <jour_de_la_semaine>) à 9h00.

- Normalement une fois que la tâche se lance à la date et heure prévu, vous pouvez vérifier si elle fonctionne avec cette commande : `sudo journalctl | grep "run_schedule.sh"`

Le fichier « run_schedule.sh » exécute une commande laravel qui se lance une fois sans répétition donc c'est la tâche CRON qui gère la répétition (une fois par semaine le lundi à 9h00).

6 TESTS DE L'APPLICATION

Maintenant que tout est configuré alors lancer l'application avec la commande en recette dans le répertoire de l'application `"/home/recette/PetitesAnnonces"` :

- `sudo npm start`

Cette commande regroupe « `php artisan serve --host=192.168.2.9` » et « `npm run dev -- host=192.168.2.9` »

Accéder à l'application avec le lien « `http://192.168.2.9:8000/` » en local sur n'importe poste.

Pour faire en sorte que le serveur laravel reste lancer même quand vous fermez votre terminal alors suivez ces étapes :

- En étant connecté à la VM en SSH, aller dans le répertoire `"/home/recette/PetitesAnnonces"`.
- Lancer une session tmux avec cette commande « `sudo tmux new -s laravel_session` »
→ Cela va créer une session tmux sous le nom « `laravel_session` » avec root donc tout utilisateur pourra y accéder.
- Faire la commande « `sudo npm start` »
- Sortez de la session en faisant « `Ctrl + B` » puis « `d` » au clavier → Cela vous ramène à votre terminal normal
- Fermez votre terminal en local et constatez si l'application tourne toujours.

Pour arrêter la session tmux :

- En étant connecté à la VM en SSH, aller dans le répertoire `"/home/recette/PetitesAnnonces"`.
- Visualisez toutes les sessions qui existe avec tmux en faisant la commande « `sudo tmux ls` » et repérer la session qui vous intéresse.
- Rentrez dans la session tmux avec cette commande « `sudo tmux attach -t <nom_de_la_session>` »
- Faire les commandes « `Ctrl + c` » et « `exit` », ce qui arrêter la session.