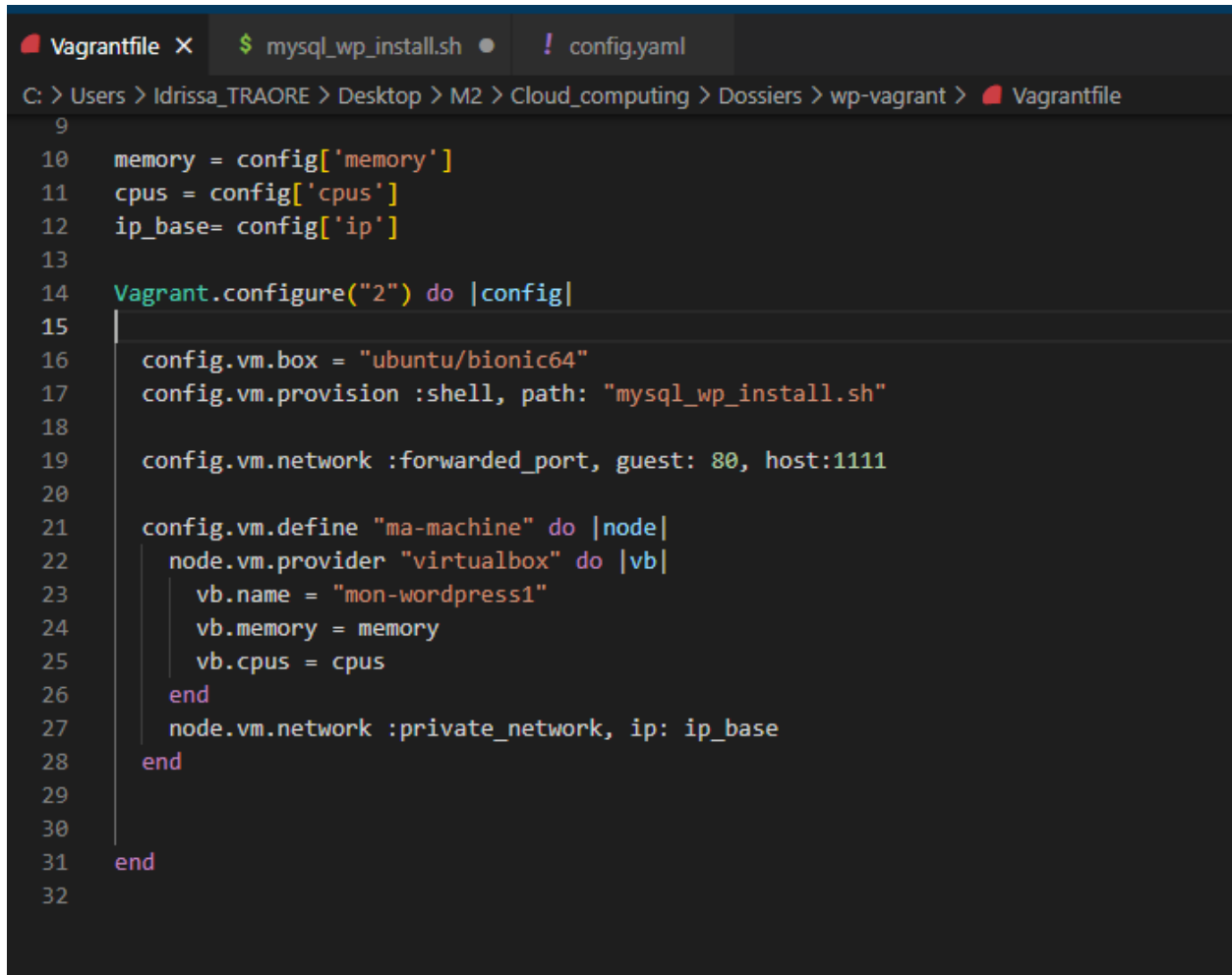


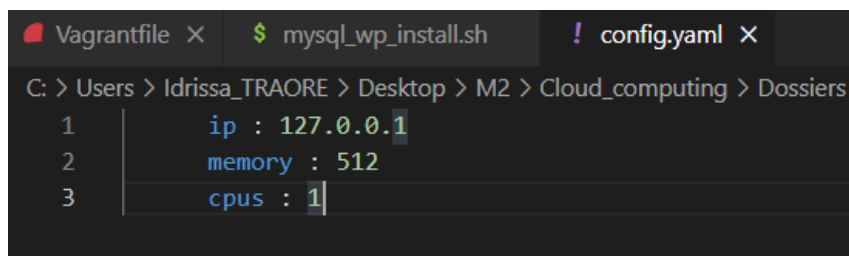
# Mini-projet Cloud computing 2022

## Partie 1 : Déploiement avec vagrant

Tout d'abord, j'ai créé mon fichier « Vagrantfile » à l'aide de la commande « vagrant init » puis « vagrant up » et j'ai renseigné les informations nécessaires du fichier « config. yaml » (ip, memory, cpus)



```
Vagrantfile X  $ mysql_wp_install.sh  ! config.yaml
C: > Users > Idrissa_TRAORE > Desktop > M2 > Cloud_computing > Dossiers > wp-vagrant > Vagrantfile
9
10  memory = config['memory']
11  cpus = config['cpus']
12  ip_base= config['ip']
13
14  Vagrant.configure("2") do |config|
15
16      config.vm.box = "ubuntu/bionic64"
17      config.vm.provision :shell, path: "mysql_wp_install.sh"
18
19      config.vm.network :forwarded_port, guest: 80, host:1111
20
21      config.vm.define "ma-machine" do |node|
22          node.vm.provider "virtualbox" do |vb|
23              vb.name = "mon-wordpress1"
24              vb.memory = memory
25              vb.cpus = cpus
26          end
27          node.vm.network :private_network, ip: ip_base
28      end
29
30
31  end
32
```



```
Vagrantfile X  $ mysql_wp_install.sh  ! config.yaml X
C: > Users > Idrissa_TRAORE > Desktop > M2 > Cloud_computing > Dossiers >
1      ip : 127.0.0.1
2      memory : 512
3      cpus : 1
```

Je démarre avec « Vagrant up »

```
C:\Users\Idrissa_TRAORE\Desktop\M2\Cloud_computing\Dossiers\wp-vagrant
λ vagrant up
Bringing machine 'ma-machine' up with 'virtualbox' provider...
==> ma-machine: You assigned a static IP ending in ".1" to this machine.
==> ma-machine: This is very often used by the router and can cause the
==> ma-machine: network to not work properly. If the network doesn't work
==> ma-machine: properly, try changing this IP.
==> ma-machine: You assigned a static IP ending in ".1" to this machine.
==> ma-machine: This is very often used by the router and can cause the
==> ma-machine: network to not work properly. If the network doesn't work
==> ma-machine: properly, try changing this IP.
==> ma-machine: Checking if box 'ubuntu/bionic64' version '20221117.0.0' is up to d
ate...
==> ma-machine: Setting the name of the VM: mon-wordpress1
==> ma-machine: Fixed port collision for 22 => 2222. Now on port 2201.
==> ma-machine: Clearing any previously set network interfaces...
==> ma-machine: Preparing network interfaces based on configuration...
    ma-machine: Adapter 1: nat
    ma-machine: Adapter 2: hostonly
==> ma-machine: Forwarding ports...
    ma-machine: 80 (guest) => 1111 (host) (adapter 1)
    ma-machine: 22 (guest) => 2201 (host) (adapter 1)
==> ma-machine: Running 'pre-boot' VM customizations...
==> ma-machine: Booting VM...
==> ma-machine: Waiting for machine to boot. This may take a few minutes...
    ma-machine: SSH address: 127.0.0.1:2201
    ma-machine: SSH username: vagrant
    ma-machine: SSH auth method: private key
```

```
ma-machine:
ma-machine: Guest Additions Version: 5.2.42
ma-machine: VirtualBox Version: 6.1
==> ma-machine: Configuring and enabling network interfaces...
==> ma-machine: Mounting shared folders...
ma-machine: /vagrant => C:/Users/Idrissa_TRAORE/Desktop/M2/Cloud_computing/Dossiers/wp-vagrant
==> ma-machine: Running provisioner: shell...
ma-machine: Running: C:/Users/IDRISS~1/AppData/Local/Temp/vagrant-shell20221123-9172-1q4vzb5.sh
ma-machine: Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
ma-machine: Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
ma-machine: Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
ma-machine: Get:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2470 kB]
ma-machine: Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
ma-machine: Get:6 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
ma-machine: Get:7 http://archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
ma-machine: Get:8 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [428 kB]
ma-machine: Get:9 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
ma-machine: Get:10 http://archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
ma-machine: Get:11 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2807 kB]
ma-machine: Get:12 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [983 kB]
ma-machine: Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [516 kB]
ma-machine: Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [1013 kB]
ma-machine: Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [140 kB]
ma-machine: Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1858 kB]
ma-machine: Get:17 http://archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [402 kB]
ma-machine: Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [24.9 kB]
ma-machine: Get:19 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [6012 B]
ma-machine: Get:20 http://archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [53.3 kB]
ma-machine: Get:21 http://archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [14.6 kB]
ma-machine: Get:22 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [18.1 kB]
ma-machine: Get:23 http://archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [8668 B]
ma-machine: Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [136 kB]
ma-machine: Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1245 kB]
ma-machine: Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [286 kB]
ma-machine: Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [19.0 kB]
ma-machine: Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [3836 B]
ma-machine: Fetched 26.5 MB in 16s (1644 kB/s)
ma-machine: Reading package lists...
```

```
C:\Users\Idrissa_TRAORE\Desktop\M2\Cloud_computing\Dossiers\wp-vagrant
λ |
```

Ce qui m'a donné la machine virtuelle « mon-wordpress1 »



Ensuite j'ai créé un fichier `mysql_wp_install.sh` dans lequel j'ai mis uniquement 2 commandes de lignes à exécuter et le reste en commentaires que j'exécuterai ligne par ligne après la connexion via la commande « `vagrant ssh` »

```
Vagrantfile  $ mysql_wp_install.sh  ! config.yaml
C: > Users > Idrissa_TRAORE > Desktop > M2 > Cloud_computing > Dossiers > wp-vagrant > $ mysql_wp_install.sh
1
2  echo "nameserver 8.8.8.8" | sudo tee /etc/resolv.conf > /dev/null # pour eviter que "sudo apt-get update" donne des erreurs
3
4  sudo apt-get update
5
6  # mysql --version
7  # sudo mysql_secure_installation # j'ai mis comme mot de passe "1234"
8
9  # sudo apt install wordpress
10
11 # sudo apt install php php-mysql
12 # php -v
13 # sudo sh -c 'echo "<?php phpinfo(); ?>" > /var/www/html/info.php' # pour creer la page d'information php
14 # vim /var/www/html/info.php # on met dans dans le fichier php :
15 # <?php
16 # phpinfo();
17 # ?>
18 # puis on enregistre
19
20 # On cree la base de donnees pour wordpress
21 # sudo mysql
22 # CREATE DATABASE wordpress_db;
23 # CREATE USER 'idrisa77'@'localhost' IDENTIFIED BY 'ABcd1234';
24 # GRANT ALL ON wordpress_db.* TO 'idrisa77'@'localhost' IDENTIFIED BY 'ABcd1234';
25 # FLUSH PRIVILEGES;
26 # Exit;
27 # cd /home/
28
29 # sudo wget https://wordpress.org/latest.tar.gz
30 # sudo tar -xzf latest.tar.gz
31 # sudo cp -R /home/wordpress /var/www/html/
32 # sudo chown -R www-data:www-data /var/www/html/wordpress/
33 # sudo chmod -R 755 /var/www/html/wordpress/
34 # sudo mkdir /var/www/html/wordpress/wp-content/uploads
35 # sudo chown -R www-data:www-data /var/www/html/wordpress/wp-content/uploads/
```

Je me suis connecte à la machine avec « `vagrant ssh` »

```
C:\Users\Idrissa_TRAORE\Desktop\M2\Cloud_computing\Dossiers\wp-vagrant
λ vagrant ssh
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-197-generic x86_64)Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-197-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Nov 24 10:00:37 UTC 2022
System load:  0.08      Processes:            84
Usage of /:   3.3% of 38.70GB   Users logged in:     0
Memory usage: 24%        IP address for enp0s3: 10.0.2.15
Swap usage:   0%          IP address for enp0s8: 127.0.0.1

0 updates can be applied immediately. 0 updates are pending.

vagrant@ubuntu-bionic:~$ ls
vagrant@ubuntu-bionic:~$
vagrant@ubuntu-bionic:~$ echo "nameserver 8.8.8.8" | sudo tee /etc/resolv.conf > /dev/null # pour eviter que "sudo apt-get update" donne des erreurs
vagrant@ubuntu-bionic:~$ sudo apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2808 kB]Get:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2470 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1858 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1245 kB]
Fetched 8641 kB in 8s (1070 kB/s)
Reading package lists... Done
vagrant@ubuntu-bionic:~$ |
```

## J'installe MySQL server

```
vagrant@ubuntu-bionic:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libbcbi-fast-perl libbcbi-pm-perl libencode-locale-perl libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7 mysql-client-core-5.7 mysql-common
  mysql-server-5.7 mysql-server-core-5.7
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinycsa
The following NEW packages will be installed:
  libaio1 libbcbi-fast-perl libbcbi-pm-perl libencode-locale-perl libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7 mysql-client-core-5.7 mysql-common
  mysql-server-5.7 mysql-server-core-5.7
0 upgraded, 21 newly installed, 0 to remove and 1 not upgraded.
Need to get 20.1 MB of archives.
After this operation, 157 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 mysql-common all 5.8+1.0.4 [7308 B]
Get:2 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libaio1 amd64 0.3.110-Subuntu0.1 [6476 B]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client-core-5.7 amd64 5.7.40-0ubuntu0.18.04.1 [6755 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client-5.7 amd64 5.7.40-0ubuntu0.18.04.1 [2028 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-server-core-5.7 amd64 5.7.40-0ubuntu0.18.04.1 [7542 kB]
53% [5 mysql-server-core-5.7 3652 kB/7542 kB 48%] 518 kB/s 14s
```

## Je vérifie la version MySQL installée

```
vagrant@ubuntu-bionic:~$ mysql --version
mysql Ver 14.14 Distrib 5.7.40, for Linux (x86_64) using Editline wrapper
vagrant@ubuntu-bionic:~$
```

Afin de mieux sécuriser l'installation de MySQL, notamment en définissant un mot de passe racine, on utilise « `mysql_secure_installation` ». Mon mot de passe entré est : 1234

```
vagrant@ubuntu-bionic:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?

Press y|Y for Yes, any other key for No: y
```

```
Estimated strength of the password: 25
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL 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? (Press y|Y for Yes, any other key for No) : 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? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL 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? (Press y|Y for Yes, any other key for No) : 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? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
```

On procède à l'installation de PHP avec « `sudo apt install php php-mysql` »

```
vagrant@ubuntu-bionic:~$ sudo apt install php php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php7.2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 libsodium23 php-common
  php7.2 php7.2-cli php7.2-common php7.2-json php7.2-mysql php7.2-opcache php7.2-readline ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom php-pear openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php7.2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 libsodium23 php
  php-common php-mysql php7.2 php7.2-cli php7.2-common php7.2-json php7.2-mysql php7.2-opcache php7.2-readline ssl-cert
0 upgraded, 22 newly installed, 0 to remove and 1 not upgraded.
Need to get 5870 kB of archives.
After this operation, 25.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 libapr1 amd64 1.6.3-2 [90.9 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 libaprutil1 amd64 1.6.1-2 [84.4 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-2 [10.6 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 libaprutil1-ldap amd64 1.6.1-2 [8764 B]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 liblua5.2-0 amd64 5.2.4-1.1build1 [108 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 apache2-bin amd64 2.4.29-1ubuntu4.25 [1072 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 apache2-utils amd64 2.4.29-1ubuntu4.25 [83.8 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 apache2-data all 2.4.29-1ubuntu4.25 [160 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 apache2 amd64 2.4.29-1ubuntu4.25 [95.1 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic/main amd64 php-common all 1:60ubuntu1 [12.1 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-common amd64 7.2.24-0ubuntu0.18.04.15 [891 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-json amd64 7.2.24-0ubuntu0.18.04.15 [18.9 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-opcache amd64 7.2.24-0ubuntu0.18.04.15 [165 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-readline amd64 7.2.24-0ubuntu0.18.04.15 [12.2 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic/main amd64 libsodium23 amd64 1.0.16-2 [143 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-cli amd64 7.2.24-0ubuntu0.18.04.15 [1412 kB]
54% [16 php7.2-cli 0 B/1412 kB 0%]
```

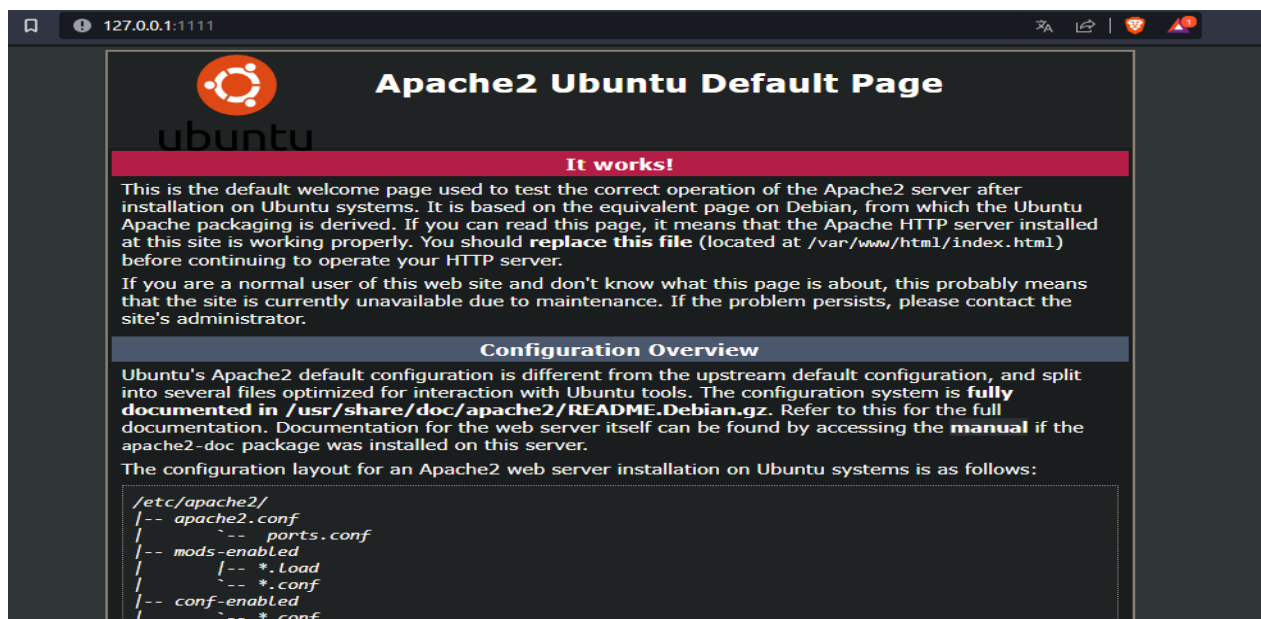
Puis nous vérifions la version installée via la commande « `php -v` »

```
vagrant@ubuntu-bionic:~$ php -v
PHP 7.2.24-0ubuntu0.18.04.15 (cli) (built: Nov 2 2022 09:09:52) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.2.24-0ubuntu0.18.04.15, Copyright (c) 1999-2018, by Zend Technologies
vagrant@ubuntu-bionic:~$ |
```

On crée une page d'informations PHP rapide pour l'afficher dans un navigateur

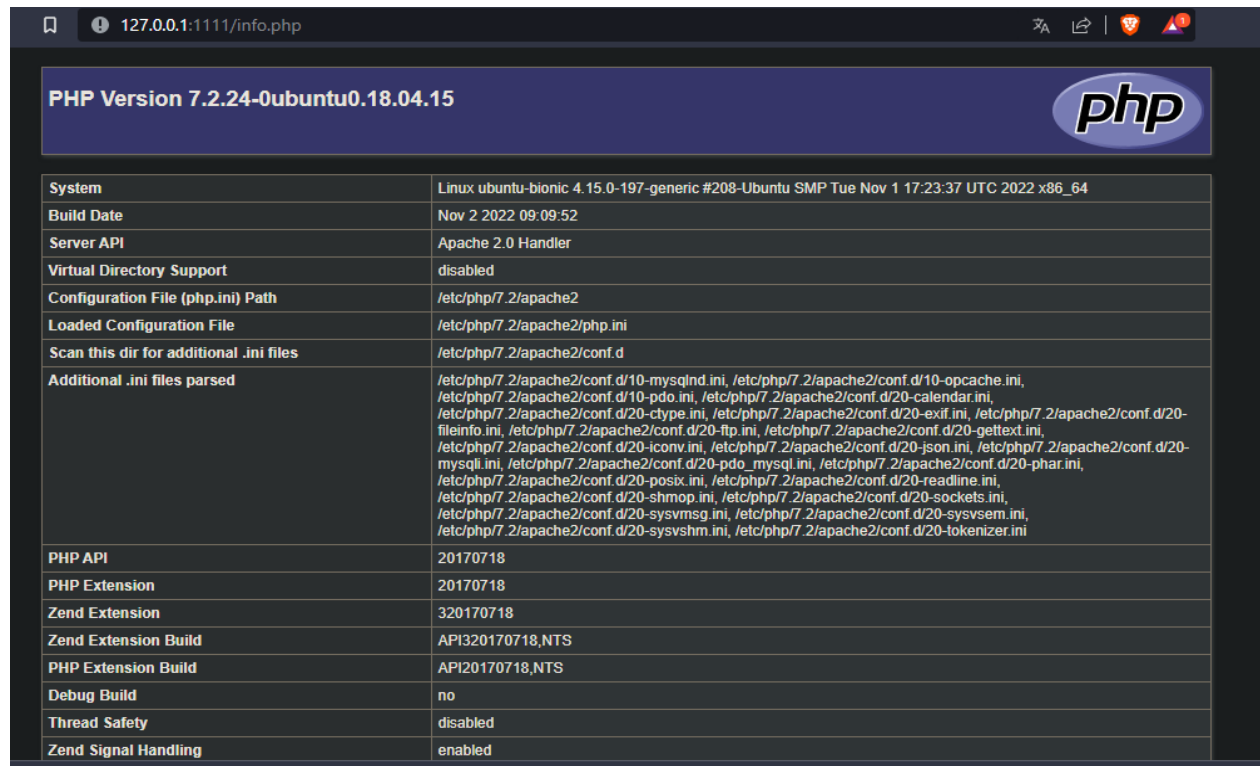
```
vagrant@ubuntu-bionic:~$ sudo sh -c 'echo "<?php phpinfo(); ?>" > /var/www/html/info.php'
```

On peut voir sur dans le navigateur les page apache en tapant <http://127.0.0.1:1111>





Et la page php en saisissant <http://127.0.0.1:1111/info.php>



PHP Version 7.2.24-0ubuntu0.18.04.15	
System	Linux ubuntu-bionic 4.15.0-197-generic #208-Ubuntu SMP Tue Nov 1 17:23:37 UTC 2022 x86_64
Build Date	Nov 2 2022 09:09:52
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled

Nous créons la base de données que nous utiliserons pour wordpress en servant des commandes suivantes : `sudo mysql`

```
CREATE DATABASE wordpress_db;
```

```
CREATE USER 'idrisa77'@'localhost' IDENTIFIED BY 'ABcd1234';
```

```
GRANT ALL ON wordpress_db.* TO 'idrisa77'@'localhost' IDENTIFIED BY 'ABcd1234'; # pour accorder des privileges a l'utilisateur
```

```
FLUSH PRIVILEGES;
```

```
Exit;
```

On se place de dossier bureau avec 'cd /home/' puis télécharge WordPress avec la commande

'sudo wget <https://wordpress.org/latest.tar.gz>' et on l'extrait avec 'sudo tar -xzf latest.tar.gz' ( vu sur le site <https://wordpress.org/support/article/how-to-install-wordpress/>)

```
wordpress/wp-admin/js/accordion.min.js
wordpress/wp-admin/js/inline-edit-post.min.js
wordpress/wp-admin/js/customize-widgets.min.js
wordpress/wp-admin/js/inline-edit-post.js
wordpress/wp-admin/js/updates.js
wordpress/wp-admin/js/media-upload.js
wordpress/wp-admin/js/media.js
wordpress/wp-admin/js/editor-expand.min.js
wordpress/wp-admin/js/media-gallery.min.js
wordpress/wp-admin/js/common.min.js
wordpress/wp-admin/js/tags-box.min.js
wordpress/wp-admin/js/svg-painter.min.js
wordpress/wp-admin/js/custom-background.js
wordpress/wp-admin/js/color-picker.min.js
wordpress/wp-admin/js/auth-app.js
wordpress/wp-admin/js/code-editor.js
wordpress/wp-admin/js/common.js
wordpress/wp-admin/js/set-post-thumbnail.min.js
wordpress/wp-admin/js/postbox.min.js
wordpress/wp-admin/js/color-picker.js
wordpress/wp-admin/js/password-strength-meter.js
wordpress/wp-admin/js/customize-nav-menus.js
wordpress/wp-admin/js/editor-expand.js
wordpress/wp-admin/js/code-editor.min.js
wordpress/wp-admin/js/set-post-thumbnail.js
wordpress/wp-admin/options-permalink.php
wordpress/wp-admin/widgets.php
wordpress/wp-admin/setup-config.php
wordpress/wp-admin/install.php
wordpress/wp-admin/admin-header.php
wordpress/wp-admin/post-new.php
wordpress/wp-admin/themes.php
wordpress/wp-admin/options-reading.php
wordpress/wp-trackback.php
wordpress/wp-comments-post.php
vagrant@ubuntu-bionic:~$ |
```

```
vagrant@ubuntu-bionic:/home$ ls
latest.tar.gz  ubuntu  vagrant  wordpress
vagrant@ubuntu-bionic:/home$ |
```

On copie le dossier WordPress vers le chemin '/var/www/html/' au moyen de

'sudo cp -R /home/wordpress /var/www/html/' .

Puis nous exécutons la commande 'sudo chown -R www-data:www-data /var/www/html/wordpress/'

Ensuite, on modifie les autorisations de fichier du dossier WordPress avec

'sudo chmod -R 755 /var/www/html/wordpress/' .

Par la suite, on crée le répertoire 'uploads' par la commande

'sudo mkdir /var/www/html/wordpress/wp-content/uploads'

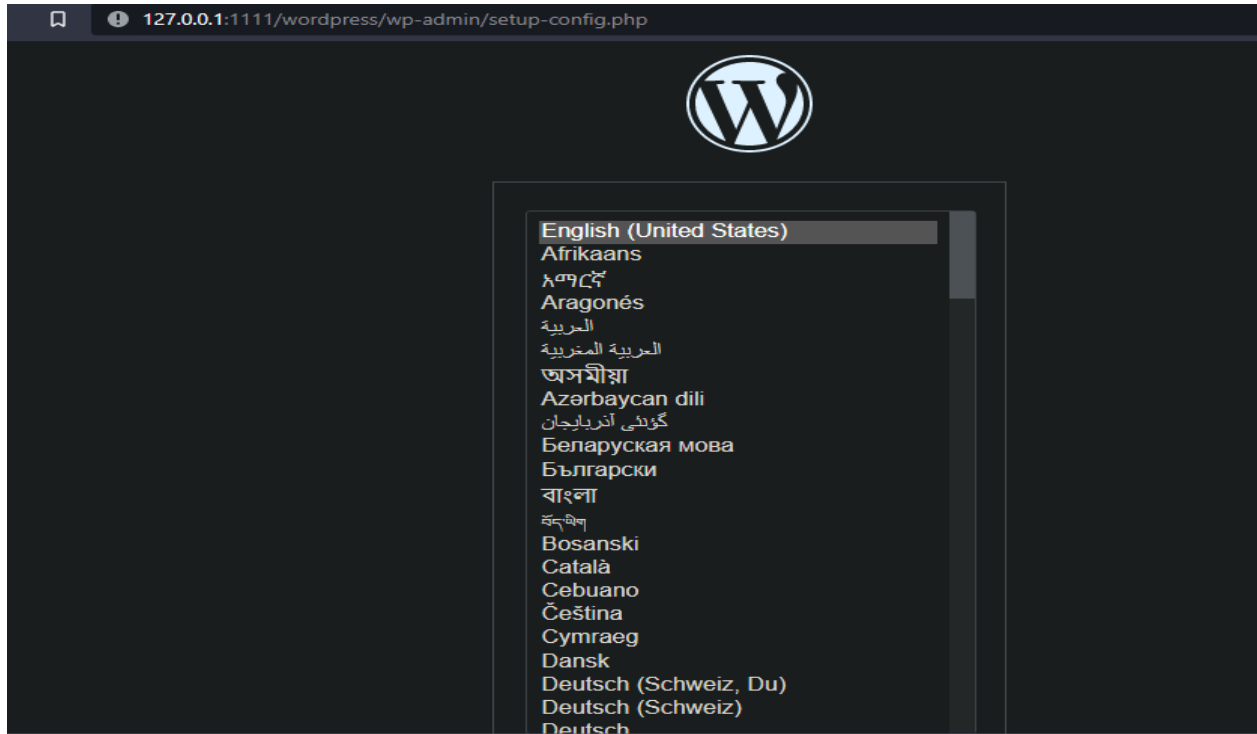
Enfin, on modifie les autorisations du répertoire 'uploads' avec




'`sudo chown -R www-data:www-data /var/www/html/wordpress/wp-content/uploads/`'

```
vagrant@ubuntu-bionic:/home$ sudo chown -R www-data:www-data /var/www/html/wordpress/
vagrant@ubuntu-bionic:/home$ sudo chmod -R 755 /var/www/html/wordpress/
vagrant@ubuntu-bionic:/home$ mkdir /var/www/html/wordpress/wp-content/uploads
mkdir: cannot create directory '/var/www/html/wordpress/wp-content/uploads': Permission denied
vagrant@ubuntu-bionic:/home$ sudo mkdir /var/www/html/wordpress/wp-content/uploads
vagrant@ubuntu-bionic:/home$ sudo chown -R www-data:www-data /var/www/html/wordpress/wp-content/uploads/
vagrant@ubuntu-bionic:/home$ |
```

On peut maintenant accéder à notre site WordPress en tapant : <http://127.0.0.1:1111/wordpress>



The screenshot shows the WordPress installation language selection screen. At the top, the URL bar displays `127.0.0.1:1111/wordpress/wp-admin/setup-config.php`. The WordPress logo is centered at the top. Below it, a list of languages is shown, with "English (United States)" selected and highlighted. Other visible languages include Afrikaans, Azərbaycan dili, Беларуская мова, Български, বাংলা, བོད་སྐད་, Bosanski, Català, Cebuano, Čeština, Cymraeg, Dansk, Deutsch (Schweiz, Du), Deutsch (Schweiz), and Deutsch.



The screenshot shows the WordPress installation welcome screen. The URL bar displays `127.0.0.1:1111/wordpress/wp-admin/setup-config.php?step=0`. The WordPress logo is centered at the top. Below it, the text reads: "Bienvenue sur WordPress. Avant de commencer, vous aurez besoin de connaître les éléments suivants." followed by a numbered list: 1. Nom de la base de données, 2. Identifiant MySQL, 3. Mot de passe de base de données, 4. Hôte de base de données, 5. Préfixe de table (si vous souhaitez avoir plusieurs WordPress sur une même base de données). Below the list, a paragraph explains that these information are used to create a `wp-config.php` file and provides instructions on how to proceed if automatic creation fails. At the bottom, there is a button labeled "C'est parti !".



Vous devez saisir ci-dessous les détails de connexion à votre base de données. Si vous ne les connaissez pas, contactez votre hébergeur.

<b>Nom de la base de données</b>	<input type="text" value="wordpress_db"/>	Le nom de la base de données avec laquelle vous souhaitez utiliser WordPress.
<b>Identifiant</b>	<input type="text" value="idrisa77"/>	Votre identifiant MySQL.
<b>Mot de passe</b>	<input type="text" value="ABcd1234"/>	Votre mot de passe de base de données.
<b>Adresse de la base de données</b>	<input type="text" value="localhost"/>	Si localhost ne fonctionne pas, demandez cette information à l'hébergeur de votre site.
<b>Préfixe des tables</b>	<input type="text" value="wp_"/>	Si vous souhaitez faire tourner plusieurs installations de WordPress sur une même base de données, modifiez ce réglage.

Envoyer



## Bienvenue

Bienvenue dans la très célèbre installation en 5 minutes de WordPress ! Vous n'avez qu'à remplir les informations demandées ci-dessous et vous serez prêt à utiliser la plus extensible et puissante plateforme de publication de contenu au monde.

## Informations nécessaires

Veillez renseigner les informations suivantes. Ne vous inquiétez pas, vous pourrez les modifier plus tard.

Titre du site

Titre1

Identifiant

Idrissa77

Les identifiants ne peuvent utiliser que des caractères alphanumériques, des espaces, des tirets bas ("\_"), des traits d'union ("-"), des points et le symbole @.

Mot de passe

ABcdef12345@

Medium

Hide

**Important :** Vous aurez besoin de ce mot de passe pour vous connecter. Pensez à le stocker dans un lieu sûr.

Votre e-mail

idrissatraore1960@gmail.com

Vérifiez bien cette adresse e-mail avant de continuer.

Visibilité par les moteurs de recherche

☒ Demander aux moteurs de recherche de ne pas indexer ce site

Certains moteurs de recherche peuvent décider de l'indexer malgré tout.

[Installer WordPress](#)



Identifiant ou adresse e-mail

Idrissa77

Mot de passe

••••••••••



☐ Se souvenir de moi

[Se connecter](#)

[Mot de passe oublié ?](#)

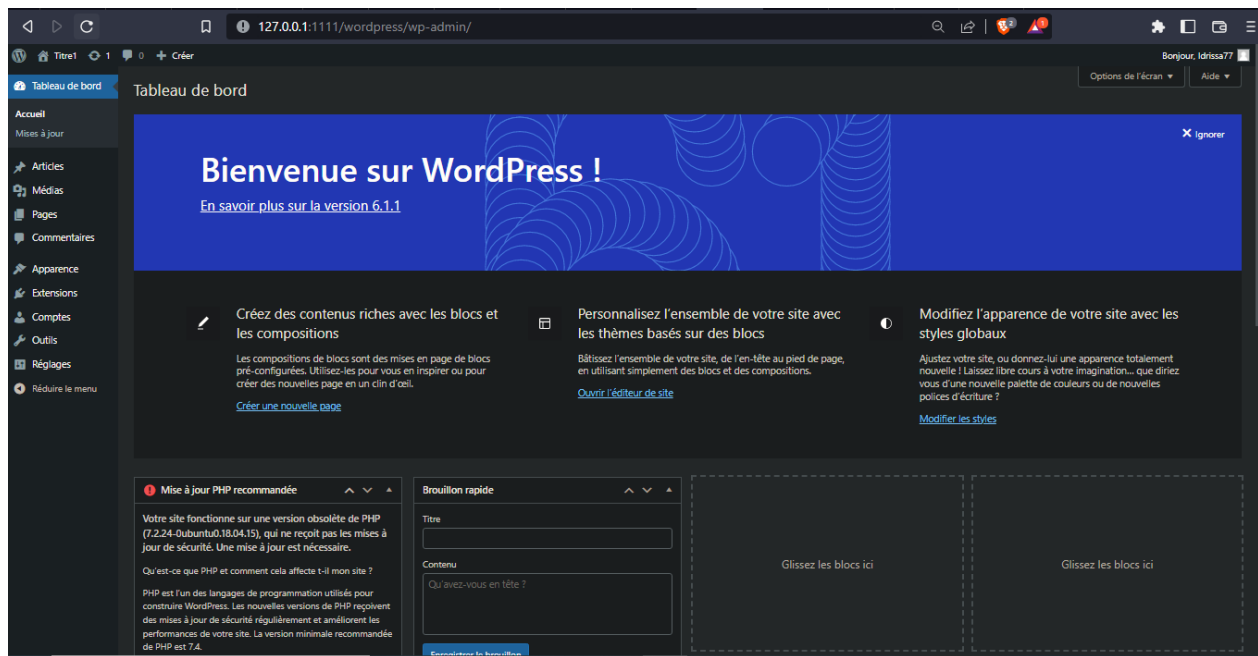
— [Aller sur Titre1](#)



Français



[Modifier](#)



## Partie 2 : Déploiement avec docker

J'ai procédé à la création d'un nouveau fichier 'Vagrantfile'

```
Vagrant.configure("2") do |config|

  config.vm.box = "ubuntu/bionic64"
  config.vm.provision :shell, path: "install-apache.sh"

  config.vm.network :forwarded_port, guest: 80, host: 8000

  config.vm.define "ma-machine3" do |node|
    node.vm.provider "virtualbox" do |vb|
      vb.name = "wp-docker3"
      vb.memory = 512
      vb.cpus = 1
    end
  end

end
```

```

λ vagrant up
Bringing machine 'ma-machine3' up with 'virtualbox' provider...
==> ma-machine3: Importing base box 'ubuntu/bionic64'...
==> ma-machine3: Matching MAC address for NAT networking...
==> ma-machine3: Checking if box 'ubuntu/bionic64' version '20221117.0.0' is up to date...
==> ma-machine3: Setting the name of the VM: wp-docker3
==> ma-machine3: Clearing any previously set network interfaces...
==> ma-machine3: Preparing network interfaces based on configuration...
    ma-machine3: Adapter 1: nat
==> ma-machine3: Forwarding ports...
    ma-machine3: 80 (guest) => 8000 (host) (adapter 1)
    ma-machine3: 22 (guest) => 2222 (host) (adapter 1)
==> ma-machine3: Running 'pre-boot' VM customizations...
==> ma-machine3: Booting VM...
==> ma-machine3: Waiting for machine to boot. This may take a few minutes...
    ma-machine3: SSH address: 127.0.0.1:2222
    ma-machine3: SSH username: vagrant
    ma-machine3: SSH auth method: private key

```

On se connecte à la machine

```

λ vagrant ssh
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-197-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Nov 25 01:34:15 UTC 2022

System load:  0.97               Processes:           90
Usage of /:   3.3% of 38.70GB    Users logged in:    0
Memory usage: 26%               IP address for enp0s3: 10.0.2.15
Swap usage:   0%

1 update can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

New release '20.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

```

On crée la base de données pour WordPress en utilisant 'touch init.sql' puis 'vim init.sql'

```

CREATE DATABASE IF NOT EXISTS wordpress;
CREATE USER IF NOT EXISTS 'wordpress'@'%' IDENTIFIED BY 'Password123';
GRANT ALL ON `wordpress`.* TO 'wordpress'@'%' IDENTIFIED BY 'Password123';
FLUSH PRIVILEGES;

```

Maintenant on utilise 'touch docker-compose.yml'

```
version: '3.3'

services:
  app:
    depends_on:
      - db
    image: wordpress:latest
    volumes:
      - ./app_files:/var/www/html
    ports:
      - "80:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: Password123
  db:
    image: mysql:5.7
    command: --init-file /data/application/init.sql
    volumes:
      - ./db_data:/var/lib/mysql
      - ./init.sql:/data/application/init.sql
    restart: always
    environment:
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: Password123
      MYSQL_ROOT_PASSWORD: root_password

volumes:
  db_data:
  app_files:
```

```
~
- port xxxx --update
```

~~~~~

"docker-compose.yml" 31L, 685C

11,11



Nous tapons 'sudo docker-compose up -d'

```
vagrant@ubuntu-bionic:~$ ls
docker-compose.yml  init.sql
vagrant@ubuntu-bionic:~$ sudo docker-compose up -d
Creating network "vagrant_default" with the default driver
Creating volume "vagrant_db_data" with default driver
Creating volume "vagrant_app_files" with default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
9a0b9cd2dfe6: Pulling fs layer
c637408ee7df: Pulling fs layer
4c517093c276: Pulling fs layer
301cc7d68c2a: Waiting
17ca9bf9231a: Waiting
9ae101e5c786: Waiting
04baa409344e: Waiting
f0b6015bf853: Waiting
6005bb052ef8: Waiting
99f303d57050: Waiting
307a9a80c1df: Waiting
|
```

```

Creating volume "vagrant_app_files" with default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
9a0b9cd2dfe6: Pull complete
c637408ee7df: Pull complete
4c517093c276: Pull complete
301cc7d68c2a: Pull complete
17ca9bf9231a: Pull complete
9ae101e5c786: Pull complete
04baa409344e: Pull complete
f0b6015bf853: Pull complete
6005bb052ef8: Pull complete
99f303d57050: Pull complete
307a9a80c1df: Pull complete
Digest: sha256:0e3435e72c493aec752d8274379b1eac4d634f47a7781a7a92b8636fa1dc94c1
Status: Downloaded newer image for mysql:5.7
Pulling app (wordpress:latest)...
latest: Pulling from library/wordpress
a603fa5e3b41: Pull complete
c428f1a49423: Pull complete
156740b07ef8: Downloading [=====> ]
48.72MB/91.63MBDownload complete
25f85b498fd5: Download complete
9b233e420ac7: Download complete
fe42347c4ecf: Download complete
d14eb2ed1e17: Download complete
10.72MB/10.76MBDownload complete
d2c43c5efbc8: Download complete
10.19MB/10.2MBDownload complete
80692ae2d067: Download complete
05e465aaa99a: Download complete
5e1d260f5864: Downloading [=====> ]
1f1b92fc1af4: Downloading [=====> ]
2.902MB/11.41MBaiting
b8532900ff77: Waiting
34541aa160dd: Waiting
179671110bb5: Waiting
c75ef74dd73f: Waiting
df68d832c77c: Waiting

```

Il avait des erreurs comme quoi le port 80/tcp était déjà utilisé, du coup, j'ai utilisé les commandes suivantes

```
sudo service apache2 stop
```

```
sudo apt-get remove apache2
```

```
curl -fsSL https://unms.com/install > /tmp/unms_inst.sh && sudo bash /tmp/unms_inst.sh --http-port  
xxxx -update
```

J'ai changé de port et ai retapé *sudo docker-compose. Up.*

```
vagrant@ubuntu-bionic:~$ sudo docker-compose up -d
Starting vagrant_db_1 ... done
Recreating vagrant_app_1 ... done
vagrant@ubuntu-bionic:~$ ls
app_files db_data docker-compose.yml init.sql
vagrant@ubuntu-bionic:~$
```

Maintenant, le site marche sur <http://localhost:80>

