

Etape 1 - Installation VM avec Debian 12

Installer ISO Debian 12 : <https://www.debian.org/download.fr.html>

Création machine virtuelle:

- 30 Go
- 2Gb/s
- Génération 1

root	Estebane42
dev08	Estebane42

Décocher Gnome et environnement de bureau Debian

Cocher serveur web et serveur ssh

Installer le programme de démarrage GRUB

Etape 2 - Configurer le ssh

Il nous faut vérifier l'adresse IP de la machine

ip a

```
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc no
    link/loopback 00:00:00:00:00:00 brd 00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    link/ether 00:15:5d:c8:35:14 brd ff:ff:ff:ff:ff:ff
    inet 192.168.80.83/24 brd 192.168.80.255 scop
        valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fec8:3514/64 scope link
        valid_lft forever preferred_lft forever
```

cd /etc/ssh

nano sshd_config

Décommenter "Port 22"

Décommenter "PermitRootLogin ..."

Réécrire cette ligne "PermitRootLogin yes"

```
Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
```

CTRL + X pour sauvegarder les modifications

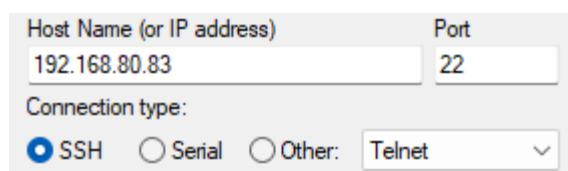
Entrer

systemctl restart ssh

Lancer Putty

Entrer l'adresse IP de la machine

Sélectionner SSH en port 22



(En cas d'erreur pour se connecter à putty, en dernier recours vérifier que la carte réseau est bien configuré)

Méthode :

décocher tout

Appliquer

cocher que IPv4 (laisser les autres se cocher automatiquement)

réessayer

Etape 3 - Installation dépendances

Mettre à jour la VM

```
apt update && apt upgrade
```

Installation paquets utilitaires (-y permet de passer la confirmation de téléchargement)

```
apt install -y wget curl zip unzip net-tools git
```

Installation php + extension

```
apt install -y php php-  
{dev,cli,common,intl,gd,xml,xsl,zip,mysql,pdo,apcu,opcache,curl,mbstring}
```

Explications des dépendances :

dev	log
cli	prompt
common	librairies communes
intl	traduction
gd	traitement image
xml, xsl	fichier balisé
pdo	gestionnaire de concession de base de donnée
apcu, opcache	gestionnaire de tache pour gagner performance
mbstring	traitement chaine de caractere

Vérifier la version de php

```
php -v
```

Installer json (format de fichier qui est une extension php qui permet d'interpréter les objets json)

```
apt install php-json
```

Vérifier que apache2 est bien installé

```
apt install apache2
```

Remarque : service php qui compresse le code : FPM et FCGI

Etape 4 - Installation des services pour le fonctionnement de la base de donnée

installation de maria db

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

on va sécuriser installation de maria db (à prendre en compte que l'on est dans un contexte de dev et non de prod)

Des questions s'affiche à l'écran

```
Enter current password for root (enter for none): █
```

ENTER

```
Switch to unix_socket authentication [Y/n] █
```

n

```
Change the root password? [Y/n] █
```

ENTER (remettre celui de root)

```
Remove anonymous users? [Y/n]
```

n

```
Disallow root login remotely? [Y/n]
```

n

```
Remove test database and access to it? [Y/n]
```

n

```
Reload privilege tables now? [Y/n]
```

ENTER

accès au prompt de maria db

```
mysql
```

création d'un utilisateur **symfony** dans Maria db

```
create user 'symfony' identified by 'Estebane42';
```

voir toute les bases de données

```
show databases;
```

Database
information_schema
mysql
performance_schema
sys

La base de donnée mysql contient toutes les infos utilisateur symfony, ce qui permet de vérifier qu'il est bien un mdp

```
select * from mysql.user;
```

on retrouve une colonne user, host et password qui nous intéresse ici

Host	User	Password
iv	Index_priv	Alter_priv
	Event_priv	Show_db_priv
	Trigger_priv	authenticatio
	String	

Accéder à ses colonnes

```
select user,host,password from mysql.user;
```

User	Host	Password
mariadb.sys	localhost	
root	localhost	*2D2724DF9D60D0175923C09CAAB46DAE74D3974B
mysql	localhost	invalid
symfony	%	*2D2724DF9D60D0175923C09CAAB46DAE74D3974B

Attention pour prod, host != % car correspond à connexion à toutes les machines sinon mettre @IP de la machine ou localhost si c'est la même machine

exit pour sortir de Maria DB

Etape 5 - Installation de composer

composer = permet de créer les projets symfony

wget <https://getcomposer.org/installer>

more installer (pour voir fichier)

php installer (pour télécharger le fichier)

on va modifier l'emplacement du fichier, pour tout le monde est accès au fichier

mv composer.phar /usr/bin/composer

On va vérifier le fichier (composer)

ls -lisa /usr/bin pipe | more

chercher composer

```
392465 2792 -rwxr-xr-x 1 root root 2855088 13 sept. 22:50 composer
```

rechercher sur google symfony.com pour télécharger le CLI symfony

wget <https://get.symfony.com/cli/installer> -0 - | bash

Plusieurs choix sont proposés

```
Use it as a local file:  
/root/.symfony5/bin/symfony  
  
Or add the following line to your shell configuration file:  
export PATH="$HOME/.symfony5/bin:$PATH"  
  
Or install it globally on your system:  
mv /root/.symfony5/bin/symfony /usr/local/bin/symfony
```

prendre "install it globally on your system"

```
mv /root/.symfony5/bin/symfony /usr/local/bin/symfony
```

Etape 6 - Créer projet symfony

ce déplacer dans le dossier pour créer le projet symfony

```
cd /var/www/html
```

il faut configurer le git (attention il faut la même adresse mail que le compte gitlab)

```
git config --global user.email 'estebanegardant0@gmail.com'  
git config --global user.name 'Estebane'
```

Créer le projet

```
symfony new Test
```

Créer clé SSH

```
ssh-keygen (tout faire entrer)
```

```
-----[RSA 3072]----+  
|     o*+          |  
|   E oo.=+        |  
|   +..Oo+o         |  
|   o oO.*oo=      o |  
|   oooo S=.= + o  |  
|   .   . +.* .    |  
|   ..o +          |  
|   .. + o          |  
|   o=.o          |  
-----[SHA256]----+
```

```
cat /root/.ssh/id_rsa.pub
```

```
-----BEGIN RSA PUBLIC KEY-----  
MIIBIjANBgkqhkiG9w0BAQEFAAQABMAsC1yc2EAAAQABAAgQDDda6dWg30tp241NVtsUc9vMvWw9Tf9qmd9H4aE57nHnlgiXa2MtjVPdV38ySdQM/7noz/+wnlwmlqQFh9nqMJeV8Sg+d+II2dNSFALAT  
DgGBS23HjszT00xzjG7j9lqNo6XpPgc9DVgDqnxscwT4u0xFksUsn59eLs2Dv3kEA3RjWa4eVoOuldu6ujS2vXtqR2/6TTDsswNbbUloSEL3jn9q3r/usk0hp83wXU4nRtj3  
N7VKe2UvccvO2FwSc6LG2yrkHyvc9mMieFZKx6cCj5eA4T/M+8mTzB/Zq8uLT5RkPr5qq62QagU1jZwrlwQMdEwi9/sklqFZBrsGEB5Lz24dcc8yONvs/h5p9upbsBaDjpfhg4GcInpT  
C93VoeqHttwIG9f7TW3/GZzult5JlBJ7Zl85KGXeSCPfLUfyWMg4az4VepuZutQmagOjmb94E/57o40JwJi8nDlow99+DuGoZc7VzFxCY5RW/4pmCrN9eDENfc= root@debian  
-----END RSA PUBLIC KEY-----
```

copier la clé

Sur Gitlabs

Profil\Préférence



Clés SSH

coller dans ajouter une clé SSH

Cle

```
ssh-rsa  
AAAAB3NzaC1yc2EAAAQABAAgQDDda6dWg30tp241NVtsUc9vMvWw9Tf9qmd9H4aE57nHnlgiXa2MtjVPdV38ySdQM/7noz/+wnlwmlqQFh9nqMJeV8Sg+d+II2dNSFALAT  
DgGBS23HjszT00xzjG7j9lqNo6XpPgc9DVgDqnxscwT4u0xFksUsn59eLs2Dv3kEA3RjWa4eVoOuldu6ujS2vXtqR2/6TTDsswNbbUloSEL3jn9q3r/usk0hp83wXU4nRtj3  
N7VKe2UvccvO2FwSc6LG2yrkHyvc9mMieFZKx6cCj5eA4T/M+8mTzB/Zq8uLT5RkPr5qq62QagU1jZwrlwQMdEwi9/sklqFZBrsGEB5Lz24dcc8yONvs/h5p9upbsBaDjpfhg4GcInpT  
C93VoeqHttwIG9f7TW3/GZzult5JlBJ7Zl85KGXeSCPfLUfyWMg4az4VepuZutQmagOjmb94E/57o40JwJi8nDlow99+DuGoZc7VzFxCY5RW/4pmCrN9eDENfc= root@debian
```

Donner un nom

Titre

root@debian

Le titre des clés est visible publiquement.

Ne pas mettre d'expiration

Date d'expiration

YYYY-MM-DD



Sur Gitlabs

Projets\Nouveau Projet

Créer un projet vide

Nom du projet : **Test**

Décocher la case READ ME

Créer le projet

Après la création du projet, différentes procédures s'affichent

Suivre "Pousser un dossier existant"

(bien faire la procédure pour chaque projet une seule fois)

Attention erreur dans la procédure (main ==> master)

vérifier que le projet symfony est bien dans le /var/www/html avec un ls

```
cd Test/
```

```
git init --initial-branch=main
```

```
git remote add origin git@gitlab.com:estebane/test.git
```

```
git add .
```

```
git commit -m "Initial commit"
```

```
git push --set-upstream origin master
```

```
yes
```

Sur Gitlabs

page accueil\projet

Vérifier que les fichiers se sont push

public	Initial commit	il y a 14 minutes
vendor	Initial commit	il y a 14 minutes
.env	Initial commit	il y a 14 minutes
composer.json	Initial commit	il y a 14 minutes
composer.lock	Initial commit	il y a 14 minutes
index.html	Initial commit	il y a 14 minutes
symfony.lock	Initial commit	il y a 14 minutes

Installer le paquet qu'on a besoin pour symfony
comme on est dans symfony on va utiliser compresser

```
composer req symfony/flex (= utiliser raccourcis commande)
```

ENTER

pour rendre compatible, installer apache

```
composer req symfony/apache-pack
```

ENTER

a

```
cd /etc/apache2
```

```
nano ports.conf
```

copier la ligne "Listen 80"

coller là à la ligne

```
Listen 80
Listen 80
```

Attention 1 à 1024 ports dédiés, 1024 à 65k c'est bon

Changer la 2eme ligne par 1025 par exemple

CTRL + X (laisser le port 80 par défaut aussi)

Créer une nouvelle config

```
cd sites-available/
```

```
cp 000-default.conf symfonyTest.conf
```

```
nano symfonyTest.conf (ex : symfonyTest)
```

1er ligne : <VirtualHost :80> remplacer par 1025 (choix entre 1025 à 65000)

```
<VirtualHost *:1025>
```

Ajouter /Test/public à Document Root (en fonction du nom du projet symfony créer au début et pas le nom du projet git)

Attention bien marqué public et non Public sinon erreur

```
DocumentRoot /var/www/html/Test/public
```

CTRL + X

Activer la conf

```
a2ensite symfonyTest.conf (permet d'activer le site qu'on vient de créer)
```

```
systemctl restart apache2
```

[http://\[@IP\]:\[port\]](http://[@IP]:[port])

ex : <http://172.27.44.129:1025>

ETAPE 6 - Installation de paquet requis pour Symfony

Installer les autres paquets

cd /var/www/html/Test/

composer req maker

(utilitaire qui permet de créer une entité)

ENTER

ENTER

Pour l'appli, installer paquet pour gerer l'intégralité de la base de donnée

composer req doctrine

ENTER

nano .env

(permet de modifier les paramètres pour ce projet symfony)

Modifier l'avant dernière ligne comme la capture d'écran juste en dessous en prendre en compte les paramètres en fonction du projet

DATABASE_URL='mysql://[UserMDB]:[password]@[IP du serveur MDB]:3306/[nom projet symfony]'

(attention @IP dépend si le serveur et la base de donnée sont installé sur le même serveur)

DATABASE_URL='mysql://symfony:Estebane42@127.0.0.1:3306/portefolioST'

Etape 7 - Configuration de la base de donnée

Donner des droits à un user dans symfony pour l'utilisateur symfony

mysql

grant all privileges on *.* to 'symfony';

flush privileges;

exit

créer base de donnée automatiquement dans symfony

php bin/console do:da:cr

Savoir les fichiers qui ne sont pas envoyé dans le git

git status

sel

Mettre à jour le git

ce qui as entre " " c'est ce qui s'affichera dans dernière vérification sur le gitlabs

git add .

git commit -m "CONF(Installation des dépendances)"

git push

git status

```
root@portefolioST:/var/www/html/portefolioST# git status
Sur la branche master
Votre branche est à jour avec 'origin/master'.
```