

ACCES EXTERIEUR MYSQL

1 – Généralités

Par défaut, et pour des raisons de sécurité évidentes, mysql (tout comme mariadb qui est pour rappel un [fork](#) de mysql) n'autorise pas l'accès extérieur à mysql. Ce que j'appelle un accès extérieur, est un accès autre qu'en localhost (127.0.0.1). Il se peut que vous ayez besoin d'échanger avec mysql, dans le cadre d'un développement par exemple. Une autre raison pourrait être que (et c'est souvent le cas) votre serveur mysql ne soit pas sur la même machine (physique ou virtuelle) que votre serveur web (apache, nginx, etc ...)

Vous devez donc autoriser votre serveur web ou votre MySQL Workbench à accéder à mysql.

Si vous n'avez pas accès à votre "machine mysql" et que celle-ci est hébergée sous Linux, vous devez utiliser le protocole SSH que nous avons vu précédemment.

Pour rappel mysql utilise par défaut le port **3306**.

Dans le cas où la "box" de notre vagrant n'est pas buggée comme celle de notre vm-2017 vous pouvez utiliser la commande "vagrant ssh" pour accéder à "l'intérieur" de notre box. Dans le cas où vous souhaitez accéder à un Linux "vrai" (et non une machine virtuelle) ou pour une box buggée nous utiliserons Putty.

ARE YOU READY ? Ca va être tendu, on est parti !!!

2 – Les infos nécessaires

Il va nous falloir connaître plusieurs choses pour pouvoir accéder à notre OS.

1. L'identifiant :

Tout comme pour mysql, tout Linux possède un compte "root" qui est super administrateur du système. La majorité des linux, par défaut n'autorisent pas un accès ssh direct avec cet identifiant. Pire encore, ubuntu par exemple ne "connait" pas ce root et utilise la commande "[sudo](#)" pour "émuler" ce fameux root ! Dans le cadre d'un Linux "physique" il y'a peut de chances que vous connaissiez ce mot de passe root il faudra donc vous adresser à l'administrateur réseau pour vous permettre d'accéder à mysql, c'est tout bon pour vous car c'est lui qui bidouillera à votre place !!!!

Dans le cas où vous travaillez sur une machine virtuelle comme notre vm-2017 ou que l'admin réseau c'est vous ... ben va falloir mettre les mains dans la console ! Si vous êtes l'admin réseau, vous connaissez le mot de passe root, vu que vous avez installé l'os, dans le cas d'ubuntu ce sera le compte que vous avez créé. Sinon, lorsque vous aurez lancé votre machine virtuelle vous aurez l'info.

CA Invite de commandes

```
D:\Vms\vm-dl2017>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'ubuntu/xenial64' is up to date...
==> default: A newer version of the box 'ubuntu/xenial64' is available! You currently
==> default: have version '20170331.0.0'. The latest is version '20170410.0.0'. Run
==> default: `vagrant box update` to update.
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
default: Adapter 1: nat
default: Adapter 2: hostonly
==> default: Forwarding ports...
default: 80 (guest) => 4567 (host) (adapter 1)
default: 3306 (guest) => 3306 (host) (adapter 1)
default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
default: SSH address: 127.0.0.1:2222
default: SSH username: ubuntu
default: SSH auth method: password
default: Warning: Remote connection disconnect. Retrying...
==> default: Machine booted and ready!
[default] GuestAdditions 5.1.18 running --- OK.
==> default: Checking for guest additions in VM...
==> default: Configuring and enabling network interfaces...
==> default: Mounting shared folders...
default: /vagrant => D:/Vms/vm-dl2017
default: /var/www/html => D:/Vms/vm-dl2017/public
==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> default: flag to force provisioning. Provisioners marked to run always will still run.

D:\Vms\vm-dl2017>
```

Nous avons maintenant l'**identifiant** ! Regardez plus haut le port 3306 qui est bindé coté **host** et **guest** ... vous devrez en cas de différence entre le host et le guest, toujours dans le notre cas, utiliser le port **host**, ici c'est le même ! Vous trouverez aussi le port SSH qui ici est bindé en 2222 (celui que vous utiliserez dans votre putty) pour le host vers un 22 pour le guest

2. Le fichier de configuration

Selon la distribution, le fichier de configuration ne sera pas le même

Par exemple sous

- **Debian** le chemin vers le fichier de configuration principal est
"/etc/mysql/my.cnf"
- **Ubuntu** (qui pourtant est construit à partir de Debian le chemin est "/etc/mysql/mysql.conf.d/mysqld.cnf" et c'est celui qui va nous intéresser

Alors je vous entend déjà pourquoi le chemin n'est pas le même, pourquoi le nom du fichier n'est pas le même ???? Ben c'est une décision des dev que vous serez bientôt !

Comment trouver alors le bon fichier ? **LA DOC, LA DOC, LA DOC !!**

Je vais tout de même vous donner une astuce pour trouvez l'emplacement de ces fichier, une chtite commande linux :

find / -name my.cnf OU find / -name mysqld.cnf

Cette commande vous permettra de trouver l'emplacement du fichier. ATTENTION : vous devez être en **root** ou mettre **sudo** avant la commande, sous ubuntu.

Je ne rentrerai pas plus loin dans les détails, ni ne vous expliquerai la démarche pour d'autres distributions Linux ...

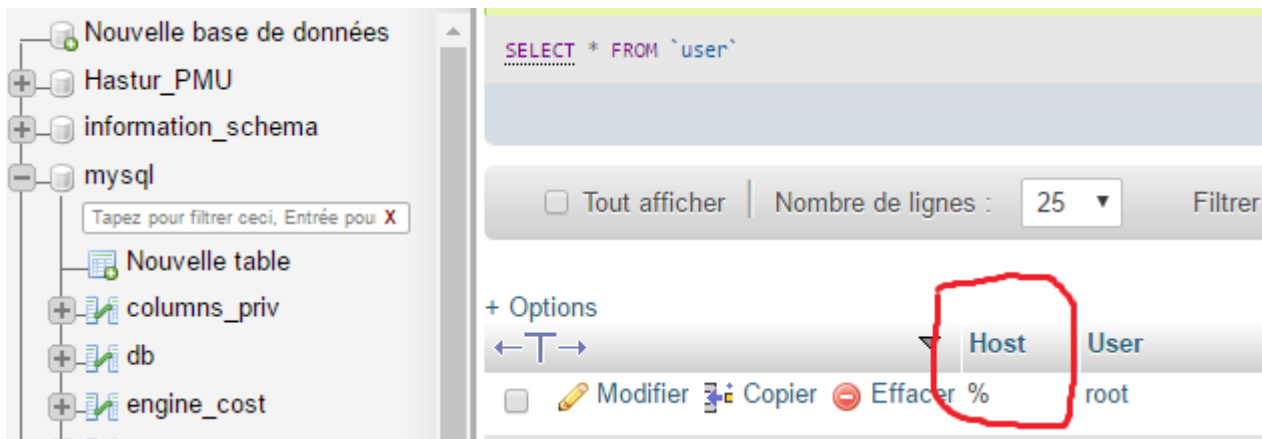
Les 2 parametres qui nous intéressent sont :

- skip-external-locking
- bind-address

3 – Rappel phpmyadmin

Nous partons du principe que phpmyadmin est installé et que vous avez accès au compte root mysql. Dans le cas contraire, il est possible de faire la même procédure en ligne de commande, mais je ne vous en parlerai pas, Thomas le fera en temps et en heure !

Dans phpmyadmin => base "mysql" => table "user", modifier le champ "host" et mettre "%" pour dire en gros "toutes les ip"



Il est possible de faire en sorte de spécifier une ou plusieurs adresses IP mais je ne rentrerai pas dans les détails, pour plus d'infos consulter Thomas ou moi !

ATTENTION : N'oubliez pas de recharger les privilèges, "flush privileges" ça vous rappelle quelque chose ? !!!!!!!

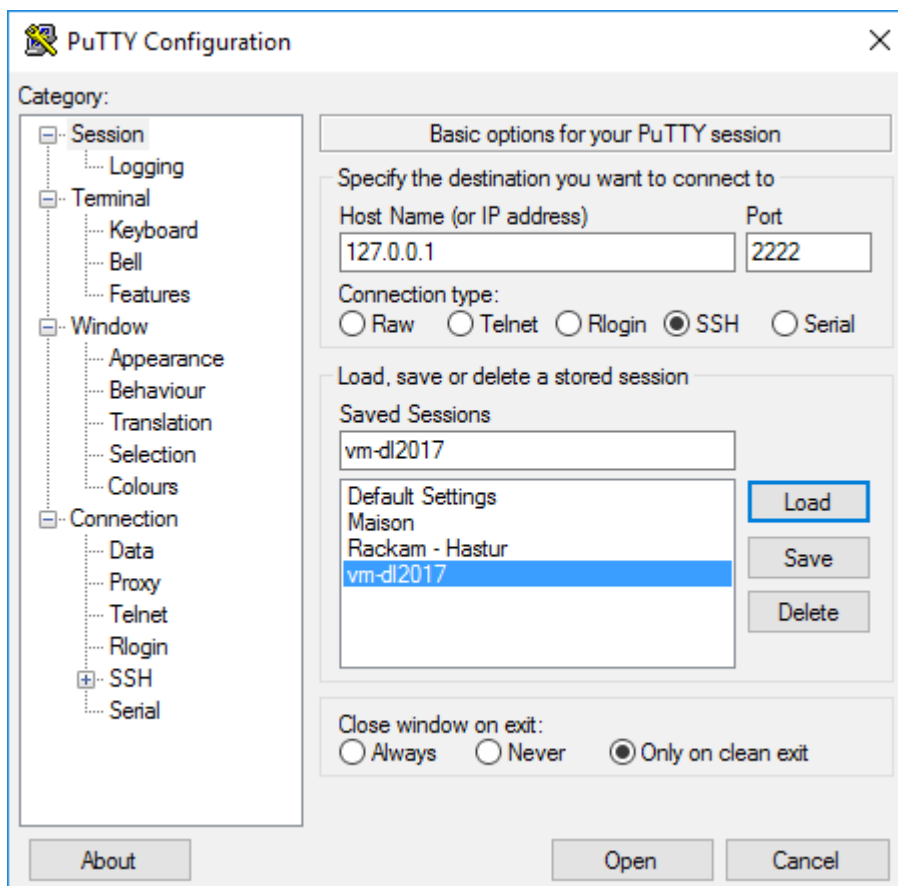
4 – Les modifications

Nous partons du principe que vous avez déjà votre clé privée Putty et qu'elle est compatible avec Putty (.ppk).

Préparons notre putty pour la connection en SSH à la VM.

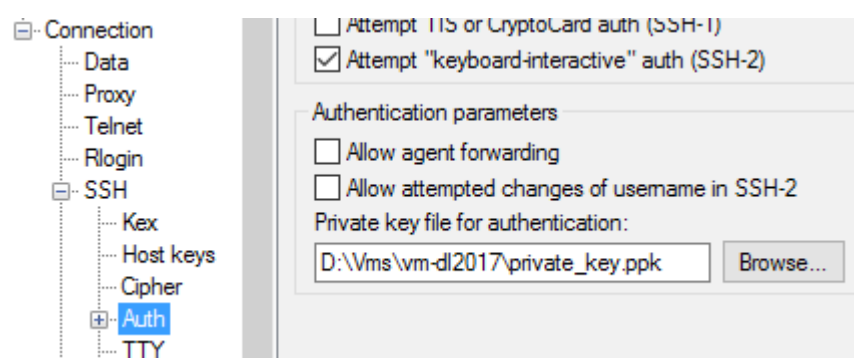
Nous utiliserons :

- Host Name : 127.0.0.1 (ou localhost)
- Port : 2202 (voir plus haut le bind 2202 => 22)



Nous allons renseigner la clé privé vu dans la 1^{er} partie.

Dans le menu de gauche => connection => ssh => auth => Private key file
=> Browse

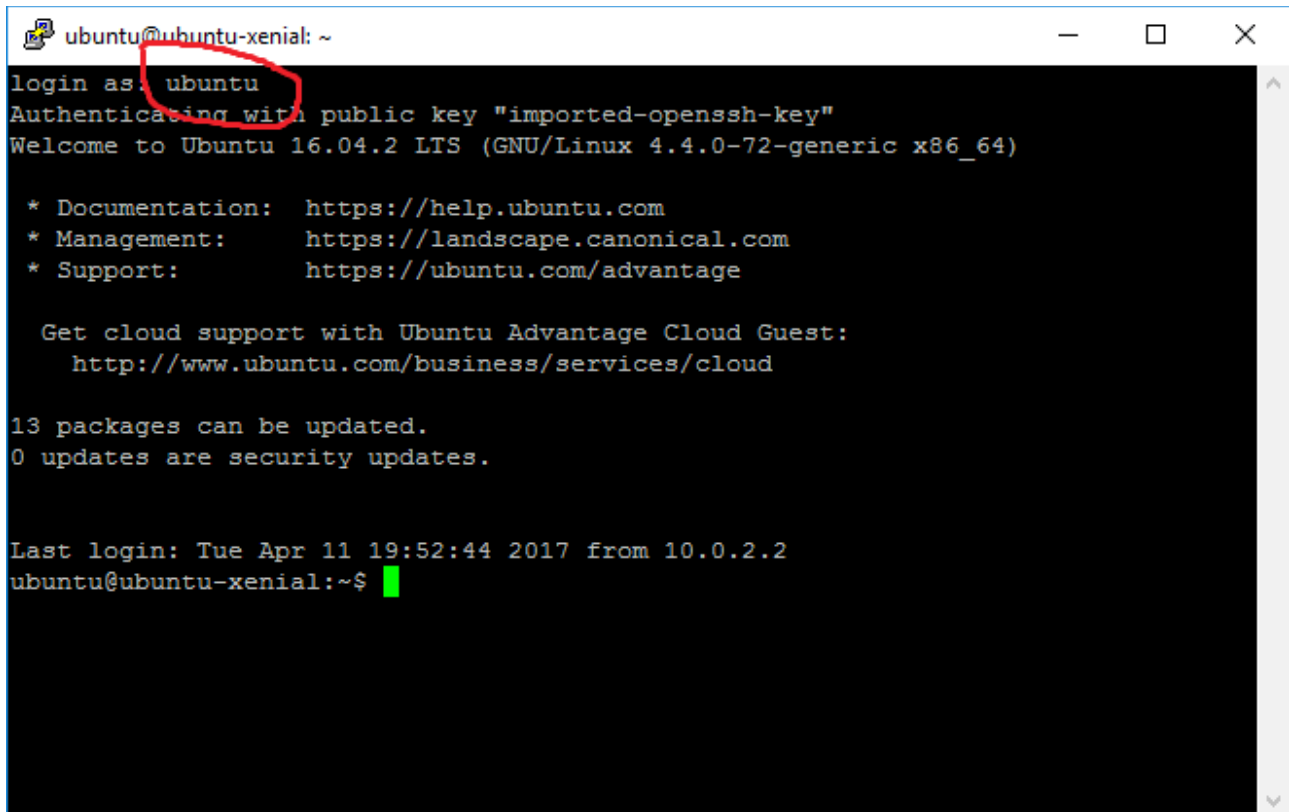


Retournez dans "session" (tout en haut du menu de gauche), donnez un

nom dans le champ "Saved Sessions" et cliquez sur "Save"

Voilà nous sommes prêts à établir notre connexion SSH. Cliquez sur "Open", si une autorisation vous est demandée, acceptez-la !

Saisissez l'identifiant que nous avons trouvé plus haut, pour nous "ubuntu" et appuyez sur enter

A terminal window titled 'ubuntu@ubuntu-xenial: ~' with standard window controls. The terminal output shows an SSH login process. The prompt 'login as:' is followed by 'ubuntu', which is circled in red. The next lines are 'Authenticating with public key "imported-openssh-key"' and 'Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-72-generic x86_64)'. This is followed by a list of links for documentation, management, and support. Then, it says 'Get cloud support with Ubuntu Advantage Cloud Guest:' with a URL. Below that, it reports '13 packages can be updated.' and '0 updates are security updates.' The last line shows the last login time and the current prompt 'ubuntu@ubuntu-xenial:~\$' with a green cursor.

```
ubuntu@ubuntu-xenial: ~
login as: ubuntu
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-72-generic x86_64)

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

Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud

13 packages can be updated.
0 updates are security updates.

Last login: Tue Apr 11 19:52:44 2017 from 10.0.2.2
ubuntu@ubuntu-xenial:~$
```

Nous sommes connecté en ssh sur notre machine, si vous êtes là, toujours avec moi , bravo, le plus difficile est fait !!!

Nous allons éditer ce fameux fichier de config qui nous fait tellement c...., euhhh pardon qui nous pose problème :)

Pour cela nous allons utiliser un éditeur en mode texte qui s'appelle **NANO**

sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

Sous ubuntu **sudo** est obligatoire pour "émuler" le root. Il se peut dans certains

cas qu'un mot de passe vous soit demandé lors d'une commande précédée par nano. Le mot de passe est celui de l'utilisateur de "base" mais ici nous n'auront pas ce soucis.

Pour info : sous nano, toutes les commandes que vous voyez en bas de l'éditeur sont à utiliser avec CTRL + <la lettre>, par exemple **CTRL + O** pour écrire après modif et **CTRL + X** pour quitter lisez ce que l'éditeur vous demande ! Pour parcourir le fichier, fleche HAUT et BAS.

Rapelez vous les 2 parametres que nous devons modifier, **skip-external-locking** et **bind-address**

```
skip-external-locking
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 127.0.0.1
```

Il faut COMMENTER ces 2 lignes grace au #

```
lc-messages-dir = /usr/share/mysql
#skip-external-locking
#
# Instead of skip-networking the default is
# localhost which is more compatible and is
#bind-address            = 127.0.0.1
#
# * Fine Tuning
#
```

Nous sauvegardons notre fichier avec **CTRL + O** , enter, puis **CTRL + X** pour quitter.

Si ca ne fonctionne pas et que vous voyez, en bas de votre éditeur **Error writing**

/etc/mysql/mysql.conf.d/mysqld.cnf: Permission denied c'est que vous n'avez pas édité avec SUDO !

Courage mes soeurs, courage mes freres, nous touchons au but !!!!!

Il nous faut maintenant redemarrer mysql, pour cela saisir la commandes

sudo service mysql restart

Et voila, votre mysql est redemarré !

***C'est fini, vous pouvez reprendre une activité normale
et taper une dernière commande, la meilleure : **exit*****

A vous maintenant de configurer MySQL Workbench !

Have a nice Day :)