

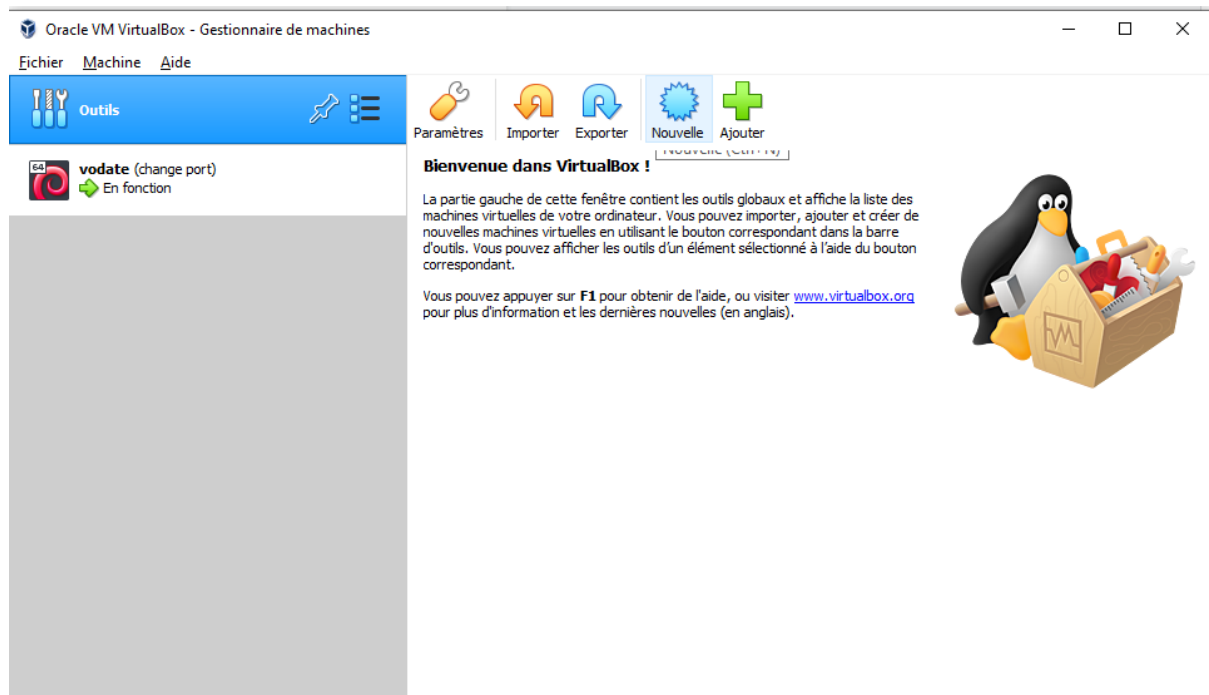
Installation de l'environnement Linux

Installation pour Windows :

Installez Virtualbox pour Windows -

<https://download.virtualbox.org/virtualbox/6.1.26/VirtualBox-6.1.26-145957-Win.exe>

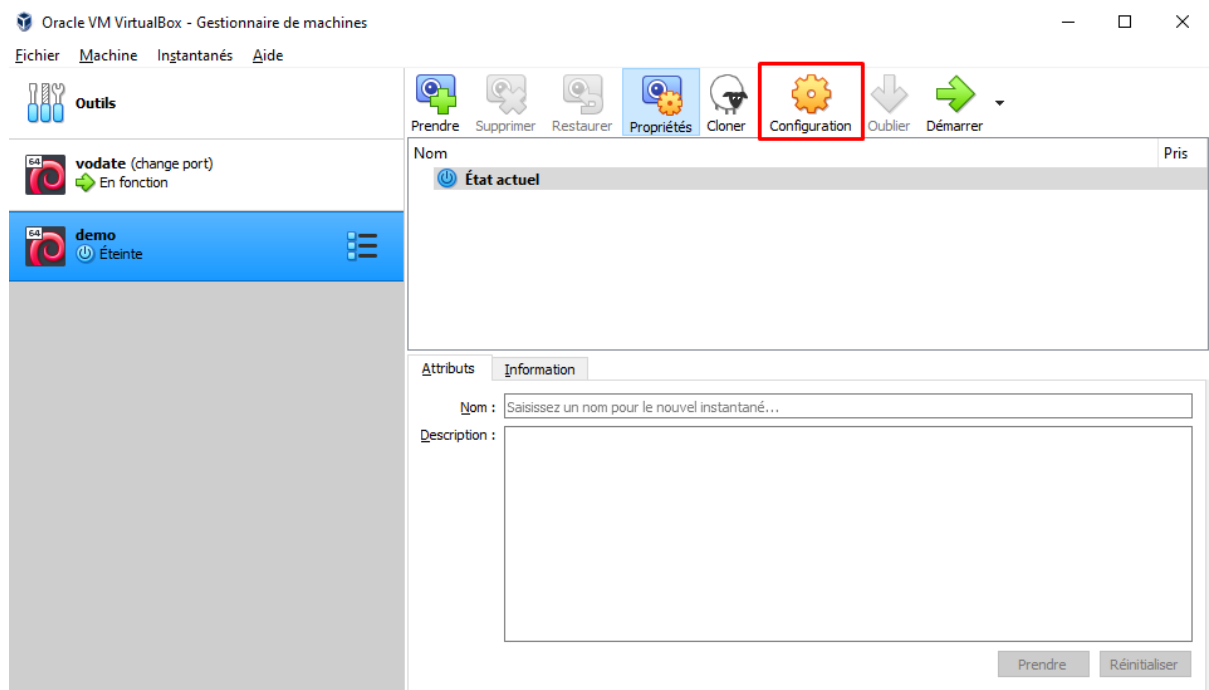
Une fois installé, faite nouvelle:



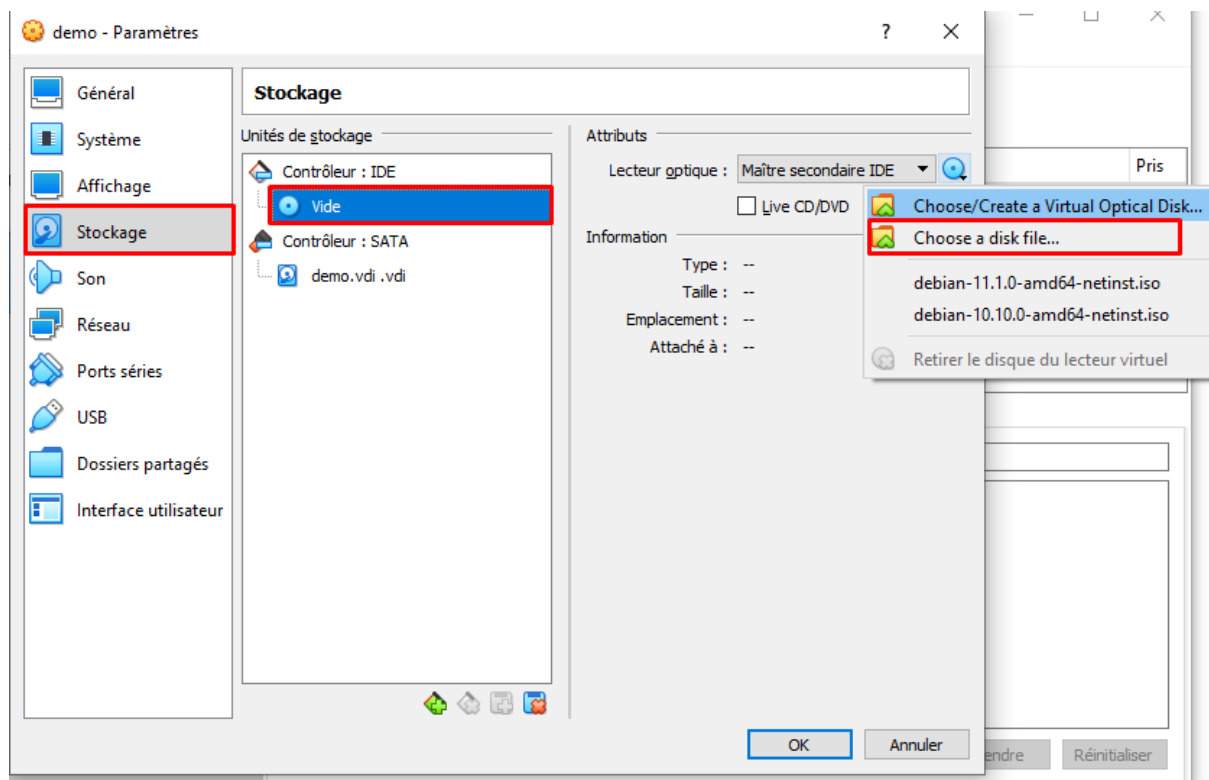
Donnez le nom que vous voulez, sélectionnez le dossier où vous voulez qu'il mette tous les fichiers pour la VM, dans le sélecteur "type", sélectionnez linux, et dans la version sélectionnez debian (64-bit). sélectionnez 1024 Go de RAM, créez un disque dur virtuel maintenant, VDI puis Dynamique alloué, enfin écrivez 10Go pour la taille du disque dur.

Téléchargez l'image de debian dans la dernière version
(<https://www.debian.org/index.fr.html>)

Cliquez sur configuration :



Dans les onglets à gauche sélectionner Stockage, cliquez sur vide et dans la partie droit cliquez sur le petit disque dans le menu qui apparaît sélectionnez “choose a disk file..” et sélectionner l’image de debian précédemment télécharger.



Une fois cela fait vous pouvez démarrer la VM:

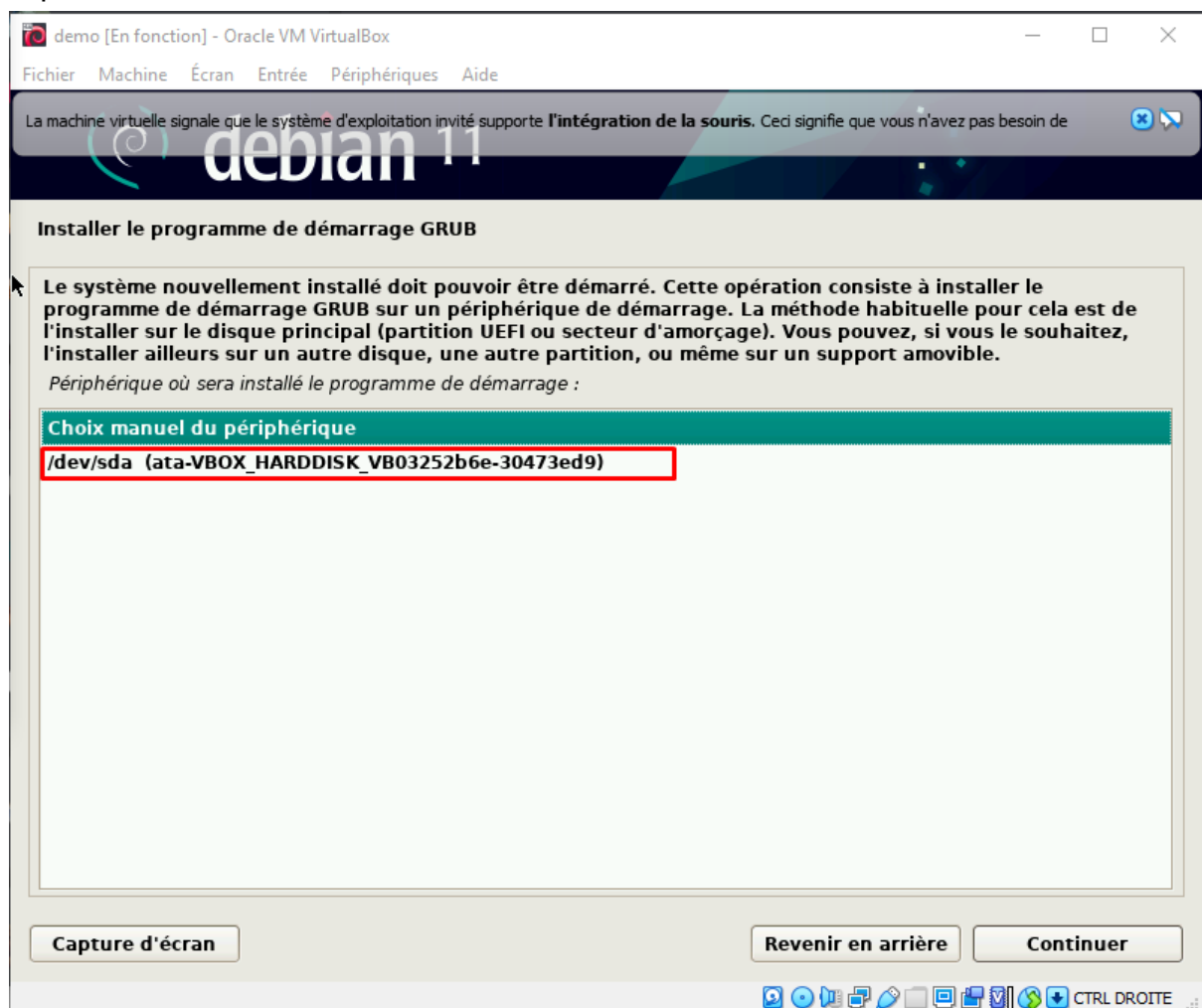


Une fois sur la page d'installation sélectionner Graphical install, sélectionner votre langue, le pays, la langue du votre clavier.

Le nom de la machine : debian
le domaine : *laisser vide*
Mot de passe super admin : root
Nom de l'utilisateur : user
Mot de passe de l'utilisateur : user

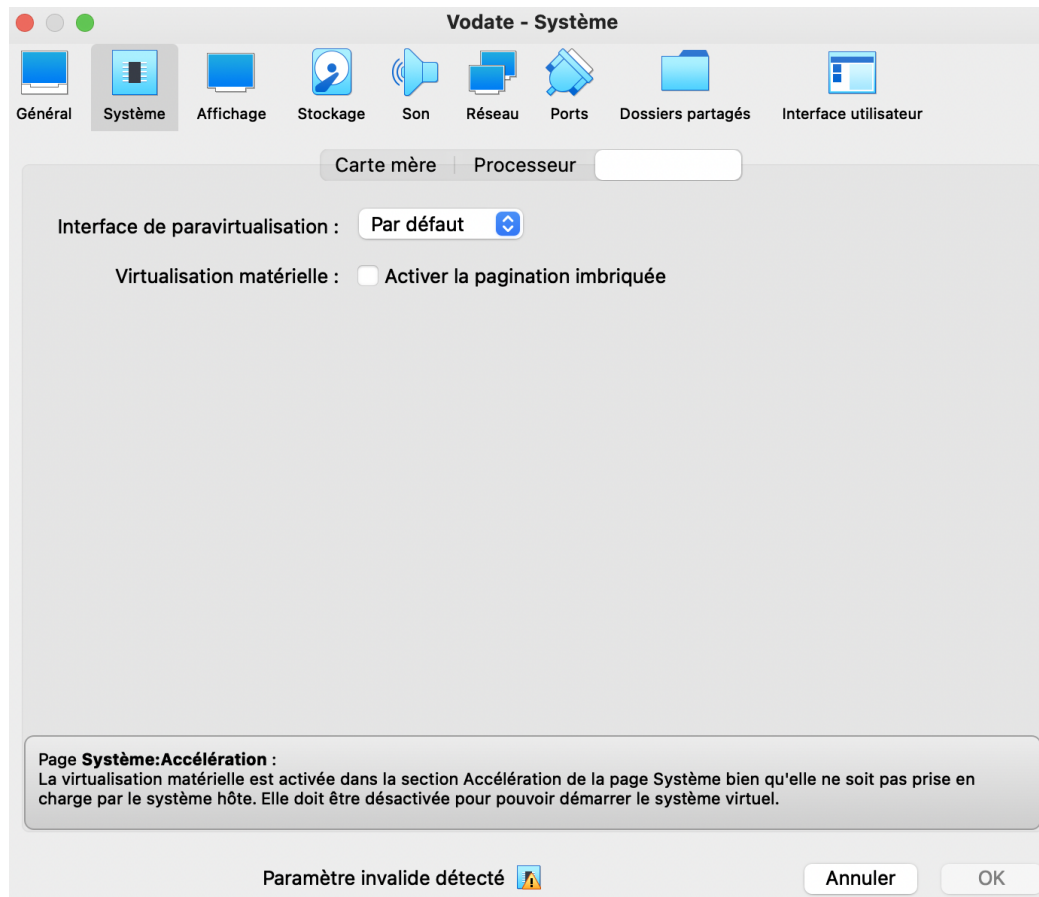
Puis fait continuer jusqu'à la page Sélection des logiciels. sur cette page sélectionner: "environnement de bureau Debian", "xfce" et "utilitaires usuels du système".

Une fois sur la page Installer le programme de démarrage GRUB, sélectionnez oui et puis sélectionner /dev/sda :



Installation pour MacOS (puce M1)

L'installation de Virtualbox n'étant pas parfaitement compatible (à date du 12/10/2021) avec les nouveaux Mac doté de la puce M1, nous avons cherché une autre solution sur internet et nous sommes tombés sur UTM un logiciel de virtualisation compatible avec les derniers Mac.



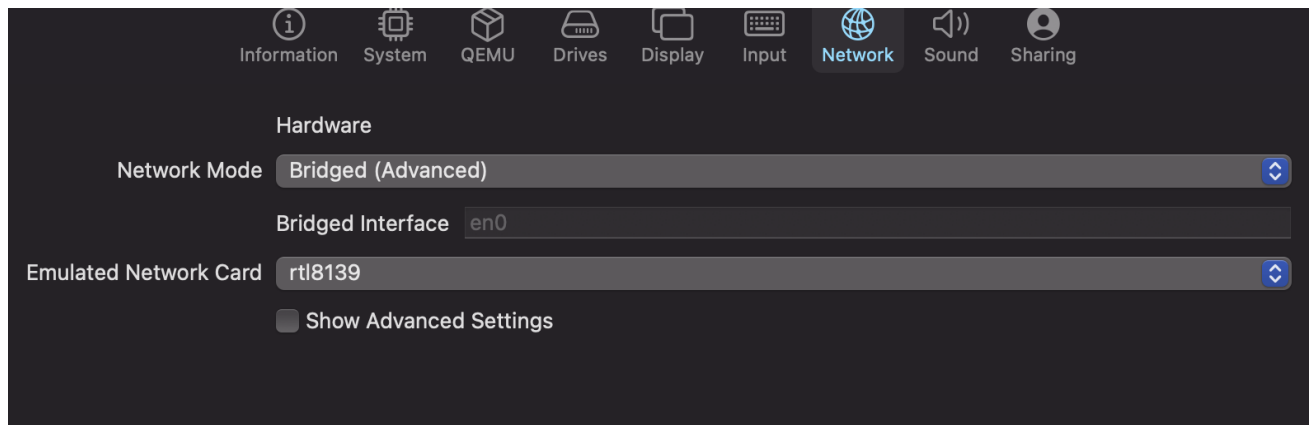
Installez l'outil UTM pour mac - <https://mac.getutm.app/>

Installez Debian 10.4 (Xfce) - <https://mac.getutm.app/gallery/debian-10-4-xfce>

Ouvrir le fichier UTM, se connecter à la session avec les identifiants suivants :

Username : debian, Password : debian.

Dans les paramètres de UTM, il faut aller dans la section Network.



Si l'option Network Mode n'apparaît pas sur UTM, pensez à vérifier que vous avez bien les dernières mises à jour système ainsi que les dernières mises à jour de UTM.

Choisir Bridged (Advanced) pour le Network Mode. Pour créer un pont directement entre la machine virtuelle ainsi que sur la machine qui accueille la machine virtuelle.

Ouvrez le terminal et tapez les lignes de commande:

Pour l'ouvrir depuis xfce, faites un clic droit sur le bureau > Applications > Émulateur de Terminal.

Cette interface est équivalente à l'invite de commandes présente sur Windows. Un terminal seul ne permet pas d'effectuer des actions. S'il est possible de taper des commandes, c'est parce que les terminaux intègrent un interpréteur de commandes ou "shell". Le shell par défaut est bash, le terminal est xterm.

Commandes système

- Afficher le répertoire en cours: `cd`
- Changer de répertoire: `cd {{ nom du répertoire }}`
- Revenir au répertoire précédent: `cd ../`
- Lister les fichiers présents dans un répertoire: `ls`
- Lister les fichiers présents dans un répertoire avec leurs droits associés, sous forme de liste et en incluant les fichiers cachés: `ls -la`

- Créer un fichier: `cat > {{ nom du fichier }}`
- Insérer du texte dans un fichier: `echo "{{texte}}" >> {{ nom du fichier }}`
- Supprimer un fichier: `rm {{ nom du fichier }}`
- Afficher le contenu d'un fichier: `cat {{ nom du fichier }}`
- Créer un répertoire: `mkdir {{ nom du répertoire }}`
- Créer un lien symbolique: `ln -s {{ chemin/vers/le/dossier/existant }} {{ /chemin/vers/le/lien/symbolique }}`
- Supprimer un répertoire: `rm -r {{ nom du répertoire }}`
- Copier un répertoire: `cp -r {{nom du répertoire}}`
- Renommer un répertoire: `mv {{ nom actuel du répertoire }} {{ nouveau nom du répertoire }}`
- Déplacer un répertoire: `mv {{ emplacement actuel du répertoire }} {{ futur emplacement du répertoire }}`
- Afficher le manuel de la commande "find": `find -help`
- Chercher un fichier sur votre disque en se basant sur son nom: `find {{ nom du fichier }}`
- Chercher du texte dans un fichier: `grep -rnw {{ nom du fichier }} -e {{ expression à chercher }}`
- Afficher le texte "Bonjour tout le monde": `echo "Bonjour tout le monde"`
- Afficher l'historique des commandes qui ont été tapées: `history`
- Afficher la version du système d'exploitation installée: `lsb_release -v`
- Afficher la date et l'heure: `date`
- Afficher la durée depuis laquelle le système d'exploitation est allumé: `uptime`
- Rechercher les mises à jour disponibles pour le système:
- Installer les nouvelles mises à jour disponibles depuis la dernière recherche: `sudo apt update`
- Se connecter en tant que superutilisateur: `su {{ opération que l'on souhaite }}`
- Installer l'éditeur de texte "emacs": `sudo apt install emacs`

- Connaître son/ses adresses ip: ip a

Debian:

SSH :

Toutes les commandes comportant une étoile sont à taper dans le terminal présent sur la machine virtuelle hôte du serveur SSH:

Installation du client *: `apt install openssh-client`

Installation du server *: `apt install openssh-server`

Vérifions maintenant que le serveur SSH soit bien actif *: `sudo systemctl status ssh`

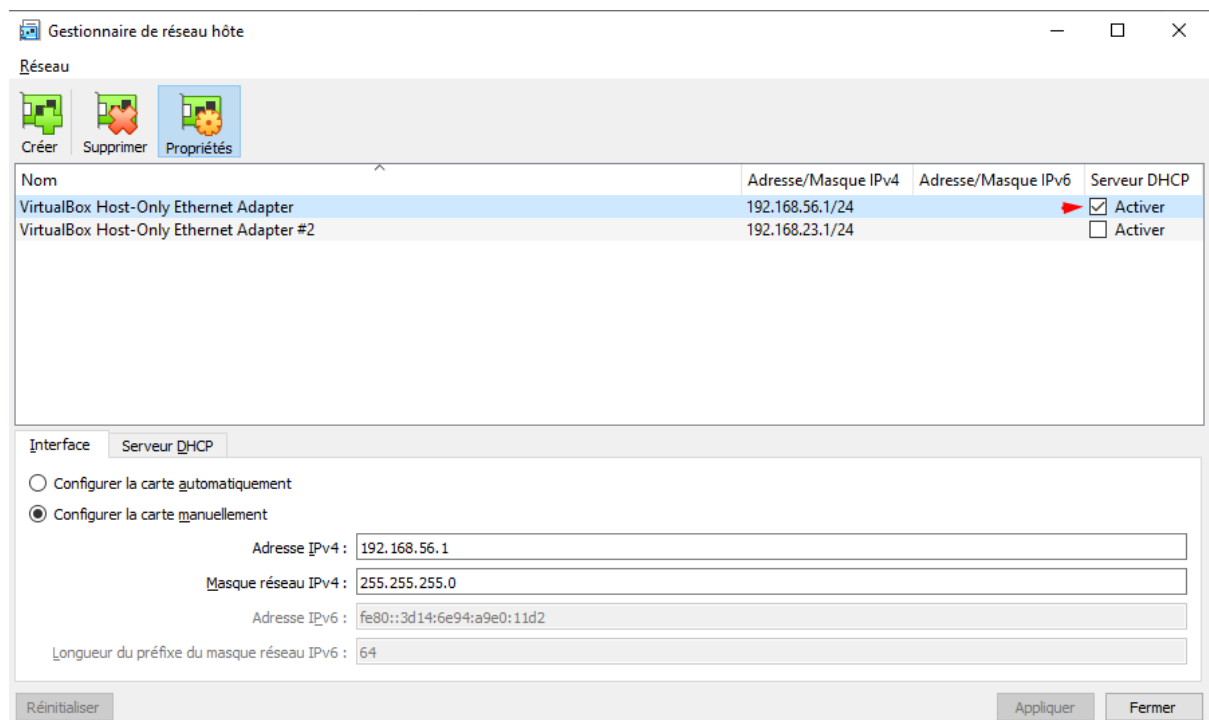
Si le serveur n'est pas actif : `sudo systemctl start ssh.service`

Si nous devons redémarrer le serveur: `sudo systemctl restart ssh.service`

Si nous devons stopper le serveur: `sudo systemctl stop ssh.service`

Windows :

Si votre machine host et sous windows il faudra alors sélectionner la carte wifi de votre machine pour cela allez dans Virtualbox > Gestionnaire de réseaux host et activer 'Server DHCP'



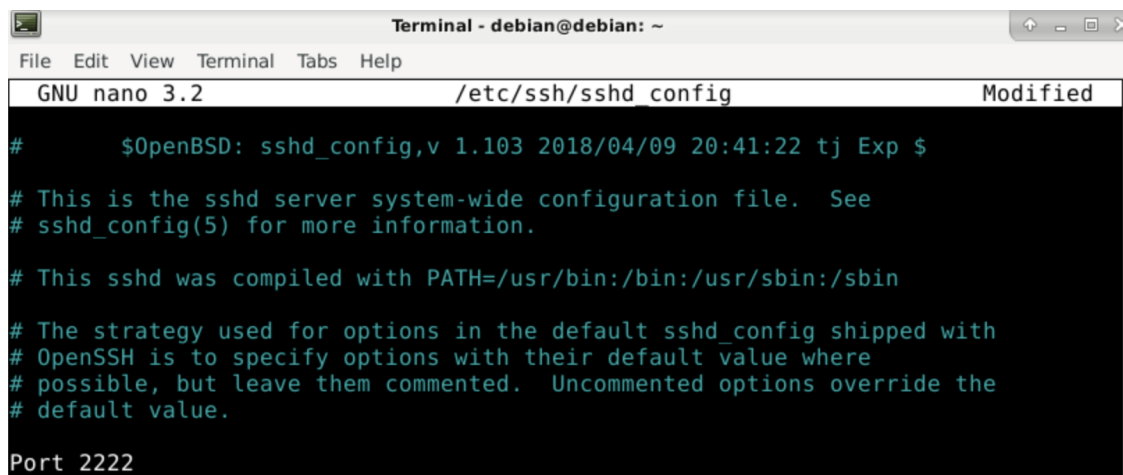
Une fois cela fait toujours, dans Virtualbox allez dans configuration > Réseau
Sélectionner :

- Mode d'accès réseau : Accès par pont
- Nom : *Sélectionner votre cart wifi de votre PC*

Nous souhaitons connaître l'adresse IP de la machine virtuelle hôte sur serveur SSH *: ip a

Ainsi que le nom de la machine virtuelle * (nous en aurons besoin pour nous connecter en ssh depuis notre machine invitée) : hostname

Continuons la mise en place du serveur en éditant le fichier de configuration pour le serveur ssh * : `sudo nano /etc/ssh/sshd_config`

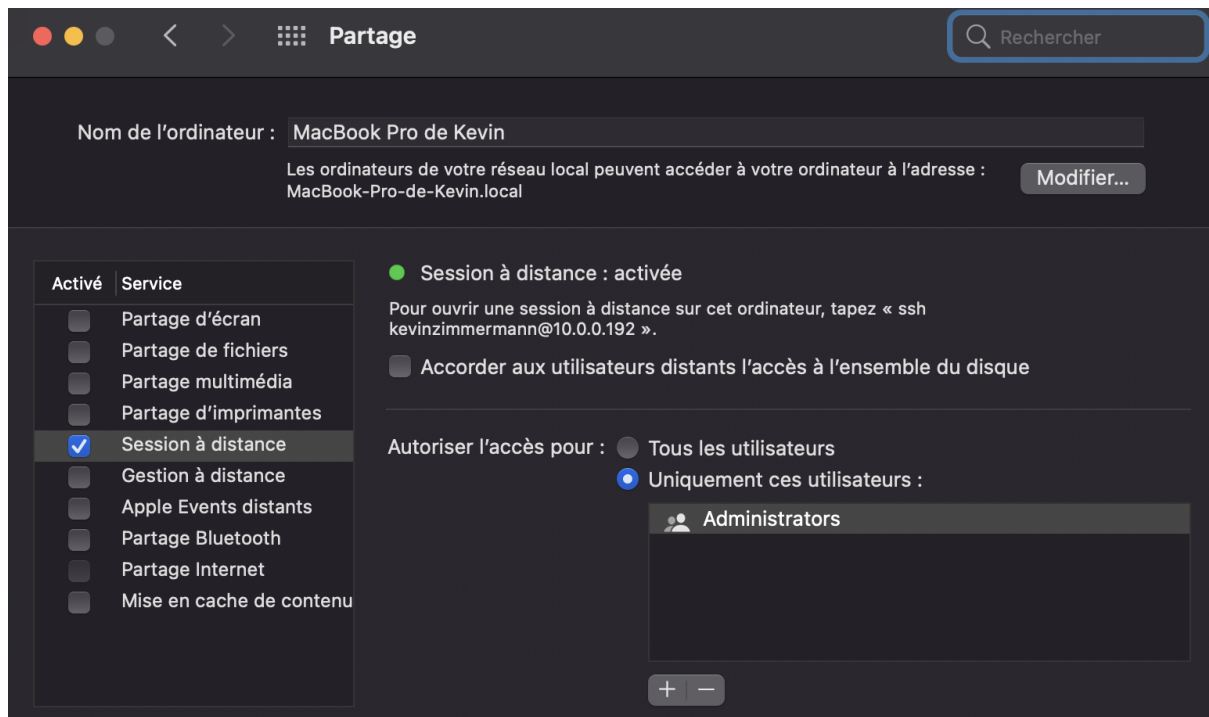


```
Terminal - debian@debian: ~
File Edit View Terminal Tabs Help
GNU nano 3.2 /etc/ssh/sshd_config Modified
# $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.
Port 2222
```

Ici nous changeons le port qui est défini par défaut sur le port 22.

Maintenant que le serveur SSH est créé et bien paramétré, nous devons nous occuper de pouvoir connecter notre serveur SSH présent sur la machine virtuelle à notre machine invitée.

Pour se connecter à notre serveur, nous devons activer dans Préférences Système > Partage et cocher le service "Session à distance" :



Pour pouvoir nous connecter à notre serveur SSH il faut générer une clé SSH sur notre machine invitée grâce à la commande: `ssh-keygen -t rsa -C {{ adresse ip obtenue grâce à "ip a" sur notre machine qui accueille le serveur SSH }}`

```
kevinzimmermann@MacBook-Pro-de-Kevin ~ % ssh-keygen -t rsa -C 10.0.0.149
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/kevinzimmermann/.ssh/id_rsa):
/Users/kevinzimmermann/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/kevinzimmermann/.ssh/id_rsa.
Your public key has been saved in /Users/kevinzimmermann/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:kl7k7bF2XLJaD8940xTJ7PzMZd+Tvc4PmHDULuta8fg 10.0.0.149
The key's randomart image is:
+---[RSA 3072]---+
|
|.      .
|.    .0..
|+ . . . =
|o S + =oo.
|. o . * %+ o
|.  + %.o=*
|. * B++B
|o.o.E*=
+-----[SHA256]-----+
```

Ici nous essayons de nous connecter à notre serveur:

Nous utilisons la commande: `ssh {{nom obtenu grâce à la commande "hostname" sur notre VM }}@{{ adresse ip obtenue grâce à "ip a" sur notre machine qui accueille le serveur SSH }} -p {{ le port que nous avons défini dans le fichier sshd_config }}`.

Nous cherchons à nous faire authentifier par le serveur sur la machine virtuelle grâce à la clé SSH que nous avons généré sur notre machine invitée. Nous choisissons donc l'option fingerprint deux fois et puis nous validons avec le yes.

Nous devons ensuite taper le mot de passe de la session du super administrateur de la VM.

Ensuite la connexion est ensuite établie entre notre machine virtuelle ainsi que notre machine invitée.

```
kevinzimmermann@MacBook-Pro-de-Kevin ~ % ssh debian@10.0.0.149 -p 2222
The authenticity of host '[10.0.0.149]:2222 ([10.0.0.149]:2222)' can't be established.
ECDSA key fingerprint is SHA256:WnhdH3Kd19Htij5PBBPMSW003SSY/W74iBAARmA9Xbc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? fingerprint
Please type 'yes', 'no' or the fingerprint: fingerprint
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '[10.0.0.149]:2222' (ECDSA) to the list of known hosts.
debian@10.0.0.149's password:
Linux debian 4.19.0-18-arm64 #1 SMP Debian 4.19.208-1 (2021-09-29) aarch64

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: Wed Oct 13 09:01:22 2021 from 10.0.4.77
debian@debian:~$
```

Apache :

Installer: `apt install apache2`

start server : `systemctl start apache2.service`

stop server : `systemctl stop apache2.service`

restart server : `systemctl restart apache2.service`

Pour créer un site, commencer par créer un dossier dans le répertoire www, pour cela utiliser cette commande : `mkdir /var/www/exemple`

Une fois cela fait utiliser cette commande :

`nano /etc/apache2/sites-available/exemple.com.conf`

et copier cela :

`<VirtualHost *:80>`

`ServerName exemple.com`

`ServerAlias www.exemple.com`

`ServerAdmin admin@exemple.com`

`DocumentRoot /var/www/exemple`

`</VirtualHost>`

Vous pouvez maintenant tester cela en allant sur votre machine host ouvre un navigateur internet et mettre l'ip de la VM.

PHP :

Installer: `apt -y install php php-common`

php version : `php -v`

Instal extension : `apt -y install php-cli php-fpm php-json php-pdo php-mysql php-zip php-gd php-mbstring php-curl php-xml php-pear php-bcmath`

Config php/apache :

`apt -y install libapache2-mod-php`

`sudo a2enmod php7.*`

`sudo systemctl restart apache2`

MYSQL :

Mariadb :

Dans un premier temps il faut installer mariadb pour cela exécuter cette commande dans votre terminal : `apt install mariadb-server`

phpMyAdmin:

Puis il faut installer phpMyAdmin pour cela exécuter c'est 4 ligne de commande :

1. `DATA="$(wget https://www.phpmyadmin.net/home_page/version.txt -q -O-)"`
2. `URL="$(echo $DATA | cut -d ' ' -f 3)"`
3. `VERSION="$(echo $DATA | cut -d ' ' -f 1)"`
4. `wget`

[https://files.phpmyadmin.net/phpMyAdmin/\\${VERSION}/phpMyAdmin-\\${VERSION}-all-languages.tar.gz](https://files.phpmyadmin.net/phpMyAdmin/${VERSION}/phpMyAdmin-${VERSION}-all-languages.tar.gz)

Puis on va extraire les fichier avec cette commande :

- `tar xvf phpMyAdmin-${VERSION}-all-languages.tar.gz`

Une fois extrait il faut placer tout ça dans le dossier `/usr/share/phpmyadmin` pour cela exécuter cela :

- `sudo mv phpMyAdmin-*/ /usr/share/phpmyadmin`

Il faut créer un dossier pour les fichier temporaire de phpMyAdmin pour cela faite ceci :

- `sudo mkdir -p /var/lib/phpmyadmin/tmp`
- `sudo chown -R www-data:www-data /var/lib/phpmyadmin`

On va créer un répertoire qui va continuer les fichier de configuration pour phpMyAdmin

- `sudo mkdir /etc/phpmyadmin/`

Puis il faut créer un fichier de configuration :

- `sudo cp /usr/share/phpmyadmin/config.sample.inc.php /usr/share/phpmyadmin/config.inc.php`

On va éditer ce fichier pour cela faite ceci :

- `sudo nano /usr/share/phpmyadmin/config.inc.php`

Il faudra éditer :

- `blowfish_secret` : il faudra mettre une clé de 32 character par exemple :
`QN.p:rDqhMp,o.h8B].N,k..P9NfxlEn`
- Il faudra ajouter (ou éditer si cette ligne existe déjà) cette ligne : `$cfg['TempDir'] = '/var/lib/phpmyadmin/tmp';`
- il faudra dé-commenter les ligne :
 - `controlhost`
 - `controlport`
 - `controluser`
 - `controlpass`
 - `pmadb`
 - `bookmarktable`
 - `relation`
 - `table_info`
 - `table_coords`
 - `pdf_pages`
 - `column_info`
 - `history`
 - `table_uiprefs`
 - `tracking`
 - `userconfig`
 - `recent`
 - `favorite`
 - `users`
 - `usergroups`
 - `navigationhiding`
 - `savedsearches`
 - `central_columns`
 - `designer_settings`
 - `export_templates`

```

/* User used to manipulate with storage */
$cfg['Servers'][$i]['controlhost'] = 'localhost';
$cfg['Servers'][$i]['controlport'] = '3306';
$cfg['Servers'][$i]['controluser'] = 'pma';
$cfg['Servers'][$i]['controlpass'] = 'PmaUser';

/* Storage database and tables */
$cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
$cfg['Servers'][$i]['bookmarktable'] = 'pma__bookmark';
$cfg['Servers'][$i]['relation'] = 'pma__relation';
$cfg['Servers'][$i]['table_info'] = 'pma__table_info';
$cfg['Servers'][$i]['table_coords'] = 'pma__table_coords';
$cfg['Servers'][$i]['pdf_pages'] = 'pma__pdf_pages';
$cfg['Servers'][$i]['column_info'] = 'pma__column_info';
$cfg['Servers'][$i]['history'] = 'pma__history';
$cfg['Servers'][$i]['table_uiprefs'] = 'pma__table_uiprefs';
$cfg['Servers'][$i]['tracking'] = 'pma__tracking';
$cfg['Servers'][$i]['userconfig'] = 'pma__userconfig';
$cfg['Servers'][$i]['recent'] = 'pma__recent';
$cfg['Servers'][$i]['favorite'] = 'pma__favorite';
$cfg['Servers'][$i]['users'] = 'pma__users';
$cfg['Servers'][$i]['usergroups'] = 'pma__usergroups';
$cfg['Servers'][$i]['navigationhiding'] = 'pma__navigationhiding';
$cfg['Servers'][$i]['savedsearches'] = 'pma__savedsearches';
$cfg['Servers'][$i]['central_columns'] = 'pma__central_columns';
$cfg['Servers'][$i]['designer_settings'] = 'pma__designer_settings';
$cfg['Servers'][$i]['export_templates'] = 'pma__export_templates';

```

Et il faudra éditer les lignes :

- controlhost : localhost
- controlport : 3306
- controlpass : *Mettre le mot de passe que vous voulez*

Configuration de phpMyAdmin avec apache il faut crée un fichier en faisant cela :

- nano /etc/apache2/conf-enabled/phpmyadmin.conf

dans ce fichier collaisie ce-ci :

Alias /phpmyadmin /usr/share/phpmyadmin

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

```
Options SymLinksIfOwnerMatch
```

```
DirectoryIndex index.php
```

```
<IfModule mod_php5.c>
```

```
<IfModule mod_mime.c>
```

```
AddType application/x-httpd-php .php
```

```
</IfModule>
```

```
<FilesMatch ".+\.php$">
```

```
SetHandler application/x-httpd-php
```

```
</FilesMatch>
```

```

    php_value include_path .
    php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
    php_admin_value open_basedir
/usr/share/phpmyadmin/:/etc/phpmyadmin/:/var/lib/phpmyadmin/:/usr/share/php/php-gettext/:
/usr/share/php/php-php-gettext/:/usr/share/javascript/:/usr/share/php/tcpdf/:/usr/share/doc/ph
pmyadmin/:/usr/share/php/phpseclib/
    php_admin_value mbstring.func_overload 0
</IfModule>
<IfModule mod_php.c>
    <IfModule mod_mime.c>
        AddType application/x-httpd-php .php
    </IfModule>
    <FilesMatch ".+\.php$">
        SetHandler application/x-httpd-php
    </FilesMatch>

    php_value include_path .
    php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
    php_admin_value open_basedir
/usr/share/phpmyadmin/:/etc/phpmyadmin/:/var/lib/phpmyadmin/:/usr/share/php/php-gettext/:
/usr/share/php/php-php-gettext/:/usr/share/javascript/:/usr/share/php/tcpdf/:/usr/share/doc/ph
pmyadmin/:/usr/share/php/phpseclib/
    php_admin_value mbstring.func_overload 0
</IfModule>

</Directory>

# Authorize for setup
<Directory /usr/share/phpmyadmin/setup>
    <IfModule mod_authz_core.c>
        <IfModule mod_authn_file.c>
            AuthType Basic
            AuthName "phpMyAdmin Setup"
            AuthUserFile /etc/phpmyadmin/htpasswd.setup
        </IfModule>
        Require valid-user
    </IfModule>
</Directory>

# Disallow web access to directories that don't need it
<Directory /usr/share/phpmyadmin/templates>
    Require all denied
</Directory>
<Directory /usr/share/phpmyadmin/libraries>
    Require all denied
</Directory>
<Directory /usr/share/phpmyadmin/setup/lib>
    Require all denied

```

</Directory>

Une fois cela fait vous pourrez redémarrer le serveur apache pour cela faite ceci dans le terminal :

- `sudo systemctl restart apache2`

Si on veut changer l'url pour accéder à phpMyAdmin il faut juste remplacer la première ligne. Par exemple on veut remplacer l'url phpmyadmin en pma le premier ligne deviendra :

- `Alias /pma /usr/share/phpmyadmin`

pour configurer phpmyadmin il faut faire quelque commande sur le terminal :

- Se connecter à un utilisateur : `mysql -u user`
 - Utilisateur de base : root:
- Crée user : `CREATE USER 'user'@'localhost' IDENTIFIED BY 'password';`
- Donné les privilège à un utilisateur : `GRANT ALL PRIVILEGES ON *.* TO 'user'@'localhost';`
 - Si on veut donner les privilège à uniquement une base de données on remplace le premier : * par le nom de la base de données, ce qui devien : `GRANT ALL PRIVILEGES ON db_name.* TO 'user'@'localhost';`
- Retirer les droit à un utilisateur : `REVOKE ALL PRIVILEGES ON *.* FROM 'user'@'localhost';`
- Crée base de données : `CREATE DATABASE db_name;`
- Change password : `SET PASSWORD FOR 'user'@'localhost' = PASSWORD('new-password');`

Certificat SSL

Un certificat SSL est un fichier de données qui lie une clé cryptographique aux informations d'une organisation. Installé sur un serveur, le certificat active le cadenas et le protocole « https », afin d'assurer une connexion sécurisée entre le serveur web et le navigateur. Le SSL est généralement utilisé pour sécuriser les transactions bancaires, le transfert de données et les informations de connexions. Il est récemment devenu la norme pour sécuriser la navigation sur les sites de réseaux sociaux.

Déploiement de l'application mobile sur l'App Store

1# étape : Inscription compte Apple Developer

La création d'un compte Apple Developer coûte 99€ par an et donne accès à l'App Store Connect pour gérer et soumettre les applications.

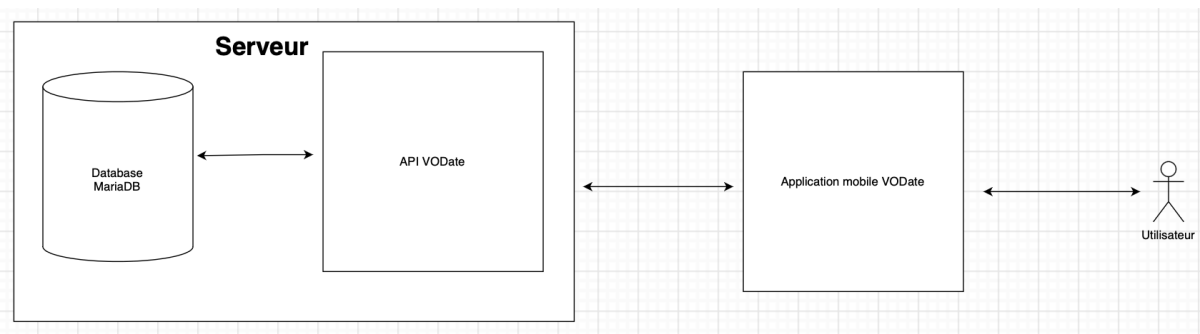
2# étapes : Tester la version Ad Hoc

Avec l'application Xcode, on génère le fichier IPA qui contient tous les éléments nécessaires au bon fonctionnement de l'application sur un appareil Apple. Ensuite nous pouvons tester la version Ad Hoc de manière approfondie (test sur plusieurs appareils, pas de bug...) celle-ci sera publiée sur l'App Store.

#3 étapes : Soumission de l'application à Apple

Les applications sont soumises à une vérification avant d'être publiées, elles doivent être conformes à la réglementation (description détaillée, capture d'écrans) de l'App Store.

Diagramme de déploiement:



Annexe:

<https://www.digitalocean.com/community/tutorials/how-to-use-ssh-to-connect-to-a-remote-server-fr>

<https://aonyx.fr/pdf/notices/procedure-installation-serveur-web.pdf>

<https://computingforgeeks.com/install-php-on-debian-linux-system/>

<https://computingforgeeks.com/install-phpmyadmin-with-apache-on-debian/>

<https://doc.ubuntu-fr.org/ssh>

<https://www.macg.co/logiciels/2021/02/virtualisation-utm-est-aussi-disponible-sur-macos-y-compris-pour-les-mac-apple-silicon-119877>

<https://computingforgeeks.com/install-phpmyadmin-with-apache-on-debian/>

https://app.diagrams.net/#G1y4pUk-8xhrKZrazPiPSFGR3_vp4bJrh2