

## Job 1 - Virtualisation

Téléchargement VirtualBox : **OS X hosts**

Si l'ouverture du fichier est bloquée : **Préférences de sécurité > autoriser le chargement du logiciel**

## Job 2 - Installation de GNU/Linux

Installation distribution debian :

**debian.org > obtenir debian > image d'installation de taille réduite > amd64**

Création d'une VM : **Nouvelle > Choix du nom de VM > type : linux > 1Go de mémoire vive (1024 mb) > disque dur VDI (Virtual box disk image) 10 Go > allocation dynamique**

Pour un démarrage via l'ISO : **clic droit VM > configuration > stockage > lecteur CD > debian**

Démarrage : **graphical install > langue : français > pays : France > clavier : français > nom système : deb64 > nom de domaine : > mdp super utilisateur : root > nom complet utilisateur : user > utilisateur login : user > utilisateur mdp : user > partitionner les disques : assisté - utiliser un disque entier > tout dans une seule partition > table de partition des périphériques - partition formatée : oui > outil de gestion des paquets : france - miroir archive debian : deb.debian.org - mandataire http : > envoi statistique : non > sélection des logiciels : environnement de bureau Debian - xfce - utilitaire usuel du système > installer programme de démarrage grub : oui > choix du périphérique de stockage ?**

Prise d'instantané :



**clic VM icône settings > instantanés > prendre**

## Job 3 - La fenêtre noire

Ouvrir le terminal : **clic droit bureau > Applications > Émulateur de terminal**

modifier le clavier (utile pour mac) : **# dpkg-reconfigure keyboard-configuration**

## Job 4 - Commandes système

Commande	Ligne de commande
Afficher le répertoire en cours	pwd
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 -l -a
Créer un fichier	touch + nom du fichier.extension
Insérer du texte dans un fichier	<p>echo + "text" &gt;&gt; chemin vers le fichier + nom fichier.extension</p> <p>ex : echo " nouveau texte " &gt;&gt; /home/user/Bureau/hello.txt</p>
Supprimer un fichier	rm + nom du fichier.extension
Afficher le contenu d'un fichier	cat + nom du fichier.extension
Créer un répertoire	mkdir + nom du dossier
Créer un lien symbolique (renvoie au fichier original)	<p>ln + -s + nom du fichier.extension + nom du lien symbolique</p> <p>ex : ln -s hello.txt + lien_symb2</p>
Supprimer un répertoire	rmdir + nom du répertoire
Copier un répertoire	<p>cp -r + Nom du répertoire à copier + Nom du nouveau répertoire</p> <p>ex : cp -r Borderland Wonderland</p>
Renommer un répertoire	mv + Nom actuel du répertoire + Nouveau nom de répertoire
Déplacer un répertoire	<p>mv + Nom du répertoire à déplacer + chemin vers la destination</p> <p>ex :</p>

	mv Borderworld Borderland/
Afficher le manuel de la commande "find"	man find
<a href="#">Chercher un fichier sur votre disque en se basant sur son nom</a>	find + -name + nom du fichier.extension find -name hello.txt
Chercher du texte dans un fichier	grep + "mon mot" + nom du fichier.extension
Afficher le texte "Bonjour tout le monde"	echo + "Bonjour tout le monde"
Afficher l'historique des commandes qui ont été tapées	ctrl + r
Afficher la version du système d'exploitation installée	lsb_release -a
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	sudo apt-get update
Installer les nouvelles mises à jour disponibles depuis la dernière recherche	sudo apt-get upgrade
Se connecter en tant que superutilisateur	su - (mdp root)
Installer l'éditeur de texte "emacs"	sudo apt-get install emacs
Connaître son/ses adresses ip	hostname -I (ceci est un i)
Vérifier la version de Debian	cat /etc/debian_version

## Job 5 - Prise en main à distance

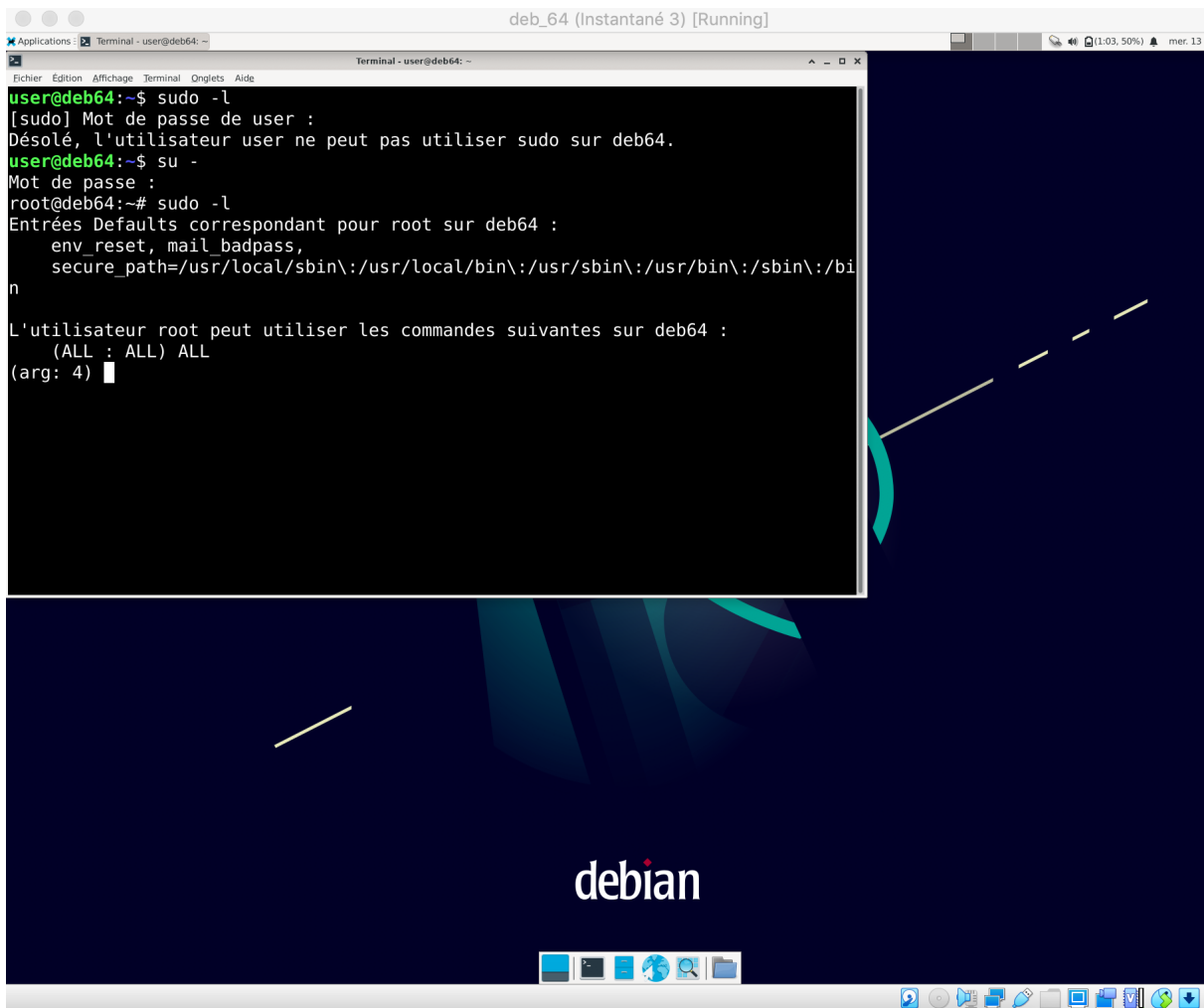
*modifier un fichier avec l'éditeur de texte nano : nano + -l + /chemin vers le fichier/fichier.extension*

find / -name sshd\_config

Il faut exécuter un certain nombre de commandes pour installer un serveur ssh et faire quelques vérifications.

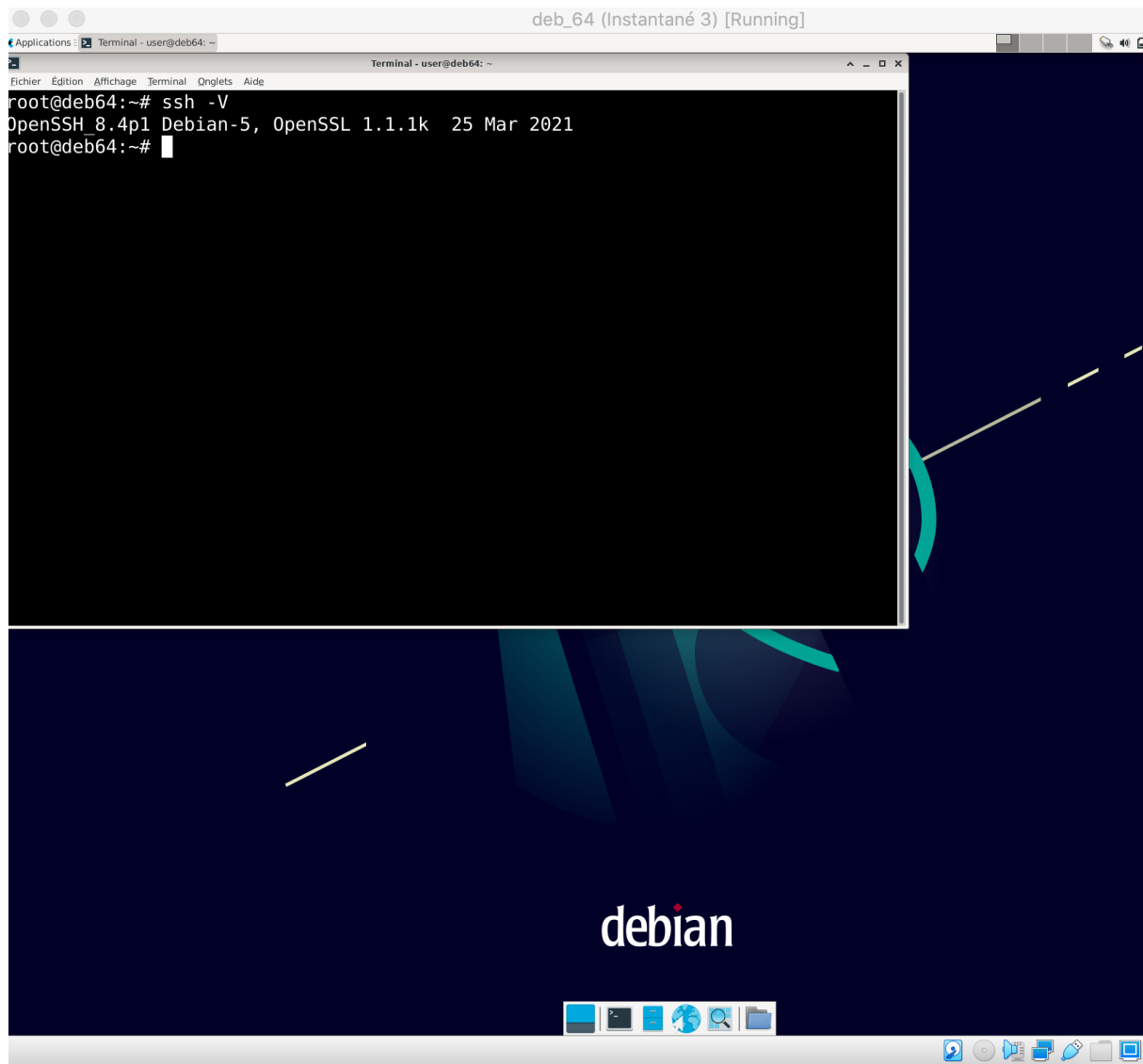
## Prerequisites

Il faut avoir les privilèges sudo pour installer un serveur SSH. Vérifier si on a les **privilèges sudo** : **# sudo -l**



```
deb_64 (Instantané 3) [Running]
Applications : Terminal - user@deb64: ~
Terminal - user@deb64: ~
user@deb64:~$ sudo -l
[sudo] Mot de passe de user :
Désolé, l'utilisateur user ne peut pas utiliser sudo sur deb64.
user@deb64:~$ su -
Mot de passe :
root@deb64:~# sudo -l
Entrées Defaults correspondant pour root sur deb64 :
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
L'utilisateur root peut utiliser les commandes suivantes sur deb64 :
    (ALL : ALL) ALL
(arg: 4)
```

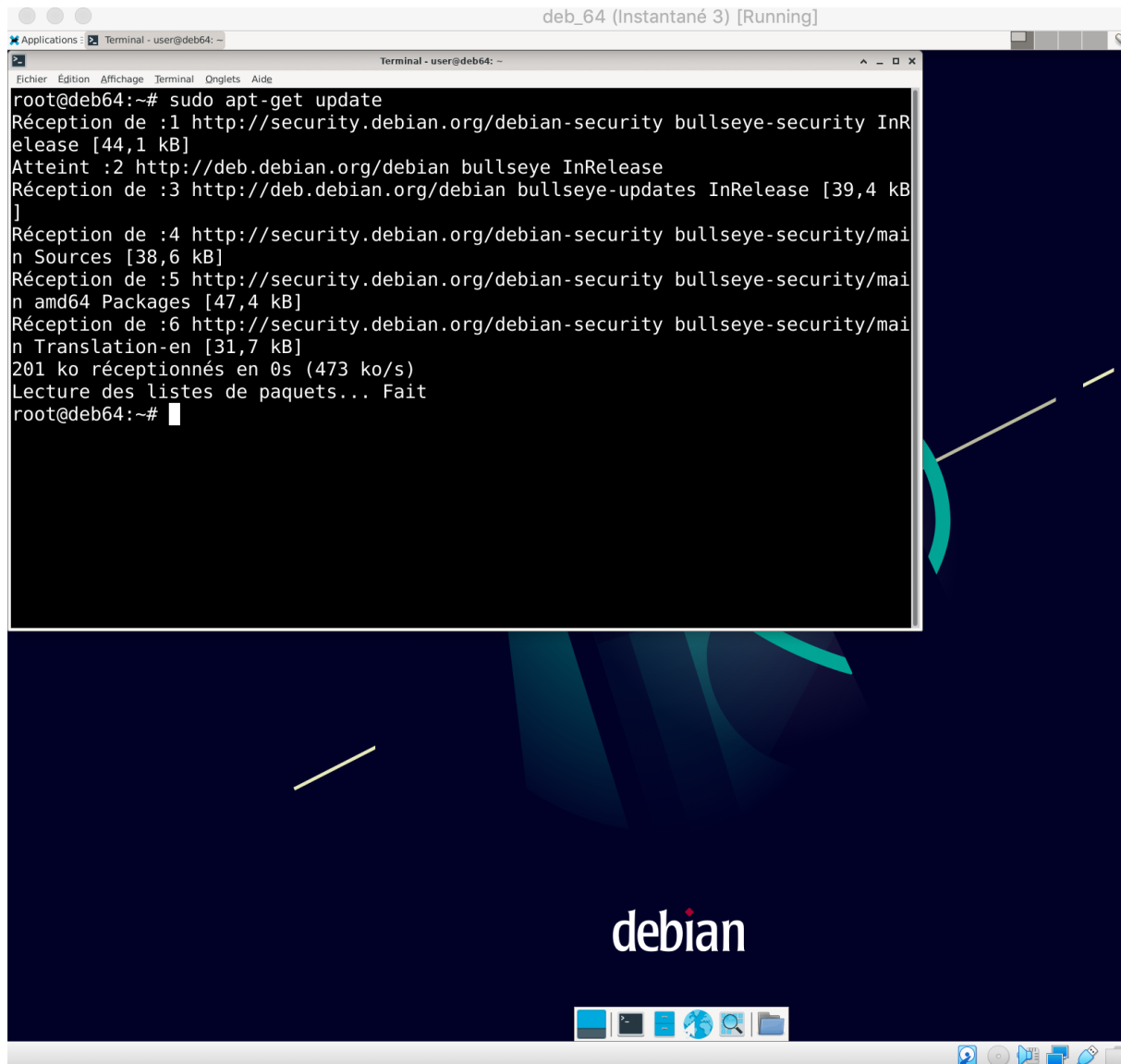
Par défaut, l'utilitaire ssh est installé sur notre hôte. Vérifier la **version** de notre utilitaire **SSH** : **# ssh -V**



## Installing OpenSSH Server

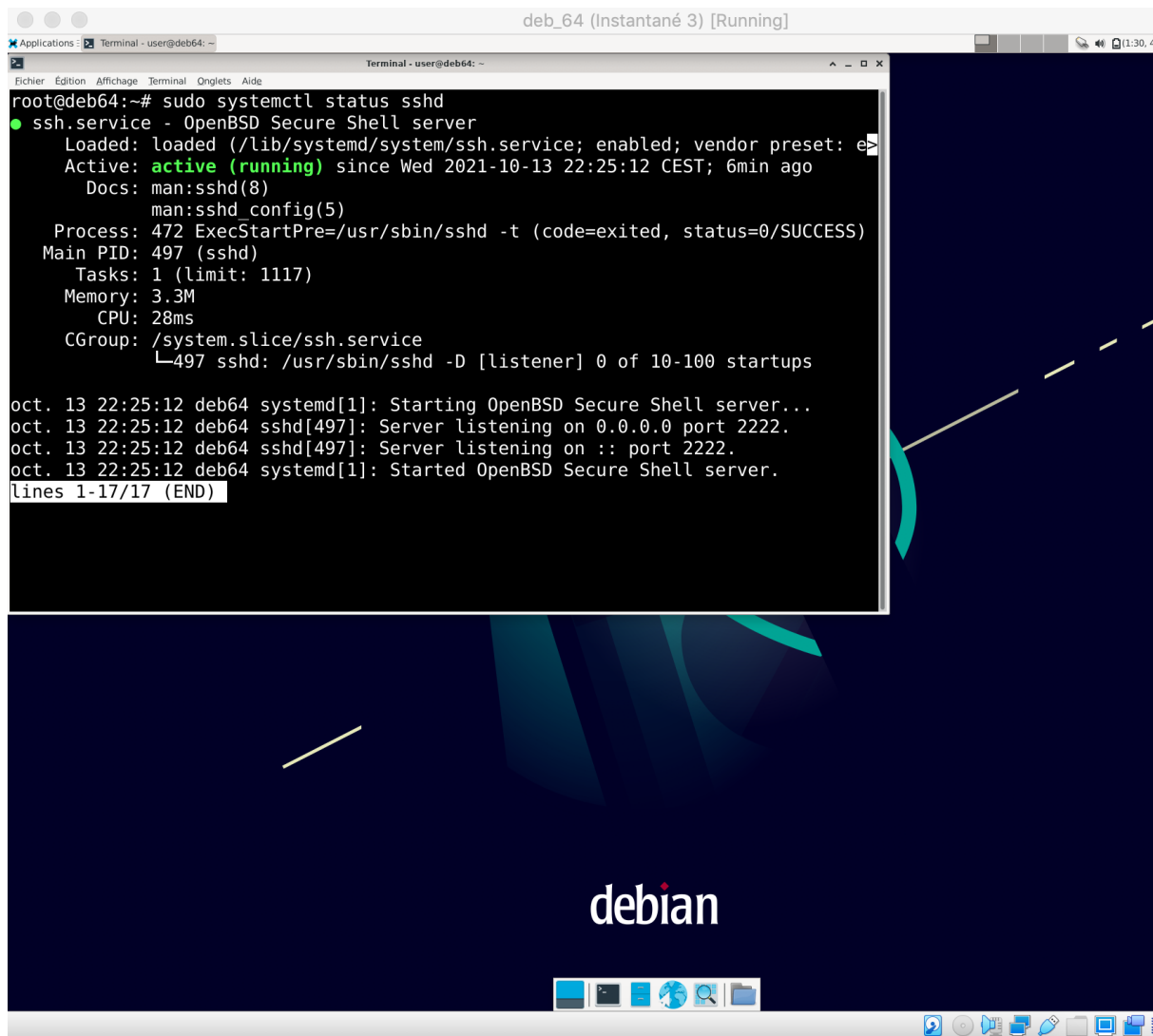
D'abord, s'assurer que les **paquets** sont **à jour** en exécutant la commande update : **# sudo apt-get update**

Si ça ne fonctionne pas il faut vérifier si la VM est bien connectée à internet (VM configuration > réseau > mode d'accès réseau > NAT)



Pour installer un serveur SSH : **# sudo apt-get install openssh-server**

Pour vérifier si le service a bien été installé : **# sudo systemctl status sshd**



Le port par défaut est 22.

Pour vérifier le port du serveur ssh : **# netstat - tulpn | grep numéro de port (ici 22)**

Si la commande est introuvable installer [# apt-get install net-tools](#)

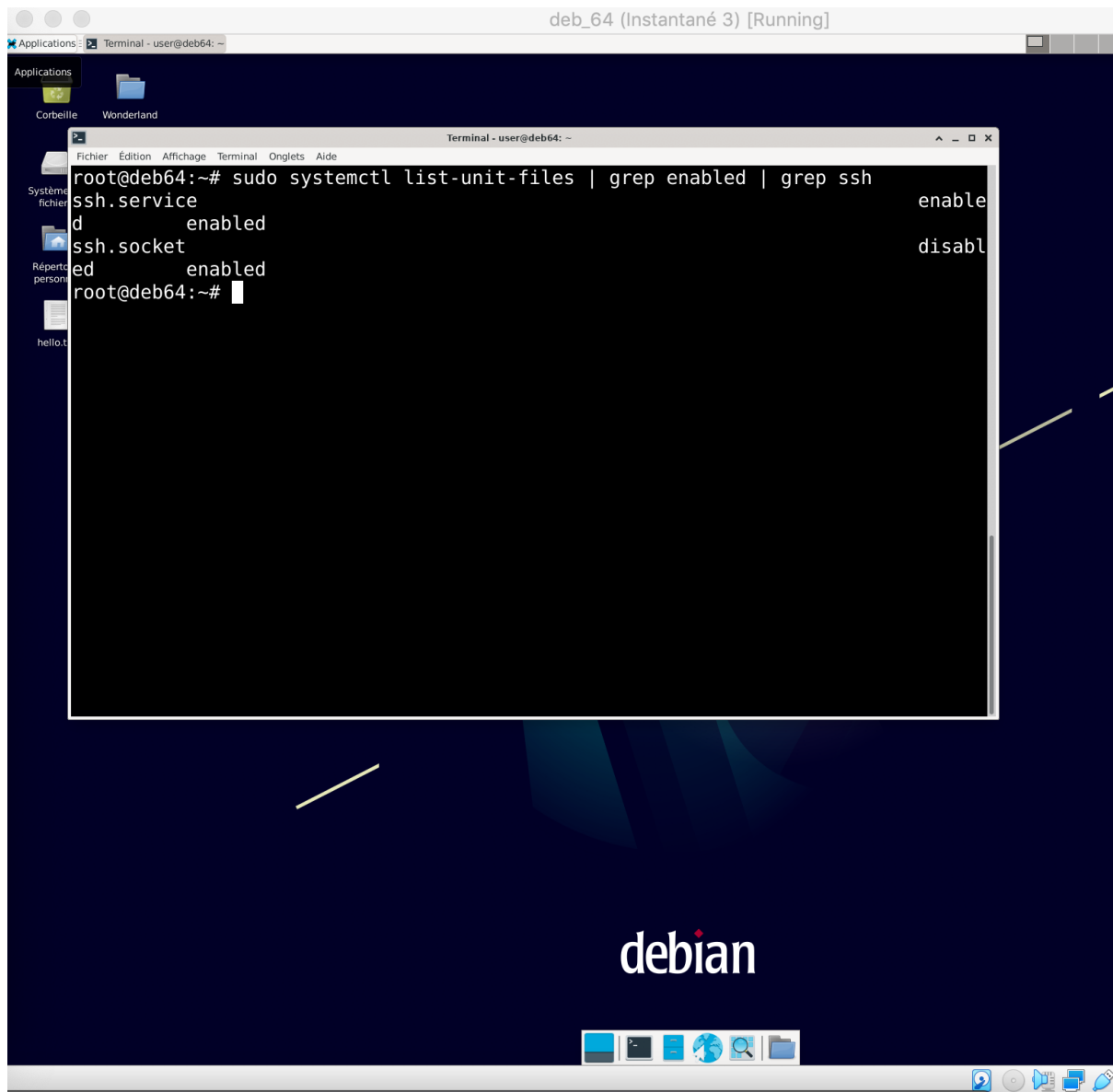
### *Autoriser le trafic SSH dans les paramètres de votre pare-feu*

*Si on utilise UFW comme pare-feu par défaut sur le système debian, il faut autoriser les connexions SSH sur l'hôte : **# sudo ufw allow ssh***

### **Activer le serveur SSH au démarrage du système**

Pour vérifier si votre service est activé ou non : **# sudo systemctl list-unit-files | grep enabled | grep ssh**

Si ça ne donne pas de résultat : **# sudo systemctl enable ssh** puis **# sudo systemctl list-unit-files | grep enabled | grep ssh**



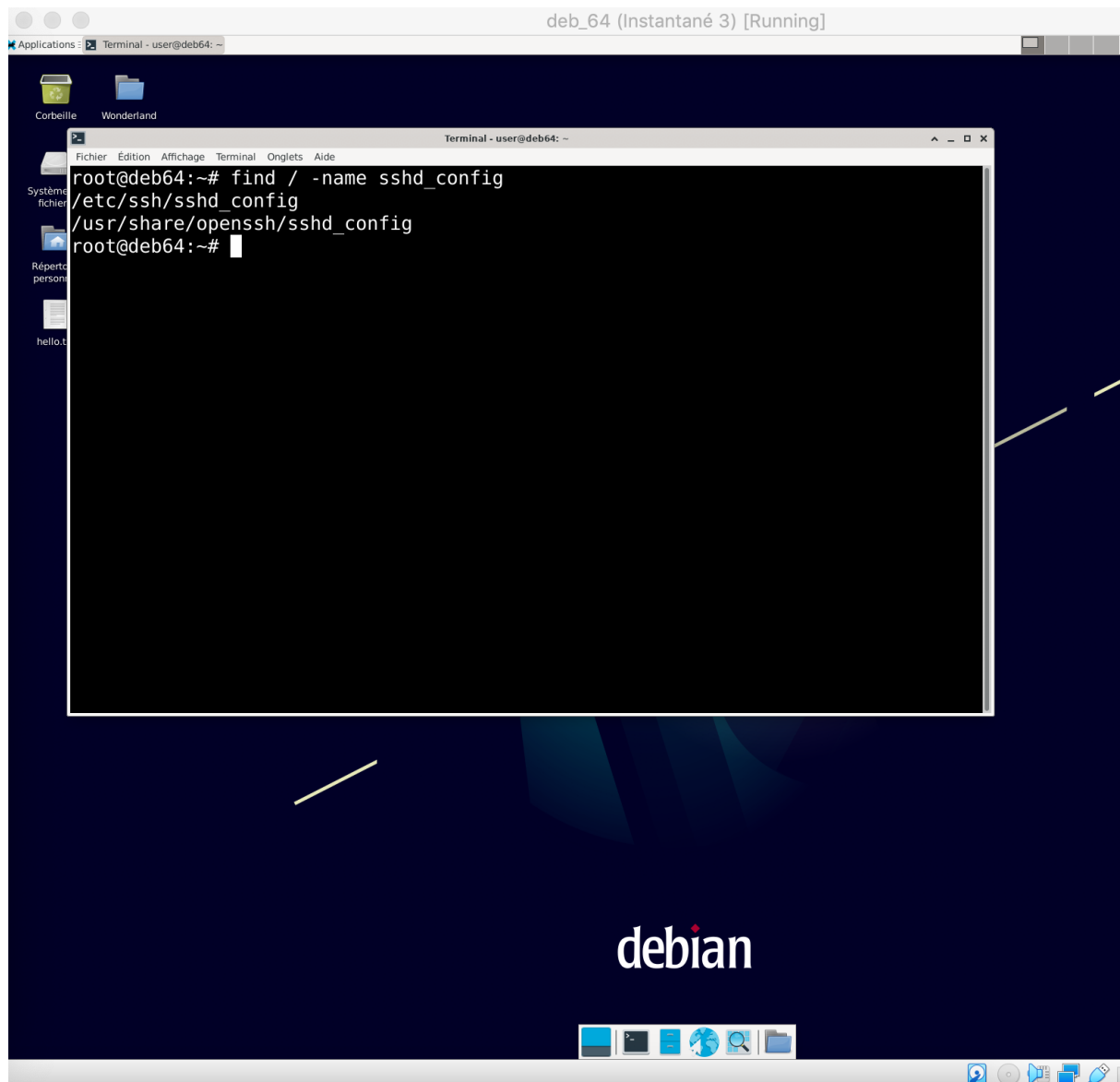
## Configurer le serveur SSH sur Debian

D'abord sécuriser en modifiant les paramètres par défaut.

### *Changer le port par défaut de SSH*

Pour changer le port par défaut il faut modifier le fichier `sshd_config`. D'abord trouver le chemin vers ce fichier : **# find / -name sshd\_config**





Pour éditer le fichier : **# nano /etc/ssh/sshd\_config**

```
deb_64 (Instantané 3) [Running]
Terminal - user@deb64: ~
GNU nano 5.4 /etc/ssh/sshd_config
$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.
Include /etc/ssh/sshd_config.d/*.conf
Port 2222
#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 no
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne
```

Modifier la ligne **#Port 22** en **Port 2222**, enregistrer et quitter (control+x et maj+0)

### **Désactiver le Root Login sur le serveur SSH**

Pour désactiver le login root sur le serveur SSH, modifiez la ligne :

**#PermitRootLogin prohibit-password** par **#PermitRootLogin no**

*Configuration de l'authentification SSH basée sur une clé : [Voir ici](#)*

**Redémarrage de votre serveur SSH pour appliquer les changements**

Il faut redémarrer le service SSH pour appliquer les changements : **# sudo systemctl restart sshd** puis vérifier le statut : **# sudo systemctl status sshd**

## Connexion à votre serveur SSH

Pour se connecter au serveur SSH, il faut utiliser la commande ssh avec la syntaxe suivante : **# ssh -p <port> <username>@<ip\_address>** (ex : **\$ ssh -p 2222 user@192.168.56.101**)

## Quitter votre serveur SSH

Quitter le serveur SSH : **Ctrl + D** ou taper **logout**

## Désactiver votre serveur SSH

Quitter

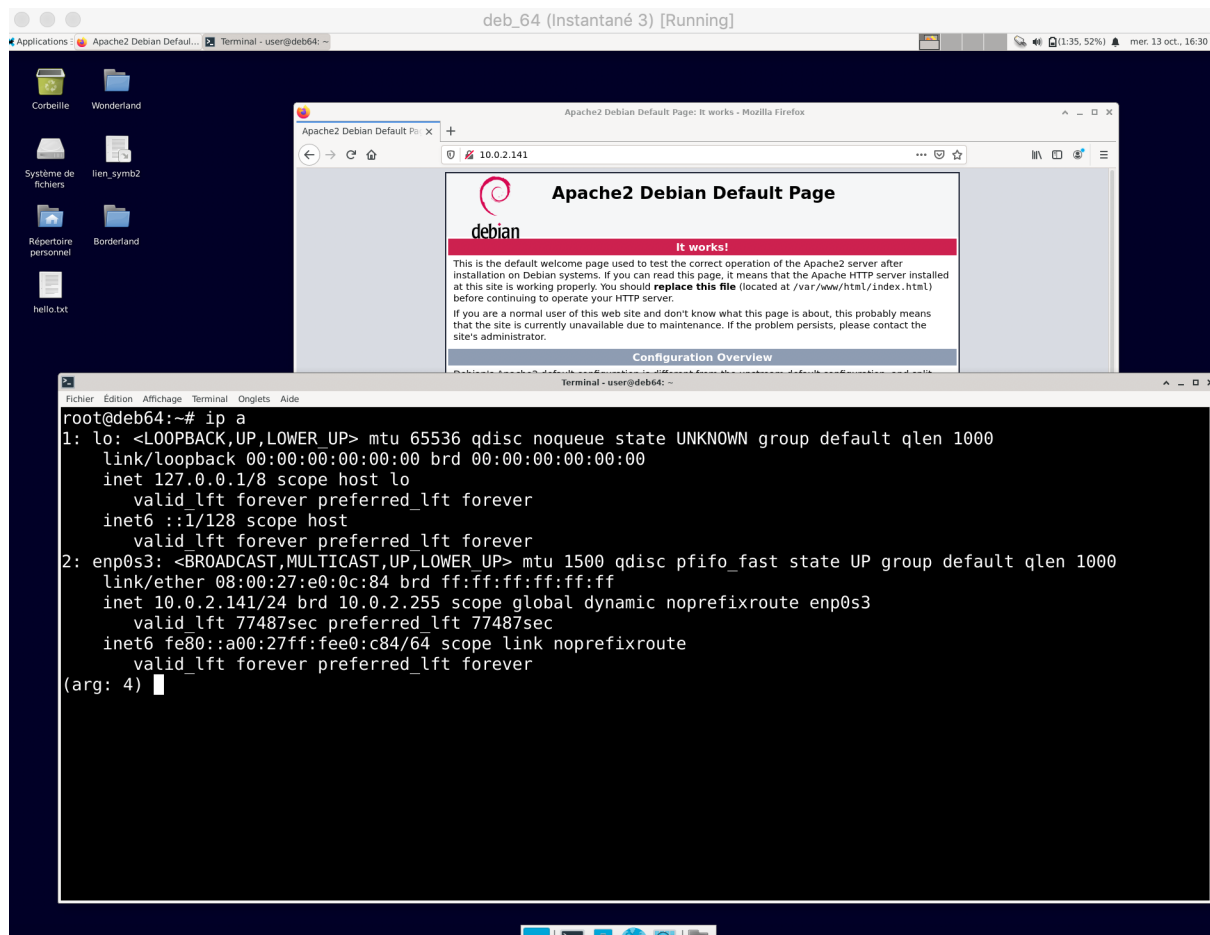
- Ajouter carte réseau

## Job 6 - Installation du serveur web

Un serveur lamp c'est linux apache mysql php

### Installation du serveur web Apache

- mise à jour des dépôts : **# apt update** puis **# apt full-upgrade**
- installation d'apache et du module qui permet d'exécuter PHP : **# apt-get install apache2 libapache2-mod-php**
- voir si le serveur web est bien actif : **# systemctl status apache2**
- voir si la page s'affiche, récupération de l'adresse ip : **# ip a**
- copier coller l'adresse ip dans la barre de recherche d'un navigateur



- *activer le mode rewrite pour réécrire les url sur certains cms : # a2enmod rewrite puis # systemctl restart apache2 ou # systemctl reload apache2*
- *pour voir la liste des modules qu'on peut activer : # ls etc/apache2/mods- puis # ls etc/apache2/mods-available*

## Installation PHP

- installer PHP et php cli qui permet d'exécuter des commandes php depuis la ligne de commande : **# apt install php php-cli**
- accéder à la liste des modules php : **# apt search ^php-**
- installation de certains modules : **# apt install php-{curl,gd,intl,memcache,xml,zip,mbstring,json}** puis **# systemctl reload apache2**
- Si on veut installer une base de données, il faut installer php mysql qui fournit les connecteurs pour se connecter à la base de données avec les requêtes mysql ou PDO : **# apt install php-mysql** puis **# systemctl reload apache2**
- Les fichiers de la racine du serveur web se trouvent dans : **# cd var/www/html**  
C'est ici qu'on stocke les fichiers de nos sites web par défaut. On y trouve un fichier index.html de base : **# cd /var/www/html ls**

- Pour vérifier le bon fonctionnement de PHP on crée un fichier : **# vi test.php =>**  
`<?php phpinfo()?>` [Comment éditer un fichier avec vi](#)

### Installation du serveur de base de données Maria DB

- installation Maria DB : **# apt install mariadb-server puis # mysql\_secure\_installation**
- réponses aux questions de l'installation : mdp = root - par sécurité suppression des utilisateurs anonymes = Y - désactivation de la connexion à root de manière distante = Y - suppression base de données test = Y - recharge des privilèges des tables = Y
- se connecter à la base de données : **# mysql -u root -p**

### Installation Phpmyadmin

On va utiliser wget qui permet de faire des requêtes http pour aller chercher un fichier html ou une image ou dans notre cas un fichier zip. On va donc récupérer l'adresse du lien de téléchargement de [Phpmyadmin](#).

- télécharger phpMyAdmin dans le dossier `/var/www/html` : **# wget https://files.phpmyadmin.net/phpMyAdmin/5.1.1/phpMyAdmin-5.1.1-all-languages.tar.gz**
- décompresser le fichier : **# tar xvf php+tab = # tar xvf phpMyAdmin-5.1.1-all-languages.tar.gz**
- supprimer du fichier zip : **# rm phpMyAdmin-5.1.1-all-languages.tar.gz**
- modifier le nom de dossier phpMyAdmin : **# mv phpMyAdmin-5.1.1-all-languages/pma**

mettre à jour php (dernière version)

## Job 7 - phpmyadmin

- afficher les commandes : `\h`
- afficher toutes les bases de données : **SHOW DATABASES;**
- travailler avec une base de données : **USE <databasename>;** (ex : **USE information\_schema;**)
- voir les tables d'une base de données : **SHOW tables;**
- créer une base de données : **CREATE DATABASE <databasename>;**
- créer une table : **CREATE TABLE <tablename> (<fieldname>\_id INT AUTO\_INCREMENT PRIMARY KEY,<fieldname> <TYPE>());**
- obtenir des informations sur la table : **DESCRIBE books;**
- [récupérer les données des tables](#)

capture

- [ajouter/modifier des données dans les tables](#)

capture

- [modifier la structure des tables](#)

capture

- [consulter les commandes basiques MariaDB](#)
- créer un utilisateur pour se connecter à phpmyadmin dans MariaDb : **#mysql -p puis mdp MariaDB puis GRANT ALL PRIVILEGES ON \*.\* TO root@localhost IDENTIFIED BY 'root' WITH GRANT OPTION;**
- L'accès à phpmyadmin est désormais possible depuis <http://localhost/pma> ou <http://192.168.56.101/pma>

## Job 8 - Déploiement du site

installer bdd dans phpmyadmin

créer un utilisateur dans la bdd

importer données de la base de données de développement

- installer git : **# apt-get install git**
- version de git : **# git --version**

### Git en ligne de commande

- cloner un repo de github :  
générer une paire de clé public clé privée rsa avec ssh : **# ssh-keygen**  
choisir un nom de fichier : **# marine-jacquens**  
passer passphrase  
déplacer les fichiers situés dans /root pour pouvoir les manipuler :  
**#cp marine-jacquens /home/user/Bureau**  
**#cp marine-jacquens.pub /home/user/Bureau**  
ajouter la clé publique dans github : **github>settings>SSH and GPG keys>New SSH key> copier le contenu de marine-jacquens.pub**

initier ssh agent pour pouvoir utiliser la clé privée : **# eval `ssh-agent`**

ajouter la clé privée : **# ssh-add marine-jacquens**

se placer dans le bon répertoire : **# cd /var/www**

cloner un répertoire : **# git clone <repo url ssh>**

**ex : git clone git@github.com:marine-jacquens/les\_plateformeurs.git**  
**OU**

recupérer un token d'authentification dans github : **github>settings>Developer settings>Personal access token>cocher uniquement repo**

**#git config --global user.name "marine-jacquens"**

**#git config --global user.email "[marine.jacquens@laplateforme.io](mailto:marine.jacquens@laplateforme.io)"**

**#git config -l**

**# git clone <repo url https>**

**ex : # git clone [https://github.com/marine-jacquens/les\\_plateformeurs.git](https://github.com/marine-jacquens/les_plateformeurs.git)**

entrer le user name : **marine-jacquens**

entrer le mdp : **token github**

- renommer le répertoire api : **# mv les\_plateformeurs api**

### Créer un projet symfony sur la machine host

- vérifier les extensions php : **\$ php -m**
- installer le gestionnaire de dépendance (packages) composer : **\$ curl -Ss https://getcomposer.org/installer | php**
- installer symfony cli qui fournit tous les outils dont on a besoin pour développer et exécuter notre application Symfony en local : **\$ curl -sS https://get.symfony.com/cli/installer | bash**

générer un .htaccess

### Modifier Virtual host

Sur apache nous pouvons créer deux ou 3 sites web et les héberger sur un seul et même serveur, c'est cela la notion de virtualhost.

créer un fichier de configuration pour le site :

- regarder les dossiers (de sites) disponibles sur apache :

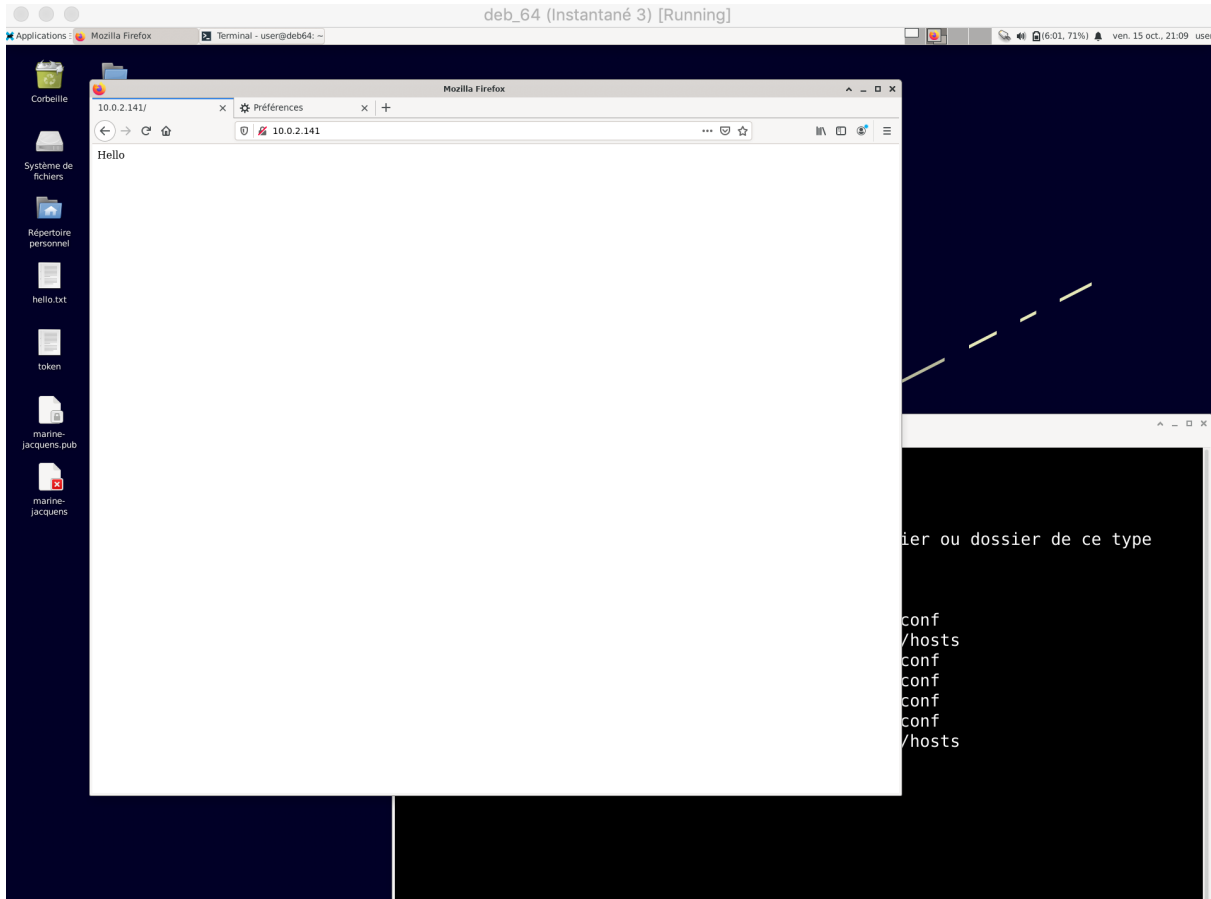
**# cd /etc/apache2/sites-available**

**# ls**

En allant sur l'adresse ip de notre debian on atterrit par défaut sur 000-default.conf, le ssl n'est pas actif par défaut

- repartir du fichier 0 : **# cp 000-default.conf api.conf**
- activer le lien vers le site : **# a2ensite api.conf** (a2 = apache2; en = enable)
- relancer serveur apache : **# systemctl reload apache2**
- revenir au dossier parent : **#cd ..**
- vérifier les sites activés : **# cd sites-enabled/**
- désactiver le site par défaut : **# a2dissite 000-default.conf**
- relancer serveur apache : **# systemctl reload apache2**
- *ls -lath pour vérifier qu'il s'agit bien de liens renvoyant vers nos dossiers de site*
- retourner dans le dossier des sites disponibles : **# cd ..** puis **# cd sites-available/**
- modifier la root vers laquelle pointe le api.conf : **# nano api.conf** puis **DocumentRoot /var/www/html/api**
- héberger le nom de domaine en local : **nano /etc/hosts** et ajouter **192.168.56.101 lesplateformeurs.com**
- **#nano api.conf** puis ajouter après **ServerAdmin** **Servername lesplateformeurs.com**  
Ainsi toutes les requêtes vers le port 80 avec le nom de domaine seront renvoyées vers lesplateformeurs.com
- redémarrer apache pour que les changements soient bien pris en compte : **#systemctl restart apache 2**
- relancer la VM : **#reboot**

- supprimer l'historique du navigateur : **Préférences>Vie privée et sécurité>Historique**
- accéder au site depuis le navigateur de la VM



- récupérer l'adresse IP de la VM : **# ip a**

Sur la machine host :

- associer l'ip de la VM au nom de domaine : **\$ sudo pico /etc/hosts** puis rentrer le mdp
- ajouter ip de la VM et nom de domaine du site : **192.168.56.101 lesplateformeurs.com**  
vider le cache : **\$ sudo dscacheutil -flushcache**
- **# ssh -p <port> <username>@<ip\_address>**  
(ex : **\$ ssh -p 2222 user@192.168.56.101**)
- entrer le mdp de user (user)

accéder au site depuis un portable connecté au wifi de l'école



répéter les opérations précédentes après avoir éteint la VM et modifier la carte réseau en accès par pont (bridge)

## Job 9 - Un peu de sécurité

- **Expliquer ce qu'est un certificat SSL**

SSL ou Secure Sockets Layer est un protocole de transmission de données sécurisées sur internet. Le **certificat SSL** (Secure Socket Layer) est un certificat électronique qui permet de sécuriser les communications entre des serveurs web et des navigateurs en empêchant des pirates de consulter ou de modifier les informations échangées entre deux systèmes. On ajoute des certificats SSL aux sites Web afin de sécuriser les transactions en ligne ou plus généralement pour préserver la confidentialité des informations client.

### Comment fonctionnent les certificats SSL ?

La sécurisation SSL fonctionne par un échange de clefs entre le client et le serveur d'application:

1. Le client se connecte au site sécurisé par SSL avec son navigateur en lui envoyant une demande d'authentification et en lui indiquant les systèmes de cryptage supportés par le navigateur.
2. Le serveur renvoie un certificat au navigateur avec la clef publique , le signature de l'autorité de certification ainsi que la longueur de cryptage maximale compatible entre les deux outils.
3. Le navigateur vérifie la validité du certificat et génère une clef d'échange à l'aide de la clef publique du serveur et lui renvoie.
4. Le serveur chiffre ensuite toutes les transactions avec le client avec cette clef unique garantissant ainsi la confidentialité des données.

- **Générer et installer son certificat SSL**

- Vérifier si la librairie openssl est installée : **# openssl version** si ce n'est pas le cas **# sudo apt-get install openssl**
- se placer dans le répertoire ssl : **# cd /etc/ssl**
- créer la clé privée du serveur : **# sudo openssl genrsa -out server.key 2048**
- générer un fichier de « demande de signature de certificat » ou CSR (Certificate Signing Request) : **# sudo openssl req -new -key server.key -out server.csr**

CSR : requête initiée par le client au certificat d'autorité qui contient toutes les informations nécessaires comme le nom de domaine ou les informations sur le site marchand, cela contient également la clé publique qui sera signée par le certificat d'autorité et qui sera retourné au client

- Visualiser le contenu du fichier généré : **# openssl req -text -noout -in server.csr**
- générez ou récupérez le certificat signé au format x509, certificat auto-signé pour 365 jours (1 an) : **# sudo openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt**

Remarque : ce certificat n'est authentifié par aucune autorité

### Installer le certificat ssl sur un serveur apache

- se placer dans le bon répertoire : **# cd /home/user**
- activer le module SSL d'Apache : **# a2enmod ssl**
- redémarrer apache : **# service apache2 restart**
- **# cp /home/user/Téléchargements/HTTPCS65577.html /var/www/html/api/HTTPCS65577.html**

## Job 10 - Déploiement de l'application mobile

## Job 11 - Diagramme de déploiement

