

Installer un serveur Mumble

et des fonctionnalités utilisables par des débutants

grâce à un accès VNC et des outils courants

sur un PC ou serveur Debian 7 ou 8

par antislash (2017)

Inconvénient : cet exploit ne peut se concevoir qu'au détriment des ressources (mémoire, processeur, espace disque et bande passante). Un accès graphique par X2GO (voir les annexes facultatives) serait plus indiqué mais moins simple.

Si vous n'avez pas de serveur pour héberger votre Mumble, le meilleur rapport qualité/prix que j'aie pu trouvé en janvier 2017 est celui à 4,69€/mois (incluant une sauvegarde Snapshot efficace et bien d'autres avantages) chez Hetzner.de. On peut utiliser ce serveur pendant 6 jours avant de payer :

https://www.hetzner.de/hosting/produkte_vserver/cx10

Malheureusement Hetzner.de n'assure pas d'assistance téléphonique en Français, si cela s'avère primordial pour l'utilisateur final, voyez donc l'offre VPS 1 SSD chez OVH à 2,99€HT (3,59TTC), seul avantage : on peut y louer une Debian 8 en CozyCloud (=préconfigurations utiles aux bots etc).

<https://www.ovh.com/fr/order/vps/#/legacy/vps/options?>

[vps=~~\(category~~'ssd~~product~~'vps_ssd_model1~~os~~'linux'\)](https://www.ovh.com/fr/order/vps/#/legacy/vps/options?vps=~~(category~~'ssd~~product~~'vps_ssd_model1~~os~~'linux'))

L'orientation du présent guide s'est érigée sur l'expérience et la recherche de 4 principes fondateurs :

- 1) **Respecter la privacité d'autrui.**
- 2) **Réfléchir avant de presser une touche.**
- 3) **En accroissant ses capacités on accroît ses responsabilités.**
- 4) **En partageant le savoir-faire conjointement à l'administration des outils, on s'apprend réciproquement comment mieux les ouvrir aux participants, mieux déployer leurs capacités.**

Avertissement : Il y aura ici plusieurs identifiants et mot de passe à noter sur bloc-notes ou, mieux, sur un papier A4, ou dans un fichier texte encrypté en PGP. Ces identifiants et mots de passe sont liés à différentes fonctionnalités et aux modes administrateur (root) ou utilisateur...

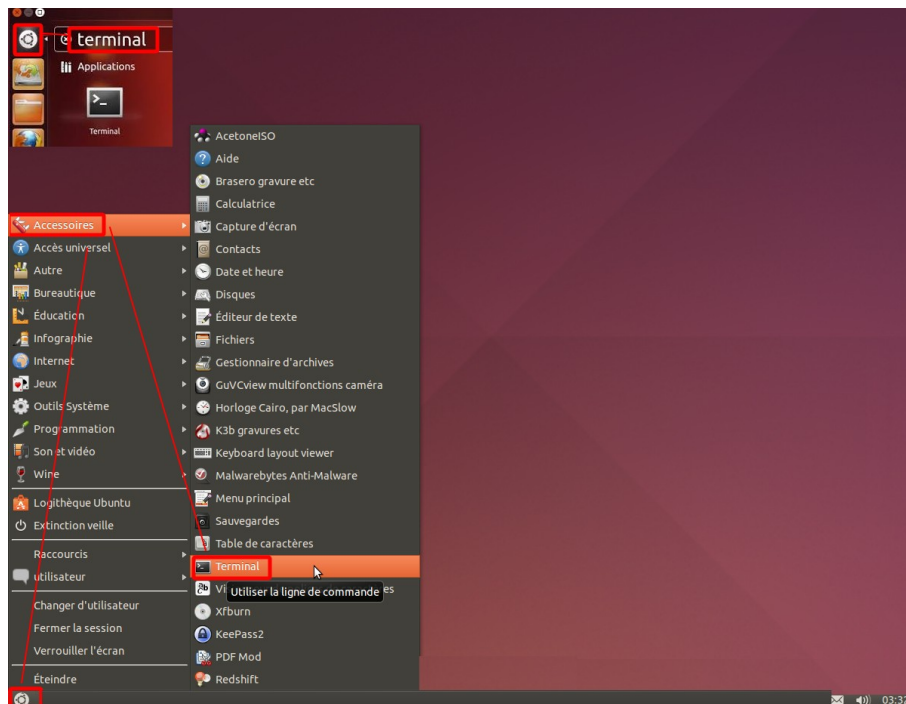
Premier geste : modifier le mot de passe root

(celui qui permet d'administrer le serveur par Terminal en SSH)

Munissez-vous de votre nom de domaine ou de l'adresse IP indiquée dans le mail de confirmation de votre fournisseur de serveur distant, notez aussi le mot de passe qui l'accompagne.

Vous devez ouvrir un terminal de commandes : la fenêtre noire avec des lettrages blancs

→ Sous Linux la fonction SSH est pré-installée...



→ mais sous Windows il faut installer Putty :

<http://www.commentcamarche.net/download/telecharger-90-putty>



Ouvrez donc un terminal dans Linux ou Putty dans Windows, et recopiez-y la commande suivante en remplaçant d'abord les XXX.XX par votre nom de domaine ou par les numéros d'adresse IP du serveur :

ssh root@XXX.XX.XXX.XX

Notez bien cette première commande, elle servira souvent, pour vous connecter au serveur en mode Root (administrateur) par un terminal en SSH, ça ressemble à ça :

```
nomdUtilisateur@nomdOrdinateur:~$ ssh root@nomdedomaine.ou.adresse.I.P
.du.serveur
```

La première fois vous devrez écrire **yes** puis presser la touche Entrée avant que ne vous soit demandé le mot de passe (qu'on ne voit pas s'afficher quand on l'écrit) puis enfin vous serez «Root»

```
Are you sure you want to continue connecting (yes/no)? yes
root@nomdedomaine.ou.adresse.I.P
.du.serveur's password: écrire le mot de passe (c'est en aveugle)
root@serveurdistant:~#
```

Pour rappel insistant, créez une note ou prévoyez un grand papier, car différents identifiants et mots de passe vont se cumuler tout au long de ce parcours.

En permanence dans ce guide, vous serez invité.e.s à entrer des commandes dans le Terminal, c'est à dire à écrire (ou coller, ou glisser) ces commandes dans la fenêtre noire puis presser Entrée.

Alors pour celles et ceux qui ne le sauraient pas, oui, sous Linux on peut glisser des textes avec la souris directement dans le Terminal. Par ailleurs, sous Linux comme sous Windows, voilà les principaux raccourcis clavier qu'on peut utiliser :

Ctrl C : copier **Ctrl V** : coller **Alt Tab** : basculer d'une fenêtre (ce guide) à une autre (le terminal)

D'abord pour définir un nouveau mot de passe de votre choix pour la connexion en SSH :
passwd

Il vous sera demandé d'écrire votre nouveau mot de passe (en aveugle) puis presser Entrée, puis il sera demandé de le réécrire une seconde fois (puis presser Entrée) pour confirmation.

Il faut ensuite actualiser la liste des paquets préinstallés dans votre système :
apt-get update

si vous voyez une erreur pointant « nodesource.list.ucf-dist » ne vous en préoccupez pas pour l'instant c'est sans incidence, vous y remédieriez plus tard si vous installez des bots en node.js.

Choisir le français comme langue utilisée par votre système Debian

Juste pour info, si vous souhaitez vérifier quelle est la langue déjà utilisée par votre système Debian, entrez la commande :

```
env | grep LANG
```

**Comme le résultat ne sera pas « Français » mais « English »,
d'abord entrez la commande de bascule vers le Français en UTF-8**

```
export LANG=fr_FR.UTF-8
```

Là, si un avis d'erreur DebConf (...)NonInteractive s'affiche, entrez cela :

```
dpkg-reconfigure debconf
```

puis acceptez tous les choix proposés par défauts avec la touche Entrée

**ensuite commandez à Debian la reconfiguration ses ressources
locales en français :**

```
dpkg-reconfigure locales
```

Là, une fenêtre vous permettra faire défiler les langues avec les flèches du clavier puis de cocher «French, France, Métropolitaine» ou «**fr_FR.UTF-8**» avec la barre Espace (la touche Entrée sert à valider et la touche Tab à se déplacer parmi les choix de la fenêtre).

Les changements ne s'appliqueront qu'au redémarrage, poursuivons d'abord quelques étapes en anglais.

Pour que les applications KDE soient elles aussi en Français, entrez la commande :

```
apt-get install kde-l10n-fr
```

juste pour info, voici comment vérifier la syntaxe exacte pour les langues disponibles :

```
apt-cache search kde-l10n
```

OK, au prochain démarrage, les applications et le système seront en français !

Mais, sans attendre, on va maintenant accomplir les installations requises...

Installer une interface graphique sur Debian 7 or 8

Infos et installations préliminaires

Il existe plusieurs interfaces graphiques (dites Bureau ou GUIs) au choix. Gnome, KDE, LXDE ... ici c'est Xfce qui a été choisie pour son équilibre entre légèreté, efficacité, simplicité d'approche pour les débutants windowsiens...

Pour installer donc xfce et du même coup presque tous les logiciels utiles à un serveur mumble enrichi, entrer la commande suivante puis validez les demandes qui s'afficheront et patientez car l'accomplissement de cette commande est de loin le plus long, ici :

```
apt-get install xorg xfce4 xfce4-goodies thunar-archive-plugin synaptic htop gdebi wicd  
iceweasel pavucontrol gedit nautilus mumble mumble-server wicd-cli wicd-curses vinagre  
filezilla audacious audacity nicotine chromium xchat ekiga keepass
```

...le terminal vous demandera de confirmer en pressant les touches o puis Entrée... puis patientez.

Une fois ces préliminaires accomplis, venons-en à l'installation de l'interface graphique VNC

Il existe diverses solutions mais celle qui inflige le moins d'incidents de mise en route aux débutants n'est malheureusement pas la plus légère, mais seulement raisonnable : vnc4server

Entrez la commande ci-après, il s'en suivra plusieurs étapes de paramétrages « utilisateur » :

```
apt-get install vnc4server
```

En mode ROOT (administrateur) vous avez tous les droits de modification, même les plus dangereux. Pour éviter des accidents aux utilisateurs, il convient donc de créer un simple compte d'utilisateur aux droits limités, comme sur Windows ou Mac.

Ici pour plus de lisibilité, ce compte a été nommé **TONPSEUDO**

C'est le nom qu'il vous faudra remplacer tout au long de ces instructions par un nom de votre choix, de préférence court, en minuscules, sans espaces ni caractères spéciaux, ni accents...

Entrez donc la commande suivante en l'adaptant :

```
adduser TONPSEUDO
```

vous serez alors invité.e à **créer un mot de passe** pour ses droits limités d'accès au système (vous serez invités aussi à confirmer en écrivant le mot de passe une seconde fois), **notez-le par ailleurs.**

SI CRÉER LE MOT DE PASSE N'A PAS ETE DEMANDE, OU POUR LE MODIFIER :

```
passwd TONPSEUDO
```

Nous sortons maintenant du mode Root (administrateur) et nous entrons dans le mode de l'utilisateur TONPSEUDO pour configurer son VNC à lui (et non pas le VNC du Root) :

su – TONPSEUDO

Au cours du premier démarrage du service VNC vous serez invité.e à créer un mot de passe spécifique qui servira à toute personne utilisant un logiciel VNC pour entrer dans votre serveur Debian (comme si elles accédaient à n'importe quel ordi distant),

démarrons donc VNC :

vncserver

Attention : choisissez un mot de passe entre 6 et 8 caractères (obligatoire) !

En effet suite à cette commande, il vous est demandé d'inscrire un mot de passe VNC de TONPSEUDO (puis de le re-saisir pour confirmer), ce mot de passe peut différer (:ou non:) de celui de l'utilisateur TONPSEUDO précédemment choisi, et pourra facilement être modifié à tout moment avec la commande : **vncpasswd**

Ok, VNC a généré ses fichiers d'utilisateur, pour les paramétrer nous devons désactiver VNC

vncserver -kill :1

Un message de confirmation s'affiche

C'est là que vous entreprendrez la première manipulation un peu plus poussée de ce guide.

Il s'agit juste de copier du texte dans un fichier vide, mais en passant par le terminal.

Ce fichier ordonne au système de démarrer en mode graphique avec VNC, et non plus seulement comme un serveur pour experts obligeant à manipuler le terminal. D'abord on ouvre ledit fichier avec un éditeur de texte (nano), on y colle le texte copié, on enregistre et valide la modification faite, et on sort de ce mode édition de texte pour revenir au Terminal. Voilà comment procéder :

cd ~

> .vnc/xstartup

nano .vnc/xstartup

L'aspect du Terminal change, vous êtes dans **nano**, suivez bien les instructions de la page suivante...

Le fichier vierge *xstartup* s'est ouvert dans nano, recopiez-y (ou glissez-y) ce texte ci-après :

```
#!/bin/sh
```

```
unset SESSION_MANAGER
```

```
unset DBUS_SESSION_BUS_ADDRESS
```

```
startxfce4 &
```

```
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
```

```
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
```

```
xsetroot -solid grey
```

```
vncconfig -iconic &
```

Puis, pour appliquer cette modification de *xstartup*, utiliser la combinaison des touches **Ctrl o**

Ensuite, il sera demandé de valider le nom du fichier à écrire en pressant la touche **Entrée**

Enfin, pour sortir de ce mode édition nano, presser la combinaison des touches **Ctrl x**

Retour au mode Root

Le fichier de démarrage de l'utilisateur est écrit, enregistré, il nous reste à paramétrer (créer le script de lancement pour) l'accès par VNC. **Cela doit se faire en mode Root (administrateur), voilà la commande qui nous y ramène**, très souvent nécessaire, à connaître :

```
su -
```

Le mot de passe est exigé, et hop, on est de retour en mode root ! (d'ailleurs pour le signaler, le \$ en fin de ligne du terminal est redevenu un #)

Maintenant on édite un autre fichier (pour l'intégration de VNC au système) avec nano :

```
nano /etc/init.d/vncserver
```

Le fichier *vncserver* s'ouvre alors dans nano, alors par clic-droit (+coller ou paste) ou par la combinaison des touches Ctrl et V, **collez-y le très long texte ci-après** que vous aurez sélectionné ici et copié par clic-droit (+copier ou copy) ou par la combinaison des touches Ctrl et C :

```
#!/bin/bash
```

```
### BEGIN INIT INFO
```

```
# Provides:      tightvncserver
```

```
# Required-Start: $syslog
```

```
# Required-Stop:  $syslog
```

```
# Default-Start:  2 3 4 5
```

```
# Default-Stop:   0 1 6
```

```
# Short-Description: vnc server
```

```
# Description: https://fr.wikipedia.org/wiki/Mumble
```

```
#
```

```
### END INIT INFO
```

```
|
```

```
unset VNCSERVERARGS
```

```
VNCSERVERS=""
```

```
[ -f /etc/vncserver/vncservers.conf ] && . /etc/vncserver/vncservers.conf
```

```
prog="$VNC server"
```

```
start() {
```

```
. /lib/lsb/init-functions
```



```
REQ_USER=$2
```

```
echo -n $"Starting $prog: "
```

```
ulimit -S -c 0 >/dev/null 2>&1
```

```
RETVAL=0
```

```
for display in ${VNCSERVERS}
```

```
do
```

```
export USER="${display##*:}"
```

```
if test -z "${REQ_USER}" -o "${REQ_USER}" == ${USER} ; then
```

```
echo -n "${display} "
```

```
unset BASH_ENV ENV
```

```
DISP="${display%%:*}"
```

```
export VNCUSERARGS="${VNCSERVERARGS[${DISP}]}"
```

```
su ${USER} -c "cd ~${USER} && [ -f .vnc/passwd ] && vncserver :${DISP} ${VNCUSERARGS}"
```

```
fi
```

```
done
```

```
}
```

```
stop() {
```

```
./lib/lsb/init-functions
```

```
REQ_USER=$2
```

```
echo -n $"Shutting down VNCServer: "
```

```
for display in ${VNCSERVERS}
```

```
do
```

```
export USER="${display##*:}"
```

```
if test -z "${REQ_USER}" -o "${REQ_USER}" == "${USER}" ; then
```

```
echo -n "${display} "
```

```
unset BASH_ENV ENV
```

```
export USER="${display##*:}"
```

```
su ${USER} -c "vncserver -kill :${display%%:*}" >/dev/null 2>&1
```

```
fi
```

```
done
```

```
echo -e "\n"
```

```
echo "VNCServer Stopped"
```

```
}
```

```
case "$1" in
```

```
start)
```

```
start $@
```

```
;;
```

```
stop)
```

```
stop $@
```

```
;;
```

```
restart|reload)
```

```
stop $@
```

```
sleep 3
```

```
start $@
```

```
;;
```

```
condrestart)
```

```
if [ -f /var/lock/subsys/vncserver ]; then
```

```
stop $@
```

```
sleep 3
```

```
start $@
```

```
fi
```

```
;;
```

```
status)
```

```
status Xvnc
```

```
::
```

```
*)
```

```
echo $"Usage: $0 {start|stop|restart|condrestart|status}"
```

```
exit 1
```

```
esac
```

une fois collé ce texte (bravo), pour appliquer cette modification du fichier vncserver, presser la combinaison des touches **Ctrl o**

ensuite, il sera demandé de valider le nom du fichier à écrire en pressant la touche **Entrée**

enfin, pour sortir de ce mode édition nano, presser la combinaison des touches **Ctrl x**

Cela fait, rendons « exécutable » ce script configurant l'intégration de VNC

```
chmod +x /etc/init.d/vncserver
```

(c'est un *marquage* du fichier pour que le système le considère *un peu comme un logiciel*)

L'étape suivante amène à indiquer vos réglages matériels pour le lancement VNC

D'abord on crée un dossier nommé vncserver dans le dossier etc :

```
mkdir -p /etc/vncserver
```

puis on crée le fichier de configuration matérielle de VNC :

```
nano /etc/vncserver/vncservers.conf
```

Dans l'éditeur nano qui s'ouvre à nouveau il faut coller le texte ci-après, mais ATTENTION À BIEN Y REMPLACER LA MENTION "TONPSEUDO" PAR LE NOM D'UTILISATEUR CHOISI PRECEDEMMENT, et vous pouvez en profiter pour ajuster la résolution 1024x768 à la taille de l'écran des utilisateurs pour qui vous accomplissez tout ce travail (si vous la connaissez).

Après avoir remplacé ci-bas le pseudo et éventuellement ajusté la résolution, **sélectionner ce court texte puis copiez-le par clic-droit +copier** (ou copy) ou par la combinaison des touches Ctrl et C **puis collez-le par clic-droit +coller** (ou paste) ou par la combinaison des touches Ctrl et V :

```
VNCSERVERS="1:TONPSEUDO"
```

```
VNCSERVERARGS[1]="-geometry 1024x768"
```

une fois collé ce texte, pour appliquer cette modification du fichier vncservers.conf, utiliser la combinaisons des touches **Ctrl o**

ensuite, il sera demandé de valider le nom du fichier à écrire en pressant la touche **Entrée**

enfin, pour sortir de ce mode édition nano, presser la combinaisons des touches **Ctrl x**

Remarque : si vous ajoutez d'autres utilisateurs, le port VNC ici réglé à 1 (devant le pseudo dans la première ligne) il faudra le faire passer à 2 et ainsi de suite pour chaque nouvel utilisateur, chacun aura ainsi son canal, avec son pseudo et sa propre résolution ajustable. De fait, pour se connecter par VNC, l'adresse du serveur qu'il faudra leur communiquer ne se finira plus par 5901 mais par 5902, ou 5903 etc (cette question s'éclairera un peu plus bas, on y arrive, patience).

La dernière étape de préparation de VNC consiste à **injecter ce lancement de VNC dans la séquence de démarrage automatique des processus à l'allumage du système**

```
update-rc.d vncserver defaults 99
```

Vous verrez sans doute un avertissement concernant l'absence de LSB, à ignorer, on s'en fiche.

On redémarre le système (extinction+rallumage), là on va voir si tout ça a bien marché :

```
reboot
```

Accès à l'interface graphique par VNC depuis votre ordi

Vous pouvez maintenant accéder graphiquement à votre serveur distant depuis votre ordinateur, si vous êtes sous windows, utilisez par exemple **PUTTY** (et débrouillez-vous pour trouver son guide si besoin), et si vous êtes sous linux, notamment sous Ubuntu ou autre Debian, c'est sans doute le *visionneur de bureaux distants* nommé **Vinagre** qui vous simplifiera le mieux cette tâche.

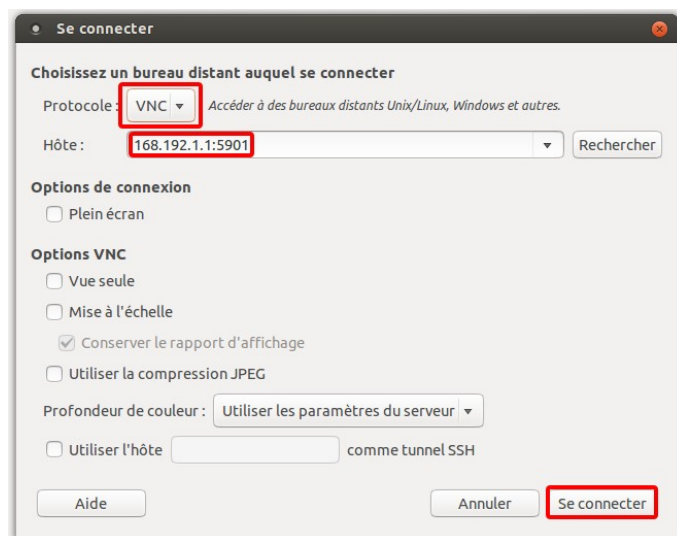
Petit soucis : le lanceur de Vinagre se loge dans votre menu des logiciels, section internet, sous le nom générique de *visionneur de bureaux distants*, or **il y en a peut-être déjà un du même nom !**

Celui de Vinagre est plus bas que l'autre dans la même liste. Mais on peut aussi **lancer Vinagre depuis le terminal** (ou par l'astucieuse combinaison Alt F2)... **entrer simplement :**

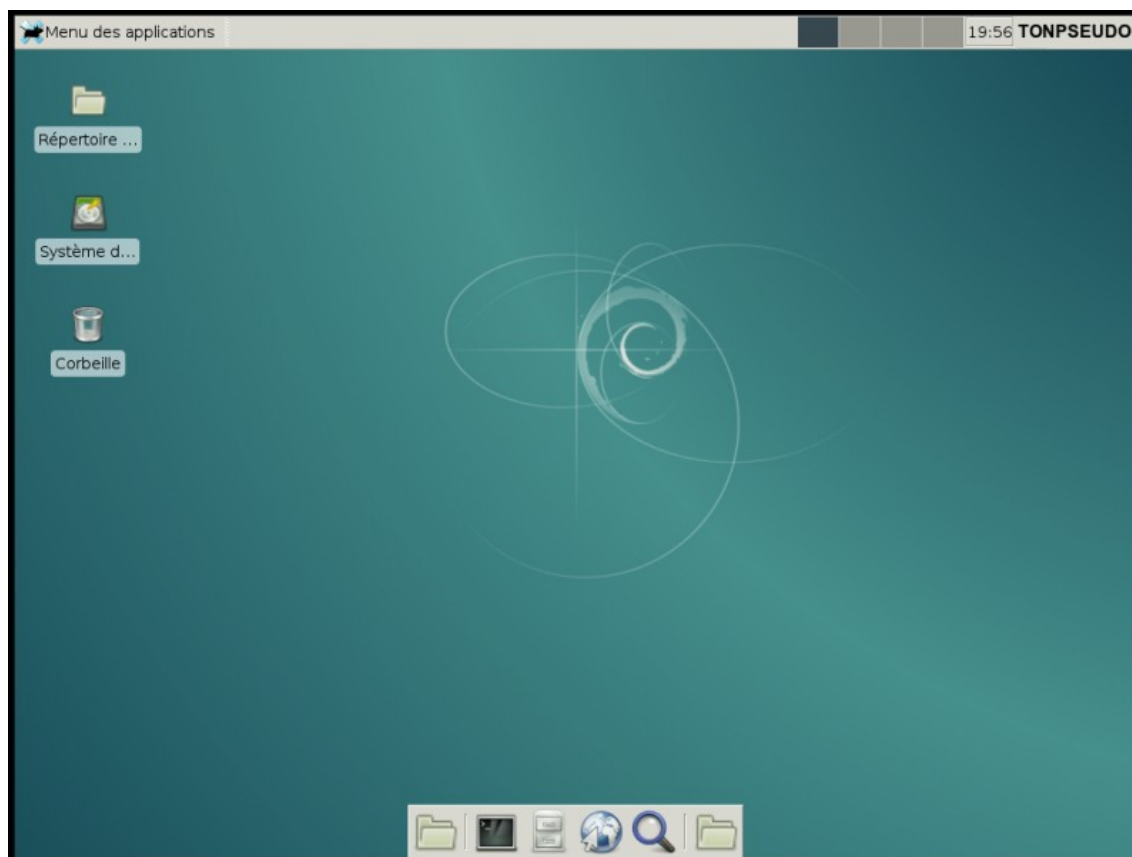
```
vinagre
```

Une fois dans la fenêtre ouverte de Vinagre

choisir le protocole VNC au lieu de SSH dans son menu déroulant, inscrire l'adresse IP du serveur distant (dans cet exemple c'est 168.192.1.1, **remplacez-la !**) puis **ajoutez le port associé** (qui par défaut est le :5901 pour le premier utilisateur) pour l'exemple ainsi : **168.192.1.1:5901**



cliquer sur le bouton «Se connecter» et hop voilà, vous y êtes enfin ! Vinagre ouvre le bureau du serveur distant dans une fenêtre ! Vous pouvez y cliquer comme dans un ordinateur !



Rappels : le mot de passe pour l'accès par VNC pourra facilement être modifié à tout moment avec la commande **vncpasswd** . Veillez à choisir un mot de passe entre 6 à 8 caractères (obligatoire) !



Quelques astuces de dépannages couramment demandées

Si l'accès par VNC ne fonctionne plus, il suffit souvent de le désactiver puis le réactiver.

▫ Pour désactiver l'accès VNC précédemment établi pour *TONPSEUDO* sur le canal :1 se connecter par SSH grâce à l'accès Root (voir le début de ce guide) puis basculer dans le terminal en tant que simple utilisateur *TONPSEUDO* :

```
su - TONPSEUDO
```

(ci-dessus remplacer *TONPSEUDO* par le nom d'utilisateur)

puis entrez la commande de désactivation du VNC

```
vnc4server -kill :1
```

▫ Enfin pour réactiver l'accès pour *TONPSEUDO* par VNC toujours sur le même canal :1, entrez la commande suivante :

```
vnc4server :1 -geometry 1024x768 -depth 16 -httpport 5900
```

SI VOUS NE POUVEZ TOUJOURS PAS ACCEDER PAR VNC A VOTRE SERVEUR, assurez-vous que VNC soit autorisé dans votre pare-feu *ip-tables* en entrant une après l'autre (séparément) les trois commandes suivantes :

```
iptables -L
```

```
iptables -A INPUT -m state --state NEW -m tcp -p tcp -m multiport --dports 5901:5903,6001:6003 -j ACCEPT
```

```
iptables-save
```


Si vous aviez une clé SSH particulière pré-intégrée et qu'elle ne fonctionne plus, l'accès VNC lui, devrait fonctionner. Alors connectez-vous via VNC, ouvrez visuellement un Terminal (vous trouverez facilement son lanceur au bas de l'écran, reconnaissable par son icône de mini-fenêtre noire) puis entrez-y la commande suivante :

```
sudo /etc/init.d/ssh restart
```

le mot de passe Root est exigé, ça y est, vous pouvez tenter à nouveau de vous connecter par SSH.

```
ssh root@XXX.XX.XXX.XX
```

(en remplaçant les X par votre nom de domaine ou par l'adresse IP du serveur)

Si vous êtes confronté au message d'erreur "access denied error from DBus. Check your user is in netdev group" :

d'abord vérifier voir si le compte d'utilisateur TONPSEUDO est bien référencé dans les groupes

du système, entrer la commande :

```
groups
```

s'il y a un soucis, entrer successivement (séparément) les trois commandes suivantes, comme toujours en remplaçant la mention TONPSEUDO par le nom d'utilisateur choisi :

```
gpsswd -a TONPSEUDO netdev
```

```
wicd-client
```

```
service dbus reload
```

Puis redémarrer le serveur (la machine distante)

```
reboot
```

Si vous êtes toujours confronté au message d'erreur "access denied error from DBus. Check your user is in netdev group", cette fois-ci entrer successivement (séparément) les trois commandes suivantes (la première est différente), comme toujours en remplaçant la mention TONPSEUDO par le nom d'utilisateur choisi :

```
adduser TONPSEUDO netdev
```

wicd-client

service dbus reload

Puis redémarrer le serveur (la machine distante)

reboot

Là ça devrait marcher, sinon faites une réclamation à la personne qui vous a filé ce guide ;)

Vous trouverez quelques
autres infos de dépannage
dans les discussions d'un forum copiées
en toute dernière partie de ce guide

Pour créer un accès limité à de l'administration ou de la vérification

Dans l'exemple de gestion collective proposé ici, l'organigramme présente 4 faces :

d'un côté les personnes qui utiliseront les identifiants de TONPSEUDO seront habilitées à modifier le message d'accueil du serveur Mumble, en plus de pouvoir utiliser les logiciels mis à disposition, les dossiers d'utilisateur (documents images musique téléchargements) et elles pourront tout explorer dans le serveur (donc valider ou faire valider que la privacité est bien respectée). Elles pourront redémarrer le service Mumble si besoin.

De l'autre côté les personnes qui auront le mot de passe du mode Root, elles, auront l'accès aux contenus du serveur donc le droit d'installer des logiciels ou de modifier les éléments du système. Elles pourront redémarrer n'importe quel service ou la machine elle-même. Par malveillance, par inconscience ou maladresse, leurs gestes peuvent induire des modifications qui ne respectent pas la privacité.

Sur un autre plan, la ou les personnes qui ont l'accès aux serveur lui-même (la machine) pourront le redémarrer s'il était planté, tout effacer et tout réinstaller, le mettre hors-service ou veiller à ce qu'il reste en service (payer les mensualités), modifier les principaux mots de passe, son adresse IP... normalement elles ne font jamais rien mais ont les boutons de contrôle supérieurs, il convient que ce rôle tourne entre personnes qui construisent des ententes confiantes.

Sur un dernier plan, l'aiguillage central dépend de la personne qui a la main sur la redirection du nom de domaine, c'est à dire de l'adresse connue et utilisée par les participants. C'est pourquoi un outil en ligne ou site web peut afficher plusieurs adresses possibles, et ainsi permettre une garantie aux participants que ce ne sont pas les mêmes personnes qui détiennent toutes les clés vitales.

Habiller l'utilisateur TONPSEUDO à bidouiller Mumble

Puisqu'on a créé un utilisateur ayant un accès par VNC, commençons par lui donner le droit de modifier le message d'accueil du serveur Mumble, de relancer le service Mumble si besoin, mais aussi d'extraire (d'exporter) des sauvegardes ou en rapatrier.

Ainsi toute personne qui aura les informations de connexion de cet utilisateur pourra aussi faire basculer toute l'architecture du Mumble d'une formule à une autre (préfabriquées) sans aucune autre aide technique que le court texte fourni en annexe de ce guide.

D'abord, si ce n'est pas déjà le cas, il se connecter au serveur par terminal en tant que root :

Dans votre ordinateur sous Linux, ouvrez un terminal (ou Putty dans Windows) et inscrivez-y la commande suivante en remplaçant les XXX.XX par votre nom de domaine ou par les

numéros d'adresse IP du serveur :

```
ssh root@XXX.XX.XXX.XX
```

Puis entrer successivement chacune de ces trois commandes suivantes en y remplaçant TONPSEUDO par le nom d'utilisateur créé/choisi lors des précédentes manipulations :

```
sudo chown TONPSEUDO /etc/mumble-server.ini
```

```
sudo chown TONPSEUDO /var/lib/mumble-server/mumble-server.sqlite
```

```
sudo chown TONPSEUDO /var/run/mumble-server/mumble-server.pid
```

Ensuite il faudra éditer manuellement le registre des sudoers (ayant-droits),
après l'avoir sauvegardé

```
sudo cp /etc/sudoers
```

```
/etc/sudoers
```

```
.bkp
```

```
sudo nano /etc/sudoers
```

Attention, allez-y soigneusement : le cadre qui s'ouvre contient déjà beaucoup d'informations précieuses pour le système, une mauvaise manip' pourrait vous obliger à tout réinstaller.

En utilisant les flèches du clavier vers le haut et vers le bas, faites défiler le texte, positionnez-vous très précisément sous la rubrique **#Allow members of group sudo to execute any command**
%sudo ALL=(ALL:ALL) ALL

Là en dessous à la ligne, pressez une fois la touche Entrée pour sauter une ligne (juste pour rendre l'ajout plus lisible) et collez les deux lignes suivantes après y avoir remplacé TONPSEUDO :

```
# Administration du serveur Mumble
```

TONPSEUDO ALL=/etc/init.d/mumble-server

puis, pour appliquer cette modification, presser la combinaison simultanée des touches

Ctrl o

puis presser la touche "Entrée" pour confirmer le nom du fichier à écrire (la question sera posée, mais pour annuler, voyez comme le menu du bas propose aussi la combinaison des touches **Ctrl c**).

Enfin, pour sortir de nano, utiliser la combinaisons des touches

Ctrl x

Note : dans ce langage, les lignes qui commencent par un # ne sont que des "commentaires facultatifs" = une info adressée aux personnes qui visitent ce fichier système

, on peut y écrire ce qu'on veut.

Voilà à quoi ressemblerait cette partie du fichier modifié :

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:
snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# Administration du serveur Mumble
TONPSEUDO ALL=/etc/init.d/mumble-server

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d|
```

Votre serveur Mumble

Sauvegarde de la configuration

Il suffit de 2 fichiers précis qui contiennent l'ensemble des propriétés nécessaires à la sauvegarde et au rapatriement d'un mumble existant : `/etc/default/mumble-server.ini` et `/var/lib/mumble-server.sqlite`. Ils sont très légers. Par précaution donc, avant toute manipulation, on crée séparément une copie de chacun de ces deux fichiers dans leur dossiers respectifs :

```
sudo cp /etc/default/mumble-server.ini /etc/default/mumble-server.ini-original.bkp
```

```
sudo cp /var/lib/mumble-server.sqlite /var/lib/mumble-server.sqlite-original.bkp
```

On peut ouvrir ces fichiers (dans un éditeur de texte simple) et en modifier le contenu, les enregistrer dans d'autres dossiers d'archives personnelles, se les envoyer par exemple encrypté en PGP avec le logiciel Keepass (fourni) et les conserver bien à l'abri.

Info : si vous rapatriez un mumble existant depuis un autre serveur, pas besoin d'éditer manuellement les configurations de mumble, si vous créez un nouveau mumble, passez ce chapitre **Rapatriement** pour aller directement à la **Configuration de votre nouveau serveur Mumble**

Rapatriement d'une sauvegarde ou d'une config modifiée

Accomplissez d'abord la sauvegarde locale conseillée ci-dessus.

Ensuite, si vous souhaitez rapatrier d'autres fichiers de sauvegarde qui viennent d'un autre dossier situé dans le même serveur, c'est simple, remplacer seulement la mention `/adresse/dudit/dossier/` de chaque ligne de commandes dans cet exemple :

```
sudo cp /adresse/dudit/dossier/mumble-server.ini /etc/default/mumble-server.ini
```

```
sudo cp /adresse/dudit/dossier/mumble-server.sqlite /var/lib/mumble-server.sqlite
```

Par contre, si la sauvegarde qu'on veut rapatrier vient d'un autre serveur ou ordi connecté à distance, d'abord il faut bien s'assurer d'être connecté en mode Root de votre serveur à vous. Soit vous y êtes entré graphiquement par VNC et y avez ouvert un terminal que vous avez passé en mode root (`su -`) soit vous y êtes connecté.e par un terminal qui indique à chaque

début de ligne, mais si ce n'est pas le cas, ou si vous n'êtes pas sûr.e, voilà une petite piqure de rappel : **Sous Linux, ouvrez un terminal (ou Putty dans Windows) et inscrivez-y la commande suivante en remplaçant les XX.XXX par votre nom de domaine ou par les numéros d'adresse IP de votre serveur (indiqués dans le mail de confirmation de vos fournisseurs) :**

```
ssh root@XXX.XX.XXX.XX
```

Ensuite se munir des identifiants (nom d'utilisateur, mot de passe) **du serveur distant** existant (ou l'ordinateur connecté) **d'où l'on tire la configuration souhaitée**. Dans les 2 commandes ci-dessous (à entrer successivement, séparément) il faudra **remplacer NOMDISTANT** par ledit nom d'utilisateur _le mot de passe sera exigé_ **puis y remplacer cette fois les XX.XXX par l'adresse IP ou le nom de domaine du serveur (ou ordinateur) d'où l'on tire la configuration souhaitée :**

```
scp /etc/default/mumble-server.ini NOMDISTANT@XX.XXX.XX.XXX :/etc/default/mumble-server.ini
```

```
scp /var/lib/mumble-server.sqlite NOMDISTANT@XX.XXX.XX.XXX :/etc/default/mumble-server.ini
```

Info : si vous rapatriez un mumble existant depuis un autre serveur, pas besoin d'éditer manuellement toutes les configurations de mumble, mais sans doute quand même quatre :

son message d'accueil, son nom visible ou pas publiquement, son site web associé, le nom du dossier racine tout en haut où arrivent les visiteurs... Voyez donc la suite ci-bas.

Configuration de votre nouveau serveur Mumble

Pour commencer

Pour « initialiser la configuration », le moyen le plus simple est de « reconfigurer le paquet ». Avis : notez bien cette commande fondamentale, si plus tard vous souhaitez réinitialiser mumble-server, ou s'il tombe en panne. Après l'avoir lancée, des choix s'afficheront dans une fenêtre interactive.

```
sudo dpkg-reconfigure mumble-server
```

Dans ce genre de fenêtre interactive du terminal, on peut se déplacer avec les flèches du clavier, la touche Tab aussi à se déplacer parmi les rubriques, choix ou menus, on peut cocher avec la barre Espace, et la touche Entrée sert à valider.

Vous serez amené à y accepter les 2 choix proposés en pressant la touche Entrée (démarrage automatique du serveur mumble à chaque allumage ; priorité élevée) mais surtout à créer le mot de passe du compte **SuperUser**, notez-le bien, gardez-le à l'abri, il sera important par la suite.

On peut aussi éditer les configurations de mumble à la main (modifier le texte qui les rassemble), c'est plus complet, plus direct, par exemple avec nano

```
nano /etc/default/mumble-server.ini
```

ou bien graphiquement avec gedit

```
gedit /etc/default/mumble-server.ini
```

Explorez et modifiez les options desquelles vous connaissez le fonctionnement et les paramètres afférents (voir les tutos sur internet, wiki...). Les principales personnalisations sont le message d'accueil et la rubrique incluant le nom du salon racine (lui aussi nommé root avant personnalisation) et la diffusion ou non dans la liste publique des mumbles.

Il vous sera par exemple possible de chercher (Ctrl F) les occurrences suivantes par exemple pour :

- Modifier le message d'accueil entre les guillemets qui suivent la mention **welcometext=** :

```
welcometext="<br />Welcome to this server running <b>Murmur</b>.<br />Enjoy your stay!<br />"
```

- qui deviendrait quelque chose comme :

```
welcometext="<br /> <br />
Bienvenue sur le serveur vocal NomDeMonProjet <br />Au bas de cette colonne de
gauche, on peut écrire aux personnes présentes, notamment pour y demander la
parole ou faire des commentaires en aparté.<br />Bon séjour !<br />"
```

- Renommer le salon racine en enlevant un # et ajoutant (ou remplaçant) le registerName :

```
#registerName=
```

- qui deviendrait quelque chose comme :

```
registerName=NomChoisi
```

- Faire apparaître votre serveur Mumble dans la liste publique en enlevant trois # et en renseignant :

```
#registerPassword=secret
```

```
#registerUrl=
```

```
#registerHostname=
```

- qui deviendraient quelque chose comme ça (ne touchez pas la mention « secret » si vous ne voulez pas assujettir l'enregistrement dans votre mumble à un mot de passe) :

```
registerPassword=secret
```

```
registerUrl=http://votrenomdedomaine-ouautresiteweb.fr
```

```
registerHostname=nompublic
```

- ou Modifier les ports et UDP pour surmonter des problèmes techniques, avoir une adresse spéciale :

```
# Port to bind TCP and UDP sockets to
port=64738
```

- Assujettir l'entrée du mumble à un mot de passe principal exigé à chaque connexion :

```
# Password to join server
serverpassword=votremotdepasse
```

- Choisir le maximum d'utilisateurs simultanés en fonction de votre bande passante ($\simeq 100$ à 500) :

```
# Maximum number of concurrent client allowed.
users=200
```

Pour beaucoup d'utilisateurs moins chevronnés, il sera bien plus pratique d'apprendre à éditer le fichier *mumble-server.ini* dans un simple éditeur de texte comme Gedit. Ce fichier ini (et le sqlite aussi, pour sauvegarde) peut être exporté puis modifié ailleurs, avant d'être rapatrié avec les commandes de la rubrique « **Rapatriement d'une sauvegarde ou d'une config modifiée** ».

Remplacez **/adresse/du/dossier/choisi** par l'adresse du dossier de votre choix :

```
sudo cp /etc/default/mumble-server.ini /adresse/du/dossier/choisi/mumble-server.ini
```

```
sudo cp var/lib/mumble-server.sqlite /adresse/du/dossier/choisi/mumble-server.sqlite /
```

Pour appliquer les modifications effectuées, redémarrez le « daemon » (service) mumble-server :

```
sudo /etc/init.d/mumble-server restart
```

Permettre la communication de Mumble si le pare-feu la bloque

Normalement votre système permet le transit des paquets et UDP entre votre serveur et les clients. Ça devrait fonctionner sans que vous n'ayez rien à faire. Si ce n'est pas le cas, entrer ces commandes

```
sudo iptables -I INPUT -p tcp --dport 64738 -j ACCEPT
sudo iptables -I INPUT -p udp --dport 64738 -j ACCEPT
```

Si vous avez précédemment défini un autre port que 64738, remplacez là aussi ce chiffre.

Puis sauvegardez les règles d'iptables

```
sudo iptables-save
```

Cas spécifique de connexion par routeur ou box nécessitant l'ouverture des ports

À quiconque étant connecté par l'intermédiaire d'un routeur, d'[une Box](#) , etc. il faudra peut être effectuer les manipulations spécifiques à ces appareils. Voyez un tuto en ligne, par exemple : https://craym.eu/tutoriels/utilitaires/ouvrir_les_ports_de_sa_box.html

INFO : démarrer éteindre ou redémarrer le service (daemon) Mumble-server

Si vous avez bien suivi les précédentes instructions, le service Mumble démarre automatiquement avec le serveur, il est censé être en permanence actif. Mais on peut vouloir l'arrêter :

```
sudo /etc/init.d/mumble-server stop
```

puis le relancer:

```
sudo /etc/init.d/mumble-server start
```

ou le redémarrer (arrêt+relance, pour appliquer des modifications du *ini*, ou parce qu'il beugue)

```
sudo /etc/init.d/mumble-server restart
```

Donner les droits d'admin à votre client Mumble

Pour débiter vous devrez vous connecter avec votre ordinateur au mumble créé sur le serveur :

Ouvrez Mumble dans votre ordinateur, dans la fenêtre qui s'ouvre en avant-plan, cliquez sur le bouton « Ajouter Nouveau... » pour inscrire le nom (libre), l'adresse exacte (l'adresse IP -ou le nom de domaine- du serveur) ainsi que Votre-Pseudo-De-Mumbleur(sans_espaces). Cela fait, cliquez sur le bouton « Connecter ». Accepter le certificat qui se propose la première fois. Enfin arrivé dans le mumble, s'il est complètement vide, faites un clic-droit sur l'icone de Votre-Pseudo-De-Mumbleur puis « S'enregistrer ».

Même si cette fonction n'est pas en fonction (grisée), tant pis, passez à l'étape suivante

Ouvrir une seconde instance Mumble, mais en tant que Superuser

Pour ouvrir plusieurs instances Mumble à la fois, la manoeuvre diffère un peu selon votre système, Linux ou Windows :

Sous Linux, entrez simplement la commande :

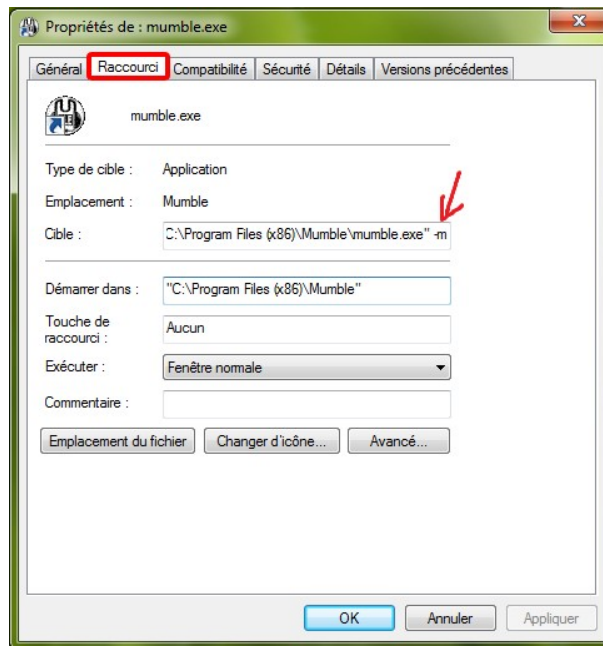
```
mumble -m
```

Sous Windows, faites une copie du raccourci de Mumble sur le Bureau, faites-y un clic droit, un menu contextuel s'ouvre, cliquez-y « Propriétés ». Dans la fenêtre des Propriétés qui s'ouvre, dans l'onglet « Raccourci » il faut modifier la cible en ajoutant un espace, le signe – et un m minuscule :

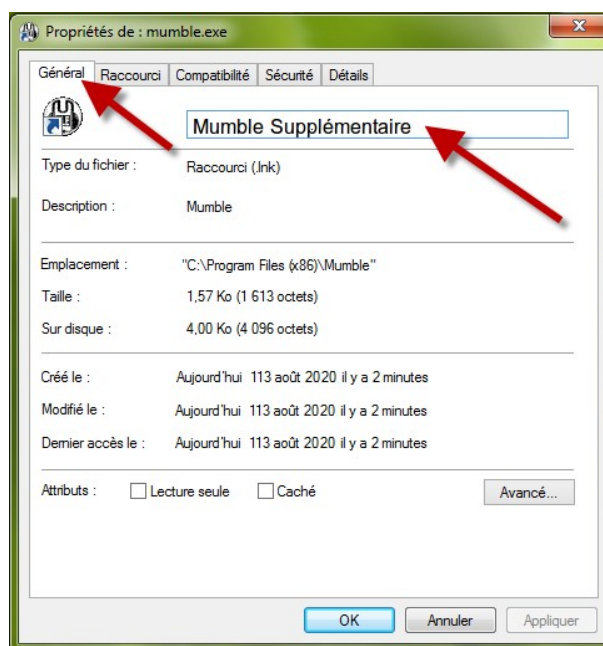
"C:\Program Files\Mumble\Mumble.exe" -m

ou

"C:\Program Files (x86)\Mumble\Mumble.exe" -m



Pour finir, aller dans l'onglet « Général » (tout en haut à gauche) et renommez le raccourci de manière éloquente, par exemple « Mumble supplémentaire ».



Dans cette seconde instance, connectez-vous en *SuperUser* :

Maintenant que vous avez ouvert une seconde instance de Mumble dans votre ordinateur, dans la fenêtre qui s'ouvre en avant-plan, cliquez sur le bouton « Ajouter Nouveau... » pour inscrire le nom SuperUser, l'adresse exacte (l'adresse IP -ou le nom de domaine- du serveur) ainsi que le pseudo-de-votre-choix(sans_espaces). Cela fait, cliquez sur le bouton « Connecter ». Une invitation vous permet alors d'entrer le mot de passe que vous avez précédemment choisi pour le SuperUser (accepter le certificat s'il s'en propose un).

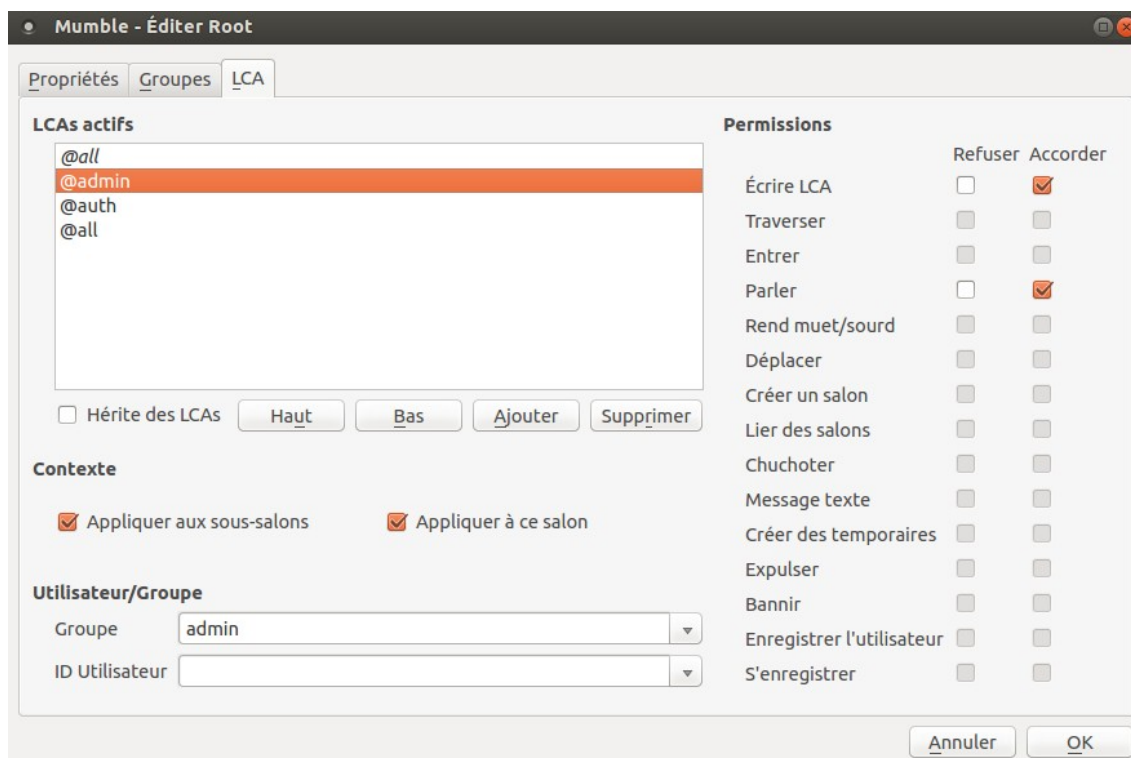
Dans cette seconde instance en *SuperUser*, certifiez votre pseudo

Enfin arrivé dans le mumble en tant que SuperUser, si votre principal pseudo d'utilisateur (resté connecté) n'apparaît pas enregistré (s'il n'arbore pas tout à droite le symbole d'utilisateur certifié), alors faites un clic-droit sur lui puis « S'enregistrer ».

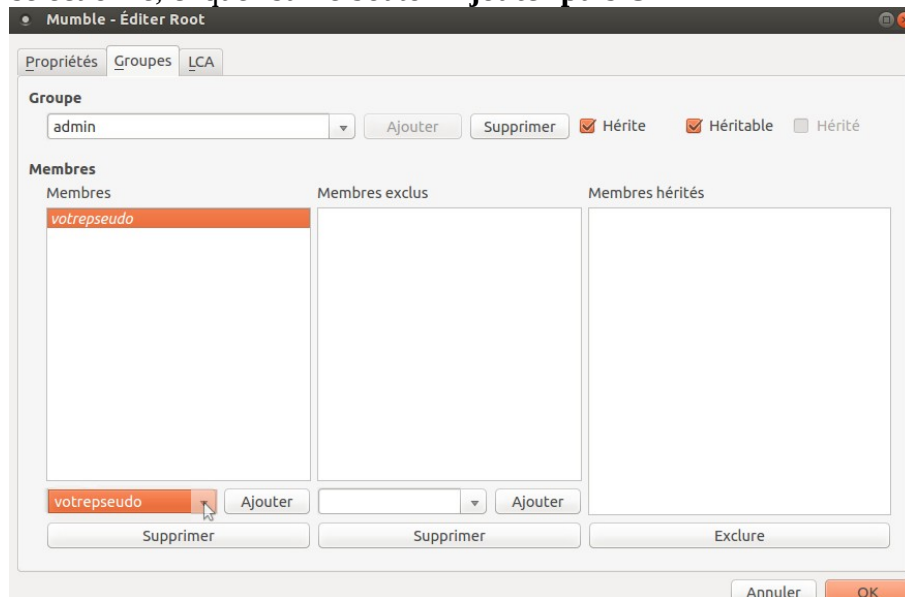
Ajouter votre principal pseudo d'utilisateur à la liste *Admin*

Tout commence par un clic-droit tout en haut sur la racine (*root*) puis « Éditer »

▫ Dans la fenêtre qui s'ouvre alors, à gauche sélectionnez « admin » et à droite assurez-vous que les cases cochées pour ce groupe sélectionné « admin » soient bien dans la colonne de droite « Accorder » pour les fonctions « Écrire LCA » et « Parler ».



☐ Dans l'onglet **Groupe** cliquez sur la flèche du menu déroulant vide sous le mot « **Groupe** ». Là sélectionnez « **admin** ». Le mot « **admin** » apparaît maintenant sous ledit mot « Groupe ». Ouvrez alors le menu déroulant qui est en bas à gauche, pour **ajouter** à ce groupe votre pseudo d'utilisateur préalablement enregistré (au groupe 'admin' ou un autre groupe que vous créez). Votre-Pseudo-De-Mumbleur sélectionné, cliquez sur le bouton **Ajouter** puis **OK**



C'est bon, vous pouvez refermer la seconde instance mumble du SuperUser et le Terminal.

Et voilà, félicitations !

Votre-Pseudo-De-Mumbleur est admin !

Il pourra à son tour en faire autant pour d'autres. Vos identifiants de SuperUser ne vous serviront plus que de roue de secours.

Sachez que vous pouvez ajouter un utilisateur directement parmi les groupes au lieu de le placer dans un groupe créé ou existant, mais ça peut entraîner des conflits et de l'illisibilité dans les droits (LCA).

Mumble - Éditer Root

Propriétés Groupes **LCA**

LCA's actifs

- @admin
- @auth
- @all

☐ Hérite des LCAs Haut Bas Ajouter Supprimer

Contexte

☒ Appliquer aux sous-salons ☒ Appliquer à ce salon

Utilisateur/Groupe

Groupe: all

ID Utilisateur: on peut écrire ici le nom d'un utilisateur (puis Entrée)

Permissions

	Refuser	Accorder
Écrire LCA	<input type="checkbox"/>	<input type="checkbox"/>
Traverser	<input type="checkbox"/>	<input type="checkbox"/>
Entrer	<input type="checkbox"/>	<input type="checkbox"/>
Parler	<input type="checkbox"/>	<input type="checkbox"/>
Rend muet/sourd	<input type="checkbox"/>	<input type="checkbox"/>
Déplacer	<input type="checkbox"/>	<input type="checkbox"/>
Créer un salon	<input type="checkbox"/>	<input type="checkbox"/>
Lier des salons	<input type="checkbox"/>	<input type="checkbox"/>
Chuchoter	<input type="checkbox"/>	<input type="checkbox"/>
Message texte	<input type="checkbox"/>	<input type="checkbox"/>
Créer des temporaires	<input type="checkbox"/>	<input type="checkbox"/>
Expulser	<input type="checkbox"/>	<input type="checkbox"/>
Bannir	<input type="checkbox"/>	<input type="checkbox"/>
Enregistrer l'utilisateur	<input type="checkbox"/>	<input type="checkbox"/>
S'enregistrer	<input type="checkbox"/>	<input type="checkbox"/>

Annuler OK

La suite vous appartient, les guides et astuces pour Mumble ne manquent pas sur internet, alors...

À vos navigateurs !



ANNEXES

Ici en annexe vous trouverez

_comment installer X2GO, une solution plus légère, plus sûre et plus appropriée que VNC mais pas toujours aussi facilement accessible pour les débutants

_comment créer un certificat SSL requis pour diverses validations internet et pour les bots

_comment installer et utiliser Docker qui facilite l'installation et l'utilisation de divers bots

_un extrait de discussions (forum de nam huy) pouvant résoudre des soucis techniques lors des installations précédemment évoquées

Installing and Configuring X2Go Server and Client on Debian 8

by [Rob Turner](#) | Published: July 3, 2015 | Last Updated: July 3, 2015

<http://www.tecmint.com/setup-remote-desktop-using-x2go-server-and-client-in-debian/>



Much of the power behind Linux comes from the command line and the ability for a system to be managed easily remotely. However, for most users from the Windows world or novice Linux administrators, there may be a preference to have access to the graphical user interface for remote management functionality.

Other users may simply have a desktop at home that may need to have graphical applications managed remotely as well. Which ever situation may be the case, there are some inherent security risks such as the remote traffic not being encrypted thus allowing malicious users to sniff the remote desktop session.

To solve this common issue with remote desktop systems, X2Go tunnels the remote desktop session through secure shell ([SSH](#)). While only one of many of the benefits of X2Go, it is a very important one!

Features of X2Go

1. Graphical remote desktop control.
2. Tunneled through SSH.
3. Sound support.
4. File and printer sharing from client to server.
5. Ability to access a single application rather than a whole desktop session.

Environment Setup

1. This guide assumes a working [Debian 8 \(Jessie\)](#) setup with LXDE (other desktop environments are supported however; please see [this link](#)).
2. Another Linux client to install the X2Go client software (This guide uses Linux Mint 17.1 with the Cinnamon desktop environment).
3. Working network connection with openssh-server already installed and working.
4. Root access

Installation of X2Go Server and Client on Debian 8

This part of the process will require setting up the X2Go server as well as an X2Go client in order to have a remote desktop connection. The guide will start first with the server setup and then proceed to the client setup.

X2Go Server Installation

The server in this tutorial will be the Debian 8 system running LXDE. The start of the installation process, is to install the X2Go Debian repository and obtain the GPG keys. The first step is to obtain the keys which can be easily accomplished the apt.

```
# apt-key adv --recv-keys --keyserver keys.gnupg.net E1F958385BFE2B6E
```

Once the keys have been obtained, a repository file needs to be created for apt to look for the X2Go packages at a specific repository location. This can all be accomplished with one simple command that creates the needed apt list file and puts the appropriate entry into that file.

```
# echo "deb http://packages.x2go.org/debian jessie main" >>
```

```
/etc/apt/sources.list.d/x2go.list
```

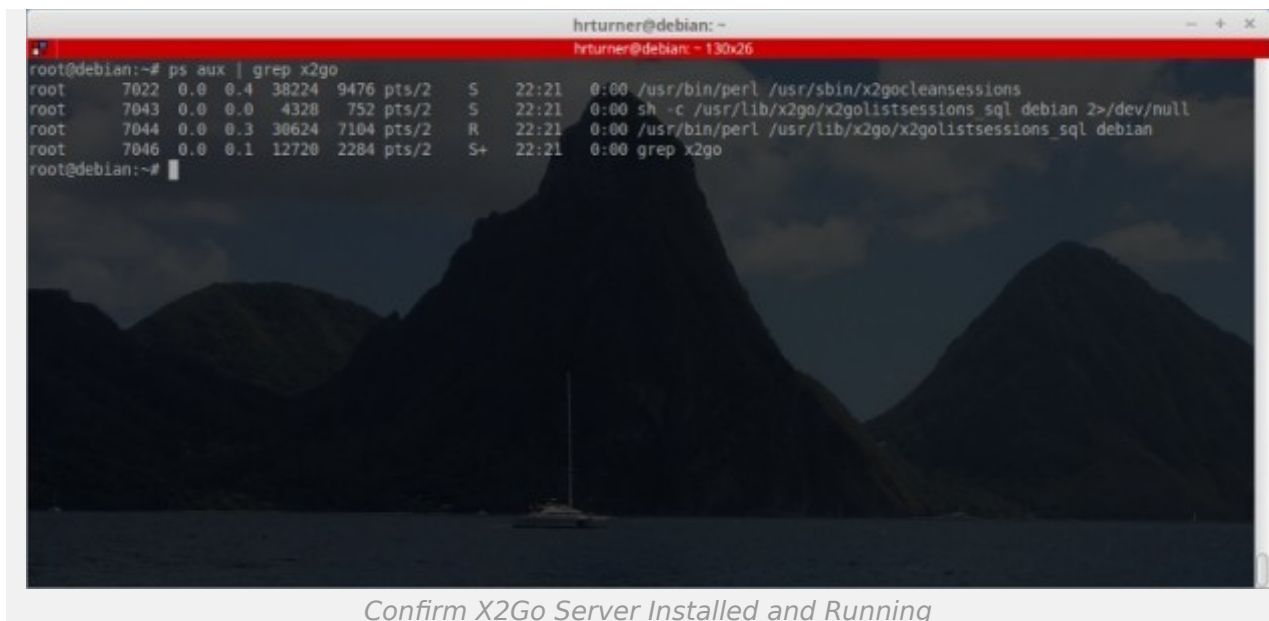
```
# apt-get update
```

The above commands will instruct apt to search this newly provided repository for packages and more specifically the X2Go packages. At this point, the system is ready to have the X2Go server installed using the apt meta-packager.

```
# apt-get install x2goserver
```

At this point the X2Go server should be installed and started. It is always a good idea to confirm that installed servers are running though.

```
# ps aux | grep x2go
```

A terminal window titled 'hrtturner@debian: ~' showing the output of the command 'ps aux | grep x2go'. The output lists four processes: a perl process for cleaning sessions (PID 7022), a shell process for a SQL session (PID 7043), another perl process for a SQL session (PID 7044), and the 'grep x2go' process itself (PID 7046). The background of the terminal window shows a dark, mountainous landscape with a small boat on the water.

```
hrtturner@debian: ~  
hrtturner@debian: ~ 130x26  
root@debian:~# ps aux | grep x2go  
root    7022  0.0  0.4 38224  9476 pts/2    S   22:21   0:00 /usr/bin/perl /usr/sbin/x2gocleansessions  
root    7043  0.0  0.0   4328   752 pts/2    S   22:21   0:00 sh -c /usr/lib/x2go/x2golistsessions_sql debian 2>/dev/null  
root    7044  0.0  0.3 30624  7104 pts/2    R   22:21   0:00 /usr/bin/perl /usr/lib/x2go/x2golistsessions_sql debian  
root    7046  0.0  0.1 12720  2284 pts/2    S+  22:21   0:00 grep x2go  
root@debian:~#
```

Confirm X2Go Server Installed and Running

In the event that the system doesn't automatically start X2Go, run the following command to attempt to start the service.

```
# service x2goserver start
```

At this point the basic server configuration should be done and the system should be waiting for connections from the X2Go client system.

X2Go Client Installation

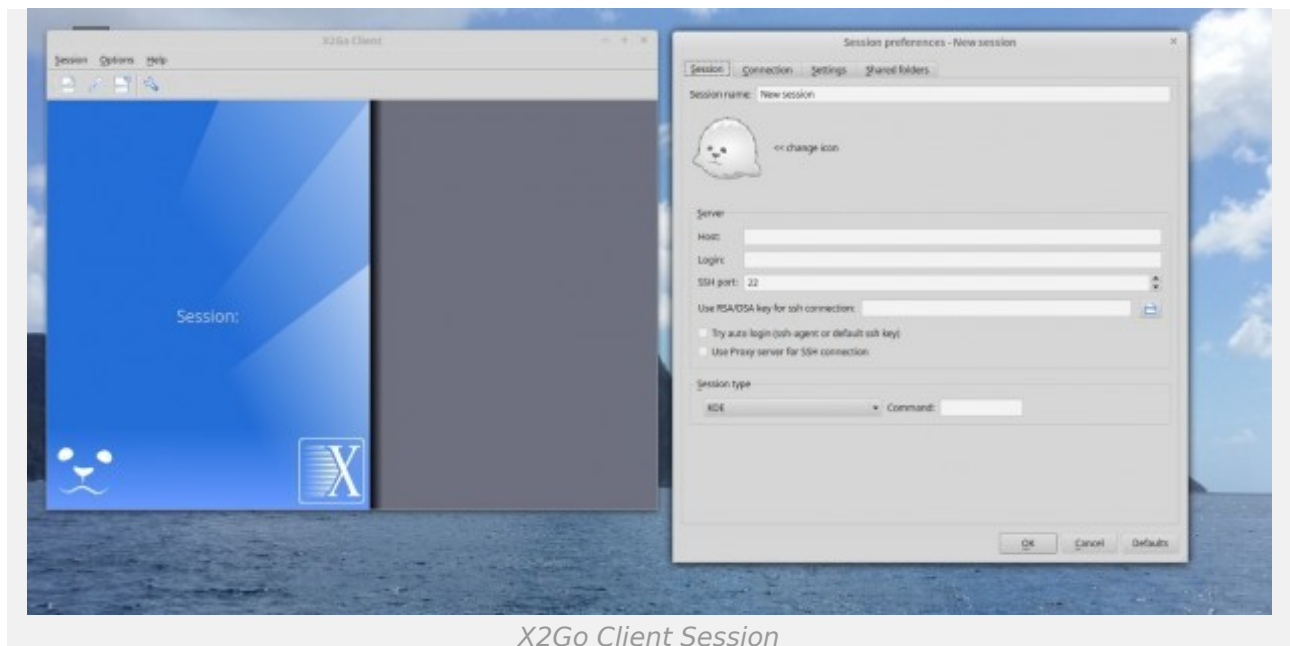
The client installation is easier than the server installation. Most distributions already have the client in their provided repositories and this package can easily be installed with the apt meta-packager.

NOTE: Remember that this is done on a computer that is going to connect to the server setup in the previous paragraphs.

```
# apt-get install x2goclient
```

Assuming that apt doesn't return any issues, the X2Go client should be ready to go. Navigate to the X2Go executable via the client's distribution's file explorer or launch the utility from the command line with the following command.

```
# x2goclient
```



X2Go Client Session

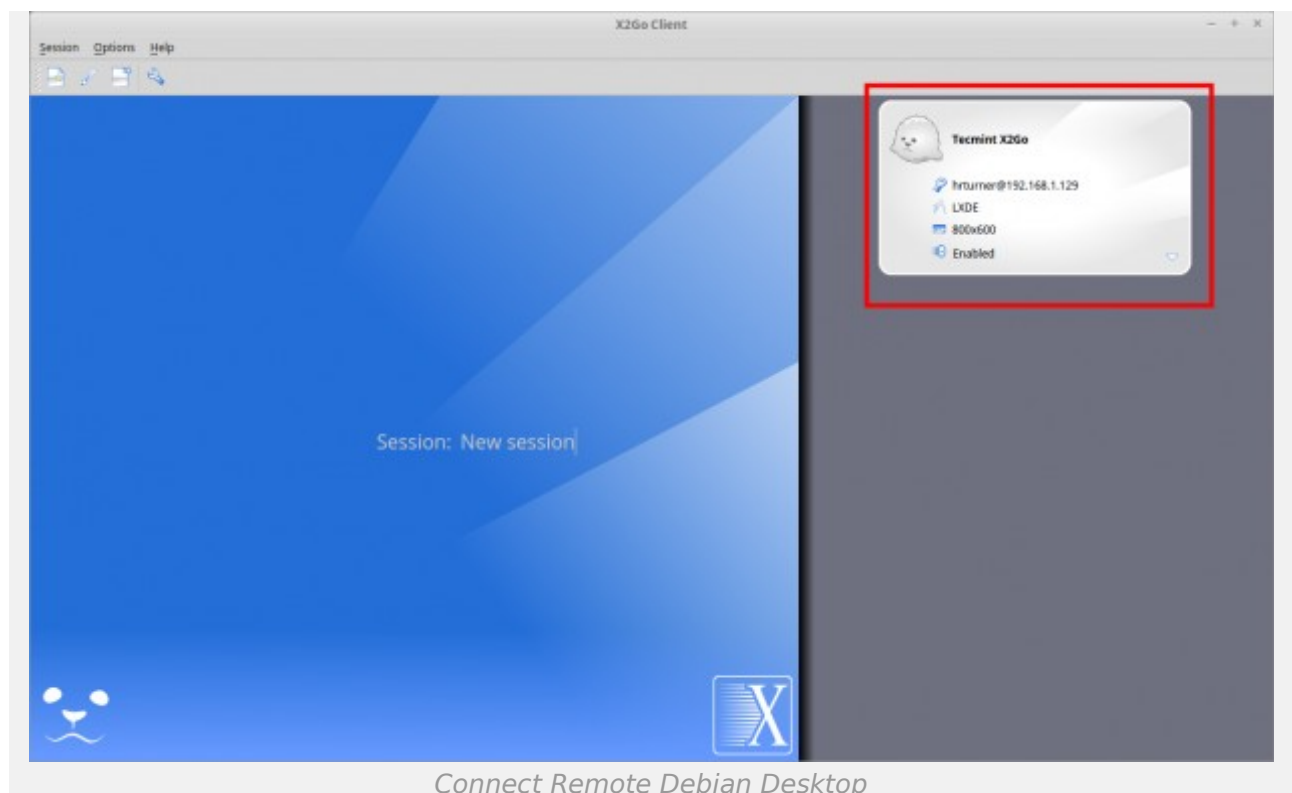
The above windows are the initial windows once the X2Go client is launched. Let's connect to the Debian Server now!

In the **server** field in the window on the right, enter the Debian system's ip address. The next box needs to have the user name of someone who can SSH into the Debian system.

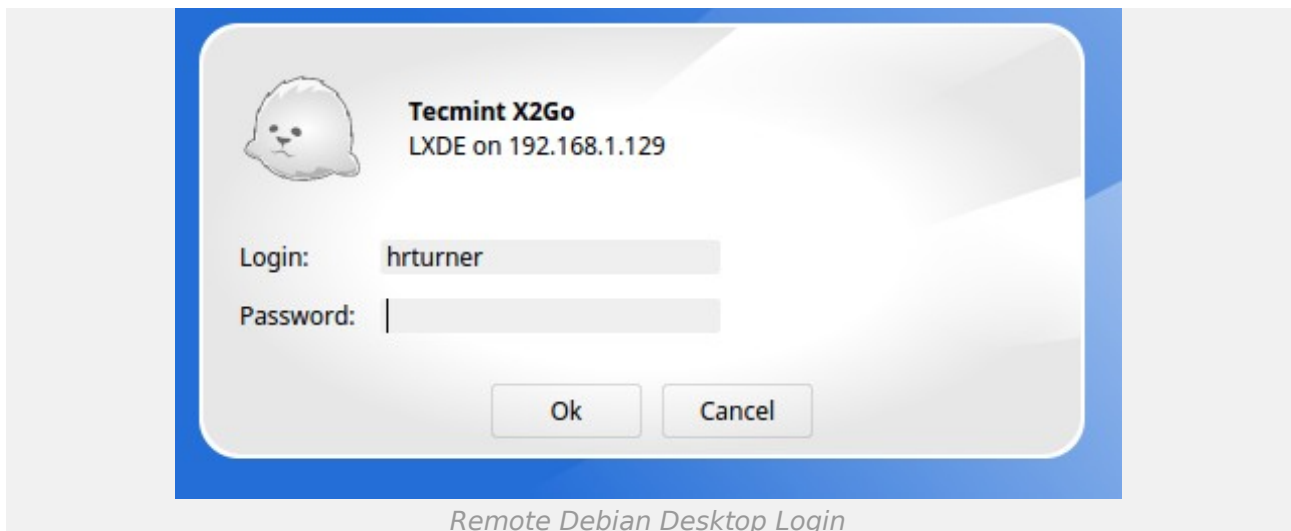
The next thing to change is the **Session Type** at the bottom. Since the Debian server is using LXDE, it is a good idea to select LXDE from the drop down.

Again, not all desktop environments are supported at the moment, please reference the link at the top of this guide to see what desktop environments are supported or if any work-arounds are needed.

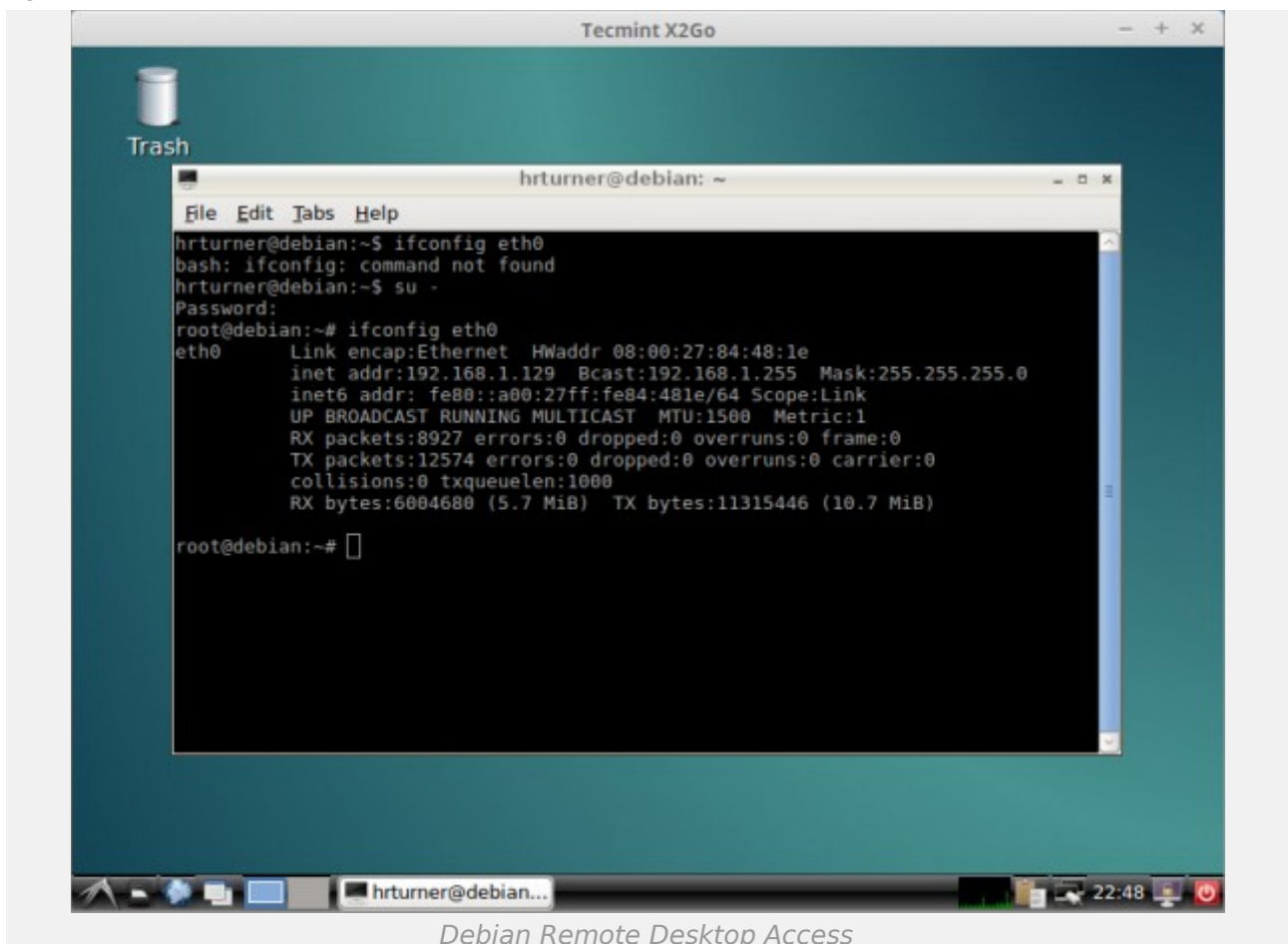
Once the above information have been input, click the “Ok” button at the bottom of the window to finish setting up the session profile. The next step is to click and activate the newly created session. To do this simply click on the session just created on the right in the X2Go Client window.



Once this session is selected, it will prompt for the user on the remote machine’s credentials. Again these credentials will be the user on the Debian server’s credentials!



Once the correct password is provided the system will then display the remote system's graphical display in a scalable window on the client system!



Hopefully at this point, your X2Go system is working like the above systems and you are enjoying a secure remote desktop connection to a Debian server!

Best of luck with this new (and more secure) remote desktop solution for a Debian Linux system!

TOUT CE QUI SUIT VOUS SERVIRA A PRÉPARER LE TERRAIN, SI VOUS COMPTEZ INSTALLER DES BOTS POUR VOTRE SERVEUR MUMBLE, IL VOUS SERA SANS DOUTE UTILE (VOIRE NECESSAIRE) DE CREER VOTRE CERTIFICAT SSL, PUIS D'INSTALLER DOCKER, VOILÀ COMMENT FAIRE :

CREER VOTRE CERTIFICAT SSL

OpenSSL est une boîte à outils open source qui implémente SSL/TLS ainsi qu'une bibliothèque de cryptographie.

Il va nous permettre dans cet article de générer un certificat auto-signé. Ce type de certificat est très utile dans un cadre d'utilisation personnelle, par exemple pour implémenter les variantes sécurisées des protocoles tels que HTTP, IMAP ou SMTP qui deviendront alors HTTPs, IMAPs et SMTPs. Certains bots pour Mumble exigent la présence d'un tel certificat lors de leur mise en oeuvre.

Par contre, si vous souhaitez un certificat spécifiquement établi pour un serveur Mumble privé et plus hautement sécurisé (non pour lesdits protocoles d'usage courant), référez-vous à des tutoriaux spécialisés comme sur ces pages là :

<http://www.waraxe.fr/sphinx/pages/cote-serveur/mumble-server.html>

https://wiki.mumble.info/wiki/Obtaining_a_StartCom_Murmur_Certificate

Installer OpenSSL

Pour installer OpenSSL, on se reportera à l'article installer OpenSSL sous Linux. Pour savoir si OpenSSL est déjà présent sur la machine, on peut consulter la version grâce à la commande suivante :

```
# openssl version
```

Si la version s'affiche, par exemple :

```
OpenSSL 1.0.1e-fips 11 Feb 2013
```

alors OpenSSL est déjà installé.

Créer le certificat

La création du certificat se déroule en trois étapes :

1. création de la clé privée
2. création d'une requête de certification
3. auto-signature

Dans le cas où le certificat sera certifié par une CA (autorité de certification) la procédure est la même, sauf que la requête de certification sera envoyée au CA au lieu de faire l'auto-signature.

Créer la clé privée

Pour info, la création de la clé privée se fait grâce à la commande `genrsa` d'OpenSSL à l'aide de quelques options :

- `genrsa` : génération d'une clé avec l'algorithme RSA
- (facultatif) `-des3` : permet de chiffrer la clé privée avec un algorithme **des3** qui nécessitera la création d'un mot de passe *ou* `passphrase`
- `-out` : fichier de sortie
- `4096` : taille de la clé, en bits

Lancez donc la création d'une clé privée de 4096 bits avec un mot de passe (remplacez 'example.com' par ce que vous voulez, préférez le nom de votre domaine le cas échéant, par exemple 'monsiteweb.org.key') :

```
# openssl genrsa -des3 -out example.com.key 4096
```

```
Generating RSA private key, 4096 bit long modulus
```

```
.....  
.....  
.....
```

```
....++
```

```
.....++
```

```
e is 65537 (0x10001)
```

```
Enter pass phrase for example.com.key:
```

```
Verifying - Enter pass phrase for example.com.key:
```

Le fichier de sortie `example.com.key` devrait être copié dans un répertoire protégé pour une utilisation ultérieure. Cette clé doit rester secrète, il faut donc aussi réduire les droits sur ce fichier (lecture uniquement pour le propriétaire) voici les commandes pour ce faire :

```
$ mkdir /usr/local/ssl/
```

```
$ cp example.com.key /usr/local/ssl/
```

```
$ chmod 400 /usr/local/ssl/example.com.key
```

Comme indiqué dans un commentaire, vous serez parfois obligé de supprimer la passphrase associée à une clé pour

pouvoir l'utiliser dans un applicatif donné tel que Apache HTTP ou Nginx. La passphrase peut être supprimée avec cette commande : `openssl rsa -in example.com.key -out example.com.key`

Vous voulez savoir si votre clé privée est chiffrée ?

Regardez les premières lignes du fichier généré : `head example.com.key`. Si la clé est chiffrée vous aurez alors une ligne du type `Proc-Type: 4, ENCRYPTED` suivi d'une ligne indiquant l'algorithme utilisé.

Créer la requête de certification

La création de la requête de certification se fait grâce à la commande `req` d'OpenSSL avec un certain nombre d'options :

- `req` : gestion des requêtes de certification
- `-new` : nouvelle demande
- `-key` : clé privée à utiliser
- `-out` : fichier de sortie

La création d'une nouvelle demande de certification implique le renseignement d'un certain nombre d'informations sur le demandeur, sachant qu'ils ne sont pas forcément obligatoires (par exemple, `Organizational Unit Name`). Si vous ne voulez pas remplir un champ, mettre un espace.

Important : l'information `Common Name` doit correspondre à l'URL que tapera l'utilisateur, si le certificat est destiné à un usage Web ou au nom complet du serveur si autre utilisation (par exemple, `imap.example.com` pour un serveur IMAP)

La *passphrase* demandée correspond au mot de passe fourni lors de la création de la clé privée.

Par exemple, la création d'une requête de certification :

```
$ openssl req -new -key example.com.key -out example.com.csr
```

Lors de cette étape, différentes questions vous seront posées afin de compléter la création de la CSR et qui apparaîtront dans le certificat SSL. *Les caractères suivants ne sont pas acceptés : < > ~! @ # \$ % ^ * / \ () ? . , &*

Champ	Détails	Exemple
Common Name/ Nom commun/CN	Le nom de domaine pleinement qualifié. Si vous avez l'intention d'acheter un certificat SSL pour l'URL https://www.monsite.com, votre Common Name sera www.monsite.com.	www.monsite.com
Organization/ Organisation	Le nom légal et exact de votre organisation / entreprise. Ne pas utiliser d'abréviation sauf pour les dénominations SA, SARL, EURL etc...	Mon Association
Organization Unit/ Département	Département au sein de l'entreprise qui va utiliser le certificat SSL	Secrétariat
City/ Ville	La ville où votre entreprise est légalement établi	Lyon
State/ Région	La région où votre entreprise est légalement établie	Rhone-Alpes
Country/ Pays	le code pays à 2 lettres où votre entreprise est légalement établie, donc pour la France :	FR

Attention : ne pas saisir les attributs supplémentaires.

Attention : ne pas rentrer de mot de passe pour le challenge (appuyez simplement sur entrée)

```
Enter pass phrase for example.com.key: UnMotDePasseComplexeOuPas
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:Paris
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:www.example.com
Email Address []:admin@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Cette commande produit le fichier `example.com.csr` qui peut être utilisé dans deux cas : utilisé pour l'auto-signature ; envoyé à une autorité de certification tierce

- Si vous vous êtes trompé ou que vous souhaitez modifier quelconque ligne de ce fichier `csr`, recommencez, tout simplement.

- _____

- Auto-signer le certificat

La création du certificat auto-signé se fait grâce à la commande `x509` d'OpenSSL avec un certain nombre d'options :

- `x509` : gestion des certificats

- `-req` : pour passer un certificat

- `-days` : durée de validité du certificat, en jours

- `-in` : fichier contenant la demande de certification

- `-signkey` : auto-signature

- `-out` : nom du certificat de sortie

La *passphrase* demandée correspond au mot de passe fourni lors de la création de la clé privée.

Par exemple, une auto-signature d'un certificat valide 1 an :

```
$ openssl x509 -req -days 365 -in example.com.csr -signkey
example.com.key -out example.com.cert
Signature ok
subject=/C=FR/ST= /L=Paris/O= /OU=
/CN=www.example.com/emailAddress=admin@example.com
Getting Private key
Enter pass phrase for example.com.key:
```

Le fichier de sortie `example.com.cert` peut être copier dans un répertoire pour une utilisation ultérieure (par exemple, le répertoire de l'application qui en aura besoin).

Utiliser le certificat

On dispose désormais d'un certificat auto-signé qui peut être utilisé avec divers services comme **Apache HTTP**, **Nginx** ou **Dovecot** par exemple pour sécuriser les différents accès. Ces services ont généralement besoin des deux fichiers :

- celui désigné comme étant la « clé privée » : dans notre cas, le fichier `example.com.key`

- celui désigné comme étant le « certificat » : dans notre cas, le fichier `example.com.cert`

Le fichier `example.com.csr` correspondant, à la requête de certification est un fichier intermédiaire qui n'est pas utilisé par les services. Il peut néanmoins être conservé pour effectuer une nouvelle auto-signature, à l'expiration du certificat par exemple.

A noter : lors de l'utilisation sur un site Web, le navigateur affichera un message d'avertissement sur la connexion en HTTPS. Ceci est dû au fait que, comme son nom l'indique, le certificat utilisé est auto-signé, c'est à dire non certifié par une autorité de confiance telle que Verisign par exemple. C'est pour cela qu'il n'est pas conseillé d'utiliser un certificat auto-signé pour un site publique par exemple, mais plutôt dans un cadre restreint.

Docker et les containers sous Debian

Publié par Florian BURNEL le 23 Oct 2014 <https://www.it-connect.fr/debuter-avec-docker-et-les-containers-sous-debian-8/>

Voir aussi cet autre guide en anglais : <https://docs.docker.com/engine/installation/linux/debian/>

Sommaire

- I. Présentation de Docker
- II. Entre virtualisation et container
- III. Installation de Docker
- IV. Utilisation d'un container Docker
 - A. Rechercher un container Docker
 - B. Installer un container Docker
 - C. Démarrer le container Docker « LAMP »
- V. Redémarrer et arrêter un container
- VI. Quels sont les containers Docker en cours ?
- VII. Consulter l'aide

I. Présentation de Docker

Aujourd'hui, nous allons parler container avec la solution Docker. L'objectif de Docker n'est pas la création de machines virtuelles, il s'agit là de la création de container, mais alors quelle est la différence ? Nous verrons cela en début d'article, pour que tout cela soit clair dès le départ.

En fait, Docker a pour objectif de faciliter le déploiement d'applications, d'avoir plusieurs versions d'une même application sur un son serveur (phase de développement, **tests**), mais aussi d'automatiser le packaging d'applications. Avec Docker, on s'oriente vers de l'intégration et du déploiement en continu grâce au système de container.



De plus, Docker permet de garder son système de base propre, tout en installant de nouvelles fonctionnalités au sein de containers. En quelque sorte, on part d'une base qui est le système d'exploitation et on ajoute différentes briques conteneurisées qui sont les applications.

Dans ce tutoriel, nous verrons ce qu'est Docker, la différence entre une VM et un container, mais également comment installer Docker et comment créer son premier container avec Docker. Pour ma part, je me trouve sur une machine virtuelle sous Debian 8 Jessie.

Note : Docker nécessite une installation 64 bits de votre distribution pour fonctionner.

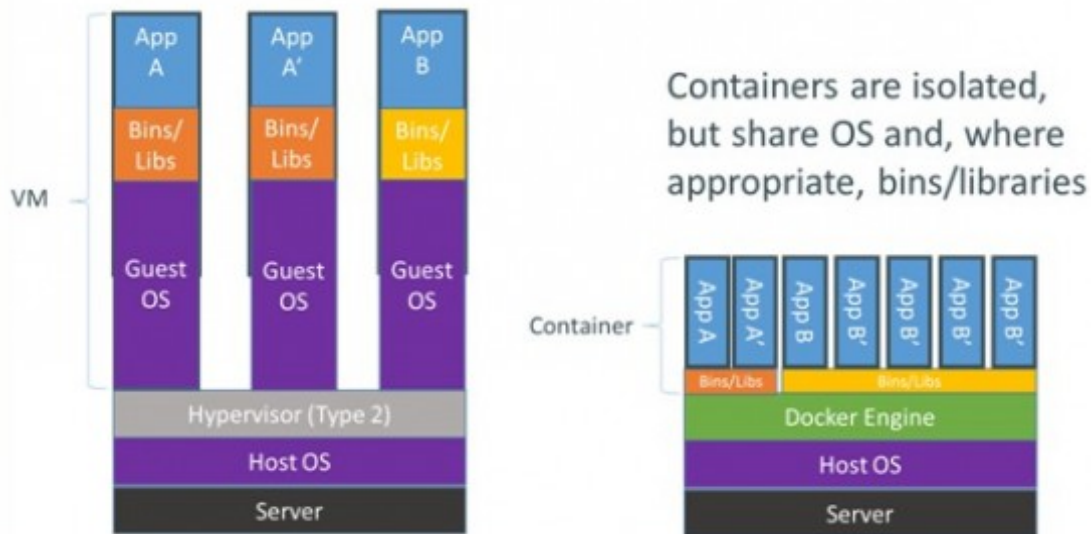
II. Entre virtualisation et container

De nos jours, les machines virtuelles sont très répandues et de nombreux logiciels (hyperviseurs) de virtualisation existent : **Hyper-V**, **VMware Workstation**, **VMware ESXi**, **VirtualBox**, **Proxmox**, etc. Pour ne citer qu'eux.

Avec une machine virtuelle, on propose une couche d'abstraction au-dessus d'un système d'exploitation existant. Il s'agit de simuler l'exécution de très bas niveau d'un système d'exploitation par-dessus un système d'exploitation existant, sans créer de liaison entre les deux. Finalement, une machine virtuelle est tout simplement la simulation d'une machine physique sur une machine physique déjà installée, et ce grâce à un hyperviseur.

Par ailleurs, un container s'appuie sur le système d'exploitation de l'hôte pour fonctionner. Il s'agit de simuler un ensemble applicatif au sein de l'OS de l'hôte, cela de façon isolée au sein d'un container. Un container est léger, performant et peut être déployé rapidement, car il partage ses ressources avec le système d'exploitation de l'hôte physique : kernel, périphériques, processeur, RAM, etc.

Containers vs. VMs



Source : [Docker.io](https://docker.io) – CONTAINER vs VM

III. Installation de Docker

Étant sous Debian 8, il est possible d'installer Docker directement grâce au paquet « `docker.io` » disponible dans les dépôts Debian :

```
apt-get update
apt-get install docker.io
```

Comme le montre la copie d'écran ci-après, Docker nécessite moins de 50 Mo pour l'installation.

```
Les NOUVEAUX paquets suivants seront installés :
 aufs-tools cgroupfs-mount docker.io git git-man libapparmor1 liberror-perl
 libnih-dbus1 libnih1 makedev mountall plymouth rsync
Les paquets suivants seront mis à jour :
 libc-bin libc-dev-bin libc6 libc6-dev libc6-i686 libgssapi-krb5-2
 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libtirpc1 locales
 nfs-common
13 mis à jour, 13 nouvellement installés, 0 à enlever et 362 non mis à jour.
Il est nécessaire de prendre 23,3 Mo/23,4 Mo dans les archives.
Après cette opération, 43,3 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] O_
```

Note : Avec Debian 7, il est nécessaire d'ajouter la source « `deb http://http.debian.net/debian wheezy-backports main` » dans le fichier `sources.list`. On pourra ensuite exécuter la commande « `apt-get update` » pour mettre à jour notre liste de source. Ensuite, exécutez la commande « `apt-get install -t wheezy-backports linux-image-amd64` » et installez Docker grâce à la commande d'exécution d'un script suivante « `curl -sSL https://get.docker.com/ | sh` ».

Lorsque l'installation sera terminée, il suffit de démarrer le service Docker comme un quelconque service :

```
service docker start
```

Nous pouvons passer à l'utilisation d'un container, nous prendrons un container LAMP comme exemple.

IV. Utilisation d'un container Docker

Docker propose de nombreux *templates* qui permettent de déployer des applications en container, très rapidement. La communauté est très active, ce qui permet aux utilisateurs de disposer de nombreux containers applicatifs prêts.

Docker est basé sur LXC (Linux Containers) qui est une référence sous Linux quant à l'utilisation des containers. Par ailleurs, Docker intègre les éléments suivants :

- cgroups (Control Groups) : Fonctionnalité du noyau Linux pour limiter, compter et isoler les ressources (CPU, RAM, etc.) utilisées par un groupe de processus.
- AppArmor et SELinux : Gestion avancée des permissions aussi bien au niveau des applications qu'au niveau du système de fichiers.
- Kernel namespace : Fonctionnalité du noyau Linux qui permet l'isolation, afin de s'assurer qu'un container ne puisse pas en affecter un autre.
- chroot : Fonctionnalité qui permet de changer la racine d'un processus, afin de l'isoler sur un système par mesure de sécurité.

Docker propose des services pour effectuer facilement différentes actions : créer, éditer, publier et exécuter des containers. Vous entendrez souvent parler de containers, d'images et de DockerFile :

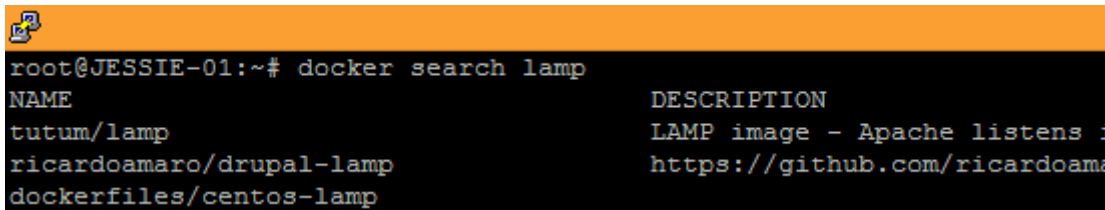
- DockerFile : Fichier source qui contient les instructions, éléments à installer, c'est un fichier de configuration.
- Image : Compilation d'un fichier DockerFile pour former une image portable, prête à être déployée
- Container : Exécution d'une image, mise en container d'une image.

Les trois termes ci-dessus sont importants, notamment si vous souhaitez vous lancer dans la création de votre propre ensemble Docker.

Le saviez-vous ? Docker est développé avec le langage Go
Passons à la pratique ! Dans ce premier tutoriel, nous allons déployer un container LAMP (Linux *Apache* MySQL PHP) afin de déployer un serveur web grâce à Docker.

A. Rechercher un container Docker

Nous souhaitons obtenir un container LAMP, pour cela nous allons regarder les images dans les dépôts Docker :



```
root@JESSIE-01:~# docker search lamp
NAME                                DESCRIPTION
tutum/lamp                          LAMP image - Apache listens 1
ricardoamaro/drupal-lamp            https://github.com/ricardoama
dockerfiles/centos-lamp
```

La liste que l'on obtient est longue... Il y a de la concurrence et du choix ! Pour faire votre choix, appuyez-vous sur trois critères principaux : la description, la popularité du container (stars) et le fait que ce soit un container officiel ou non.

Vous pouvez aussi consulter au préalable le *repository* de Docker : [Docker Browser](#)

Pour ma part, je pars sur l'image « tutum/lamp » qui est à l'heure actuelle l'image container de LAMP la mieux notée.

B. Installer un container Docker

La commande pour installer (récupérer, déployer) une image de container sur sa machine est la suivante :

```
docker pull tutum/lamp
```

Il suffit de remplacer tutum/lamp par le nom de l'image choisie.

```
root@JESSIE-01:~# docker pull tutum/lamp
Pulling repository tutum/lamp
4b32789c7d66: Download complete
511136ea3c5a: Download complete
1c9383292a8f: Download complete
9942dd43ff21: Download complete
d92c3c92fa73: Download complete
```

C. Démarrer le container Docker « LAMP »

Le démarrage d'un container s'effectue grâce à « docker run », comme ceci :

```
docker run -d -p 80:80 -p 3306:3306 tutum/lamp
```

Ce container écoute sur deux ports : le 80 pour le HTTP, le 3306 pour les connexions MySQL. On peut alors accéder à notre serveur web :



Hello world!

MySQL Server version: 5.5.38-0ubuntu0.14.04.1

Il faut savoir que chaque container peut disposer de ces options d'exécution et de configuration. Pour cela, il faut consulter la « fiche » du container concerné sur le *repository* de Docker.

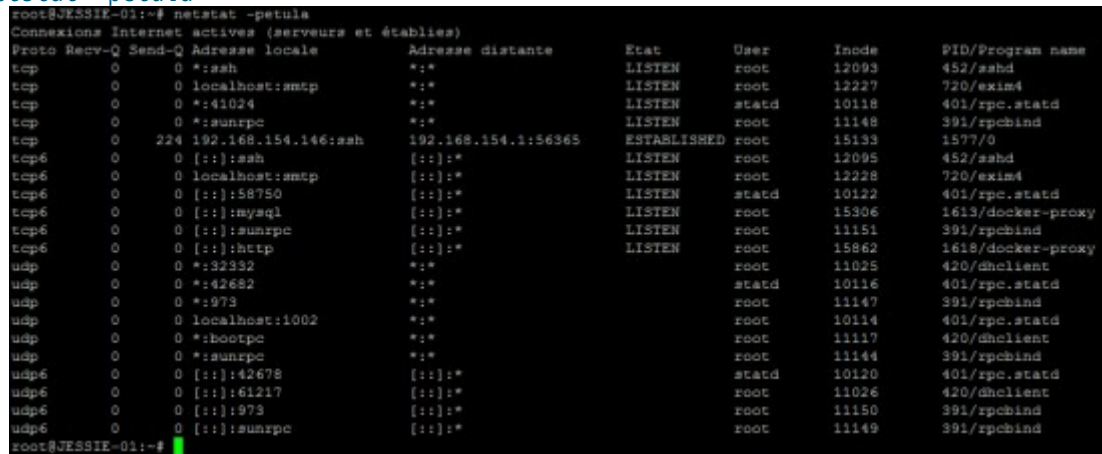
L'option `-p` permet d'indiquer les ports d'écoutes du container et fonctionne sous cette forme :

```
docker run -p <hôte_port1>:<container_port1> -p <hôte_port2>:<container_port2>
```

Le port d'hôte correspond au port sur lequel vous devez contacter l'hôte pour accéder au service, le port container correspond au port vers lequel sera redirigée la requête au niveau du container. Il y a en fin de compte un « NATage » de ports.

Si l'on regarde les ports *listen* sur notre serveur Debian 8, on remarque qu'il y a « docker-proxy » en écoute sur les deux ports que nous avons indiqués précédemment (80 et 3306) :

```
netstat -petula
```



```
root@JESSIE-01:~# netstat -petula
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale Adresse distante Etat User Inode PID/Program name
tcp 0 0 *:*ssh *:.* LISTEN root 12093 452/sshd
tcp 0 0 localhost:smtp *:.* LISTEN root 12227 720/exim4
tcp 0 0 *:41024 *:.* LISTEN statd 10118 401/rpc.statd
tcp 0 0 *:sunrpc *:.* LISTEN root 11148 391/rpcbind
tcp 0 224 192.168.154.146:ssh 192.168.154.1:56365 ESTABLISHED root 15133 1577/0
tcp6 0 0 [::]:ssh [::]:.* LISTEN root 12095 452/sshd
tcp6 0 0 localhost:smtp [::]:.* LISTEN root 12228 720/exim4
tcp6 0 0 [::]:58750 [::]:.* LISTEN statd 10122 401/rpc.statd
tcp6 0 0 [::]:mysql [::]:.* LISTEN root 15306 1613/docker-proxy
tcp6 0 0 [::]:sunrpc [::]:.* LISTEN root 11151 391/rpcbind
tcp6 0 0 [::]:http [::]:.* LISTEN root 15862 1618/docker-proxy
udp 0 0 *:32332 *:.* root 11025 420/dhclient
udp 0 0 *:42682 *:.* statd 10116 401/rpc.statd
udp 0 0 *:973 *:.* root 11147 391/rpcbind
udp 0 0 localhost:1002 *:.* root 10114 401/rpc.statd
udp 0 0 *:bootpc *:.* root 11117 420/dhclient
udp 0 0 *:sunrpc *:.* root 11144 391/rpcbind
udp6 0 0 [::]:42678 [::]:.* statd 10120 401/rpc.statd
udp6 0 0 [::]:61217 [::]:.* root 11026 420/dhclient
udp6 0 0 [::]:973 [::]:.* root 11150 391/rpcbind
udp6 0 0 [::]:sunrpc [::]:.* root 11149 391/rpcbind
root@JESSIE-01:~#
```

V. Redémarrer et arrêter un container

Démarrer un container c'est bien, savoir l'arrêter et le redémarrer c'est encore mieux. Pour cela, deux commandes sont à connaître :

- Docker : Redémarrer un container

```
docker start <container-id-ou-container-name>
```

- Docker : Arrêter un container

```
docker stop <container-id-ou-container-name>
```

Note : Le fait de redémarrer un container (d'ailleurs, c'est bien *start* et non *restart*), ne réinitialise pas ses options il sera exécuté avec les mêmes options que l'exécution initiale.

Comment savoir le nom ou l'ID d'un container ? Réponse à l'étape suivante.

VI. Quels sont les containers Docker en cours ?

Un container par ci, un container par là... L'exécution des containers peut vite se multiplier. Pour visualiser quels sont les containers en cours d'exécution, on utilisera la commande suivante :


```

root@JESSIE-01:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
3e5b6f93819a       tutum/lamp:latest   "/run.sh"          5 hours ago        Up 5 hours          0.0.0.0:80->80/tcp, 0.0.0.0:3306->3306/tcp   furious_hoover
root@JESSIE-01:~#

```

Vous remarquerez sur la copie d'écran que le container tutum/lamp est bien en cours d'exécution (c'est celui que nous avons exécuté précédemment).

Voici la signification des différentes colonnes :

- CONTAINER ID : Identifiant du container.
- IMAGE : Identifiant nominatif de l'image du container, avec notamment la version comme ici « *latest* » puisque c'est la plus récente.
- COMMAND : Commande passée en paramètre lorsque le container a été créé.
- CREATED : Date de création du container, ici « *Il y a 5 heures* » .
- STATUS : État du container, ici « *En cours d'exécution depuis 5 heures* » .
- PORTS : Les différentes redirections de ports configurées, rappelez-vous l'option -p...
- NAMES : Nom aléatoire donné au conteneur, ceci est personnalisable grâce à l'option -name lors de l'exécution *docker run*.

VII. Consulter l'aide

Pour vous guider, comptez sur la documentation officielle de Docker mais également sur les fichiers `man` :

`man docker`

Mais également plus précisément pour chaque commande :

`man docker run`

Ainsi, vous obtiendrez des informations précieuses quant à l'utilisation de certaines options, certaines commandes, si vous désirez déjà aller plus loin dans l'utilisation de Docker.

Documentation officielle : docs.docker.com

Ce premier tutoriel sur Docker touche à sa fin, il vous permettra de débiter sous Docker et de mieux comprendre la documentation officielle et les différents *man* fournis. N'hésitez pas à partager vos avis et expérience sur ce très bel outil !

Quelques discussions de dépannage extraites du forum de Nam Huy

source : <https://www.namhuy.net/3111/install-vnc-server-debian-7.html>

Comments and replies :

1. Ron 08/20/2014 at 14:15

EXCELLENT how-to. I've been bashing my head against the desk, looking at all of the outdated tutorials online.

One thing I am curious of though (and the only things I have seen that reference this are either a) old, or b) using a different VNC software; not sure if everything applies)... How would I go about integrating this into SSH, to secure the connection? I didn't really care about that, until I realized that vnc4server (and others, apparently) only utilize an 8 character password. For some reason, that just makes me paranoid (even though my VNC user is just a regular user, that user has quite a bit of control on the server).

Thanks in advance!

- namhuy 08/20/2014 at 23:16

you can use this command on your local machine to make a tunnel to your remote ssh/vnc server

```
ssh -p 22 -L 5901:127.0.0.1:5901 -N -f -l remoteuser remoteaddress
```

replace *remoteuser* with your user on your remote server, and replace *remoteaddress* with your remote ssh/vnc server. After you used the above command, it will ask you for your ssh password (not vnc password)

The next step is to start vncviewer or xtightvncviewer

(<http://www.namhuy.net/3106/install-vnc-server-ubuntu-14-04.html#comment-1179>) on your local machine to connect to your remote vnc server.

```
Xtightvncviewer 127.0.0.1:5901
```

Yes you will vnc to your localhost or 127.0.0.1 since you already have an opened ssh tunnel from your local machine to your remote ssh/vnc server. This time you should type in your vnc password.

Sample output when I connect to my remote vnc server through ssh tunnel

```
namhuy@namhuy-virtual-machine:~$ xtightvncviewer 127.0.0.1:5901
Connected to RFB server, using protocol version 3.8
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "vnc:5901 (vncuser)"
VNC server default format:
  16 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 31 green 63 blue 31, shift red 11 green 5 blue 0
Warning: Cannot convert string "-*-helvetica-bold-r-*-*16-*-*-*-*-*" to type FontStruct
Using default colormap which is TrueColor. Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Same machine: preferring raw encoding
```

2. etnbln 10/12/2014 at 09:33

I followed this tutorial, and it works mostly fine.

My problem is, that the vnc session that is created after reboot always has 1024 x 768 resolution, no matter what i enter in /etc/vncserver/vncservers.conf any idea what is happening?

3. etnbln 10/12/2014 at 09:35

oh, stupid mistake. i left the [1] put changed the port. problem fixed.

4. Dicko 01/25/2015 at 21:48

Then there is also

sudo apt-get install xrdp

and you have vnc over windoze rdp, no muss no fuss.

5. Tim 04/10/2015 at 14:04

Great tutorial but a question if I may, how do I get a VNC session to to the current desktop that is currently in use rather than starting a new desktop or if the PC is at the gui login stage I get a vnc session to the gui login screen??

- namhuy 04/10/2015 at 18:59

I believe vnc4server does not let you to vnc your current desktop session like you wanted to. I heard NX ([NoMachine](#)) lets you access the running X session on the console.

6. Totò 06/05/2015 at 08:18

great.

How can I ask user to login to Debian after he connect via VNC?

- namhuy 06/09/2015 at 02:47

you can create separate vnc session for each user. The vnc's authentication is the same as user login authentication

7. Neil 06/18/2015 at 05:03

I followed your tutorial and in lots of ways it works really well. However, when i try to configure a VPN connection i get an error "Connection add failed no session found for uid 1001(unknown)".

I can log in normally using the vncuser account and save the vpn connection info with no errors but its not visible when I connect remotely over the vnc.

Could this be something to do with way the vnc session is set up?

I am new to Linux so its a very steep leaning curve at the moment.

- namhuy 06/18/2015 at 08:39

what does the vnc's log give you? the logs should be located in \$HOME/.vnc/ and /var/log/auth.log

8. Neil 06/19/2015 at 19:39

First I would like to say thank you for your quick reply.

In the .VNC log I get below when I connect and try to configure the VPN. I am using the same gateway information that I have used to configure through the normal keyboard.

Sat Jun 20 12:19:55 2015

VNCSTConnST: Server default pixel format depth 16 (16bpp) little-endian rgb565

VNCSTConnST: Client pixel format depth 24 (32bpp) little-endian rgb888

Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.

** (nm-connection-editor:9393): WARNING **: Invalid setting VPN: gateway

** (nm-connection-editor:9393): WARNING **: Failed to get zones from FirewallID: (2) The name org.fedoraproject.FirewallID1 was not provided by any .service files

** (nm-connection-editor:9393): WARNING **: Invalid setting VPN: gateway

** (nm-connection-editor:9393): WARNING **: Invalid setting VPN: gateway

** (nm-connection-editor:9393): WARNING **: Invalid setting VPN: gateway

Sat Jun 20 12:23:03 2015

Connections: closed: 0.0.0.0::49684 (Clean disconnection)

SMsgWriter: framebuffer updates 286

SMsgWriter: ZRLE rects 606, bytes 718880

SMsgWriter: raw bytes equivalent 20098916, compression ratio 27.958652

I dont get anything in the auth.log

Any thoughts

- namhuy 06/19/2015 at 20:48

its likely an old bug, sorry i can't help much on the error.

<http://ubuntuforums.org/showthread.php?t=2028072>

<https://bugs.launchpad.net/ubuntu/+source/network-manager-openvpn/+bug/1254220>

tell me about your configuration? debian with gnome?

9. Neil 06/19/2015 at 22:22

I set it up just as you suggested with xfce

- namhuy 06/20/2015 at 00:47

did you install on a vps or dedicated server? how many network card do you have? static ip or from dhcp? any firewall/iptables?

10.Neil 06/22/2015 at 02:08

Its a normal standalone server, maybe 2 years old that was running win XP. I use a static IP address so its easy to connect to. I have not changed anything with firewall/iptables from default. I read the bug links above. I guess its similar I just get an error when I try and save new vpn connection I have created. I only need to set them up once, Is there anyway to create them normally and have them visible when I connect with the VNC.

I really appreciate your help with this. I am totally new to Linux so just about everything is a struggle.

- namhuy 06/22/2015 at 21:28

how did you set your network? can you copy/paste your ifconfig output here?

11.Neil 06/29/2015 at 04:07

ok to make things easy I enabled DHCP and set everything to default. When I connect over the vnc I can hover the mouse pointer over the network connection in the top right hand corner and a bubble comes up and tells me the ethernet connection is active.

However if I right click and select connection information I get an error showing no valid active connections found.

Not sure how relivent this is but the network connection icon showing in the top right is different when I connect over vnc compared to a normal log in.

12.Neil 07/20/2015 at 02:37

if I can provide more information that would help I would need a but more instruction.

- namhuy 07/20/2015 at 08:27

which part you don't understand? I can guide you from there.

13.Csaba 07/22/2015 at 00:54

Hi!

Thanks!

I have a problem. After I connect i see this:

<http://postimg.org/image/bb0sva46z/>

Please help me!

- namhuy 07/22/2015 at 03:13

can you copy and paste the logs which are located at \$HOME/.vnc/
and /var/log/auth.log

14.VNCuser9 01/16/2016 at 08:24

Hi,

Thank you for the tuto.

I have debian 7.8 wheezy stable.

I did exactly what you said but, when I reboot, vncserver doesn't start, I need to manually connect on "vncuser" to type "vncserver".

That's not my real problem.

The first time I logged on vnc, the display was good, menu etc... I logged out the user and closed vnc on my computer, rebooted the server.

now the problem is, I reconnect on vnc, but, I don't have bar menu, I just have the menu when I right click somewhere, and the 3 default icons.

How Do I get back the bar menus ?

Sorry, I'm not bilingual, I'm french.

Thank you.

- namhuy 01/16/2016 at 15:07

How many users do you have that have vnc's connection right? I think you have reconnected to the wrong user with wrong xstartup file What

do you have in your .vnc/xstartup file? Did you use the xstartup file I provided? By the way can you copy/paste the log file in vnc user directory?

```
cat $HOME/.vnc/xstartup
```

and

```
cat $HOME/.vnc/*.log
```

15.VNCuser9 02/12/2016 at 11:24

Hi, sorry for the long wait.

The panel is back when I go to panel settings.

```
#!/bin/sh
unset SESSION_MANAGER
unset DBUS_SESSION_BUS_ADDRESS
startxfce4 &

[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
xsetroot -solid grey
```

Xvnc Free Edition 4.1.1 - built Jan 30 2013 16:11:39
Copyright (C) 2002-2005 RealVNC Ltd.
See <http://www.realvnc.com> for information on VNC.
Underlying X server release 40300000, The XFree86 Project, Inc

Fri Feb 12 20:21:18 2016
vncext: VNC extension running!
vncext: Listening for VNC connections on port 5901
vncext: created VNC server for screen 0
error opening security policy file /etc/X11/xserver/SecurityPolicy
Could not init font path element /usr/X11R6/lib/X11/fonts/Type1/, removing from list!
Could not init font path element /usr/X11R6/lib/X11/fonts/Speedo/, removing from list!
Could not init font path element /usr/X11R6/lib/X11/fonts/misc/, removing from list!
Could not init font path element /usr/X11R6/lib/X11/fonts/75dpi/, removing from list!
Could not init font path element /usr/X11R6/lib/X11/fonts/100dpi/, removing from list!
sh: /home/vncuser/.vnc/xstartup: /bin/sh^M: bad interpreter: No such file or directory

I use only one user (vncuser) and root. I never tried to connect with root using vnc.

To start vncserver I type :

```
su vncuser  
vncserver
```

- namhuy 02/12/2016 at 14:42

How do you connect to your VNC? Do you connect to your VNC with root or vncuser user? You might have permission problem if you start vncserver with root but using VNC as vncuser. The best option is start vncserver as vncuser.

Also try this

```
chmod +x /etc/X11/xinit/xinitrc
```

and

```
chown vncuser:vncuser -Rf /home/vncuser/.vnc/
```