

# x11vnc

- Objet : Prise de controle d'un Bureau graphique à distance
- Niveau requis :  
[avisé](#)
- Commentaires : *x11vnc est un serveur utilisant le protocole vnc*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊 , [ssh: shell distant sécurisé](#)
- Suivi :  
[à-tester](#)
  - Création par [Switch](#) 23/03/2019
  - Co-créé par [Debian\\_Alain](#) 24/03/2019
  - Testé par [Debian\\_Alain](#) le 24/03/2019
- Commentaires sur le forum : <https://debian-facile.org/viewtopic.php?id=23958> <sup>1)</sup>

## Introduction

Ce programme permet de prendre le contrôle physique d'un ordinateur à distance ( contrôler clavier, souris, voir l'écran etc ...). Utile pour dépanner une personne à distance ou utiliser un ordinateur sans écran ( serveur, ou écran abîmé... ). x11vnc est un programme à installer sur l'ordinateur distant à contrôler ( partie serveur ). Il fonctionne de pair avec un "client VNC" ( un second programme à installer sur votre machine ex : xtightvncviewer ), qui vous donnera le contrôle de la machine distante. x11vnc vient de X11 ( le serveur graphique de Linux ) et VNC, le nom du protocole utilisé ici ( Virtual Network Computing ).

les commandes ci-dessous ont été testés sur

un serveur [debian stretch](#) 9.8(amd64) [bureau mate](#)(1.16.2) en [V.irtual M.achine Virtual Box](#)(6.0.4) .

[kernel](#) : Linux Host-004 4.19.0-0.bpo.4-amd64 #1 SMP Debian 4.19.28-2~bpo9+1 (2019-03-27)  
x86\_64 GNU/Linux

ou un [pc hp 6005](#) pour le script systemd . stretch 9.8 mate 1.16.2

[kernel](#): Linux Host-001 4.19.0-0.bpo.2-amd64 #1 SMP Debian 4.19.16-1~bpo9+1 (2019-02-07)  
x86\_64 GNU/Linux

et un client debian [buster testing](#) amd64 [bureau gnome](#)(3.30.2) [wayland](#) .

[kernel](#) : Linux Alain-PC 4.19.0-4-amd64 #1 SMP Debian 4.19.28-2 (2019-03-15) x86\_64 GNU/Linux

- par ailleurs, tout ce qui suit a été testé sur un réseau local . derrière une box .

- vu le nombre faramineux de box rien que dans les F.A.I. Français, j'ai sciemment shunté ce point .

- je ne suis pas assez calé là - dessus (les box) .

# Installation

Sur le poste distant à contrôler, que nous nommerons "serveur"

```
apt install x11vnc
```

Puis sur votre poste local ( client ) : Il existe de nombreux clients VNC ( graphiques, ou qui se lancent en console , multiprotocols etc). nous prendrons en exemple un client simple : tigervnc-viewer

```
apt install tigervnc-viewer
```



xtightvncviewer est un autre client VNC proche de tigervnc-viewer



vncviewer est un alias de ces deux client cités avant. C'est pourquoi dans les tutos sur x11vnc, vous verrez souvent ces 3 noms pour parler du client. Ca reste en gros la meme chose

## Exemple



L'exemple suivant n'est pas configuré NI SECURISE et n'est la qu'a titre éducatif. Prenez l'exemple du chapitre suivant avec un mot de passe pour une réelle utilisation de VNC !

Sur le serveur distant à contrôler ayant une session graphique démarrée : lancez le serveur x11vnc

```
x11vnc
```

Le serveur [x11vnc](#) se lance, affiche ses informations dans votre [console](#), et vous indique qu'il est prêt et attend une demande de connection d'un [client](#) sur le port par défaut 5900 :


[retour de la commande](#)

```
...  
The VNC desktop is:      vnd:0  
PORT=5900  
...
```

A présent, votre poste local, lancez le client vnc ( xtigervncviewer ) en indiquant l'adresse IP de votre serveur ( ex: 192.168.122.140 ):

```
xtigervncviewer IP_du_SERVEUR
```

Une fenetre graphique s'ouvre, affichant l'écran distant, dont vous pouvez contrôler la souris et le clavier 😊.




Pour quitter depuis le poste client : fermez la fenetre. Ou, pour quitter depuis le poste serveur; stoppez le programme x11vnc depuis la console ou il a été lancé avec `CTRL + C`

## Configuration

`x11vnc` permet beaucoup de configurations : des méthodes pour sécuriser par mots de passe, simplement montrer l'écran ou donner le controle complet ( avec souris clavier ), multi utilisateurs ... Lisez le manuel

```
man x11vnc
```

Quelques options essentielles [parmi toute celle disponibles](#):

Option	Explication	Exemple
<code>-display disp</code>	Précise l'affichage X11 à utiliser. Généralement :0 si vous n'avez qu'un écran.	<code>-disp :0</code>
<code>-auth fichier</code>	Nécessaire si x11vnc est lancé depuis une connection distante ssh ou en tant que service, ou par un script au démarrage du systeme ( donc lancé par root et non par vous ). Dans ce cas x11vnc doit s'identifier auprès du "serveur X" ( serveur graphique de Linux ) en montrant un "cookie" ( Xauthority ); sans quoi le serveur X rejettera x11vnc et l'écran sera inaccessible. Le chemin de ce cookie dépend de votre gestionnaire de connection ( Lightdm, gdm3 etc ). Pour le trouver, dans une console sur votre serveur, faites : <code>pgrep -a Xorg   grep -o '[-]auth [^]*'</code>	<div>Pour lightdm : <code>-auth /var/run/lightdm/root/:0</code> Pour gdm3 : <code>-auth /run/user/1000/gdm/Xauthority</code> etc...</div>
<code>-shared</code>	Autorise plusieurs clients a se connecter en meme temps au serveur x11vnc	
<code>-forever</code>	(identique à -many ) A la déconnexion du client, x11vnc reste allumé en attente d'autres demandes. Utile pour lancer un serveur vnc toujours allumé .	
<code>-xkb</code>	 Aide a configurer le clavier si différent entre serveur et client.	
<code>-bg</code>	(background) Va en arrière plan ( apres le message de lancement ). Utile pour les scripts ( ex: au demarrage du systeme ). Si besoin, activez les "logs" en fichiers, pour retrouver les messages de x11vnc.	

Option	Explication	Exemple
-o fichier	Définit un fichier de log de VNC	x11vnc [OPTION] -o /var/log/x11vnc.log
Option sécurité	Choisissez une des méthodes suivantes pour sécuriser votre serveur vnc	Exemple
<b>-rfbauth</b> <b>fichier_mot_de_pass</b>	<b>Méthode conseillée !</b> Protège votre serveur vnc par un mot de passe que vous définirez dans un fichier. La première fois avant d'utiliser cette option vous devrez créer et enregistrer un mot de passe ( par défaut dans <b>~/.vnc/passwd</b> ) avec la commande <b>x11vnc -storepasswd</b>	-rfbauth ~/.vnc_passwd
<b>-usepw</b>	Si pas de mot de passe déjà stocké sur la machine, alors x11vnc vous demande d'entrer un mot de passe dans la console. Puis vous propose de l'enregistrer dans <b>~/.vnc/passwd</b> pour les prochains lancements de x11vnc	
<b>-passwd</b>	<b>Méthode déconseillée</b> car le mot de passe sera visible dans votre ligne de commande depuis la liste des processus en cours Définit un mot de passe.	-passwd secret

### 1) première méthode sécurisation par mot de passe

de même que la sécurisation par mot de passe ,



il y a possibilité de créer un "tunnel ssh" (voir ci - après et liens)

ou de créer une connexion vnc par ssl (ssvnc) .



**ATTENTION !** ne confondez pas le mot de passe de vnc et le mot de passe user de votre systeme !

## utilisation de ssh

note : les exemples suivants s'entendent dans un seul sens :

aidant vers demandeur .



pour que le demandeur initialise la liaison afin d'être aidé ,

on peut parler alors de reverse ssh .

mais je n'y connais rien .

je laisse volontiers à plus calé le soin de vous l'expliquer .

**ssh** est un protocole de communication en console .

il permet , sur une console locale , de se connecter sur un ordinateur distant .

exactement comme si on y ouvrait une console .(sur le pc distant)

particularité :

comme toute architecture client / serveur , ssh nécessite un serveur sur la machine distante

et un client sur la machine locale ( [l'interface ssh](#))

aussi : en ssh , (sur la machine locale) vers la machine distante :

```
ssh id_user_distant@ip_distante
```

exemple :

où l'on voit la commande (ssh) le user distant(alain) et l'ip distant (192.168.1.32 : adresse réseau local)

[I.P.](#) vient de l'anglais I.nternet P.rotocol , soit : Protocole Internet .

il s'agit d'un procédé d'adressage à l'origine sur une séquence de 4 chiffres (IPV4) séparés par un point .

à venir , l' IPV6 : séquence et zone d'adressage beaucoup plus grands .



pour plus ample information , voire les liens en bas du document .

n.b.: ssh demande souvent une [authentification](#) par mot de passe , voire clé R.S.A. .

en réseau local , ssh est très bien . sur internet , c'est de suite plus difficile ...

en ayant au préalable enregistré le mot de passe (au dessus) :

cette commande peut lancer x11vnc sur le pc distant depuis la console locale .

on peut aussi la lancer en direct sur le pc distant où est installé le logiciel

```
x11vnc -q -many -rfbauth ~/.vnc/passwd
```

#### IMPORTANT !



on parle bien ici du mot de passe de vnc !

stocké dans le fichier par défaut “~/.vnc/passwd” de la commande “x11vnc -storepasswd”

```
alain@debian:~$ x11vnc -q -many -rfbauth ~/.vnc/passwd  
26/03/2019 03:42:44
```

```
26/03/2019 03:42:44 *** XOpenDisplay failed. No -display or DISPLAY.  
26/03/2019 03:42:44 *** Trying ":0" in 4 seconds. Press Ctrl-C to abort.  
26/03/2019 03:42:44 *** 1 2 3 4  
26/03/2019 03:42:48 *** XOpenDisplay of ":0" successful.  
26/03/2019 03:42:48
```

```
The VNC desktop is:      debian:0  
PORT=5900
```

## connexion à distance

il n'y a pas beaucoup de solutions pour l'accès à distance sur un pc .

la seule , (presque) disponible par défaut , c'est ssh . en mode texte , donc pas facilement abordable .

pour prendre la main en graphique sur le pc distant , il faut passer par là pour installer et configurer x11vnc , par exemple .

c'est aussi le seul moyen , en premier , pour configurer le chargeur de session .

une fois cette étape franchie , on peut accéder en graphique au pc distant (serveur) .

il permet l'accès à un serveur distant mais ne permet pas de lancer "bêtement" le bureau .

pour lancer le bureau , il faut modifier son lanceur (gdm / gdm3) pour activer l'autologin .

donc , par la même occasion , profitez en pour installer et paramétrer x11vnc puis , ensuite , redémarrage .

au reboot du serveur qui loggue donc automatiquement et sous x , vous pourrez lancer tigervnc et ,

donc , obtenir un beau bureau distant sur votre machine locale .

note : dans gdm3 , ne pas oublier de décommenter (activer , donc) la ligne suivante :

```
WaylandEnable=false
```

faute de quoi , il y a des chances que votre pc démarre sous wayland , donc , pas de serveur x et pas de x11vnc (même s'il est installé) .

conséquence directe : pas de bureau distant .

il y a une autre solution : toujours en ssh , installer x11vnc en systemd .

sous lightdm , vous aurez alors accès au login en graphique .

vous pouvez , dans lightdm , modifier le chapitre [vnc] (optionnel et pas obligatoire) qui devrait vous permettre

en théorie d'accéder à l'écran de login à distance et , donc , de lancer le login en graphique .

mais il faut , en préalable , installer x11vnc dans systemd ("/etc/systemd/system") . voir le chapitre

“systemd” plus bas .

qui lui , réellement , donne accès au login à distance , uniquement , pour l'instant , dans lightdm .

```
/etc/lightdm/lightdm.conf

[VNCServer]
enabled=true
command=Xvnc -rfbauth /home/<user>/.vnc/passwd
port=5900
listen-address=localhost
width=1024
height=768
depth=24
```

La connexion au login avec VNC lors de l'utilisation de GDM3 n'est actuellement pas possible.

Le moyen le plus simple de récupérer cette fonctionnalité VNC est simplement de revenir de GDM3 à LightDM.

LightDM est toujours activement développé et utilisé par de nombreuses distributions Linux

Pour réveiller un ordinateur à distance , on peut utiliser le W.O.L. (wake on lan) .

réveil de la machine par le réseau .

on a besoin de l'adresse mac (sorte de no de série de l'interface) et de l'adresse ip (adresse internet)

(voire bibliographies)

note :

curieusement , le lancement de x11vnc par lightdm , ne se fait pas . ce dernier en effet lance tigervnc(server) grâce à un lien codé en dur dans le programme (lightdm) . pour lancer x11vnc , il faut , soit le lancer indéfiniment par systemd , soit modifier le lien du chargeur de démarrage , soit reprogrammer ce dernier pour , enfin , lancer x11vnc au démarrage .

pour ma part , je suis partisan de la première solution (systemd) , switch , lui semble plutôt partisan de la deuxième solution (travailler sur les liens). elle aura un gros avantage . en ce sens qu'elle permettra d'utiliser moins de mémoire . mais que se passe t il en cas de crash ? dans ce dernier cas ? . je crois que la première solution est plus “sécuré” mais aussi plus gourmande en mémoire vu que le processus est lancé constamment et non pas à l'appel uniquement .

## Créer un tunnel ssh :

dans ce tuto , vous trouverez 3 méthodes de sécurisation de vnc :

1) le mot de passe (plus haut)

2) le tunnel ssh , ici .

cette méthode consiste à faire passer par ssh(port 22 , chiffré par défaut) une connexion vnc(port 5900 en clair)

3) et le protocole ssl (à suivre).

qui repose sur l'authentification par clé de chiffrement ssl .

ici , L'inconvénient de cette méthode est que vnc passe par un port non crypté (5900 pour le display :0).

Pour traverser une box ou un routeur il faut ouvrir des ports spécifiques pour vnc.

L'idéal est de passer par un tunnel.

Seul le port 22 pour ssh est routé dans la box, tout ce qui passe à travers est crypté ainsi que les tunnels qu'il ouvrira.

```
ssh -L 5900:localhost:5900 machine_distante 'x11vnc -q -display :0'
```

Pour connecter le client à l'autre bout du tunnel

dans une nouvelle console :

```
vncviewer localhost
```

hop ! le bureau distant ...

ici , un script qui fonctionne à peu près .

il est un peu capricieux , aussi , n'hésitez pas à le relancer si au bout de 10 secondes vous n'avez toujours rien à l'écran .

[scriptvnc.sh](#)

```
#!/bin/bash

#create SSH tunnel
ssh -f -L 5900:ip_distante:5900 -t user@ip_distante 'x11vnc -q -
geometry 1280x800; /bin/bash'

#open vncviewer
vncviewer localhost:5900

exit 0
```

pour vous simplifier le travail , voici une commande à taper :

```
cat > ~/scriptvnc.sh <<EOF
#!/bin/bash

#create SSH tunnel
```



```
ssh -f -L 5900:ip_distante:5900 -t user@ip_distante 'x11vnc -q -geometry 1280x800; /bin/bash'

#open vncviewer
vncviewer localhost:5900

exit 0
EOF
```

n'oubliez pas les droits :

```
chmod u+x ~/scriptvnc.sh
```

dorénavant ,

```
./scriptvnc.sh
```

suffit pour lancer sur votre machine locale l'affichage de votre machine distante .

qui plus est , en passant par un tunnel ssh . (sécurité)

NB: n'hésitez pas à remplacer <user> par le nom de l'utilisateur distant

et <ip\_distante> par l'adresse ip de votre machine distante .

vous pouvez accéder à l'ip de votre machine distante dans les paramètres de certaines box , ou , sur la machine distante , en tapant

```
ip a
```

et en cherchant les données "inet" pour l'interface concernée .



vous pouvez éditer le script avec nano (voir plus amples infos dans les notes de bas de page)



on parle , ici , de la connexion au systeme (en ssh) , donc , du mot de passe systeme (user)

par ailleurs, bien que la connexion x11vnc ne soit pas protégée (port 5900) , elle passe par ssh , donc , dans un tunnel (port 22) , et est , de ce fait , cryptée par ssh.

écran :

```
alain@Alain-PC:~$ ssh -L 5900:localhost:5900 192.168.1.32 'x11vnc -q -display :0'
```

```
alain@192.168.1.32's password:
#####
#@#####
#@
#@ ** WARNING ** WARNING ** WARNING ** WARNING ** @#
#@
#@ YOU ARE RUNNING X11VNC WITHOUT A PASSWORD!! @#
#@
#@ This means anyone with network access to this computer @#
#@ may be able to view and control your desktop. @#
#@
#@ >>> If you did not mean to do this Press CTRL-C now!! <<< @#
#@
#@#####
#####

PORT=5900
```

The VNC desktop is:        debian:0

```
alain@Alain-PC:~$ vncviewer localhost
```

```
TigerVNC Viewer 64-bit v1.9.0
Built on: 2018-12-01 21:51
Copyright (C) 1999-2018 TigerVNC Team and many others (see README.rst)
See http://www.tigervnc.org for information on TigerVNC.
```

```
Tue Mar 26 04:04:18 2019
DecodeManager: Detected 16 CPU core(s)
DecodeManager: Creating 4 decoder thread(s)
CConn:         connected to host localhost port 5900
CConnection:   Server supports RFB protocol version 3.8
CConnection:   Using RFB protocol version 3.8
CConnection:   Choosing security type None(1)
CConn:         Using pixel format depth 24 (32bpp) little-endian rgb888
CConn:         Using Tight encoding
```

## connexion avec ssl (préambule)

3 ème solution pour sécuriser vnc

authentification par certificat ssl :



SSVNC est un client ssl - vnc dérivé de tightvnc . il gère aussi le ssh ainsi que le ssl + ssh .

je ne parle pas de ces deux options annexes ici , je ne me suis pas penché dessus .

mais libre à vous . vous pourrez ainsi développer ce tuto .

installer ssvnc en local

```
apt install ssvnc
```

on crée le dossier `./vnc/certs/accepted/`

avant de le créer , vérifier , avec `tree` , s'il n'existe pas déjà :

auquel cas , l'étape de création est inutile .

```
tree ~/.vnc/
```

```
mkdir ~/.vnc/certs/
```

```
mkdir ~/.vnc/certs/accepted/
```

en ssh , on se connecte à la machine distante :

```
ssh <user_distant>@<ip_distant>
```



ici , il est question du mot de passe systeme (en cas de besoin)

on initialise la connexion :

```
x11vnc -ssl SAVE-key
```

après avoir rentré la passphrase , qu'on note sur un carnet de peur de l'oublier ,



ICI , il est question du mot de passe ssl (de la passphrase)

on va copier la clé depuis le serveur vers le client :

```
scp -r -p ~/.vnc/certs/server-key.crt  
id_locale@ip_locale:~/.vnc/certs/accepted/server-key.crt
```

## connexion avec ssl (utilisation manuelle)

relancer x11vnc en ssh

```
x11vnc -ssl SAVE-key -forever
```



rentrer la passphrase que l'on a noté (pour ne pas l'oublier).

on ne ferme pas la console distante (ssh) et

on ouvre une console locale :

```
ssvnc
```

puis rentrer comme demandé l'adresse de l'hôte :

ici , le pc distant .

et le mot de passe du pc distant .

ne pas rentrer de proxy .

dans le cas présent , c'est inutile .

laisser les réglages par défaut et

cliquer sur connexion .

ssvnc demandera confirmation pour le certificat (la 1ere fois) et

lancera vncviewer en ssl .

voilà !

## Connexion à ssl : écrans types ssvnc

graphiques :

[image de départ ssvnc](#)



[ssvnc menu "options"](#)



[écran d'accueil ssvnc](#)



[demande confirmation mot de passe systeme \(ssh\)](#)



[résultat par exemple \(ssvnc d'une V.M. à l'origine\)](#)



# Démarrage automatique

## Au démarrage d'une session graphique avec "autostart"

```
nano ~/.config/autostart/x11vnc.desktop
```

```
[Desktop Entry]
```

```
Type=Application
```

```
Name=x11vnc
```

```
Exec=x11vnc -forever -rfbauth /home/alain/.vnc/passwd
```



sur votre configuration , remplacez `"/home/alain/"` par `"/home/<votre_user>/"`

au redémarrage de la session , on s'aperçoit que l'on a un processus de lancé :

```
pgrep x11vnc
```

voilà son P.I.D. par exemple

```
1536
```

le lancement de vncviewer (ou tigervnc , si vous préférez ) , vous donnera un résultat positif :

```
vncviewer 192.168.1.32:5900
```

```
TigerVNC Viewer 64-bit v1.9.0
```

```
Built on: 2018-12-01 21:51
```

```
Copyright (C) 1999-2018 TigerVNC Team and many others (see README.rst)
```

```
See http://www.tigervnc.org for information on TigerVNC.
```

```
Tue Mar 26 10:21:38 2019
```

```
DecodeManager: Detected 16 CPU core(s)
```

```
DecodeManager: Creating 4 decoder thread(s)
```

```
CConn:      connected to host 192.168.1.32 port 5900
```

```
CConnection: Server supports RFB protocol version 3.8
```

```
CConnection: Using RFB protocol version 3.8
```

```
CConnection: Choosing security type VncAuth(2)
```

```
Tue Mar 26 10:21:50 2019
```

```
CConn:      Using pixel format depth 24 (32bpp) little-endian rgb888
```

```
CConn:      Using Tight encoding
```

## Au démarrage du système avec les "services systemd"

testé sur stretch 9.8.0 mate 1.16.2 et lmde 3 cindy cinnamon : pas de souci .

[Lmde3 cindy cinnamon download](#)

[debian stretch 9.8.0 mate](#)

pour créer un démon , c.a.d. une routine qui lancera x11vnc en arrière plan au démarrage ,  
on peut faire ceci :



ce n'est pas une solution universelle . il faut parfois l'adapter .

```
cat > /etc/systemd/system/x11vnc.service <<EOF
[Unit]
Description=Service x11vnc
Requires=display-manager.service
After=display-manager.service

[Service]
Type=simple
ExecStart=/usr/bin/x11vnc -rfbauth /path/.vncpasswd -o /var/log/x11vnc.log -
display :0 -auth /var/run/lightdm/root/:0 -forever -shared -rfbport 5900
Restart=on-failure
RestartSec=3

[Install]
WantedBy=multi-user.target
EOF
```

N.B.: remplacer

```
/path/.vncpasswd
```

par l'adresse du fichier de mot de passe .

ici (dans ce tuto):

```
/home/<user>/.vnc/passwd
```

par exemple :

```
/home/switch/.vnc/passwd
```

on attribue les bons droits au fichier :

```
chmod 0644 /etc/systemd/system/x11vnc.service
```

on l'installe dans systemd

```
systemctl enable x11vnc.service
```

on le lance :

```
systemctl start x11vnc.service
```

on vérifie son état :

```
systemctl status x11vnc.service
```

et c'est bon , le démon est installé . merci Delurk (irc) . reste plus qu'à redémarrer la machine pour tests .

mais normalement , vncviewer ou remmina vous donneront accès à l'interface graphique .

N.B.: si , comme moi , vous recopiez bêtement le texte et que , donc , vous devez le corriger ...

après la rectification , faites ceci :



```
systemctl daemon-reload
```

```
systemctl restart name.service
```

dans le cas présent , remplacez "name.service" par "x11vnc.service" (le nom de votre démon)

## Démarrage à la demande avec Inetd

## Astuces :)

Lancer un serveur vnc à distance , puis s'y connecter de chez vous, en une commande grâce à ssh : il s'agit d'un "tunnel ssh" .

```
ssh -f -L 5900:localhost:5900 your.ssh.server "x11vnc -safer -localhost -nopw -once -display :0"; vinagre localhost:5900
```

dans le cas présent , vous pouvez remplacer "vinagre" par "vncviewer" .

ce qui donne :

```
ssh -f -L 5900:localhost:5900 your.ssh.server "x11vnc -safer -localhost -nopw -once -display :0"; vncviewer localhost:5900
```

pareil : s'il est un peu capricieux , n'hésitez pas à le relancer plusieurs fois de suite (2 ou 3 fois suffiront

, normalement)

Lancer un serveur qui détecte le fichier X Authority :

```
x11vnc -display :1 $(pgrep -a Xorg | grep -o '[-]auth [^ ]*')
```

## création d'un alias

pour simplifier l'usage des commandes de x11vnc :

en cas de démarrage manuel :

pour créer un alias , c'est simple :

```
nano ~/.bash_aliases
```

y copier le code suivant :

```
alias assistance='x11vnc -many -rfbauth ~/.vnc/passwd -xkb'
```

il suffira , dès lors , de taper la commande

```
assistance
```

en lieu et place de la longue ligne d'instructions d'origine

N.B.: pour quitter nano :

[retour de la commande](#)

```
CTRL + 0 <enter>
```

puis

[retour de la commande](#)

```
CTRL + X <enter>
```

si vous préférez , c'est plus facile , copiez - collez le bloc suivant :

```
cat > /home/switch/.bash_aliases <<EOF
alias assistance='x11vnc -many -rfbauth ~/.vnc/passwd -xkb'
EOF
```



adaptez /home/switch/ à votre configuration . normalement /home/<utilisateur>/



avec un tunnel ssh , on peut obtenir ceci :

```
nano .bash_aliases
```

```
alias tunnelssh='ssh -f -L 5900:localhost:5900 your.ssh.server "x11vnc -safer -localhost -nopw -once -display :0"; vncviewer localhost:5900'
```



je ne sais plus s'il faut redémarrer la session ou le pc après la création d'un alias .

On peut lister les alias . ce qui permet de verifier si le nôtre est bien enregistré .

```
alias
```

```
alias agf='sudo apt full-upgrade'
alias agg='sudo apt upgrade'
alias agu='sudo apt update'
alias dir='dir --color=auto'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias fullmaj='agu && agf'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -l'
alias logs='sudo multitail -s 2 -f /var/log/messages -f /var/log/kern.log -f /var/log/syslog -f /var/log/dmesg -f /var/log/ufw.log'
alias ls='ls --color=auto'
alias maj='agu && agg'
alias vdir='vdir --color=auto'
```

## culture générale <bibliographies>

lectures utiles à compulser librement ou en cas de curiosité :

x11vnc :

[x11vnc -- wikipedia --](#)

[ubuntu - fr](#)

[linuxtricks \(intéressant mais peu fiable\)](#)

[linux mint perso](#)

[karlrunge \(eng.\)](#)

[x11vnc : commandes \(eng.\)](#)

Site généraliste très intéressant (ssh , vnc , ssvnc , ...) :

<https://pronflette.developpez.com/tutoriels/controle-distance/>

vnc :

[install and configure vnc \(eng.\)](#)

tigervnc :

[tigervnc \(eng.\) -- le site officiel --](#)

tightvnc :

[tightvnc \(eng.\) -- le site officiel --](#)

ssh :

[S.S.H. Secure Shell -- wikipedia --](#)

[page d'accueil ssh](#)

[client ssh](#)

[serveur ssh](#)

[tunnel ssh :](#)

[vnc : accès par tunnel ssh](#)

[super user : tunel ssh \(eng.\)](#)

ssl :

[ssl x11vnc \(eng.\)](#)

[ssl visualisation en graphique](#)

[explications sur ssl](#)

nano :

[petit mode d'emploi de nano](#)

ssvnc :

[site officiel ssvnc \(eng.\)](#)

[sourceforge ssvnc \(eng.\)](#)

adressage I.P. :

[SUPINFO : IPV4 VS IPV6](#)

systemd :

[Red Hat : systemd \(eng.\)](#)

[inetd / xinetd :](#)

[inetd wikipedia](#)

[xinetd wikipedia](#)

[inetd debian : le superserveur](#)

[/etc/inetd.conf](#)

[démarrer vnc comme service inetd](#)

[Lightdm et VNC :](#)

[lightdm archwiki](#)

[lightdm et vnc archwiki](#)

[GDM et VNC :](#)

[GDM3 et VNC](#)

[W.O.L. :](#)

[W.ake O.n L.an WIKIPEDIA](#)

[adresse MAC WIKIPEDIA](#)

[G wake on lan](#)

<sup>1)</sup>

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:reseau:x11vnc>

Last update: **01/10/2023 11:27**

