Table of Contents

Background information

If you are used to UNIX and Linux, you probably know that you can connect with `ssh` to a Linux/UNIX server and start a graphic application, which will then appear (if your ssh client is configured correctly) on your screen. This works because `ssh` automatically tunnels the X11 protocol so that the X11 applications running on the server can talk to the X11 server running on your box. Unfortunately the X11 protocol is very heavy, so on slow connections it can become very painful.

Several tools have been developed to address the slowness of X11, the main one being VNC⬀, with the relative TightVNC⬀ and the alternative NoMachine NX⬀. All of them start a virtual X11 (VNC) server on the remote machine so that graphic applications can talk X11 via the local network, while the VNC client (or viewer) running on your box uses a lightweight protocol to communicate with the server and display the content of the remote virtual screen.

Here I'm describing the easiest set-up, based on standard software packages already installed or easy to install. See also the video tutorial How to run to run vnc on lxplus⬀.

Prerequisites

Before you start, you need to have a valid CERN AFS account, to be able to connect to `lxplus` and start commands there.

On your box (Linux, Mac or Windows) you need a VNC native client or to be able to run Java applications on the web. Possible choices for VNC clients are:

- Linux (few examples, check you distribution repository for more)
  - ♦ xvnc4viewer
  - ♦ xtightvncviewer
  - ♦ gvncviewer (gtk+)
  - ♦ krdc (KDE)
  - ♦ gtkvncviewer (Gnome)
  - ♦ vinagre (Gnome)
- Mac
  - ♦ Chicken Of The VNC⬀ (open source)
- Windows
  - ♦ RealVNC⬀ (no need to fill in the details, just *Proceed to Downloads*)
  - ♦ TightVNC⬀ (both client and server with some nice features)

TightVNC offers a cross-platform viewer⬀ in Java too, which comes as a single file that can be executed from any directory without special privileges.

If for any reason you cannot install a VNC viewer on your machine, you can still access a remote VNC server if you can start Java applications from the web (you can test your Java installation on the Java web site⬀.

Initial configuration (server side, `lxplus`)

These steps have to be performed only once, before the first time you start `vncserver`.

Connect to `lxplus` your usual way. On the shell prompt check if you have a directory called `.vnc` in your home directory:

```
[lxplus] ~ $ ls ~/.vnc
ls: /afs/cern.ch/user/u/username/.vnc: No such file or directory
```

If you see the error message above, it's OK, otherwise it means you already started `vncserver` once, which can mean two things:

- you know what you are doing, in which case you can skip this chapter
- you must have started `vncserver` by mistake, so remove/rename the `.vnc` directory and follow the instructions

We need to set the password needed to access the VNC server we are going to start (type a password when asked):

```
[lxplus] ~ $ vncpasswd
Password:
Verify:
```

Let's create the VNC server start-up file: copy, with your favorite editor, the following text in a file called `~/.vnc/xstartup`

```
#!/bin/sh

# Uncomment the following two lines for normal desktop:
# unset SESSION_MANAGER
# exec /etc/X11/xinit/xinitrc

[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
xsetroot -solid grey
vncconfig -iconic &
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &

# display a dialog box to acquire the AFS token
(zenity --entry --title "AFS token" \
 --hide-text --text "Password for $USER@cern.ch" | kinit \
 || zenity --error --title "AFS token" \
 --text "Invalid credentials. You must run 'kinit' on the terminal.")&

wm=`which icewm mwm twm  | head -1`
$wm &
```

This is a slightly modified version of the `xstartup` created by `vncserver` on `lxplus`.

Now you have to make it executable:

```
[lxplus] ~ $ chmod a+x ~/.vnc/xstartup
```

Start the remote session and set up the ssh tunnels

First a few words about `vncserver`:

```
[lxplus] ~ $ vncserver --help

usage: vncserver [:<number>] [-nohttpd] [-name <desktop-name>] [-depth <depth>]
                 [-geometry <width>x<height>]
                 [-pixelformat rgbNNN|bgrNNN]
                 <Xvnc-options>...

       vncserver -kill <X-display>
```

VNC creates a virtual display for your applications. As any display, it has a size and a color depth. By default, it should start with a size of `1024x768` and a color depth of 24 bits (16M colors). Some times you may need to tune these parameters to better fit the memory constraint of the server or the screen resolution of the client. For example, to use all the real estate of your `1600x1200` monitor, you can pass to `vncserver` the option `-geometry 1600x1200`.

Once started, `vncserver` goes to background and listens at two TCP ports ranging from 5800 and 5900 respectively. The actual port numbers depend on the virtual display numeric id. For example, if `vncserver` prints that it started the desktop on *hostname*:5, then it listens at the ports 5805 and 5905 (respectively 5800+5 and 5900+5). To be able to connect to the server we have to create ssh tunnels for them.

When you want to close the virtual display you started, you have to call `vncserver -kill :`.

The connection to `lxplus`, and the creation of the tunnels highly depends on your client-side configuration, so I'll give few options.

Putty

Putty⬚ is a handy ssh client that is available for Windows (from the Putty web site⬚) and for Linux (check your package repositories).

Start Putty, you will get the window:

Putty main window

Insert `lxplus.cern.ch` in the *Host Name* field, select the `SSH` connection type (if not already selected) and click on the *Open* button.

Putty connection details

The window will change in a terminal window and it will ask for the user name and for the password (before that it may ask you to accept the key of the ssh server).

Once you have you shell prompt, you can start `vncserver`, optionally with the switches to change the screen size or the color depth:

```
[lxplus123] ~ $ vncserver

New 'lxplus123.cern.ch:7 (username)' desktop is lxplus123.cern.ch:7

Starting applications specified in /afs/cern.ch/user/u/username/.vnc/xstartup
Log file is /afs/cern.ch/user/u/username/.vnc/lxplus255.cern.ch:7.log
```

In this case, the display id is **7**, so we have to create the ssh tunnels for the ports 5807 and 5907.

Hold down the CTRL key of your and click with the right button of your mouse in the Putty console, release and select the entry *Change Settings...* from the menu that appears.

Look for *Connectioon->SSH->Tunnels* in the menu on the left hand side. In the configuration that appears on the right hand side we can add the forwarding of the two ports 5807 and 5907. To do it, insert the number (5807) in the *Source port* field, then `localhost:` (`localhost:5807` in the example) in the *Destination* field

and click on the button *Add*, then do it for the other port.

Configuration of the tunnels

Now you are ready to connect to the server.

Command-line ssh client

On (almost) any UNIX-like system (i.e. Linux and Mac) you have the `ssh` command installed. If not, you can easily install it:

- Ubuntu:
    - `sudo apt-get install openssh-client`
- Debian, as root:
    - `apt-get install openssh-client`
- RedHat/Fedora, as root:
    - `yum install openssh-clients`

To start `vncserver` on `lxplus`, you can simply start a terminal and type (optionally adding the switches to change the screen size or the color depth):

```
[mybox] ~ $ ssh username@lxplus.cern.ch vncserver
username@lxplus.cern.ch's password:

New 'lxplus403.cern.ch:7 (marcocle)' desktop is lxplus403.cern.ch:7

Starting applications specified in /afs/cern.ch/user/u/username/.vnc/xstartup
Log file is /afs/cern.ch/user/u/username/.vnc/lxplus403.cern.ch:7.log

[mybox] ~ $
```

In this case, the display id is **7**, so we have to create the ssh tunnels for the ports 5807 and 5907 to the host `lxplus403.cern.ch` (note that we have to bypass the generica lias `lxplus`):

```
[mybox] ~ $ ssh username@lxplus403.cern.ch -L5807:localhost:5807 -L5907:localhost:5907
```

```
The authenticity of host 'lxplus403.cern.ch (137.138.210.203)' can't be established.
RSA key fingerprint is a4:9f:57:6b:d5:4e:4d:56:85:ba:99:db:8c:2a:8e:b7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'lxplus403.cern.ch' (RSA) to the list of known hosts.
username@lxplus403.cern.ch's password:
...
...
[lxplus403] ~ $
```

The ssh tunnels will be active as long as the ssh connection is open, but you do not need to use this terminal until you want to kill the display.

Now you are ready to connect to the server.

Start the VNC viewer

Again, this depends on the configuration of your box, but we can consider few basic cases.

Client that can be started from command line

This includes all the Linux clients and the multi-platform TightVNC Java client.

When you can start the client from the command line, you just have to pass to it the string "localhost:". Few examples using the id 7 we already used before:

- xvncviewer localhost:7
- xvnc4viewer localhost:7
- krdc localhost:7
- java -jar tightvnc-jviewer.jar localhost:7

In all cases you will be asked to enter a password, which is the one you used at the beginning of this page.

IMPORTANT: you must start the viewer on your box and not on `lxplus`!

Client with graphical interface

This includes most Linux (KDE, Gnome, GTK, ...), Windows and Mac clients (including the multi-platform TightVNC Java client).

In this case, just start the client and give it the server name (localhost:*id*) and the password. The way to do it may vary a lot depending on the client, but is it usually easy to understand.

For example with `xvncviewer` you get

xvncviewer: server name

xvncviewer: password

with TightVNC Java client

TightVNC Java client: server name

TightVNC Java client: password

and with Chicken Of The VNC

Chicken Of The VNC: server name and password

Java-enabled web browser

If you do not want to install a local client (I'd like to remind you that the TightVNC Java client is a just a single file), you can still use a Java-enabled browser and connect to the URL `http://localhost:`, where is the port 5800+*id* (in our example it becomes http://localhost:5807☑).

Working with the VNC session

Once the connection to the server has been established, you will see a window displaying the remote virtual display.

xvncviewer: virtual display

Because of the way AFS Kerberos tokens are handled, the first thing you have to do is to acquire a new token. To simplify your life, the `xstartup` script above produces a dialog which asks for your AFS password and calls `kinit` for you. If you are not using the suggested script or you make a mistake when typing the password (sorry, no retrial), you will have to call `kinit` yourself from the console.

If you do not like the default terminal `xterm`, you can use others, like `gnome-terminal` or `konsole` (both available on `lxplus`).

Shut down the VNC server

Once you have finished with you session, you should go back to the terminal you left open (Putty or ssh) and call `vncserver -kill` passing it the display id with the ':' in front.

In our example with the display id 7, it becomes:

```
[lxplus403] ~ $ vncserver -kill :7
Killing Xvnc process ID 12345
[lxplus403] ~ $ exit
```

-- MarcoClemencic - 12-Sep-2011

This topic: LHCb > RemoteLxplusConsoleHowTo
Topic revision: r8 - 2012-04-23 - MarcoClemencic