DUNE Computing/Using VNC Connections on the dunegpvms

Contents

Using VNC Connections on the dunegpvms (Instructions slightly updated for AL9)

VNC provides you with a remote window to easily view GUIs from a remote machine with very little delayed response. The new instructions require a little bit more setup than the old instructions (linked at the bottom of the page) but the setting up only needs doing once. If you get stuck, feel free to contact me (Dom Brailsford: d.brailsford@lancaster.ac.uk).

You need to do a few things prior to following the rest of the instructions:

1. Pick a number for your VNC server (Pick any number between 0 and 99 -- in this example I'm going to use 22)
2. Pick a dunegpvm that you like to work on (I'm going to use dunegpvm06 in this example)
3. Install a VNC viewer on your local machine (Macs come with one already installed, an example of how to use it is shown at the bottom)

On your chosen dunegpvm (i.e. ssh to it)

Firstly, check whether your chosen VNC server number is available. Run the following on the command line (change the 22 to the number you picked - **do not forget the -localhost option. If you do, then your VNC session will be automatically killed by a cron job set up to detect these.)**:

```
vncserver :22 -localhost -bs -geometry 1680x1050
```

It will tell you if the VNC server already exists; pick a different number in that case and test again. If the specific server does not exist then it will be created (it may ask you to pick a password, remember it for later). To optimize the resolution and window size, you may need to experiment with the number of x and y pixels in the -geometry option. The numbers above work okay with a 13-inch retina screen on a mac when in full-screen mode.

Add the following snippet to the bottom of your $HOME/.profile or $HOME/.bash_profile (open either in a text editor). Again, change the 22 to your VNC server number.

```
#VNC stuff
VNCNUM=22 #CHANGE THIS NUMBER TO WHATEVER VNC SERVER NUMBER YOU PICKED
if [[ `hostname` == *"gpvm"* ]] #only start VNC servers on the gpvms (i.e. not on the build machines)
then
  #DBrailsford2024.  AL9 does not like the format used for DISPLAY in SL7 (DISPLAY=localhost:PORT).  It now needs to
be DISPLAY=:PORT
  export DISPLAY=":$VNCNUM" #Export the display to point to the VNC server
  if [ `lsof -i -P -n | grep $(expr 5900 + ${VNCNUM}) | wc -l` -eq 0 -o `lsof -i -P -n | grep $(expr 6000 + ${VNCNUM})
| wc -l` -eq 0 ]
  then
    echo "vncserver :$VNCNUM not running.  Starting now...."
    vncserver :$VNCNUM -localhost -bs    #Check if the VNC server is running and start it if not (-localhost
mandatory!)
  else
    echo "vncserver :$VNCNUM already running (hopefully owned by you).  Not attempting to start the vncserver..."
  fi
fi
```

On your local machine (i.e. open a new terminal window and do not ssh anywhere)

Add the following snippet to the bottom of your $HOME/.ssh/config. Change dbrailsf to your own kerberos principal, change the two occurrences of dunegpvm06 to whatever dunegpvm you chose at the start and change the 22 to your VNC server number (if your VNC number is between 0 and 9 then include a preceding 0 e.g. 00, 01, 02 etc.)

```
Host dunegpvm06
  HostName dunegpvm06.fnal.gov
  User dbrailsf
  ForwardAgent yes
  ForwardX11 yes
  ForwardX11Trusted yes
  GSSAPIAuthentication yes
  GSSAPIDelegateCredentials yes
  LocalForward 5901 localhost:5922
```

This snippet does a couple of things. Firstly, it allows you to connect to your chosen dunegpvm without specifying your kerberos principal or the full address of the gpvm. In the case of the above snippet, the following ssh command connects me to dunegpvm06

```
  ssh -Y dunegpvm06
```

Secondly, and more importantly, it automatically forwards the remote port used by the VNC server on the dunegpvm to a specific port (5901 in the example above - you do not need to change this number) on your local machine.

You should now be permanently setup to use VNC

How to use your VNC server

ssh to your chosen dunegpvm using its new shortened name e.g.

```
  ssh dunegpvm06
```

The ssh connection should automatically handle the port forwarding for you in the background. Now fire up root and open a TBrowser as a test.

```
source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
setup root v6_10_04d -q "e14:nu:prof"
```

and once the root prompt appears

```
 new TBrowser
```

On your local machine (open a new terminal window and do not ssh anywhere), open your vnc viewer software and point it towards localhost:5901. On a mac, the command is:

```
open vnc://localhost:5901
```

If you set a password when originally creating the VNC server on the dunegpvm, you will be prompted for it. Hopefully you now see the TBrowser.

To use VNC from now on, you should only need an active ssh connection to your chosen dunegpvm (e.g. when you're working on it) and a VNC viewer open and pointing to localhost:5901 on your local machine.

Disconnecting and Reconnecting

If you exit out of your ssh session that forwards the VNC's port, the VNC server will still be running on the dunegpvm machine. If you hadn't shut down the local screen sharing window, it will now say "Reconnecting" and display a spinning wheel. You can exit the screen sharing app. At a future time, you can re-log in to the gpvm with port forwarding, and restart the local screen sharing app, and you should be able to pick up where you left off.

Note -- it is easy to leave many (many, many) VNC sessions open on a dunegpvm machine. Please clean them up so they do not take memory or VNC slot numbers.

Old instructions

This is the text of an e-mail from Dom Brailsford, April 29, 2016:

```
There is already vncserver software set up on the gpvms (at least there is for dune).
You will need to install VNC viewer software on your laptop.  If you are a mac user,
this is already installed and I'll show you how to run this in the example below.

STEP 1: Start a VNC server on the gpvm

Log into the gpvm of your choice.
Start the VNC server.  The command is:

vncserver :X -localhost

where X is a number of your choice.  In my case, I chose 8, so my command was:

vncserver :8 -localhost

The number specifies the display you are going to use.  I don't think this will work if
someone is already using that display so, in that case, it may demand you use a different one.
If this is the first time setting up a server, it will ask you to pick a password.  Pick one.
The -localhost option is needed to block connections that come from outside not through the ssh tunnel.
If omitted, then SCD will block the gpvm from the network (!).  Update March 25, 2019:  SCD has installed
cron jobs that seek and kill non -localhost VNC sessions, in an attempt to get ahead of the node blocker.
You will receive an e-mail if your VNC session is killed.

STEP 2: Push the output of a remote terminal to the VNC desktop
```

```
In a terminal where you are connected to the gpvm and are doing work which
requires a GUI, issue the following command:

export DISPLAY=localhost:X

Where X is the number you chose.  In my case the command was:

export DISPLAY=localhost:8

STEP 3: Tunnel the VNC through ssh to keep it all encrypted

On your local machine, via a terminal, issue the following command:
ssh -K -L 59X:localhost:59X -N -f -l USERNAME GPVMADDRESS
Something to be aware of here.  The port forwarding for VNC is via ports 59[0..99] and for
numbers less than 10 you have to include the leading 0.  So my command was

ssh -K -L 5908:localhost:5908 -N -f -l dbrailsf dunegpvm06.fnal.gov

The -L 5908:localhost:5908 says forward information from the local side on port 5908 to
the remote host via its port 5908.  You have to make sure that the port number (in my case 08)
matches with the display used when setting in the vncserver.

STEP 4: Open the VNC window locally

Tell whatever local vncviewer you have installed to open the localhost window using the
port (localhost:59X or localhost:5908 in my case).  For mac users, there is already
one installed which you can access very easily via open.  The mac command is

open vnc://localhost:59X

In my case, the command was:

open vnc://localhost:5908
A desktop window should pop up.  In any remote terminal window in which you have pushed the
output to the VNC window (like step 2), the GUIs should open in this desktop.

AAACK -it asked for a password - that means you set up a password for your vnc on the gpvm at some point.  If you
don't remember it, you can go to $HOME/.vnc on the gpvm and remove the passwd file then try to restart your vnc server
(killing it is nontrivial) on the gpvm.

Then  on your localhost your open vnc should ask for that password and proceed happily.

STEP 5: Open a GUI remotely and watch it appear in the desktop window

Go back to the terminal in step 2 and open a GUI.  A quick test would be a TBrowser
root -l
new TBrowser
Confirm that the TBrowser opens in the VNC desktop window.  Try doing stuff
with it and note how quick it is.

I also believe that you can push multiple terminal displays to a single VNC server so
you could repeat step 2 for a bunch of different terminal windows.

You should re-use existing VNC sessions.  If you close down the window for the vncviewer on your local computer, the
VNC session is still running on the server.  Failing to clean them up will clutter the memory and swap file of the
server.  Leaving sessions running is a feature, not a bug, as it allows you to reconnect to your old session without
having to start from scratch.
```