Running CASA in a VNC Session at NRAO

Paul La Plante
plaplant@berkeley.edu

Last Updated June 3, 2021

1 Introduction

Getting CASA working inside of a VNC at NRAO is (unfortunately) non-trivial, though hopefully this guide can be somewhat demystifying. The basic steps can be summarized thusly:

1. Start a VNC session on NRAO (either on `herapost-master` or a compute node).
2. Launch a noVNC server that will serve as the client for the VNC session.
3. Connect to the noVNC server from your local machine via port-forwarding in `ssh`.

We go through each of these in turn.

2 Starting a VNC session on NRAO

The first step to launching a VNC session at NRAO is getting the `vncserver` command to play nicely with an Anaconda installation. Anaconda provides a number of Linux system libraries, one of which is d-bus. Unless care is taken, the VNC session will attempt to use the Anaconda-provided d-bus rather than the system one. Making VNC cooperate consists of: (i) adding a section to the user's `.bashrc` file to avoid loading Anaconda when instantiating a VNC session, and (ii) setting the `PATH` appropriately.

2.1 Updating `.bashrc`

Please add the following to your `.bashrc`, **above** the part where Anaconda is initialized:

```
if [ "$0" = "/etc/X11/xinit/xinitrc" ]; then
  return
fi
```

This will prevent VNC from getting the Anaconda-provided d-bus library.

2.2 Set PATH appropriately

The PATH variable must be adjusted when calling vncserver, again to avoid the wrong d-bus libraries from being used. We recommend creating an alias in your .bashrc file. Note the following command should be one single line:

```
alias casa_vnc='PATH=/opt/services/torque-6.1.2/bin:/opt/services/torque-6.1.2/sbin:
/usr/lib64/qt-3.3/bin:/opt/local/bin:/usr/local/cuda/bin:/usr/local/bin:/usr/bin:
/usr/local/sbin:/usr/sbin vncserver'
```

When the command casa_vnc is run, this creates a VNC session on the host on which it is run.

2.3 Launch vncserver

The vncserver command that is aliased to casa_vnc can be run either on herapost-master or a compute node. To request an interactive job for this, one can run the following command:

```
$ qsub -q hera -I -l nodes=1:ppn=1,vmem=16G,pmem=16G,walltime=4:00:00
```

This will request an interactive node with 1 processor, 16 GB of RAM, and 4 hours of time. Either on a compute node or herapost-master, one should then run:

```
$ casa_vnc
```

This will invoke the vncserver command with the PATH set appropriately. If this is the first time you have run the vncserver command, it will ask you to set a password. You should see something like this:

```
(base) [plaplant@herapost004 ~]$ casa_vnc

You will require a password to access your desktops.

Password:
Verify:
Would you like to enter a view-only password (y/n)? n
A view-only password is not used

New 'herapost004:1 (plaplant)' desktop is herapost004:1

Starting applications specified in /users/plaplant/.vnc/xstartup
Log file is /users/plaplant/.vnc/herapost004:1.log
```

Note that your VNC session will be safe behind an SSH tunnel, so a weak password is fine. It must be at least 6 characters, so I typically use hera.. for mine.

Note the VNC session number! This is the number after the `hostname:` in the start up message. In my case, my session is `1`. When specifying the port to forward to, you should add your session number to 5900 (because that's how VNC works). So for me, 5900 + 1 = 5901.

3 Using noVNC to connect to the VNC Session

noVNC is a browser-based way of connecting to a VNC session. It avoids having to install a VNC client on the user's machine, and instead uses a web browser window for interacting. Under the hood noVNC is a sort of client + server, where it acts as a client connecting to the VNC session we started in Sec. 2.3, which then serves the session to a web browser via an SSH tunnel.

3.1 Clone noVNC directory

noVNC requires no explicit installation, and is graciously hosted at NRAO in Danny Jacobs's home directory. Run the following commands:

```
$ cd
$ mkdir src
$ cd src
$ git clone ~djacobs/src/noVNC
```

This only needs to be done one time.

3.2 Launch noVNC

We now launch the noVNC client on `herapost-master`, being careful to forward the ports and hosts appropriately. Suppose as above I am running my `vncsession` on `herapost004`. My noVNC command will look like this:

```
$ cd ~/src/noVNC
$ ./utils/launch.sh --vnc herapost004:5901 --listen 6081
Warning: could not find self.pem
No installed websockify, attempting to clone websockify...
Cloning into '/users/plaplant/hera/noVNC/utils/websockify'...
remote: Enumerating objects: 4352, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 4352 (delta 11), reused 18 (delta 11), pack-reused 4327
Receiving objects: 100% (4352/4352), 4.63 MiB | 12.41 MiB/s, done.
Resolving deltas: 100% (2853/2853), done.
Using local websockify at /users/plaplant/hera/noVNC/utils/websockify/run
```

```
Starting webserver and WebSockets proxy on port 6081
WebSocket server settings:
  - Listen on :6081
  - Web server. Web root: /users/plaplant/hera/noVNC
  - No SSL/TLS support (no cert file)
  - proxying from :6081 to herapost004:5901


Navigate to this URL:

    http://herapost-master:6081/vnc.html?host=herapost-master&port=6081

Press Ctrl-C to exit
```

Note that in my command, I specify the VNC host + port to connect the client to as part of the `--vnc` option. I also specify the `--listen` option to specify which port to listen on for incoming connections. Note that unlike Jupyter Notebooks, noVNC is not clever enough to pick an alternative if the default (6080) is taken. You may have to try several different ones until you find an open port.

If you would like to avoid a dropped connection killing your noVNC server, you may run this inside of a screen session instead.

Note the URL that noVNC prints out at the end! This is *almost* right for connecting to the noVNC server, but must be modified to make use of the ssh tunnel we're about to open up. In particular, you must substitute `herapost-master` for `localhost` in all places.

4 Connecting to the noVNC Server

The noVNC server we launched in Sec. 3.2 is now listening on a particular port on `herapost-master`. We will open a new SSH session with this port forwarded from our local machine to NRAO.

4.1 Forward the Appropriate Port

On your local machine, start a new SSH connection to NRAO forwarding the port you selected with `--listen` in Sec. 3.2:

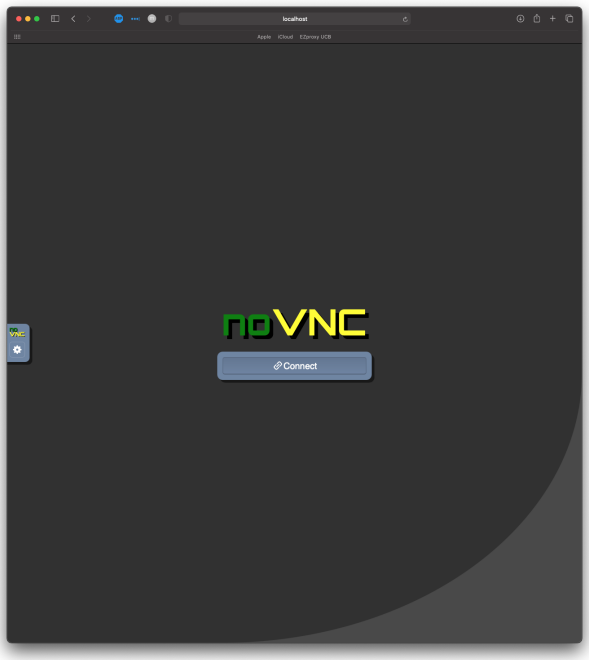
```
$ ssh plaplant@login.aoc.nrao.edu -L 6081:herapost-master:6081
```

4.2 Connect to noVNC

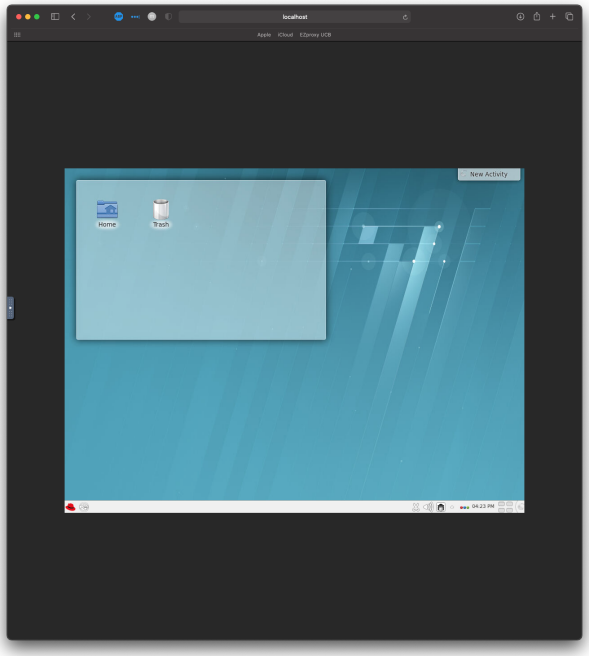
Paste the URL that noVNC printed at the end of the startup (with the substitution of `herapost-master -> localhost`) into a window on a local browser. In my case:

```
http://localhost:6081/vnc.html?host=localhost&port=6081
```

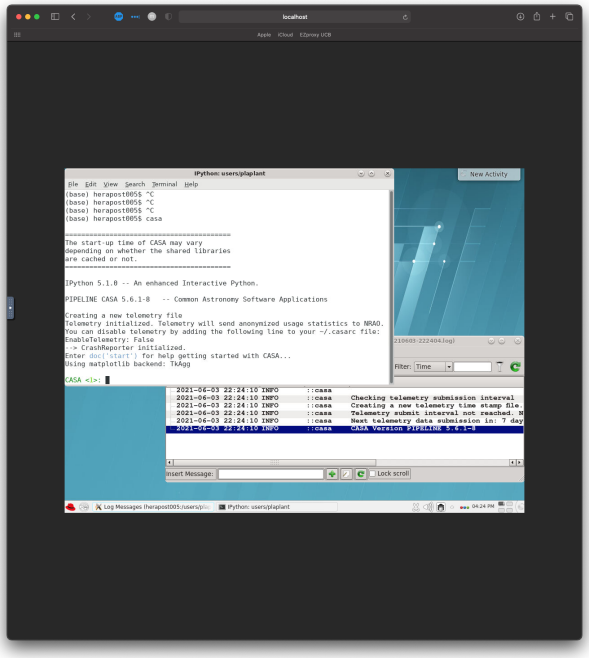
The result should be something like this:



Click “Connect”, and then enter the password (`hera..` or whatever, from Sec. 2.3). You should then get a Red Hat desktop:



Open a terminal (click the hat icon in the lower-left, type “Terminal” in the text box) and run `casa`:



Ta-da! We did it!