Let's say you'd like to submit a batch job to analyze/post process some of your results of your previous batch jobs. Sure, you got your data on /scratch which could be accessed on login nodes. Great. Right away, you fire up whatever the application through X windows and then you realize it is very slow. And at some point, right before you’re about to save the picture your window disappears and then your terminal windows get swamped with messages asking you not to run jobs on the login nodes. On top of that you’d get emails telling you the same annoying message. By the way, I'm the one who made this happen. Meaning, I installed Cgroups on login nodes that would kill current job once user's memory goes beyond allowed 2GB.

Oho, wait up. Don't curse at me. There is a reason I did this. Previously, on Bowery we didn't have Cgroups. Even if I wanted to implement I couldn't have as the OS was so old it wouldn't support it. Anyway, let's say there are few users who would end up using so much memory on login nodes, they become either slow or completely freeze or crash altogether. This means, every user on this login node feels it. Do you want this happen to you? I am sure you don't. So, I ended up doing Cgroups. This is a great feature that comes with RedHat/CentOS 6.X. Wait a second, why am I talking about Cgroups? Let's move on to what I wanted to talk.

So, now you know that your application in X-windows was killed because it used memory more than allowed 2GB. So, what is the solution? You'll need to submit an interactive job requesting more memory. Then, from there open the application. Again, you realize that application is extremely slow to respond. In these types of situations, there is a better solution. VNC: virtual network computing. This is definitely faster. So, how do we do it on our cluster Mercrer. For Mac users there is a separate document. You can look for it by searching for VNC Remote Desktop in the search bar.

### Running VNC on Windows

**Step 1: Open Putty and ssh directly into Prince directly. If you are off-campus this technique won't work.**

**Step 2: Run the below command to gain access to a compute node**

```bash
$qsub -I
```

Note: `-I` is caps i.

You should see something like the below text. Getting access to a compute node takes time (a few seconds to a few minutes based on how busy the cluster is)

```
________________________________________________________________________
WARNING no memory limit specified, assuming <2GB/CPU (2gb total)
qsub: waiting for job 8756849 to start
qsub: job 8756849 ready
________________________________________________________________________
```

**Step 3: Once you get access to a compute node you need the IP address of that node. For that run the command**

```bash
$hostname -i
```

You should see something like the below text in Putty.

```
________________________________________________________________________
10.0.254.174
________________________________________________________________________
```

**Step 4: Run the command**

```bash
$vncserver
```

You should see some text like the text below.

```
________________________________________________________________________
```
New `compute-4-12.local:1 (trc311)` desktop is `compute-4-12.local:1`

Starting applications specified in `/home/trc311/.vnc/xstartup`

Log file is `/home/trc311/.vnc/compute-4-12.local:1.log`

---------------------------------------------------------------------------------------------------------------------

Here you can see the login node port which is 1. (local:1 this could also be zero). Make a note of this because it is needed for the next step. In case you haven’t set a password for vnc, you will be asked to create a password for security purposes. You can alternatively set it or change it using the command `$vncpasswd`.

Step 5: Change some options in Putty now. Click on the Putty icon on the top left corner of the Putty window. Go to Change Settings. Now go to Connections and then SSH option. Select Tunnels. Here you enter a source port (any number greater than 1024) and the destination would be the IP address of the compute node followed by the login port i.e. 590x where X is either 1 or 0. Once you’ve entered the source port and destination click on Add and then Apply. The destination address should look like 10.0.xxx.xxx:590x. Remember to put a colon (:) between ip address and port number.

![PuTTY Reconfiguration](image)

Step 6: Now run the vncviewer of your choice. Enter the vncserver details. I have used tightvnc here but you can choose the vnc of your choice. You enter the remote host address as

```
localhost:<source_port>
```

Here the source_port is the source port you entered in your Putty settings. Once you run that a popup will ask you to verify login. Here you enter
your vnc password that you set before.

You are good to go now. You can see a display and run the module of your choice now.

**Running VNC on Linux**

**Step 1: ssh into Prince**

**Step 2: qsub -l**

You will be assigned a compute node for that particular job.

**Step 3: Setting up the VNC**

a) Enter the command:

```
vncserver
```

New 'compute-8-7.local:1 (trc311)' desktop is compute-8-7.local:1

Starting applications specified in /home/trc311/.vnc/xstartup

Log file is /home/trc311/.vnc/compute-8-7.local:1.log

b) If you haven’t set up a vncserver before it might prompt you to setup a password. Enter the password and remember it. Or you can set it up using the command $vncpasswd

c) Type the command

```
hostname -i
```

This should return the ip address for your compute node.

```
10.0.254.76
```

----------------------------------------------------------------------------------------------------------
d) Once the vncserver has been setup, port forwarding has to be done. For this first go to a new line of the terminal and then you press the shift key + tilde (~) at once followed by shift key + C

```
ssh>
```

Something like the above should be seen in your terminal. If not go to a new line in the terminal and repeat the d)

Now type the following:

```
ssh> -L<port_no>:<compute node ip address>:590x
```

Here, the port_no has to be any number greater than 1024. Make sure the port is not being used by another vncserver, otherwise you will get an error. In 590x, the x is the login port of localhost i.e. 1 or 0. Make sure there are no spaces between the forwarding command.

Step 4: Now go to your VNC application and enter the server information as local host information.

```
localhost:<port_no>
```

For some applications ssh might be auto checked, remember to uncheck that option. Once vnc session starts it might prompt you for a password which should be the password you entered while setting up the vncserver.

Now open the terminal on the display i.e. the compute node and run the module that you need.