SSH tunneling overview

<table>
<thead>
<tr>
<th>Quick Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPC at NYU</td>
</tr>
<tr>
<td>HPC Newsletter</td>
</tr>
<tr>
<td>Getting started on Prince</td>
</tr>
<tr>
<td>Getting an account</td>
</tr>
<tr>
<td>Prince How-to Articles</td>
</tr>
<tr>
<td>Logging in</td>
</tr>
<tr>
<td>Windows</td>
</tr>
<tr>
<td>Mac / Linux</td>
</tr>
<tr>
<td>Clusters and Storage July 2017</td>
</tr>
<tr>
<td>Mercer (HPC, retired)</td>
</tr>
<tr>
<td>Prince (HPC)</td>
</tr>
<tr>
<td>Dumbo (Hadoop)</td>
</tr>
</tbody>
</table>
Overview

The HPC clusters cannot be accessed directly from the Internet - access is a two-step process:

1. Log in to `hpc.nyu.edu`
   This is a "bastion host", providing a secure gateway between the NYU HPC clusters and the Internet. You will find that on `hpc.nyu.edu` itself you can do very little save but to log in to an NYU HPC cluster

2. Log in to the cluster you wish to use
   This will put you on a "login" node for that cluster, from where you can manipulate files and initiate jobs. This is not the place for heavy compute work: you submit that as a job to the compute nodes via the batch queuing system. Instructions for using the batch system can be found [here](#) (but you don't need to worry about that yet)

To log in you will use the program `ssh` from a terminal window. If you are unfamiliar with the command line interface this may seem daunting - relax, it's easy and vastly more powerful than point-and-click. We have a basic tutorial [here](#).

If you wish to use any software with a graphical interface, your Mac or Linux workstation must be configured to handle remote windows. This is a standard feature of Linux, but **Mac users will first need to download and install an X server** such as [XQuartz](#).

For easier access and to transfer files between your workstation and the clusters, you will eventually want to set up **SSH tunneling**

Logging In - the easy, primitive way

Now you are ready to begin the two-step process.

In the boxes below and elsewhere in this wiki, the symbol "$" at the beginning of a line represents the command prompt - don't type the "$"; type only the remainder of the line following the "$". Also wherever `NetID` appears, replace it with your NYU NetID.

1. **Log in to `hpc.nyu.edu`**
   The `-Y` option allows X forwarding, that is, GUI applications running on the cluster can draw windows on your screen (as long as you have an X server such as [XQuartz](#) installed)

   ```
   $ ssh -Y NetID@hpc.nyu.edu
   ```
2. From there, log in to the cluster you wish to use
The clusters are named as follows:

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Host name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercer</td>
<td>mercer.es.its.nyu.edu</td>
</tr>
<tr>
<td>Babar</td>
<td>dumbo.es.its.nyu.edu</td>
</tr>
</tbody>
</table>

Therefore, log in using one of the following commands, according to which cluster you wish to use:

$ ssh -Y NetID@mercer.es.its.nyu.edu
$ ssh -Y NetID@dumbo.es.its.nyu.edu

On Mercer you may notice that you are now on a host named "login-0-0" or "login-0-3" or something similar. The cluster uses multiple login nodes and which one you get depends how busy each is at the time. The login nodes are configured identically and see the same filesystems, so the specific node you are logged in to is not important.

Setting up SSH Tunneling

In computer networking, a computer decides what to do with an incoming network packet according to the "port" it arrived on. The port is simply a number attached to the packet. Certain ports are reserved for specific functions, for example packets arriving on port 22 are assumed to be intended for the SSH handler, so the computer passes those packets to SSH to interpret. Other port numbers are available to use for whatever you like, and as long as the same port is not used for different things on the same computer, everything works.

With SSH Tunneling, you will start an SSH session between your workstation and the bastion host hpc.nyu.edu, and instruct that session to create a tunnel. Your workstation will make one end of the tunnel, at "localhost, port 8023" ("localhost" is the computer's name for itself, so packets arriving at your workstation port 8023 will be sent into the tunnel). The bastion host will make the other end of the tunnel, at "mercer.es.its.nyu.edu, port 22", so anything coming through the tunnel will be forwarded to the normal SSH port (22) of Mercer. The fact that your workstation cannot see Mercer does not matter, it only needs to see its end of the tunnel.

The following diagram illustrates the process. It looks complex, but only requires 2 steps: the blue text shows what happens when you create the tunnel (step 1) and the green arrows indicate using the tunnel (step 2).
You only need to do step 1 once, and then you can use the tunnel (step 2) as many times as you like - for example, you might have two terminal sessions and a WinSCP session all using the same tunnel created with step 1.

In these instructions we are using port 8023. If it happens that another program on your computer is watching this port (which is fairly unlikely) then it won't work, and you'll need to choose a different port number, eg 9020, and substitute that throughout these instructions. 4-digit numbers starting with an 8 or a 9 are usually good ones to choose.

(Mac only) Preparing your Mac for SSH Tunneling

Recent versions of OSX do not require changes to System Preferences, so first try skipping to Setting up a tunnel you can reuse first. If you are then unable to connect through your tunnel, try the 4 steps below.

We will be instructing your Mac to forward certain incoming packets to a tunnel, so first the Mac must be willing to accept the incoming packets at all. To enable this:

1. Open System Preferences and click Sharing.
2. Select the Remote Login checkbox.
3. Return to system preferences and click “Security” and then “Firewall options” (if you are using Mavericks) or “Advanced” (for older versions of OSX)
4. Uncheck “Block all incoming connections”.

We have a video guide of this process.

Also, make sure you have prepared your Mac for X in accordance with the instructions above.

Note that Linux users do not need to do this: remote logins are enabled by default under Linux.

Setting up a tunnel you can reuse (the best approach)

To avoid repeatedly setting up a tunnel, we write the details of the tunnel into your SSH configuration file. This is found in the hidden ".ssh/" directory under your home directory. To access it, first open a Terminal window.

The process is the same for Linux and Mac, and is demonstrated in the first minute and a half of this video guide (the remainder of the video demonstrates transferring files over the tunnel, which is covered here).

To see the contents of a directory, enter "ls -la" at the command prompt. The "a" is important, without it files and directories starting with "." will be hidden. We are looking for a directory called ".ssh".
If you do not have a ".ssh" directory, create one as follows. The permissions of this directory are important, hence the `chmod` command.

```bash
mkdir ~/.ssh
chmod 700 ~/.ssh
```

Using your favorite editor, open the file ".ssh/config". If you are not familiar with Unix-ish editors, we have some help [here](#).

Add the following lines to ".ssh/config:

```plaintext
# first we create the tunnel, with instructions to pass incoming packets on ports 8020, 8021 and 8022 through it and to specific locations
Host hpctunnel
  HostName hpc.nyu.edu
  ForwardX11 yes
  LocalForward 8023 mercer.es.its.nyu.edu:22
  # you can add other hosts to the tunnel too - for example, to specifically use login node 1 on mercer:
  LocalForward 8024 mercer1.es.its.nyu.edu:22
  # for the Hadoop cluster:
  LocalForward 8025 dumbo.es.its.nyu.edu:22
  User NetID

# next we create an alias for incoming packets on port 8023. The alias corresponds to where the tunnel forwards these packets
Host mercer
  HostName localhost
  Port 8023
  ForwardX11 yes
  User NetID

# repeat the last step for each LocalForward entry under "Host hpctunnel", eg:
Host mercer1
  HostName localhost
  Port 8024
  ForwardX11 yes
  User NetID

Host dumbo
  HostName localhost
  HostKeyAlias dumbo
  Port 8025
  ForwardX11 yes
  User NetID
```

Starting the tunnel

To create the tunnel, ssh to it with the following command:

```bash
$ ssh hpctunnel
```

**Important:** you must leave this window open for the tunnel to remain open. It is best to start a new terminal window for subsequent logins.
Logging in via the tunnel

1. Open a new terminal window

2. Use ssh to log in to the cluster, as shown below. Note that you must use the short name defined above in your .ssh/config file, not the fully qualified domain name

   $ ssh mercer

   In the command above we have not used the -X flag to ssh. This is no longer needed because we already specified "ForwardX11 yes" in the .ssh/config file.

Creating a once-off tunnel (not the best approach)

You can set up a once-off tunnel without editing .ssh/config by running the following command:

   $ ssh -L 8023:mercer:22 NetID@hpc.nyu.edu

This is the equivalent to running "ssh hpctunnel" in the reusable tunnel instructions, but the port forwarding is specified on the command line. You can setup tunneling to a second cluster (eg Babar) by adding a second -L option to the command line.

Logging in via a once-off tunnel

However this does not create the convenient alias, so when connecting with ssh, or scp or rsync, you must explicitly connect to that port on your workstation (localhost):

1. Open a new terminal window

2. Use ssh to log in to the cluster by logging into "localhost" on the appropriate port, as shown below.

   $ ssh -X -p 8023 localhost

   With a once-off tunnel we do need the -X flag to ssh, and also the -p flag (lowercase) to specify the port number.

Logging Out

At the end of your session, you can log out of a cluster, and of hpc.nyu.edu, with the command:

   $ exit

   Don't forget to also log out from the session on hpc.nyu.edu, which you opened to start the tunnel!