Remote access session on NoMachine or FastX (and add aliases to your .bashrc file)

This page is meant as a primer for initiating remote access session to hutch laser controls systems. If the desire is to access accelerator-side controls systems such as those for the LCLS-I and LCLS-II photoinjectors, the following page is also recommendable: Remotely connecting to S20 laser controls and diagnostic (for on-shift QLOs)

TLDR

When working off-site, it may be important access the critical controls and computing architecture needed to support hutch activities. NoMachine , an application-based tool, and FastX, a browser-based utility, are two different ways to get this access. This brief tutorial walks through some of the basic steps of use.

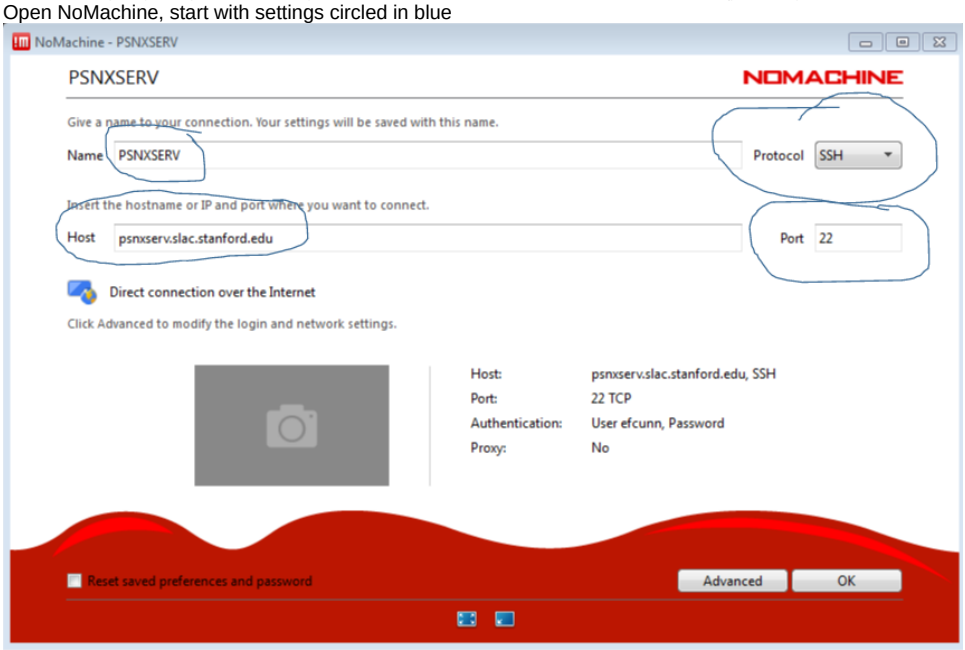
Note also that MobaXterm is another well-recommended remote access platform that, while not shown in this tutorial, is often preferred above NoMachine for its stability and added conveniences.

Before getting started: remote access permissions

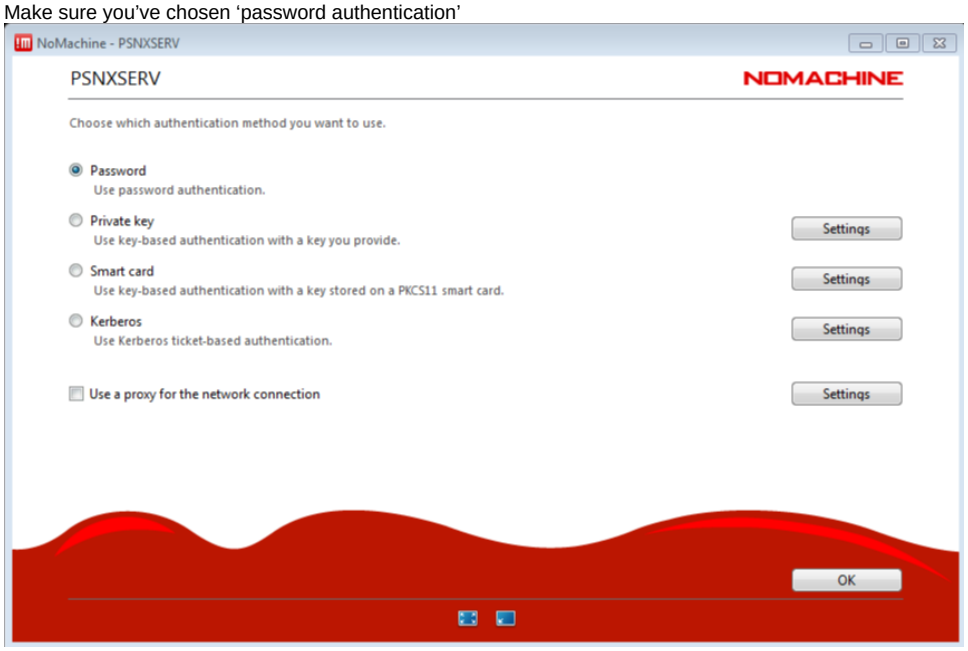
If this is your first time ever accessing SLAC's networks, you may need to review the procedure for securing the correct access permissions to the networks needed: Unix account permissions for accessing SLAC networks and remote machines

NoMachine

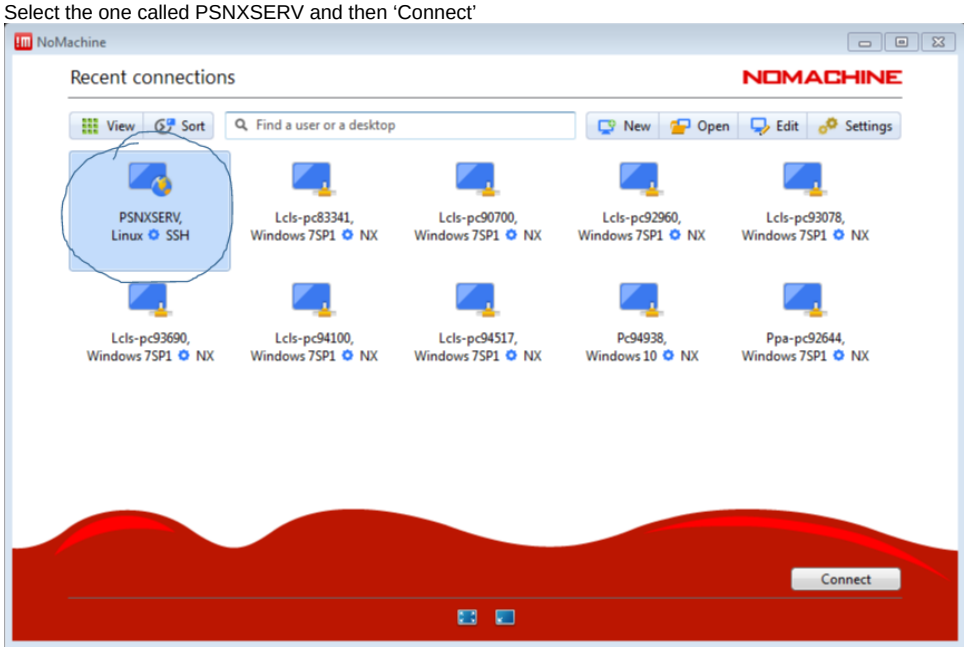
- Before doing anything, make sure you start with a machine on the SLAC network (you may need to VPN using Cisco AnyConnect)
- Open NoMachine, start with settings circled in blue



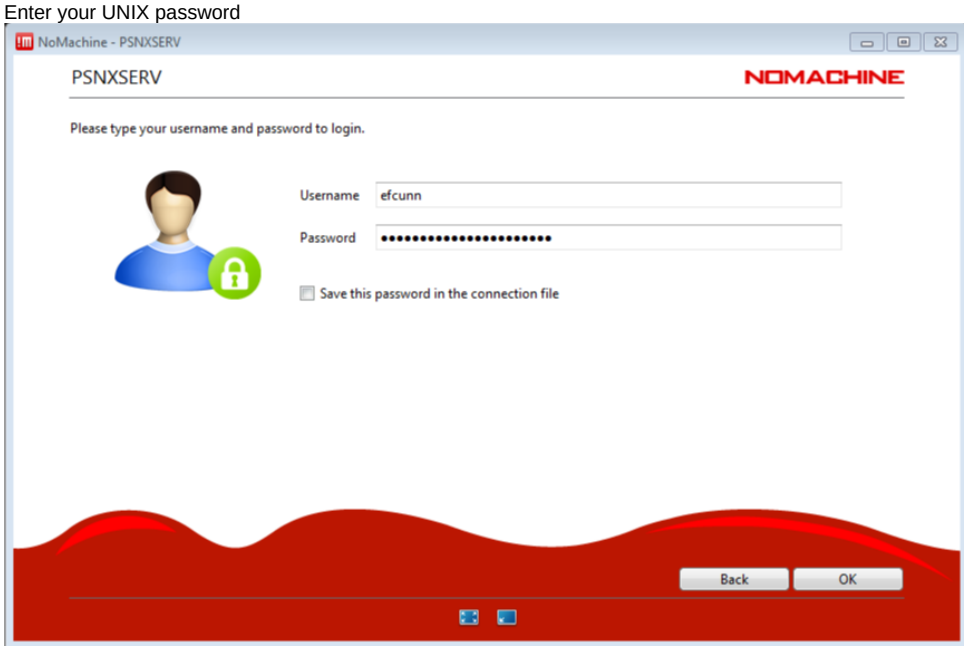
- Make sure you've chosen 'password authentication'



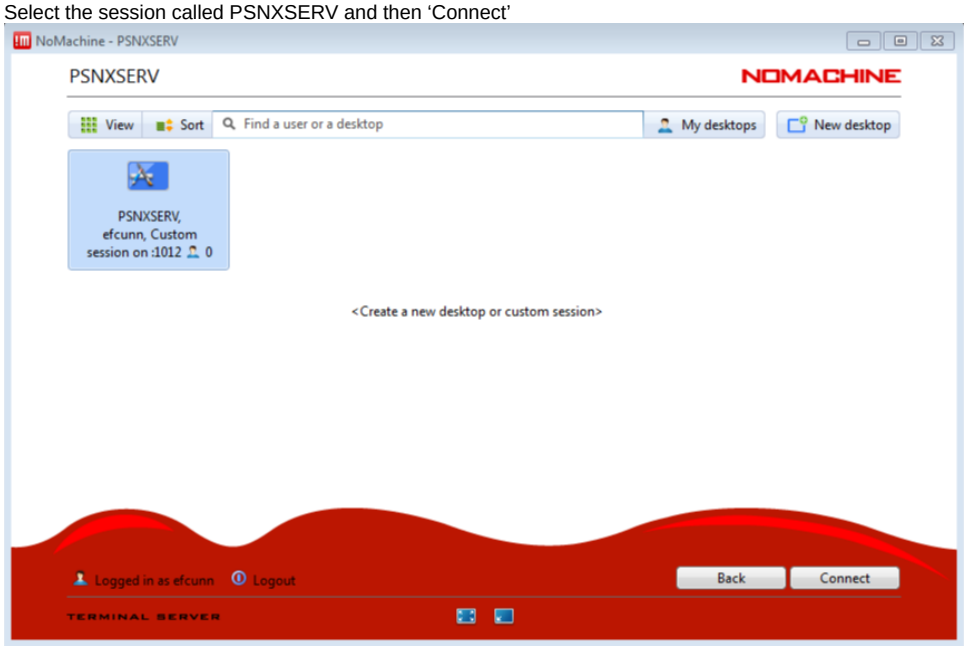
- Select the one called PSNXSERV and then 'Connect'



- Enter your UNIX password



- Select the session called PSNXSERV and then 'Connect'



- You will be presented with a terminal window.

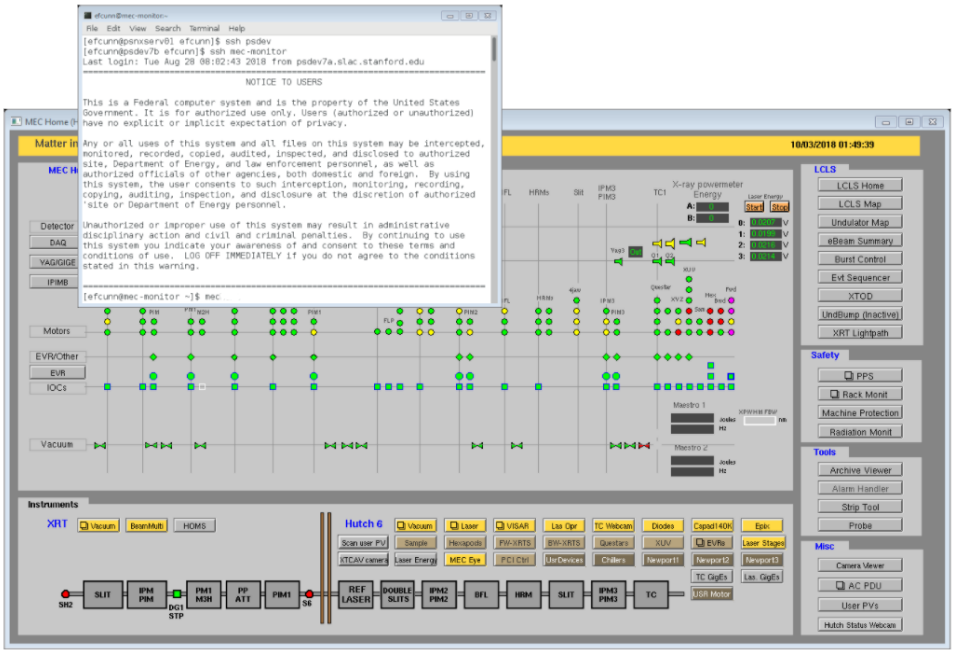
  Type the following to tunnel and connect to `psdev`, which is the main hub for connecting to hutch machines:

  ```
  ssh psdev
  ```

  Wait until you've connected, then connect to the computer you want (e.g. `mec-monitor`) using `ssh`:

  ```
  ssh mec-monitor
  ```

  From there, for example, you can run MEC Home, MEC Python, VNC Viewer, or whatever!
  (Note that instead of tunneling through `psdev` to an LCLS machine, it may occasionally be important to access `mcclogin` and and `physics@lcls-srv01` instead for e.g. supporting work at the LCLS-I photinjector.)



.bashrc trouble

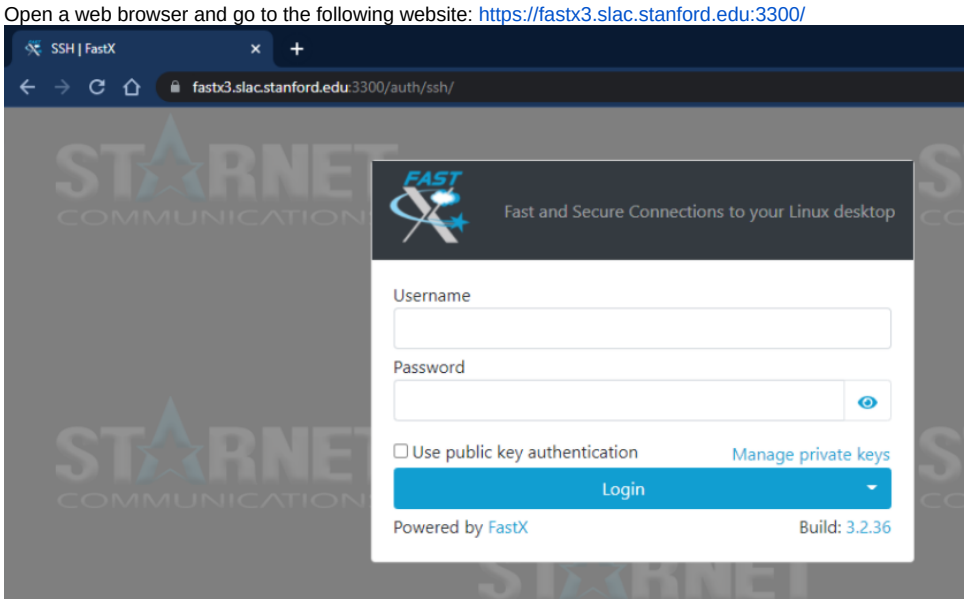
NOTE: If you follow the steps above, but the terminal says something like `-bash-4.2$` instead of something like `[username@hostname ~]$` then there's a good chance that you won't be able to type in the shortcut command you're used to using to open a controls GUI (e.g. `las` for launching the NEH Laser Hall Laser Home Lucid screen or `mec` for launching the MEC Home EDM screen).

```
-bash-4.2$
```

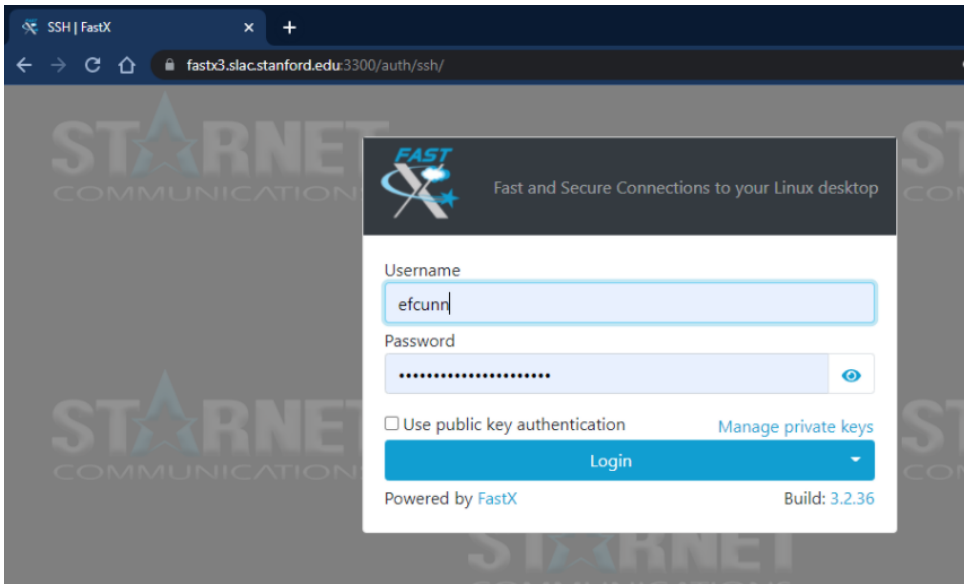
If this is the case, you'll need to make sure you follow the directions below about configuring your `.bashrc` file (or, alternatively, figure out why your `.bashrc` file isn't being sourced correctly). Until then, you still won't be able to get what you want 🙂.

FastX

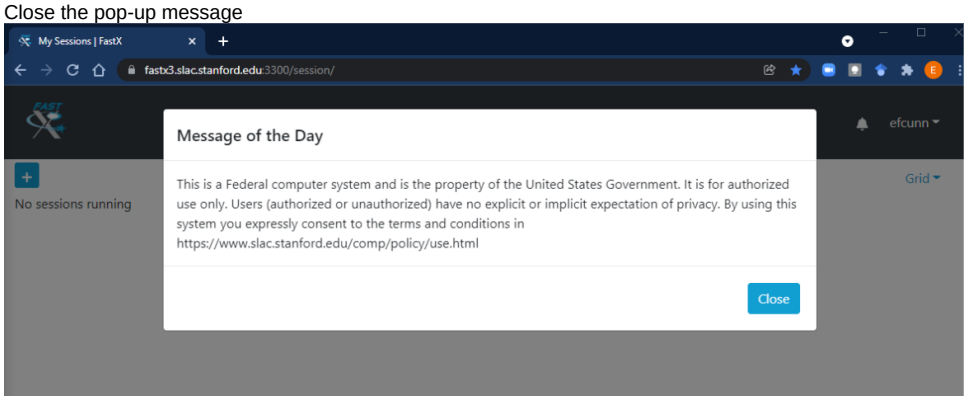
- Open a web browser and go to the following website: https://fastx3.slac.stanford.edu:3300/



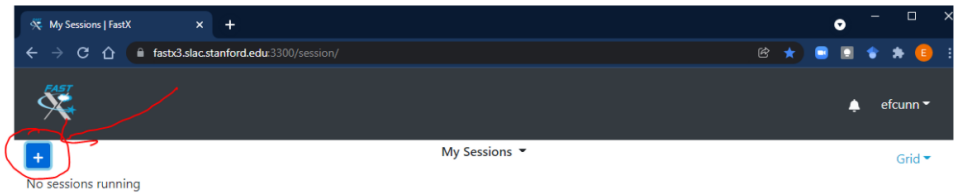
- Log in using your Unix account credentials



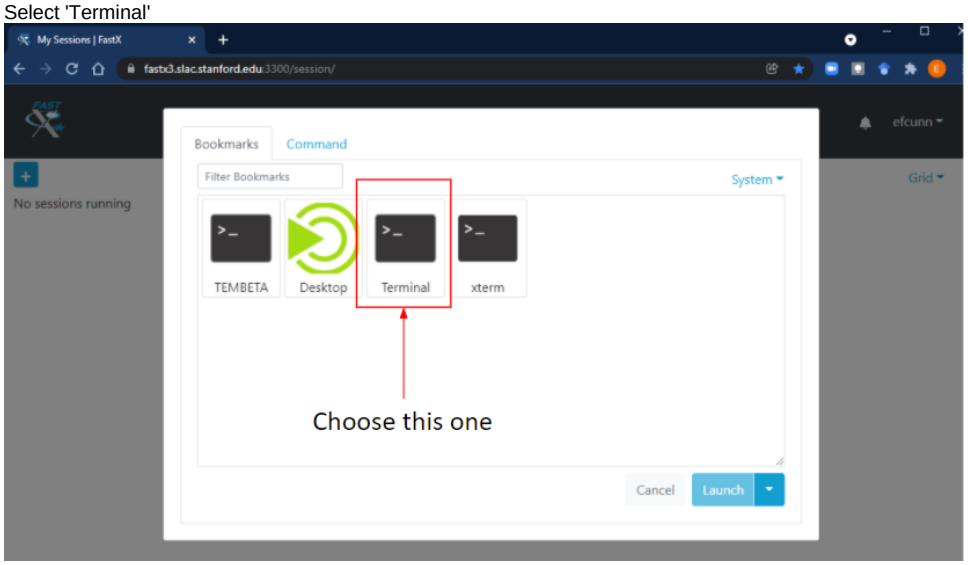
- Close the pop-up message



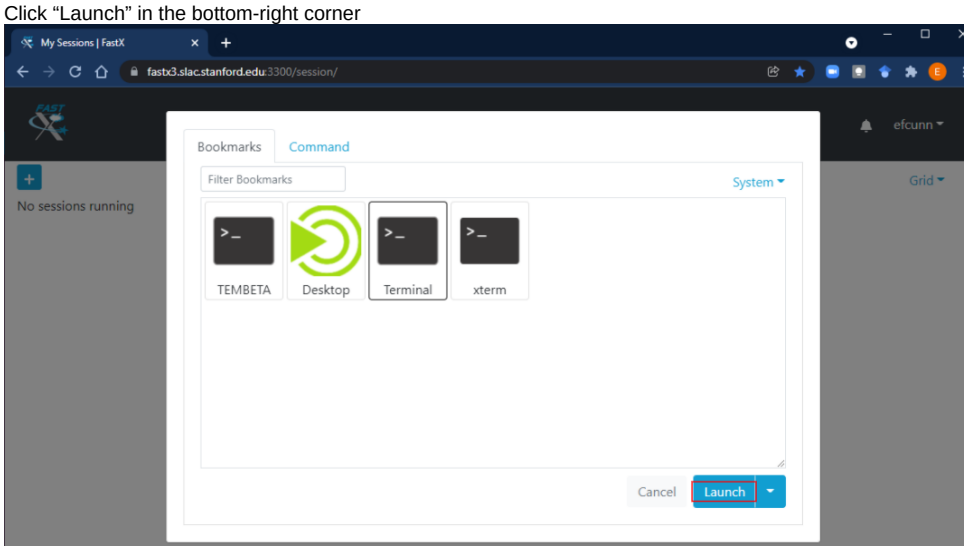
- Start a new session by clicking the '+' sign



- Select 'Terminal'



- Click “Launch” in the bottom-right corner



- You will be presented with a terminal window.

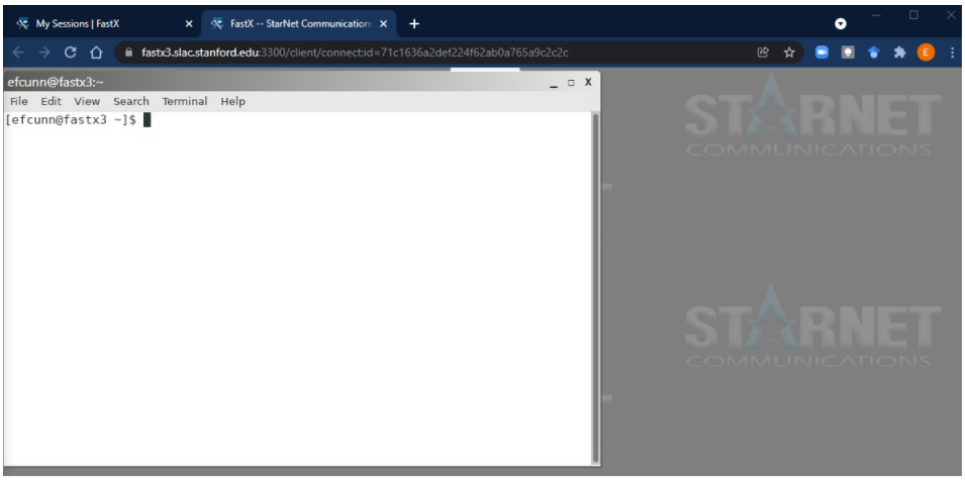
  Type the following to tunnel and connect to `psdev`, which is the main hub for connecting to hutch machines:

  ```
  ssh psdev
  ```

  Wait until you've connected, then connect to the computer you want (e.g. `mec-monitor`) using `ssh`:

  ```
  ssh mec-monitor
  ```

  From there, for example, you can run MEC Home, MEC Python, VNC Viewer, or whatever!
  (Note that instead of tunneling through `psdev` to an LCLS machine, it may occasionally be important to access `mcclogin` and and `physics@lcls-srv01` instead for e.g. supporting work at the LCLS-I photinjector.)



Editing your alias shortcuts in your .bashrc file

If it's your first time using the terminal line under your own log-in credentials (whether using FastX, NoMachine, etc.), you'll need to define some commands in order to run the utilities you want.

To do so, from your terminal type `gedit ~/.bashrc` (or otherwise open the `.bashrc` file in another favorite text editor like `emacs` or `vim`). This will bring up a new window.

Copy the text below and paste it into the file you've opened. [Note: the NoMachine windows share the same clipboard as your desktop, though this may be trickier using FastX.] Once successful, click 'save' and exit the text editor. Now from the terminal, you should be able to do things like launch MEC Home, mecpython, VNC, etc.

If you're still having trouble, try typing `source ~/.bashrc` or try exiting your session completely and starting with a fresh terminal to make sure it sees the new definitions.

WARNING: THERE MAY BE SPECIAL CHARACTERS PROBLEMS WHEN COPYING AND PASTING – WATCH OUT FOR PROBLEMS!!

SEE ALSO: https://github.com/pcdshub/shared-dotfiles and https://github.com/pcdshub/shared-dotfiles/blob/master/on_site/bashrc !!

----------Copy everything below this line into your .bashrc file----------

```
# .bashrc

# Source global definitions

if [ -f /etc/bashrc ]; then

  . /etc/bashrc

fi

#### PATH definitions starts ####
# clear the path before assigning relevant values
export PATH=$PATH
export PATH=/reg/common/package/python/2.7.2/bin:$PATH # commented out by Zhou
export PATH=/reg/g/pcds/package/epics/3.14/base/current/bin/linux-x86/:$PATH
export PATH=/reg/g/pcds/epics-dev/screens/edm/cxi/current/:$PATH


#snelson: added what we have in xpp

export PATH=/reg/neh/operator/mecopr/bin:${PATH}

export PATH=/reg/common/tools/bin:${PATH}

export PATH=/reg/g/pcds/engineering_tools/mec/scripts:${PATH}

export PATH=/reg/g/pcds/pyps/apps/iocmanager/latest:${PATH}

#### PATH definitions ends   ####

##################################################
```

```
### Basic PCDS environment setup for all logins ###
#################################################
source /reg/g/pcds/setup/pathmunge.sh
if [ -f "${HOME}/.pcds_setup.sh" ]; then
    source "${HOME}/.pcds_setup.sh"
fi
source /reg/g/pcds/setup/epics-ca-env.sh
source /reg/neh/home/sioan/setupEpics &> /dev/null
source /reg/g/pcds/setup/epicsenv-cur.sh &> /dev/null
export PSPKG_ROOT=/reg/g/pcds/pkg_mgr
export PSPKG_RELEASE="sxr=3.0.0"
PATH="$PATH:/reg/common/tools/bin"
```

Final note:

This page is also mirrored here on the LCLS Laser Confluence: How to remotely connect to the EPICS control system for lasers

Related articles

- How to make a new recipe for the LPL
- Remote access session on NoMachine or FastX (and add aliases to your .bashrc file)
- Fully power down the SPL
- Cold start the SPL
- New big compressor alignment procedure