THE UNIVERSITY OF CHICAGO | Research Computing Center

Intro to High Performance Computing

Software Carpentry Python in HPC module

Accessing Midway: Mac/Linux Users

Midway logins:

midway.rcc.uchicago.edu	Midway I
midway2.rcc.uchicago.edu	Midway 2

- **Mac/Linux users:**
  Open the terminal app: Applications – utilities -> terminal
  Drag it to your dock so that it is easy to locate in future.

```
w/o X11 [jhskone@volpe]$ ssh $USER@midway.rcc.uchicago.edu

X11 enabled: [jhskone@volpe]$ ssh -X $USER@midway.rcc.uchicago.edu
```

- **Mac users – enabling X11 forwarding**
  Mac users will not have XQuartz X server installed by default. You will need to download and install X11 server and client from the XQuartz project page.
  NOTE: Users are encouraged to install the older 2.7.8 version of XQuartz, if they intend to run OpenGL applications (e.g. VMD, gaussview, etc).

Accessing Midway: Windows Users

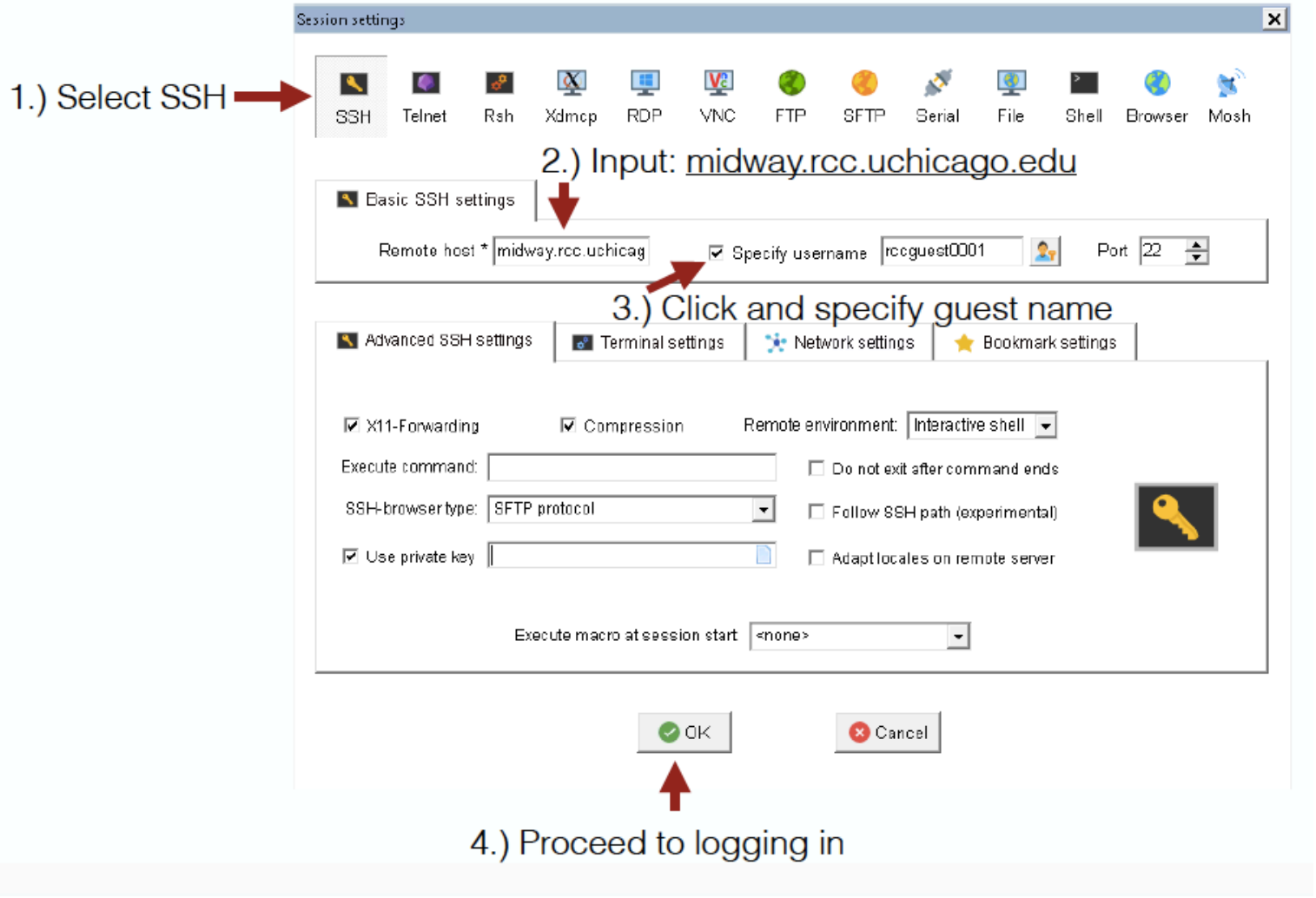
Midway logins:

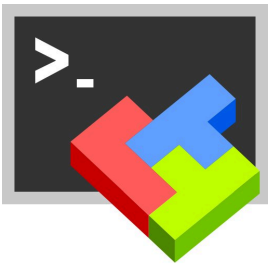
midway.rcc.uchicago.edu	Midway I
midway2.rcc.uchicago.edu	Midway 2

- **Windows users:**
  Download either mobaxterm or cygwinX if you don't have it already.
  Strongly encourage windows users to use mobaXterm even if you have previously used putty.
  Windows10 users can use the Windows subsystem for Linux – activate dev mode.

- **MobaXterm**
  Comes bundled with X Server so it is X11 forwarding capable.
  Also has an sftp tab built in to the client permitting easy drag and drop file transfer from remote to local machine.
  To get MobaXterm click the Download tab at top of their site and choose the "Free Home Edition" of MobaXterm.
  Download the MobaXterm Home (Installer edition) zip file and extract contents.

Windows Users: MobaXterm client





Accessing Midway via GUI

We use the Thinlinc Client.
You can either access the webversion of this or download the client

Access from browser the following:
https://midway-login1.rcc.uchicago.edu/main/

This is the safest bet for displaying applications that require X-forwarding if you are having trouble working with your ssh X11 forwarding enabled session.

From terminal X forwarding can be achieved with –X or –Y For example:

```
[jhskone@volpe]$ ssh -Y $USER@midway.rcc.uchicago.edu
```

For Linux distributions X Server is typically installed by default.
For Windows users using MobaXterm, the default ssh connection is with X11 forwarding enabled.

Accessing Materials on Midway

This class has access to the /project/rcc/jhskone/sw-carpentry. This is where materials for this lecture will be located.

Your first task is to copy & untar the HPC-python tar file to your home directory:

```
[jhskone@midway1]$ cd /home/$USER

[jhskone@midway1]$ cp /project/rcc/jhskone/HPC-python.tar.gz ./

[jhskone@midway1]$ tar –xzvf HPC-python.tar.gz

[jhskone@midway1]$ cd HPC-python
```

Will paste the path/filename into etherpad

Module System

HPC centers typically use a software module system to manage the software packages that loaded into your environment.

This is useful in that you don't have to install the software your self and can selectively choose which software packages are accessible to you so that you can possibly avoid software conflicts.

Useful Module Commands

```
[jhskone@midway1]$ module help       # information about using module

[jhskone@midway1]$ module list       # list your currently loaded modules

[jhskone@midway1]$ module avail      # list all avail software packages

[jhskone@midway1]$ module load  <package>  # load <package> into your env

[jhskone@midway1]$ module unload <package> # unload <package> from env
```

Using Midway Module System

- To use a particular software package load the package/version name. The default version will load if the version is not specified

```
module load python
module list
```

- The module load command appends to your $PATH and $LD_LIBRARY_PATH the locations of the compilers and their accompanying libraries.

```
echo $PATH; echo $LD_LIBRARY_PATH
```

- Notice that not just the python module has been loaded. Any other software that the module you load is dependent upon will also be loaded. Keep this in mind as this can potentially lead to conflicts with other software you may load/use. List info about the module package:

```
module show python
```

Python Modules on Midway

There is a plethora of python modules on midway. How then to choose which to use?

```
[jhskone@midway1]$ module avail python

------------------------ /software/modulefiles ------------------------
python/2.7                              python/2.7-2015q1
python/3.3python/2.7-2013q4             python/2.7-2015q2(default)
python/3.4-2015q1python/2.7-2014q1      python/2.7.12+gcc-4.7
python/3.5.2+gcc-4.8python/2.7-2014q2   python/2.7.12+gcc-6.1
python/3.5.2+intel-16.0python/2.7-2014q3 python/2.7.12+intel-16.0
---------------------------- /etc/modulefiles ------------------------
```

```
[jhskone@midway1]$ module avail Anaconda2
   Anaconda2/4.1.1(default)

[jhskone@midway1]$ module avail Anaconda3
Anaconda3/4.1.1(default)
```

Python Modules on Midway

There is a plethora of python modules on midway. How then to choose which to use?

- There are two main different python versions (python2.7 and python3.x) which are indicated in the module version.

- Some installs are built with different system compiler (gcc, intel) This is important if coupling your python code with other compiled software (want to have homogeneity in compilers/libs). It can have influence on speed of linear algebra operations. For example those python versions built with intel mkl will likely have faster numpy routines.

- Not all python modules have the same packages. You can check this with pip or conda (Anaconda distribution only).

Running a python code from CLI

- The first exercise (exercise0) is to obtain an interactive session:

```
sinteractive --partition=sandyb –time=0:30:00
```

- Then we will run the python code in example0.py

  which python

```
python example0.py
```

- We can access the same compute node we have an interactive session started on by ssh'ing from midway-login to the compute node hostname. Open separate terminal and do so.

Installing Python Packages

- People will commonly request that they have xyz python module installed.
- Users can do this themselves with pip.

```
module load python/versionXXX
pip install –user <package>
```

- Will install it by default to ~/.local unless you specify PYTHONUSERBASE

- Users can also use the virtual environment

```
virtualenv <Directory to store python environment>
cd <dir store python env>
source activate
```

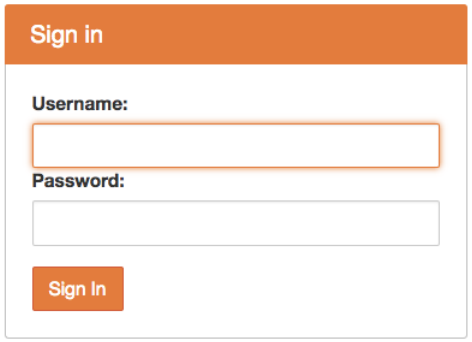
Running Jupyter Notebooks on Midway

1.) From the login nodes: (ssh **–Y** midway.rcc.uchicago.edu)

```
[jhskone@midway1]$ module load python

[jhskone@midway1]$ jupyter notebook
                      OR
[jhskone@midway1]$ jupyter notebook <file.ipynb>
```

2.) Accessing the jupyterhub portal: https://jupyter.rcc.uchicago.edu



Username is your CNetID
Password is your CNetID password

You will land in /home/$USER directory upon login

Inspecting Your Submitted Job

- The user can directly login to the node that their job is running on and observe the progress of the job.

```
squeue –u <userid>
```

Will return job ids of jobs you have running or pending.
Get a running jobs list of nodes it is using:

```
squeue -O nodelist -j <job_id>
```

Login directly to that node and run commands `top` and `free –g`

If the job is not running check its priority:

```
squeue -O prioritylong -j <job_id>
```

```
squeue -p sandyb –O jobid,numnodes,reason,timelimit,prioritylong
```