New User Tutorial

Using a Terminal

Most computer users are familiar with a graphical user interface (GUI). A GUI consists of visual icons and directories that are navigated using a mouse. Most high performance computing (HPC) systems use a command-line interface because it allows a user to efficiently do many common HPC tasks. While there is a steep learning curve associated with using a terminal, it has a set of features that allows researchers to automate tasks, saving time and money.

Logging into Cowboy

In order to log into Cowboy, you need to use a terminal with Secured Shell (SSH). After logging in, you will be on one of the cluster’s login nodes. Node refers to a server or computer. Cowboy has two login nodes, and 254 compute nodes. A private network connects the compute nodes together. A cluster consists of these nodes, the network, and a file system.

Windows

Most Windows operating systems do not come with an SSH terminal application. Putty is a free, open-source terminal emulator that supports SSH.

1. Download Putty from the Putty website.
2. Choose the file "putty.exe," save it to your desktop, then run it.
3. A window will open that looks like the image below.
4. Enter `cowboy.hpc.okstate.edu` in Host Name text box.
5. Verify Connection type is "SSH," and Port number is 22.
6. In Saved Sessions, you can name these settings, e.g. Cowboy, and click Save.
7. Next time you open PuTTY, you can double-click this saved session and it will open a login screen directly.
8. After saving your settings, click Open. Enter your Cowboy username and press the Enter key.
9. Enter your password, followed by the Enter key. Passwords are case sensitive.

    Nothing will display while you type your password. This is a security feature.

10. To end your session, type `exit` or `logout`, followed by the Enter key.

**Linux**

Linux operating systems already have a terminal application. You can launch a terminal from your computer's application menu.

1. In a terminal, type `ssh username@cowboy.hpc.okstate.edu`, where `username` is your username.

    If your username is `pete`, you would enter `ssh pete@cowboy.hpc.okstate.edu`.

2. Enter your password, followed by the Enter key. Passwords are case sensitive.

    Nothing will display while you type your password. This is a security feature.

3. To end your session, type `exit` or `logout`, followed by the Enter key.

**Mac**

Macintosh operating systems already have a terminal application.

1. Open your Finder (hard drive) window.
2. Double-click on the following sequence of icons: Applications folder, Utilities folder, Terminal icon.
3. In the terminal, type `ssh username@cowboy.hpc.okstate.edu`, where `username` is your username.

    If your username is `pete`, you would enter `ssh pete@cowboy.hpc.okstate.edu`.

4. Enter your password, followed by the Enter key. Passwords are case sensitive.

    Nothing will display while you type your password. This is a security feature.

5. To end your session, type `exit` or `logout`, followed by the Enter key.

**Using your mouse**

Your mouse has no function within the terminal. Apart from copy and paste commands, you can only use your keyboard.

**Copy**
### Changing Your Password

It is important to change your password when you log in for the first time. Change your password using the `passwd` command.

1. After logging in, type `passwd`.
2. Enter your old password, then your new password twice.
3. Please use a strong password. Passwords should be at least eight characters long and include numbers, letters, and special characters.

### Linux Commands

Below are Linux commands and terminal shortcuts that are useful on Cowboy.

#### Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>pwd</td>
<td>Print Working</td>
<td>Displays your current location.</td>
</tr>
<tr>
<td></td>
<td>Directory</td>
<td></td>
</tr>
<tr>
<td>ls</td>
<td>List</td>
<td>List all the files and directories within a directory.</td>
</tr>
<tr>
<td>cd</td>
<td>Change Directory</td>
<td>Moves the user to a different directory.</td>
</tr>
<tr>
<td>cp</td>
<td>Copy</td>
<td>Copies a file.</td>
</tr>
<tr>
<td>Command</td>
<td>Action</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>mv</td>
<td>Move</td>
<td>Moves a file (also renames a file).</td>
</tr>
<tr>
<td>rm</td>
<td>Remove</td>
<td>Permanantly deletes a file.</td>
</tr>
<tr>
<td>grep</td>
<td>Grep</td>
<td>Searches a file's contents.</td>
</tr>
<tr>
<td>nano -w</td>
<td>Nano</td>
<td>Is a text editor.</td>
</tr>
<tr>
<td>diff</td>
<td>Diff</td>
<td>Compares differences between files.</td>
</tr>
<tr>
<td>tar</td>
<td>Tar</td>
<td>Creates file archives.</td>
</tr>
<tr>
<td>man</td>
<td>Manual</td>
<td>Allows users to look up manual entries.</td>
</tr>
<tr>
<td>exit</td>
<td>Exit</td>
<td>Ends the session.</td>
</tr>
</tbody>
</table>

Pipe
The pipe character is placed between two commands. It "pipes" the first command's output into the second command's input.

Terminal Shorthand

<table>
<thead>
<tr>
<th>Shorthand</th>
<th>Terminal's Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>..</td>
<td>One directory up from the current directory.</td>
</tr>
<tr>
<td>.</td>
<td>The current directory.</td>
</tr>
<tr>
<td>~</td>
<td>The user's home directory.</td>
</tr>
</tbody>
</table>

Key Combinations

<table>
<thead>
<tr>
<th>Key Combination</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tab</td>
<td>Completes partially typed commands or file names.</td>
</tr>
<tr>
<td>The up arrow</td>
<td>Cycles through previous commands.</td>
</tr>
<tr>
<td>Ctrl + d</td>
<td>Exits the terminal (same as typing exit).</td>
</tr>
<tr>
<td>Ctrl + c</td>
<td>Stops a running command.</td>
</tr>
</tbody>
</table>

The OSU HPCC website maintains a page of Linux tutorials.

File System

Each user has access to two directories: /home and /scratch. When you log in, your working directory is /home/username, where username is your username. /home/username has a quota of 25GB. To check your current usage, type du - sh. You should store your source code and executable files here.

OSU HPCC staff does not back up your home folder.

Use /scratch/username for large files and large file collections.

OSU HPCC staff does not back up your scratch folder. If you need archival storage, please contact Dana Brunson.

Users also have access to the /opt directory. /opt contains additional software.

If Cowboy does not have software you need, send us an email and request a software install.
File Transfers

Several options exist for moving files between your computer and Cowboy. Globus is a file transfer application for large data transfers. Smaller data transfers can utilize one of the GUI options below. If you prefer a command-line interface, please see the last set of directions in this section.

Globus

Globus is an online service which facilitates fault tolerant, large data transfers.

1. Set up your computer as a Globus endpoint by downloading the Globus Personal Client.
2. Create an account.
3. Verify the account and log into Globus.
4. Find the Manage Data drop-down list and click "Transfer Files".
5. Your screen will look similar to this one:

![Globus Transfer Files](image)

6. The Cowboy 'Endpoint' is `okstate\cowboy`.
7. Once you have entered both endpoints, click one of the arrows to start the transfer.

For more information, visit the [Globus website](#).

WinSCP (Windows)

1. Download "Portable executables" from the WinSCP Website.
2. The download is a zip archive with the file `WinSCP.exe`. Place `WinSCP.exe` on your computer's desktop. This is the only file you need.
3. Double-click the WinSCP icon.
4. A window like the one below will open:
5. In the Host name box, type `cowboy.hpc.okstate.edu`.
6. In the User name box, enter your username.
   If you want to save the above information, click Save. In the future, you can double-click the session name to directly log in.
7. Click Login.
8. You will see a window similar to the one below once you have connected.

9. The left side contains your files, and the right side has your Cowboy home directory.
10. You can drag and drop files in either direction.

### Cyberduck (Windows and Mac)

1. Download Cyberduck from their website
2. Click the Download icon below the duck.
   Warning! The Start Download icon on the right of the duck is an advertisement.
3. After downloading and opening Cyberduck, you should see a window similar to this:
4. Click the Open Connection icon at the top left.
5. Choose the "SFTP" (SSH File Transfer Protocol) option in the drop down menu at the top.
6. For Server, type coyboy.hpc.okstate.edu.
7. Change the Port to 22.
8. Enter your Cowboy username and password.
9. Click Connect.
10. Your Cyberduck window now looks like the picture below.
11. You can drag and drop files in both directions.

FileZilla (Windows, Linux, and Mac)

1. Download FileZilla from their website.
2. Click the Download FileZilla Client – All platforms icon.
3. After downloading and opening FileZilla, your window looks like the following image:
4. For Host, type `cowboy.hpc.okstate.edu`.

5. Enter your Cowboy username and password.
6. Change Port to 22.
7. Click Quickconnect.
8. Your window is now like the picture below.
9. You can drag and drop files in either direction.

Command-line File Transfers (Linux and Mac)

Copy a file from Cowboy to your local machine

1. Open a terminal on your local machine.
2. Navigate to the directory where you want to place the file.
3. Type
   
   `scp yourusername@cowboy.hpc.okstate.edu:directory_where_your_file_is_stored/yourfilename .`

   If Pete wanted to move the file `examplefile.txt` from his scratch folder to his local computer, he would type
   
   `scp pete@cowboy.hpc.okstate.edu:/scratch/pete/data/examplefile.txt .`

4. Enter your Cowboy password.
5. You will see the file name and “100%” once the transfer completes.

Copy a file from your local machine to Cowboy

1. Open a terminal on your local machine.
2. Navigate to the directory where your file is located.
3. Type
   
   scp localfilename yourusername@cowboy.hpc.okstate.edu:/directory_where_you_want_to_put_the_file/.

   If Pete wanted to put examplefile.txt in his /data folder on /scratch, he would type
   
   scp examplefile.txt pete@cowboy.hpc.okstate.edu:/scratch/pete/data/.

4. Enter your Cowboy password.
5. You will see the file name and "100%" once the transfer completes.

### Scheduler

When you log into Cowboy, you are located on one of Cowboy’s login nodes. You can edit files here, but you must use the scheduler to run your job. The scheduler takes information from you, finds the best compute node(s) to use, and runs your job.

![Scheduler Diagram]

The most effective way to use Cowboy is as follows:

1. Log in.
2. Save information about your job in a submission script.
3. Submit your submission script to the scheduler.
4. Log out and wait for your job to finish.

   You can set up your submission script so the scheduler sends you an email once your job has finished.

The following sections contain a guided exercise. This exercise will show you how to do each of these steps. You may find it useful to reference this tutorial’s list of Linux commands.

### Submission Scripts

Submission scripts give information about your job to the scheduler. Remember to use your username in place of pete in the following examples.

1. Change to the scratch directory and create a new directory called sample.

   $ cd /scratch/pete
   $ mkdir sample
   $ cd sample
   $ pwd
   /home/pete/sample

2. Copy an example submission script to your new directory.

   $ cp /opt/examples/helloworld.pbs .
   $ ls
   helloworld.pbs

3. Display the contents of this submission script by using cat.

   $ cat helloworld.pbs
   #!/bin/bash
   #PBS -q express
   #PBS -V
# specify the queue batch, express or bigmem
#PBS -l nodes=1:ppn=1
#PBS -l walltime=10:00
#PBS -j oe

cd $PBS_O_WORKDIR
module load examples
helloworld

Lines beginning with # are comments. They serve as your notes, but are ignored the computer. Any line that starts with #PBS is a special comment that gives information to the scheduler.

Submission Script Commands

<table>
<thead>
<tr>
<th>Line</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>#PBS -q express</td>
<td>Determines what queue your job will wait in before it begins.</td>
</tr>
<tr>
<td>#PBS -l nodes=1:ppn=1</td>
<td>Determine how many nodes, and processors per node (ppn), your job will need.</td>
</tr>
<tr>
<td>#PBS -l walltime=10:00</td>
<td>Determines your job's time limit.</td>
</tr>
<tr>
<td>#PBS -j oe</td>
<td>Please make sure all your submission scripts include this line. It will help HPCC staff assist you if your job has an error.</td>
</tr>
<tr>
<td>cd $PBS_O_WORKDIR</td>
<td>Tells the system where your job is located. Please make sure all your submission scripts include this line.</td>
</tr>
<tr>
<td>module load</td>
<td>Loads any needed software to run your job.</td>
</tr>
<tr>
<td>helloworld</td>
<td>The command to start your job. This command will change depending on your job.</td>
</tr>
</tbody>
</table>

Please try to accurately predict how long your job will take. If the wall time is too long, it may delay your job’s start time. If your job is taking longer than you expected, please email HPCC staff to request a wall time extension.

If you have questions about job queues, nodes and processors, or wall time limits, please email HPCC staff.

Submitting Jobs

Please submit all jobs to the scheduler. Your job will begin as resources become available. Since Cowboy is a shared resource, the scheduler determines when to begin your job. Do not run a job on a login node.

4. Continue the exercise by using `qsub` to submit your job.

```bash
$ qsub helloworld.pbs
```

This command submits the job to the scheduler and returns a job ID number.

5. To see all running jobs, enter the command `showq`.

6. To view only your jobs, add `grep` and `|` to your command.

```bash
$ showq | grep pete
```

7. Use `qpeek` and the job ID number to see the output of your job while it is running.

```bash
$ qpeek 414
Hello pete I am running on node n245.cluster
```
8. You can stop a running job by using `qdel` and the job ID number: `qdel 414`.

9. Once your job finishes, view the output by looking at the scheduler’s output file. It is the file with the naming convention `jobname.ojobidnumber`.

```
$ ls
helloworld.pbs.o414 helloworld.pbs
$ cat helloworld.pbs.o414
Hello pete I am running on node n245.cluster
waiting for 10 seconds
successfully finished!
```

**Nano**

Nano is a terminal text editor. You can use Nano to edit the example script so it fits your job’s needs.

10. Use `nano` to add a command telling Cowboy to email you once the job has finished.

```
$ nano helloworld.pbs
GNU nano 2.0.9 File: helloworld.pbs
#!/bin/bash
#PBS -q express
#PBS -l nodes=1:ppn=1
#PBS -l walltime=10:00
#PBS -j oe
cd $PBS_O_WORKDIR
module load examples
helloworld
```

11. Using the arrow keys, move the cursor to the end of the line `cd $PBS_O_WORKDIR`.
12. Press Enter to add a blank line and type: `#PBS -m aBe -M your.email@university.edu`.
13. Hold down Ctrl + x to exit nano.

```
Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
Y Yes
N No
```

14. Press `Y` to save.

```
File Name to Write: helloworld.pbs
```

15. Press Enter to keep the same file name.

**Additional Information**

This section explains more about job queues and "processors per node."

**Job Queues**

Job queues are where jobs wait to begin. Edit your script to use one of Cowboy’s four job queues.
Job Queues

<table>
<thead>
<tr>
<th>Queue</th>
<th>Wall Time Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch</td>
<td>120 hours (20:00:00)</td>
<td>The default queue.</td>
</tr>
<tr>
<td>express</td>
<td>1 hour (01:00:00)</td>
<td>For running short jobs and testing scripts. Has only two compute nodes.</td>
</tr>
<tr>
<td>bigmem</td>
<td>120 hours (20:00:00)</td>
<td>Jobs run on one of two compute nodes that have 256GB of RAM and a GPU card.</td>
</tr>
<tr>
<td>kiliable</td>
<td>504 hours (80:00:00)</td>
<td>For long jobs. HPCC administrators may stop jobs in this queue at any time.</td>
</tr>
</tbody>
</table>

Processors Per Node

Each compute node on Cowboy has 12 processors (cores) and 32 GB of RAM.

Applications that use a single node

If your code only uses a single core, it is better to request one processor.

```
#PBS -l nodes=1:ppn=1
```

If your code can use all 12 cores efficiently, request all 12 cores.

```
#PBS -l nodes=1:ppn=12
```

Here is an example script for a job that uses all 12 cores.

```
#!/bin/bash
#PBS -q batch
#PBS -l nodes=1:ppn=12
#PBS -l walltime=24:00:00
#PBS -j oe
module load <software_module_name>
cd $PBS_O_WORKDIR
<put the commands to run your application here>
```

Make sure to remove the enclosing brackets (<<>>) in the example. Don’t make the mistake of putting your module name and job commands inside them.

Applications that use multiple nodes

Here is an example submission script for a job that uses Message Passing Interface (MPI) to run on more than one node.

```
#!/bin/bash
#PBS -q batch
#PBS -l nodes=16:ppn=12
#PBS -l walltime=24:00:00
#PBS -j oe
module load mvapich2-1.8/intel
cd $PBS_O_WORKDIR
NP=$(cat $PBS_NODEFILE | wc -l)
mpirun -np $NP ./yourexecutable
```

Download Tutorial
Contact Information

Please contact us if you have any questions.

Email: hpcc@okstate.edu
Website: hpcc.okstate.edu