This is written as if you are interacting with BlueWaters from an Ubuntu operating system. This should also work with Mac. I don't own a Mac so some of the commands might be slightly different. In the past, I've seen maybe only 2 students use Windows and it required a lot of extra work on their part to figure it out. Let me know if there are any discrepancies and it'd be greatly appreciated.

Logging In

ssh user@bwbay.ncsa.illinois.edu

Use your provided trainee account in place of 'user'. You will be prompted for your password. After you login, you will be at a terminal indicated by something like user@h2ologin3:~>. This is strictly a login node. ***Do not attempt to run any programs here. This is not a compute node and it is against BlueWaters policy to do so.***

You will be in the home directory of your BlueWaters account and can type **ls** to see an empty ~/bin directory and ~/scratch directory.

Transfering Files

https://bluewaters.ncsa.illinois.edu/education-training-allocation-data-transfer

On your local machine, install openssh-server:

sudo apt-get install openssh-server

For mac Users:
https://superuser.com/questions/104929/how-do-you-run-a-ssh-server-on-mac-os-x

This allows you to access your local machine remotely from other servers. You will then be able to access any file on your local machine and transfer it over to BlueWaters. (This was necessary for my machine running linux but may not be necessary for Mac.)

You can copy files <1GB and directories <10GB using the command scp. This command must be run from a login node (which is the node you'll be on when you first login) and cannot be run from your local machine.

scp localuser@localIP:~/codeForBlueWaters.py ~/scratch/.
scp -r localuser@localIP:~/directoryForBlueWaters ~/scratch/

The first command is for copying a single file from your local machine to the scratch directory. The second command is for copying an entire directory (the -r option is

to do so recursively). Type ifconfig at the command line on your local machine to get your IP address. Another way is if you type exit at your BlueWaters login (user@h2ologin3) and then log back in, you will be prompted with a message such as the following:

Last login: Thu Sep 15 13:41:06 2016 from YOURIPADDRESS

For files >1GB and/or directories >10GB (which should only really be datasets for this class), let the TAs or the professor know what it is you need to transfer and I believe we will have to do this ourselves since the student accounts don't allow it. We can put the datasets in the shared projects directory for the class so everyone can have access to them. The shared projects directory for the class is /projects/eot/balq.

ls /projects/training/bauh/
OR
cd /projects/training/bauh/

This will list files we provide for the class. You can copy these into your own local ~/scratch directory to run them yourself or make changes.

cp /projects/training/bauh/*CIFAR* ~/scratch/.

This command will copy all of the files with 'CIFAR' somewhere in the name to your local scratch directory. **Note: /scratch is different than ~/scratch. The ~ is shorthand for your own local home directory where /scratch is on the root directory. For example, my ~/scratch is actually located at /u/training/instr002/scratch. Don't try to do anything with the root /scratch.**

The scratch directory is supposed to be wiped occasionally (I believe every 30 days although I don't think I've ever actually seen it wiped). It's probably not a good idea to keep important files in here. You can scp files back from BlueWaters to your local computer for storage if they are small enough.

scp ~/scratch/file_to_transfer_back localuser@localIP:~/some_directory/.

Warnings from their website:

- Interactive limits will apply on login nodes; long running transfers (hours) will require special arrangements

- Use no more than 2 login nodes for moving files. If load or interactive response becomes an issue, the admins will terminate data transfers without warning.
- Interactive response for compiling and building is the top priority for login nodes

Interactive Jobs

https://bluewaters.ncsa.illinois.edu/interactive-jobs

https://bluewaters.ncsa.illinois.edu/cluster-compatibility-mode

Interactive jobs allow you to work directly on a compute node for running code. This is very useful when writing new code and/or debugging existing code. At the login node (user@h2ologin3), type the following command:

qsub -I -l gres=ccm -l nodes=1:ppn=16:xk -l walltime=01:00:00 -X

-I (capital i) is for interactive mode. gres=ccm is for "cluster compatibility mode" (see link above). nodes=# specifies how many compute nodes you want to open up (most of the time just request 1 node). ppn=16:xk is for how many of the cpu cores you want to use on the compute nodes (you might as well use all of them) and the xk portion is for requesting a node with a GPU (always request a node with a gpu for this class). walltime=##:##:## specifies the max time you want this compute node to run for before it shuts down automatically (in case you lose internet connection or your computer freezes or any other issue like this). You can always end the interactive session earlier than the specified walltime and it won't waste any of the hours we have allocated for BlueWaters.

You will be prompted with a message about waiting for a job to start. This can take a variable amount of time depending on how many compute nodes are available at the time of requesting and how many compute nodes requested (a few minutes to a few hours).

Once the request goes through, if you only requested one node, you will immediately be at a terminal for a PBS MOM node (user@nid#####). **Just like the login node, you cannot perform any calculations here as it's against BlueWaters usage policy.** If you choose to do an interactive session for more than one node, follow the BlueWaters tutorial link above for cluster compatibility

mode. This will step you through the process of how to switch between the different compute nodes requested. I will assume here in this write up that only one node was requested.

Type the following commands to login to the compute node from the PBS MOM node:

module add ccm

ccmlogin

You will once again be at a terminal indicated by user@nid#####. However, this number should be different than the previous number indicating the PBS MOM node. You are now at an actual compute node where computations can be performed.

In order to be able to run python, type "**module load bwpy**" which loads the BlueWaters packages for python. You can also type "**module load bwpy-mpi**" which allows python to use MPI for parallel computations but this causes an error when trying to import chainer.serializers so don't load it you plan on using this (*This was for last year when we used chainer instead of pytorch. I haven't seen any issues this year with loading bwpy-mpi but I also haven't actually used any of these modules so I wouldn't worry about loading it). You will also need to type "**module load cudatoolkit"** to utilize the GPU with pytorch.

Although you are on a compute node, you are still in the same home directory of the login node where your scratch and bin directories are and any code/datasets you have. You can now safely run any code (python code_to_run.py) or start an "ipython" session.

Type "exit" if you ever want to end the interactive session before the specified walltime. You may need to exit multiple times depending on how many nodes requested and to quit the PBS MOM node and the login node as well.

Submitting Jobs

https://bluewaters.ncsa.illinois.edu/running-your-jobs

Under the /projects/eot/balq/, there are three example files: CIFAR10_TF.py, Train_CIFAR_TF.pbs, Train_CIFAR_TF.sh. These can be copied to your own local directory along with CIFAR10.hdf5 (the dataset).

Now whichever directory these are located, if you were using an interactive compute node in that location, executing "python CIFAR10_TF.py" would work. However, this particular code takes a long time (~10 hours maybe). An easier approach is to submit a job which can execute and finish without needing to interfere by using these .pbs and .sh files.

At the top of the .pbs file, there are similar arguments as those used when requesting an interactive node. Set the number of nodes and type of gpu (nodes=1:ppn=16:xk), the walltime, the name, email (-M command), and whether you want to receive emails (-m) command. The rest of the file is essentially a list of commands that would be run at the command line. There is a 'cd' command changing to the location of the files you want to run (do not use the location /projects/training/bauh that is currently there, change this to wherever you copy your code to). There are a few commands that should always be needed:

. /opt/modules/default/init/bash

module load bwpy

module load cudatoolkit

And lastly there is 'aprun -n 1 -N1 Train_CIFAR_TF.sh'. The script it calls here simply calls 'python CIFAR10_TF.py'.

At a login node (not a compute node), you can simply type 'qsub Train_CIFAR_TF.pbs' and this will submit a job to be run on BlueWaters. Make sure your code is saving the output or any models you want to keep as you won't be able to interact with the job at all.

The 'qstat' command will list all of the submitted jobs currently on BlueWaters with their status (Q for waiting in the queue and R for running). If you specified an email address above, you will receive an email when it begins running and after it is complete. You can type 'qstat | grep user' where user is your trainee account to see only the jobs you've submitted.

If you submit a job with a mistake, simply typing ‘qdel job_id.bw’ will get rid of it as to not waste any compute hours.

This is a very rough overview. It is heavily recommended to read the BlueWaters links above to get a better understanding of submitting jobs on BlueWaters.