## ROAR (incomplete) User Guide

We will assume the student already has a Roar account with ICDS. This can be done with https://accounts.aci.ics.psu.edu/users/

In this short guide we will go over submitting a job on Roar's OpenOnDemand. This will include enough information to get you started. For a complete user guide, please visit Roar's website https://www.icds.psu.edu/computing-services/roar-user-guide/

Note that you will need to connect to Penn State's VPN GlobalProtect https://ithelp.ssri.psu.edu/guides/use-penn-state-vpn-windows when connecting to Roar and you will need to use whatever 2FA method it is that you use. In my case it's DuoMobile.

## Data Transfer

When working on your project, you may want to first upload your data to the Roar server. Here we will achieve this from the command line, but you may also use GUI applications such as FileZilla. When transferring data to Roar, we will connect to *datamgr.aci.psu.edu.*

### SCP

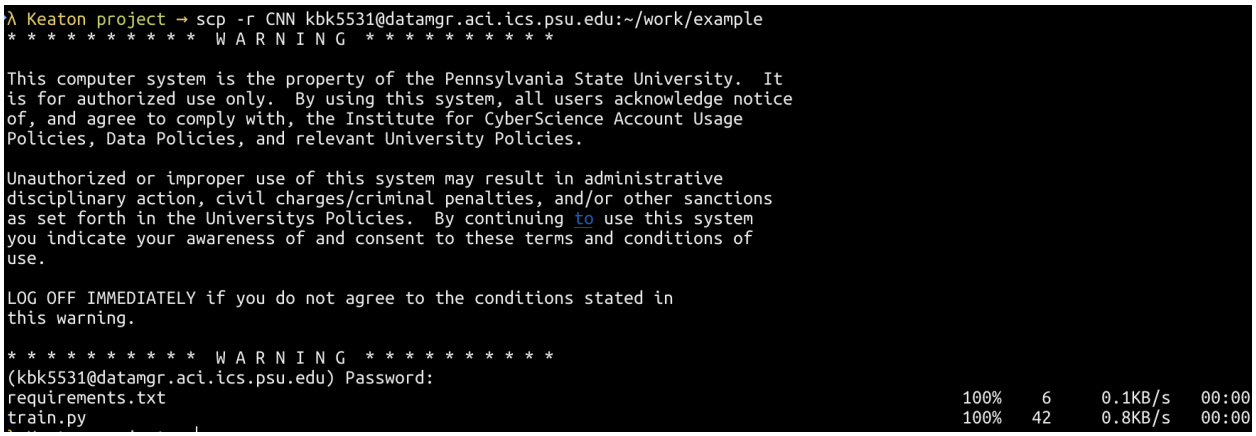
Lets copy a file to the Roar server from our local machine using secure copy with
> scp <file_to_upload> username@datamgr.aci.ics.psu.edu:<target_directory>.
For example

```
λ Keaton project → scp data.txt kbk5531@datamgr.aci.ics.psu.edu:~/work/example
* * * * * * * * * * W A R N I N G * * * * * * * * * *

This computer system is the property of the Pennsylvania State University.  It
is for authorized use only.  By using this system, all users acknowledge notice
of, and agree to comply with, the Institute for CyberScience Account Usage
Policies, Data Policies, and relevant University Policies.

Unauthorized or improper use of this system may result in administrative
disciplinary action, civil charges/criminal penalties, and/or other sanctions
as set forth in the Universitys Policies.  By continuing to use this system
you indicate your awareness of and consent to these terms and conditions of
use.

LOG OFF IMMEDIATELY if you do not agree to the conditions stated in
this warning.

* * * * * * * * * * W A R N I N G * * * * * * * * * *
(kbk5531@datamgr.aci.ics.psu.edu) Password:
data.txt                                                    100%   13     0.3KB/s   00:00
```

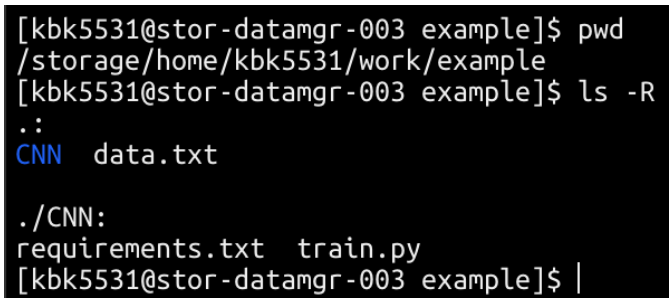
In the case of transferring a directory, we will include the -r flag in our scp command.

Note that before initiating the transfer, I was connected to Penn State's VPN and after initiating the transfer and entering my password, I had to authenticate it via my DuoMobile account.

Let's also upload the code we want to run in the directory *CNN*.




We can now check all our files were properly copied.




You can read more on data transfering here:
https://www.icds.psu.edu/computing-services/roar-user-guide/how-to-transfer-data-at-the-command-line/

**Connecting to ACI**

For transferring data to Roar, we were using datamgr. For running jobs and making submissions to Roar's queue, we will connect to submit.aci.ics.psu.edu using ssh. As noted in the Roar documentation "users connecting with ssh are encouraged to use the secure x-window forwarding flag (-Y) if x-windows will be used during the session". Lets connect to Roar, as an example

> ssh kbk5531@submit.aci.ics.psu.edu

In many cases, you may want to create a virtual environment. Here we will be doing this with Anaconda, a popular package manager for Python that comes installed on the Roar Server.

**Requesting a GPU via the Open queue**

If you requested to use Roar for this class, we are assuming you will need a GPU. Here we will go through an example of requesting a GPU on the Open Roar queue. You can read more here: https://www.icds.psu.edu/computing-services/roar-user-guide-legacy/.

For this example, we will be doing an interactive job on the Roar server. Note you may also submit jobs in a non-interactive way. First we will request access to GPU resources. We request this with

> qsub -I -A open -l nodes=1:ppn=1:gpus=1 -l pmem=1gb -l walltime=0:05:00

```
[kbk5531@submit-002 work]$ qsub -I -A open -l nodes=1:ppn=1:gpus=1 -l pmem=1gb -l walltime=0:05:00
Job will run under the 'open' account, as requested.
qsub: waiting for job 35003987.torque01.util.production.int.aci.ics.psu.edu to start
```

We recommend you read more about the arguments I've included here:
https://www.icds.psu.edu/computing-services/roar-user-guide-legacy/
In your case, you will likely need a greater walltime (maximum run time), pmem (maximum physical memory used by any processor in the job), and mem (maximum amount of physical memory used in total) which I did not include here. It will be left up to the student to determine these values.

**Running your program**

Note that in the following section I created a new job that does not request a GPU to not take up resources I don't actually need.

If you choose to run your program in an interactive session and your job has started, we will likely first want to create our conda environment.

```
[kbk5531@comp-bc-0383 example]$ pwd
/storage/home/kbk5531/work/example
[kbk5531@comp-bc-0383 example]$ module load anaconda3
[kbk5531@comp-bc-0383 example]$ conda create -n example python=3.8 --file CNN/requirements.tx
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Here we first *load* Anaconda, create a virtual environment running Python 3.8, and install any requirements in our *CNN/reuirements.txt*. Now that we have our environment created, we want to activate it so we can use any necessary packages. You can do this with *conda activate env_name*. Note if this gives an error, you may first need to run *source activate env_name*. Once in our env, we can run our code.

```
[kbk5531@comp-bc-0383 example]$ conda activate example
(example) [kbk5531@comp-bc-0383 example]$ python CNN/
requirements.txt  train.py
(example) [kbk5531@comp-bc-0383 example]$ python CNN/train.py
Hello world!
(example) [kbk5531@comp-bc-0383 example]$
```

**Resources + Notes**

- We highly recommend you look through the Roar user guide here: https://www.icds.psu.edu/computing-services/roar-user-guide-legacy/ for more examples and information on running jobs and the Roar server in general.
- For those who have not worked with Unix/Linux before, ICDS has a useful cheat sheet of commands you can find here: https://www.icds.psu.edu/computing-services/roar-user-guide/useful-unix-linux-commands/
- In this document, we went through submitting an interactive job. Often when you want to run code, you may want to simply submit a job (non-interactive). This can be faster and you can automate more parts of running your code. We HIGHLY recommend that you first try a "test run" in interactive mode first, to ensure you can properly create an environment, install packages, etc before creating a submission script.