HPC Jupyter notebook/server

Leveraging a lot of the Pedro's work on the HPC, this bash script would streamline what you would need to do to have an 8 hour* Jupyter notebook session that uses the HPC resources. Since I am currently using a loaner laptop with limited resources, I needed to use the HPC for any coding and debugging for TPOT.

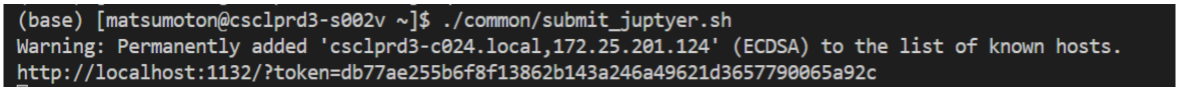
1. SSH into one of the submit nodes:
    a. Csclprd3-s001v.csmc.edu
    b. csclprd3-s002v.csmc.edu
    c. csclprd3-s003v.csmc.edu
2. Install Miniconda in your respective common folder

```
cd /common/<username>
wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
chmod +x Miniconda-latest-Linux-x86_64.sh
./Miniconda-latest-Linux-x86_64.sh
```

3. Create a conda environment. Keep note of the name (I used tpot_env)
    a. Sample:

```
conda create --name tpot_env tpot matplotlib xgboost jupyter
```

4. Put the 2 bash scripts into your respective common directory (submit_jupyter.sh and _submit_jupyter.sh)
    a. You will need to edit the _submit_jupyter.sh file to make sure your activate the right coda environment. tpot_env is the default in the file.
5. Then return back to your home directory (cd ~)
6. Run submit_jupyter.sh bash script. After about 30 seconds, you should see the output look like:



```
(base) [matsumoton@csclprd3-s002v ~]$ ./common/submit_juptyer.sh
Warning: Permanently added 'csclprd3-c024.local,172.25.201.124' (ECDSA) to the list of known hosts.
http://localhost:1132/?token=db77ae255b6f8f13862b143a246a49621d3657790065a92c
```

7. You can then open that URL in your browser for a jupyter notebook or use the link for your VSCode Jupyter plugin.

You can edit the bash scripts if you like to have a Jupyter session longer than 8 hours by editing the 28800 number on each script.

The opened port number is actually the first 4 digits of your checksum hash of your username so we likely won't have to worry about collision of ports without a hash collision.

Happy programming folks.