

Jacob Barger

07/07/2020

Creating a TensorFlow / Cuda Virtual Environment in Anaconda for Lewis Cluster

1. First you need to create a conda configuration file and insert a conda storage location, so you can tell conda where to store its files. This is to prevent the environment from filling up your home directory. Run the following command in your home directory:

```
vim ~/.condarc
```

After creating and opening the file, insert/paste the following four lines:

```
envs_dirs:  
- /storage/hpc/data/${USER}/miniconda/envs  
pkgs_dirs:  
- /storage/hpc/data/${USER}/miniconda/pkgs
```

Save & exit the file by pressing escape, and typing :wq into the vim command line, then press enter.

Afterwards you should be able to view the contents of the file as such:

```
[jsbp67@lewis4-r630-login-node675 ~]$ cat ~/.condarc  
envs_dirs:  
- /storage/hpc/data/${USER}/miniconda/envs  
pkgs_dirs:  
- /storage/hpc/data/${USER}/miniconda/pkgs  
[jsbp67@lewis4-r630-login-node675 ~]$
```

2. Next we need to grab a compute node, as we begin in a login node, run the following command:

```
srun --partition Interactive --pty /bin/bash
```

You should see your node change:

```
[jsbp67@lewis4-r630-login-node675 ~]$ srun --partition Interactive --pty /bin/bash  
[jsbp67@lewis4-lenovo-hpc2-node282 ~]$
```

3. Load the miniconda module into the command line via the following command:

```
module load miniconda3
```

4. Now we need to create an environment for which to install our packages. This is so later we can load this environment onto the GPU node for our jobs.

Create the node with the following command, and substitute the environment name, *my_environment*, with something meaningful to you and your project, and the python version with what is required for your packages:

```
conda create --name my_environment python=3.7
```

Select yes on any prompts given

5. Next we need to activate/enable our environment with the following command, substitute *my_environment* with your environment name.

```
source activate my_environment
```

Note: You can view your environments at any time with the command: `conda env list`

And switch between them with using activate or deactivate in the syntax shown above.

Verify your environment is selected as it will be in parenthesis before your shell information:

```
[jsbp67@lewis4-lenovo-hpc2-node282 ~]$ source activate pdnet  
(pdnet) [jsbp67@lewis4-lenovo-hpc2-node282 ~]$
```

6. Now we need to install whatever packages are necessary for your project to run on the GPU.

For example, to run tensorflow on GPU we need to install the tensorflow-gpu package to our environment with the following command:

```
conda install tensorflow-gpu
```

Press yes to any prompt given for the dependencies, this may take quite a while depending on the packages.

7. Afterwards you may run GPU jobs in the terminal by requesting a GPU node, activating your conda environment, and then invoking your scripts, which can be done with the following command (note this is the two hour access GPU):

```
srun --mem 20G -p Gpu --gres gpu:1 --pty /bin/bash -login
```

```
source activate my_environment python3
```

```
./my_script.py -p parameters_here
```