Assignment 05 - Containers

DSE 512

Spring 2021

NOTE: Please use singularity on ISAAC for this assignment. Singularity 3.5.2 is available by default and does not require loading modules or any setup at all.

Please refer to the singularity documentation here: https://sylabs.io/guides/3.5/user-guide

and to the ISAAC-specific singularity documentation here: https://oit.utk.edu/hpsc/isaac-open/singularity-user-guide/

In this assignment, you will build a singularity container, update the environment inside to include code your program requires, run that code inside the image, and extract the results.

You may experiment with these steps interactively. For the assignment, you do not need to launch these in a qsub job or used distributed computing. Instead, put all of your commands into a shell script called `assignment5.sh`, launch it with `sh assignment5.sh | tee assignment5.log` and submit both the shell script and the log file that is generated.

0. (0 points) **Setup** We will be downloading containers, which will exhaust your home directory disk quota. So first, create a new cache directory for singularity, and export the following environment variable in your shell session.

```
export SINGULARITY_CACHEDIR=/lustre/haven/proj/UTK0150/$USER/singularity_cache
mkdir $SINGULARITY_CACHEDIR
```

Now, within this terminal any singularity commands will use the given directory to store the large image files we will download, and you should not run out of disk space.

1. (25 points) **Ephemeral containers**. We will launch a shell in an "ephemeral container" that is downloaded on demand and deleted when we are done. This is useful when trying out an image when you are unsure whether you will use it long-term or not. First, from an ISAAC login node, run the following commands and note the output:

```
grep PRETTY_NAME /etc/os-release
ls /
```

Next, Run the following command do drop into an ephemeral singularity container. This may take a few minutes.

```
singularity shell docker://dceoy/pydata:dnn-cpu
```

Re-run the first to commands from inside this shell and note the differences. You are inside a container which has applied an overlay to the true filesystem. Press Ctrl-d to exit.

2. (25 points) **Pulling containers** Containers can also be persisted to disk by "pulling". Run the following command to pull the docker image into a singularity image file, and verify that you can shell into it.

```
singularity build pydatacpu.sif docker://dceoy/pydata:dnn-cpu
```

3. (25 points) **Bind mounting directories**. By default, singularity containers mount `/home/$USER`, `/tmp`, and `$PWD`, meaning you can see them inside your container. Let's also mount your user home

directory and print its contents from within the container. *If you named your user directory something other than your username, you will need to make that edit to this command.*

```
singularity exec --bind /lustre/haven/proj/UTK0150/$USER:/myproj  pydatacpu.sif ls -l /myproj
```

4. (25 points) **Run kmeans_vectorized.py in the container**. Now, use the skills you developed in the first few problems to run your kmeans_vectorized.py script from previous assignments. Verify that the output matches what we saw in Assignment 01.

More exercises (0 points) There you have it. A singularity container that has pandas, numpy, and other machine learning packages already installed, without having to manage a conda environment. Note that mpi4py is missing, and if you'd like to install it, you would need to extend this container. However, it is difficult to do so, since you must build new containers on another computer that you have root access on, like an Ubuntu virtual machine. That is not part of this assignment, but is the next step in understanding and using Singularity, so if you are looking for more to learn, try modifying this container on a local machine (using the `singularity bootstrap` command and editing a Singularity file), then copying it back onto ISAAC to verify that it ran.