

Introduction to Containers

Advanced Content

featuring **Singularity** on the **Terra** cluster

by Richard Lawrence

Date: 04/30/2021

Spring 2021

Outline

- More Container Examples
- Build your own Containers

Learning Resources

- HPRC Wiki <https://hprc.tamu.edu/wiki/SW:Singularity>
- HPRC on Youtube <https://www.youtube.com/c/TexasAMHPRC>
(video of this course will be posted)
- Singularity Manual <https://sylabs.io/guides/3.7/user-guide/>
- Docker Manual <https://docs.docker.com/>
- Other container courses:
 - NBIS <https://nbis-reproducible-research.readthedocs.io/en/latest/singularity/>
 - Arizona <https://learning.cyverse.org/projects/Container-camp-2020/>
 - TACC <https://learn.tacc.utexas.edu/mod/page/view.php?id=95>

More Container Examples

With exercises

Singularity with GPU

- Containers should be built with CUDA version compatible with local GPUs (CUDA \geq 9)
- Just add the `--nv` flag to your singularity command

Many repositories on Docker Hub have GPU-ready images. Search for images with “gpu” in tags

The nvidia cloud also provides GPU-ready images. See: https://hprc.tamu.edu/wiki/SW:Singularity:Examples#NVIDIA_GPU_Cloud

TensorFlow GPU Exercise

Image file: `tensorflow_2.4.1-gpu.sif` from
`docker://tensorflow/tensorflow:2.4.1-gpu`

Located at `/scratch/data/Singularity/images/`

Also: `tf-gpu-test.py` in the same location.

```
$ srun --mem=512m --time=01:00:00 \
--gres=gpu:1 --partition=gpu --pty bash -i
$ singularity exec --nv tensorflow_2.4.1-gpu.sif \
python3 tf-gpu-test.py
Num GPUs Available:  1
```

Containers in HPRC Portal

Container support for **Interactive Apps** on the HPRC Portal

Currently:

- Jupyter Notebook (a popular Python IDE)

Will extend container support to other Apps. If you would like something, please ask help@hprc.tamu.edu

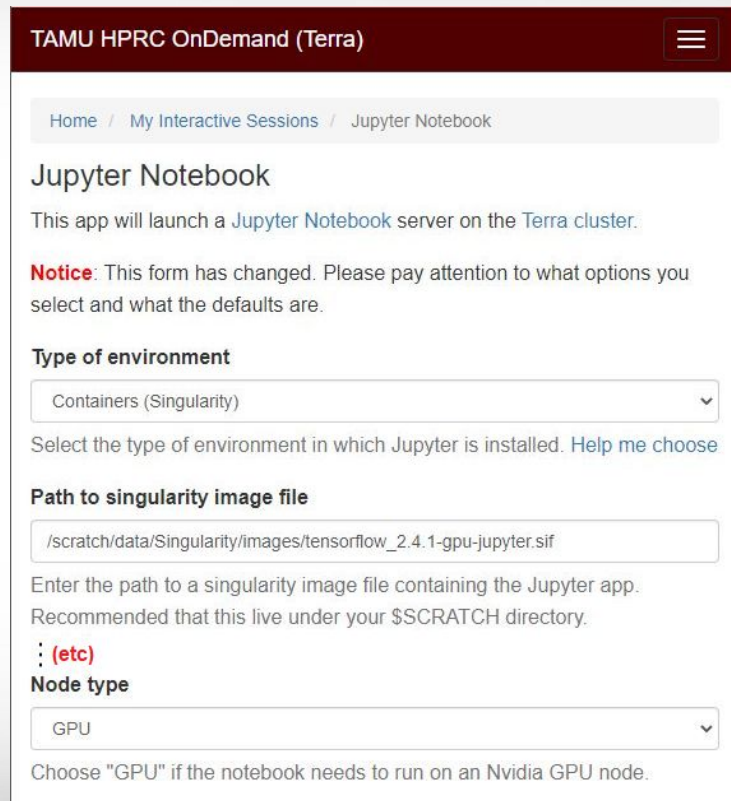
Jupyter Exercise

- Choose an image file on disk that contains Jupyter
- Image file: `tensorflow_2.4.1-gpu-jupyter.sif` from `docker://tensorflow/tensorflow:2.4.1-gpu-jupyter`

Located at `/scratch/data/Singularity/images/`

Jupyter Exercise

- Navigate to HPRC Portal → Terra OnDemand → Interactive Apps → Jupyter Notebook
- Select “Containers (Singularity)” from “Type of environment” drop-down menu
- Enter the path to the image file.



The screenshot shows the TAMU HPRC OnDemand (Terra) interface for configuring a Jupyter Notebook. The page title is "Jupyter Notebook" and the breadcrumb trail is "Home / My Interactive Sessions / Jupyter Notebook". The main heading is "Jupyter Notebook" with a sub-heading "This app will launch a Jupyter Notebook server on the Terra cluster." A red "Notice" states: "This form has changed. Please pay attention to what options you select and what the defaults are." The "Type of environment" dropdown menu is set to "Containers (Singularity)". Below this, a text input field contains the path "/scratch/data/Singularity/images/tensorflow_2.4.1-gpu-jupyter.sif". The "Node type" dropdown menu is set to "GPU".

TAMU HPRC OnDemand (Terra)

Home / My Interactive Sessions / Jupyter Notebook

Jupyter Notebook

This app will launch a Jupyter Notebook server on the Terra cluster.

Notice: This form has changed. Please pay attention to what options you select and what the defaults are.

Type of environment

Containers (Singularity)

Select the type of environment in which Jupyter is installed. [Help me choose](#)

Path to singularity image file

/scratch/data/Singularity/images/tensorflow_2.4.1-gpu-jupyter.sif

Enter the path to a singularity image file containing the Jupyter app.
Recommended that this live under your \$SCRATCH directory.

⋮ (etc)

Node type

GPU

Choose "GPU" if the notebook needs to run on an Nvidia GPU node.

Singularity with MPI

There are several ways to containerize MPI.

Check with your container source to see how it is meant to be used.

- MPI installed and run inside container.

```
singularity exec <sif> mpirun <exe>
```

- MPI installed and run outside container.

```
mpirun singularity exec <sif> <exe>
```

- MPI installed outside, run inside.

```
singularity exec --bind /dir <sif> /dir/mpirun <exe>
```

See <https://sylabs.io/guides/3.7/user-guide/mpi.html>

LAMMPS MPI Exercise

Image file: `lammps-stable_29Oct2020[...].sif` from
`docker://lammps/lammps:stable_29Oct2020[...]`

Located at `/scratch/data/Singularity/images/`

Also: `lammps-test-input.txt` in the same location.

```
$ srun --ntasks=4 --mem=2560m --time=01:00:00 --pty bash -i
```

```
$ singularity run --bind "/scratch,$TMPDIR:/work" \  
lammps-stable_29Oct2020_ubuntu20.04_openmpi_py3.sif mpirun -np 4 \  
/bin/lmp_mpi -in lammps-test-input.txt
```

Biocontainers Exercise

Image file: `bcbio-nextgen-1.2.8.sif`

From docker://quay.io/biocontainers/bcbio-nextgen

<https://biocontainers.pro/tools/bcbio-nextgen>

Located at `/scratch/data/Singularity/images/`

```
$ srun --mem=512m --time=01:00:00 --pty bash -i
$ singularity exec bcbio-nextgen-1.2.8.sif bcbio_nextgen.py --help
usage: bcbio_nextgen.py [...]
$ singularity exec bcbio-nextgen-1.2.8.sif python -c \
  "import pysam; print(pysam.__version__)"
0.16.0.1
```

Build your own Containers

Overview, no exercises

Singularity Image Formats

Singularity has two image formats:

- A compressed read-only file (extension `.sif`)
- A writable directory structure called “sandbox”.

Singularity `build` command can convert from one to the other. This is mainly useful for modifying images, such as installing new software.

Installing new software usually requires **root privileges**, which must be done on your personal machine.

Modify Container Example

Starting with an existing image located at `<source>`

```
$ singularity build --sandbox <sandbox-name> <source>
```

This creates a directory `<sandbox-name>/` that works the same as an image file, except it can be used in write mode.

```
$ sudo singularity shell --writable <sandbox-name>/
```

Now you install software in the root directories, as the root user. When you're done, you exit the sandbox and:

```
$ singularity build <final-name>.sif <sandbox-name>/
```

This creates the read-only file suitable for production use.

See https://sylabs.io/guides/3.7/user-guide/build_a_container.html

Container Recipe

Singularity `build` can build containers from a recipe.

A recipe might contain these facts:

- “Bootstrap”: what your build starts from. It could be an existing recipe or image, or it could be from scratch.
- What software to install in the container and how.
- “Runscript”: what your container does by default.

Once you have the recipe file defined:

```
$ sudo singularity build <container> <recipe-file>
```

See https://sylabs.io/guides/3.7/user-guide/definition_files.html

Automatic Singularity Build

You can create an account on Singularity Hub and use it to automatically build and publish your containers.

- Start with a github repository you can push to.
- Commit a singularity recipe file named “Singularity”.
- Link that github repository in Singularity Hub.
- Define what triggers a new build. It can be whenever a new tag is committed, for example.
- Pull your freshly-built container from Singularity Hub.

See <https://singularityhub.github.io/singularityhub-docs/>

Advanced Content Complete

Conclusion

- Run Containers on clusters! It's easy.
- HPRC supports Singularity
- Convert Docker to Singularity!
- Expect Charlie Cloud support in the near future
- Ask for help!

Survey

Please fill out the survey to let us know how you feel about this short course. This will help us improve.

Questions



Learning Resources

- HPRC Wiki <https://hprc.tamu.edu/wiki/SW:Singularity>
- HPRC on Youtube <https://www.youtube.com/c/TexasAMHPRC>
(video of this course will be posted)
- Singularity Manual <https://sylabs.io/guides/3.7/user-guide/>
- Docker Manual <https://docs.docker.com/>
- Other container courses:
 - NBIS <https://nbis-reproducible-research.readthedocs.io/en/latest/singularity/>
 - Arizona <https://learning.cyverse.org/projects/Container-camp-2020/>
 - TACC <https://learn.tacc.utexas.edu/mod/page/view.php?id=95>

Thank you

Contact: help@hprc.tamu.edu