Containers

Shipping containers have frequently been used as an analogy for computing containers because the container is standard, it does not care what is put inside, and it will be carried on any ship, or in the case of computing containers can run on many systems different from the one it was created on. Hence the logo for Docker, the most common container platform.

Docker is widely used by researchers for reasons we won’t get into here. See their documentation

Docker images cannot be run in a HPC environment. This has to do with the privileges required to run a Docker container. This issue is addressed with Singularity. It is a container technology that completely contains the authority so that when the image is run all privileges stay inside the container. This makes it ideal for the shared environment of a supercomputer. And even better is that a Docker image can be encapsulated inside a Singularity image. So the documentation here instructs how to take either a Docker image and run it from Singularity or create an image using Singularity only.

Singularity Overview

Singularity containers let users run applications in a Linux environment of their choosing. This is similar to but not the same as Docker.

The most important thing to know is that you create the singularity container called an image on a workstation where you have root privileges, and then transfer the image to HPC where you can execute the image. If root authority is an issue then the answer might be a virtual environment on your laptop, like Vagrant for MacOS

For an overview and more detailed information refer to:
http://singularity.lbl.gov
Here are some of the use cases we support using Singularity:

- You already use Docker and want to run your jobs on HPC
- You want to preserve your environment so that a system change will not affect your work
- You need newer or different libraries than are offered on HPC systems
- Someone else developed the workflow using a different version of linux
- You prefer to use something other than Red Hat / CentOS, like Ubuntu

**Singularity Commands**

```
$ singularity --help
USAGE: singularity [global options...] <command> [command options...] ...
```

**GLOBAL OPTIONS:**
- `-d` `--debug` Print debugging information
- `-h` `--help` Display usage summary
- `-s` `--silent` Only print errors
- `-q` `--quiet` Suppress all normal output
- `--version` Show application version
- `-v` `--verbose` Increase verbosity +1
- `-x` `--sh-debug` Print shell wrapper debugging information

**GENERAL COMMANDS:**
- `help` Show additional help for a command or container
- `selftest` Run some self tests for singularity install

**CONTAINER USAGE COMMANDS:**
- `exec` Execute a command within container
- `run` Launch a runscript within container
- `shell` Run a Bourne shell within container
- `test` Launch a testscript within container

**CONTAINER MANAGEMENT COMMANDS:**
- `apps` List available apps within a container
- `bootstrap` *Deprecated* use build instead
- `build` Build a new Singularity container
- `check` Perform container lint checks
- `inspect` Display a container’s metadata
- `mount` Mount a Singularity container image
- `pull` Pull a Singularity/Docker container to $PWD

**COMMAND GROUPS:**
- `image` Container image command group
- `instance` Persistent instance command group

**CONTAINER USAGE OPTIONS:**
- `see singularity help <command>`

For any additional help or support visit the Singularity website: http://singularity.lbl.gov/

**Singularity Tutorials**

There are tutorials located here

**Singularity and GPU's**

One of the most significant use cases for Singularity is to support machine learning workflows. The details are in the GPU section.

**Singularity Python and Machine Learning**

More tutorials