

# Containers

- Singularity
  - Background
  - Build a Singularity Container
    - Build a container from a Singularity recipe (ubuntu1604-cuda92-mpi400.def)
    - Pull a container from Singularity Hub or Docker Hub
  - Run a Singularity Container
    - Run a container with "exec" command
    - Run a container with shell
  - Build a writable Singularity container
  - Build a writable Singularity container based on an existing read-only image
  - Frequently Asked Questions

## Singularity

### Background

Singularity is a container solution created by necessity for scientific and application-driven workloads. The Singularity containers can be used to package entire scientific workflows, software and libraries, and even data. This means that you don't have to ask your cluster admin to install anything for you - you can put it in a Singularity container and run.

(see webpage <https://www.sylabs.io> and/or <http://singularity.lbl.gov/> for details)

### Build a Singularity Container



#### Admin assistance required

Singularity now requires fakeroot privilege for non-root users to create container images. Please contact an admin to set up your account with fakeroot.

### Build a container from a Singularity recipe ([ubuntu1604-cuda92-mpi400.def](#))

```
wget https://wiki.ncsa.illinois.edu/download/attachments/82518873/ubuntu1604-cuda92-mpi400.def
singularity build ubuntu1604-cuda92-mpi400.sif ubuntu1604-cuda92-mpi400.def
```

### Pull a container from Singularity Hub or Docker Hub

```
singularity pull docker://nvidia/cuda-ppc64le:9.2-cudnn7-devel-ubuntu16.04
```

### Run a Singularity Container



#### Mounted home filesystem

Unlike docker (which provides an isolated filesystem), Singularity mounts your home directory into the container. Be aware of this so you don't accidentally delete things in your home directory.

### Run a container with "exec" command

```
mpirun -n 4 singularity exec --nv /opt/apps/samples-image/ubuntu1604-cuda92-mpi400.sif hostname
```

### Run a container with shell

```
singularity shell --nv /opt/apps/samples-image/ubuntu1604-cuda92-mpi400.sif
```

## Build a writable Singularity container

By default, the Singularity container filesystem is read-only. There are 2 options to build a writable Singularity container.

The "--sandbox" option allows users to create a container within a writable directory (called a *sandbox*). It's possible to create a sandbox without root privileges, but to ensure proper file permissions it is recommended to do so as root.

```
singularity build --sandbox cuda-ppc64le/ docker://nvidia/cuda-ppc64le:9.2-cudnn7-devel-ubuntu16.0
singularity shell --writable cuda-ppc64le/
```

The "--writable" option allows users to create a writable ext3 image. Root privileges are required to create writable containers.

```
singularity build --writable cuda-ppc64le.sif docker://nvidia/cuda-ppc64le:9.2-cudnn7-devel-ubuntu16.04
singularity shell --writable cuda-ppc64le.sif
```

## Build a writable Singularity container based on an existing read-only image

Convert it to a writable directory (a sandbox):

```
singularity build --sandbox cuda-ppc64le-mod/ cuda-ppc64le.sif
singularity shell --writable cuda-ppc64le-mod/
```

Use an existing container as a target to build a new container

```
singularity build --writable cuda-ppc64le-mod.sif cuda-ppc64le.sif
singularity shell --writable cuda-ppc64le-mod.sif
```

## Frequently Asked Questions

1. Can I run a Singularity container with x86 architecture image on an IBM ppc64le system?  
A: Singularity cannot directly run x86 binaries on an IBM Power system because the bytes of machine code are different. Users need to make sure their own images are built for "ppc64le" rather than "x86".
2. What is the option "--nv" means?  
A: The "--nv" allows the Singularity container to leverage the NVIDIA GPU on the host system. Use this option if the container asks for nvidia-docker support.