

# Singularity & Amber

August 28, 2019

## 1 Foreword

[Singularity](https://sylabs.io) (<https://sylabs.io>) is a software that manages and operates containers. Unlike [docker](#), [singularity](#) often does not need [sudo](#) to run and is therefore suitable for production in High Performance Computing centres by regular (i.e., non-admin) users.

With [singularity](#), containers are stored on SIF (Singularity Image Format) files. These containers can be built from definition files (usually without root access).

For Amber development, the main interest in using [singularity](#) is to be able to build and to use Amber using various flavors of Linux, therefore easing the debugging process for these different platforms. Another interest lies in the fact that, at runtime, [singularity](#) mount the [\\$HOME](#) and the current working directory within the container. That way, an Amber tree on the host machine can be directly accessed within the container by running it directly inside [\\$AMBERHOME](#).

## 2 Requirements

- [singularity](#)  $\geq$  3.3 ([root](#) or [sudo](#) access is not necessary for standard usage, but is required for the installation of [singularity](#) or the build of sandbox directories)
- Amber source code(!)

## 3 Container building

All container images are named as [linux/major.minor.sif](#) for Singularity Image Format files (e.g., [centos/7.6.sif](#)), and [linux/major.minor-sandbox](#) for *sandbox* directories. Container definition files are stored in the directory [\\$AMBERHOME/containers/singularity](#). They are in the form [linux/major.minor.def](#).

A script called [amberity](#) helps managing these containers:

```
export AMBERHOME=...
alias amberity=$AMBERHOME/containers/singularity/amberity
amberity -h
```

```
Usage: amberity [options] COMMAND [ARGS]
```

Command: a Amber command (e.g., sander, cpptraj, ...) with its optional arguments

Options:

```
-a | --avail      : List available Singularity containers
-b | --build <arg> : Build a specific Singularity container (e.g., centos/7.6.sif)
-d | --def <arg>  : Specify a default Singularity container (then quit)
-g | --get        : Get the default Singularity container
-h | --help       : Print command line help and exit.
-s | --sif <arg> : Specify a Singularity container to run (e.g., centos/7.6.sif)
-t | --targets    : List Singularity containers that can be built
```

To get the list of the containers that can be built:

```
amberity -t
```

The following containers can be built:

```
centos/7.6.sif
centos/6.10.sif
debian/10.sif
debian/9.sif
fedora/30.sif
opensuse/15.1.sif
opensuse/15.0.sif
ubuntu/18.04.sif
ubuntu/16.04.sif
nvidia/10.1/ubuntu/18.04.sif
nvidia/10.1/ubuntu/16.04.sif
nvidia/10.1/centos/7.sif
```

You can choose and build a container like this:

```
amberity -b centos/7.6.sif
```

The container centos/7.6.sif can now be used and any command can be executed withing the container:

```
amberity --sif centos/7.6.sif cat /etc/os-release
```

```
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
[...]
```

To avoid the specification of the same container over and over. One can define a *default* container:

```
amberity --def centos/7.6.sif
```

```
Set $AMBERHOME/containers/singularity/default to centos/7.6.sif
```

Then:

```
amberity cat /etc/os-release
```

will execute `cat /etc/os-release` within the default container.

If you don't remember which container you defined as a default, you can retrieve the information via the `--get` command:

```
amberity --get
```

```
Default container is: centos/7.6.sif
```

## 4 Amber building

The `amberity` script allows also the configuration, the building and the testing of the Amber source code using a specific command named `amber`.

**Note:** the `amber` command is only defined within the container built in `$AMBERHOME/container/singularity`. It is not a command that can be run directly on the host machine.

Calling the `amber` script is simple. Provided that you defined a default container and that your current working directory is `$AMBERHOME`, just type:

```
# your current directory must be AMBERHOME
cd $AMBERHOME
amberity amber
```

```
Usage: amber <command> [options]
```

Available commands:

<code>config</code>	same as <code>config.serial</code>
<code>config.serial</code>	configure serial (e.g., default) version of Amber
<code>config.openmp</code>	configure OpenMP version of Amber
<code>config.parallel</code>	configure parallel MPI version of Amber
<code>config.mpi</code>	same as <code>config.parallel</code>
<code>build</code>	build amber (e.g., 'make install' after configuration)
<code>build.serial</code>	<code>config.serial</code> + <code>build</code>
<code>build.openmp</code>	<code>config.openmp</code> + <code>build</code>
<code>build.parallel</code>	<code>config.parallel</code> + <code>build</code>
<code>build.mpi</code>	same as <code>build.parallel</code>
<code>build.full</code>	<code>build.serial</code> + <code>build.openmp</code> + <code>build.parallel</code>
<code>test</code>	test current installed version
<code>test.serial</code>	test serial version
<code>test.openmp</code>	test openmp version
<code>test.parallel</code>	test parallel version (with 2 cores)
<code>test.mpi</code>	same as <code>test.parallel</code>
<code>full</code>	<code>full.serial</code> + <code>full.openmp</code> + <code>full.parallel</code>

```
full.serial      build.serial + test.serial
full.openmp     build.openmp + test.openmp
full.parallel   build.parallel + test.parallel
```

Warning: only 'gnu' installation is currently supported...

Build options:

```
-np X           requests building Amber with X cores using parallel make
```

To configure Amber using your container, just type:

```
amberity amber config
```

After configuration you can build Amber using 4 cores (parallel make):

```
amberity amber build -np 4
```

## 5 Amber program usage

After successfully building (and testing) your Amber source code using your favorite container, you can call any program built within the Amber code:

```
amberity cpptraj
```

```
CPPTRAJ: Trajectory Analysis. V4.15.0 (AmberTools V20.00)
```

```
  ---  ---  ---  ---
  | \ / | \ / | \ / |
  _|_/\_|_/\_|_/\_|_
```

```
| Date/time: 08/28/19 00:03:40
```

```
| Available memory: 7.306 GB
```