Hello all, I’ll be putting up a step-by-step guide on how to install and use Singularity.

In simple terms, Singularity (or Apptainer- it’s the same thing essentially) can be used to create an “image” of an environment- including the folders containing your code, the python packages already installed, already compiled ready-to-run Fortran executables etc. The advantage of using a Singularity image is that you can simply copy it from your local system, and run it anywhere without having to compile it in your new system- whether it be HPC, or another computer.   
  
For example: Suppose you have to run a code in a new system or HPC which does not give you root privileges. And the code that you need to install requires libraries that need root access to compile the code. In that case you can simply make a singularity image, move it to the HPC, extract it and simply run it without needing to compile!  
  
The ‘Singularity image’ of the environment is in ‘.sif’ format (Singularity Image Format) and it is built based on a user-defined ‘.def’ file. You can make ‘.sif’ images of both local directories as well as GitHub repositories. Here is an example of the singularity ‘.def’ file that Abel wrote <https://github.com/Ai33L/climt-heatwaves/blob/main/climt.def> (Hope you have access to his private repo). I will share my ‘.def’ file once I update my git repository.

**Step-by-step on installing and using Singularity:**As mentioned earlier, singularity and Apptainer are the same thing. I used Apptainer because I had some minor issues (probably skill issues) with Singularity. But it works the same. You can use Singularity to unpack/build a ‘.sif’ image made by Apptainer in your local system.

* Install apptainer on your local PC:sudo apt install apptainer
* Check if your apptainer is installed and your system can find it:  
  apptainer --version
* Write your ‘.def’ file and place it right outside your *climt* directory. (Check Abel’s def file or simply find guides online, I suggest the former)
* Navigate to right outside your *climt* directory and run the code to make the ‘.sif’ file  
  sudo apptainer build <image name>.sif <def file name>.def

If you don’t have sudo access then run the code below. Really helpful if you want to create a sandbox directly inside an HPC.   
apptainer build --fakeroot <image name>.sif <def file name>.def

* Voila!, you should have your successful ‘.sif’ file build if the Linux Gods permit. If things go wrong (which they probably will, let me know).
* You might encounter an error while installing Apptainer, as Ubuntu doesn’t ship with it by default. Therefore, you’ll need to add it manually.  
  sudo add-apt-repository -y ppa:apptainer/ppa

sudo apt update

**Alternate workflow if you are not comfortable with creating a .def file**

* sudo apptainer build --sandbox <sandbox\_name>/ library://ubuntu

This pulls Ubuntu from the Singularity library and creates an image. Switch ‘library’ with  
 ‘docker’ if you run into an error. You can use ‘--fakeroot’ instead of sudo if you don’t have access

* apptainer shell --containall --writable <sandbox\_name>

This should open up a shell inside the container. You can install all the necessary packages within this container. Try to run a sample code/application to see if everything works fine. After this, you can exit the shell.

* sudo apptainer build <image\_name>.sif <sandbox\_name>

### **Non-sudo local install of apptainer:**

You need to have Go compiler to install Apptainer. If your system already has one, then skip the installation.

wget https://go.dev/dl/go1.22.6.linux-amd64.tar.gz

tar -C $HOME -xzf go1.22.6.linux-amd64.tar.gz

Add to PATH temporarily for this shell

export PATH=$HOME/go/bin:$PATH

Download apptainer source:

mkdir -p ~/apptainer-build

cd ~/apptainer-build

wget https://github.com/apptainer/apptainer/releases/download/v1.3.2/apptainer-1.3.2.tar.gz

tar -xzf apptainer-1.3.2.tar.gz

cd apptainer-1.3.2

Install and compile:

./mconfig --prefix=$HOME/apptainer-local

make -C builddir

make -C builddir install

export PATH=$HOME/apptainer-local/bin:$PATH

echo 'export PATH=$HOME/apptainer-local/bin:$PATH' >> ~/.bashrc add this line to bash such that this is added to PATH permanently.

apptainer --version to check for if the installation works

### **Running apptainer/singularity:**

Either you open a shell inside the image and run it as usual, or you can directly execute it.

If your codes/data are in some other folder you use --bind argument to specify where to look for the required files. The bound files will be available in the folder you specify, /home in this case.

Eg. apptainer shell --bind climt\_singularity/model\_runs/base\_config:/home climt\_singularity/climt.sif

This code binds the base\_config directory to /opt/climt of the image. This can be accessed through the image/container shell.

To use image shell:

Go to the folder with the image (.sif) file.  
apptainer shell <file\_name>.sif

Once you are in the container shell, use python -s python\_file.py to run the script. This forces it to use the container packages and not the local ones. This might not be required in your case, but I seem to have had trouble with package mismatch since the script accidentally used the local packages.

To run it without accessing image shell:

You have to add the command after the code. For example python3 scartch1.py

apptainer exec --bind climt\_singularity/model\_runs/base\_config:/opt/climt climt\_singularity/climt.sif <command>

To create and use sandbox

Apptainer build --sandbox <sandbox\_name> <file\_name>.sif

apptainer shell --writable --containall --bind model\_runs/:/home climt\_sandbox

**Interactive job on HPC**

salloc --nodes=1 --ntasks=1 --cpus-per-task=4 --mem=8G --time=02:00:00

srun --pty bash

Or

srun --nodes=1 --ntasks=1 --cpus-per-task=4 --mem=8G --time=02:00:00 --pty bash