Singularity Containers on Minerva HPC

Patricia Kovatch
Lili Gai, PhD

Eugene Fluder, PhD
Hyung Min Cho, PhD
Jielin Yu, PhD
Wei Guo, PhD
Kali Mclennan

October 07 2022

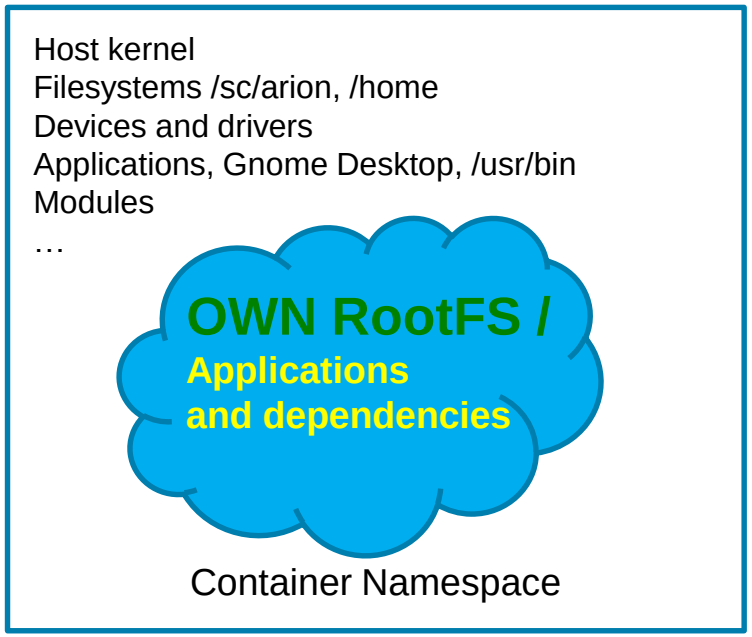
Mount Sinai

Outline

- What is a container?
- Why Singularity, and docker?
- Basic Usage of Singularity as a Minerva User
- Example use cases: RStudio Web and Jupyter Notebook on-the-fly

Containers

- A standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another.



By-function containers provide:

- Software bundles for applications
- Self contained environment, BYOE
- Platform/Host agnostic

https://www.docker.com/resources/what-container

https://portal.biohpc.swmed.edu/content/guides/singularity-containers-biohpc/

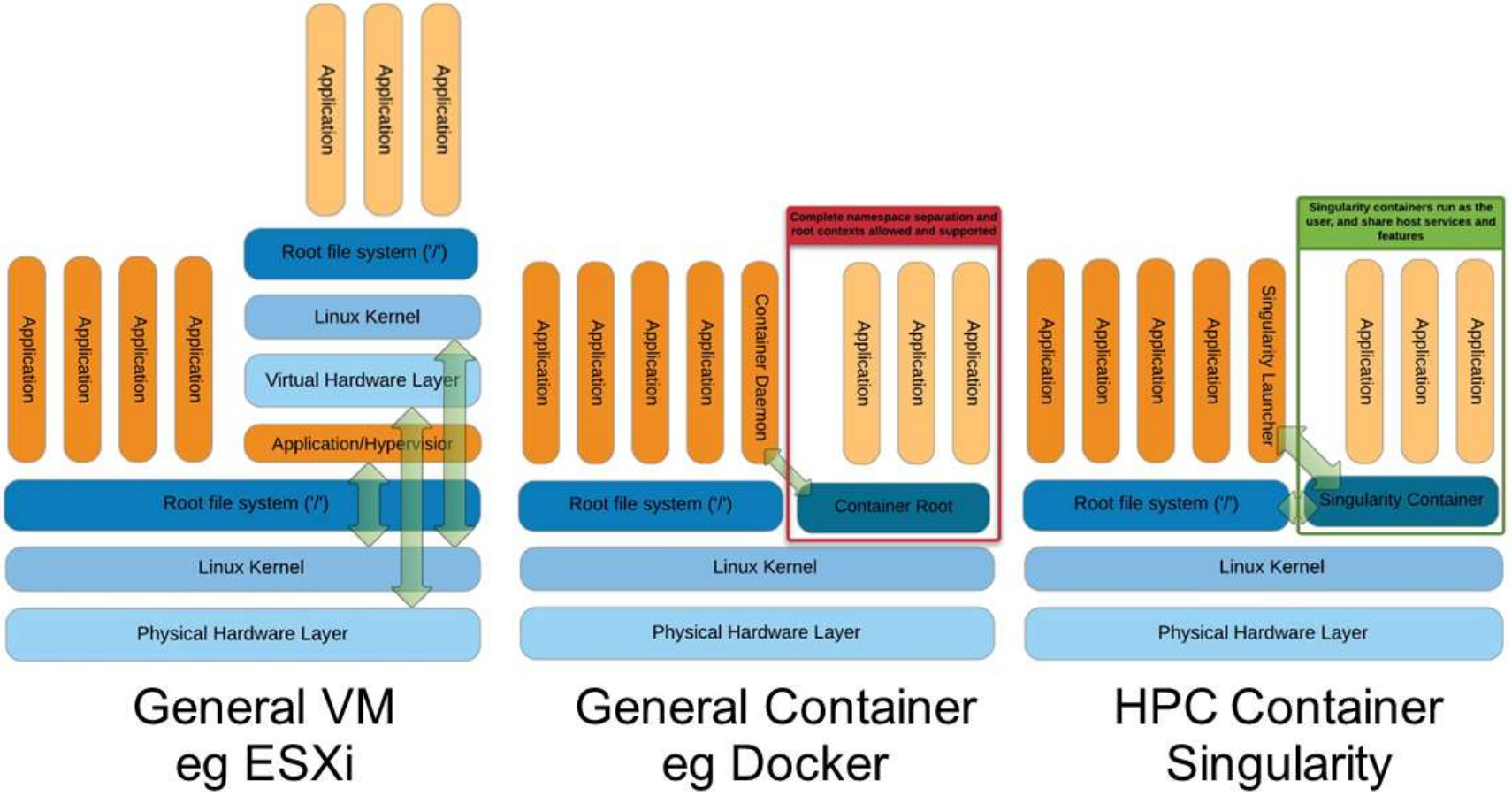
Why Containers?

- The software I want to use is too complicated that I can't get it work on my computer anyhow.

- The software can't be installed on the cluster because of new kernel or system level library requirements

- I want to rerun my analysis sometime ago; I want to reproduce my collaborator's pipelines or results

VM vs Container, Singularity vs Docker



Only Singularity is supported on Minerva HPC

Reference: https://tin6150.github.io/psg/blogger_container_hpc.html

Singularity on Minerva HPC - 1

Use singularity module on Minerva nodes

```
$ bsub -q gpu -XF -P acc_hpcstaff -n 4 -W 3:00 -R v100 -R "rusage[mem=3000],
rusage[ngpus_excl_p=1]" -Is /bin/bash

$ module load singularity/3.6.4

# If on non-login non-interactive compute node, set proxy first

$ module load proxies

# or

$ export http_proxy=http://172.28.7.1:3128
$ export https_proxy=http://172.28.7.1:3128
$ export all_proxy=http://172.28.7.1:3128
$ export no_proxy=localhost,*.chimera.hpc.mssm.edu,172.28.0.0/16
```

Pull image from Docker Hub docker://, and Sylabs Cloud library://

```
$ singularity pull docker://gcc:10

$ ls -l
-rwxr-xr-x 1 yuj25 hpcstaff 396431360 Oct  4 17:04 gcc_10.sif

$ singularity pull library://vigo332/default/singularity-rstudio-r4
```

Singularity on Minerva HPC - 2

Images layers are cached in $HOME/.singularity/cache/, may blow up your $HOME quota

```
$ singularity cache list -v
NAME                     DATE CREATED           SIZE           TYPE
0e3f4c426c9e5994ac625c   2021-04-23 16:46:57    440.43 MB      blob
0f46f97746e4df5959e8c8   2021-04-26 12:57:43    213.09 MB      blob

$ singularity cache clean
```

You can change the cache directory by specifying the SINGULARITY_CACHEDIR environment parameter:

```
$ SINGULARITY_CACHEDIR=/sc/arion/scratch/yuj25/containers/cache \
   singularity pull docker://gcc:10

INFO:    Converting OCI blobs to SIF format
INFO:    Starting build...
Getting image source signatures
Copying blob 723254a2c089 done
Copying blob abe15a44e12f done
Copying blob 409a28e3cc3d done
Copying blob 503166935590 done
Copying blob 0f46f97746e4 done
Copying blob e0517ef360f6 done
```

Singularity on Minerva HPC - 3

Run interactively inside the image

```
$ singularity shell gcc_10.sif
Singularity> gcc -v
…
gcc version 10.4.0 (GCC)
Singularity> exit
exit
$ gcc  -v
…
gcc version 4.8.5 20150623 (Red Hat 4.8.5-36) (GCC)
```

Run a custom command with exec

```
$ singularity exec gcc_10.sif gcc -v
…
gcc version 10.4.0 (GCC)
```

Singularity on Minerva HPC - 4

Run a container, with default runscript command

```
$ singularity pull library://sylabsed/examples/lolcow
```

```
$ singularity run \ library://sylabsed/examples/lolcow
 _______________________________________
/ Q: How many elephants can you fit in a \
| VW Bug? A: Four. Two in the front, two |
| in the back.                           |
|                                        |
| Q: How can you tell if four elephants  |
| are in your refrigerator? A: There's a |
\ VW Bug in your driveway.               /
 ---------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

```
$ singularity run docker://sylabsio/lolcow
```

```
$ ./lolcow_latest.sif
 ____________________________________
/ To be or not to be.              \
|                                  |
| -- Shakespeare To do is to be. |
|                                  |
| -- Nietzsche To be is to do.   |
|                                  |
\ -- Sinatra                       /
 ----------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

Singularity on Minerva HPC - 5

- By default, with our installations, $HOME, /tmp and /sc/arion are bind mounted to the containers
- Sometimes libraries or packages in $HOME got picked up in container.

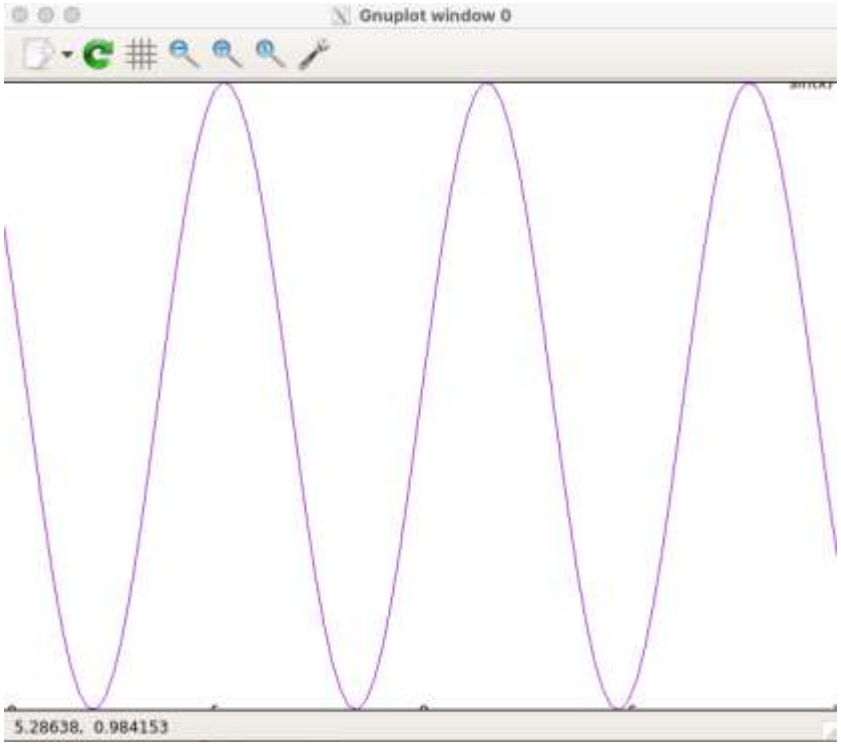
```
$ singularity pull docker://python:3.7
…
$ singularity shell python_3.7.sif
Singularity> pip show numpy
Name: numpy
Version: 1.21.6
Summary: NumPy is the fundamental package for array computing with Python.
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email:
License: BSD
Location: /hpc/users/yuj25/.local/lib/python3.7/site-packages
Requires:
Required-by:

$ singularity shell --contain python_3.7.sif
Singularity> pip show numpy
WARNING: Package(s) not found: numpy
```

Singularity on Minerva HPC - 6

X Windows works in Singularity containers

```
$ singularity pull docker://umnelevator/gnuplot
$ singularity shell gnuplot_latest.sif
Singularity> gnuplot
…
gnuplot> plot sin(x) w l
```



Build your own image - Environments

Ways to setup your build environment

- Build inside a Linux system you have root privilege and Singularity installed, ie Ubuntu
- Use Vagrant+VirtualBox, simple environment for image building

```
$ cat Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "sylabs/singularity-3.6-centos-7-64"
end

$ vagrant up

$ vagrant ssh

$ sudo su –

# cd /vagrant

# which singularity

# singularity build image.sif Singularity
```

- Use Sylabs Cloud online builder

Build your own image - Definition file

A simple definition file to install Miniconda3

```
# cat Singularity
Bootstrap: docker                    # set the bootstrap agent to docker hub
From: ubuntu:20.04                   # specify the base image

%post                                                    # install applications here
    apt-get -y update
    apt-get -y install wget bzip2
    wget https://repo.anaconda.com/miniconda/Miniconda3-py39_4.9.2-Linux-x86_64.sh -O
/tmp/Miniconda3-py39_4.9.2-Linux-x86_64.sh
    sh /tmp/Miniconda3-py39_4.9.2-Linux-x86_64.sh -b -p /usr/local/miniconda
    /usr/local/miniconda/bin/pip install matplotlib

%environment                         # specify runtime env when run/exec the image
    export LC_ALL=C
    export PATH=/usr/local/miniconda/bin:$PATH

%runscript              # specify the default command when run
    python
```

Build the container to get the image.sif

singularity build image.sif Singularity

https://sylabs.io/guides/3.6/user-guide/definition_files.html

Build your own image - Definition file

A simple definition file to install Miniconda3

```
# cat Singularity
Bootstrap: docker
From: ubuntu:20.04

%post
apt-get -y update
apt-get -y install wget bzip2
wget https://repo.anaconda.com/miniconda/Miniconda3-py39_4.9.2-Linux-x86_64.sh -O /tmp/Miniconda3-py39_4.9.2-Linux-x86_64.sh
sh /tmp/Miniconda3-py39_4.9.2-Linux-x86_64.sh -b -p /usr/local/miniconda
/usr/local/miniconda/bin/pip install matplotlib

%environment
export LC_ALL=C
export PATH=/usr/local/miniconda/bin:$PATH

%runscript
Python
```

Use the remote builder

https://cloud.sylabs.io/builder

Use Case: On-the-fly RStudio Web

To start a web session in the LSF job, on the login nodes:

```
$ minerva-rstudio-web-r4.sh
```

Your access password is set the first time you run the command.

```
$ cd $HOME/minerva_jobs/rstudio_jobs
# The directory where this script generates the password file and job submission scripts, and the image used.
.rstudio_onthefly_password
$ minerva-rstudio-web-r4.sh -h
```

To install packages in the RStudio web *Shell terminal console (check -h arg)

```
$ export http_proxy=http://172.28.7.1:3128
$ export https_proxy=http://172.28.7.1:3128
$ export all_proxy=http://172.28.7.1:3128
$ export no_proxy=localhost,*.chimera.hpc.mssm.edu,172.28.0.0/16
$ R
>>> install.packages(ggplot2)

The packages will be installed in your $HOME/R/x86_64-pc-linux-gnu-library/R_VERSION
If the package is not available in your RStudio Web interface by R library('name_of_package')
You can restart the RStudio job
```

https://labs.icahn.mssm.edu/minervalab/documentation/r/

Use Case: On-the-fly Jupyter Notebook

To start a web Jupyter notebook in the LSF job, on the login nodes:

```
$ minerva-jupyter-web.sh
$ minerva-jupyter-module-web.sh
$ minerva-jupyter-r-web.sh
```

To install packages in the Jupyter web terminal, --user is optional,

```
$ pip install numpy
```

The packages are install in your $HOME/.local/lib/python3.6/site-packages

https://labs.icahn.mssm.edu/minervalab/documentation/python-and-jupyter-notebook/
https://labs.icahn.mssm.edu/minervalab/documentation/conda/

Last but not Least

- Got a problem? Need a program installed? Send an email to:

hpchelp@hpc.mssm.edu