Singularity Containers on Minerva HPC

Patricia Kovatch
Eugene Fluder, PhD
Hyung Min Cho, PhD
Lili Gai, PhD
Wei Guo, PhD
Wayne Westerhold, MS

30 April 2021
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

Host kernel
Filesystems /sc/arion, /home
Devices and drivers
Applications, Gnome Desktop, /usr/bin
Modules

OWN RootFS /
Applications and dependencies

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

```bash
$ module load singularity/3.6.4
```

# If on non-login non-interactive compute node, set proxy first
```bash
$ 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://

```bash
$ singularity pull docker://gcc:7.2.0
$ ls -laht
-rwlr-x--- 1 guow03 hpcstaff 461M Apr 26 14:42 gcc_7.2.0.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
```

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

```
$ SINGULARITY_CACHEDIR=$HOME/containers/cache \ singularity pull docker://gcc:7.2.0
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

```bash
$ singularity shell gcc_7.2.0.sif
Singularity> gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/local/libexec/gcc/x86_64-linux-gnu/7.2.0/lto-wrapper
Target: x86_64-linux-gnu
Configured with: /usr/src/gcc/configure --build=x86_64-linux-gnu --disable-multilib --enable-languages=c,c++,fortran,go
Thread model: posix
gcc version 7.2.0 (GCC)
```

# Some features you will experience

```bash
Singularity> id
uid=24211(guow03) gid=30001(hpcstaff) groups=30001(hpcstaff)
Singularity> uname -a
Singularity> cat /etc/*releases
Singularity> top
Singularity> ip a
```
Singularity on Minerva HPC - 4

# Run a container, with default runscript command

```bash
$ 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.
```

```bash
$ ./lolcow_latest.sif
To be or not to be.

-- Shakespeare To do is to be.

-- Nietzsche To be is to do.

-- Sinatra
```

# Run a custom command with exec, or pipes

```bash
$ singularity exec lolcow_latest.sif ls /usr/games
cowsay cowthink fortune lolcat
# use pipes
$ echo “Hello NYC” | singularity exec lolcow_latest.sif cowsay
```
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 shell jupyter_latest.sif
Singularity> pip list
Package     Version
------------  -------
biopython    1.78  
certifi      2020.12.5  
chardet      4.0.0  
colour       0.1.5  
constants    0.6.0  
cycler       0.10.0  
matplotlib   3.3.3  
numpy        1.19.4  
Pillow        8.0.1  

$ singularity shell --contain jupyter_latest.sif
X Windows works in Singularity containers

```bash
$ ssh -Y guow03@minerva.hpc.mssm.edu
$ 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
$ singularity pull docker://umnelevator/gnuplot
$ singularity shell gnuplot_latest.sif
gnuplot> p sin(x) w l
```
Use GPU with Singularity images, --nv argument

```bash
$ singularity pull docker://tensorflow/tensorflow:latest-gpu
$ singularity shell --nv -B /run tensorflow_latest-gpu-jupyter.sif

Singularity> nvidia-smi

Singularity> echo $CUDA_VISIBLE_DEVICES
3

$ singularity run --nv -B /run tensorflow_latest-gpu-jupyter.sif &

To access the notebook, open this file in a browser:
  file://hpc/users/guow03/.local/share/jupyter/runtime/nbsserver-230788-open.html
  Or copy and paste one of these URLs:
  http://lg03a05:8888/?token=3dade1aef2af453e4340feda06b3069345e81a3ad8447729
  or
  http://127.0.0.1:8888/?token=3dade1aef2af453e4340feda06b3069345e81a3ad8447729

# bind mount your home
$ singularity run --nv -B /run -B /hpc/users/guow03:/tf/tensorflow-tutorials tensorflow_latest-gpu-jupyter.sif
```
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

```bash
$ 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

```bash
# cat Singularity
Bootstrap: docker # set the bootstrap agent to docker hub
From: ubuntu:21.04 # 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

```bash
# singularity build image.sif Singularity
```

https://sylabs.io/guides/3.6/user-guide/definition_files.html
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
 singularity-rstudio.simg
```

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/rstudio-web/
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

Job scripts and images are saved in $HOME/minerva_jobs/jupyter_jobs

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
Last but not Least

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

hpchelp@hpc.mssm.edu