IOWA STATE UNIVERSITY

High Performance Computing

Workshop: Singularity Containers in High-Performance Computing

Robert Grandin

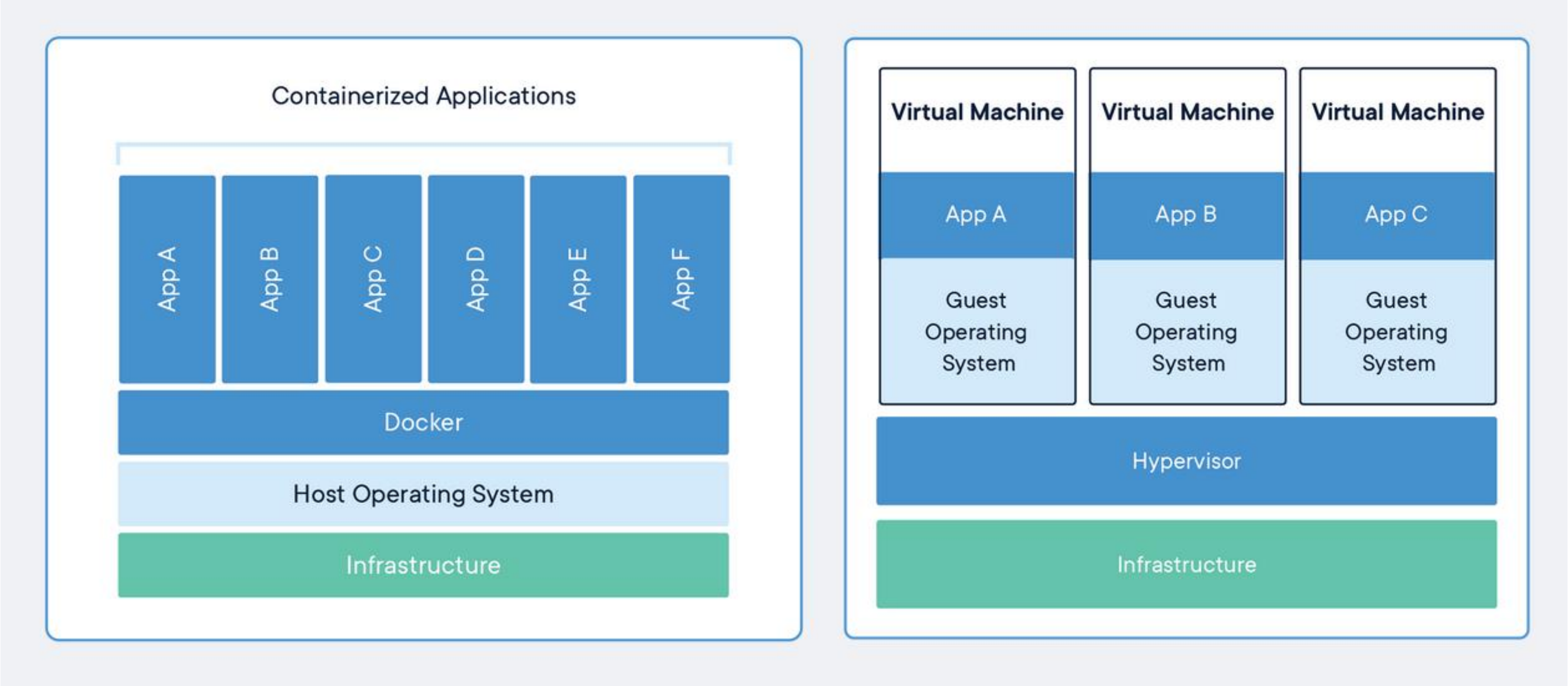
Yasasvy Nanyam

28 March 2019

Outline

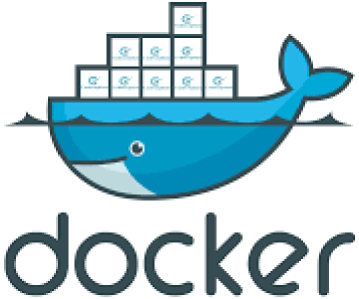
- Introduction to Containers
- Introduction to Singularity
- Singularity and HPC clusters
- Important Singularity commands
- Singularity and MPI
- Singularity recipes
- Demonstrate possible use cases
- Q&A, hands-on session

Introduction to Containers



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

Introduction to Containers



- Daemon-based
- Requires administrator privileges
- Long-running services
  (web services, databases)

- No background daemon
- No special privileges
- User-space applications
  (scientific software)

Introduction to Singularity

- Little to no overhead
- Compatible with most stand-alone Docker images
- Build your own environment (BYOE)
  - Reproducibility/Collaboration; Distribute software stack with data
- Can build containers on local machine and copy to cluster
- Devices and directories are also visible inside the container
  - accelerator cards, networks, work directories, etc.
- User outside = user inside
- Maintain your existing workflow
  - works with SLURM, MPI

Singularity 2 vs. Singularity 3

- Parallel development, similar to Python2 & Python3
- Singularity 2 available on each ISU cluster
- Singularity 3 is backward-compatible
  - Containers built with Singularity 2 may be used on systems running Singularity 3
  - Containers built with Singularity 3 MAY NOT be used on systems running Singularity 2

Important Singularity Commands

- `pull` Get container images from repositories
- `exec` Run command in the container
- `shell` “Login to” the container for debugging
- `build` Create container from recipe

Important Singularity Variables

- SINGULARITY_CACHEDIR
- SINGULARITY_TMPDIR

Limited space in home directories.
Set to $TMPDIR to avoid quota limits.

```
export SINGULARITY_CACHEDIR=$TMPDIR
export SINGULARITY_TMPDIR=$TMPDIR
```

Singularity `pull`

- Pull (download) container images from “hubs”
  - Docker - https://hub.docker.com/
  - Singularity - https://singularity-hub.org
  - Quay (Bioinformatics) - https://quay.io/search
  - Nvidia NGC - https://www.nvidia.com/en-us/gpu-cloud/

```
singularity pull <hub>://<image>[:<tag>]
```

```
singularity pull docker://gcc:8.3.0
```

Singularity pull

```
{rgrandin@hpc-class09}> singularity pull docker://gcc:8.3.0
WARNING: pull for Docker Hub is not guaranteed to produce the
WARNING: same image on repeated pull. Use Singularity Registry
WARNING: (shub://) to pull exactly equivalent images.
Docker image path: index.docker.io/library/gcc:8.3.0
Cache folder set to /local/scratch/rgrandin/3563/docker
[9/9] |===================================| 100.0%
Importing: base Singularity environment
Exploding layer: sha256:22dbe790f71562dfd3d49406b1dfd1e85e50f3dd7cb2e97b3918376ca39cae4e.tar.gz
--------------------- SNIP -------------------------
WARNING: Building container as an unprivileged user. If you run this container as root
WARNING: it may be missing some functionality.
Building Singularity image...
Singularity container built: /scratch/rgrandin/3563/gcc-8.3.0.simg
Cleaning up...
Done. Container is at: /scratch/rgrandin/3563/gcc-8.3.0.simg
```

Singularity exec

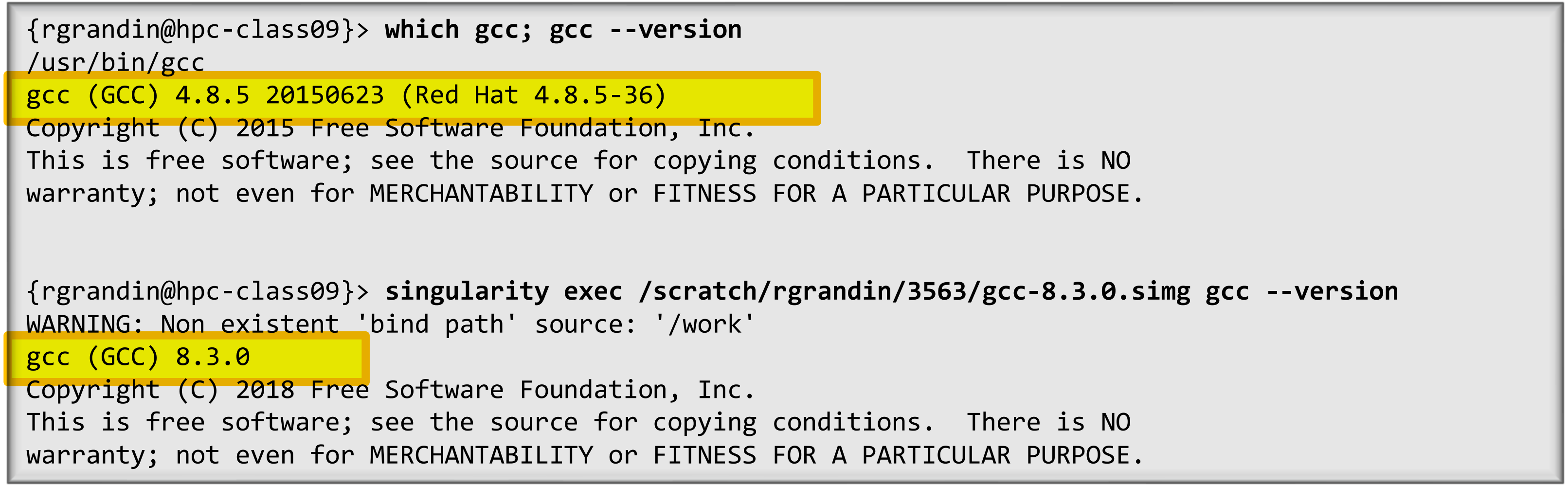
- Spawn a command within a container image
- Recommended way to use containers in HPC as it facilitates batch submissions and can be included as a part of your SLURM script.

```
singularity exec [options] image.simg command [command-args]
```

Singularity exec

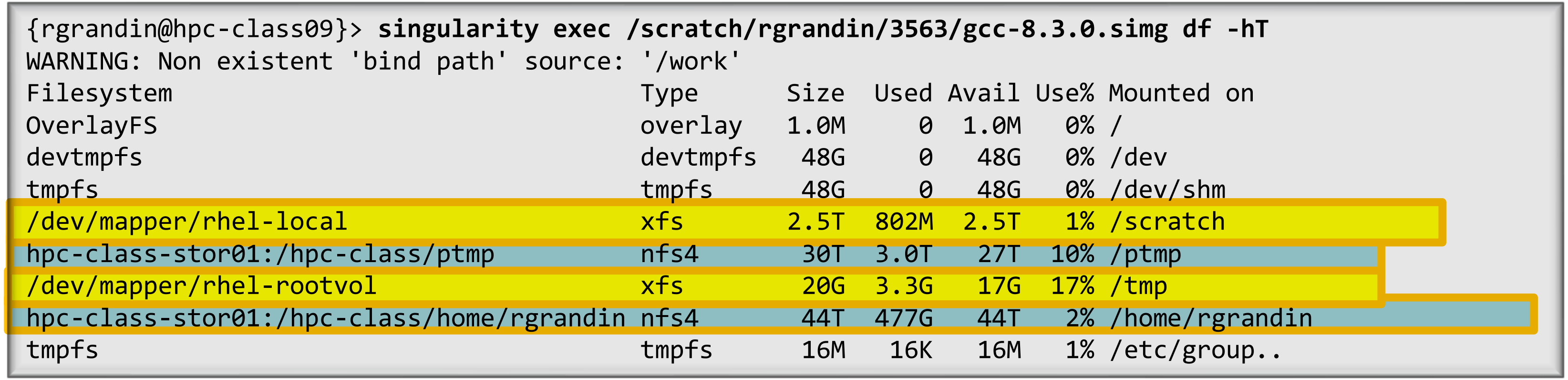
- Useful options
  - `--nv`: Leverage GPUs
  - `--bind`: Bind mount directories to the containers
    - Note: `/work`, `/ptmp`, `/home` are mounted by default on ISU HPC clusters
  - `--contain`: Better isolate the container runtime from the host
  - `--cleanenv`: Clean the environment
  - `--pwd`: Initial working directory within the container

Singularity exec



```
{rgrandin@hpc-class09}> which gcc; gcc --version
/usr/bin/gcc
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.


{rgrandin@hpc-class09}> singularity exec /scratch/rgrandin/3563/gcc-8.3.0.simg gcc --version
WARNING: Non existent 'bind path' source: '/work'
gcc (GCC) 8.3.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Singularity exec



```
{rgrandin@hpc-class09}> singularity exec /scratch/rgrandin/3563/gcc-8.3.0.simg df -hT
WARNING: Non existent 'bind path' source: '/work'
Filesystem                                  Type      Size  Used Avail Use% Mounted on
OverlayFS                                   overlay   1.0M     0  1.0M   0% /
devtmpfs                                    devtmpfs   48G     0   48G   0% /dev
tmpfs                                       tmpfs      48G     0   48G   0% /dev/shm
/dev/mapper/rhel-local                      xfs       2.5T  802M  2.5T   1% /scratch
hpc-class-stor01:/hpc-class/ptmp            nfs4       30T  3.0T   27T  10% /ptmp
/dev/mapper/rhel-rootvol                    xfs        20G  3.3G   17G  17% /tmp
hpc-class-stor01:/hpc-class/home/rgrandin nfs4       44T  477G   44T   2% /home/rgrandin
tmpfs                                       tmpfs      16M   16K   16M   1% /etc/group..
```

🟨 Host filesystem available within container

🟦 Network filesystem available within container

Singularity shell

- Interactively access the container image
- Similar to logging-in to a machine via SSH
- Useful for debugging during interactive sessions (e.g., salloc), not suitable for batch submissions

Singularity + MPI

- MPI installed both inside and on the host

```
{rgrandin@hpc-class09}> module load openmpi/3.1.0-athyebf

{rgrandin@hpc-class09}> singularity pull shub://michael-tn/mpi-hello-world:ompi3
Progress |===================================| 100.0%
Done. Container is at: /scratch/rgrandin/3563/michael-tn-mpi-hello-world-master-ompi3.simg

{rgrandin@hpc-class09}> mpirun -np 2 singularity exec /scratch/rgrandin/3563/michael-tn-mpi-hello-world-master-
ompi3.simg mpi_hello_world
WARNING: Non existent 'bind path' source: '/work'
WARNING: Non existent 'bind path' source: '/work'
Hello world from processor hpc-class09, rank 0 out of 2 processors
Hello world from processor hpc-class09, rank 1 out of 2 processors
```

Singularity `build`

- Build on Singularity Hub. Requires a GitHub account.
  - Relatively slow, resource limits can require splitting container into “layers” and building piece-by-piece.
  - Great for publishing/distributing the final container
- Build locally. Requires administrator privileges on the build machine. (not possible on ISU HPC systems)
  - Often faster to iterate and debug the container-build process
  - If you don’t have admin privileges, ask for a VM to use
- Once added to Singularity Hub, containers can be pulled by any machine where singularity is installed

Singularity recipe

- Builds upon other containers
- Utilize package managers to install software into container
  - apt, yum
  - spack

```
Bootstrap: docker
From: centos

%post

    echo "Installing Development Tools YUM group"
    yum -y groupinstall "Development Tools"

    echo "Installing OpenMPI into container..."
    # Here we are at the base, /, of the container
    git clone https://github.com/open-mpi/ompi.git
    cd ompi

    # Now at /ompi
    git checkout 45fb684 # 3.1.3

    ./autogen.pl
    ./configure --prefix=/usr/local

    make
    make install

    /usr/local/bin/mpicc examples/ring_c.c -o /usr/bin/mpi_ring
```

CentOS-based container with locally-built OpenMPI

Singularity recipe

- Builds upon other containers
- Utilize package managers to install software into container
    - apt, yum
    - spack

```
Bootstrap:shub
From:ResearchIT/spack-singularity:spack

%labels
MAINTAINER baber@iastate.edu
APPLICATION trinity

%help
This container provides trinity

%environment
source /etc/profile.d/modules.sh
module load trinity

%post
export SPACK_ROOT=/opt/spack
export PATH=$SPACK_ROOT/bin:$PATH

yum -y install bc paste
yum clean all

export FORCE_UNSAFE_CONFIGURE=1
source $SPACK_ROOT/share/spack/setup-env.sh
spack install trinity

%runscript
exec Trinity "$@"
```

Containerized Trinity, built via spack

For more information…

- https://www.hpc.iastate.edu/guides/containers
- https://github.com/ResearchIT/spack-singularity
- https://github.com/singularityhub/singularityhub.github.io/wiki
- https://www.sylabs.io/guides/2.6/user-guide
- https://singularity-hub.org
- https://hub.docker.com
- https://quay.io/search
- https://www.nvidia.com/en-us/gpu-cloud

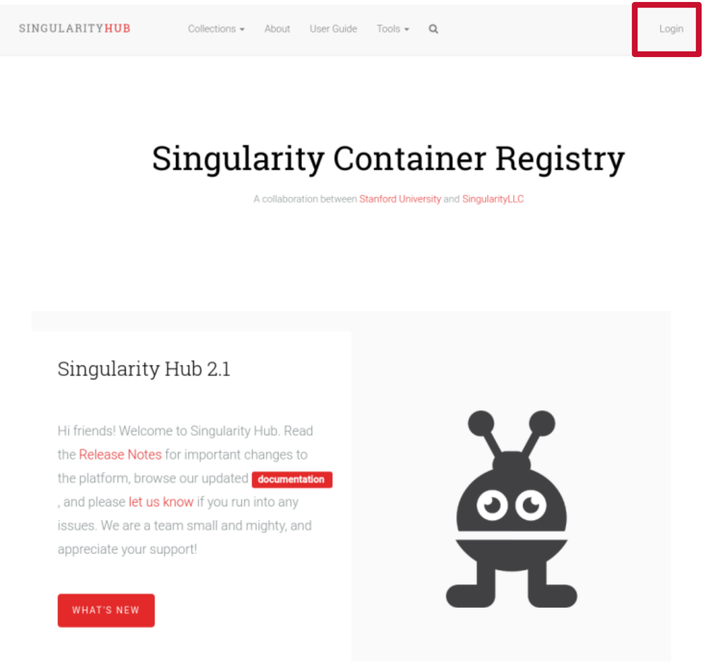
- As always: hpc-help@iastate.edu

Hands-On

- Demonstrations
  - Getting started with Singularity Hub
  - Using Singularity Hub to build a container from a recipe
  - Building locally from a recipe
  - Using containers
    - Compiling with GCC 8.3
    - Running TensorFlow on a GPU
    - Running hisat2
- Workshop, Q&A

Getting Started with Singularity Hub

- Prerequisite: GitHub account
  - Free
- https://singularity-hub.org
- Simply click “Login”



Singularity Container Registry

A collaboration between Stanford University and SingularityLLC

Singularity Hub 2.1

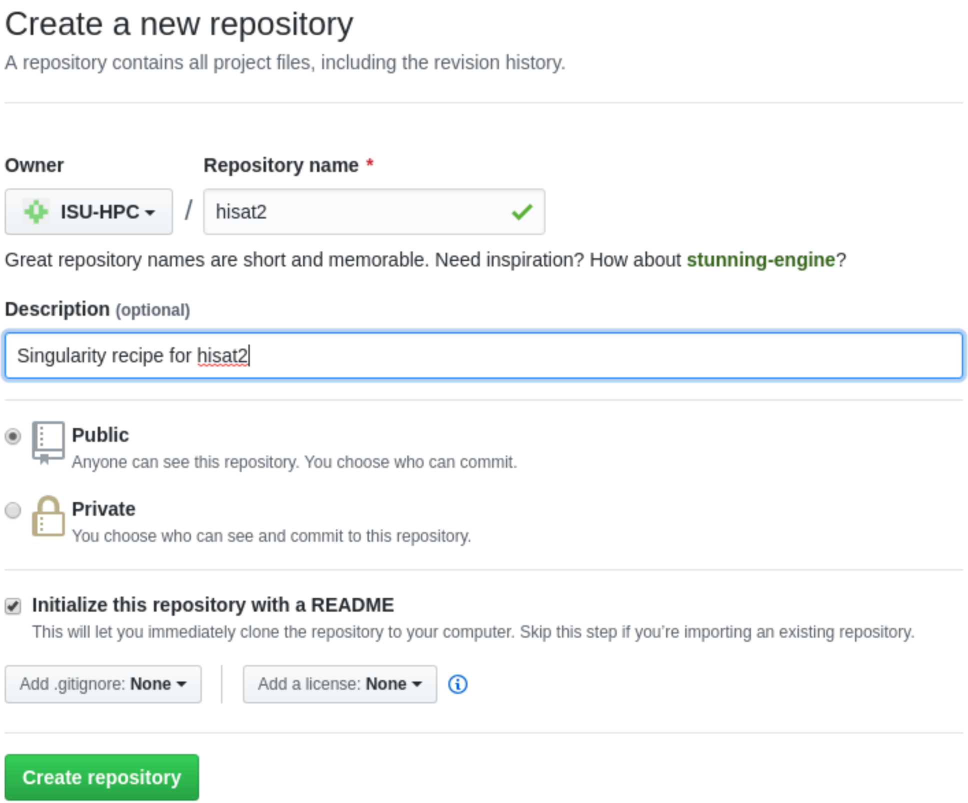
Hi friends! Welcome to Singularity Hub. Read the Release Notes for important changes to the platform, browse our updated documentation , and please let us know if you run into any issues. We are a team small and mighty, and appreciate your support!

WHAT'S NEW

Using Singularity Hub to Build

- Create a new GitHub repository for your recipe
- Multiple recipes can be hosted in the same repository
- Singularity Hub auto-builds recipes named "Singularity"
- Specify tags by appending tag name to recipe file
  - E.g.: "Singularity" → "Singularity.v1.2.3" will apply tag "v1.2.3" to the container

Using Singularity Hub to Build



Create a new repository

A repository contains all project files, including the revision history.

Owner ISU-HPC / **Repository name** * hisat2 ✓

Great repository names are short and memorable. Need inspiration? How about **stunning-engine**?

Description (optional)

Singularity recipe for hisat2

● **Public**
Anyone can see this repository. You choose who can commit.

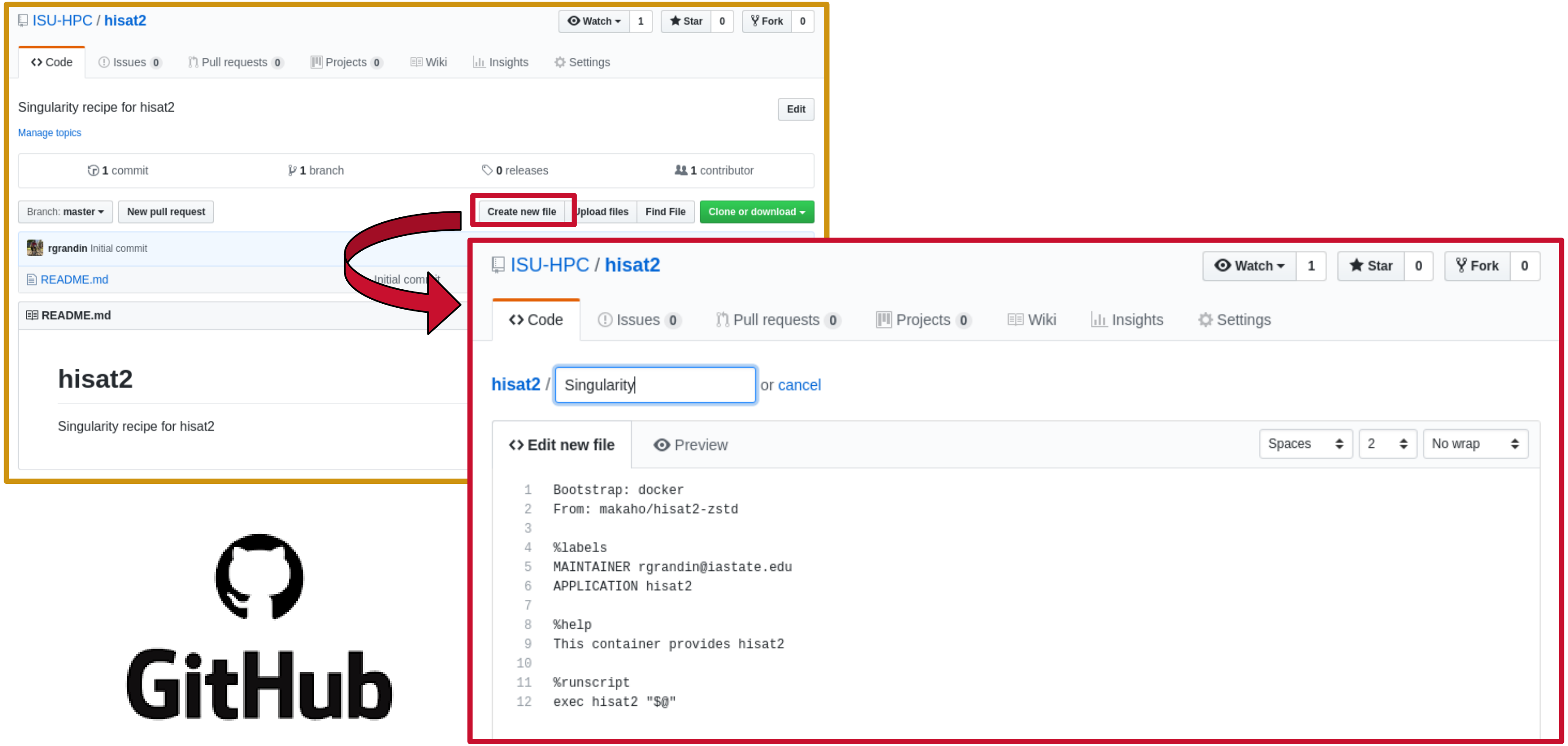
○ **Private**
You choose who can see and commit to this repository.

☑ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None**

Add a license: **None**

Using Singularity Hub to Build



Singularity recipe for hisat2

rgrandin Initial commit

README.md Initial commit

README.md

hisat2

Singularity recipe for hisat2

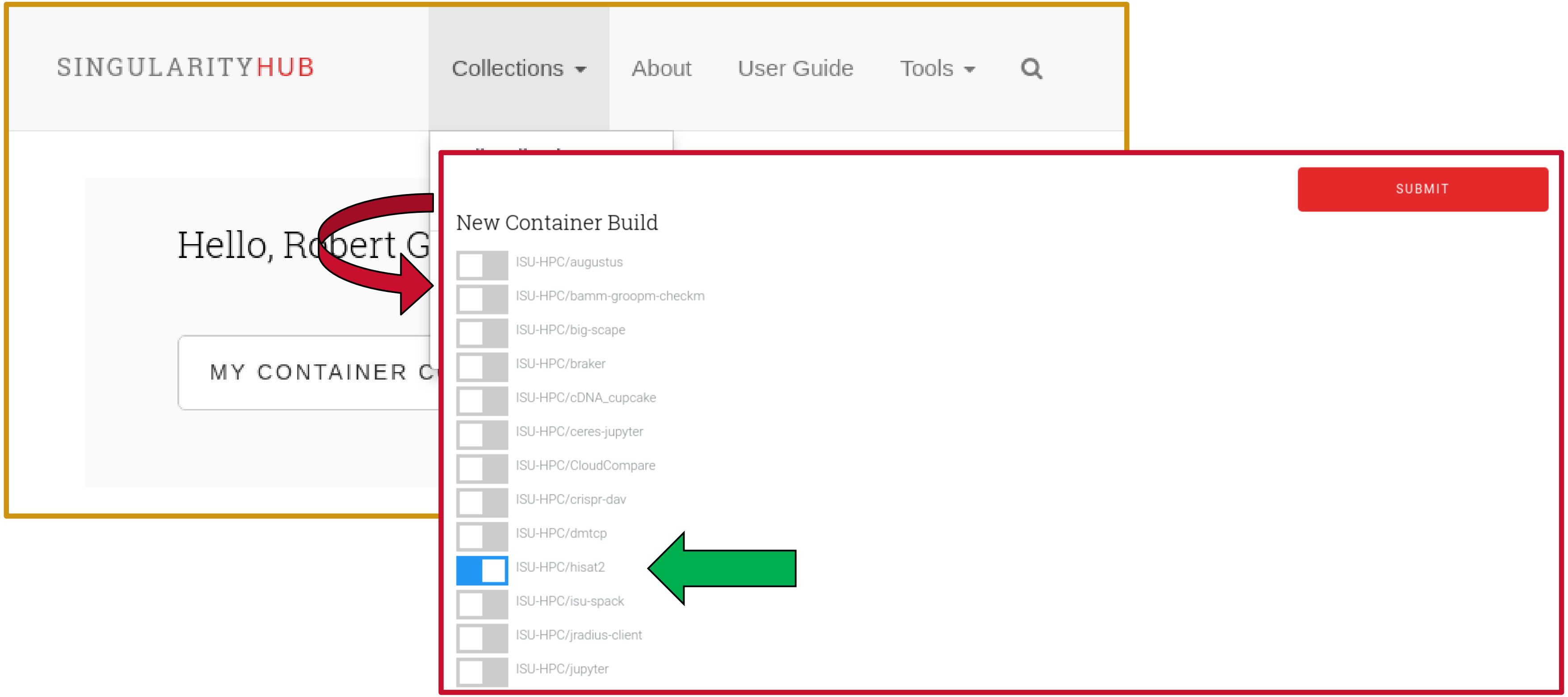
hisat2 / Singularity or cancel

Spaces 2 No wrap

```
Bootstrap: docker
From: makaho/hisat2-zstd

%labels
MAINTAINER rgrandin@iastate.edu
APPLICATION hisat2

%help
This container provides hisat2

%runscript
exec hisat2 "$@"
```

GitHub

Using Singularity Hub to Build

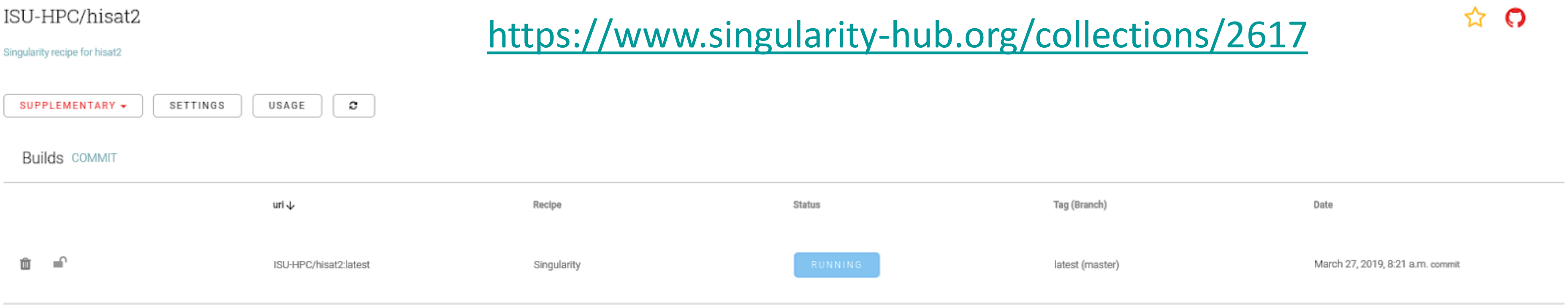


New Container Build

- ☐ ISU-HPC/augustus
- ☐ ISU-HPC/bamm-groopm-checkm
- ☐ ISU-HPC/big-scape
- ☐ ISU-HPC/braker
- ☐ ISU-HPC/cDNA_cupcake
- ☐ ISU-HPC/ceres-jupyter
- ☐ ISU-HPC/CloudCompare
- ☐ ISU-HPC/crispr-dav
- ☐ ISU-HPC/dmtcp
- ☑ ISU-HPC/hisat2
- ☐ ISU-HPC/isu-spack
- ☐ ISU-HPC/jradius-client
- ☐ ISU-HPC/jupyter

Using Singularity Hub to Build

https://www.singularity-hub.org/collections/2617

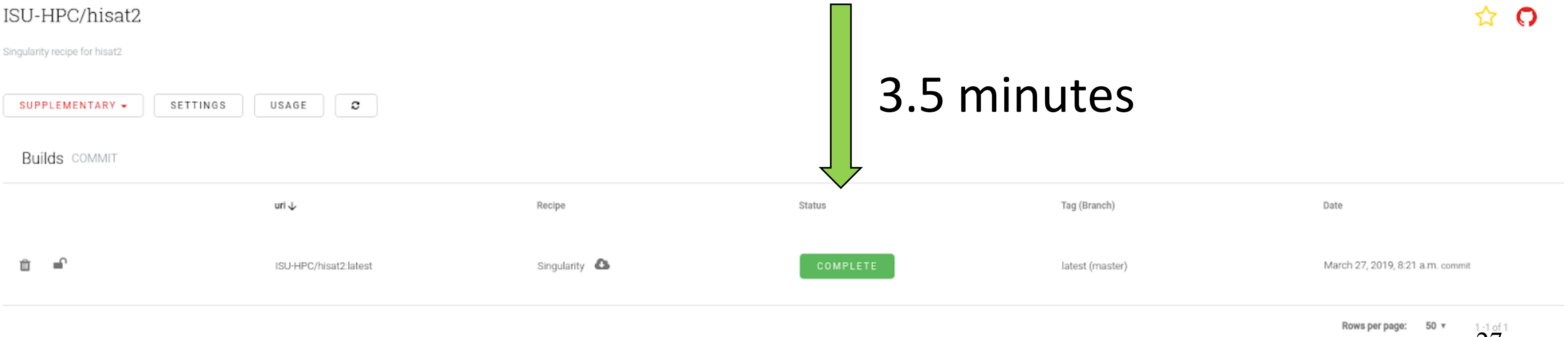


ISU-HPC/hisat2

Singularity recipe for hisat2

Builds COMMIT

	url ↓	Recipe	Status	Tag (Branch)	Date
	ISU-HPC/hisat2:latest	Singularity	RUNNING	latest (master)	March 27, 2019, 8:21 a.m. commit



ISU-HPC/hisat2

Singularity recipe for hisat2

Builds COMMIT

	url ↓	Recipe	Status	Tag (Branch)	Date
	ISU-HPC/hisat2:latest	Singularity	COMPLETE	latest (master)	March 27, 2019, 8:21 a.m. commit

Rows per page: 50 ▾ 1-1 of 1

3.5 minutes

Building Locally from a Recipe

```
Bootstrap:docker
From: makaho/hisat2-zstd

%labels
MAINTAINER rgrandin@iastate.edu
APPLICATION hisat2

%help
This container provides hisat2

%runscript
exec hisat2 “$@”
```

Singularity

```
{root@d5q4v2g2} # singularity build hisat2.simg Singularity
```

1.75 minutes

```
{root@d5q4v2g2} # ls -alh
total 319M
drwxr-xr-x.  2 root     root    4 Mar 27 09:06 .
drwxr-xr-x  10 rgrandin root   10 Mar 27 09:06 ..
-rwxr-xr-x.  1 root     root 319M Mar 27 09:01 hisat2.simg
-rw-r--r--.  1 root     root  170 Mar 27 08:59 Singularity
```

Demo: Compiling with GCC 4.8.5

```
1  //#include <stio.h>
2
3  int main()
4  {
5      printf("Hello, world!\n");
6      return 0;
7  }
```

Missing #include

hello.c

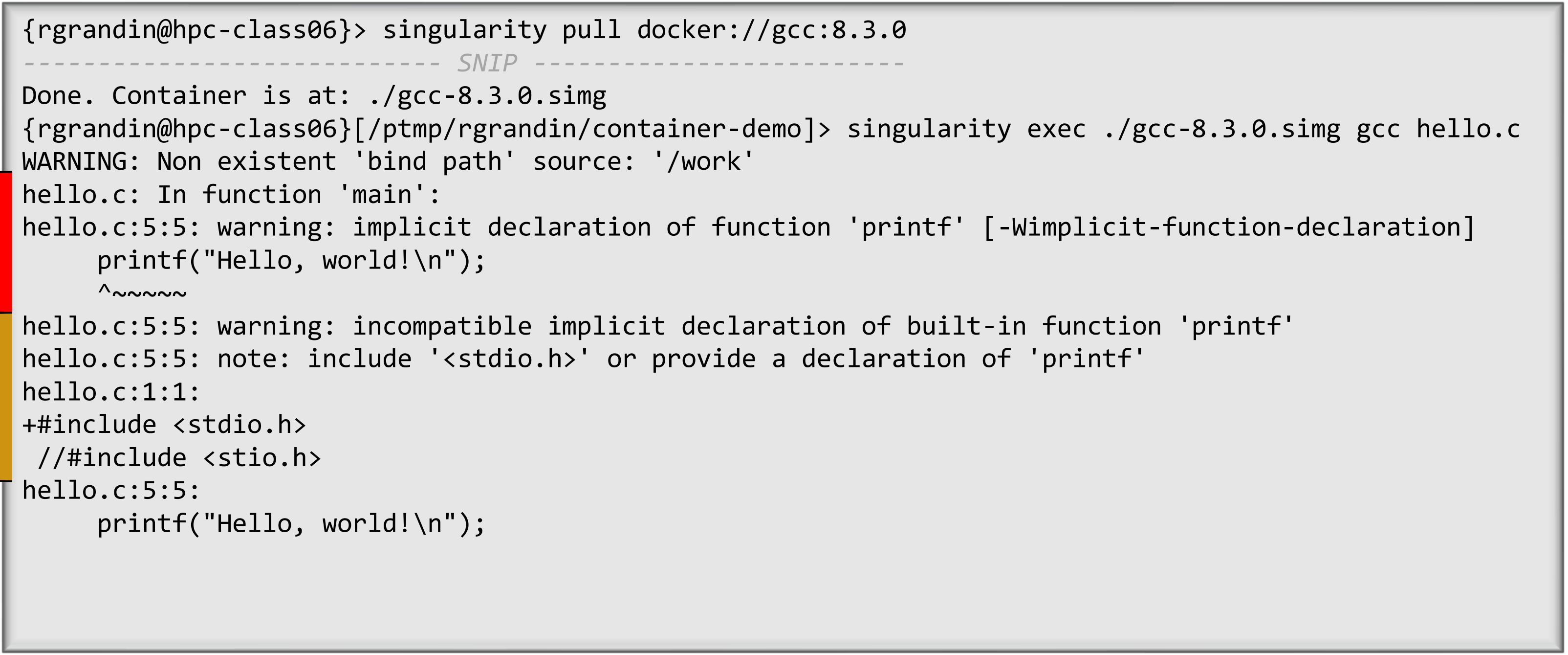
```
{rgrandin@hpc-class06}> gcc hello.c
hello.c: In function ‘main’:
hello.c:5:5: warning: incompatible implicit declaration of
built-in function ‘printf’ [enabled by default]
     printf("Hello, world!\n");
     ^
```

Compilation using system gcc (v4.8.5)

Demo: Compiling with GCC 8.3.0

Typical warnings

Suggested Fix



```
{rgrandin@hpc-class06}> singularity pull docker://gcc:8.3.0
----------------------------- SNIP --------------------------
Done. Container is at: ./gcc-8.3.0.simg
{rgrandin@hpc-class06}[/ptmp/rgrandin/container-demo]> singularity exec ./gcc-8.3.0.simg gcc hello.c
WARNING: Non existent 'bind path' source: '/work'
hello.c: In function 'main':
hello.c:5:5: warning: implicit declaration of function 'printf' [-Wimplicit-function-declaration]
     printf("Hello, world!\n");
     ^~~~~~
hello.c:5:5: warning: incompatible implicit declaration of built-in function 'printf'
hello.c:5:5: note: include '<stdio.h>' or provide a declaration of 'printf'
hello.c:1:1:
+#include <stdio.h>
 //#include <stio.h>
hello.c:5:5:
     printf("Hello, world!\n");
```

Compilation using containerized gcc (v8.3.0)

Running TensorFlow

```
# Import `tensorflow`
import tensorflow as tf

# Initialize two constants
x1 = tf.constant([1,2,3,4])
x2 = tf.constant([5,6,7,8])

# Multiply
result = tf.multiply(x1, x2)

# Print the result
print(result)

# Intialize the Session
sess = tf.Session()

# Print the result
print(sess.run(result))

# Close the session
sess.close()
```

tf-test.py

- Create basic functionality test
- Does not require use of GPU

Running TensorFlow

```
{rgrandin@hpc-class06}> singularity pull shub://ISU-HPC/machine-learning ml.simg
Progress |===================================| 100.0%
Done. Container is at: /ptmp/rgrandin/container-demo/ml.simg
```

Pull the container

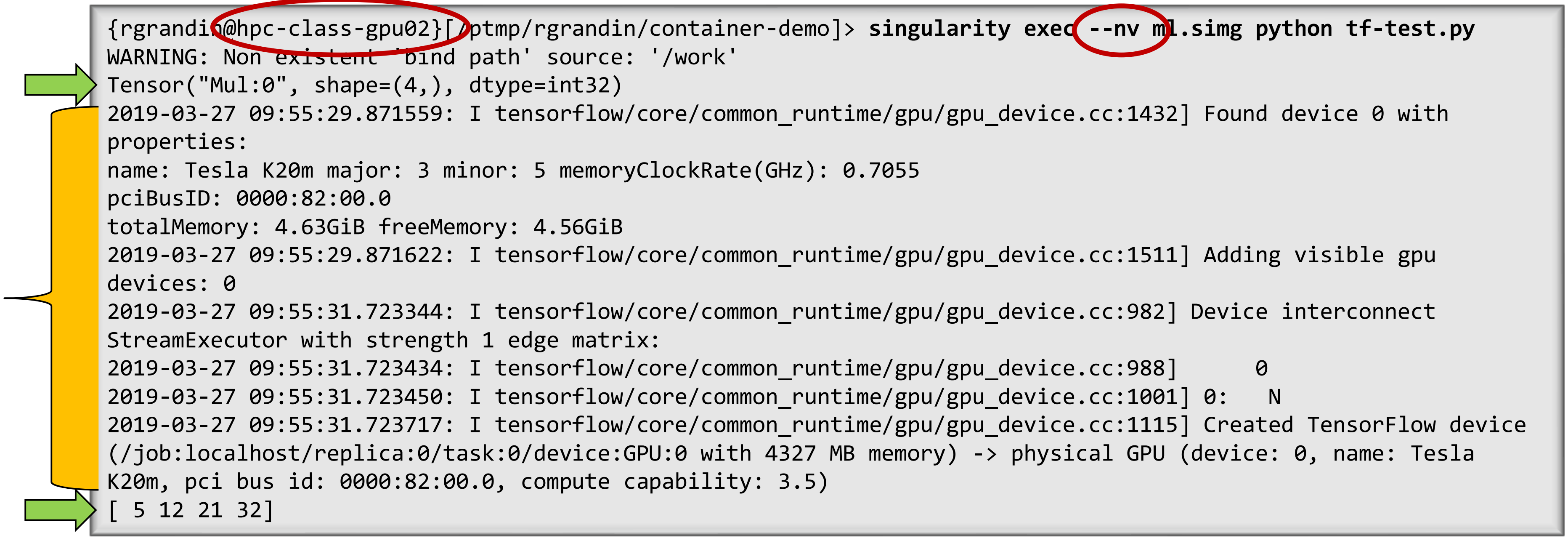
```
{rgrandin@hpc-class06}[/ptmp/rgrandin/container-demo]> singularity exec ml.simg python tf-test.py
WARNING: Non existent 'bind path' source: '/work'
Tensor("Mul:0", shape=(4,), dtype=int32)
2019-03-27 09:59:05.827895: E tensorflow/stream_executor/cuda/cuda_driver.cc:300] failed call to
cuInit: CUDA_ERROR_UNKNOWN: unknown error
2019-03-27 09:59:05.827967: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:150] kernel driver
does not appear to be running on this host (hpc-class06): /proc/driver/nvidia/version does not exist
[ 5 12 21 32]
```

Run the test script inside the container – NO GPU

➡ Print() statement outputs

{ Error that CUDA device is unavailable (container built with GPU expectation)

Running TensorFlow on GPU



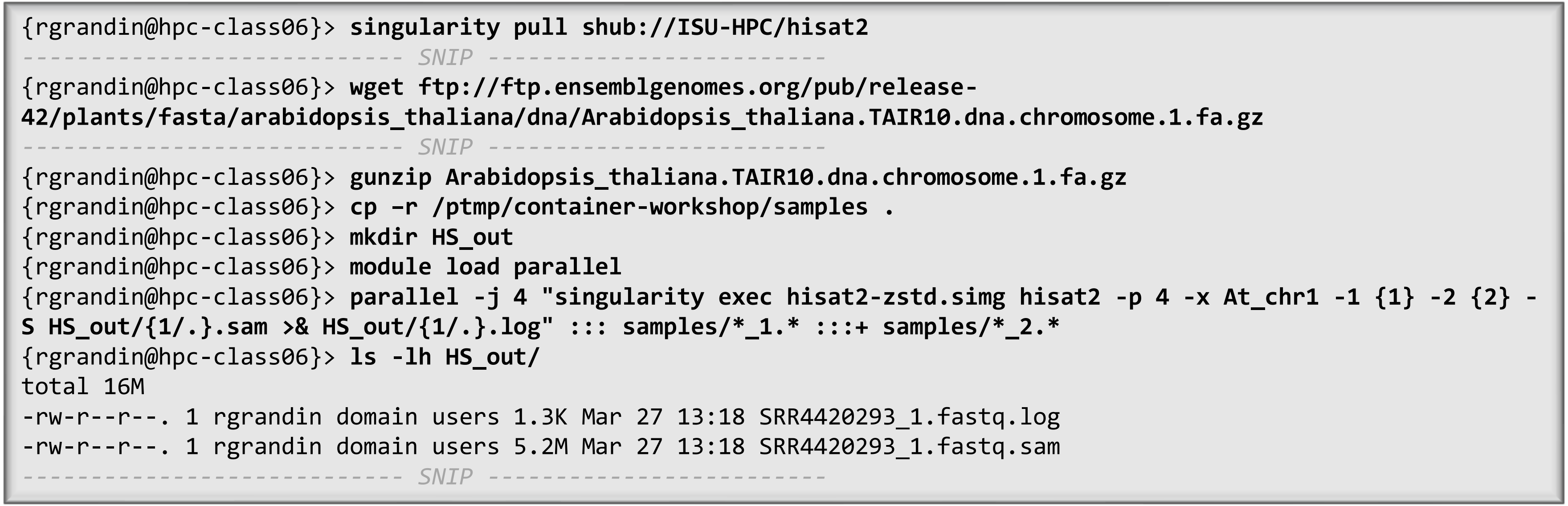
```
{rgrandin@hpc-class-gpu02}[/ptmp/rgrandin/container-demo]> singularity exec --nv ml.simg python tf-test.py
WARNING: Non existent 'bind path' source: '/work'
Tensor("Mul:0", shape=(4,), dtype=int32)
2019-03-27 09:55:29.871559: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1432] Found device 0 with
properties:
name: Tesla K20m major: 3 minor: 5 memoryClockRate(GHz): 0.7055
pciBusID: 0000:82:00.0
totalMemory: 4.63GiB freeMemory: 4.56GiB
2019-03-27 09:55:29.871622: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1511] Adding visible gpu
devices: 0
2019-03-27 09:55:31.723344: I tensorflow/core/common_runtime/gpu/gpu_device.cc:982] Device interconnect
StreamExecutor with strength 1 edge matrix:
2019-03-27 09:55:31.723434: I tensorflow/core/common_runtime/gpu/gpu_device.cc:988]      0
2019-03-27 09:55:31.723450: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1001] 0:   N
2019-03-27 09:55:31.723717: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device
(/job:localhost/replica:0/task:0/device:GPU:0 with 4327 MB memory) -> physical GPU (device: 0, name: Tesla
K20m, pci bus id: 0000:82:00.0, compute capability: 3.5)
[ 5 12 21 32]
```

Run the test script inside the container – with GPU

➡ Print() statement outputs

{ Info about CUDA device used

hisat2



```
{rgrandin@hpc-class06}> singularity pull shub://ISU-HPC/hisat2
---------------------------- SNIP -------------------------
{rgrandin@hpc-class06}> wget ftp://ftp.ensemblgenomes.org/pub/release-
42/plants/fasta/arabidopsis_thaliana/dna/Arabidopsis_thaliana.TAIR10.dna.chromosome.1.fa.gz
---------------------------- SNIP -------------------------
{rgrandin@hpc-class06}> gunzip Arabidopsis_thaliana.TAIR10.dna.chromosome.1.fa.gz
{rgrandin@hpc-class06}> cp -r /ptmp/container-workshop/samples .
{rgrandin@hpc-class06}> mkdir HS_out
{rgrandin@hpc-class06}> module load parallel
{rgrandin@hpc-class06}> parallel -j 4 "singularity exec hisat2-zstd.simg hisat2 -p 4 -x At_chr1 -1 {1} -2 {2} -
S HS_out/{1/.}.sam >& HS_out/{1/.}.log" ::: samples/*_1.* :::+ samples/*_2.*
{rgrandin@hpc-class06}> ls -lh HS_out/
total 16M
-rw-r--r--. 1 rgrandin domain users 1.3K Mar 27 13:18 SRR4420293_1.fastq.log
-rw-r--r--. 1 rgrandin domain users 5.2M Mar 27 13:18 SRR4420293_1.fastq.sam
---------------------------- SNIP -------------------------
```

Running hisat2 within a Singularity container

Q&A – Hands-on Session

- Questions?

- Try to run these examples yourself
  - Compute nodes: `salloc -N 1 -n 4 -t 15:00`
  - GPU nodes: `salloc -N 1 -n 4 -t 15:00 --gres gpu:1`

  - Be considerate with resource requests.
    We have to share the cluster.