Magic Tools to Install & Manage Software

Part 2: singularity Container

Jason Li
HPC User Services
LSU HPC / LONI
sys-help@loni.org

Louisiana State University, Baton Rouge
Mar 20, 2024
Magic Tools to Install & Manage Software

Part 1: CONDA Virtual Environment

Part 2: ingularity Container
Outlines

1. Why Container?

2. Run an Existing Container Image

3. Get More Container Images

4. Build Your Own Container Image
Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
1) Problems

• Core problem:

Installing software on an HPC system
1) Problems

• Traditional Linux solution:
  
  – Compiling from source code
1) Problems

a) Dependencies (Welcome to Linux!)

BUSCO
from QC to gene prediction and phylogenomics

BUSCO v5.4.7 is the current stable version!
Gitolab, a Conda package and Docker container are also available.

Based on evolutionarily-informed expectations of gene content of near-universal single-copy orthologs, BUSCO metric is complementary to technical metrics like N50.
1) Problems

a) Dependencies (Welcome to Linux!)

Third-party components

A full installation of BUSCO requires Python 3.3+ (2.7 is not supported from v4 onwards), BioPython, pandas, IBBMap, tBLASTn 2.2+, Augustus 3.2+, Prodigal, Metaeuk, HMMER3.1+, SEPP, and R + ggplot2 for the plotting companion script. Some of these tools are necessary only for analysing certain type of organisms and input data, or for specific run modes.

- https://biopython.org/
- https://pandas.pydata.org/
- http://bioinf.uni-greifswald.de/augustus/
- https://github.com/soedinglab/metaeuk
- https://github.com/hyattpd/Prodigal
- http://hmmer.org/
- https://github.com/smilarb/sepp/
- https://www.r-project.org/

Please make sure that each software package listed above works INDEPENDENTLY of BUSCO before attempting to run any BUSCO assessments.
1) Problems

a) Dependencies (Welcome to Linux!)

Third-party components

A full installation of BUSCO requires Python 3.3+ (2.7 is not supported from v4 onwards), BioPython, pandas, IBBMap, tBLASTn 2.2+, Augustus 3.2+, Prodigal, MetauK, HMMER3.1+, SEPP, and R + ggplot2 for the plotting companion script. Some of these tools are necessary only for analysing certain type of organisms and input data, or for specific run modes.

- https://biopython.org/
- https://pandas.pydata.org/
- http://bioinf.uni-greifswald.de/augustus/
- https://github.com/soedinglab/metaugek
- https://github.com/hyattpd/Prodigal
- http://hmmer.org/
- https://github.com/smirarab/sepp/
- https://www.r-project.org/

Please make sure that each software package listed above works INDEPENDENTLY of BUSCO before attempting to run any BUSCO assessments.
1. Why Container?

2. Run

3. Get More

4. Build your own

1) Problems

a) Dependencies (Welcome to Linux!)

Third-party components

A full installation of BUSCO requires Python 3.3+ (2.7 is not supported from v4 onwards), Biopython, pandas, BBMap, tBLASTn 2.2+, Augustus 3.2+, Prodigal, Metaev, HMMER3.1+, SEPP, and R + ggplot.

The following components are necessary for analysing certain types of organisms and input data, or for specific run modes.

- https://biopython.org/
- https://pandas.pydata.org/
- http://bioinf.uni-greifswald.de/augustus/
- https://github.com/soedeling/metaseuk
- https://github.com/hyattpd/Prodigal
- http://hmmer.org/
- https://github.com/smirarab/sepp/
- https://www.r-project.org/

Please make sure that each software package listed above works INDEPENDENTLY of BUSCO before attempting to run any BUSCO assessments.
b) Permission denied (Welcome to HPC!)
1) Problems

b) Permission denied (Welcome to HPC!)

[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module li
Currently Loaded Modulefiles:
1) python/3.6.2-anaconda-tensorflow
1) Problems

b) **Permission denied** (Welcome to HPC!)

```
[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module list
Currently Loaded Modulefiles:
  1) python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ pip install geos
```

Download from https://files.pythonhosted.org/packages/49/5b/b8ac28c74c0112b9036a41b652b3de9baa59c409/6b 3.0MB/s
1) Problems

b) Permission denied (Welcome to HPC!)

```
[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module list
Currently Loaded Modulefiles:
  1) python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ pip install geos
Collecting geos
  Downloading https://files.pythonhosted.org/packages/49/5b/b8ac74c01187a36aa4165b53d9baa59c
  100% |████████████████████████████████| 409kB 3.0MB/s

File “/usr/local/packages/python/3.6.2-anaconda/lib/python3.6/os.py”, line 220, in makedirs
You are using pip version 9.0.1, however version 20.0.1 is available.
You should consider upgrading via the ‘pip install --upgrade pip’ command.
[jasonli3@smic2 ~]$
```
b) **Permission denied** (Welcome to HPC!)

- If you ask Google / ChatGPT…

```bash
$ sudo yum install ...
$ sudo apt-get install ...
$ sudo make install
```
c) Conflicted packages

- What if I need two packages with conflicted dependencies?

Tensorflow 1.13

PyTorch > 1.5
1) Problems

d) Sharing / Migrating your environment

- Huge effort & large disk quota to install
  
  • What if my colleagues want to use?
  
  • What if I want to migrate a different cluster?
1) Problems

Any of those apply to you?
1) Problems

Magic Tools to Install / Manage Software

Part 1: CONDA Virtual Environment

Part 2: ingularity Container
1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
2) Container & Singularity

a) What is a container?

- Virtual machine
  - “Virtualize” / “mimic” an entire computer on another computer
  - Virtualize both hardware and software
2) Container & Singularity

a) What is a container?

- Container:
  - A lightweight and fast virtual machine
  - Only virtualize the Operation System (meaning, does not virtualize hardware)
  - Only virtualize Linux on Linux
2) Container & Singularity

a) What is a **container**?

A collection of files and their hierarchy (e.g., system files of a "guest" OS)

```
/usr /bin /lib /etc
```

```
Host OS
/usr /bin /lib /etc /home /work /project
```

Container Engine
2) Container & Singularity

a) What is a container?

- A “chimera” system:
  - Can virtualize an entirely different OS!
  - Can contain other software packages (inc. dependencies, environment settings, etc.) installed in the guest OS
2) Container & Singularity

a) What is a container?

Container

```
/usr  /bin  /lib  /etc
```

Container Image

```
*.sif  *.simg
```
2) Container & Singularity

a) What is a container?

- Properties:
  - Self-contained
    All dependencies can be installed within the container
  - Isolated
    Whatever happens in a container stays in that container…

Container

```
/usr /bin /lib /etc
```

Host OS

```
/usr /bin /lib /etc /home /work /project
```

“Chimera”
b) How does it solve my problems?

- **Dependency issue**
  - Pack all dependencies (even OS) in container
  - Can use `apt-get` or `yum`
  - Developers now release containers!

- **Permission issue**
  - Can’t write to certain paths on HPC, but **CAN** write to them in container

- **Conflicted packages**
  - Install in different containers.

- **Share / Migrate**
  - Copy-paste a container image!
c) What is Singularity?

Technology →
c) What is **Singularity**?

↑ **Software** system that implements the technology
c) What is **Singularity**?
2) Container & Singularity

c) What is Singularity?

- Does **NOT** need root privileges
- “Container for HPC”

- **Needs** root privileges
Technology that helps with software installation →

↓ Software system that implements the technology

Container

```
/usr  /bin  /lib  /etc
```

Host OS

```
/usr  /bin  /lib  /etc  /home  /work  /project
```

“Chimera”

```
/usr  /bin  /lib  /etc  /home  /work  /project
```
1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
1) What you need

- Singularity availability

  a) On all clusters
     - LSU HPC: SMIC, Deep Bayou, SuperMike 3
     - LONI: QB2, QB3

  b) Only on computing nodes
     - Unavailable on head nodes
     - Must start a job (interactive & batch)

  c) To all users
     - No additional action required
## 1) What you need

<table>
<thead>
<tr>
<th>To …</th>
<th>What you need …</th>
</tr>
</thead>
</table>
| Run an Existing Container Image | • Access to our HPC systems  
  - An active account  
  - An active allocation |
1) What you need

- Run:
  - All users
1. **Why Container?**
   1) Problems
   2) Container & Singularity

2. **Run an Existing Container Image**
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. **Get More Container Images**
   1) What you need
   2) Where to get
   3) How to get

4. **Build Your Own Container Image**
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
2) Basic commands

- Available images

  - On all clusters: /home/admin/singularity
2) Basic commands

a) Common usage 1: Open a shell in the image

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>singularity shell &lt;container&gt;</td>
<td>Starts a shell in the image</td>
</tr>
</tbody>
</table>

Try me: /home/admin/singularity/ubuntu-training.sif
2) Basic commands

a) Common usage 1: Open a shell in the image

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>singularity shell [options] &lt;container&gt;</td>
<td>Starts a shell in the image</td>
</tr>
</tbody>
</table>

[Options]
- -B /path/to/bind
  - /home is bound by default
- --nv
  - Enable Nvidia GPU
## 2) Basic commands

### b) Common usage 2: Execute a single command in the image

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>singularity exec &lt;container&gt; &lt;command&gt;</td>
<td>Execute a command in the image</td>
</tr>
</tbody>
</table>

**Try me:** /home/admin/singularity/ubuntu-training.sif
### 2) Basic commands

#### b) Common usage 2: Execute a single command in the image

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>singularity exec [options] &lt;container&gt; &lt;command&gt;</code></td>
<td>Execute a command in the image</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-B /path/to/bind</code></td>
<td>Bind a path(s)  &lt;br&gt;  - /home is bound by default</td>
</tr>
<tr>
<td><code>--nv</code></td>
<td>Enable Nvidia GPU</td>
</tr>
</tbody>
</table>
c) Another (less) common usage: Run a prewritten script

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>singularity run [options] &lt;container&gt;</td>
<td>Run a prewritten script</td>
</tr>
<tr>
<td>[Options] -B /path/to/bind</td>
<td>Bind a path(s)</td>
</tr>
<tr>
<td></td>
<td>• /home is bound by default</td>
</tr>
<tr>
<td>--nv</td>
<td>Enable Nvidia GPU</td>
</tr>
</tbody>
</table>
### 2) Basic commands

- Quick recap

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>singularity shell [options] &lt;container&gt;</code></td>
<td>Run a prewritten script</td>
</tr>
<tr>
<td><code>singularity exec [options] &lt;container&gt; &lt;command&gt;</code></td>
<td>Execute a command in the image</td>
</tr>
<tr>
<td><code>singularity run [options] &lt;container&gt;</code></td>
<td>Run a prewritten script</td>
</tr>
</tbody>
</table>
1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
3) Run jobs with Singularity

- Job types and commands

<table>
<thead>
<tr>
<th>Job Type</th>
<th>Commands</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Interactive** | • singularity shell [options] <container>
• singularity exec [options] <container> <command> | • Debugging & testing |
| **Batch**   | • singularity exec [options] <container> <command> | • Production              |
3) Run jobs with Singularity

a) Interactive job

- Start an interactive job
- Open a shell in the image with `singularity shell`
- Do your things...
- Run your software with `singularity exec`
# 3) Run jobs with Singularity

## b) Batch job

<table>
<thead>
<tr>
<th>PBS</th>
<th>Slurm</th>
</tr>
</thead>
</table>
| #!/bin/bash  
#PBS -A <Allocation name>  
#PBS -q workq  
#PBS -l nodes=1:ppn=20  
#PBS -l walltime=24:00:00  

cd /to/work/directory  
IMG=/home/admin/singularity/ubuntu-training.sif  
singularity exec -B /work,/project $IMG  
   python myjob.py |  
#!/bin/bash  
#SBATCH -A <Allocation name>  
#SBATCH -p workq  
#SBATCH -N 1  
#SBATCH -n 64  
#SBATCH -t 24:00:00  

cd /to/work/directory  
IMG=/home/admin/singularity/ubuntu-training.sif  
singularity exec -B /work,/project $IMG  
   python myjob.py |
2. Run
   - All users

Summary

1. Why Container?
2. Run
3. Get More
4. Build your own
## Summary

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>singularity shell [options] &lt;container&gt;</code></td>
<td>Run a prewritten script</td>
</tr>
<tr>
<td><code>singularity exec [options] &lt;container&gt; &lt;command&gt;</code></td>
<td>Execute a command in the image</td>
</tr>
<tr>
<td><code>singularity run [options] &lt;container&gt;</code></td>
<td>Run a prewritten script</td>
</tr>
</tbody>
</table>
Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
3. Get More Container Images

- Available images

  - On all clusters: /home/admin/singularity

```
(base) [jasonli3@mike1 ~]$ ls /home/admin/singularity
abinit.v9.8.3.sif
alphafold-catgumag-2.2.sif
bowtie2-2.5.1.sif
busco-5.4.7.sif
clara.400-1.sif
deepmd-kit_2.0.3_cuda11.3_gpu.sif
deepmd-kit_2.2.1_cuda11.6_gpu.sif
delft3dfm_r142632_05032023.sif
delft3dfm_r142632.sif
delft3dfm_r142586.sif
jax-0.4.13-gpu-jupyter.sif
MuST-1.8.7.sif
openpose.sif
pytorch-2.0.1-gpu-jupyter.sif
rstudio-2023.03.1-446-rocky8.sif
salmon-1.10.2.sif
tensorflow-2.13.0-gpu-jupyter.sif
trinity-2.15.1.sif
ubuntu-training.sif
```
<table>
<thead>
<tr>
<th>Outlines</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Why Container?</td>
</tr>
<tr>
<td>1) Problems</td>
</tr>
<tr>
<td>2) Container &amp; Singularity</td>
</tr>
</tbody>
</table>

| 2. Run an Existing Container Image |
|   1) What you need |
|   2) Basic commands |
|   3) Running jobs with Singularity |

| 3. Get More Container Images |
|   1) What you need |
|   2) Where to get |
|   3) How to get |

| 4. Build Your Own Container Image |
|   1) What you need |
|   2) Typical workflow |
|   3) Make it easier - Recipe |
1) What you need

1. Why Container?
2. Run
3. Get More
4. Build your own

```bash
(base) [jasonli3@mike4]
```

```
ls /home/admin/singularity
```

```
total 78269032
-rw-r-xr-x 1 root singularity 4555411456 Jun 7 20:27 abinit.v9.8.3.sif
-rw-r-xr-x 1 root singularity 3167383496 Oct 3 2022 alphafold-catgumag-2.2.sif
-rw-r-xr-x 1 root singularity 118206464 Sep 11 11:00 bowtie2-2.5.1.sif
-rw-r-xr-x 1 root singularity 822177792 Aug 22 09:00 busco-5.4.7.sif
-rw-r-xr-x 1 root singularity 2155438880 Nov 9 2022 clara.400-1.sif
-rw-r-xr-x 1 root singularity 3285417984 Jun 6 13:13 deepmd-kit 2.0.3_cuda11.3_gpu.sif
-rw-r-xr-x 1 root singularity 3390902272 Jun 5 20:34 deepmd-kit 2.2.1_cuda11.6_gpu.sif
-rw-r-xr-x 1 root singularity 9305526272 May 3 12:56 delft3dfm_r142633_05032023.sif
-rw-r-xr-x 1 root singularity 11812220928 Jun 27 23:50 delft3dfm_r142632.sif
-rw-r-xr-x 1 root singularity 11594326016 Jul 26 00:05 delft3d_r142586.sif
-rw-r-xr-x 1 root singularity 5152387072 Sep 15 23:52 jax-0.4.13-gpu-jupyter.sif
-rw-r-xr-x 1 root singularity 7603736576 Jul 20 14:57 MuST-1.8.7.sif
-rw-r-xr-x 1 root singularity 5322223616 Jun 20 14:26 openpose.sif
-rw-r-xr-x 1 root singularity 4016316416 Aug 7 23:56 pytorch-2.0.1-gpu-jupyter.sif
-rw-r-xr-x 1 root singularity 911499264 Aug 25 09:48 rstudio-2023.03.1-446-rocky8.sif
-rw-r-xr-x 1 root singularity 42639360 Sep 21 12:22 salmon-1.10.2.sif
-rw-r-xr-x 1 root singularity 4079706112 Aug 8 09:32 tensorflow-2.13.0-gpu-jupyter.sif
-rw-r-xr-x 1 root singularity 2739630080 Aug 30 19:40 trinity-2.15.1.sif
-rw-r-xr-x 1 root singularity 71102464 Sep 7 11:50 ubuntu-training.sif
1) What you need

Singularity images must belong to "singularity" group to run on our clusters!
## 1) What you need

<table>
<thead>
<tr>
<th>To …</th>
<th>What you need …</th>
</tr>
</thead>
</table>
| Run an Existing Container Image | • Access to our HPC systems  
  - An active account  
  - An active allocation |
| Get More Container Images | • (Everything above)  
  • Added to **Singularity** group  
  - Needed to change **group ownership**  
  - Send email to [sys-help@loni.org](mailto:sys-help@loni.org) to request |
1) What you need

- Run:
  - All users

- Get more:
  - Added to singularity group
1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
2) Where to get

• You can get container images from a lot of places
  ➢ Not that you should!

• Concerns?
  – Reliability (some third-party or untested images may not work)
  – Security risk (some untrustworthy publishers may pack something you don’t know about)

• Solution
  – Always get from a source that you can trust
## 2) Where to get

- **Tier 1: Developer release (official release)**
  - On software’s official website, look for “Docker” / “Singularity” / “Container” / etc.
  - E.g., Tensorflow, Trinity, Salmon

- **Tier 2: Trustworthy third party**

<table>
<thead>
<tr>
<th>Name</th>
<th>Notes</th>
</tr>
</thead>
</table>
<pre><code>              | • For biology                                                          |
</code></pre>
| Nvidia NGC       | • [https://catalog.ngc.nvidia.com/containers](https://catalog.ngc.nvidia.com/containers)  
                  | • For Nvidia GPU                                                       |
| Bitnami          | • [https://bitnami.com/stacks/containers](https://bitnami.com/stacks/containers)  
                  | • By VmWare                                                           |
| **Docker Hub**   | • [https://hub.docker.com/](https://hub.docker.com/) & [https://quay.io/](https://quay.io/)  
                  | • Don’t just trust everything you see there!                          |
| **Quay.io**      | • Look for Docker Official Image or Verified Publisher                 |
|                  | • Avoid third-party publishers that you don’t know                    |
1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
3) How to get

• Steps:
  
  a) Step 1: Pull the image
  
  b) Step 2: Change group ownership
### 3) How to get

#### a) Step 1: Pull the image

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>singularity <strong>pull</strong></td>
<td>Pull an image from source</td>
</tr>
<tr>
<td><strong>&lt;source&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><code>docker://container[:tag]</code></td>
<td>Pull a Docker container and convert to Singularity</td>
</tr>
<tr>
<td>• (Compare to Docker command)</td>
<td><em>Many software’s official container release is in Docker form (may or may not on Docker Hub)</em></td>
</tr>
<tr>
<td><code>docker pull container[:tag]</code></td>
<td></td>
</tr>
<tr>
<td><strong><a href="http://www.myexample.com/container_image.sif">http://www.myexample.com/container_image.sif</a></strong></td>
<td>Download an image file from a web source</td>
</tr>
</tbody>
</table>
### 3) How to get

#### a) Step 1: Pull the image

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>singularity <code>build</code></td>
<td>Build an image from source (Advanced)</td>
</tr>
<tr>
<td><code>&lt;target&gt; &lt;source&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>docker://container[:tag]</code></td>
<td>Build from a Docker container</td>
</tr>
<tr>
<td><code>container_image.sif</code></td>
<td>Build from a local image file</td>
</tr>
<tr>
<td><code>container_sandbox/</code></td>
<td>Build from a local <strong>sandbox</strong> (A directory form of a container)</td>
</tr>
<tr>
<td><code>container_recipe.def</code></td>
<td>Build from a <strong>recipe</strong> (an instruction script of how to build an image)</td>
</tr>
</tbody>
</table>
3) How to get

a) Step 1: Pull the image

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>singularity pull [options] [target] &lt;source&gt;</td>
<td>Simple pull</td>
</tr>
<tr>
<td>singularity build [options] &lt;target&gt; &lt;source&gt;</td>
<td>Advanced build command</td>
</tr>
</tbody>
</table>
3) How to get

b) Step 2: Change group ownership

– What if you do not?

```
FATAL: singularity image is not owned by required group(s)
```

– To solve it, run this:

```
$ chgrp singularity <container>
```

* You must be added to singularity group to finish this step
Summary

1. Why Container?
2. Run
   - All users
3. Get More
   - Added to singularity group
4. Build your own
• Steps:

a) Step 1: Pull the image

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>singularity pull [options] [target] &lt;source&gt;</td>
<td>Simple pull</td>
</tr>
<tr>
<td>singularity build [options] &lt;target&gt; &lt;source&gt;</td>
<td>Advanced build command</td>
</tr>
</tbody>
</table>

b) Step 2: Change group ownership
1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
4. Build Your Own Container Image

• Scenarios:

  – I did not find any container release. Need to DIY.

  – Installation can be easily done using `sudo apt` or `sudo yum` if I have the permission.

  – I found a container, but need to make changes (e.g., adding something else).
4. Build Your Own Container Image

- Idea

```
sudo
yum
apt-get
```

HPC systems
4. Build Your Own Container Image

- Idea

Software

HPC systems

Build & transfer container images

A local machine

sudo

yum

apt-get

...
4. Build Your Own Container Image

- Idea

- Build & transfer container images

- A local machine

- HPC systems

- Software

- Container
  - /usr
  - /bin
  - /lib
  - /etc

- Host OS
  - /usr
  - /bin
  - /lib
  - /etc
  - /home
  - /work
  - /project

- "Chimera"
  - /usr
  - /bin
  - /lib
  - /etc
  - /home
  - /work
  - /project

- sudo
- yum
- apt-get

- Build your own container image
1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
1) What you need

<table>
<thead>
<tr>
<th>To …</th>
<th>What you need …</th>
</tr>
</thead>
</table>
| Run an Existing Container Image | • Access to our HPC systems  
  - An active account  
  - An active allocation |
| Get More Container Images | • (Everything above)  
  • Added to *Singularity* group  
    - Needed to change **group ownership**  
    - Send email to sys-help@loni.org to request |
| Build Your Own Container Image | • (Everything above)  
  • A local Linux machine, where:  
    - You have **root permission** |
1) What you need

- **Run:**
  - All users

- **Get more:**
  - Added to *singularity* group

- **Build your own:**
  - A Linux machine where:
    - You have *root permission*
1) What you need

- Install Singularity

• Joined Linux Foundation
• Easier installation

Compatible

- Community supported
• Installed on our clusters

Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
2) Typical workflow

On HPC

Choose a base image

Build a sandbox

Make modifications

Upload image to HPC

Your image is ready to go

On local machine

Build an image
## 2) Typical workflow

a) Choose a base image

<table>
<thead>
<tr>
<th>Common choices</th>
<th>Typical scenarios</th>
</tr>
</thead>
</table>
| A minimum, “mint” OS (e.g., Ubuntu, Rocky, Debian, …) | • You cannot find an existing image with the software you need, and need to install from the scratch.  
• You need to build a minimum size image |
| A container with software already installed (e.g., TensorFlow, PyTorch, …) | • You need to modify an existing working image (e.g., add a Python module to TensorFlow image) |
2) Typical workflow

b) Build a sandbox

– What’s a sandbox?

• A directory form (unlike a single image file) of a container

• Allows modification
### 2) Typical workflow

#### b) Build a sandbox

```bash
$ singularity build [options] <target> <source>
```

<table>
<thead>
<tr>
<th><code>&lt;source&gt;</code></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>docker://container[:tag]</code></td>
<td>Build from a Docker container</td>
</tr>
<tr>
<td><code>container_image.sif</code></td>
<td>Build from a local image file</td>
</tr>
<tr>
<td><code>container_sandbox/</code></td>
<td>Build from a local sandbox (A directory form of a container)</td>
</tr>
<tr>
<td><code>container_recipe.def</code></td>
<td>Build from a recipe (an instruction script of how to build an image)</td>
</tr>
</tbody>
</table>
### 2) Typical workflow

b) **Build a sandbox**

```bash
$ singularity build [--sandbox] [options] <target> <source>
```

<table>
<thead>
<tr>
<th><code>&lt;source&gt;</code></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>docker://container[:tag]</code></td>
<td>Build from a Docker container</td>
</tr>
<tr>
<td><code>container_image.sif</code></td>
<td>Build from a local image file</td>
</tr>
<tr>
<td><code>container_sandbox/</code></td>
<td>Build from a local sandbox (a directory form of a container)</td>
</tr>
<tr>
<td><code>container_recipe.def</code></td>
<td>Build from a recipe (an instruction script of how to build an image)</td>
</tr>
</tbody>
</table>
2) Typical workflow

c) Make modifications

$ singularity shell [options] <container>
2) Typical workflow

c) Make modifications

$ singularity shell --writable [options] <container>

i. Allows **writing** to the sandbox
   - Without it, just like running a regular container image
2) Typical workflow

c) Make modifications

ii. Run the container as **root**
   - Grants root privilege in container
   - Needed in most cases
   - Technically not required, but cannot run things like `sudo apt` or `sudo yum` without it

i. Allows **writing** to the sandbox
   - Without it, just like running a regular container image

```
$ sudo singularity shell --writable [options] <container>
```
2) Typical workflow

c) Make modifications

```bash
$ sudo singularity shell --writable [options] <container>
Singularity>
Singularity> apt update
Singularity> apt install ...
```
2) Typical workflow

d) Build an image from sandbox

```bash
$ singularity build [options] <target> <source>
```

<table>
<thead>
<tr>
<th><code>&lt;source&gt;</code></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>docker://container[:tag]</code></td>
<td>Build from a Docker container</td>
</tr>
<tr>
<td><code>container_image.sif</code></td>
<td>Build from a local image file</td>
</tr>
<tr>
<td><code>container_sandbox/</code></td>
<td>Build from a local sandbox (A directory form of a container)</td>
</tr>
<tr>
<td><code>container_recipe.def</code></td>
<td>Build from a recipe (an instruction script of how to build an image)</td>
</tr>
</tbody>
</table>
2) Typical workflow

d) Build an image from sandbox

```
$ sudo singularity build [options] <target> <source>
```

Modify with “sudo”? Build with “sudo”!
## 2) Typical workflow

- Quick recap

<table>
<thead>
<tr>
<th>To ...</th>
<th>You need to ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build a sandbox</td>
<td><code>$ singularity build --sandbox ...</code></td>
</tr>
<tr>
<td>Modify a sandbox</td>
<td><code>$ sudo singularity shell --writable ...</code></td>
</tr>
<tr>
<td>Build an image from sandbox</td>
<td><code>$ sudo singularity build ...</code></td>
</tr>
</tbody>
</table>
2) Typical workflow

e) Upload image to HPC and run

Now! The moment of truth!
2) Typical workflow

On HPC

On local machine

1. Why Container?
2. Run
3. Get More
4. Build your own
2) Typical workflow

On HPC

- Upload image to HPC
- Your image is ready to go

On local machine

1. Choose a base image
2. Build a sandbox
3. Make modifications
4. Build an image

Your image is ready to go

1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
3) Make it easier - Recipe

- **Why?**

On HPC

- Upload image to HPC
- Your image is ready to go

On local machine

1. Choose a base image
2. Build a sandbox
3. Make modifications
4. Build an image

Your image is ready to go
### 3) Make it easier - Recipe

**Why?**

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Flexibility</td>
<td>• Repeatability</td>
</tr>
<tr>
<td></td>
<td>• Minimizing image size</td>
</tr>
</tbody>
</table>

**Solution:**

- **Recipe**: A text file containing instructions to build a container
3) Make it easier - Recipe

- **Why?**

On HPC

- Upload image to HPC
- Your image is ready to go

On local machine

1. Choose a base image
2. Build a sandbox
3. Make modifications
4. Build an image

Your image is ready to go
3) Make it easier - Recipe

- Why?

On HPC

- Write / Obtain a recipe

On local machine

- Build an image from recipe

- Upload image to HPC

Your image is ready to go
3) Make it easier - Recipe

a) What does a recipe look like?

```ruby
def
  BootStrap: docker
  From: ubuntu:latest

  %labels
  Author       Jason Li
  Description  A container with Ruby installed

  %post
  apt update
  apt install -y ruby

  %environment
  export MYENV="Some environmental variable"

  %runscript
  ruby -e "puts 'Hello from container!'"
```

3) Make it easier - Recipe

a) What does a recipe look like?

```
ruby.def

Header
- Base image info (how, where, what to pull)

BootStrap: docker
From: ubuntu:latest

%labels
Author       Jason Li
Description  A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```
3) Make it easier - Recipe

a) What does a recipe look like?

```
ruby.def

BootStrap: docker
From: ubuntu:latest

%labels
Author       Jason Li
Description  A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```
3) Make it easier - Recipe

a) What does a recipe look like?

```
ruby.def
Bootstrap: docker
From: ubuntu:latest

%labels
Author       Jason Li
Description  A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```

**Post**
- Commands to execute after the base image is pulled
3) Make it easier - Recipe

a) What does a recipe look like?

```
ruby.def

BootStrap: docker
From: ubuntu:latest

%labels
Author       Jason Li
Description  A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```

Environment
- Define environmental variables every time the container is executed
3) Make it easier - Recipe

a) What does a recipe look like?

ruby.def

BootStrap: docker
From: ubuntu:latest

%labels
Author       Jason Li
Description  A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"

Runscript - Commands to be run with singularity run
3) Make it easier - Recipe

a) What does a recipe look like?

ruby.def

```
BootStrap: docker
From: ubuntu:latest

%labels
Author       Jason Li
Description  A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```
3) Make it easier - Recipe

a) What does a recipe look like?

```ruby
def
    # Bootstrap: docker
    BootStrap: docker
    From: ubuntu:latest

    # Labels
    %labels
    Author       Jason Li
    Description  A container with Ruby installed

    # Post
    %post
    apt update
    apt install -y ruby

    # Environment
    %environment
    export MYENV="Some environmental variable"

    # Runscript
    %runscript
    ruby -e "puts 'Hello from container!""
```

3) Make it easier - Recipe

b) Build the recipe

$ singularity build [options] <target> <source>

docker://container[:tag] | Build from a Docker container
container_image.sif | Build from a local image file
container_sandbox/ | Build from a local sandbox (A directory form of a container)
container_recipe.def | Build from a recipe (an instruction script of how to build an image)
Summary

1. Why Container?
2. Run
- All users
3. Get More
- Added to singularity group
4. Build your own
- A Linux machine where:
  - You have root permission
Summary

1. Why Container?
2. Run
3. Get More
4. Build your own

On HPC
- Upload image to HPC
- Your image is ready to go

On local machine
- Choose a base image
- Build a sandbox
- Make modifications
- Build an image

Your image is ready to go
Conclusion
Conclusion

1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe
Take home message

- **Run:**
  - All users

- **Get more:**
  - Added to singularity group

- **Build your own:**
  - A Linux machine where:
    - You have root permission
To conclude our mini series...
Conda vs Singularity

- **Virtual Environment v.S. Container?**

```
/path/to/envs
  |____env1
  |      • Python 3.7
  |            • numpy 1.13
  |            • scipy 0.19
  |____env2
  |      • Python 3.9
  |            • pandas 1.5
  |____env3
  |      • Perl 5.8
  |            • Bioperl 1.6
  |____env4
  |      • R 4.2.3
  |            • Seurat 4.3
```

Container

```
/usr  /bin  /lib  /etc
```

+ Host OS

```
/usr  /bin  /lib  /etc  /home  /work  /project
```

“Chimera”

```
/usr  /bin  /lib  /etc  /home  /work  /project
```
<table>
<thead>
<tr>
<th></th>
<th>Conda / Virtual Environments</th>
<th>Singularity / Containers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>All users</td>
<td>All users, but may need additional things</td>
</tr>
<tr>
<td>Self-contained</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Isolated</td>
<td>Yes (but still accessible from outside)</td>
<td>Perfect (completely isolated from outside)</td>
</tr>
<tr>
<td>Editability</td>
<td>Yes</td>
<td>No (Must create a new image)</td>
</tr>
<tr>
<td>Disk usage</td>
<td>Large</td>
<td>Smaller</td>
</tr>
<tr>
<td>Portability</td>
<td>Possible (with .yml recipe)</td>
<td>Great (just copy-paste one file)</td>
</tr>
<tr>
<td>Security</td>
<td>Fair</td>
<td>Good</td>
</tr>
<tr>
<td>Ease of use</td>
<td>Good</td>
<td>May require a little more understanding</td>
</tr>
</tbody>
</table>
## Conda vs Singularity

<table>
<thead>
<tr>
<th>Good for</th>
<th>Conda / Virtual Environments</th>
<th>Singularity / Containers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Less hassle to create and install software from scratch</td>
<td>• Less hassle if the developer releases a working container</td>
</tr>
<tr>
<td></td>
<td>• If you need to frequently make modifications</td>
<td>• If you don’t need to make changes after it is created</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Portability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reduce disk usage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Your system admins yelled at you about security risk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Portability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reduce disk usage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Your system admins yelled at you about security risk</td>
</tr>
</tbody>
</table>
Contact us

- Contact user services
  - Email Help Ticket: sys-help@loni.org
  - Telephone Help Desk: +1 (225) 578-0900
“Commercial” time!

- Are you tired of wring the long, tedious singularity commands?

```bash
$ singularity exec --nv -B /work,/project,/usr/local/package \
    /home/admin/singularity/ubuntu-training.sif \
    python helloworld.py
```
“Commercial” time!

- Try SIMPLE-MOD!
  - [https://github.com/lsuhpchelp/SIMPLE-MOD](https://github.com/lsuhpchelp/SIMPLE-MOD)
  - A GUI tool to create module key from container-based software.
  - Using the software in containers is as easy as:

```bash
$ module load busco
$ busco --version
BUSCO 5.6.1
```