

An intro to Singularity

- These slides and examples

https://github.com/martinjvickers/install_scripts

- Official docs for our version

<http://singularity.lbl.gov/archive/docs/v2-4/>

- CiS documentation

<https://docs.cis.nbi.ac.uk/display/CIS/Installing+software+on+the+cluster+using+Singularity>

Presented by Martin and Sreya

Not exactly an interactive tutorial

- However if you want to try the examples;
 - Best to have two terminals on SLURM
 - ssh software7
 - To create singularity containers
 - interactive
 - To interact with them correctly
 - On software7
 - source git-1.8.1.2
 - git clone https://github.com/martinjvickers/install_scripts
 - cd install_scripts/singularity_intro

Contents

- A little background
- Working with Singularity
 - Creating a basic recipe
 - Building an image from a recipe
 - CentOS and Ubuntu package manager example
 - Sandbox
 - Environment variable
 - Singularity Hub pre-built example

Why Singularity?

- Singularity aims:
 - Make applications and pipelines portable
 - Cross platform
 - Software reproducibility
 - Support legacy OS and apps
 - Reduce burden of installing software on SysAdmins and allow users more flexibility

Interactive Development

```
sudo singularity build --sandbox tmpdir/ Singularity
```

```
sudo singularity build --writable container.img Singularity
```

BUILD ENVIRONMENT

Build from Recipe

```
sudo singularity build container.img Singularity
```

Build from Singularity

```
sudo singularity build container.img shub://vsoch/hello-world
```

Build from Docker

```
sudo singularity build container.img docker://ubuntu
```

Container Execution

```
singularity run container.img  
singularity shell container.img  
singularity exec container.img ...
```

Reproducible Sharing

```
singularity pull shub://...  
singularity pull docker://... *
```

PRODUCTION ENVIRONMENT

* Docker construction from layers not guaranteed to replicate between pulls

Create a recipe file

- Defines the container you're creating
 - e.g. OS and version
- Install software and dependencies
- Provide a record of how the software was installed
 - can be version controlled e.g. using git
- Can be shared/published

Samtools example

- Installing samtools
- A CentOS and a Ubuntu example
- Both built using a recipe file
- Both installed using the OS package manager

Quick note on directory structure

- `./x86_64`
- `./x86_64/samtools-0.1.19-centos.img`
- `./x86_64/README.md`
- `./x86_64/bin`
- `./x86_64/bin/samtools`
- `./x86_64/bin/singularity.exec`
- `./src`
- `./src/samtools-0.1.19-centos.def`

Create a recipe

BootStrap: yum

OSVersion: 7

MirrorURL: [http://yum-repos.hpccluster/centos/7/os/\\$basearch/](http://yum-repos.hpccluster/centos/7/os/$basearch/)

Include: yum

UpdateURL: [http://yum-repos.hpccluster/centos/7/updates/\\$basearch/](http://yum-repos.hpccluster/centos/7/updates/$basearch/)

%post

```
mkdir /jic && mkdir /nbi  
mkdir /tgac && mkdir /tsl
```

Installing software in the recipe

```
%post
```

```
    mkdir /jic
```

```
    mkdir /nbi
```

```
    mkdir /tgac
```

```
    mkdir /tsl
```

```
    yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

```
    yum update && yum -y install samtools
```

Build a writeable image

Build the container from the recipe which installs the OS and anything you're specified in `%post`

```
sudo singularity build -w container.img recipe.def
```

This creates an image which is writeable.

Without the `-w` flag, a read only image is created which is perfect for deploying and sharing

Running samtools using exec

singularity exec container.img <command>

```
[mvickers@ ~]$ ls
bin  README.md  samtools-0.1.19centos.img
[mvickers@ ~]$ samtools
-bash: samtools: command not found
[mvickers@ ~]$ singularity exec samtools-
0.1.19centos.img samtools
```

Program: samtools (Tools for alignments in the SAM format)

Version: 0.1.19-44428cd

Usage: samtools <command> [options]

Same example using Ubuntu

BootStrap: debootstrap

OSVersion: trusty

MirrorURL: <http://us.archive.ubuntu.com/ubuntu/>

%post

```
    mkdir /jic && mkdir /nbi && mkdir /tgac && mkdir /tsl  
    apt-get install --force-yes -y software-properties-  
common  
    add-apt-repository "deb  
http://archive.ubuntu.com/ubuntu $(lsb_release -sc)  
universe"  
    apt-get update --force-yes -y  
    apt-get install --force-yes -y samtools
```

Same build process as before

```
[mvickers@NBI-HPC ~]$ sudo singularity build -w samtools-0.1.19ubuntu.img ../src/samtools-0.1.19-ubuntu.def
```

```
[mvickers@NBI-HPC ~]$ singularity exec samtools-0.1.19ubuntu.img samtools
```

```
Program: samtools (Tools for alignments in the SAM format)  
Version: 0.1.19-96b5f2294a
```

Why a read only image?

- What is the purpose of a read only image
 - Simply: compressed and easier to share

129M May 10 08:45 samtools-0.1.19ubuntu_readonly.img

404M May 10 08:41 samtools-0.1.19ubuntu.img

Expanding a image

- When building a writeable image, you get around ~20% more than the space required to build it.
- Can you expand an image? Short answer: Not easily

```
[mvickers@ ~]$ sudo singularity shell -w samtools-0.1.19ubuntu.img
```

Singularity: Invoking an interactive shell within container...

```
Singularity samtools-0.1.19ubuntu.img:/> df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
singularity	383M	310M	53M	86%	/

The answer, use a sandbox

- Creates a directory structure of the container so that you can develop interactively

```
[mvickers@NBI-HPC sandbox]$ sudo singularity build --sandbox centos_sandbox centos_sandbox.def
```

```
[mvickers@NBI-HPC sandbox]$ ls centos_sandbox
```

bin	environment	jic	media	opt	run	srv	tmp	var
boot	etc	lib	mnt	proc	sbin	sys	tsl	
dev	home	lib64	nbi	root	singularity	tgac	usr	

Interacting with the sandbox

- Same as usual.
- Must use `-w` to write to shell otherwise changes will not persist

```
[mvickers@NBI-HPC sandbox]$ sudo singularity shell -w  
centos_sandbox
```

```
Singularity: Invoking an interactive shell within  
container...
```

```
Singularity centos_sandbox:~>
```

Making an image from a sandbox

- Once you've installed your software, you can make an image the same way as before

```
sudo singularity build production_app.img centos_sandbox
```

- Just in case you wondered why bother:

```
[mvickers@NBI-HPC sandbox]$ du -h --max-depth=1
```

```
2.0G ./centos_sandbox
```

```
[mvickers@NBI-HPC sandbox]$ ls -lath production.img
```

```
-rwxrwx--x 1 root root 263M May 10 00:20 production.img
```

Environment variables

- RNAstructure example (courtesy of Hugh)
- Setting variable in the recipe

```
%environment
```

```
DATAPATH=/opt/software/RNAstructure/data_tables
```

- Setting a variable at runtime
 - using SINGULARITYENV_*

```
SINGULARITYENV_DATAPATH=/opt/software/RNAstructure/data_
tables singularity exec <command>
```

Perfect, a more complex example

- Installing RNA-seq tools in bioconda
 - CentOS
 - Install OS dependencies
 - Install miniconda
 - Install bioinformatics channels
 - Install some standard bioinformatics tools
 - ***Append miniconda binaries to the PATH***

%post

```
mkdir /jic && mkdir /nbi  
mkdir /tgac && mkdir /tsl  
yum -y update  
yum -y install yum-utils which wget  
yum -y groupinstall development
```

```
wget https://repo.continuum.io/miniconda/Miniconda3-  
latest-Linux-x86_64.sh -O miniconda.sh  
bash miniconda.sh -b -p /usr/local/miniconda
```

```
##### Required here as %environment is added after build  
export PATH=$PATH:/usr/local/miniconda/bin  
conda config --add channels conda-forge  
conda config --add channels bioconda  
conda install -y trimmomatic tophat cufflinks transdecoder  
trinity
```

%environment

```
export PATH=$PATH:/usr/local/miniconda/bin
```

Build as usual






- We can see in the environment variables that the PATH is set correctly, and we can run a command

```
[mvickers@NBI-HPC ~]$ singularity exec bioconda_centos.img env |  
grep miniconda  
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:/u  
sr/local/miniconda/bin
```

```
[mvickers@NBI-HPC ~]$ singularity exec bioconda_centos.img bowtie2  
No index, query, or output file specified!  
Bowtie 2 version 2.3.4.1 by Ben Langmead (langmea@cs.jhu.edu,  
www.cs.jhu.edu/~langmea)
```

Singularity Hub (shub://)

<https://singularity-hub.org>.

Name	Buids	Description	Stars ★	Downloads	Last Modified
researchapps/quantum_state_diffusion	109	Solving quantum state diffusion numerically.	1	15	2017-10-19 
vsoch/singularity-hello-world	102		1	6042	2018-03-02 
vsoch/pe-predictive	100	run pe-finder to predict pumonary embolism from radiology reports	0	14	2017-10-17 
khanlab/prepdwi	15	BIDS-app for pre-processing DWI (denoise, unring, top-up, eddy, bedpost..)	0	22	2018-05-08 
shahzebsiddiqui/eb-singularity	12	Stuff related to integrating EasyBuild & Singularity	0	137	2018-02-21 

Downloading prebuild shub images using pull

```
[mvickers@NBI-HPC ~]$ singularity pull shub://vsoch/hello-  
world  
Progress |=====| 100.0%  
Done. Container is at: vsoch-hello-world-master-latest.simg
```

```
[mvickers@NBI-HPC ~]$ ls  
vsoch-hello-world-master-latest.simg
```

```
[mvickers@NBI-HPC ~]$ singularity exec vsoch-hello-world-  
master-latest.simg ls  
bin      environment  lib      mnt rawr.sh  sbin      sys  var  
boot     etc          lib64    opt  root      singularity tmp  
dev      home        media    proc run    srv        usr
```

Extra points

- `singularity image.expand image.img` does not work. It only works on empty image, before an OS has been installed
- When running `singularity exec` on the software node, you can't access your file system correctly

`WARNING: Could not chdir to home: /usr/users/JIC_c1/mvickers`

`open: No such file or directory`

`[main_samview] fail to open "70SC_1_val_1.bam" for reading.`

Extra points

- If publishing only a recipe that installs from github, use specific releases e.g.

```
git clone <repository> --branch v0.1.19
```

- Rather than simply `git clone <repository>`
 - As the software in the repository develops, so do the dependencies
- Probably best to use CentOS recipes where possible (at least at the moment) for NBI specific things

Development of Singularity

- It's moving very fast
 - We are using 2.4.2 as of yesterday, latest is 2.5.1
 - Prior to that was 2.2 which is quite different
 - Significant changes and compatibility
- The move from 2.2 to 2.4.2 yesterday on the software node changed a lot of things
 - including all of these slides and examples

What we've not manage to cover

- Docker
 - `singularity pull docker://ubuntu`
 - Quite like a little buggy at times as still in active development
- How to install on the `/nbi/testing` and sourcing the file
 - Getting the right directory structure
 - Following the CiS instructions
 - <https://docs.cis.nbi.ac.uk/display/CIS/Installing+software+on+the+cluster+using+Singularity>
- Publishing to Singularity Hub
- Converting images (e.g. `image->sandbox->image`)

Thank you

- Tjelvar Olsson
- Matthew Hartley
- Hugh Wollfenden
- Michael Burrell
- Sam Gallop
- CiS Team