Building an Easybuild Container Library in Sylabs Cloud

Shahzeb Siddiqui (Shahzeb.Siddiqui@3ds.com)

Dassault Systemes

Jan 29th 2020

5th Easybuild User Meeting

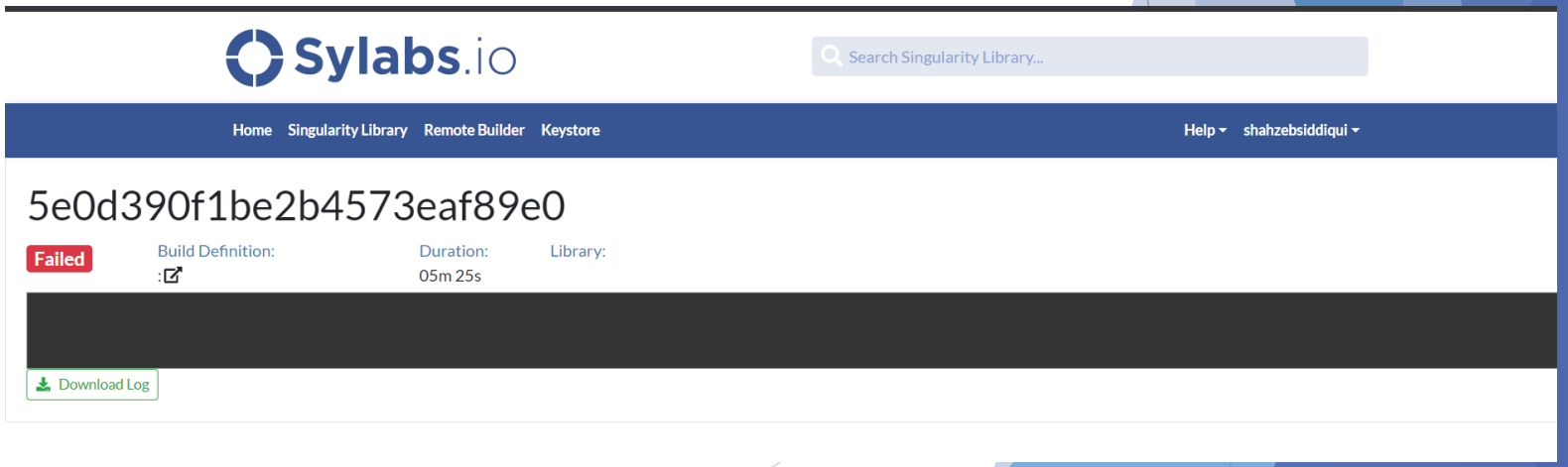
Background

- Easybuild Singularity support was first introduced in v3.6.0 and subsequent release added support for Docker containers
- Easybuild Containers is an experimental feature and that requires community testing and feedback
- Today talk will cover the following
  - My experience building easybuild containers in Sylabs Cloud
  - Cover some of the bugs in the easybuild-framework
  - Building Easybuild containers with different bootstrap agents
  - Present Singularity template in easybuild-framework and ways to improve template
  - Discuss my Easybuild Container Library in Sylabs Cloud
  - Demo & QA

Known Issues

- Bug Fix #3135: Need to upgrade Pip and install wheel package in singularity def file. Issue is resolved in eb version 4.1.1
- Can't switch to easybuild user inside container see #3171
- Missing dependencies when using other bootstrap agents (Docker, Library, etc…)
- 60min build timeout in Sylabs Cloud.
- In some odd cases, Sylabs does not stream logs even after build failure.

```
== processing EasyBuild easyconfig /usr/easybuild/easyconfigs/g/GCCcore/GCCcore-6.4.0.eb
== building and installing GCCcore/6.4.0...
== fetching files...
== creating build dir, resetting environment...
== unpacking...
== patching...
== preparing...
== configuring...
== building...
5e0e105f1be2b4573eaf89e3 build exceeded max build time
```



Easybuild Singularity Container Workflow

```
#1. Generate Singularity Definition File from Easybuild
eb binutils-2.28.eb -C --experimental --container-config bootstrap=yum,osversion=7

#2. Create Access Token from Sylabs and login
singularity remote login

#3. build container on Sylabs builder
singularity build --remote binutils-2.28.sif Singularity.binutils-2.28

#4a. Generate Key if you dont have one. If you have key then sign the container
singularity key newpair

#4b. List your Key pair
singularity key list

#4c. In Step 4a. if you select N in Would you like to push it to the keystore? [Y,n]
singularity key push <KEY>

#5. Sign & Verify your container
singularity sign binutils-2.28.sif
singularity verify binutils-2.28.sif

#6. Push container to your library
singularity push binutils-2.28.sif library://shahzebmsiddiqui/easybuild/binutils:2.28
```

Choosing Different Bootstrap Agent

- Singularity has several bootstrap agent (base image) to start build containers
- Easybuild supports several bootstrap agents that can be tweaked with --container-config option
- Bootstrap Agents:
- Yum: --container-config bootstrap=yum,osversion=7
- Library: --container-config bootstrap=library,from=library/default/centos:7
- Docker: --container-config bootstrap=docker,from=centos:7

```
Bootstrap: docker
From: centos:7
```

```
Bootstrap: library
From: library/default/centos:7
```

```
Bootstrap: yum
OSVersion: 7
MirrorURL: http://mirror.centos.org/centos-%{OSVERSION}/%{OSVERSION}/os/x86_64/
Include: yum
```

Testing with Centos Bootstrap from SyLabs

- Recipe Build: eb M4-1.4.18.eb -C --experimental --container-config bootstrap=library,from=library/default/centos:7
- Issue seems quite trivial but it is not practical from easybuild-framework to correctly generate the recipe file
- Letting user pick any base image from a library is a problem!

```
(ssi29) ssi29@ag-mxg-hulk090> singularity build --remote m4-1.4.18.sif /mxg-hpc/users/ssi29/easybuild/containers/Singularity.M4-1.4.18
INFO:    Remote "default" added.
INFO:    Authenticating with remote: default
INFO:    API Key Verified!
INFO:    Remote "default" now in use.
INFO:    Starting build...
INFO:    Downloading library image
INFO:    Running post scriptlet
+ pip install -U pip
/.build-script-post: line 4: pip: command not found
FATAL:   failed to execute %post proc: exit status 127
FATAL:   While performing build: while running engine: while running /usr/local/libexec/singularity/bin/starter: exit status 255
FATAL:   While performing build: build image size <= 0
```

```
(ssi29) ssi29@ag-mxg-hulk090> singularity exec library://library/default/centos:7  pip --version
/.singularity.d/actions/exec: line 9: exec: pip: not found
```

Testing with Docker Centos Bootstrap

- **Recipe Build:** eb Anaconda3-5.3.0.eb -C --experimental --container-config bootstrap=docker,from=centos:7
- Container Build: singularity build --remote anaconda3-5.3.0.sif Singularity.Anaconda3-5.3.0

```
Storing signatures
2020/01/23 02:51:01  info unpack layer: sha256:ab5ef0e
INFO:    Running post scriptlet
+ pip install -U pip
/.build-script-post: line 4: pip: command not found
FATAL:   failed to execute %post proc: exit status 127
FATAL:   While performing build: while running engine:
FATAL:   While performing build: build image size <= 0
```

Trying Again

- Let's try installing pip in the %post section
- Build: eb Anaconda3-5.3.0.eb -C --experimental --container-config bootstrap=docker,from=centos:7,post_commands="curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py && python get-pip.py" --force
- Turns out post_commands place the command string after pip which doesn't help.
- Another issue is eb doesn't create **easybuild** user inside container. See #3172

```
%post


# install EasyBuild using pip
pip install -U pip
pip install wheel
pip install -U setuptools
pip install 'vsc-install<0.11.4' 'vsc-base<2.9.0'
pip install easybuild

curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py && python get-pip.py
```

Trying Again

- Adding the easybuild user manually and create directory paths where easybuild will install. This is auto generated when using **yum bootstrap**

```
# create 'easybuild' user (if missing)
id easybuild || useradd easybuild

# create /app software installation prefix + /scratch sandbox directory
if [ ! -d /app ]; then mkdir -p /app; chown easybuild:easybuild -R /app; fi
if [ ! -d /scratch ]; then mkdir -p /scratch; chown easybuild:easybuild -R /scratch; fi
```

- Turns out the build fails due to package dependencies. For complete summary see issue #3172

```
# use EasyBuild to install specified software
eb Anaconda3-5.3.0.eb --robot
+ eb Anaconda3-5.3.0.eb --robot
/bin/eb: line 77: which: command not found
ERROR: You seem to be running EasyBuild with root privileges which is not wise, so let's end this here.
== temporary log file in case of crash /scratch/tmp/eb-6A5vDL/easybuild-4_ubs8.log
FATAL:   failed to execute %post proc: exit status 1
FATAL:   While performing build: while running engine: while running /usr/local/libexec/singularity/bin/
FATAL:   While performing build: build image size <= 0
```

Lessons Learned

- Using different bootstrap agents breaks container builds due to difference in base images such as package deps for easybuild.
- Easybuild generates Singularity Def file compatible with Centos (**yum**).
- Proposal:
  - Standardize on one bootstrap that means remove all bootstrap support from easybuild-framework (**--container-config**)
  - Build a base image that installs all easybuild dependencies with exception to easybuild (**pip install easybuild**)
  - Remove feature that allows user to specify their own template file **--container-template-recipe**. Let users edit the Definition file after creation as required

Default Singularity Template

```
SINGULARITY_TEMPLATE = """
Bootstrap: %(bootstrap)s
%(bootstrap_config)s

%%post
%(install_os_deps)s

%(install_eb)s

%(post_commands)s
```

```
Bootstrap: yum
OSVersion: 7
MirrorURL: http://mirror.centos.org/centos-%{OSVERSION}/%{OSVERSION}/os/x86_64/
Include: yum
```

```
%post
yum install --quiet --assumeyes epel-release
yum install --quiet --assumeyes python setuptools Lmod
yum install --quiet --assumeyes python-pip
yum install --quiet --assumeyes bzip2 gzip tar zip unzip xz
yum install --quiet --assumeyes curl wget
yum install --quiet --assumeyes patch make
yum install --quiet --assumeyes file git which
yum install --quiet --assumeyes gcc-c++
yum install --quiet --assumeyes perl-Data-Dumper
yum install --quiet --assumeyes perl-Thread-Queue
yum --skip-broken --quiet --assumeyes install libibverbs-dev libibverbs-devel rdma-core-devel
yum --skip-broken --quiet --assumeyes install openssl-devel libssl-dev libopenssl-devel
```

Required for OpenMPI

```
# install EasyBuild using pip
pip install -U setuptools
pip install 'vsc-install<0.11.4' 'vsc-base<2.9.0'
pip install easybuild
```

Easybuild user must be created in container.
See issue #3172 when passing post_commands into --container-config

```
        # if no custom value is specified for 'post_commands' keyword,
        # make sure 'easybuild' user exists and that installation prefix + scratch dir are in place
        if 'post_commands' not in template_data:
            template_data['post_commands'] = '\n'.join([
                "# create 'easybuild' user (if missing)",
                "id easybuild || useradd easybuild",
                '',
                "# create /app software installation prefix + /scratch sandbox directory",
                "if [ ! -d /app ]; then mkdir -p /app; chown easybuild:easybuild -R /app; fi",
                "if [ ! -d /scratch ]; then mkdir -p /scratch; chown easybuild:easybuild -R /scratch; fi",
            ])
```

```
# create 'easybuild' user (if missing)
id easybuild || useradd easybuild

# create /app software installation prefix + /scratch sandbox directory
if [ ! -d /app ]; then mkdir -p /app; chown easybuild:easybuild -R /app; fi
if [ ! -d /scratch ]; then mkdir -p /scratch; chown easybuild:easybuild -R /scratch; fi
```

Default Singularity Template

Is this too strict?

- Just recently, switching to easybuild user in **%post section** was not working and resulted in eb to run as root and complained. See #3171
- The easybuild user inside container will install software in /app and everything else in /scratch
- After the build, easybuild will update the system spider cache to ensure spider is up to date

```
# change to 'easybuild' user
su - easybuild

# verbose commands, exit on first error
set -ve

# configure EasyBuild

# use /scratch as general prefix, used for sources, build directories, etc.
export EASYBUILD_PREFIX=/scratch

# also use /scratch for temporary directories
export EASYBUILD_TMPDIR=/scratch/tmp

# download sources to /scratch/sources, but also consider files located in /tmp/easybuild/sources;
# that way, source files that can not be downloaded can be seeded in
export EASYBUILD_SOURCEPATH=/scratch/sources:/tmp/easybuild/sources

# install software & modules into /app
export EASYBUILD_INSTALLPATH=/app

# use EasyBuild to install specified software
eb %(easyconfigs)s --robot %(eb_args)s

# update Lmod cache
mkdir -p /app/lmodcache
$LMOD_DIR/update_lmod_system_cache_files -d /app/lmodcache -t /app/lmodcache/timestamp /app/modules/all

# exit from 'easybuild' user
exit
```

```
# use EasyBuild to install specified software
eb Anaconda3-5.3.0.eb --robot
+ eb Anaconda3-5.3.0.eb --robot
/bin/eb: line 77: which: command not found
ERROR: You seem to be running EasyBuild with root privileges which is not wise,
```

Default Singularity Template

- After the build, the root user will purge /scratch where easybuild stores the source files, build directory, log files etc…
- %environment section allows you to define environment variable set at runtime.
- Currently, easybuild will attempt to load modules inside container, and avoid mixing modules from host environment
- We can do better!

```
# cleanup, everything in /scratch is assumed to be temporary
rm -rf /scratch/*

%%runscript
eval "$@"

%%environment
# make sure that 'module' and 'ml' commands are defined
source /etc/profile
# increase threshold time for Lmod to write cache in $HOME (which we don't want to do)
export LMOD_SHORT_TIME=86400
# purge any modules that may be loaded outside container
module --force purge
# avoid picking up modules from outside of container
module unuse $MODULEPATH
# pick up modules installed in /app
module use /app/modules/all
# load module(s) corresponding to installed software
module load %(mod_names)s

%%labels

"""
```

Easybuild Base Image

Yum bootstrap using Centos 7

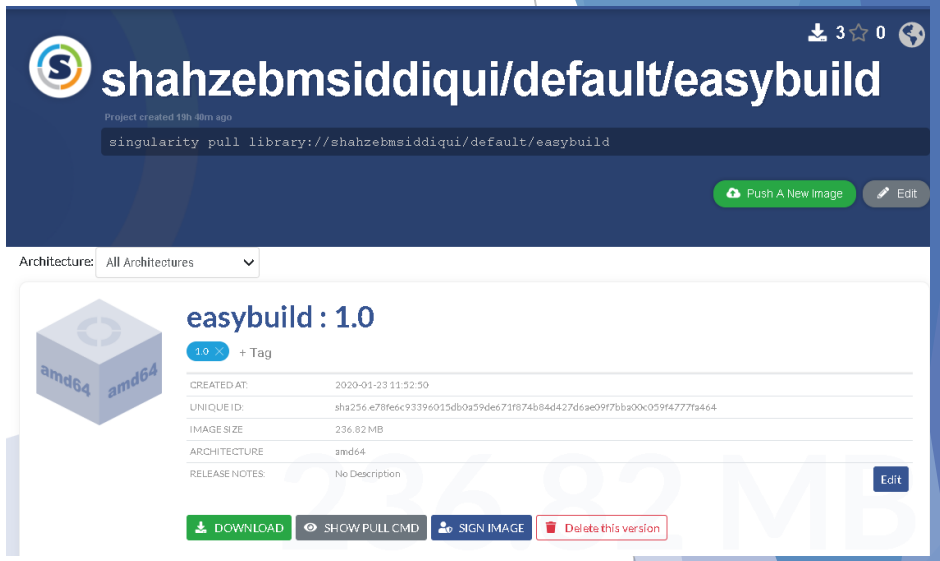
Package Dependencies for Easybuild with exception package dependency for OpenMPI.

Create easybuild user and inject easybuild configuration into startup files

Directory where easybuild user needs access to build app.

System Lmodrc configuration required for updating Lmod cache

```
Bootstrap: yum
OSVersion: 7
MirrorURL: http://mirror.centos.org/centos-%{OSVERSION}/%{OSVERSION}/os/x86_64/
Include: yum

%post
yum install --quiet --assumeyes epel-release
yum install --quiet --assumeyes python setuptools Lmod
yum install --quiet --assumeyes python-pip
yum install --quiet --assumeyes bzip2 gzip tar zip unzip xz
yum install --quiet --assumeyes curl wget
yum install --quiet --assumeyes patch make
yum install --quiet --assumeyes file git which
yum install --quiet --assumeyes gcc-c++
yum install --quiet --assumeyes perl-Data-Dumper
yum install --quiet --assumeyes perl-Thread-Queue
yum --skip-broken --quiet --assumeyes install openssl-devel libssl-dev libopenssl-devel


# create 'easybuild' user (if missing)
id easybuild || useradd easybuild

cat >> /home/easybuild/.bashrc <<-EOM
export EASYBUILD_PREFIX=/scratch
export EASYBUILD_TMPDIR=/scratch/tmp
export EASYBUILD_SOURCEPATH=/scratch/sources:/tmp/easybuild/sources
export EASYBUILD_INSTALLPATH=/app
EOM

cat >> /home/easybuild/.bash_profile <<-EOM
export EASYBUILD_PREFIX=/scratch
export EASYBUILD_TMPDIR=/scratch/tmp
export EASYBUILD_SOURCEPATH=/scratch/sources:/tmp/easybuild/sources
export EASYBUILD_INSTALLPATH=/app
EOM

# create /app software installation prefix + /scratch sandbox directory
if [ ! -d /app ]; then mkdir -p /app; chown easybuild:easybuild -R /app; fi
if [ ! -d /scratch ]; then mkdir -p /scratch; chown easybuild:easybuild -R /scratch; fi

# install Lmod RC file
cat > /etc/lmodrc.lua << EOF
scDescriptT = {
  {
    ["dir"]       = "/app/lmodcache",
    ["timestamp"] = "/app/lmodcache/timestamp",
  },
}
EOF

%runscript
eval "$@"

%environment

%labels
```



Bootstrap with easybuild:1.0

```
Bootstrap: library
From:  shahzebmsiddiqui/default/easybuild:1.0

%post

# install EasyBuild using pip
pip install -U pip
pip install wheel
pip install -U setuptools
pip install 'vsc-install<0.11.4' 'vsc-base<2.9.0'
pip install easybuild


# change to 'easybuild' user
su - easybuild << 'EOF'
set -ve
eb M4-1.4.17.eb --robot
mkdir -p /app/lmodcache
$LMOD_DIR/update_lmod_system_cache_files -d /app/lmodcache -t /app/lmodcache/timestamp /app/modules/all
EOF

# cleanup, everything in /scratch is assumed to be temporary
rm -rf /scratch/*

%runscript
eval "$@"

%environment
# make sure that 'module' and 'ml' commands are defined
source /etc/profile
# increase threshold time for Lmod to write cache in $HOME (which we don't want to do)
export LMOD_SHORT_TIME=86400
# purge any modules that may be loaded outside container
module --force purge
# avoid picking up modules from outside of container
module unuse $MODULEPATH
# pick up modules installed in /app
module use /app/modules/all
# load module(s) corresponding to installed software
module load M4/1.4.17

%labels
```

Bootstrap with easybuild:1.0

Installing easybuild and its dependencies via pip

Build application in container and update Lmod Cache

```
ssi29@ag-mxg-hulk090> ml

Currently Loaded Modules:
  1) GCCcore/9.2.0   2) zlib/1.2.11   3) binutils/2.32   4) GCC/9.2.0-2.32


ssi29@ag-mxg-hulk090> ./m4-1.4.17.sif ml

Currently Loaded Modules:
  1) M4/1.4.17
```

Easybuild Container that can't download source files automatically

- Java is an example where one needs to download source file before building. If you try building without downloading the tarball you will get this error

```
eb Java-1.8.0_92.eb --robot
== temporary log file in case of crash /scratch/tmp/eb-uKrjoR/easyl
== resolving dependencies ...
== processing EasyBuild easyconfig /usr/easybuild/easyconfigs/j/Ja
== building and installing Java/1.8.0_92...
== fetching files...
== FAILED: Installation ended unsuccessfully (build directory: /sc
ing it didn't work either... Paths attempted (in order): /usr/easyl
asyconfigs/j/Java/jdk-8u92 (took 0 sec)
== Results of the build can be found in the log file(s) /scratch/t
ERROR: Build of /usr/easybuild/easyconfigs/j/Java/Java-1.8.0_92.eb
```

- To fix this place the tarball in /tmp/easybuild/sources and rebuild container.
- Recall that /tmp/easybuild/sources on host is bind inside container at /scratch/sources

```
export EASYBUILD_SOURCEPATH=/scratch/sources:/tmp/easybuild/sources
```

Java Example

```
+ su - easybuild
eb Java-1.8.0_92.eb --robot
== temporary log file in case of crash /scratch/tmp/eb-giWOaQ/easybuild-AD_DSC.log
== resolving dependencies ...
== processing EasyBuild easyconfig /usr/easybuild/easyconfigs/j/Java/Java-1.8.0_92.eb
== building and installing Java/1.8.0_92...
== fetching files...
== creating build dir, resetting environment...
== unpacking...
== patching...
== preparing...
== configuring...
== building...
== testing...
== installing...
== taking care of extensions...
== restore after iterating...
== postprocessing...
== sanity checking...
== cleaning up...
== creating module...
== permissions...
== packaging...
== COMPLETED: Installation ended successfully (took 8 sec)
== Results of the build can be found in the log file(s) /app/software/Java/1.8.0_92/easybuild/easybuild-Java-1.8.0_92-20200124.043503.log
== Build succeeded for 1 out of 1
== Temporary log file(s) /scratch/tmp/eb-giWOaQ/easybuild-AD_DSC.log* have been removed.
== Temporary directory /scratch/tmp/eb-giWOaQ has been removed.
mkdir -p /app/lmodcache
$LMOD_DIR/update_lmod_system_cache_files -d /app/lmodcache -t /app/lmodcache/timestamp /app/modules/all
+ rm -rf /scratch/build /scratch/ebfiles_repo /scratch/tmp
INFO:    Adding environment to container
INFO:    Adding runscript
INFO:    Creating SIF file...
INFO:    Build complete: java-1.8.0_92.sif
```

Easyconfig from upstream easybuild-easyconfigs repo.

Cleaning up inside container

Creating SIF image.

Shelling into container

- Lmod is configured such that **singularity shell** will load modules inside container.

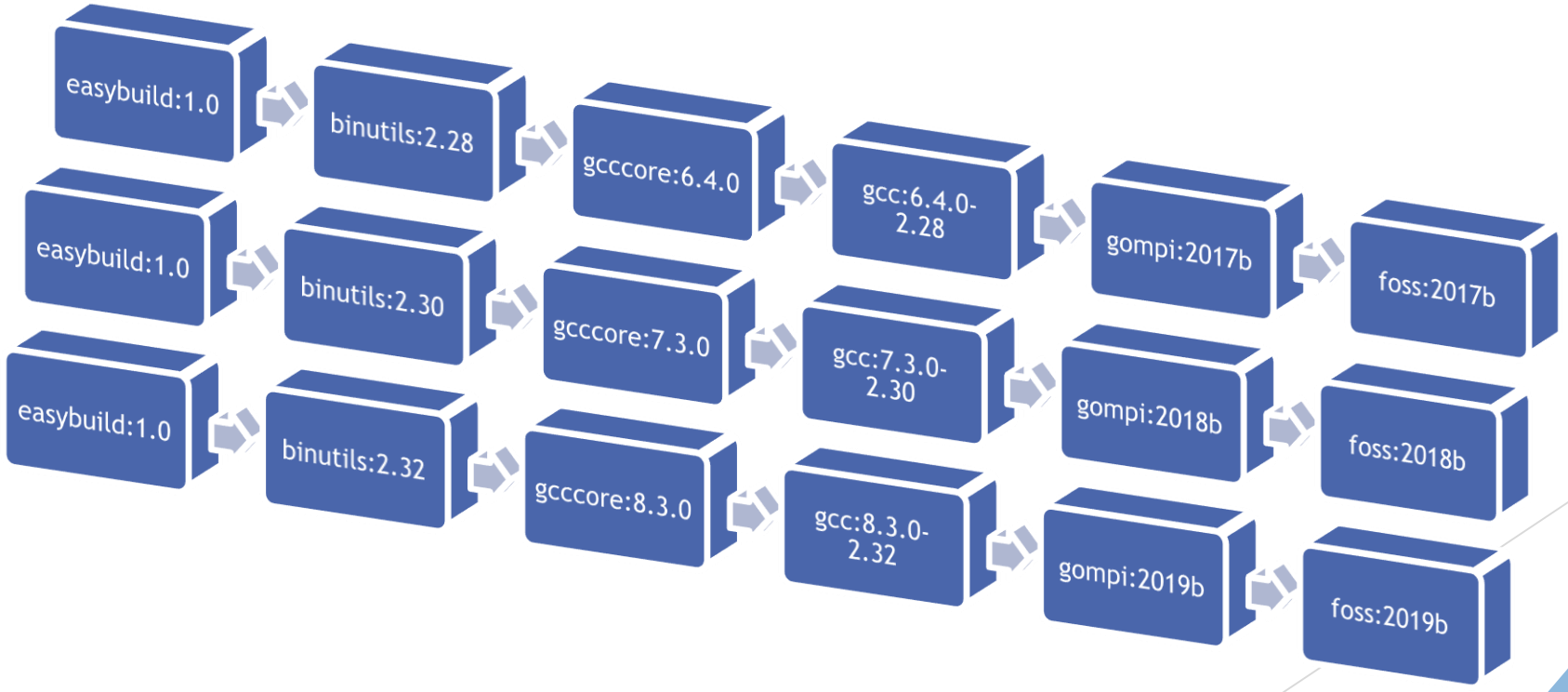
```
ssi29@ag-mxg-hulk090> singularity shell binutils-2.28.sif
Singularity binutils-2.28.sif:~/gpfs/easybuild/containers> ml av

------------------------------------------------------------------------------------------ /app/modules/all ---
   Bison/3.0.4    M4/1.4.17    M4/1.4.18 (D)    binutils/2.28 (L)    flex/2.6.3    help2man/1.47.4    zlib/1.2.11

  Where:
   L:  Module is loaded
   D:  Default Module
```

- Recall that host environment variables are passed into to container so you might be confused if you try searching for MODULEPATH inside container

```
ssi29@ag-mxg-hulk090> singularity run binutils-2.28.sif echo  $MODULEPATH  | sed 's/:/\n/g'
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core
/mxg-hpc/users/ssi29/easybuild/modules/all
/etc/modulefiles
/usr/share/modulefiles
/usr/share/modulefiles/Linux
/usr/share/modulefiles/Core
/usr/share/lmod/lmod/modulefiles/Core
```

Easybuild Toolchain Stacking

- All easybuild container come from a base image. Currently easybuild:1.0 is a Centos 7 image with Lmod using EasybuildMNS.
- Container Stacking reduces build time for application and it gives user freedom to pick any toolchain container as a starting point.



Current state of Easybuild Containers

m4 : 1.4.17

1.4.17 + Tag

CREATED AT:	2020-01-23 12:12:48
UNIQUE ID:	sha256.6c8770f1c44dbf503d17488ac5901a537a17083dda00fc16e4fbbf2f2cb28c1
IMAGE SIZE	269.62 MB
ARCHITECTURE	amd64
RELEASE NOTES:	No Description

m4 : 1.4.18

1.4.18 + Tag

CREATED AT:	2020-01-23 12:35:43
UNIQUE ID:	sha256.0706a3d4c92e89d3ffa293edf90c2a321448c11a8a51a84095bd0351d6378eed
IMAGE SIZE	269.60 MB
ARCHITECTURE	amd64
RELEASE NOTES:	No Description

binutils : 2.28

2.28 + Tag

CREATED AT:	2020-01-23 12:35:05
UNIQUE ID:	sha256.dc127177ded90f7a5f1701556494516a98b6a8d6a47d7271b8f2d195d33a4641
IMAGE SIZE	321.39 MB
ARCHITECTURE	amd64
RELEASE NOTES:	No Description

anaconda3 : 5.3.0

5.3.0 + Tag

CREATED AT:	2020-01-23 09:46:04
UNIQUE ID:	sha256.6015a2438defdd603ca36e6b4b1ad1bfd55954ecf3517406fbe9023607786613
IMAGE SIZE	1.66 GB
ARCHITECTURE	amd64
RELEASE NOTES:	No Description

easybuild : 1.0

1.0 + Tag

CREATED AT:	2020-01-23 11:52:50
UNIQUE ID:	sha256.e78fe6c93396015db0a59de671f874b84d427d6ae09f7bba00c059f4777fa464
IMAGE SIZE	236.82 MB
ARCHITECTURE	amd64
RELEASE NOTES:	No Description

java : 1.8.0_92

1.8.0_92 + Tag

CREATED AT:	2020-01-23 11:49:19
UNIQUE ID:	sha256.3fdbbdcfa646407e7102682249b0edf7d7ccdc10961a741f2ef0b2b85fa09f08
IMAGE SIZE	442.92 MB
ARCHITECTURE	amd64
RELEASE NOTES:	No Description

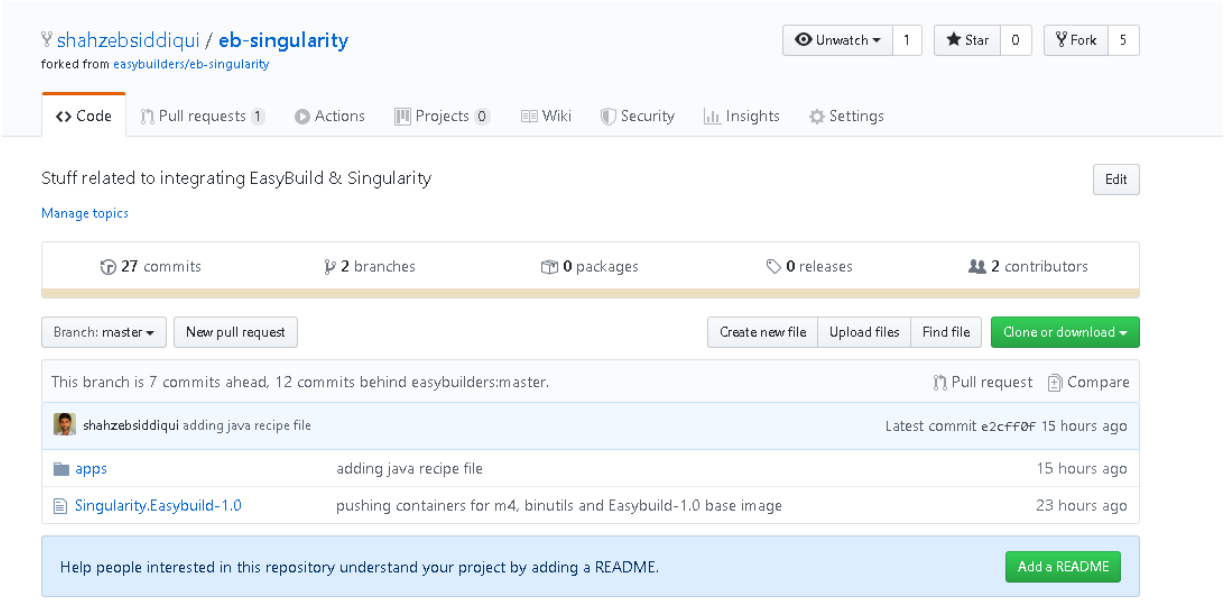
anaconda2 : 5.2.0

5.2.0 + Tag

CREATED AT:	2020-01-23 11:08:07
UNIQUE ID:	sha256.3865a9b397521078c586b1cfcf1abef922dd925ce9a2ca28249f56575d0a0004
IMAGE SIZE	1.61 GB
ARCHITECTURE	amd64
RELEASE NOTES:	No Description

Repository

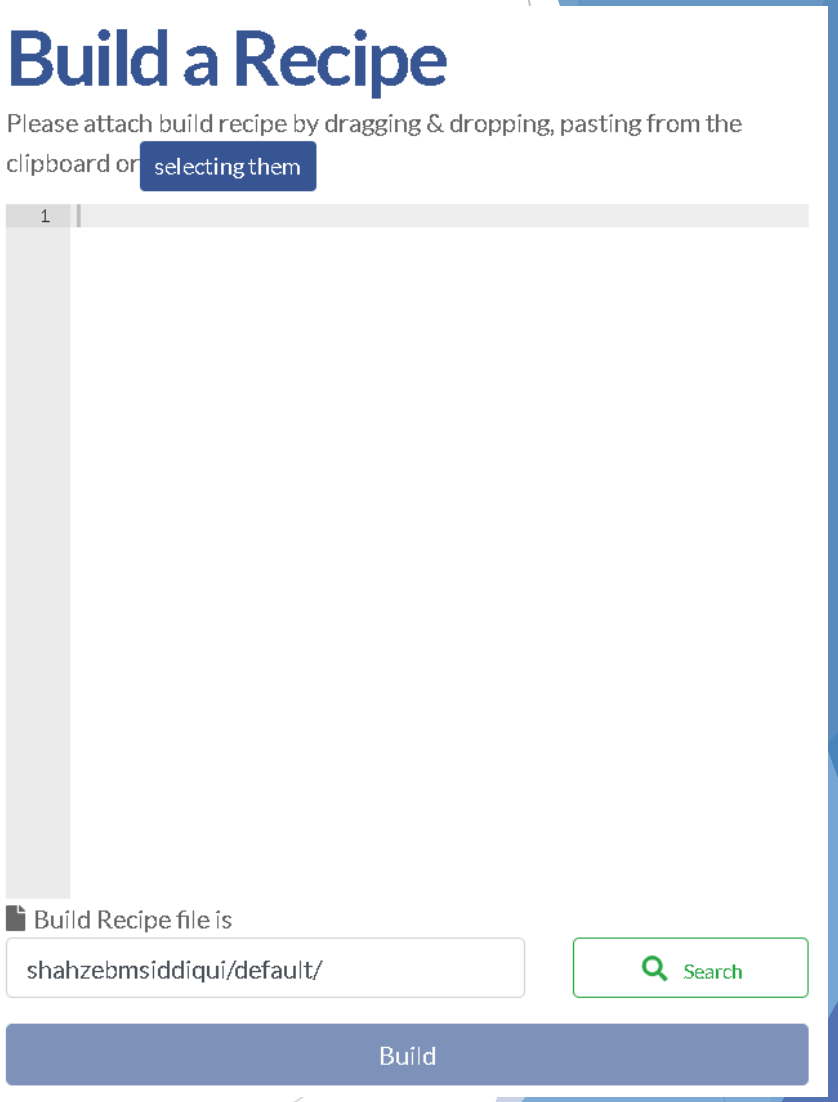
- Currently all recipe files are accessible in master branch at https://github.com/shahzebsiddiqui/eb-singularity which soon should be pushed to upstream at https://github.com/easybuilders/eb-singularity




```
.
├── apps
│   ├── anaconda2
│   │   └── Singularity.Anaconda2-5.3.0
│   ├── anaconda3
│   │   └── Singularity.Anaconda3-5.3.0
│   ├── binutils
│   │   └── Singularity.binutils-2.28
│   ├── bzip2
│   │   └── Singularity.bzip2-1.0.6
│   ├── java
│   │   └── Singularity.Java-1.8.0_92
│   └── m4
│       ├── Singularity.M4-1.4.17
│       └── Singularity.M4-1.4.18
└── Singularity.Easybuild-1.0
```

Sylabs Cloud

- According to Sylabs all builds under the free edition are performed on AWS T3 machine with a 60min timeout.
- Containers hosted in Sylabs Cloud are hosted in a library associated to a user account. Currently, there is no concept of hosting container under an organization (group library) to host easybuild containers.
- Sylabs Web Builder https://cloud.sylabs.io/builder can build containers on web by specifying recipe file, though it is not clear how to pass additional source files (tarballs, or %files section).
- SyLabs Enterprise is the on-premise instance of Sylabs Clouds that can be used for full control of building containers locally.



QA?

- Currently, easybuild automates recipe generation and container build (--container-build-image). Should easybuild support both features?
- Should we store recipe files in GitHub (https://github.com/easybuilders/eb-singularity)
- Should we build and publish all containers in Sylabs Cloud library?
- Can we allow easybuild containers to be built locally? If so which architecture do we support?
- Should we sign all containers that is pushed to Sylabs? If so which user signs it?
- Can we integrate eb --new-pr for recipe files?
- Shall we create multiple easybuild base images (Centos 7, 8)
- Should we create domain specific base images (Bio-informatics, Genomics, Chemistry, Statistics)
- Do we agree on the naming scheme for container recipe Singularity.<APP>-<VERSION>-<TOOLCHAIN>