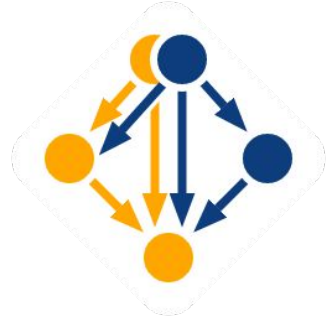




EIC Spack-tacular

- github.com/spack: “A flexible package manager supporting multiple versions, configurations, platforms, and compilers.”
- spack.io: “Spack is a package manager for supercomputers, Linux, and macOS. It makes installing scientific software easy.”
- Benefits for EIC:
 - Support for environments of scientific software, natively compiled on HPC architectures
 - Entirely controlled by user (think `conda create myenv, conda activate myenv`)
- Disadvantages:
 - Yet another package manager to support when “everyone can just run `cmake .. && make`”
 - Primarily automates build from source; not a binary distribution system without add'l effort



EIC Spack-tacular: How To Get Started?

- spack.readthedocs.io:
 - `git clone https://github.com/spack/spack.git`
 - `export SPACK_ROOT=`realpath spack``
 - `export PATH=$SPACK_ROOT/bin:$PATH`
 - `source $SPACK_ROOT/share/spack/setup-env.sh`
 - `spack install root`
- **Find packages:** `spack list root`
- **Info on packages:** `spack info root`
- **Use variants:** `spack install root@6.14.04+pythia8`
- **Load packages:** `spack load root (like module load root)`
- **Load environment (like conda env):**
 - `spack env create myenv`
 - `spack env activate myenv`
 - `spack env deactivate myenv`



EIC Spack-tacular: Writing Packages

- From source location, e.g.
 - `spack create https://gitlab.com/eic/eic-smear`
 - Imports released version, supports git branches (`spack install eic-smear@master`)
 - Autodetection of build system not always successful (eic-smear needed cmake hint)
- Package recipe in `repos/builtin/eic-smear/package.py`

```
class EicSmear(CMakePackage):  
    """Monte Carlo analysis package developed by BNL."""
```

```
homepage = "https://wiki.bnl.gov/eic/index.php"  
url      = "https://gitlab.com/eic/eic-smear"  
git      = "https://gitlab.com/eic/eic-smear.git"
```

```
variant("pythia6", default=False,  
        description="Include Pythia6 support")
```

```
version('master', branch='master')  
version('1.0.4', branch='1.0.4')  
version('1.0.3', branch='1.0.3')
```

```
version('1.0.2', branch='1.0.2')  
version('1.0.1', branch='1.0.1')
```

```
depends_on('root')  
depends_on('cmake', type='build')  
depends_on('pythia6', when='+pythia6')
```

```
def cmake_args(self):  
    args = []  
    if self.spec.variants['pythia6']:  
        args.append('-DPYTHIA6_LIBDIR={0}'.format(  
            self.spec['pythia6'].prefix.lib))  
    return args
```



EIC Spack-tacular: Repositories

- Builtin repository though pull request on github.com/spack
 - “Your PR must pass Spack's unit tests and documentation tests, and must be [PEP 8](#) compliant. We enforce these guidelines with Travis CI.”
- Dedicated repositories with `git repo add`
 - `git clone https://gitlab.com/eic/spack.git spack-eic`
 - `spack repo add spack-eic`
 - `spack install eic-smear`
- Binary distribution through build caches (with `http mirror`)
 - `spack gpg init`
 - `spack gpg create `git config --get user.name` `git config --get user.email``
 - `spack buildcache create -d ~/scratch/spack/ root`
 - `spack mirror add data file://$HOME/scratch/spack/`
 - `spack buildcache list`
 - `spack buildcache install`



EIC Spack-tacular: Containers

- From environments to Docker containers

- `spack env create myenv`
- `spack env activate myenv`
- `spack install eic-software-stack`
- `spack env deactivate myenv`
- `spack containerize myenv > Dockerfile`

```
# Build stage with Spack pre-installed and ready to be used
FROM spack/ubuntu-bionic:latest as builder
```

```
# What we want to install and how we want to install it
# is specified in a manifest file (spack.yaml)
```

```
RUN mkdir /opt/spack-environment \
&& (echo "spack:" \
&& echo " specs:" \
&& echo " - eic-smear" \
&& echo " view: /opt/view" \
&& echo " concretization: together" \
&& echo " config:" \
&& echo " install_tree: /opt/software") > /opt/spack-environment/spack.yaml
```

```
# Install the software, remove unnecessary deps
```

```
RUN cd /opt/spack-environment && spack install && spack gc -y
```

```
# Strip all the binaries
```

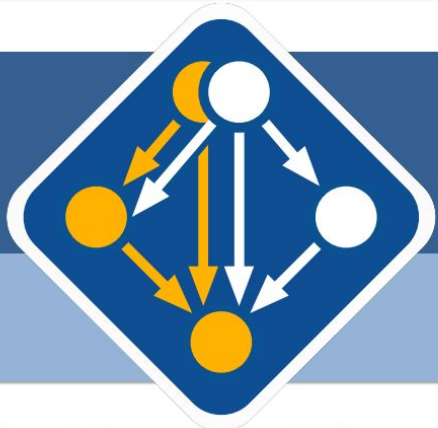
```
RUN find -L /opt/view/* -type f -exec readlink -f '{}' \; | \
xargs file -i | \
grep 'charset=binary' | \
grep 'x-executable|x-archive|x-sharedlib' | \
awk -F: '{print $1}' | xargs strip -s
```

```
# Modifications to the environment that are necessary to run
```

```
RUN cd /opt/spack-environment && \
spack env activate --sh -d . >> /etc/profile.d/z10_spack_environment.sh
```

```
# Bare OS image to run the installed executables
```

```
FROM ubuntu:18.04
COPY --from=builder /opt/spack-environment /opt/spack-environment
COPY --from=builder /opt/software /opt/software
COPY --from=builder /opt/view /opt/view
COPY --from=builder /etc/profile.d/z10_spack_environment.sh /etc/profile.d/z10_spack_environment.sh
ENTRYPOINT ["/bin/bash", "--rcfile", "/etc/profile", "-l"]
```



Spack

Building & Deploying the ECP Software Ecosystem

Gregory Becker*, Peter Scheibel*, Tamara Dahlgren*, Mario Melara†, Scott Wittenburg‡, Omar Padron‡, Bryon Bean‡, Zack Galbreath‡, Aashish Chaudhary, and Massimiliano Culpo[§], and Todd Gamblin*

*Lawrence Livermore National Laboratory, †Lawrence Berkeley National Laboratory, ‡Kitware, Inc., §Sylabs, Inc.

Spack is enabling delivery of the exascale software stack

- ECP asks us to build a software stack that will have broad impact beyond DOE.
 - Needs to be robust, tested, and reliable
 - Needs to be easy to get up and running
- Spack will provide the infrastructure necessary to make this tractable through automation:
 - A dependency model that can handle HPC software
 - A hub for coordinated software releases (like xSDK)
 - Build and test automation for large packages across facilities
 - Hosted binary and source software distributions for *all* ECP HPC platforms

Easy Installation

```
$ git clone https://github.com/spack/spack
$ . spack/share/spack/setup-env.sh
$ spack install hdf5
```

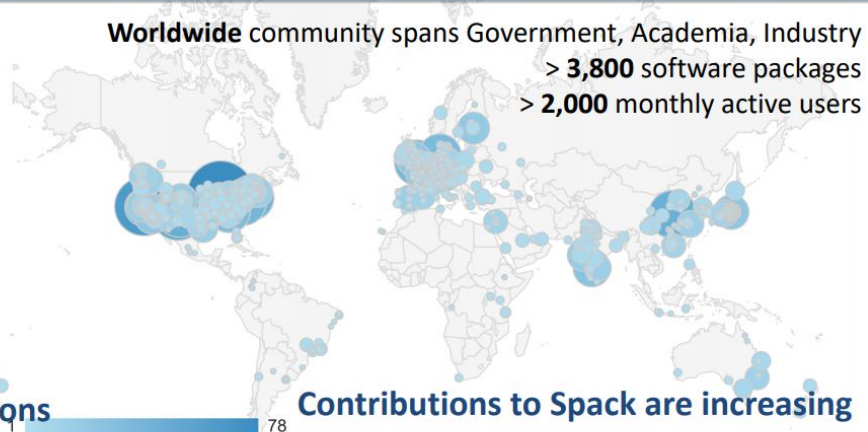
- Clone from github and you're ready to go!
- Sourcing configuration script is optional

Easily Experiment with Build Options

```
$ spack install mpileaks
$ spack install mpileaks@3.3
$ spack install mpileaks@3.3 %gcc@4.7.3
$ spack install mpileaks@3.3 %gcc@4.7.3 +threads
$ spack install mpileaks@3.3 cppflags="-O3 -g3"
$ spack install mpileaks@3.3 target=skylake
$ spack install mpileaks@3.3 ^mpich@3.2 %gcc@4.9.3
```

unconstrained
 @ custom version
 % custom compiler
 +/- build option
 set compiler flags
 set target microarchitecture
 ^ dependency information

Worldwide community spans Government, Academia, Industry
 > 3,800 software packages
 > 2,000 monthly active users



Contributions to Spack are increasing

