# ONE STACK TO BUILD THEM ALL:

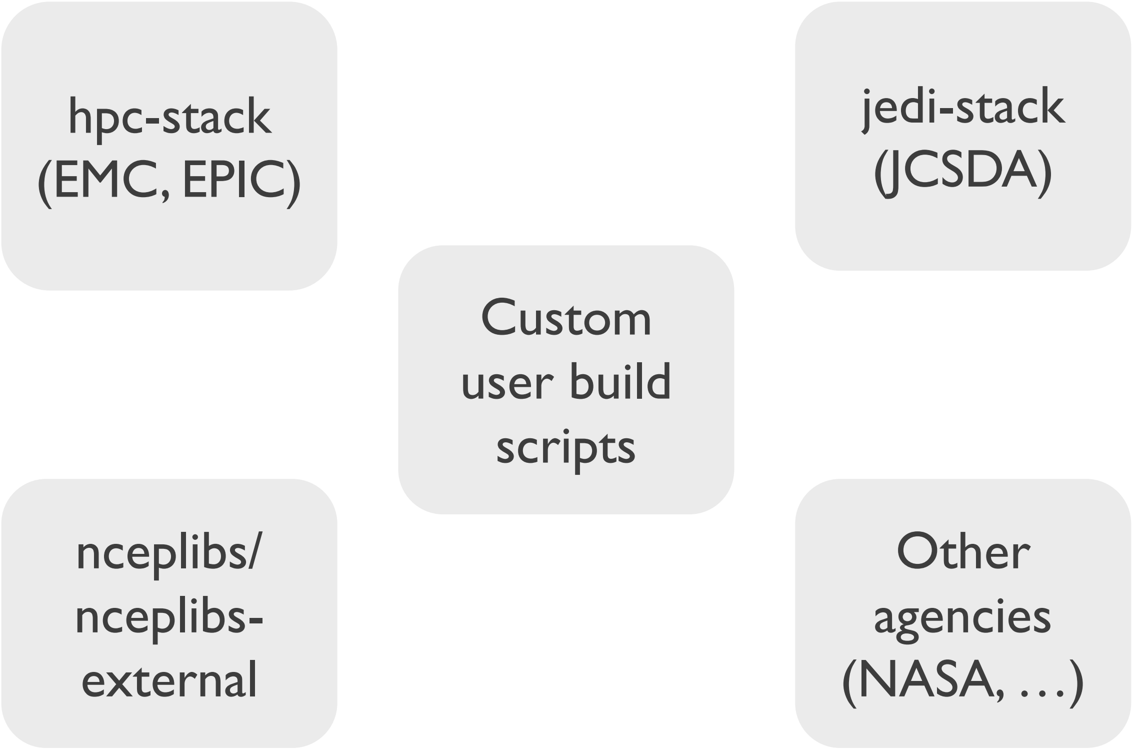
# THE NOAA-EMC/JCSDA SPACK-STACK

DOM HEINZELLER[1], KYLE GERHEISER[2],
STEVE HERBENER[1], ED HARNETT[3] ,
ARUN CHWALA[3]

[1] UCAR/UCP/JCSDA, BOULDER, CO, USA
[2] REDLINE PERFORMANCE AT NOAA EMC
[3] NOAA EMC, COLLEGE PARK, MD, USA

# CONTENTS

- The problem(s)
- A bold idea
- spack-stack
- User interface
- Quo vadis?



hpc-stack (EMC, EPIC)

jedi-stack (JCSDA)

Custom user build scripts

nceplibs/ nceplibs-external

Other agencies (NASA, …)

## DUPLICATION OF EFFORTS

## MULTIPLE STACKS MAINTAINED ON HPC, CLOUD, …

Compiled code

- MPI
- netCDF
- most nceplibs
- eckit, fckit, atlas

Mixed packages

- matplotlib
- numpy
- scipy
- bufr

Python packages

- MPI4py
- netCDF4py
- Boto3
- Pybind11

## PYTHON WHEELS, CONDA, VENV

## INCONSISTENT SOFTWARE STACK

## SWITCH STACK FOR DIFFERENT APPLICATION

## FRUSTRATION FOR USERS AND DEVELOPERS

Support

Documentation

# A BOLD IDEA

What if …

- we had only one software stack to develop, maintain, and support,
- that stack provided everything to build the UFS and the JEDI applications,
- there was a single authoritative installation on each supported system,
- this stack solved our Python environment issues,
- it had solid automatic testing capabilities,
- and it had comprehensive documentation?

**spack-stack**

# *SPACK-STACK*

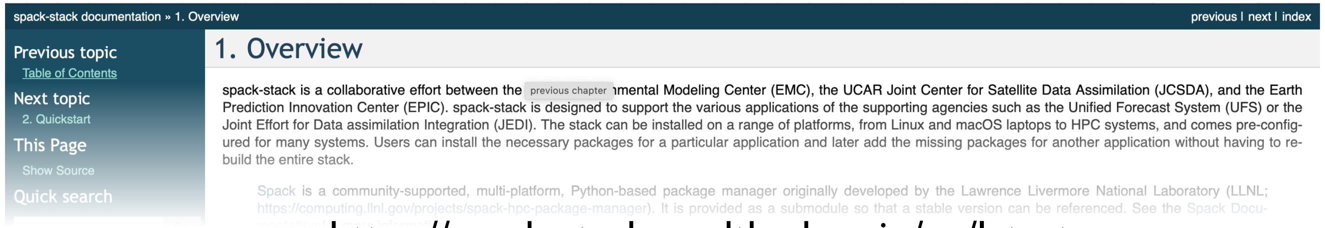
https://github.com/noaa-emc/spack-stack

Joint effort of EMC and JCSDA

Spack package manager (LLNL)

ufs-weather-model soca jedi-mpas gfs-wafs jedi-um ufs-srw-app gldas ufs-utils jedi-tools jedi-ufs global-workflow jedi-ewok nceplibs gsi ufswm-python gw-python upp jedi-fv3 ww3

One stack to build them all



## 1. Overview

spack-stack is a collaborative effort between the previous chapter mental Modeling Center (EMC), the UCAR Joint Center for Satellite Data Assimilation (JCSDA), and the Earth Prediction Innovation Center (EPIC). spack-stack is designed to support the various applications of the supporting agencies such as the Unified Forecast System (UFS) or the Joint Effort for Data assimilation Integration (JEDI). The stack can be installed on a range of platforms, from Linux and macOS laptops to HPC systems, and comes pre-configured for many systems. Users can install the necessary packages for a particular application and later add the missing packages for another application without having to rebuild the entire stack.

Spack is a community-supported, multi-platform, Python-based package manager originally developed by the Lawrence Livermore National Laboratory (LLNL; https://computing.llnl.gov/projects/spack-hpc-package-manager). It is provided as a submodule so that a stable version can be referenced. See the Spack Docu-

https://spack-stack.readthedocs.io/en/latest

Spack

https://spack.io

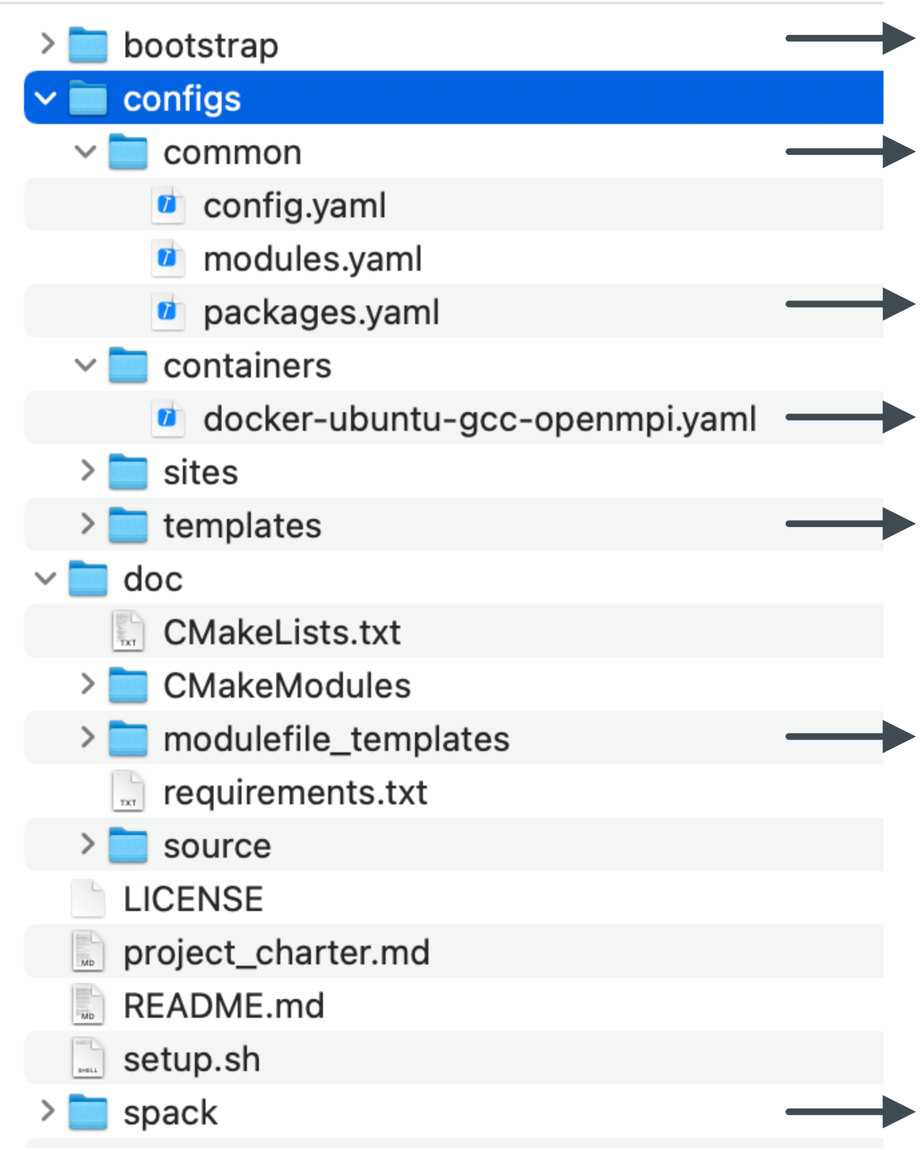
*“Spack is a package manager for supercomputers, Linux, and macOS. It makes installing scientific software easy. Spack isn't tied to a particular language; you can build a software stack in Python or R, link to libraries written in C, C++, or Fortran, and easily swap compilers or target specific microarchitectures”*

https://computing.llnl.gov/projects/spack-hpc-package-manager

- Compile all packages with the same compiler
- Use compiled dependencies when building Python packages
- Full control over versions, build types, and variants
- Create lua/lmod or tcl/tk environment modules
- Configure compiler toolchains
- Build containers from spack configuration files
- Currently 6400+ packages (1800+ Python packages)
- A huge community contributing packages and updates

# *SPACK-STACK*

https://github.com/noaa-emc/spack-stack
https://spack-stack.readthedocs.io/en/latest



```
> bootstrap
v configigs
  v common
      config.yaml
      modules.yaml
      packages.yaml
  v containers
      docker-ubuntu-gcc-openmpi.yaml
  > sites
  > templates
v doc
    CMakeLists.txt
  > CMakeModules
  > modulefile_templates
    requirements.txt
  > source
  LICENSE
  project_charter.md
  README.md
  setup.sh
> spack
```

bootstrap → Miniconda install script etc.

common → Common config files

packages.yaml → Package versions (latest)

docker-ubuntu-gcc-openmpi.yaml → Container recipes

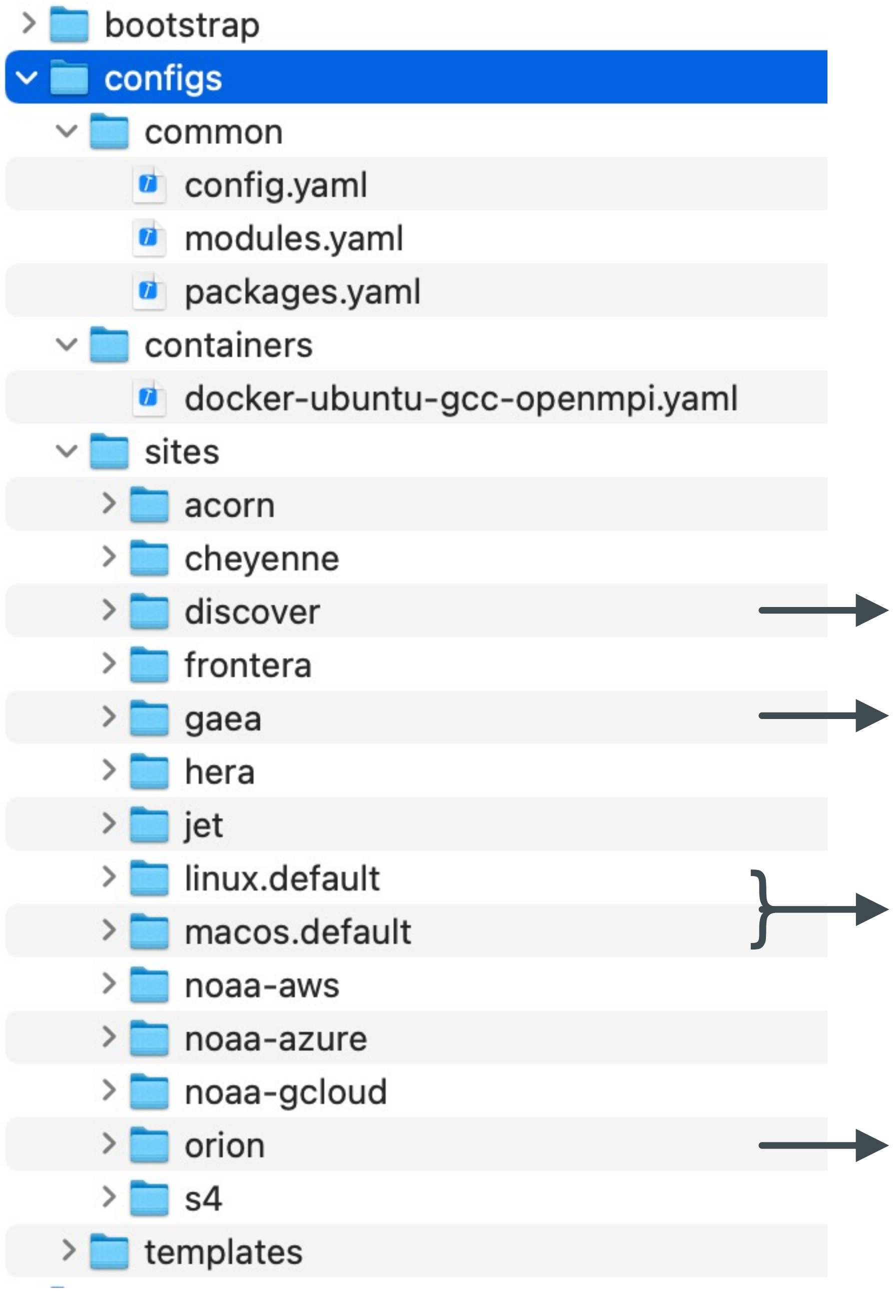
templates → Releases (packages/versions defined), e.g. skylab-1.0.0

modulefile_templates → Module templates for external dependencies (e.g. ecflow)

spack → Fork of spack repo with more packages and spack extension

# SPACK-STACK

https://github.com/noaa-emc/spack-stack
https://spack-stack.readthedocs.io/en/latest



```
> bootstrap
v configs
  v common
      config.yaml
      modules.yaml
      packages.yaml
  v containers
      docker-ubuntu-gcc-openmpi.yaml
  v sites
    > acorn
    > cheyenne
    > discover
    > frontera
    > gaea
    > hera
    > jet
    > linux.default
    > macos.default
    > noaa-aws
    > noaa-azure
    > noaa-gcloud
    > orion
    > s4
  > templates
```

discover → Fully supported in 1st release

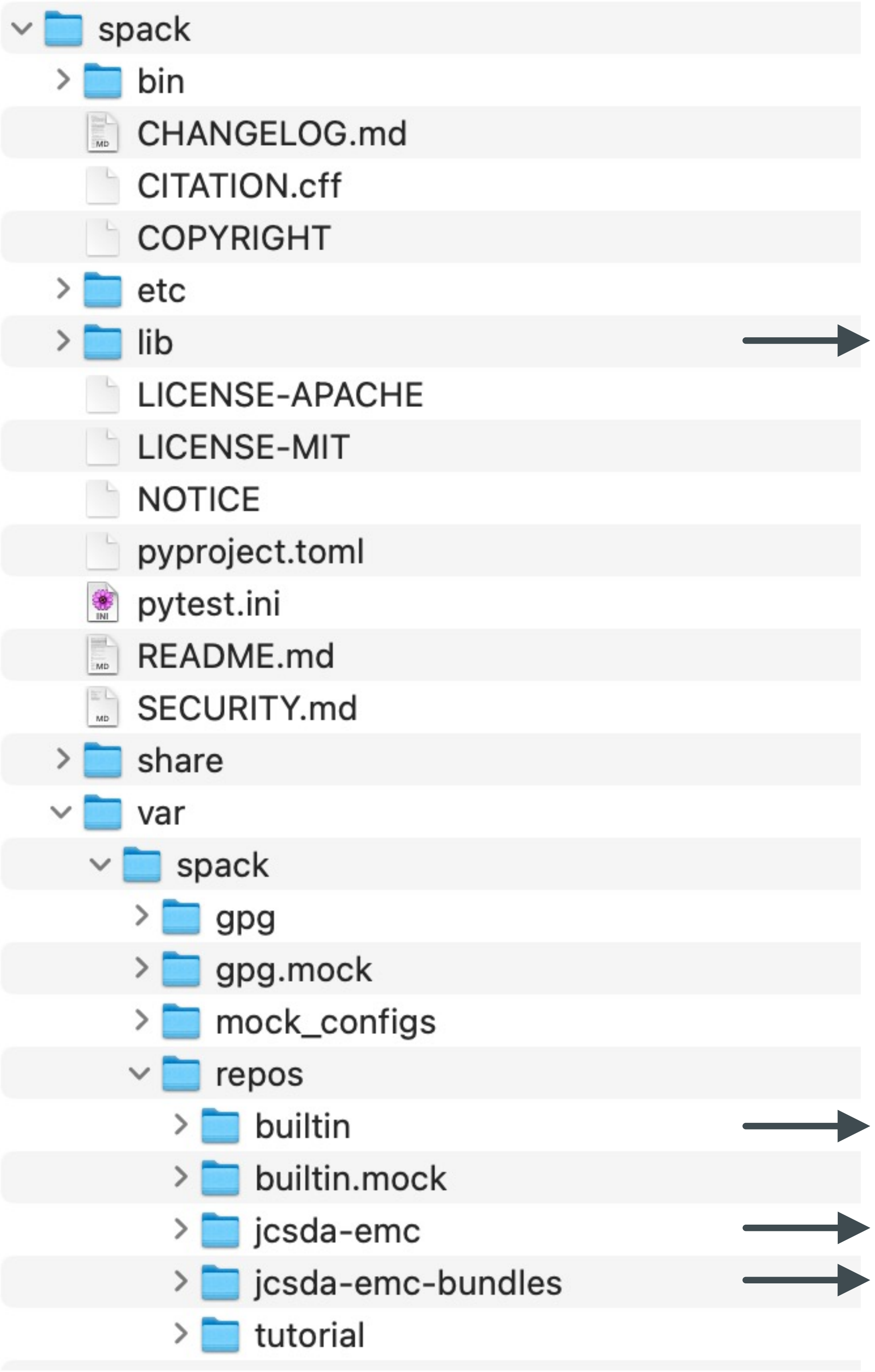
gaea → Used by NG-GODAS!

linux.default, macos.default } → Generic site configs

orion → Fully supported in 1st release

# SPACK-STACK

https://github.com/noaa-emc/spack-stack
https://spack-stack.readthedocs.io/en/latest



```
spack
  bin
  CHANGELOG.md
  CITATION.cff
  COPYRIGHT
  etc
  lib
  LICENSE-APACHE
  LICENSE-MIT
  NOTICE
  pyproject.toml
  pytest.ini
  README.md
  SECURITY.md
  share
  var
    spack
      gpg
      gpg.mock
      mock_configs
      repos
        builtin
        builtin.mock
        jcsda-emc
        jcsda-emc-bundles
        tutorial
```

lib → spack cmd extension included in this directory

builtin → Packages in auth. spack repo

jcsda-emc → Additional packages

jcsda-emc-bundles → Convenience (env) packages

*SPACK-STACK*

https://github.com/noaa-emc/spack-stack
https://spack-stack.readthedocs.io/en/latest

## Creating a new environment on a preconfigured platform

```
$ source setup.sh

$ spack stack create env --site=orion \
        --template=skylab-1.0.0 \
        --name=skylab-1.0.0

$ spack env activate -p \
        envs/skylab-1.0.0

$ spack concretize

$ spack install
```

```
$ spack module lmod refresh

$ spack stack setup-meta-modules
```

*SPACK-STACK*

https://github.com/noaa-emc/spack-stack
https://spack-stack.readthedocs.io/en/latest

## Creating a new environment on a configurable/user platform

https://spack-stack.readthedocs.io/en/latest/Platforms.html#macos

```
$ source setup.sh

$ spack stack create env --site=macos.default ...

$ spack env activate -p envs/skylab-1.0.0
```

First time only

```
$ export SPACK_SYSTEM_CONFIG_PATH="envs/skylab-.../site"

$ spack external find --scope system # finds build pkgs
# Run a few more times for other pkgs (python, ...)

$ spack compiler find --scope system # finds compilers

$ export -n SPACK_SYSTEM_CONFIG_PATH

$ spack config add ... # add package config options
```

```
$ spack concretize && spack install

$ spack module lmod refresh

$ spack stack setup-meta-modules
```

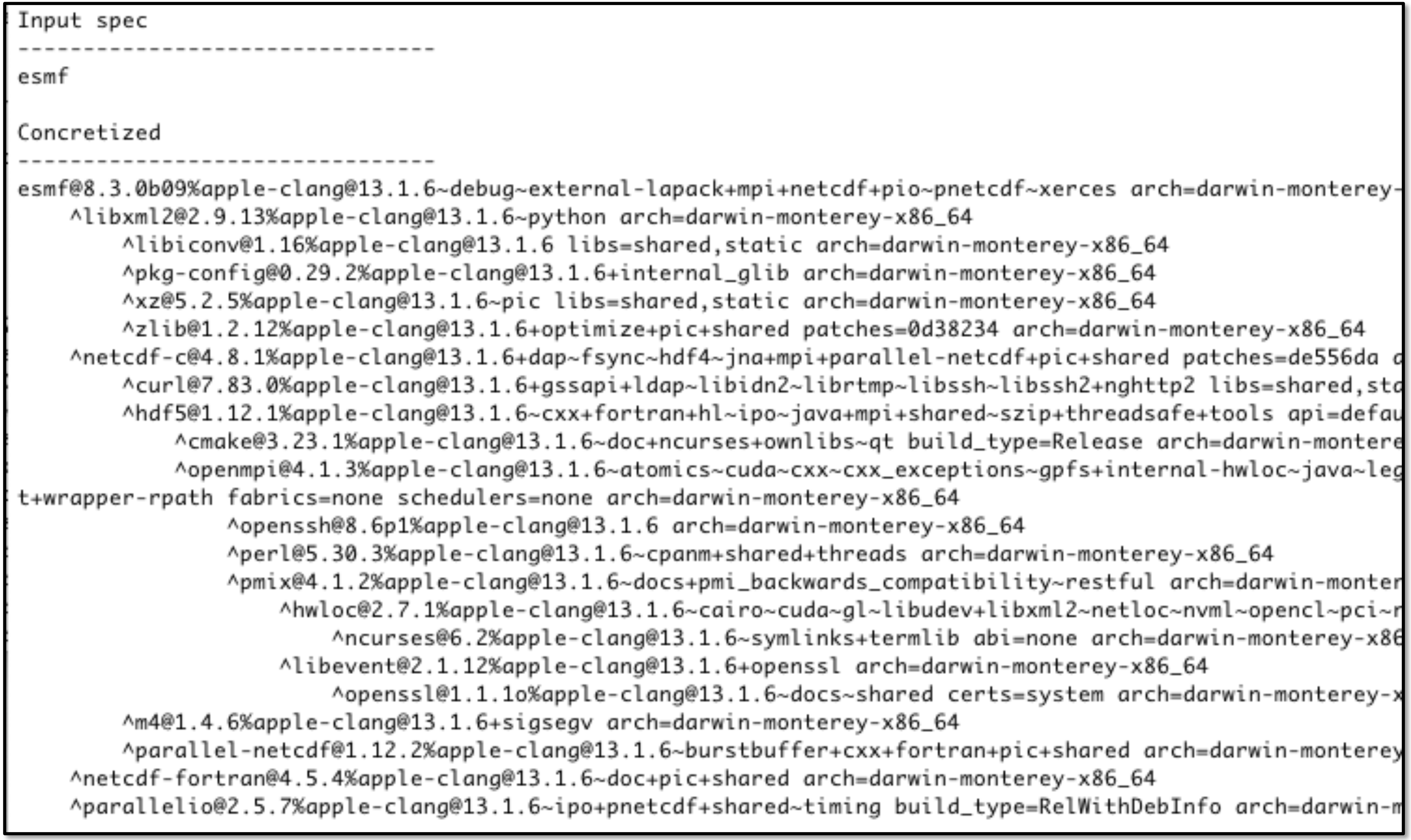
SPACK-STACK

https://github.com/noaa-emc/spack-stack
https://spack-stack.readthedocs.io/en/latest

## Useful (native) spack commands

```
$ spack env activate -p envs/skylab-1.0.0

$ spack info ecmwf-atlas

$ spack spec esmf

$ spack find [-v -l ecmwf-atlas]

$ spack add esmf@8.4.0bs08 +debug

$ spack clean -a

$ spack env deactivate

...
```

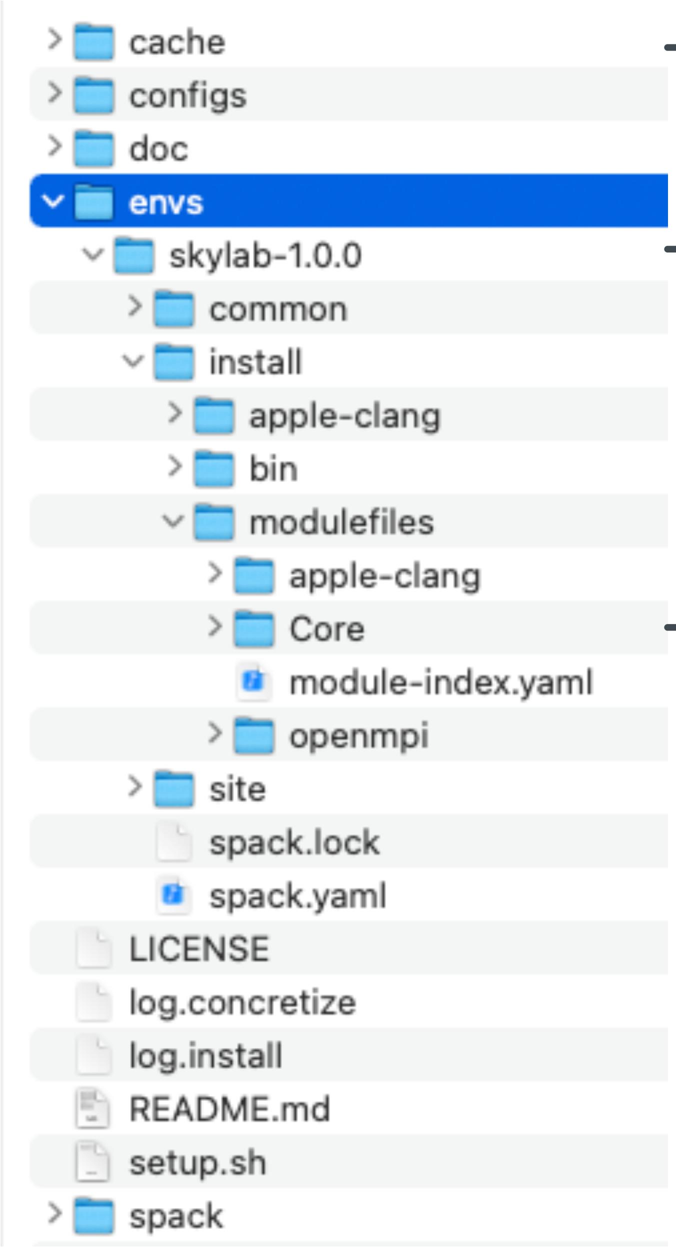
https://spack.readthedocs.io/en/latest/



```
Input spec
--------------------------------
esmf

Concretized
--------------------------------
esmf@8.3.0b09%apple-clang@13.1.6~debug~external-lapack+mpi+netcdf+pio~pnetcdf~xerces arch=darwin-monterey-
    ^libxml2@2.9.13%apple-clang@13.1.6~python arch=darwin-monterey-x86_64
        ^libiconv@1.16%apple-clang@13.1.6 libs=shared,static arch=darwin-monterey-x86_64
        ^pkg-config@0.29.2%apple-clang@13.1.6+internal_glib arch=darwin-monterey-x86_64
        ^xz@5.2.5%apple-clang@13.1.6~pic libs=shared,static arch=darwin-monterey-x86_64
        ^zlib@1.2.12%apple-clang@13.1.6+optimize+pic+shared patches=0d38234 arch=darwin-monterey-x86_64
    ^netcdf-c@4.8.1%apple-clang@13.1.6+dap~fsync~hdf4~jna+mpi+parallel-netcdf+pic+shared patches=de556da a
        ^curl@7.83.0%apple-clang@13.1.6+gssapi+ldap~libidn2~librtmp~libssh~libssh2+nghttp2 libs=shared,sta
        ^hdf5@1.12.1%apple-clang@13.1.6~cxx+fortran+hl~ipo~java+mpi+shared~szip+threadsafe+tools api=defau
            ^cmake@3.23.1%apple-clang@13.1.6~doc+ncurses+ownlibs~qt build_type=Release arch=darwin-montere
            ^openmpi@4.1.3%apple-clang@13.1.6~atomics~cuda~cxx~cxx_exceptions~gpfs+internal-hwloc~java~leg
t+wrapper-rpath fabrics=none schedulers=none arch=darwin-monterey-x86_64
                ^openssh@8.6p1%apple-clang@13.1.6 arch=darwin-monterey-x86_64
                ^perl@5.30.3%apple-clang@13.1.6~cpanm+shared+threads arch=darwin-monterey-x86_64
                ^pmix@4.1.2%apple-clang@13.1.6~docs+pmi_backwards_compatibility~restful arch=darwin-monter
                    ^hwloc@2.7.1%apple-clang@13.1.6~cairo~cuda~gl~libudev+libxml2~netloc~nvml~opencl~pci~r
                        ^ncurses@6.2%apple-clang@13.1.6~symlinks+termlib abi=none arch=darwin-monterey-x86
                    ^libevent@2.1.12%apple-clang@13.1.6+openssl arch=darwin-monterey-x86_64
                        ^openssl@1.1.1o%apple-clang@13.1.6~docs~shared certs=system arch=darwin-monterey-x
        ^m4@1.4.6%apple-clang@13.1.6+sigsegv arch=darwin-monterey-x86_64
        ^parallel-netcdf@1.12.2%apple-clang@13.1.6~burstbuffer+cxx+fortran+pic+shared arch=darwin-monterey
    ^netcdf-fortran@4.5.4%apple-clang@13.1.6~doc+pic+shared arch=darwin-monterey-x86_64
    ^parallelio@2.5.7%apple-clang@13.1.6~ipo+pnetcdf+shared~timing build_type=RelWithDebInfo arch=darwin-m
```

# *SPACK-STACK*

https://github.com/noaa-emc/spack-stack
https://spack-stack.readthedocs.io/en/latest



```
cache
configs
doc
envs
  skylab-1.0.0
    common
    install
      apple-clang
      bin
      modulefiles
        apple-clang
        Core
        module-index.yaml
        openmpi
    site
    spack.lock
    spack.yaml
LICENSE
log.concretize
log.install
README.md
setup.sh
spack
```

Source/build cache

Spack environment

Meta-modules

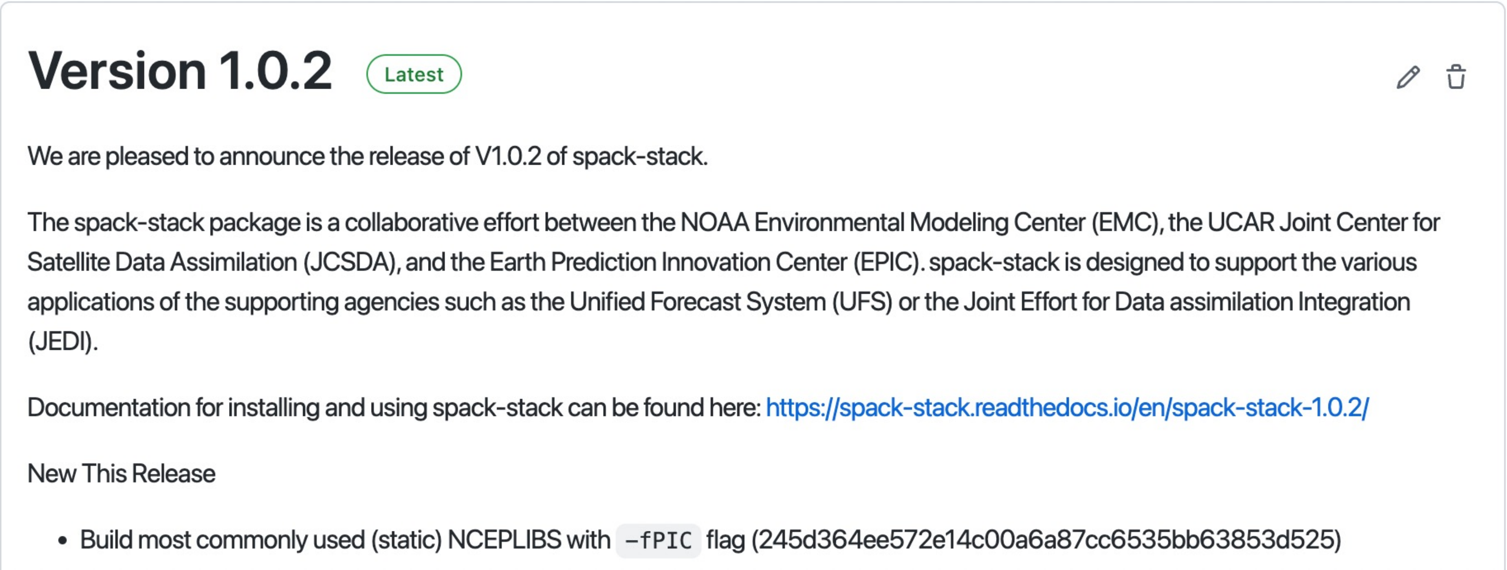
# USER INTERFACE

Almost no changes for users of hpc-stack or jedi-stack

Example: MSU Orion with Intel 2022.0.2

```
module purge

module use /path/to/external/modulefiles
module load miniconda/3.9.7
module load ecflow/5.8.4

module use /path/to/spack/env/install/modulefiles/Core
module load stack-intel/2022.0.2
module load stack-intel-oneapi-mpi/2021.5.1
module load stack-python/3.9.7

module available
```

# SPACK-STACK 1.0.2 RELEASE



## Version 1.0.2

We are pleased to announce the release of V1.0.2 of spack-stack.

The spack-stack package is a collaborative effort between the NOAA Environmental Modeling Center (EMC), the UCAR Joint Center for Satellite Data Assimilation (JCSDA), and the Earth Prediction Innovation Center (EPIC). spack-stack is designed to support the various applications of the supporting agencies such as the Unified Forecast System (UFS) or the Joint Effort for Data assimilation Integration (JEDI).

Documentation for installing and using spack-stack can be found here: https://spack-stack.readthedocs.io/en/spack-stack-1.0.2/

New This Release

- Build most commonly used (static) NCEPLIBS with `-fPIC` flag (245d364ee572e14c00a6a87cc6535bb63853d525)

- Supported platforms: NASA Discover (Intel, GNU), MSU Orion (Intel, GNU), AWS Red Hat and Ubuntu (GNU)
  - Will be expanded over the next weeks and months
- Supported application: Skylab 1.0 (JEDI-FV3 + EWOK)
  - Basically ready to use with other applications, e.g. the ufs-weather-model, but these haven't been fully tested
- CI tests using GitHub actions and self-hosted runners

# SPACK-STACK

https://github.com/noaa-emc/spack-stack
https://spack-stack.readthedocs.io/en/latest

## Work in progress: Creating containers from spack

Important: Uses a spack branch that must be accessible on Github. Packages must be accessible w/o authentication (so far).

```
$ source setup.sh
```

```
$ spack stack create ctr \
        docker-ubuntu-gcc-openmpi \
        --template=empty
```

```
$ cd envs/docker-ubuntu-gcc-openmpi

# Edit spack.yaml and add specs to build

$ spack containerize > Dockerfile

$ docker build -t myimage .
```

```
$ docker run -it myimage
```

# QUO VADIS?

- Formalizing collaboration between EMC and JCSDA on spack-stack (developing a project charter)
- Looking for additional partners
  - **EPIC?** Use spack-stack for public releases, support to the community, manage one or more platforms
  - **NASA?** Build GEOS GCM out of the box with spack-stack, share Discover management
- Release cycles, directory structures, mechanisms for automated testing on hard iron and automated updates
- Sharing platform management (installation, maintenance): roles and responsibilities of partner organizations
- Mid-term future: install UFS and JEDI releases via spack

```
spack install ufs-srw-app@3.0.1 skylab@2.0.0
```

***spack-stack***

https://github.com/NOAA-EMC/spack-stack/

https://spack-stack.readthedocs.io/en/latest/