CADRE Workshop Container

Background info and initial set up

By: Edward Snyder

https://epic.noaa.gov/

Outline

- Containers
  - What are they?
  - How do they work?
  - Software
  - NOAA EPIC Container Usage
    - spack-stack
- CADRE Workshop Container
  - What's inside?
  - How is it built?
  - Helpful commands
  - Initial set up

Resource Slide

For help today:

- slack channel - #cadre-epic-data-assimilation-training

Commands from this presentation:

- CADRE_Container_Commands.txt

Scientific/Theoretical Q's about Land DA-workflow repo:

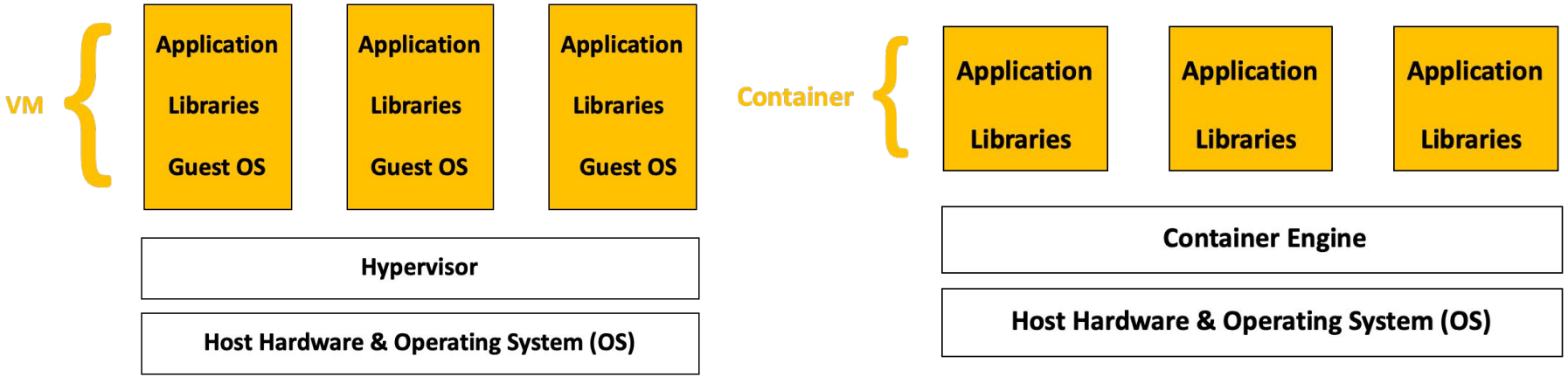
- GitHub Discussion

What are containers?

- Software that packages up an application's code and dependencies allowing for reliable runs across different computing environments.
- Benefits
  - Consistent runs
  - Faster development
  - Isolation from OS env
  - Portable
  - Scalable

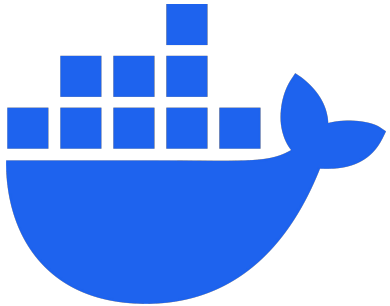
How do containers work?

- Virtualise the OS instead of the hardware (VMs)



Container Software: Docker

- Pioneer in PaaS (platform as a service)
- Client-server architecture
  - Docker commands
  - Docker daemon
  - Dockerhub
- See container benefits slides



Docker Key Terms

Dockerfile - text file containing instructions to build a Docker image. Sets base image, env variables, and commands. A recipe for a Docker image.

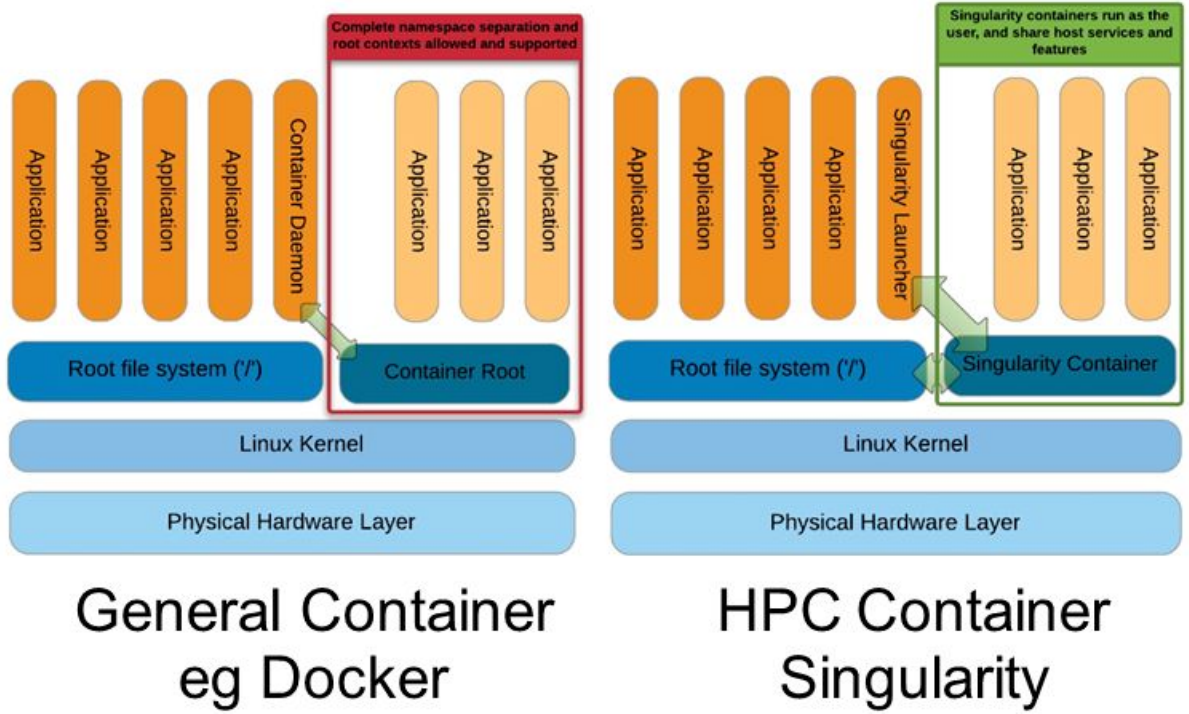
Docker image - reusable file created from the Dockerfile. It is read-only and the individual commands from the Dockerfile are layers. Blueprint for Docker container

Docker container - a running instance of the Docker image. Run the application. Can be modified and each instance is the same but separate environment.

Container Software: Apptainer (Singularity)

- Rebranded to Apptainer in 2021
- High-Performance Computing (HPC) focused
  - User permissions (Docker requires root access)
  - Ability to run MPI and GPU workflows
  - Communication with job schedulers
  - Can mount to local file system
- Compatible with Docker
- Apptainer Singularity Image Format (SIF) file
  - Not writable
  - Sandbox (writable)

Docker & Apptainer Architecture



NOAA EPIC Container Usage

- Workshops
- Application or feature releases
  - Short-Range Weather (SRW)
  - Land Data Assimilation (DA) System
  - Unified Forecast System (UFS) Weather Model (WM) Hierarchical System Develop (HSD)
- Land-DA_workflow ctest
  - Dockerfile
- spack-stack deployment
  - Research and Development High Performance Computing (RDHPC)

spack-stack

- One-stop shop for app dependency packages
  - Unified Forecast System (UFS)
  - Joint Effort for Data assimilation Integration (JEDI)
- Based on spack - a python-based package manager developed by Lawrence Livermore National Laboratory (LLNL)
- Supported by: NOAA Environmental Modeling Center (EMC), the UCAR Joint Center for Satellite Data Assimilation (JCSDA), the Earth Prediction Innovation Center (EPIC), and the U.S. Naval Research Laboratory (NRL)

CADRE Workshop Container Contents

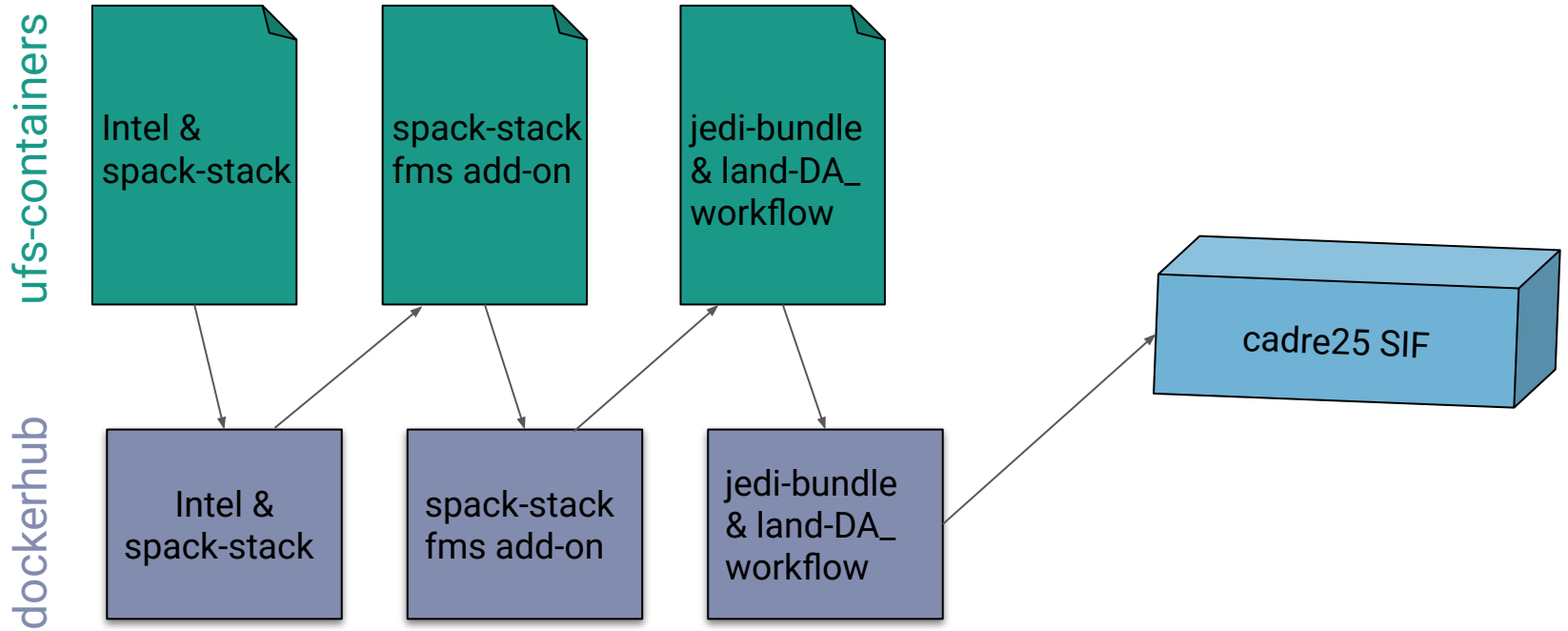
- Includes:
  - intel
  - spack-stack
  - jedi-bundle
  - land-DA_workflow

```
ubuntu@ip-10-29-93-178:~$ singularity shell -B /home/ ubuntu22.04-intel-landda-cadre25.img
Singularity> ls /opt
intel  jedi-bundle  land-DA_workflow  spack-stack
```

CADRE Workshop Container Build Steps

- Three Dockerfiles:
  - Dockerfile.ubuntu22.04-intel-unified
    - build intel and spack-stack
  - Dockerfile.ubuntu22.04-intel-unified-fms2024.01
    - spack fms add-on
  - Dockerfile.ubuntu22.04-intel21.10-ue160-fms202401-cadre25
    - build jedi-bundle and land-DA_workflow
- Leverage Jenkins to create Docker image
- Conversion to Apptainer SIF occurs on NOAA Parallel Works Cloud Platform

CADRE Workshop Container Build Diagram



Intel & spack-stack Dockerfile

```
From intel/ubuntu22.04-hpckit:2023.1.0
ARG branch_name

ENV branch=$branch_name
RUN mkdir -p /opt/build
RUN apt-get update &&   DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends     ca-certificates curl &&   rm -rf /var/lib/apt/lists/*
# install cmake
RUN cd /opt/build &&     curl -LO https://github.com/Kitware/CMake/releases/download/v3.23.1/cmake-3.23.1-linux-x86_64.sh && /bin/bash cmake-3.23.1-linux-x86_64.sh
#
# update apt
RUN apt-get update -yq --allow-unauthenticated
#
RUN DEBIAN_FRONTEND=noninteractive TZ=Etc/UTC apt-get -y install tzdata
# Install some basics that will be needed by all apps
RUN ln -fs /usr/share/zoneinfo/America/New_York /etc/localtime && \
    dpkg-reconfigure --frontend noninteractive tzdata && \
apt install -y --no-install-recommends  vim && \
apt install -y --no-install-recommends  wget && \
```

Intel & spack-stack Dockerfile

```
RUN git clone -b release/$branch --recursive https://github.com/jcsda/spack-stack.git spack-stack-$branch && \
  cd /opt/spack-stack/spack-stack-$branch && \
# set the spack directories to be safe for use by git even though they are in /opt
    git config --global --add safe.directory /opt/spack-stack/spack-stack-$branch && \
    git config --global --add safe.directory /opt/spack-stack/spack-stack-$branch/spack && \
#
# Get any recent updates to the release
    git remote update && git checkout release/$branch && git pull origin release/$branch && \
    git submodule sync && git submodule update  && \
    mv /opt/spack-stack/ubuntu-intel configs/sites && \
#
# Create the environment
    . ./setup.sh && spack compiler rm gcc@11.4.0 && \
    export SPACK_SYSTEM_CONFIG_PATH="$PWD/env/unified-env/site" && spack stack create env --site ubuntu-intel --template unified-dev --name unified-env && \
    spack compiler add oneapi && spack compiler list && sed -i "s/'%aocc', '%apple-clang', '%gcc', //g" /opt/spack-stack/spack-stack-$branch/envs/unified-env/spack.yaml && \
    sed -i 's/\=2021/2021/g' /root/.spack/linux/compilers.yaml && \
```

spack-stack fms add-on Dockerfile

```
FROM noaaepic/ubuntu22.04-intel-unified:v1.6.0

WORKDIR /opt/spack-stack/spack-stack-1.6.0

# Create the fms-2024.01 env
RUN . ./setup.sh && \
    #spack compiler rm gcc@11.4.0 && \
    export SPACK_SYSTEM_CONFIG_PATH="/opt/spack-stack/spack-stack-1.6.0/envs/fms-2024.01/site" && \
    spack stack create env --name fms-2024.01 --template empty --site ubuntu-intel --upstream /opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install && \
   # spack compiler add oneapi && spack compiler list && \
   # sed -i 's/\=2021/2021/g' /root/.spack/linux/compilers.yaml && \
    sed -i '20 i \    version("2024.01", sha256="29ac23a5a2a4765ae66d218bb261cb04f7ad44618205ab0924c4e66c9ef8fa38")' spack/var/spack/repos/builtin/packages/fms/package.py && \
    sed -i '28 i \    version("3.5.1", sha256="a9acdb5d23eca532838f21c4a917727ac85851fc9e1f100d65a6f27c1a563998")' spack/var/spack/repos/builtin/packages/g2/package.py && \
    sed -i '24 i \    version("1.13.0", sha256="7e52cccc91277bcedbd9e13ee3478480e744eb22d13c5b636bd0ad91bf43d38e")' spack/var/spack/repos/builtin/packages/g2tmpl/package.py && \
    sed -i '34 i \    depends_on("g2c@1.7:", when="^g2@3.5:")' spack/var/spack/repos/builtin/packages/grib-util/package.py && \
    cd /opt/spack-stack/spack-stack-1.6.0/envs/fms-2024.01 && \
    spack env activate . && \
    spack add upp-env %intel ^g2tmpl@1.13.0 ^g2@3.5.1 && \
    spack add ip %intel && \
    spack add nco@5.0.6 %intel && \
```

jedi-bundle & land-DA_workflow Dockerfile

```
FROM noaaepic/ubuntu22.04-intel-unified:v1.6.0-fms202401

WORKDIR /opt
#clone the jedi bundle and checkout the appropriate release
RUN git clone -b sync_gdas https://github.com/NOAA-EPIC/jedi-bundle.git && mkdir jedi-bundle/build
RUN cd /opt/spack-stack && rm -rf spack/var/spack/cache
WORKDIR /opt/jedi-bundle
#delete a few unnecessary packages
RUN source /opt/spack-stack/spack-stack-1.6.0/.bashenv && \
    cd build && ecbuild -DCMAKE_INSTALL_PREFIX=../install .. && \
    make -j 4 && make install
RUN rm -rf /opt/jedi-bundle/build
WORKDIR /opt
#clone and build the land-DA workflow
RUN git clone -b cadre25 --recursive https://github.com/ufs-community/land-DA_workflow.git
#RUN git clone -b da-consortium-container --recursive https://github.com/EdwardSnyder-NOAA/land-DA_workflow.git
COPY build_singularity_intel.lua /opt/land-DA_workflow/modulefiles
RUN chmod a+rx /opt/land-DA_workflow/modulefiles/build_singularity_intel.lua
RUN sed -i -e $'$a\\\n  - numpy=2.1.1' $PWD/land-DA_workflow/parm/conda_environment.yml && sed -i -e $'$a\\\n  - matplotlib=3.9.2' $PWD/land-DA_workflow/parm/conda_environment.yml
RUN sed -i -e $'$a\\\n  - scipy=1.14.1' $PWD/land-DA_workflow/parm/conda_environment.yml && sed -i -e $'$a\\\n  - cartopy=0.23.0' $PWD/land-DA_workflow/parm/conda_environment.yml
RUN sed -i -e $'$a\\\n  - xarray=2024.9.0' $PWD/land-DA_workflow/parm/conda_environment.yml && sed -i -e $'$a\\\n  - netcdf4=1.7.1' $PWD/land-DA_workflow/parm/conda_environment.yml
```

jedi-bundle & land-DA_workflow Dockerfile

```
RUN  source /opt/spack-stack/spack-stack-1.6.0/envs/fms-2024.01/.bashenv-fms  && \
    # module load bacio cmake crtm ecbuild esmf fms gftl-shared g2 g2tmpl hdf5 ip jasper libpng mapl netcdf-c netcdf-fortran parallelio && \
     #module load prod_util py-netcdf4 py-numpy py-pyyaml py-xarray sp ufs-pyenv w3emc zlib scotch && \
     cd land-DA_workflow/sorc && \
     export os=$(uname) && \
     export hardware=$(uname -m) && \
     export installer=Miniforge3-${os}-${hardware}.sh && \
     curl -L -O "https://github.com/conda-forge/miniforge/releases/download/23.3.1-1/${installer}" && \
     bash ./${installer} -bfp conda && \
     rm ${installer} && \
     source conda/etc/profile.d/conda.sh && \
     conda activate && mamba env create -n land_da --file /opt/land-DA_workflow/parm/conda_environment.yml && \
     git clone https://github.com/NOAA-EMC/jcb && \
     cd jcb && \
     /opt/land-DA_workflow/sorc/conda/envs/land_da/bin/pip install . && \
     cd ../.. && \
     module use /opt/land-DA_workflow/modulefiles && \
     module load build_singularity_intel && \
     mkdir build && cd build && \
     ecbuild -DCMAKE_INSTALL_PREFIX=../install ../sorc -DAPP=LND -DCCPP_SUITES="" && make VERBOSE=1 -j 8 && \
     cp ufs_model.fd/src/ufs_model.fd-build/ufs_model ../install/bin/ && \
     #cd .. && rm -rf build && mkdir -p sorc/build/bin && mkdir exec
     cd ../sorc && mkdir build-atml && cd build-atml && \
     ecbuild -DCMAKE_INSTALL_PREFIX=. ../ -DAPP=ATML && make VERBOSE=1 -j 8 && \
     cd ../../ && mkdir exec
COPY run_container_executable.sh /opt/land-DA_workflow/parm
```

Questions?

- Docs
  - spack-stack
  - jedi-bundle
  - land-DA_workflow
  - Docker
  - Apptainer
- Repos
  - spack-stack
  - jedi-bundle
  - land-da_workflow
  - ufs-containers
  - dockerhub

Common Apptainer (Singularity) Commands

Build Apptainer SIF File

singularity build --force ubuntu22.04-intel-landda-cadre25.img
docker://noaaepic/ubuntu22.04-intel21.10-landda:ue160-fms202401-cadre25

Singularity sandbox

singularity build --sandbox ubuntu22.04-intel-landda-cadre25
ubuntu22.04-intel-landda-cadre25.img

Common Apptainer (Singularity) Commands (cont.)

Singularity shell cmd

singularity shell -B /home ubuntu22.04-intel-landda-cadre25.img

```
ubuntu@ip-10-29-93-209:~$ pwd
/home/ubuntu
ubuntu@ip-10-29-93-209:~$ singularity shell -B /home /home/ubuntu/ubuntu22.04-intel-landda-cadre25.img
Singularity> pwd
/home/ubuntu
Singularity> ls
Land-DA_v2.1_inputs.tar.gz  inputs  rocoto  ubuntu22.04-intel-landda-cadre25.img
Singularity> ls /opt/
intel  jedi-bundle  land-DA_workflow  spack-stack
Singularity> exit
exit
ubuntu@ip-10-29-93-209:~$
```

Common Apptainer (Singularity) Commands (cont.)

Singularity execute cmd

- singularity exec -H $PWD ubuntu22.04-intel-landda-cadre25.img /usr/bin/which python

```
ubuntu@ip-10-29-93-178:~$ which python
/opt/parallelcluster/pyenv/versions/3.9.20/envs/awsbatch_virtualenv/bin/python
ubuntu@ip-10-29-93-178:~$ singularity exec -H $PWD ubuntu22.04-intel-landda-cadre25.img /usr/bin/which python
/opt/land-DA_workflow/sorc/conda/envs/land_da/bin/python
```

Common Apptainer (Singularity) Commands (cont.)

Singularity execute cmd

- singularity exec -H $PWD ubuntu22.04-intel-landda-cadre25.img sh -c ‘echo $PATH’

```
ubuntu@ip-10-29-93-178:~$ echo $PATH
/opt/amazon/openmpi/bin:/opt/amazon/efa/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/aws/bin:/opt/parallelcluster/pyenv/versions/3.9.20/envs/awsbatch_virtualenv/bin:/opt/slurm/bin:/home/ubuntu/rocoto/bin
ubuntu@ip-10-29-93-178:~$ singularity exec -H $PWD ubuntu22.04-intel-landda-cadre25.img sh -c  'echo $PATH'
/opt/intel/oneapi/compiler/2024.0/bin:/opt/intel/oneapi/compiler/2023.2.3/linux/bin/intel64:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/py-xlsxwriter-3.1.7-oa7jczb/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/py-xlrd-2.0.1-jyfdepy/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/py-numpy-1.22.3-eh5axr4/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/openblas-0.3.24-aikecg4/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/py-netcdf4-1.5.8-jx4ron5/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/py-f90nml-1.4.3-2apzn7c/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/py-cython-0.29.36-wlq7bsf/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/nccmp-1.9.0.1-pkfbsfh/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/nemsio-2.5.4-blen2v4/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/scotch-7.0.4-iphwv6d/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/mapl-2.40.3-gkwr2pv/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/esmf-8.6.0-p4rin2p/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/parallel-netcdf-1.12.2-k22zinh/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/netcdf-fortran-4.6.1-yls3min/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/netcdf-c-4.9.2-bszu7f6/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/curl-8.4.0-qzsbdvy/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/nghttp2-1.57.0-tucncxq/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/zstd-1.5.2-gteuzsa/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/hdf5-1.14.0-hysklr6/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/pkg-config-0.29.2-56zcxb7/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/libpng-1.6.37-pmrvxkg/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/jasper-2.0.32-diukxzw/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/libjpeg-turbo-2.1.0-nguovr5/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/ecbuild-3.7.2-uq373s5/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/intel-oneapi-mpi-2021.9.0-6bnjcwc/mpi/2021.9.0/libfabric/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/intel-oneapi-mpi-2021.9.0-6bnjcwc/mpi/2021.9.0/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/intel-oneapi-mpi-2021.9.0-6bnjcwc/bin:/opt/spack-stack/spack-stack-1.6.0/envs/unified-env/install/intel/2021.10.0/prod-util-2.1.1-rfbi5bq/bin:/opt/land-DA_workflow/sorc/conda/envs/land_da/bin:/opt/jedi-bundle/install/bin:/opt/land-DA_workflow/install/bin:/opt/intel/oneapi/vtune/2023.1.0/bin64:/opt/intel/oneapi/mpi/2021.9.0/libfabric/bin:/opt/intel/oneapi/mpi/2021.9.0/bin:/opt/intel/oneapi/mkl/2023.1.0/bin/intel64:/opt/intel/oneapi/itac/2021.9.0/bin:/opt/intel/oneapi/inspector/2023.1.0/bin64:/opt/intel/oneapi/dev-utilities/2021.9.0/bin:/opt/intel/oneapi/debugger/2023.1.0/gdb/intel64/bin:/opt/intel/oneapi/compiler/2023.1.0/linux/lib/oclfpga/bin:/opt/intel/oneapi/compiler/2023.1.0/linux/bin/intel64:/opt/intel/oneapi/compiler/2023.1.0/linux/bin:/opt/intel/oneapi/clck/2021.7.3/bin/intel64:/opt/intel/oneapi/advisor/2023.1.0/bin64:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local
```

Common Apptainer (Singularity) Commands (cont.)

Singularity execute cmd

- singularity exec -H $PWD ubuntu22.04-intel-landda-cadre25.img cp -r /opt/land-DA_workflow/setup_container.sh .

NOTE: ensure you are in your home directory first before running it (/home/ubuntu)

You should see *setup_container.sh* in your working directory now.

```
ubuntu@ip-10-29-93-178:~$ singularity exec -H $PWD ubuntu22.04-intel-landda-cadre25.img cp -r /opt/land-DA_workflow/setup_container.sh .
ubuntu@ip-10-29-93-178:~$ ls
Land-DA_v2.1_inputs.tar.gz  inputs  rocoto  setup_container.sh  ubuntu22.04-intel-landda-cadre25.img
```

setup_container.sh

- Script that sets up the land-DA-workflow to run with the container
- What does it do?
  - Copies out the land-DA_workflow and jedi-bundle
  - Links the input data to the land-DA_workflow/fix
  - Update singularity modulefiles, conda, and exlandda and experiment related scripts
  - Create executable wrappers that point to the container
    - land-DA_workflow/exec

setup_container.sh (cont.)

```
ubuntu@ip-10-29-93-226:~$ ls -l land-DA_workflow/exec/
total 0
lrwxrwxrwx 1 ubuntu ubuntu 35 May 19 18:12 apply_incr.exe -> ../parm/run_container_executable.sh
lrwxrwxrwx 1 ubuntu ubuntu 35 May 19 18:12 calcfIMS.exe -> ../parm/run_container_executable.sh
lrwxrwxrwx 1 ubuntu ubuntu 35 May 19 18:12 chgres_cube -> ../parm/run_container_executable.sh
lrwxrwxrwx 1 ubuntu ubuntu 35 May 19 18:12 err_chk -> ../parm/run_container_executable.sh
lrwxrwxrwx 1 ubuntu ubuntu 35 May 19 18:12 fv3jedi_letkf.x -> ../parm/run_container_executable.sh
lrwxrwxrwx 1 ubuntu ubuntu 35 May 19 18:12 fv3jedi_var.x -> ../parm/run_container_executable.sh
lrwxrwxrwx 1 ubuntu ubuntu 35 May 19 18:12 ndate -> ../parm/run_container_executable.sh
lrwxrwxrwx 1 ubuntu ubuntu 35 May 19 18:12 prep_step -> ../parm/run_container_executable.sh
lrwxrwxrwx 1 ubuntu ubuntu 35 May 19 18:12 python -> ../parm/run_container_executable.sh
lrwxrwxrwx 1 ubuntu ubuntu 35 May 19 18:12 setpdy.sh -> ../parm/run_container_executable.sh
lrwxrwxrwx 1 ubuntu ubuntu 35 May 19 18:12 tile2tile_converter.exe -> ../parm/run_container_executable.sh
lrwxrwxrwx 1 ubuntu ubuntu 35 May 19 18:12 ufs_model -> ../parm/run_container_executable.sh
```

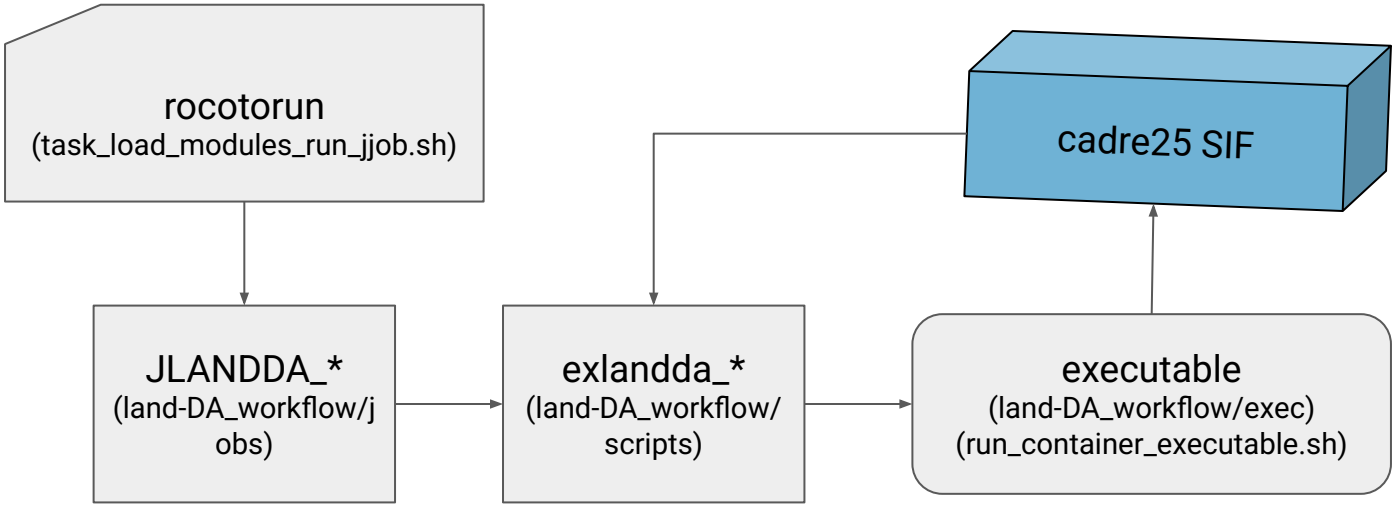
run_container_executable.sh

- Wrapper script to run executables inside of the container
- Utilizes singularity exec command
- Has other singularity runtime variables

singularity exec -B /home ubuntu22.04-intel-landda-cadre25.img ufs_model

```
# Uncomment the line below when running the ATML experiment
#export SINGULARITYENV_PREPEND_PATH=SINGULARITY_WORKING_DIR/land-DA_workflow/sorc/build/bin:$SINGULARITYENV_PREPEND_PATH
${SINGULARITYBIN} exec -B $BINDDIR:$BINDDIR -B $CONTAINERBASE:$CONTAINERBASE $INPUTBIND $img $cmd $arg
```

Running Land-DA_workflow with container



Setting up the container

- Ensure you are in /home/ubuntu
  - exit out of container if you haven't already
  - cd ~
- export LANDDA_INPUTS=/home/ubuntu/inputs
- export img=/home/ubuntu/ubuntu22.04-intel-landda-cadre25.img
- singularity exec -H $PWD $img cp -r /opt/land-DA_workflow/setup_container.sh .
- ./setup_container.sh -c=intelmpi/2021.13 -m=intelmpi/2021.13 -i=$img

Setting up the container (output)

- Home directory after running the setup_container.sh script

```
ubuntu@ip-10-29-93-178:~$ export LANDDA_INPUTS=/home/ubuntu/inputs
ubuntu@ip-10-29-93-178:~$ export img=/home/ubuntu/ubuntu22.04-intel-landda-cadre25.img
ubuntu@ip-10-29-93-178:~$ singularity exec -H $PWD ubuntu22.04-intel-landda-cadre25.img cp -r /opt/land-DA_workflow/setup_container.sh .
ubuntu@ip-10-29-93-178:~$ ./setup_container.sh -c=intelmpi/2021.13 -m=intelmpi/2021.13 -i=$img
Copying out land-DA_workflow from container
Checking if LANDDA_INPUTS variable exists and linking to land-DA_workflow
Land DA data exists, creating links
Updating scripts files
Updating singularity modulefiles
Updating run related scripts
Setup conda
Getting the jedi test data from container
Update experiment variables
Creating links for exe
Done
ubuntu@ip-10-29-93-178:~$ ls
Land-DA_v2.1_inputs.tar.gz  inputs  jedi-bundle  land-DA_workflow  rocoto  setup_container.sh  ubuntu22.04-intel-landda-cadre25.img
```

Setting up the container

- Initial home directory on AMI

```
ubuntu@ip-10-29-93-226:~$ pwd
/home/ubuntu
ubuntu@ip-10-29-93-226:~$ ls
Land-DA_v2.1_inputs.tar.gz  inputs  rocoto  ubuntu22.04-intel-landda-daconsortium.img
```