

Containerization of simulation applications for frequently re-run configurations

SIParCS project

Si Chen

Emory University, PhD student

Mentors: Haiying Xu, Jian Sun, Sheri Mickelson

August 1, 2023



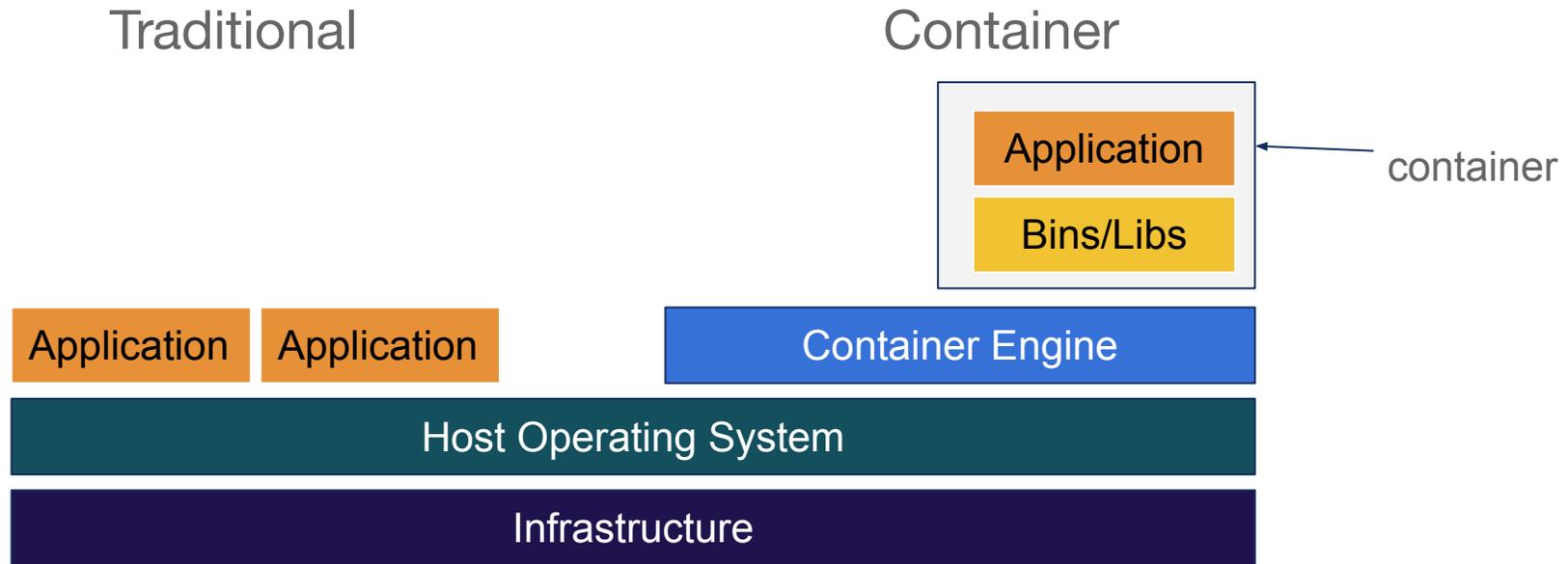
Project Goal: **Reproducibility** and **Portability**

Containerization is a software deployment process that bundles an application's code with all the files and libraries it needs to run on any infrastructure.

- What is a container?
- How to build containers?
- How to use containers in HPC?
- How to help applications development? (CM1)

What is a container?

Container Concept



Two main concepts



- **Image:** the **static** file that includes executable code, representation of the environment.
- **Runtime:** the **dynamic** instance where the image is executed.

Docker is the most widely used container platform

How to build containers?

How to build docker images from Dockerfile



Dockerfile
(recipe)

```
FROM centos:7.8.2003

WORKDIR /home

COPY entrypoints.sh /home/entrypoints.sh

RUN yum update -y && yum install -y which && \
  yum install -y centos-release-scl

RUN git clone -c feature.manyFiles=true
https://github.com/spack/spack.git /home/spack && \
./home/spack/share/spack/setup-env.sh && \
CXX=g++ && CC=gcc && FC=gfortran && \
spack install nvhpc@22.2 %gcc@9.3.1

ENV PATH="/home/spack/bin:$PATH"

ENTRYPOINT ["/entrypoint.sh"]
```



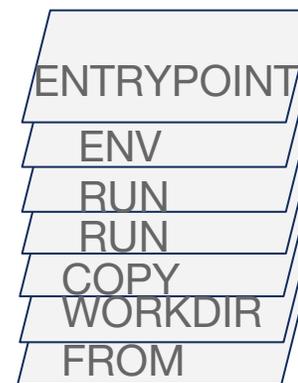
Build

base_image
work directory
local files to image
command for **install packages**
and libraries
environment
script run at beginning



Docker image

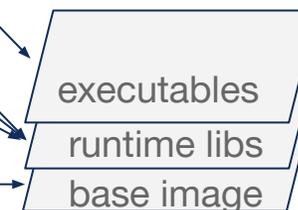
70+GB



Cached layers

Developer

400+MB



Lightweight

Shrink the size: **multi-stage**

How to build/ship/run docker container

- Build image:

```
$ docker build -t meditates/cm1:spack .
```

Docker Hub
account

tag

repository
name

Local

- Run container:

```
$ docker run -it --name u22 meditates/cm1:spack
```



- Push image to repository:

```
$ docker push meditates/cm1:spack
```

meditates / cm1

Description

This repository does not have a description

Last pushed: a day ago

Tags

This repository contains 10 tag(s).

Tag	OS	Type	Pulled	Pushed
spack_nvnhpc_gpu		Image	a day ago	a day ago
spack_3		Image	4 days ago	4 days ago
spack_nvnhpc		Image	6 days ago	6 days ago
spack_intel		Image	8 days ago	8 days ago

How to Install Packages

Modern scientific codes rely on icebergs of dependency libraries.
Multiple versions, configurations, platforms, and compilers...

How can all of these coexist on the same machine?



Spack is a package manager.

- Custom **versions & configurations**
- Customize **dependencies**
- Packages can peacefully coexist
- Group packages with **Environment**

Spack usage

Install (uninstall/load) packages and compilers

`$ spack install openmpi@4.1.5 %nvhpc@23.1 +cuda ^ucx +cuda`

Annotations for the command above:

- package**: points to `openmpi`
- version**: points to `@4.1.5`
- compiler**: points to `%nvhpc@23.1`
- enable variants**: points to `+cuda`
- customized dependencies**: points to `^ucx +cuda`

For compilers:

- `spack compiler add `spack location -i intel-oneapi-compilers@2021.4.0`
/compiler/latest/linux/bin/intel64`
- `spack compiler add `spack location -i intel-oneapi-compilers@2021.4.0`
/compiler/latest/linux/bin`

target: [x86_64_v3]

How to use containers in HPC?

Singularity Container

Singularity was specifically for HPC environment

- **Security.** Using singularity containers does not need sudo rights
- Convenience. Singularity containers are handled like regular files and directories

Building images

- **Building from external repositories** (e.g.docker hub)
- Building from a definition file (need sudo)
- Building from an existing image

Two different image formats

- A Singularity Image Format (SIF) file
- Sandbox directory format

Process on HPC cluster

- Build image: cm1_spack.sif

build from
docker image

```
$ singularity build --fix-perms cm1_spack.sif docker://meditates/cm1:spack
```

give the correct permission
to delete a folder on Casper

- Build application CM1

enable nvhpc

mount the host files into
the container

```
$ singularity exec --nv --bind /glade:/glade /glade/scratch/sichen/sing/cm1_spack.sif  
/glade/scratch/sichen/sing/intel-openmpi.sh
```

spack load
packages

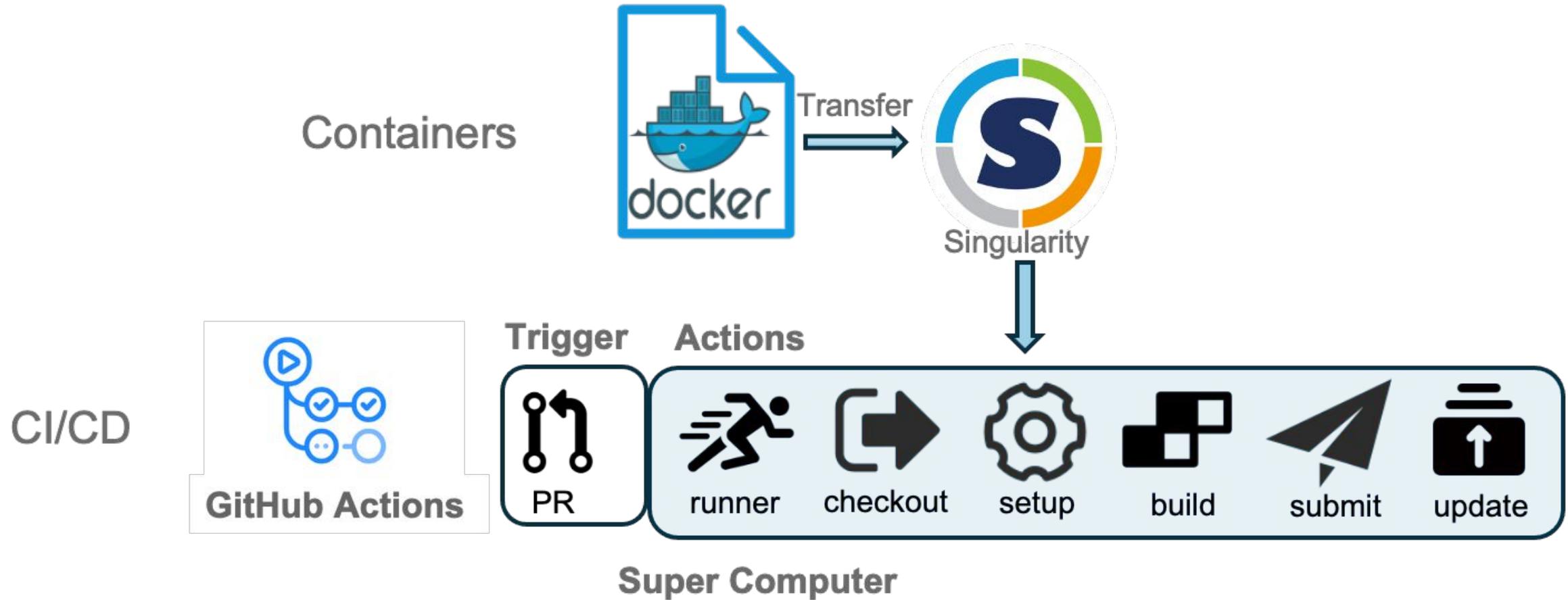
- Submit CM1 job

interactive job
submission

```
$ qcmd -q casper -l select=1:ncpus=2:mpiprocs=2 -A <myaccount> -l  
walltime=3:00:00 -- "/glade/scratch/sichen/sing/submit_intel.sh"
```

How to help applications development?

Validation Workflow



Check the Capabilities of Containers

Triggered via push 3 days ago

 meditates pushed  e5b8a8d `gpu-opt`

Summary

Jobs

- ✓ test_without_contai...
- ✓ test_nvhpc_container
- ✓ test_intel_container

Artifacts

—

actions.yml

on: push

✓ test_nvhpc_container 9m 8s

✓ test_intel_container 9m 12s

✓ test_without_container 8m 48s

cuda@11.6
openmpi@2021.4.0
nvhpc@23.1

openmpi@2021.4.0
oneapi@4.1.5

modules on Casper
(bare-metal)

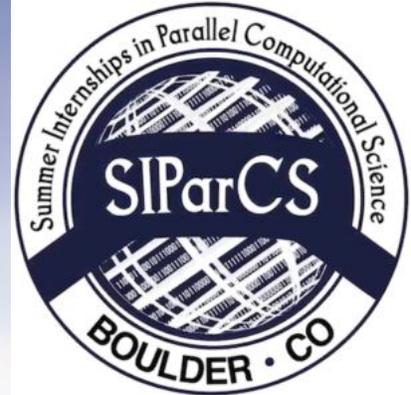
Next Step

For the next stage, my planned work includes:

- Build more general containers that can run both on Casper and Derecho
- Experiment with multiple mpich and nvhpc versions on containers
- Validate and compare the result of different containers

Lessons Learned

- Build multi-stage docker container
- Use Spack to install packages
- Build singularity container and run it on Casper
- Set environment variables and configurations in container
- Apply the GitHub Action CI/CD workflow
- Debug the install error and compiling error



Many thanks to

Mentors: Haiying Xu, Jian Sun, Sheri Mickelson

Technical Support: Brian Vanderwende, ASAP team

SIParCS organizers: Virginia Do, Julius Owusu Afriyie, Ben Fellman

