KONTEJNERY - CONTAINERS
NVIDIA NGC
PODPORA AI – SUPPORT OF AI

Jan Hoidekr

CESNET

duben 2023
Praha

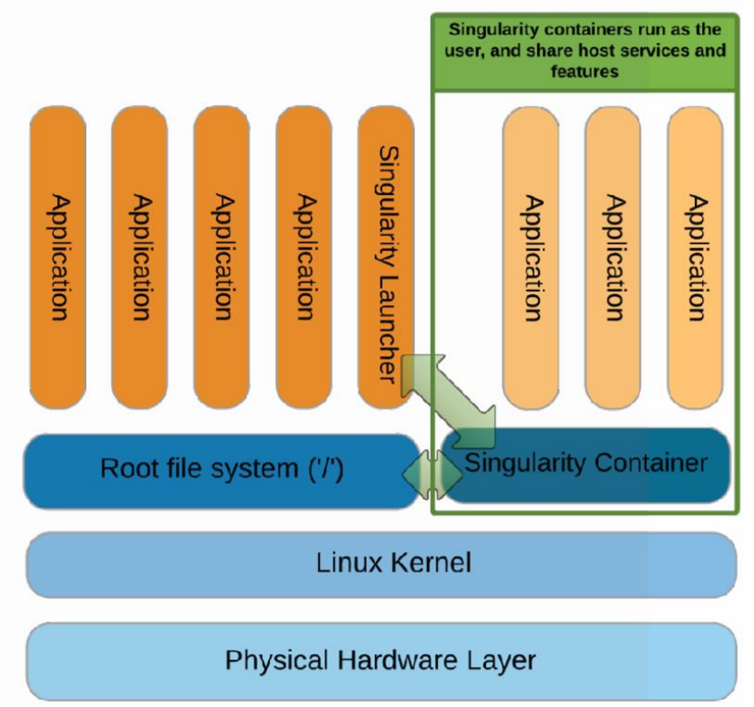
CONTAINERS

Containers

- Virtualization of Applications
- Image - Container
- Docker, Singularity, Podman, Apptainer, …

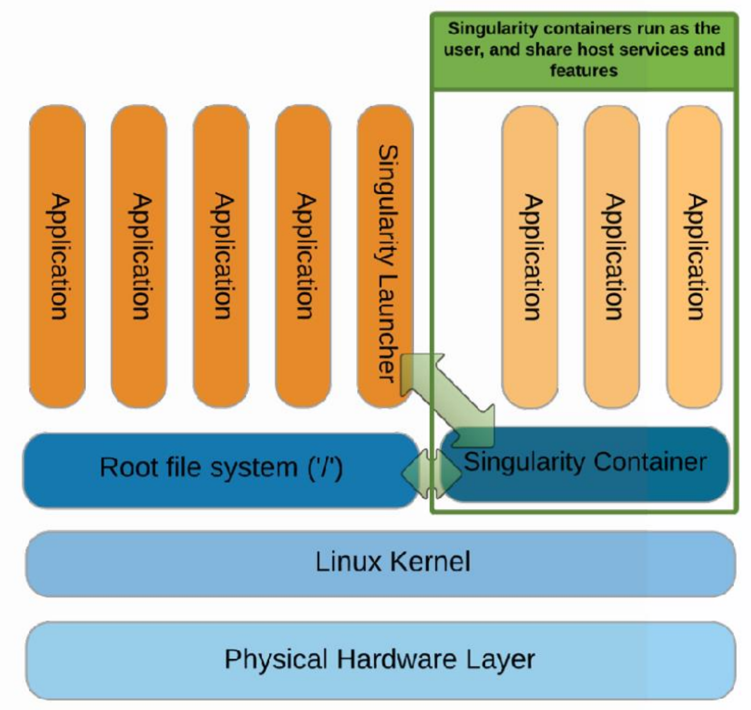
Why to use containers?

- Own environment – bins + libs
- Reproducibility
  - easy to share environment



Infrastructure for containers

=



Infrastructure for containers

CONTAINERS

SingularityCE in Metacentrum

- open source code from Sylabs Inc., BSD licence
- designed for HPC
- focused on OCI compatibility
- v3.11 allows image builds on all nodes with small limitation
  - builder.metacentrum.cz with userns
- back in Debian repositories (unstable)

Apptainer project

- fork of original Singularity project
- imports more code from SingCE then SingCE from Apptainer
- focused on running with userns with non-setuid execution

SIF - Singularity Image Format

- developed by Sylabs Inc.

Singularity example

- HelpDesk request – „please install sw truvari https://github.com/ACEnglish/truvari “

```
$ git clone https://github.com/ACEnglish/truvari && cd truvari
```

Singularity example

- HelpDesk request – „please install sw truvari https://github.com/ACEnglish/truvari “

```
$ git clone https://github.com/ACEnglish/truvari && cd truvari


$ export SINGULARITY_TMPDIR=$SCRATCHDIR
$ singularity build truvari.sif Singularity.def
… INFO:    Build complete: truvari.sif
```

Singularity example

- HelpDesk request – „please install sw truvari https://github.com/ACEnglish/truvari “

```
$ git clone https://github.com/ACEnglish/truvari && cd truvari


$ export SINGULARITY_TMPDIR=$SCRATCHDIR
$ singularity build truvari.sif Singularity.def
… INFO:    Build complete: truvari.sif


$ singularity run truvari.sif
usage: truvari [-h] CMD ...
Truvari v4.1.0-dev Structural Variant Benchmarking and Annotation
```

Singularity example

- HelpDesk request – „please install sw truvari https://github.com/ACEnglish/truvari “

```
$ git clone https://github.com/ACEnglish/truvari && cd truvari


$ export SINGULARITY_TMPDIR=$SCRATCHDIR
$ singularity build truvari.sif Singularity.def
… INFO:    Build complete: truvari.sif


$ singularity run truvari.sif
usage: truvari [-h] CMD ...
Truvari v4.1.0-dev Structural Variant Benchmarking and Annotation


$ alias truvari=“singularity run /mypath/truvari.sif”
$ truvari [params]
```

Comparison

Dockerfile

```
FROM ubuntu:22.04

ADD . /opt/truvari-source
WORKDIR /opt/truvari-source

RUN apt-get -qq update \
 && DEBIAN_FRONTEND=noninteractive apt-get install -yq
  bcftools curl python3-dev python3-pip samtools tabix \
  vcftools wget  && \
  rm -rf /var/lib/apt/lists/*

RUN python3 -m pip install --upgrade pip && \
    python3 -m pip install setproctitle pylint && \
    python3 -m pip install ./

ENTRYPOINT ["truvari"]
```

Singularity Definition file

```
Bootstrap: docker
From: ubuntu:20.04

%files
. /opt/truvari-source

%post
apt-get -qq update \
 && DEBIAN_FRONTEND=noninteractive apt-get install -yq \
  bcftools curl python3-dev python3-pip samtools tabix \
  vcftools wget  && \
  rm -rf /var/lib/apt/lists/*

python3 -m pip install --upgrade pip && \
    python3 -m pip install setproctitle pylint && \
    python3 -m pip install ./

%runscript
exec truvari "$@"
```

(*) Example of similar parts, not complete files

Singularity commands I

- run
  - executes the runscript inside container, typical for dedicated tools

```
$ singularity run truvari.sif
```

- exec
  - executes command inside container environment, typical for script using tools inside container

```
$ singularity exec pytorch.sif train_model.py
```

- pull
  - Get image from registry into local cache

```
$ singularity pull truvari.sif docker://truvari
```

- cache list / clean
  - location ~/.singularity or SINGULARITY_CACHEDIR

```
$ singularity cache list
$ singularity cache clean
```

Singularity commands II

- instance
  - Running instance in background, similar to docker instances

```
$ singularity instance start mysql.sif mysql
```

- build
  - builds image, more possibilities

```
$ export SINGULARITY_TMPDIR=$SCRATCHDIR

$ singularity build truvari.sif Singularity.def

$ singularity build buster.sif docker://debian:buster

# using sandbox – extracted directories, only builder.metacentrum.cz
$ singularity build -s buster.sbox docker://debian:buster
$ singularity shell -f -w buster.sbox
$ singularity build -f buster.sif buster.sbox
```

Definition files

- recipe for building image
- similar to Dockerfile
  - conversion

```
$ module add spython
$ spython recipe Dockerfile singularity.def
```

```
Bootstrap: docker
From: python:3.12.0a7-bullseye

%files
    ./sources /opt
%environment
    export LISTEN_PORT=12345
%post
    pip3 install numpy
%runscript
    echo "Container was created $NOW"
    echo "Arguments received: $*"
%labels
    Author Jan Hoidekr @ MetaCentrum
```

Build image

- `singularity build image.sif sing.def`
- SINGULARITY_TMPDIR
  SINGULARITY_CACHEDIR
- most **builds from definition files on all nodes**
  - experts can use builder.metacentrum.cz

Running containers

- `singularity run image.sif [parameters]`
- `singularity exec image.sif [script/binary with parameters]`

- Bind directories -B

```
$ singularity exec –B /my_sw:/sw image.sif
```

  - /storage, /home, /scratch* - default binds

- GPU --nv

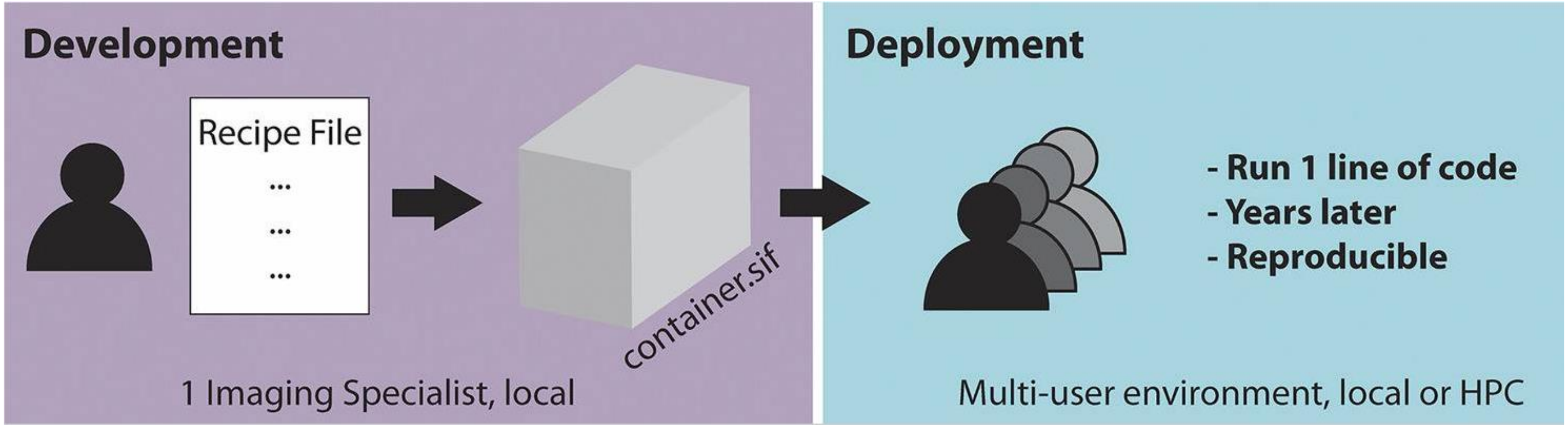
```
$ singularity exec --nv image.sif
```

  - Access to nvidia GPU inside container

- Location of images
  - first run could be delayed [seconds] due to caching
  - no need to copy into $SCRATCHDIR with data

Reproducibility – Singularity definition files

- Singularity development with defintion files
- container.sif - Image with environment

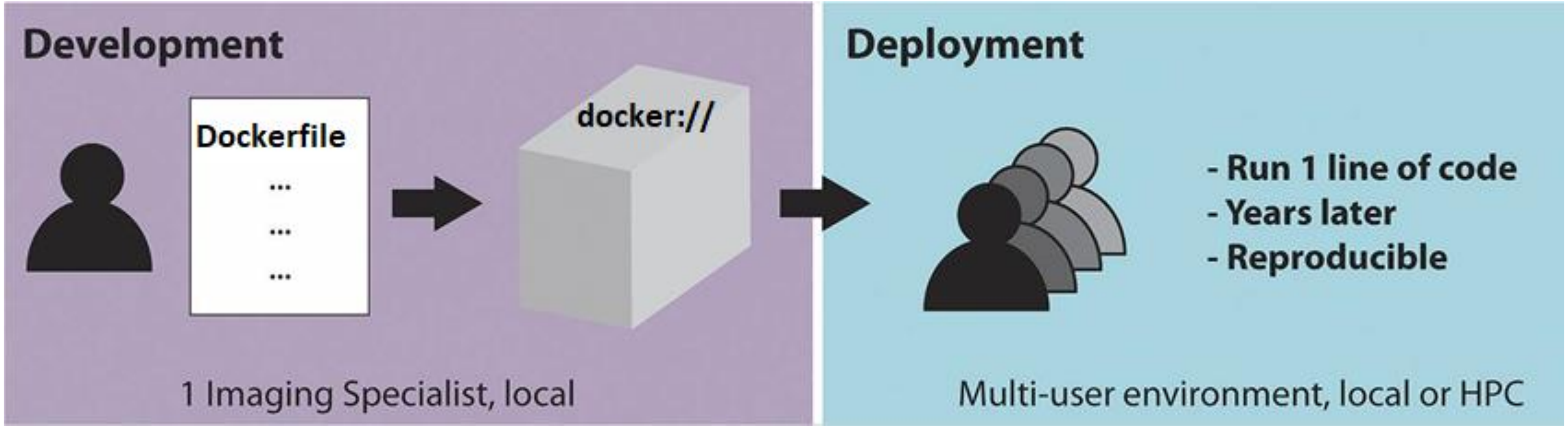


Mitra-Behura, Shilpita et al. "Singularity Containers Improve Reproducibility and Ease of Use in Computational Image Analysis Workflows." *Frontiers in bioinformatics* vol. 1 757291

Reproducibility – Docker registry + singularity in Metacentrum

- Docker development
- same SINGULARITY_CACHEDIR for cached image for ALL users

```
$ singularity run docker://imagename
```



QUESTIONS ABOUT SINGULARITY ?

NVIDIA GPU CLOUD

NVIDIA GPU CLOUD https://catalog.ngc.nvidia.com/

- Prepared docker images for GPU computing
  - TensorFLow, PyTorch, …
- Easy to build customized image

```
$ singularity build my.sif my.def
```

```
Bootstrap: docker
From: nvcr.io/nvidia/pytorch:23.03-py3

%post
    pip3 install ipywidgets
%labels
    customized NGC PyTorch for MetaCentrum seminar
```

NVIDIA GPU CLOUD

NVIDIA GPU CLOUD https://catalog.ngc.nvidia.com/

- Prepared docker images for GPU computing
  - TensorFLow, PyTorch, …

- Easy to build customized image

```
$ singularity build my.sif my.def
```

```
Bootstrap: docker
From: nvcr.io/nvidia/pytorch:23.03-py3

%post
    pip3 install ipywidgets
%labels
    customized NGC PyTorch for MetaCentrum seminar
```

- Use PBS parameters to select GPU
  - **gpu_mem** – minimum of GPU memory
  - **gpu_cap** – minimum allowed capability
  - **cuda_version** – exact version, related with driver version
  - **cluster** – each cluster has one type of GPU

see GPUs at docs.metacentrum.cz

```
$ qsub -l -q gpu select=1:ncpus=4:ngpus=1:gpu_mem=16gb:gpu_cap=75:cuda_version=12.1
```

NVIDIA GPU CLOUD in MetaCentrum

- Pulled images in `/cvmfs/singularity.metacentrum.cz/NGC/`
  - anything missing? Ask meta@cesnet.cz
- Versions 23.xx need CUDA 12 and newer driver version
  - use PBS param `cuda_version=12.1`
- See *Release notes* of NGC images
  - versions
  - known bugs!
- Jupyter notebooks via OnDemand

NVIDIA GPU CLOUD, AI SUPPORT

AI computing

- Start with small jobs – 1node + 1GPU
  - ➢ 1node + multiGPU
    - ➢ multinode + multiGPU

- datasets copy to $SCRATCHDIR
  - big datasets => scratch.shared – cluster galdor

    saves time for copying datasets for every job, but slower then local filesystem

- check usage of GPU
  - tools `nvidia-smi`, `nvtop`
  - ! prevent blocking GPU HW with jobs w/o GPU support

NVIDIA GPU CLOUD, AI SUPPORT

AI computing

- Use of general frameworks – TensorFlow, PyTorch, …
  - start with NGC images - own installation is not recommended
- Jupyter notebooks
  - OnDemand
  - JupyterHub
- Problems? Ask us sooner then later meta@cesnet.cz
  - many faults are known

cesnet

DĚKUJI ZA POZORNOST

MÁTE NĚJAKÉ DOTAZY?