Shrinking the size of container images

[11]

Containers

Containerized Applications

App A	App B	App C	App D	App E	App F

Docker

Host Operating System

Infrastructure

Virtual Machine	Virtual Machine	Virtual Machine
App A	App B	App C
Guest Operating System	Guest Operating System	Guest Operating System

Hypervisor

Infrastructure

[10]
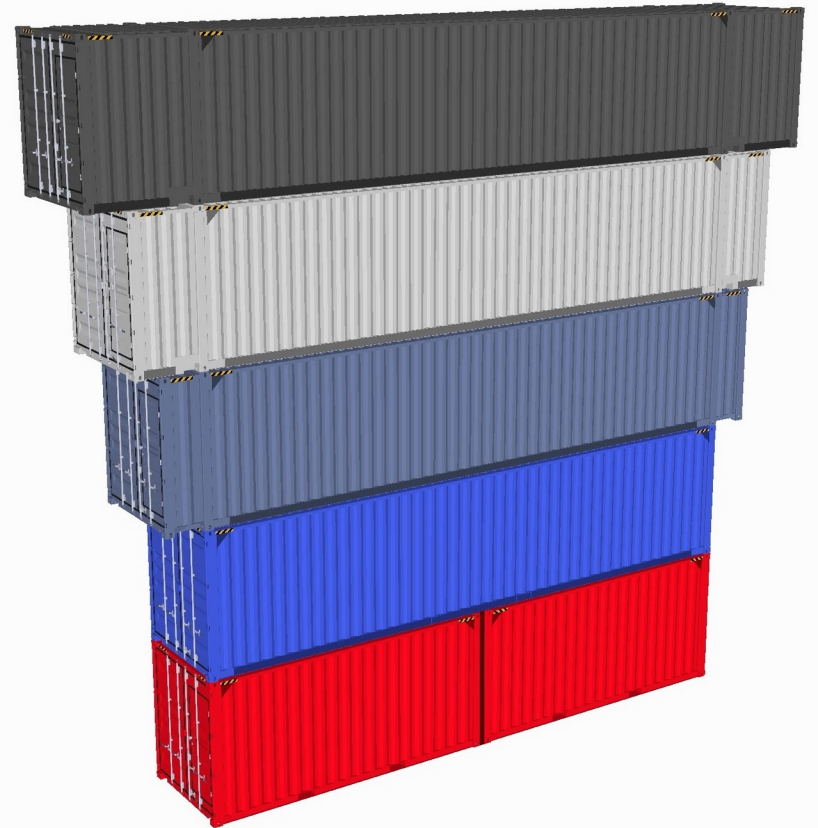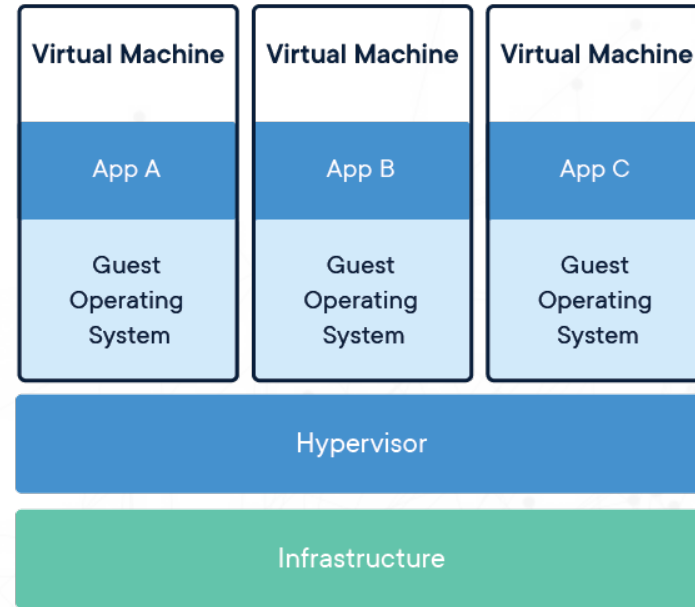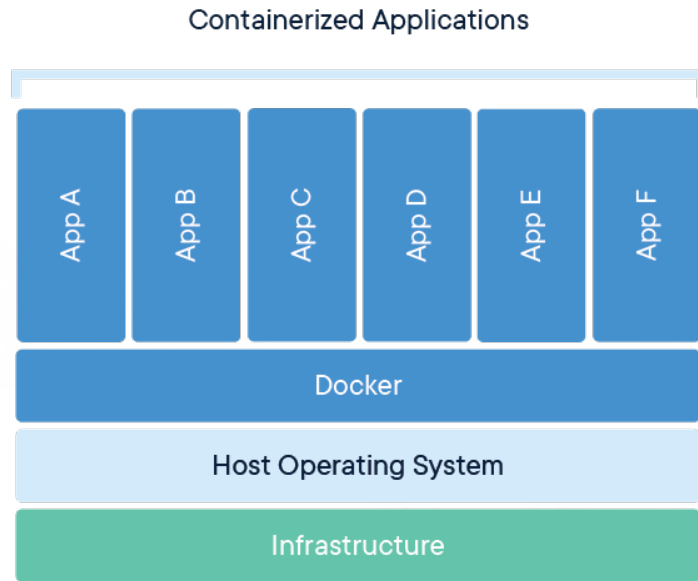
Building containers

- Container definition file

- HPC Container Maker (HPCCM)

  - Generate Dockerfiles or Singularity definition files

  - Uses high level python instructions

  - Python has increased flexibility

  - Applies latest best practices

HPCCM

```python
#!/usr/bin/python3
Stage0 += baseimage(image='docker.io/library/ubuntu:22.04')


# add compiler
Stage0 += gnu()


# add mpi
Stage0 += openmpi(version='4.0.3')
```

```
$ hpccm --recipe hpccm_recipe.py --format docker > Dockerfile.hpccm

FROM docker.io/library/ubuntu:22.04

# GNU compiler
RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        g++ \
        gcc \
        gfortran && \
    rm -rf /var/lib/apt/lists/*


# OpenMPI version 4.0.3
RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        bzip2 \
        file \
        hwloc \
        libnuma-dev \
```
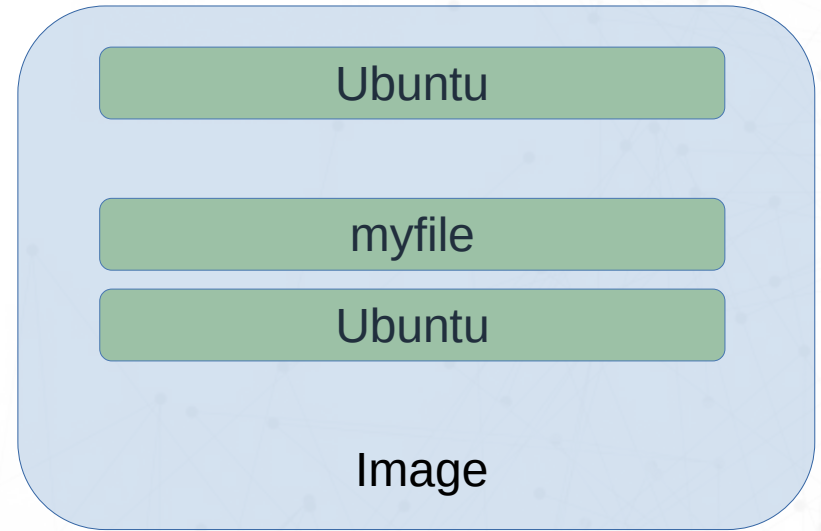
Project

- Multi image build:

  - Base: ubuntu with a spack environment

  - MPI: spack installs gcc, fftw, mpi

  - Gromacs: spack installs gromacs

- Problem is a big final image (4.3 GB)

- Howto reduce the size?

Layers in Docker

```
FROM docker.io/library/ubuntu:22.04

RUN touch myfile
RUN rm myfile
```

Ubuntu
myfile
Ubuntu

Image

- Multi image build:
  - Base: ubuntu with a spack environment
  - MPI: spack installs gcc, fftw, mpi
  - Gromacs: spack installs gromacs

Reduce the image size

- put all install cmd's in one RUN command and clean everything up

- start container from the image, then export and import it

- build the last image with the experimental --squash option

- Using multi stage build

Multi stage build

```
# -----------
#    stage0
# -----------
FROM gwdg/hpc-gromacs-ubuntu-csl:latest AS base

# clean
RUN spack uninstall -y gcc@9.3.0 target=x86_64 && \
    spack gc -y


# -----------
#    stage1
# -----------
FROM FROM docker.io/library/ubuntu:22.04

# copy spack env
COPY --from=base /opt/spack /opt/spack
```

Result

- Shell script which

  - modifies hppc recipes (compiler, target, version, name)

  - Calls hpccm

  - Starts build process

    - Use squash or multi stage build

Image Sources

- [10] https://commons.wikimedia.org/wiki/File:Docker-containerized-and-vm-transparent-bg.png

- [11] https://commons.wikimedia.org/wiki/File:Container_sizes.jpeg