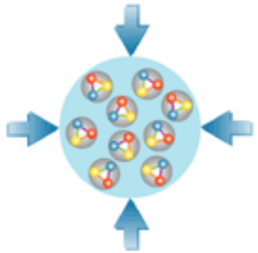
CBM Containers

Dirk Hutter
hutter@compeng.uni-frankfurt.de

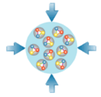
FIAS Frankfurt Institute for Advanced Studies
Goethe-Universität Frankfurt am Main, Germany



Online Meeting

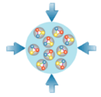
General Nomenclature

- I refer mostly to Docker / Open Container Initiative (OCI) container infrastructure
- (Container-)Image:
  - The **data** needed to start a container, e.g. the os user land and libraries
  - Immutable, shared between containers, stored in a registry and cashed locally.
  - Is layered, different images can share the same (lower) layers
  - Image tag <repository_url>:<version>, e.g., hub.cbm.gsi.de/computing/cbmroot/cbm_online:master
- Container:
  - The **runtime environment** created from an image
  - Adds a mutable but only for the lifetime of the container persistent layer above the image
  - Mounts external storage for persistent data
- (Container-)Registry
  - Central storage/repository for images



How to build a container image

- In Docker:
- Define your image layers in one or multiple docker files
  - Lowest layer is usually a minimal OS image from the OS maintainer
  - Any image can be the starting point for your image
- Build the image with docker build
- Tag and publish the created image
- We do this mainly in CI jobs, but can be done manually
- Good practice:
  - One docker file per image
  - On gitlab: one (or multiple simultaneously versioned) docker image per project/repository

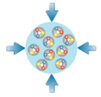


Zoo of Containers

- A Container can serve many purposes and may be adapted to a specific purpose
- We can create multiple „levels“ of containers/images
  - For storage efficiency it is important to get the layers right
- runtime container
  - minimal os + runtime libraries
- build container
  - minimal os + development libraries + dev tools to build (and run) the application
- development container
  - „interactive“ os + development libraries + dev tools + debug tools
- containerized application
  - runtime container + application
- containerized „interactive“ application
  - development container + application
  - or any other combination

How to develop inside a container

- General Idea:
  - Start a container with all needed development tools and libraries
  - Mount a folder with the sources you want to work into the container
  - Start a shell in the container and work
- Depending on the container runtime the exact procedure can be more complex
  - e.g., Docker will not inherit your UIDs from the host
- Do this manually
  - e.g., Sergey cbmdock https://git.cbm.gsi.de/se.gorbunov/cbmtools
- Virgo style
  - Start a container on virgo, needed magic is handled by the virgo environment
  - Caveat: no documentation how to setup slurm in a user supplied container, quiet tricky to reverse engineer.
  - Can be done automatically for each ssh session using RemoteCommand
- Vscode devcontainers plugin
  - https://code.visualstudio.com/docs/devcontainers/containers
  - Manages everything for you,
  - Runs the VScode server inside the container -> plugins see the correct environment
  - Can run in combination with the remote ssh plugin



Current status „Computing Containers“

- https://git.cbm.gsi.de/computing/images/base
  - **Mirror** of external images
- https://git.cbm.gsi.de/computing/images/online
  - **Online containers** build and runtime base images
- https://git.cbm.gsi.de/dockerfiles
  - BaseImages, **FairRoot, FairSoft, CbmRoot**, Flesnet
  - Latex, Podman_test
- https://git.cbm.gsi.de/nora_containers/cbmdev/
  - container for **cbmroot** development (via Vscode dev containers)

„Dockerfiles“ Images

- Debian -> FairSoft -> FairRoot -> CbmRoot
- Debian -> FairSoft -> Flesnet
- BaseImages: just a copy of some singularity containers
- Unmaintained (last commit 2022) but good starting point
- Issues
  - Not clear to me which purpose they target
  - Build flow is outdated
  - Naming is sub-optimal
    - hub.cbm.gsi.de/dockerfiles/fairroot/debian11:jan24p5_v18.8.2
  - Depending on usage and management they should move to the computing group

Why?

- https://git.cbm.gsi.de/CbmSoft/
  - User! owning geometry, parameters and input repositories
  - Is this a shared account? Who controls this user?