HPC Container Conformance
Overview

Christian Kniep, 2023-11-13, SC'23

Christian Kniep
QNIB Solutions
Berlin area, Germany
info@qnib.org

Conformance What!?

What are we trying to achieve?

Guidance!

- Collect ways of building container images for HPC use cases
- Derive best-practices on how to build and annotate a container
- Use best-practices and annotation and take the SystemAdmin perspective

Expectation Management in terms of Portability/Performance

- Container images might be specific to a system or generic; how to we guide folks what to expect?

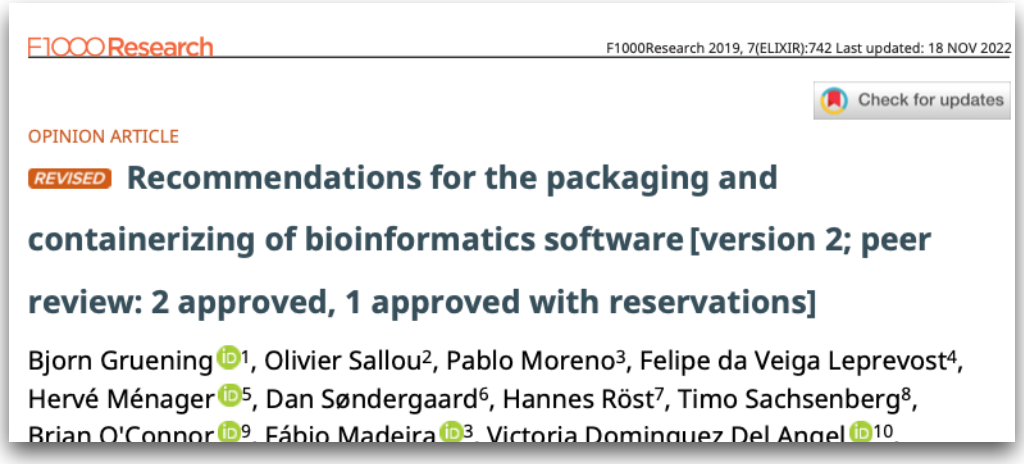
Application we start with: GROMACS, PyTorch (, WRF?)

Biocontainer Paper

Recommendations for the packaging and containerising of bioinformatics software

https://f1000research.com/articles/7-742/v2

1. A package first
2. One tool, one container
3. Tool and container versions should be explicit
4. Avoid using ENTRYPOINT
5. Reduce size as much as possible
6. Keep data outside of the container
7. Add functional testing logic
8. Check the license of the software
9. Make your package discoverable
10. Provide reproducible and documented builds
11. Provide helpful usage message



Expected Image Behaviour

Expected Image Behavior

Login Container vs. Application Container

For HPC containers we expect to be dropped into a shell (most likely bash)

```
docker run -ti -v $(pwd):/data quay.io/cqnib/gromacs-2021.5_gcc-7.3.1:aarch64
bash-4.2#
```

The look and feel should be similar to logging into a compute node. The environment is prepared to have the application already at your fingertips.

```
$ docker run -ti -v $(pwd):/data -w /data \
     quay.io/cqnib/gromacs-2021.5_gcc-7.3.1:aarch64 gmx  mdrun -s benchRIB.tpr -resethway
                    :-) GROMACS - gmx mdrun, 2021.5-spack (-:
Using 1 MPI thread
Using 8 OpenMP threads
```

ENTRYPOINT	CMD
/bin/bash —rcfile /etc/profile -l -c $* --	/bin/bash

Expected Image Behaviour

USER within Container

The container is going to spawn a process under the UID:GID of an beforehand unknown user. This implies the following:

- Make sure that scripts within the container do not use `whoami` or anything that needs a ‘real’ username
- Make sure the container is able to run as `nobody`

Expected Image Behaviour

Help message as a default CMD

Shim ENTRYPOINT to print a help message and `exec bash` afterwards.

```
(base) M1BookPro → motd  docker run -ti --rm  test

  ### GROMACS Container

  This container uses thread MPI; therefore it is not to be used f
  The container defines a scratch volume at  /scratch  which is us

26f1e7f6c961:/# exit
exit
(base) M1BookPro → motd  docker run -ti --rm  test bash
358f511cba86:/#
```

```
1    #!/bin/bash
2    glow /usr/share/hpc3/help.md
3    exec bash
```

```
COPY hpc3-help /usr/local/bin/
COPY help.md /usr/share/hpc3/help.md
CMD ["hpc3-help"]
```

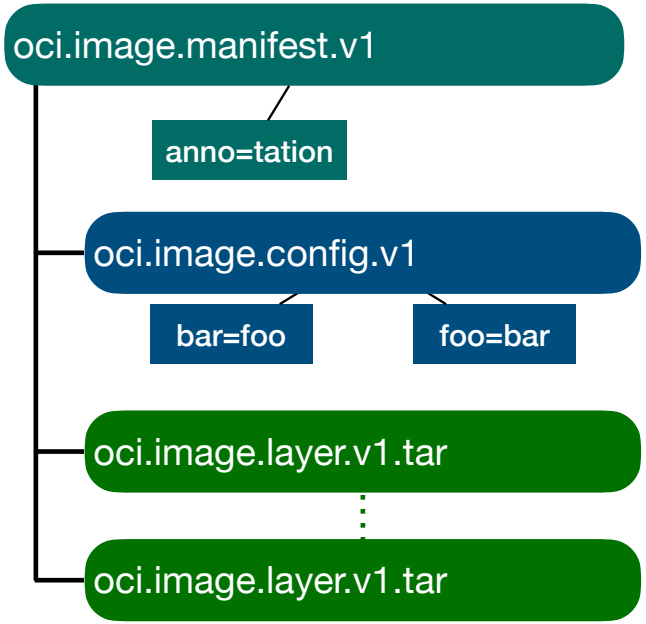
Annotations

What are annotations and labels?

Labels vs. Annotations

```
FROM alpine:3.17.2

LABEL foo=bar
LABEL bar=foo
```



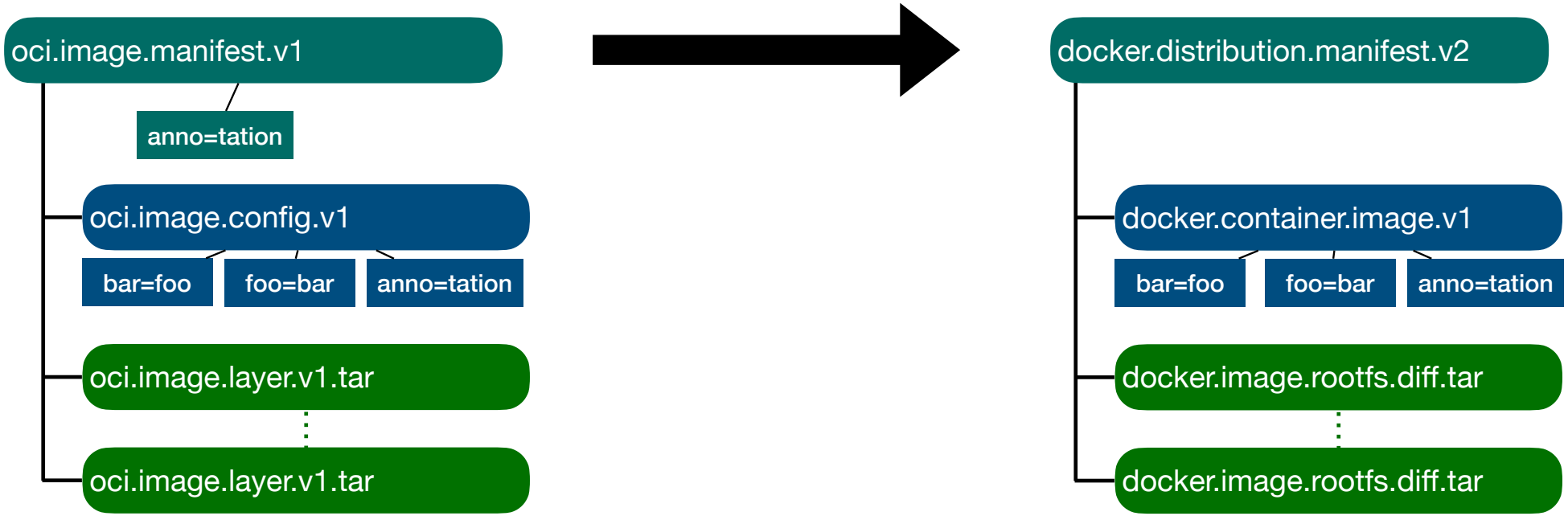

```
$ podman build -q -f Dockerfile --annotation anno=tation -t podman/labels .
cf73eacc27043d167b697dfb5fa84bfee1ca70f102a0b2eedf49ef52458e1044
$ podman save podman/labels --format oci-archive |tar xf - -C oci
$ jq . oci/blobs/sha256/b36b594a83b3d4bd44d94b4a63dd0aa80896ad535ed3883205eb459527ae362a
{
  "schemaVersion": 2,
  "mediaType": "application/vnd.oci.image.manifest.v1+json",
  "config": {
    "mediaType": "application/vnd.oci.image.config.v1+json",
    "digest": "sha256:cf73eacc27043d167b697dfb5fa84bfee1ca70f102a0b2eedf49ef52458e1044",
```

```
$ jq .config.Labels oci/blobs/sha256/cf73eacc27043d167b697dfb5fa84bfee1ca70f102a0b2eedf49ef52458e1044
{
  "bar": "foo",
  "foo": "bar",
  "io.buildah.version": "1.27.0"
}
```

```
    }
  ],
  "annotations": {
    "anno": "tation",
    "org.opencontainers.image.base.digest": "sha256:60cfb06536035a143bbfbac665bae52d493c58b
    "org.opencontainers.image.base.name": "localhost/alpine:3.17"
  }
}
```

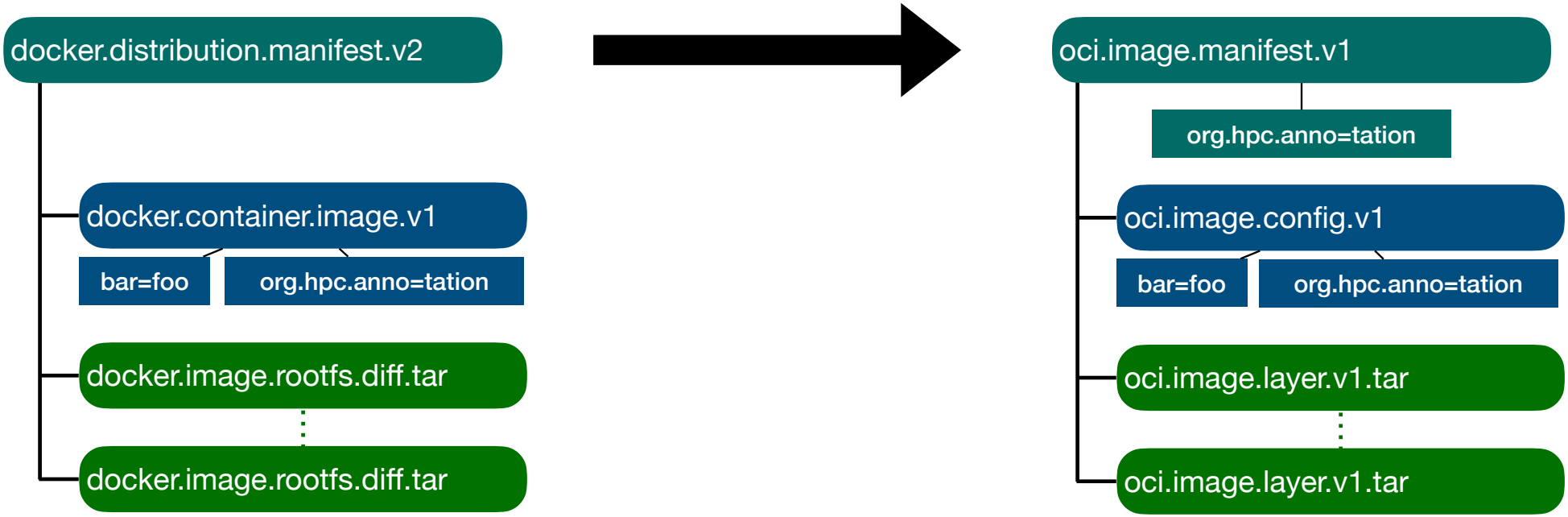
What are annotations and labels?

Labels should be considered ground truth for all key/value pairs



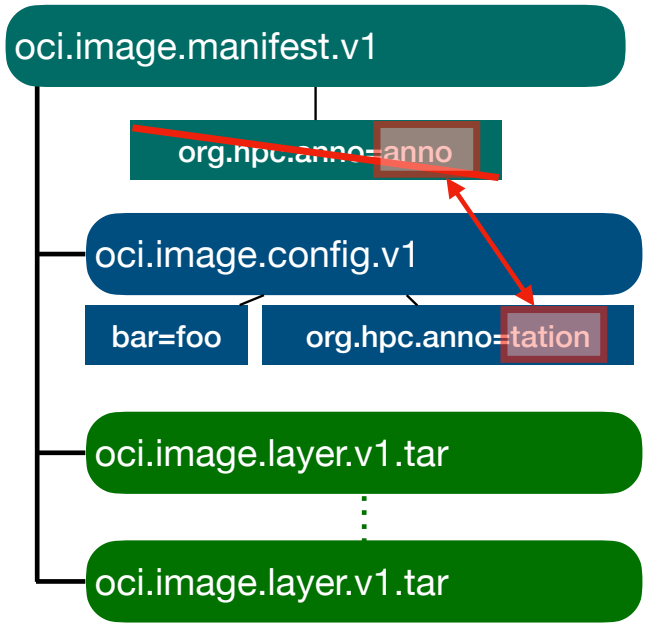
What are annotations and labels?

Labels w/ reverse domain notation might be elevated



Consistency between Labels and Annotations

Labels are ground truth!



Annotation Groups

Hardware Annotations

CPU/GPU/...

Information about what the application in the containers user-land is compiled for.

- Will the application segfault due to architecture mismatch (beyond the platform specification ARM/x86)?
- What CUDA version and GPU architecture is the application build against?

org.supercontainers.hardware.cpu.target	generic / microarch
org.supercontainers.hardware.cpu.target.generic	x86_64_v4
org.supercontainers.hardware.cpu.target.microarch	skylake / skylake_avx512

org.supercontainers.hardware.gpu.vendor	nvidia / and / intel
org.supercontainers.hardware.gpu.nvidia.cuda.version	12.1
org.supercontainers.hardware.gpu.nvidia.architecture	sm_35 (kepler), sm_86 (ampere)

MPI/Interconnect Annotations

Information about what the user-land is compiled for and what methods to tweak the container is the container designed for?

org.supercontainers.mpi.implementation	(openmpi,mpich,threadmpi)
org.supercontainers.mpi.communication.framework	(ucx, libfabrics)
org.supercontainers.mpi.openmpi.version	1.16.1
org.supercontainers.mpi.libfabric.abi.version	1.6
org.supercontainers.mpi.portability.optimization	stock, cray-xc-cnl10
org.supercontainers.mpi.portability.mode	mpi_replace, libfabric_inject, ucx_replace

System Annotations

What can the user expect

Scripting Environment: What does the container carry to support scripts?

org.supercontainers.system.libc.implementation	glibc,musl
org.supercontainers.system.glibc.version	2.35
org.supercontainers.system.python.version	3.10
org.supercontainers.system.shell.implementation	bash,sh,zsh
org.supercontainers.system.tools.includes	jq,wget,awscli
org.supercontainers.system.path.extra	/usr/local/bin (empty dir already in PATH)

System Annotations

What can the user expect

Scripting Environment: What does the container carry to support scripts?

org.supercontainers.system.libc.implementation	glibc,musl
org.supercontainers.system.glibc.version	2.35
org.supercontainers.system.python.version	3.10
org.supercontainers.system.shell.implementation	bash,sh,zsh
org.supercontainers.system.tools.includes	jq,wget,awscli
org.supercontainers.system.path.extra	/usr/local/bin (empty dir already in PATH)

What is expected from the host system

org.supercontainers.host.kernel.version.min	5.1
org.supercontainers.host.kernel.modules.expectation	user-namespaces

Documentation Annotations

Further information

Hello-world example as minimalistic as possible

Benchmark how-to with meaningful, representative result

How to use the container?

org.supercontainers.docs.quickstart.link	https://external.website.org/how-to-gromacs
org.supercontainers.docs.quickstart.base64	base64-encoded-markdown
org.supercontainers.docs.benchmark.link	https://external.website.org/how-to-bench-gromacs
org.supercontainers.docs.benchmark.base64	base64-encoded-markdown

How to reproduce/tweak the container build

org.supercontainers.docs.build.dockerfile	base64-encoded-dockerfile
org.supercontainers.docs.build.spack.env	base64-encoded-spack.env
org.supercontainers.docs.build.quickstart.link	https://external.website.org/how-to-build-gromacs
org.supercontainers.docs.build.quickstart.base64	base64-encoded-markdown

Next

What's next

- **ISC'24: 10th annual High Performance Container Workshop**
  Mark your calendar

- https://github.com/HPC-Container-Conformance/paper-2023
  Put together a paper similar to the Bioinformatic one to solidify the ideas around HPC containers.