CernVM-FS for Containers

Jakob Blomer (CERN)
EGI Webinar, 15 October 2020
CernVM-FS Worldwide Deployment

Mission statement: global delivery of experiment software, platforms, and conditions data
Mission statement: global delivery of experiment software, platforms, and conditions data

> 1B files under management
~ 10 large stratum 1s
~ 400 site caches

CERN stratum 1 fully on Ceph/S3
Mission statement: global delivery of experiment software, platforms, and conditions data

Zoomed into Europe:
## Principal Content Types

<table>
<thead>
<tr>
<th>🔷️ Production Software</th>
<th>🔷️ Unpacked Container Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: <code>/cvmfs/atlas.cern.ch</code></td>
<td>Example: <code>/cvmfs/unpacked.cern.ch</code></td>
</tr>
<tr>
<td>✓ Most mature use case</td>
<td>✓ Enables large scale container deployment</td>
</tr>
<tr>
<td>🌟 Containerizing publish workflows</td>
<td>🌟 Integration for most container runtimes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>🔷️ Integration Builds</th>
<th>🔷️ Auxiliary data sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: <code>/cvmfs/lhcbdev.cern.ch</code></td>
<td>Example: <code>/cvmfs/alice-ocdb.cern.ch</code></td>
</tr>
<tr>
<td>✓ Most challenging application (massive load)</td>
<td>✓ Benefits from internal versioning</td>
</tr>
<tr>
<td>🌟 Continous improvements: parallel publishing, faster GC, shorter propagation</td>
<td>• Depending on volume requires more planning for the CDN components</td>
</tr>
</tbody>
</table>

🌟 Current focus of developments

---

```plaintext
jblomer@cern.ch
CernVM-FS for Containers, EGI Webinar 10/2020
```
## The Grid’s Container Hub

### /cvmfs/unpacked.cern.ch
- > 500 images
- > 2TB
- > 35M files

### /cvmfs/singularity.opensciencegrid.org
- > 500 images
- > 2TB
- > 40M files

A large variety of images readily available to run with singularity on the grid, including **base operating systems**, **experiment software stacks**, **explorative tools (ML etc.)**, **user analyses**, and special-purpose containers such as **folding@home**

```bash
[jblomer@lxplus.cern.ch]$ singularity exec \\
'#/cvmfs/unpacked.cern.ch/registry.hub.docker.com/library/debian:stable' \\
cat /etc/issue
Debian GNU/Linux 10
```

---

CernVM-FS for Containers, EGI Webinar 10/2020
Containers and HEP
### Why Containers

We are aiming for the isolation & orchestration capabilities from standard industry tools with the flexibility and efficiency of CernVM-FS based software distribution.

<table>
<thead>
<tr>
<th>Desired Property</th>
<th>Standard Technology</th>
<th>HEP Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isolation</td>
<td>Job X does not impact job Y on the same node</td>
<td>cgroups, namespaces</td>
</tr>
<tr>
<td>Independence</td>
<td>Application brings all its dependencies</td>
<td>chroot, overlayfs</td>
</tr>
<tr>
<td>Distribution</td>
<td>Make software available on the nodes when needed</td>
<td>tar, container registry</td>
</tr>
<tr>
<td>Orchestration</td>
<td>Schedule distributed jobs on multiple nodes</td>
<td>kubernetes</td>
</tr>
</tbody>
</table>
File System Gymnastics

Image  The root file system of the container, either tarball-packed or exploded in a directory

Layer  A part of an image, e.g. only the files of the base operating system

Overlay  Stack of layers that create a (writable) root file system

Underlay  A writable root file system created from a read-only root file system
Image  The root file system of the container, either tarball-packed or exploded in a directory

Layer  A part of an image, e.g. only the files of the base operating system

Overlay  Stack of layers that create a (writable) root file system

Underlay  A writable root file system created from a read-only root file system
Image  The root file system of the container, either tarball-packed or exploded in a directory

Layer    A part of an image, e.g. only the files of the base operating system

Overlay  Stack of layers that create a (writable) root file system

Underlay A writable root file system created from a read-only root file system
Image  The root file system of the container, either tarball-packed or exploded in a directory

Layer  A part of an image, e.g. only the files of the base operating system

Overlay  Stack of layers that create a (writable) root file system

Underlay  A writable root file system created from a read-only root file system
Distribution of container images sizes in /cvmfs/unpacked.cern.ch and /cvmfs/singularity.opensciencegrid.org

It’s easy to overflow the worker node scratch space with multi-gigabyte images
De-duplication

Comparison of de-duplication efficiency between layers and file-based storage (CernVM-FS)

De-duplication works properly only on file-level granularity

jblomer@cern.ch

CernVM-FS for Containers, EGI Webinar 10/2020
Observations from CERN lxbatch farm:

- Looked at 3 different images (ATLAS and CMS base images)
- Found on > 250 worker nodes
- For each image:
  - 2% to 9% of the image volume in the worker node cache
  - Site-wide: ~15% of the volume cached
- Not yet including de-duplication effects in the worker node cache

→ ×10 to ×50 higher image distribution efficiency from /cvmfs hosted images compared to docker pull ...
Containers and CernVM-FS
Provide /cvmfs in a container

If /cvmfs is available on the host
- Bind mount from host to container

On opportunistic resources:
- Unprivileged mounting inside container
  Uses of user-level fuse mounts (EL >7.8);
  challenge on sharing the cache among containers
- As of CernVM-FS 2.8: “service container”
  CernVM-FS client pre-packaged as a container
- Pre-mounted by singularity

Provide container from /cvmfs

Unpacked images on /cvmfs for scalable distribution

Requires:
1. Container image conversion:
   automated with the CernVM-FS DUCC service
2. Container runtime plug-in required for layered images

Reminder:
- “Flat image”: starts container from unpacked root file system
- “Layered image”: constructs root file system with Overlay-FS from several directories

Use cases ① and ② can be combined
$ docker run -v /cvmfs:/cvmfs:shared busybox ls /cvmfs/sft.cern.ch
README.md lcg

$ singularity exec -B /cvmfs docker://busybox ls /cvmfs/sft.cern.ch
README.md lcg

Important: use *shared* bind mount with docker so that repositories can be mounted on demand from inside the container
Technical foundations

- User namespaces completing container support
- As of Linux kernel version 4.18 (EL8, but also EL 7.8), fuse mounts are unprivileged in user name spaces
- Overlay-FS implementation available as a fuse module

```
$ cvmfsexec grid.cern.ch atlas.cern.ch -- ls /cvmfs
atlas.cern.ch cvmfs-config.cern.ch grid.cern.ch
```
CernVM-FS Service Container

- CernVM-FS client in a minimal container
- Mounted /cvmfs inside container can be “leaked” to the outside host
- Alternative to system package based installation, e.g. on container-only operating systems
- Foundation of the kubernetes daemonset deployment

```bash
$ docker run -d --rm \
  -e CVMFS_CLIENT_PROFILE=single -e CVMFS_REPOSITORIES=sft.cern.ch \ 
  --cap-add SYS_ADMIN --device /dev/fuse \ 
  --volume /cvmfs:/cvmfs:shared
  cvmfs/service

$ ls /cvmfs/
cvmfs-config.cern.ch sft.cern.ch
```
With the new Fuse3 libraries, mounting can be handed off to a trusted, external helper.

- Fuse3 libraries have been backported to EL6 and EL7 platforms.
- Gives access to /cvmfs in containers started by singularity (singularity --fusemount)
- Required cvmfs client to be installed and prepared in the container

```
$ CONFIGREPO=config-osg.opensciencegrid.org
$ mkdir -p $HOME/cvmfs_cache
$ singularity exec -S /var/run/cvmfs -B $HOME/cvmfs_cache:/var/lib/cvmfs 
  --fusemount "container:cvmfs2 $CONFIGREPO /cvmfs/$CONFIGREPO"
  --fusemount "container:cvmfs2 sft.cern.ch /cvmfs/sft.cern.ch"
  docker://davedykstra/cvmfs-fuse3 ls /cvmfs/sft.cern.ch
README.md lcg
```
Container Deployment from CernVM-FS
**Wishlist**

https://gitlab.cern.ch/unpacked/sync

```yaml
version: 1
user: cvmfsunpacker
cvmfs_repo: 'unpacked.cern.ch'
output_format: >
   https://gitlab-registry.cern.ch/unpacked/sync/$(image)
input:
- 'https://registry.hub.docker.com/library/fedora:latest'
- 'https://registry.hub.docker.com/library/debian:stable'
- 'https://registry.hub.docker.com/library/centos:*'
```

Multiple wishlists possible, e.g. experiment specific

```
# Singularity
/registry.hub.docker.com/library/fedora:latest -> \
/unpacked.cern.ch/.flat/d0/d0932...
# containerd, k8s, podman
/.layers/f0/1af7...
```

**Ongoing work on direct registry integration, i.e. docker push triggers image conversion (ETA 2021)**
Container Runtime Integration

```
[jblomer@lxplus.cern.ch]$ singularity exec \\
'/cvmfs/unpacked.cern.ch/registry.hub.docker.com/library/debian:stable' \\
cat /etc/issue
Debian GNU/Linux 10
```

<table>
<thead>
<tr>
<th>Runtime</th>
<th>Type</th>
<th>CernVM-FS Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singularity</td>
<td>flat (+ layers)</td>
<td>native</td>
</tr>
<tr>
<td>runc</td>
<td>flat (+ layers)</td>
<td>native</td>
</tr>
<tr>
<td>docker</td>
<td>layers</td>
<td>&quot;graph driver&quot; image storage plugin</td>
</tr>
<tr>
<td>containerd / k8s</td>
<td>layers</td>
<td>pre-release ▶ remote snapshotter</td>
</tr>
<tr>
<td>podman</td>
<td>layers (+ flat)</td>
<td>pre-release ▶ pull request</td>
</tr>
</tbody>
</table>

Currently improving documentation, examples, integration tests for different deployment options
### Team: CERN & LHC Computing

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date of last work unit</td>
<td>2020-10-13 20:13:49</td>
</tr>
<tr>
<td>Active CPUs within 50 days</td>
<td>418,716</td>
</tr>
<tr>
<td>Team Id</td>
<td>38188</td>
</tr>
<tr>
<td>Grand Score</td>
<td>81,674,915,475</td>
</tr>
<tr>
<td>Work Unit Count</td>
<td>16,082,482</td>
</tr>
<tr>
<td>Team Ranking</td>
<td>17 of 255121</td>
</tr>
<tr>
<td>Fast Teampage URL</td>
<td><a href="https://apps.foldingathome.org/teamstats/team38188.html">https://apps.foldingathome.org/teamstats/team38188.html</a></td>
</tr>
</tbody>
</table>

Runs on the grid off containers served from `/cvmfs`

---

Example for Large-Scale Deployment: Folding@Home

**Runs on the grid off containers served from `/cvmfs`**
Ephemeral Publish Container
A new command, `cvmfs_server enter`, creates a sub-shell with a writable `/cvmfs`.

- Uses internally user namespaces and `fuse-overlayfs`
- Works unprivileged on any modern Linux that can mount the client
- Meant to be used on build nodes
- **Ongoing work, to be released in CernVM-FS 2.8** [pull request](#)

```bash
$ cvmfs_server enter hsf.cvmfs.io
...Opens a shell with write access to /cvmfs/hsf.cvmfs.io
$ cvmfs_server diff --tarball | gzip > changes.tar.gz
...Back to read-only mode
```
Summary
Announcements

EGI Clinic on HT-Condor and CernVM-FS
3 November 2020
https://indico.egi.eu/event/5000/sessions/4508/

CernVM Workshop 2021
1–3 February 2021
https://indico.cern.ch/e/cvm21
A container strategy for the grid:

1. Exploit isolation and orchestration capabilities from standard tools (singularity, k8s, . . .)
2. Exploit distribution efficiency from CernVM-FS

There is a continuous effort from the CernVM-FS team to integrate well with container runtimes

Possible future landscape of containerized software stacks

<table>
<thead>
<tr>
<th>Production workflows (e.g. simulation)</th>
<th>Explorative workflows (e.g. testing ML tools)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full app universe provided by /cvmfs</td>
<td>Full container deployed from /cvmfs</td>
</tr>
<tr>
<td>Minimal Compatibility Container</td>
<td></td>
</tr>
<tr>
<td>glibc + $\varepsilon$</td>
<td></td>
</tr>
</tbody>
</table>
Backup Slides
Enabling Feature for Container Publishing: Tarball Ingestion

Direct path for the common pattern of publishing tarball contents

```
$ cvmfs_server transaction
$ tar -xf ubuntu.tar.gz
$ cvmfs_server publish
```

```
$ cat ubuntu.tar.gz |
    cvmfs_server ingest -t -
```

```
Uses libarchive: support for rpm, zip, etc. could be easily added
```

```
Performance Example
Ubuntu 18.04 container – 4 GB in 250k files: 56 s untar + 1 min publish vs. 74s ingest
```
As part of opening the transaction, "4.2" is cloned to "4.2-patches"

- Meta-data only copy, thus extremely fast
- Only changes on top need to be published