



Publishing to CernVM-FS on Kubernetes



Andrea Valenzuela Ramírez & Jakob Blomer
(andrea.valenzuela.ramirez@cern.ch) (jakob.blomer@cern.ch)

Context:

The CernVM-File System (CernVM-FS) provides a scalable, reliable and low-maintenance software distribution service. It is asymmetric by construction: it has some writers, but many readers.

Classic approach:

The classic publishing process needs a dedicated *release manager machine* that provides the editable repository copy.

This approach was improved thanks to the introduction of a Gateway that provides concurrent access to the repository backend storage through a REST API.

Our approach:

We present further improvements to the CernVM-FS publishing process based on:

- 1 The construction of *ephemeral containers* on-demand.
- 2 The usage of new command to commit changes to the Gateway.

1 The CernVM-FS **ephemeral writable container** can provide a short-lived shell with writable access to a regular, read-only CernVM-FS repository. A writable mountpoint is normally a functionality that only publisher nodes provide. **With the ephemeral writable container, this capability becomes available to every regular client.**

The ephemeral writable container uses Linux user namespaces and fuse-overlays in order to construct the writable repository mountpoint.

Try it:

```
cvmfs_server enter <repository name> --transaction --repo-config <config path>
```

2 From the ephemeral shell, it is possible to publish changes directly to a Gateway by using the `cvmfs commit` command. It requires an extra configuration:

CVMFS_SERVER_URL
CVMFS_HTTP_PROXY

```
/etc/cvmfs/config.d/  
<repository name>.conf
```

CVMFS_UPSTREAM_STORAGE
CVMFS_KEYS_DIR

```
/customed_path/<repository  
name>/server.conf
```

Try it:

```
cvmfs_publish commit <repository name>
```

MAIN IMPLICATION

Kubernetes workflow:

With this approach, both the Gateway and the containers used for publishing can be instantiated as pods in a kubernetes cluster.

Thus, it is possible to have a kubernetes-native CernVM-FS publishing workflow.

