

Publishing to CernVM-FS on Kubernetes

Andrea Valenzuela Ramírez

andrea.valenzuela.ramirez@cern.ch

CERN - Technical Student project 2020-21

CernVMFS Workshop 2022

Nikhef, Amsterdam

September 12, 2022



1. Introduction
2. Required capabilities on the client side
3. Kubernetes-native CernVM-FS publishing workflow
4. Conclusion

There are different options to **publish content** into a CernVM-FS repository:

Serialized publishing

Dedicated *release manager machine* that provides the editable repository copy.

+ S3 storage

Parallel publishing

Gateway that provides concurrent access to the repository back-end storage, so that multiple *release managers* can publish at different directory subpaths.

Publishing from ephemeral containers

Short-lived containers can be created on demand to provide a temporary, editable repository copy for a single publish operation on regular (cloud) clients.

It is now possible thanks to the recent developments in the Linux kernel and in the Fuse user-space libraries that enable **fully unprivileged mounting** for Fuse file systems.

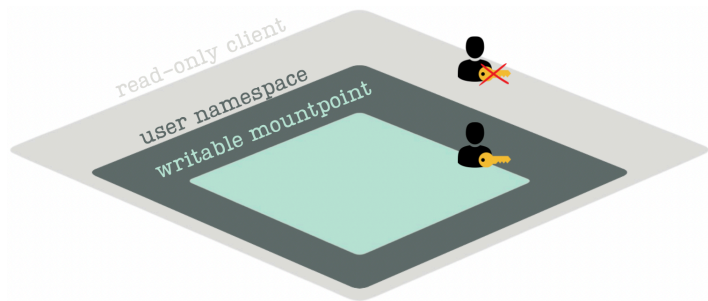


Figure: The ephemeral shell spawns a new user (Linux) namespace and provides the writable `/cvmfs` mountpoint using the Fuse implementation of the union file system overlaysfs.

Requirements

- Linux user namespaces.
- Fuse-overlaysfs.
- Recent enough kernel.
 - Vanilla ≥ 4.18
 - EL 8

This deployment aims to:

- Move CernVM-FS towards a **serverless model** where the server infrastructure components can be replaced by cloud services.

Use case: Publishing on Kubernetes

CernVM-FS can be deconstructed into their core constituents -storage, gateway and client-, which can be then individually hosted and orchestrated in cloud.

- Have the possibility to encapsulate **publisher nodes in containers**.
⇒ It can bring many benefits for the operations of publisher clusters.
- *Eventually* give the capability to any regular client to **become a publisher**.

The CernVM-FS command to start the ephemeral writable shell is so-called `enter` command:

```
cvmfs_server enter <repository-name> --transaction  
--repo-config <path/to/config/<repository-name>/server.conf>
```

The ephemeral shell needs two **extra configuration files**:

`<path/to/config/<repository-name>/server.conf>`

CVMFS_UPSTREAM_STORAGE
CVMFS_KEYS_DIR

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

CVMFS_SERVER_URL
CVMFS_HTTP_PROXY

Temporary directories

- **Session directory**
within the user's home directory.
- **Cache directory** for
the lifetime of the container.

From the ephemeral shell, it is necessary to **submit any change set to the gateway** so that it can be written into the authoritative storage. It is the so-called commit operation:

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

To see the **new published content** after closing the shell:

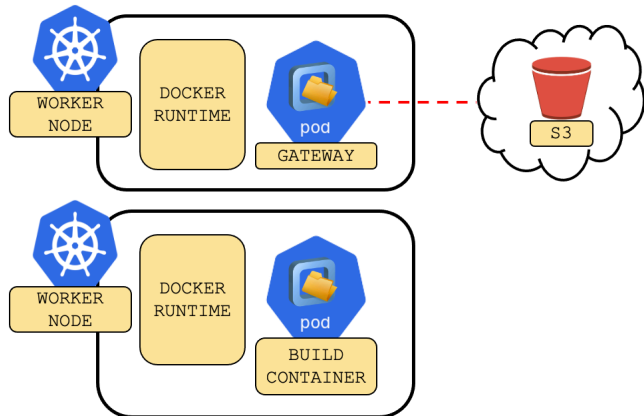
```
cvmfs_talk -i <repository-name> remount sync
```

To **discard changes**, if any, and exit the ephemeral shell:

```
exit or cvmfs_server abort <repository-name>
```

Kubernetes setup:

- Gateway and build container in **different worker nodes**.
- Default docker runtime.
- The Gateway has been configured to directly publish to **S3**.
- Regular CernVM-FS 2.9 installation in the build container.



► Demo

Figure: Schema of the presented use case where the build container gets temporarily promoted to a publisher.








► Gitlab repository

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: centos
  name: client
spec:
  containers:
    - image: aandvalenzuela/commitcommand:1.2
      name: client
      command: ["/bin/sleep", "3650d"]
      resources: {}
      securityContext:
        privileged: true
      volumeMounts:
        - mountPath: /dev/fuse
          name: fuse-device
  volumes:
    - name: fuse-device
      hostPath:
        path: /dev/fuse
        type: CharDevice
      dnsPolicy: ClusterFirst
      restartPolicy: Never
  status: {}
```

Figure: Pod configuration file prepared to work with a 2.9 CernVM-FS client installation. It is necessary to set the **Fuse mount point** as volume mount point.

- This CernVM-FS publishing workflow provides two new capabilities:
 - **Create ephemeral containers on demand** that give writable access to a repository.
 - Submit the change set to the gateway for **publishing before closing the writable environment**.
- The ephemeral shell improves other current working scenarios:
 - Software builder nodes can now **directly publish** their build products to the repository.
 - It helps deploying **non-relocatable packages**.
- Any client with suitable keys can **publish new content directly though the gateway in the cloud**.
 - ⇒ More work is planned to move the whole publishing infrastructure into the serverless paradigm and on demand publishing.

Questions? :)

-  Blomer J, Buncic P, Meusel R, Ganis G, Sfiligoi I and Thain D 2015 *Computing in Science Engineering* **17(6)** 61-71
-  Blomer J, Buncic P and Meusel R 2013 *The CernVM file system* CERN, Geneva, Switzerland, Tech. Rep, 2-1
-  Bocchi E, Blomer J, Mosciatti S and Valenzuela A 2021 *EPJ Web of Conferences* **251** 02033
-  Blomer J, Dykstra D, Ganis G, Mosciatti S and Priessnitz J 2020 *EPJ Web of Conferences* **245** 07012
-  Mondal S. K, Pan R, Kabir H. M, Tian T and Dai H. N 2022 *The Journal of Supercomputing* **78(2)** 2937-2987
-  Popescu R, Blomer J and Ganis G 2019 *Web of Conferences* **214** 03036
-  Blomer J, Ganis G, Mosciatti S and Popescu R 2019 *EPJ Web of Conferences* **214** 09007