

Table of Contents

Maintaining a CernVM-FS Repository.....	1
Publish Changes.....	1
Manage Named Snapshots.....	1
Recommendations for Named Snapshots.....	1
Rollback.....	2
Manage Nested Catalogs.....	2
Recommendations for Nested Catalogs.....	2
Final Remarks.....	3

Maintaining a CernVM-FS Repository

CernVM-FS repositories are maintained on dedicated machines, "release manager machines" or "installation boxes", that provide writable access to `/cvmfs/$repository`. Similar to versioning systems, changes to `/cvmfs/$repository` are temporary until they are *committed* or *discarded*. That allows you to test and verify your changes, for instance to test a newly installed release before publishing it to clients.

CernVM-FS is a versioning, snapshot-based file system. Whenever changes are *published* (committed), a new file system snapshot of the current state is created. These file system snapshots can be *tagged* with a name, which makes them *named snapshots*. A named snapshot is meant to stay in the file system. You can rollback to named snapshots and you can, on client side, decide to mount any of the named snapshots in lieu of the newest available snapshot.

Two named snapshots are managed automatically by `cvmfs`, *trunk* and *trunk-previous*. This allows for easy unpublishing of a mistake, by rolling back to the *trunk-previous* tag.

Publish Changes

Changes to a repository are encapsulated in transactions. In order to publish changes, use the following commands

```
cvmfs_server transaction
# Make changes to /cvmfs/...
cvmfs_server publish
```

If you want to discard changed *before* they have been published, use

```
cvmfs_server abort
```

Manage Named Snapshots

At the point of publishing, the resulting snapshot can be named. To do so, use the `-a` option like

```
cvmfs_server transaction
# Changes
cvmfs_server publish -a release-1.0
```

As a tag name, use an identifier without spaces and special characters. You can list all named snapshots by

```
cvmfs_server lstags
```

In order to remove (unpublish) a named snapshot, use the `-r` option like

```
cvmfs_server transaction
cvmfs_server publish -r release-1.0
```

Recommendations for Named Snapshots

Use named snapshots whenever you do larger modifications to the repository, for instance when you install a new software release. Only with named snapshots you have the ability to easily undo modifications and to preserve the state of the file system for the future. Nevertheless, do not use named snapshots excessively. Start cleaning up unnecessary snapshots once you have more than ~50.

Rollback

You can rollback your repository to any of the named snapshots. Technically, this means that the given snapshot is *re-published*, while all intermediate snapshots are removed from the history.

In order to rollback, do

```
cvmfs_server transaction
cvmfs_server rollback -t release-1.0
```

Note: a rollback is, like restoring from backups, not something you would do often. Use caution. A rollback is irreversible.

Manage Nested Catalogs

CernVM-FS stores meta-data (path names, file sizes, ...) in *file catalogs*. When a client accesses a repository, it has to download the file catalog first and then it downloads the files as they are opened. A single file catalog for an entire repository can quickly become large and impractical. At the same time, clients typically do not need all of the repository's meta-data at the same time. For instance, clients using software release 1.0 do not need to know about the contents of software release 2.0.

With nested catalogs, CernVM-FS has a mechanism to partition the directory tree of a repository into many catalogs. Repository maintainers are responsible for sensible cutting of the directory trees into nested catalogs. They can do so by creating and removing the magic file ".cvmfscatalog".

For example, in order to create a nested catalog for software release 1.0 in the hypothetical repository *experiment.cern.ch*, do

```
cvmfs_transaction
touch /cvmfs/experiment.cern.ch/software/1.0/.cvmfscatalog
cvmfs_server publish
```

If you want to merge a nested catalog with its parent catalog, remove the corresponding .cvmfscatalog file. Nested catalogs can be nested on arbitrary many levels.

Recommendations for Nested Catalogs

Nested catalogs should be created having in mind which files and directories are accessed together. This is typically the case for software releases, but can be also on the directory level that separates platforms. For instance, for a directory layout like

```
/cvmfs/experiment.cern.ch
|- /software
|   |- /i686
|   |   |- 1.0
|   |   |- 2.0
|   |   `-- common
|   |- /x86_64
|   |   |- 1.0
|   |   `-- common
|- /grid-certificates
|- /scripts
```

it makes sense to have nested catalogs at

- /cvmfs/experiment.cern.ch/software/i686

- /cvmfs/experiment.cern.ch/software/x86_64
- /cvmfs/experiment.cern.ch/software/i686/1.0
- /cvmfs/experiment.cern.ch/software/i686/2.0
- /cvmfs/experiment.cern.ch/software/x86_64/1.0

It could also make sense to have a nested catalog under grid-certificates, if the certificates are updated much more frequently than the other directories. It would *not* make sense to create a nested catalog under /cvmfs/experiment.cern.ch/software/i686/common, because this directory needs to be accessed anyway whenever its parent directory is needed.

As a rule of thumb, a single file catalog should contain more than 1000 files and directories but not contain more than ~200000 files.

Final Remarks

Be careful when publishing changes in a repository. Changes take ~1 hour to get populated through the caches to all grid worker nodes. That means if a change breaks something, the Grid might stop for an hour or longer until the fix is populated.

Restructuring the repository's directory tree is an expensive operation in CernVM-FS. Moreover, it can easily break clients when they switch to a restructured file system snapshot. Therefore, your software directory tree layout should be relatively stable before you start filling the CernVM-FS repository.

We strongly discourage the use of installation cron jobs. A human being should verify the changes that are published.

-- JakobBlomer - 08 Nov 2013

This topic: CvmFS > MaintainRepositories

Topic revision: r2 - 2013-11-08 - SteveTraylen



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback