

Table of Contents

Modernising CVMFS as a Service.....	1
Project Description.....	2
Phase 1: Get to know the technologies.....	3
Current Status of CVMFS Release Management.....	3
Linux Containers with Docker.....	3
Remote storage access with NFS/HTTP/S3.....	3
Phase 2: Prototyping.....	4
The release manager container.....	4
Test an S3-backed CVMFS repository.....	4
Phase 3: Putting it together.....	5
CvmFS Web Utilities.....	5

Modernising CVMFS as a Service

Project Description

The CernVM Filesystem (CVMFS) is used by LHC experiments to deliver applications and content to the Worldwide LHC computing grid. Presently, publishing new content to CVMFS requires a clumsy combination of shared interactive VMs and attached block storage; this approach is not highly available, not highly performing, and not scalable.

This summer student project will be to develop a prototype of a new approach to CVMFS publishing. First, the backend storage for CVMFS will be moved to cloud storage, which will be more scalable and flexible to handle the current and future growth of the service. Next, the interactive publishing VMs will be replaced with Linux containers, launched on-demand and allowing CVMFS maintainers to distribute applications more efficiently than at present.

The student will get hands-on experience developing a modern "devops" service, using S3 cloud storage and Linux containers on our scalable cloud infrastructure. Further, if the project is successfully accomplished, the student's work would make a tangible impact in CVMFS publishing -- which will improve the day-to-day operations of the experiments' computing teams.

Phase 1: Get to know the technologies

Current Status of CVMFS Release Management

We maintain a **testing** CVMFS release manager machine at `cvmfs-test.cern.ch`. Practise the release management process there: open a new transaction, write some changes to `/cvmfs/test.cern.ch`, publish the transaction, then go look on `lxplus/cvmfs/test.cern.ch` to see the new repository revision. Do this a few times, and study how CVMFS is managing its data.

The CVMFS release manager machines are puppet-managed, using our `cvmfs/lx` hostgroup. Study their configuration here: <https://gitlab.cern.ch/ai/it-puppet-hostgroup-cvmfs>

Linux Containers with Docker

Get a test machine, e.g. a VM in OpenStack, where you can install and test docker. Maybe you can do this with Docker on your Mac. One of our IT-CM colleagues has prepared a recipe to build an "lxplus-like" docker image. Use that to prepare your own image and run it. Compare with the `lx.pp` configuration to start thinking about what we need for a "release manager" docker image.

For more info:

- <https://cern.service-now.com/service-portal/article.do?n=KB0004284>
- <https://cern.service-now.com/service-portal/article.do?n=KB0004553>

Remote storage access with NFS/HTTP/S3

One of the key steps in this project will be to remotely store and serve up the CVMFS backend data. We can do this a few ways:

- release manager machines mount a large NFS server, and apache/nginx serve that NFS data via HTTP.
- Use CVMFS's S3 backend functionality:
<http://cvmfs.readthedocs.io/en/stable/cpt-repo.html#s3-compatible-storage-systems>

Study the technologies, getting a test account on our Ceph S3 service to understand how S3 works and how it might work in combination with CVMFS repositories.

Phase 2: Prototyping

The release manager container

Build a Linux container image that can be used to manage an existing, e.g. /cvmfs/test.cern.ch, repository.
Unknowns include:

- how do we pass the repo signing secret into the image?
- CVMFS needs unionfs or aufs -- does this work with docker?

Test an S3-backed CVMFS repository

Follow the recipe here <http://cvmfs.readthedocs.io/en/stable/cpt-repo.html#s3-compatible-storage-systems> to build an S3-backed CVMFS repository. Does this have any advantages over our current repo storage?

Phase 3: Putting it together

Combine the docker and remote-storage CVMFS repo to build up a full solution. Document the release manager guidelines for how they can use this prototype.

Extra credit: Can we make a docker image that can manage **existing** cvmfs repositories?

CvmFS Web Utilities

- - advanced search
- WebTopicList - all topics in alphabetical order
- WebChanges - recent topic changes in this web
- WebNotify - subscribe to an e-mail alert sent when topics change
- WebRss, WebAtom - RSS and ATOM news feeds of topic changes
- WebStatistics - listing popular topics and top contributors
- WebPreferences - preferences of this web

-- DanielVanDerSter - 2017-06-06

This topic: CvmFS > CVMFSaaS

Topic revision: r1 - 2017-06-06 - DanielVanDerSter



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