



# Updates on ergonomics for “thin docker images”



# Docker images vs. Thin images

- Docker image
  - Manifest + List of tarfiles
- Thin image
  - Manifest + List of directory on CVMFS
- Stack them one on top of the other using an union filesystem
- Start your container



# How to create a thin image

1. Download the image
2. Push each layer into CVMFS
  - a. Manually
  - b. `cvmfs_server ingest`
3. Create the new manifest for the thin image
  - a. Already automated process with a go software



# New Ergonomics

Easier creation of thin image

1. [Docker2cvmfs](#)
  - a. Convert the image and import it into dockerd
  - b. Requires an operator
2. [Docker2cvmfs daemon](#)
  - a. Convert images and push them into registries
  - b. (Mostly) Automatic process

# Docker2cvmfs



1. Pull a docker image
2. Download each layer
3. Ingest into CVMFS each layer
4. Create the thin image
5. Import the thin image into dockerd
6. Image is now locally available to the operator

```
./docker2cvmfs make-thin \  
  -i library/redis:4 \  
  -o siscia/thin/redis:4 \  
  -r test.cern.ch
```



# Automate the last step removing the operator

- What to do with the thin image just created?
- Push it into the registries
- Introducing “wish”
  - `< ImageInput, CVMFS repository, ImageOutput >`



# What is an image

```
Registry    +      https://registry.hub.docker.com
Repository  +      library/redis
Reference   =      ( :4 | @sha256:abc... )

Image
```

# Docker2cvmfs\_daemon workflow



Start by adding as many wishes as necessary:

```
./daemon add-wish -i InputImage \  
                -r repo.cvmfs.ch -o OutputImage
```

Convert all the wishes

```
./daemon convert
```

Some image can change, like when pushing a new “latest” tag

Run convert on a loop

```
./daemon loop
```





## Few warnings

- Mostly an append only repository
- What layer to delete?
  - a. Delete a layer could break running images
- Use “stable” tags in the wishes, or even better the hashes.