

# cvmfs at GSI

## cvmfs Workshop Amsterdam 2022

Sören Fleischer

13.09.2022

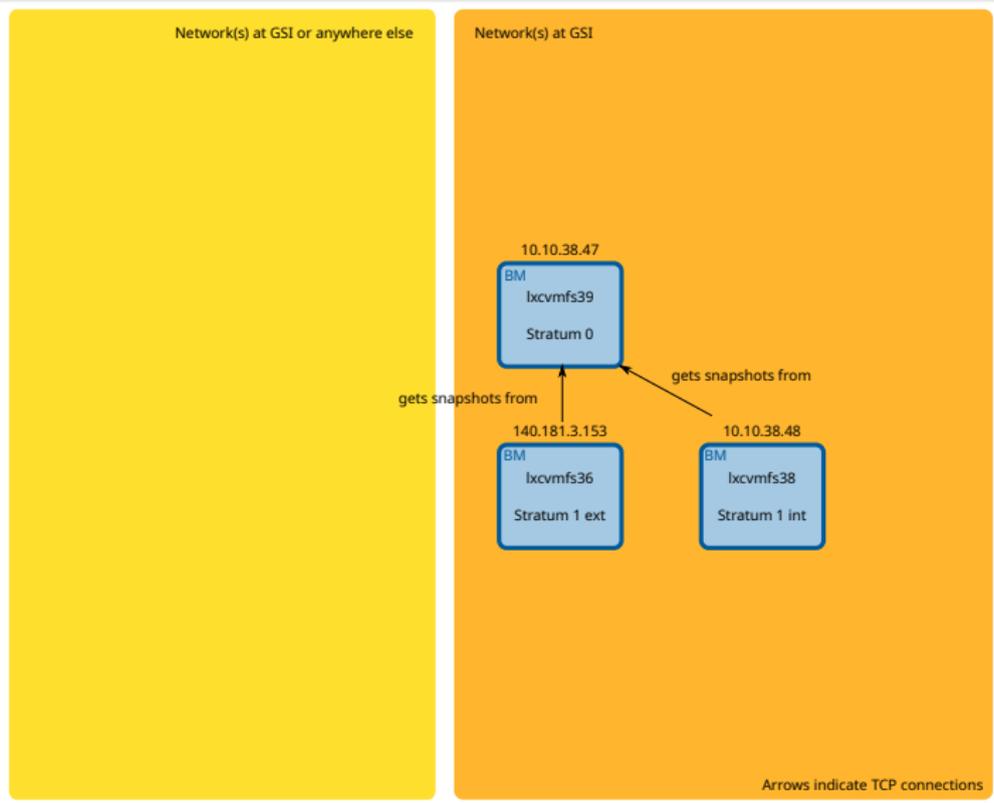
# Operations

- OS: Debian 10
- OS-level configuration: Chef/CINC
- Updates: unattended-upgrades
- cvmfs workflows: Ansible

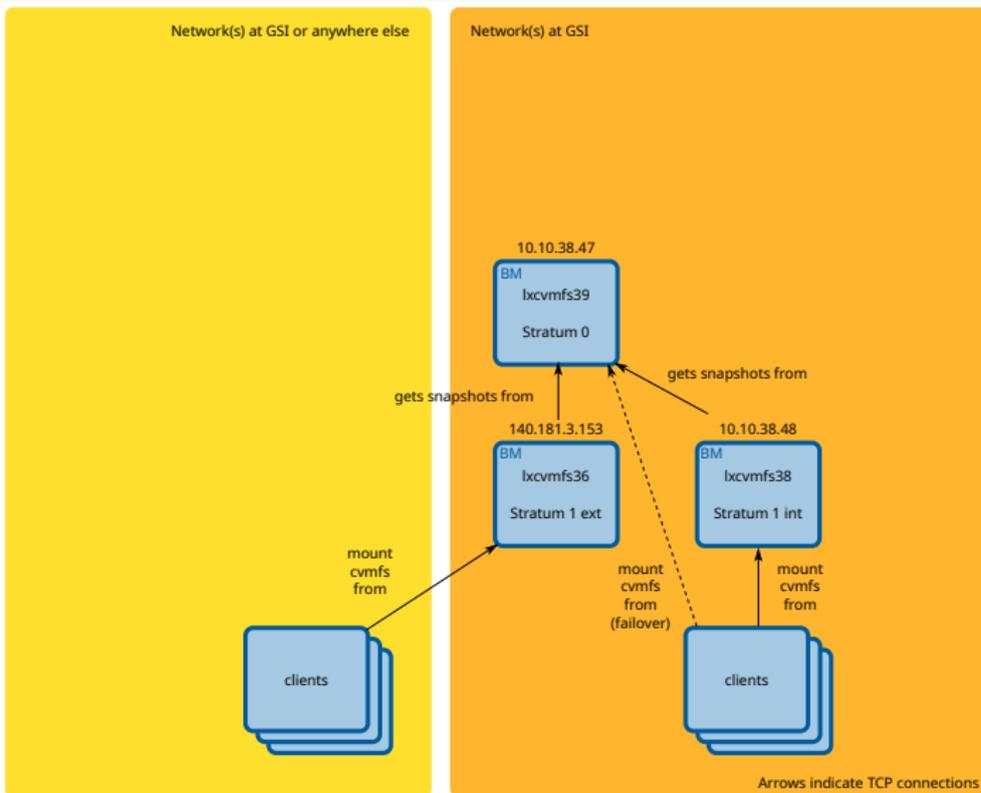
# Current cvmfs Design at GSI



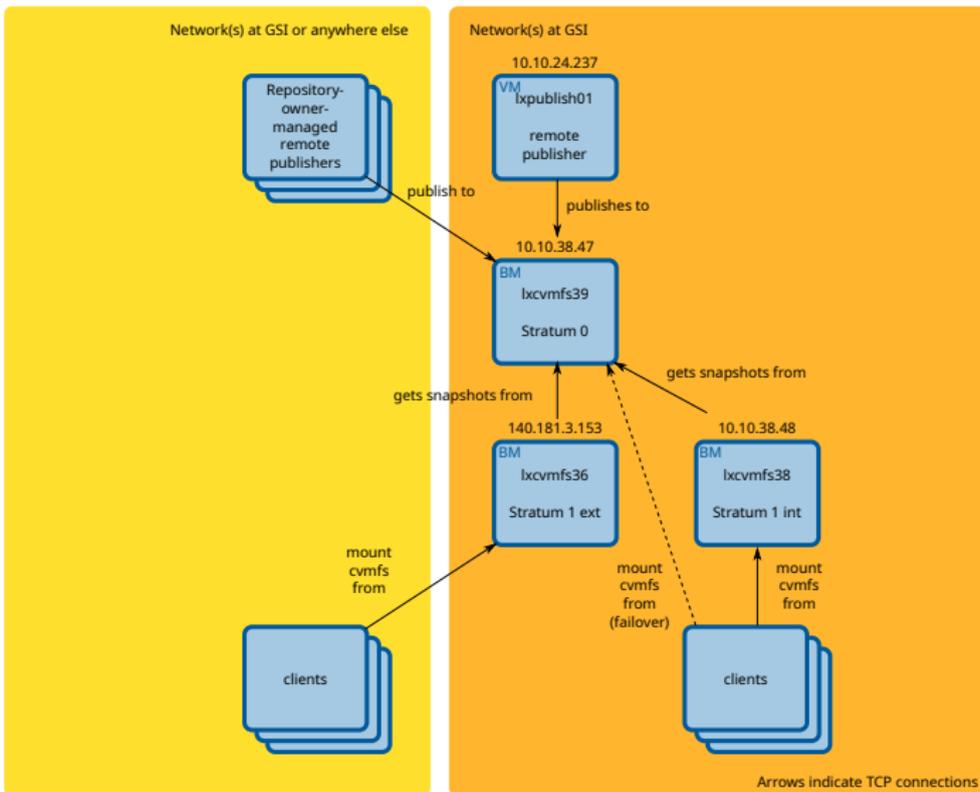
# Current cvmfs Design at GSI



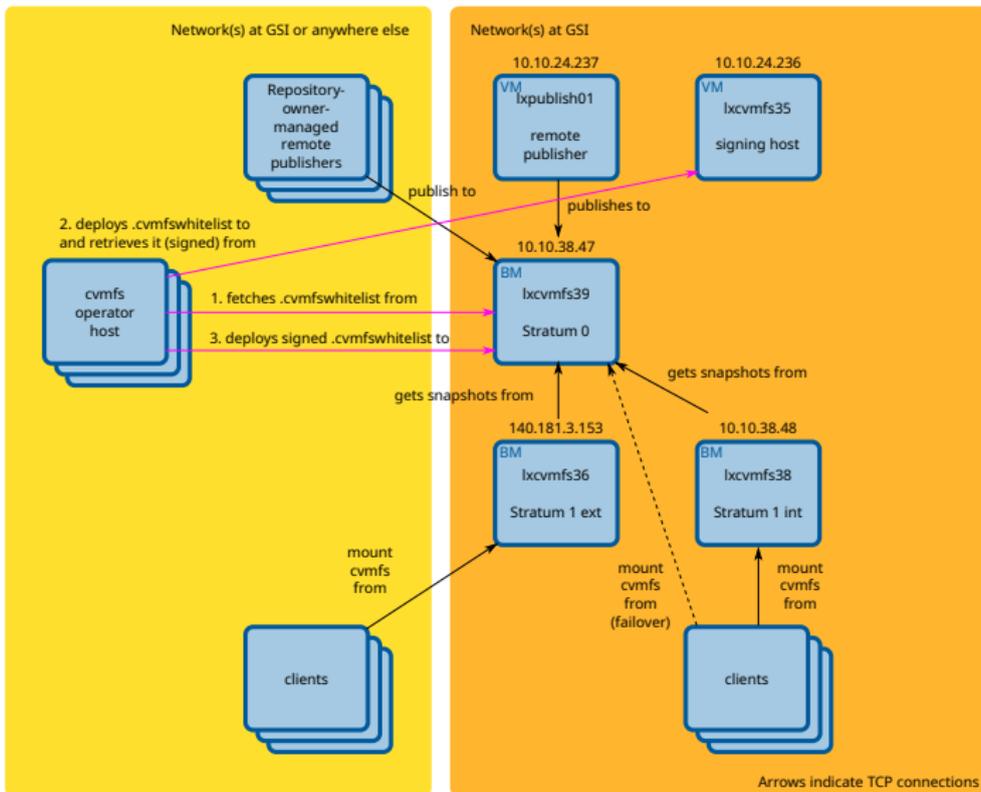
# Current cvmfs Design at GSI



# Current cvmfs Design at GSI



# Current cvmfs Design at GSI



## Items to be backed up

- `/etc/cvmfs/`
- `/srv/cvmfs/`
- `/var/spool/cvmfs/`
- `/etc/apache2/`
- `/etc/fstab`

# Backup Workflow Stratum 0

- Every day a script
  - stops **cvmfs-gateway** and **apache2**
  - umounts all cvmfs repositories
  - creates a **btrfs** snapshot
  - starts **apache2** and **cvmfs-gateway**
  - mounts all cvmfs repositories
- Effective downtime each day:  $\approx 5..10$  seconds
- The 2 most recent snapshots are kept on the Stratum 0
- Those 2 snapshots are picked up by **TSM** every day
- TSM is used with config DAILY to keep some number of daily copies

# Stats

- 16 repositories
- 7 repositories on external Stratum 1
- 4 repositories use their own publisher host
- $\approx$  270 GB total

# Main Use Case

- Making executables/libraries accessible to batch farm / cluster nodes
  - Traditional "copying pre-compiled software packages to cvmfs"
  - Spack [package management tool]
  - "VAEs" (Virtual Execution Environments)

# VAEs / Containers

Main(?) purpose of containers:

- Making a specific set of libraries visible to an executable
  - Without installing those libraries to the host system
  - Eliminating or greatly reducing Dependency Hell
  - Saving resources compared to proper virtualisation

# VAEs / Containers

- The host system can be different than what the users see when they log in
- The host system can be upgraded/replaced without breaking every user's workflows
- Users who need different library environments can be accommodated, given they supply an Apptainer container image
- However, the Linux Kernel seen by the application is always that of the host system

# VAEs / Containers

Submit Nodes	VAE	Support Until
<a href="#">virgo-debian10.hpc.gsi.de</a>	Debian 10	TBD
<a href="#">virgo-centos7.hpc.gsi.de</a>	CentOS 7	TBD
<a href="#">virgo-amdgpu.hpc.gsi.de</a>	CentOS 7	TBD

- Log in to Virgo (GSI Cluster/Batch Farm) using any of these hostnames will create a shell **inside** the corresponding VAE/container.
- For each batch job submission, the user can configure which VAE they want to use

# SLURM Singularity Plugin

## Usage

The plugin adds following command-line options to `salloc`, `srun` and `sbatch` :

Option	Description
<code>--singularity-container=</code>	Path to the Singularity container. Equivalent to using the environment variable <code>SLURM_SINGULARITY_CONTAINER</code> .
<code>--singularity-bind=</code>	<a href="#">User-defined bind paths</a> will be appended to the defaults specified in the plug-in configuration. Equivalent to using the environment variable <code>SLURM_SINGULARITY_BIND</code> .
<code>--singularity-args=</code>	List of <code>singularity exec</code> <a href="#">command-line arguments</a> .
<code>--singularity-no-bind-defaults</code>	Disable the bind mount defaults specified in the plug-in configuration.

The implementation of [slurm-singularity-wrapper.sh](#) adds additional environment variables:

Env. Variable	Description
<code>SLURM_SINGULARITY_DEBUG</code>	Set <code>true</code> to enable debugging output
<code>SLURM_SINGULARITY_GLOBAL</code>	Optional <a href="#">global options</a> to the <code>singularity</code> command

<https://git.gsi.de/hpc/cluster/slurm-singularity-exec>

# SLURM Singularity Plugin

```
SLURM_SINGULARITY_DEBUG=true SLURM_SINGULARITY_GLOBAL=--silent \  
  srun --singularity-container=/tmp/debian10.sif \  
    --singularity-bind=/srv \  
    --singularity-args="--no-home" \  
    -- /bin/grep -i pretty /etc/os-release
```

Executing it will generate output similar to:

```
Start Singularity container /tmp/debian10.sif  
Debug: SLURM_SINGULARITY_CONTAINER=/tmp/debian10.sif  
Debug: SLURM_SINGULARITY_ARGS=--no-home  
Debug: SLURM_SINGULARITY_BIND=/etc/slurm,/var/run/munge,/var/spool/slurm  
Debug: SLURM_SINGULARITY_GLOBAL=--silent  
Debug: singularity --silent exec --bind=/etc/slurm,/var/run/munge,/var/spool/slurm,/srv --no-home /tmp/debian10.s  
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
```

<https://git.gsi.de/hpc/cluster/slurm-singularity-exec>