



FermiGrid images available for FIFE

Marco Mambelli

Fermilab – FIFE meeting

May 30, 2019

Announcement

- Singularity images of FermiGrid nodes are available
 - This will give you the same OS, binaries and libraries also outside of Fermilab (on OSG, cloud, ...)
 - Will be like a FermiGrid worker node
 - Images published on CVMFS via OSG
 - You can pick the OS (SL6, SL7)

[/cvmfs/singularity.opensciencegrid.org/fermilab/fnal-wn-sl7:latest](https://cvmfs/singularity.opensciencegrid.org/fermilab/fnal-wn-sl7:latest)
[/cvmfs/singularity.opensciencegrid.org/fermilab/fnal-wn-sl6:latest](https://cvmfs/singularity.opensciencegrid.org/fermilab/fnal-wn-sl6:latest)

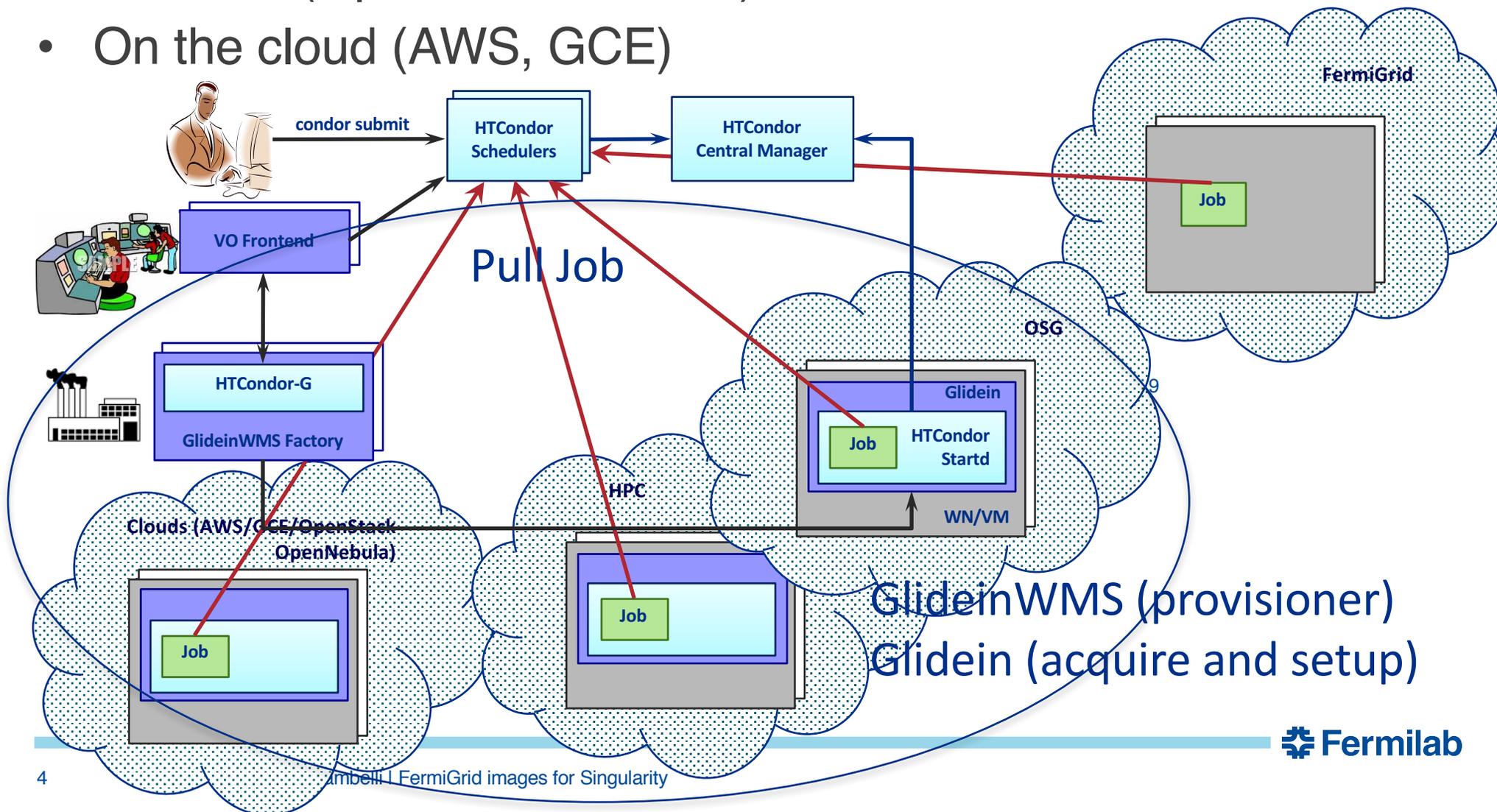
- Some background and details are in the next slides

Some definitions – simple version

- Worker Node
 - Hardware capabilities (CPUs, memory, GPU, ...)
- Containers
 - A way to provide uniform environment
- Image (Container image)
 - The environment your job will run in
 - OS, libraries, binaries, ...
- Singularity, Docker
 - Software that allows to use containers

Jobs run on a Worker Node, it may be...

- At Fermilab
- On OSG (Open Science Grid)
- On the cloud (AWS, GCE)



Using Singularity images

- On FermiGrid (direct submission) the FermiGrid images for SL6 and SL7 is what is provided to you (no need to use Singularity here)
 - You may still want Singularity if you want to run you different custom image
 - You have to use a wrapper script to invoke Singularity with the right image (in red below) and your job (in green):

```
/usr/bin/singularity exec -B /cvmfs --ipc --pid -contain \  
/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el7:latest /bin/sleep 60
```

- Using GlideinWMS you are running on unknown resources, so is recommended to use Singularity to see the same OS and libraries you have at Fermilab.
 - The setup is done for you by the Glidein if Singularity is available
 - Normally select the FermiGrid images (fnal-wn-sl7, fnal-wn-sl6)
 - Or your custom image if needed

FermiGrid images on GlideinWMS

- New images similar to FermiGrid nodes
 - /cvmfs/singularity.opensciencegrid.org/fermilab/fnal-wn-sl7:latest
 - /cvmfs/singularity.opensciencegrid.org/fermilab/fnal-wn-sl6:latest
- To use these (or other) images for your jobs
 - Singularity must be enabled
 - Add to the condor submit file of the job:
`+SingularityImage = "full_path_to_the_image"`
 - Or use a -I option in JobSub:
`-I '+SingularityImage="full_path_to_the_image"'`
- The VO can set these as a default image for you
 - The job will simply run in a FermiGrid like node by default

<https://cdcvs.fnal.gov/redmine/projects/glideinwms/wiki/GlideinwmsSingularity>

How to enable Singularity in GlideinWMS (for VO admins)

- VOs and Resources negotiate the use of Singularity
 - GLIDEIN_Singularity_Use (FE) and GLIDEIN_SINGULARITY_REQUIRE (FA)
 - REQUIRED, PREFERRED, OPTIONAL or NEVER
 - REQUIRED or PREFERRED in the intersection in the table means that Singularity is enabled

	NEVER	OPTIONAL	PREFERRED	REQUIRED (REQUIRED_GWMS)
NEVER	NEVER	NEVER	NEVER	FAIL
OPTIONAL	NEVER	NEVER	PREFERRED	REQUIRED
PREFERRED	NEVER	PREFERRED	PREFERRED	REQUIRED
REQUIRED	FAIL	REQUIRED	REQUIRED	REQUIRED
DISABLE_GWMS	DISABLE	DISABLE	DISABLE	DISABLE (FAIL)

- VOs can define short names (key/ID, path or URL)
 - SINGULARITY_IMAGES_DICT (FE, FA)
- User jobs can then select an image from the dictionary
 - REQUIRED_OS
- Specify custom software and libraries to be mounted
 - GLIDEIN_SINGULARITY_BINDPATH

OSG uses extracted Images on CVMFS

OSG stores container images on CVMFS in extracted form (Singularity calls this “sandbox” mode). That is, OSG takes the Docker image layers or the Singularity img/simg/sif files and export them onto CVMFS. For example, ls on one of the containers looks similar to ls / on any Linux machine:

```
$ ls /cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el7:latest/  
cvmfs  host-libs  proc  sys  anaconda-post.log  lib64  
dev    media     root  tmp  bin                 sbin  
etc    mnt       run   usr  image-build-info.txt  singularity  
home   opt       srv   var  lib
```

This is very efficient: Most container instances only use **a small part** of the container image (**50-150 MB**) and that part is **cached** in CVMFS! And CVMFS deduplicates everything so Docker layers are not a problem.

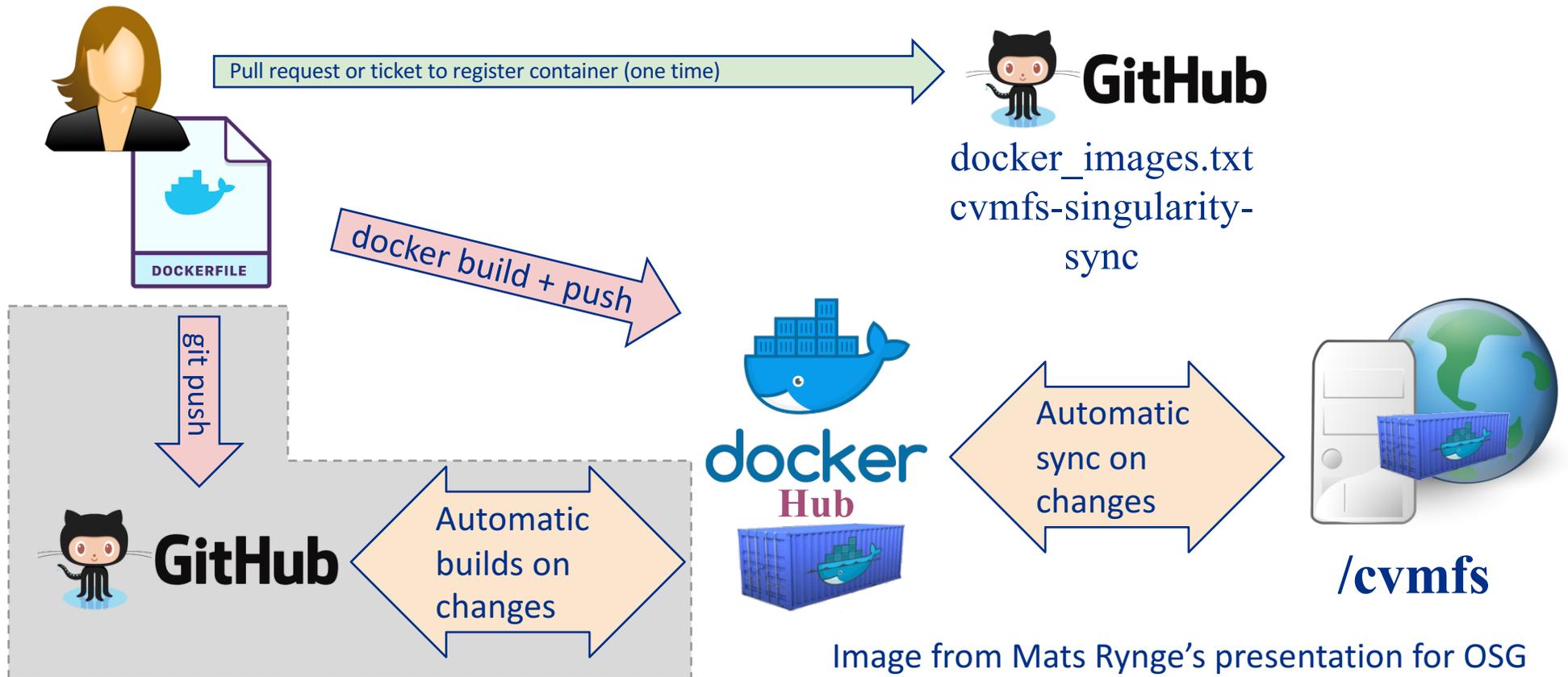
Available Container Images you can choose in OSG

ls /cvmfs/singularity.opensciencegrid.org/

	Image Location	Definition	Description
EL 6	/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el6:latest	GitHub	A basic Enterprise Linux (CentOS) 6 based image. This is currently our default image
EL 7	/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el7:latest	GitHub	A basic Enterprise Linux (CentOS) 7 based image.
Ubuntu Xenial	/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-ubuntu-xenial:latest	GitHub	A good image if you prefer Ubuntu over EL flavors
Ubuntu 18.04 (Bionic)	/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-ubuntu-18.04:latest	GitHub	A good image if you prefer Ubuntu over EL flavors
TensorFlow	/cvmfs/singularity.opensciencegrid.org/opensciencegrid/tensorflow:latest	GitHub	Base on the TensorFlow base image, with a few OSG packages added
TensorFlow GPU	/cvmfs/singularity.opensciencegrid.org/opensciencegrid/tensorflow-gpu:latest	GitHub	Used for running TensorFlow jobs on OSG GPU resources

~150 images, consisting of pre-defined ones by OSG staff, base images from Docker (different OSes, Python, r-base, ...) and custom images by our users

You can build your image and publish it in OSG



Fermilab provides support to Experiments build Docker images
Here to publish them in OSG:

<https://github.com/opensciencegrid/cvmfs-singularity-sync>