SINGULARITY & CONTAINERS A TIER-1 PERSPECTIVE

ANDREW LAHIFF

15TH SEPTEMBER 2017, GRIDPP 39 - LANCASTER

GridPP
UK Computing for Particle Physics

Science & Technology Facilities Council

OVERVIEW

- RAL batch farm
- xrootd access to ECHO from RAL worker nodes
- Singularity

RAL BATCH FARM

We fully migrated to the HTCondor Docker universe early this year

- all jobs run in Docker containers, running on bare metal
- all previous functionality still works
  - machine job features, glexec, ...

Some statistics

- over 700 hosts running HTCondor & Docker engine
- around ~12K containers running concurrently
  - corresponding to over 22000 cores
- up to ~0.5M containers created per week

RAL BATCH FARM

HS06 benchmarks from the 2016 WNs

- SL7 bare metal
- SL7 Docker containers on SL7 -0.84%
- SL6 Docker containers on SL7 -5.69%

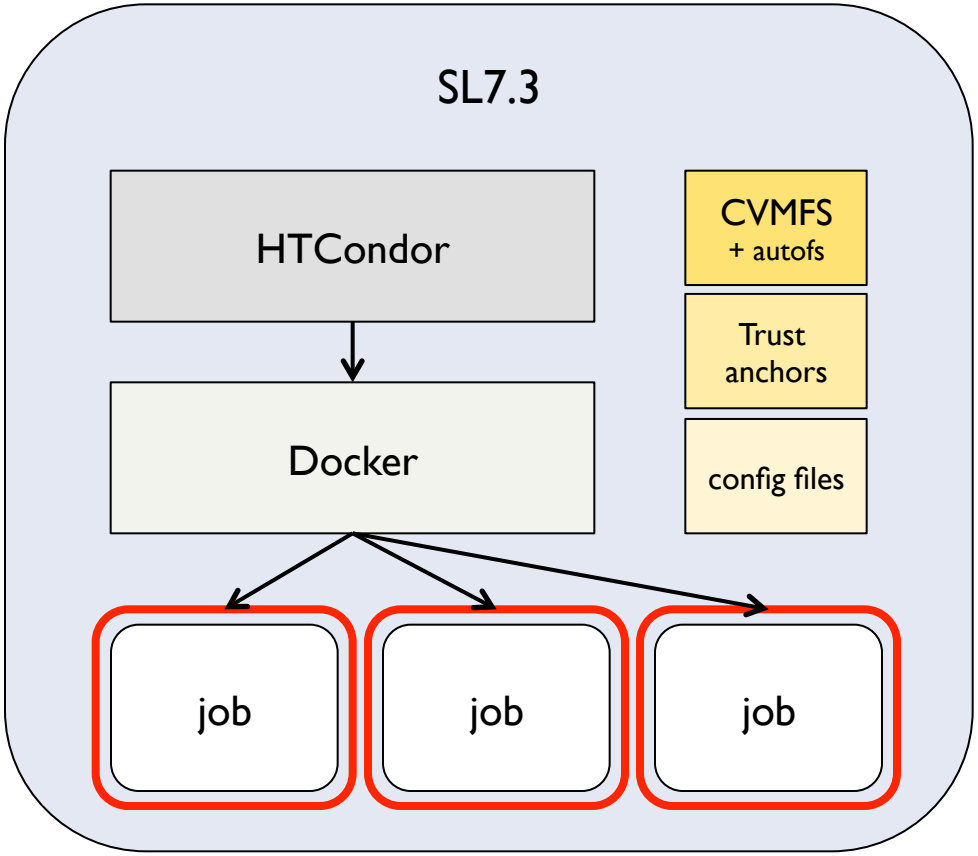
We're using the "SL7 Docker containers on SL7" value for all capacity & accounting purposes

Benchmarking done in a way consistent with how jobs are run

- each benchmarking thread was run in a separate container
- in our case 40 in total on machines with 40 ht cores

RAL BATCH FARM

A RAL worker node



- containers run as unprivileged pool account users
- users don't have access to the Docker daemon at all
- no way for users to specify arbitrary images via the Grid
- CVMFS, Trust anchors, config bind mounted into containers

RAL BATCH FARM

Only site-provided images can be used

- SL6 & SL7 images available

Added an SL7 queue on ARC CEs (only done on 2 of 4 so far)

- So far mostly ALICE have been using SL7
  - their jobs work on either SL6 or SL7
- Some issues
  - queue publishing is not correct
    - ARC assumes queues correspond to partitioned resources
  - some VOs don't specify a queue for SL6
    - jobs can end up on either SL6 or SL7

RAL BATCH FARM

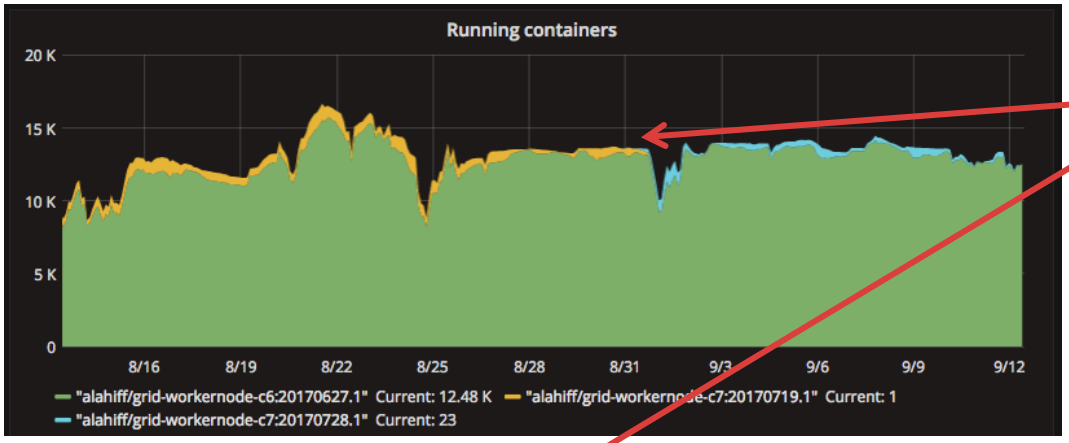
Schedd job transform on the CEs (HTCondor 8.6.x)

- sets Docker image based on queue
- updates jobs' Requirements expression

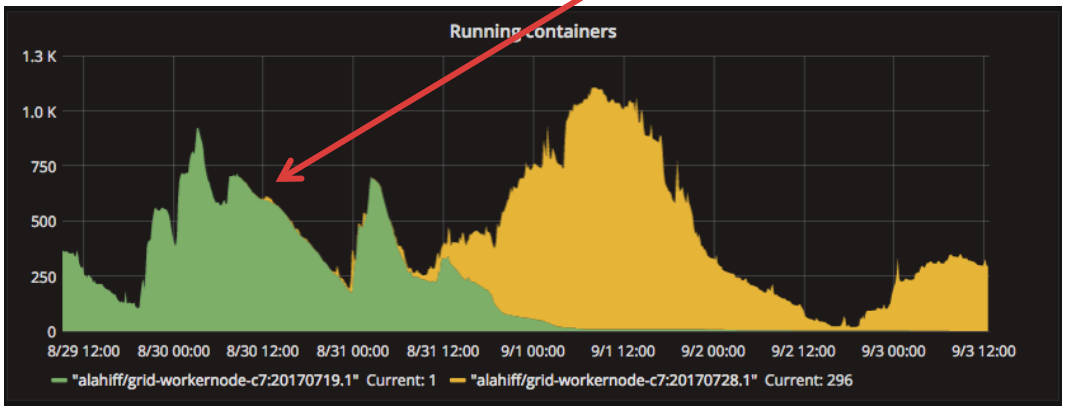
```
# Convert job to Docker universe
JOB_TRANSFORM_NAMES = $(JOB_TRANSFORM_NAMES), DefaultDocker
JOB_TRANSFORM_DefaultDocker @=end
[
   Requirements = JobUniverse == 5 && DockerImage =?= undefined && Owner =!= "nagios";
   set_WantDocker = true;
   eval_set_DockerImage = ifThenElse(NordugridQueue =?= "EL7", "alahiff/grid-workernode-c7:20170728.1", "alahiff/grid-worke
rnode-c6:20170627.1");
   set_Requirements = ( TARGET.HasDocker ) && ( TARGET.Disk >= RequestDisk ) && ( TARGET.Memory >= RequestMemory ) && ( TAR
GET.Cpus >= RequestCpus ) && ( TARGET.HasFileTransfer ) && ( x509UserProxyVOName =?= "atlas" && NumJobStarts == 0 || x509Us
erProxyVOName =!= "atlas");
   copy_TransferInput = "OriginalTransferInput";
   eval_set_TransferInput = strcat(OriginalTransferInput, ",", Cmd);
]
@end
```

RAL BATCH FARM

Monitoring in place so we know what images are being used



SL7 image updated



Running SL7 containers only shown

RAL BATCH FARM

Additions to the HTCondor health check script

- checks Docker daemon (runs "docker ps")
- checks containerd (checks if the containerd process exists)
- checks if /pool is an XFS filesystem formatted with ftype=1
  - needed for overlayfs Docker storage backend

We use the "DOCKER" knob in HTCondor

- Python script to modify "docker run" arguments generated by HTCondor, e.g.
  - implement both soft & hard cgroup memory limits
  - add required capabilities for Singularity

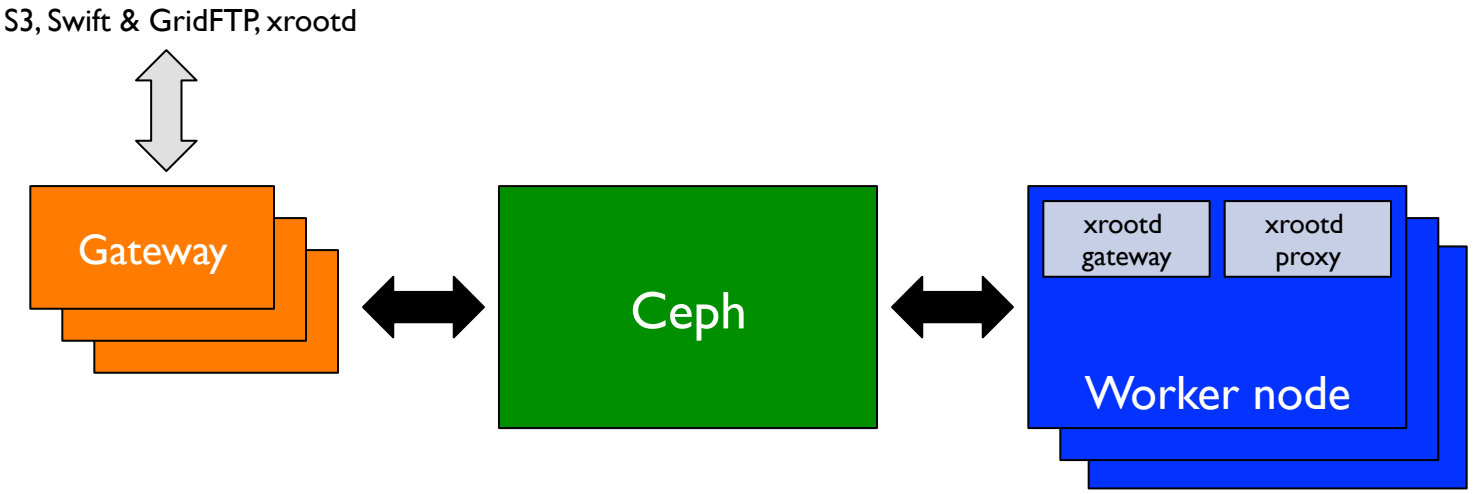
XROOTD ACCESS TO ECHO

We have more than just jobs running in containers on WNs

- xrootd Ceph gateways & proxies deployed on each WN
- provides highly scalable data access
  - avoids gateways being a bottleneck

Running important services on a WN is not trivial

- jobs could affect the xrootd services
- the xrootd services could affect jobs



XROOTD ACCESS TO ECHO

We found that pids cgroups can be very useful

- during the recent ECHO problems the xrootd gateways started leaking threads, affecting jobs
- now we limit the number of pids allowed in the xrootd gateway containers

Jobs can affect gateways & proxies

- very recently experienced issues where proxies could no longer allocate any memory (jobs were using it all)
- will try putting all containers running jobs into a parent cgroup with a hard memory limit
  - leaving memory for xrootd (4 GB + 8 GB), the kernel etc

XROOTD ACCESS TO ECHO

5 external gateways

gateways on worker nodes

XROOTD ACCESS TO ECHO

External gateways treated as pets

- broken external gateways callout & need manual intervention

Gateways on WNs treated as cattle

- too many to treat as pets
- a health check runs every 3 mins
- unhealthy gateways on WNs are destroyed & re-created automatically

HTCondor aware of health status

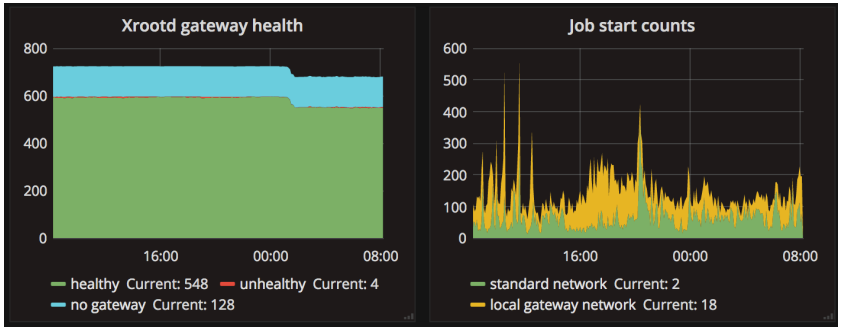
- job ClassAds specify if xrootd access to ECHO is needed
- WNs will only run new such jobs if the xrootd gateway is healthy

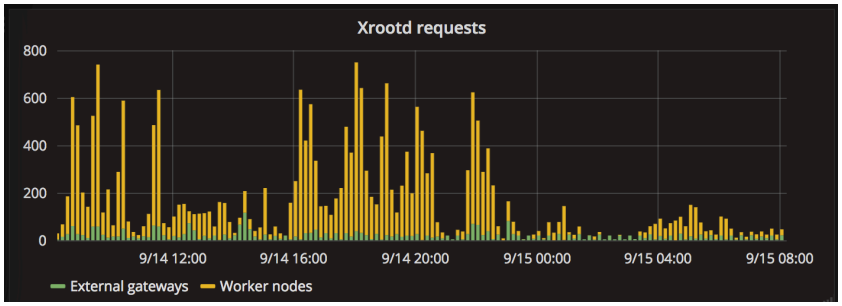
Callout only if number of broken gateways gets too large

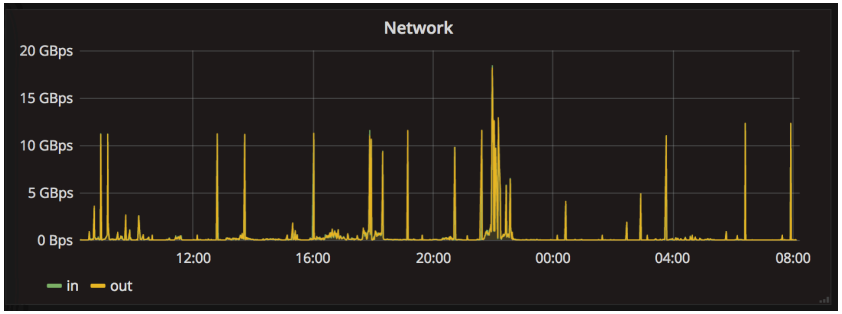
This limits the effects of any problems & reduces effort required

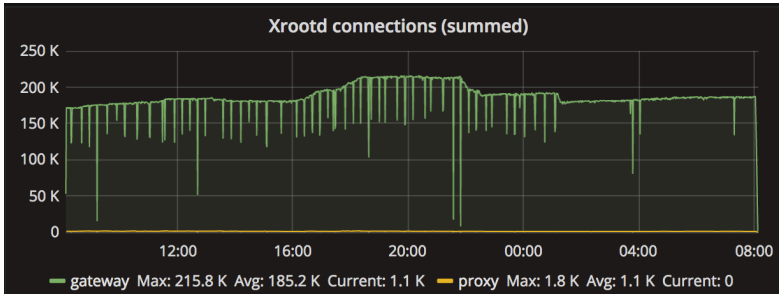
XROOTD ACCESS TO ECHO

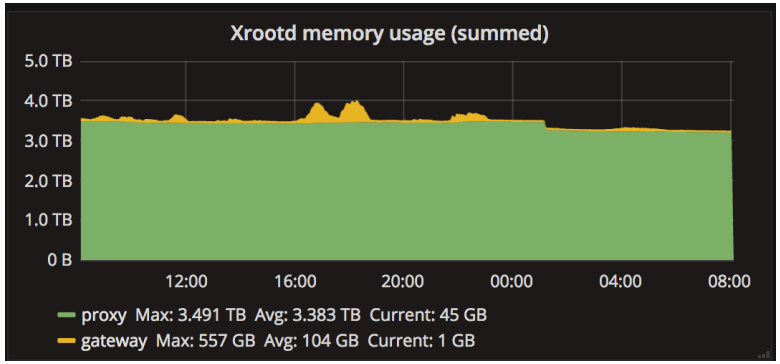
Monitoring











Using

- Telegraf to collect per-container metrics
- xrootd UDP monitoring to ElasticSearch
- collect xrootd connection stats by entering the appropriate network namespace

CONTAINER USE CASES

There are several different ways of using containers with batch systems

CONTAINER USE CASES (1)

Benefits for the site

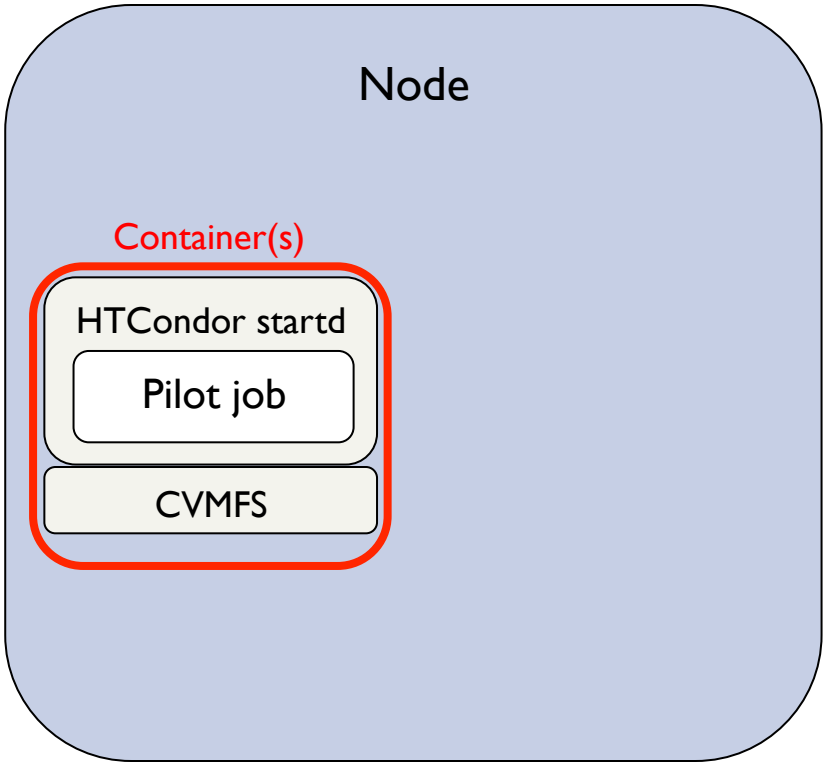
- entire worker node in container(s)
- don't need to have resources dedicated as "grid worker nodes"
- can run other activities on the same host

Complications

- CVMFS

Example:

- running LHC jobs on public clouds using Kubernetes – RCUK Cloud WG Particle Physics Pilot*

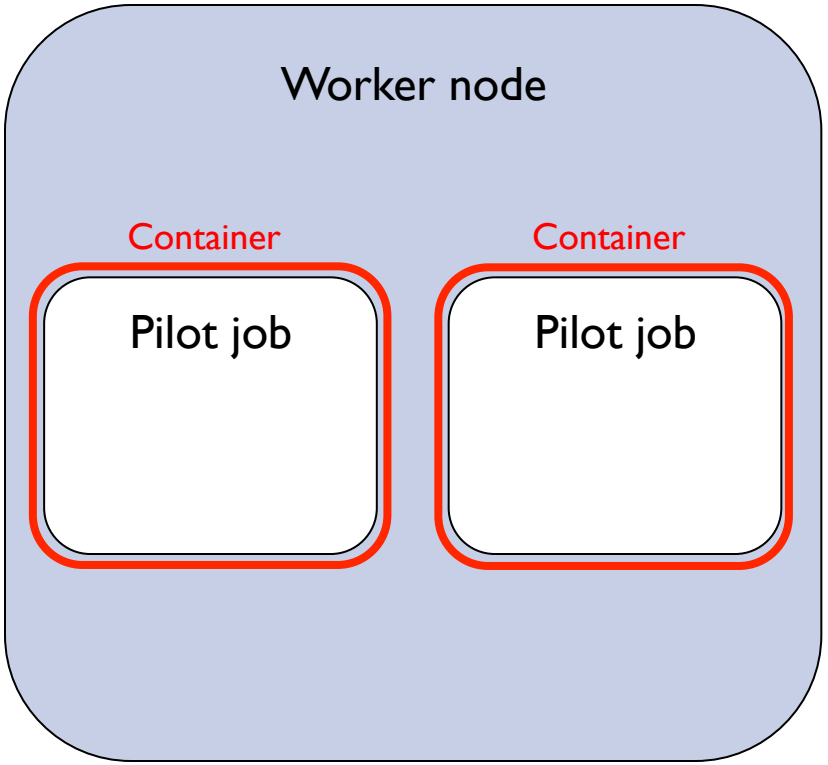


* https://indico.cern.ch/event/578974/contributions/2695215/attachments/1521436/2376946/ATLAS_AZURE-v1.pdf

CONTAINER USE CASES (2)

Benefits for the site

- isolate jobs & impose resource limits
- provide appropriate environment for jobs, without tying it to the host
- site can provide different job environments without partitioning resources
- even if VO runs payload jobs in containers, the pilot still needs a specific environment



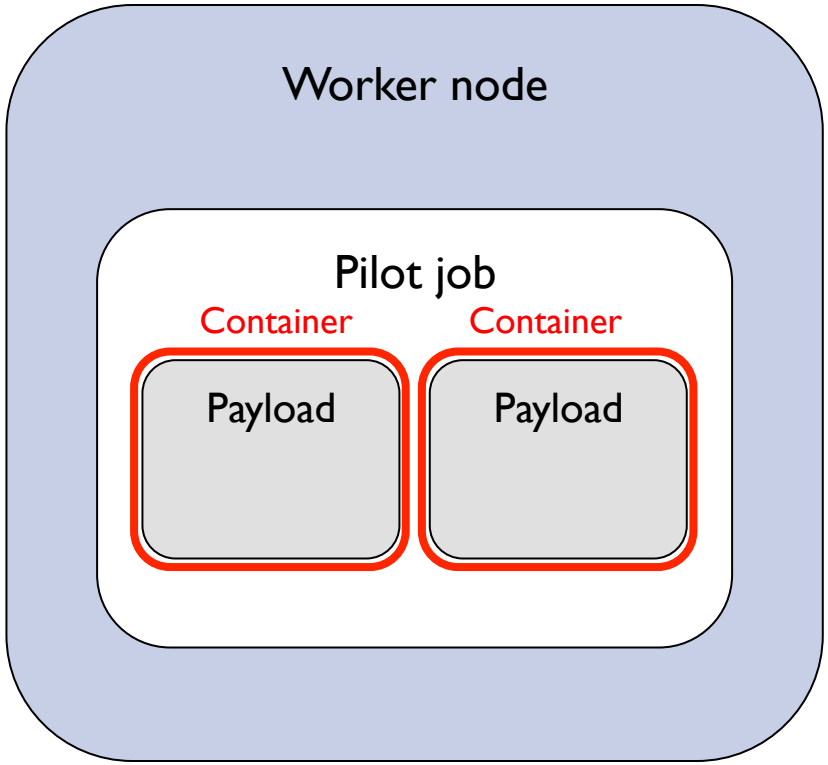
CONTAINER USE CASES (3)

Benefits for the VO

- don't have to rely on the site to provide the appropriate job environments
- a single pilot can run jobs in different environments
- payload jobs isolated from pilot

Example VOs:

- CMS



SINGULARITY

Docker not suitable for containers within pilot jobs

- root access needed to run containers

Singularity is a tool for unprivileged users to run containers

- provides much less isolation than Docker – only:
  - file isolation
  - process isolation
- Docker
  - namespaces (pid, mount, ipc, utc, user, ...)
  - cgroups (cpu, memory, pids, devices, blk io, ...)
  - network isolation
  - Linux capabilities
  - AppArmor & Seccomp security profiles

SINGULARITY AT RAL

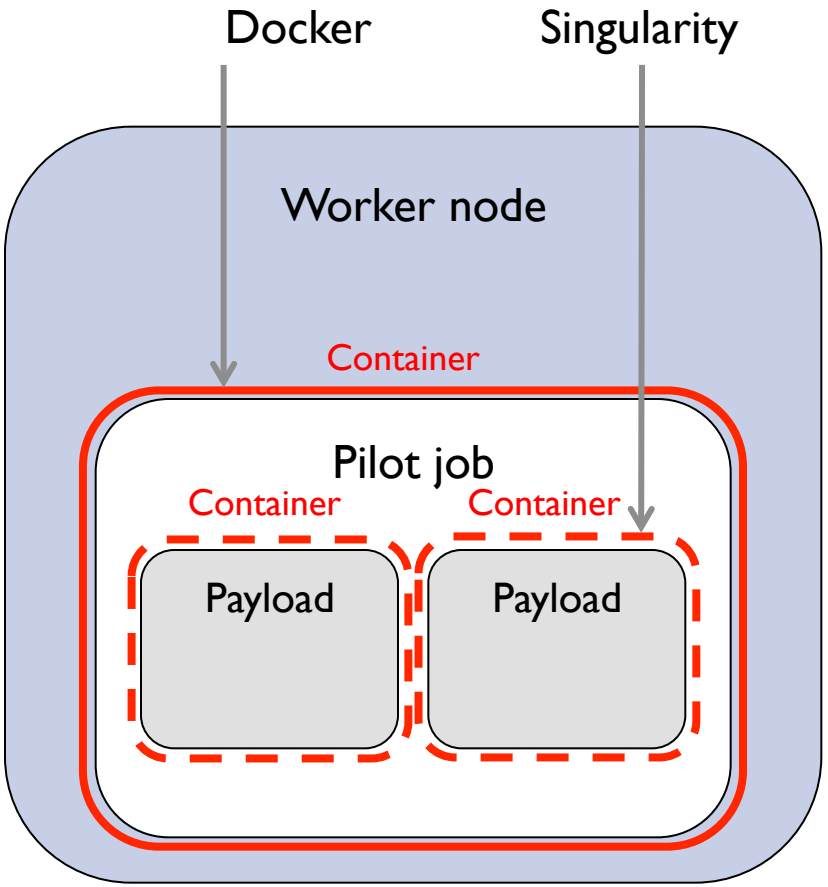
Singularity installed in SL7 container used at RAL

- zero configuration required
- VOs can use it if they want
- enables the CMS model to work

There are two different types of containers with differing levels of isolation

Unfortunately

- CAP_SYS_ADMIN needed for Singularity to work



SINGULARITY AT RAL

Singularity version

- EPEL only has 2.2.1
  - it works but gives an error message when run inside a Docker container as an unprivileged user
- Moved to using 2.3.1 built from source
  - OSG will start providing RPMs

Current status

- CMS glidein factory people have successfully tested Singularity at RAL
- Not yet being used for jobs, only SAM tests...

Summary	ttcms048-7312554.0-lcg1935.gridpp.rl.ac.uk: OK
Details	/bin/singularity exec --home /pool/condor/dir_16654/tmp.1w7NBht2mv:/srv --bind /cvmfs --bind /pool/condor/dir_16654/tmp.1w7N WARNING: Not mounting requested bind point (already mounted in container): /srv WARNING: Container does not have an exec helper script, calling 'echo' directly Hello World summary: OK Singularity test passed

SINGULARITY AT RAL

Future

- With SL7.4 sites don't need to install Singularity, just need to
  - enable unprivileged access to namespaces
  - set user.max_user_namespaces to be non-zero
- VOs can then run Singularity directly from CVMFS

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/7.4_Release_Notes/technology_previews_kernel.html

Unprivileged access to name spaces can be enabled as a Technology Preview

You can now set the `namespace.unpriv_enable` kernel command-line option if required, as a Technology Preview.

The default setting is off.

SINGULARITY & HTCONDOR

HTCondor can be configured to run jobs in Singularity containers

- this is what CMS does within glideinWMS

Does it make sense to do this for a batch system with local users?

- absolutely

Does it make sense to do this at Grid sites?

- possibly not
- this would prevent CMS pilots from being able to use Singularity
  - can only run Singularity within Singularity as root
  - HTCondor won't even let you run jobs as root

SUMMARY

- Containers being used in production at a large scale
  - both for batch jobs and services (xrootd)
- VOs can use either SL6 or SL7
- Singularity available for VOs using SL7

Questions?