Brookhaven National Laboratory

U.S. DEPARTMENT of ENERGY

HTCondor Modernization at BNL

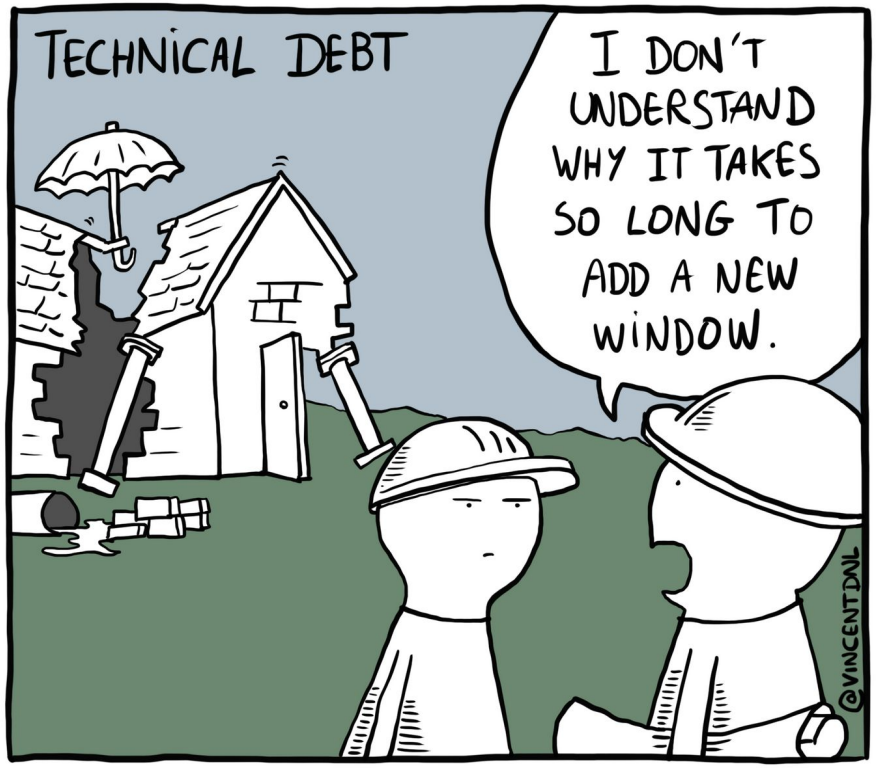
Tom Smith, BNL Scientific Computing and Data Facility (SCDF)

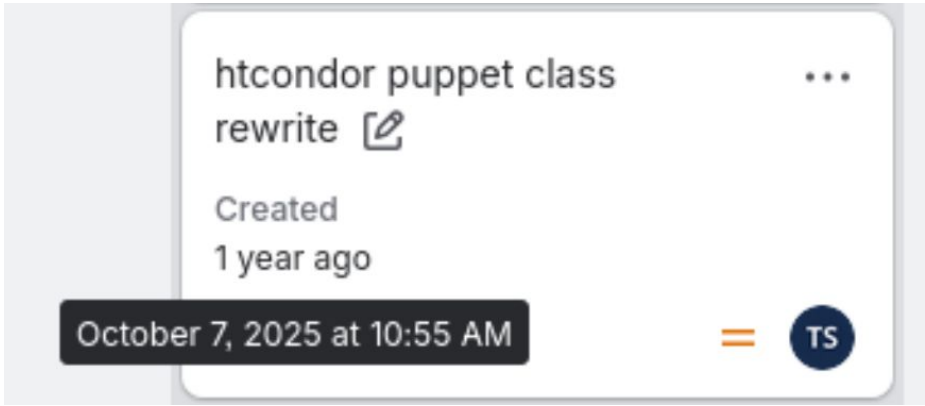
2026 June 9, HTC26

@BrookhavenLab

Motivations

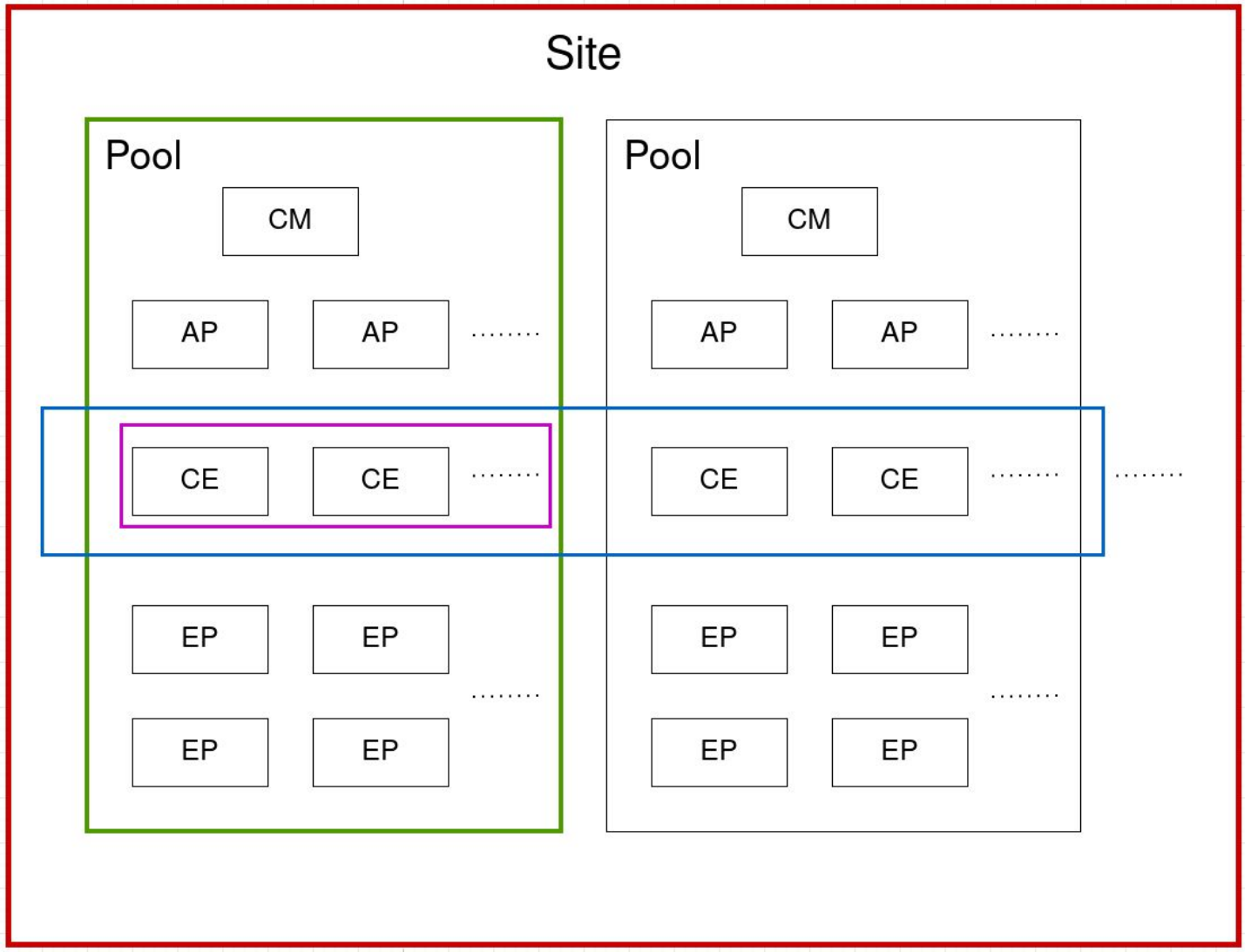
- **Accumulation of technical debt over the course of many years of operations**
  - Continuous uptime creates a focus on “Just fix it enough to keep on running”
  - Culture of Maintenance over Development
  - Adding any feature to the existing codebase was non-trivial
  - Deprecated features and workarounds everywhere
- **No time like the present**
  - sl7 is gone, alma 9 is here
  - Powerful tools at our disposal (Foreman)
  - Need to adapt to a culture of rapid change





HTCondor config refactor

- Condor configs are read in lexicographical order
- This allows a hierarchy of files, overriding values where applicable:
  - condor_config (site)
  - config.d/00_pool (pool)
  - config.d/08_common_ce (CE)
  - config.d/10_local_ce (pool CE)
- Configure things at highest level of commonality
  - a value will be exactly where you expect it to be!



Config refactor cont'd

- Configs gone through line by line
  - some lines were moved to different level (startd configurations don't need to be on site level!)
  - deprecated things were removed
  - some old code existed to provide things that are now 1st class features
  - lines that match the default value were removed (in some cases)
  - values that didn't match the default were scrutinized
    - comments added to explain values or complex logic
  - comments added in general
  - old syntax transforms were transformed
  - IDTokens instead of Password
- Total size of the configs (by lines) reduced by more than half, including new comment lines added!

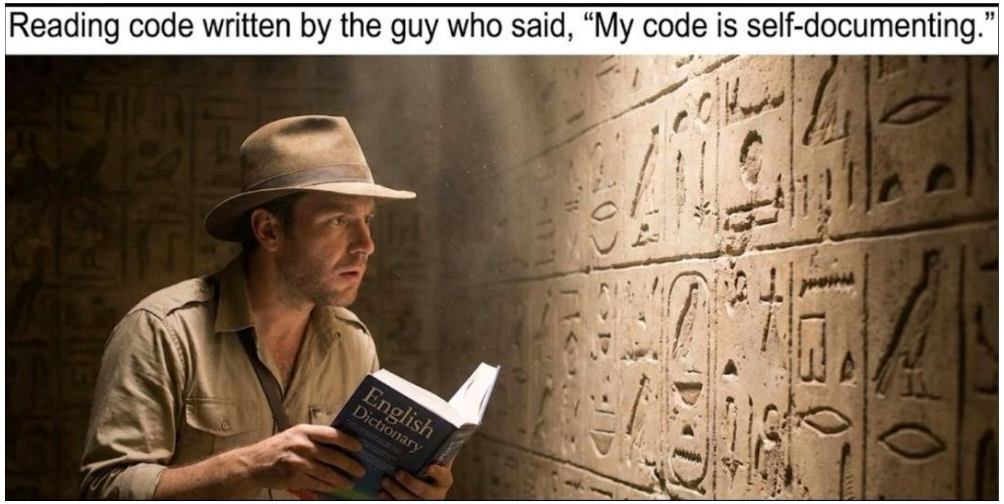
Puppet infrastructure code

- Simplify the structure - avoid a web of interconnected classes
- Simple is better than clever - avoid complex and fragile logic (when possible)
- Written with Foreman / Hiera in mind - make use of host group parameters
- Fail safe - default values reflect a safe value (usually test pool)
- Useful and plentiful comments
  - parameters / variables fully described (use / purpose / origin / defaults)
  - files created by puppet include a header describing their source in git tree
- Written with testing in mind, not an afterthought

```
### Tom - attempt at a complete rewrite of condor class and all that stuff (here we go...) ###
#
# HTCondor main class
# Top level class, common across all condor nodes regardless of specific function
#
# Parameters :
#   condor_version    : version of the condor package desired.
#                       defaults to 'present' if not set
#
#   condor_pool       : the pool that the node is a member of (atlas,sphnx,spool,jupyter,test)
#                       defaults to 'test' if not set (safe default)
#
#   condor_experiment : the particular experiment this node is affiliated with, for example star,phenix,eic,dune,etc
#                         this parameter is only really relevant on APs, and has an impact on 30_experiment template
#                       defaults to the value of $condor_pool if not set
#
#   condor_test_cm    : optional parameter to point host to a different CM. Useful if you want to build nodes with production-
#                         like config, but want to point them to a different CM, like the testpool
#                       defaults to empty string ''
#
#  condor_pool_key    : optional parameter if you wish to specify a /home/condor/pool_key different from the normal one we use for
#                         our pools. Useful for any pool that has portions external to our site, so as not to leak secrets
#                       defaults to empty string ''
#
#   Parameters can be set in hiera or foreman, but foreman has the highest authority
#   Note: parameters set in hiera or as foreman smart class parameters are class scope ($facts['parameter_name'])
#         parameters set in foreman (global, host group, host level) are considered global scope ($::parameter_name)

class htcondor (  $condor_version    = pick($facts['condor_version'],$::condor_version,'present'),
                  $condor_pool       = pick($facts['condor_pool'],$::condor_pool,'test'),
                  $condor_experiment = pick($facts['condor_experiment'],$::condor_experiment,$condor_pool),
                  $condor_test_cm    = pick_default($facts['condor_test_cm'],$::condor_test_cm),
                  $condor_pool_key   = pick_default($facts['condor_pool_key'],$::condor_pool_key) ) {
```

```
├── manifests
│   ├── ap.pp
│   ├── ce.pp
│   ├── cm.pp
│   ├── ep_arm.pp
│   ├── ep_gpu.pp
│   ├── ep.pp
│   ├── init.pp
│   └── misc
│       ├── adstash.pp
│       └── condor_autostart.pp
```

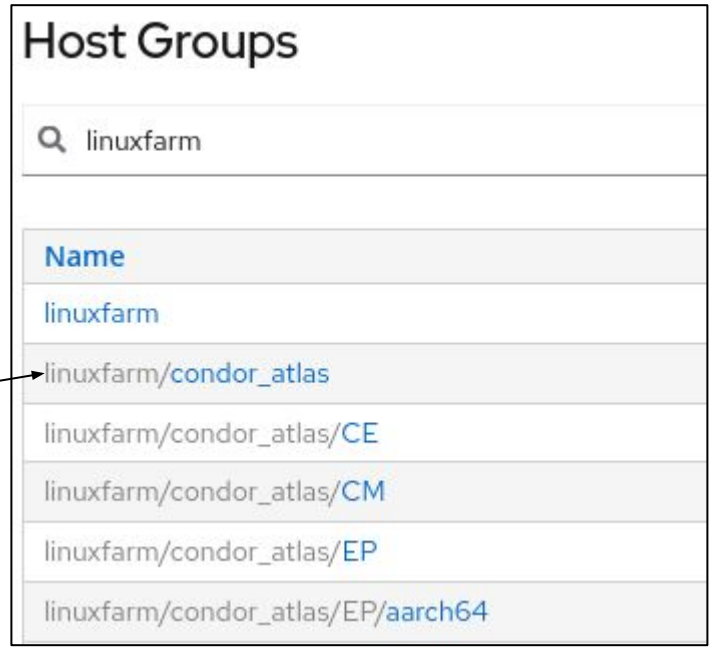


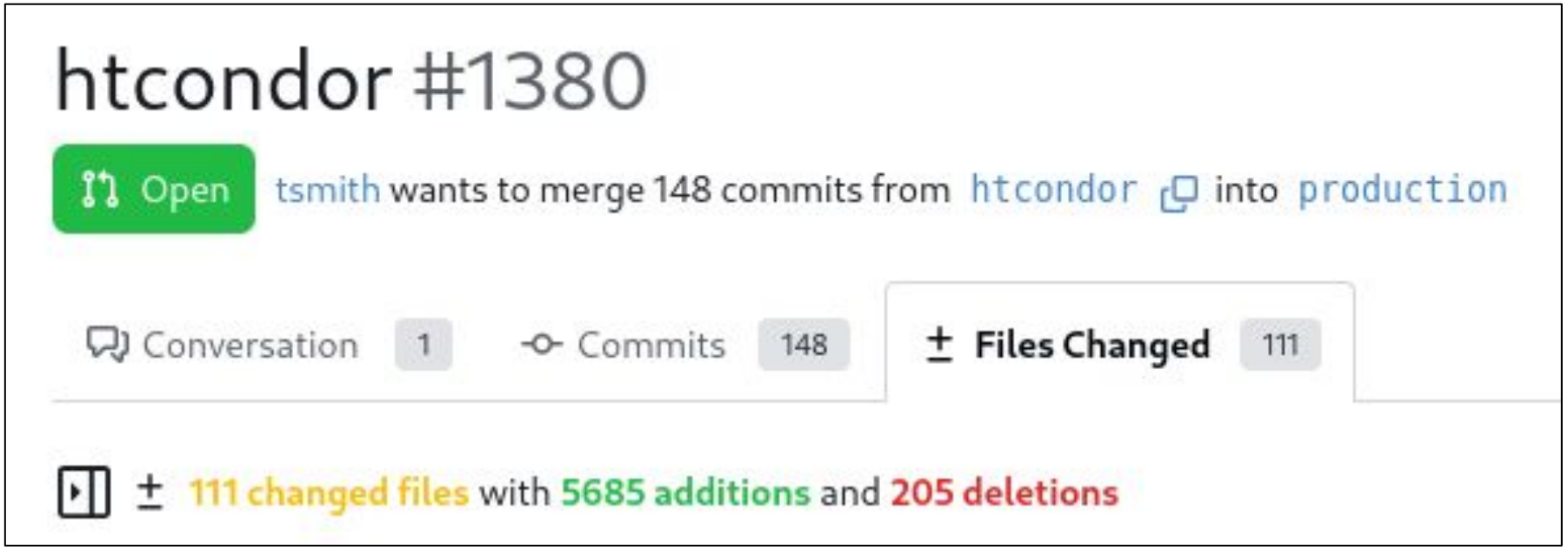

```
htcondor
├── files
│   ├── adstash
│   │   ├── 19_adstash
│   │   └── opensearch.sdcc.bnl.local.pem
│   ├── condor_autostart
│   │   ├── condor_autostart.py
│   │   ├── condor_autostart.service
│   │   └── condor_autostart.timer
│   ├── configs
│   │   ├── ap
│   │   │   ├── common_ap
│   │   │   ├── sphnx_ap
│   │   │   ├── spool_ap
│   │   │   └── test_ap
│   │   ├── ce
│   │   │   ├── atlas_ce
│   │   │   ├── common_ce
│   │   │   └── spool_ce
│   │   ├── cm
│   │   │   ├── atlas_cm
│   │   │   ├── common_cm
│   │   │   ├── jupyter_cm
│   │   │   ├── sphnx_cm
│   │   │   ├── spool_cm
│   │   │   └── test_cm
│   │   ├── common
│   │   │   ├── condor_config
│   │   │   ├── condor_config.old
│   │   │   └── condor_config_source
│   │   ├── ep
│   │   │   ├── atlas_ep
│   │   │   ├── common_ep
│   │   │   ├── sphnx_ep
│   │   │   ├── spool_ep
│   │   │   └── test_ep
│   │   ├── misc
│   │   │   └── 20_aarch
│   │   └── pool
│   │       ├── atlas_pool
│   │       ├── atlas_pool.old
│   │       ├── sphnx_pool
│   │       ├── sphnx_pool.old
│   │       ├── spool_pool
│   │       ├── spool_pool.old
│   │       └── test_pool
│   ├── groups
│   │   ├── sdcc-group-definitions
│   │   └── sphnx-group-definitions
│   ├── htctl
│   └── pool_key
```

Hierarchy and naming of things

- Conform to HTC naming of things (worker -> EP, submit -> AP, gatekeeper -> CE)

- Apply this across the various parts of the infrastructure:
  - HTCondor config files
  - puppet manifests, files, and templates
  - puppet profiles
  - foreman host groups

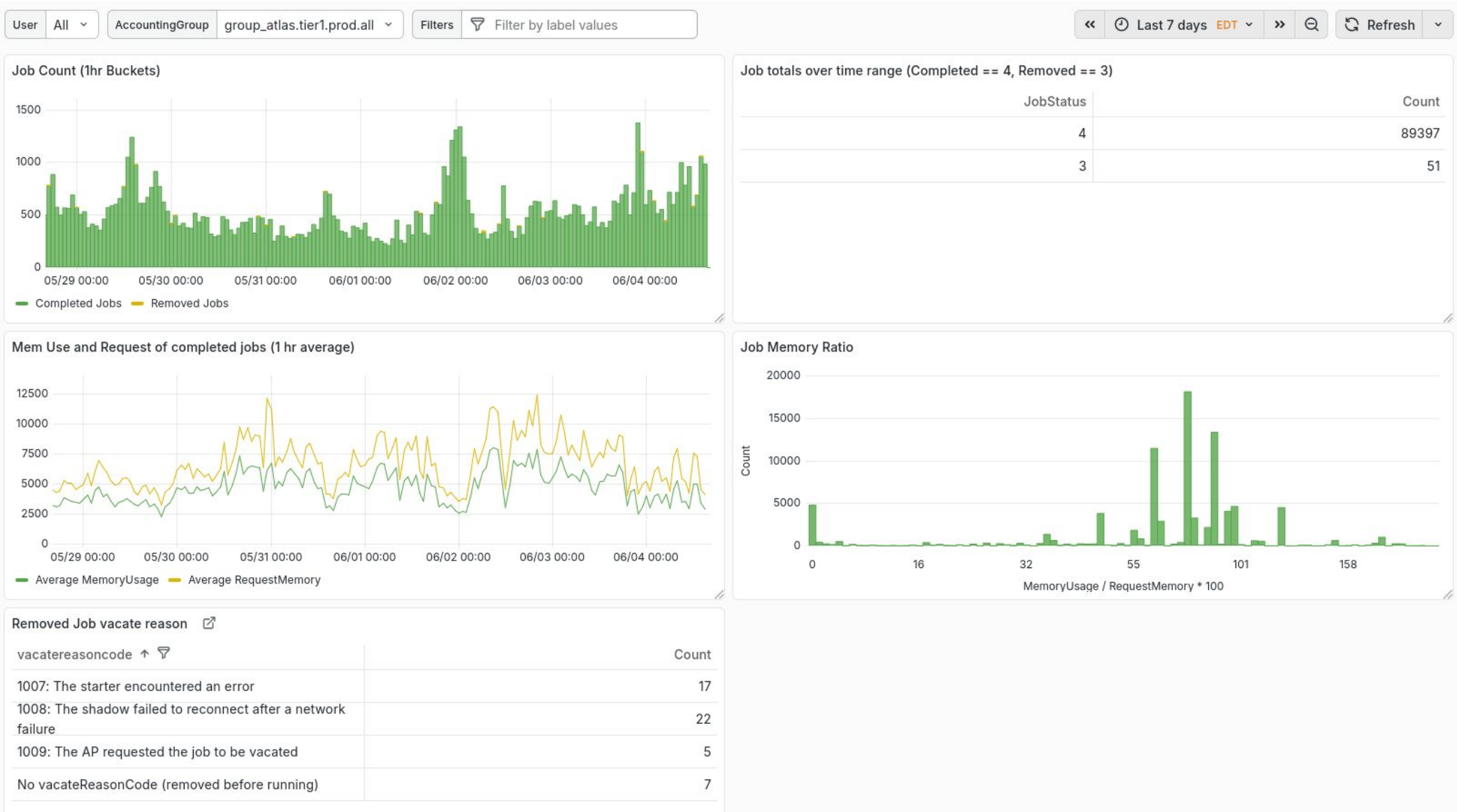




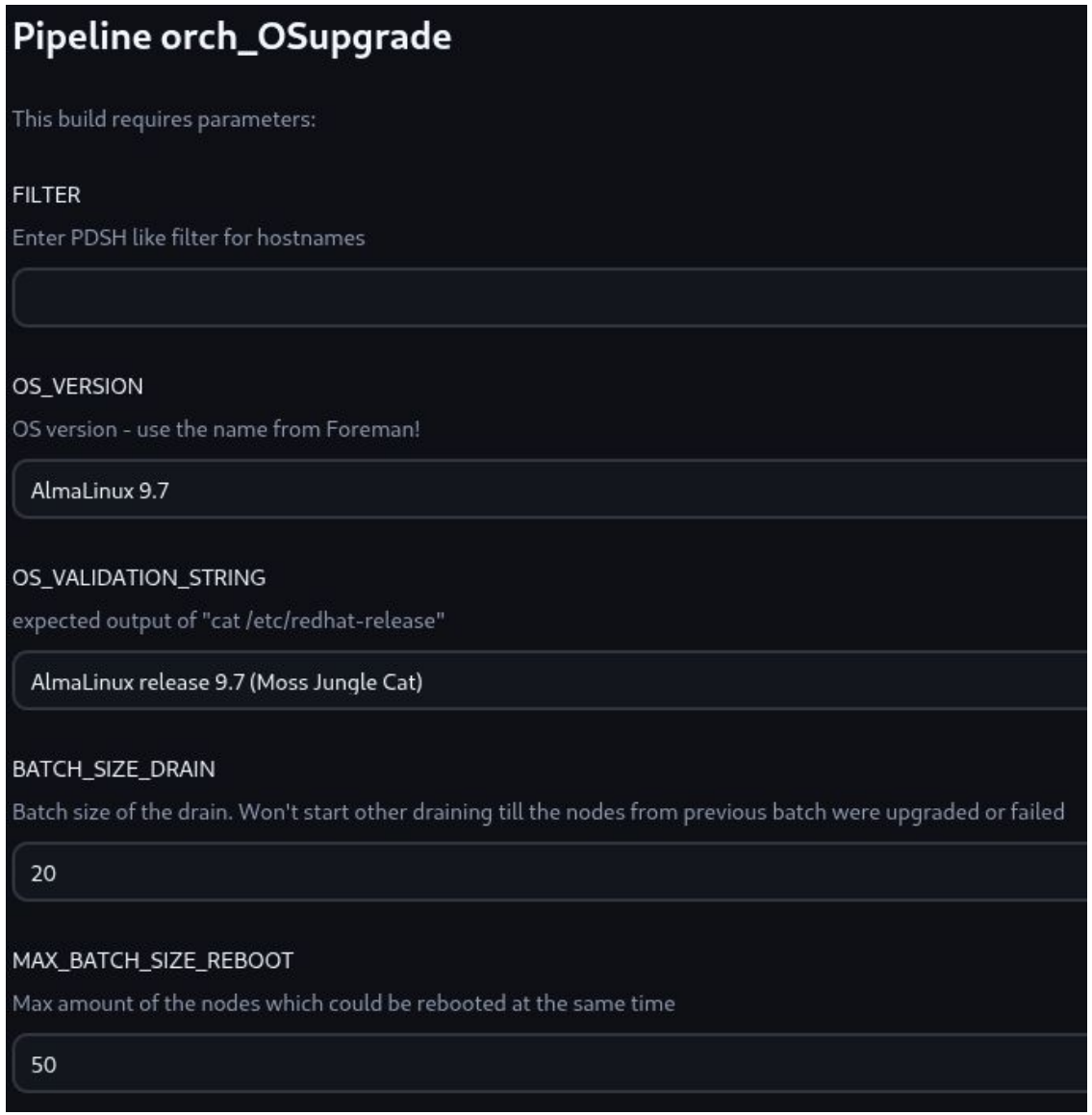
- New code (htcondor class) exists alongside old code (condor class)
  - Easy to test, can migrate pools one at a time
  - If something doesn't quite work out, can always go back while you iterate (No point of no return)
  - Something as big as a full code rewrite can be worked on over long periods of time (Maintenance vs Development)
  - The old code can be retired when the new code is fully deployed

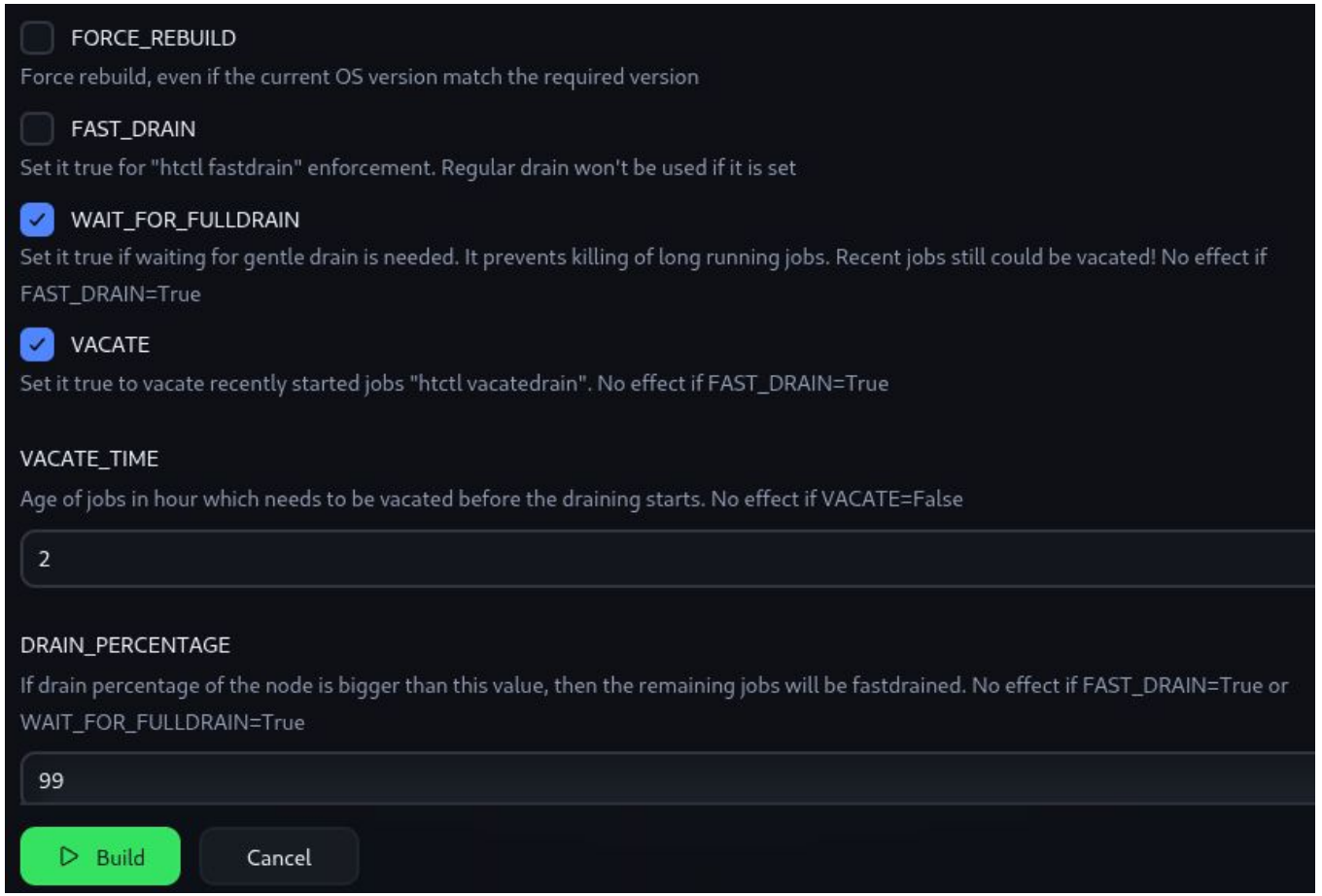
Other fun stuff

- HTCondor adstash -> Opensearch -> grafana

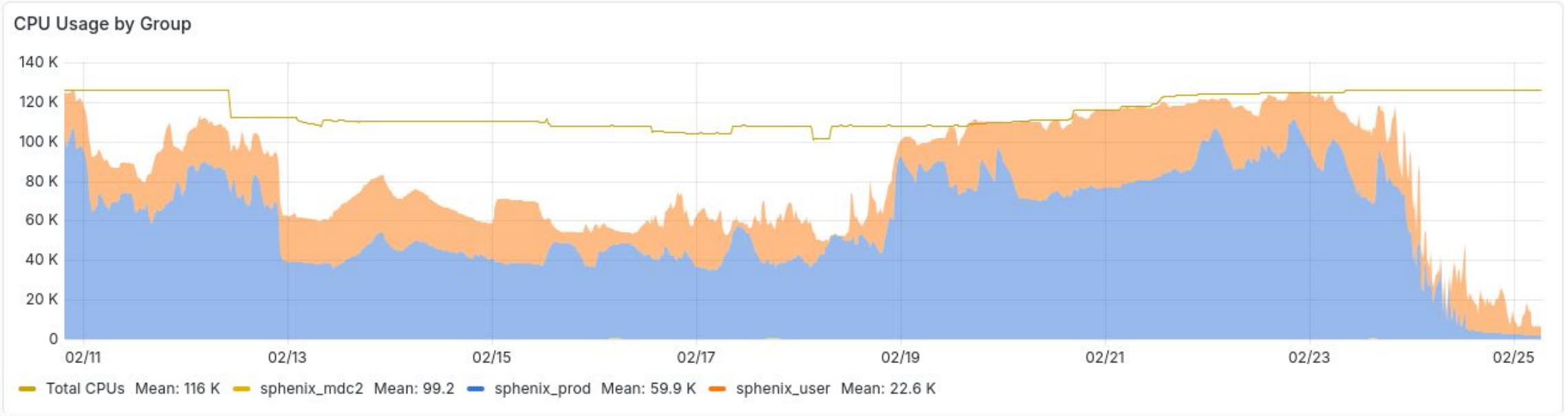


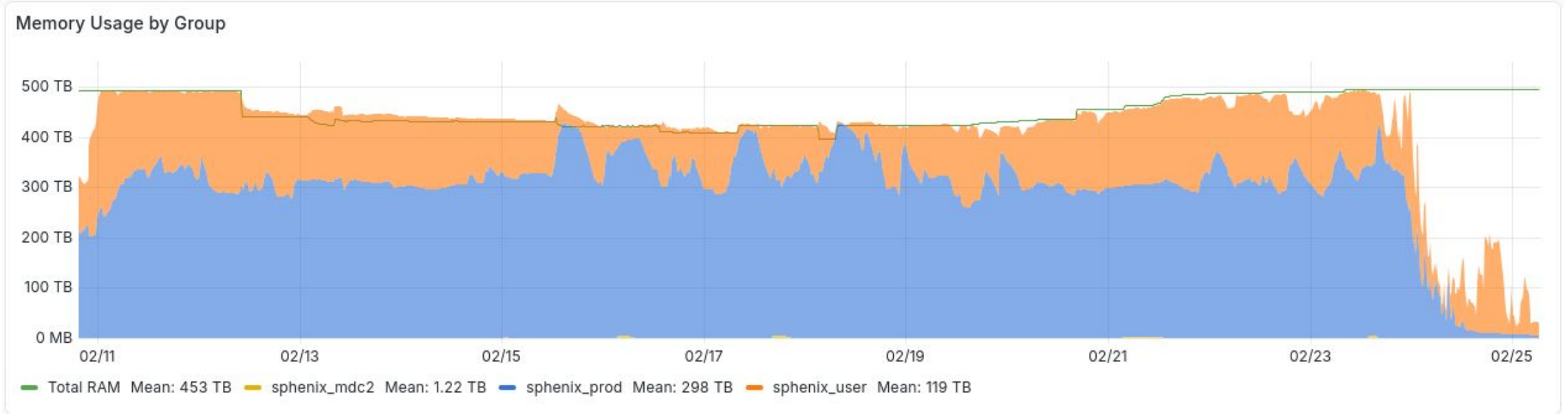
- Jenkins drain / rebuild / start pipeline (Thanks Oszkar!)





- Jenkins drain / rebuild / start pipeline (Thanks Oszkar!)





Brookhaven National Laboratory

- htctl (HTCondor control)
  - terminal wrapper program
  - written in python
  - provides quality of life / additional functionality for common condor functions
    - pre-flight checks for EP
  - some advanced features not present in normal condor
    - alternative draining strategies
    - AP backup / restore

```
--------- htctl help ----------------------------------------------------------
  htctl is a wrapper program for common condor actions performed on the linux farm

  Subsystem is defined in /etc/htctl.conf, or assumed ep if file not present

  Subsystems: cm = Central Manager,            ce = Compute Entrypoint (gatekeeper)
              ap = Access Point (submit node), ep = Execution Point (worker)

  Usage: htctl <action> <optional sub-action>
  Note - not all actions have sub-actions

--------- Actions for all condor subsystems -----------------------------------
  start         - Starts HTCondor daemons
  stop          - Stops HTCondor daemons
  restart       - Restarts HTCondor daemons
  status        - Shows status of HTCondor daemons

--------- Actions for EP (worker) only ----------------------------------------
  enable        - Enables EP to accept new jobs
  disable <message>
                - Disables EP from accepting jobs with optional message in /NOCONDOR
  fullstart     - On an EP, same as running "htctl start" followed by "htctl enable"
  fs            - Same as "fullstart"
  drain         - Drains EP gracefully and stops daemons (respects MaxJobRetirementTime)
  vacatedrain <hours>
                - Drains EP gracefully and vacates jobs started within last <hours>
  fastdrain     - Drains EP fast and stops daemons (ignores MaxJobRetirementTime)
  canceldrain   - Cancels the drain of an EP and resumes operation

--------- Actions for AP (submit / schedd) only -------------------------------
  backup        - Copies log and spool directories for later restoration after rebuild
  restore       - Retreives aforementioned backups and puts them back
  clear_backup - Clears any backups from the NFS backup directory for this host

-------------------------------------------------------------------------------
```

Summary

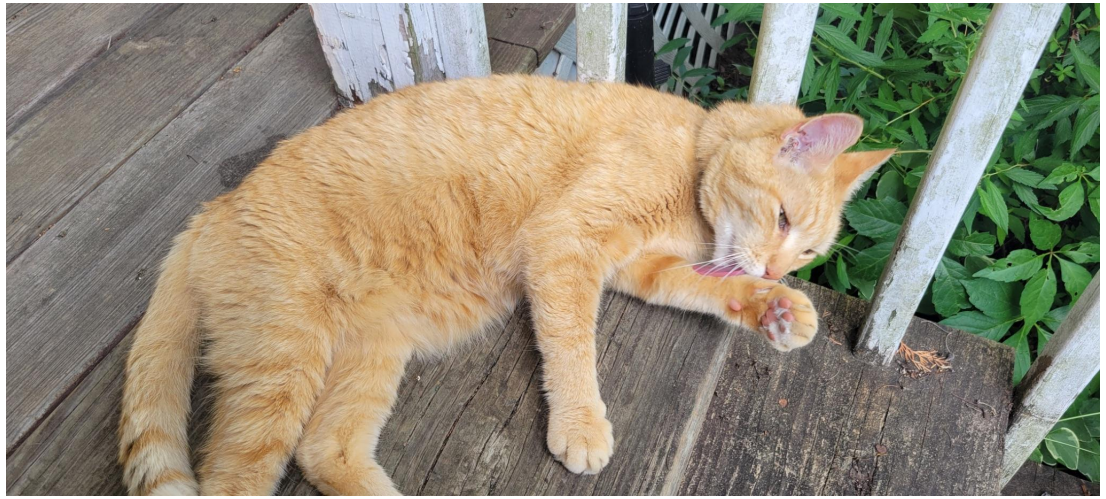
- Respect the classics - don’t change just for the sake of change
- Refactor all the HTC configurations to a modern standard
- Rewrite all of the infrastructure code related to the HTC system
- Standardization across the board of hierarchy and the naming of things
- Other fun stuff

Acknowledgment / Any questions ?

Thanks to:
- BNL SCDF Staff
- ATLAS community

Special Thanks to:
- SCDF IT Fabric team
  - Matt Cowan, Costin Caramarcu, Kevin Casella, Oszkar Tarjan, Zhihua Dong, Shigeki Misawa
- HTCondor team

Very Special Thanks to:
- TJ (John Knoeller) from HTCondor team



Bonus Slides

Brief Overview of our HTC system

(BNL Site Report in 1 slide)

	ATLAS Tier 1 pool	sPHENIX pool (new!)	Shared pool
Worker nodes (EPs)	~320	~1200	~490
Logical cores	32k	133k	41k
HEPscore23	~460k	~2.05M	~516k
Condor CEs	4	0	4
Submit nodes (APs)	0	12	35

ATLAS EXPERIMENT

sPHENIX

STAR

Belle II

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

PHENIX

…and more!

It was a busy year..

- HTCondor major version upgrades 23.0 -> 24.0 -> 25.0
- Major refactor of HTCondor related puppet infrastructure code, and configurations

Condor adstash / Opensearch