

AF Debrief and Planning Meeting Notes

- Thoughts from Lincoln to guide this discussion: [pdf](#)
- Difficult to plan details until we know the scrubbing outcome
- SSH still dominant access modality
 - 36% of UC AF users using VSCode
 - People probably scp'ing files locally between laptop and facility
 - Eventually will likely need to get 2FA implemented
- Users will start using Code assisting tools and LLMs more
 - Will cost be an incentive in the future to develop local LLMs (e.g. oLlama)
- AF-specific Jupyter plugins development
 - E.g. standardize Rucio plugin across facilities
 - Fengping has already invested significant time into getting Binderhub working well
 - At SLAC, each working group has their own customized environment
 - Users in "ML space" have their own specific requirements and not really ATLAS tools
 - Other users need ATLAS specific interactive environment - happier with ATLAS standard provided images
 - Does UC scan the containers?
 - Maybe done in Harbor?
 - What about privilege escalation with shared volumes?
 - Could be caught after the fact by CrowdStrike at UC, but not explicitly prevented otherwise
 - Have not had to ban any users yet, "scolded" a handful
 - Another example: Jupyter "wall" plugin
 - Do we have sufficient personnel resources to support plugin development and maintenance?
 - Could some of this be driven by other experiments (e.g. Belle, FCC, etc. at BNL)
 - SLAC provides "[OpenOnDemand](#)" platform and users set up their own local jupyter environments
- Federation
 - Base all facilities on CERN identity
 - Should be made "trivial" if in ATLAS VO
 - Some potential restrictions at DOE labs
 - Would be nice to have a SSO portal
 - Federation more-or-less "solved" at BNL, how to get access at SLAC?
 - How do we solve cross-site identity, particularly for storage access?
 - How to federate home areas and data areas?
 - Persistent storage at the site of origin (may need to ship files back)
 - Leverage Condor/Glide-ins to move data around?

- Does caching really make sense for AFs?
 - Will not solve everything
- Stretched Clusters?
 - Different approaches exist - technically interesting, get stuck at policy level
 - How to stretch Jupyter resources across sites?
 - Batchspawner to send jupyter kernel to external site?
 - Dask job-queue spawner?
 - Run on secure VPN?
 - We know how to do Condor flocking, should start with that?
 - How to use Dask at external site?
 - Use local harbor proxy cache to provide container environments
 - Need to solve data access
 - Need to solve security/firewall issues
 - How do users know where to send the Dask jobs? Condor decides this (using glide-ins)?
 - What about k8s native solutions, like kueue?
 - Would need to provide dedicated k8s compute resources, could not use Condor resources.
- What approach to prioritize?
 - Jupyterhub at UC with overlay batch cluster, using batchspawner to send kernels to external sites
 - No need to open specific ports
 - Only outbound connections?
 - Dask to local resources
 - Containers from Harbor proxy?
 - Need to solve identity and authorization
 - Probably don't need local accounts in this scenario, only token for data access
 - Need to be able to generate, transport and refresh user tokens
 - How to handle GPU requests?
 - Leverage k8s for a development platform at UC
 - Shared development of distributed AF portal
 - Enforce use of ArgoCD? Flux?
 - Will need some root access for development and support (troubleshooting by stop/start Condor, investigating logs, etc.)