



Analysis Facilities activities in (some) DOE multi-purpose computing centers

Doug Benjamin (BNL), **Burt Holzman** (FNAL),
Ofer Rind (BNL), Wei Yang (SLAC)

March 25, 2022

In partnership with:



DOE National Laboratories

Office of Science Laboratories

- 1 Ames Laboratory
Ames, Iowa
- 2 Argonne National Laboratory
Argonne, Illinois
- 3 Brookhaven National Laboratory
Upton, New York
- 4 Fermi National Accelerator Laboratory
Batavia, Illinois
- 5 Lawrence Berkeley National Laboratory
Berkeley, California
- 6 Oak Ridge National Laboratory
Oak Ridge, Tennessee
- 7 Pacific Northwest National Laboratory
Richland, Washington
- 8 Princeton Plasma Physics Laboratory
Princeton, New Jersey
- 9 SLAC National Accelerator Laboratory
Menlo Park, California
- 10 Thomas Jefferson National Accelerator Facility
Newport News, Virginia

Other DOE Laboratories

- 1 Idaho National Laboratory
Idaho Falls, Idaho
- 2 National Energy Technology Laboratory
Morgantown, West Virginia
Pittsburgh, Pennsylvania
Albany, Oregon
- 3 National Renewable Energy Laboratory
Golden, Colorado
- 4 Savannah River National Laboratory
Aiken, South Carolina

NNSA Laboratories

- 1 Lawrence Livermore National Laboratory
Livermore, California
- 2 Los Alamos National Laboratory
Los Alamos, New Mexico
- 3 Sandia National Laboratory
Albuquerque, New Mexico
Livermore, California



- 17 national labs
- 4 with large HEP funding: **Fermilab, Brookhaven, SLAC, Lawrence Berkeley**
- Will highlight work at first 3 today (but we should have LBNL at next forum!)

Analysis Facilities at National Labs

- Pre-existing computing facilities
 - **Long history** of providing user analysis facilities

RHIC Computing Facility (RCF)

➤ Organizationally established in 1997

The first scientific non-data computer acquisition by the Laboratory occurred in 1970. About \$500K had been allocated for the acquisition of a medium-sized computer to service the bubble-chamber film-measuring and analysis needs generated by FAF. The

- Today we will focus on the **AFs in development** (support fast columnar analyses) that complement our existing AFs
- Security
 - As .gov sites, labs are generally subject to increased scrutiny and oversight
 - Certification of software / path to FedRAMP certification is helpful
- Multi-tenancy
 - Serve broad communities, not single experiments (and not necessarily just HEP)

Fundamental principles:

- Create a user-oriented analysis facility based on our own experiences supporting scientists on traditional grid technologies.
- Explore, deploy and collaborate on industry-level technologies and strategies for optimizing data analysis partly in preparation for HL-LHC and upcoming experiments with large data demands such as DUNE.
- Foster collaboration with HEP experiments in order to better understand science analysis needs and provide computing solutions accordingly.

Secure

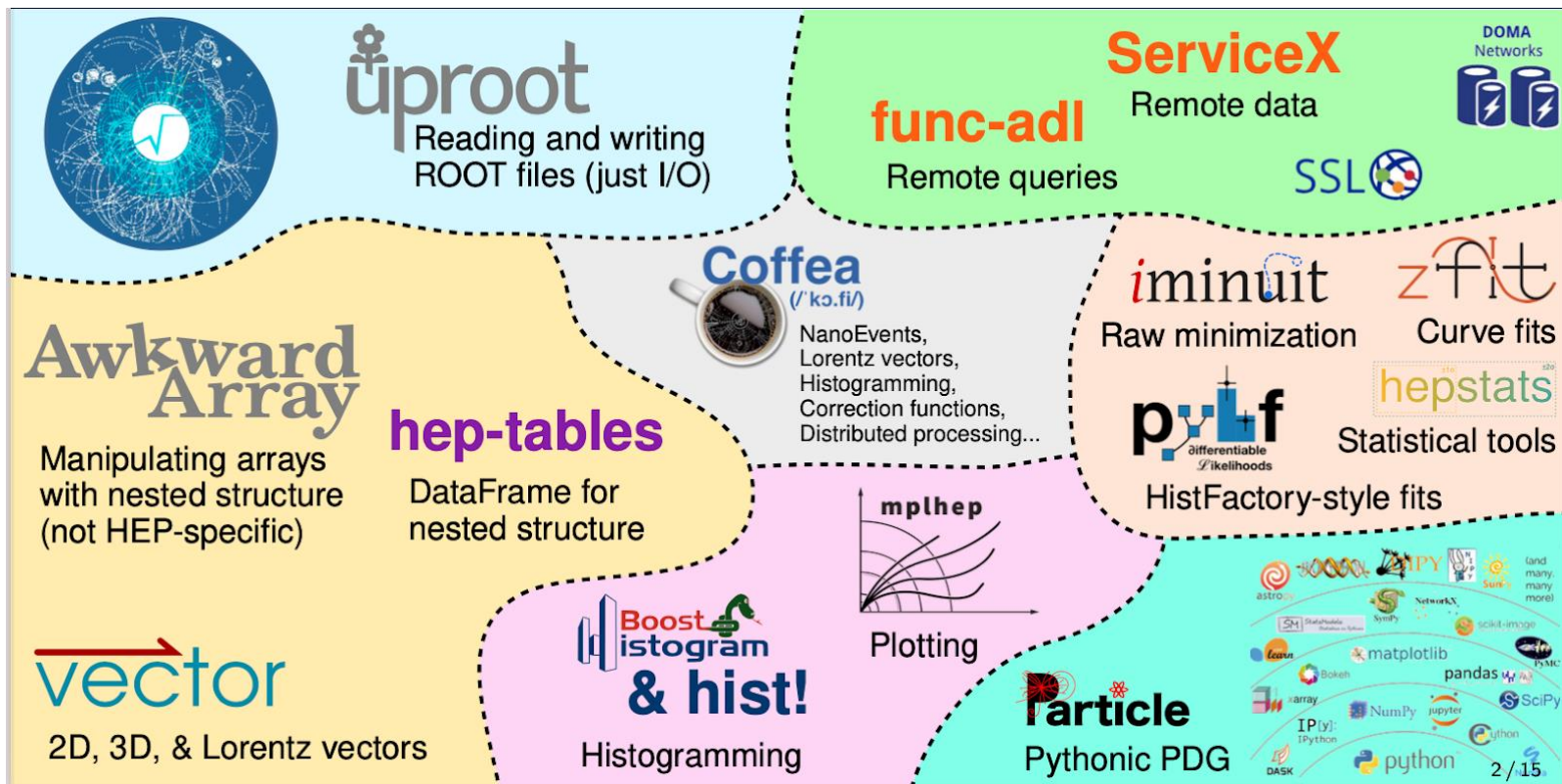
Integrated & functional

Multi-VO

DevOps (operational
sustainability)

Active collaboration

Analysis Systems Ecosystem



Brookhaven National Laboratory

Overview of US ATLAS Analysis Facilities

US ATLAS has three shared Tier 3 analysis facilities providing software & computing

- Resources that fill gaps between grid jobs and interactive analysis on local computers
- Interactive ssh login, local batch, non-grid storage, LOCALGROUPDISK, PanDA
- GPU resources available
- Documentation: <https://usatlas.readthedocs.io/projects/af-docs/en/latest/>

DOE

Launched Oct 2021



BNL Facility

~2000 cores, part of a larger shared pool,
opportunistic access up to 40k cores
User quota: 500GB GPFS plus 5TB dCache
~200 users



SLAC Facility

~1200 cores, part of larger shared pool,
opportunistic access up to 15k cores
User quota: 100GB home, 2-10TB for data
~100 users



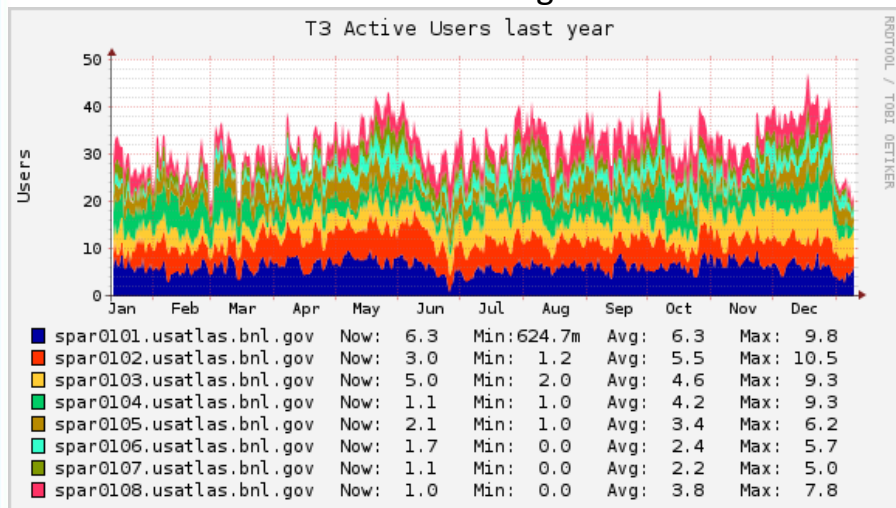
U Chicago Facility

~1000 cores, co-located and close
integration with MWT2
User quota: 100GB home, 10TB for data
~50 users

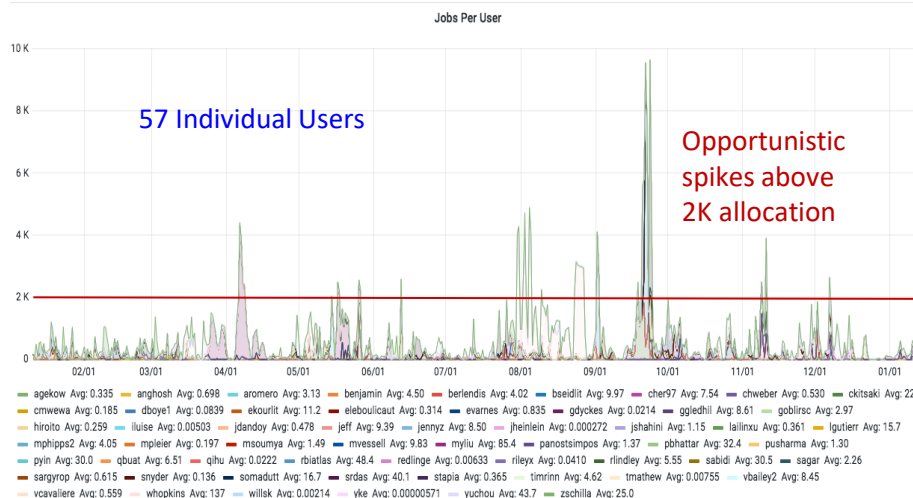
BNL Tier-3

"Traditional" ssh + interactive/batch usage

Total Number of Interactive Users across BNL Tier-3 Login Hosts



Number of Running Condor Jobs Per Tier-3 User



BNL SDCC Jupyterhub Portal



SDCC JupyterHub

The SDCC offers multiple JupyterHub instance and back-end combinations for different users and accounts. Choose the appropriate option from the instances displayed below.

[More information](#) [Questions and support](#)



[Run Jupyter](#) [Manage Sessions](#)

Click "Run" above to navigate to jupyter and choose to launch a notebook via HTC, HPC or directly on our condor pool.
Click on the "Manage Sessions" button to be able to start and stop an already running session or manage named sessions

Custom Jupyterhub interface to multiple resources backends with account-based access control:

- Dedicated set of Jupyterhub nodes
 - Dask scheduling onto HTCondor pools
- Notebook spawning onto HTCondor pools
- Notebook spawning onto HPC cluster via Slurm
- Shared environment with SLAC on BNL cvmfs server

jupyterhub Home Token Admin rind Logout

SDCC Jupyter Launcher

[HTC / Standard](#) [HTCondor Pool](#) [IC / HPC Systems](#)

Run a notebook on a standard interactive HTCondor submit-node

Select JupyterLab Environment

- Default
- Default HPC
- USATLAS

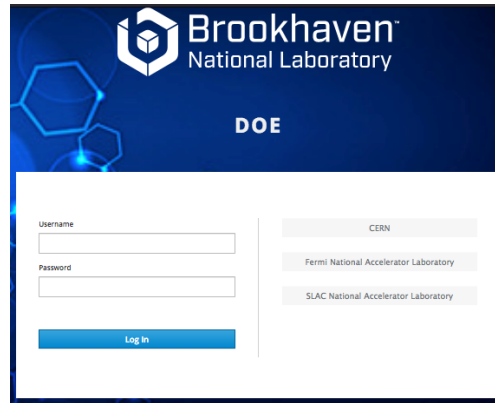
Singularity Container

- None
- Custom

Start

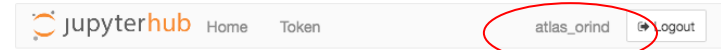
Federated Access

- Preproduction Jupyterhub with federated login in testing at BNL (accepts CERN, SLAC, FNAL logins)
 - Access for all ATLAS users collaborating with a US institution
 - Simple form for creating lightweight AF account
 - DOE lab policy requires multi-factor auth for interactive access

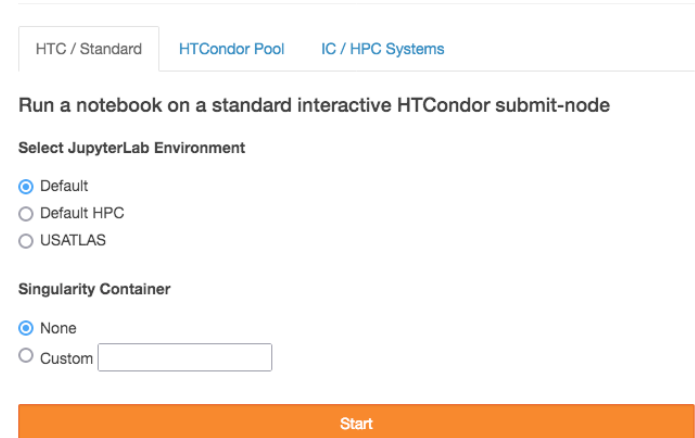


The screenshot shows the Brookhaven National Laboratory DOE login interface. It features a dark blue header with the Brookhaven logo and the text 'DOE'. Below the header, there are input fields for 'Username' and 'Password', and a 'Log In' button. To the right of the input fields, there are three buttons for different institutions: 'CERN', 'Fermi National Accelerator Laboratory', and 'SLAC National Accelerator Laboratory'.

Automatic account creation based on CERN login



SDCC Jupyter Launcher



The screenshot shows the SDCC Jupyter Launcher interface. It has a header with three tabs: 'HTC / Standard', 'HTCondor Pool', and 'IC / HPC Systems'. Below the tabs, there is a text prompt: 'Run a notebook on a standard interactive HTCondor submit-node'. Underneath, there is a section for 'Select JupyterLab Environment' with three radio buttons: 'Default' (selected), 'Default HPC', and 'USATLAS'. Below that is a section for 'Singularity Container' with two radio buttons: 'None' (selected) and 'Custom' with an adjacent input field. At the bottom, there is a large orange 'Start' button.

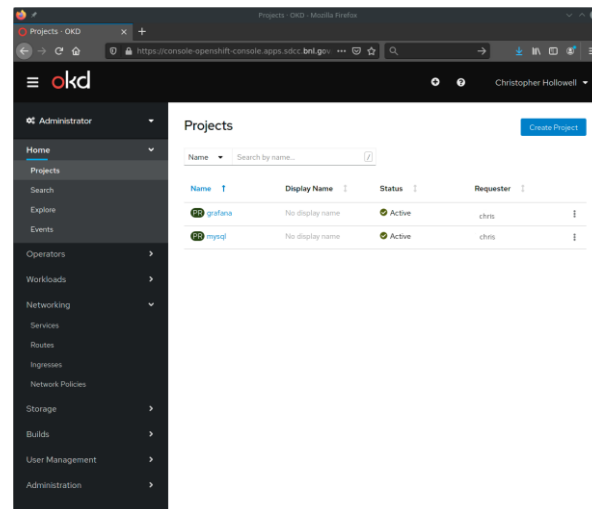
OKD at BNL SDCC

OKD provides a platform for container orchestration, similar to Kubernetes (k8s)

- Community-supported release of Openshift
- Allows for simplified deployment of services via helm charts and Openshift templates
- Contains numerous security enhancements out of the box vs k8s
 - Users are never root by default

OKD 4.7 cluster online at SDCC

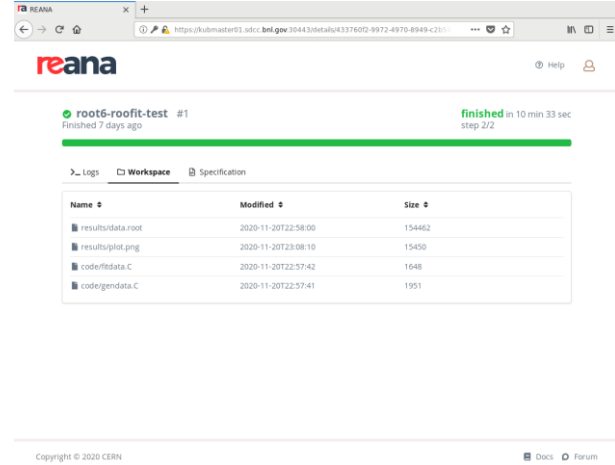
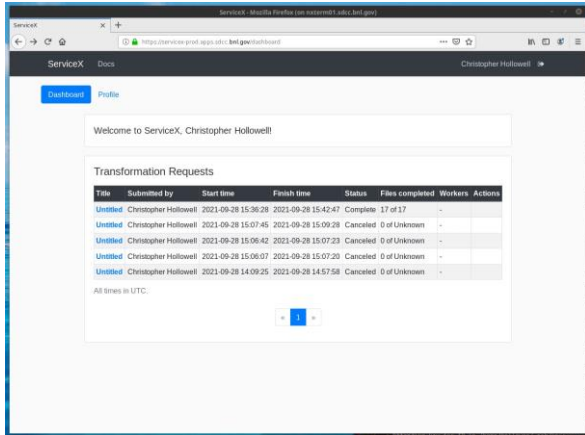
- 4 user nodes - 20C/40T, 128 GB RAM
- 15 TB NMVE storage (NetAPP 250)
- Console authentication tied to our keycloak IDP
- Currently only accessible from inside BNL



AF at BNL SDCC on OKD

REANA testbed

- User testing in progress
 - Web interface currently accessible via SSH tunnel/SOCKS proxy
- Can interface and submit container jobs to SLURM on the HPC cluster



ServiceX test deployment

- Authentication using SDCC IDP
- Currently only available from within the SDCC network
- Working with developers on integrating needed changes to containers and helm charts to support OKD upstream
 - Currently only in our modified deployment

SLAC National Accelerator Laboratory



- SLAC AF established in 2017
 - Serving US ATLAS users and their international collaborators
 - Initially inherit hardware from US ATLAS Western Tier 2. Most of them are retired now.
 - Added new storage (800TB), CPU (512cores)
 - adding a GPU node
 - 40-70 active users (depend on how/what to count)
- Services:
 - Traditional login/batch: ssh, SLURM, user home and private data space
 - Grid services: RSE & Xcache (for users to access official ATLAS data products)
 - Jupyter/DASK: a user expandable Jupyter environment, back by SLURM
 - Kernels to access ATLAS release (needs updates), Tensorflow/Keras, Rapids
 - These kernels are shared with BNL via CVMFS.
 - Opportunistic access to a large pool of CPUs and GPUs
- Support:
 - SLAC specific document site and email/ticket
 - US ATLAS AF document site in [github/readthedocs.io](https://github.com/readthedocs.io)
 - Mattermost channel/e-mail support, to be moved to a central US ATLAS Discourse service

SLAC Analysis Facility

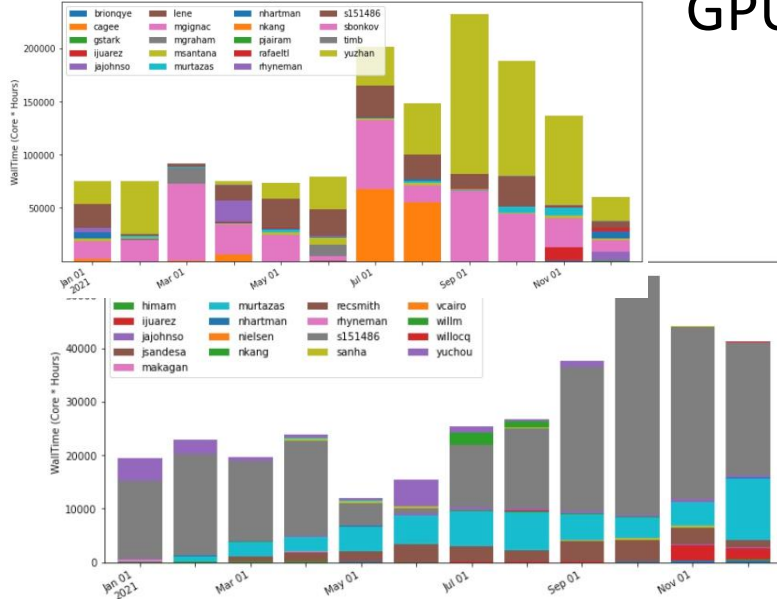


- Heavy Jupyter/DASK usage
 - More Jupyter/DASK users than batch users in 2021
- Users are utilizing SLAC's large pool of GPUs

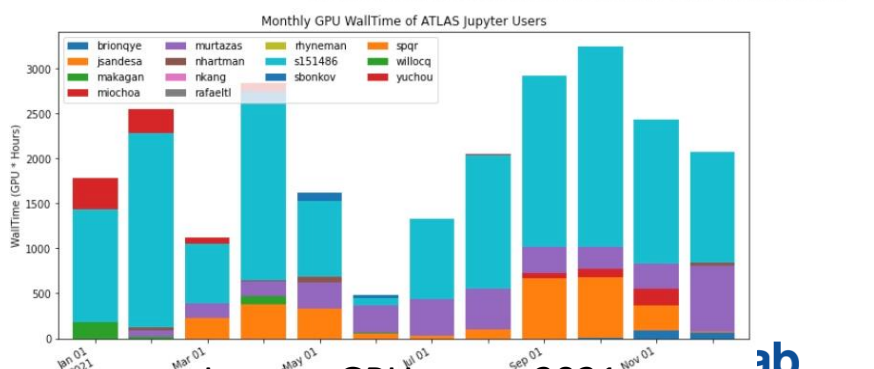
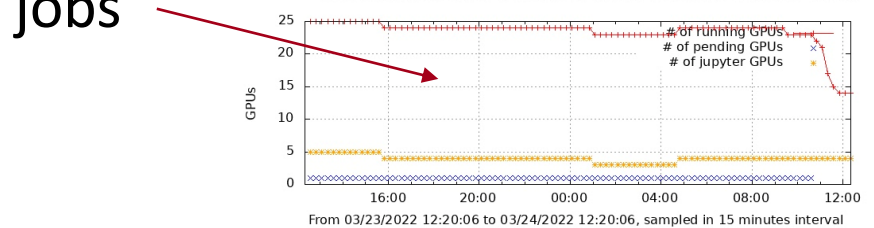
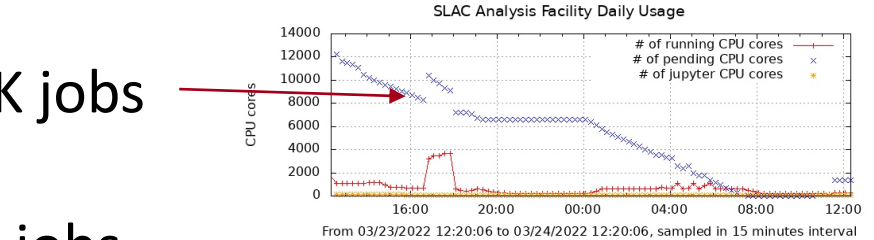
DASK jobs

GPU jobs

Batch CPU usage 2021



Jupyter CPU usage 2021



Jupyter GPU usage 2021

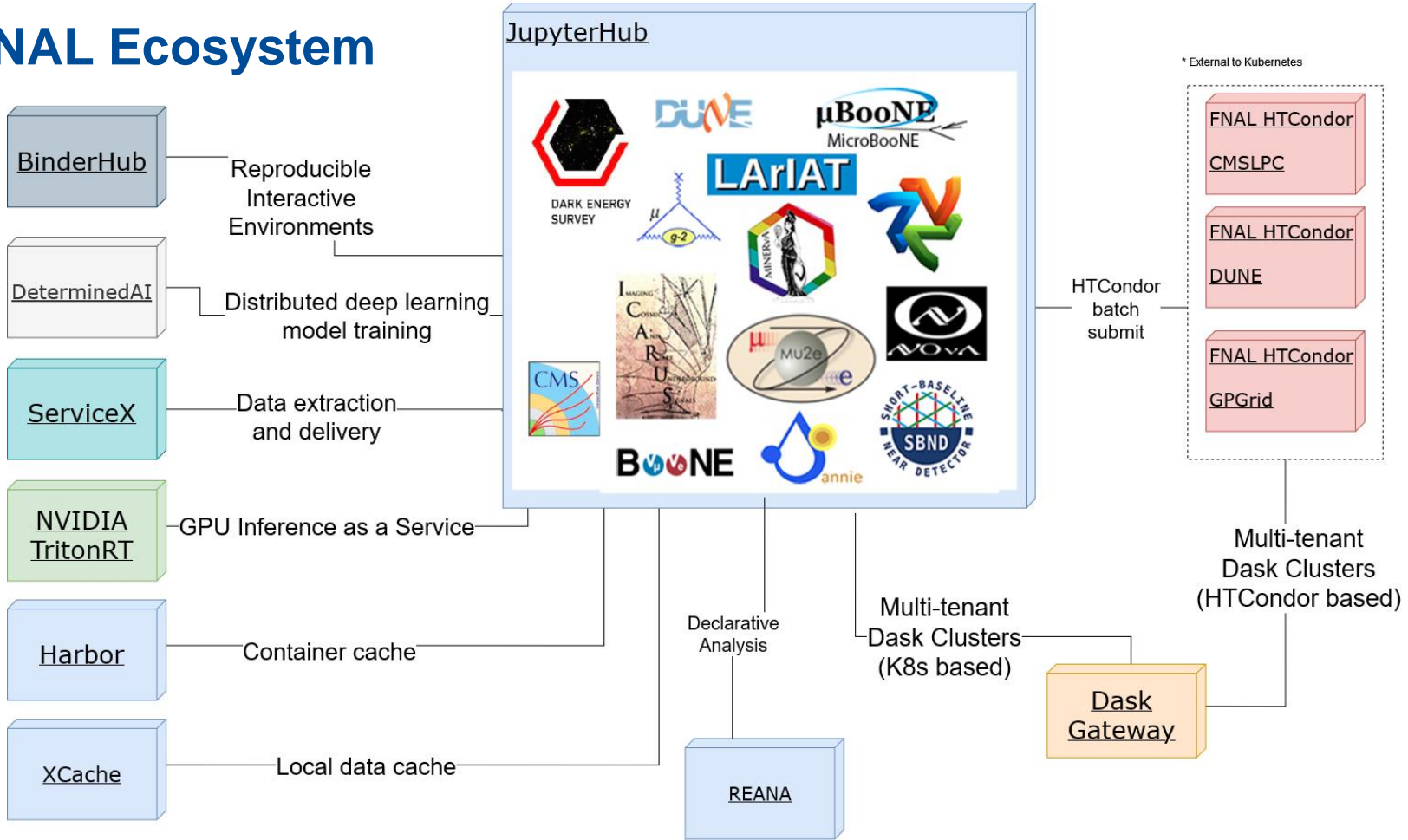


- Modernizing the SLAC overall computing infrastructure
 - SLAC AF is embedded in the SLAC computing infrastructure. So SLAC AF will benefits
- Storage:
 - Moving from a mixed of AFS/GPFS/Lustre for \$HOME to Weka SSD based home.
 - Have very positive experience with Weka posix home at LCLS
 - Hopefully this will significantly boost the start up speed for Python
 - Not completely settled for data space.
 - Object store is preferred by Rubin. But Weka may provide a tiered posix with hidden S3 backend.
- Jupyter environment:
 - Currently using OpenOnDemand to spawn Jupyter on SLURM nodes (as SLURM jobs).
 - Good: utilize SLURM scheduling
 - Bad: exclusive allocation of GPUs is expensive (even for a 400+ GPU cluster)
 - Maybe looking at traditional methods: allowing several users to freely login to a few shared GPU nodes.
- Authentication/Authorization:
 - Looking at Federated Access. A grey area with a policy side and a technical side. On tech side:
 - Easy for web based access but hard for SSH
 - Experimenting ways to integrate SSH
- K8s for future service
 - Like will use k8s to host most of the services. VM are still needed in some cases
 - This is mostly driven by Rubin but ATLAS and other will benefits from the experience/expertise.



Fermi National Accelerator Laboratory

FNAL Ecosystem



* External to Kubernetes

Canonical Jupyterhub Screenshots

jupyterhub

Fermilab

Welcome to JupyterHub @ the Fermilab Elastic Analysis Facility

Use your Fermi SERVICES domain credentials to log in

The OpenShift Kubernetes cluster will have upgrades installed on 2/16 from 10:00 – 11:00 am. As a result, Jupyter notebook pods may be killed and re-launched during this time.

If you have an existing environment and want to run it as a notebook, go to EAF BinderHub

EAF is in beta testing phase. This is the point where we need your help:

- Please note that GPU availability is on a first come, first serve basis. If you request a notebook with a GPU and it times out, please try again later.
- Inactive/Idle notebooks will be automatically stopped after 8 hours
- To report your feedback please visit the following [GitHub issue](#), open as a safe feedback space.
- If you uncover a security issue, please report it privately by emailing esf-admins@fnal.gov
- If you find any other regressions, please open an issue in the [EAF GitHub repository](#)
- If you don't find any issues, we also appreciate positive input. Make sure to add the successful update on the [feedback space](#).

Sign in

Username:

Password:

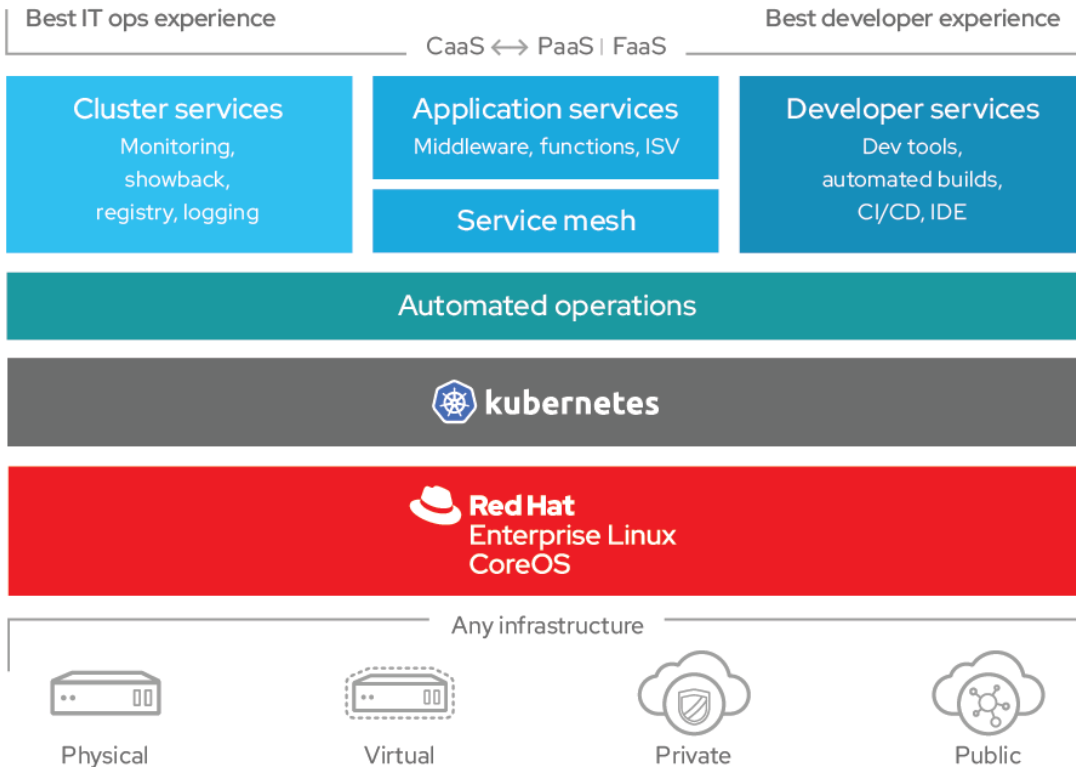
Sign in

jupyterhub Home Token Help/FAQ Admin Services macosta Logout

Server Options

ACCEL-AI	DES/LSST/ASTRO	CMSLPC
<ul style="list-style-type: none">● SL7 Interactive● SL7 Interactive General Purpose Notebook● GPU SL7 Interactive (NVIDIA Tesla K40m)● GPU SL7 Interactive (NVIDIA Tesla T4)	<ul style="list-style-type: none">● SL7 Interactive● COFFEE-DASK SL7 Interactive● GPU SL7 Interactive (NVIDIA Tesla K40m)● GPU SL7 Interactive (NVIDIA Tesla T4)● GPU SL7 Interactive (NVIDIA Tesla T4)● GPU SL7 Interactive (NVIDIA Tesla T4)● GPU SL7 Interactive (NVIDIA Tesla T4)	<ul style="list-style-type: none">● SL7 Interactive● SL7 Interactive General Purpose Notebook● GPU SL7 Interactive (NVIDIA Tesla T4)● GPU SL7 Interactive (NVIDIA Tesla T4)
LBNF/DUNE/ProtoDUNE	FIFE/Neutrinos	Fermi generic SL7/CC8
<ul style="list-style-type: none">● SL7 Interactive General Purpose Notebook● GPU SL7 Interactive (NVIDIA Tesla T4)● GPU SL7 Interactive (NVIDIA Tesla K40m)	<ul style="list-style-type: none">● SL7 Interactive General Purpose Notebook● GPU SL7 Interactive (NVIDIA Tesla K40m)● GPU SL7 Interactive (NVIDIA Tesla T4)	<ul style="list-style-type: none">● Basic SL7 Interactive● Basic CC8 Interactive

Start



OKD: open-source OpenShift

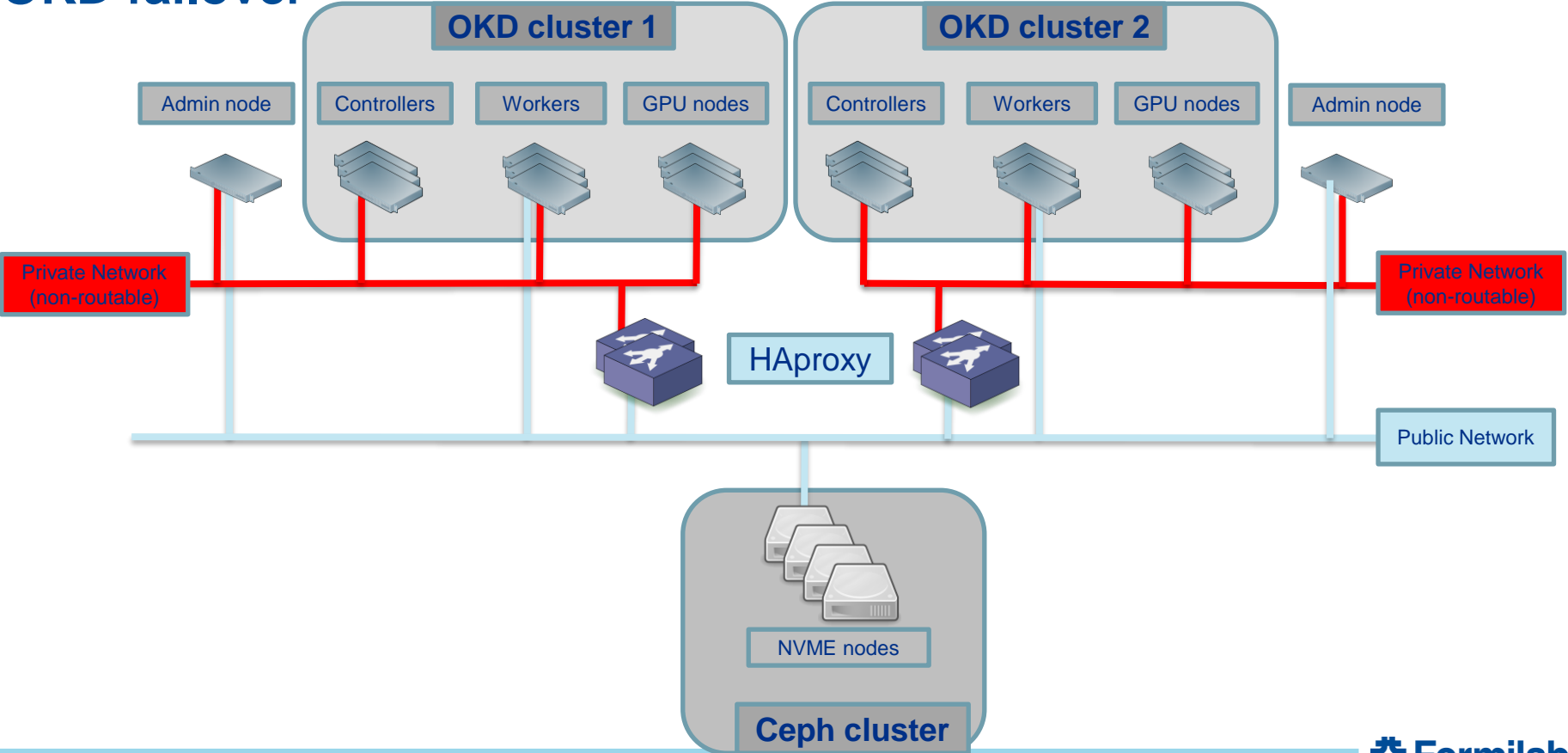
Provides:

- Namespace isolation
- Good default security policies (e.g. containers are unprivileged and non-root)
- Monitoring/logging primitives

OKD deployment overview

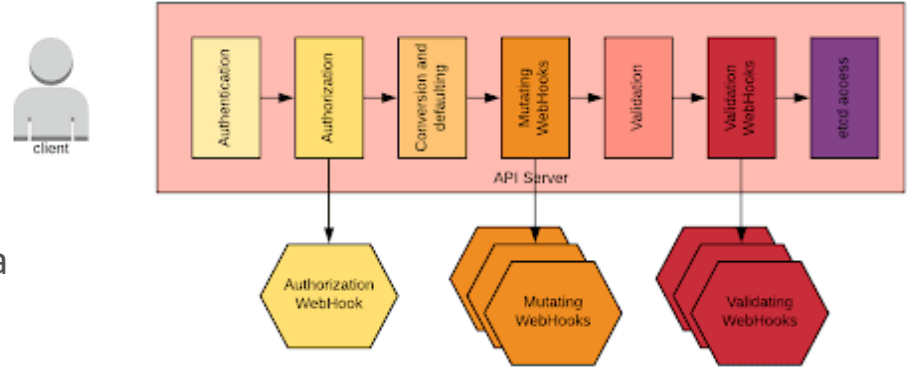
- OKD 4.x on thin VM on bare metal
- Installation fully scripted
 - Configuration files generated by Puppet
 - Additional configs (chrony, yum, RPM's Ceph keys) done through OKD `MachineConfig` objects
 - One single command to install OKD and other components (OPA, ceph-CSI)
- Using isolated network, users can access through HAproxy using LDAP authentication
- Redundant cluster strategy
 - Second cluster used as a cold spare and for testing upgrade or troubleshooting
 - Permanent storage on external Ceph cluster so it can be accessed from both clusters

OKD failover



Open Policy Agent (OPA)

- OPA on OKD
 - OKD allows creation of webhooks in the validation stage
 - OPA can inject homemade policies through the webhook to allow/reject objects
- OPA Policies:
 - Rego language
 - OKD agnostic, policies just parse yaml files
 - Policies can be updated live from Puppet Hiera
- Currently used for:
 - NFS mounts
 - External IPs



Operating a multi-VO analysis facility

- EAF and its components have evolved over a couple of years of prototyping
- Applications, integration strategies, infrastructure and storage provisioning has been designed following **Cloud Native** recommendations and best practices.
- Among other reasons like **operational efficiency** and **faster development cycles**, cloud native allows us to take advantage of our on-prem OKD resources and scaling capabilities – hence the ‘Elastic’ on the name



The entire system is described **declaratively**



The canonical desired system state is **versioned** in git



Approved changes can be **automatically applied** to the system



Software agents ensure correctness and alert (diffs & actions)

Principles of GitOps: <https://gitops-community.github.io/kit/>

Operating a multi-VO analysis facility

GitOps: Why?

- Takes advantage of a declarative system to manage the configuration and operations of every element of the platform, from the infrastructure through to the applications.
- Provides observability and control - ensuring that the platform is reliable and operable.

GitOps: How?

- All our production applications are deployed and managed via Helm charts
- Application/IaaS code is version controlled, securely hosted in lab's internal GitLab instance
- Published OKD templates to deploy specific applications for hosting CVMFS as well as containerized HTCondor worker nodes for Kubernetes
- CI/CD pipelines implemented for build, test, audit and push to image repository
- Orchestration and deployment via Argo CD



Operating a multi-VO analysis facility – Lessons learned

- UNIX User management must be consistent lab wide: Not trivial to achieve in JupyterHub as LDAP authenticator does not support it.
- Built multiple custom hooks in KubeSpawner which handles UNIX user creation via FERRY (lab source-of-truth for user account information) for Pod customization, CVMFS mounts, access control, NFS areas, CephFS persistent volumes and initialization containers.
- Layered container builds allow for code recycling while providing flexibility for diverse custom environments.
- BinderHub provides an extra layer of flexibility for notebook customization by users – Requirements are wide as we are not only supporting VOs but specific research groups within experiments.
- Some of the open-source applications deployed are specifically designed for vanilla installations of Kubernetes. Notably, there is a general absence of Service Accounts, poorly defined and delimited security contexts and wide role definition.
- A considerable R&D effort has been destined to patching applications, submitting pull-requests to code owners and making recommendations in open, collaborative workspaces within the open-source community.

Summary and Common Themes

- HEP Analysis Facilities at US National Labs are deployed and in use at Brookhaven, Fermilab, SLAC
- Use of declarative installation and infrastructure tools gives us agility to meet evolving analysis needs
- Elasticity provided to already-existing batch software systems