Plans for Facility R&D

Lincoln Bryant
US ATLAS Computing Facility Biweekly
August 2, 2023

Facility R&D

- Facility R&D broadly refers to activities related to the exploration and innovation of systems, services and physical infrastructure that provide platforms suitable for HL-LHC service environments and runtime ecosystems.
  - These can be purely local facilities (platforms deployed within a local area network) or distributed, in the sense of interoperating services over wide area networks.
  - c.f. IRIS-HEP 2.0 strategic roadmap sections on *Facility R&D and Integration*
- In US ATLAS we have adopted cloud-native application management methods for infrastructure and services
  - Kubernetes with GitOps-style management of resources
- Flexible strategies are being tried out in the IRIS-HEP Scalable Systems Laboratory (SSL) and the UC shared Tier3 Analysis Facility
  - These patterns can be adopted at other sites

Aligning with IRIS-HEP 2.0

- Why?
  - Concepts in IRIS-HEP 1.0 (K8s substrates, GitOps services) now in production (Run 3 Tier3 center, analysis facilities)
  - We cannot afford an independent R&D effort in the Facility
  - Align with 0.75 FTE expected from IRIS-HEP 2.0
  - Inform the next generation Tier2 and Analysis Facilities
  - Must figure a way to evolve what's below to provide needed flexibility above
- SSL - Scalable Systems Laboratory
  - For the past four years, a dedicated K8s cluster at UChicago to support R&D
  - *River* has been an important space for prototypes and even called into production use for analytics and hosted CE services

Motivation for Today's Talk

- Today's presentation will be focusing on our efforts surrounding Kubernetes and "cloud native" technologies throughout the facility
- Broadly, we will put existing and new efforts in that space under Facility R&D and work to prepare our sites (at all tiers) for the eventual requirements of HL-LHC computing
- Will cover a number of the milestones we have laid out for Facility R&D, including:
  - Evaluating OKD for the facility
  - Building a stretched K8S across the T2 facility
  - Using federated login via the ATLAS IAM
  - Tagging resources for scheduling decisions
  - Evaluating various batch schedulers in the K8S space
  - Dynamically scaling services with Horizontal Pod Autoscalers
  - Growing the stretched T2 with retired workers
  - Backfilling with ATLAS production

Evaluating Kubernetes distribution (MS #332)

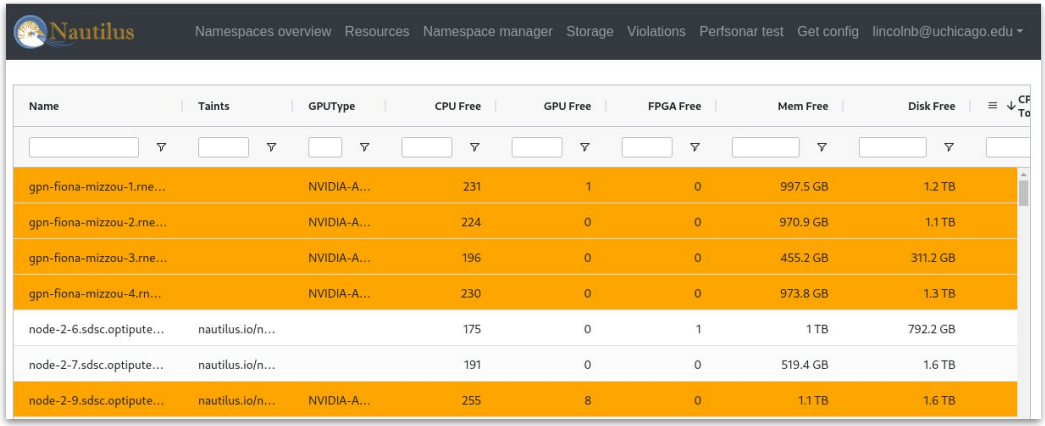
- Much like Linux itself, there are a variety of Kubernetes flavors
  - Many national labs have chosen OKD for its tighter security out of the box and alignment with Red Hat OpenShift
  - Others (e.g. NERSC) have chosen the SUSE-backed Rancher platform instead
  - Many universities have deployed Kubernetes with the tools provided directly by the Cloud Native Computing Foundation, such as kubeadm or kubespray
- Given the OKD experience at BNL and now NET2, it makes the most sense to start there
  - From a policy perspective, it is easier for sites nominally using vanilla Kubernetes to adopt OKD rather than the reverse
- Many things we need to investigate for compatibility
  - GitOps - Flux, Argo, something else?
  - Running our various services: Squid/Varnish/XCache under a stricter environment
  - Compatibility with tooling - Lens for example is very popular
  - Compatibility with federation, etc.
- If this appears workable, we would consider moving the UChicago Tier 3 to OKD as well

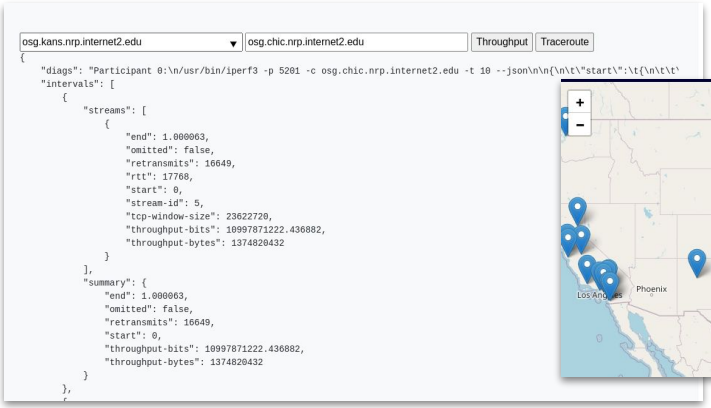
Stretched Kubernetes Multi-Site Cluster (MS #333)

- One of the major goals in Facility R&D is to build a multi-site Kubernetes that stretches over our T2 complex
- Each Tier 2 site contributing some resources to a managed R&D platform
- Enable rapid development and iteration for new services that can be operated close to the data at each site
- Draw from our own experience with federated K8S as well as others, e.g.:
  - PRP/NRP
  - PATh Facility
  - SLATE

Multi-site K8S examples - PRP/NRP

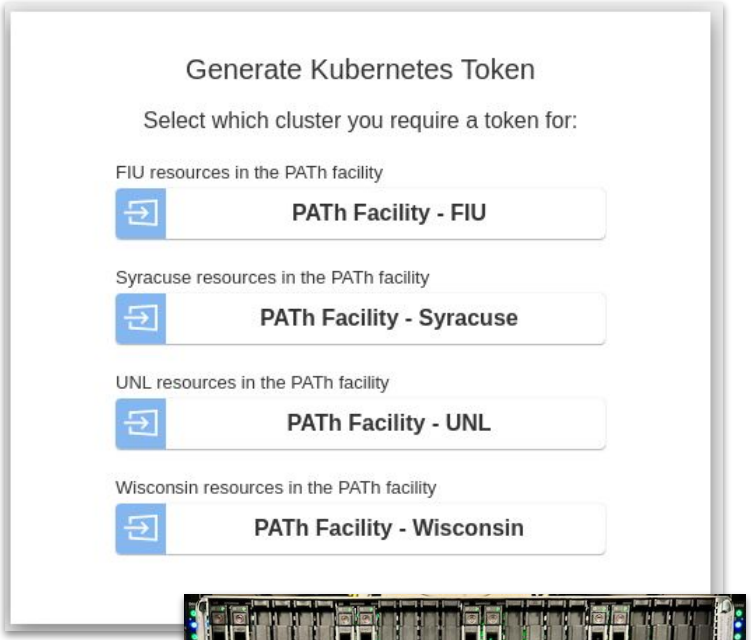
- PRP/NRP approaches multi-site K8S by directly stretching over the WAN
- Single control plane / endpoint
  - Very simple and natural experience for developers
- PRP/NRP team manages the whole stack from IPMI, through OS, through K8S
  - Easy for sites to bring resources to the table

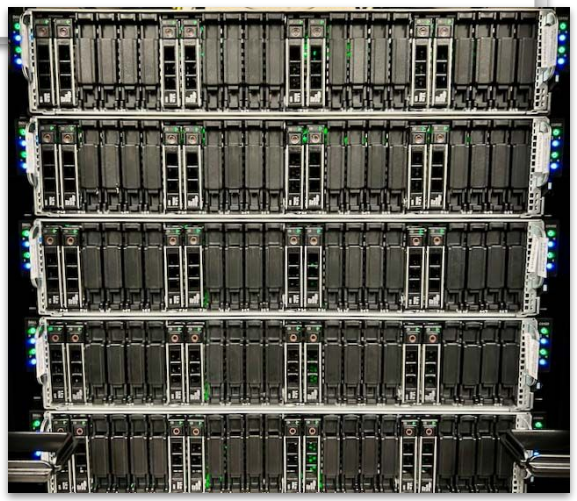




Multi-site K8S examples - PATh Facility

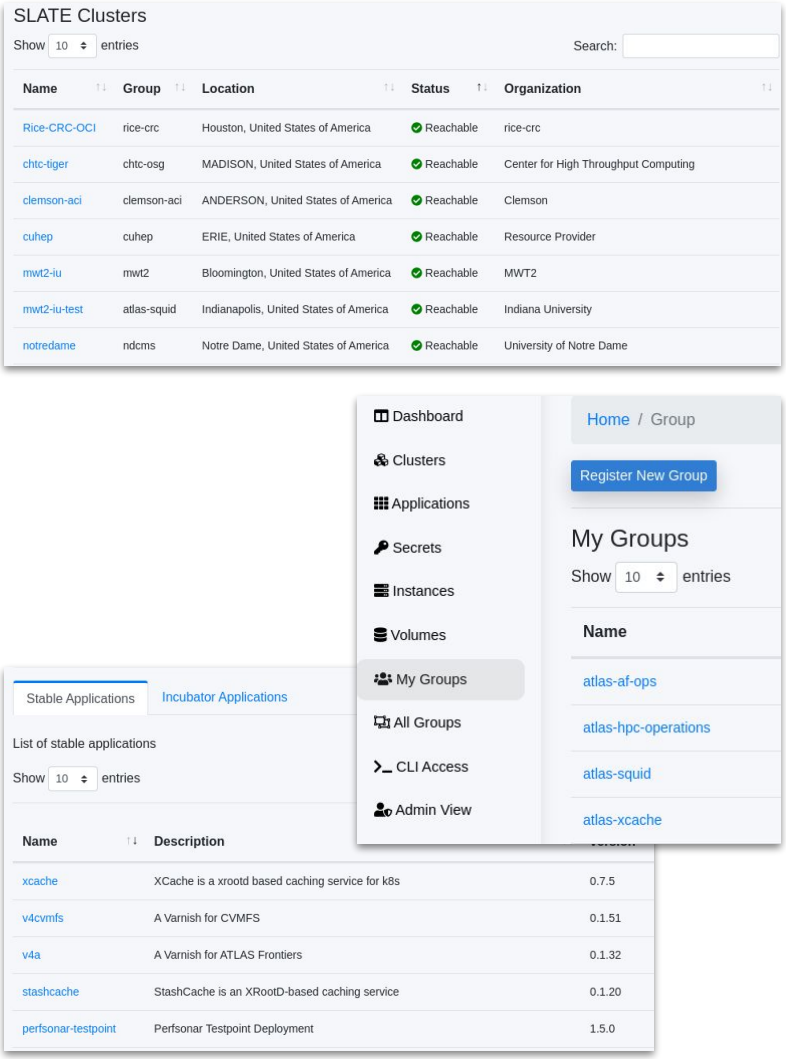
- In the PATh Facility, each site is treated as a separate cluster
- PATh team uses Flux tenants to manage the Kubernetes objects at each site via GitOps
- Hardware/networking managed by the site
- OS, Kubernetes, Applications managed by the PATh team





Multi-site K8S examples - SLATE

- SLATE federates at a high-level with no assumption of direct control over any resource
  - Minimizes its presence and privilege on clusters
- Operates as a sort of concierge between resource providers and resource consumers
- Tightly curated application catalog with strong focus on security
  - Platform is agnostic to *who* uses it and *where* they run applications, so the SLATE team focuses on *what* can be run

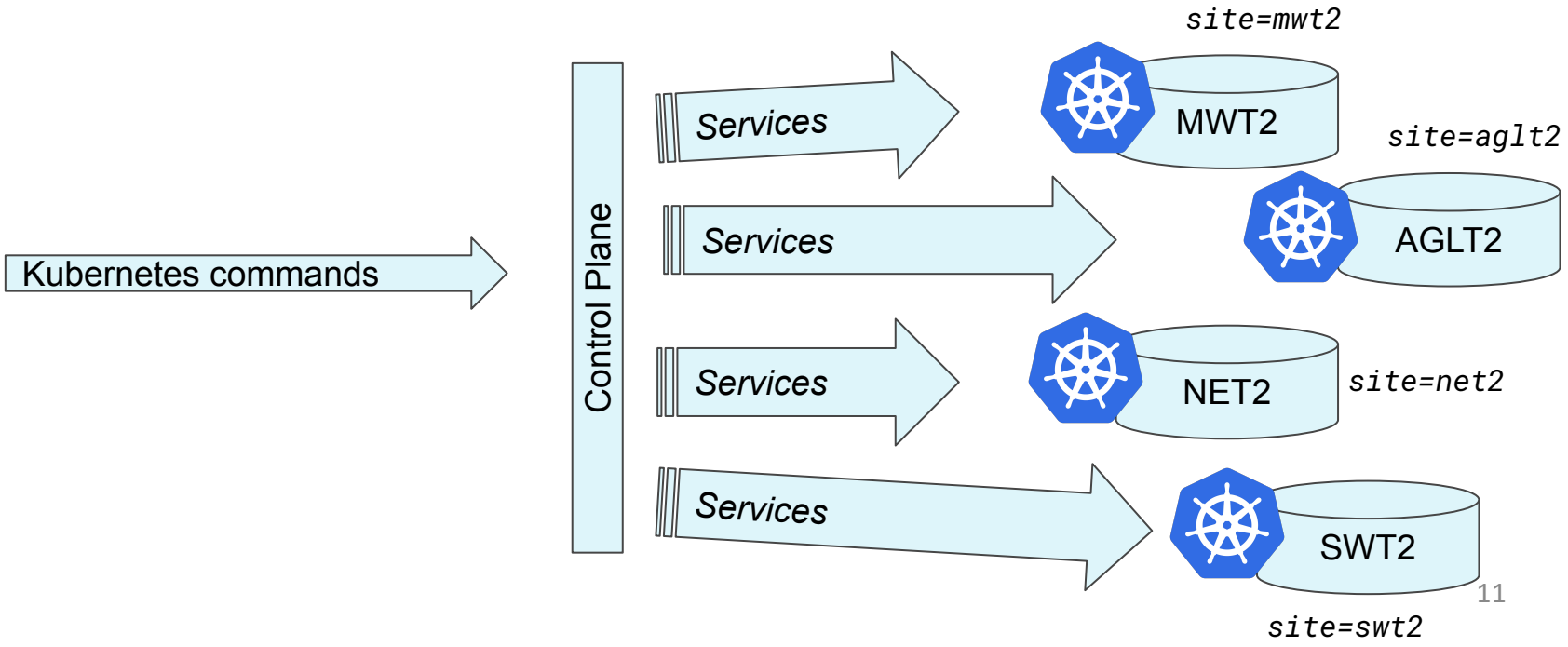


Multi-site K8S for US ATLAS

- We want to build a K8S platform for US ATLAS that blends the best aspects of these approaches
- Start very modest (2 sites), perhaps using OKD
  - Later adding the rest of the T2s (MS#340)
- We can make managing the node part of our Facility R&D effort, if desired
  - This requires someone on the Facility R&D team having root
- Stretched over the WAN, single API endpoint
- Tag resources by geographic location
  - Run services close to where data lives
- Single identity system via the ATLAS IAM
- Fold in resources such as:
  - Existing SLATE servers at each site
  - Retired workers (CPU, services) and storage nodes (caches or object stores)
- Backfill with ATLAS production, no CPU wasted

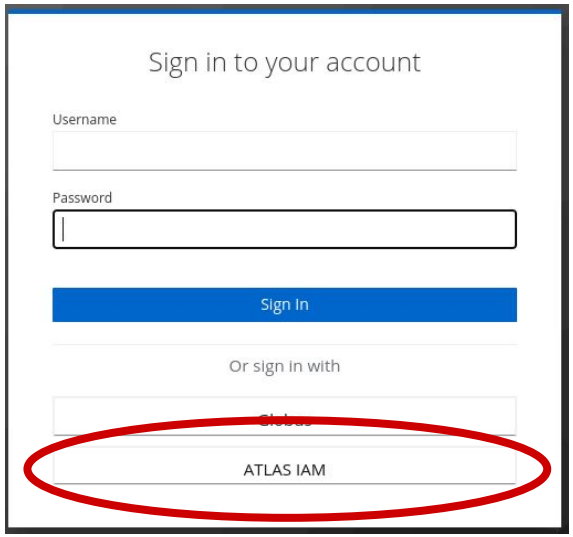
Multi-site Resource Tagging (MS #336)

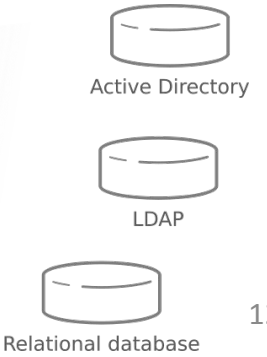
- Once we have established our multi-site K8S, want to add labels to each site for scheduling decisions
- Selectors for geographic location, data availability, and so on
- Tools like PanDA-Dask, ServiceX, etc could use these tags to place jobs nearest to the site that has the appropriate data in Rucio



Identity & Authorization (MS#337)

- Our stretched Facility R&D platform ought to use a single sign on technology that uses existing identity providers
- Minimize the need for yet another account
- We will plan to leverage Keycloak connected to the ATLAS IAM
  - Others (BNL) have experience in this area as well
- Users will login with CERN credentials, get their Kubernetes credentials via OIDC authentication flow to the IAM
- Also useful for any other applications we may want to incorporate (e.g. Jupyter)
- As well as other identity providers (Globus, CI Logon, etc) supporting OIDC





Continuous Integration Framework (MS#341)

- The Facility R&D platform will support two styles of object deployment
  - Production Deployment
    - Utilizing GitOps via Flux or ArgoCD
    - All long-lived services must have provenance in a Git repository
    - Monitored, alerted upon, etc
  - Scratch deployment
    - Namespaces attached to individual users
    - Directly deploy things via ‘kubectl’ or your favorite tool after IAM login
    - Objects are aggressively cleaned after a few days
- Idea is to cut down on ‘junk’ deployments of broken and forgotten stuff.
- Additional policies can be applied with Pod Security Standards, Gatekeeper, Kyverno, etc

Backfilling with Retired & Overpledged CPU (MS#342)

- Plan to add additional resources to our seedling cluster
- At MWT2, we will start off by adding 25-35kHS06 of over-pledge and retired worker nodes to this cluster
  - Invite other T2s to do the same
- This will add non-trivial resources to the platform for any R&D efforts
- We will backfill these with production using the same approach used by the UTA cluster as well as the Google Cloud project
- Can also consider adding retired storage for scratch purposes, either as some form of Ceph storage (Object, Block, Filesystem) or XRootD caches with varying QOS expectation

Evaluating Scheduling technologies (MS#338)

- Assuming we're successful and we keep the R&D Platform full of work, we'll want to look into scheduling technologies to
- Simplest strategy to start will be to use Pod Priorities, including preemption
- However we will also want to investigate a number of other technologies including:
  - Volcano Batch Scheduler, originally developed by Huawei
  - Kueue, a "Kube Native" scheduler that came out of the Kubernetes Batch Working Group
  - Descheduler, a tool for rescheduling Kubernetes pods based on changing cluster conditions

Horizontal Pod Autoscaling (MS#339)

- The applications that we run on the Facility R&D Platform need to be responsive to changes in demand
- For example, ServiceX transformers should scale up or down as appropriate based on the number of files needing to be transformed
- Caches could likewise scale up/down depending on the number of jobs demanding data or changes in the working set size
- We will provide a demonstrator of a working HPA recipe for services on the Platform

Monitoring, Alerting, APEL accounting (MS#335)

- Adopt tools used in the Kubernetes community for monitoring and alerting, including Prometheus, Grafana, etc.
- These can be especially noisy, need to refine them to have a good signal to noise ratio
- Accounting for jobs run on the various Tier 2 K8S efforts (UTA, NET2, stretched Facility R&D platform) need to be reported to APEL
  - Adopt KAPEL from UVic?
  - Work with our OSG-LHC colleagues as appropriate

Bursting to cloud/T2s (MS#332)

- Analysis Facility workloads are more latency sensitive (in terms of turnaround time) than our traditional jobs
- We plan to investigate a number of approaches for getting resources to AFs quickly, including:
  - Demonstrating the ability to burst into Tier 2 facilities via HTCondor flocking/glideins
  - Demonstrating the ability to burst to cloud resources
- This will require understanding the limitations/requirements for users.
  - Many workflows depend on shared filesystems, local accounts, etc like a traditional HPC-style cluster
  - How much effort for users to adapt their workloads?
  - What can we do to adapt the resources to the user?

Bi-Weekly Meeting

- Incorporate input from our various K8S-related efforts
  - K8S T2 testbed at UTA
  - K8S T2 at NET2
  - K8S SLATE GitOps platforms at each Tier 2
- Include interested folks from PATh, IRIS-HEP
- Discuss common issues/solutions, experiences, technology choices
- Build & share recipes and documentation
- Newdle poll for figuring out an appropriate timeslot:
  - https://newdle.cern.ch/newdle/zTGxgRkZ

Summary

- A flexible R&D platform is needed within the facility to explore and innovate with new systems, services, and infrastructure
- This will inform the next generation T2 and future analysis facilities
- This work aligns nicely with the IRIS-HEP 2.0 strategic plan and effort we are receiving
- We expect this work will also greatly benefit sites that have already taken the plunge into the world of Kubernetes/GitOps/etc