Evolving AI Research Infrastructure with Kubernetes at Meta

Scale 22x

Shaun Hopper
Production Engineer

Chandan Avdhut
Production Engineer

Meta

Introductions

- Who we are
- We support instance level compute for all of Meta in Public Cloud.
- We build compute platforms, tooling, managed infra.
- We don’t run every host ourselves, make the distinction between customer managed hosts vs our platform.

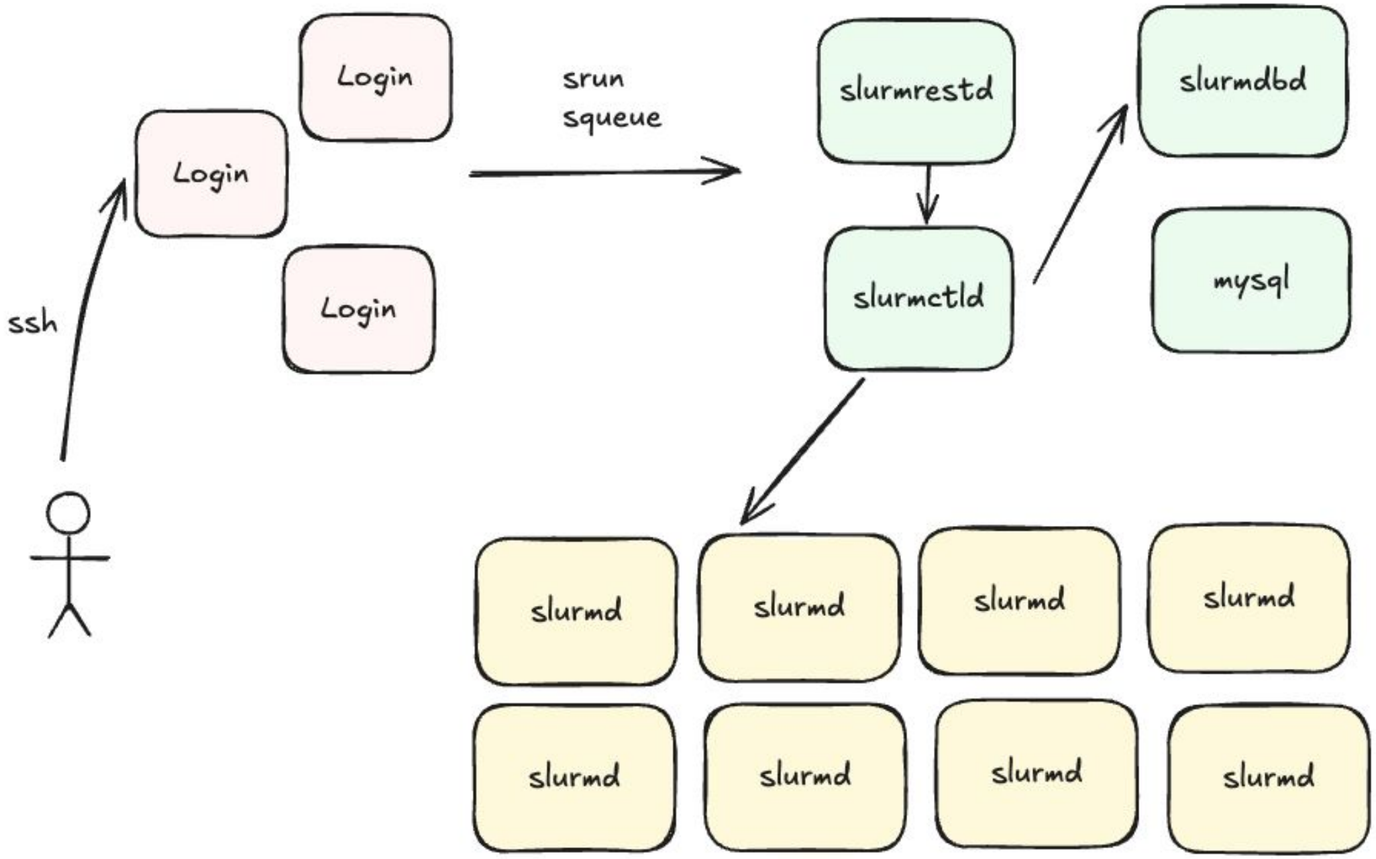
Shaun Hopper

Chandan Avdhut

Agenda

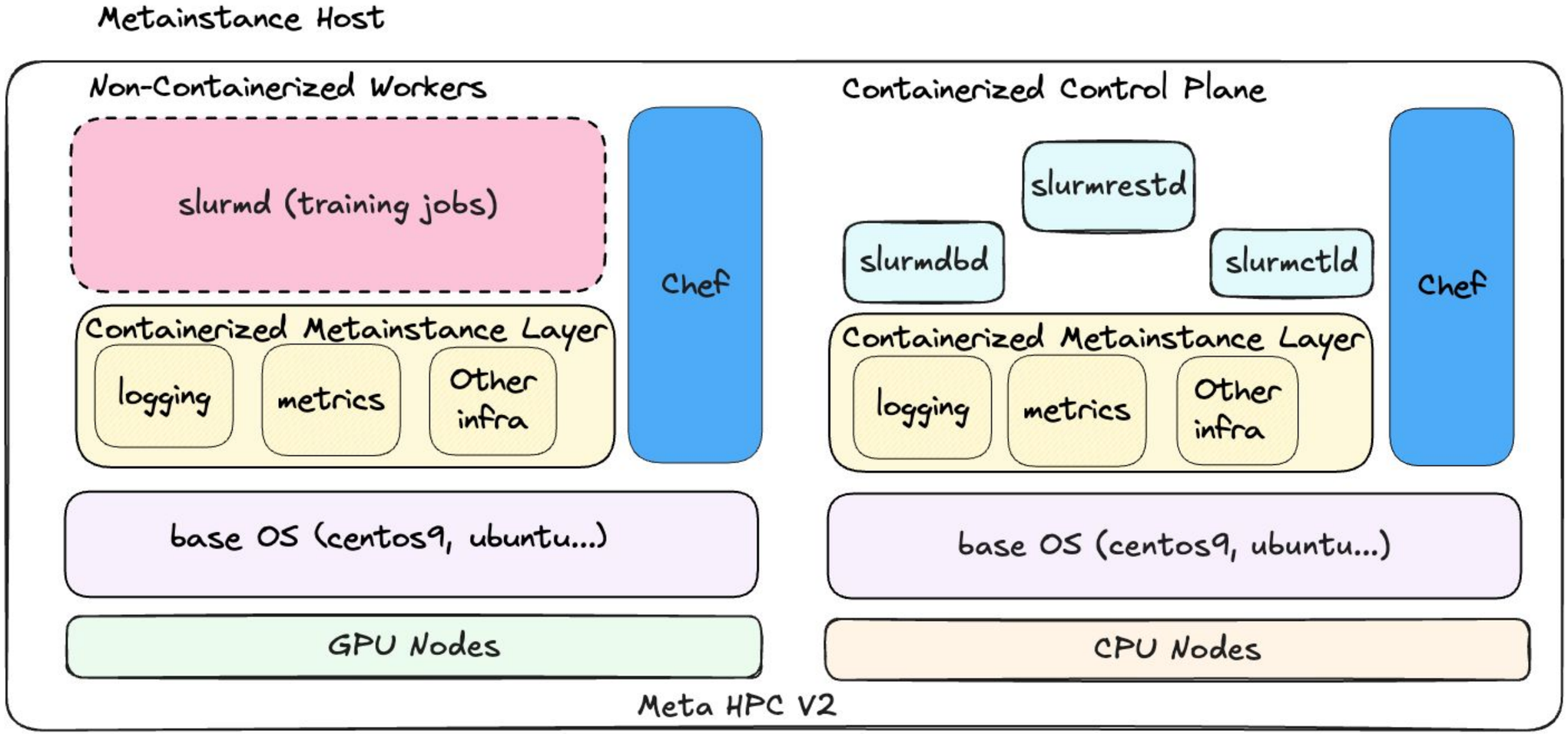
- Preserving the Research Experience
- New Ways of Host Management
- Proper Data Access Controls

A typical slurm research cluster

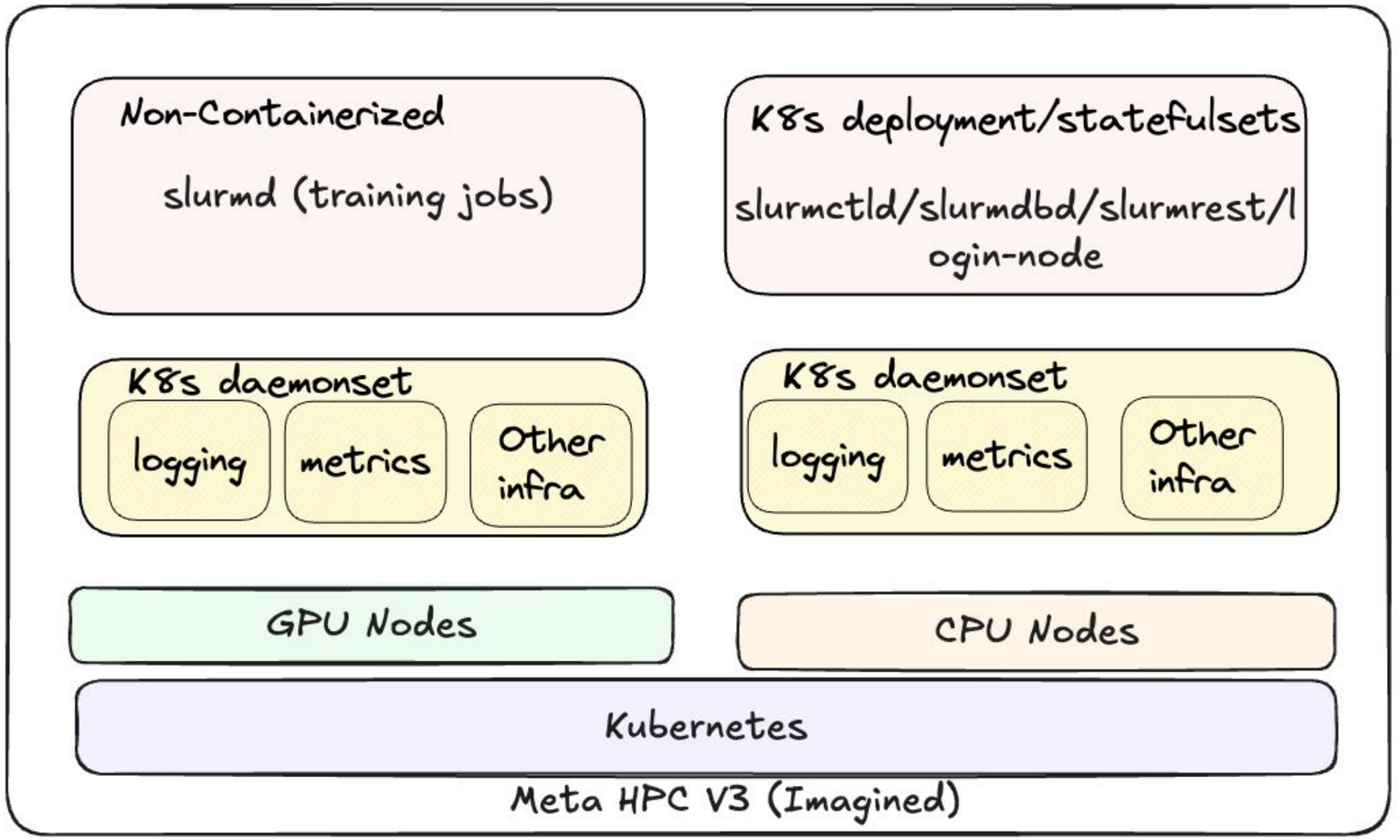


Existing Architecture

- MetaInstance based HPC (v2) with Chef on top
- All research clusters ran Slurm.
- We were heavily invested in immutability & containers.
- We knew we were going to use Kubernetes for the control plane and various platform components.

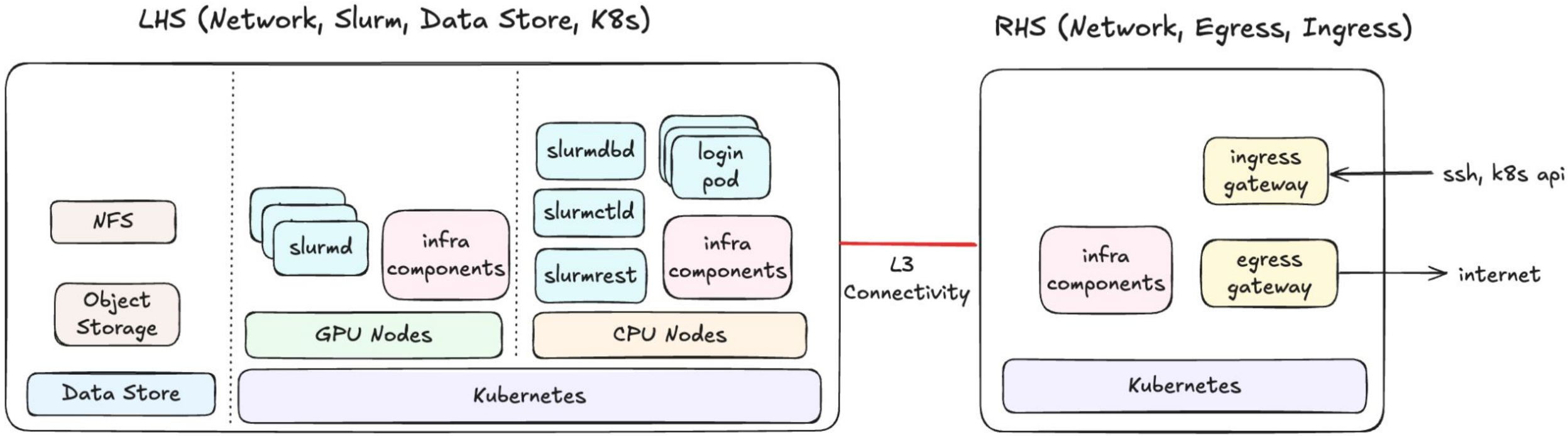


Where we thought we would go



Where we went

- We ended up completely rebasing the entire HPC cluster on top of kubernetes.



Preserving the Research Experience

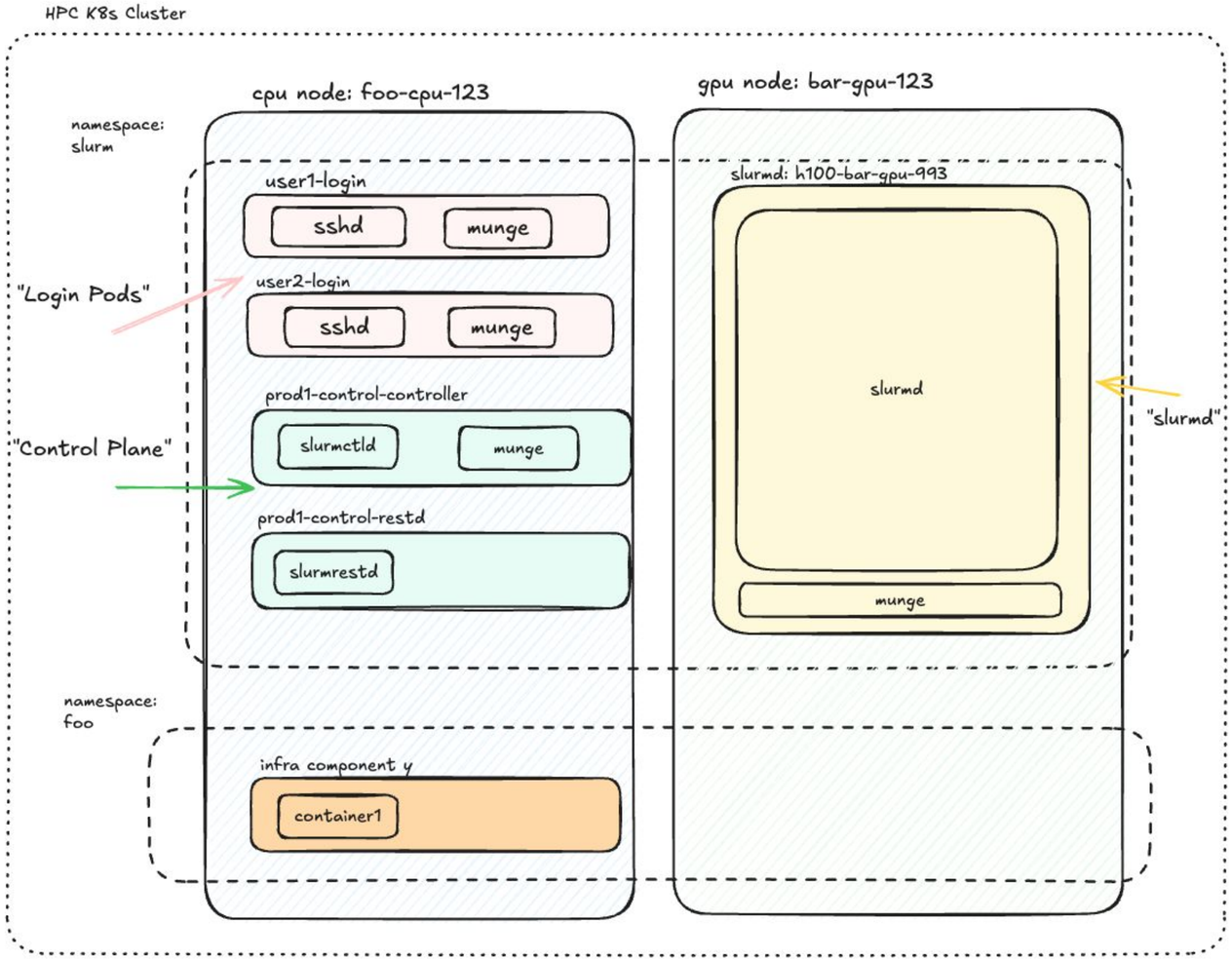
The research experience needed to remain consistent

Challenges

- A researcher should have no idea that they are inside of a kubernetes cluster.
- They never run helm, kubectl, etc.

Solutions

- Custom CLI that presents a facade over k8s API interactions.
- Since k8s is just http, this is easy.



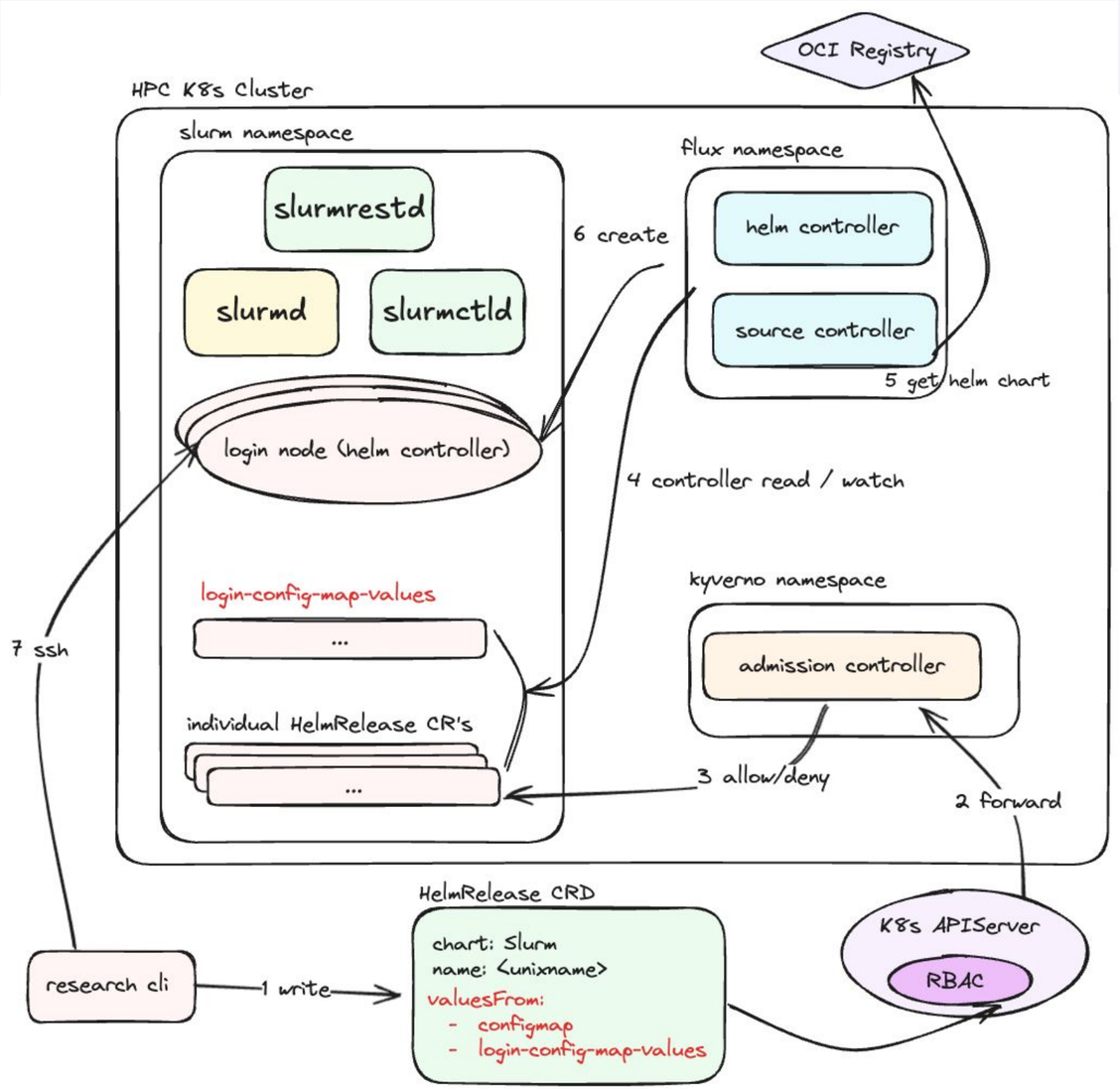
Researcher’s login environments went from multi-tenant to individual

Challenges

- Researchers gained isolated environments with resource constraints preventing noisy neighbors and enhancing security.
- Resource guarantees meant we needed login pods needed to be provisioned on-demand and cleaned up automatically.

Solutions

- Flux helm controller and a login pod chart
- Kyverno admission controller to enforce HelmRelease values.



New Ways of Host Management

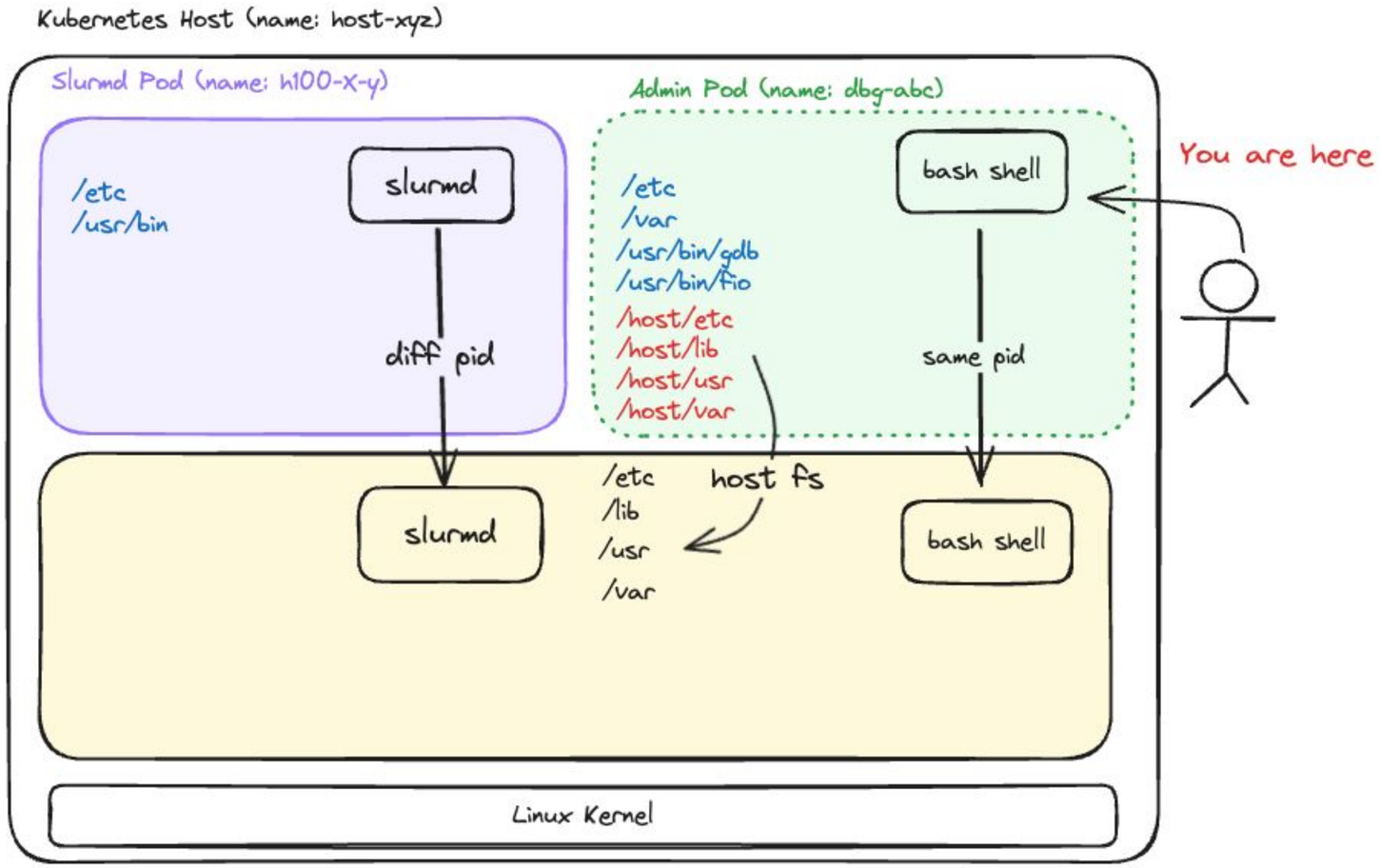
Managing hosts looks completely different now

Challenges

- Hosts are now immutable
- No systemd besides starting the containerd/kubelet
- No config management via chef or ansible
- Everything has to be defined in kubernetes
- kube exec became the new ssh

Solutions

- admin-pod daemonset
  - A landing zone for kubectl exec
  - Tools can be installed without dirtying the host
- Host Management daemonsets
  - Host-cfg: set sysctls
  - Automount: Continuously mounting new NFS datasets



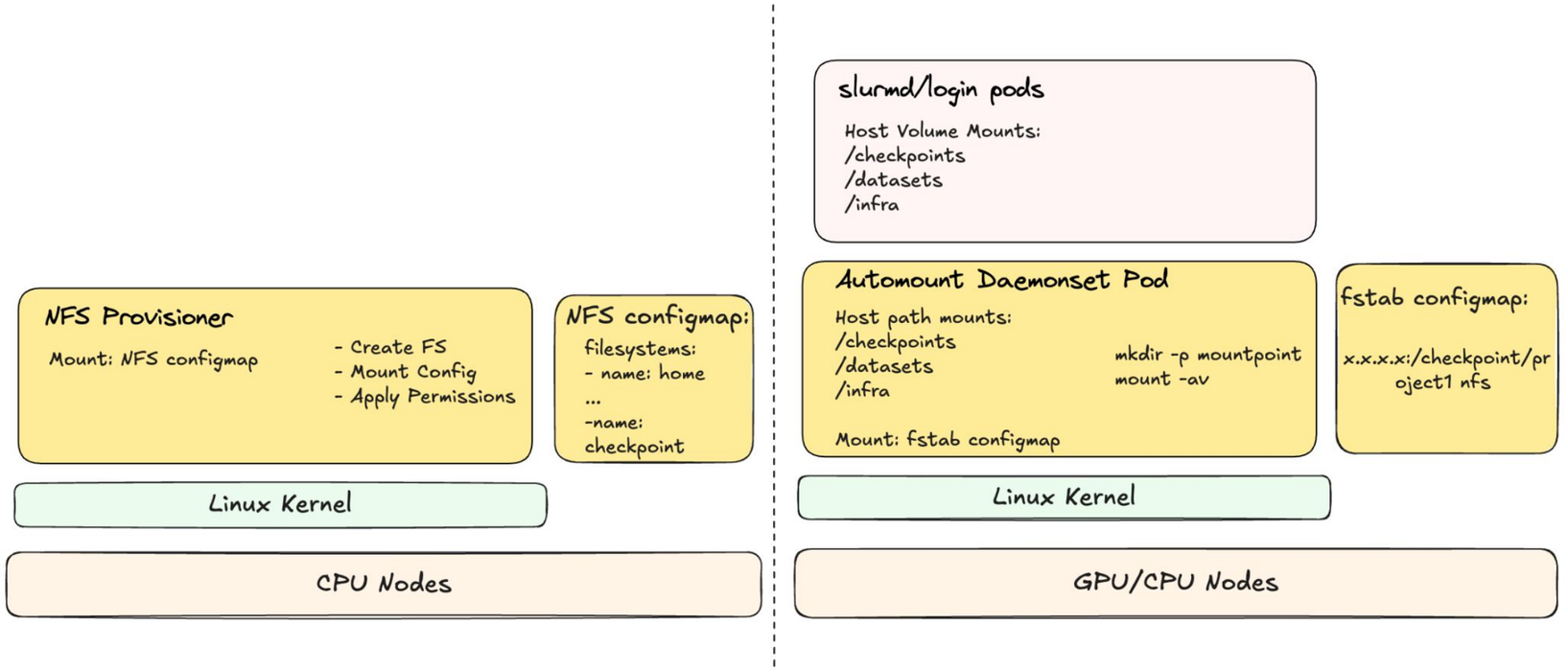
Datasets should come and go without disruption

Challenges:

- New datasets should automatically appear in login & worker pods.
- Using CSI Drivers, PVCs, and PVs causes pod restarts for login and worker pods

Solutions:

- Use NFS everywhere, and do all mounting at the host level.
- Bind mount NFS into login & worker pods.



Proper Data Access Controls

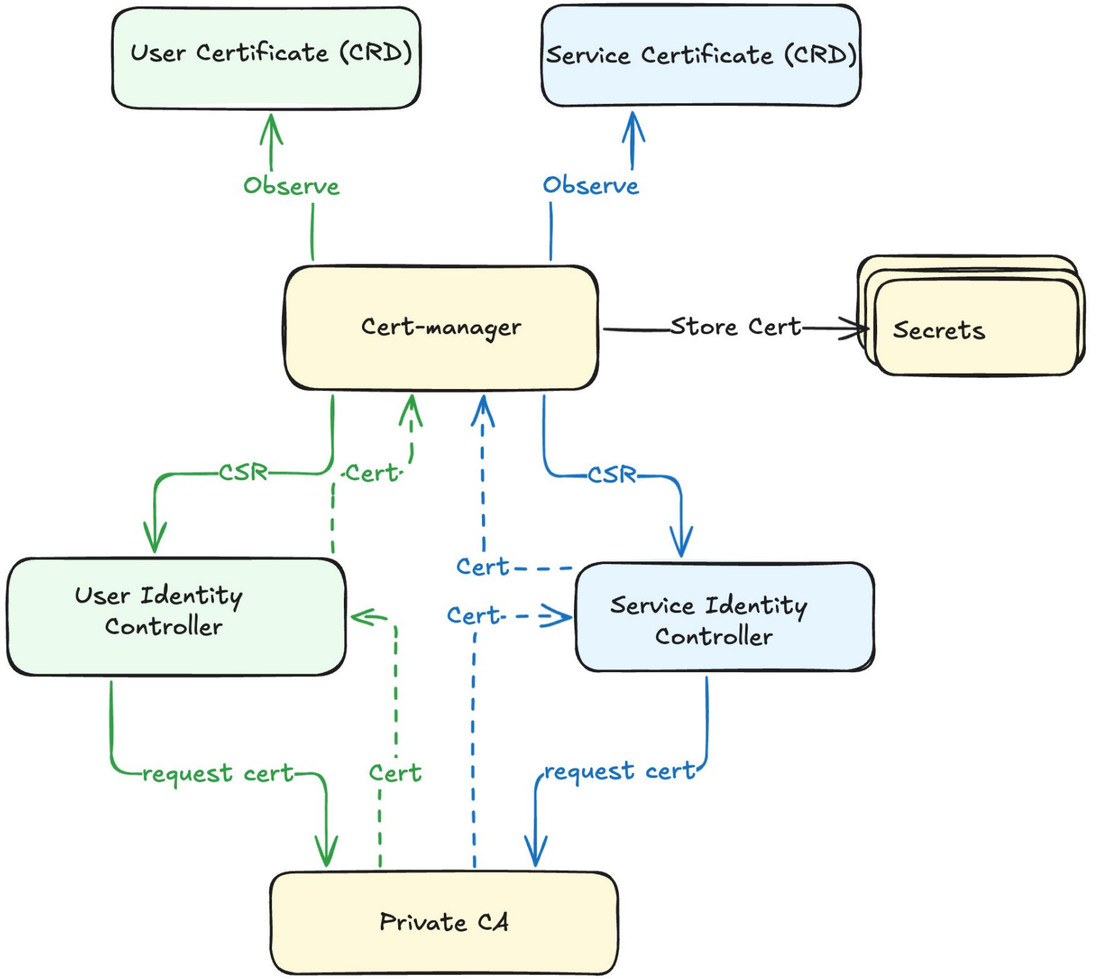
All data access needed to be locked down and auditable

Challenges:

- HPC cluster-internal and external access needs MTLS.
- MTLS requires certificates, but we want them signed by a private CA.

Solutions:

- cert-manager was used to vend service and user certificates
  - External-issuer allowed us to vend them backed by a private CA
- MTLS enables egress gateway to able to log every request with user or service identity



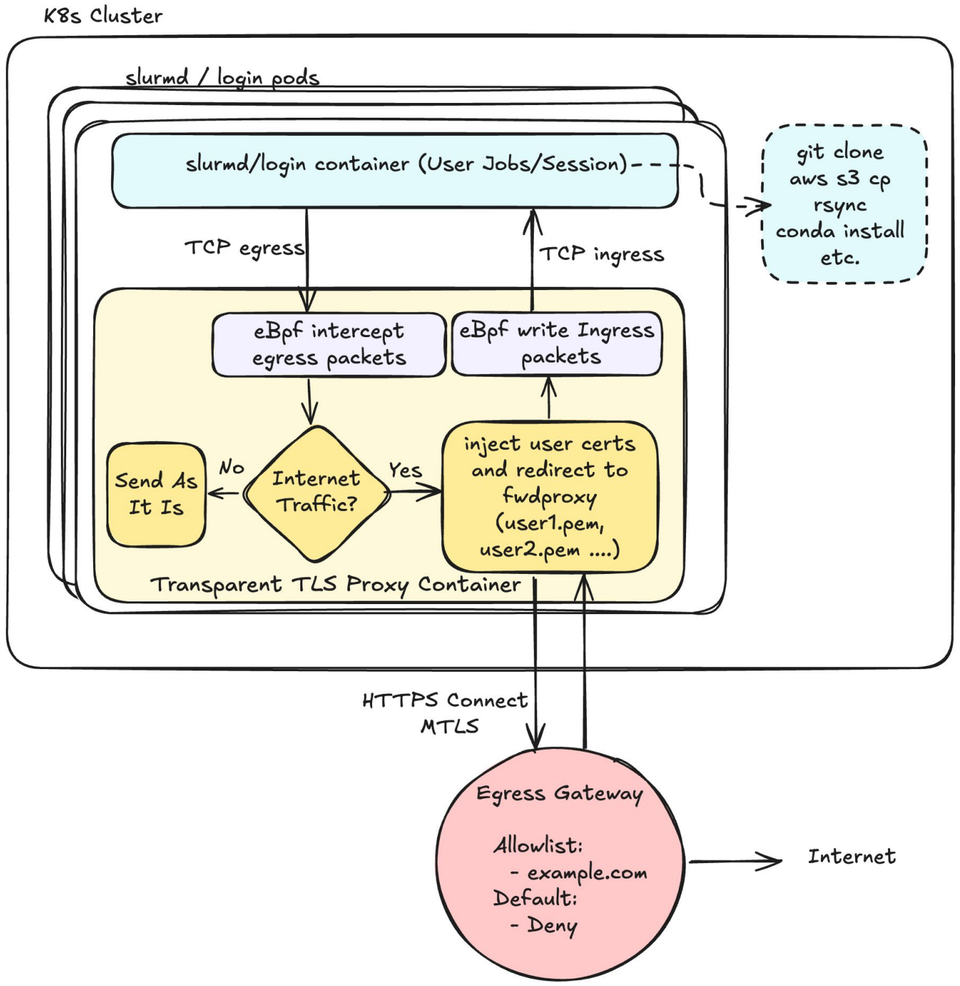
mtls & certificates: what's so hard about that?

Challenges:

- Research workflows are sensitive, git clone, pip install, etc are routine. We don't want to get in the way, and not everything support client certs.
- Researchers workflows should be portable from cluster to cluster. They shouldn't have to explicitly configure proxies, client certificates, etc.

Solution:

- A e-bpf hook transparently proxies tls (ttls) connections with user certificates.



Whats Next?

We plan to focus on

- Addressing login pod restart sensitivity & image push frequency
- Optimizing image size as more features are added
- Implementing canary mechanisms
- Improving debuggability of components like TTLS


kubernetes


slurm
workload manager


containerd


HELM

Thanks


Kyverno


flux


eBPF


CERT
MANAGER


Meta