Cloudy Topics from the Hutch

- Traditional HPC and Kubernetes Co-Existence
- Hybrid Cloud HPC with Slurm
- Lessons Learned in the Cloud
- Cloud Native HPC with AWS Batch
- Unified Kubernetes Cluster Management

Robert McDermott
IT Enterprise Architect
rmcdermo@fredhutch.org
Traditional HPC and Kubernetes Co-Existence

- **Goal**: Allow Slurm and K8s batch jobs to share the same cluster hardware

**Diagram**:

1. Researchers
   - Create batch job definition
   - Submit job using submission script
   - $ kbatch -N 2 myjob2.yml

2. Slurm Batch Jobs
   - $ sbatch -N 2 myjob1.sh

3. Slurm
   - $ salloc –N 2
   - Allocate nodes

4. Hybrid Slurm / Kubernetes Cluster
   - Pod1
   - Pod2
   - Job: user1-myjob2

5. K8s Master
   - Add k8s node labels

6. K8s Node
   - N1
   - N2
   - N3
   - N4
   - N5
   - N6

7. Find completed k8s jobs
8. Delete completed jobs
9. Remove node labels
Traditional HPC and Kubernetes Co-Existence

- **Is it possible?** Yes, but I wouldn’t recommend it on a shared HPC cluster
- Allowing users to run docker containers on a multi-user host with shared file systems and NFSv3 exports secured only via AUTH_SYS is **risky**
- You can use pod security policies to mitigate some of this risk
- The only safe way to give users Docker like functionality on a shared host is with Singularity: [https://www.sylabs.io/singularity/](https://www.sylabs.io/singularity/)

![Diagram showing HPC Node, Container, Bridge, and NFS v3 Server with mount points and export details.](image-url)
SLURM Hybrid HPC Cluster

Number of CPU cores in use by Lab

Number of HPC nodes by status
Cloud VPN Performance Problems

- Read performance from our on-prem storage systems (Isilon, NetApp and Avere) was very slow (2-3MB/s); Write performance was good (150MB/s)
- Assumed it was protocol (NFSv3) related
- Found performance was very good with standard Linux NFS servers
- Discovered that the protocol was not the problem (iperf had the exact same results)
- All poor performing NFS systems were running a BSD flavor operating system
- Working with our vendor (Fortinet) to solve this bug
- Investigating other VPN solutions
WireGuard  Our Future VPN?

- 3-6x performance improvement to IPsec
- 4k lines of code (LOC) vs. 100k LOC in the OpenVPN implementation
- Linux Kernel based code is faster
- More secure; weak cryptography modules have been removed

"Can I just once again state my love for it [wireguard] and hope it gets merged soon? Maybe the code isn't perfect, but I've skimmed it, and compared to the horrors that are OpenVPN and IPSec, it's a work of art."
https://lists.openwall.net/netdev/2018/08/02/124

Linus Torvalds
Some Lessons We’ve Learned Operating in the Cloud

- Hand built cloud infrastructure is easy to get started, but soon becomes a hard to manage and understand mess: **Automate the creation and evolution of infrastructure with a tool like Terraform and keep all code in a source code management system.**

- Large, multi-departmental, kitchen-sink style cloud accounts are hard to manage and present a very large blast radius if compromised: **Departments that need administrative access, mission critical applications and DR resources should all have their own dedicated child accounts.**

- Resources missing tags and inconsistent tag keys/values make it difficult reconcile the bill at the end of the month: **Create a mandatory tagging scheme that covers the dimensions you’d like to break costs down by (department, service, etc…) and enforce it.**

- If you use the default VPC you’ll have problem routing traffic from multiple accounts to your campus over a VPN. **Centrally track and allocate IP space to prevent any overlap. Also, don’t use the IP range 172.17.0.0/16 or you’ll have a support headache with Docker users (that’s the default private docker network).**
Bitcoin Mining Account Takeover

May 9 20:27 2018: Committed file containing API Key

We were very lucky!
Cloud Native HPC with AWS Batch

Docker Hub

mycontainer

Build

Pull

AWS Batch

Input

Output

mycontainer

S3

PI Buckets

github

fredhutch/mycontainer

- Dockerfile
- myscript.py

$ git commit
$ git push

1: Push Dockerfile and code to Github

$ aws batch submit-job --job-name myjob

2: Submit Job to AWS Batch
3: Collect results from S3

Fred Hutch Campus
Unified Kubernetes Cluster Management

- Abstraction layer for vendor and platform agnostic k8s architectures
- Single management console, centralized auth, RBAC, API and catalogs

Amazon AWS
- IaaS
  - EC2
- PaaS
  - EKS

Google GCP
- IaaS
- PaaS
  - CE
  - GKE

Microsoft Azure
- IaaS
- PaaS
  - VMs
  - AKS

Fred Hutch Campus
- VMware
- Hardware

User Interface
- $ kubectl
- Web Console
- App Catalogs
- Auth / RBAC
- https://rancher.com
Thank You