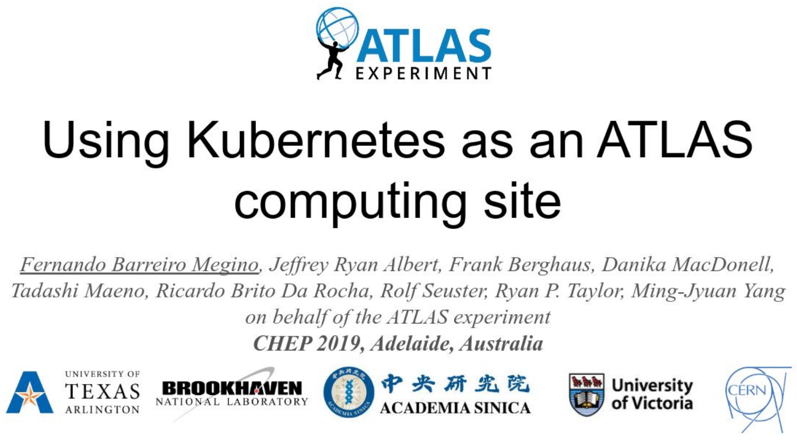
Building a fully cloud-native ATLAS Tier 2 on Kubernetes

Ryan Taylor
on behalf of UVic Research Computing Services

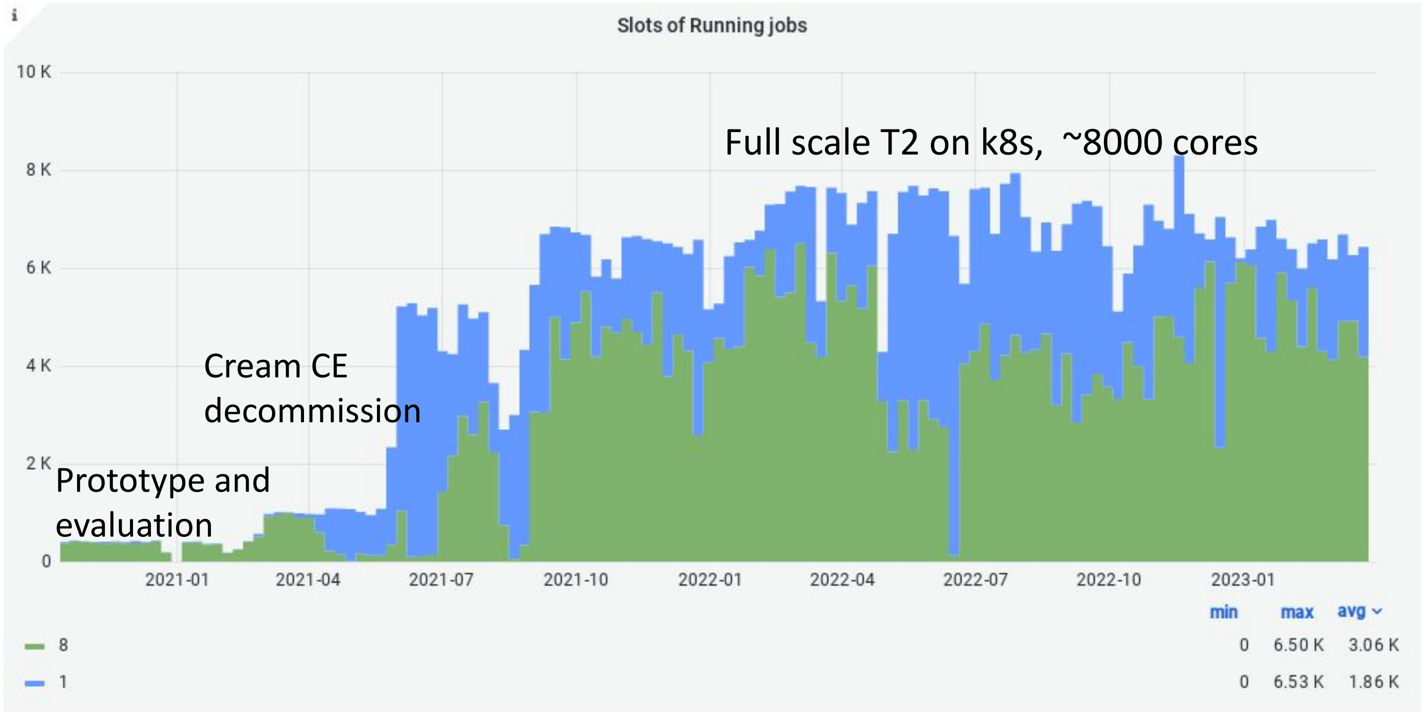
Background

CHEP 2019 presentation



Using Kubernetes as an ATLAS computing site

Fernando Barreiro Megino, Jeffrey Ryan Albert, Frank Berghaus, Danika MacDonell, Tadashi Maeno, Ricardo Brito Da Rocha, Rolf Seuster, Ryan P. Taylor, Ming-Jyuan Yang
on behalf of the ATLAS experiment
CHEP 2019, Adelaide, Australia



Full scale T2 on k8s, ~8000 cores

Cream CE decommission

Prototype and evaluation

CA-VICTORIA-WESTGRID-T2 uses Kubernetes for container-native batch computing. Harvester submits ATLAS grid jobs to k8s API, which runs them as pods. No traditional batch system or Compute Element.

The eventual goal: a fully k8s-native T2

Installable with Helm

- Helm: application manager for Kubernetes
  - One command to install/upgrade everything
  - Comprehensive configuration via one YAML file
- `helm install T2Site`

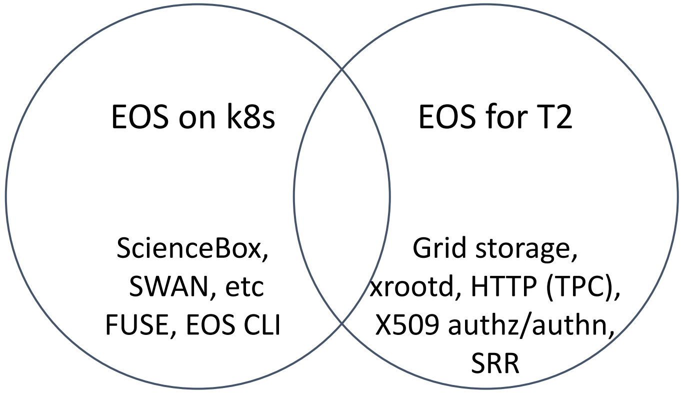
(K)APEL accounting	done
frontier-squid	done
compute (security rules, Harvester setup)	done (static YAML)
EOS SE	in progress
CVMFS-CSI	optional
~~Compute Element~~	built-in
~~Batch system~~	built-in

EOS SE on k8s with CephFS

- Physical consolidation: all storage on Ceph (CephFS)
- Logical consolidation: services on k8s
- EOS can be installed on k8s via Helm chart
  - reproducible, single step deployment
  - easier to manage and maintain
  - easy to set up another instance, e.g. for dev
- EOS + CephFS is an established solution
- Opportunity: direct data access for jobs on CephFS
  - Thanks to Andreas

EOS Helm chart

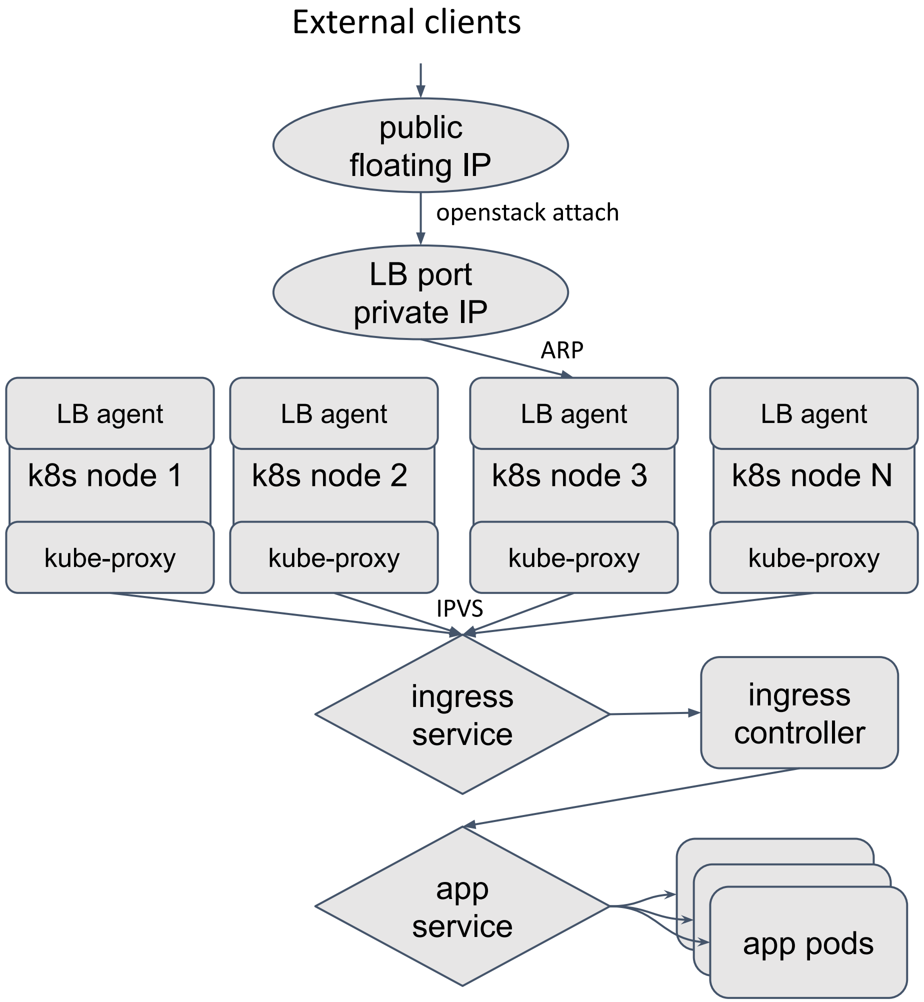
- Generally only used for internal clients so far
  - Different from typical grid storage use case
- Need enhancements for T2 SE use cases (#74, #75)
  - configure X509 VOMS authz/authn
  - install host certs via secrets
  - fetch-crl, grid-security CAs, etc.
  - **external network access**


EOS on k8s
ScienceBox, SWAN, etc FUSE, EOS CLI
EOS for T2
Grid storage, xrootd, HTTP (TPC), X509 authz/authn, SRR

Big thanks to Enrico for collaboration and support on Helm charts!

Network architecture on k8s

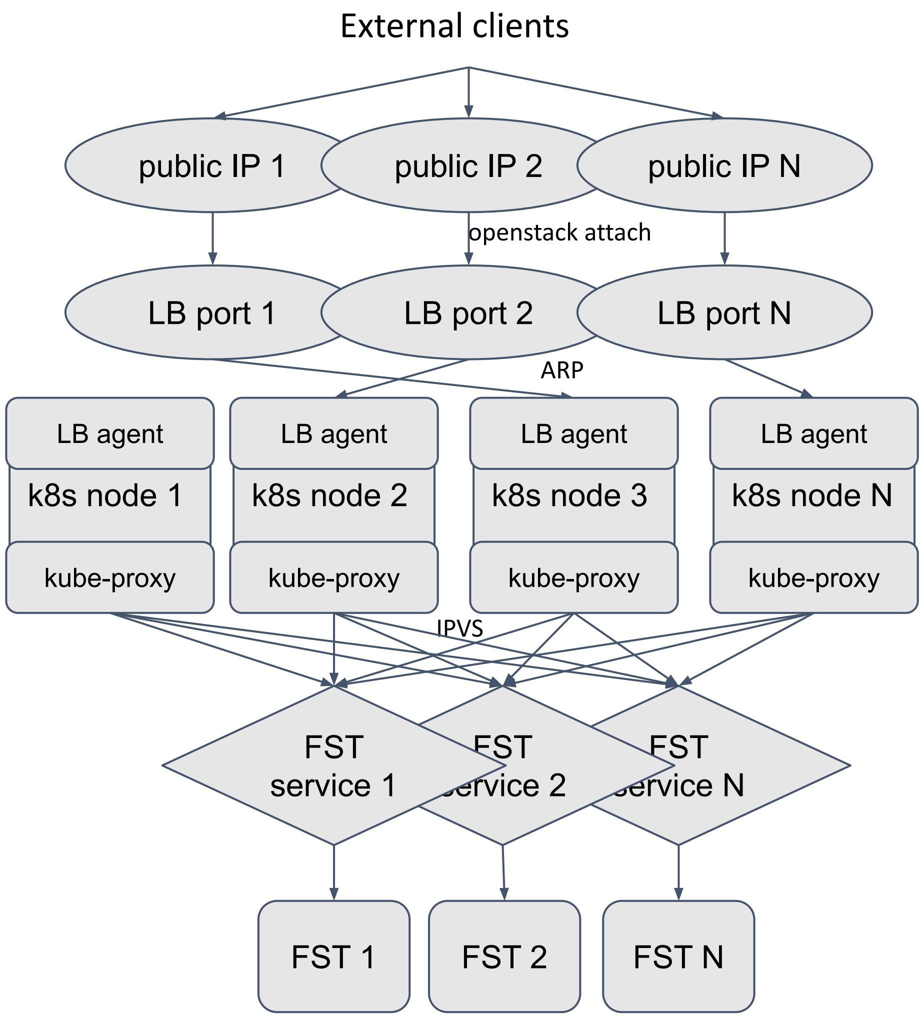
- Simple architecture for typical k8s app
  - Web app, minimal bandwidth
  - Single ingress IP
  - Ingress controller and LBaaS
  - L7 (HTTP) routing
- Won’t work for EOS
  - Need to scale bandwidth >> 1 NIC
  - FSTs need to be individually addressable
  - Ingress (Traefik) can do L4 (TCP) routing
    - but only with SNI (Server Name Indication)
    - XrootD can not support SNI



Network architecture on k8s for EOS

- One LB service for each of N FSTs
    - Total bandwidth = 1 NIC * N
    - L3 routing: 1 IP per FST
    - Ingress controller not a bottleneck
    - Solves multi-homing
        - With hostAliases (/etc/hosts)

- Challenges
    - Requires manual certificate management
        - instead of using Ingress (certs-aaS)
    - Need a way to specify FST hostnames [#7]()
        - Since EOS does its own routing/redirection



EOS CephFS layout on k8s

“Usual” way: separate volume per FST

- /volume01
    - /volume01/fst01
- /volume02
    - /volume02/fst02
- /volume03
    - /volume03/fst03
- …

Instead try one volume for all FSTs

- /volume01
    - /volume01/fst01
    - /volume01/fst02
    - /volume01/fst03
    - …

- Hopefully simplify cloud volume management
- Facilitate direct data access for compute jobs on k8s cluster
- Ideal: any/all FSTs on CephFS could serve data together
    - proxy groups?
- CephFS bug encountered: 55090
    - Ceph fixes: #46902 #46905

Summary

- All services/resource for UVic T2 are on k8s, except storage
- Developing proof-of-concept EOS SE deployment with k8s, CephFS
- Enhancements of EOS Helm chart
- Scalable k8s network architecture for external access to EOS
  - Need a way to specify FST host names

Why Kubernetes?

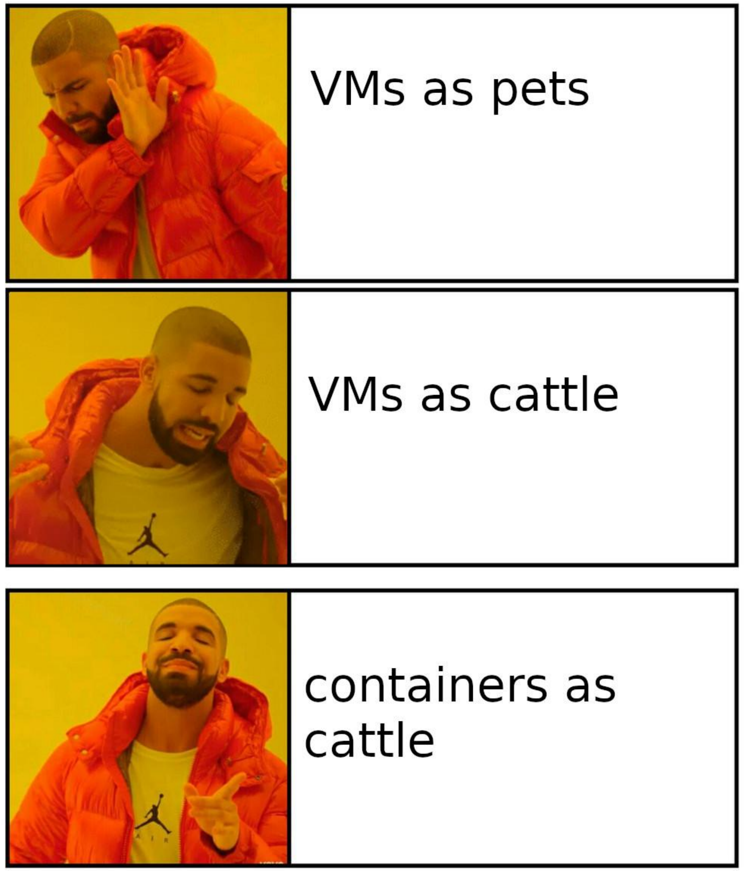
- We are a cloud site

Arbutus Science Cloud

ARBUTUSCLOUD
OPEN FOR RESEARCH
vCPUs
vGPUs
MASSIVE STORAGE
https://computecanada.ca/research-portal/national-services/compute-canada-cloud
compute canada | calcul canada
University of Victoria

openstack ceph

- Cloud + k8s provides:
  - Flexible & dynamic infrastructure
  - Resilience and automated remediation
  - Rapid application deployment
  - Application lifecycle management
  - Horizontal scalability



Openstack

Openstack + ???

Openstack + k8s

Prior talks on UVic k8s T2

- 2019 Nov CHEP
- 2019 Dec pre-GDB
- 2020 Dec k8s HEP meetup
- 2020 Dec WFM SW TIM
- 2021 May ADC TCB
- 2022 June pre-GDB
- 2022 Nov WLCG workshop
- 2022 Dec US ATLAS Computing Facilities F2F

Ingress and LBaaS

- Initial basic approach used keepalived and nginx-ingress to receive traffic from outside world into clusters
- Migrated to PureLB and Traefik
  - More maintainable/manageable, via Helm charts
  - Cohesive access to dashboards etc across all clusters
- PureLB: like MetalLB but simpler, lightweight
  - relies on Linux network stack of host
  - Programmable (LB -> LBaaS)
- Traefik Ingress controller
  - Widely used, full featured, nice web UI, CRDs
  - Better TCP and UDP support