

# Building a fully cloud-native ATLAS Tier 2 on Kubernetes

Ryan Taylor, Jeff Albert, Fernando Harald Barreiro Megino  
on behalf of the ATLAS Computing Activity



**University  
of Victoria**



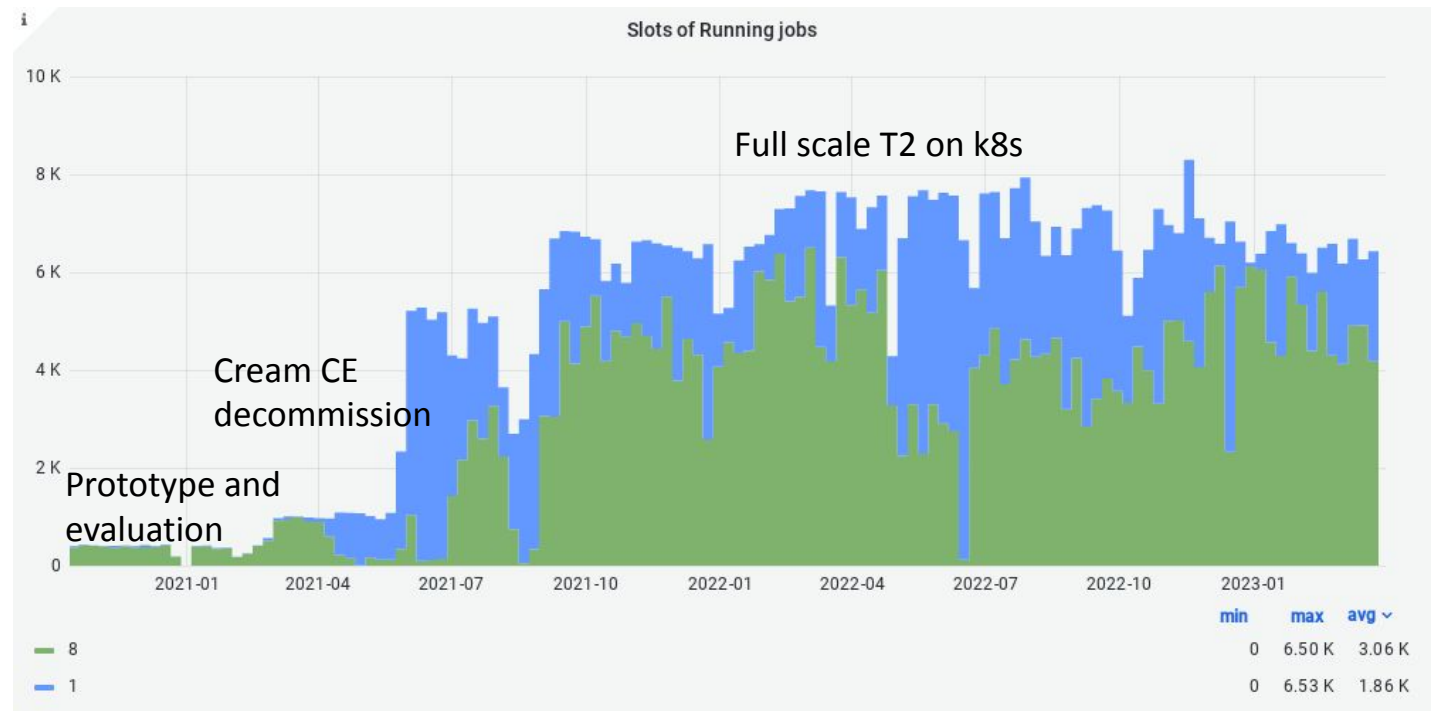
# Background

## [CHEP 2019 presentation](#)



### Using Kubernetes as an ATLAS computing site

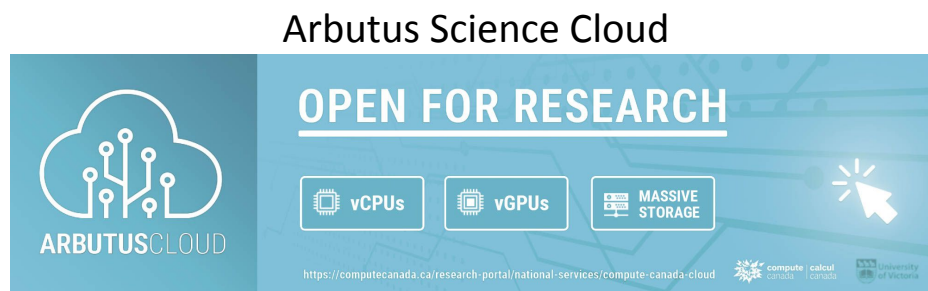
*Fernando Barreiro Megino, Jeffrey Ryan Albert, Frank Berghaus, Danika MacDonell, Tudashi Maeno, Ricardo Brito Da Rocha, Rolf Seuster, Ryan P. Taylor, Ming-Jyuan Yang on behalf of the ATLAS experiment  
CHEP 2019, Adelaide, Australia*






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.

# Why Kubernetes?

- We are a cloud site



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

	VMs as pets	Openstack
	VMs as cattle	Openstack + ???
	containers as cattle	Openstack + k8s

# The eventual goal: a fully k8s-native T2

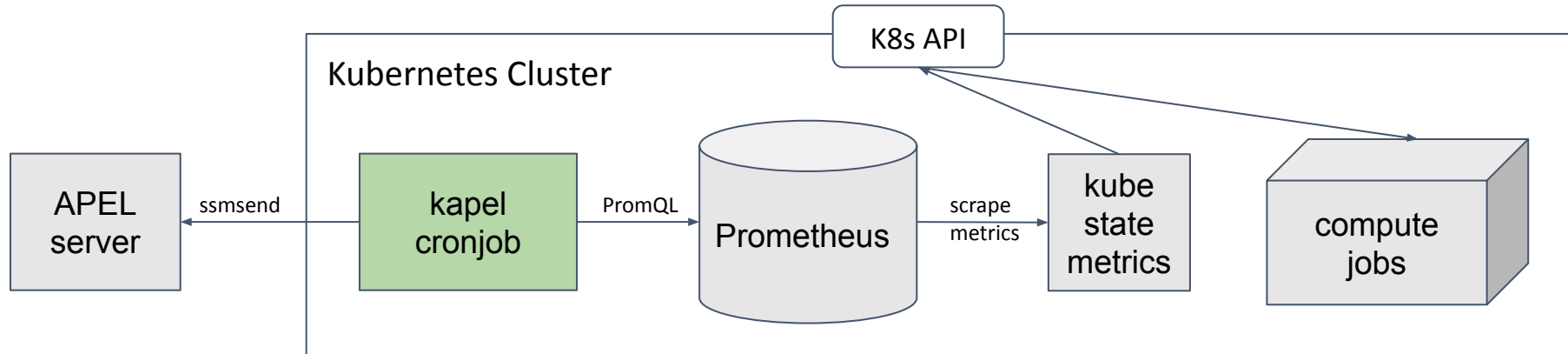
## Installable with Helm



- 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

# KAPEL

## Container-native APEL accounting for Kubernetes

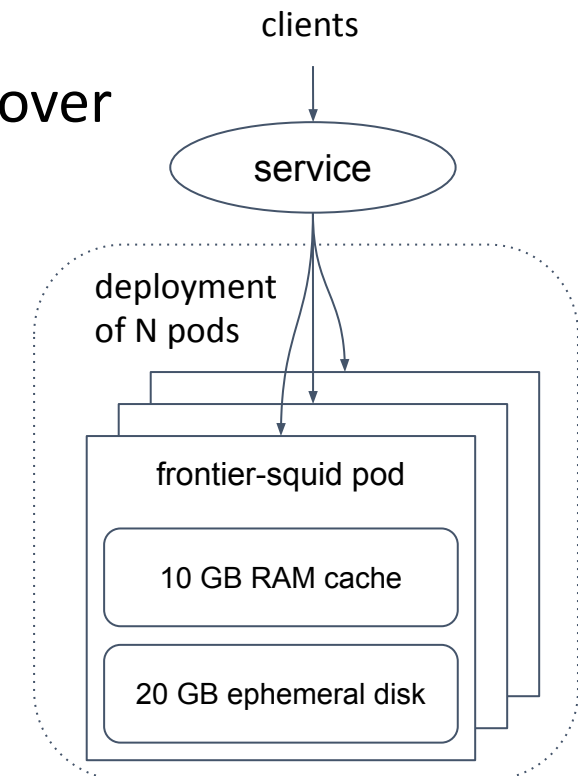


- Standard k8s add-ons do most of the work
  - k8s cron job instead of APEL node
  - Prometheus instead of MySQL DB for data collection and storage
  - PromQL for data querying, analytics
  - kube-state-metrics (KSM) instead of batch log parser
  - Only needed to write ~200 lines of python (and some YAML)
- Available as Helm chart: <https://github.com/rptaylor/kapel>

# Frontier-squid

## Deployed on Kubernetes

- Using frontier-squid [Helm chart](#) from CERN ScienceBox
  - Simple, lightweight, container-native approach
  - Trivial to scale, with automatic load-balancing and failover
- UVic contributed enhancements
  - Run as unprivileged squid user [#61](#)
  - Allow configuration of service details [#63](#)
  - Support for priorityClass and pod resource requests/limits [#64](#)
  - Send access logs to stdout [#69](#)
  - Configurable ACL activation [#72](#)
  - Harmonize configuration with upstream package [#73](#)
  - Add backup readiness probe URL for redundancy [#74](#)
  - Update ACLs for Frontier servers [#78](#)
  - Expand list of safe ports [#81](#)
- Suitable for new CVMFS proxy sharding feature



# EOS SE on k8s with CephFS

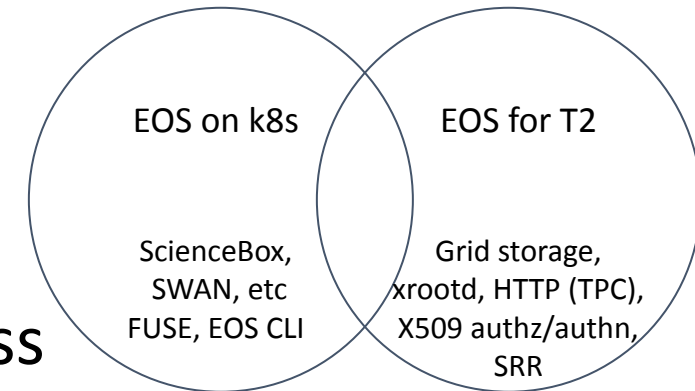


- Physical consolidation: all storage on Ceph
- Logical consolidation: services on k8s
- EOS can be installed on k8s via Helm chart
  - reproducible, single step deployment
  - easier to manage and maintain
  - easy to set up another instance, e.g. for dev
- EOS + CephFS is an established solution
-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

# EOS SE on k8s with CephFS



- Enhancements of Helm chart for T2 use case
  - VOMS authz/authn
  - Set up host certs as secrets, fetch-crl, CAs, etc.
- Kubernetes network architecture for external access
  - A LoadBalancer Service for each storage pod (FST)
- Need a way to specify FST hostnames [#7](#)
- CephFS bug encountered: [55090](#)
  - Ceph fixes: [#46902](#) [#46905](#)
- Benchmarking and performance tuning





# Extra slides

# CVMFS proxy sharding with k8s Squids

- New feature in CVMFS v2.10 to improve cache hit rates
- CVMFS understands round-robin DNS
  - dereferences multiple A records
- Solution using k8s Services: [headless ClusterIP](#)

```
service:
```

```
  clusterIP: None
```

- Should decrease CVMFS\_DNS\_MIN\_TTL to a small value
  - CVMFS default is 1 min
  - K8s deployment upgrade could be < 1 min (and DNS TTL is 5 s)
  - Details: [#97](#)

# 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

