

# Kubernetes Tutorial / Hackathon Closeout

---

April 24-26, 2024

<https://indico.cern.ch/event/1384683/>

# Executive Summary

---

- Half-day tutorial covering Kubernetes cluster installation with Kubespray and cluster management with GitOps using Flux.
  - 20 single-node Kubernetes clusters successfully built over the course of the morning
- Two days spent hacking on a huge number of Kubernetes-related topics including:
  - Stretched Kubernetes constructed over encrypted VPN (WireGuard)
  - Workload scheduling across multiple competing namespaces (Kueue)
  - Federating existing clusters and deploying various objects across the fabric (Karmada)
  - Gathering and sending K8S job metrics for WLCG accounting (KAPEL? Kount.ly? Kuantifier?)
  - Deployment of a reproducible analysis platform on various clusters (REANA)
  - Attaching clusters to our federated BinderHub instance (<https://rp1.hl-lhc.io/>)
  - Exploring graphical Kubernetes management (Lens)
  - Monitoring and Dashboards (Prometheus)
  - Configuring Anycast DNS to minimize latency to clients of the stretched K8S (Cloudflare)
  - Deployment of ServiceX Lite transformers to K8S clusters made the same day
  - Provisioning additional nodes at IU for the stretched K8S (s1)



# Wireguard

- Encrypted VPN mesh
- Used as the control plane network for 's1' stretched Kubernetes cluster
- Working group spent time testing latency, throughput, CPU utilization
  - Can add a non-trivial amount of load when operating at line rates
- Investigating feasibility for deployment within BNL and other restrictive site networks
- Next steps: Follow up with this periodically, try to prove feasibility at BNL

```
[ 5] 35.00-36.00 sec 1.05 GBytes 9.01 Gbits/sec 0 2.14 MBytes
[ 5] 36.00-37.00 sec 1.06 GBytes 9.07 Gbits/sec 0 2.14 MBytes
```

---

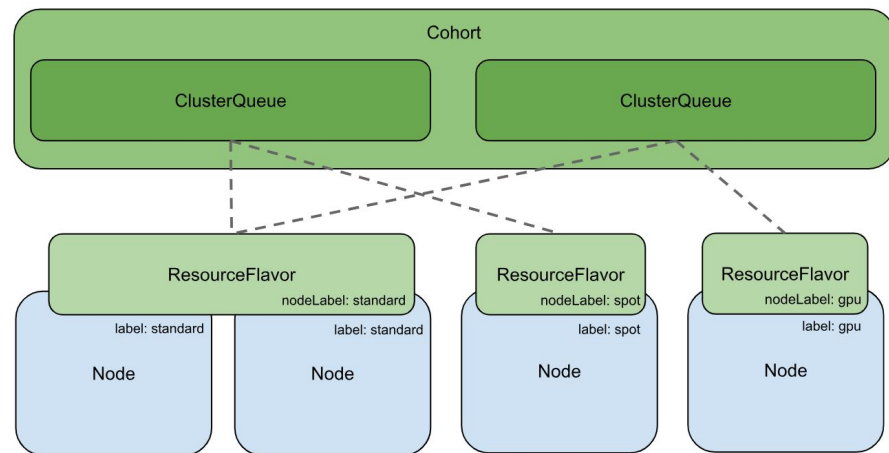
```
top - 15:17:15 up 23:15, 3 users, load average: 1.90, 1.34, 0.79
Tasks: 659 total, 4 running, 655 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.4 us, 5.8 sy, 0.0 ni, 92.5 id, 0.0 wa, 0.0 hi, 1.3 si, 0.0 st
MiB Mem : 128169.0 total, 118082.9 free, 5450.1 used, 5680.4 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used, 122718.9 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
778943	root	20	0	8928	3840	3200	R	37.4	0.0	0:13.36	iperf3
731586	root	20	0	0	0	0	I	20.9	0.0	0:26.22	kworker/5:1-wg-cr+
773528	root	20	0	0	0	0	I	17.5	0.0	0:25.75	kworker/5:4-wg-cr+
779132	root	20	0	0	0	0	I	13.6	0.0	0:03.61	kworker/5:2-wg-cr+
759661	root	20	0	0	0	0	I	8.9	0.0	0:19.76	kworker/5:0-wg-cr+
2422	root	20	0	5894628	1.7g	1.5g	S	5.0	1.3	24:52.88	falcon-sensor-b
766952	root	20	0	0	0	0	I	4.3	0.0	0:02.13	kworker/25:0-wg-c+
769034	root	20	0	0	0	0	I	4.3	0.0	0:02.95	kworker/30:2-wg-c+
769539	root	20	0	0	0	0	I	4.3	0.0	0:02.04	kworker/34:2-wg-c+

```
[116] 0:root@093:/etc/wireguard* 1:root@uchicago001:> "uct2-int.mwt2.org" 15:17 24-Apr-24
```

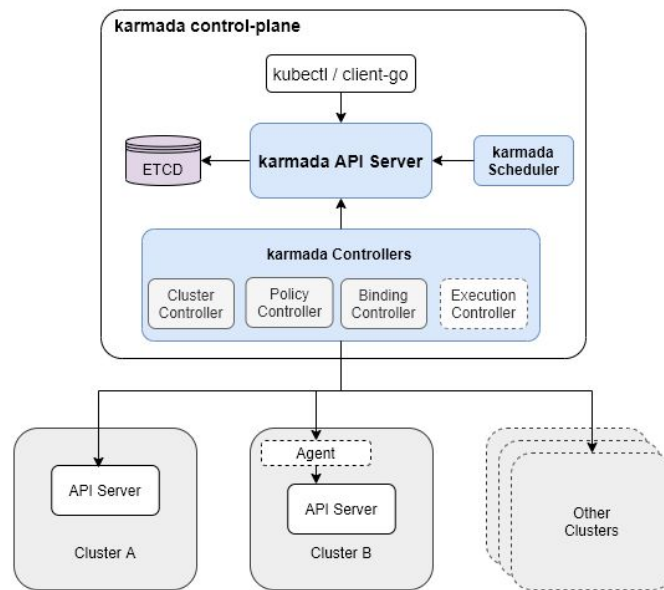
# Kueue

- APIs and controller for Job queuing in Kubernetes
- Promising technology, a lot of momentum in the community
- Proof of Concept cluster built by WG:
  - Larger tutorial-style Kubespray cluster built with ~5 nodes, ~100 cores
  - Gathered familiarity with the concepts (ClusterQueues, Cohorts, ResourceFlavors, etc) and how they map onto familiar HPC/HTC concepts in our community
  - Created and configured two artificial workload groups, 'usatlas' and 'osg'
  - Submitted sleep-like jobs for each workload group, 40 and 20 cores respectively.
- Next steps: Try OSG and ATLAS on a prod cluster?



# Karmada

- Multi-cluster Kubernetes federation platform
  - CNCF incubation project
  - Spiritual successor of KubeFed et al
- Results from the WG:
  - Installed all of the CLI tools on tutorial bastion
  - Control plane on a tutorial host as the leader of the federation
  - Joined 4 of the WG's tutorial clusters to the pool
  - Deployed an nginx example on each cluster, scaling up and scaling down
  - Investigated different mechanisms (Push vs Pull mode)
- Next steps: Try between some of our prod clusters?



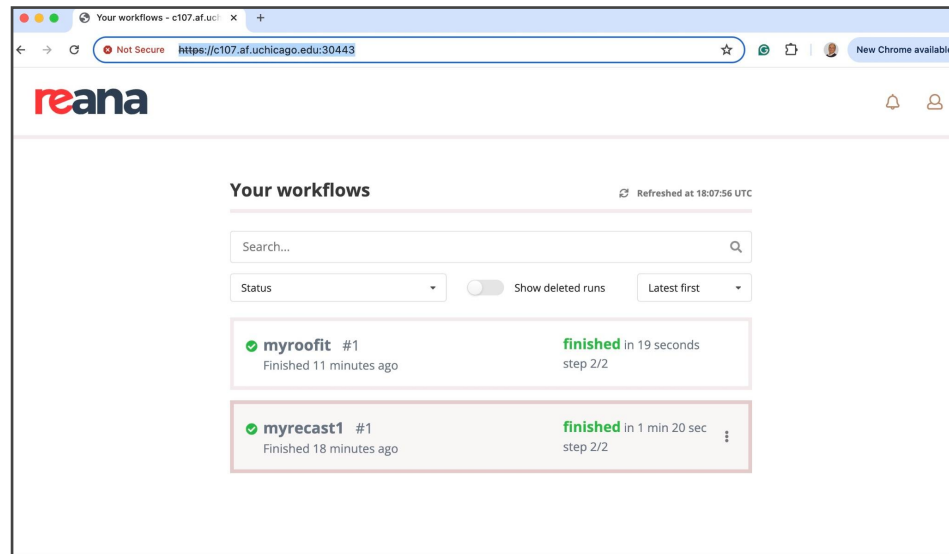
# Job Metrics

---

- KAPEL processes metrics from Prometheus and puts them into a format suitable for APEL
- Team worked on a way to ship those metrics from KAPEL directly into GRACC, which then reports to APEL
- Test OSPool Glideins setup and installed on Tiger
- Metrics are being gathered via KAPEL
- OSG already runs a probe for glideins, so results will be compared
- At OKD sites, additional authorization needed and is being worked on
- Perhaps in production in May.

# REANA

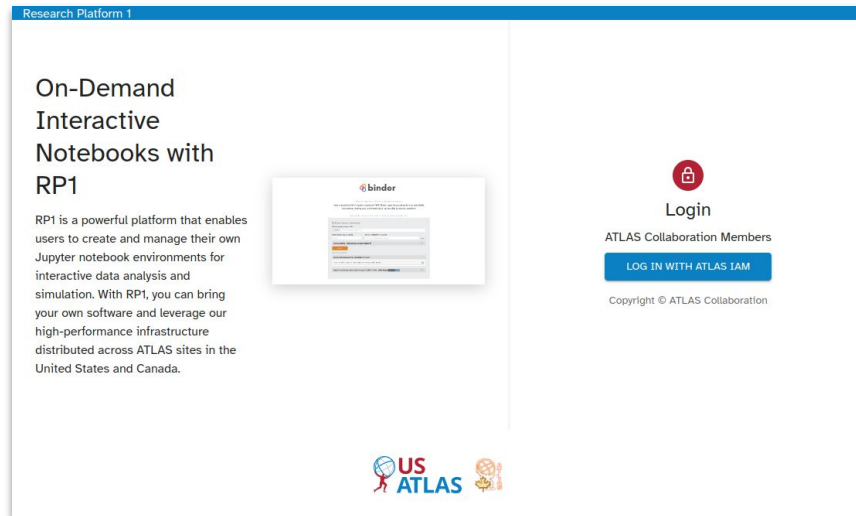
- REANA deployed on Kubespray tutorial node
- Was able to run some canned analyses on the REANA deployment
- Role needed to deploy REANA sorted out, deployed on river-dev
- To deploy on RP1 or UC AF:
  - For 'prod' will be a tenant in Flux, i.e. deployed via GitOps
  - Need to add federated authentication (ATLAS IAM)
  - Understand how to collect metrics
    - Built-in ? Prometheus ?





# RP1

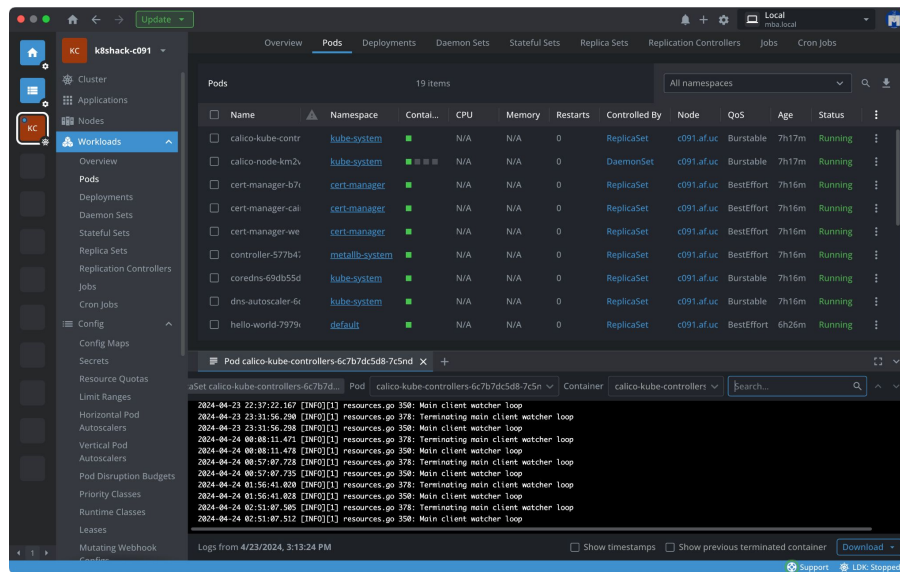
- Binder-based platform for training events, e.g. for the US ATLAS Summer Workshop in Seattle
- Instructions written on how to join RP1
- Were able to add s1 and 3 Kubespray test clusters into RP1 Binder
- Connected to ATLAS IAM
- To follow up:
  - Can we add NET2?

A screenshot of the Research Platform 1 (RP1) landing page. The page has a blue header with the text "Research Platform 1". The main content area is white and contains the following elements:

- On-Demand Interactive Notebooks with RP1**: A heading followed by a paragraph: "RP1 is a powerful platform that enables users to create and manage their own Jupyter notebook environments for interactive data analysis and simulation. With RP1, you can bring your own software and leverage our high-performance infrastructure distributed across ATLAS sites in the United States and Canada."
- Binder Screenshot**: A small screenshot of the Binder web interface showing a list of available notebook environments.
- Login Section**: A red lock icon, the text "Login", "ATLAS Collaboration Members", a blue button labeled "LOG IN WITH ATLAS IAM", and the text "Copyright © ATLAS Collaboration".
- Logos**: The US ATLAS logo and the ATLAS logo are positioned at the bottom right of the page.

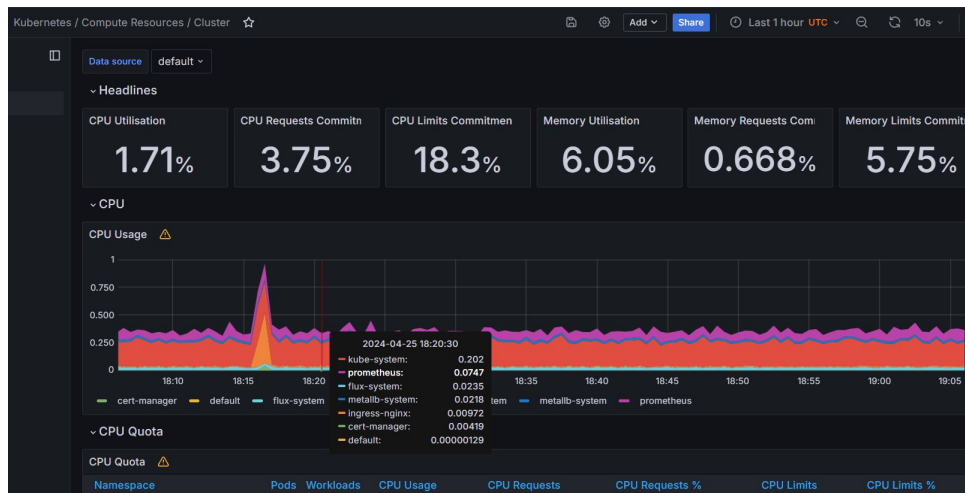
# Lens

- Desktop application, Electron-based
- Graphically navigate Kubernetes cluster
- Can optionally install Prometheus on your cluster to gather metrics
- Quickly get a terminal on a Pod, easily jump between multiple clusters etc
- Easy to filter, select objects, delete broken things etc.



# Monitoring and dashboards

- WG installed Prometheus via the 'Prometheus Stack' Helm chart, including Grafana
- Also installed via Flux HelmRelease
- Modified the 'hello world' ingress from the tutorial to make a Grafana ingress
- Gathered and plotted metrics for test clusters
- Next steps: Learn PromQL?



# Cloudflare Anycast DNS

- Cloudflare offers latency / geo-aware DNS resolution
- We combine this with our stretched cluster 's1' to provide access to services based on location
- E.g., 'curl [rp1.hl-lhc.net:8080](http://rp1.hl-lhc.net:8080)' will respond with information from the closest node to your location
- Next steps: XCache, Varnish ? e.g. 'varnish.usatlas.org' ?

```

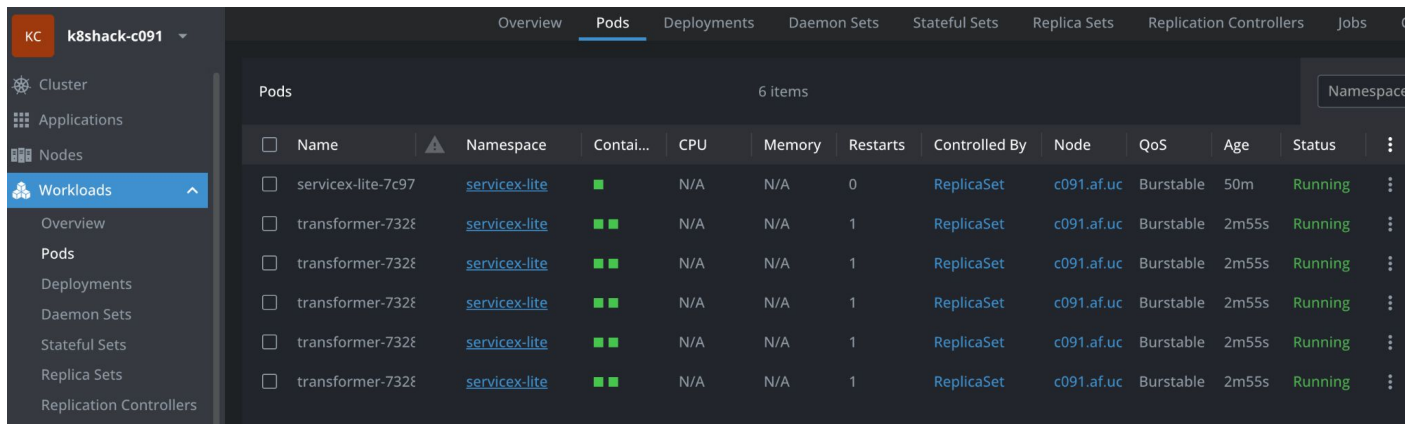
~: bash — Konsole
File Edit View Bookmarks Plugins Settings Help
[09:04]:~ $ curl -s http://rp1.hl-lhc.net:8080 | jq
{
  "host": {
    "hostname": "rp1.hl-lhc.net",
    "ip": "::ffff:172.69.7.3",
    "ips": []
  },
  "http": {
    "method": "GET",
    "baseUrl": "",
    "originalUrl": "/",
    "protocol": "http"
  },
  "request": {
    "params": {
      "0": "/"
    }
  }
}

```

**Cloudflare Chicago endpoint**

# ServiceX Lite Deployment

- Small application that listens for events in the 'main' ServiceX and runs them on a separate cluster
- We were successfully able to put ServiceX Lite transformers on clusters created during the workshop
- These transformers ran real workloads and returned real results to users



Name	Namespace	Contain...	CPU	Memory	Restarts	Controlled By	Node	QoS	Age	Status
servicex-lite-7c97	servicex-lite	■	N/A	N/A	0	ReplicaSet	c091.af.uc	Burstable	50m	Running
transformer-7328	servicex-lite	■■	N/A	N/A	1	ReplicaSet	c091.af.uc	Burstable	2m55s	Running
transformer-7328	servicex-lite	■■	N/A	N/A	1	ReplicaSet	c091.af.uc	Burstable	2m55s	Running
transformer-7328	servicex-lite	■■	N/A	N/A	1	ReplicaSet	c091.af.uc	Burstable	2m55s	Running
transformer-7328	servicex-lite	■■	N/A	N/A	1	ReplicaSet	c091.af.uc	Burstable	2m55s	Running
transformer-7328	servicex-lite	■■	N/A	N/A	1	ReplicaSet	c091.af.uc	Burstable	2m55s	Running

# Additional nodes added to S1

---

- Five workers (R630) drained and rebuilt at IU with AlmaLinux 9
- Wireguard installed and configured to join the VPN mesh
- Will finish up today

# Building Expertise



- In US ATLAS Facilities R&D we are planning on creating a formal professional development program on many topics touched on here
- We will update you if this gets formalized and funded

***Certified Kubernetes Administrator (CKA)***

***Certified Kubernetes Application Developer (CKAD)***

***Certified Kubernetes Security Specialist (CKS)***

Kubernetes and Cloud Native Associate (KCNA)

Kubernetes and Cloud Security Associate (KCSA)

Prometheus Certified Associate (PCA)

Istio Certified Associate (ICA)

Cilium Certified Associate (CCA)

Certified Argo Project Associate (CAPA)

GitOps Certified Associate (CGOA)



## CKA Curriculum

### 25% - Cluster Architecture, Installation & Configuration

- Manage role based access control (RBAC)
- Use Kubeadm to install a basic cluster
- Manage a highly-available Kubernetes cluster
- Provision underlying infrastructure to deploy a Kubernetes cluster
- Perform a version upgrade on a Kubernetes cluster using Kubeadm
- Implement etcd backup and restore

### 15% - Workloads & Scheduling

- Understand deployments and how to perform rolling update and rollbacks
- Use ConfigMaps and Secrets to configure applications
- Know how to scale applications
- Understand the primitives used to create robust, self-healing, application deployments
- Understand how resource limits can affect Pod scheduling
- Awareness of manifest management and common templating tools

good for  
newcomers

### 20% - Services & Networking

- Understand host networking configuration on the cluster nodes
- Understand connectivity between Pods
- Understand ClusterIP, NodePort, LoadBalancer service types and endpoints
- Know how to use Ingress controllers and Ingress resources
- Know how to configure and use CoreDNS
- Choose an appropriate container network interface plugin



# Reference Material

---

- Google drive folder of materials [here](#)
- The main tutorial document is [located here](#)
- Tutorial recordings [here](#)
- Hack Brainstorming and linked group documents [here](#)
- Slack channel [here](#)



**safe travels home!**

