The Elastic Analysis Facility at Fermilab

Lindsey Gray (slides from Maria P. Acosta) – On behalf of the EAF team

subMIT internal workshop @ MIT

January 31st , 2024

Elastic Analysis Facility team

- Burt Holzman – Project lead
- Maria Acosta – Technical lead for applications
- Chris Bonnaud – Technical lead for infrastructure
- Elise Chavez
- Melis Erkinbaev

- Dave Mason
- Joe Boyd
- Glenn Cooper
- Lindsey Gray
- Nick Smith
- Farrukh Khan
- Ed Simmonds

Outline

- Infrastructure
    - OKD4 & Fedora CoreOS
    - OKD installation
    - Cluster specs
    - Redundant clusters
- Applications
    - Fundamental principles
    - Security
    - Multi-VO support
    - EAF applications ecosystem (Dask Gateway on EAF & Triton autoscaling)
    - DevOps (operational sustainability)
    - Active collaboration
- Summary and questions

Infrastructure

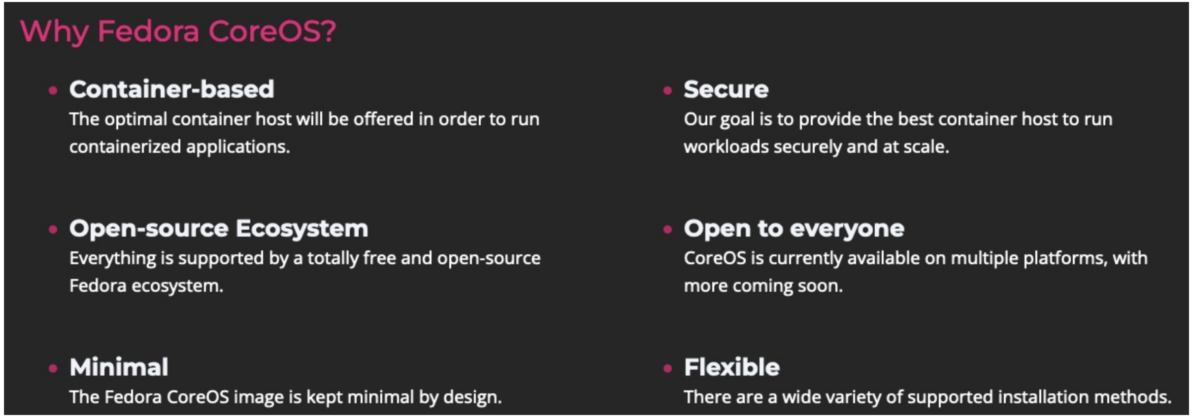
OKD4

- Open-source version of RedHat's OpenShift Container Platform, maintained by the community.
- Based on Vanilla Kubernetes, provides many features out of the box:
  - Multi-tenancy and security
  - SDN (configures an overlay network using Open vSwitch)
  - Ingress
  - CI/CD
  - GUI
  - System monitoring
- Use CRI-O as container runtime

Fedora CoreOS

- OKD requires Fedora CoreOS for all hardware nodes in the cluster.
- Minimal OS, designed specifically for running containers.
- Mostly immutable

  Configuration done by ignition file during installation
- Cannot be managed via Puppet

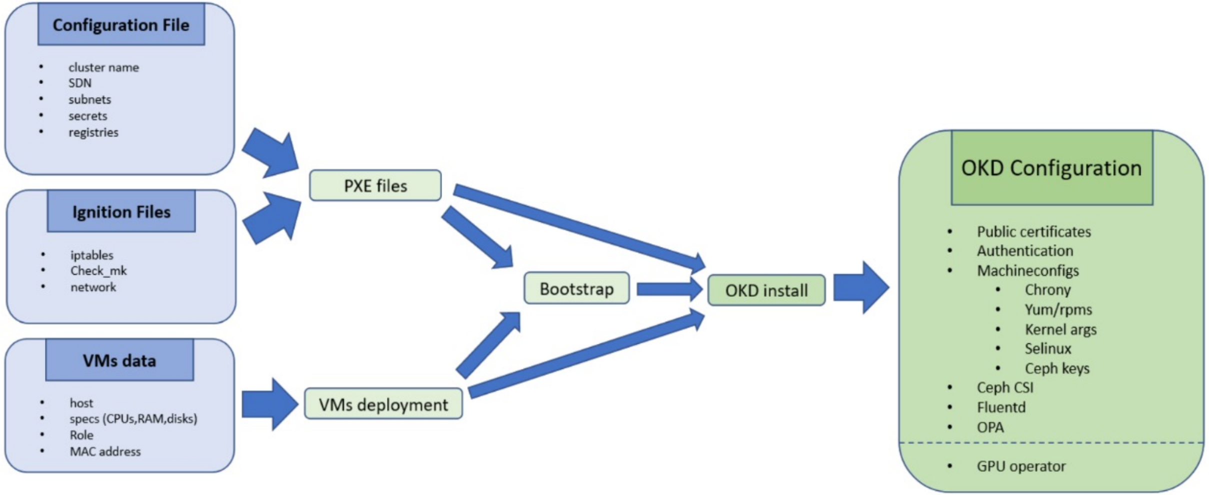


Why Fedora CoreOS?

- **Container-based**
  The optimal container host will be offered in order to run containerized applications.
- **Secure**
  Our goal is to provide the best container host to run workloads securely and at scale.
- **Open-source Ecosystem**
  Everything is supported by a totally free and open-source Fedora ecosystem.
- **Open to everyone**
  CoreOS is currently available on multiple platforms, with more coming soon.
- **Minimal**
  The Fedora CoreOS image is kept minimal by design.
- **Flexible**
  There are a wide variety of supported installation methods.

OKD4 installation

- All OKD nodes are deployed in the form of VMs (libvirt/kvm on standard linux host)
- Configuration files: static files from Git + Dynamic files generated by puppet
- Single bash script to go through all steps
- Evaluating existing tools (ArgoCD, terraform…) to consolidate the installation process



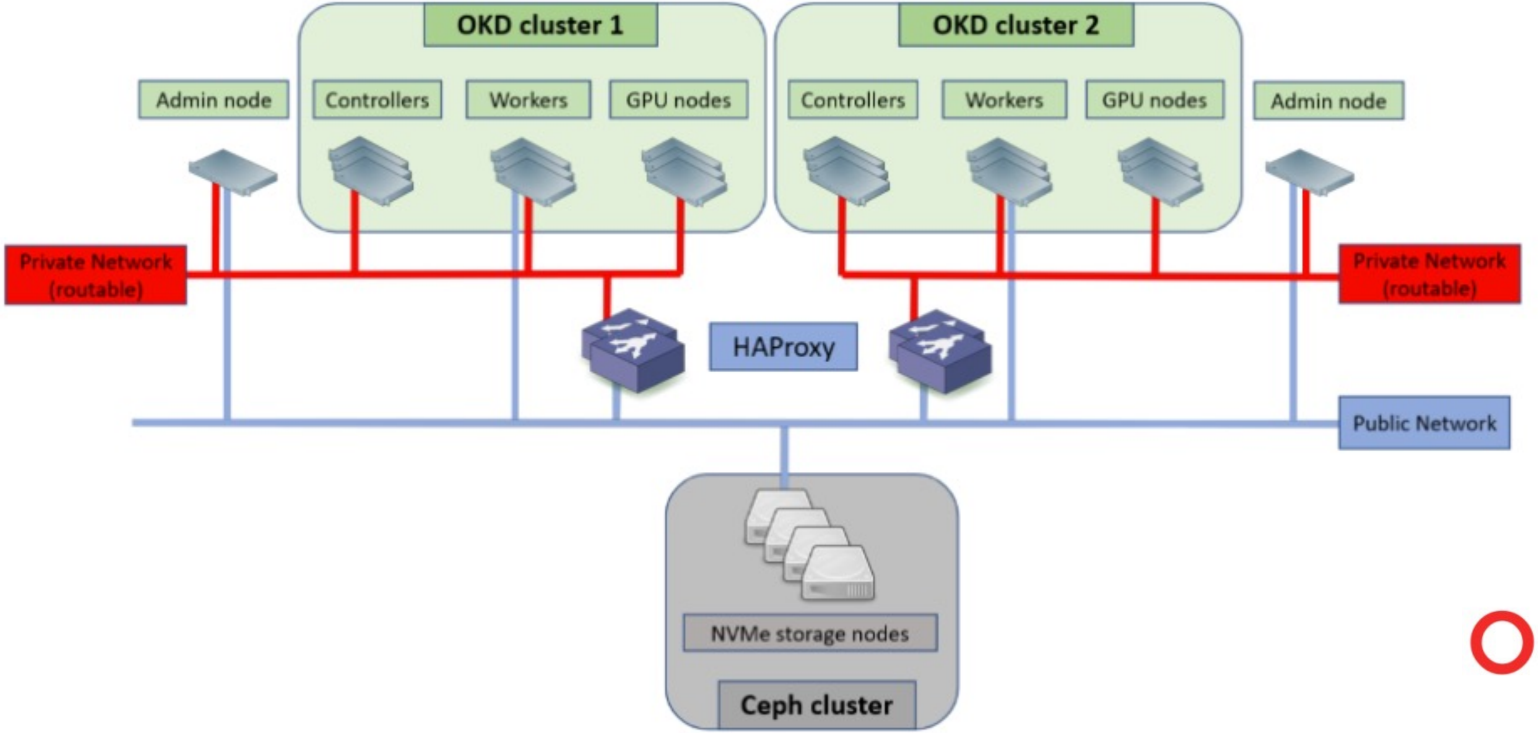
Cluster specifications

- OKD Dev
  - 3 controllers (4 cores, 16GB RAM, connection 10Gb/s)
  - 3 workers (22 cores, 88GB RAM, 10Gb/s connection)
  - 2 A100 servers (62 cores, 480GB RAM, 100Gb/s) segmented into 30 multi-instance GPU partitions
  - 4 old GPU nodes used as simple worker nodes (15 cores, 100GB RAM. 1Gb/s connection)
  - Running Kubernetes v1.23.5, Fedora CoreOS 35 and cri-o://1.23.3
- OKD Prod
  - 3 controllers (38 cores, 180GB RAM, 10Gb/s connection)
  - 3 workers (78 cores, 360GB RAM , 100Gb/s connection)
  - A100 GPU nodes will be migrated

Redundant clusters

- Production environment needs to be reliable but:
  - OKD 4 is a complex product with many different components working together in the background
  - No Red Hat support, troubleshooting issues could take days/weeks
  - Upgrading basically means reinstalling from scratch
- Mitigation
  - 2 production clusters
    - Second cluster can be used as a cold spare
    - Second cluster can be used to test changes without impacting production
    - Upgrade can be done by migrating users from one cluster to the other
  - EAF has been running on the development cluster (OKD4) and is scheduled to be migrated by end of 2023

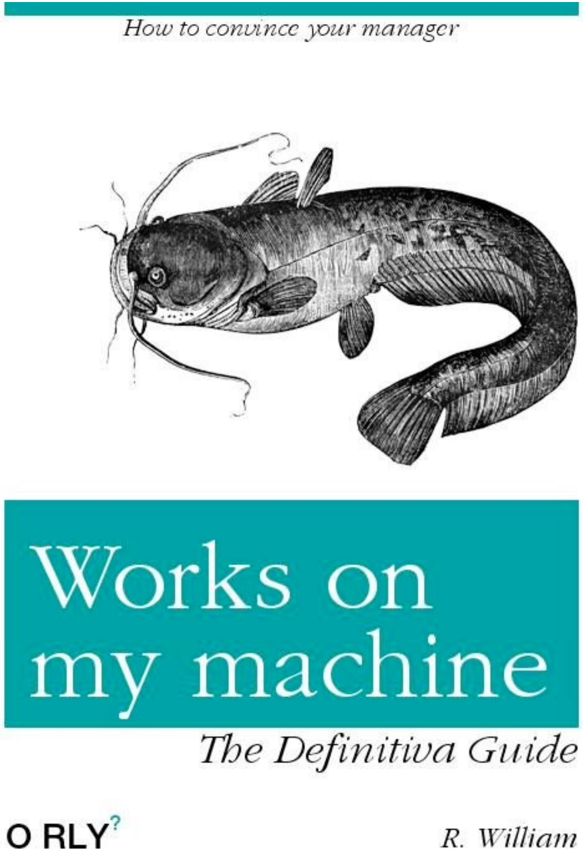
Redundant clusters



What does an EAF migration look like?

- Preparation:
  - Backup user data and any ephemeral components as needed
  - Ensure consistency between Git and current deployments, synchronize all changes and branches
  - Documentation check
  - Checklist and inventory
- Teardown of ALL components, applications, services and data
- Semi-automated deployment of application ecosystem into new cluster via Helm
- Testing and validation

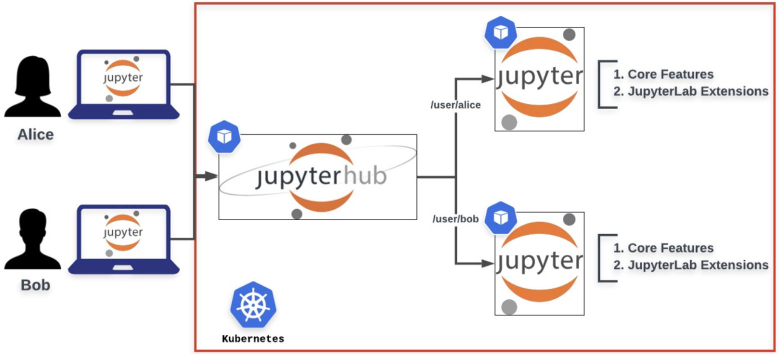
Applications



A few words about modern Scientific data analysis:

- Needs to be fast, reliable, secure, accessible + bonus points for replicability and UI/UX features.
- Requires persistent and non-persistent data storage
- Fosters collaborative environments, enables distributed teams and multi-disciplinary groups to make science using computing tools
- Work smart, improve where there's room for it.. but don't abandon the old, wise ways

A JupyterHub-based deployment



- Originally standalone Jupyter Notebooks.
- Evolved to a self-hosted, multi-user platform for hosting multiple notebooks, kernels and highly customizable environments.
- Can be deployed in multiple platforms including Cloud, on prem and Kubernetes.

✓ Implements authentication, login pages and token-based roles

✓ Tracks activity and does effective resource management

✓ Proxying is done behind the scenes

Fundamental principles

- Create a user-oriented analysis facility based on our own experience supporting scientists on traditional technologies.
- Explore, deploy and collaborate on industry-level tools and strategies for optimizing data analysis.
- Facilitate the use and access of a pool of large, specialized hardware for all Fermilab users in and Elastic way.
- Foster collaboration with experiments and science groups in order to better understand current and future analysis needs.
- Provide effective, requirement-oriented computing solutions.

Secure

Integrated & functional

Multi-VO

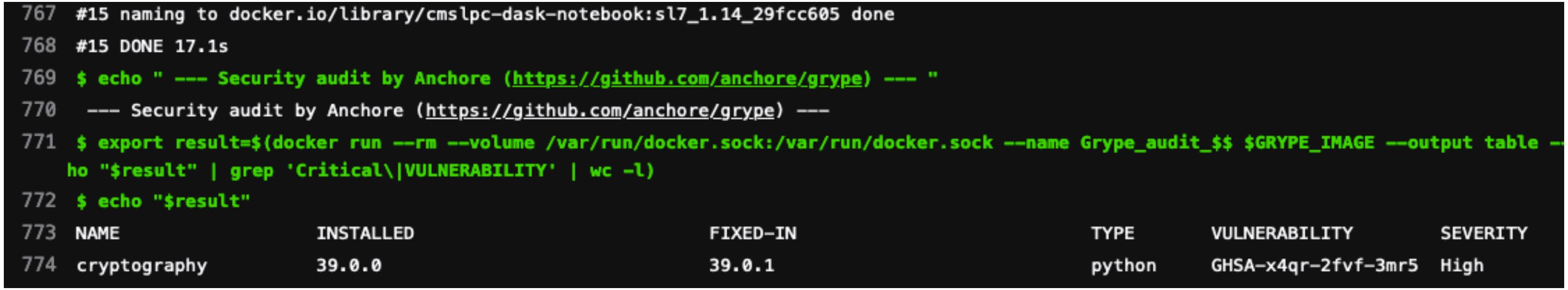
DevOps (operational sustainability)

Active collaboration

Security

Secure | Integrated & functional | Multi-VO | DevOps (operational sustainability) | Active collaboration

- JupyterHub is integrated with lab's PingFederate SSO, other apps can authenticate and authorize with JupyterHub.
- Keycloak facilitates authorization with MLFlow
- Dev instance running with FedID capabilities via CILogon
- User management done via FERRY, the central attribute repository for all Fermilab experiments.
- Docker image vulnerability scanning via Anchore Grype (https://github.com/anchore/grype)
- Ongoing security reviews with Fermilab's CST to enable offsite access.
- [Upcoming] Tailored profile options to avoid accidental access to experiment data



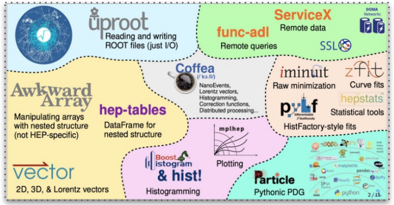
```
#15 naming to docker.io/library/cmslpc-dask-notebook:sl7_1.14_29fcc605 done
#15 DONE 17.1s
$ echo " --- Security audit by Anchore (https://github.com/anchore/grype) --- "
 --- Security audit by Anchore (https://github.com/anchore/grype) ---
$ export result=$(docker run --rm --volume /var/run/docker.sock:/var/run/docker.sock --name Grype_audit_$$ $GRYPE_IMAGE --output table --
ho "$result" | grep 'Critical\|VULNERABILITY' | wc -l)
$ echo "$result"
NAME                INSTALLED                          FIXED-IN                           TYPE      VULNERABILITY         SEVERITY
cryptography        39.0.0                             39.0.1                             python    GHSA-x4qr-2fvf-3mr5   High
```

Multi-VO support

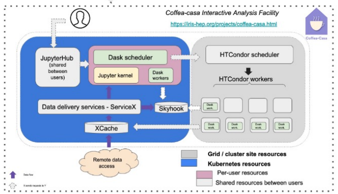
Secure | Integrated & functional | **Multi-VO** | DevOps (operational sustainability) | Active collaboration

- EAF is meant to be a facility for all experiments and science groups at FNAL.
- Robotics and acceleratorOps workloads starting to pop up, as well as heavy Astro image processing.
- Summer peak saw students from all over the lab and the world.
- New catalog display with multiple choices/options per VO
- Periodic stakeholders meeting is key to maintain communication channels open with experiments and science groups

Analysis Tools

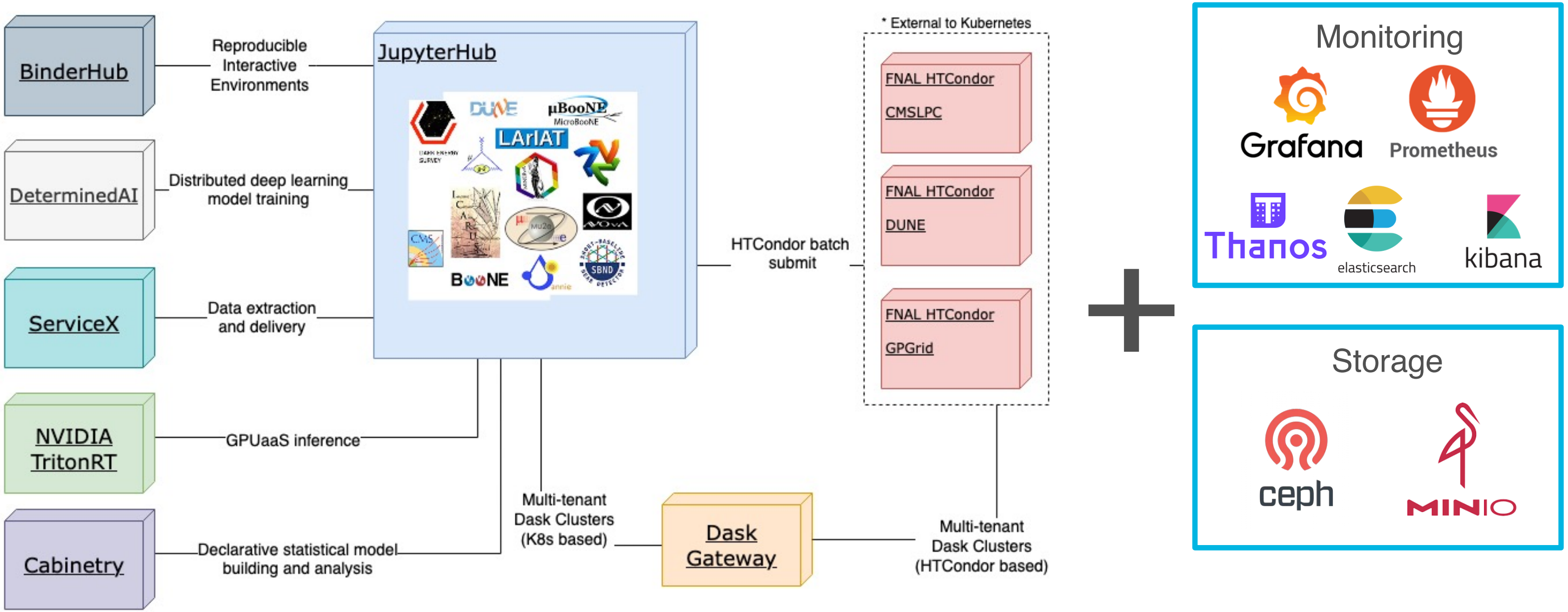


Analysis Facilities


Coffea-casa Interactive Analysis Facility
https://iris-hep.org/projects/coffea-casa.html
Coffea-Casa
JupyterHub (shared between users)
Dask scheduler
Jupyter kernel
Dask workers
HTCondor scheduler
HTCondor workers
Dask work.
Data delivery services - ServiceX
Skyhook
XCache
Remote data access
Grid / cluster site resources
Kubernetes resources
Per-user resources
Shared resources between users
Data flow
X sends requests to Y

EAF applications ecosystem

Secure | Integrated & functional | Multi-VO | DevOps (operational sustainability) | Active collaboration

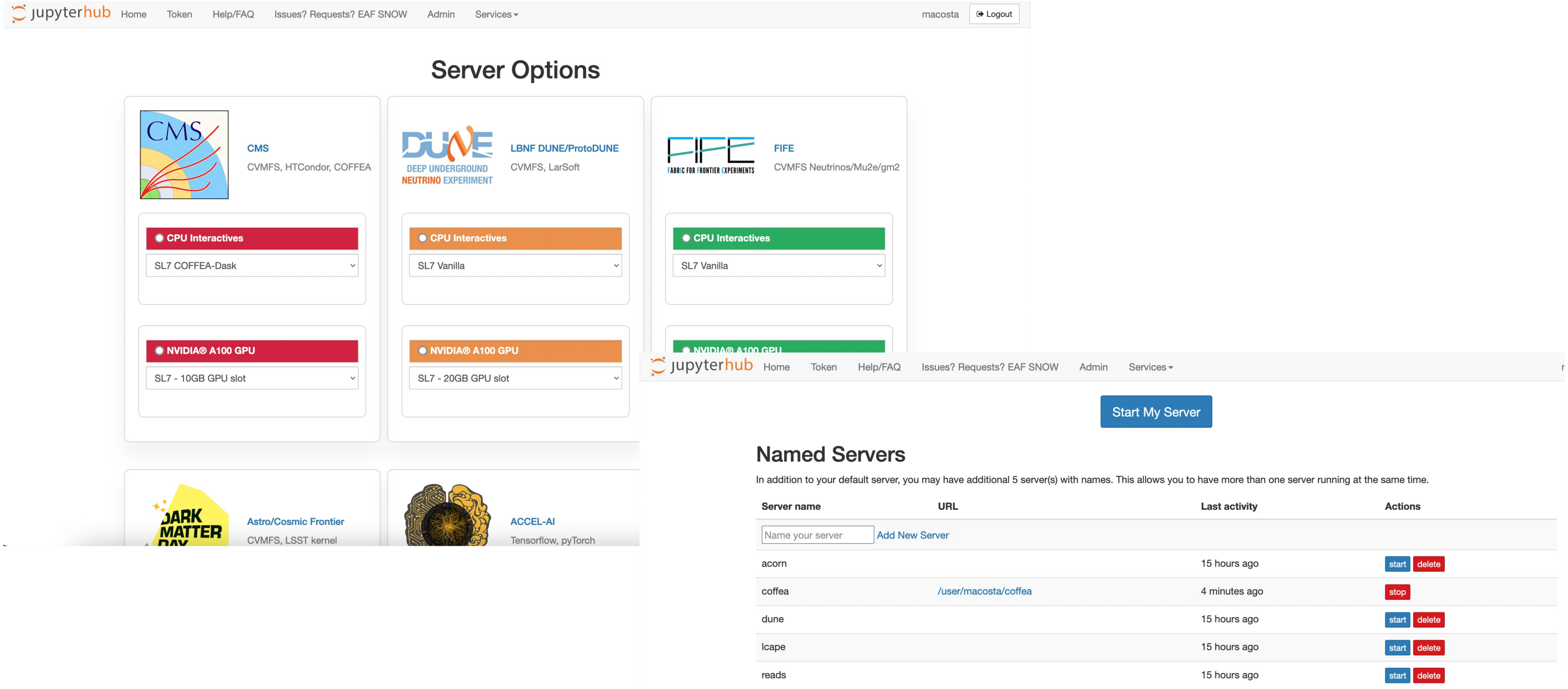


What does an EAF user get?

Secure | Integrated & functional | Multi-VO | DevOps (operational sustainability) | Active collaboration

- 25 GB cross-notebook persistent area for user storage, with UI features for file upload, download and folder management.
- Extra 40 GB scratch space for GPU notebooks
- JupyterHub extension catalog, Git labextension, Dask labextension, draw.io graphic environment
- CVMFS mounts dependent on notebook flavor
- HTCondor remote job submission to CMSLPC, FermiGrid (In progress), Wilson Cluster HPC (upcoming)
- Central laboratory NFS home areas for users /nashome/<username>
- Experiment-specific NFS areas – LPC NFS /uscms/home, /uscms/data*
- Environments tailored with experiment analysis software i.e LarSoft, CMSSW
- Up to 4 ’named servers’ running concurrently, sharing persistent area
- Access to our full applications ecosystem
- In-notebook resource usage monitoring and Landscape Grafana metrics
- Instant access to 560GB of A100 GPU memory power (divided into 10, 20, 40 GB partitions)

What does an EAF user get?


jupyterhub
Home
Token
Help/FAQ
Issues? Requests? EAF SNOW
Admin
Services
macosta
Logout
CMS
DUNE
DEEP UNDERGROUND NEUTRINO EXPERIMENT
FIFE
FABRIC FOR FRONTIER EXPERIMENTS
DARK MATTER DAY

Server Options

CMS
CVMFS, HTCondor, COFFEA

○ CPU Interactives
SL7 COFFEA-Dask

○ NVIDIA® A100 GPU
SL7 - 10GB GPU slot

LBNF DUNE/ProtoDUNE
CVMFS, LarSoft

○ CPU Interactives
SL7 Vanilla

○ NVIDIA® A100 GPU
SL7 - 20GB GPU slot

FIFE
CVMFS Neutrinos/Mu2e/gm2

○ CPU Interactives
SL7 Vanilla

○ NVIDIA® A100 GPU

Astro/Cosmic Frontier
CVMFS, LSST kernel

ACCEL-AI
Tensorflow, pyTorch

Named Servers

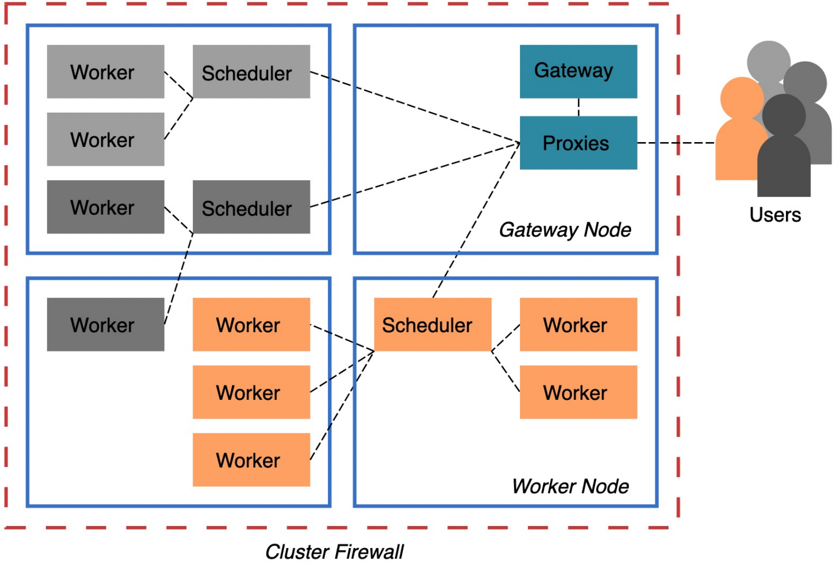
In addition to your default server, you may have additional 5 server(s) with names. This allows you to have more than one server running at the same time.

Server name	URL	Last activity	Actions
Name your server Add New Server			
acorn		15 hours ago	start delete
coffea	/user/macosta/coffea	4 minutes ago	stop
dune		15 hours ago	start delete
lcape		15 hours ago	start delete
reads		15 hours ago	start delete

A different approach to Dask: Dask Gateway (https://gateway.dask.org/)

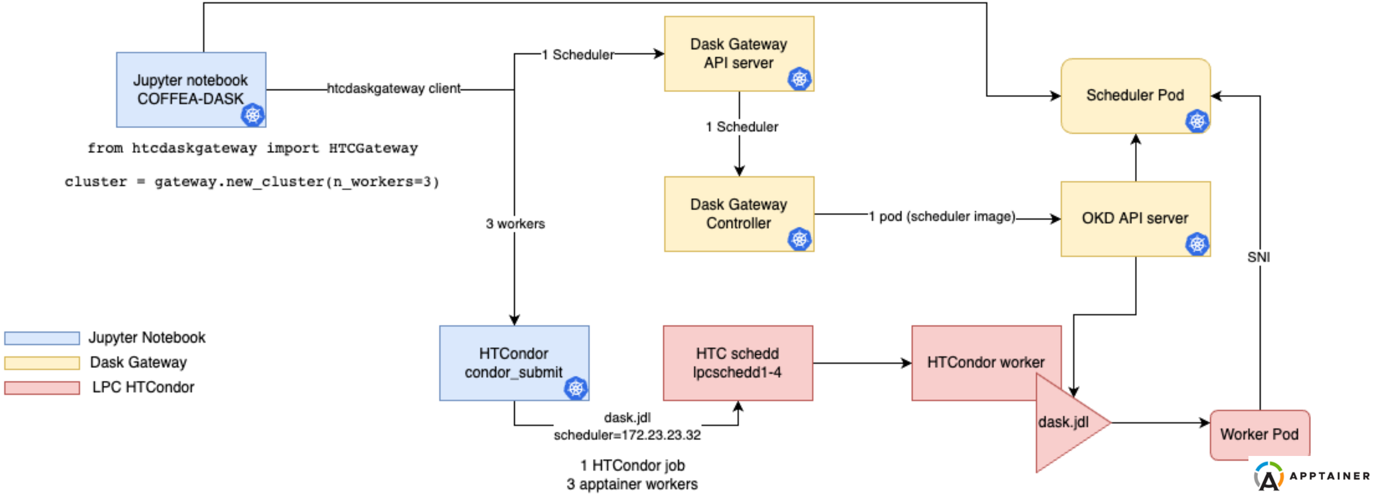
Provides a secure, multi-tenant server for managing Dask clusters. Allows users to launch and use Dask clusters in a shared, centrally managed cluster environment, without requiring users to have direct access to the underlying cluster backend (e.g. Kubernetes, Hadoop/YARN, HPC Job queues, etc…)

- ✓ Helm chart deployment
- ✓ REST api for managing clusters
- ✓ Proxy for client to scheduler traffic (TLS)
- ✓ Proxy for dashboards (HTTP)
- ✓ Flexible design
  - ✓ Configurable backend (Kubernetes, YARN, HPC, ...)
  - ✓ Configurable authentication (Kerberos, JupyterHub, ...)
- ✓ Most actions done server-side (simple client, more complicated server)



Dask Gateway on EAF

- Modified client side to perform HTCondor job submission directly from the EAF COFFEA-Dask notebook.
- Modified server side to 'outsource' scaling to HTCondor – can also scale in the form of Kubernetes pods and form hybrid clusters (experimental)
- Scheduler and Workers use COFFEA-team curated & maintained image: /cvmfs/unpacked.cern.ch/registry.hub.docker.com/coffeateam/coffea-dask-cc7-gateway:<hash>



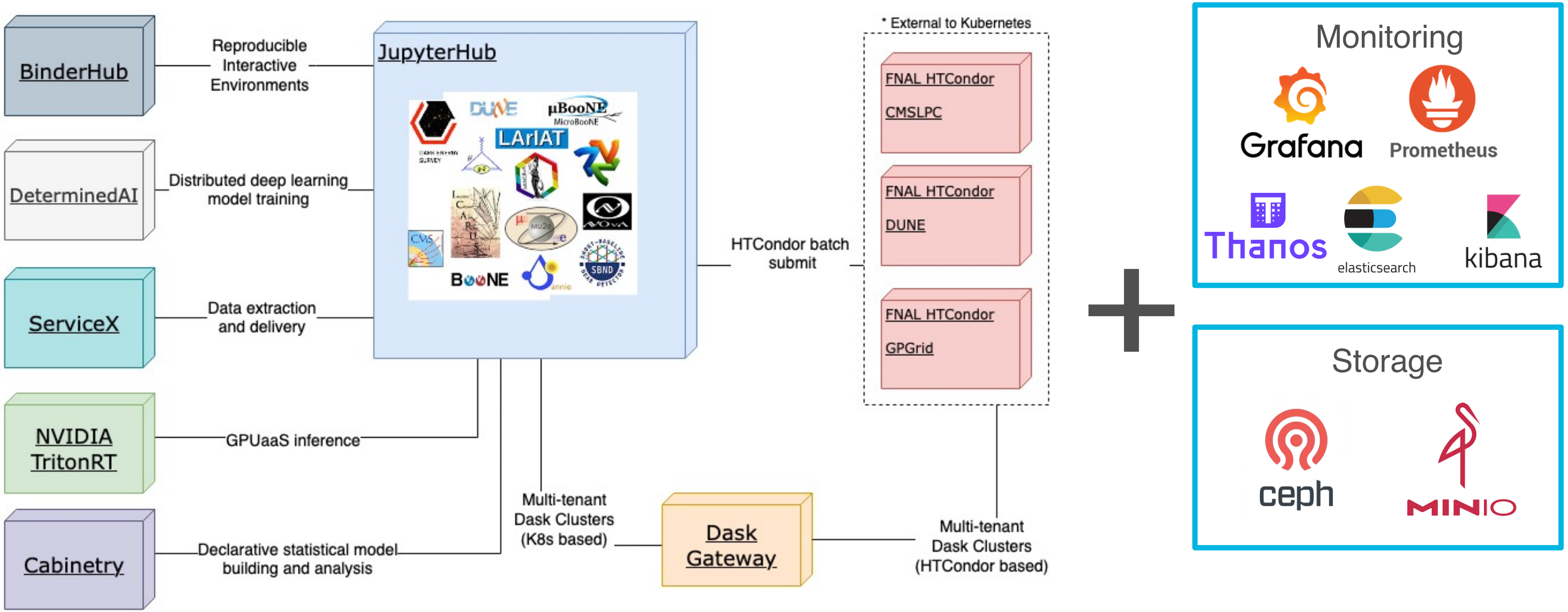
```
from htcdaskgateway import HTCGateway

cluster = gateway.new_cluster(n_workers=3)
```

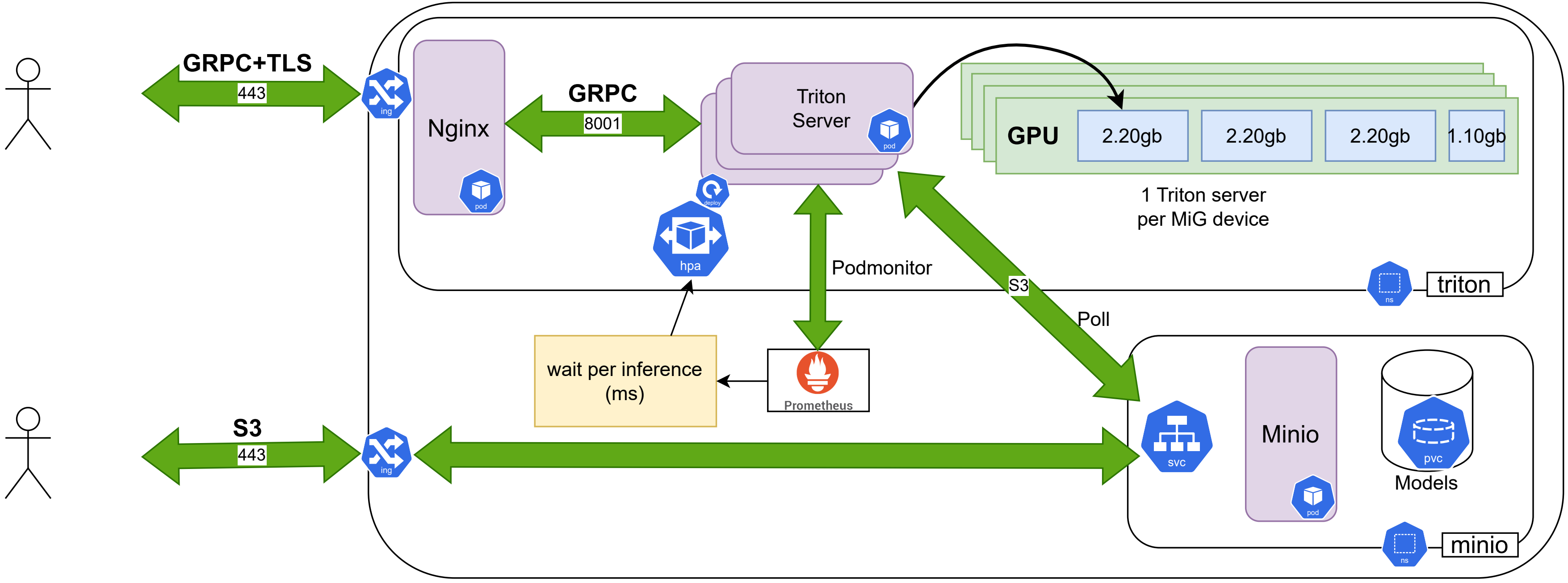
🟦 Jupyter Notebook
🟨 Dask Gateway
🟥 LPC HTCondor

EAF applications ecosystem

Secure | Integrated & functional | Multi-VO | DevOps (operational sustainability) | Active collaboration



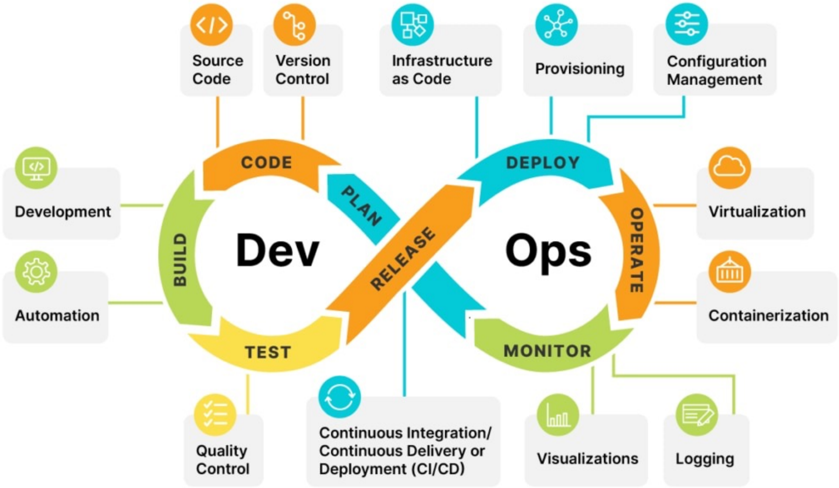
Triton Autoscaling



DevOps (operational sustainability)

Secure | Integrated & functional | Multi-VO | DevOps (operational sustainability) | Active collaboration

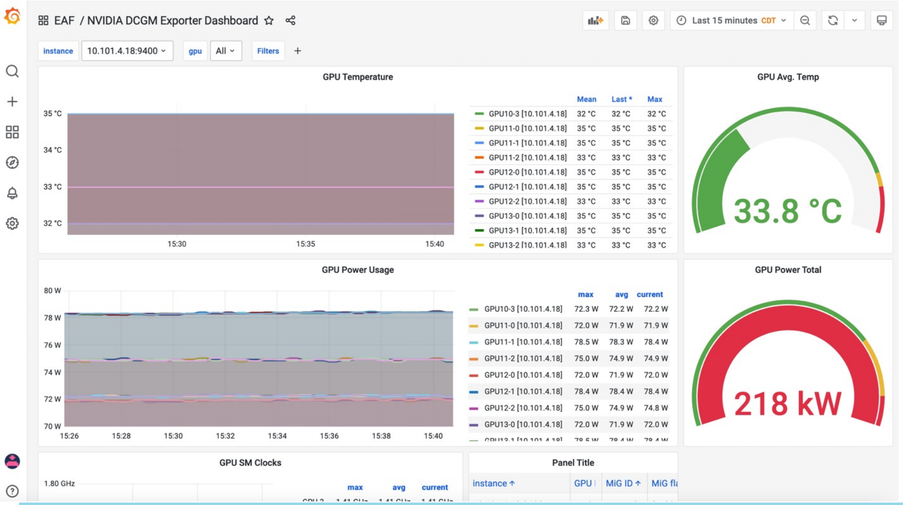
- GitOps with GitLab: Using JupyterHub Helm charts for quick deployment, rollback and configuration consistency.
- Monitoring and observability are key parts of the AF.
- Metrics, logs and events are being pushed to the lab’s monitoring platform: Landscape
- CI/CD pipelines for automated image builds including security audit, functional testing and library versions validation.
- Operational tools: Checklists, git repo documentation, pre-upgrade spreadsheets, code checks and teamwork!



Monitoring and metrics

https://landscape.fnal.gov/monitor/dashboards/f/kngVRjPVz/eaf

- Grafana + Prometheus + InfluxDB monitoring hosted at FNAL Landscape.
- GPU statistics, CPU/Memory usage, network usage per notebook, JupyterHub metrics, TritonRT inference dashboards
- Having trouble on EAF? Check the status page, JupyterHub may be having trouble! (Hint: look for spikes in 400 or 500 HTTP errors)



GPU Temperature

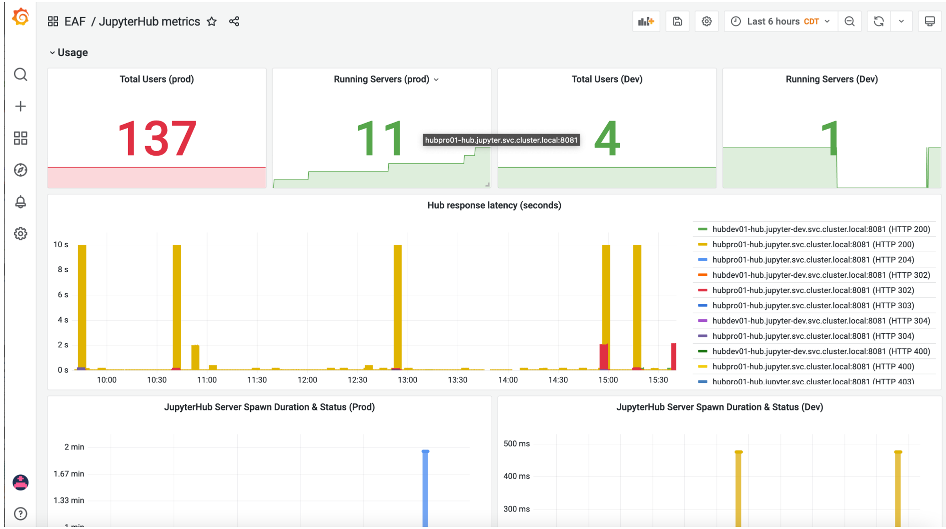
	Mean	Last *	Max
GPU10-3 [10.101.4.18]	32 °C	32 °C	32 °C
GPU11-0 [10.101.4.18]	35 °C	35 °C	35 °C
GPU11-1 [10.101.4.18]	35 °C	35 °C	35 °C
GPU11-2 [10.101.4.18]	33 °C	33 °C	33 °C
GPU12-0 [10.101.4.18]	35 °C	35 °C	35 °C
GPU12-1 [10.101.4.18]	35 °C	35 °C	35 °C
GPU12-2 [10.101.4.18]	33 °C	33 °C	33 °C
GPU13-0 [10.101.4.18]	35 °C	35 °C	35 °C
GPU13-1 [10.101.4.18]	35 °C	35 °C	35 °C
GPU13-2 [10.101.4.18]	33 °C	33 °C	33 °C

GPU Power Usage

	max	avg	current
GPU10-3 [10.101.4.18]	72.3 W	72.2 W	72.2 W
GPU11-0 [10.101.4.18]	72.0 W	71.9 W	71.9 W
GPU11-1 [10.101.4.18]	78.5 W	78.3 W	78.4 W
GPU11-2 [10.101.4.18]	75.0 W	74.9 W	74.9 W
GPU12-0 [10.101.4.18]	72.0 W	71.9 W	72.0 W
GPU12-1 [10.101.4.18]	78.4 W	78.4 W	78.4 W
GPU12-2 [10.101.4.18]	75.0 W	74.9 W	74.8 W
GPU13-0 [10.101.4.18]	72.0 W	71.9 W	72.0 W
GPU13-1 [10.101.4.18]	78.5 W	78.4 W	78.4 W

Panel Title

instance ↑	GPU	MiG ID ↑	MiG fl

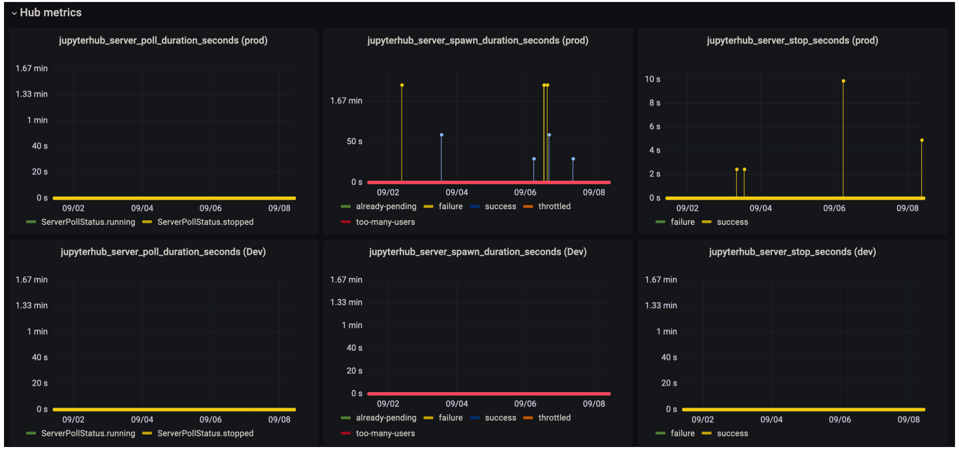


Hub response latency (seconds)

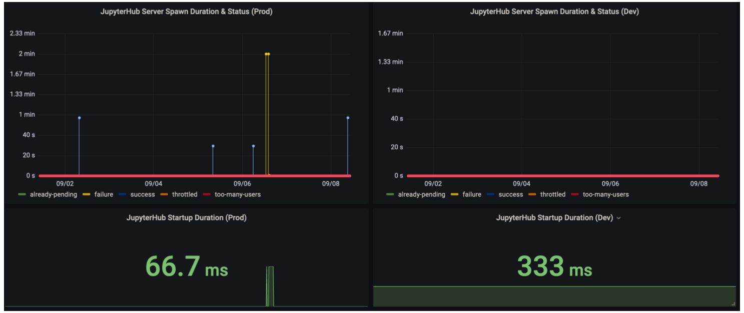
- hubdev01-hub.jupyter-dev.svc.cluster.local:8081 (HTTP 200)
- hubpro01-hub.jupyter.svc.cluster.local:8081 (HTTP 200)
- hubpro01-hub.jupyter.svc.cluster.local:8081 (HTTP 204)
- hubdev01-hub.jupyter-dev.svc.cluster.local:8081 (HTTP 302)
- hubpro01-hub.jupyter.svc.cluster.local:8081 (HTTP 302)
- hubpro01-hub.jupyter.svc.cluster.local:8081 (HTTP 303)
- hubdev01-hub.jupyter-dev.svc.cluster.local:8081 (HTTP 304)
- hubpro01-hub.jupyter.svc.cluster.local:8081 (HTTP 304)
- hubdev01-hub.jupyter-dev.svc.cluster.local:8081 (HTTP 400)
- hubpro01-hub.jupyter.svc.cluster.local:8081 (HTTP 400)
- hubpro01-hub.jupyter.svc.cluster.local:8081 (HTTP 403)

Monitoring and metrics

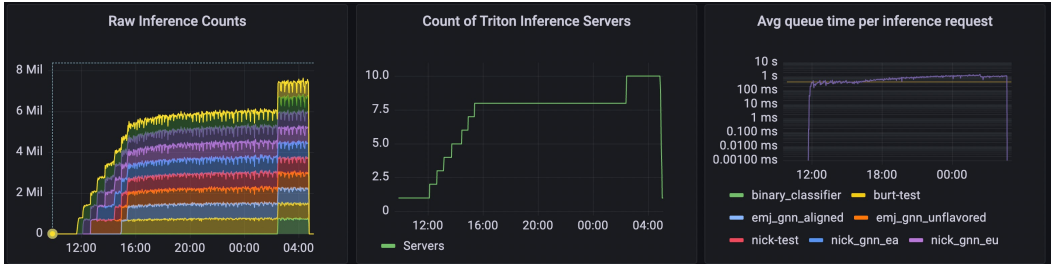
Insight on spawning process duration and outcomes for each step: poll, spawn, stop:



Spawning process duration and Hub (application) startup time:

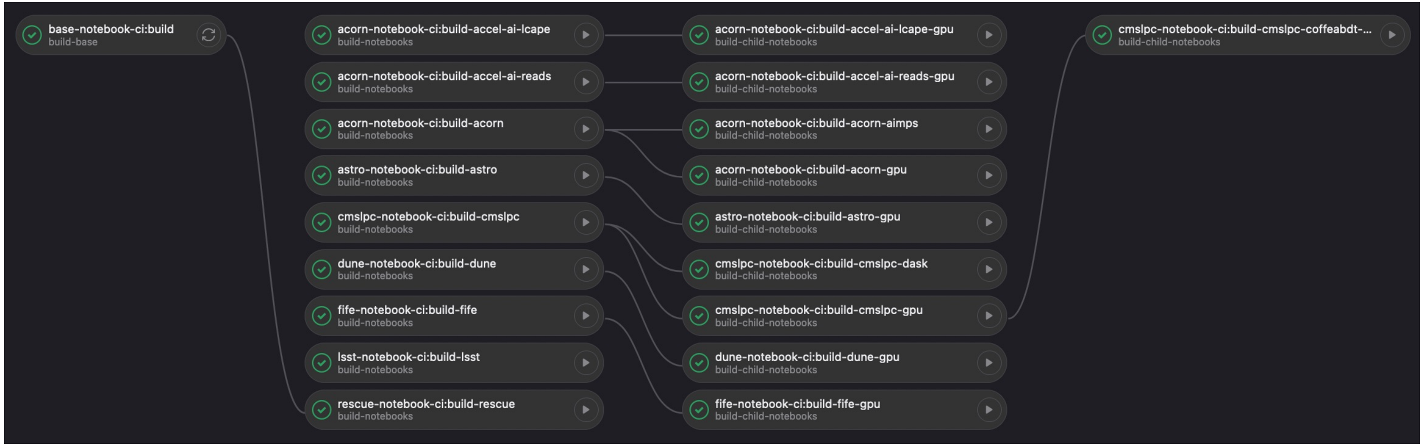


Triton autoscaling:



CI/CD for image builds via GitLab

Simultaneous builds with dependency relationships:

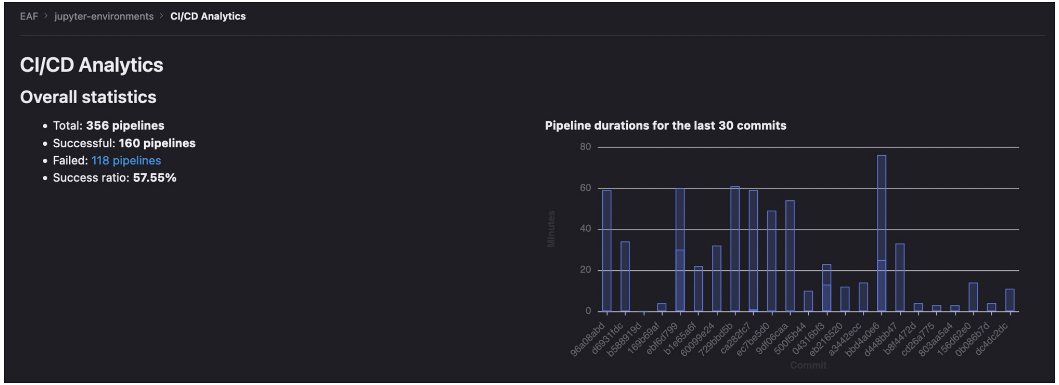


- base-notebook-ci:build
  build-base

- acorn-notebook-ci:build-accel-ai-lcape
  build-notebooks
- acorn-notebook-ci:build-accel-ai-reads
  build-notebooks
- acorn-notebook-ci:build-acorn
  build-notebooks
- astro-notebook-ci:build-astro
  build-notebooks
- cmslpc-notebook-ci:build-cmslpc
  build-notebooks
- dune-notebook-ci:build-dune
  build-notebooks
- fife-notebook-ci:build-fife
  build-notebooks
- lsst-notebook-ci:build-lsst
  build-notebooks
- rescue-notebook-ci:build-rescue
  build-notebooks

- acorn-notebook-ci:build-accel-ai-lcape-gpu
  build-child-notebooks
- acorn-notebook-ci:build-accel-ai-reads-gpu
  build-child-notebooks
- acorn-notebook-ci:build-acorn-aimps
  build-child-notebooks
- acorn-notebook-ci:build-acorn-gpu
  build-child-notebooks
- astro-notebook-ci:build-astro-gpu
  build-child-notebooks
- cmslpc-notebook-ci:build-cmslpc-dask
  build-child-notebooks
- cmslpc-notebook-ci:build-cmslpc-gpu
  build-child-notebooks
- dune-notebook-ci:build-dune-gpu
  build-child-notebooks
- fife-notebook-ci:build-fife-gpu
  build-child-notebooks

- cmslpc-notebook-ci:build-cmslpc-coffeabdt-...
  build-child-notebooks



EAF › jupyter-environments › **CI/CD Analytics**

CI/CD Analytics

Overall statistics

- Total: **356 pipelines**
- Successful: **160 pipelines**
- Failed: 118 pipelines
- Success ratio: **57.55%**

Pipeline durations for the last 30 commits

Pipeline Analytics

Active collaboration (and communication)

Secure | Integrated & functional | Multi-VO | DevOps (operational sustainability) | Active collaboration

- Currently outlining plans for formal communication and support channels as well as feedback spaces.
- ServiceNow service offering for user issues and requests
- Public-facing documentation site for users: https://eafjupyter.readthedocs.io/en/latest/
- Feedback, issues and user/VO requests are always welcome, we encourage SNOW for all formal requests.
- Other support channels include
  - Slack (@macosta, @burt, @elisec and the #eaf-users channel)
  - Mattermost (@macostaf @holzman)
  - EAF mailing lists: eaf-users@fnal.gov, eaf-admins@fnal.gov

- Active Participation on IRIS-HEP AGCs and Fermilab Users Meetings. Check out our Demo and presentation material from September on indico!

Summary and questions:

- GPUs are in demand. Access to specialized hardware is one of the key aspects of EAF. - How to partition resources in a fair way?
- Experiments and users concerned with data access for analysis - How can we effectively bring data in and out of analysis facilities?
- The facility is gaining traction and interest from multiple groups not traditionally considered 'experiments' – How can we properly onboard users and groups?
- Current focus on documentation, user channels, feedback and stronger authorization models as well as Dask Gateway and BinderHub.
- Inter-facility collaboration and communication is key – How to avoid duplicate work? How to benchmark AFs? What are our channels and spaces to talk to each other?

Thanks ☺ Questions?

Maria Acosta – EAF, ACORN
macosta@fnal.gov
@macosta on Slack