



université
PARIS-SACLAY



Fink on Kubernetes

Efficient management of massive alert streams for astronomical objects identification



Fabrice Jammes

Research engineer

Laboratoire de Physique de Clermont

Julien Peloton

Research engineer

IJCLab, Orsay

Etienne Fayen

Research engineer

Université Paris Saclay



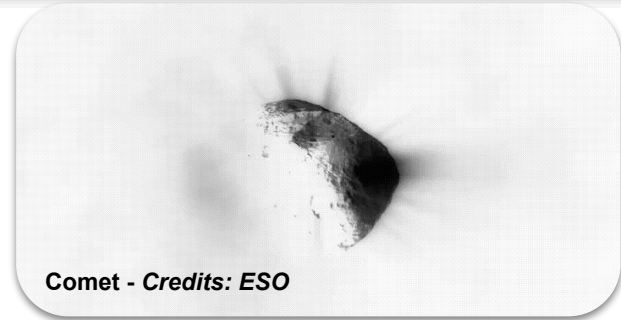
Fink: astronomy & computing

What are we doing?

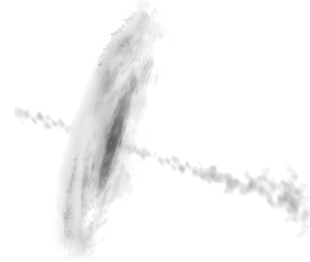
Tracking **changes** in the sky

- Solar system, galactic and extragalactic science

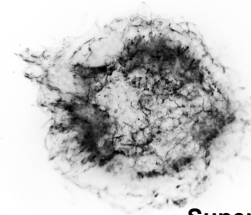
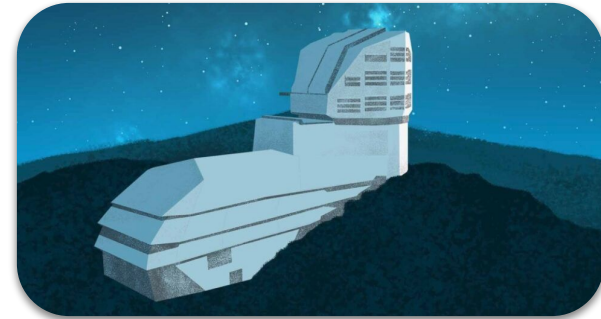
Fink is a **broker**, serving the scientific community by ingesting, classifying, filtering, and redistributing **alerts** from telescopes and surveys.



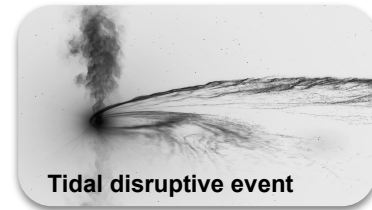
Comet - Credits: ESO



Active Galactic nucleus



Supernova



Tidal disruptive event



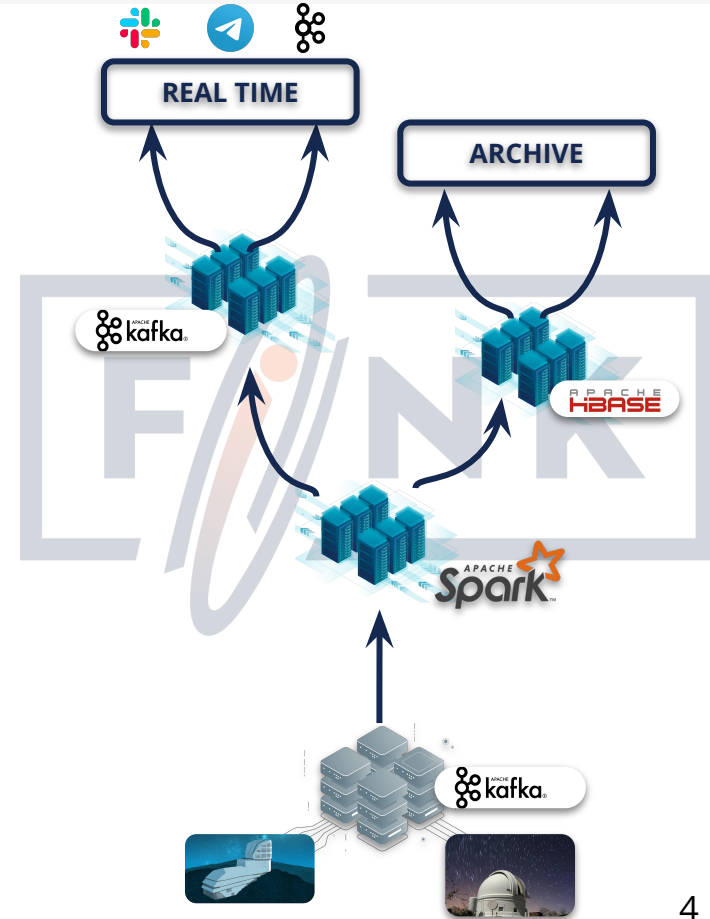
Current status

Services deployed on large academic
OpenStack clouds

Operating **24/7**, serving 100+ unique users
per day: observatories, scientists &
amateurs

- Real-time components (million+
event/night)
- Event database (~1B entries)

Community-driven: scientists bring
building bricks to the project



Why using kubernetes?



The long run is difficult


- Number of maintainers is low → **deployment and operations should be made easy**
- From 0 to million of events per night → **auto-scaling**
- Domain experts are the crucial agents for scientific discoveries, but low computer skills → **high-level abstractions**
- Large ecosystem of dependencies, often incompatible → **microservices**
- Interplay between the computing model & user software → **Infrastructure should adapt to specific user needs**
- From development to production → **end-to-end testing is necessary: code & infrastructure**



Harnessing the power of Kubernetes

Empowering academic cloud

Beyond improving Fink capabilities, the **objective is to develop skills to operate large Academic On-Premise clouds** (e.g. Paris Saclay University cloud)

The experience gained with Fink benefits other communities in return, e.g. available with Kubernetes 

- On-demand Kubernetes / Apache Spark clusters
- On-demand self-hosted GitHub Action runners
- Training on cloud technologies for students & staff

A scalable self-hosted CI

“Let’s turn those big static CI/CD pipelines into lightweight, easy to create, easy to run pipelines”

Validation the full production platform

The screenshot shows a GitHub Actions workflow run for the repository 'astrolabsoftware / fink-broker'. The workflow is named 'run science on branch #167' and is currently in an 'In progress' state. The workflow file is 'e2e-science.yml' with a trigger of 'on: push'. The workflow consists of three jobs: 'call-workflow-passing-data', 'e2e-science.yml', and 'call-workflow-passing-data'. The 'e2e-science.yml' job is currently running and contains three steps: 'Build image' (39s), 'Run integration tests' (25m 36s), and 'Analyze image' (5m 14s). The 'Build image' and 'Analyze image' steps are completed, while the 'Run integration tests' step is in progress. The 'call-workflow-passing-data' job is not yet started.

← e2e: science, self-hosted

run science on branch #167 Cancel workflow Latest #5 ⋮

Summary

Jobs

- call-workflow-passing-data
- Build image
- Run integration tests
- Analyze image

Run details

- Usage
- Workflow file

Re-run triggered 26 minutes ago	Status	Total duration	Artifacts
fjammes → 2d01be4 816-deploy-fink-broker-on...	In progress	-	-

e2e-science.yml
on: push

```
graph LR; A[call-workflow... / Build image 39s] --> B[c... / Run integration t... 25m 36s]; A --> C[call-wo... / Analyze image 5m 14s]; B --> D[c... / Push fink-broker image t...];
```

9

CI replicates production deployment and execution

The screenshot shows a GitHub Actions workflow run for the repository `astrolabssoftware/fink-broker`. The workflow is titled `call-workflow-passing-data / Run integration tests` and is in a `Beta` state. It started 26m 28s ago. The workflow consists of several jobs, with the current job being `Run integration tests`.

The `Run integration tests` job is highlighted with an orange rounded rectangle and contains the following steps:

- Install ciux
- Ciux project ignition
- Create k8s (kind) cluster
- Install olm and argocd operators
- Install argo-workflows (fink-alert-simulator pre-requis...
- Run argoCD
- Download image
- Load container image inside kind
- Run fink-alert-simulator
- Install fink-broker pre-requisites (JDK, Spark)
- Run fink-broker
- Check results
- Promote fink-broker image
- Post Run actions/setup-go@v4

The right side of the screenshot shows a visual representation of the workflow steps. A large rounded rectangle highlights the steps from `Install ciux` to `Run argoCD`. Inside this rectangle, there is a logo for `k8s-toolbox` and a logo for `OPERATOR LIFECYCLE MANAGER`. Below the highlighted steps, there are logos for `kafka`, `MINIO`, and `FINK`.

The bottom right corner of the screenshot shows the `FINK` logo.

Setup the Fink pre-requisites in CI with Gitops



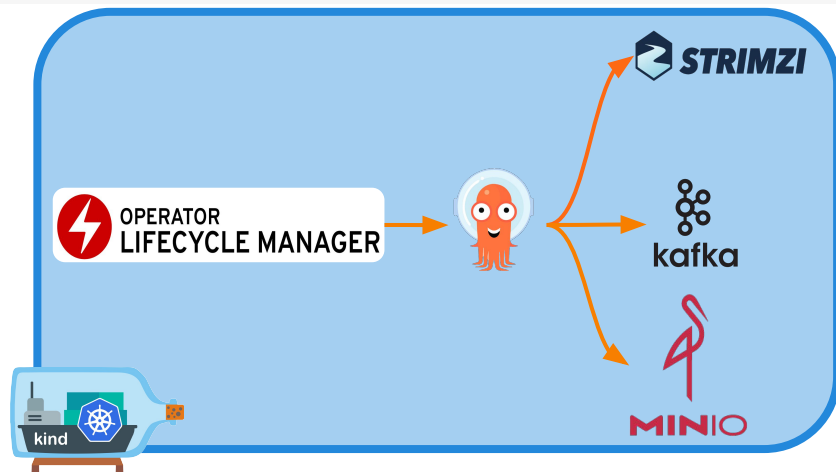
k8s-toolbox



<https://github.com/k8s-school/ktbx>

- Easy Kubernetes Installation on Linux
- One-line installations for OLM, ArgoCD, Argo Workflow
- Instant access to k8s sysadmin tools

→ Simulate Kafka servers and S3 storage for telescopes in seconds



```
# 3 nodes k8s cluster with Calico CNI
ktbx create -c

# Install components
ktbx install kubect1
ktbx install olm
ktbx install argocd
ktbx install argowf
```

Extensive use of operators



- Apache Spark Operator
- Kafka Operator (Strimzi)
- HDFS Operator
- Minio Operator



Ciux : automated build/e2e tests pre-requisites checks

<https://github.com/k8s-school/ciux>

- avoids rebuilding fink image in CI if no code changed
- deploys all microservices, including those under development, necessary for the fink stack e2e testing
- Check if images are available before running e2e tests

```
Project fink-broker
/home/runner/work/fink-broker/fink-broker v3.1.2-rc1-40-g75a25ee
Dependencies:
/home/runner/work/fink-broker/fink-alert-simulator v3.1.2-rc1-2-g244067f in-place=false pull=true
/home/runner/work/fink-broker/finkctl v3.1.1-rc2-16-g86cc680 in-place=false
https://github.com/astrolabsoftware/fink-cd remote-only=true
Package: github.com/k8s-school/ktbx@v1.1.1-rc17
Go modules installed:
https://github.com/astrolabsoftware/finkctl from-src=true
github.com/k8s-school/ktbx@v1.1.1-rc17
Available Images:
gitlab-registry.in2p3.fr/astrolabsoftware/fink/fink-alert-simulator:v3.1.2-rc1-2-g244067f
```

Automated
management of
versions numbers

Insights on self-hosted runner for GHA

Essential for testing Fink with its production environment

- Runs on a k8s cluster, each CI job runs in a pod
- Based on ARC which is stable in recent versions
- Considering VCluster as an alternative to Kind

While powerful, CI still requires **significant** maintenance ;-(
but it saves a lot of time when deploying production!

K8s-based Self-Hosted Runner Advantages

- **Scalability:** Easily scale resources to match workload demands
- **Interactive Debugging:** access from workstation in 2 lines:

```
user@workstation:~$ export KUBECONFIG=$HOME/.kube/self-hosted-ci kubeconfig
user@workstation:~$ kubectl exec -it -n arc-runners arc-runners-cllkn-runner-jzgr9 -- bash
Defaulted container "runner" out of: runner, dind, init-dind-externals (init)
runner@arc-runners-cllkn-runner-jzgr9:~$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
distribute-py-10fc7d8e41697017-driver	1/1	Running	0	21m
distribute-sims-20200101-7766b68e416b3f78-exec-10	1/1	Running	0	18m
distribute-sims-20200101-7766b68e416b3f78-exec-9	1/1	Running	0	18m
raw2science-py-721c198e41696faa-driver	0/1	Error	0	21m
stream2raw-py-d247da8e41697082-driver	1/1	Running	0	21m
stream2raw-sims-ee7de18e416b4246-exec-10	1/1	Running	0	18m
stream2raw-sims-ee7de18e416b4246-exec-9	1/1	Running	0	18m

Easy interactive access to GHA self-hosted runner

Interactive troubleshooting:

```
runner@arc-runners-ctlkn-runner-jzgr9:~$ kubectl logs raw2science-py-721c198e41696faa-driver
24/03/15 09:23:26 INFO main (raw2science.py line 88): Applying quality cuts
Traceback (most recent call last):
  File "/home/fink/fink-broker/bin/raw2science.py", line 147, in <module>
    main()
  File "/home/fink/fink-broker/bin/raw2science.py", line 93, in main
    df = apply_science_modules(df, args.noscience)
  File "/home/fink/fink-broker/fink_broker/science.py", line 225, in apply_science_modules
    df = xmatch_cds(df)
NameError: name 'xmatch_cds' is not defined
24/03/15 09:23:27 WARN ExecutorPodsWatchSnapshotSource: Kubernetes client has been closed.
```

Save time by avoiding the need to reinstall the entire `fink-broker` stack and up-to-date microservices.

Regular GitHub Actions (GHA) runners lack interactive access and scalability.

DIY self-hosted CI

Goal: eliminates the dependency on remote services like GitHub Actions, GitLab CI

Currently e2e tests are launched through a cronjob on a LPC server which push a notification to GHA

A project is under development to make it simpler and more powerful:

<https://github.com/k8s-school/home-ci>

DEMO

<https://github.com/k8s-school/demo-kafka>

<https://github.com/astroabsoftware/fink-broker>

- 1 Fink is set to run until 2035 (at least!)
- 2 Hundreds of users watching the sky
- 3 Full deployment on k8s ongoing, fine-tuning in progress
- 4 Opportunity to learn better how to operate academic on-premise clouds

Q&A

Fabrice JAMMES

*Laboratoire de Physique de
Clermont*

Julien Peloton

IJCLab, Orsay

Etienne Fayen

Université Paris Saclay



@FinkBroker

