

Exploring fink-broker on Kubernetes

Fabrice Jammes

Research engineer
Laboratoire de Physique de Clermont

Julien Peloton

Research engineer
IJCLab

Etienne Fayen

Research engineer
Université Paris Saclay

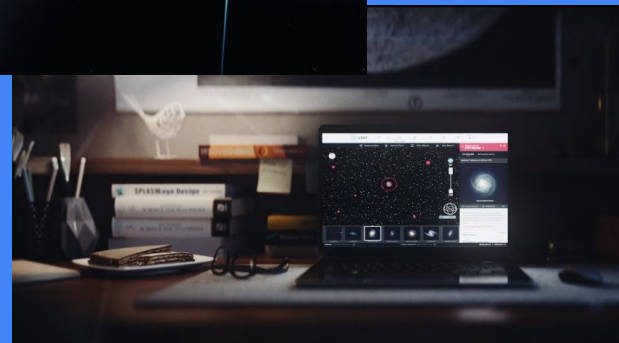
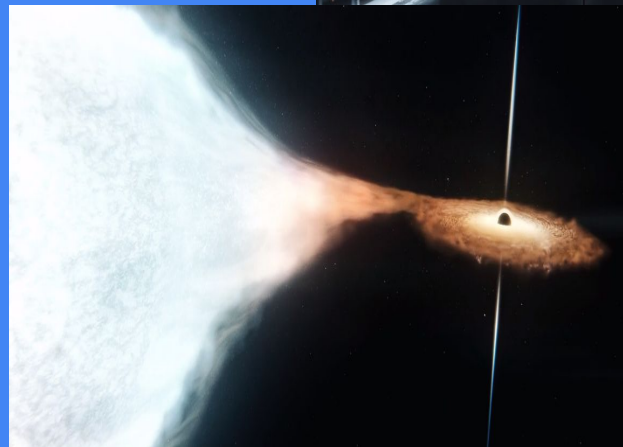
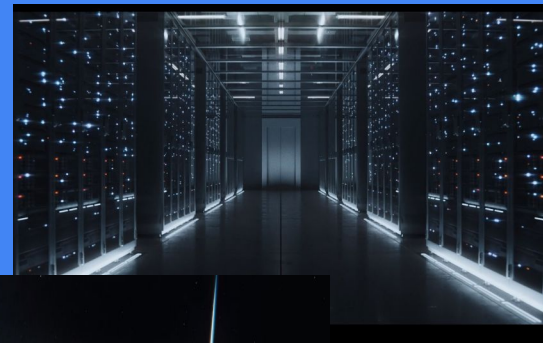
1 Containers

2 Kubernetes

3 Fink broker overview

4 Devops essentials and CI/CD

5 Toolbox: ktbx, finkctl, ciux

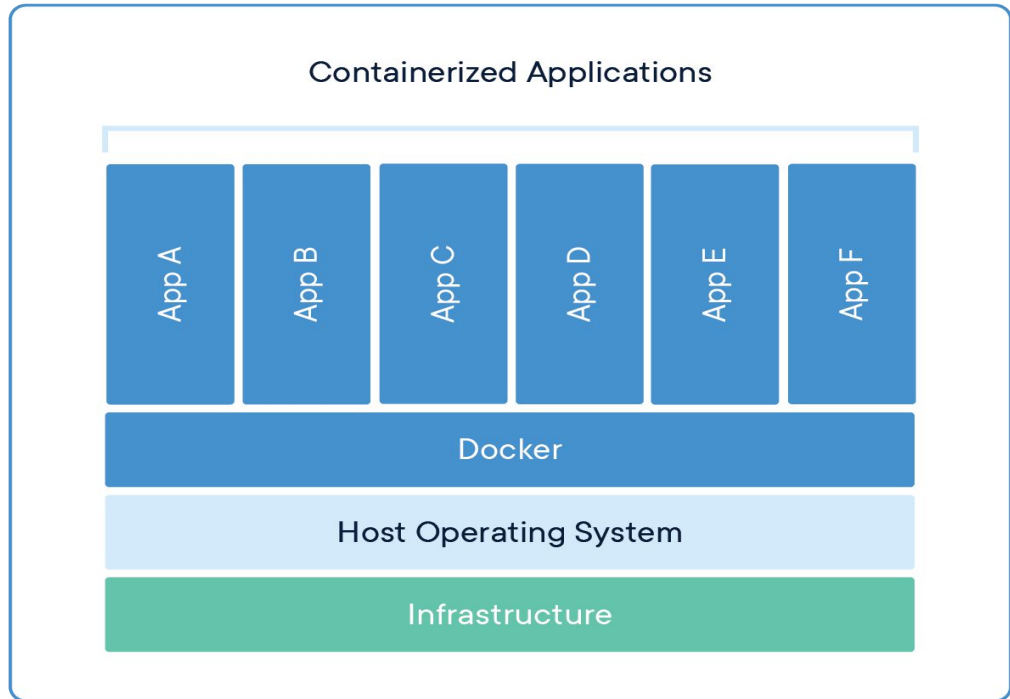


Linux Container: the Docker example

Definition

Docker is a tool that can package an application and its dependencies in an isolated container, which can be run on any server.

[Source: fr.wikipedia.org]



Deployment: target applications and infrastructures


Multiplicity of Stacks

 **Static website**
nginx 1.5 + modsecurity + openssl + bootstrap 2

 **User DB**
postgres + pgv8 + v8

 **Queue**
Redis + redis-sentinel

 **Analytics DB**
hadoop + hive + thrift + OpenDK

 **Background workers**
Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs

 **Web frontend**
Ruby + Rails + sass + Unicorn

 **API endpoint**
Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

Do services and apps interact appropriately?

Multiplicity of hardware environments

 **Development VM**

 **QA server**

Customer Data Center



Public Cloud

Disaster recovery

Production Servers



Production Cluster









Contributor's laptop



Can I migrate smoothly and quickly?



Matrix from hell

	Static website	?	?	?	?	?	?	?
	Web frontend	?	?	?	?	?	?	?
	Background workers	?	?	?	?	?	?	?
	User DB	?	?	?	?	?	?	?
	Analytics DB	?	?	?	?	?	?	?
	Queue	?	?	?	?	?	?	?
		Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers



Similarity with maritime transport

Multiplicity of Goods



Do I worry about how goods interact (e.g. coffee beans next to spices)

Multiplicity of methods for transporting/storing



Can I transport quickly and smoothly (e.g. from boat to train to truck)

Intermodal containers

Multiplicity of Goods



A standard container that is loaded with virtually any goods, and stays sealed until it reaches final delivery.

Do I worry about how goods interact (e.g. coffee beans next to spices)



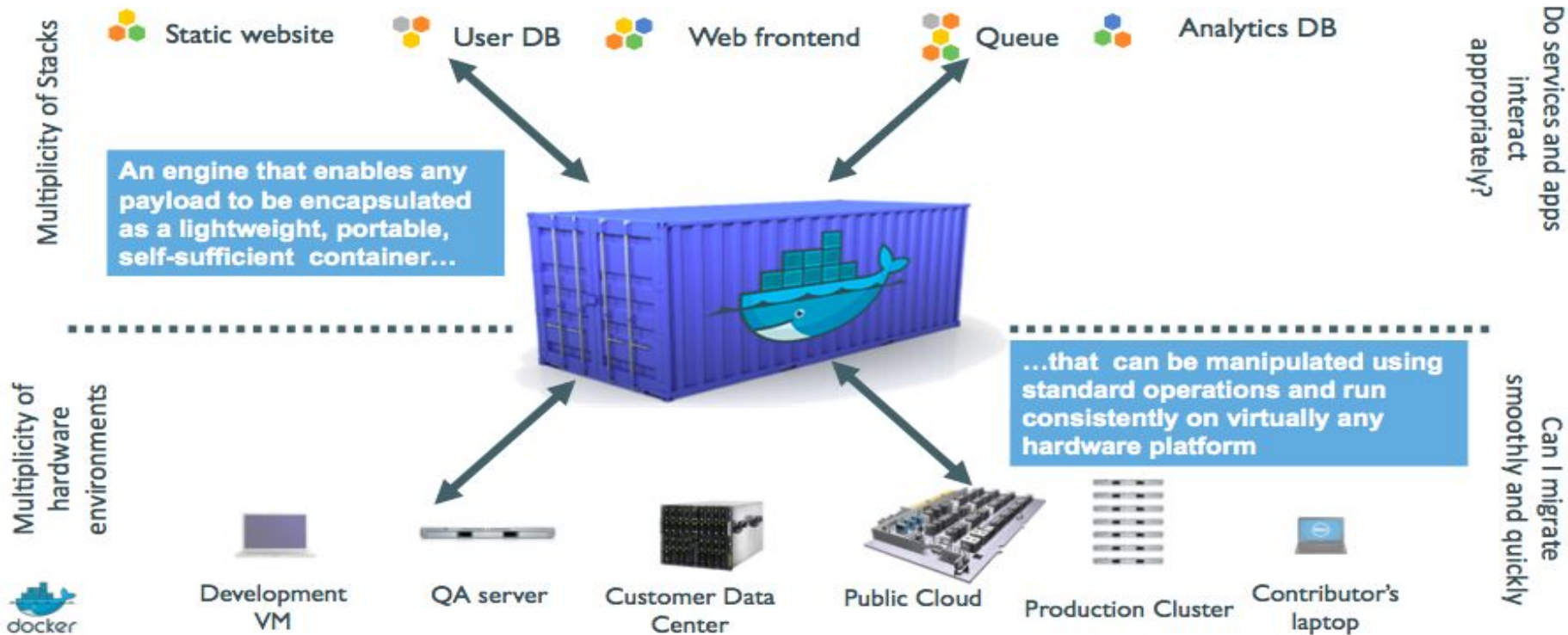
...in between, can be loaded and unloaded, stacked, transported efficiently over long distances, and transferred from one mode of transport to another

Can I transport quickly and smoothly (e.g. from boat to train to truck)
























































Multiplicity of methods for transporting/storing



A container system for applications



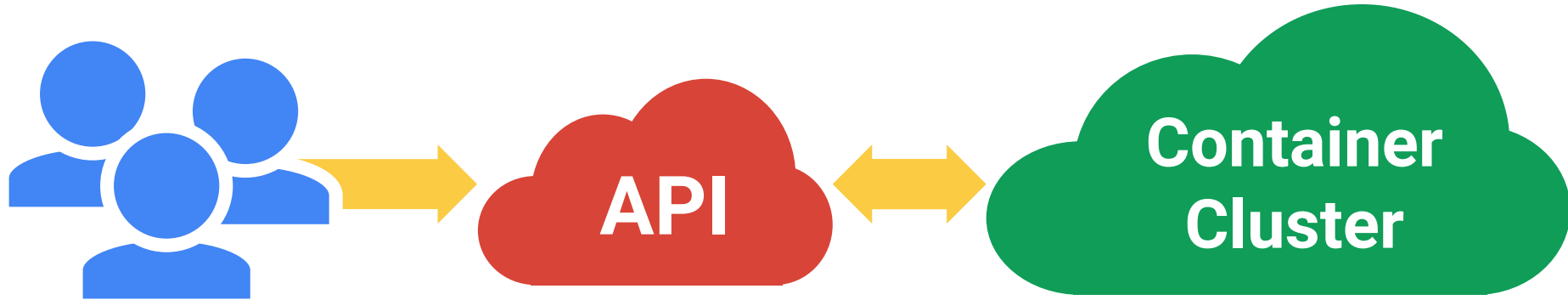
Exit matrix from hell!

	Static website							
	Web frontend							
	Background workers							
	User DB							
	Analytics DB							
	Queue							
		Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
								

Kubernetes (*shortname: k8s*)

Operating containerized applications at
scale

All you really care about



Many containers
running on many
servers

Workload Portability

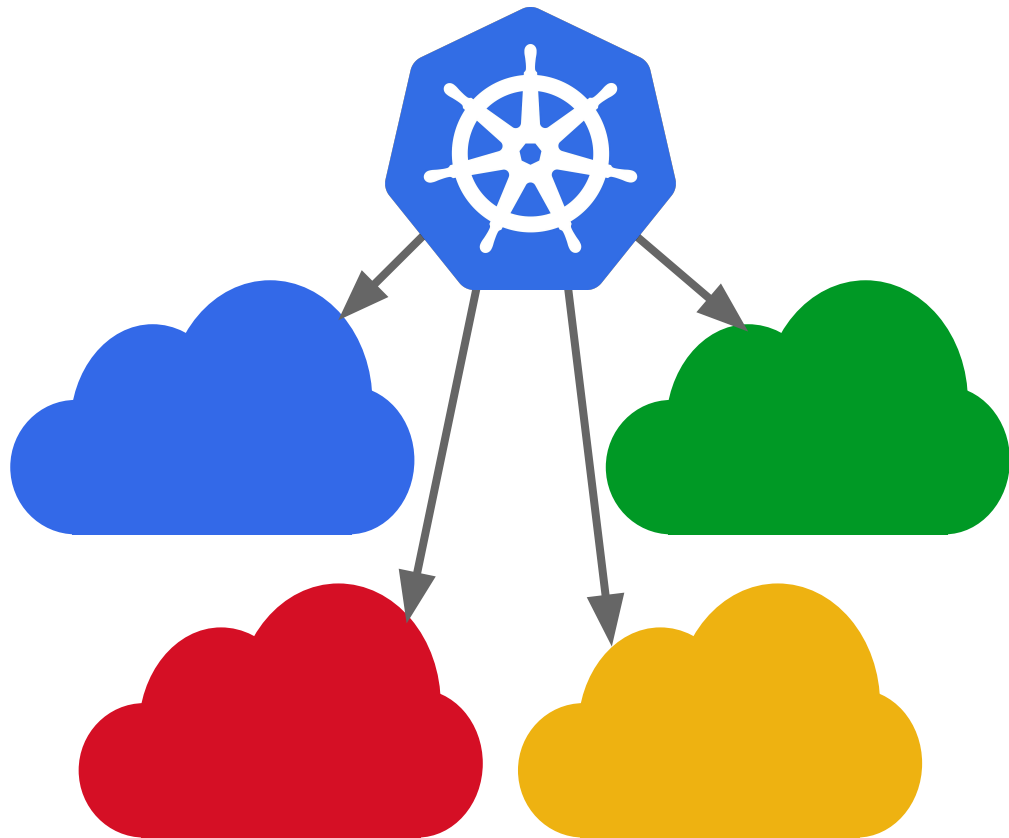
Workload portability

Goal: Avoid vendor lock-in

Runs in many environments, including
“**the cloud**”, “**bare metal**”, and “**your laptop**”

The API and the implementation are
100% open

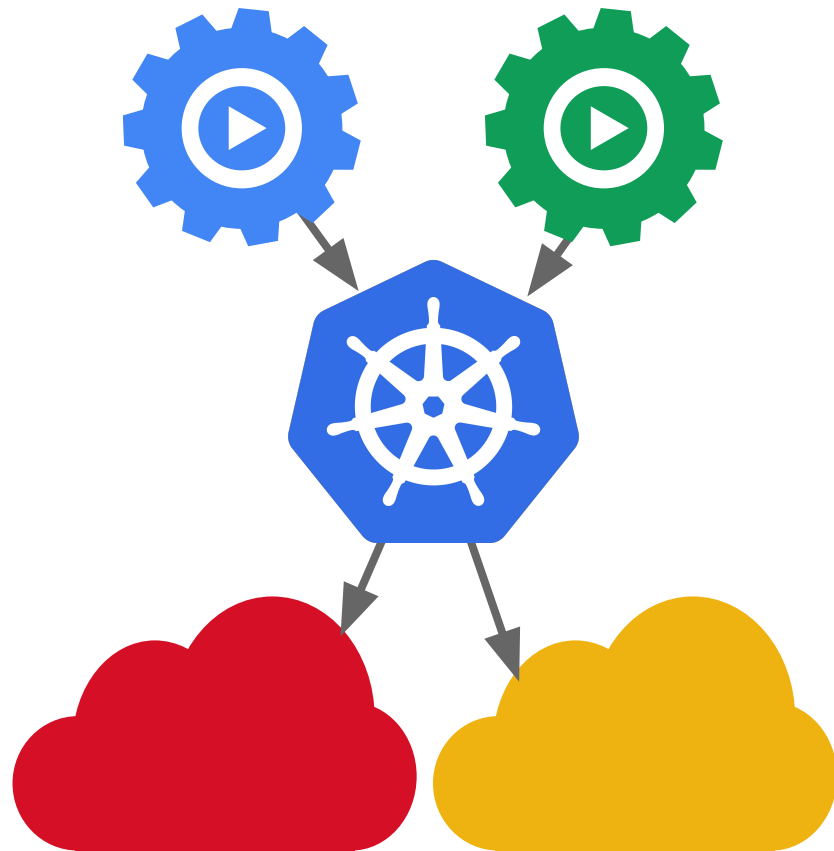
The whole system is modular and
replaceable



Workload portability

Goal: Write once, run anywhere*

Don't force apps to know about concepts that are cloud-provider/infrastructure-specific



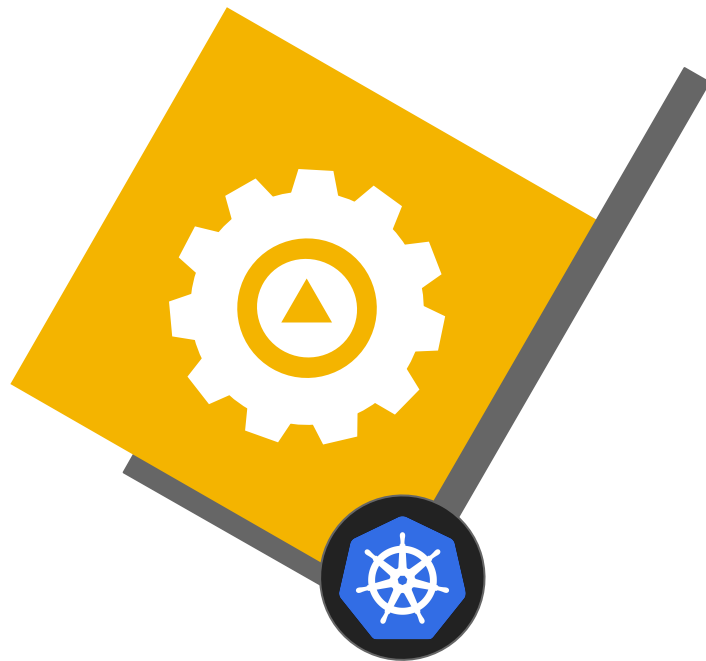
Workload portability

Result: Portability

Build your apps on-prem, lift-and-shift into cloud when you are ready

Don't get stuck with a platform that doesn't work for you

Put your app on wheels and move it whenever and wherever you need



Devops Essential

Combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity

CI / CD

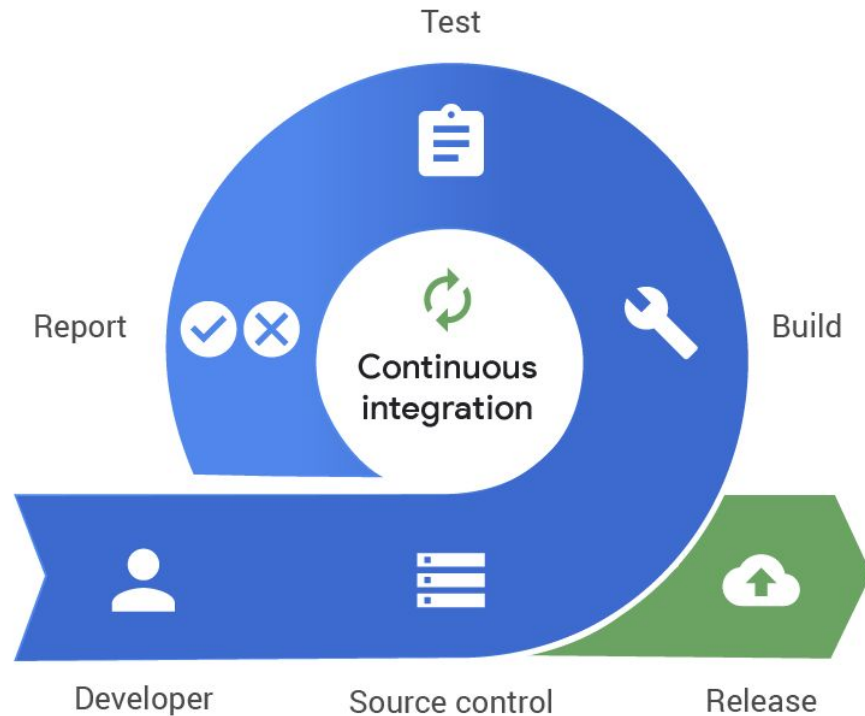
CI and CD stand for continuous integration and continuous delivery.

Continuous integration

Modern software development practice

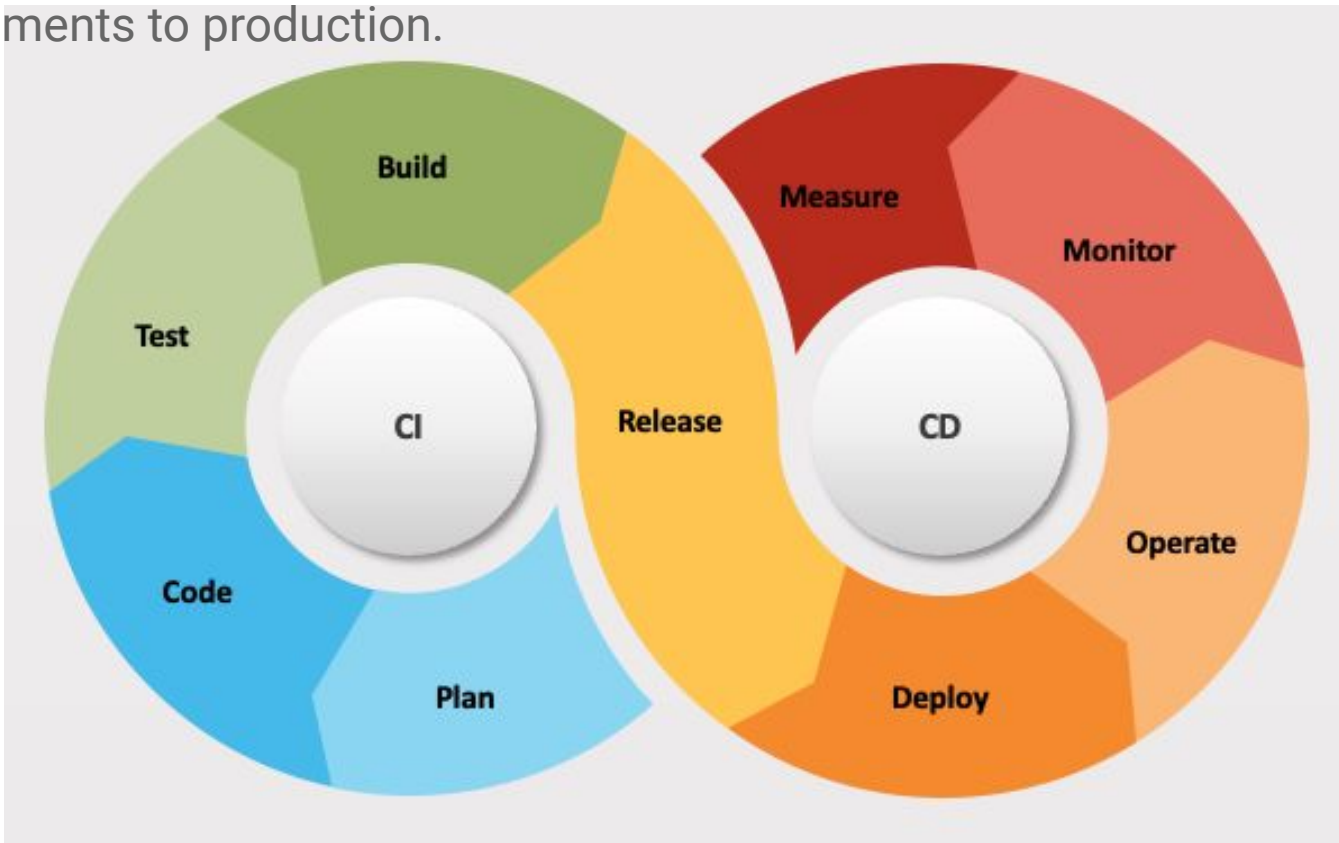
Incremental code changes are made frequently and reliably

Automated build-and-test steps ensure that code changes being merged into the repository are reliable



The CI/CD Pipeline

Automates the delivery of incremental code changes from developers' local environments to production.



Fink-broker on k8s

Fink: at scale, 100+ processes

Kafka

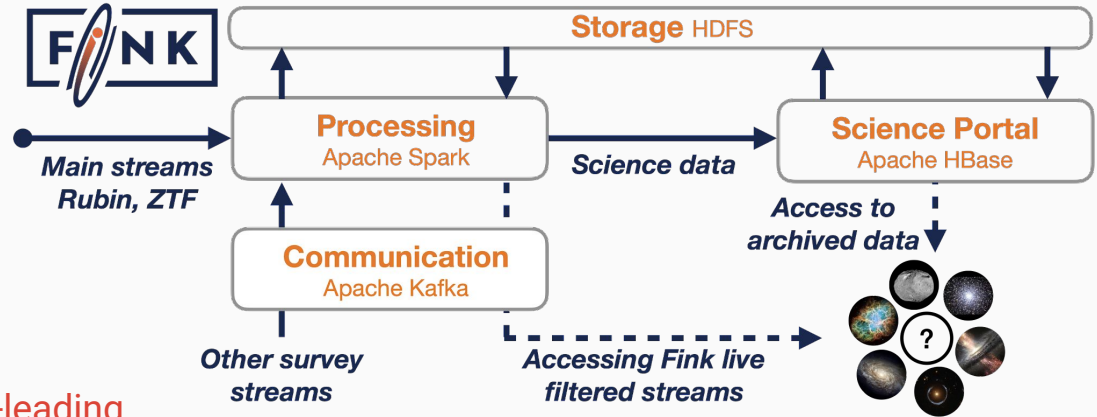
A data streaming technology and a complex distributed computing system.

Spark

Distributed processing system used for big data workloads

S3 Storage

Object storage service that offers industry-leading scalability, data availability, security, and performance



Objective:
Ease Fink-broker deployment and management at scale

Fully automated deployment

For each git commit, the fink stack is automatically deployed and tested on k8s

Jobs

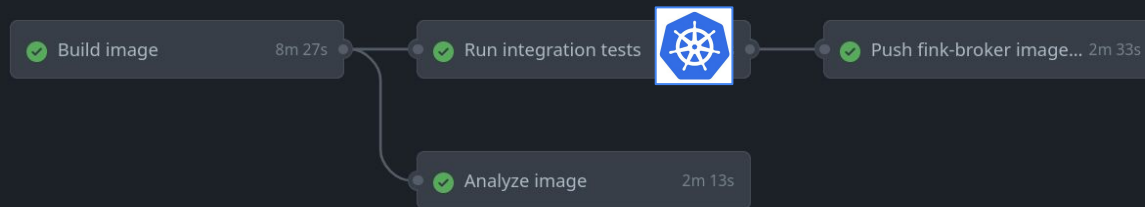
- ✓ Build image
- ✓ Run integration tests
- ✓ Analyze image
- ✓ Push fink-broker im...

Run details

- 🕒 Usage
- 📄 Workflow file

itest-gha.yml

on: push



A generic, push-button, install procedure

The very same procedure to install fink on:

- production/integration/qualification platforms
- developer workstations
- In the cloud
- On premise/Bare-metal

Run integration tests
succeeded yesterday in 9m 26s

Search logs

- ✓ Create k8s (kind) cluster
- ✓ Install olm and argocd operators
- ✓ Run argoCD
- ✓ Download image
- ✓ Load container image inside kind
- ✓ Install fink-alert-simulator pre-requisites (argo-workflows)
- ✓ Run fink-alert-simulator
- ✓ Install fink-broker pre-requisites (JDK Spark)
- ✓ Run fink-broker
- > ✓ Check results

CI Setup with KIND



For each commit

- Build Fink image
- Start Kubernetes
- Start Fink stack
- Launch integration tests
- Push image to registry



kind-control-plane
(master)



Container Runtime

KIND

kind-worker



Fink-broker

Kafka

Container Runtime

kind-worker2



Minio (S3)

Fink-broker

Container Runtime



Docker runtime

Github Action VM

<https://kind.sigs.k8s.io/>

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

Toolbox, written in golang

ktbx

Kubernetes install and management

ciux

Software stack versioning, release management

finkctl

Fink command line interface

Objective:

Manage a k8s-based multi-project software stack



Ciux will be proposed to Qserv team when more mature

- 1 Fink is running in a Kubernetes POC
- 2 Production platform coming at CC-IN2P3
- 3 TODO Security, monitoring, automation++
- 4 TODO Scalability test for LSST data deluge?

Conclusion

Q&A

Fabrice JAMMES
Laboratoire de Physique de
Clermont

Julien Peloton
IJCLab

Etienne Fayen
Université Paris Saclay

