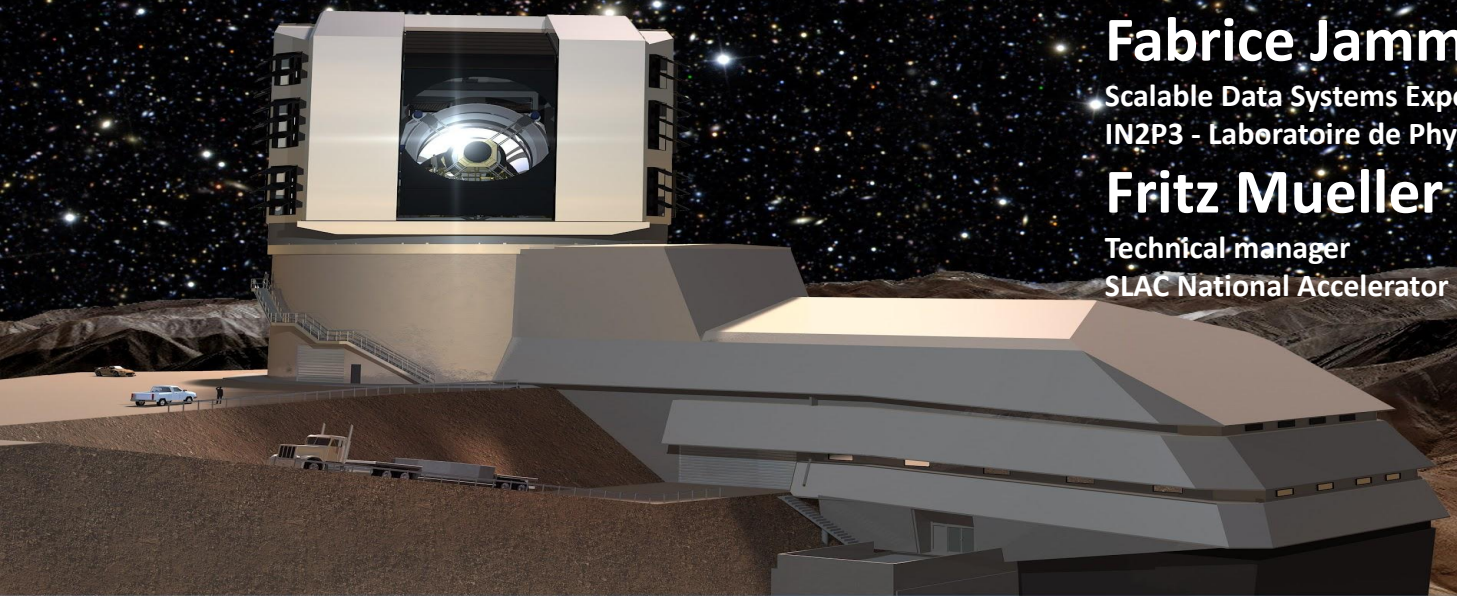
Système d'information dans le Cloud 16/11/2021

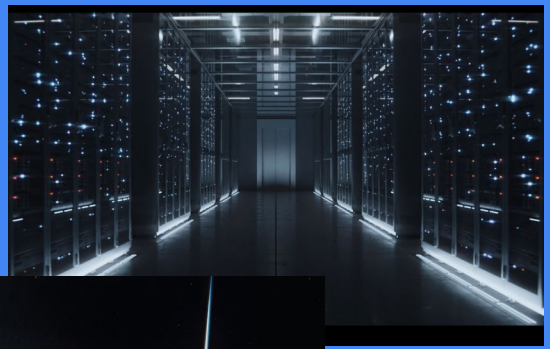
Large Synoptic Survey Telescope
From Cloud-Native to Dark Energy

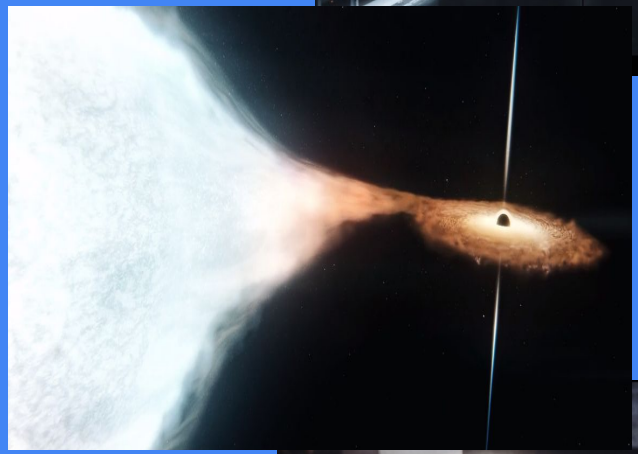


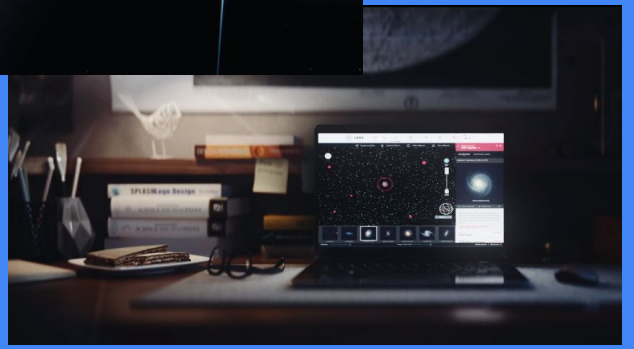
Fabrice Jammes
Scalable Data Systems Expert
IN2P3 - Laboratoire de Physique de Clermont

Fritz Mueller
Technical manager
SLAC National Accelerator Laboratory

1. Large Synoptic Survey Telescope
2. The largest astronomical catalog
3. Cloud-Native: Kubernetes
4. Cloud-Native: Gitops & CI
5. Cloud-Native: Kubernetes Operators
6. Cloud-Native: Storage management
7. Cloud-Native: Workflows







A project that makes you dream

A revolutionary telescope

The **largest digital camera** in the world

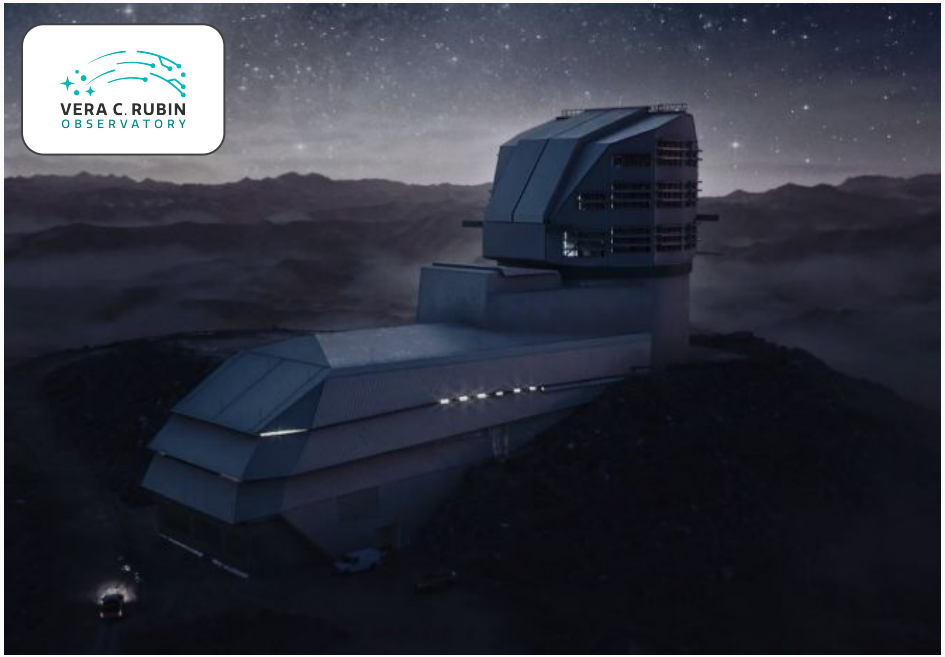
The **largest celestial catalogs** ever made

Funding

~$1 billion, 20% dedicated to data management

Key role of CNRS/IN2P3

Objective:
Define the nature of dark energy

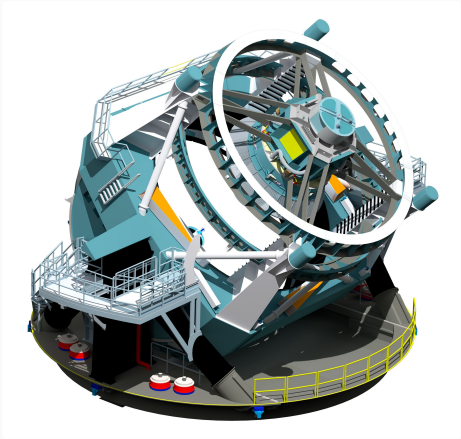


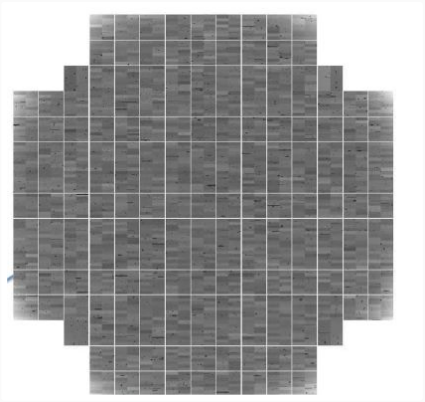
NSF

U.S. DEPARTMENT OF ENERGY

cnrs

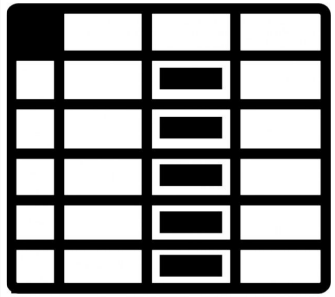
The largest astronomical catalog





Raw data

Processed image

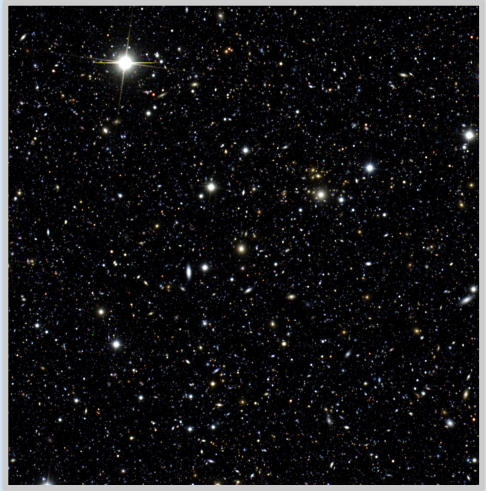


Catalog (stars, galaxies, objects, sources,transients, exposures, etc.)

LSST will produce a catalog of **40 billion galaxies and stars** and their associated physical properties, i.e. **500 PB** of data

Data

Images
Persisted: **~38 PB**
Temporary: **~½ EB**



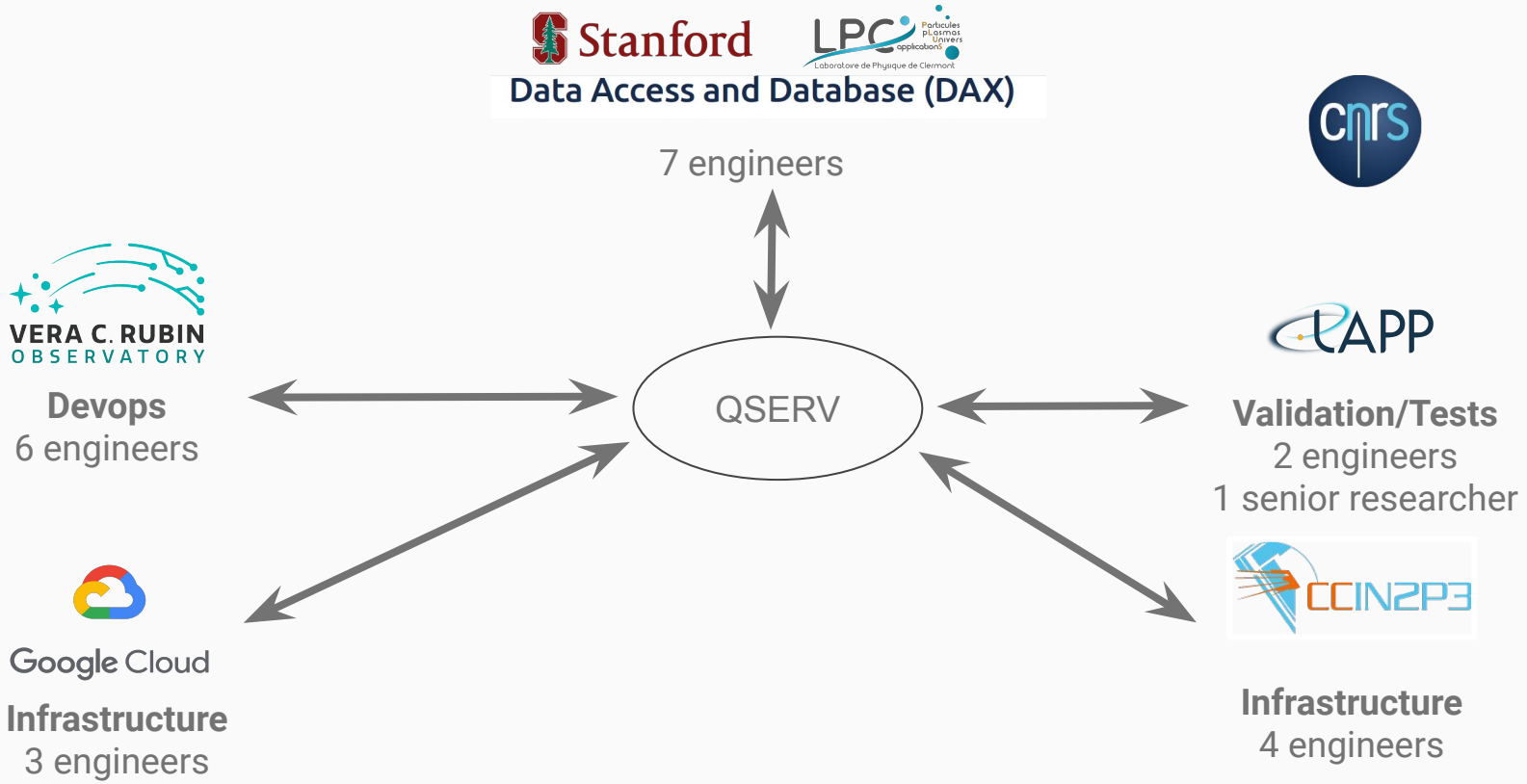
- ★ ~3 million “visits”
- ★ ~47 billion“objects”
- ★ ~9 trillion “detections”

- ★ Largest table: **~5 PB**
- ★ Tallest table: ~50 trillion rows
- ★ Total (all data releases, compressed): **~83 PB**

Ad-hoc user-generated data
Rich provenance

Qserv The Petascale database

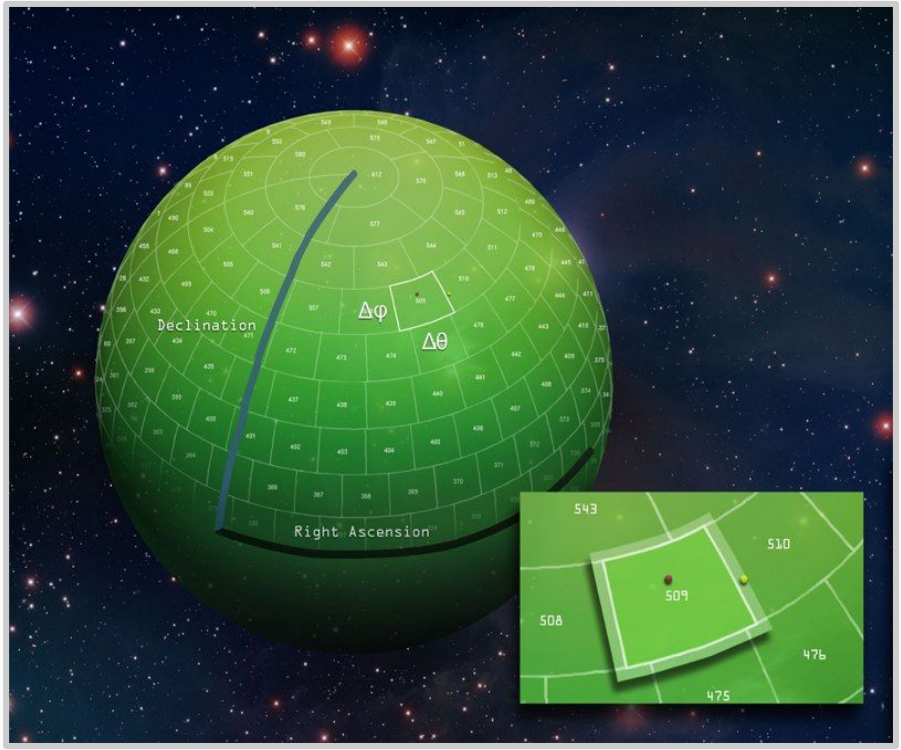
International context



The DAX team

Data Access and **Database**

- ★ Data and metadata
- ★ Images and databases
- ★ Persisting and querying
- ★ For pipelines and users
- ★ Real time Alert Production and annual Data Release Production
- ★ For Archive Center and all Data Access Centers
- ★ For USA, France and international partners
- ★ Persisted and virtual data
- ★ **Estimating, designing, prototyping, building, and productizing**

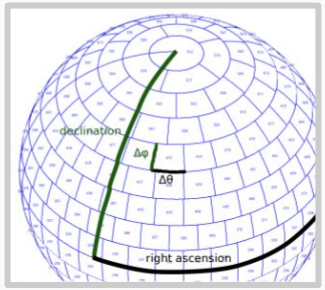


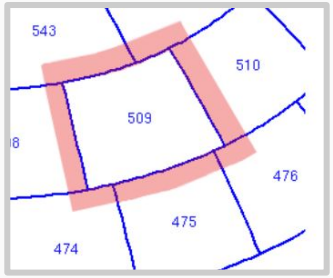
Qserv design

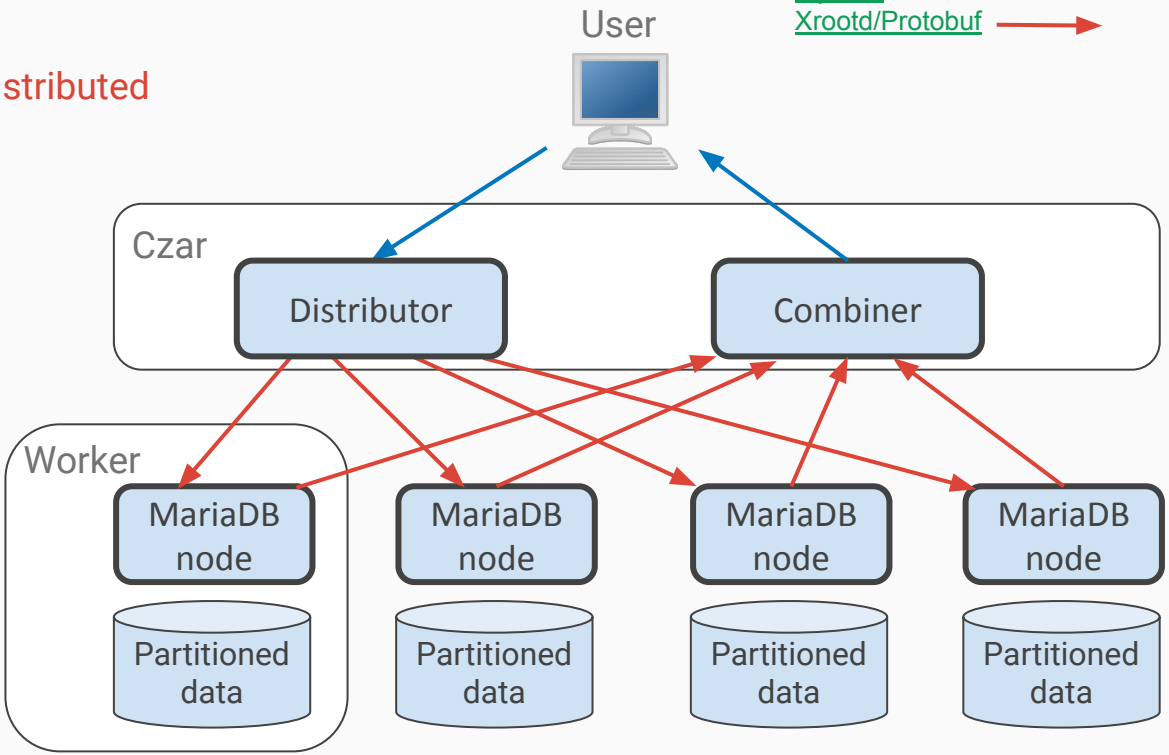
Relational database, 100% open source

Spatially-sharded with overlaps

Map/reduce-like processing, highly distributed







~1000 workers, 20 chunks/5TB per workers

Highly automated deployment

Goals

In France

CC-IN2P3 will analyze **50% of the data** stream and provide **access to the entire catalog**

In the US

Google hosts the **Interim Data Facility**

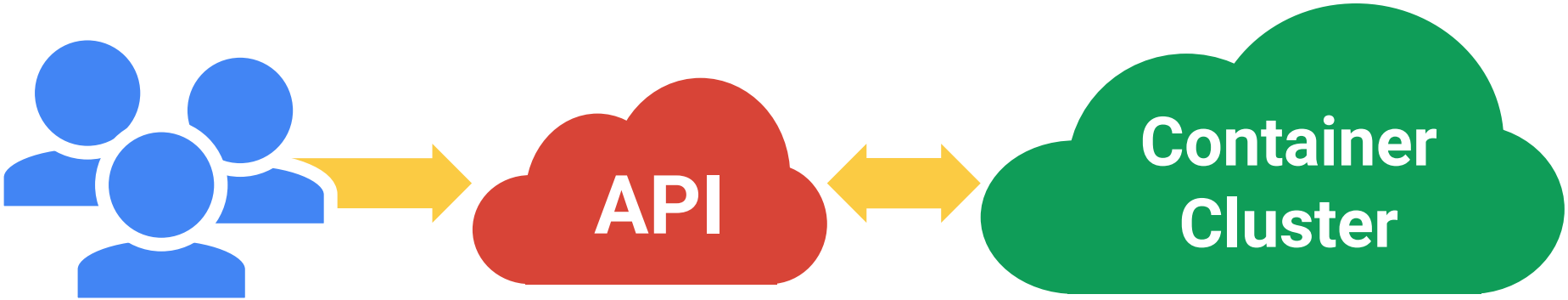
~1000 machines per database instance

Coordination of Rubin Observatory, IN2P3 and Google

Kubernetes accepted by the project and validated for 20% of the target

Cloud-Native
Kubernetes

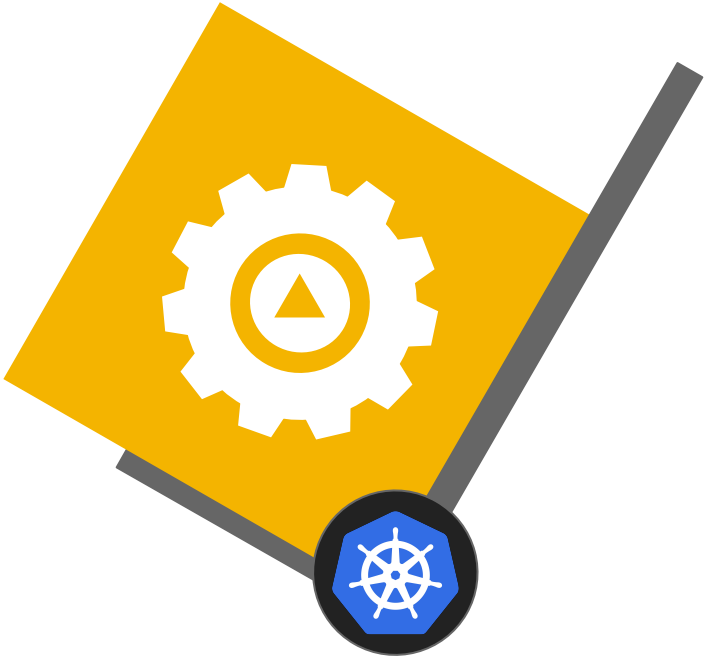
All you really care about



Workload portability

Portability

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

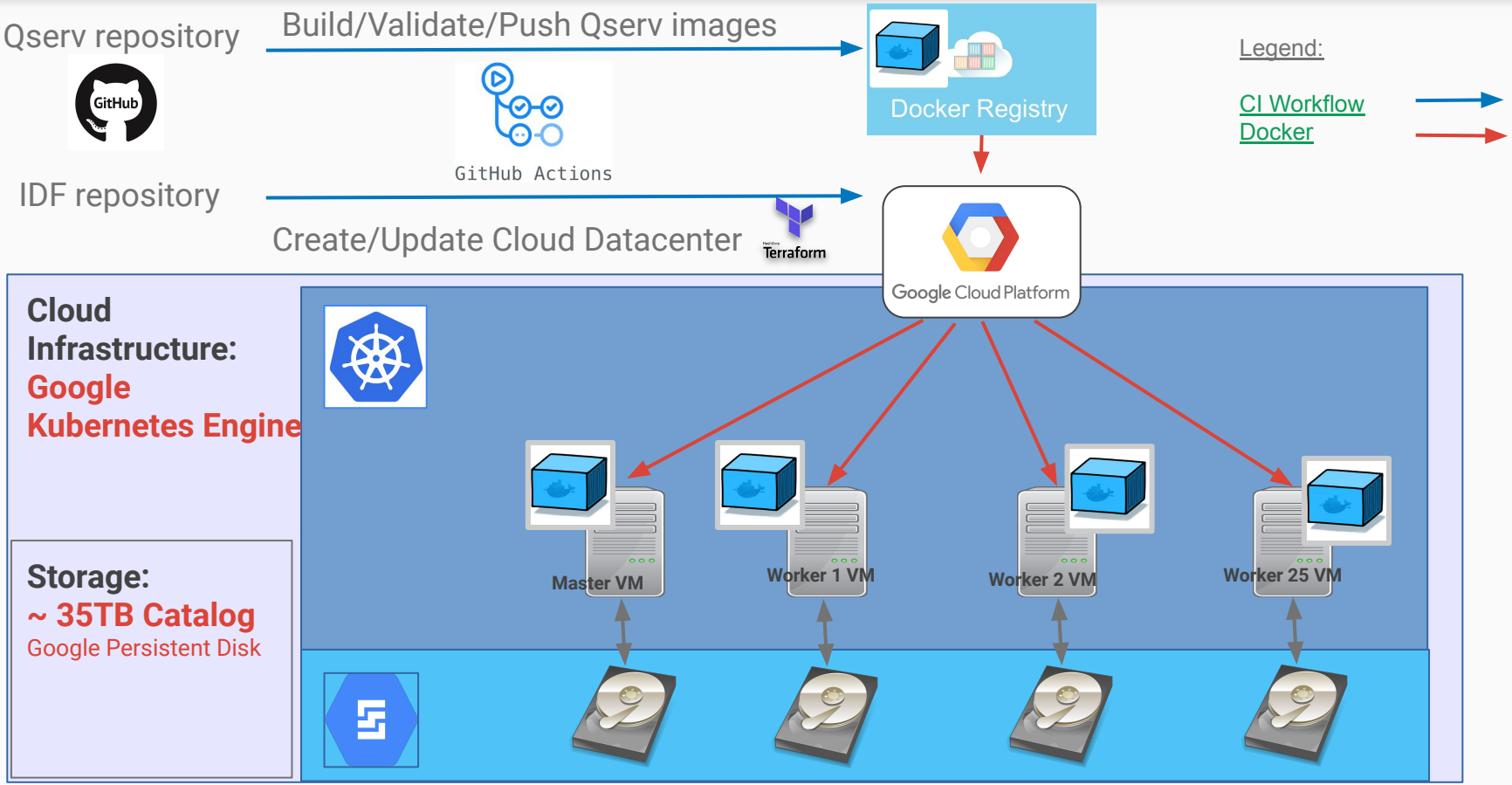


Before Kubernetes
~3 monthes to deploy Qserv inside a new cluster

With Kubernetes
5 minutes to 1 day

Cloud-Native
Gitops & CI

Automated deployment: Cloud Native



CI Setup with KIND

For each commit
- Build Qserv image
- Start kind
- Start Qserv
- Launch integration tests
- Push image to registry

KIND

kind-control-plane (master)

Container Runtime

kind-worker

Qserv Czar

Qserv Worker 1

Container Runtime

kind-worker2

Qserv Worker 2

Qserv Worker 2

Container Runtime

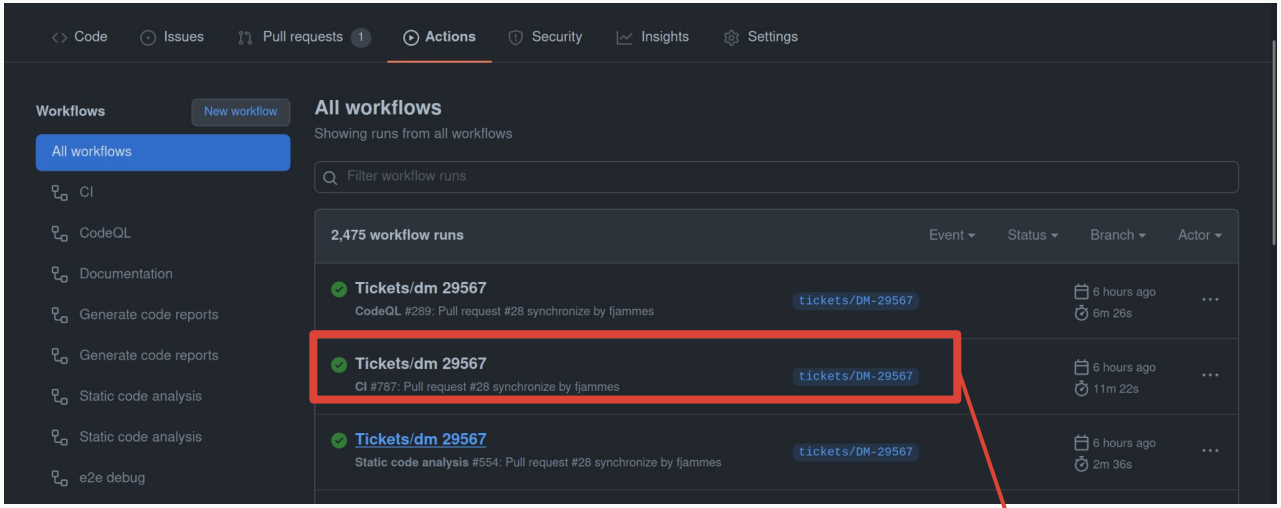
Docker runtime

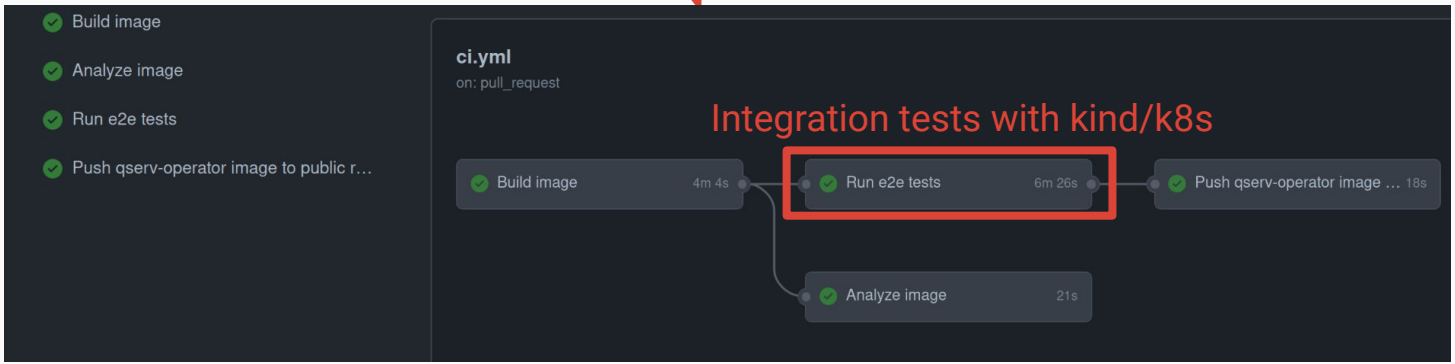
Github Action VM

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

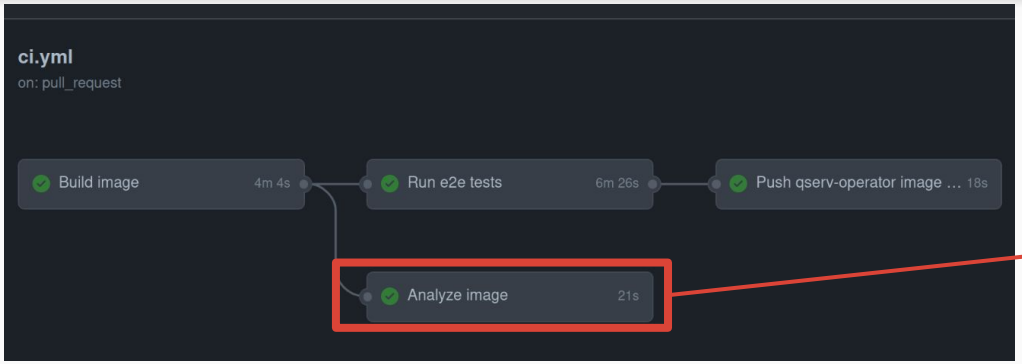
Wrapper for CI: https://github.com/k8s-school/kind-helper

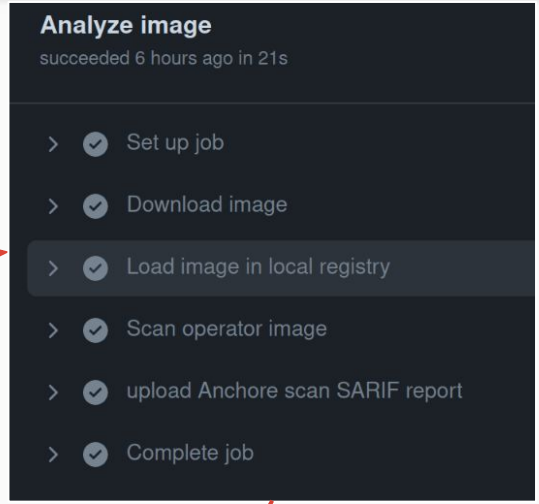
CI in practice: Qserv integration tests





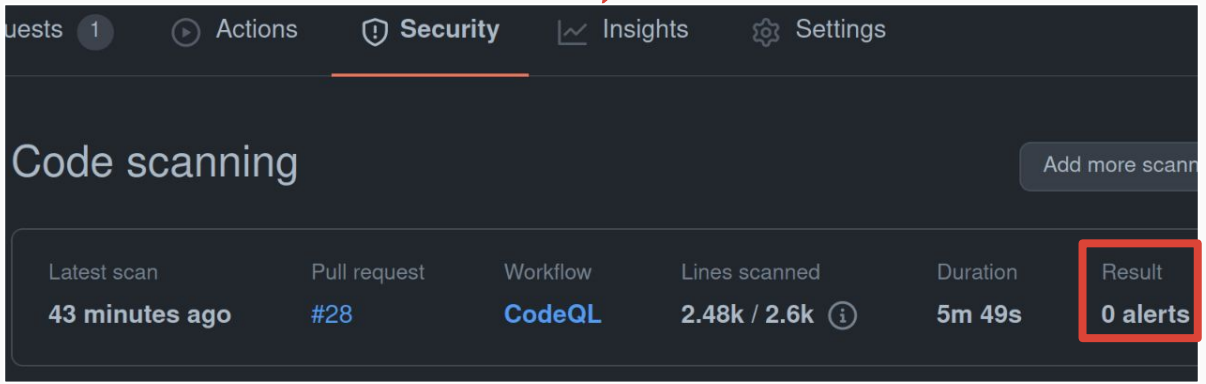
CI in practice: Qserv image scanning



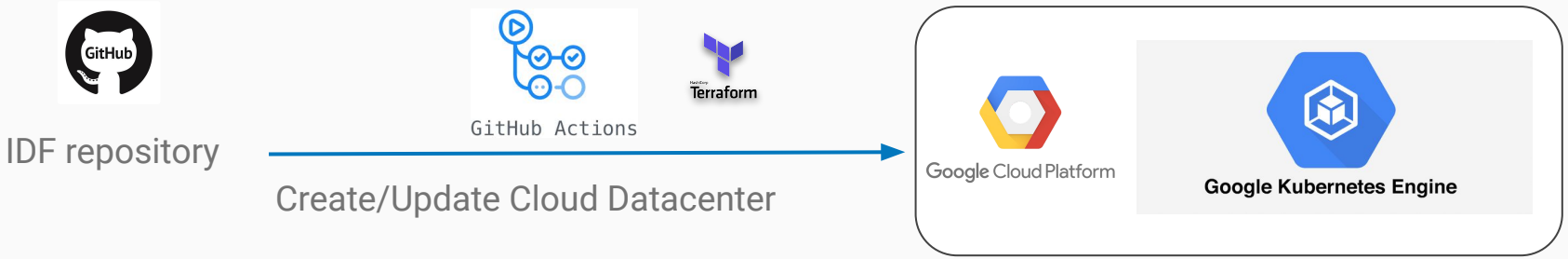


anchore

Vulnerability Scanning & Policy-Compliance for Containers



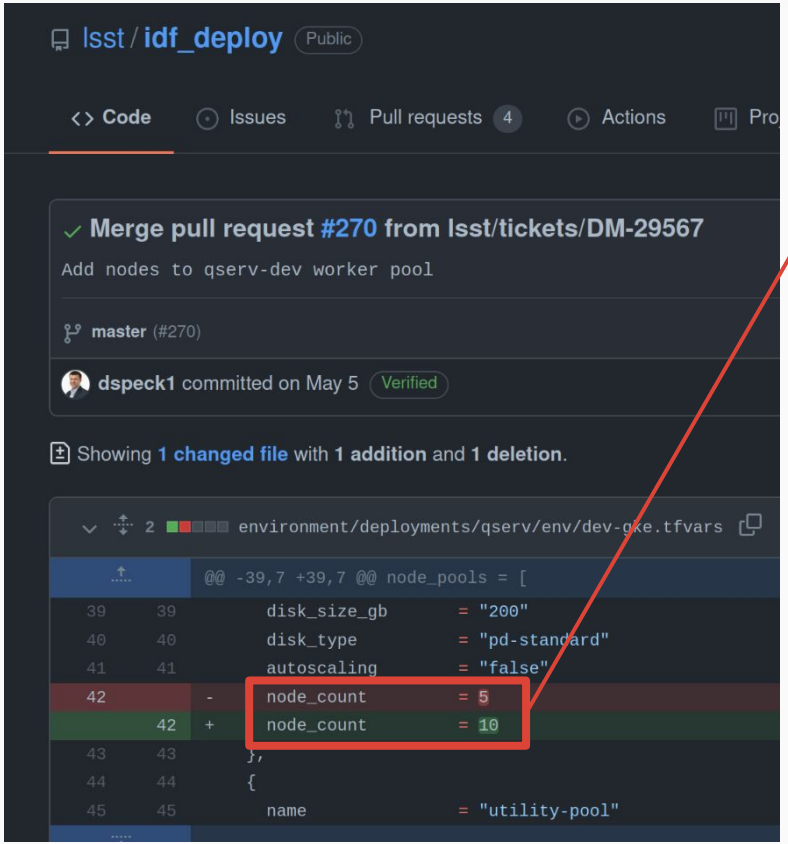
Gitops: CI + IaC

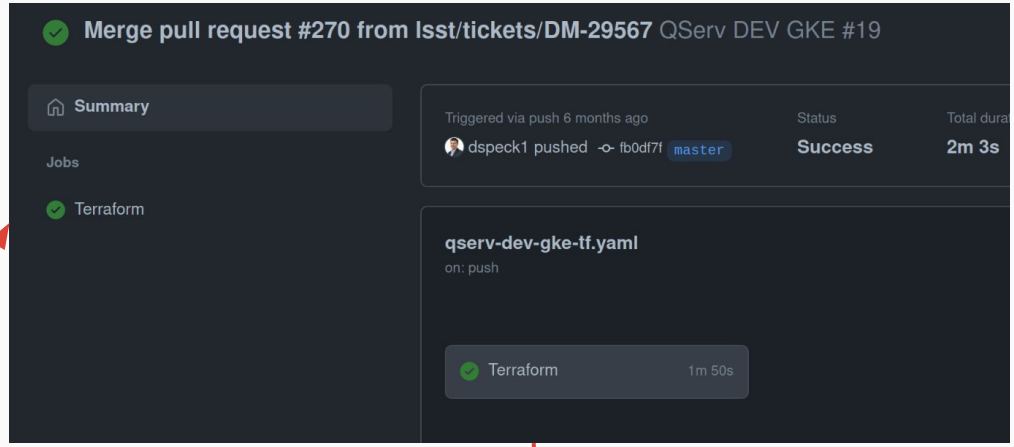


Delegate access to infrastructure management
Track who does what on infrastructure
Recreate infrastructure from scratch
Ease Kubernetes upgrade

Kubernetes is fully managed by Google Cloud / GKE

In practice

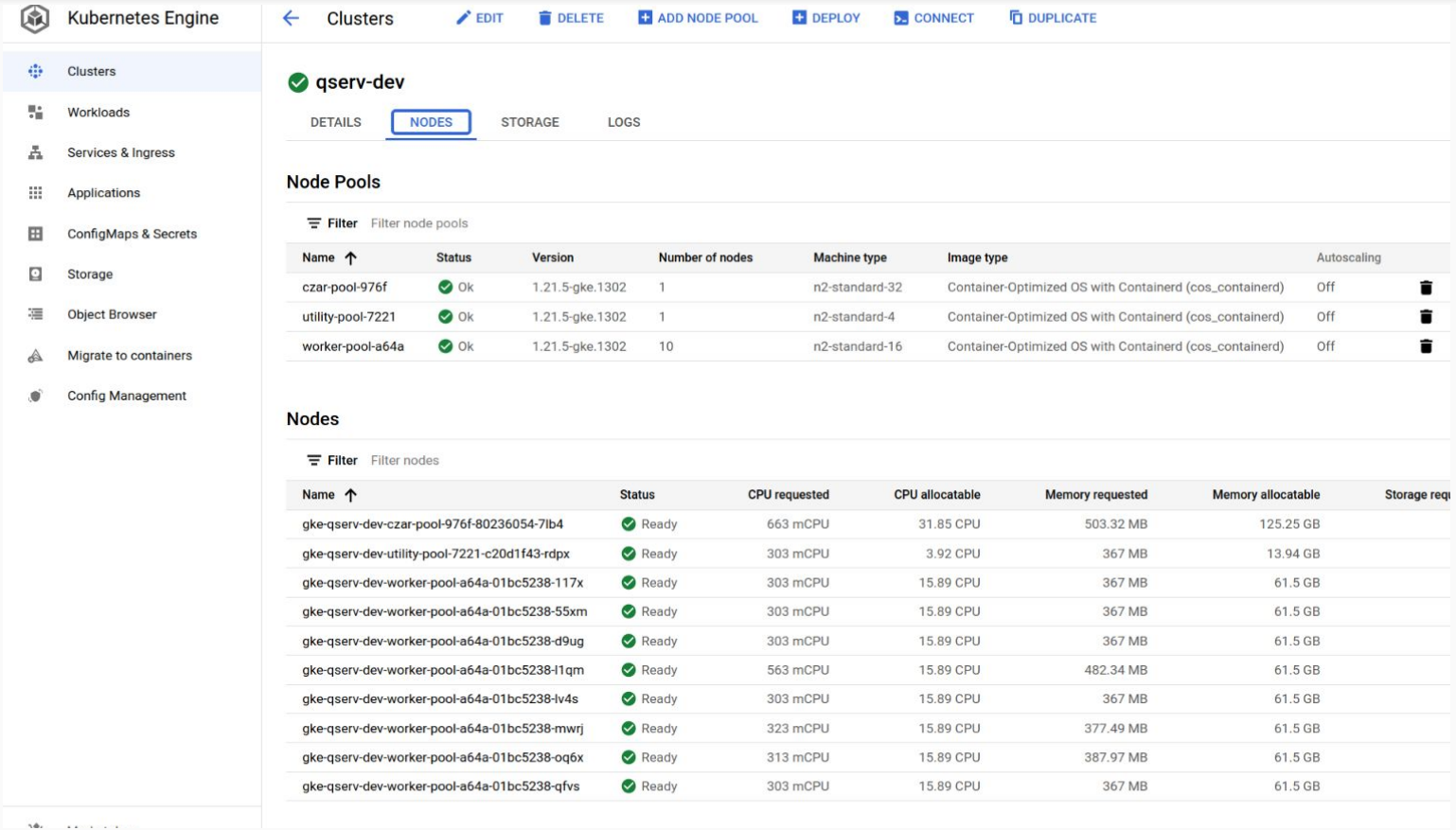




Add five nodes to the GKE cluster

Kubernetes will then allow to easily scale Qserv

Gitops: the Google Cloud Console



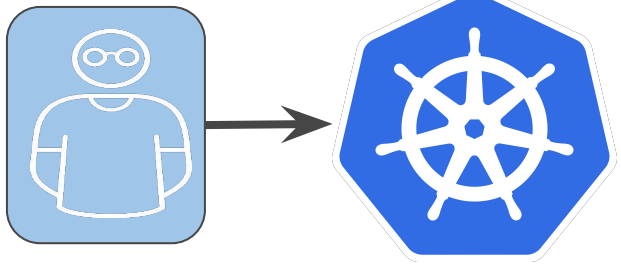
Cloud-Native
Kubernetes operators

Operators embed ops knowledge from the experts

See

- https://kubernetes.io/docs/concepts/extend-kubernetes/operator/
- https://cloud.google.com/blog/products/containers-kubernetes/best-practices-for-building-kubernetes-operators-and-stateful-apps

How does an operator works?



Allow to deploy a complex application with only a few lines of yaml

Software Developer
Kubernetes user

K8s API

Kubernetes operator

Native Kubernetes resources

Custom resource

```
apiVersion: qserv.lsst.org/v1alpha1
kind: Qserv
metadata:
 name: qserv
 namespace: database
spec:
 czar:
   image:
qserv/lite-qserv:2021.10.1-rc1-37-g45646a1cb
   replicas: 1
   storage: 1Ti
 worker:
   image:
qserv/lite-qserv:2021.10.1-rc1-37-g45646a1cb
   replicas: 10
   ...
```

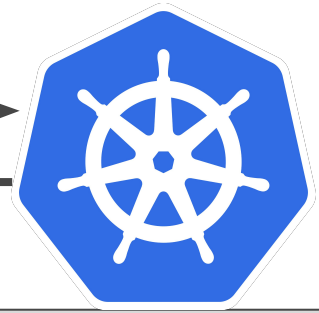
Custom Kubernetes controller

Watch Event

Reconcile

Custom resource definition

here Qserv



Deployments
StatefulSets
Autoscalers
Secrets
Config maps

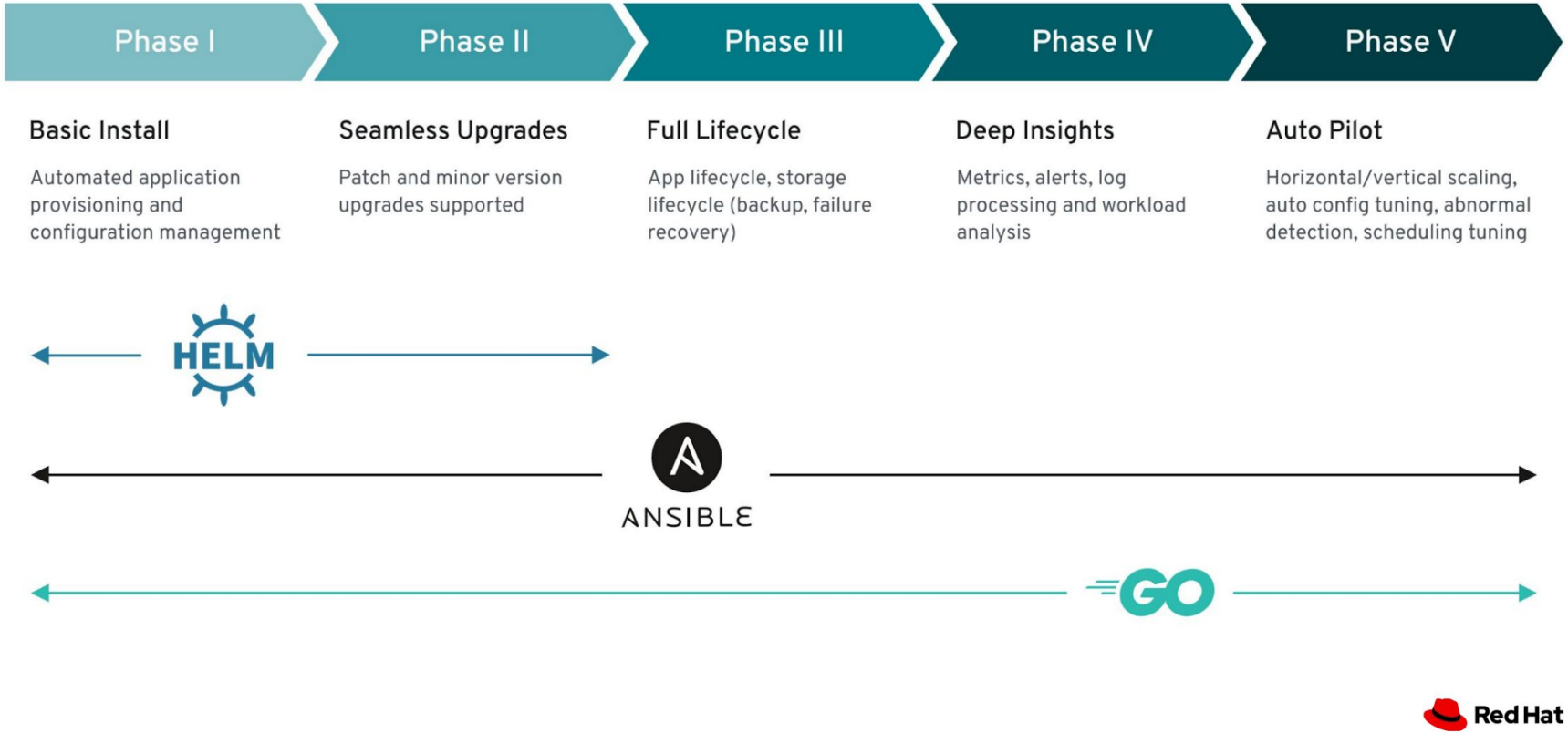
Why should you use an operator?

Operators: both sysadmin + application experts

- Resize/Upgrade
- Reconfigure
- Backup
- Healing

The Sysadmin

Types of operators

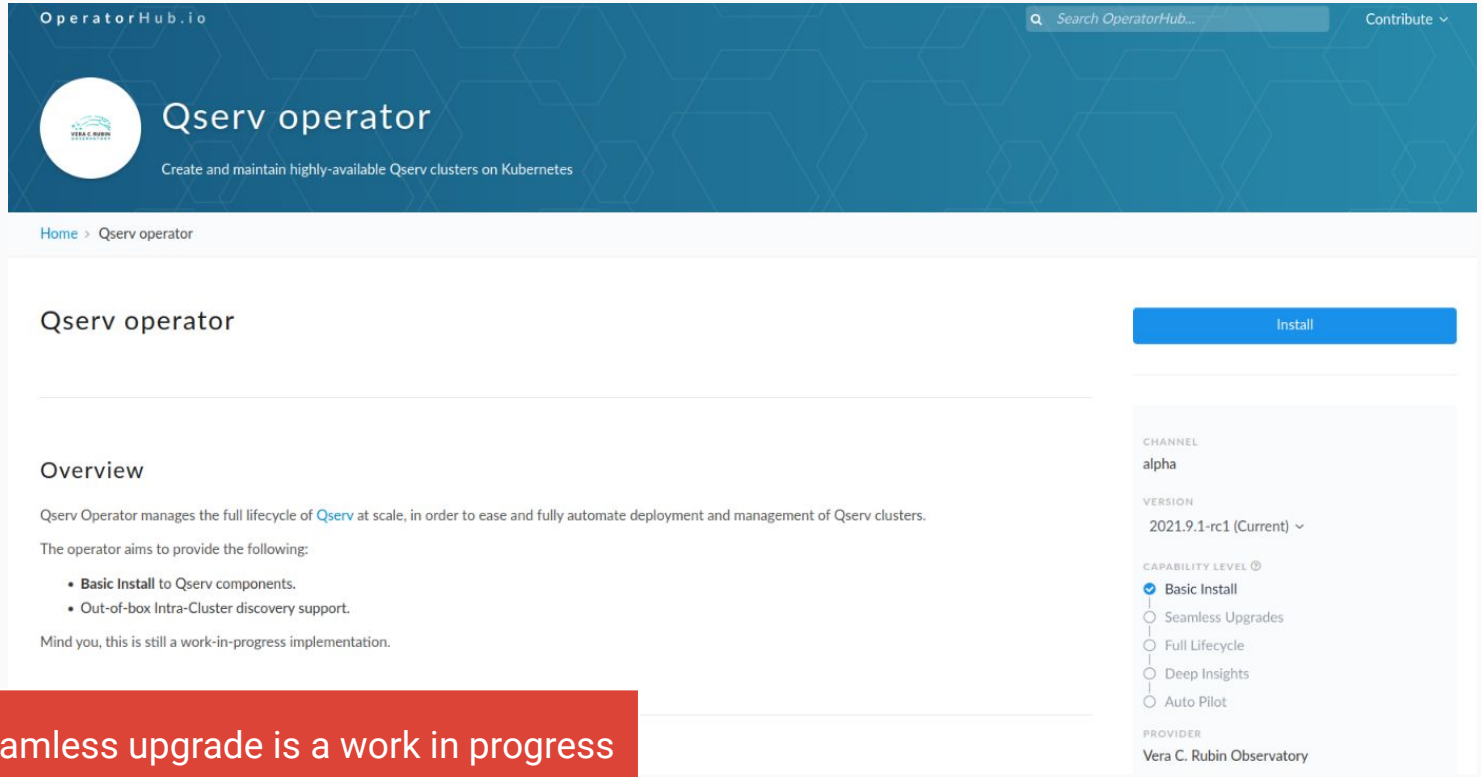


Operator across the industry

OperatorHub.io | The registry for Kubernetes Operators

Qserv is available on operatorHub

https://operatorhub.io/operator/qserv-operator



Seamless upgrade is a work in progress

Cloud-Native
Storage management

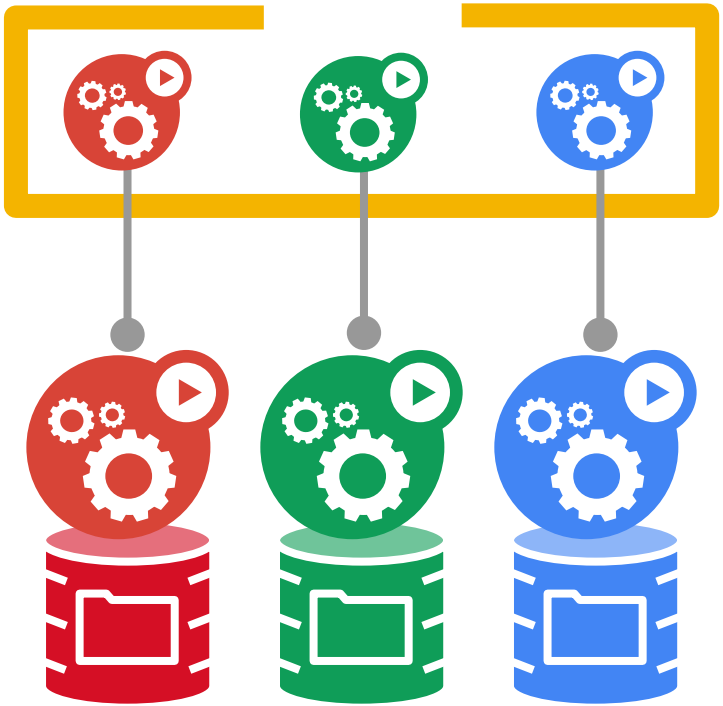
Qserv is deployed using StatefulSets

Goal: enable clustered software on Kubernetes

- Qserv, but also mysql, redis, zookeeper, ...

Clustered apps need “identity” and have sequencing constraints

- stable hostname, available in DNS
- an ordinal index
- stable storage: linked to the ordinal & hostname
- discovery of peers for quorum
- startup/teardown ordering



Storage management

GKE: Dynamic storage provisionning

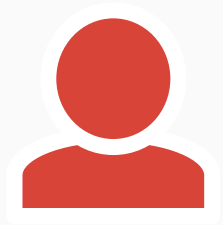
User deploy Qserv instance

Create PVClaims

Google Storage creates automatically PersistentVolume+Google Disks (ssd+hdd)

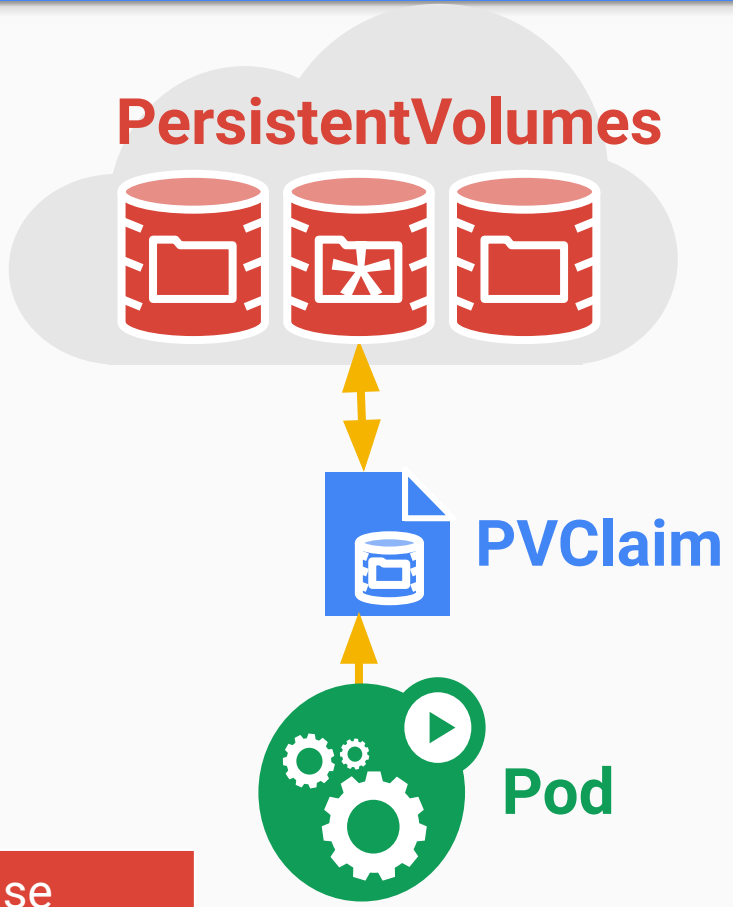
On-premise:

Storage is manually declared to Kubernetes (via PV) and created



Cluster Admin

User



Easier on GKE, but better performance on-premise

What we need to improve

Backup and restore

Use Kubernetes standard API (i.e. VolumeSnapshots)?
Use more advanced but non standard tools like Portworx, StorageOS, Longhorn?

Dynamic provisioning on-premise

Looking at CNCF Landscape, but no standard solution yet

Objective:
Use the very same procedures to manage the storage On-Premise and on Commercial Cloud

Cloud-Native Workflows

A powerful data ingest workflow

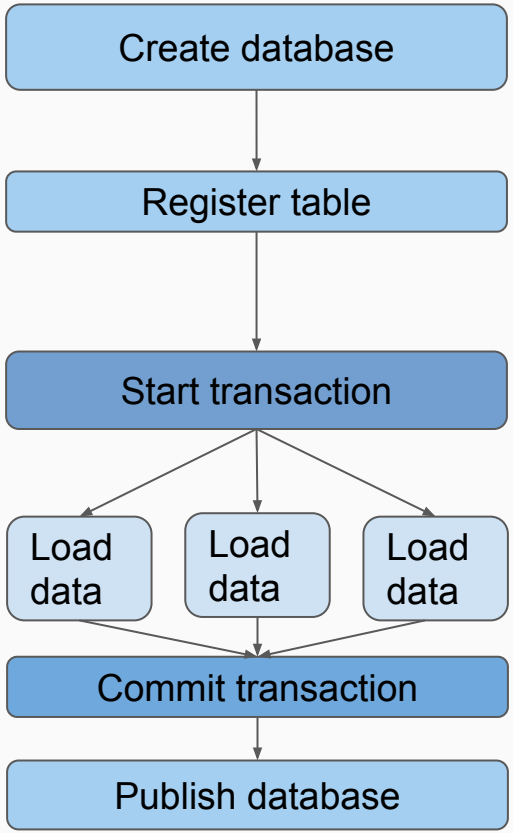
Qserv has a powerful distributed ingest algorithm

Flexible but require orchestrating tasks (DAG)

Argo Workflow project help us a log

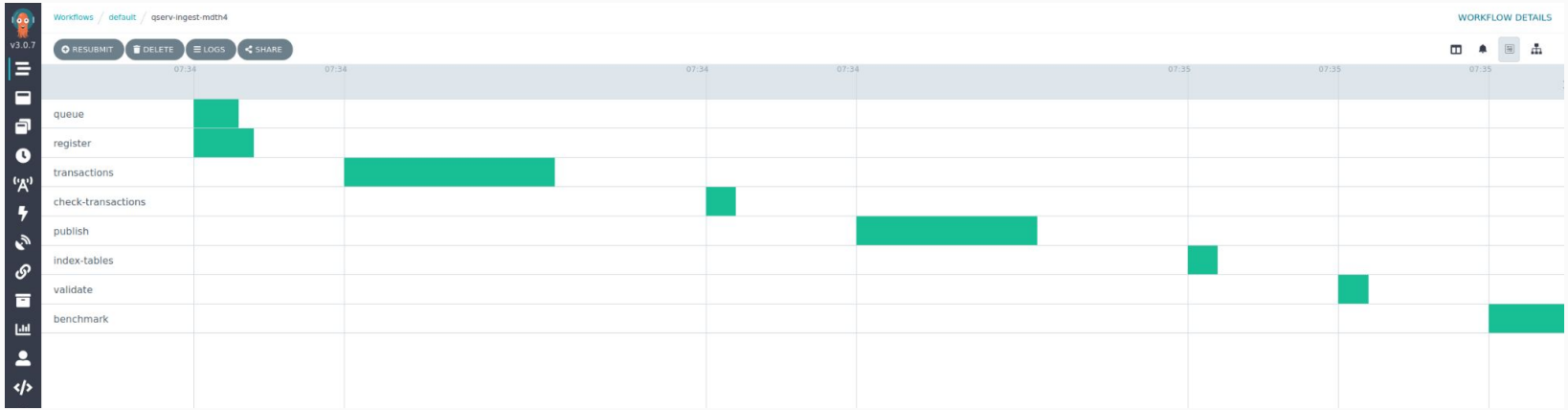
Case study 2021: **Implementation of a large-scale data loading algorithm**

Ingestion of 15000 files and 15TB in 1h30

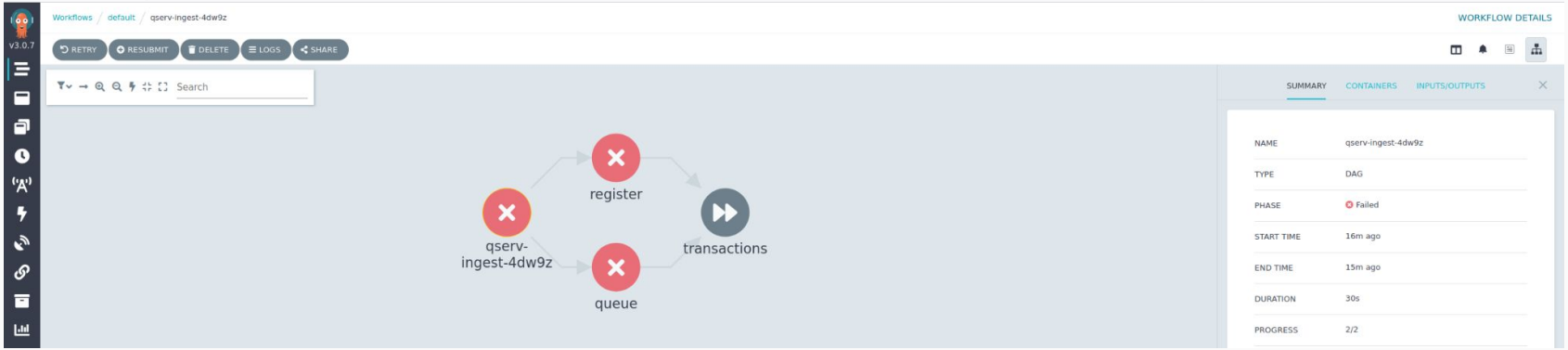


Argo: screenshots

```
fjammes@clrinfoport18 ~ argo get @latest | tail -n 15
STEP                        TEMPLATE      PODNAME                          DURATION
 ✔ qserv-ingest-mdth4       main
 ├─✔ queue                  ingest-step   qserv-ingest-mdth4-2075476264    3s
 ├─✔ register               ingest-step   qserv-ingest-mdth4-964548720     4s
 ├─✔ transactions           transactions  qserv-ingest-mdth4-1041421248    14s
 ├─✔ check-transactions     ingest-step   qserv-ingest-mdth4-3195504171    2s
 ├─✔ publish                ingest-step   qserv-ingest-mdth4-4256901816    12s
 ├─✔ index-tables           index-tables
 │  └─✔ index-tables        ingest-step   qserv-ingest-mdth4-1866502525    2s
 ├─✔ validate               ingest-step   qserv-ingest-mdth4-493206715     2s
 └─✔ benchmark              benchmark
    └─✔ benchmark           ingest-step   qserv-ingest-mdth4-1797710727    5s
```



Argo: screenshots



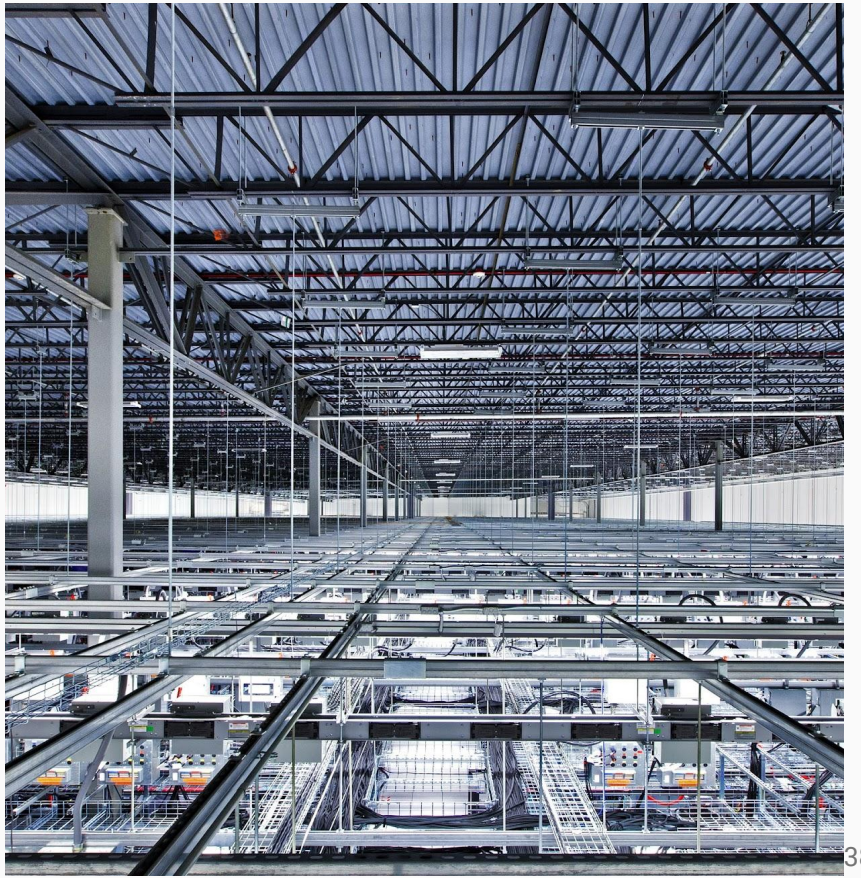
Public cloud: pros and cons

Pros

- ★ Flexibility (access, provisioning)
- ★ Excellent support
- ★ Low maintenance
- ★ Cool proprietary features

Cons

- ★ Expensive / Cost difficult to manage
- ★ Vendor lock-in
- ★ Hide Kubernetes internals (black box)
- ★ Run slower than bare-metal (~25%)



Conclusion

1. Qserv is going on
2. Container orchestration helps a lot
3. Commercial cloud might be an alternative

Q&A

Fabrice JAMMES
Laboratoire de Physique de Clermont

Fritz Mueller
SLAC National Accelerator Laboratory