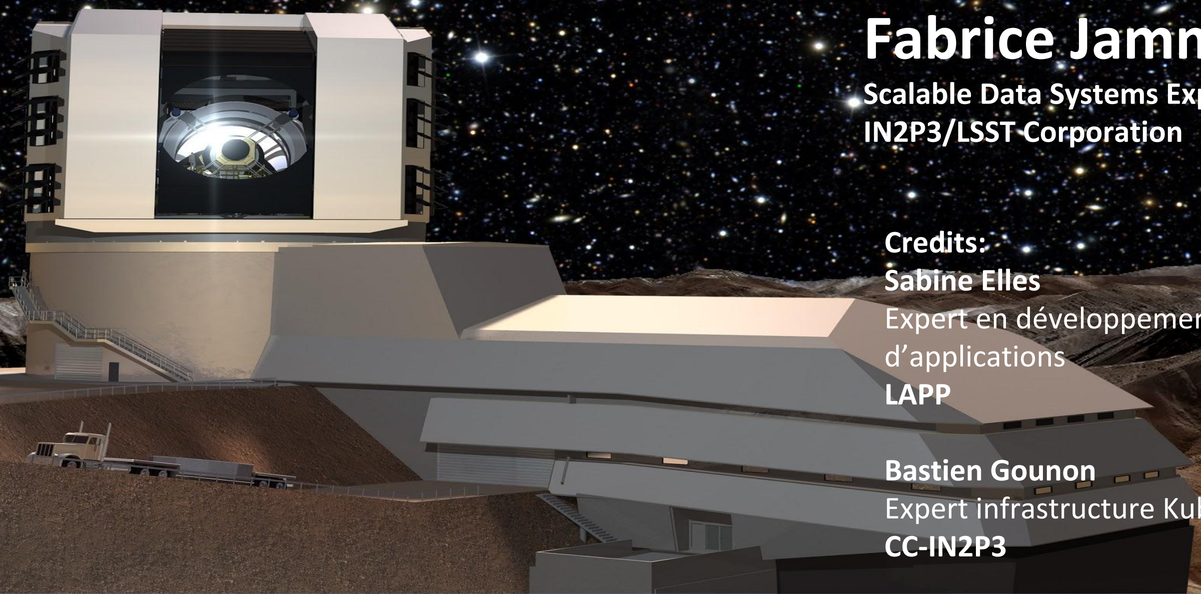
Des péta-octets de données dans Kubernetes, c'est possible!



Fabrice Jammes
Scalable Data Systems Expert
IN2P3/LSST Corporation

Credits:
Sabine Elles
Expert en développement d'applications
LAPP

Bastien Gounon
Expert infrastructure Kubernetes
CC-IN2P3

Agenda

1. Large Synoptic Survey Telescope
2. Qserv: LSST Petascale database
3. Benefits of Cloud-Native
4. On-premise vs Public Cloud

LSST in short

LSST
Large Synoptic Survey Telescope

Large Synoptic Survey Telescope

Large aperture, wide-field, ground-based survey telescope
The largest imager ever built for astronomy

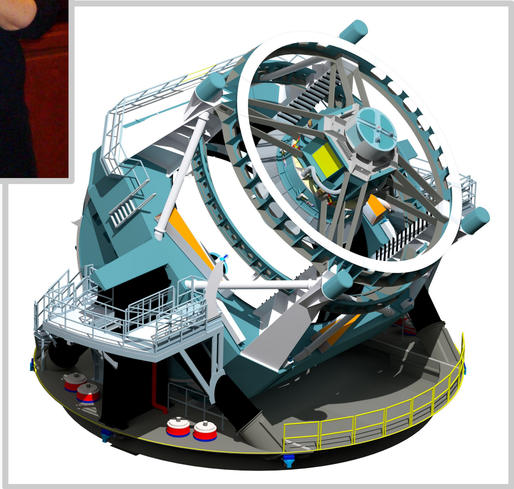
NSF

U.S. DEPARTMENT OF ENERGY

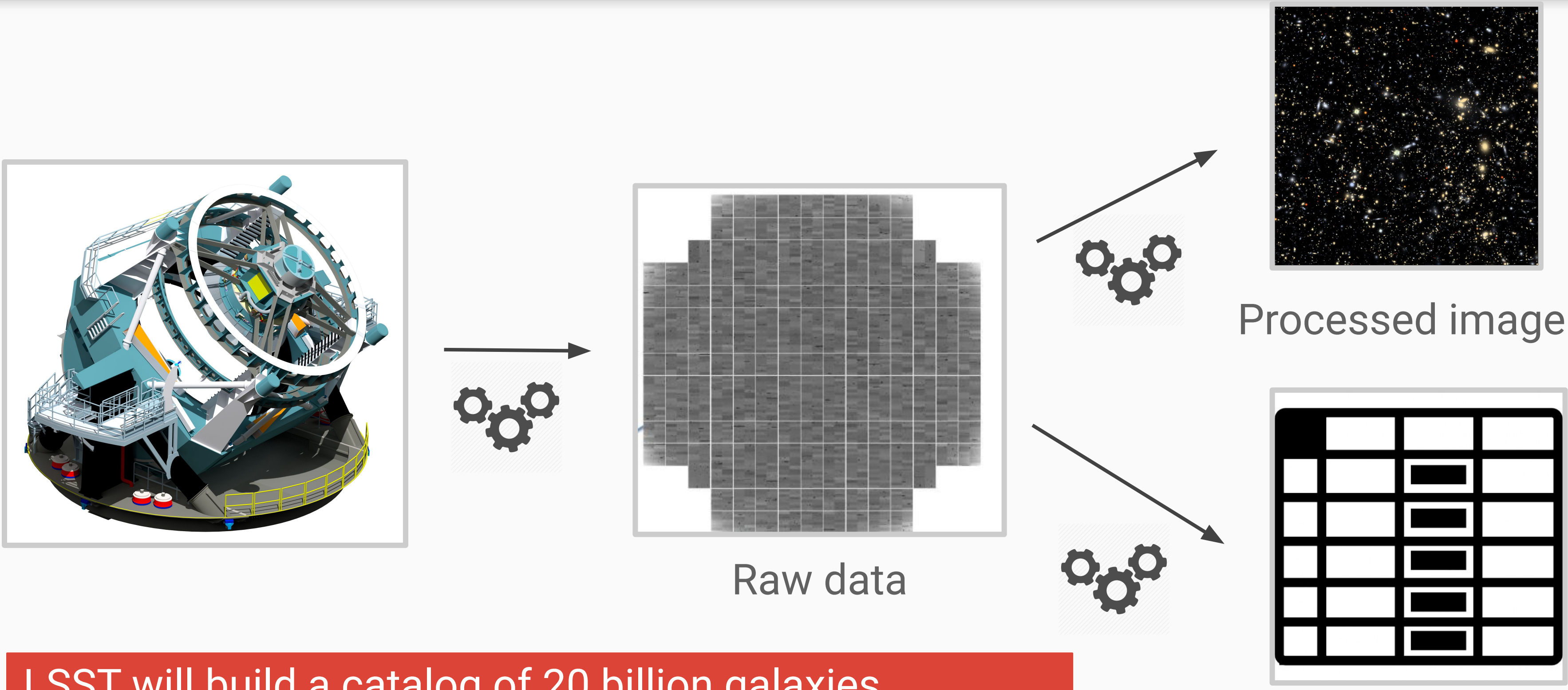
Characteristics

- ★ All visible sky in 6 bands
- ★ ~20000□
- ★ 15 seconds exposures, 1 visit/3 days
- ★ During 10 years!
- ★ **60 PB of raw data**

IN2P3
Institut National de Physique Nucléaire et de Physique des Particules



80+ PB of astronomical catalog



Raw data

Processed image

LSST will build a catalog of 20 billion galaxies and 17 billion stars and their associated physical properties

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

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 LSST Petascale database

Who we are

Database and Data access team

- ★ 10 engineers at Stanford University + 1 IN2P3
  - ○ *Software development*

Operations teams

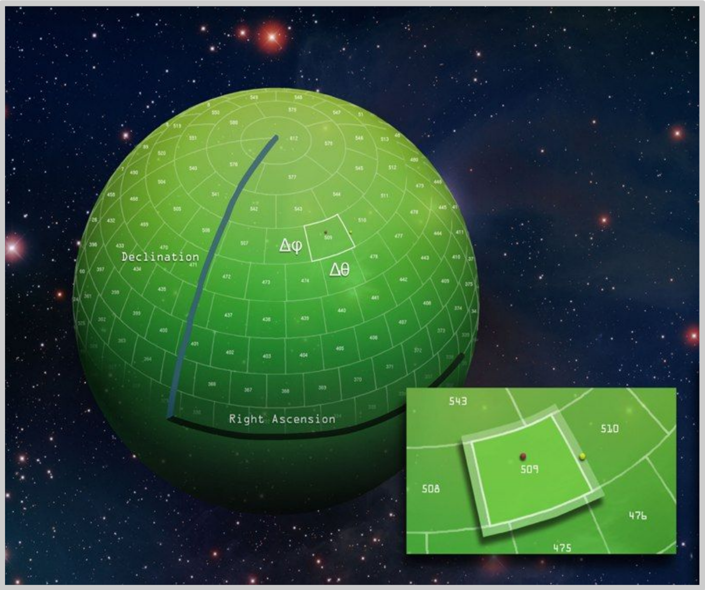
- ★ 5 sysadmins at NCSA/IN2P3
  - ○ *Large Scale development platforms*
  - ○ *Cloud Native / Kubernetes*
  - ○ *System administration, Monitoring*

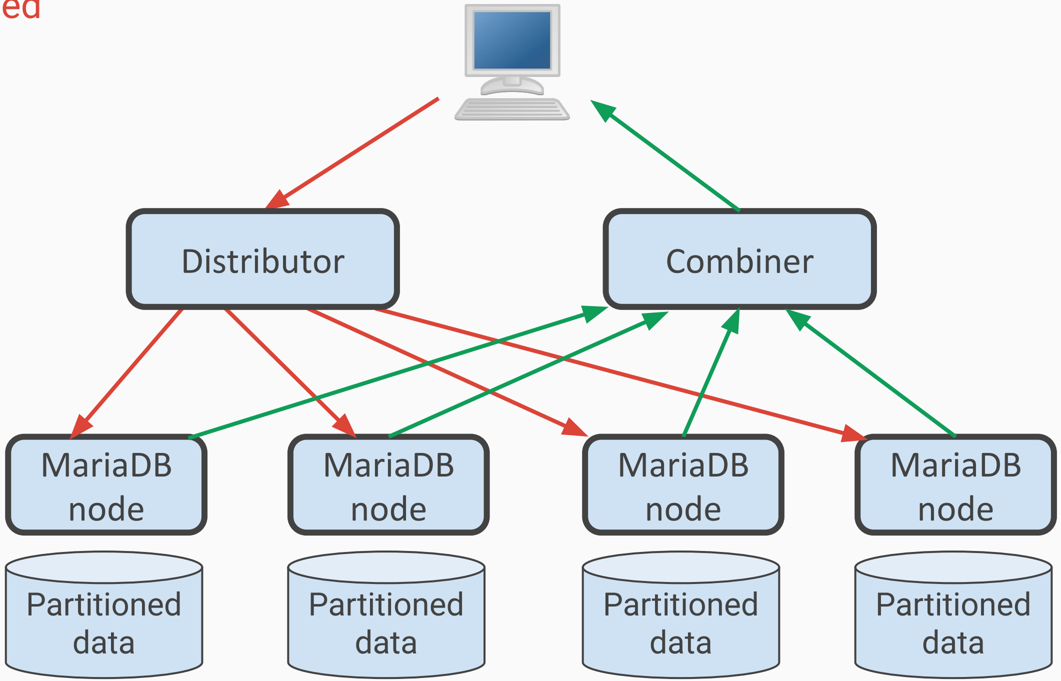
Qserv design

Relational database, 100% open source

Spatially-sharded with overlaps

Map/reduce-like processing, highly distributed





From Cloud-Native to Bare-Metal

Target for production

~1000 nodes cluster in 2 international Academic data-centers

Running now

Development platform (CC-IN2P3)

1000 cores, 15 TB memory

15 PB storage

*=> Large scale test: **300 TB synthetized data***

=> Ingestion of DESC-DC2 data (1 TB)

Protótype Data Access Center (NCSA)

500 cores, 4 TB memory

700 TB storage,

=> WISE catalog ("real" dataset)

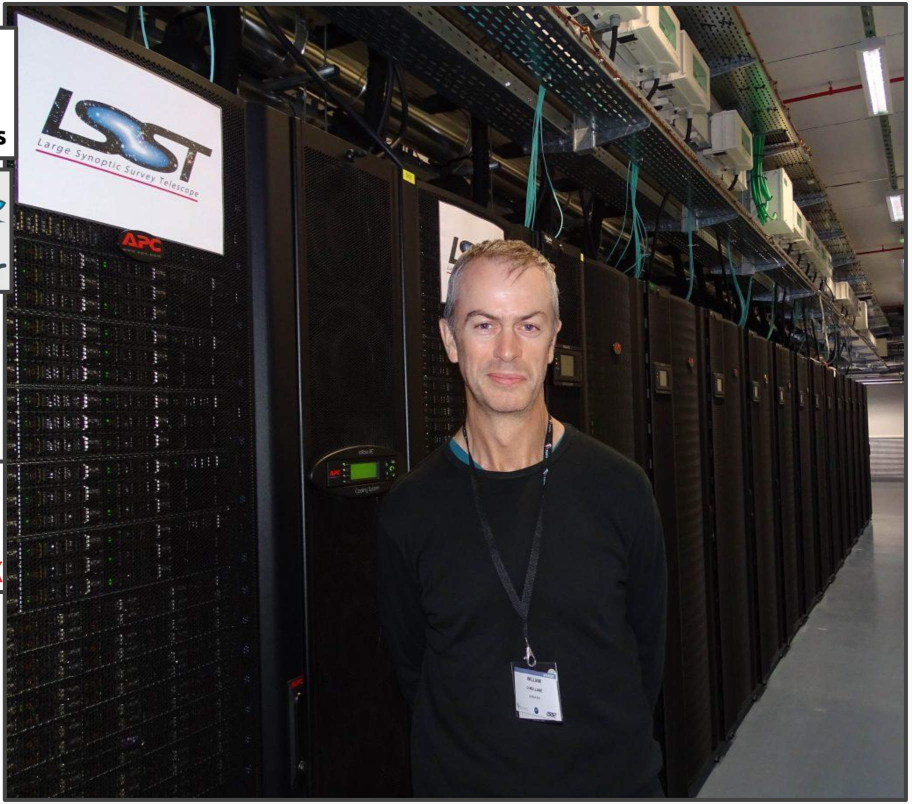
kubernetes

docker

HashiCorp Terraform

openstack

ceph



Centre de Calcul
de l'Institut National de Physique Nucléaire
et de Physique des Particules

Qserv Platform @ CC-IN2P3

Bastien Gounon

Platform Overview

dedicated hardware:

3 x Kubernetes masters (40x2.2GHz, 64GB RAM)

control-plane

2 x Qserv masters (40x2.2GHz, 256GB RAM, 8TB SSD RAID1)

user interaction, result aggregation

20 x Qserv workers (40x2.2GHz, 256GB RAM, 48TB HDD RAID5)

database workload and storage

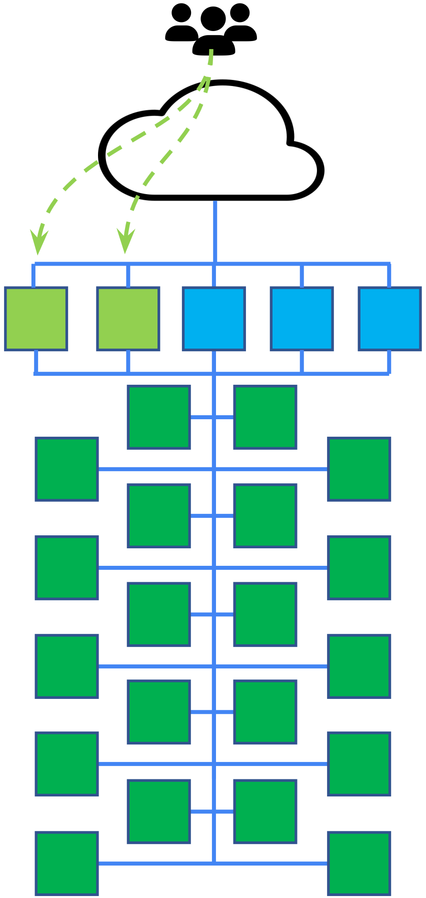
=> 25 nodes Kubernetes cluster (v1.15.3)

deployed via Puppet using puppetlabs/kubernetes plugin

CRI : containerd

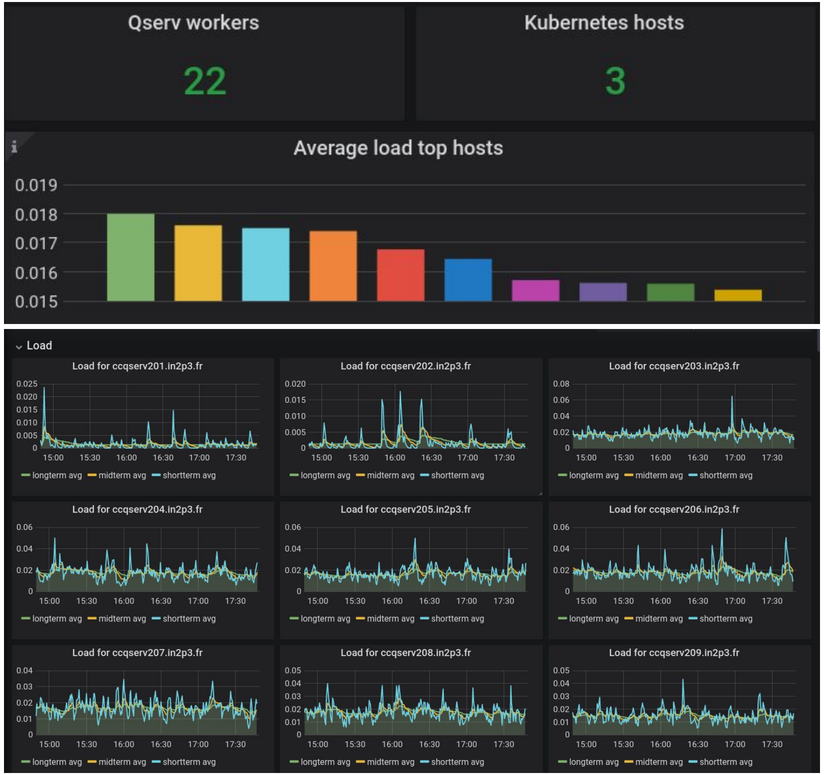
CNI : weave

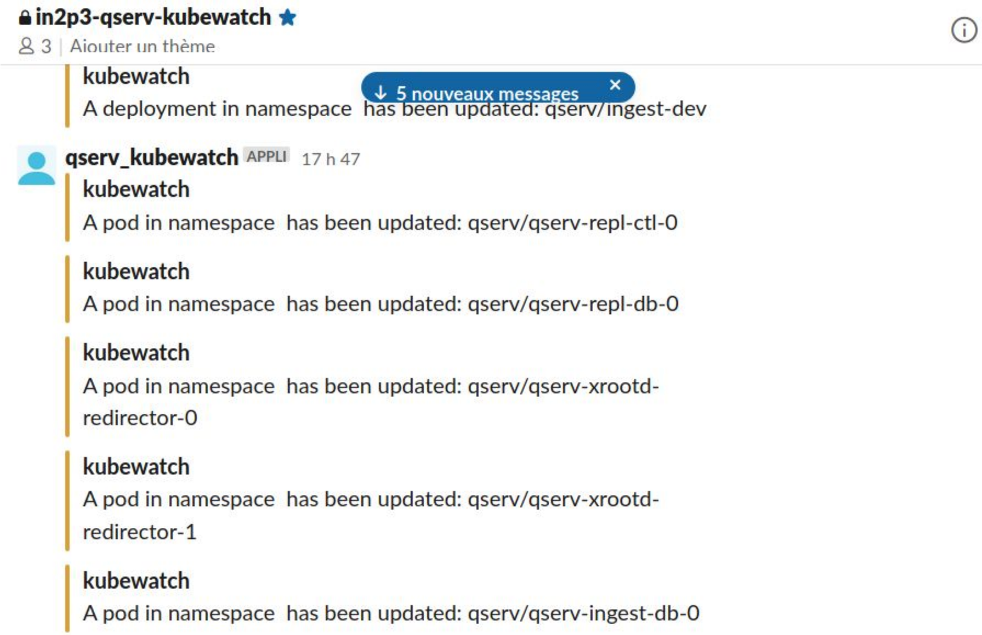
token-based authentication for cluster administration



Monitoring Tools

ElasticSearch/Grafana activity dashboard





in2p3-qserv-kubewatch

3 | Ajouter un thème

kubewatch

A deployment in namespace has been updated: qserv/ingest-dev

qserv_kubewatch APPLI 17 h 47

kubewatch

A pod in namespace has been updated: qserv/qserv-repl-ctl-0

kubewatch

A pod in namespace has been updated: qserv/qserv-repl-db-0

kubewatch

A pod in namespace has been updated: qserv/qserv-xrootd-redirector-0

kubewatch

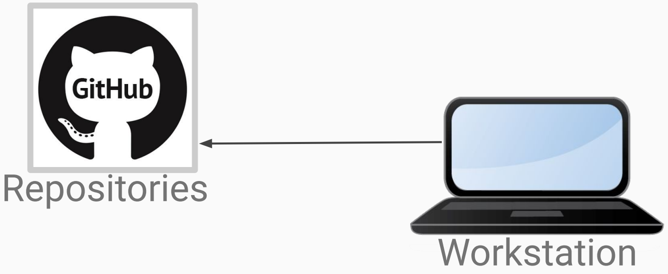
A pod in namespace has been updated: qserv/qserv-xrootd-redirector-1

kubewatch

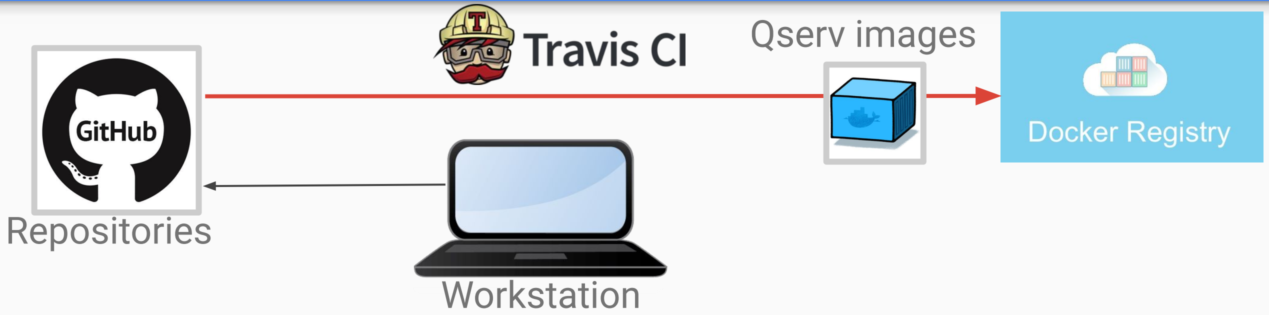
A pod in namespace has been updated: qserv/qserv-ingest-db-0

Benefits of Cloud-Native

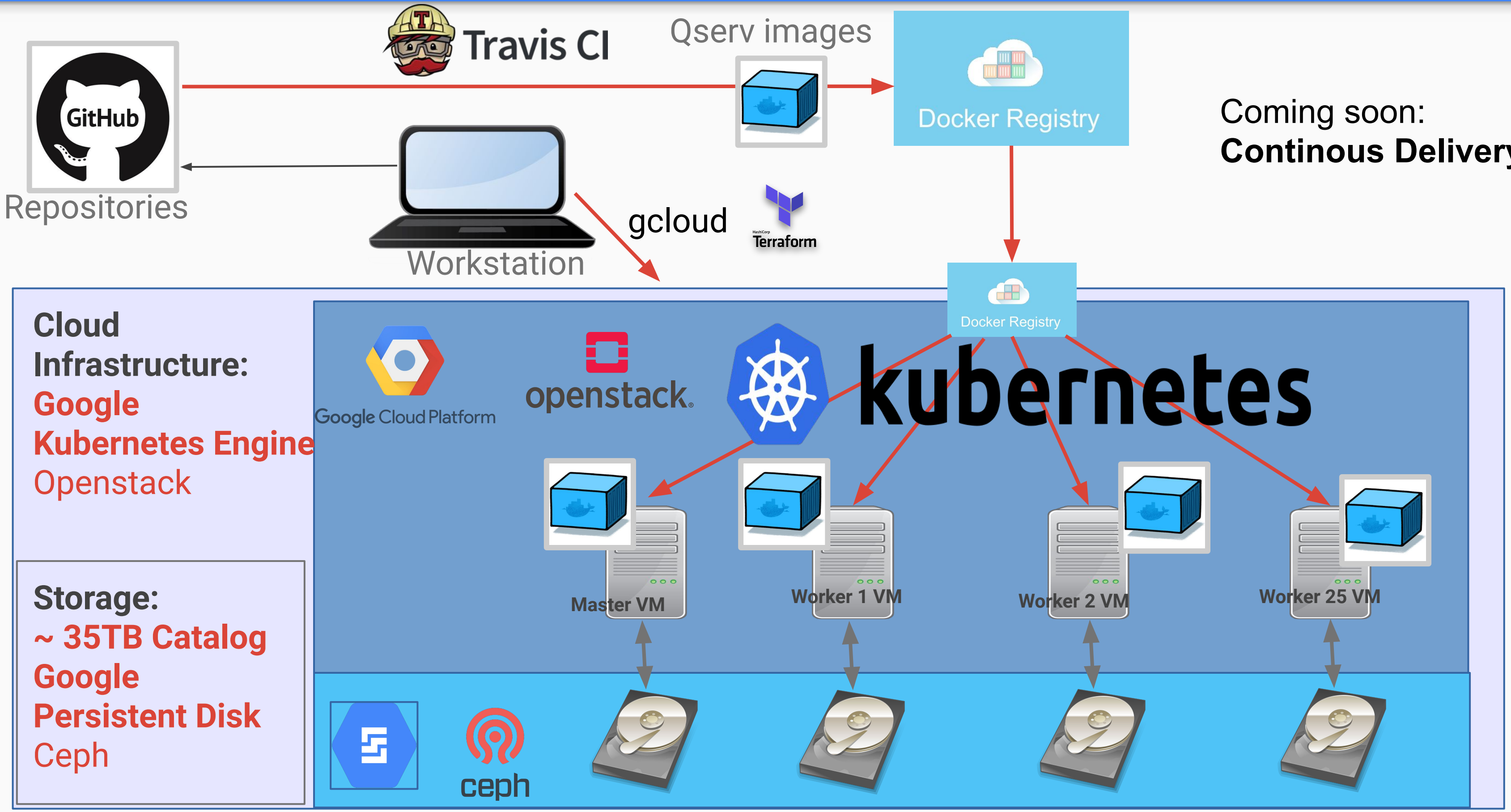
Automated Qserv deployment



Automated Qserv deployment



Automated deployment: Cloud Native



Travis CI

Qserv images

Docker Registry

Repositories

Workstation

gcloud

Terraform

Cloud Infrastructure:
Google Kubernetes Engine
Openstack

Google Cloud Platform

openstack.

Docker Registry

kubernetes

Master VM

Worker 1 VM

Worker 2 VM

Worker 25 VM

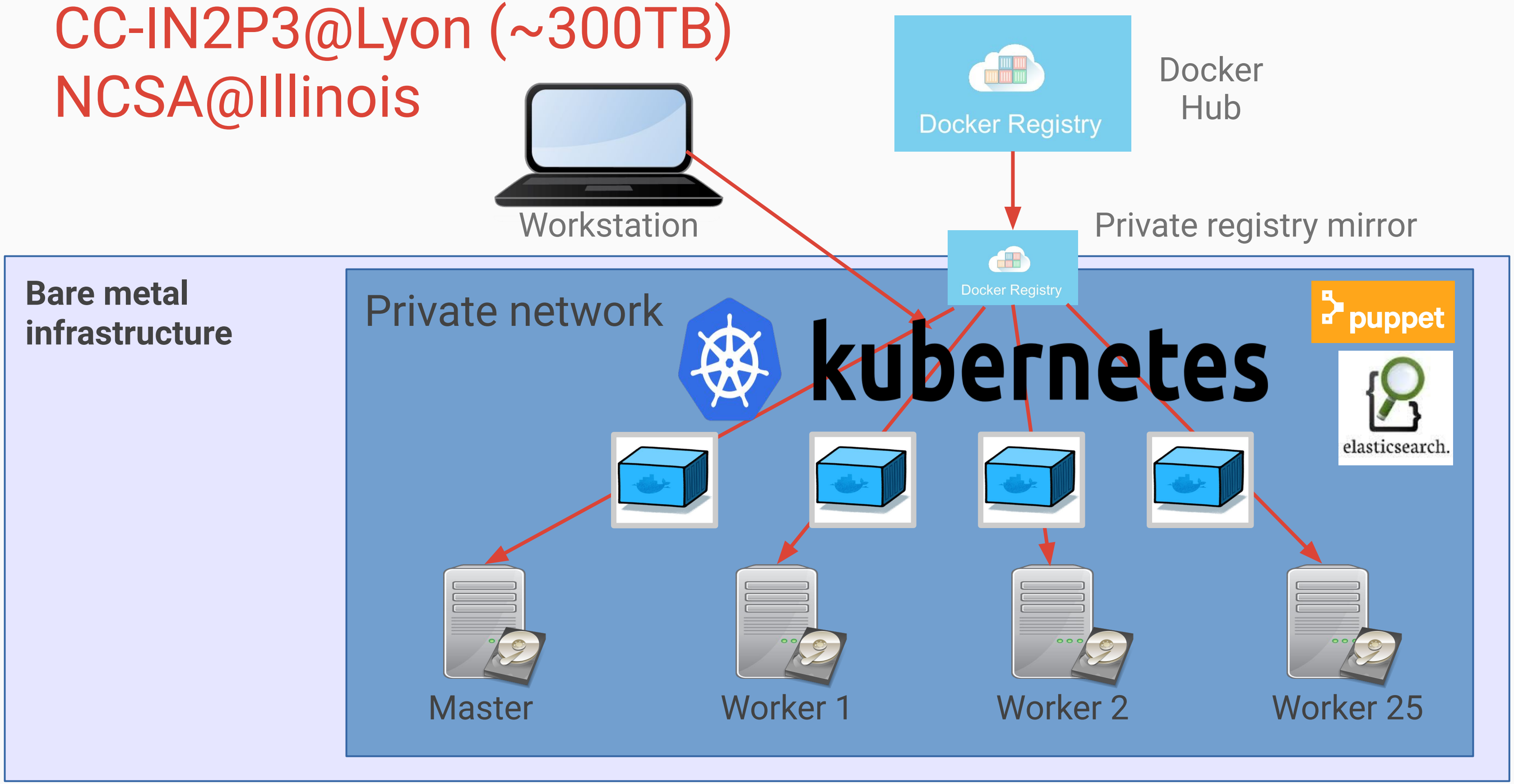
Storage:
~ 35TB Catalog
Google Persistent Disk
Ceph

ceph

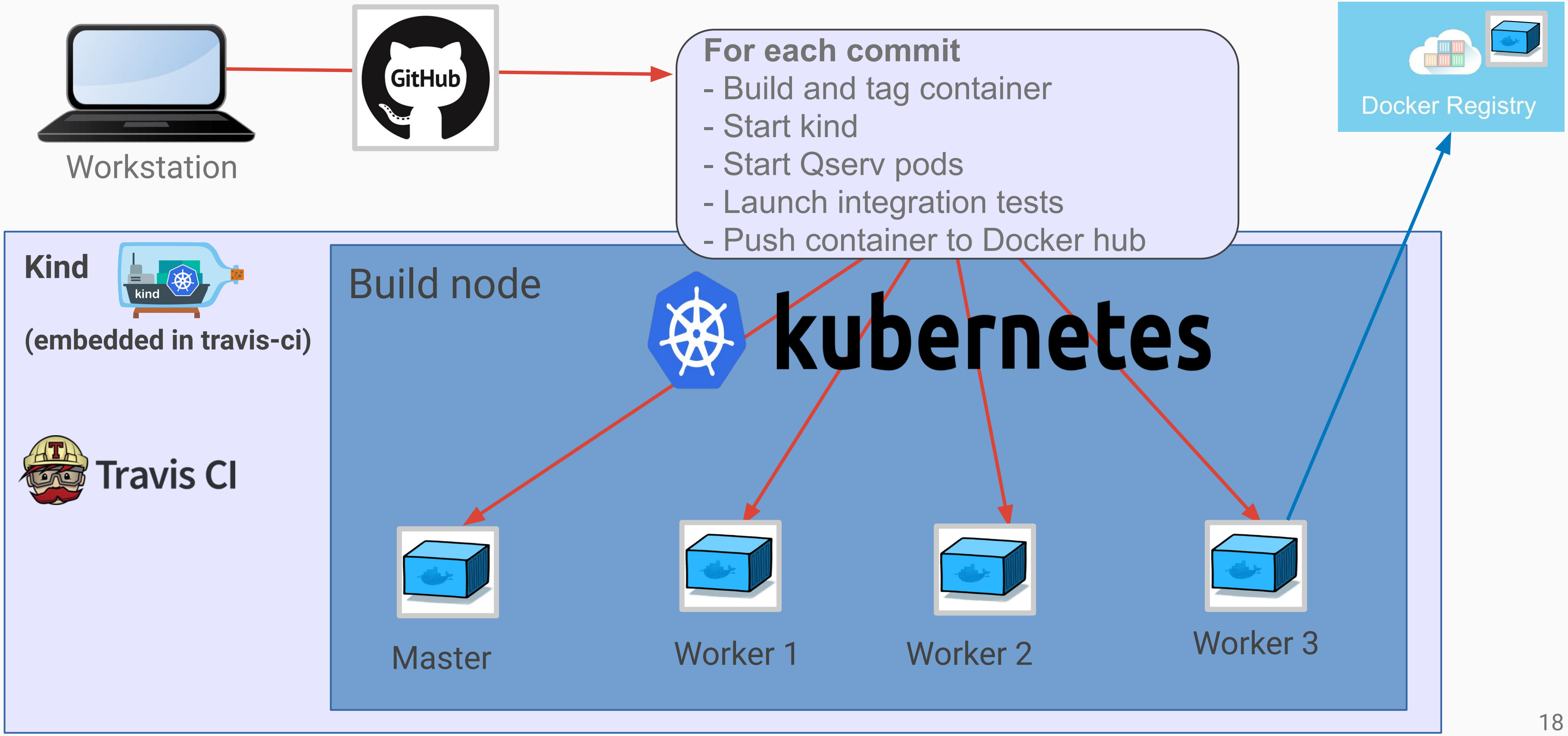
Coming soon:
Continous Delivery

Automated deployment: bare-metal

CC-IN2P3@Lyon (~300TB)
NCSA@Illinois



Automated deployment: CI

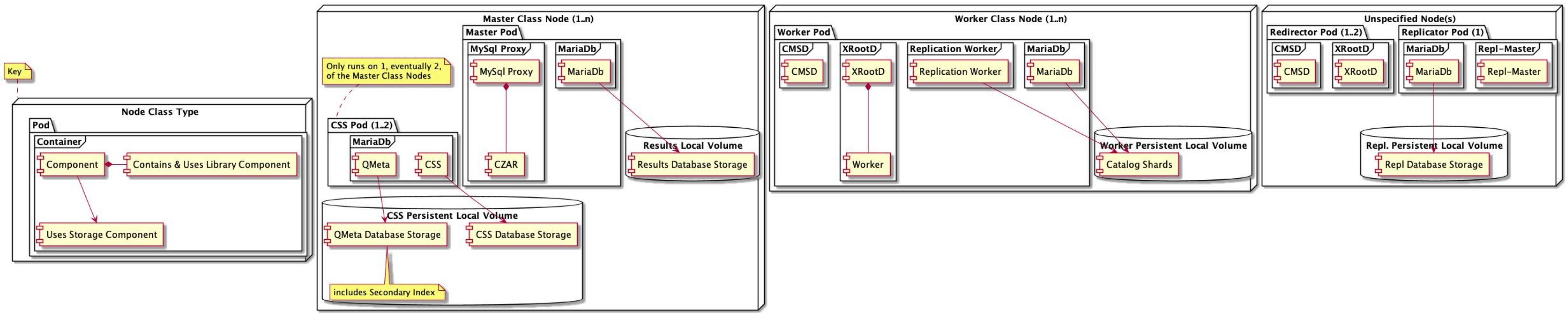


For each commit
- Build and tag container
- Start kind
- Start Qserv pods
- Launch integration tests
- Push container to Docker hub

K8s + Microservice features

- ★ Automated scaling
- ★ Container scheduling
- ★ Auto-healing
- ★ Continuous deployment

- ★ Volume management (storage)
- ★ Easy monitoring
- ★ Healthcheck
- ★ Security

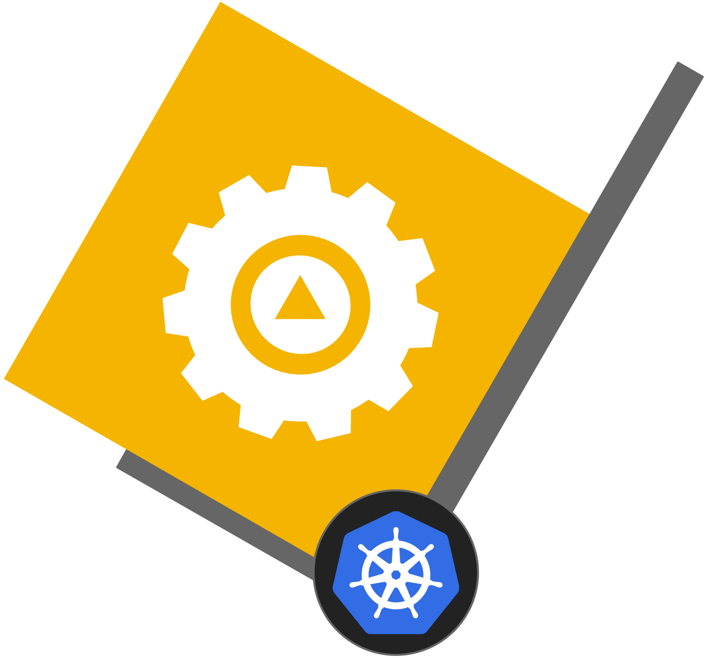


The killer feature: workload portability

Result: Portability

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

Easily move your distributed application anywhere Kubernetes is supported, in seconds.



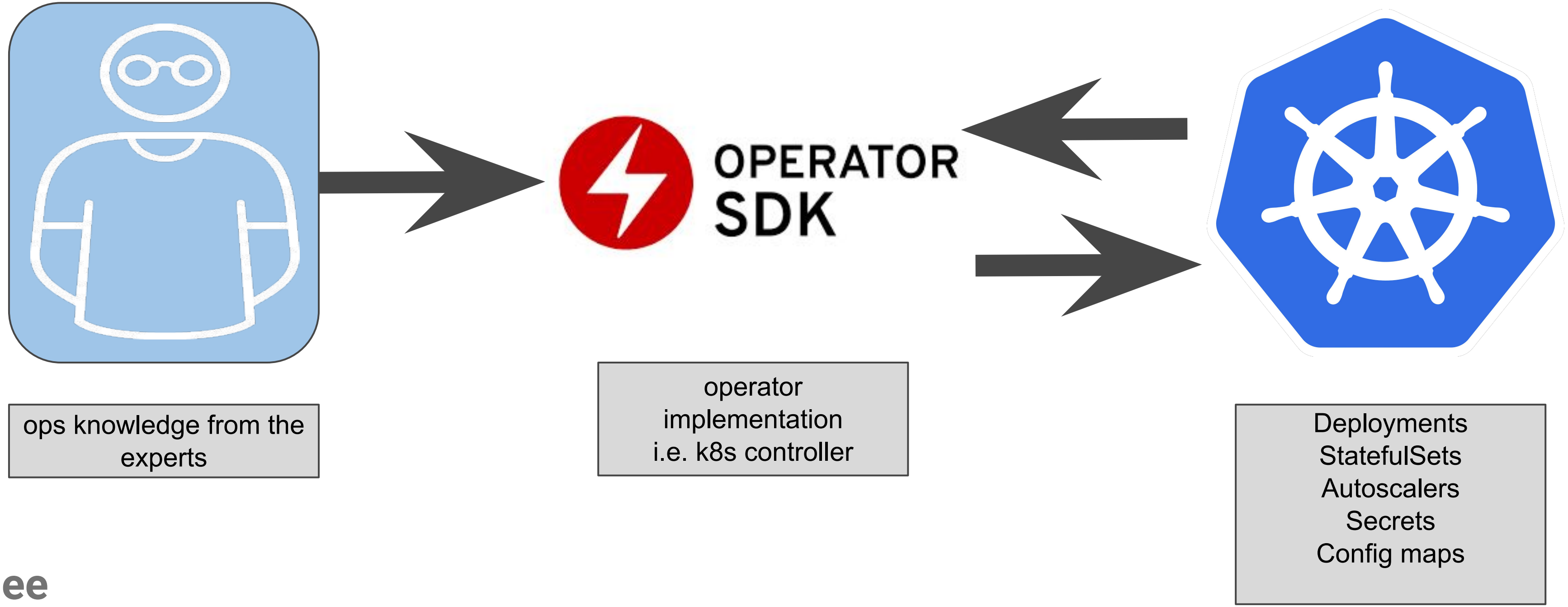
Operators: adding sysadmin knowledge inside k8s

Operators: both sysadmin + application experts

- Resize/Upgrade
- Reconfigure
- Backup
- Healing

The Sysadmin

Operators embed ops knowledge from the experts



ops knowledge from the experts

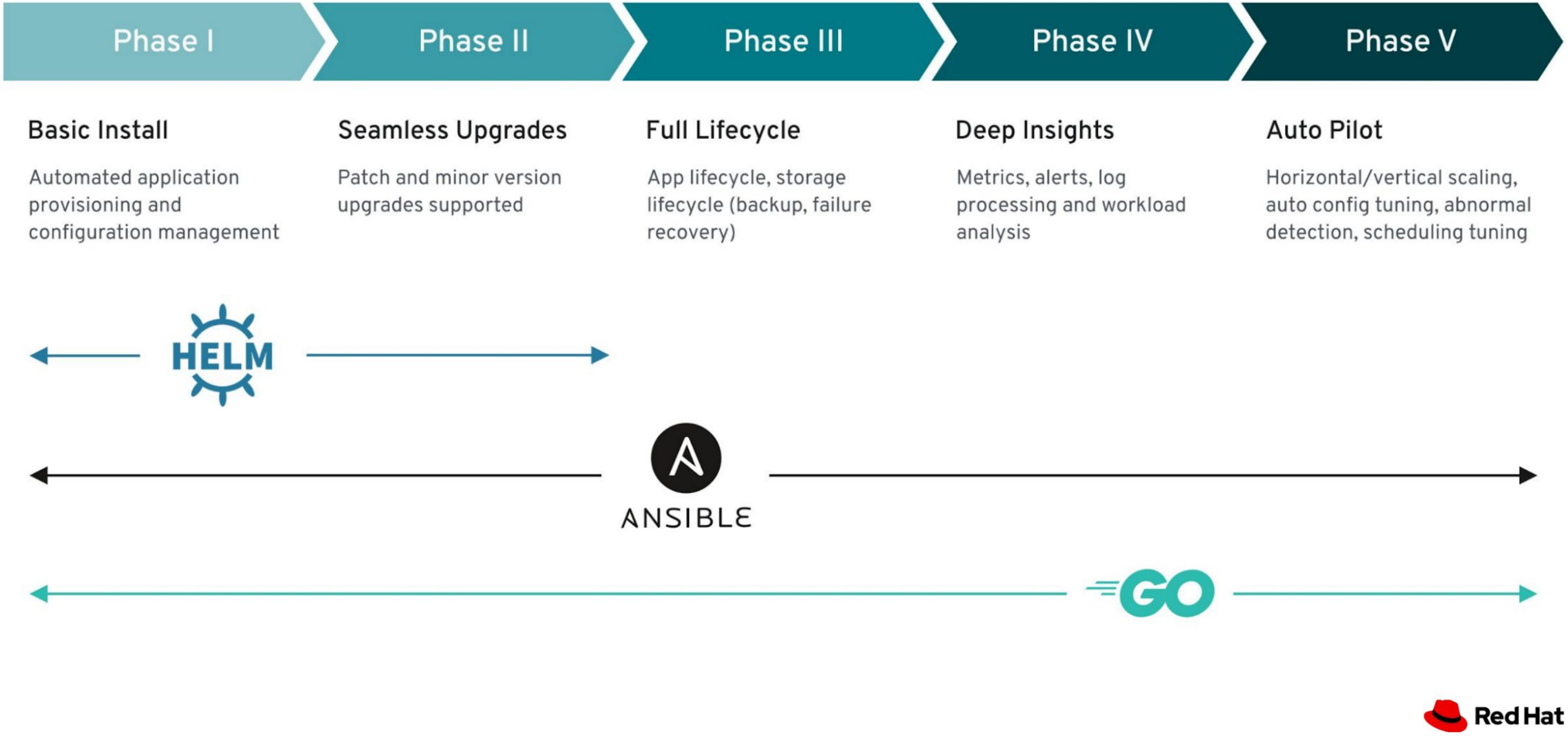
operator implementation
i.e. k8s controller

Deployments
StatefulSets
Autoscalers
Secrets
Config maps

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

Operator SDK: types of operators



Phase I

Basic Install

Automated application provisioning and configuration management

Phase II

Seamless Upgrades

Patch and minor version upgrades supported

Phase III

Full Lifecycle

App lifecycle, storage lifecycle (backup, failure recovery)

Phase IV

Deep Insights

Metrics, alerts, log processing and workload analysis

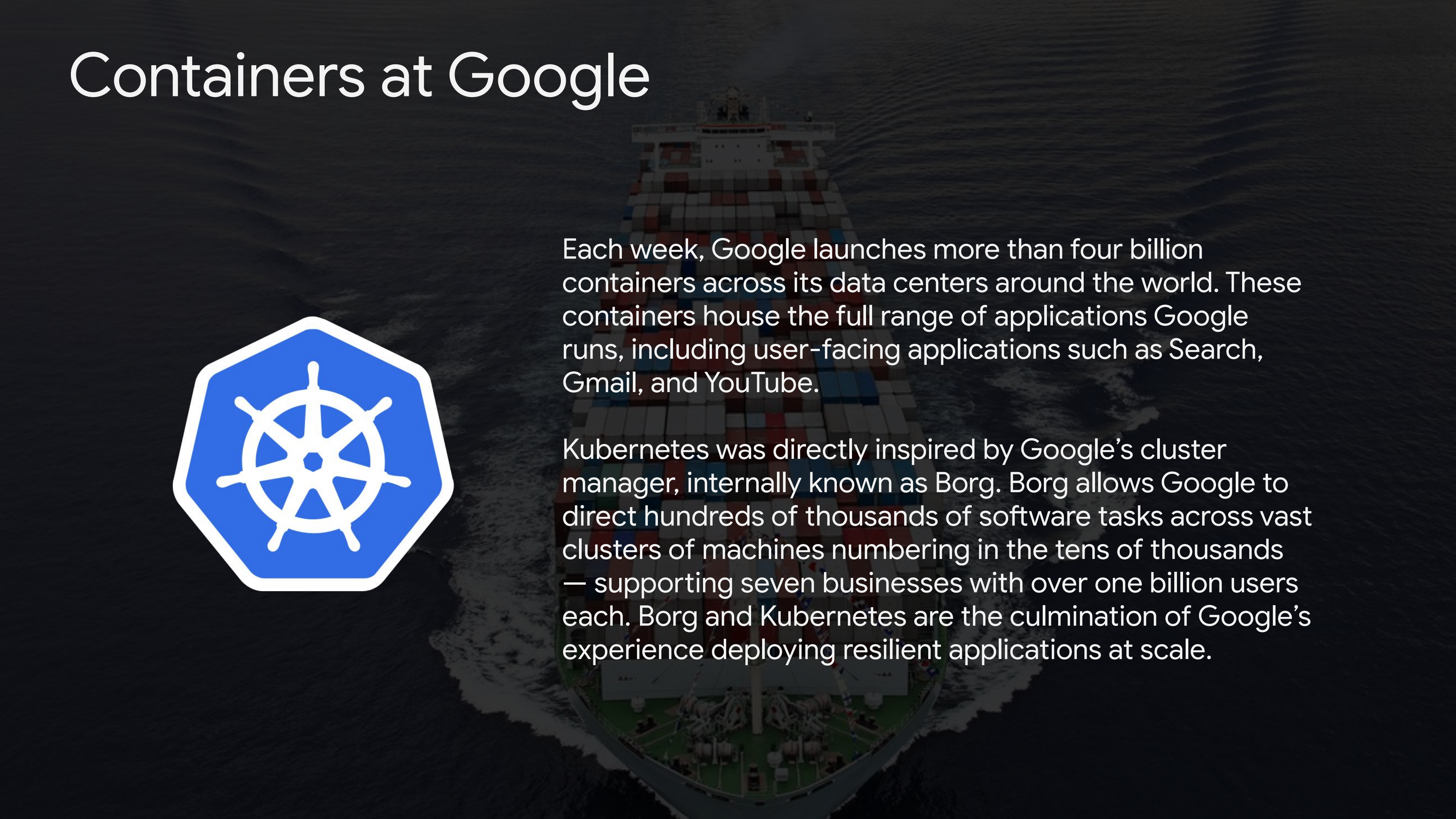
Phase V

Auto Pilot

Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning

On-premise
vs
Public Cloud

Containers at Google



Each week, Google launches more than four billion containers across its data centers around the world. These containers house the full range of applications Google runs, including user-facing applications such as Search, Gmail, and YouTube.

Kubernetes was directly inspired by Google's cluster manager, internally known as Borg. Borg allows Google to direct hundreds of thousands of software tasks across vast clusters of machines numbering in the tens of thousands — supporting seven businesses with over one billion users each. Borg and Kubernetes are the culmination of Google's experience deploying resilient applications at scale.

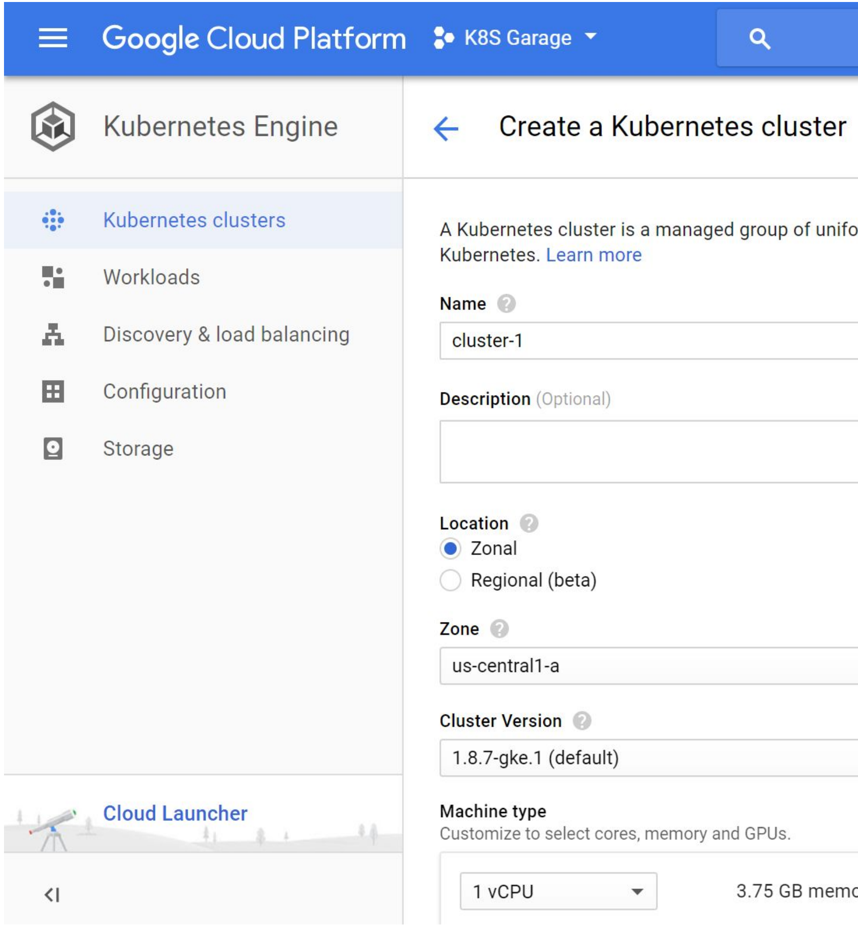
Kubernetes the Easy Way

Start a cluster with one-click

View your clusters and workloads in a single pane of glass

Google keeps your cluster up and running

Google Cloud



Google Cloud Platform
K8S Garage

Kubernetes Engine

Create a Kubernetes cluster

A Kubernetes cluster is a managed group of unifo
Kubernetes. Learn more

Name
cluster-1

Description (Optional)

Location
● Zonal
○ Regional (beta)

Zone
us-central1-a

Cluster Version
1.8.7-gke.1 (default)

Machine type
Customize to select cores, memory and GPUs.

1 vCPU 3.75 GB memo

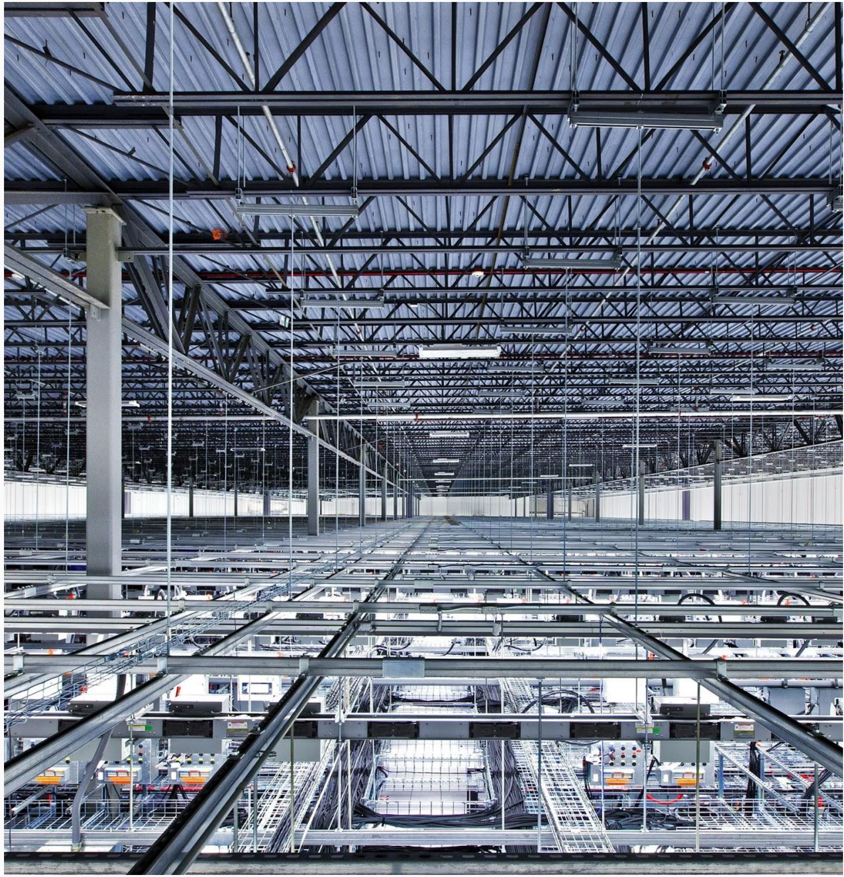
Public cloud: pros and cons

Pros

- ★ Flexibility for infrastructure provisionning:
  - ○ setup a 40 nodes Qserv cluster in 0.5 days
  - ○ extend it to 50 nodes in 10 seconds
- ★ Excellent support from Google engineers
- ★ Easy to setup development clusters with few maintenance
- ★ Cool proprietary features

Cons,

- ★ Expensive for production platform
  - ○ 100K in 3 months for LSST
- ★ Easy to get stuck with proprietary features
- ★ Hide Kubernetes internals so may be difficult to setup
- ★ Run slower than bare-metal (~25%)



On-premise: pros and cons

Pros

★ Flexibility on cluster setup

  ○ DIY Kubernetes

  ○ Fine-tune your components (local HDD)

★ Require skilled engineers

★ Ease to guarantee your workload portability

★ Run faster than public cloud

Cons

★ Difficult to retrieve the global cost

★ Require manpower for setup and maintenance

★ Hardware upgrade are cost-effective and slow

★ Difficult to rebuild the cluster from scratch

© PHOTOTHEQUE IN2P3 / CNRS

Thanks!

Contact:

Fabrice JAMMES
Formation et conseil Kubernetes
https://k8s-school.fr

fabrice.jammes@k8s-school.fr