Cats vs. Dogs

a Hybrid Cloud Storage Story

Yuval Lifshitz
Principal Software Engineer,
Red Hat

OVERVIEW

What is Ceph?

RADOS Gateway and Object Storage

PubSub in Ceph

Knative, Kubernetes and Lambda

Live Demo

WHAT IS CEPH?

The buzzwords

"Software defined storage"

"Unified storage system"

"Scalable distributed storage"

"The future of storage"

"The Linux of storage"

The substance

Ceph is free and open source **software**

Runs on commodity hardware

- Commodity servers
- IP networks
- HDDs, SSDs, NVMe, NV-DIMMs, ...

Unified - a single cluster can serve **object**, **block**, and **file** workloads

CEPH IS FREE AND OPEN SOURCE

Free to use (...as in beer)

Free to change (...as in speech)

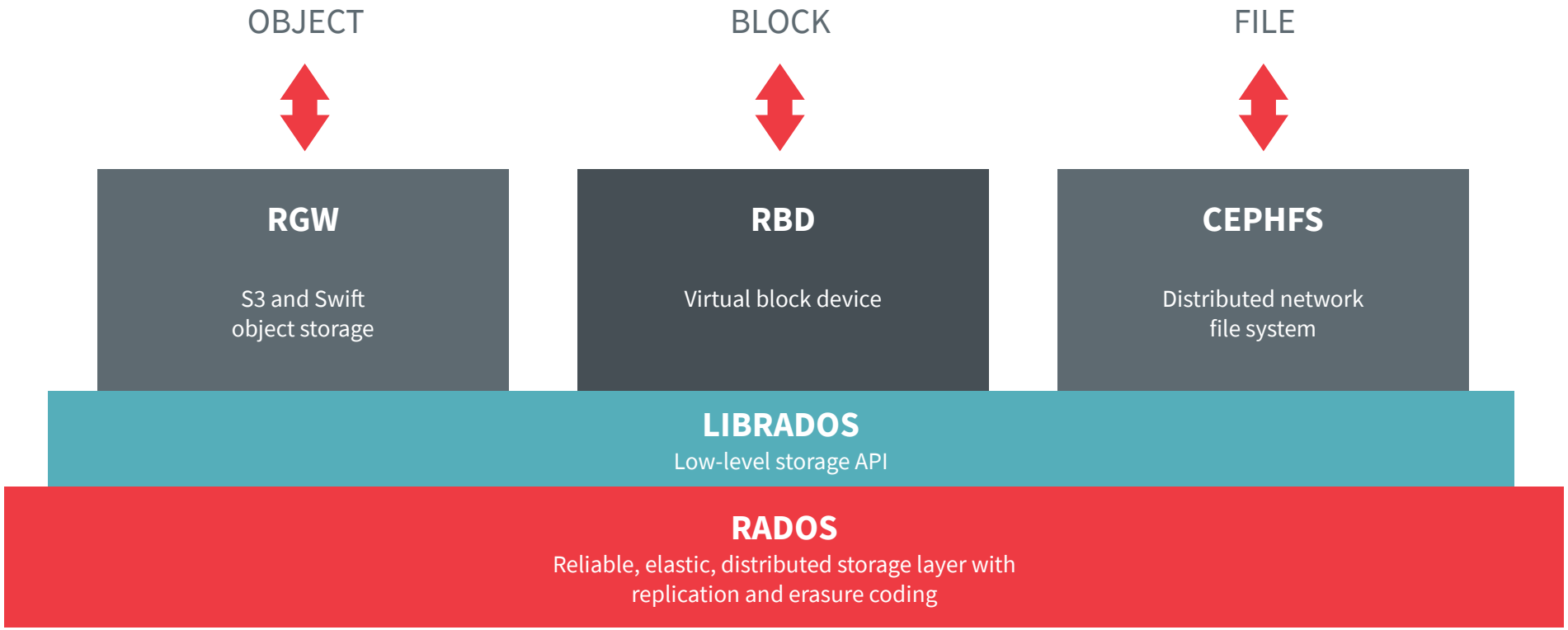
Free from vendor lock-in

Open to the world (not everyone knows C++):

- PubSub
- Cloud
- Dynamic Object Interface (Lua)
- ....

Red Hat

UNIFIED STORAGE SYSTEM



Red Hat

RGW: RADOS Gateway

S3 and Swift-compatible object storage

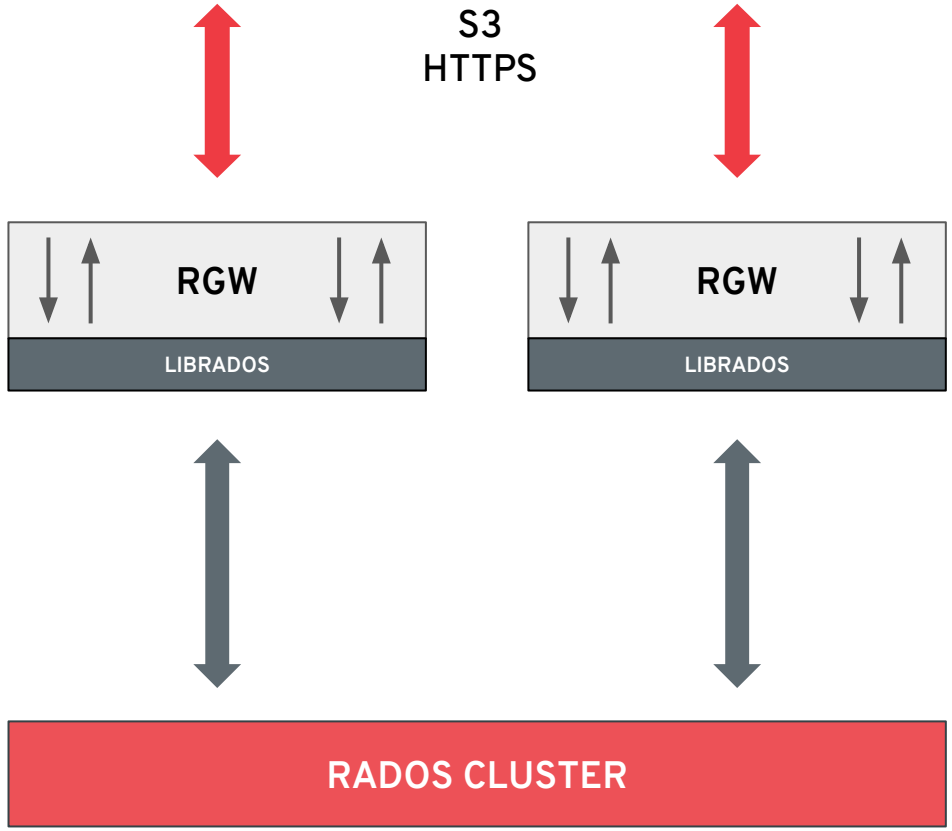
- HTTPS/REST-based API
- Often combined with load balancer to provide storage service to public internet

Users, buckets, objects

- Data and permissions model is based on a superset of S3 and Swift APIs
- ACL-based permissions, enforced by RGW

RGW objects not same as RADOS objects

- S3 objects can be very big: GB to TB
- RGW stripes data across RADOS objects



Red Hat

CEPH PUB-SUB: WHY?

External monitoring systems

Application taking actions on the objects (*as will be shown in the demo*)

External tiering logic

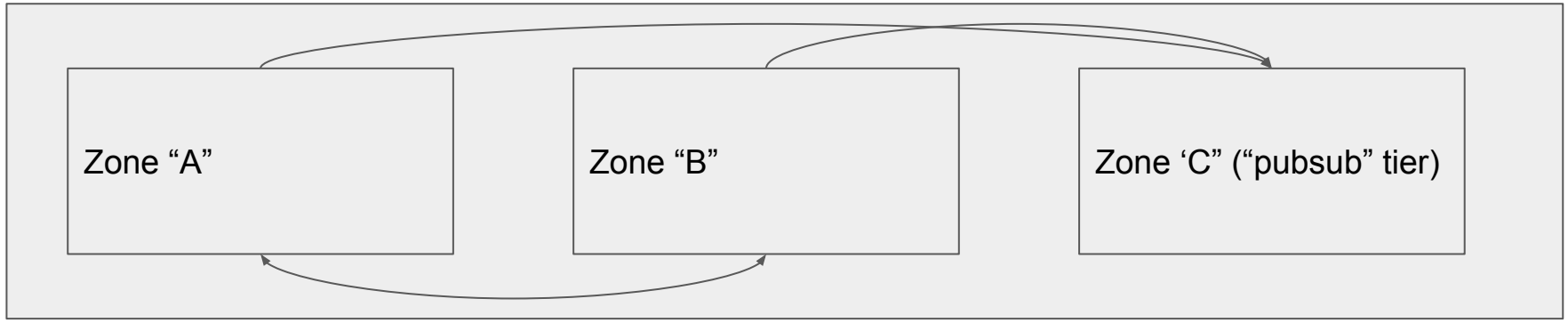
Hacking in general (e.g. migration...)

CEPH PUB-SUB: SELF CONTAINED SOLUTION

Uses the pluggable **sync-module** architecture. An RGW on a specific zone called “pubsub” zone, will be synced with metadata from the zonegroup

Since pubsub is a sync-module it benefits from the multisite reliable messaging and ack semantics

The events are stored in a special bucket in Ceph (we call a **subscription**) and could be **pulled** and removed (acked)



Red Hat

OPENING PUB-SUB: PUSH ENDPOINTS

Has pluggable endpoint architecture for **pushing** the events. Currently we support HTTP and AMQP0.9.1 (RabbitMQ)

We plan on adding: Kafka, AMQP1.0 (ActiveMQ), native serverless support… and more!

CEPH PUB-SUB: ALTERNATIVE ARCHITECTURE

Message busses like: AMQP, Kafka, Knative etc. have their own reliability mechanism, so, as we evolve the feature, we may want to allow for **push only mode** where events are not stored in Ceph, and just pushed into one or more endpoints

In such a case we may add a **simpler deployment** option, where there won't be a need for a dedicated "pubsub" zone, and an event would be considered as "acked" once it is successfully sent to the endpoint

This will allow a richer set of event triggers without overloading the zone sync mechanism. In case of more **ephemeral events** we may chose not to persist them to storage at all (at the risk of losing them if and RGW restart)

CEPH PUB-SUB: DATA MODEL

Topics are named object that could contain the definition of a specific endpoint

Notifications associate topics with a specific bucket, and may also include filter definition on the events (e.g. do we want to see all events or only object creations?). If associated topic has an endpoint, the events will be pushed there - **push mode**

If a topic used by a notification does not have an endpoint, events will be stored in a **subscription** in Ceph and could be fetched from the subscription - **pull mode**

KNATIVE OVERVIEW

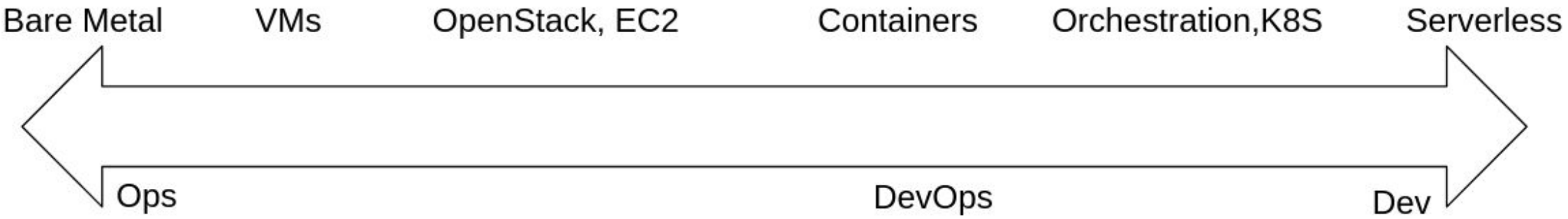
A framework build on top of kubernetes and Istio to: build, trigger, monitor and run **serverless** function

In our case we use 2 components of Knative:

Eventing: reliable event delivery to single or multiple data sinks

Serving: route traffic to functions; triggering and scale up/down function containers

A “WORD” ABOUT SERVERLESS



Simplicity: “I’m just a dev, don’t want to run a server”

Efficiency: “I pay for what I run”

DEMO: A “HYBRID” STORYLINE

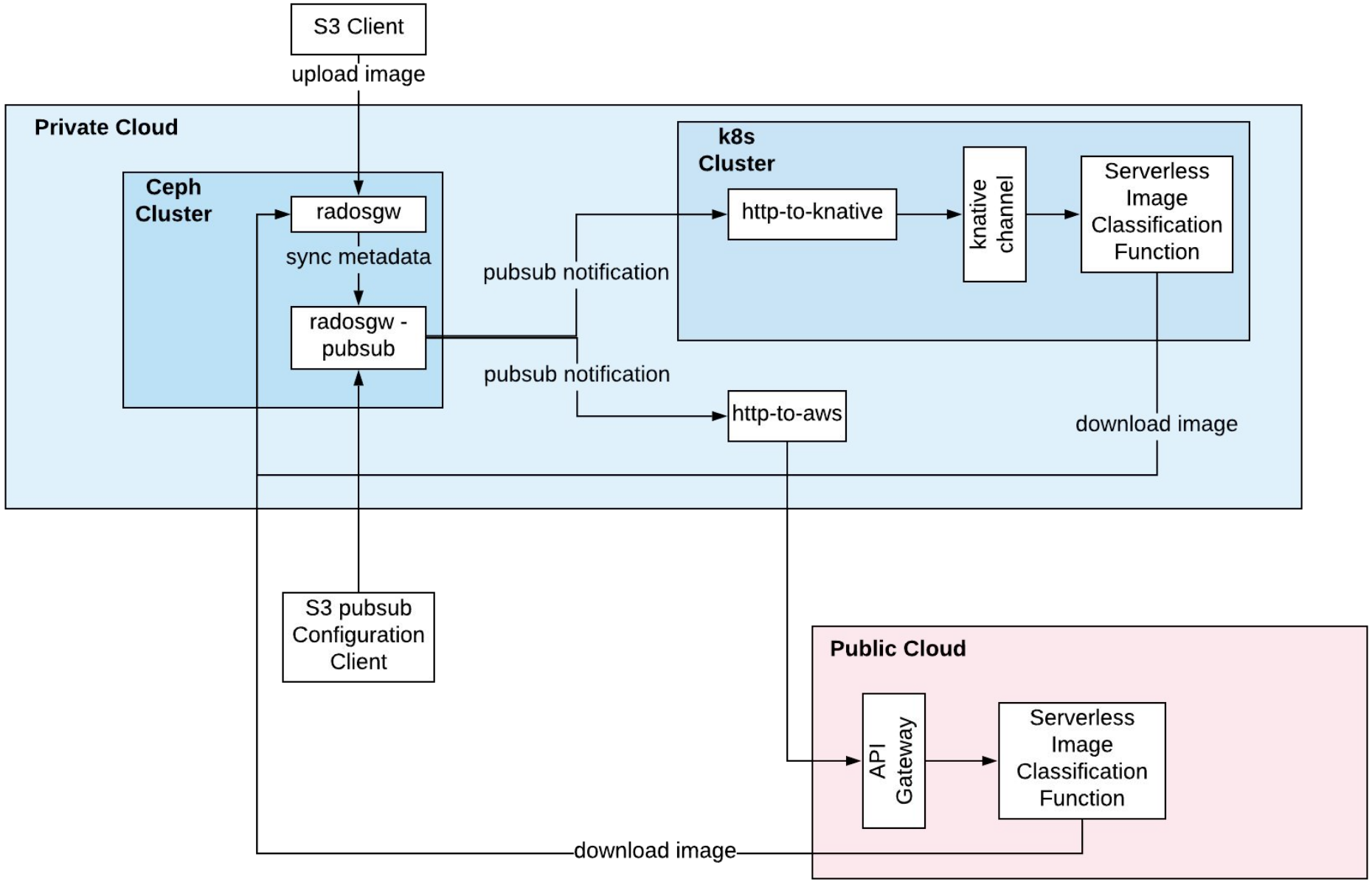
We want to do image classification for every new image uploaded into Ceph. And we have two types of users: **cat** owners and **dog** owners

Our **cat** classification algorithm is homegrown, and specializes in cats, and we want to run it in our on-premise serverless environment (knative on k8s)

For the **dog** images we want to use a publicly available algorithm, in public cloud serverless environment (lambda on AWS)

There could be many other reasons for splitting work between functions on private/public clouds: Cost, Performance, Bandwidth, Privacy …

DEMO: HYBRID CLOUD



Red Hat

DEMO: CEPH SETUP

First we create two **buckets**, one for cats and one for dogs

Using the PubSub mechanism in Ceph we create two **topics** and **notifications**:

One that would track new objects on the **cat** bucket and push notifications to the **http-to-knative** process

Another that track new objects on the **dog** bucket and push notifications to the **http-to-aws** process

DEMO: FORMAT TRANSLATORS

The http-to-knative process run as a pod (ContainerSource in Knative terms) in k8s, and put all incoming PubSubs events it receives into a predefined knative channel

The http-to-aws process convert incoming PubSub events into lambda function invocations

In the future we could have knative-flavored HTTP endpoint and lambda-flavored HTTP endpoint natively in RGW, so these conversion processes are ***not needed anymore***

DEMO: PUBLIC/PRIVATE CLOUD SETUP

Upload to AWS a lambda function that knows how to fetch an object from Ceph (based on information in the incoming trigger event), and classify it

Configure on the k8s cluster a Knative service that will launch a pod that knows how to fetch an object from Ceph (based on information in the incoming trigger event), and classify it

We use Istio on the k8s cluster to allow for data to flow between the cluster and the RGW

DEMO

https://github.com/ceph/rgw-pubsub-api

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat