Cats vs. Dogs

a Hybrid Cloud Storage Story

Huamin Chen
Principal Software Engineer, Red Hat

Yehuda Sadeh-Weinraub
Senior Principal Software Engineer, Red Hat

Yuval Lifshitz
Principal Software Engineer, Red Hat
OVERVIEW

Knative and Kubernetes
Publish-Subscribe in Ceph
Demo
A framework build on top of kubernetes and istio to: build, trigger, monitor and run **serverless** function.

In our case we use 2 features 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 know how to manage a server”

Cost: “I pay for what I run”
CEPH PUBLISH-SUBSCRIBE: RGW COMPONENTS

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 and could be **pulled** and removed (acked)
CEPH PUBLISH-SUBSCRIBE: 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!
Topics are named objects that could contain the definition of a specific endpoint.

Subscriptions 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 of a subscription does not have an endpoint, events will be stored in Ceph and could be fetched from the subscription - pull mode.

Same topic and endpoint could be used by multiple subscriptions. Multiple subscriptions could be defined on the same bucket for different filters and endpoints.
We want to 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 …
First we create two buckets, one for cats and one for dogs

Using the Publish-Subscribe mechanism in Ceph we create two topics and subscriptions:

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 Publish-Subscribe events it receives into a predefined knative channel.

The http-to-aws process convert incoming Publish-Subscribe 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.
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