Data Management in a Cloud-Agnostic World

Vianney Rancurel
Scality
Agenda

- Current state of data
- Zenko cloud-agnostic data management platform and use cases
- Zenko new workflow engine and use cases
Current State of Data
Current State of Data

- Files and objects are in separate storage silos (business and technical).
- Data locations are increasingly dispersed (on-premise and in different clouds).
- Admins must also maintain legacy storage platforms.
- Companies need to integrate and leverage new object storage platforms.
- Creating data workflows—especially hybrid or multi-cloud flows—is difficult.
- Building apps that can leverage data is hard/slow.
Cloud-Agnostic Data Management

- Single interface for data control across legacy, hybrid, and multi-cloud storage architectures
- Global data visibility and search
- Data governance
- Data mobility
- Programmatic and GUI control
- No vendor lock-in
- Faster app development
Workflow Engine

- Can execute operations on data, wherever it is.
- Leverages data.
- Automates tasks (antivirus, auto-tagging, de-identification, anonymization, etc).
- Gets insights from data (machine learning, AI, etc).
- Instead of automating using duct tape (e.g. scripts), we manage it with a data-centric UI.
Architecture of a Cloud-Agnostic Data Management Platform (Zenko)
- Storage location: a bucket (somewhere on-premise or in the cloud) or a mountpoint (on a filesystem).
Zenko (presented SDC 2018)
Zenko Today: Predefined “Workflows”

- Replication of storage locations
- Lifecycle (expiration / transition) per object storage location
- Out-of-band updates (metadata ingestion)
Cloud-Agnostic Platform Use Cases

- **ENTERPRISE**
  - Cloud DR/HA for Data

- **MEDIA & ENTERTAINMENT**
  - Cloud Media Workflows

- **HEALTHCARE**
  - Long-term Archive with active search

- **SERVICE PROVIDERS**
  - Data Management as a Service

- **BANKS, MANUFACTURERS, PHARMA**
  - TV, BROADCASTERS, STUDIOS, POST-PROD, DISTRIBUTION

- **HOSPITALS, MEDICAL GROUPS, UNIVERSITIES**
  - CABLE, TELCO, MOBILE, HOSTERS & CLOUD
Zenko Architecture
Zenko’s New Workflow Engine
Workflow Engine

What if I could manage my data flows with a graphical UI?

- Parallelize some processing
- Serialize other tasks
- Call “on-premise” or remote “cloud” functions
# Workflow Engine Use Cases

## ENTERPRISE
- BANKS, MANUFACTURERS, PHARMA
- Antivirus, Encryption, Compression, compliance

## MEDIA & ENTERTAINMENT
- TV, BROADCASTERS, STUDIOS, POST-PROD, DISTRIBUTION
- Smart tiering, video transcoding

## HEALTHCARE
- HOSPITALS, MEDICAL GROUPS, UNIVERSITIES
- De-identification for AI training, auto-tagging

## SERVICE PROVIDERS
- CABLE, TELCO, MOBILE, HOSTERS & CLOUD
- Antispam, Antivirus
Zenko Workflow Engine Architecture

- **User**
- **Object Storage Protocol**
- **UI / Management Portal**
- **Cloud Server**
  - Temporary Storage
  - Data
  - Metadata
- **Redis**
  - Metrics
- **MongoDB**
  - Buckets, Objects, Users
- **Kafka**
  - Queues
- **Backbeat**
  - Workflow Engine
  - Async Operations
- **Cosmos**
  - K8s Function Engine
  - Data
  - Metadata
- **Clouds 1, 2, 3**
  - Cloud Functions
  - Out-of-band Updates (Metadata)
- **On Premise FS Storage**
- **On Premise Object Storage**
- **Vault**
  - Auth
  - Metadata
- **Kubernetes / Prometheus**
- **Etc**
Kubernetes Function Engines

- We leverage Kubernetes-based function engines (also called serverless engines) e.g., Kubeless, Fission, KNative, as runtime engines for workflow blocks.
- Users create “functions” as Kubernetes pods through existing serverless CLI. Zenko can list them and register them to the Workflow Manager UI.
- The serverless engine can autoscale “functions” based on load (e.g. heavy CPU processing).
- “Functions” can be triggered by:
  - CLI (for tests)
  - HTTP (for synchronous calls), e.g., for custom authorization in Zenko.
  - Kafka topics: for asynchronous calls, the “function” is triggered by a Kafka event in a topic, and the status (success/failure) is stored in another Kafka topic. This is what we leverage in Zenko for asynchronous processing.
- The serverless engine has its own Prometheus sidecars for monitoring the “functions” activity.
- Zenko does the control (retry policy, reporting successes and failures through a UI dashboard).
Workflow Engine: A Real-Life Use Case

An auto-tagging feedback loop:

1) Use TensorFlow locally to auto-tag pictures
2) Check the score
3) Call a cloud AI service
4) Train our model
5) Update tags
Workflow Engine: How it works

A simple workflow with two external functions (serverless engine)
- Writes to two clouds
- Only the metadata (including the data’s location) is transferred in the topics (e.g., Function1 reads/writes data from/to the RING).
Questions?

1,000+ registered Zenko Orbit users

Forum: https://forum.zenko.io
Website: www.zenko.io/blog
Github: https://github.com/scality/zenko
Thank you.