KubeCon | CloudNativeCon
North America 2021

RESILIENCE
REALIZED

Real-Time Data Anonymization: the Serverless Way

Yuval Lifshitz, Github: yuvalif
Huamin Chen, Github: rootfs, Twitter: root_fs

Red Hat Inc.

Agenda

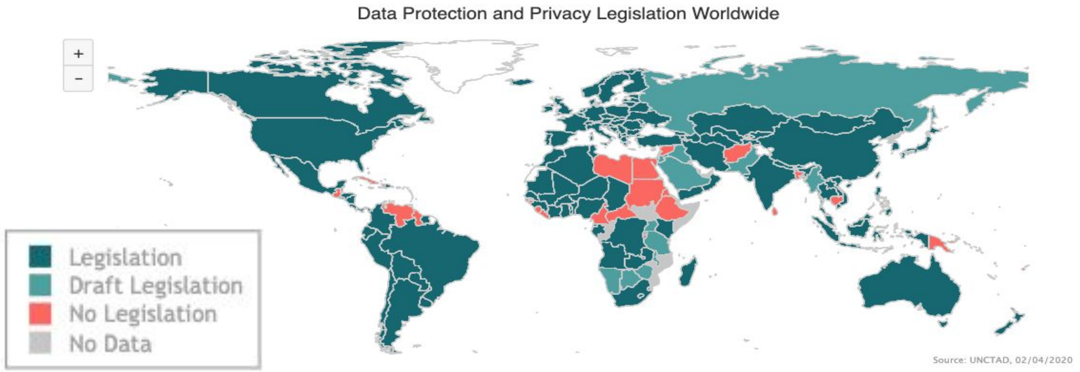
- Data Protection and Privacy Preservation
- Solution Overview
- Rook
- Ceph Rados Gateway (RGW) Bucket Notification
- Message Queue
- Serverless
- MicroShift

Data Protection and Privacy Preservation

Data protection and privacy is an increasingly important issue for global data controllers. Care must be taken to process, exchange, or store sensitive personal identifiable data and honor privacy preferences.

66%	10%	19%	5%
COUNTRIES WITH **LEGISLATION**	COUNTRIES WITH **DRAFT LEGISLATION**	COUNTRIES WITH **NO LEGISLATION**	COUNTRIES WITH **NO DATA**

SELECT A COUNTR | SELECT A REGION | DOWNLOAD FULL DATA


Data Protection and Privacy Legislation Worldwide
+
-
Legislation
Draft Legislation
No Legislation
No Data
Source: UNCTAD, 02/04/2020

Per Recital 26 EU GDPR, data anonymization does not fall into the scope of GDPR:

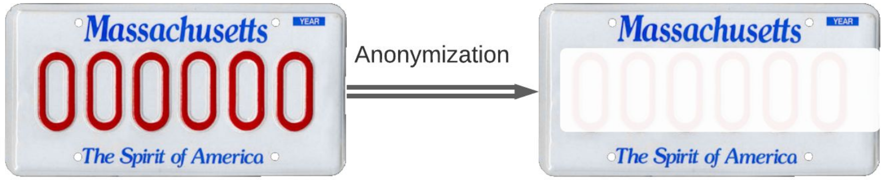
"The principles of data protection should therefore not apply to anonymous information, namely information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable.
This Regulation does not therefore concern the processing of such anonymous information, including for statistical or research purposes." - Recital 26 EU GDPR

Use Case

Under GDPR, data, such as faces and license plates, are considered personal. Data controllers should comply with the requirements and handle with care.

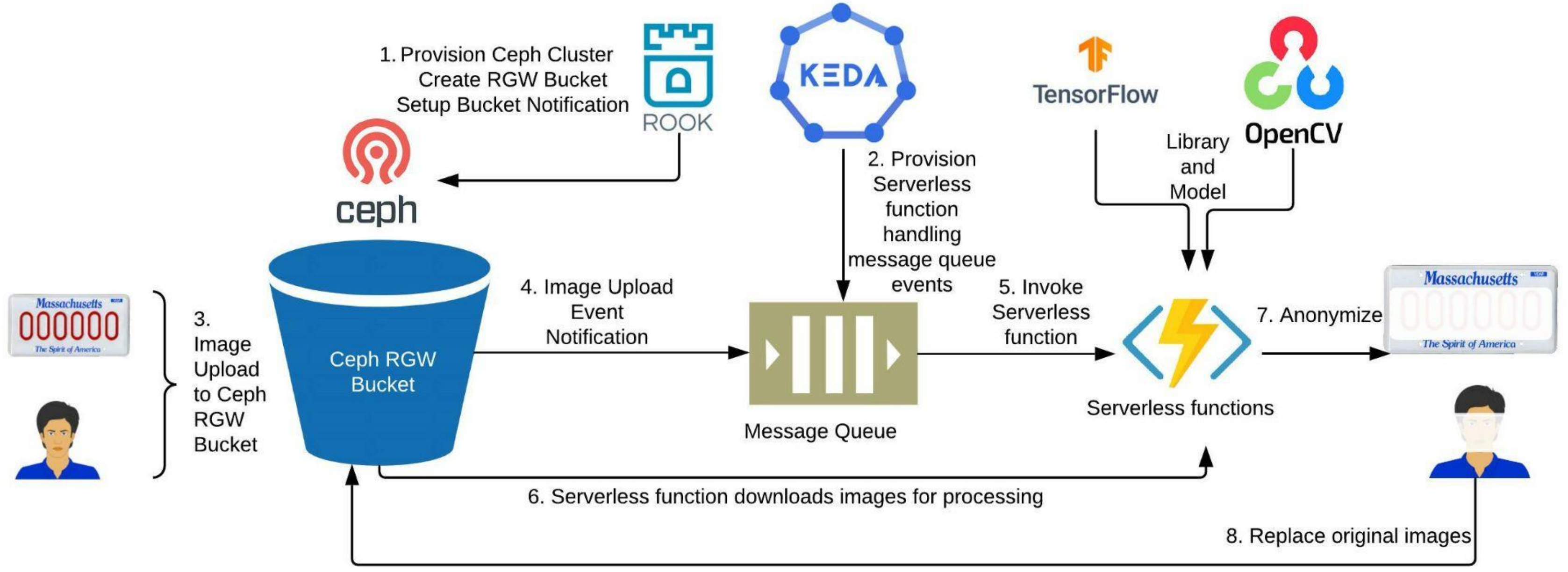
Encryption/decryption, a way of pseudonymization, incurs computational, operational, and sometimes financial overhead.

Our target use case is images uploaded to hosted Object Storage. Our CNCF ecosystem based solution helps storage providers anonymize these images.



Anonymization

Solution Architecture


1. Provision Ceph Cluster
Create RGW Bucket
Setup Bucket Notification
ROOK
ceph
KEDA
2. Provision Serverless function handling message queue events
TensorFlow
OpenCV
Library and Model
Massachusetts
000000
The Spirit of America
3. Image Upload to Ceph RGW Bucket
Ceph RGW Bucket
4. Image Upload Event Notification
Message Queue
5. Invoke Serverless function
Serverless functions
7. Anonymize
6. Serverless function downloads images for processing
8. Replace original images

MicroShift, a lightweight OpenShift/Kubernetes: Start Cloud Native workloads with minimal Control Plane overhead

Ceph RGW Bucket Notifications

Functionality

- Tracking object changes in a bucket
- Provides AWS compatible REST API

Topic - “where to?”

- Aggregates different published events to an endpoint
- Endpoints could be: Kafka, AMQP0.9.1 (RabbitMQ), HTTP and soon also: AWS SNS, AWS Lambda and AMQP1.0 (ActiveMQ)

Notification - “when?”

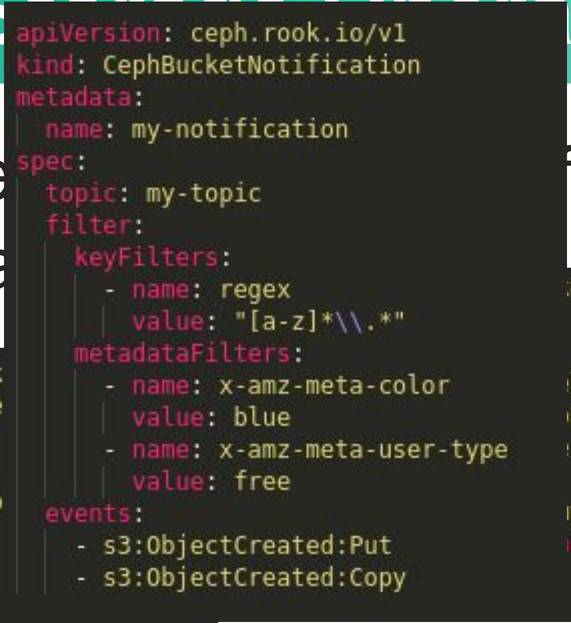
- Changes on bucket are published to a topic
- Filtering based on object name, attributes and tags

Event - “what?”

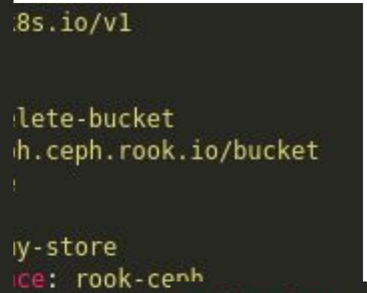
- S3 compatible event schema

Rook - S tor for K8S

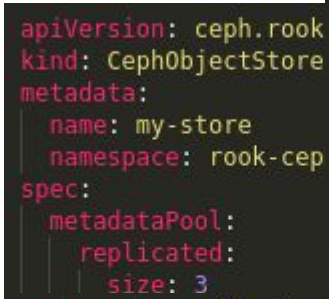
Ceph obje fications is:
“easy as a



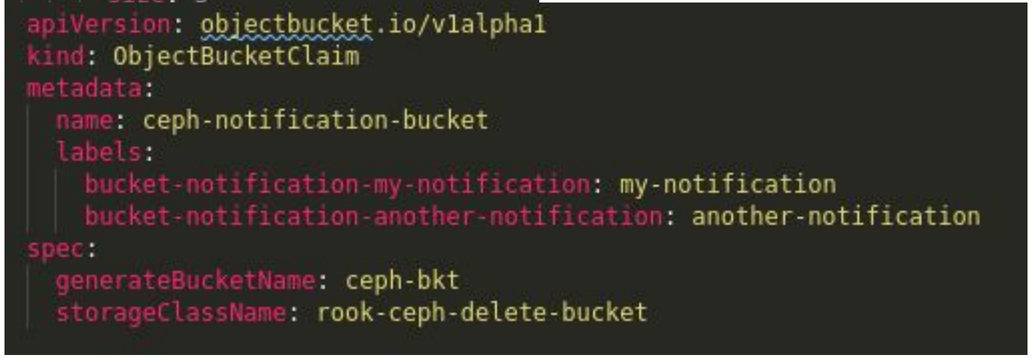
```yaml
apiVersion: ceph.rook.io/v1
kind: CephBucketNotification
metadata:
  name: my-notification
spec:
  topic: my-topic
  filter:
    keyFilters:
      - name: regex
        value: "[a-z]*\\.*"
    metadataFilters:
      - name: x-amz-meta-color
        value: blue
      - name: x-amz-meta-user-type
        value: free
  events:
    - s3:ObjectCreated:Put
    - s3:ObjectCreated:Copy
```



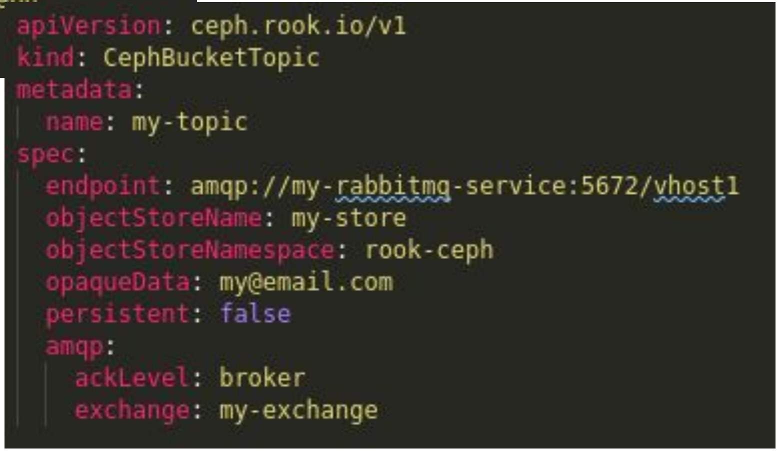
```
8s.io/v1

lete-bucket
h.ceph.rook.io/bucket

y-store
ce: rook-ceph
```



```yaml
apiVersion: ceph.rook
kind: CephObjectStore
metadata:
  name: my-store
  namespace: rook-cep
spec:
  metadataPool:
    replicated:
      size: 3
```

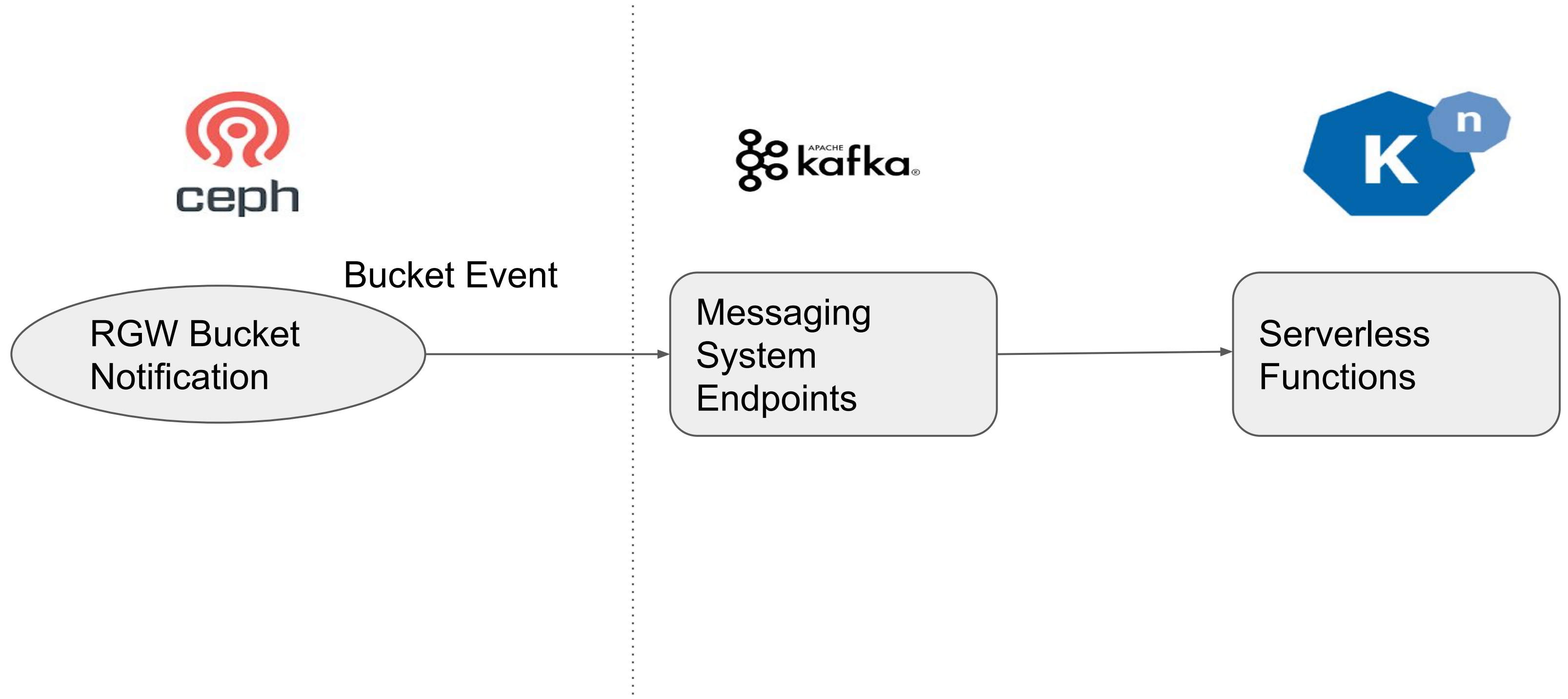


```yaml
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: ceph-notification-bucket
  labels:
    bucket-notification-my-notification: my-notification
    bucket-notification-another-notification: another-notification
spec:
  generateBucketName: ceph-bkt
  storageClassName: rook-ceph-delete-bucket
```



```yaml
apiVersion: ceph.rook.io/v1
kind: CephBucketTopic
metadata:
  name: my-topic
spec:
  endpoint: amqp://my-rabbitmq-service:5672/vhost1
  objectStoreName: my-store
  objectStoreNamespace: rook-ceph
  opaqueData: my@email.com
  persistent: false
  amqp:
    ackLevel: broker
    exchange: my-exchange
```

Push to a Serverless Function



Why Native MQ API in Ceph RGW?

Pushing bucket notifications **directly** into the serverless functions (e.g. to Knative over Kafka or HTTP) works well for **simple** event handling

Pushing bucket notifications to a **message broker** can handle more **complex** cases (e.g. long running executions that may fail midway)

- But may introduce new complexity in the form of a message broker…
- Message broker need to be exposed to the Ceph (firewall, security etc.)

So… Native MQ APIs are the answer!

Push vs. Pull

	Message Push	Message Pull
Message Delivery Mechanism	Notifications sent to an external message broker	Notifications stored in RADOS backed message queue
Serverless Function Programming Model	Based on the external message queue	Function reading from the message queue based on autoscaling trigger
Autoscaling Trigger	Based on Serverless function utilization	Based on the approximated queue size
Producer Reliability	RADOS until acked by external message queue	RADOS
Consumer Reliability	Based on the external message queue	Notifications deleted after consumer acks or timeout expires. Stateless consumer

Ceph RGW Native Message Queue

Based on AWS SQS API

- Implements a subset of it (the parts needed for bucket notifications) with minor modifications
- Allows for standard tools (e.g. boto3) integration
- “at least once” + “visibility timeout” + “retention period”

Using Ceph storage for durability and scalability

- MQ is based on RADOS objects

Co locate data and processing

- Mostly implemented inside the OSD

Expose via REST only the APIs needed for bucket notifications

- In the future we may expose a fully functional REST based message FIFO for different applications

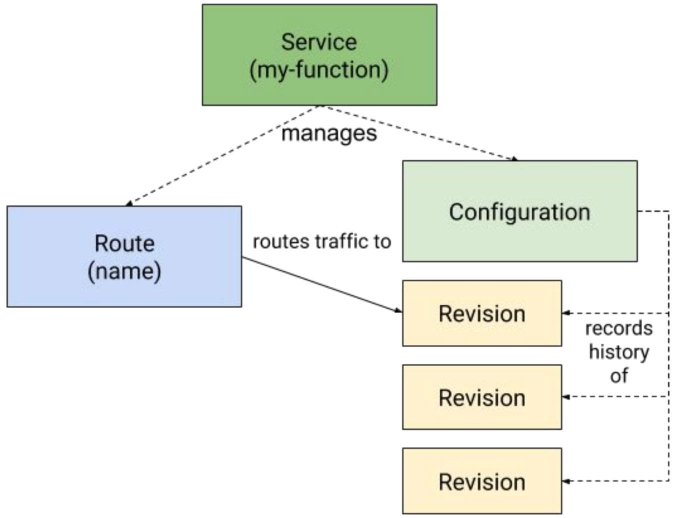
External Message Queues

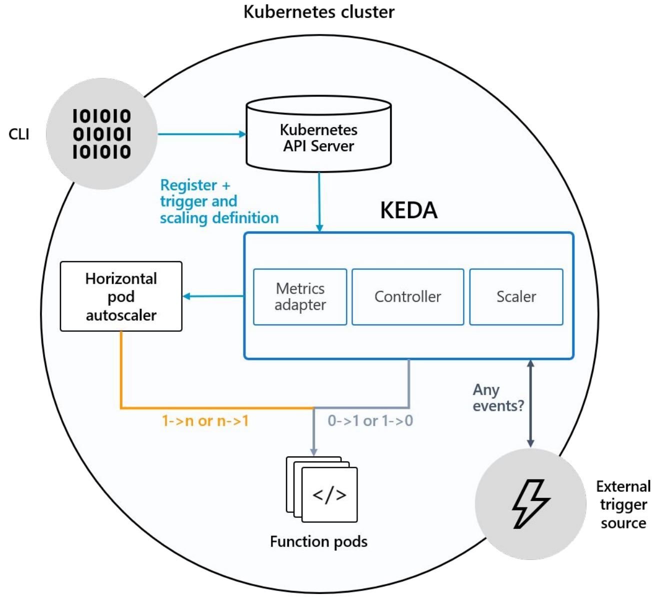
Kafka (scalable cloud solution)

AMQP (lightweight)

Cloud Native Serverless Framework

Knative and KEDA are CNCF Serverless frameworks. Both support scale down to zero through autoscaling. Knative is more network oriented, and KEDA is more service oriented.





Source: https://knative.dev/docs/serving/

Source: https://keda.sh/docs/2.0/

KEDA Use Case

- Serve long lasting Serverless functions
- Need no networking components. This lightweight feature is especially suitable for Edge computing or single purpose applications.
- Serverless functions do not need external endpoint (thus not HTTP triggered as in Knative). External endpoints are not always available due to connectivity or security issues.
- Preemptively autoscale Serverless functions to meet incoming requests and can process data in real-time

Anonymization Serverless Function

The Serverless function downloads the images from Ceph RGW Bucket, detects such personal information as faces and license plates, and blurs (thus anonymizes) the region of interest, before replacing the original images in the Bucket.

There are many object detection mechanisms, such as Haar-Feature Cascade Classifiers and Deep Neural Network models.

- Haar-Feature Cascade Classifier is fast but can only detect one class at a time.
- DNN models are more complicated but can do multiclass detection.

Without loss of generality, this solution uses OpenCV Haar-Feature Cascade classifier for face detection and a pre-trained Tensorflow model for license plate detection (LPD).

MicroShift

- Lightweight implementation of OpenShift/Kubernetes optimized for edge computing and small factor devices with resource constraints or single purpose workloads
- Provides a *minimal and customizable* OpenShift experience
- Single binary that can be deployed as an RPM package or container, running on Linux, MacOS, and Windows, supporting both amd64 and arm64. Also deployable on RISC-V and POWER.