Deduplicating Cloud Functions - Demo 2

Members:

Beliz Kaleli

Vikash Sahu

Paritosh Shirodkar

Asutosh Patra

Mentor:

Shripad Nadgowda

Boston University EC528

BOSTON UNIVERSITY

Some Recap..

How Does Serverless Work?

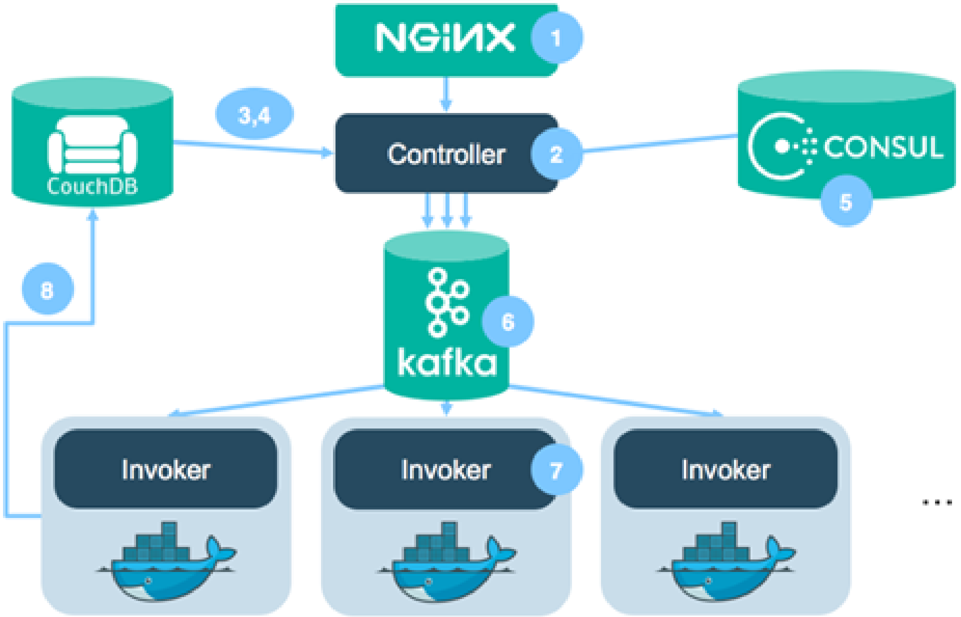


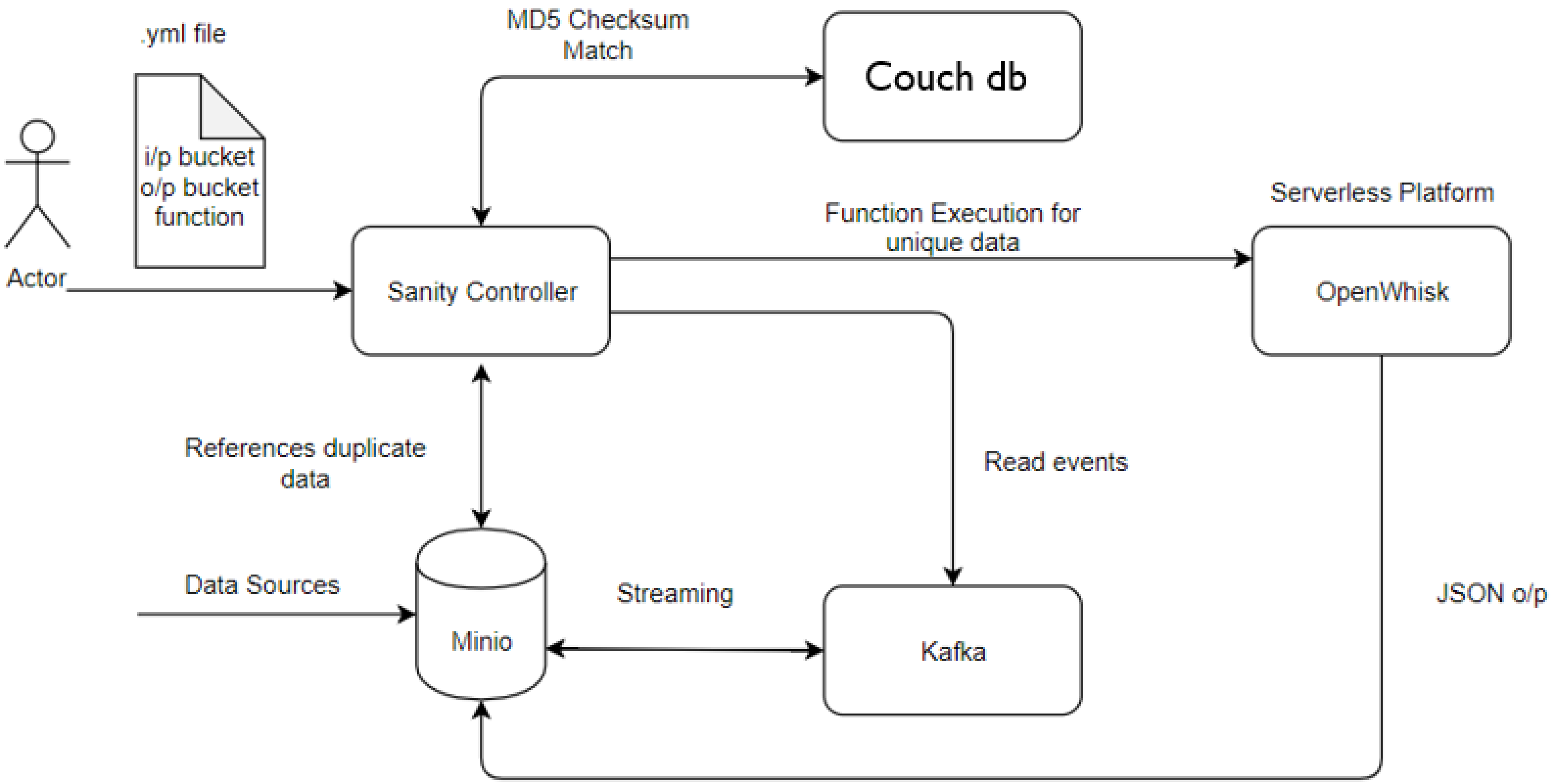
Fig 1. OpenWhisk architecture, Retrieved from:

https://medium.com/openwhisk/uncovering-the-magic-how-serverless-platforms-really-work-3cb127b05f71

Steps:

- SSL Terminator forwards HTTP Req.(user command)
- Check if user exists in OpenWhisk and their privileges
- Consul finds available invokers
- Controller chooses one invoker
- Controller publishes message to Kafka(action+parameters)
- ActivationId is sent to user
- A Docker container spawned, code injected, executed, result obtained, container destroyed
- Result is stored in DB

Architecture Diagram

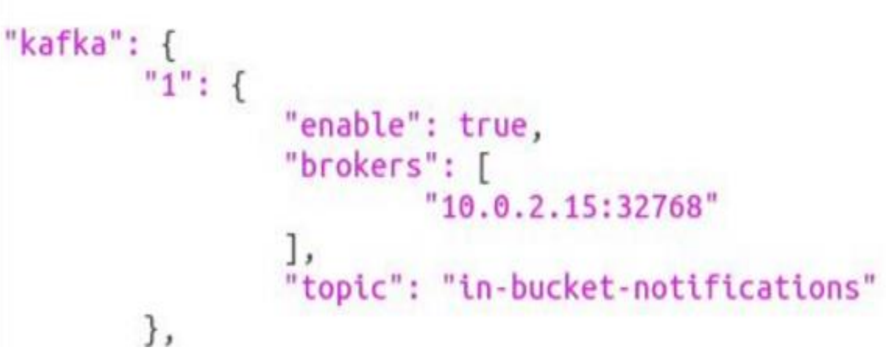


Integration of Components

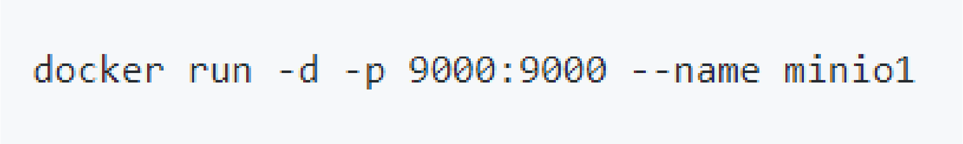
- Minio
- Kafka
- Couchdb
- Openwhisk

Minio

- Client and Server
- Configure server with Kafka
- Run server

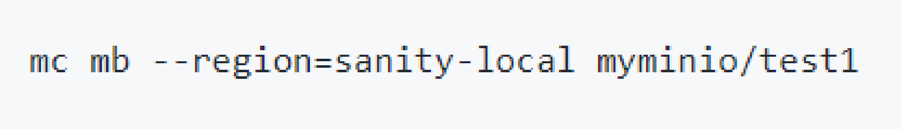


```
"kafka": {
        "1": {
                "enable": true,
                "brokers": [
                        "10.0.2.15:32768"
                ],
                "topic": "in-bucket-notifications"
        },
```

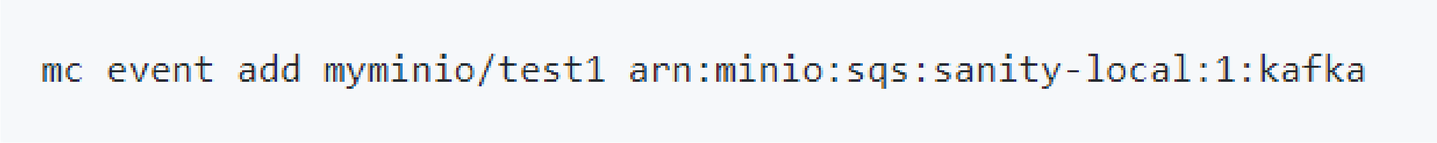


```
docker run -d -p 9000:9000 --name minio1
```

- Client makes a bucket
- and setup event



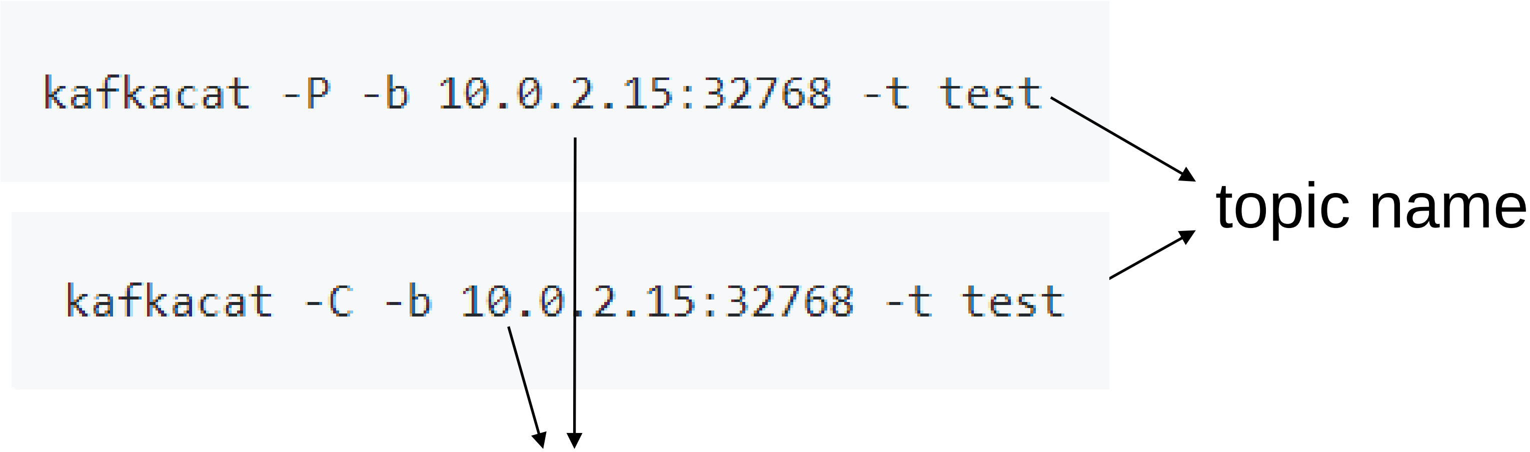
```
mc mb --region=sanity-local myminio/test1
```



```
mc event add myminio/test1 arn:minio:sqs:sanity-local:1:kafka
```

Kafka

- Bucket events are monitored using Kafka
- Command line Consumer and Producer
- Runs in Docker
- Docker -compose:
  - Define and run multi-container **Docker** applications.
  - Run entire app with a single command: $docker-compose up



```
kafkacat -P -b 10.0.2.15:32768 -t test
```

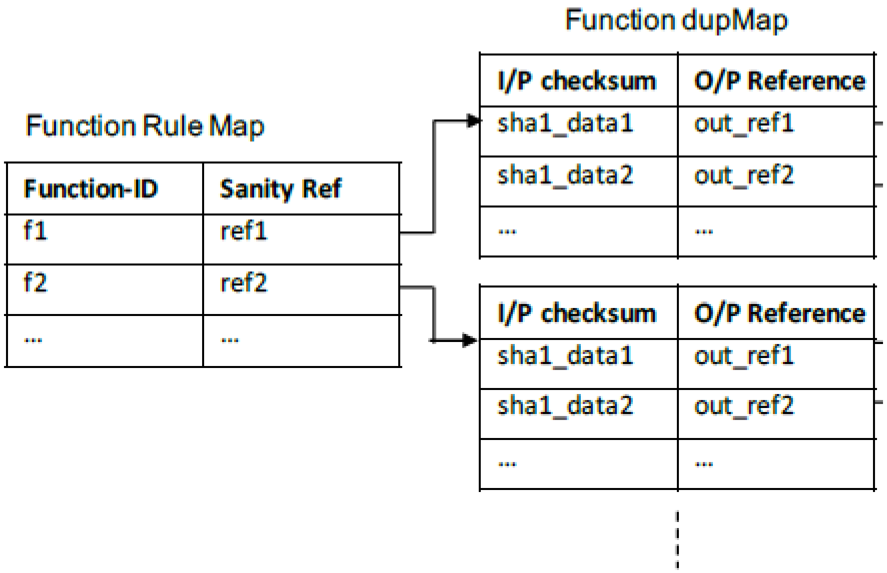
```
kafkacat -C -b 10.0.2.15:32768 -t test
```

topic name

broker name

Couch DB

- Nosql Database
- Easier to grow a cluster
- Built-in managed caching
- Supports Distributed environment for data replication,fault-tolerance, load balancing



Function Rule Map

Function-ID	Sanity Ref
f1	ref1
f2	ref2
...	...

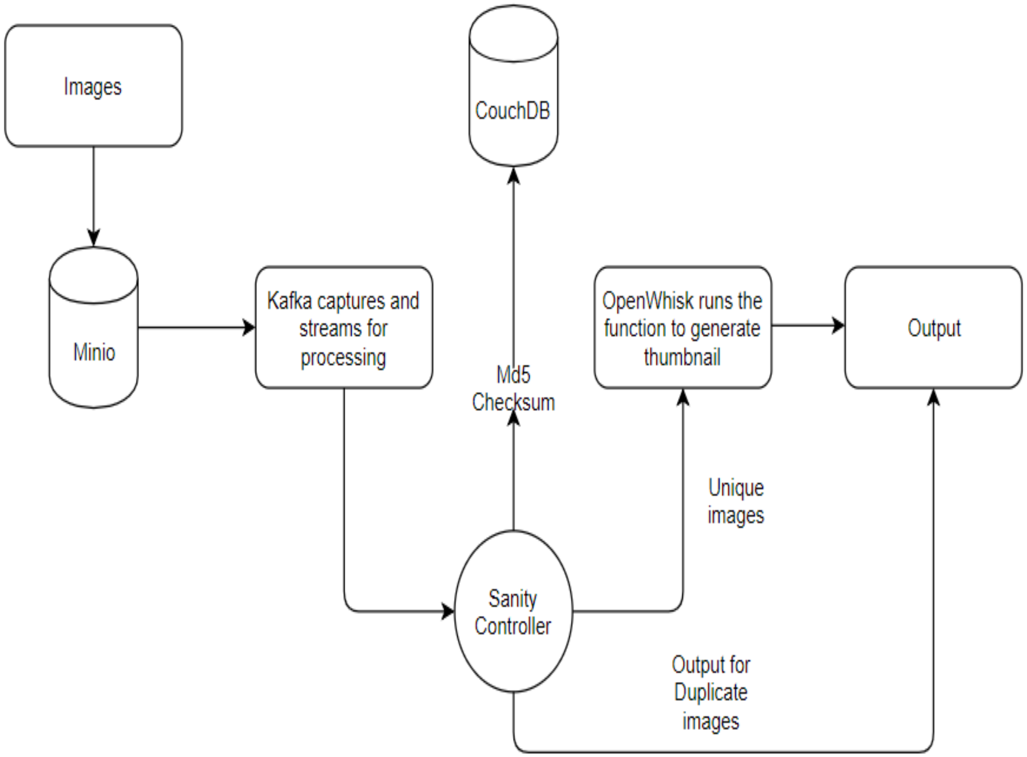
Function dupMap

I/P checksum	O/P Reference
sha1_data1	out_ref1
sha1_data2	out_ref2
...	...

I/P checksum	O/P Reference
sha1_data1	out_ref1
sha1_data2	out_ref2
...	...

Use Case

- Taking function as input from user(finding a thumbnail of an image)
- Checking the MD5 checksum to identify unique data
- Query is done in Couchdb
- Serverless is invoked only if new data comes into play


Images
CouchDB
Minio
Kafka captures and streams for processing
Md5 Checksum
OpenWhisk runs the function to generate thumbnail
Output
Unique images
Sanity Controller
Output for Duplicate images

DEMO

Challenges in Current Sprint

- OpenWhisk Devtools (Docker) version not stable. Lot of functionalities are developed daily
- Difficulty understanding the internal architecture and pipeline of OpenWhisk
- Understanding how containers interact with each other
- Understand how Kafka, Minio and Couchdb works w.r.t. Openwhisk

Next Steps (Sprint - 3)

- Running components in different containers
- Write Sanity Controller and integrate with other elements.
- Implement a function deduplication system
- Design new event management and function invocation framework for Cloud Object Storage

Learning from this Sprint

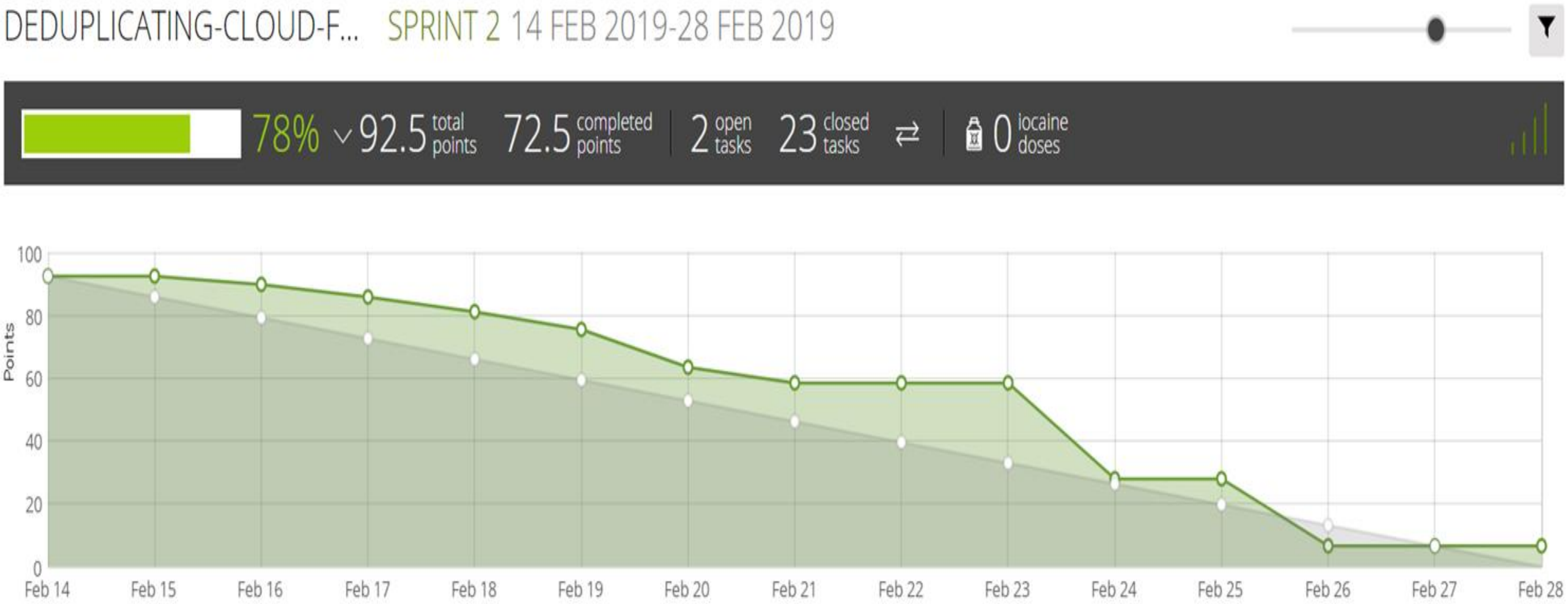
- Got exposed to Open source community
- Had interaction with the members of OpenWhisk
- Pull requests were created and contributed in discussion
- Learnt about Kafka streaming and Minio

Questions from Previous Sprint

- **Why is this being implemented as a layer on top of OpenWhisk rather than inside it?**
  - Due to complex constraints involved, we are trying to build POC on the same architecture. Once this prototype is successful, stretch goal is to integrate it with OpenWhisk framework(difficult to complete in this time frame)

- **How do you know that a function has no side effect, or uses external data (e.g., from a DB)?**
  - Serverless allows only the stateless functions to be executed and not stateful. The stateful functions are not in our scope.

Burndown Chart



DEDUPLICATING-CLOUD-F... SPRINT 2 14 FEB 2019-28 FEB 2019

78% 92.5 total points 72.5 completed points | 2 open tasks 23 closed tasks ⇄ | 0 iocaine doses

THANK YOU