Automate and scale your data pipelines the Cloud Native Way!

OpenShift Commons briefing
March 2020

Guillaume Moutier - gmoutier@redhat.com
Sr. Principal Technical Evangelist

Red Hat

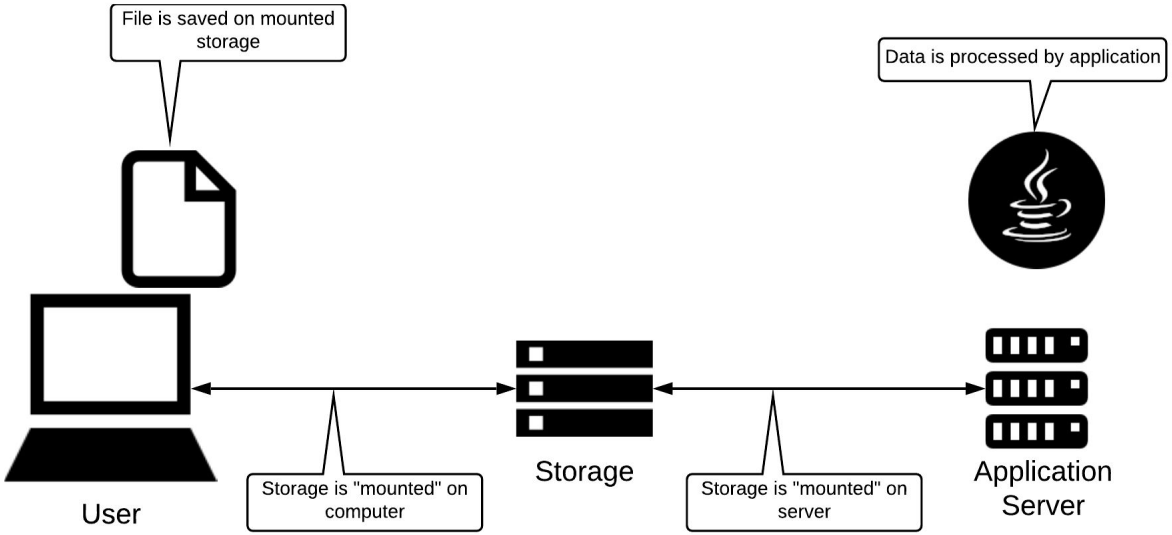
(Opinionated) Characteristics for a cloud native data platform:

1. Agility and Elasticity: as tools, frameworks and datasets evolve constantly and rapidly, you must be able to act accordingly.
2. Cloud standards: avoid vendor lock-in with proprietary tools and formats, and embrace widely recognized open-source protocols and standards.
3. Hybrid cloud architecture: you architecture must run anywhere without any change (some configs may be adapted, but not the architecture itself).
4. Automation: embrace the devops philosophy. Everything must automated and code-based.
5. Separate Compute from Storage: take advantage of the rich computing ecosystem against object storage.

Business outcomes: speed, efficiency, adaptability

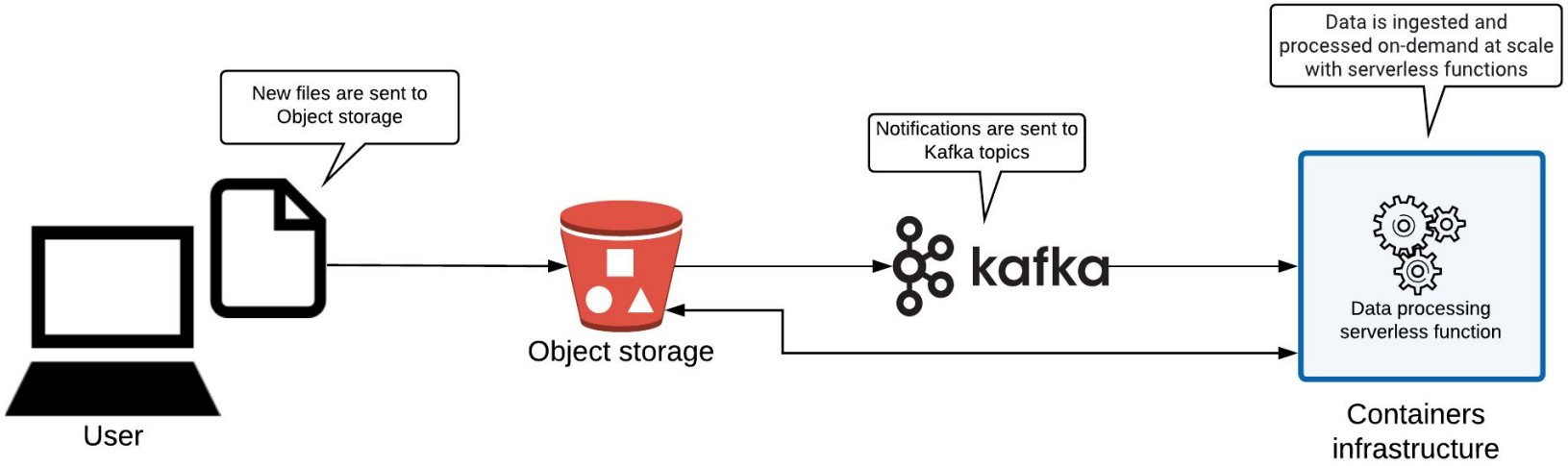
Legacy data pipeline architecture (still standard?):

Everything tightly coupled and not easily scalable



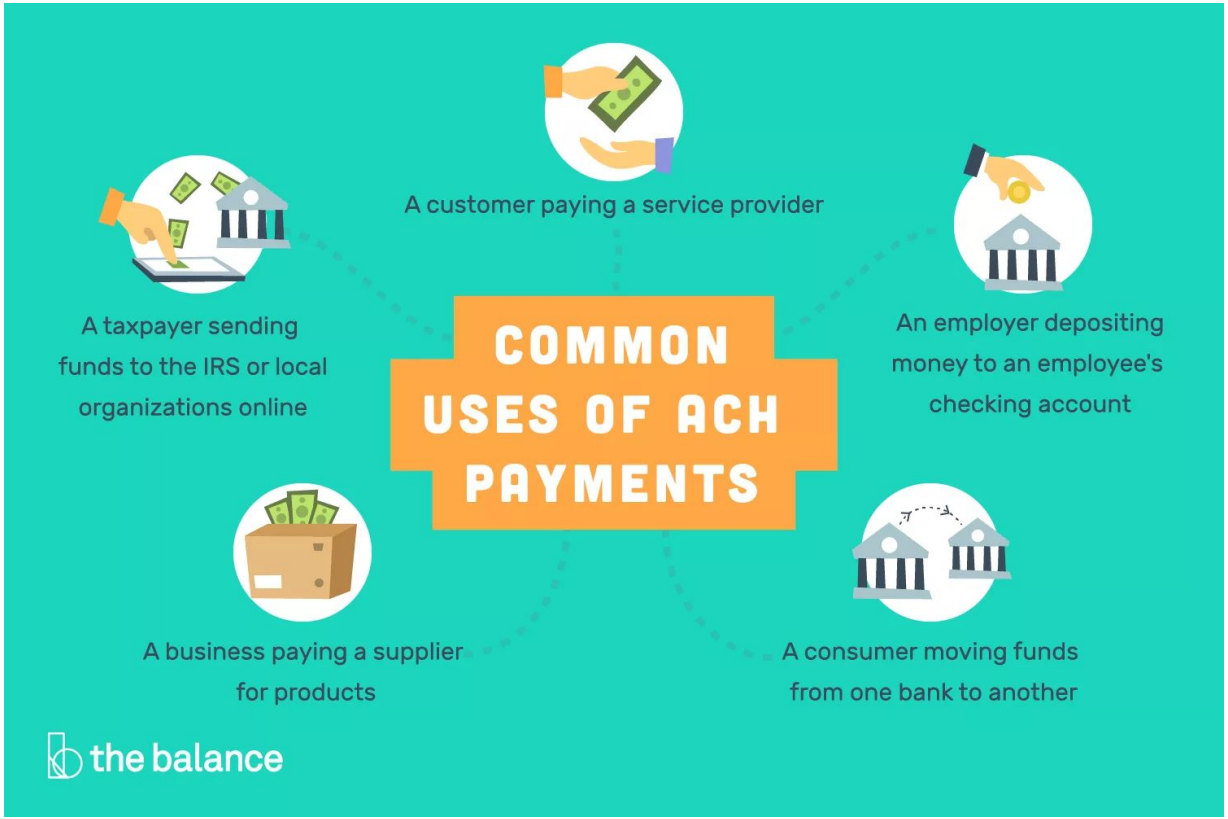
Example of a cloud native architecture pattern:

Everything disconnected and automatically scalable

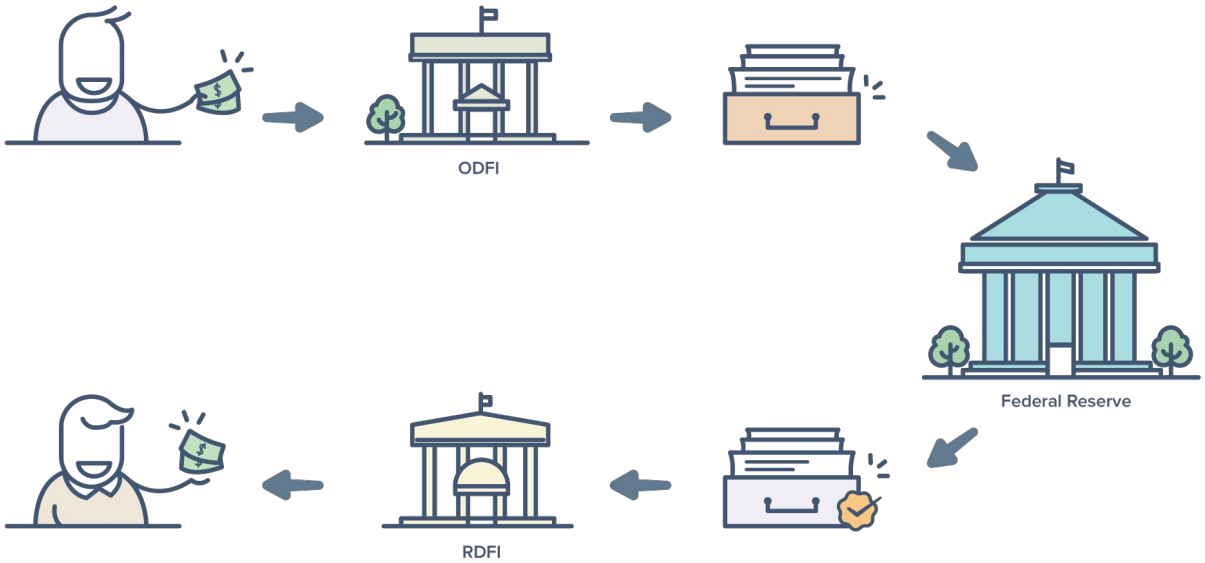


Let’s do it for real!

An example based on ACH payments



The ACH pipeline:



Credit: https://www.chargebee.com/blog/ach-payments-explained/

The ACH format

Record 1 Example:

```
101 062000019199999999905092113 17A094101Regions Bank          XYZ Company          Ref Code
5200XY
622062
622062
622062
622064
622062
622062
622084
622062
622062
622062
622062
622062
622061
820000
900000
999999
999999
999999
```

RECORD TYPE	
1	
1	
1	
1	
1	
1	
1	
1	
1	
1	
1	
1	

Record 5 Example:

```
101 062000019199999999905092113 17A094101Regions Bank          XYZ Company
5200XYZ Com
622062000
622062000
622062000
622064000
622062000
622062000
622084300
622062000
622062000
622062000
622061204
820000001
900000010
999999999
999999999
999999999
```

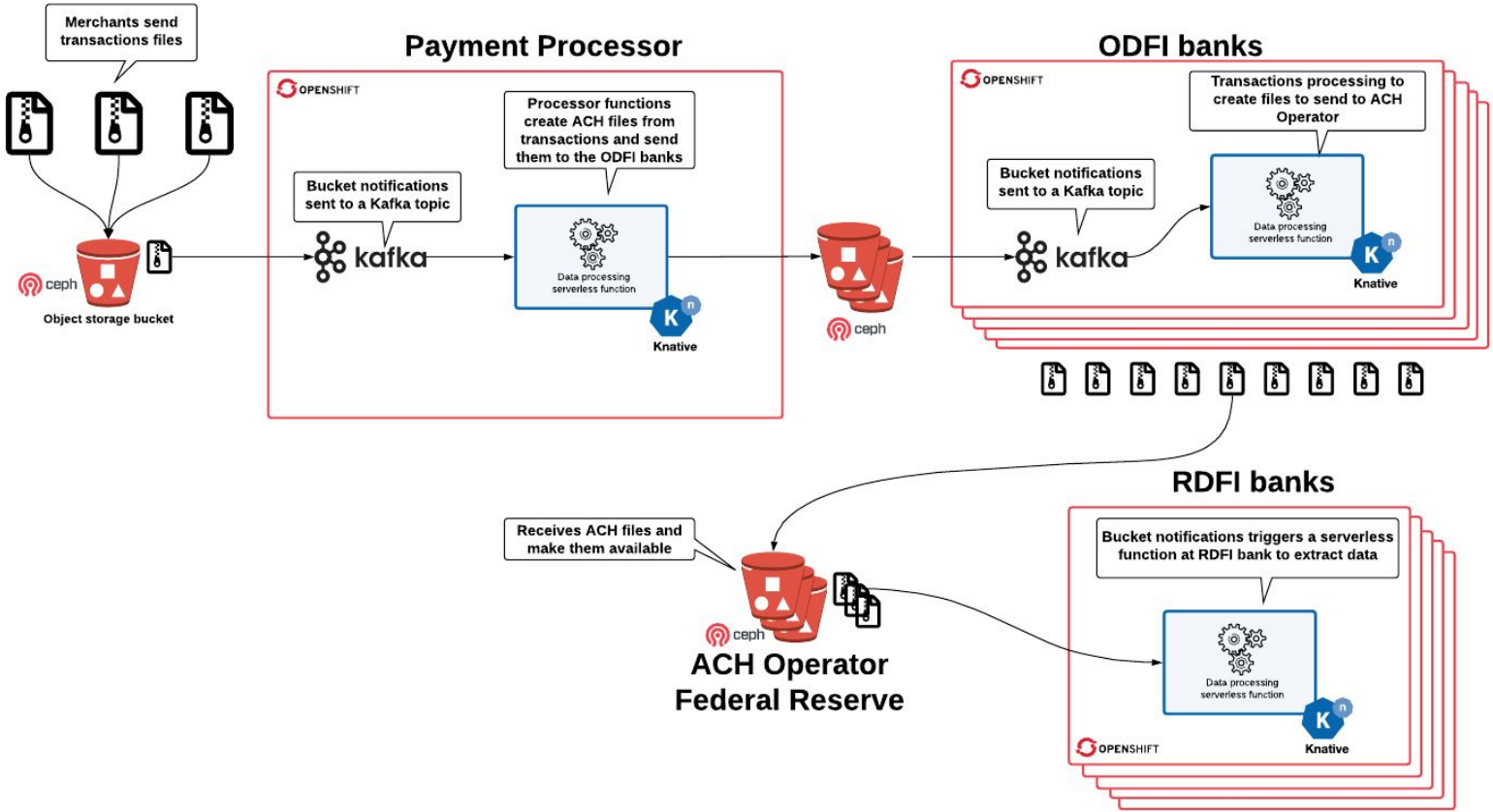
RECORD TYPE	FIELD N
5	Service Cla
5	Company
5	Discretiona
5	Compa Identific
5	Standard Clas
5	Company's Descrip
5	Compan Descriptiv
5	Effective Date
5	(BLANK) Set Date
5	Originator Code
5	Originati Identific
5	Batch Nu

Record 6 Example:

```
101 062000019199999999905092113 17A094101Regions Bank          XYZ Company
5200XYZ Company     Discretionary Data  1999999999PPDPAYROLL    050921050923    1062000010000001
622062000019123456789        00001250251001          Harper, John            0062000010000001
622062000019234567890 1      00001303251002          Brown, Grg              0062000010000002
622062000080334567890 1      00001516221003          Jones, Sara             0062000010000003
622064000017456789012 3      00001528851004          Agnew, Spiro            0062000010000004
622062000019456893012        00002212001005          Clinton, Bill           0062000010000005
622062000019888522231        00002222001006          Reeves, Charles         0062000010000006
622084300593997751137 87     00002121221007          Patterson, Marie        0062000010000007
622062000019753894125        00002331451008          Wade, Marie             0062000010000008
622062000080741325896        00000555001009          Pigue, Philip           0062000010000009
622062000080951324568        00000655001010          Hiscox, George          0062000010000010
622062000019356897501 00     00000932281011          Lafont, William         0062000010000011
622062000080777752235        00000800101012          Abels, Charles          0062000010000012
622061204654111122255 3      00001212151013          Morris, William         0062000010000013
820000013008295056300000000000000000186397719999999 9                       062000010000001
900000100000200000013008295056300000000000000000186397 7
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
```

RECORD TYPE	FIELD NAME	SIZE	POSITION	DESCRIPTION	STANDARD/SAMPLE VALUE
6	Transaction Code	2	02-03	Identifies the account Type at the receiving Bank: 22/32=Deposit Checking/Savings; 27/37= Debit Checking/Savings;	"22"
6	Receiving DFI Identification	8	04-11	Routing number of Receiving Bank	
6	Check Digit	1	12-12	Ninth Digit of Receiving Bank's Routing number	
6	DFI Account Number	17	13-29	ACH recipients account number at receiving bank	"123456789"
6	Amount	10	30-39	Amount of Transaction, including cents; no decimal !--$$$$cc	"0000125025" =$1,250.25
6	Individual Identification Number	15	40-54	Identifies the Receivers ID in batch. May be printed on Stmt.	"1001"
6	Individual Name	22	55-76	Name of Individual receiving ACH (Credit/or Debit)	"John Doe"
6	(BLANK) Discretionary Data	2	77-78	For company's internal use. No Format required- Typically BLANK	" "
6	Addenda Record Indicator	1	79-79	Addenda present = "1", no addenda = "0"	"0"
6	Trace Number	15	80-94	Bank will assign trace number. Company's software will also create a trace that will be "stripped" away by bank and recreated.	"062000010000001"

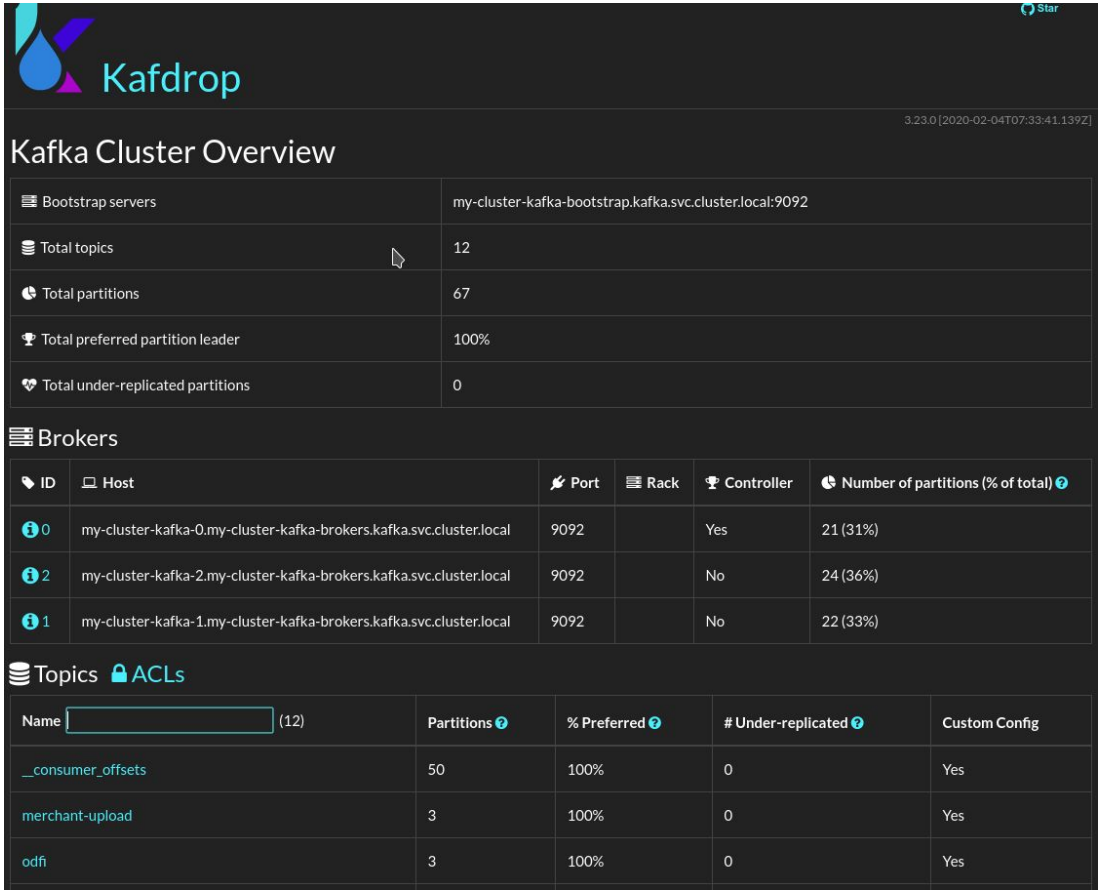
Demo pipeline



Tools used in the demo:

1. RHCS4 object storage: allows for “disconnected” operations. Files will be sent through a simple HTTP upload.
2. RHCS bucket notifications: when a new object is created, an event is sent to a Kafka “topic” (can also be simple HTTP endpoint or AMQ).
3. AMQ Streams (Kafka): messaging bus with high resiliency and ultra-low latency. Will buffer and handle the different notifications.
4. OpenShift Serverless (KNative), serverless workloads manager, with two components:
    a. Eventing: will send a “cloudevent” object to a service when a new message comes into a Kafka topic.
    b. Serving: processing containers will be spawned and scaled as new “cloudevents” are coming in.

Kafka topics



Red Hat

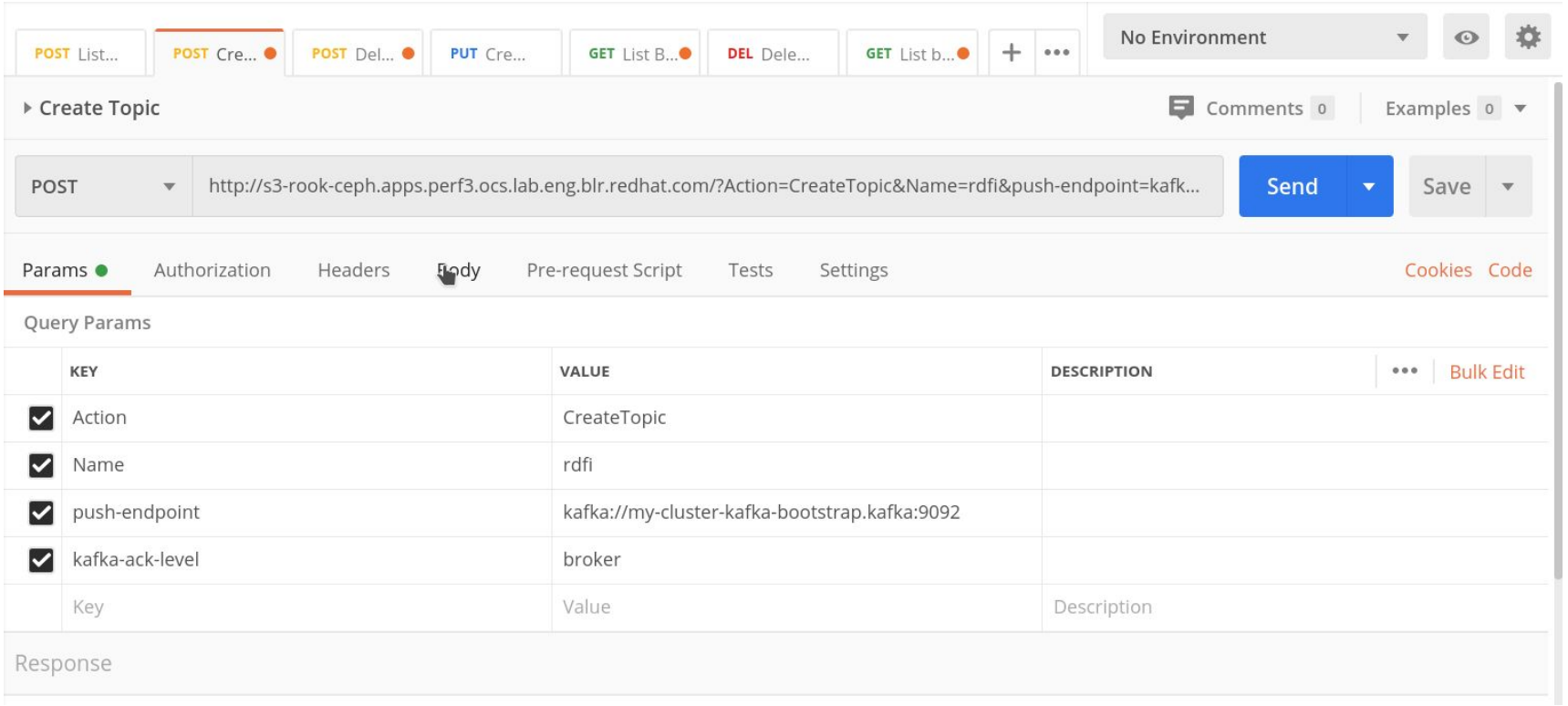
Buckets

In [1]:
```
import boto3

access_key =
secret_key =
service_point = 'http://s3-rook-ceph.apps.perf3.ocs.lab.eng.blr.redhat.com/'

s3client = boto3.client('s3', 'us-east-1', endpoint_url=service_point,
                        aws_access_key_id=access_key,
                        aws_secret_access_key=secret_key,
                        use_ssl=True if 'https' in service_point else False)
```

In [2]:
```
# List buckets
for bucket in s3client.list_buckets()['Buckets']:
    print(bucket['Name'])
```

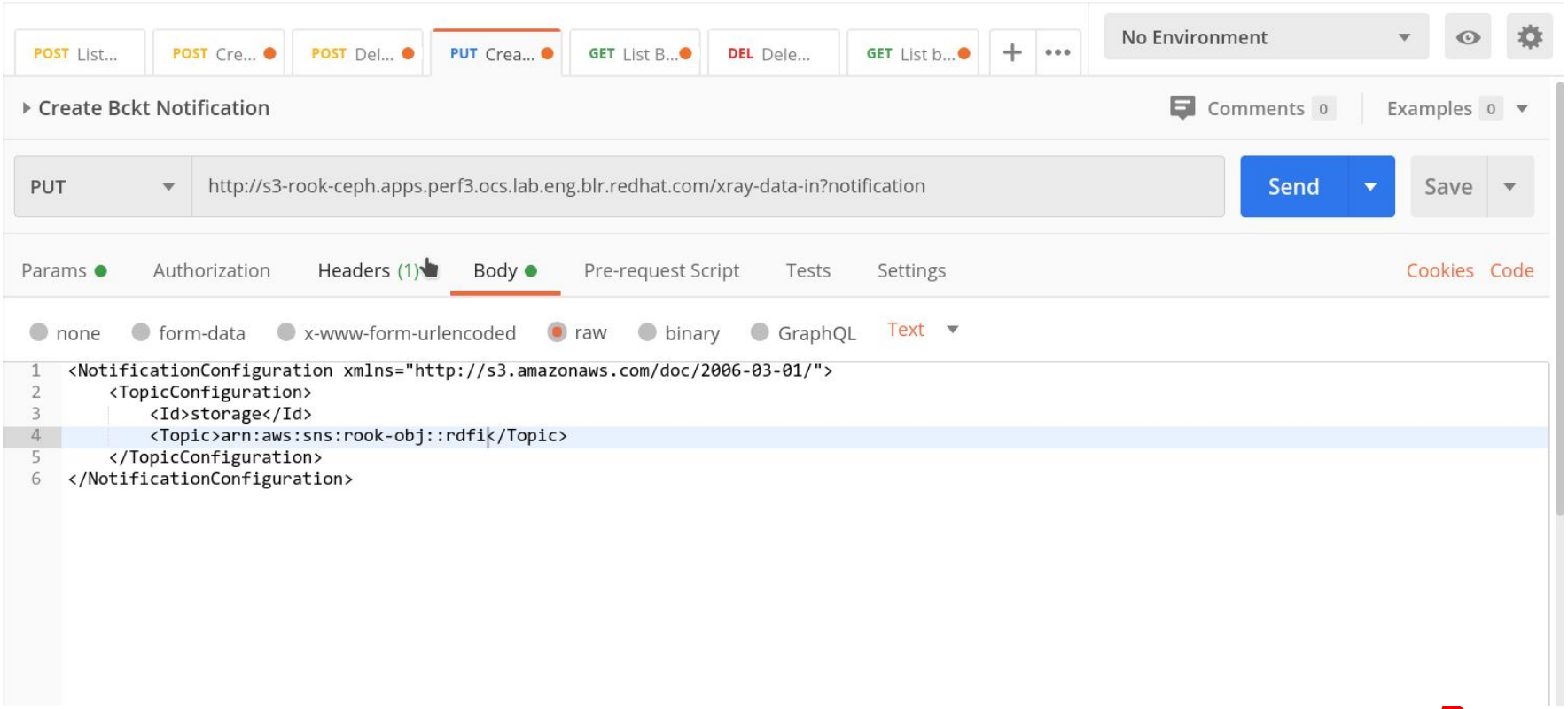
```
ach-merchant-upload
ach-odfi-06200001
ach-odfi-06200002
ach-odfi-06200003
ach-odfi-06200004
ach-odfi-06200005
ach-odfi-06200006
ach-odfi-06200007
ach-rdfi-06200001
ach-rdfi-06200002
ach-rdfi-06200003
ach-rdfi-06200004
ach-rdfi-06200005
ach-rdfi-06200006
ach-rdfi-06200007
```

Bucket notification: topic creation



Red Hat

Bucket notification: notification creation



Transactions Job

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: create-transaction
  namespace: ach
spec:
  backoffLimit: 6
  completions: 60
  parallelism: 5
  template:
    metadata:
      labels:
        job-name: create-transaction
      name: create-transaction
    spec:
      containers:
...
        image: quay.io/guimou/ach-transactions-generator:latest
        name: gen
...
```

Demo time!

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

Repo: https://github.com/guimou/datapipelines/tree/master/demos/ach

Red Hat