All about

Apache Kafka

in Oracle Event Hub Cloud Service

15-Jul-2018

aioug

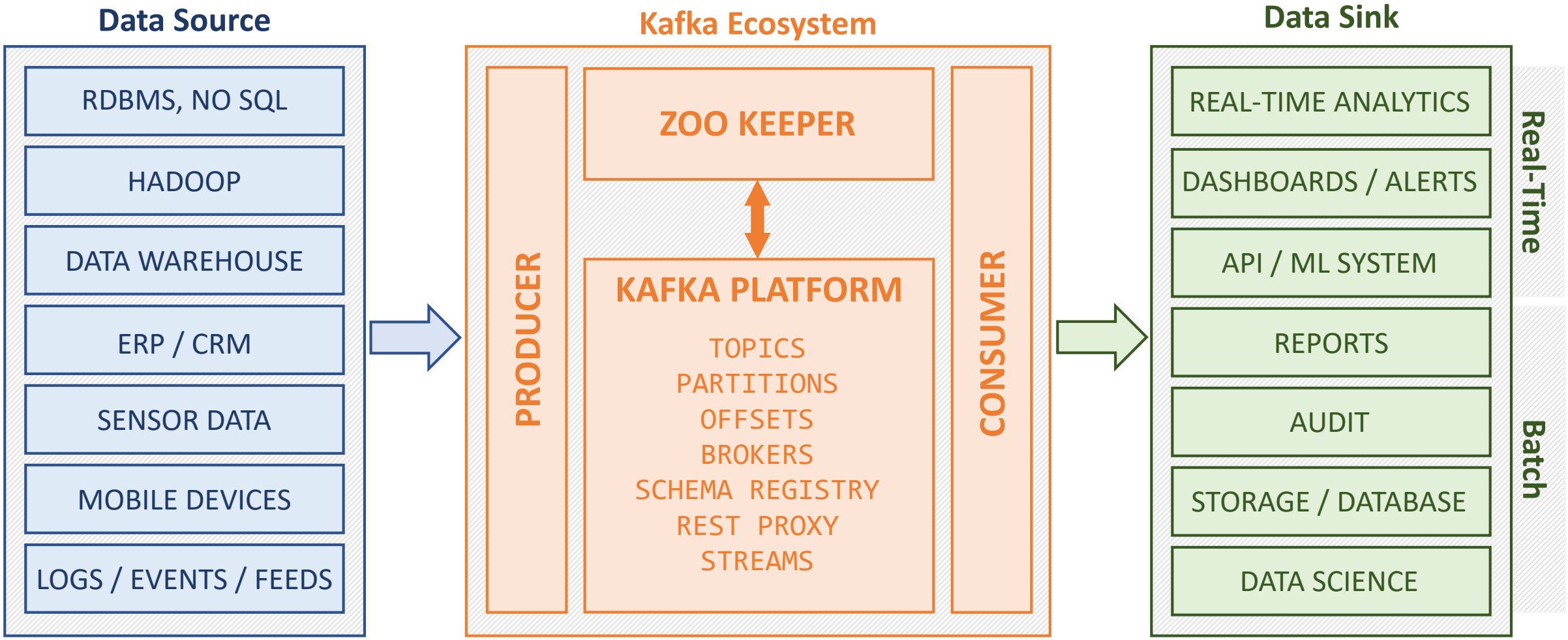
Agenda

- Why need Streaming / Messaging platform?
- Introduction to Apache Kafka
- Architecture of Kafka
- Kafka Core Concepts
- Kafka Components
- Oracle Event Hub Cloud Service
- Kafka in Event Hub Cloud

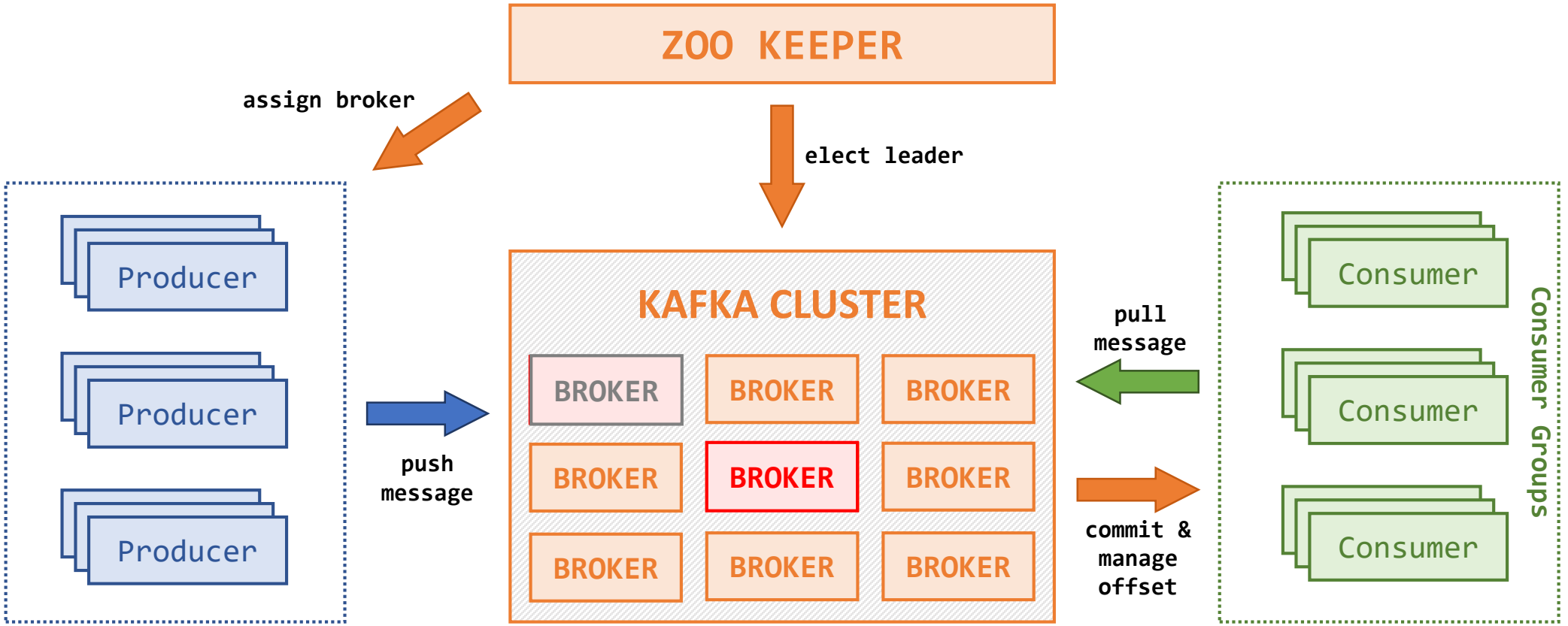
What is Kafka

- Apache Kafka, is an open source technology for developing real-time, fault tolerant, highly scalable and immutable messaging system.
- Developed by LinkedIn and donated to Apache Software Foundation
- Its key strength is its ability to make high volume data available as a real-time stream.
- It can work with OLTP systems, Batch systems like Hadoop, Real time systems that require low-latency access, Stream processing engines etc.,
- It is very well suitable for distributed and disconnected applications.

How Kafka works… ?



Kafka Core Architecture



Real-time use case...

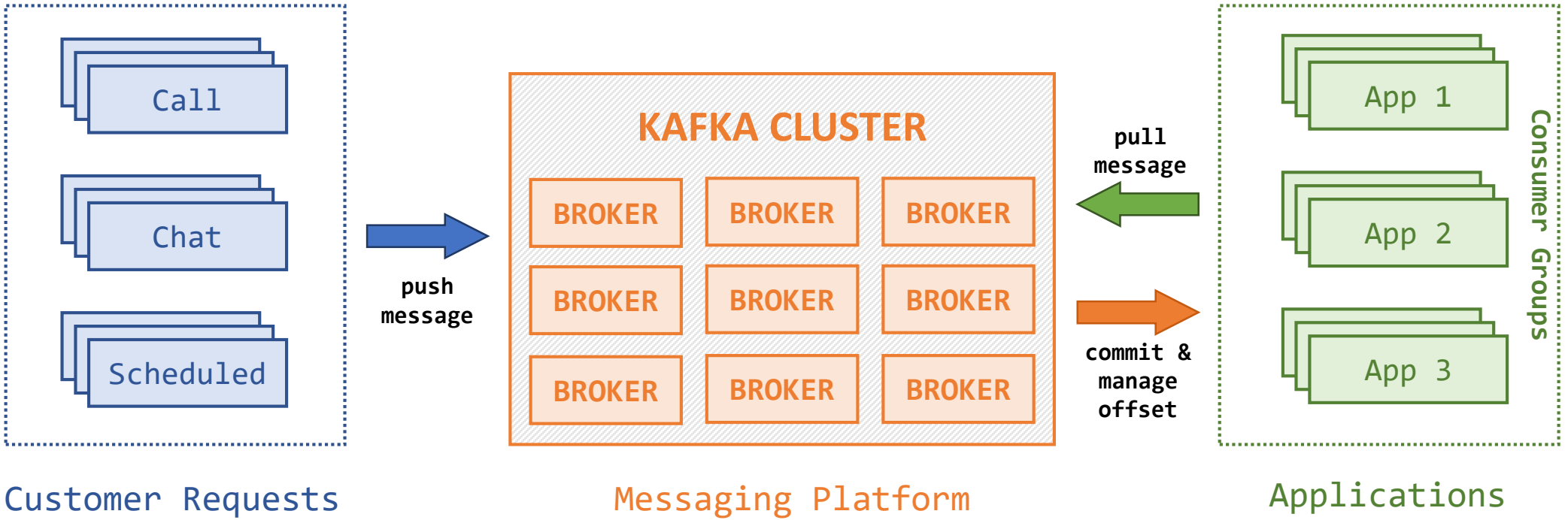
Customer calls for investment advice

service request → portfolio analysis → provide recommendations → investment is made

Create
Exchange
Analyse
Aggregate
Data

(100 calls per hr) (1 request processed per hr)

Kafka Architecture (contd.,)



Kafka Ecosystem

- Topics, Partitions and Offset
- Brokers
- Producers
- Consumers
- Zookeeper

1. Topics, Partitions & Offset

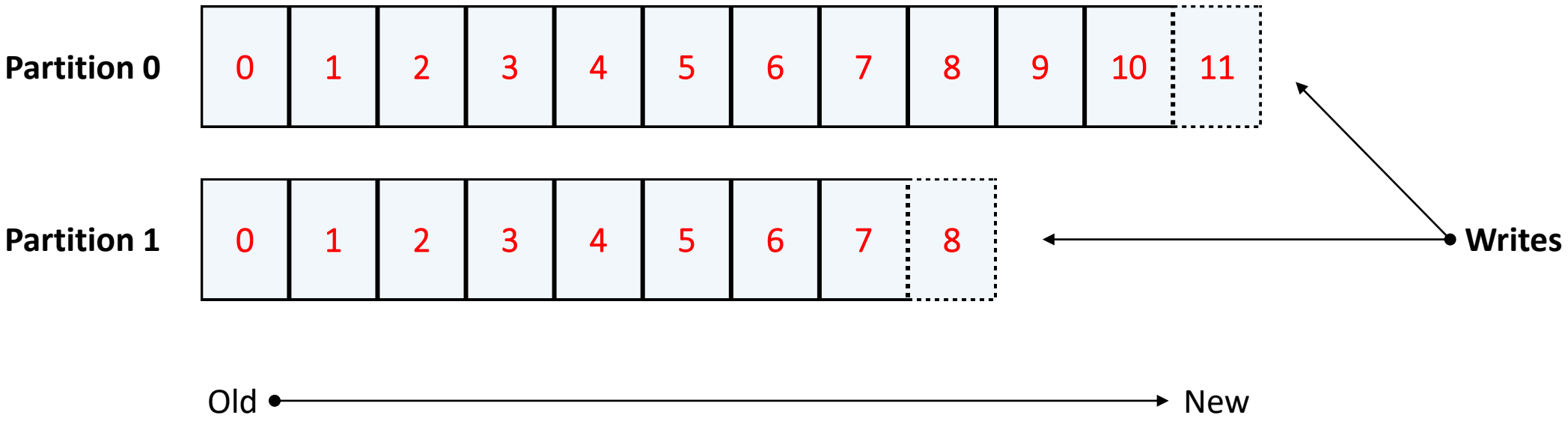
Topics:

- Topic is a particular stream of data
- It is basically a log file that stores data, and can be compared to a Table in the database
- A topic is always identified by its name

Partitions:

- A topic is split into Partitions, and stores message in them
- Partitions are ordered
- The messages in a partition are identified by an incremental id called Offset
- Data written into a partition cannot be changed (immutable)

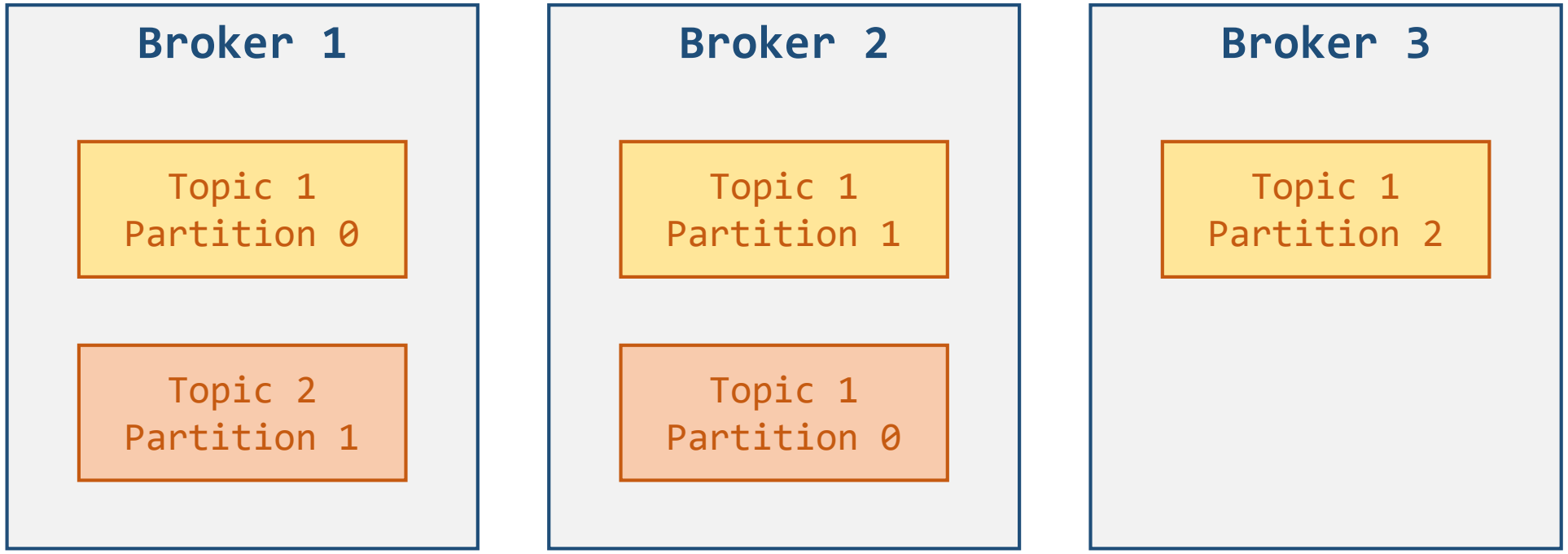
Anatomy of a Topic



2. Brokers (a.k.a Kafka Server)

- A Kafka cluster contains 1 or many Brokers (servers)
- A Broker is identified by its id
- A Broker takes care of certain partitions in a topic
- In a cluster, a connection to any Broker provides access to the entire cluster

Sample Kafka Cluster



3 Brokers
Topic 1 – 3 partitions
Topic 2 – 2 partitions

3. Producers

- Producers are responsible for writing data into a Topic
- The producer will require the Topic Name and Broker to connect to the Kafka cluster, and produce the messages
- In case more than one Broker is there, the Kafka cluster is responsible for Leader election and routing the message to the right broker
- Producer can choose to add a Key to the message. Messages with same key value will always be routed to the same Partition.

4. Consumers

- Consumers are responsible for writing data into a Topic
- The consumer will require the Topic Name and Broker to connect to the Kafka cluster, and produce the messages
- In case more than one Broker is there, the Kafka cluster is responsible for Leader election and pulling the message from the right broker
- Data is read in order for each Partition.
- Consumers read the data in Groups. A consumer in a group reads from only one Partition exclusively.
- So the number of Consumer should be less or equal to the number of Partition.
- Kafka broker stores the offsets read by the Consumer.

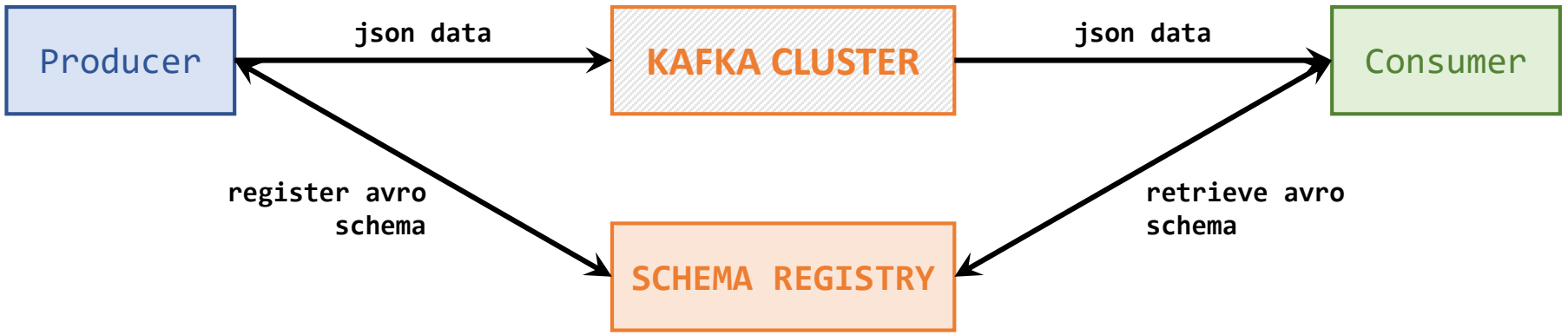
5. Zookeeper

- Zookeeper manages the Kafka brokers
- It helps the Brokers in performing the leader election for partitions
- Zookeeper send notification to Brokers in the event of any new Topic creation, Topic deleted, Broker dies, Brokers comes up, etc.,
- A Zookeeper cluster is called Quorum, and it always contain odd number of servers in it (1,3,5,7…)
- Zookeeper quorum elects its own leader, and the others will be followers

Kafka Toolsets

- Kafka Schema Registry
- Kafka REST Proxy
- Kafka Connect API
- Kafka Streams API

Kafka Schema Registry



Avro Schema

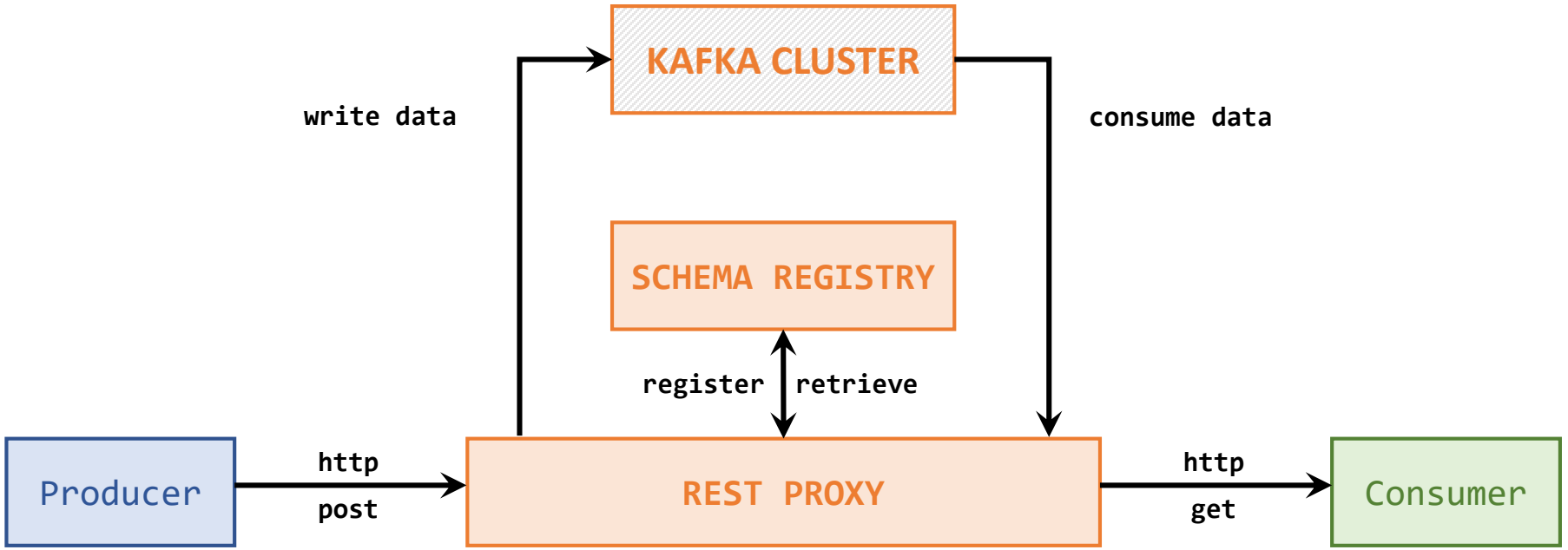
JSON Data:

```
{ "id": "83011050001",
  "student_id": 100,
  "university_id": 908,
  "course_details": {
    "course_id": 100,
    "enroll_date": "2012-02-13 00:00:00",
    "verb": "completed",
    "result_score": 0.9
  }
}
```

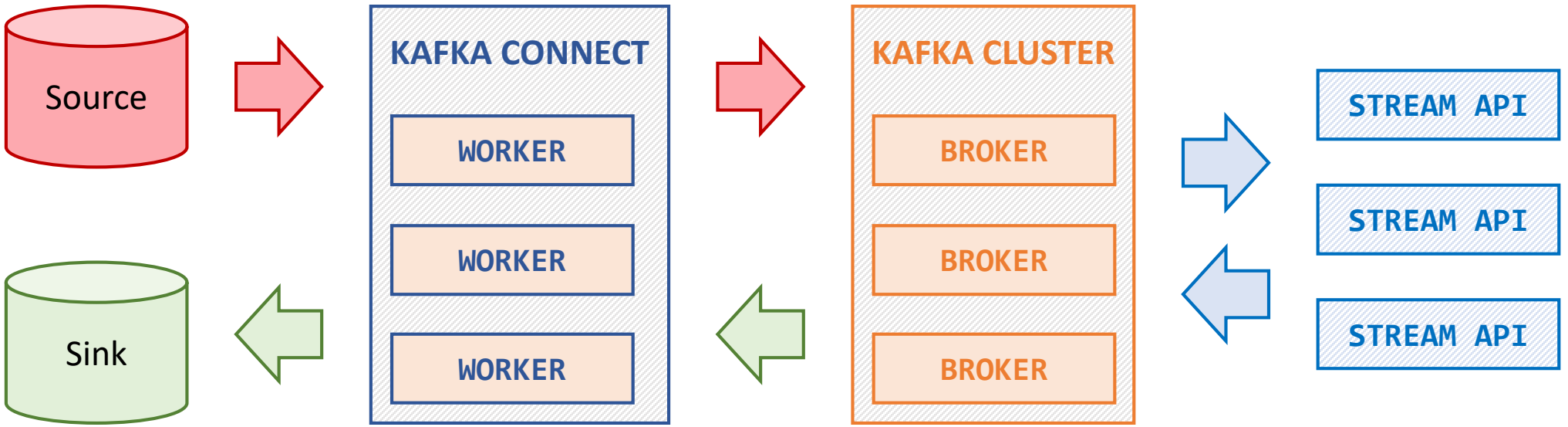
Avro Schema:

```
{ "namespace": "com.aioug.avro",
  "type": "record",
  "name": "StudentActivity",
  "fields": [
  {"name": "id", "type": "string"
  },
  { "name": "student_id", "type": "int"
  },
  { "name": "university_id", "type": "int"
  },
  { "name": "course_details",
    "type":
    { "name": "Activity",
      "type": "record",
      "fields": [
      { "name": "course_id", "type": "int"
      },
      { "name": "enroll_date", "type": "string"
      },
      { "name": "verb", "type": "string"
      },
      { "name": "result_score", "type": "double"
      }]
    }
  }]
}
```

Kafka REST Proxy



Kafka Connect & Streams Architecture



Oracle Event Hub Cloud

Oracle Event Hub Cloud Service delivers the power of Kafka as a managed streaming data platform integrated with the rest of Oracle's Cloud.

This enables the rapid, secure and cost-effective operation on streaming data by leveraging Kafka.

Instances,

- An instance in ***Oracle Event Hub Cloud Service – Dedicated*** refers to a Kafka Cluster.
- An instance in ***Oracle Event Hub Cloud Service*** refers to a Kafka Topic.

Oracle Event Hub Cloud

Features

- Apache Kafka delivered as a managed service
- Available in dedicated and multi-tenant flavours
- Elastic by nodes and by partitions

Benefits

- Realtime streaming platform
- Easy to use with REST APIs
- High performance Native API support
- Lift and Shift Kafka workloads from on-premise
- Elastic – scale from thousand to million of events per second
- Reliable – Highly available with in-cluster replication and cluster-mirroring

Oracle Event Hub Cloud – REST APIs

Use the Event Hub Cloud Platform REST API to

- Create and manage Oracle Event Hub Cloud Service clusters and topics
- View and manage network security rules
- Monitor the health of your service
- Apply patches and Scale clusters on-demand
- Manage the life cycle of your Oracle Event Hub Cloud Service

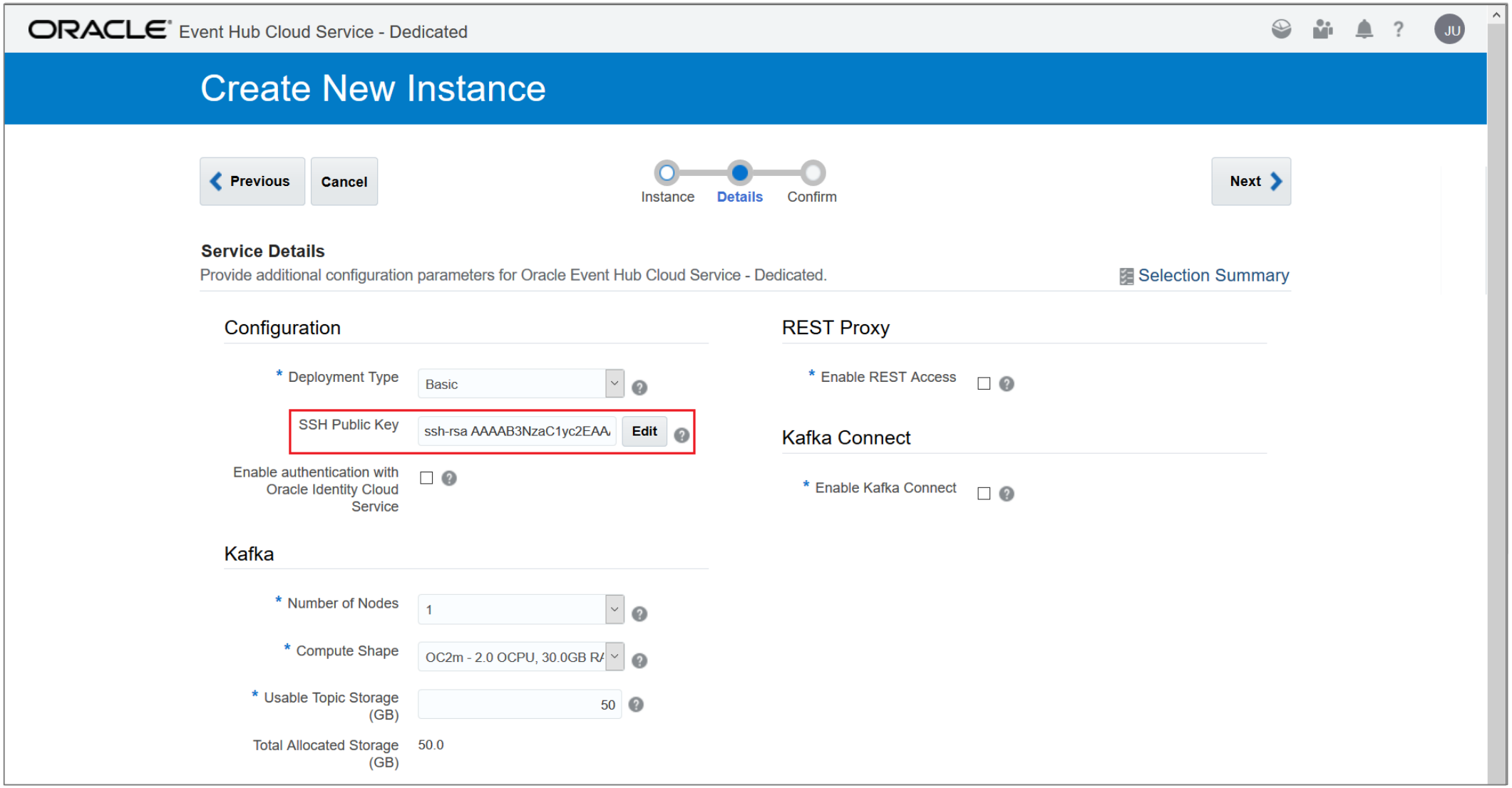
REST APIs offer easy provisioning and lifecycle management of Apache Kafka topic on Oracle Public Cloud with ability to create partitions, consume and produce messages.

These options are also available from Oracle PaaS Service Manager CLI as well.

Oracle Event Hub Cloud – Setup Steps

- Access the Oracle Event Hub Cloud Service Console
- Create an SSH Key Pair
- Create a Kafka Cluster
- Create a Kafka Topic
- Produce to and Consume from a Topic
- View the runtime metrics for the Topic
- Kafka Connect and REST configuration are optional

Create a Cluster



Create New Instance

Previous | Cancel

Instance > Details > Confirm

Next

Service Details

Provide additional configuration parameters for Oracle Event Hub Cloud Service - Dedicated.

Selection Summary

Configuration

* Deployment Type: Basic

SSH Public Key: ssh-rsa AAAAB3NzaC1yc2EAA... Edit

Enable authentication with Oracle Identity Cloud Service: ☐

Kafka

* Number of Nodes: 3

* Compute Shape: OC2m - 2.0 OCPU, 30.0GB R...

* Usable Topic Storage (GB): 50

Total Allocated Storage (GB): 150.0

Credentials

Username: admin

Password: ••••••••••

Confirm Password: ••••••••••

REST Proxy

* Enable REST Access: ☑

* Number of Nodes: 1

Compute Shape: OC1m - 1.0 OCPU, 15.0GB R...

Kafka Connect

* Enable Kafka Connect: ☑

* Number of Connect Nodes: 1

Compute Shape: OC1m - 1.0 OCPU, 15.0GB R...

service_payload_AIOUGKafkaCluster.json


```
{ "edition": "EE",
  "vmPublicKeyText": "ssh-rsa ....",
  "enableNotification": "true",
  "notificationEmail": "justinmraj@gmail.com",
  "userPassword": "<Fill_Here>",
  "serviceVersion": "0.10",
  "serviceLevel": "PAAS",
  "subscriptionId": "1776365",
  "serviceDescription": "creating a cluster in event hub dedicated cloud service",
  "serviceName": "AIOUGKafkaCluster",
  "useIdcsSecurity": "false",
  "userName": "admin",
  "isBYOL": "false",
  "components": {
    "connect": {
      "createConnect": "true",
      "connectShape": "oc1m",
      "connectClusterSize": "1"
    },
    "kafka": {
      "dataStorage": "50",
      "kafkaZkClusterSize": "3",
      "deploymentType": "Basic",
      "shape": "oc2m"
    },
    "restprxy": {
      "restprxyClusterSize": "1",
      "createRestprxy": "true",
      "restprxyShape": "oc1m"
    }  }
 ,"meteringFrequency": "HOURLY"
}
```

Demo…

Thank You !!!

Hariharaputhran & Justin Michael Raj

AIOUG Evangelists