KIP-80: Kafka Rest Server

Status

Current state: *DISCARDED*

Discussion thread: *here*

JIRA: *KAFKA-3294*

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

The goal is to add Kafka Rest Server to Kafka Repository. This will allow any language/tool that
can work with HTTP to produce and consume messages, and perform administrative actions with Kafka service.

Why add Kafka Rest Server to Kafka?

There are already some open-source REST proxies are available. Confluent REST Proxy got comprehensive interface.
But we would like to add REST server that many users ask for under Apache Kafka repo.

We want to add Kafka Rest Server to Kafka for the following reasons.

1) Many data Infra tools comes up with Rest Interface. It is useful to have inbuilt Rest API support for Produce,
   Consume messages and admin interface for integrating with external management and provisioning tools.
2) Shipping Kafka Rest Server as part of a normal Kafka release makes it immediately available to every user that downloads Kafka.
3) Helps to maintain the version compatibility between Kafka and Rest Server.

Some of the good practices and ideas will be borrowed from existing tools.

Proposed Changes

Rest Server

The REST server is a separate server, sitting between Kafka cluster and client applications. It is wrapper around existing client libraries.
Request/Responses supports JSON format with embedded data format (JSON and Base64 encoded strings). Proxy uses vendor specific
content types in Content-Type and Accept headers to make the format of the data explicit. This approach is borrowed from Confluent Rest Proxy.
Supported Content-Types are: application/vnd.kafka.binary.v1+json, application/vnd.kafka.json.v1+json
Rest server can be easily scaled by deploying multiple proxy instances. This way we can spread the load across multiple proxy instances.

Producer API

REST server accepts produce requests for specific topics or partitions. It internally uses java producer instance to write messages into Kafka.

Consumer API

REST Proxy uses the new consumer API to consume the messages from the subscribed topic on behalf of the specified consumer group. Consumer instances are stateful and tied to a particular Rest server instance.
A full URL is provided when the instance is created and it should be used to construct any subsequent requests.
When a message is consumed on behalf of a consume group for the first time, then Kafka Consumer instance joins the consumer group and subscribes to the topic. All Consumer instances that are currently members of that group and subscribed to that topic divide topic partitions among themselves. If a Consumer instance has not consumed from a particular topic on behalf of a particular consumer group for configured interval (normally large interval), then it unsubscribes from the topic on behalf of that group. This is for cleaning unused consumer instances due to dead clients.
Offset commit can be either automatic or manual as requested by the user.

We can also retrieve the messages for a consumer, from a specific partition, starting with an offset.

We also want to take community opinion on other ways of implementing consumer group functionality.

Admin API and Security Integration:

This will be taken up as future work after the KIP-4 implementation.

Public Interfaces

Producer API

POST /topics/:topic
Description : Produce messages to a given topic

Parameters:
Topic (String) - topic name

Request:
JSON Object contains array of produce records

Response:
JSON Object contains response objects

Status Codes:
404 Not Found
Error Code 40401 - topic Not Found
Error Code 40402 - Version Not Found
500 Internal Server Error
Error Code 50001 - Kafka Error

Example request:

```
POST /topics/test HTTP/1.1
Host:kafkarest.host.com
Content-Type: application/vnd.kafka.binary.v1+json
Accept: application/vnd.kafka.v1+json
{
 "records": [
  {
    "key": "a2V5",
    "value": "dmFsdWU="
  },
  {
    "value": "dmFsdWU=",
    "partition": 1
  }
 ]
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.kafka.v1+json

{
 "offsets": [
  {
    "partition": 2,
    "offset": 1
  },
  {
    "partition": 1,
    "offset": 2
  }
 ]
}
```

POST /topics/:topicName/partitions/:partition
Description : Produce messages to one partition of the topic

Parameters:
topicName (String) - topic name
partition (int) - partition number

Request:
JSON Object contains array of produce records

Response:
JSON Object contains response objects

Status Codes:
404 Not Found
Error Code 40401 - topic Not Found
Error Code 40402 - partition Not Found
500 Internal Server Error
Error Code 50001 - Kafka Error

Example request:

```
POST /topics/test/partitions/1 HTTP/1.1

Host:kafkarest.host.com

Content-Type: application/vnd.kafka.binary.v1+json

Accept: application/vnd.kafka.v1+json

{

  "records": [

   {

     "key": "a2V5",

     "value": "dmFsdWU="

   },

   {

     "value": "dmFsdWU="

   }

 ]

}
```

Example response:

```
HTTP/1.1 200 OK
 Content-Type: application/vnd.kafka.v1+json


{
 "offsets": [
  {
    "partition": 1,
    "offset": 1,
  },
  {
    "partition": 1,
    "offset": 2,
  }
 ]
}
```

Consumer API

Description : Create a new consumer instance in the consumer group.

POST /consumers/:group

Parameters:

- group (String) - group name

Request:

- JSON Object contains name, dataformat, consumer properties.

Response:

- JSON Object contains response objects.
- Response includes a URL including the host since the consumer is stateful and tied to a specific REST proxy instance

Status Codes:

- 404 Not Found
  - Error Code 40401 - topic Not Found
  - Error Code 40402 - partition Not Found
- 500 Internal Server Error
  - Error Code 50001 - Kafka Error

Example request:

```
POST /consumers/group/ HTTP/1.1
Accept: application/vnd.kafka.v1+json, application/vnd.kafka+json, application/json

{
 "name": "Instance1",
 "format": "binary",
 "auto.offset.reset": "smallest",
 "auto.commit.enable": "false"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.kafka.v1+json

{
 "id": "Instance1",
 "url": "http://kafkarest1.com/consumers/group/instances/Instance1"
}
```

Description : consumes a message from the specified topic on behalf of the specified consumer group

GET /consumers/:group/:instance/:topic

Parameters:

- topic (String) - topic name
- group (String) - consumer group id
- Instance (string) - consumer instance name

Response:

- JSON Object contains response objects

Status Codes:

- 404 Not Found
  - Error Code 4040X - topic Not Found
  - Error Code 4040Y - group Not Found
  - Error Code 4040Z - instance Not Found
- 500 Internal Server Error
  - Error Code 50001 - Kafka Error

Example request:

```
GET /consumers/group/instance1/topic HTTP/1.1
Accept: application/vnd.kafka.binary.v1+json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.kafka.binary.v1+json

[
 {
   "key": "a2V5",
   "value": "dmFsdWU=",
   "partition": 1,
   "offset": 1,
 },
 {
    "key": "a2V5",
   "value": "dmFsdWU=",
   "partition": 1,
   "offset": 2,
 }
]
```

Description : Commit offsets for the consumer instance associated with group.

POST /consumers/:group/:instance/offsets

Parameters:

- group (String) - consumer group id
- Instance (String) - consumer instance name

Response:

- JSON Object contains response objects

Status Codes:

- 404 Not Found
  - Error Code 40401 - group Not Found
  - Error Code 40402 - consumer instance Not Found
- 500 Internal Server Error
  - Error Code 50001 - Kafka Error

Example request:

```
POST /consumers/group/instance1/offsets HTTP/1.1
Accept: application/vnd.kafka.v1+json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.kafka.v1+json

[
 {
   "topic": "test",
   "partition": 1,
   "committed": 100
 },
 {
   "topic": "test",
   "partition": 2,
   "committed": 200
 },
 {
   "topic": "test2",
   "partition": 1,
   "committed": 50
 }
]
```

GET /topics/:topic_name/partitions/:partition?offset=(int)
Description : Consume messages from one partition of the topic.

Parameters:

topic_name (String) - topic name
partition (int) - partition number
offset (int) - offset to fetch

Response:
JSON Object contains response objects

Status Codes:
404 Not Found
Error Code 40401 - topic Not Found
Error Code 40402 - partition Not Found
500 Internal Server Error
Error Code 50001 - Kafka Error

Example request:

```
GET /consume/test/partitions/1?offset=1 HTTP/1.1
Accept: application/vnd.kafka.binary.v1+json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.kafka.binary.v1+json

[
 {
   "key": "a2V5",
   "value": "dmFsdWU=",
   "partition": 1,
   "offset": 1,
 },
 {
    "key": "a2V5",
   "value": "dmFsdWU=",
   "partition": 1,
   "offset": 2,
 }
]
```

TODO
I will update required config options and response Error codes/messages.

Compatibility, Deprecation, and Migration Plan

This KIP only proposes additions. There should be no compatibility issues.

Rejected Alternatives

Make Kafka Rest Server an external third-party tool

Main Reason for this KIP is to add Rest Server to Kafka. Shipping Kafka Rest Server as part of a normal Kafka release makes it immediately available to every user that downloads Kafka. Also helps to maintain the version compatibility between Kafka clients and Rest Server.

Push/Stream messages to end clients:

End clients can register for the listening of events. This can be implemented used Web socket push notifications.
But This is against kafka consumer's pull model. It is good maintain Kafka consumer's semantics.