KIP-59: Proposal for a kafka broker command

This page is meant as a template for writing a KIP. To create a KIP choose Tools->Copy on this page and modify with your content and replace the heading with the next KIP number and a description of your issue. Replace anything in italics with your own description.

Status

Current state: Under Discussion

Discussion thread:

JIRA: ⚠ Unable to render Jira issues macro, execution error.

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

Motivation

This is a proposal for an admin tool - say, ***kafka-brokers.sh*** to provide broker related useful information. Some of the key factors for Kafka's success are its performant architecture and operational simplicity. This is further complemented with a set of commandline tools and utilities for managing topics as well as testing/stress-testing. However currently Kafka lacks commands/tools to get cluster and broker overview. Although it should be mentioned that Kafka does expose cluster information via API and broker metrics via JMX.

Public Interfaces

The kafka-broker.sh command is modeled after the kafka-topic.sh and has options as described later below.

Proposed Changes

The command will essentially provide the following pieces of information:

Broker Information

- Cluster Id
- Controller info
- And for each broker
- Broker Id
- Hostname
- Rack
- Endpoints
- Topic count (i.e. how many topics are hosted by the broker)
- All partition count (i.e. how many partition replicas are hosted by the broker)
- Leader partition count (i.e. how many leader partitions are hosted by the broker)
- Insync partition count (i.e, how many follower partitions are insync)
- Trailing partition count (i.e. how many partitions are not insync)
- When details are printed, A leader partition will be prefixed by a "+" symbol and a trailing partition will be prefixed by a "-" symbol

With command line options, topic names and partition numbers can be enumerated.

Command Options

The command kafka-brokers.sh requires zookeeper information and additional accepts other options as show below:

kafka-brokers.sh Usage/Options

```
$ ./kafka-brokers.sh
Missing required argument "[bootstrap-server]"
Option                                  Description
------                                  -----------
--bootstrap-server <String: The server  REQUIRED: the server(s) to use for
  (s) to use for bootstrapping>           bootstrapping
--broker <Integer: broker>              Filter for a broker. Option can be
                                          used multiple times for multiple
                                          broker-ids
--details                               if specified, shows detailed listing
                                          of topics and partitions
--host <String: host>                   Filter for a hostname. Option can be
                                          used multiple times for multiple
                                          hostnames
--partition-details                     if specified, shows partitions in each
                                          topic
--rack <String: rack>                   Filter for a rack. Option can be used
                                          multiple times for multiple racks
--topic <String: topic>                 Filter for a topic. Option can be used
                                          multiple times for multiple topics
--topic-details                         if specified, shows topics and
                                          partition counts in each topic
```

The options "–broker", "–host", "–rack" and "–topic" provide filtering for the specified broker (broker-id), host, rack or topic. Each of the options can be specified multiple times. The option "--topic-details" makes the command to enumerate all the topic names in addition to giving the topic count. The option "--partition-details" makes the command to enumerate all the partitions for each topic. The partitions have an an optional prefix of "+" or "-" to indicate that the partition is a leader partition or an under-replicated partition respectively. An insync replica partition will not have any prefix.

Here are examples usages of the command.

Summary Output Without Any Details

```
$ ./kafka-brokers.sh --bootstrap-server host1:9092,host2:9092
BrokerId: 1        Hostname: host1        Rack: rack1        Topics: 2        Partitions: 12        Leaders:
4        InSync: 8        Trailing: 0
BrokerId: 2        Hostname: host2        Rack: rack2        Topics: 2        Partitions: 12        Leaders:
4        InSync: 8        Trailing: 0
BrokerId: 3        Hostname: host3        Rack: rack3        Topics: 2        Partitions: 12        Leaders:
4        InSync: 8        Trailing: 0
```

Output with Topic Details

```
BrokerId: 1        Hostname: host1        Rack: rack1        Topics: 2        Partitions: 12        Leaders:
4        InSync: 8        Trailing: 0        Topic Details: topic2 with 6 partitions, topic1 with 6
partitions
BrokerId: 2        Hostname: host2        Rack: rack2        Topics: 2        Partitions: 12        Leaders:
4        InSync: 8        Trailing: 0        Topic Details: topic2 with 6 partitions, topic1 with 6
partitions
BrokerId: 3        Hostname: host3        Rack: rack3        Topics: 2        Partitions: 12        Leaders:
4        InSync: 8        Trailing: 0        Topic Details: topic2 with 6 partitions, topic1 with 6 partitions
```

Output with Partition Details

```
BrokerId: 1        Hostname: host1        Rack: rack1        Topics: 2        Partitions: 12        Leaders:
4        InSync: 8        Trailing: 0        Partition Details: (topic2: 5,4,+3,2,1,+0), (topic1: 5,4,+3,2,1,
+0)
BrokerId: 2        Hostname: host2        Rack: rack2        Topics: 2        Partitions: 12        Leaders:
4        InSync: 8        Trailing: 0        Partition Details: (topic2: 5,+4,3,2,+1,0), (topic1: 5,+4,3,2,+1,
0)
BrokerId: 3        Hostname: host3        Rack: rack3        Topics: 2        Partitions: 12        Leaders:
4        InSync: 8        Trailing: 0        Partition Details: (topic2: +5,4,3,+2,1,0), (topic1: +5,4,3,+2,1,0)
```

Output with Topic and Partition Details

```
BrokerId: 1        Hostname: host1        Rack: rack1        Topics: 2        Partitions: 12        Leaders:
4        InSync: 8        Trailing: 0        Topic Details: topic2 with 6 partitions, topic1 with 6
partitions        Partition Details: (topic2: 5,4,+3,2,1,+0), (topic1: 5,4,+3,2,1,+0)
BrokerId: 2        Hostname: host2        Rack: rack2        Topics: 2        Partitions: 12        Leaders:
4        InSync: 8        Trailing: 0        Topic Details: topic2 with 6 partitions, topic1 with 6
partitions        Partition Details: (topic2: 5,+4,3,2,+1,0), (topic1: 5,+4,3,2,+1,0)
BrokerId: 3        Hostname: host3        Rack: rack3        Topics: 2        Partitions: 12        Leaders:
4        InSync: 8        Trailing: 0        Topic Details: topic2 with 6 partitions, topic1 with 6
partitions        Partition Details: (topic2: +5,4,3,+2,1,0), (topic1: +5,4,3,+2,1,0)
```

Compatibility, Deprecation, and Migration Plan

This command is based on the new AdminClient, so is compatible with Kafka versions 0.10.1 and higher only.

Rejected Alternatives

If there are alternative ways of accomplishing the same thing, what were they? The purpose of this section is to motivate why the design is the way it is and not some other way.