Log Management in Apache Kafka

Lee Dongjin | dongjin@apache.org

Introducing Myself …

- Software Developer, Big Data Technologies
  - Hadoop, Giraph, HBase, Spark, Kafka, …
- Apache Kafka Consultant
- Apache Kafka Contributor
  - KIP-110: Add Codec for ZStandard Compression
  - KIP-508: Make Suppression State Queriable
  - KIP-653: Upgrade log4j to log4j2 (Today's Topic!)
  - etc…

Log - What Log?

- Log = Set of events

- Meaning 1:
  - Append-only data structure
  - In Kafka, stored in ${log.dirs}, ${log.dir}

- Meaning 2:
  - A file that records the events occur in the system internals.
  - In Kafka, stored in ${kafka.logs.dir}

- We will focus on meaning 2 only in this talk.

Log Management in Kafka

- Logging in Kafka Broker
  - What to monitor?
  - How to manage it?

- Logging in Kafka Connect
  - What to monitor?
  - How to manage it?

- Upcoming: Log4j2 (Preview)
  - Differences?
  - How to activate it?

Logging in Kafka Broker (1)

- So many loggers in Kafka Broker

  - {number of loggers} > 110 (as of 2.7.0)

- Too many loggers to keeping eyes on them…

  - Then, which loggers are more important?

Logging in Kafka Broker (2)

- `kafka.request.logger`
  - Receives various Kafka requests.
  - Default: ${kafka.logs.dir}/kafka-request.log

- `kafka.server.KafkaApis`
  - Handles various Kafka requests.
  - Not defined in the default logging configuration.

- `kafka.log.LogCleaner`
  - Log Compaction, Retention, etc.
  - Default: ${kafka.logs.dir}/log-cleaner.log

How to manage logging in Kafka Broker? (1)

- Modifying the log configuration & restarting?

  - A squad of brokers?

  - Kubernetes pods?

- We need to configure the log settings dynamically!

- Alternatives

  - Method 1: Use JMX Console

  - Method 2: Use Command Line Tool (kafka-configs.sh)

    - … or Admin API

How to manage logging in Kafka Broker? (2)

- Method 1: Use JMX Console
  - You can change the logger's logging level dynamically with JMX bean.
  - Similar Features: Apache Storm, Apache Zookeeper
- How to use it?

How to manage logging in Kafka Broker? (3)

- Start Kafka with JMX enabled.

- Connect to {kafka-broker}:{jmx-port} with JMX Console

- Open `kafka` > `kafka.request.logger` MBean

  - To get Logger Level

    - click getLogLevel from operations tab

    - p1 = logger name (e.g., 'kafka.request.logger') ↵

  - To set Logger Level

    - click setLogLevel from operations tab

    - p1 = logger name (e.g., 'kafka.request.logger'), p2 = logger level (e.g., 'INFO') ↵

How to manage logging in Kafka Broker? (4)
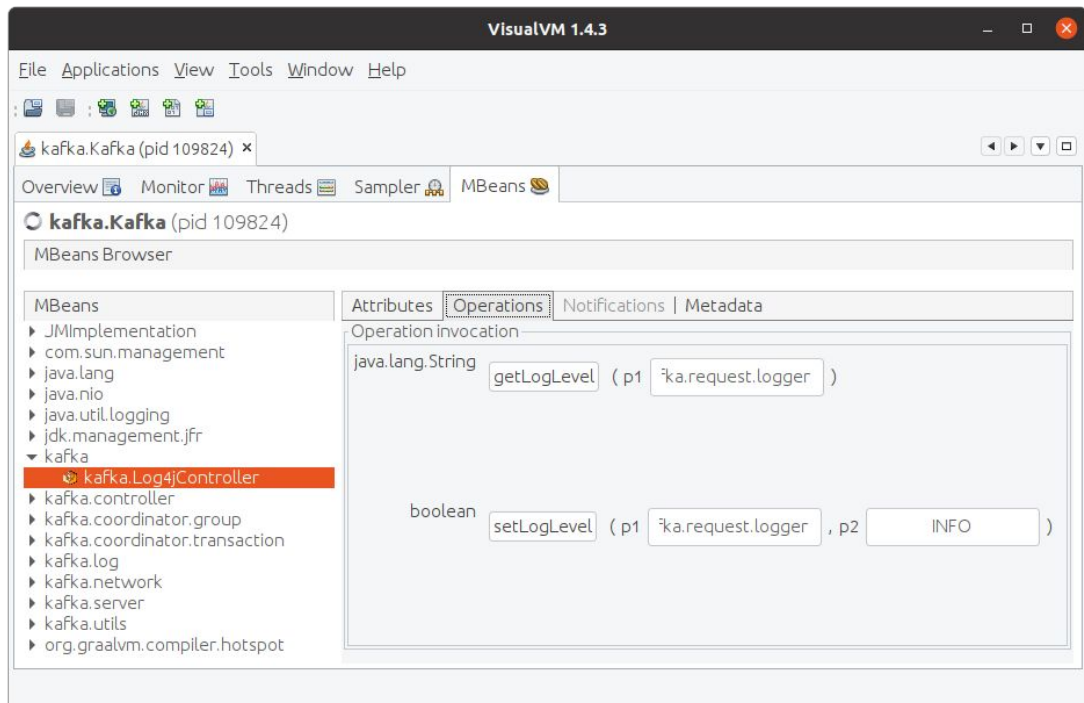
How to manage logging in Kafka Broker? (5)

- Method 2: Use Command Line Tool
  - To check current loggers:

```
$ bin/kafka-configs.sh --bootstrap-server {kafka-broker}:9092 \
        --entity-type broker-loggers --entity-name {broker-id} \
        --describe
Dynamic configs for broker-logger {broker-id} are:
  org.apache.zookeeper.CreateMode=INFO sensitive=false synonyms={}
  kafka.coordinator.transaction.ProducerIdManager=INFO sensitive=false synonyms={}
  kafka.utils.KafkaScheduler=INFO sensitive=false synonyms={}
  ...
```

How to manage logging in Kafka Broker? (6)

- Method 2: Use Command Line Tool

  - To update current logger level (e.g., `kafka.request.logger` to TRACE):

```
$ bin/kafka-configs.sh --bootstrap-server {kafka-broker}:9092 \
          --entity-type broker-loggers --entity-name {broker-id} \
          --alter --add-config kafka.request.logger=TRACE
Completed updating config for broker-logger {broker-id}.
```

How to manage logging in Kafka Broker? (7)

- Dynamic Configuration API was updated in 2.3.0
    - Admin#incrementalAlterConfigs
    - KIP-339: Create a new IncrementalAlterConfigs API
    - KIP-412: Extend Admin API to support dynamic application log levels
- Before use this tool, please update the installation first.

Logging in Kafka Connect (1)

- Question: Too many logging messages are mixed in one log file.

    - ${kafka.logs.dir}/connect.log

    - How can I separate the logging messages I need...?

    - Like, the log messages from a specific task.

- [KIP-449: Add connector contexts to Connect worker logs](#)

    - From 2.3.0

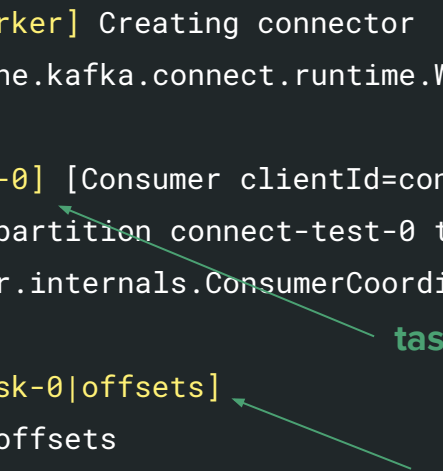   - Provides Logging Context

Logging in Kafka Connect (2)

- Logging format: [<connectorName>|<scope>] <message>
  - [my-jdbc-connector|worker] :
    - Connector Creation, Config Validation, Starting, Stopping, …
  - [my-jdbc-connector|task-0]:
    - Task Creation, Consuming, Producing, …
  - [my-jdbc-connector|task-0|offsets]:
    - Task's offset commit.

Logging in Kafka Connect (3)

- Example

```
[2019-04-02 17:01:38,315] INFO [local-file-source|worker] Creating connector
local-file-source of type FileStreamSource (org.apache.kafka.connect.runtime.Worker:227)
...
[2019-04-02 17:01:38,487] INFO [local-file-sink|task-0] [Consumer clientId=consumer-1,
groupId=connect-local-file-sink] Setting offset for partition connect-test-0 to the
committed offset 2 (org.apache.kafka.clients.consumer.internals.ConsumerCoordinator:511)
...
[2019-04-02 17:01:40,901] INFO [local-file-source|task-0|offsets]
WorkerSourceTask{id=local-file-source-0} Committing offsets
(org.apache.kafka.connect.runtime.WorkerSourceTask:398)
```

worker log

task log

task commits offset!

How to manage logging in Kafka Connect? (1)

- Checking current loggers with corresponding levels
    - GET to http://{kafka-connect-host}:8083/admin/loggers
- Example

```
$ curl -s http://{kafka-connect-host}:8083/admin/loggers
{
  "org.apache.kafka.connect.runtime.rest": {
    "level": "WARN"
  },
  ...
}
```

How to manage logging in Kafka Connect? (2)

- Modifying Logging Levels
    - Put Json to http://{kafka-connect-host}:8083/admin/loggers/{logger-name}
- Example
    - Change logger 'io.debezium.*'s level into 'TRACE'

```
$ curl -s -X PUT -H "Content-Type:application/json" \
    http://{kafka-connect-host}:8083/admin/loggers/io.debezium \
    -d '{"level": "TRACE"}' | jq '.'
[
  "io.debezium",
  "io.debezium.connector.mysql.MySqlConnector",
  ...
]
```

Upcoming Feature: Log4j2 (1)

- Log4j 1.x: deprecated from May, 2012 (1.2.17)
  - Obsolete Configuration Syntax
  - Security Problems
    - [CVE-2019-17571: SocketServer Vulnerability](#)
  - **Problem: Kafka is still using it.**
    - Only Log4j 1.x to Kafka is supported.

- Log4j 2.x: current de-facto standard
  - Will be adopted to Apache Kafka > 2.7.0
    - [KIP-676: Respect logging hierarchy](#) (Bug Fix)
    - [KIP-653: Upgrade log4j to log4j2](#)

Upcoming Feature: Log4j2 (2)

- log4j 1.x vs. log4j2 Configuration Syntax

```
# root logger
log4j.rootLogger=INFO, stdout, kafkaAppender


...


log4j.logger.kafka=INFO
log4j.logger.org.apache.kafka=INFO


log4j.logger.kafka.request.logger=WARN,
requestAppender
log4j.additivity.kafka.request.logger=false
```

```
# root logger
rootLogger.level=INFO


...


# in log4j2, list loggers first
loggers=org.apache.zookeeper, ...


logger.org.apache.kafka.name=org.apache.kafka
logger.org.apache.kafka.level=INFO
logger.kafka.request.logger.additivity=false
```

Upcoming Feature: Log4j2 (3)

- Replaces log4 to log4j2

  - Removes Security Vulnerability

- Default: Log4j configuration

  - For backward-compatibility (with log4j-1.2-api)

- How can I activate log4j2 mode?

  - Use JVM configuration

    - Log4j: log4j.configuration

    - Log4j2: log4j.configuration**File**

  - export

    KAFKA_LOG4J_OPTS="-Dlog4j.configurationFile=file:$base_dir/../config/log4j2.properties"

Upcoming Feature: Log4j2 (3)

- What Changes?

  - Except Configuration Syntax, NONE.

  - Log4j2 version default configurations will be provided, also.

- Note: Root Logger's name

  - Log4j: "root"

  - Log4j2: "" (empty string!)

  - Kafka: "root"

    - For backward-compatibility

Upcoming Feature: Log4j2 (3)

- Question 1: When can we use the feature?
  - Public Beta will be provided soon.
    - 2.7.0+log4j2, 2.6.1+log4j2, etc.
    - with English, Korean documentation.
- Question 2: When log4j2 configuration will be default?
  - Not Decided Yet.
- Question 3: What about Log4j2 Appender?
  - Working in Progress.

Summary

- You can manage Kafka Brokers' log with …
  - JMX Console
  - Command Line Tool
- You can manager Kafka Connects' log with…
  - Rest API
- Kafka will migrate into log4j2
  - With Backward-compatibility!

Questions?