

# Школа Java Middle Developer

## Kafka

Инструменты для работы с Kafka

# Содержание

1. Обзор инструментов для работы с Kafka;
2. Kafka CLI.

# Зачем нужны специализированные инструменты для работы с Kafka?

1. kcat (или Kafka-CLI) – командная строка
2. Kafka Monitoring Tools – инструменты мониторинга
3. Kafka UIs – управление кластером Kafka
4. Debezium – используется для связи реляционных БД с Kafka
5. Kafka Streams – потоковая обработка
6. MirrorMaker - утилита репликации
7. Kafka Connect – взаимодействие с внешними системами
8. Cruise Control (LinkedIn) – анализ использования ресурсов
9. Kafka Security Manager (KSM) – обеспечение контроля доступа

# Kafka CLI

- Создание и использование сообщений непосредственно из топиков Kafka;
- Просмотр метаданных и конфигураций топика;
- Управление группами потребителей (перечисление, описание и сброс настроек).

# Администрирование Kafka

# Операции с топиками

➤ kafka-topics.sh

➤ kafka-configs.sh

Обязательный аргумент --bootstrap-server

# Создание нового топика

## ➤ Создание топика

```
kafka-topics.sh --bootstrap-server <server connect> --create --topic  
<string> --replication-factor <integer> --partitions <integer>
```

## ➤ Проверка топика

```
kafka-topics.sh --describe --topic my-topic --bootstrap-server  
localhost:9092
```

# Добавление партиций

Увеличение количества партиций для топика my-topic до 16

```
kafka-topics.sh --bootstrap-server localhost:9092 --alter  
--topic my-topic --partitions 16
```

Для игнорирования ошибки в случае отсутствия топика для команды  
--alter существует аргумент --if-exists

Уменьшить количество партиций топика невозможно



# Удаление топика

Удаление топика с названием my-topic

```
kafka-topics.sh --bootstrap-server localhost:9092 --delete  
--topic my-topic
```

Удаление топика приводит к удалению всех его сообщений

# Вывод списка всех топиков кластера

Список всех топиков кластера:

```
kafka-topics.sh --bootstrap-server localhost:9092 --list
```

Формат списка - по одному топик в строке, без сортировки

# Подробное описание топикиов

## Описание всех топикиов

```
kafka-topics.sh --bootstrap-server localhost:9092 --describe
```

- Можно ограничиться информацией по одному топику, указав для команды аргумент `--topic`
- Для поиска всех топикиов, у которых были переопределены настройки, воспользуйтесь аргументом `--topics-with-overrides`
- Аргумент `--under-replicated-partitions` выводит все партиции, одна или более реплик которых не согласованы с ведущей репликой
- Аргумент `--un-available-partitions` выводит все партиции, у которых нет ведущей реплики

# Группы потребителей

# Вывод списка и описание групп

## Список групп

```
kafka-consumer-groups.sh --bootstrap-server localhost:9092  
--list
```

Более подробное описание любой из перечисленных групп можно получить, заменив параметр `--list` на `--describe` и добавив параметр `--group`

# Вывод списка и описание групп

Поле	Описание
GROUP	Название группы потребителей
TOPIC	Название читаемого топика
PARTITION	Идентификатор читаемой партии
CURRENT-OFFSET	Последнее смещение, зафиксированное группой потребителей для данной партии. Представляет собой позицию потребителя в партии
LOG-END-OFFSET	Текущее максимальное смещение для данной партии топика из имеющихся в брокере
LAG	Разница между Current-Offset потребителя и Log-End-Offset брокера для данной партии топика
OWNER	Член группы потребителей, который в настоящий момент выполняет потребление данной партии топика. Представляет собой произвольный идентификатор, задаваемый самим членом группы. Не обязан содержать имя хоста потребителя

# Удаление группы

Удаление группы потребителей testgroup

```
kafka-consumer-groups.sh --bootstrap-server localhost:9092 --delete  
--group testgroup
```

Удаление группы потребителей приводит к удалению всей группы

# Динамическое изменение конфигурации

kafka-configs.sh

Обязательный атрибут --bootstrap-server



# Переопределение значений настроек топиков по умолчанию

Изменение настроек топика

```
kafka-configs.sh --bootstrap-server localhost:9092 --alter  
--entity-type topics --entity-name <topic name>  
--add-config <key>=<value>[, <key>=<value>...]
```

# Допустимые ключи конфигурации топиков

Ключ конфигурации	Описание
file.delete.delay.ms	Длительность (в миллисекундах) ожидания перед удалением сегментов журнала и индексов для данного топика с диска
flush.messages	Количество сообщений, до принудительного сброса сообщений данного топика на диск
flush.ms	Промежуток времени (в миллисекундах) перед принудительным сбросом сообщений данного топика на диск
index.interval.bytes	Допустимое количество байтов сообщений, генерируемых между записями индекса сегмента журнала
max.message.bytes	Максимальный размер отдельного сообщения данного топика (в байтах)
message.format.version	Используемая брокером при записи сообщений на диск версия формата сообщений. Должна представлять собой допустимую версию API (например, 0.10.0)
message.timestamp.difference.max.ms	Максимально допустимая разница (в миллисекундах) между меткой даты/ времени отправки сообщения и меткой даты/времени брокера о его получении
message.timestamp.type	Используемая при записи сообщений на диск метка даты/времени. Текущие значения - CreateTime для задаваемой клиентом метки даты/времени и LogAppendTime при записи сообщения в партицию брокером
min.cleanable.dirty.ratio	Частота попыток сжатия партиций данного топика утилитой сжатия журналов (в виде отношения числа несжатых сегментов журнала к общему числу сегментов). Имеет смысл только для топиков со сжатием журналов
min.insync.replicas	Минимальное количество согласованных реплик, необходимое для того, чтобы партиция топика считалась доступной
preallocate	При установке в true место под сегменты журналов для этого топика будет выделяться заранее, при создании нового сегмента
retention.bytes	Объем хранимых сообщений этого топика (в байтах)
retention.ms	Длительность (в миллисекундах) хранения сообщений данного топика
segment.bytes	Объем сообщений (в байтах), записываемый в отдельный сегмент журнала в партиции
segment.index.bytes	Максимальный размер (в байтах) отдельного индекса сегмента журнала
segment.jitter.ms	Максимальное число миллисекунд, задаваемых случайным образом и добавляемых к ключу segment.ms при создании сегментов журналов
segment.ms	Частота (в миллисекундах) чередования сегментов журналов для каждой из партиций
unclean.leader.election.enable	При установке в false для данного топика будет запрещен «нечистый» выбор ведущей реплики

# Переопределение настроек клиентов по умолчанию

Команда изменения настроек клиента

```
kafka-configs.sh --bootstrap-server localhost:9092
```

```
--alter --entity-type clients --entity-name <client ID>
```

```
--add-config <key>=<value>[, <key>=<value>...]
```

Ключ конфигурации	Описание
producer_bytes_rate	Допустимый объем генерируемых для одного брокера в секунду по отдельному идентификатору клиента сообщений (в байтах)
consumer_bytes_rate	Допустимый объем потребляемых из одного брокера в секунду по отдельному идентификатору клиента сообщений (в байтах)

# Описание переопределений настроек

Список всех переопределений настроек для топика my-topic

```
kafka-configs.sh --bootstrap-server localhost:9092
```

```
--describe --entity-type topics --entity-name my-topic
```

Описание конфигурации выводит только переопределения и не включает настройки кластера по умолчанию

# Удаление переопределений настроек

Удаление переопределения настроек ключа retention.ms для топика my-topic

```
kafka-configs.sh --bootstrap-server localhost:9092  
--alter --entity-type topics --entity-name my-topic  
--delete-config retention.ms
```

# Управление партициями

# Выбор предпочтительной ведущей реплики

Выбор предпочтительной ведущей реплики для всех партиций  
всех топиков кластера:

```
kafka-leader-election.sh --bootstrap-server localhost:9092  
--election-type PREFERRED --all-topic-partitions
```

# Смена реплик партиции

Примеры случаев, когда необходимо изменить распределение реплик для партиции:

- Партиции топики не сбалансированы в масштабах кластера, что приводит к неравномерной нагрузке брокеров;
- Один из брокеров отключился, и партиция недореплицирована;
- Был добавлен новый брокер, и ему необходимо выделить часть нагрузки кластера.



# Смена реплик партиции

Для генерации набора перемещений партиций необходимо создать файл с JSON-объектом, содержащим список топигов. Формат JSON-объекта следующий:

```
{
  "topics": [
    {
      "topic": "foo"
    },
    {
      "topic": "foo1"
    }
  ],
  "version": 1
}
```

# Смена реплик партиции

Следующий пример команды, генерирует перемещения партиций, чтобы перенести перечисленные в файле topics.json партиции на брокеры с идентификаторами 0 и 1:

```
kafka-reassign-partitions.sh --bootstrap-server localhost:9092  
--generate --topics-to-move-json-file topics.json --broker-list 0,1
```

Current partition replica assignment

```
{"version":1,"partitions":[{"topic":"quickstart-  
events","partition":0,"replicas":[1],"log_dirs":["any"]}]}
```

Proposed partition reassignment configuration

```
{"version":1,"partitions":[{"topic":"quickstart-  
events","partition":0,"replicas":[1],"log_dirs":["any"]}]}
```

# Смена реплик партиции

Выполнение предлагаемого перераспределения партиций из файла:

```
kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --execute  
--reassignment-json-file reassign.json
```

Проверка состояния выполняющегося переназначения партиций из файла reassign.json можно командой:

```
kafka-reassign-partitions.sh --bootstrap-server localhost:9092 --verify  
--reassignment-json-file reassign.json
```

# Изменение коэффициента репликации

- ❖ Изменить коэффициент репликации можно посредством создания JSON-объекта в том же формате, который применялся на этапе переназначения партиций для добавления/удаления реплик с целью задания нужного коэффициента репликации.

```
{
  "partitions": [
    {
      "topic": "my-topic",
      "partition": 0,
      "replicas": [
        1
      ]
    }
  ],
  "version": 1
}
```

```
{
  "partitions": [
    {
      "topic": "my-topic",
      "partition": 0,
      "replicas": [
        1,
        2
      ]
    }
  ],
  "version": 1
}
```

# Проверка реплик

Проверка реплики для топиков, название которых начинается с «my-», на брокерах 1 и 2:

```
kafka-replica-verification.sh --broker-list localhost:9092  
--topic-white-list 'my-.*'
```

# Потребление и генерация

# Консольный потребитель

- ❖ Утилита `kafka-console-consumer.sh` позволяет потреблять сообщения из одного или нескольких топиков кластера Kafka
- ❖ Важно использовать потребитель той же версии, что и кластер Kafka

```
# kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic my-topic
```

`--formatter CLASSNAME` — задает используемый при декодировании сообщений класс подпрограммы форматирования сообщений. По умолчанию

`kafka.tools.DefaultMessageFormatter`;

`--from-beginning` – потреблять сообщения, начиная с первого смещения;

`--max-messages NUM` – читать не более NUM сообщений, до завершения работы;

`--partition NUM` — читать только из партии с идентификатором NUM.

# Параметры подпрограммы форматирования сообщений

Существуют три реализованные подпрограммы форматирования сообщений:

- ❖ `kafka.tools.LoggingMessageFormatter` – выводит сообщения посредством механизма журналирования вместо стандартного потока вывода. Сообщения выводятся на уровне INFO и включают метку даты/времени, ключ и значение;
- ❖ `kafka.tools.NoOpMessageFormatter` – потребляет сообщения, но не выводит их вообще.



# Параметры подпрограммы форматирования сообщений

У `kafka.tools.DefaultMessageFormatter` есть также несколько удобных параметров, которые можно задавать с помощью параметра командной строки `--property`:

- `print.timestamp` – установите в `true` для отображения метки даты/времени каждого из сообщений (при наличии);
- `print.key` – установите в `true` для отображения наряду со значением ключа сообщения;
- `key.separator` – задает символ-разделитель, выводимый между ключом и значением сообщения;
- `line.separator` – задает символ-разделитель, выводимый между сообщениями;
- `key.deserializer` – позволяет задать имя класса для десериализации ключа;
- `value.deserializer` – позволяет задать имя класса для десериализации значения.

# Чтение топи́ков смещений

Для чтения отдельного сообщения из топика смещений, необходимо выполнить следующую команду (для версии Kafka начиная с 0.11):

```
# kafka-console-consumer.sh --consumer.config config/consumer.properties  
  --formatter  
  "kafka.coordinator.group.GroupMetadataManager\${OffsetsMessageFormatter}"  
  --bootstrap-server localhost:9092  
  --topic __consumer_offsets  
  --from-beginning  
  --max-messages 1
```

# Консольный производитель

Команда запуска консольного производителя:

```
# kafka-console-producer.sh --topic my-topic --bootstrap-server localhost:9092
```

# Аргументы командной строки консольного производителя

- `--key-serializer CLASSNAME` - задает используемый при сериализации ключей сообщений класс подпрограммы кодирования. По умолчанию `kafka.serializer.DefaultEncoder`;
- `--value-serializer CLASSNAME` - задает используемый при сериализации значений сообщений класс подпрограммы кодирования. По умолчанию `kafka.serializer.DefaultEncoder`;
- `--compression-codec STRING` - задает используемый при генерации сообщений тип сжатия. Возможные значения - `none`, `gzip`, `snappy` и `lz4`. Значение по умолчанию `gzip`;
- `--sync` — генерирует сообщения синхронно, ожидая подтверждения получения каждого из них перед отправкой следующего.

**Спасибо за внимание**