OTUS

Онлайн образование

otus.ru

Проверить, идет ли запись

Меня хорошо видно && слышно?

Тема вебинара

Топики

Заигрин Вадим

Ведущий эксперт по технологиям, Сбербанк

vzaigrin@yandex.ru
https://t.me/vzaigrin

Преподаватель

Вадим Заигрин

Более 30 лет в ИТ:
- Big Data
  - Data Engineer
  - Data Science
- Разработка
  - Scala, Java, Python, C, Lisp
- IT Infrastructure
  - Администрирование
  - Сопровождение
  - Архитектура

Big Data проекты в банках, телекоме и в рознице.

Правила вебинара

Активно
участвуем

Off-topic обсуждаем
в Telegram

Задаем вопрос
в чат или голосом

Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения

Индивидуально

Время, необходимое
на активность

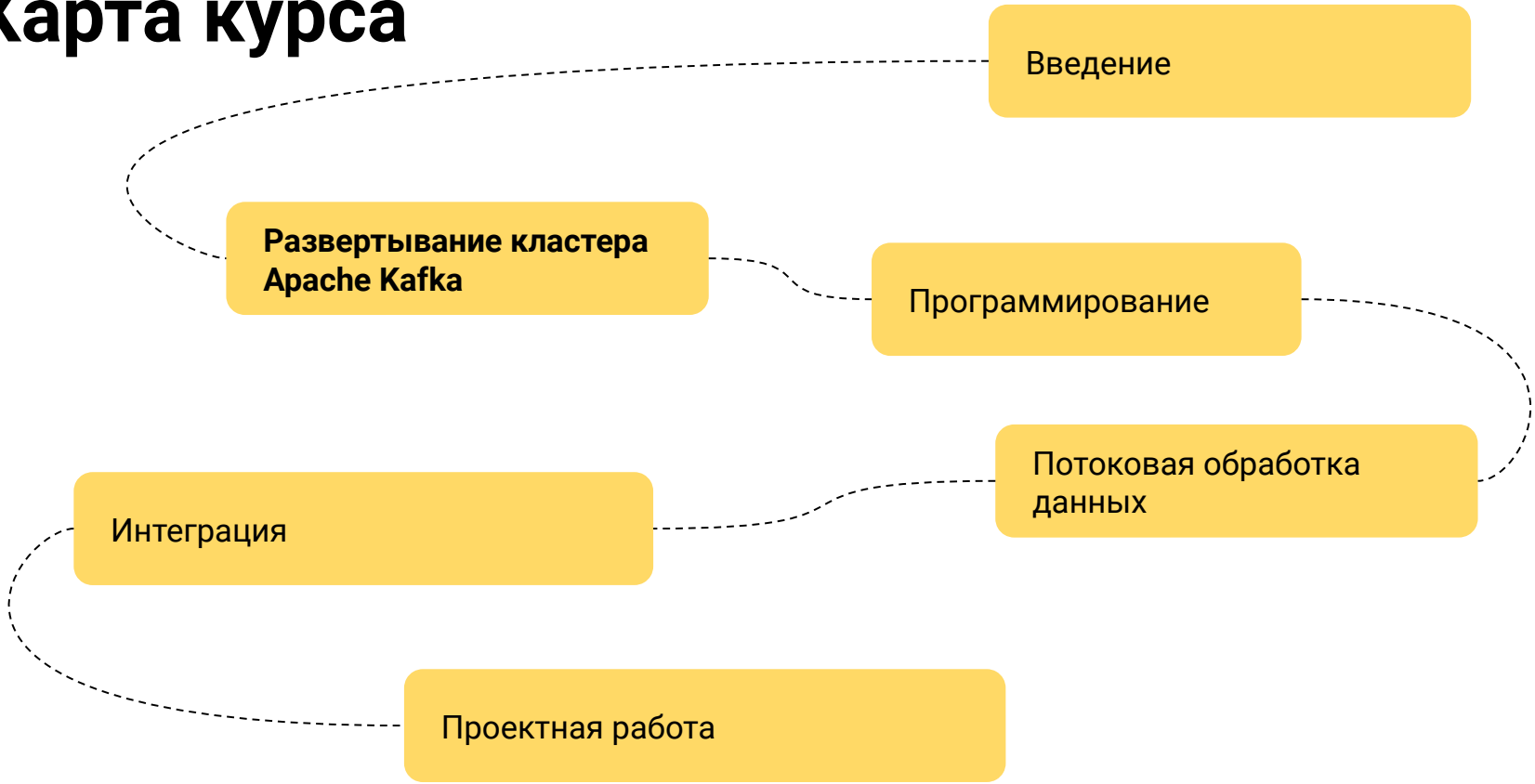
Пишем в чат

Говорим голосом

Документ

Ответьте себе или
задайте вопрос

Карта курса


Введение
Развертывание кластера Apache Kafka
Программирование
Потоковая обработка данных
Интеграция
Проектная работа

Маршрут вебинара

- Что такое топик
- Операции с топиками
- Хранение сообщений
- Параметры топиков
- Рефлексия

Цели вебинара

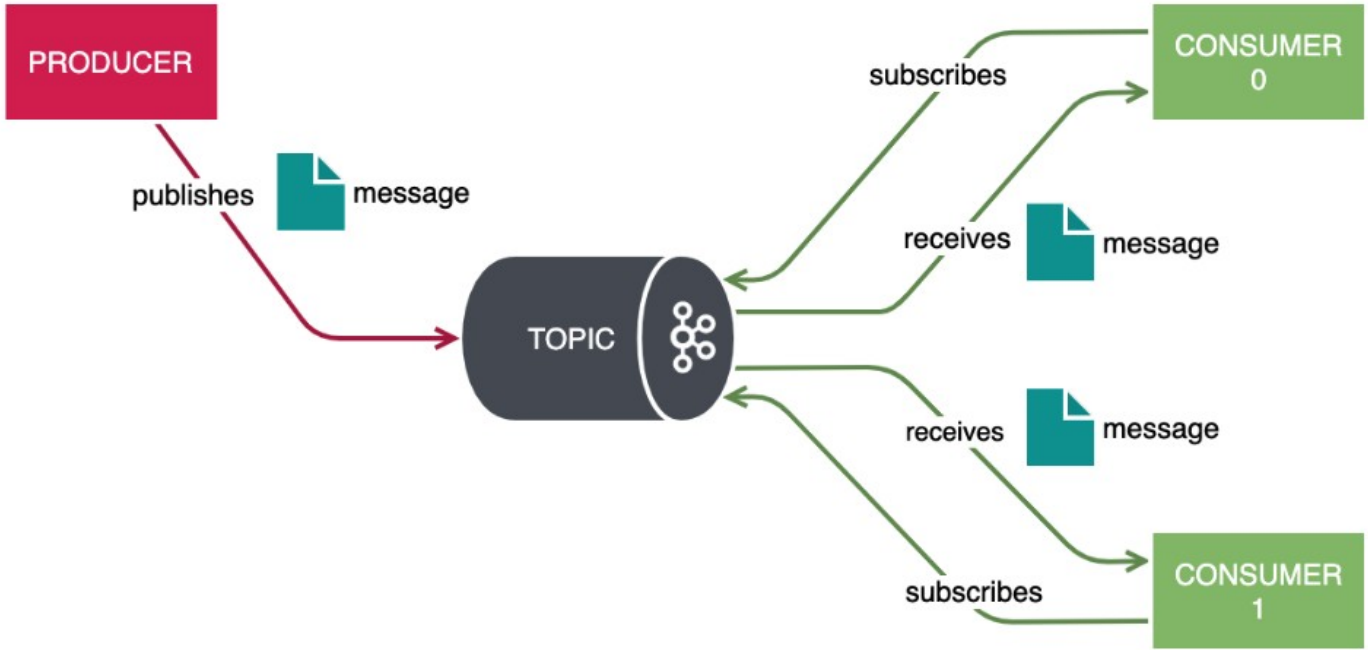
1.	Научимся работать с топиками
2.	Узнаем зачем нужны партиции и репликации
3.	Посмотрим как хранятся сообщения

Смысл

1.	Сможем работать с топиками
2.	Сможем настраивать топики

Топик

Publish / Subscribe

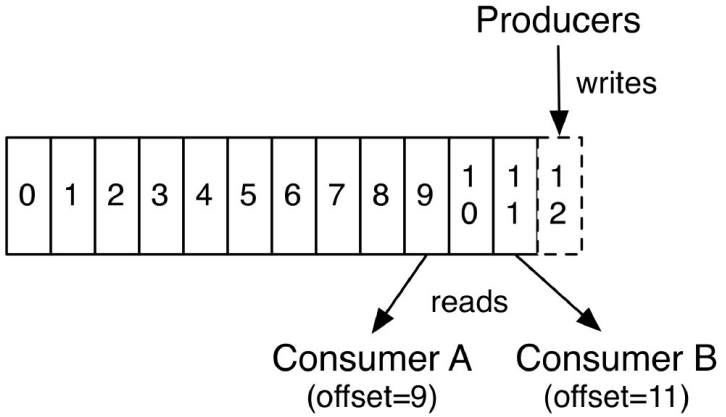


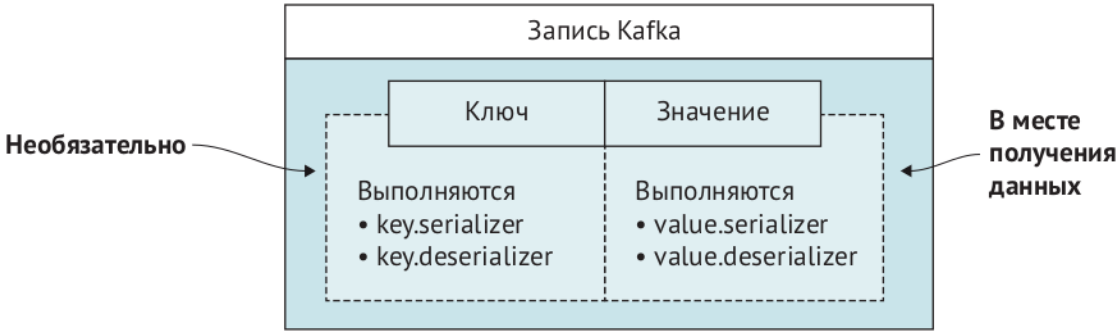
Topic

Topic – имя потока с данными

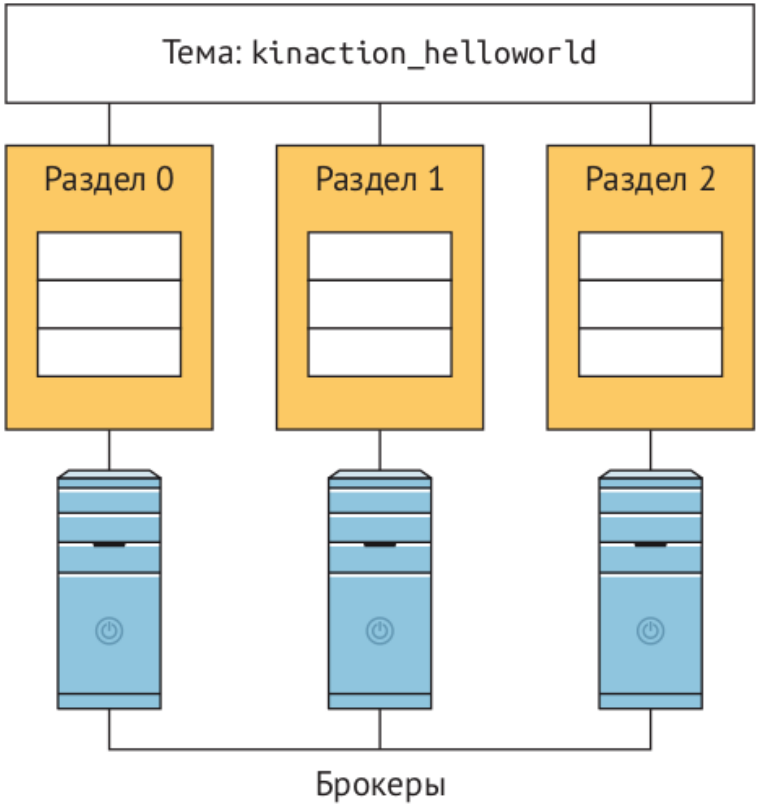
Record – элемент данных типа ключ-значение

Offset – позиция записи





Топики и партиции (разделы, секции)



Основные операции с топиками

`kafka-topics.sh` – работа с топиками:

`--list` — вывести список топиков

`--describe` — вывести описание топиков

`--create` — создать топик

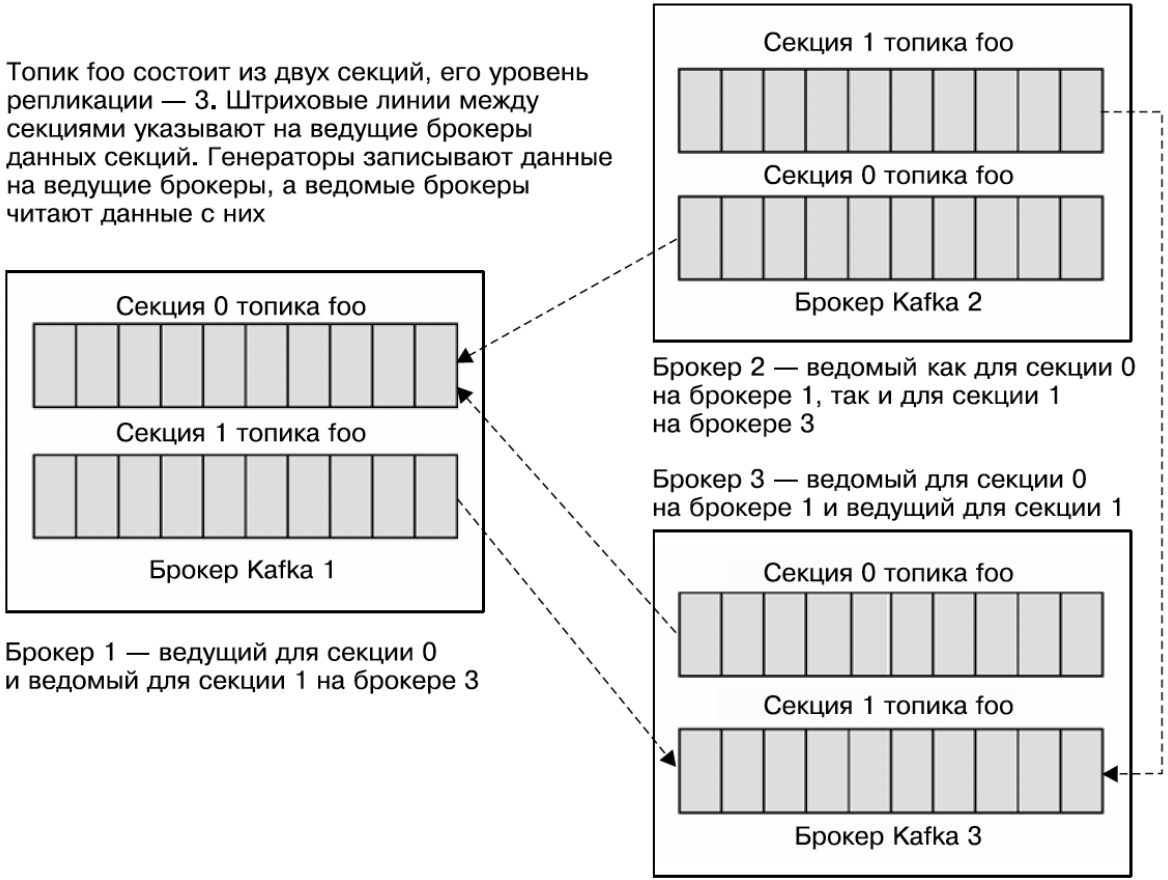
`--delete` — удалить топик

`--alter` — изменить топик

LIVE

Репликация

Репликация

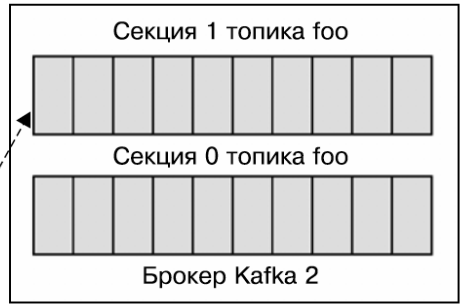

Топик foo состоит из двух секций, его уровень репликации — 3. Штриховые линии между секциями указывают на ведущие брокеры данных секций. Генераторы записывают данные на ведущие брокеры, а ведомые брокеры читают данные с них
Секция 1 топика foo
Секция 0 топика foo
Брокер Kafka 2
Брокер 2 — ведомый как для секции 0 на брокере 1, так и для секции 1 на брокере 3
Брокер 3 — ведомый для секции 0 на брокере 1 и ведущий для секции 1
Секция 0 топика foo
Секция 1 топика foo
Брокер Kafka 1
Брокер 1 — ведущий для секции 0 и ведомый для секции 1 на брокере 3
Секция 0 топика foo
Секция 1 топика foo
Брокер Kafka 3

Поведение при отказе брокера

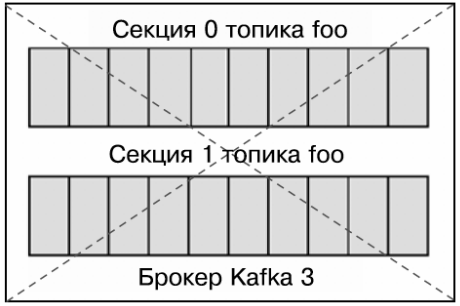
Топик foo состоит из двух секций, его уровень репликации — 3. Изначально у него следующие ведущие и ведомые брокеры:
Брокер 1 — ведущий для секции 0
и ведомый для секции 1
Брокер 2 — ведомый для секции 0 и секции 1
Брокер 3 — ведомый для секции 0
и ведущий для секции 1

Брокер 3 перестал реагировать
на контрольные сигналы ZooKeeper

Шаг 1: будучи ведущим, брокер 1 обнаружил сбой брокера 3



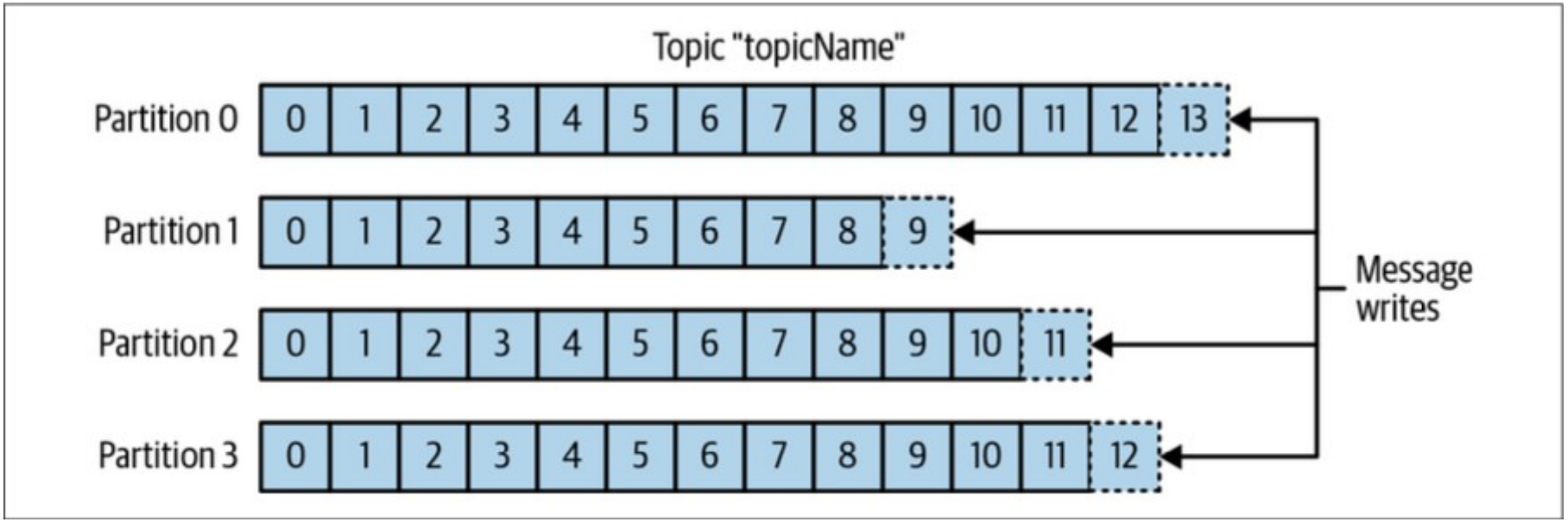
Шаг 2: контроллер делает ведущим для секции 1 брокер 2 вместо брокера 3. Все записи секции 1 теперь будут попадать на брокер 2, а брокер 1 будет потреблять сообщения для секции 1 с брокера 2



LIVE

Данные в топиках

Запись в топик с партициями



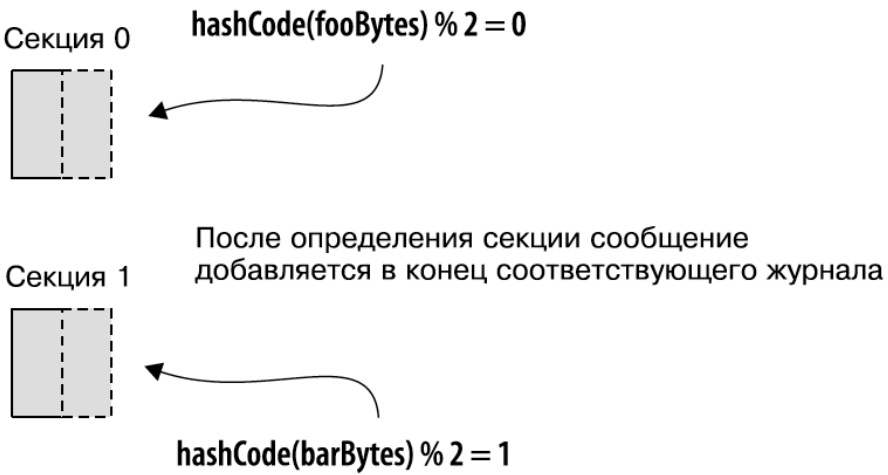
Партиции группируют данные по ключу

Входящие сообщения:

{foo, данные сообщения}
{bar, данные сообщения}

Секция, в которую будет отправлено сообщение, определяется его ключами. Эти ключи не пусты.

Байтовое представление ключа используется для вычисления хеша



Журналы

- `log.dir` – место хранения журналов
- Подкаталог – топик
- Кол-во подкаталогов – кол-во партиций
- Формат названия: имя-топика_номер-партиции

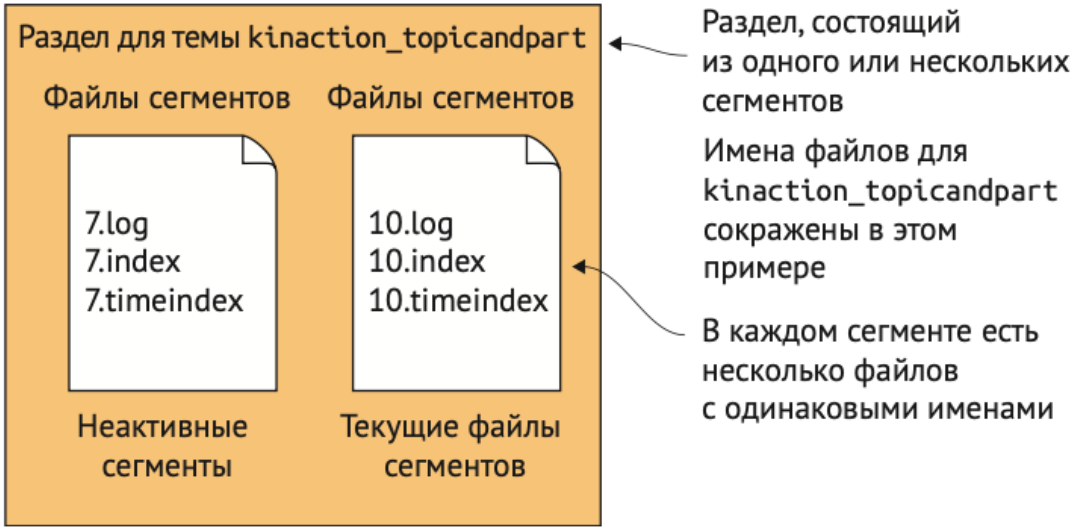
```
/logs
    /logs/topicA_0        В topicA — одна секция
    /logs/topicB_0        В topicB — три секции
    /logs/topicB_1
    /logs/topicB_2
```

Параметры хранения

Ключ	Назначение
`log.retention.bytes`	Максимальный размер журнала в байтах, при превышении которого будет производиться удаление данных
`log.retention.ms`	Максимальная продолжительность хранения данных в журнале в миллисекундах
`log.retention.minutes`	Максимальная продолжительность хранения данных в журнале в минутах. Если в конфигурации установлен параметр `log.retention.ms`, то используется он, а параметр `log.retention.minutes` игнорируется
`log.retention.hours`	Максимальная продолжительность хранения данных в журнале в часах. Если в конфигурации установлен параметр `log.retention.ms` и/или `log.retention.minutes`, то значение `log.retention.hours` игнорируется и используется параметр с наибольшей точностью

Сегменты журналов

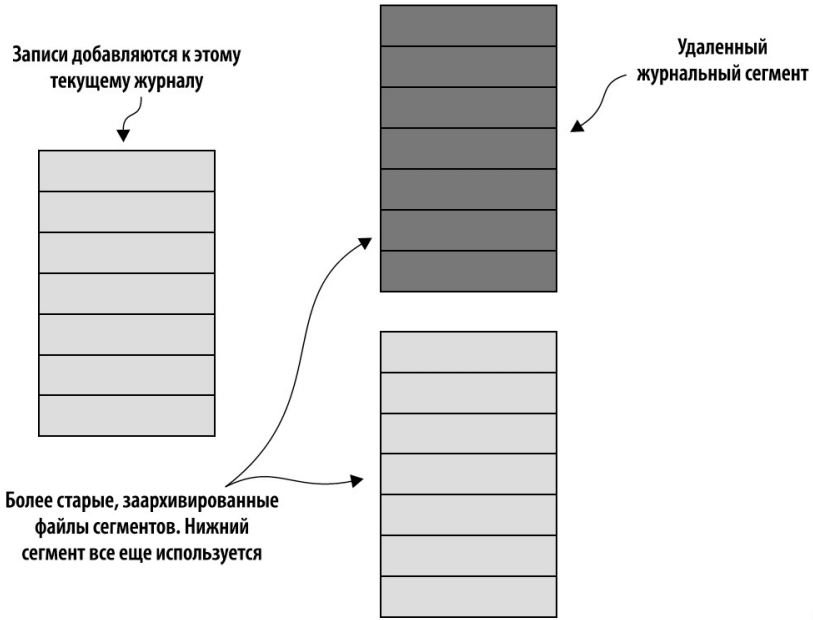
- `.log` – данные
- `.index` и `.timeindex` – хранения соответствий между смещением сообщения и физической позицией внутри индексного файла



Удаление журналов

Двухэтапная стратегий удаления:

- архивация сегментов
- удаление старых сегментов



Операции с журналами

`kafka-log-dirs.sh` — информация о журналах

`kafka-get-offsets.sh` — информация о смещениях

`kafka-dump-log.sh` — просмотр содержимого журналов

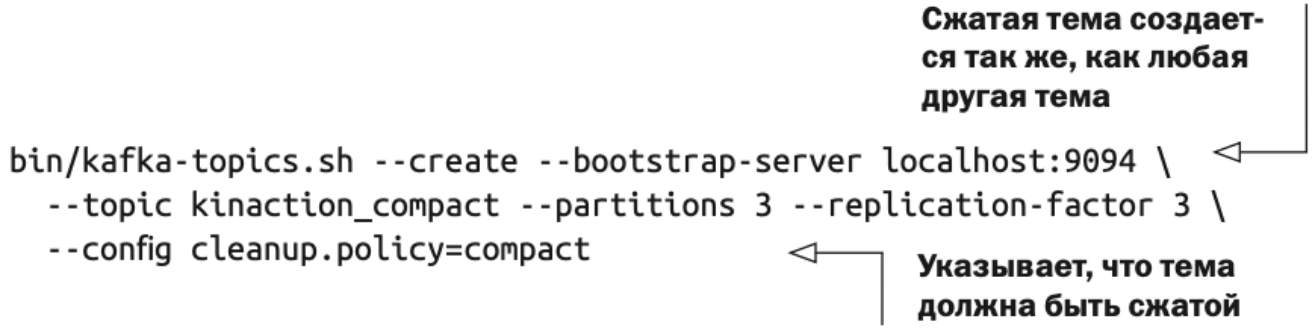
`kafka-delete-records.sh` — удаление записей

LIVE

Сжатие топиков

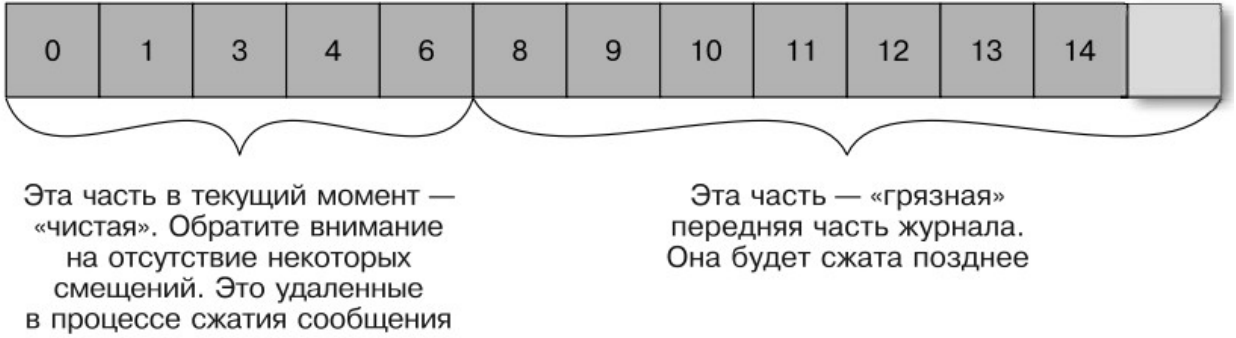
Сжатие топиков

- Стратегии хранения:
  - `delete` – удаление событий, чей возраст превышает срок хранения
  - `compact` – **сохранение только последних значений для каждого ключа**
- Сжатие возможно только для тем с непустыми ключами

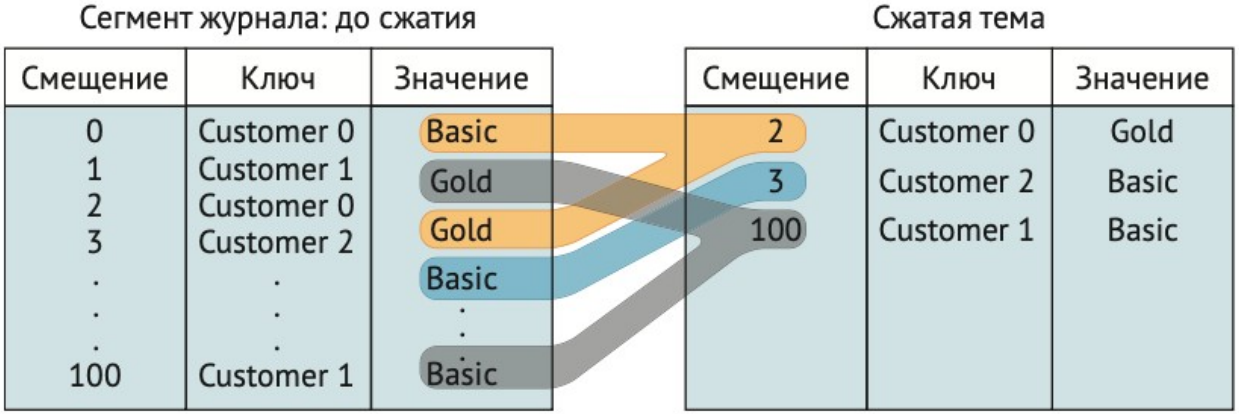


Как происходит сжатие

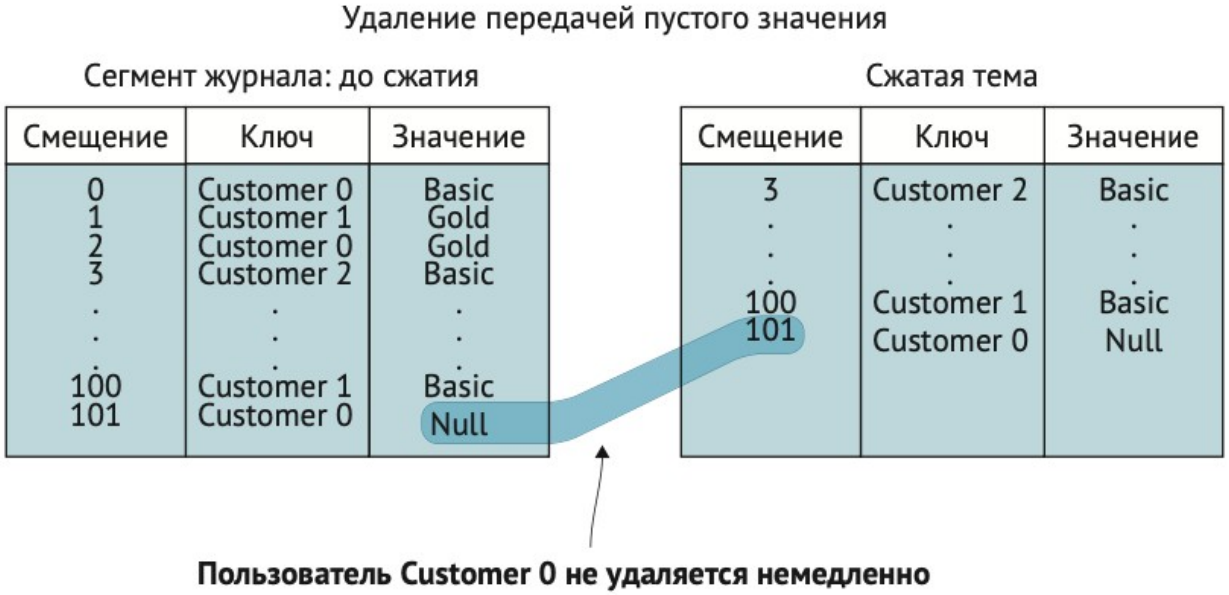
- Каждый из журналов делится на две части:
  - «чистую» — сжатые ранее сообщения. Она содержит только по одному значению для каждого ключа — последнему на момент предыдущего сжатия
  - «грязную» — сообщения, записанные после последнего сжатия



Пример


Сегмент журнала: до сжатия
Смещение
Ключ
Значение
0
1
2
3
100
Customer 0
Customer 1
Customer 0
Customer 2
Customer 1
Basic
Gold
Gold
Basic
Basic
Сжатая тема
Смещение
Ключ
Значение
2
3
100
Customer 0
Customer 2
Customer 1
Gold
Basic
Basic

Удаление события



Когда выполняется сжатие

- Сжимаются только неактивные сегменты
- Сжатие начинается, когда 50% топика содержат «грязные» данные (по умолчанию)
- Параметры времени сжатия:
  - `min.compaction.lag.ms` — минимальная задержка после записи сообщения до его сжатия
  - `max.compaction.lag.ms` — максимальная задержка между моментом записи сообщения и моментом, когда оно пригодно для сжатия

Параметры топиков

Задание параметров топиков

- `kafka-topics.sh --config <String: name=value>` – при создании топика
- `kafka-configs.sh` — динамическое изменение параметров:
  - `--describe` — вывести параметр
  - `--alter --add-config` — изменить параметр
  - `--alter --delete` — удалить параметр
- `https://kafka.apache.org/documentation/#topicconfigs`

Параметры топиков

cleanup.policy	При значении compact сообщения топика будут отбрасываться и из сообщений с заданным ключом будет сохраняться только самое последнее (сжатые журналы)
compression.type	Тип сжатия, используемый брокером при записи на диск пакетов сообщений для данного топика
delete.retention.ms	Длительность (в миллисекундах) хранения отметок об удалении для данного топика. Имеет смысл только для топиков со сжатием журналов

Параметры топиков

file.delete.delay.ms	Длительность (в миллисекундах) ожидания перед удалением сегментов журнала и индексов для данного топика с диска
flush.messages	Количество сообщений, которое может быть получено, прежде чем будет выполнен принудительный сброс сообщений данного топика на диск
flush.ms	Промежуток времени (в миллисекундах) перед принудительным сбросом сообщений данного топика на диск
max.compaction.lag.ms	Максимальное время, в течение которого сообщение не будет сжиматься в журнале
max.message.bytes	Максимальный размер отдельного сообщения данного топика (в байтах)

Параметры топиков

min.cleanable.dirty.ratio	Частота попыток сжатия разделов 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.compaction.lag.ms	Минимальное время, в течение которого сообщение будет оставаться в журнале в несжатом виде
min.insync.replicas	Минимальное число согласованных реплик, необходимое для того, чтобы раздел топика считался доступным

Параметры топиков

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 разделе
segment.index.bytes	Максимальный размер (в байтах) отдельного индекса сегмента журнала

LIVE

Вопросы?

Ставим “+”,
если вопросы есть

Ставим “–”,
если вопросов нет

Резюме

Подведем итоги

1.	Топики — это логические, а не физические структуры
2.	Топики состоят из партиций, что обеспечивает возможность параллельной обработки данных
3.	Сегменты файлов журнала записываются в каталоги партиций
4.	Сжатые топики позволяют хранить последнее значение определённого события

Рефлексия

Рефлексия

С какими впечатлениями уходите с вебинара?

Как будете применять на практике то,
что узнали на вебинаре?

Следующий вебинар

Мониторинг кластера Kafka

- Ссылка на вебинар будет в ЛК за 15 минут
- Материалы к занятию в ЛК — можно изучать
- Обязательный материал обозначен красной лентой

Заполните, пожалуйста, опрос о занятии по ссылке в чате

Спасибо за внимание!

Приходите на следующие вебинары

Заигрин Вадим

Ведущий эксперт по технологиям, Сбербанк

vzaigrin@yandex.ru
https://t.me/vzaigrin