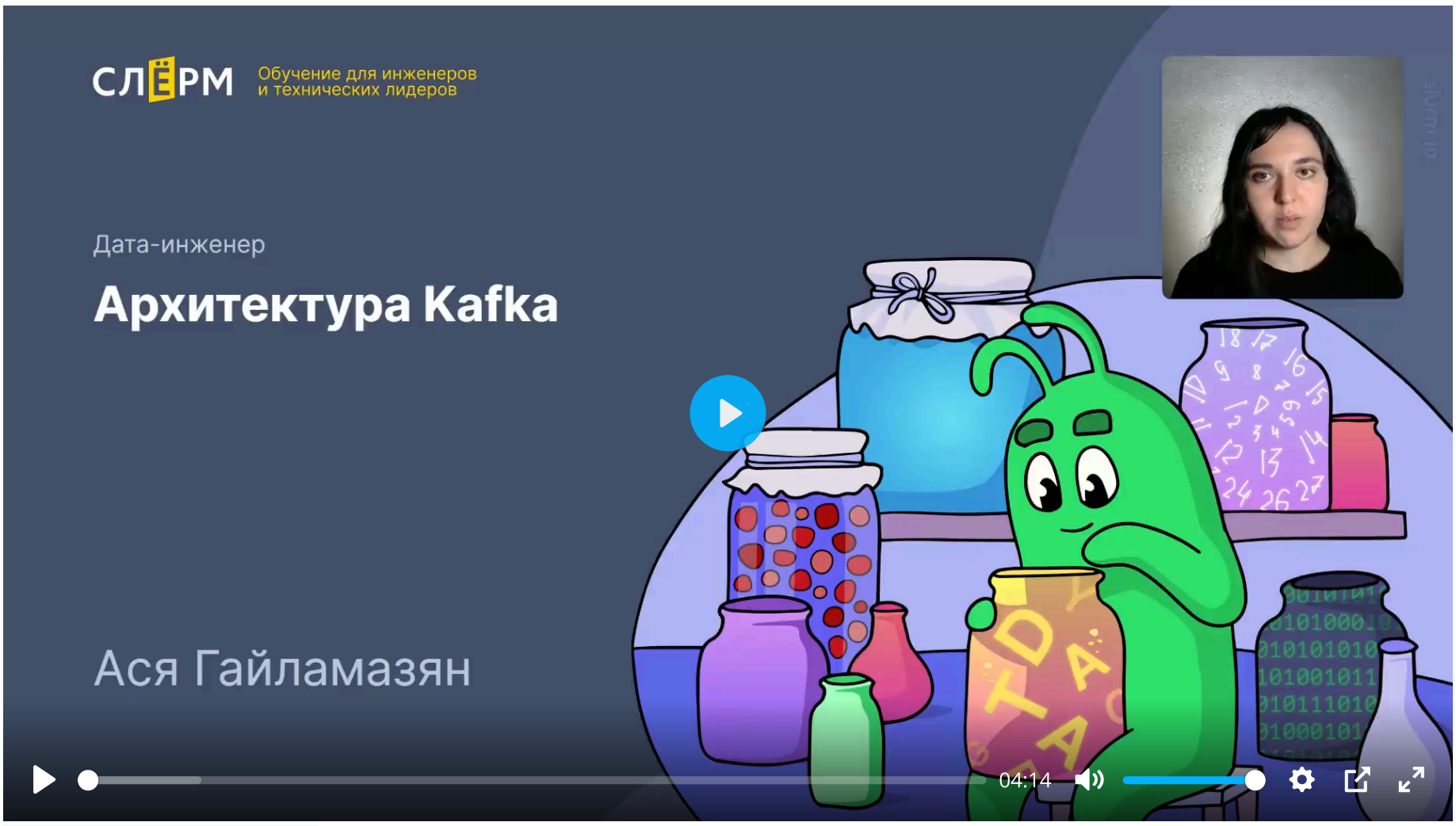
9.3 Kafka - знакомство и архитектура



Презентация к уроку 9.3.2

Текстовая расшифровка видео:

АРХИТЕКТУРА KAFKA

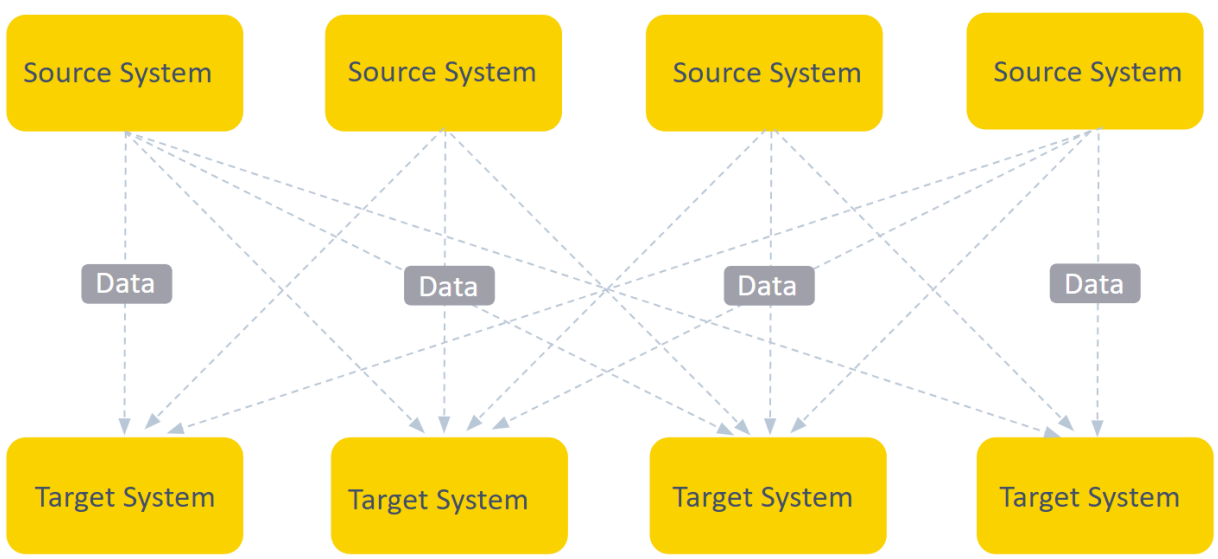
План:

- Интеграции;
- Общая интеграция с Kafka;
- Архитектура инстанса Kafka;
- Архитектура Kafka;
- Репликация.

Интеграции

Обычно перед компаниями встает вопрос, когда у них есть большое количество источников и получателей данных, о том, чтобы интегрировать эти данные друг с другом.

В данном примере 14 интеграций между источниками и получателями:



Также нужно учитывать, что бывают разные протоколы данных, HTTP, TCP, FTP, разные форматы данных (JSON, XML, CSV и др.). Периодические источники могут менять свою конфигурацию, из-за чего нужно переписывать всю интеграцию. Для решения этого вопроса был придуман Apache Kafka.

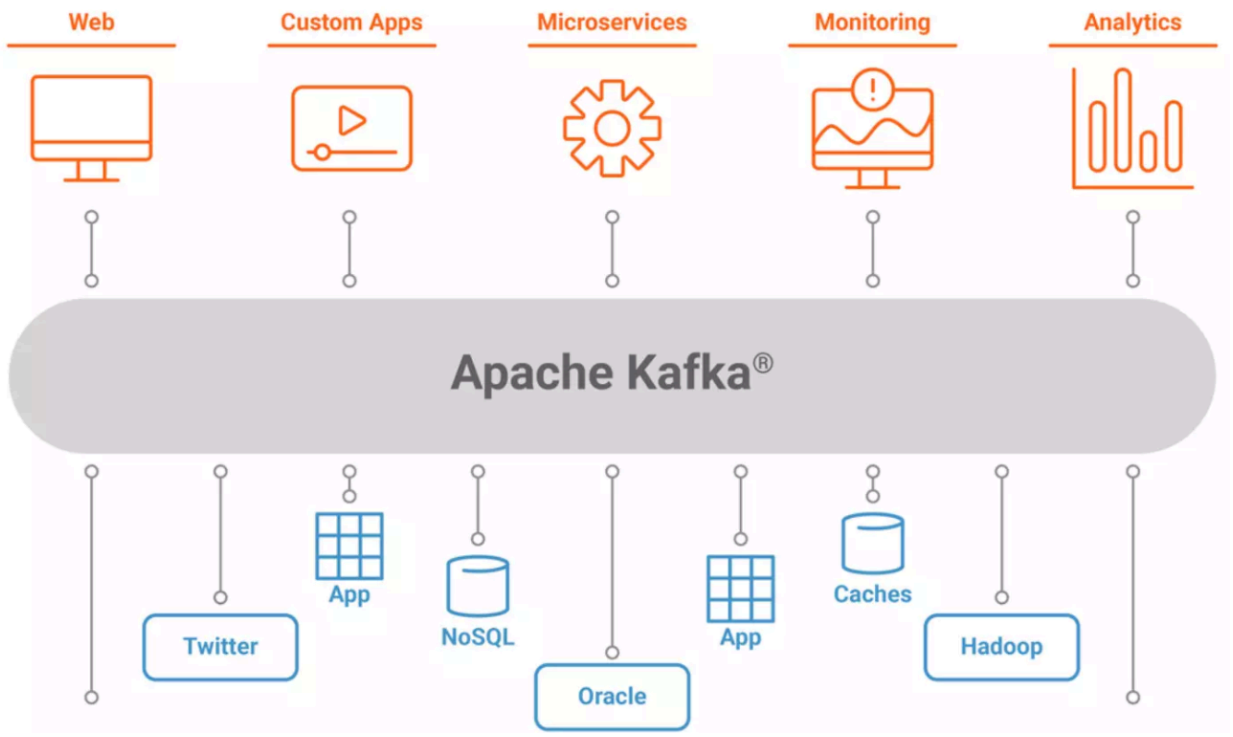
Общая интеграция с Kafka

Apache Kafka решает этот вопрос тем, что позволяет не писать все эти коннекторы, а подключаться к единому источнику Apache Kafka напрямую.

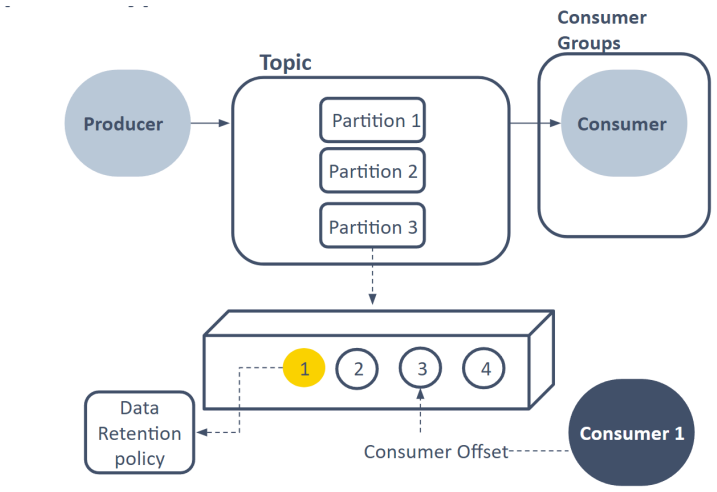
У Apache Kafka минимальные задержки, поэтому его присутствие в системе некритично (задержки до 10 микросекунд).

Apache Kafka поддерживает распределенную архитектуру. Он устойчив к падениям благодаря репликации.

Apache Kafka – это Opensource-продукт.



Архитектура инстанса Kafka



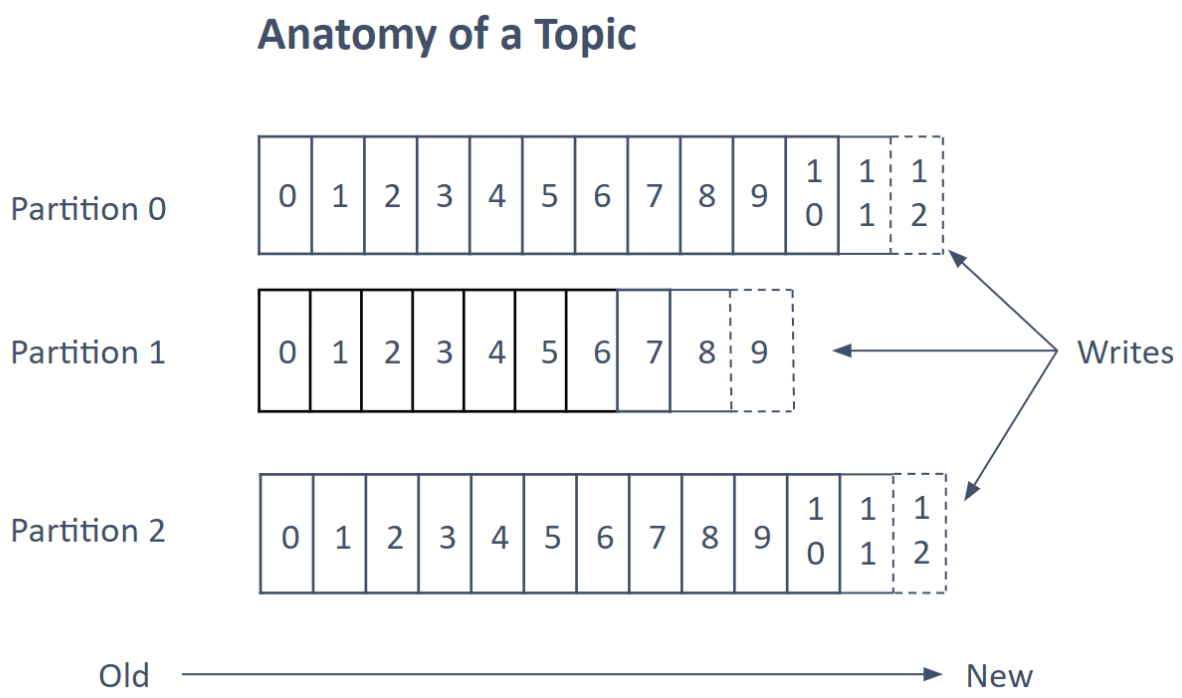
Сообщения приходят с неким ключом, со значением, с timestamp’ом, с информацией с метаданными в заголовке (опционально). Это сообщение кладется в очередь «Topic».

В Topic’ах хранятся очередь сообщений, разделенная на партиции. Когда топик создается, создается количество партиций (оно может меняться). Инстанс может быть распределенным по серверам. Для одного инстанса Kafka может быть неограниченное количество топиков, но не более 200 000 partition’ов.

Partition’ы в сообщение кладутся по порядку. Этот порядок там и сохраняется. Эти сообщения считываются консьюмером и не удаляются.

Можно настроить хранение. Хранение в Kafka по умолчанию – неделя.

Архитектура Kafka



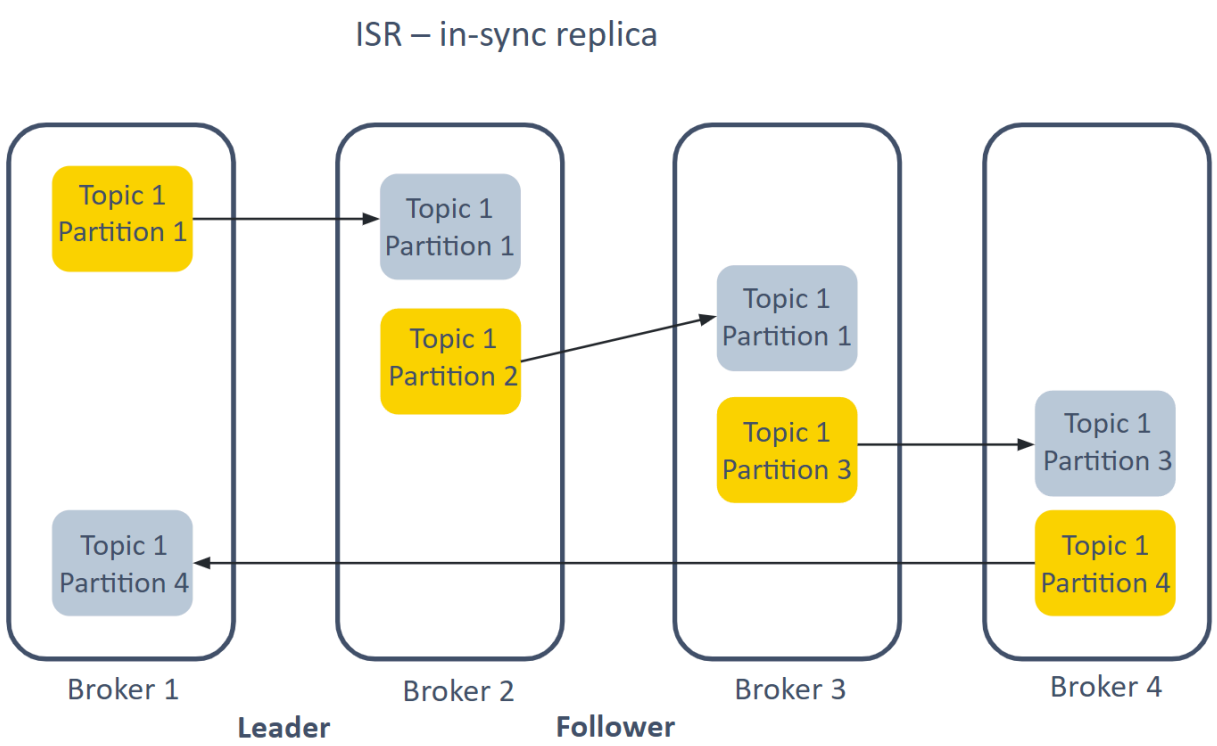
Partition’ы представляют собой rey-лист, в котором есть некий офсет начала хранения данных.

Офсет инкрементный и формируется с нуля. Каждый раз, когда добавляются новые сообщения, они не перетирают старые, а добавляются к концу partition’а.

Офсет для каждого partition’а уникален: если данные с офсета «0» в partition «0», то в partition «1» с офсета «0» будут храниться другие данные.

Следует отметить, что данные пишутся по правилу «Fifo», то есть, если они первые пришли, то первыми и считываются.

Репликация



Устойчивость Kafka поддерживается с помощью механизма репликации. Задается коэффициент репликации, который указывает, сколько уникальных реплик должно быть.

В Kafka репликации происходят на уровне partition’ов, топиков. То есть, каждый partition каждого топика будет реплицироваться отдельно. При репликации Kafka гарантирует, что partition’ы хранятся на разных брокерах (уникальные реплики одного partition’а будут размещены по разным брокерам). Таким образом, коэффициент репликации может быть не больше, чем количество брокеров, не больше, чем количество серверов.

Так происходит реплицирование master-реплики. Master – основной partition, с которым происходит работа консьюмера и продюсера, чтение и запись данных. После того, как данные меняются на master’е, они асинхронно (in-sync replica) подъезжают на реплике.

С Kafka 2.4 появилась возможность чтения консьюмером из реплик тоже.

В случае каких-либо сбоев переизбираются master-реплики (из работоспособных выбирается новая master-нода, так называемая «Master Partition» и ее реплики.

Как вам урок? 😠 😐 🔥

Изучил, далее >