МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

Введение в Kafka

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

Что такое Apache Kafka

Основные понятия

Особенности Kafka.

Kafka — это система обмена сообщениями, разработанная как быстрая, масштабируемая и надежная.

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

- Это платформа обработки потоков с открытым исходным кодом.
- Создан в LinkedIn, а затем в 2011 году стал проектом Apache с открытым исходным кодом.
- Kafka написана на Scala и Java.
- Он направлен на предоставление платформы с высокой пропускной способностью и малой задержкой для обработки потоков данных в реальном времени.

Apache описывает Kafka как распределенную потоковую платформу, которая позволяет нам:

- Публиковать и подписываться на потоки записей.
- Хранить потоки записей отказоустойчивым способом.
- Обрабатывать потоки записей по мере их появления.

Kafka понятия

Записи(Records)

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Records = message = event

Каждая запись состоит из ключа, значения и метки времени.

Categorized into Topics

Сообщение = массив байтов

Сериализатор не нужен

Темы(Topics)

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Тема — это категория, в которой публикуются записи.

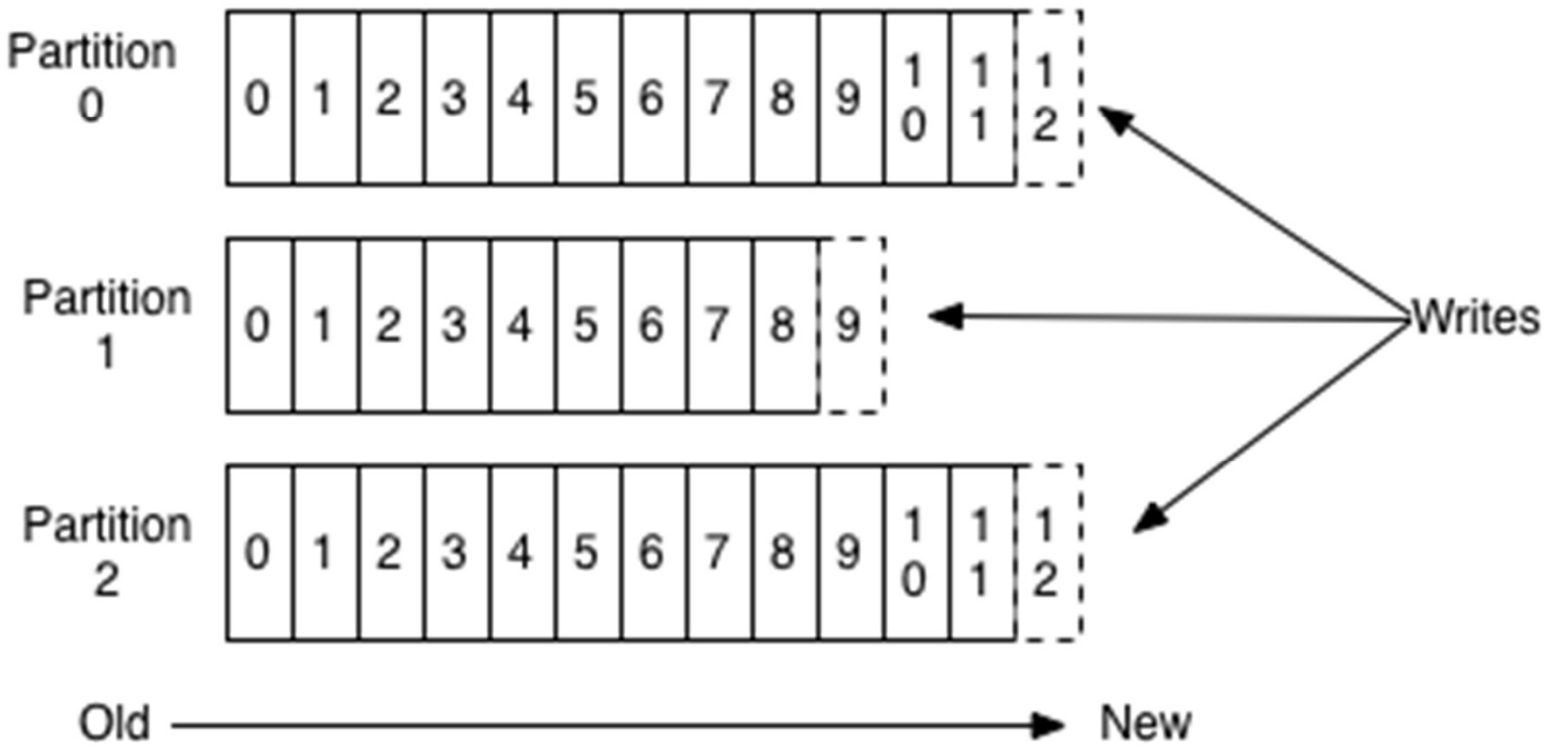
Производители(Producers) записывают данные в темы, а потребители(consumers) читают их.

Темы в Kafka всегда мультиподписные

Темы делятся на разделы.

Темы постоянны — записи сохраняются на диске

Anatomy of a Topic



Source: https://kafka.apache.org/intro

Разделы

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Каждая тема имеет один или несколько разделов

Порядок сохраняется в пределах одного раздела

Разделы реплицируются по всему кластеру

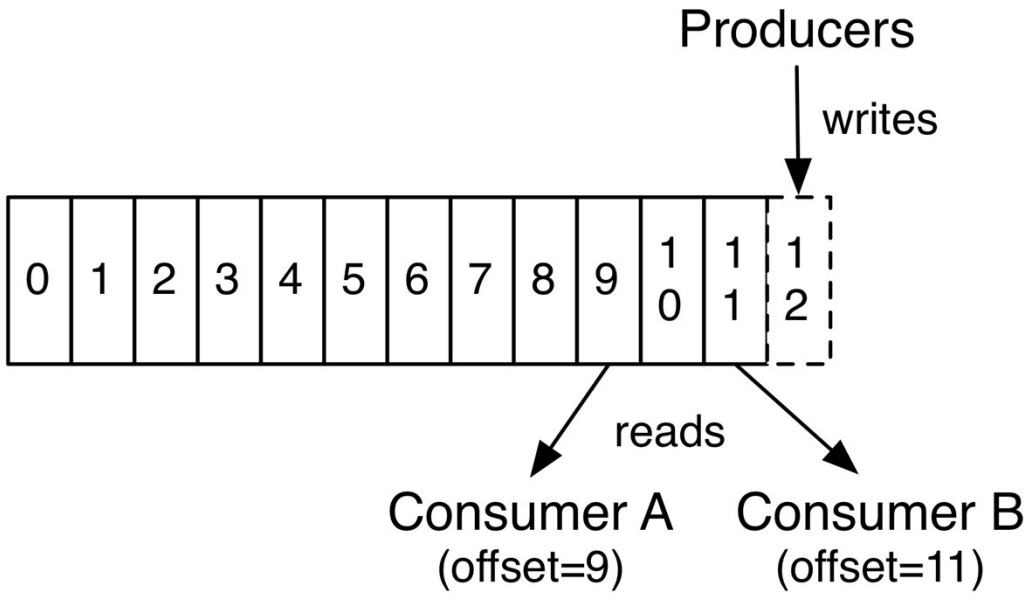
Записи добавляются в самый конец каждого раздела

Смещение раздела(Partition Offset)

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Каждое секционированное сообщение имеет уникальный идентификатор последовательности, который называется смещением.

Смещение уникально только для каждого раздела



Source: https://kafka.apache.org/intro

Реплики(Replicas)

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Реплика – это резервная копия раздела

Фактор репликации сообщает кластеру, сколько одинаковых реплик должно быть создано в кластере.

Реплики никогда не используются для чтения или записи данных.

Они используются для предотвращения потери данных.

Может быть одна или несколько реплик раздела

Один лидер и много последователей

In-Sync Replica – это реплика, содержащая достаточно записей, чтобы учитываться при выборе лидера раздела.

Брокеры(Brokers)

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Kafka Broker — это сервер Kafka, который управляет записями.

- Получает сообщения
- Назначает смещения
- Фиксирует сообщения в хранилище на диске

У каждого брокера может быть ноль или более разделов для каждой темы.

Кластер(Cluster)

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Когда у Kafka более одного брокера, это называется кластером Kafka.

Кластер Kafka можно масштабировать без простоев.

Zookeeper

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Он создает резервные копии критически важных метаданных Kafka Cluster.

- Ответственный за выбор очереди
- Сохраняет ACL(Списки контроля доступа)
- Хранит список доступных брокеров Kafka

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

Kafka
Особенности

Мульти аренда

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Вы можете развернуть Kafka как многопользовательское решение.

Мульти аренда включается путем настройки тем, которые могут создавать или потреблять данные.

Поддержка квот — администраторы могут определять квоты для управления ресурсами брокера, используемыми клиентами.

Geo Replication

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Kafka MirrorMaker обеспечивает поддержку георепликации.

- Сообщения реплицируются в нескольких центрах обработки данных или облачных регионах;
- активные/пассивные сценарии резервного копирования и восстановления;
- активные/активные сценарии для размещения данных ближе к вашим пользователям.

Guarantees

На высоком уровне Kafka дает следующие гарантии:

- Сообщения, отправленные производителем в определенный раздел Темы, будут добавляться в порядке их отправки.
- Потребитель видит записи в том порядке, в котором они хранятся в журнале.
- Тема с коэффициентом репликации N, мы допустим до N-1 сбоев сервера.

Удержание

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

Сообщение остается в теме для:

Конфигурации периода времени или
Достижения настраиваемого размера

Хранения каждой Темы.

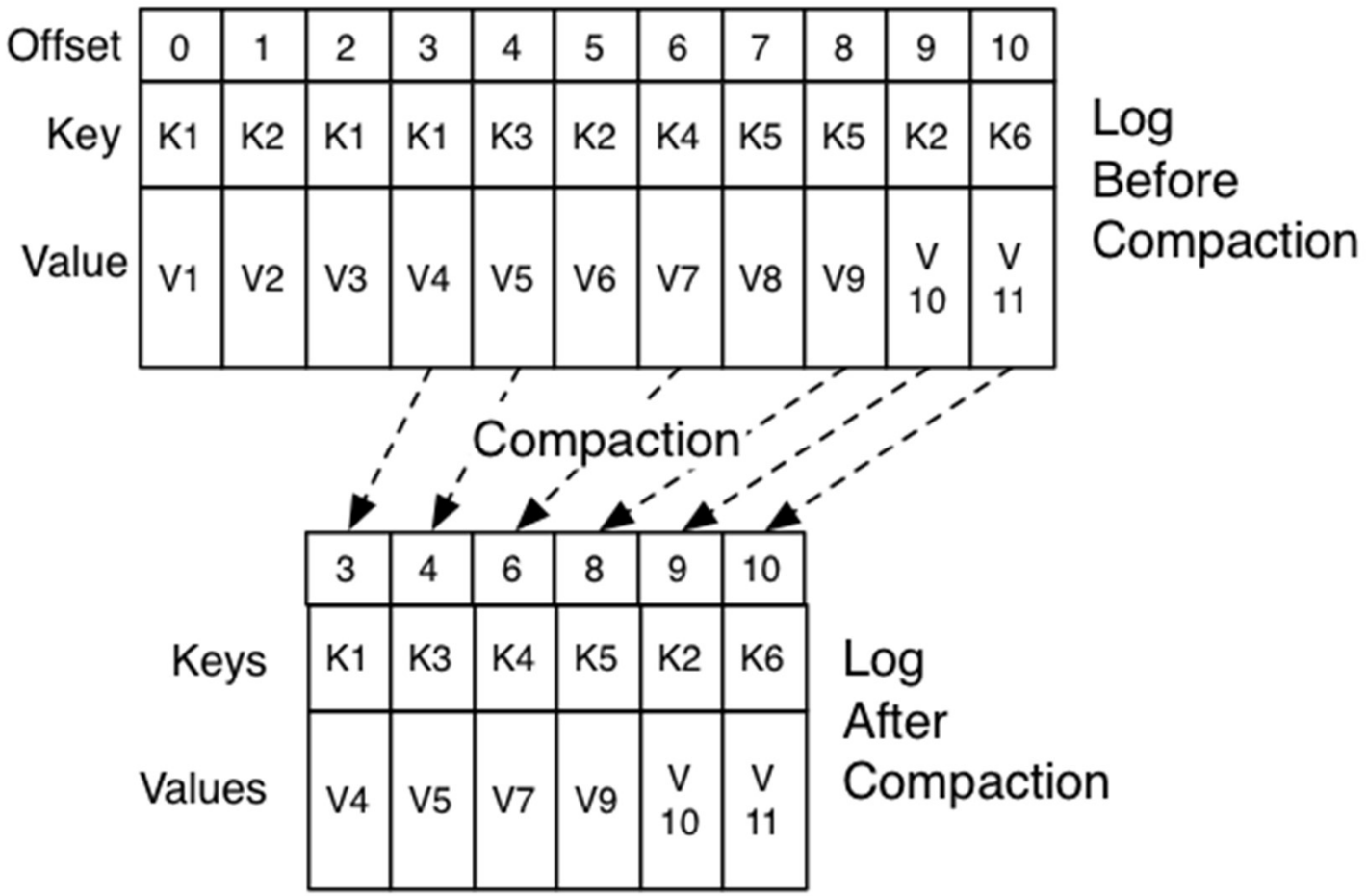
Сжатие логов

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

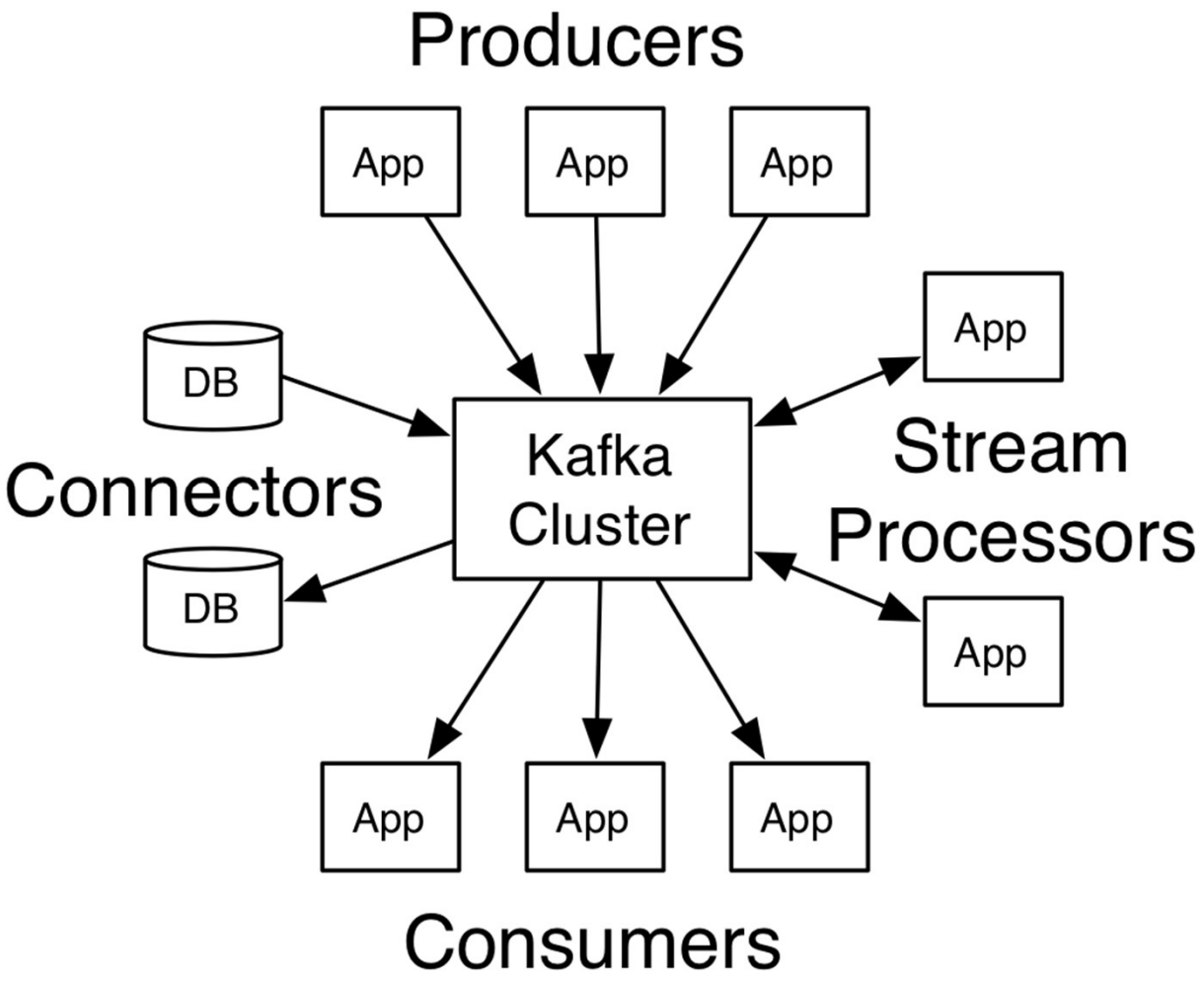
Сохраняет по крайней мере последнее известное значение для каждого ключа записи для одного тематического раздела.

Сжатые журналы полезны для восстановления состояния после сбоя или сбоя системы.

Используется также для удаления одиночного сообщения



Source: https://kafka.apache.org/intro



Source: https://kafka.apache.org/intro

Producer API

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Producer API позволяет приложению публиковать поток записей в одной или нескольких темах Kafka.

Consumer API

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Consumer API позволяет приложению подписываться на одну или несколько тем и обрабатывать поток создаваемых для них записей.

Streams API

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Streams API позволяет приложению выступать в качестве потокового процессора, потребляя входной поток из одной или нескольких тем и создавая выходной поток для одной или нескольких выходных тем, эффективно преобразовывая входные потоки в выходные потоки.

Connector API

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Connector API позволяет создавать и запускать многоразовых производителей или потребителей, которые связывают темы Kafka с существующими приложениями или системами данных. Например, коннектор к реляционной базе данных может фиксировать каждое изменение в таблице.

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

Введение в Strimzi

Strimzi

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ
МГПУ

- Какие проблемы решает Strimzi
- Определение
- Концепция

Какие проблемы решает Strimzi

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

- Обновление Kafka
- Первоначальное развертывание
- Управление ZooKeeper
- Замена брокеров
- Перебалансировка Темы
- Выведение из эксплуатации или добавление брокеров

Strimzi

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Kafka на основе оператора в проекте Kubernetes

Strimzi

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Strimzi — это open source проект Apache Kafka с открытым исходным кодом для Kubernetes и OpenShif.

Представлен 25 февраля 2018 г.

Разработан на основе проекта, известного как Barnabas, Паоло Патьерно, Red Hat.

Часть программы разработчиков Red Hat

Компонент “Streams” Red Hat AMQ, коммерческого продукта технологий обмена сообщениями Red Hat.

Strimzi

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Детали проекта:

Apache Kafka project for Kubernetes and OpenShift

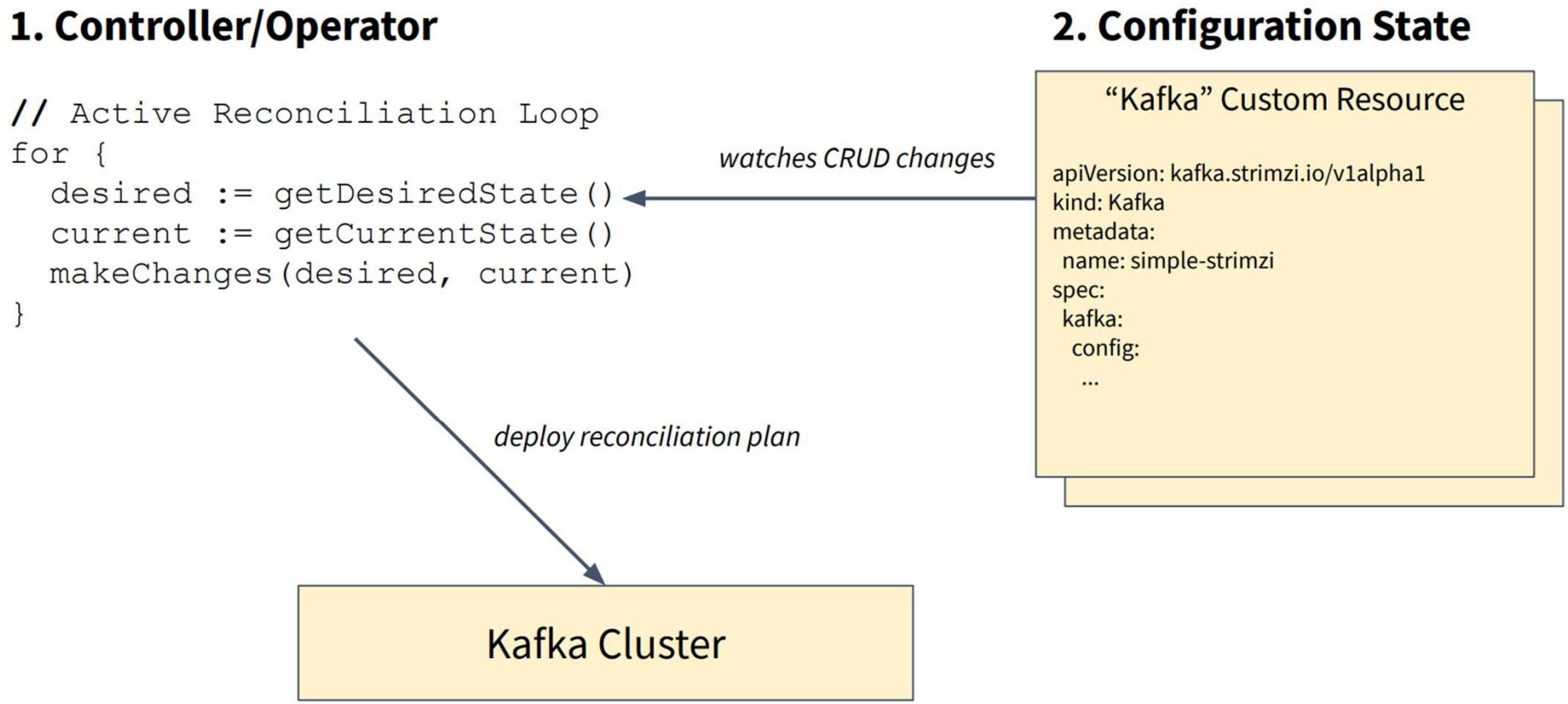
Licensed under Apache License 2.0

Web site: http://strimzi.io/

GitHub: https://github.com/strimzi

Strimzi – Operator Pattern

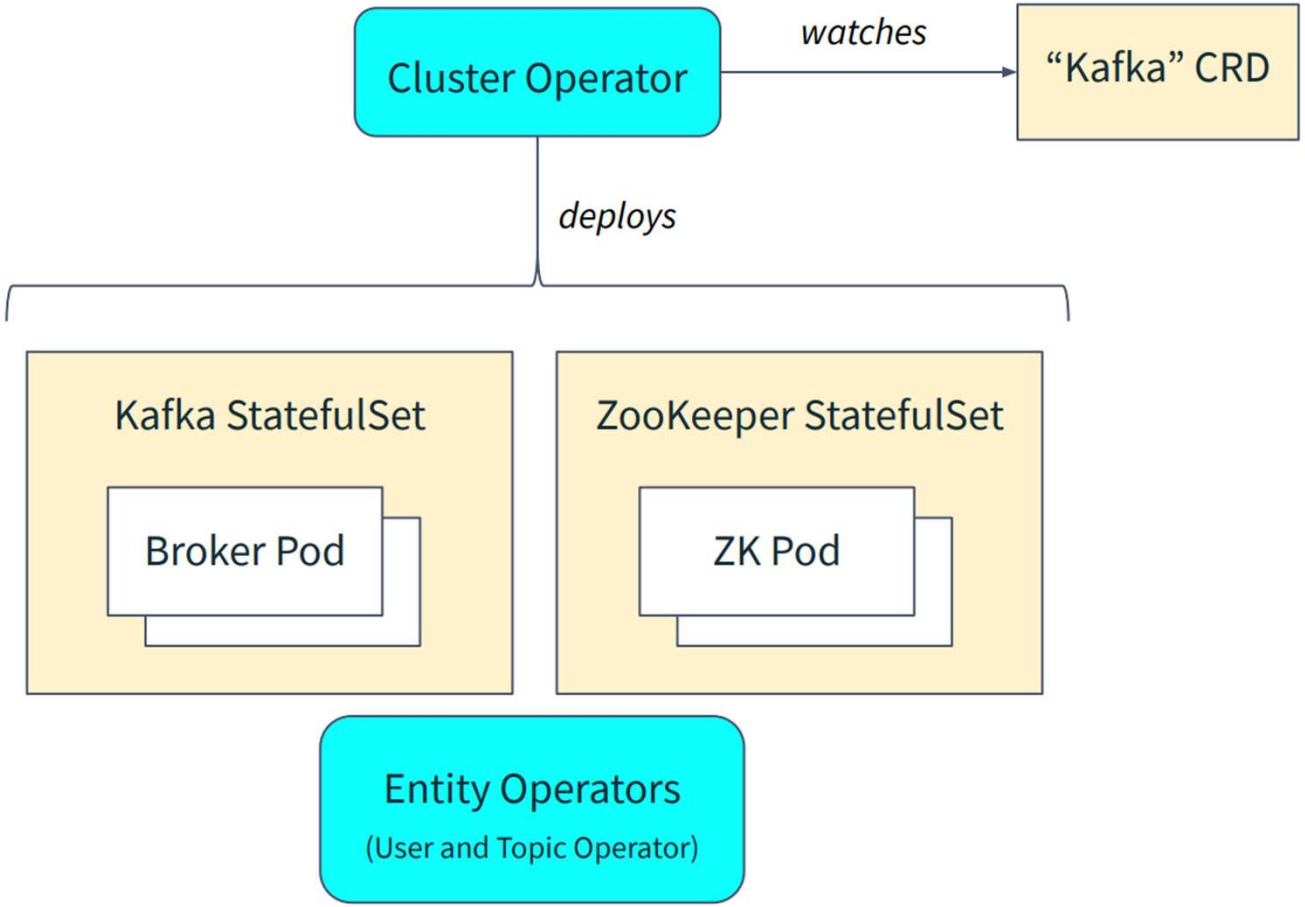
МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ



Source: https://www.slideshare.net/Lightbend/running-kafka-on-kubernetes-with-strimzi-for-realtime-streaming-applications

Strimzi – Cluster Operator

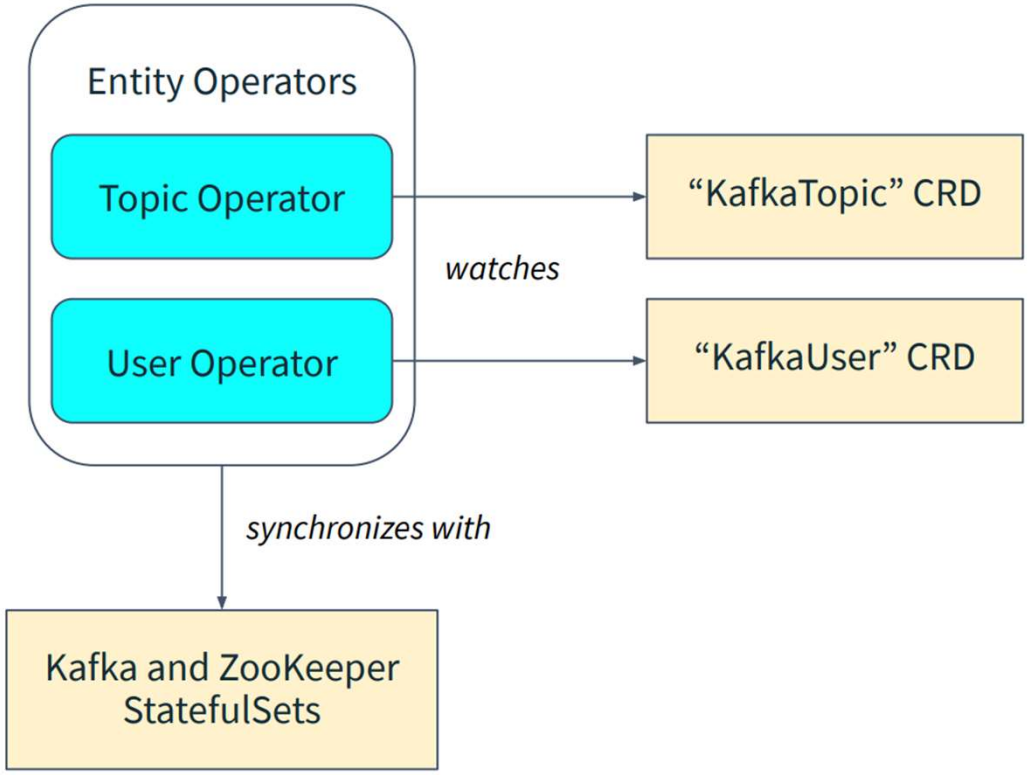
МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ



Source: https://www.slideshare.net/Lightbend/running-kafka-on-kubernetes-with-strimzi-for-realtime-streaming-applications

Strimzi – User and Topic Operators

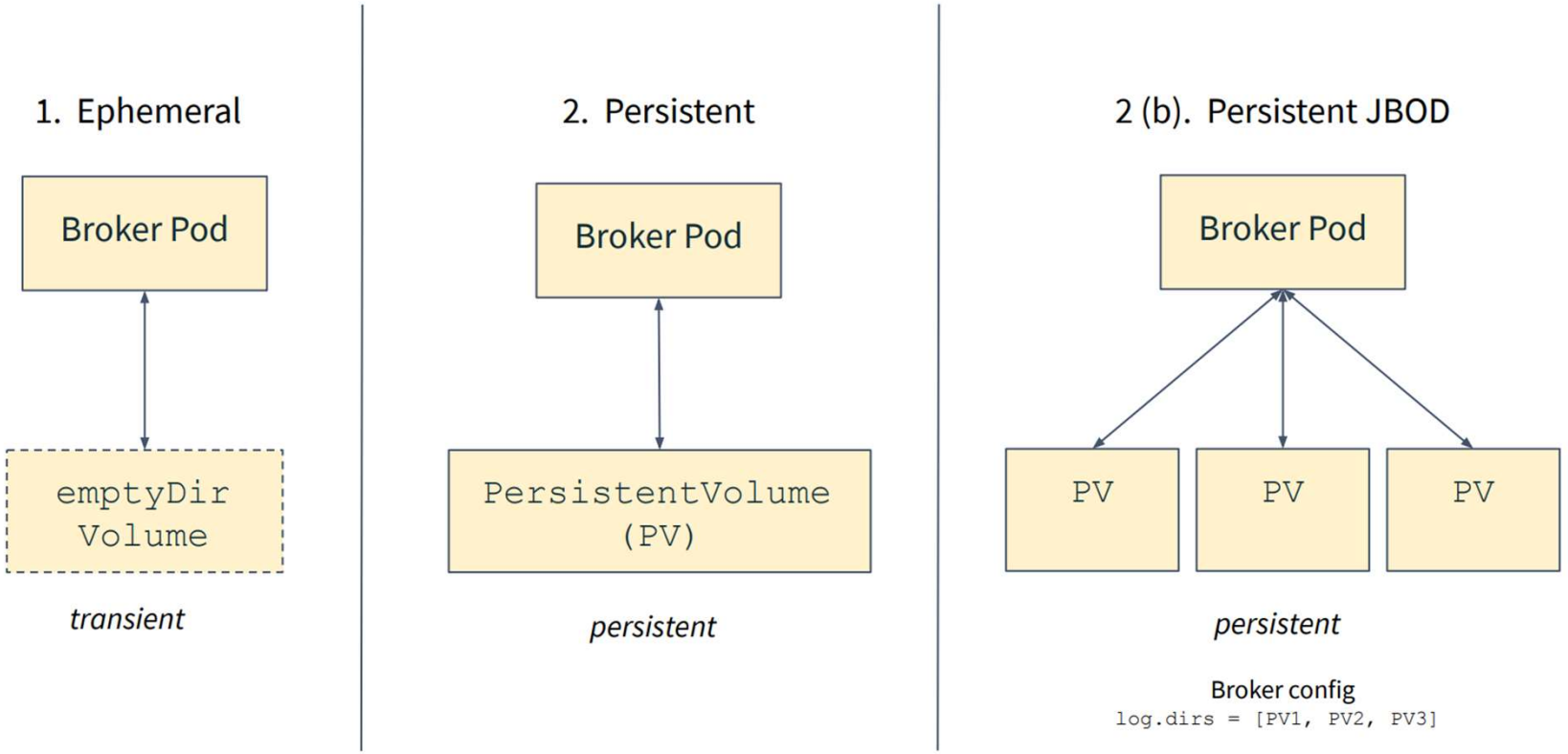
МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ



Source: https://www.slideshare.net/Lightbend/running-kafka-on-kubernetes-with-strimzi-for-realtime-streaming-applications

Strimzi – Режимы хранения

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ



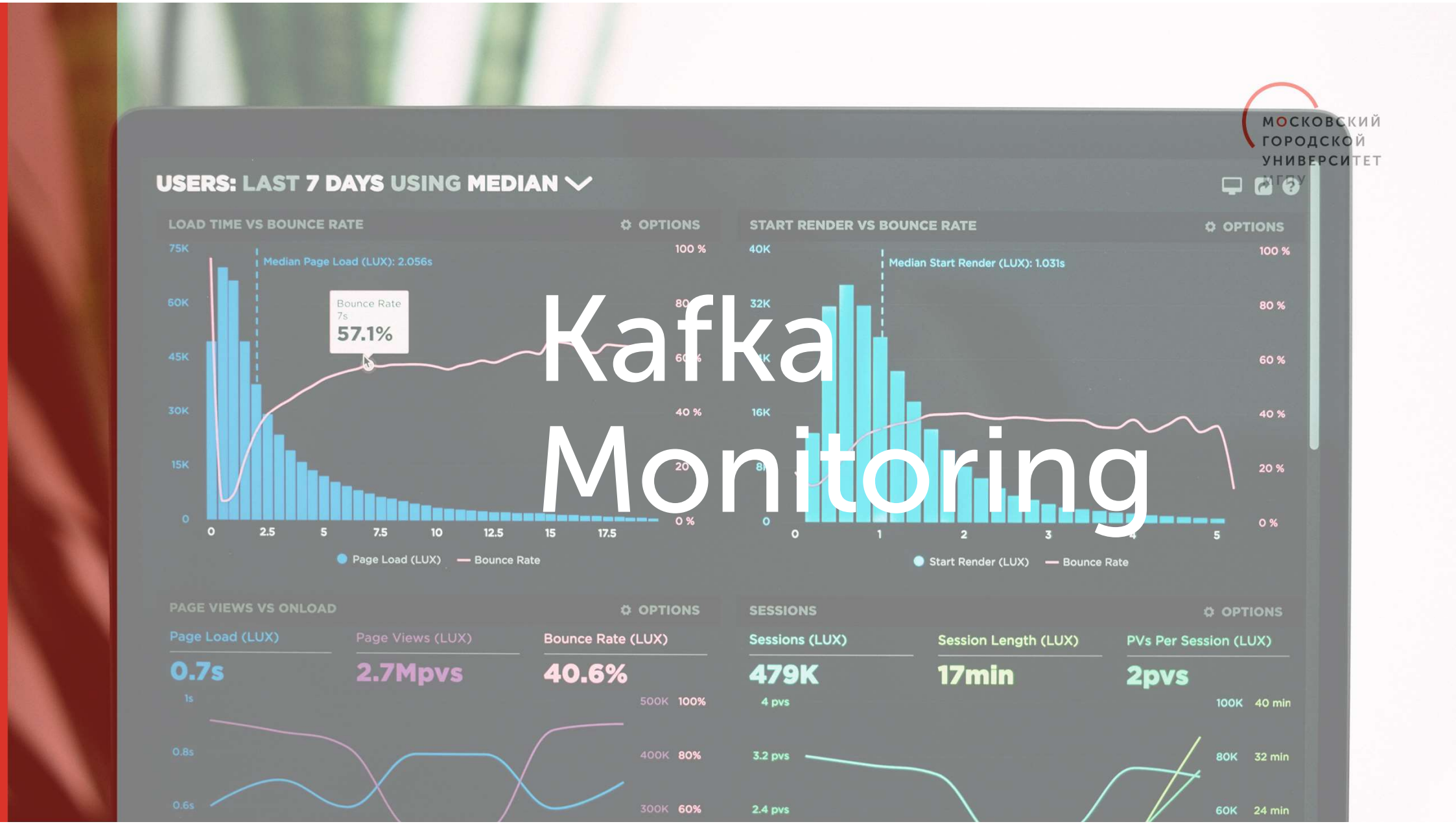
Source: https://www.slideshare.net/Lightbend/running-kafka-on-kubernetes-with-strimzi-for-realtime-streaming-applications

Strimzi – Connecting Clients

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

“Plain”	9092
TLS	9093
Interbroker	9094
Prometheus	9404

Source: https://www.slideshare.net/Lightbend/running-kafka-on-kubernetes-with-strimzi-for-realtime-streaming-applications



Kafka Monitoring

Kafka Monitoring

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Опции:

https://github.com/linkedin/kafka-monitor – Kafka Monitor

- FileBeat + Kibana
- Cruise Control
- Burrow
- https://prometheus.io/ - Prometheus + Grafana
- Внешние инструменты, такие как:
- Datadog
- Splunk

Kafka Monitoring

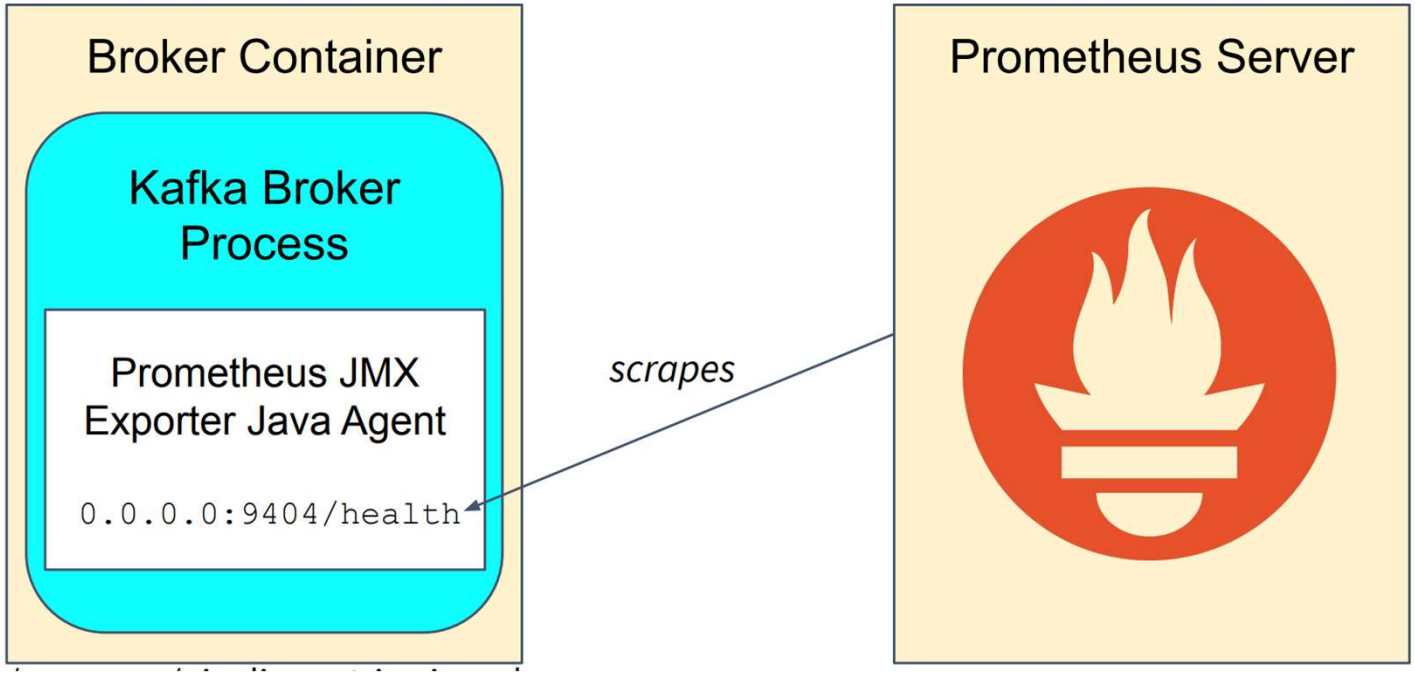
МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Kubernetes + Prometheus + Grafana

Source: https://www.slideshare.net/Lightbend/running-kafka-on-kubernetes-with-strimzi-for-realtime-streaming-applications

Kafka Monitoring

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ



Source: https://www.slideshare.net/Lightbend/running-kafka-on-kubernetes-with-strimzi-for-realtime-streaming-applications

Key Metrics to Monitor

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Древовидные семейства метрик:

- Kafka Broker metrics
- JVM metrics
- Host/server metrics

Key Metrics to Set Alerts

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

kafka.server:type=ReplicaManager,name=UnderReplicatedPartitions

Number of under-replicated partitions (| ISR | < | all replicas |). Alert if value is greater than 0.

kafka.controller:type=KafkaController,name=OfflinePartitionsCount

Number of partitions that don't have an active leader and are hence not writable or readable. Alert if value is greater than 0.

kafka.controller:type=KafkaController,name=ActiveControllerCount

Number of active controllers in the cluster. Alert if value is anything other than 1.

Key Metrics to Monitor

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

kafka.server:type=BrokerTopicMetrics,name=MessagesInPerSec

Aggregate incoming message rate.

kafka.server:type=BrokerTopicMetrics,name=BytesInPerSec

Aggregate incoming byte rate.

kafka.server:type=BrokerTopicMetrics,name=BytesOutPerSec

Aggregate outgoing byte rate.

kafka.network:type=RequestMetrics,name=RequestsPerSec,request={Produce|FetchConsumer|FetchFollower}

Request rate.

Key Metrics to Monitor

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

kafka.log:type=LogFlushStats,name=LogFlushRateAndTimeMs

Log flush rate and time.

kafka.controller:type=ControllerStats,name=LeaderElectionRateAndTimeMs

Leader election rate and latency.

kafka.controller:type=ControllerStats,name=UncleanLeaderElectionsPerSec

Unclean leader election rate.

kafka.server:type=ReplicaManager,name=PartitionCount

Number of partitions on this broker. This should be mostly even across all brokers.

Key Metrics to Monitor

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

kafka.server:type=ReplicaManager,name=LeaderCount

Number of leaders on this broker. This should be mostly even across all brokers. If not, set auto.leader.rebalance.enable to true on all brokers in the cluster.

kafka.server:type=ReplicaManager,name=IsrShrinksPerSec

If a broker goes down, ISR for some of the partitions will shrink. When that broker is up again, ISR will be expanded once the replicas are fully caught up. Other than that, the expected value for both ISR shrink rate and expansion rate is 0.

kafka.server:type=ReplicaManager,name=IsrExpandsPerSec

When a broker is brought up after a failure, it starts catching up by reading from the leader. Once it is caught up, it gets added back to the ISR.

kafka.server:type=ReplicaFetcherManager,name=MaxLag,clientId=Replica

Maximum lag in messages between the follower and leader replicas. This is controlled by the replica.lag.max.messages config.

Key Metrics to Monitor

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

kafka.server:type=FetcherLagMetrics,name=ConsumerLag,clientId=([-.\w]+),topic=([-.\w]+),partition=([0-9]+)

Lag in number of messages per follower replica. This is useful to know if the replica is slow or has stopped replicating from

kafka.network:type=RequestMetrics,name=TotalTimeMs,request={Produce|FetchConsumer|FetchFollower}

Total time in ms to serve the specified request.

kafka.server:type=DelayedOperationPurgatory,delayedOperation=Produce,name=PurgatorySize

Number of requests waiting in the producer purgatory. This should be non-zero acks=-1 is used on the producer.

kafka.server:type=DelayedOperationPurgatory,delayedOperation=Fetch,name=PurgatorySize

Number of requests waiting in the fetch purgatory. This is high if consumers use a large

Kafka Monitoring – best practicies

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Настроить оповещения

Не забывайте следить за Zookeeper

Не забывайте отслеживать метрики Consumer и Producer.

Kafka Monitoring – best practicies

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Настроить оповещения

Не забывайте следить за Zookeeper

Не забывайте отслеживать метрики Consumer и Producer.

Собирайте журналы и сохраняйте их в ElasticSearch или другом агрегаторе журналов.

Все важные метрики и их описание:

https://docs.confluent.io/3.2.2/kafka/monitoring.html#server-metrics

Key Metrics to Monitor - materials

МОСКОВСКИЙ ГОРОДСКОЙ УНИВЕРСИТЕТ МГПУ

Articles:

https://www.datadoghq.com/blog/monitoring-kafka-performance-metrics/

https://docs.confluent.io/current/kafka/monitoring.html

МОСКОВСКИЙ
ГОРОДСКОЙ
УНИВЕРСИТЕТ
МГПУ

СПАСИБО ЗА ВНИМАНИЕ