

Зачем необходим мониторинг Kafka:

- Отслеживание поведения системы в реальном времени;
- Трекинг изменений в поведении с течением времени;
- Заблаговременное планирование ресурсов.

Что необходимо мониторить:

- Собирать метрики Apache Kafka и Apache Zookeeper;
- Собирать метрики JVM;
- Собирать метрики хоста системы (загрузки памяти, ЦПУ сетевого стека, диска).

Мониторинг Kafka - основные тезисы:

- Kafka репортит несколько сотен метрик из коробки;
- По умолчанию, метрики репортятся через JMX;
- Можно подключать кастомные репортеры, изменяя настройку `metric.reporters`.

Варианты архитектуры (несколько способов сбора и отображения метрик из Kafka):

- JmxTrans + Graphite/StatsD/TSDB: <https://github.com/jmxtrans/jmxtrans>
- Prometheus & JMX Exporter: https://github.com/prometheus/jmx_exporter
- Custom Reporter

Статьи о проблемах с EDR:

- <https://medium.com/expedia-group-tech/your-latency-metrics-could-be-misleading-you-how-hdrhistogram-can-help-9d545b598374>

- <https://engineering.salesforce.com/be-careful-with-reservoirs-708884018daf>

Ключевые метрики Kafka (на основе 4 Golden Signals:

https://sre.google/sre-book/monitoring-distributed-systems/#xref_monitoring_golden-signals):

Доступность	Скорость	Загруженность	Трафик
<u>UnderReplicatedPartitions</u> <u>UnderMinIsrPartitionCount</u> <u>Isr[Shrinks Expands]PerSec</u> <u>ActiveControllerCount</u> <u>LeaderElectionRateAndTimeMs</u> ErrorsPerSec SessionState	TotalTimeMs PurgatorySize <u>ZooKeeperRequestLatencyMs</u> GC CollectionCount <u>GC CollectionTime</u>	PartitionCount <u>LeaderCount</u> <u>NetworkProcessorAvgIdlePercent</u> <u>RequestHandlerAvgIdlePercent</u> Free Disk Space DISK IOPS/Latency CPU Usage Network Bytes In/Out	<u>MessagesInPerSec</u> <u>Bytes[In Out]PerSec</u> <u>RequestsPerSec</u>

Ключевые метрики Zookeeper (на основе 4 Golden Signals:

https://sre.google/sre-book/monitoring-distributed-systems/#xref_monitoring_golden-signals):

Доступность	Скорость	Загруженность	Трафик
Leader (sum = 1) PartOfEnsemble (sum = N)	AvgRequestLatency GC CollectionCount <u>GC CollectionTime</u>	NodeCount WatchCount <u>OutstandingRequests</u> Free Disk Space Disk IOPS/Latency	<u>PacketsSent</u> <u>PacketsReceived</u> NumAliveConnections

Мониторинг клиентов и сбор метрик с клиентских библиотек:

- Клиенты Кафки также репортят кучу метрик из коробки;
- Желательно оборачивать “ванильных клиентов” в собственную библиотеку;
- Для мониторинга лага коньюмеров лучше использовать <https://github.com/linkedin/Burrow>

SLI:

- SLI (Service level indicator) - ключевая метрика работоспособности сервиса;
- Доступность записи = $(\text{Total Produce Requests} - \text{Produce Request Errors}) / \text{Total Produce Requests} * 100\%$

SLO:

- SLO (Service Level Objective) - целевое значение SLI;
- Повышает стабильность конечного продукта и помогает в приоритизации нашей работы;
- Выступает в следующих гарантиях, что доступность записи $\geq 99.9\%$ в течение последних 14 дней.

Как считать SLI/SLO:

- Использование метрик внешнего монитора

Xinfra Monitor: <https://github.com/linkedin/kafka-monitor>