

Apache Kafka 모니터링을 위한 Metrics 이해

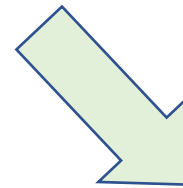
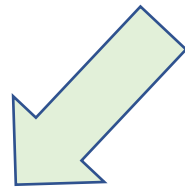
박상원

Kafka Monitoring?

현재 상황을 파악하고

이상 상태를 감지하여

대응하는 활동

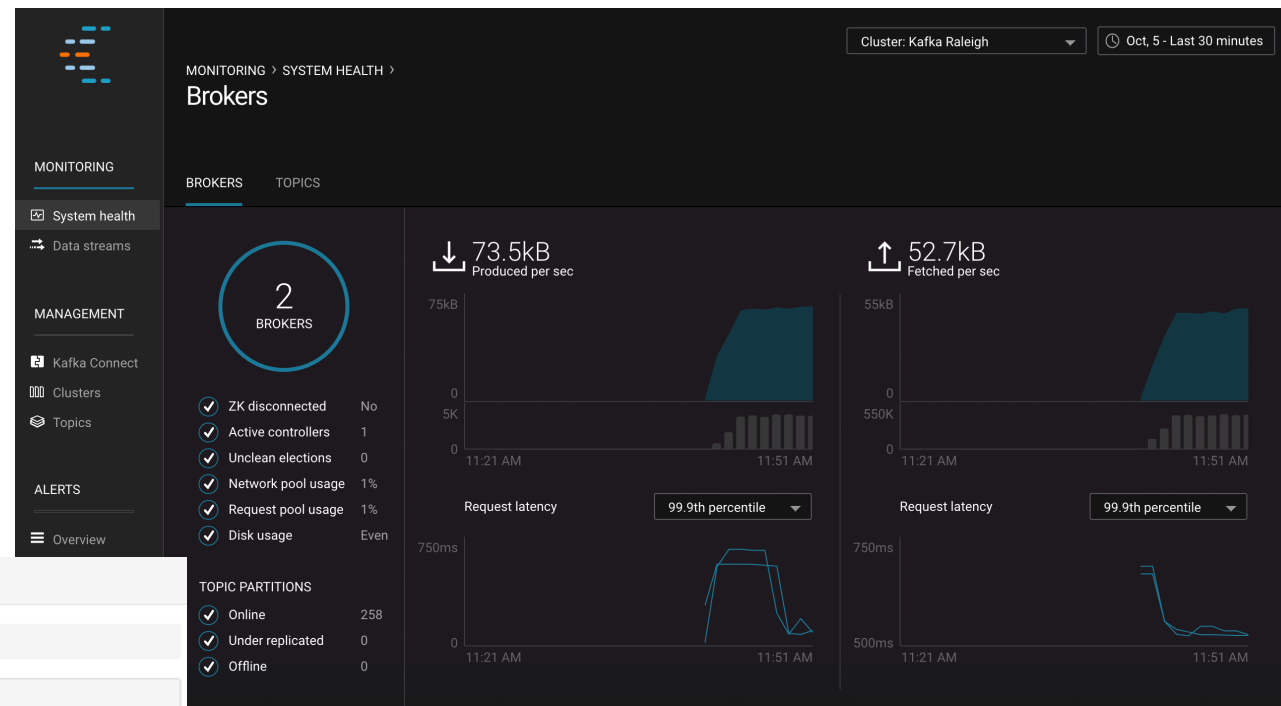


장애 방지

성능 개선

어떻게

현재 상황을 파악할 수 있을까?



Brokers Topic Preferred Replica Election Reassign Partitions Consumers

Topic Summary

Replication	1
Number of Partitions	1
Sum of partition offsets	0
Total number of Brokers	1
Number of Brokers for Topic	1
Preferred Replicas %	100
Brokers Skewed %	0
Brokers Spread %	100
Under-replicated %	0

Operations

Delete Topic Reassign Partitions Generate Partition Assignments

Add Partitions Update Config Manual Partition Assignments

Partitions by Broker

Broker	# of Partitions	Partitions	Skewed?
0	1	(0)	false

Consumers consuming from this topic

Please enable consumer polling [here](#).

Metrics

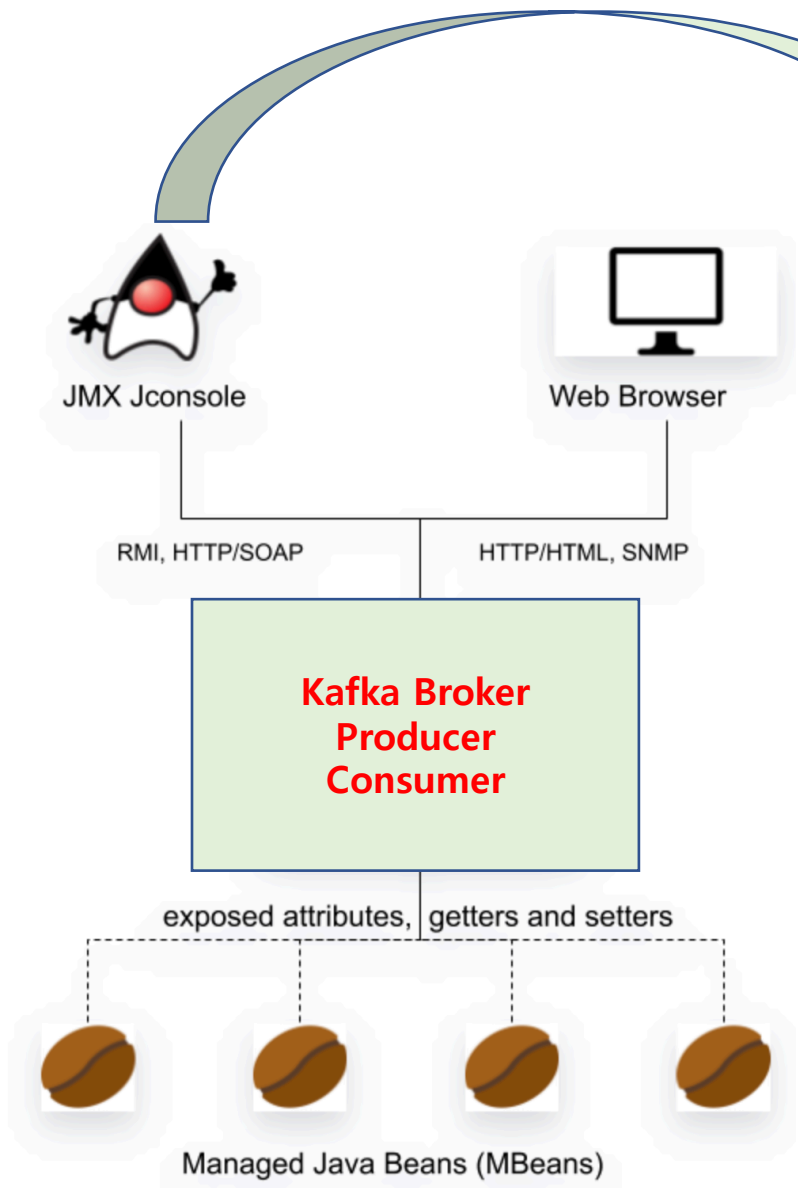
Please enable JMX polling [here](#).

ManageEngine Applications Manager

내 업무에 최적화된 모니터링을 하려면?

다른 Metric과 연계한 새로운 지표 생성

측정된 지표에 따른 Alarm 등의 업무 자동화



MBeans

- ▶ com.sun.management
- ▶ java.lang
- ▶ java.nio
- ▶ java.util.logging
- ▶ kafka
- ▶ kafka.controller
- ▶ kafka.coordinator.group
- ▶ kafka.coordinator.transaction
- ▶ kafka.log
- ▶ kafka.network
- ▼ kafka.server
 - ▼ BrokerTopicMetrics
 - 🔍 BytesInPerSec
 - 🔍 BytesOutPerSec
 - 🔍 BytesRejectedPerSec
 - 🔍 FailedFetchRequestsPerSec
 - 🔍 FailedProduceRequestsPerSec
 - 🔍 FetchMessageConversionsPerSe
 - 🔍 **MessagesInPerSec**
 - 🔍 ProduceMessageConversionsPe
 - 🔍 ReplicationBytesInPerSec
 - 🔍 ReplicationBytesOutPerSec
 - 🔍 TotalFetchRequestsPerSec
 - 🔍 TotalProduceRequestsPerSec
 - ▶ DelayedOperationPurgatory
 - 🔍 Fetch
 - ▶ FetchSessionCache
 - ▶ KafkaRequestHandlerPool
 - ▶ KafkaServer

Attributes | **Operations** | **Notifications** | **Metadata**

Attribute values

Name	Value
Count	0
EventType	messages
FifteenMinuteRate	0.0
FiveMinuteRate	0.0
MeanRate	0.0
OneMinuteRate	0.0
RateUnit	SECONDS

[Refresh](#)

수집할 JMX Metrics을 선택

- ▶ kafka
- ▶ kafka.cluster
- ▶ kafka.controller
- ▶ kafka.coordinator.group
- ▶ kafka.coordinator.transaction
- ▶ kafka.log
- ▶ kafka.network
- ▼ kafka.server
 - ▼ BrokerTopicMetrics
 - ▶ BytesInPerSec
 - ▶ BytesOutPerSec
 - ▶ BytesRejectedPerSec
 - ▶ FailedFetchRequestsPerSec
 - ▶ FailedProduceRequestsPerSec
 - ▶ FetchMessageConversionsPerSec
 - ▶ MessagesInPerSec
 - ▶ ProduceMessageConversionsPerSec
 - ▶ ReplicationBytesInPerSec
 - ▶ ReplicationBytesOutPerSec
 - ▶ TotalFetchRequestsPerSec
 - ▶ TotalProduceRequestsPerSec
 - ▶ DelayedFetchMetrics
 - ▶ DelayedOperationPurgatory
 - ▶ Fetch
 - ▶ FetchSessionCache
 - ▶ KafkaRequestHandlerPool
 - ▶ KafkaServer
 - ▶ Produce

```
{  
  "object_name" : "kafka.server:type=BrokerTopicMetrics,name=MessagesInPerSec",  
  "attributes" : [ "OneMinuteRate" ],  
  "object_alias" : "${type}.${name}"  
},  
  
{  
  "object_name" : "kafka.producer:type=producer-metrics,client-id=*",  
  "attributes" : ["outgoing-byte-rate"],  
  "object_alias" : "Producer.BytesRate"  
},
```


어떤 정보를 모니터링 해야 할까?

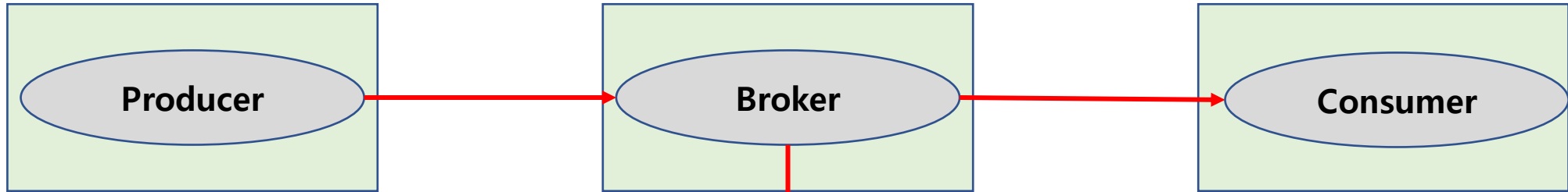
(클러스터 안정성, 메세지 적시성, 처리성능)

적시성

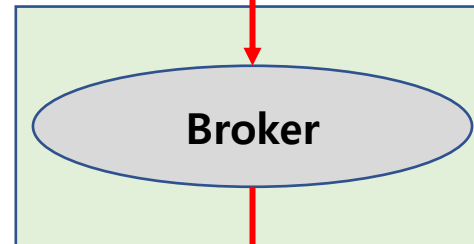
Producer에서 Consumer로 지연 없이 전달되는가?

적시성을 위해 확인해야 할 Metrics 유형은?

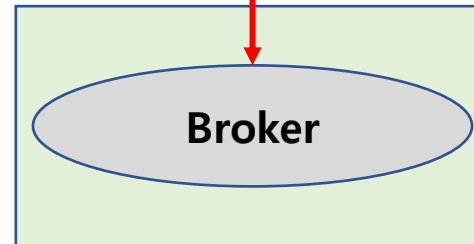
- ✓ Producer에서 Consumer까지 얼마나 빨리/많이 전달되는지 확인 필요



- ✓ 얼마나 빨리/많이 보내고 있는가?



- ✓ 얼마나 빨리/많이 수신하고 있는가?



- ✓ Producer와 Consumer 양쪽의 요청을 얼마나 빨리/많이 처리하고 있는가?

Producer Metrics (Kafka 0.8.2 이후)

Metric	Comments	Alert
request-rate	<ul style="list-style-type: none">초당 요청(to broker) 건수	
response-rate	<ul style="list-style-type: none">초당 응답(from broker) 건수	요청과 응답 비율이 유사해야 함.
outgoing-byte-rate	<ul style="list-style-type: none">초당 Broker로 전송한 bytes	처리량 개선 확인
io-ratio	<ul style="list-style-type: none">I/O 작업을 위해 I/O thread가 사용한 시간 비율	
io-wait-ratio	<ul style="list-style-type: none">I/O thread가 waiting에 소요한 시간 비율	

Consumer Metrics (Kafka 0.9.0.0 이후)

전체 Consumer, Consumer Group, Topic 별 구분하여 모니터링 가능

Metric	Comments	Alert
records-lag-max	<ul style="list-style-type: none">최신의 메세지 offset값과, consumer가 읽어간 offset값의 최대 차이값이 증가한다면, consumer group이 데이터를 빠르게 가져가지 못함	MaxLag > (자체 기준)
fetch-rate	<ul style="list-style-type: none">Broker에 초당 요청하는 회수만약 consumer가 중지되었다면, 0으로 낮아질 것이다.	fetch-rate < 0.5.
records-per-request-avg	<ul style="list-style-type: none">초당 읽은 메세지(record) 개수	
bytes-consumed-rate	<ul style="list-style-type: none">초당 읽은 byte size	
commit-rate	<ul style="list-style-type: none">Consumer가 kafka에 offset을 commit한 비율 (초당 commit수)	
assigned-partitions	<ul style="list-style-type: none">현재 Consumer가 가지고 있는 partition의 갯수	

Broker Metrics

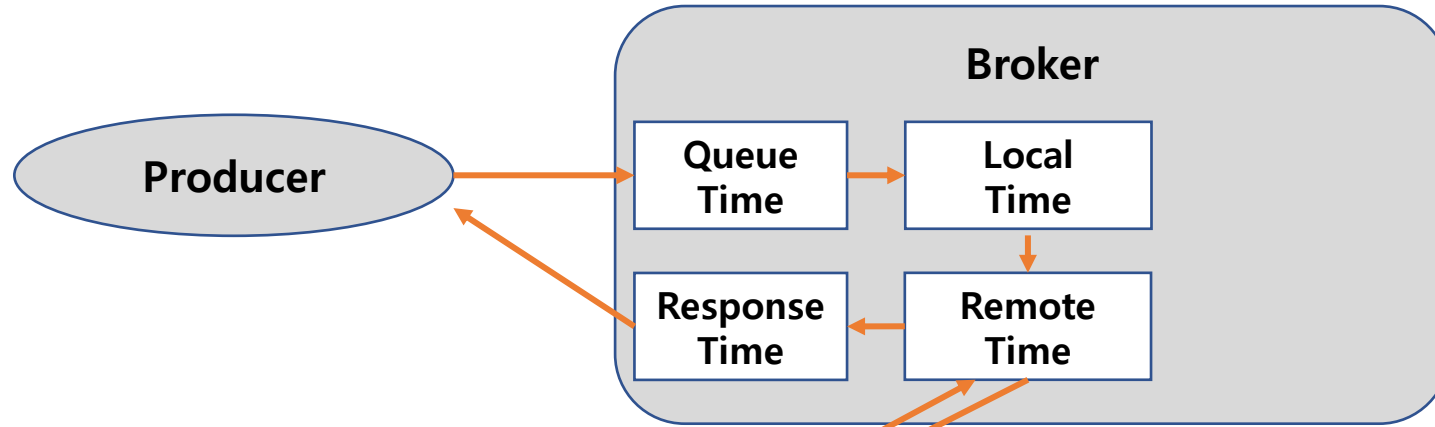
Cluster, Broker, Topic 단위로 구분하여 모니터링 가능

Metric	Comments	Alert
MessagesInPerSec	<ul style="list-style-type: none">• 초당 유입되는 메세지 count (많을 수록 처리성능이 높음)• (kafka.server: type=BrokerTopicMetrics)	
BytesInPerSec / BytesOutPerSec	<ul style="list-style-type: none">• 초당 유입 & 유출 되는 byte (많을 수록 처리성능이 높음)• (kafka.server: type=BrokerTopicMetrics)	
RequestsPerSec	<ul style="list-style-type: none">• 초당 요청 건수 {Produce FetchConsumer FetchFollower}• (kafka.network: type=RequestMetrics)	
TotalTimeMs	<ul style="list-style-type: none">• 하나의 요청을 처리하는데 소요된 전체 시간 {Produce FetchConsumer FetchFollower}• 구간별로 분할하여 시간 측정 가능• (kafka.network: type=RequestMetrics)	

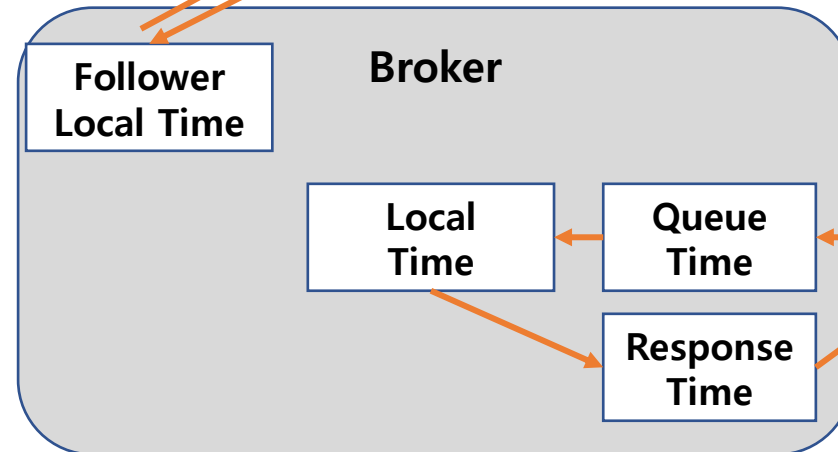
TotalTimeMs 란?

요청을 처리에 소요된 전체 시간으로, 3가지 요청으로 구분

Producer 관점



Follower 관점



Consumer 관점



TotalTimeMs 란? (세부 Metrics 확인)

TotalTimeMs이 너무 오래 걸리면, Bottleneck이 어디서 발생하는지 상세 metrics 확인

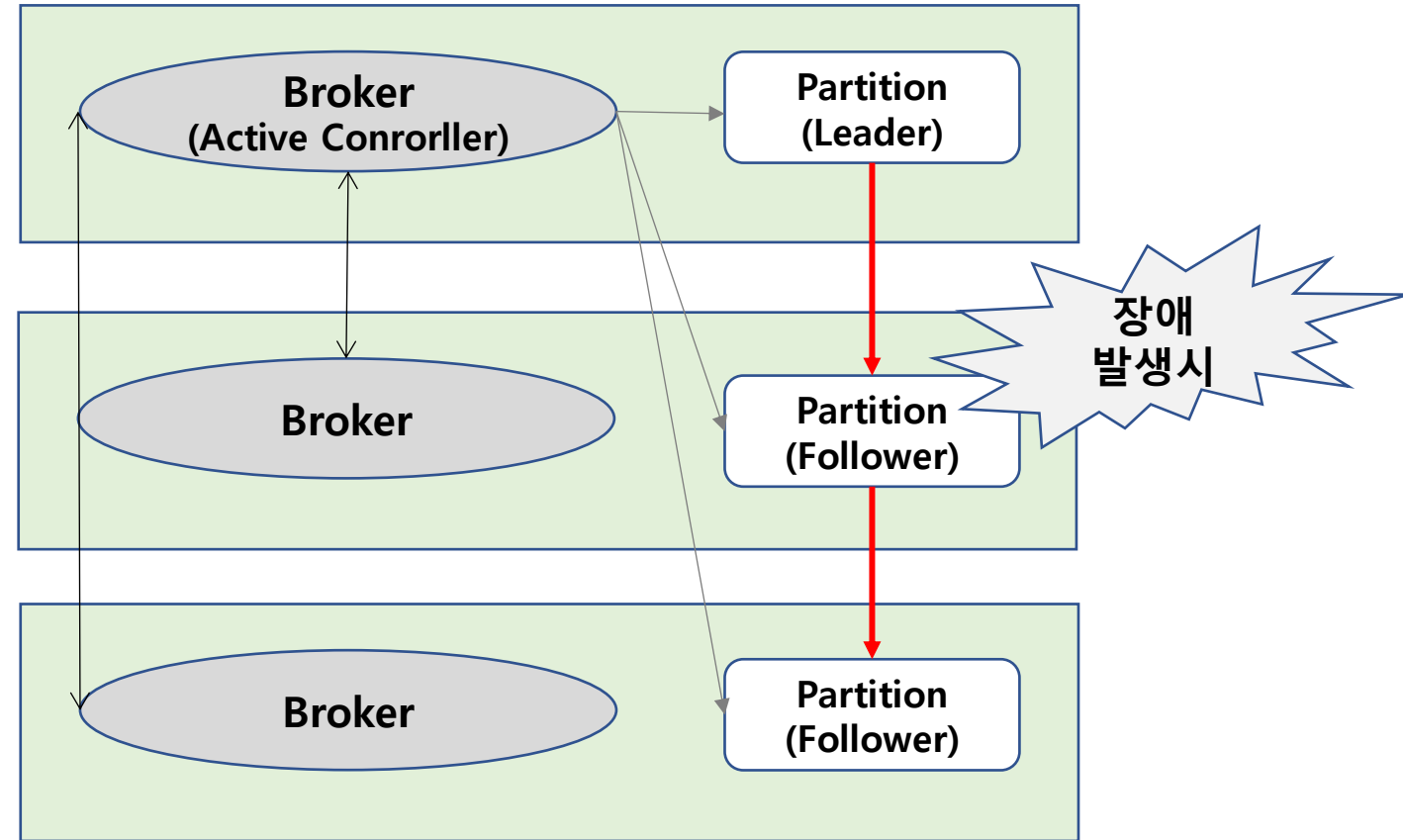
Metric	Comments
RequestQueueTimeMs	<ul style="list-style-type: none">요청 큐에서 기다리는 시간을 producer, consumer fetch, follower fetch별로 측정높은 값(대기 시간이 길어지는 현상)은 I/O thread가 부족하거나, CPU 부하 예상(kafka.network:type=RequestMetrics)
LocalTimeMs	<ul style="list-style-type: none">전달된 요청을 leader에서 처리하는 시간. (leader가 local data 처리)높은 값은 disk i/o가 낮음을 의미(kafka.network:type=RequestMetrics)
RemoteTimeMs	<ul style="list-style-type: none">요청이 Follower를 기다리는 시간높은 값은 NW 연결이 늦어짐을 의미Fetch 요청에서 이 값이 높은 것은, 가져올 데이터가 많지 않음을 의미할 수 있음(kafka.network:type=RequestMetrics)
ResponseQueueTimeMs	<ul style="list-style-type: none">요청이 응답 큐에서 대기하는 시간.높은 값은 NW thread가 부족함을 의미.(kafka.network:type=RequestMetrics)
ResponseSendTimeMs	<ul style="list-style-type: none">Client의 요청에 응답한 시간.높은 값은 NW thread 또는 CPU가 부족하거나, NW부하가 높음을 의미(kafka.network:type=RequestMetrics)

안정성

데이터 유실없이 중단 없는 서비스를 할 수 있는가?

안정성을 위해 확인해야 할 Metrics 유형은?

- ✓ 데이터 처리의 핵심인 Partition이 장애없이 정상적으로 운영되는가?



- ✓ Server의 불필요한 Disk I/O 가 발행하지 않는가?

System Metrics (Swap usage)

Kafka Server에서 Swap이 발생하면, Disk I/O가 발생하여 성능에 영향

Swap이 발생하는 원인

- Kafka 구동시 설정한 Heap Memory를 초과하는 경우
- 데이터를 swap(Disk) 공간으로 복사하게 됨. (프로그램이 중지되지 않도록 하는 역할)
- 한번 swap공간으로 이동하면, 다시 메모리로 돌아오게 할 수 없다. → 성능 저하 유발

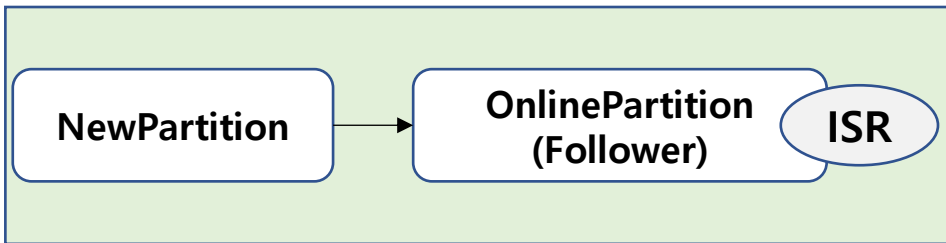
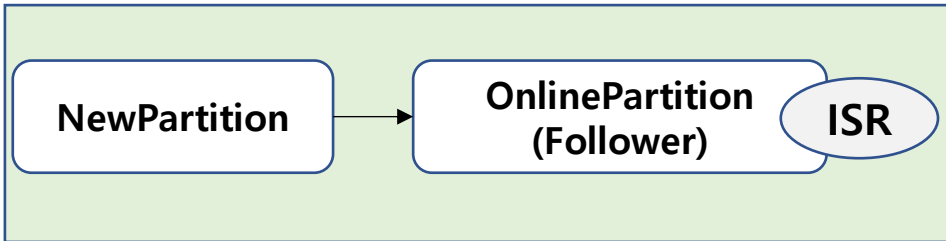
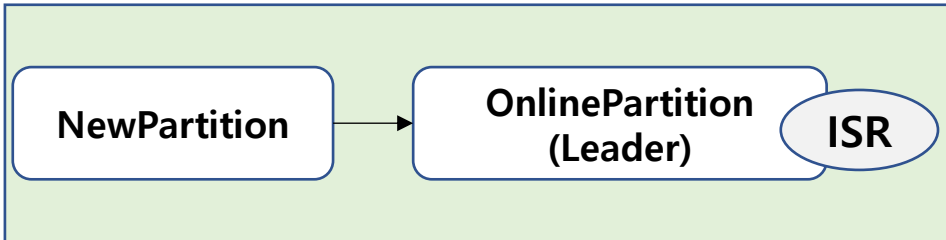
Swap 발생 조건 변경

- `vm.swappiness`
- 메모리에서 swap으로 이동을 언제 할지 결정
- `vm.swappiness = 10` → 메모리 사용률 90% 이상일 때
- Kafka 성능을 극대화 하려면,
- **`vm.swappiness = 0`** → 메모리에서만 처리하도록 설정

Partition의 생성 및 장애 시 상태 변화

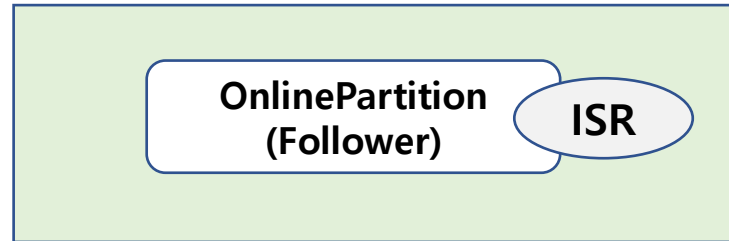
✓ 데이터 처리의 핵심인 Partition이 장애없이 정상적으로 운영되는가?

Topic 생성시 상태 변화



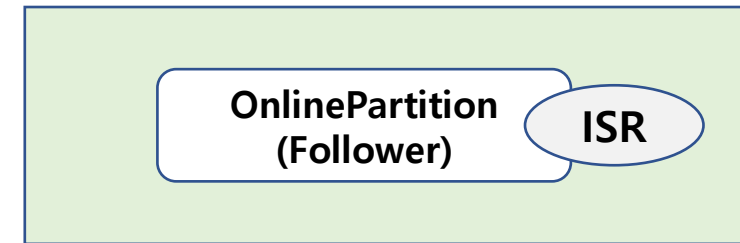
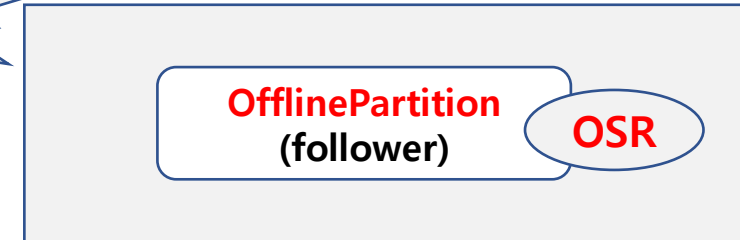
Topic 생성, leader 선출 및 복제가 완료된 상태

Broker 장애 시 상태 변화



장애 Broker의 partition은 OfflinePartition으로 표시

신규 Leader 선출



신규 leader가 선출되고 정상 서비스 가능

Broker Metrics

Cluster와 Partition이 정상적으로 동작하고 있는지 확인하는 Metrics

Metric	Comments	Alert
ActiveControllerCount	<ul style="list-style-type: none">클러스터의 Active Controller 개수 (클러스터 당 1개만 존재)(kafka.controller: type=KafkaController)	1개가 아닌 경우
IsrShrinksPerSec IsrExpandsPerSec	<ul style="list-style-type: none">Broker가 다운되었을때, 일부 partition의 ISR이 줄어든다.Broker가 정상으로 회복하면, OSR상태에서 ISR로 복귀되는 비율(kafka.server: type=ReplicaManager)	!= 0
OfflinePartitionsCount	<ul style="list-style-type: none">쓰기가 불가능한 leader partition의 개수 (partiton의 장애)(kafka.controller: type=KafkaController)	> 0
UnderReplicatedPartitions	<ul style="list-style-type: none">데이터 복제가 완료되지 못한 partition의 개수즉, 전체 partition수에서 ISR을 제외한 수(kafka.server: type=ReplicaManager)	> 0
UncleanLeaderElectionsPerSec	<ul style="list-style-type: none">Leader election이 진행중인 비율진행중이라면 그 동안 leader가 정상동작을 못하므로, 0이 되어야 함.(kafka.controller: type=ControllerStats)	!= 0

Broker Metrics

처리 성능에 영향을 주는 Metrics

Metric	Comments	Alert
PurgatorySize	<ul style="list-style-type: none">• Broker에서 요청을 처리하지 않고, 격리시킨 요청(request)의 건수• 어떤 경우에 이렇게 요청을 격리하나?<ul style="list-style-type: none">• Produce (Producer 관점)<ul style="list-style-type: none">• Request.required.acks = -1(all) 일 때,• 모든 복제가 완료되기 전까지, producer의 request는 대기(격리)• Fetch (consumer 관점)<ul style="list-style-type: none">• Fetch.wait.max.ms 시간 또는• fetch.min.bytes만큼의 데이터가 없는 경우• (kafka.server:type=DelayedOperationPurgatory)	운영자의 설정에 따라 판단
NetworkProcessorAvgIdlePercent	<ul style="list-style-type: none">• 네트워크 프로세서가 유휴상태인 비율• 낮을 수록 thread가 많은 작업을 하고 있음을 의미• (kafka.network: type=SocketServer)	< 0.3.
RequestHandlerAvgIdlePercent	<ul style="list-style-type: none">• Request handler thread가 유휴상태인 시간의 평균• 이 수치가 낮으면 일을 안한다는 의미.• (kafka.server: type=KafkaRequestHandlerPool)	< 0.3.

Consumer Metrics (Kafka 0.9.0.0 이후)

Consumer 관점에서 Broker의 부하를 확인하는 Metrics

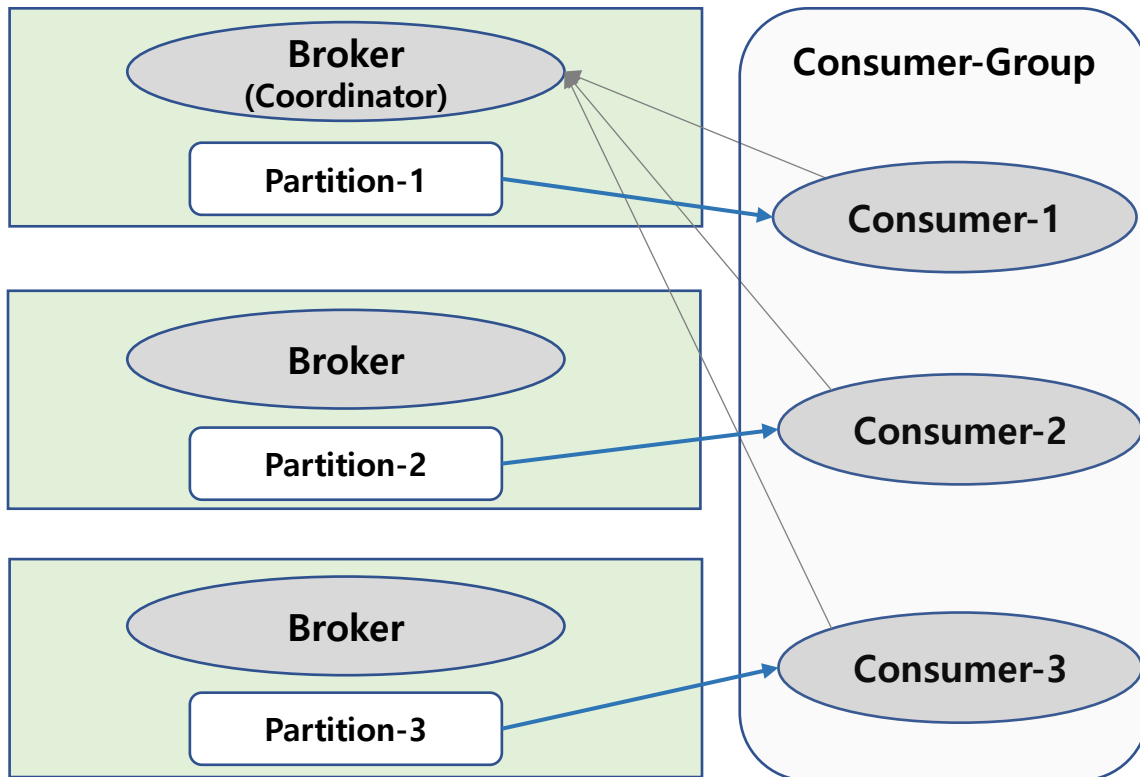
Metric	Comments	Alert
fetch-throttle-time-avg	<ul style="list-style-type: none">Consumer의 요청으로 Broker의 과도한 자원(cpu,network 등)을 사용하는 경우, Broker가 임의로 Consumer 요청을 제한한 시간	지속 증가 시 Broker 추가 고려
join-rate	<ul style="list-style-type: none">Consumer 장애로 인하여, partition이 다른 consumer로 할당되는 비율	증가 시 Consumer 안정성 개선 필요

Consumer 장애를 판단하는 기준은?

Consumer와 Broker(Coordinator)간의 주기적 신호를 통해 장애 판단

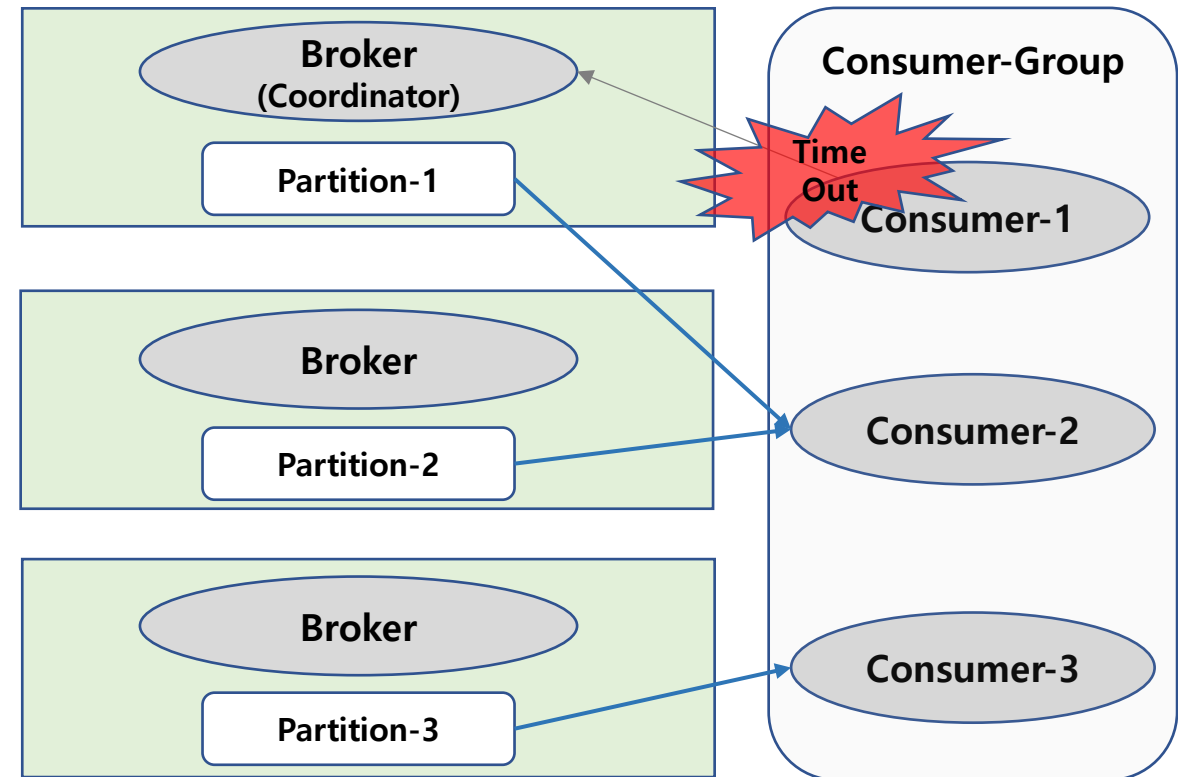
Consumer의 신호 확인 (`session.timeout.ms`, `max.poll.interval.ms`)

- Consumer가 주기적 신호를 Coordinator로 전송하여 장애 여부 판단
- Session Time, Poll(데이터 요청) Time 시간을 확인



Consumer 장애 시 Partition 재할당

- Consumer의 신호가 없으면 장애로 판단하고, partition 재 할당
- JMX Metric의 Join 발생

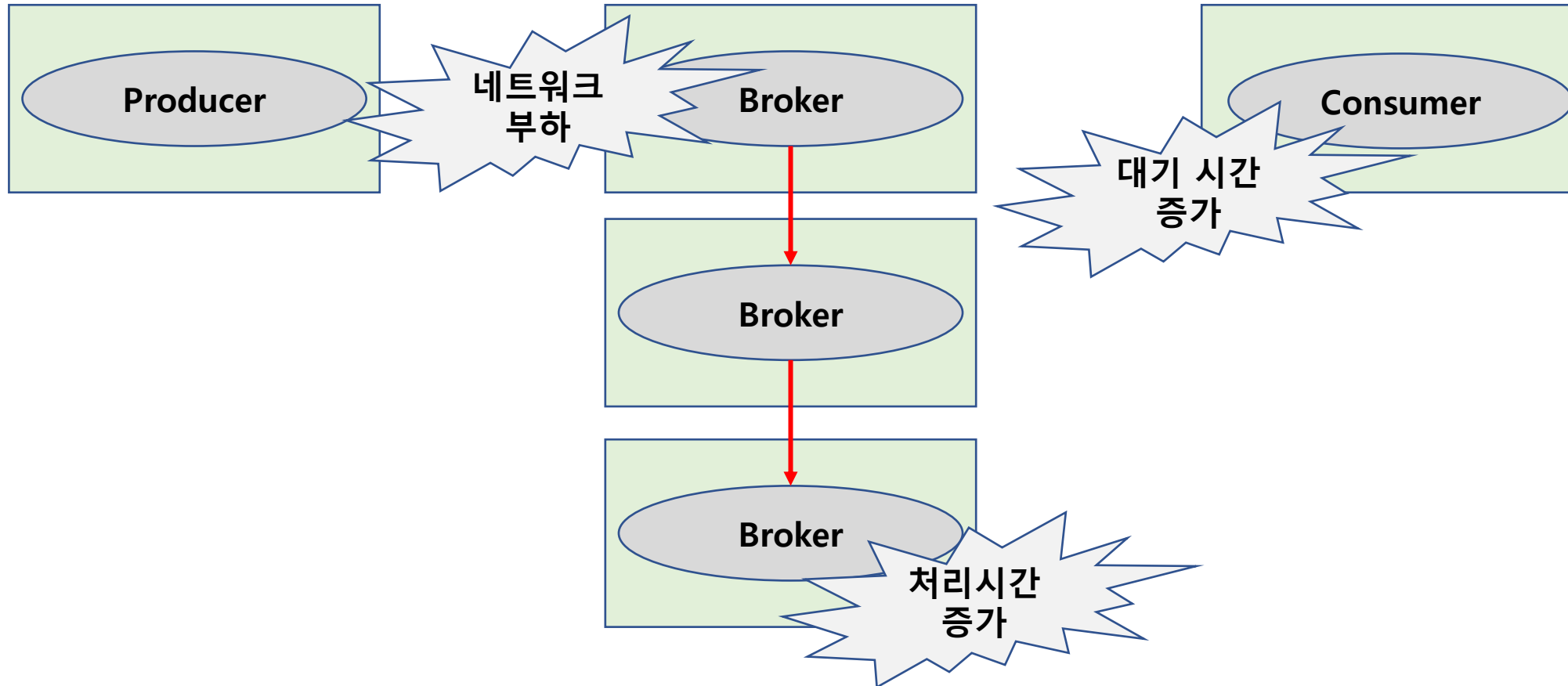


확장 시점

언제 클러스터를 확장해야 할까?

확장성을 위해 확인해야 할 Metrics 유형은?

- ✓ 과도한 부하가 발생하는 구간이 어디인가? (NW, CPU, Memory)



- ✓ **Broker 처리 성능이 저하되고 있는가?**

Broker Metrics

Broker 서버 자원의 상태를 예측하는 Metrics

Metric	Comments	대응
RequestsPerSec	<ul style="list-style-type: none">• 너무 많은 요청으로 Broker의 네트워크 대역폭을 지속 초과하면,• Broker를 확장하여 분산 처리 필요• (kafka.network: type=RequestMetrics)	
LogFlushRateAndTimeMs	<ul style="list-style-type: none">• Cache에 저장된 데이터를 Log(Disk)로 저장하는 비율• 비율이 점차 늦어진다면,• Broker를 추가하여 partition에 쓰는 부하를 분산 필요• (kafka.log:type=LogFlushStats)	지연 발생시, 데이터 유실 가능성 높음
ErrorsPerSec	<ul style="list-style-type: none">• 즉, Network Error가 없는 상태에서 Broker 처리량이 줄어든다면,• Broker 내부 자원 부족 가능성 높음• (kafka.network: type=RequestMetrics)	

**자신의 운영 시스템에 최적화된
모니터링 지표의 조합이 가장 중요**

END