Kafka Day 8 – Revision Notes (Monitoring, Scaling, Throughput Tuning)

---

1. Kafka Monitoring Essentials

- Track key metrics:
    - **Broker**: CPU, disk I/O, GC pause time
    - **Topics**: under-replicated partitions, partition skew
    - **Consumers**: lag, rebalance frequency
- Tools: Prometheus, Grafana, JMX Exporter, Burrow, Confluent Control Center

---

2. Consumer Lag Root Causes

- Causes: slow processing, network issues, GC pauses, unbalanced partitions
- Fixes: increase parallelism, improve consumer logic, use async processing, optimize batch size

---

3. Diagnosing Time-Based Lag Spikes

- Analyze daily batch jobs or downstream bottlenecks
- Check JVM GC logs and processing capacity during peak times
- Use alerts to catch patterns early and scale consumers proactively

---

4. Scaling Kafka Producers and Consumers

- **Producers**: tune `batch.size`, `linger.ms`, `compression.type`
- **Consumers**: increase partitions, run multiple consumers in same group
- Ensure rebalance stability and monitor throughput

---

5. Compression Impact on Throughput

- Compression reduces network/disk I/O but increases CPU
- **Snappy** = faster, lower compression; **Zstd** = slower, better compression ratio
- Large batch + Zstd may hurt latency

---

6. Broker-Side Tuning for High Throughput

- Tune configs:
  - `num.network.threads`
  - `num.io.threads`
  - `socket.request.max.bytes`
- Optimize replication settings and thread pool sizing

---

7. Partition-Level Lag Diagnosis

- Causes: skewed key distribution, slow processing in specific partitions
- Solution: rebalance keys, distribute load evenly, analyze consumer assignment

---

8. Handling Throughput Drop After Compression

- Monitor CPU load, GC pauses, thread pools
- Try Snappy if Zstd is causing latency
- Adjust `linger.ms` and `batch.size` for better batching

---

9. Reducing Lag During High Throughput

- Increase consumer instances
- Reduce `max.poll.records`
- Use async writing to sink (e.g., DB)
- Monitor downstream systems to ensure they're not choking

---

10. Key Configs for Rebalance Stability

- `session.timeout.ms`, `heartbeat.interval.ms`, `max.poll.interval.ms`
- Use **CooperativeStickyAssignor** to reduce partition movement and rebalance load gracefully