



Monitoring Apache Kafka

Gwen Shapira, Principal Data Architect
@gwenshap

#1 Kafka Monitoring Tip:

Please Monitor Apache Kafka.

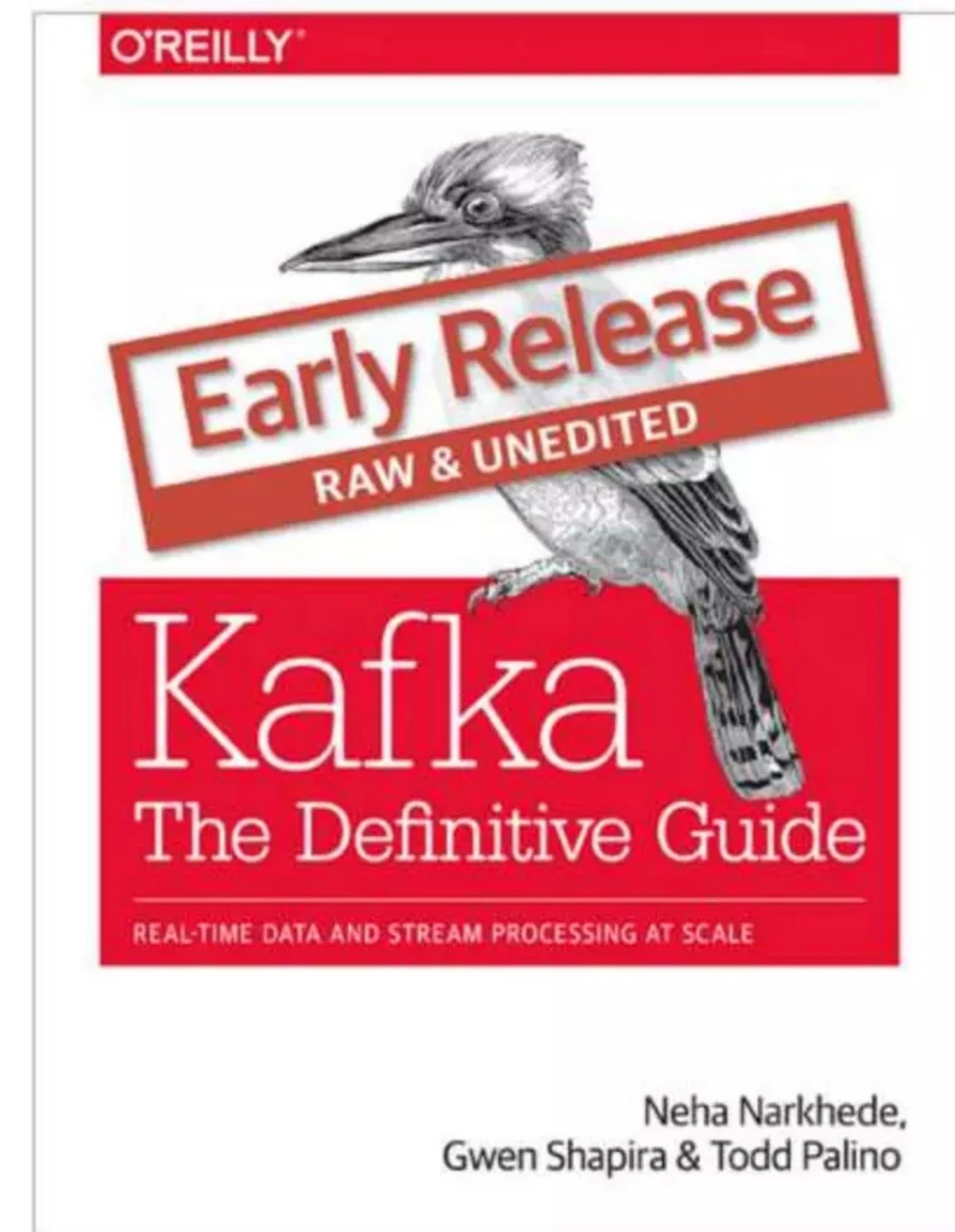


Breaking tuning
production databases
since 1997

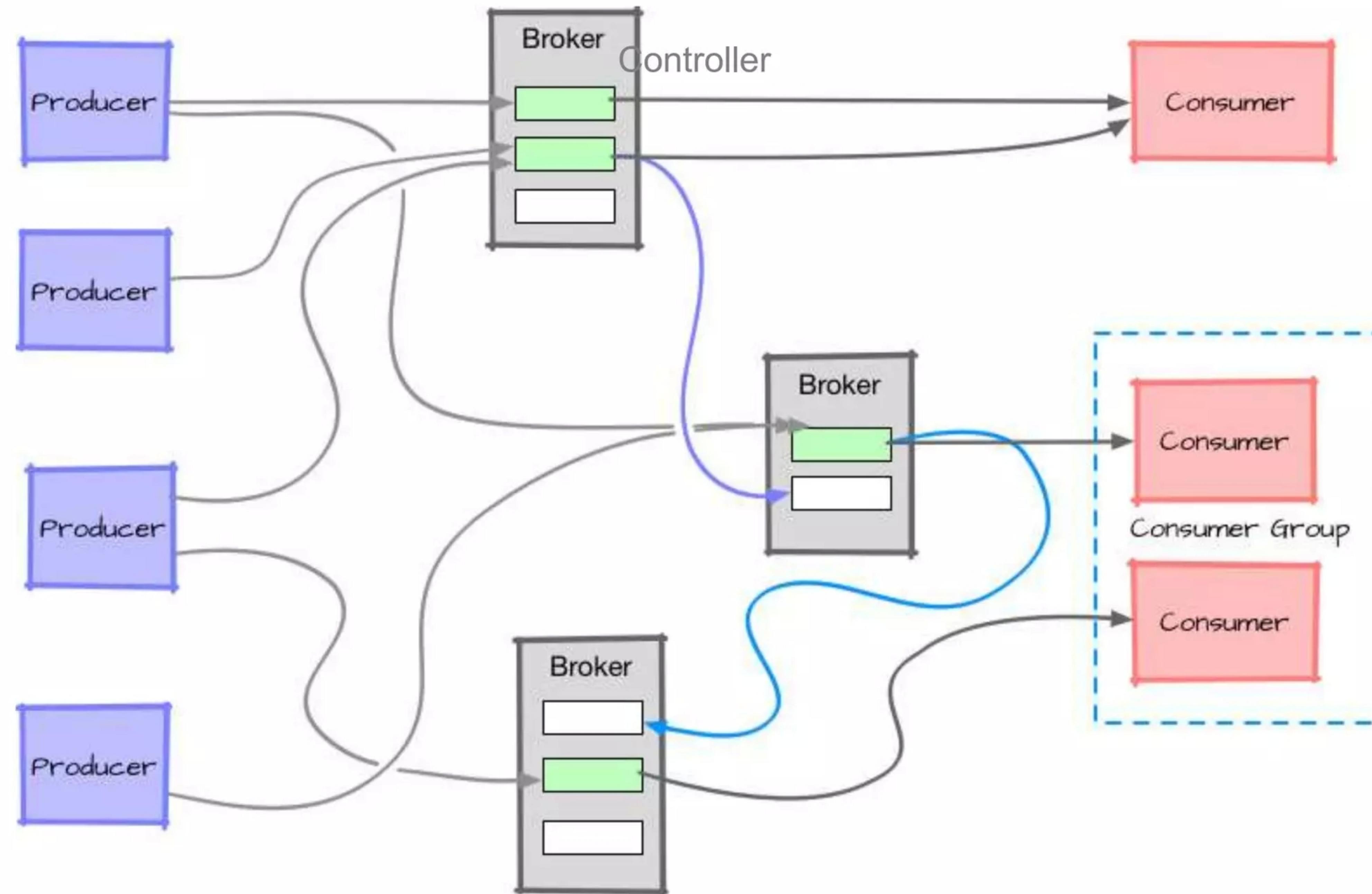
Apache Kafka PMC

Principal Data Architect

Tweeting a lot
@gwenshap



Apache Kafka is a distributed system and has many components



Things to keep an eye on

- Broker health
- Message delivery
- Performance
- Capacity



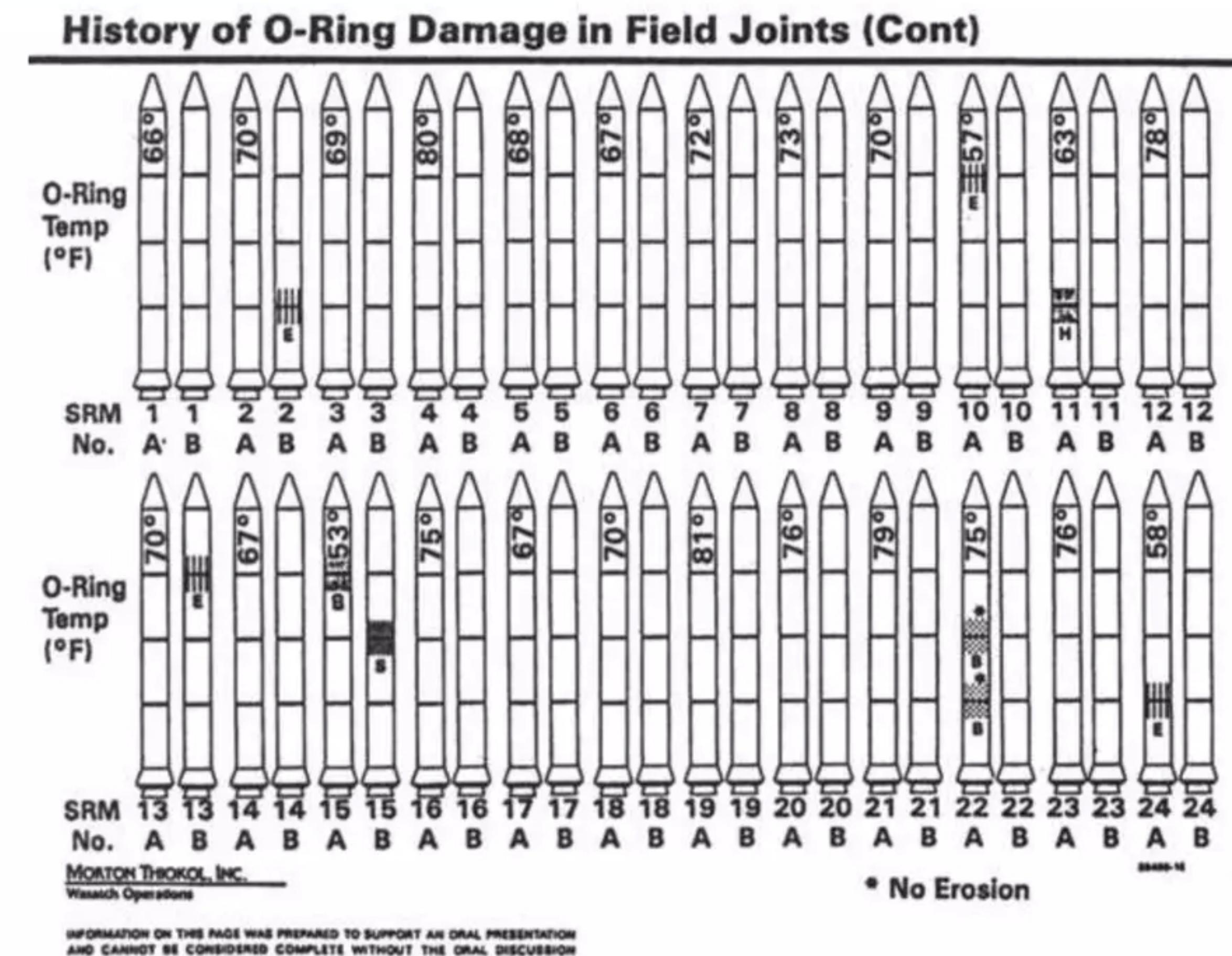
Broker Health Monitoring

Kafka & Metrics Reporters

- Pluggable interface
- JMX reporter built in
- TONS of metrics

TONS of Metrics

- Broker throughput
- Topic throughput
- Disk utilization
- Unclean leader elections
- Network pool usage
- Request pool usage
- Request latencies – 30 request types, 5 phases each
- Topic partition status counts: online, under replicated, offline
- Log flush rates
- ZK disconnects
- Garbage collection pauses
- Message delivery
- Consumer groups reading from topics
- ...



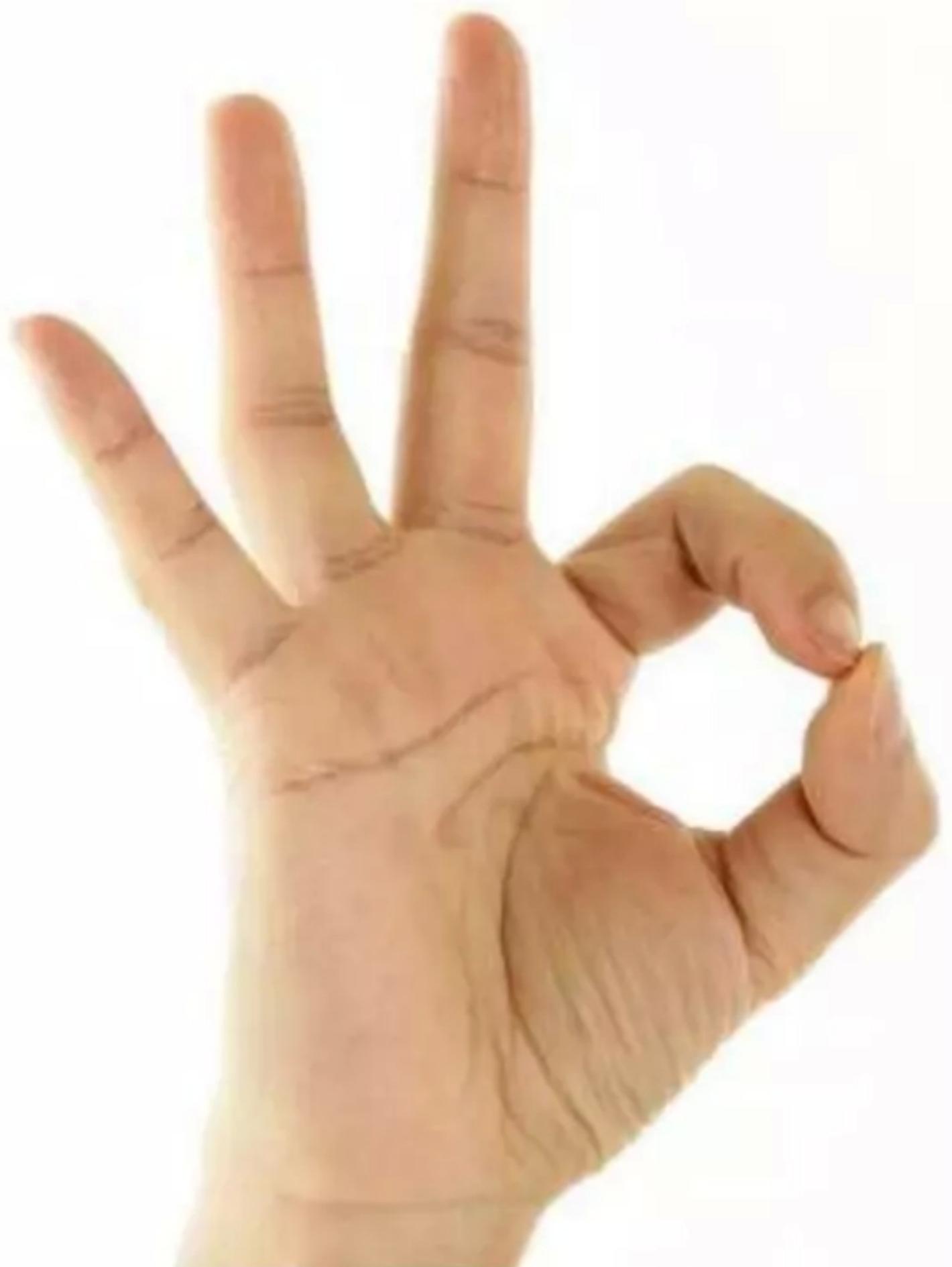
#2 Kafka Monitoring Tip:

Have a dashboard

that lets you know

“is everything ok?”

in one look



Is the broker process up?

- ps -ef | grep kafka
- Do we receive metrics?

Under-replicated partitions

- If you can monitor just one thing...
- Is it a specific broker?
- Cluster wide:
 - Out of resources
 - Imbalance
- Broker:
 - Hardware
 - Noisy neighbor
 - Configuration
 - Garbage collection

TOPIC PARTITIONS		
 Online		242
 Under replicated		208
 Offline		33

Drill Down into Broker and Topic: Do we see a problem right here?

Broker ID	Throughput		Latency (produce)				Latency (fetch)	
	Bytes in/sec	Bytes out/sec	99.9th %ile	99th %ile	95th %ile	Median	99.9th %ile	99th %ile
1 ***	19.8kB	10.6kB	531ms	134ms	50ms	11ms	538ms	509ms
2 ***	20.4kB	9.93kB	456ms	116ms	47ms	10ms	528ms	512ms
3 ***	20.5kB	9.05kB	590ms	195ms	55ms	12ms	614ms	510ms

Topic Name	Throughput		Replication		
	Bytes in/sec	Bytes out/sec	Total	In sync	
connect-config	...	0	0	1	1
connect-offsets	...	1.00B	1.00B	25	25
connect-status	...	0	0	5	5
wikipedia.failed	...			6	6
wikipedia.parsed	...	2.76kB	2.76kB	6	6

Check partition placement - is the issue specific to one broker?



Partitions		Data flow Consumer groups	Replica placement Broker list	1	2
id	Replicas				
0		2	2,1		
1		2	1,2		

#3 Kafka Monitoring Tip:

Monitor
Under-replicated Partitions



Canary

- Produce an event
- Try to consume the event 1s later
- Did you get it?



Other important health metrics

- Active Controller
- ZK Disconnects
- Unclean leader elections
- ISR shrink/expand
- # brokers

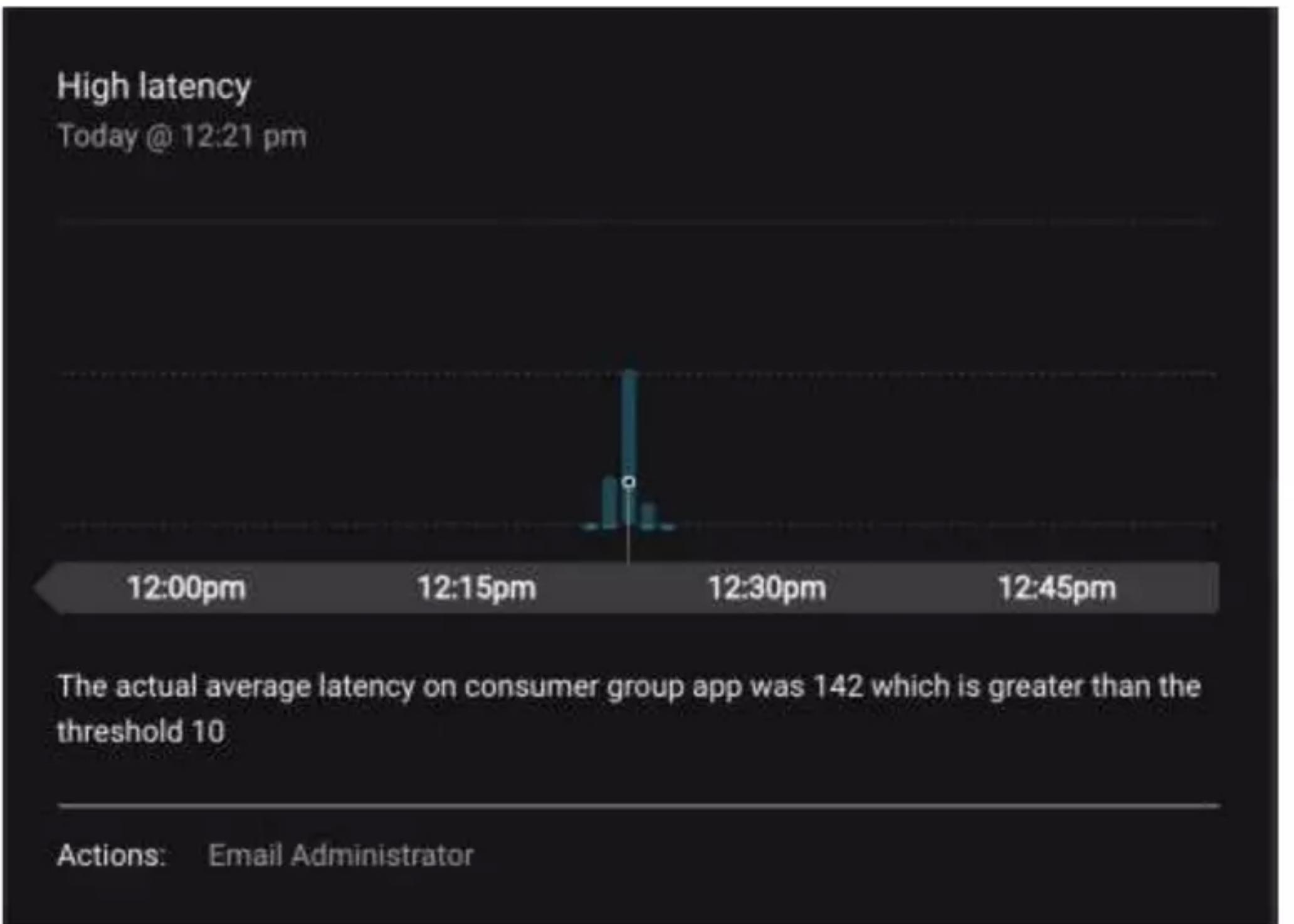
The screenshot shows the Confluent Cloud monitoring dashboard under the 'SYSTEM HEALTH' section. On the left, a sidebar lists 'MONITORING', 'MANAGEMENT', 'ALERTS', and 'INTEGRATION' sections. The 'BROKERS' tab is selected in the top navigation bar. A large circular summary indicates there are 2 brokers. Below this, a table provides detailed status for various metrics:

Metric	Status
ZK disconnected	No
Active controllers	1
Unclean elections	0
Network pool usage	1%
Request pool usage	1%
Disk usage	Even

At the bottom, a 'TOPIC PARTITIONS' section shows counts for Online (258), Under replicated (0), and Offline (0) partitions.

#3 Kafka Monitoring Tip:

Don't Watch the Dashboard



PSA regarding broker health

- Before 0.11.0.0 – restarting a broker is risky
- Even after...
- Before 1.0.0 – restarting a broker is slow
- Especially with 5000+ partitions per broker

Lesson:

Only restart if you know why this will fix the issue.

Message Delivery

Delivery Guarantees

- At most once
- At least once
- Exactly once
- ... and within N milliseconds



Are you meeting your guarantees, right now?

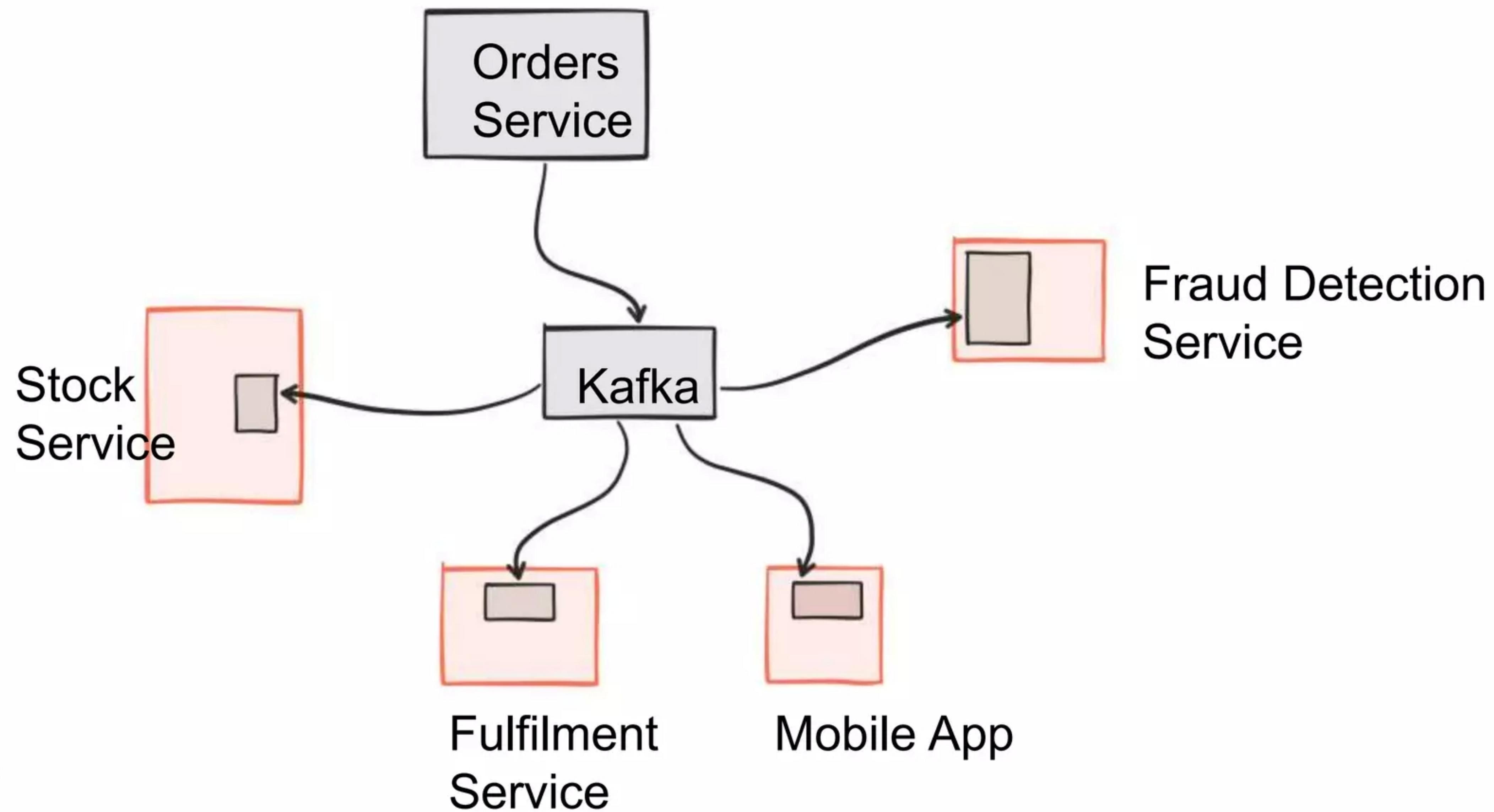
#4 Kafka Monitoring Tip:

Monitoring brokers isn't enough.

You need to monitor events



Every Service that uses Kafka is a Distributed System



How to monitor?

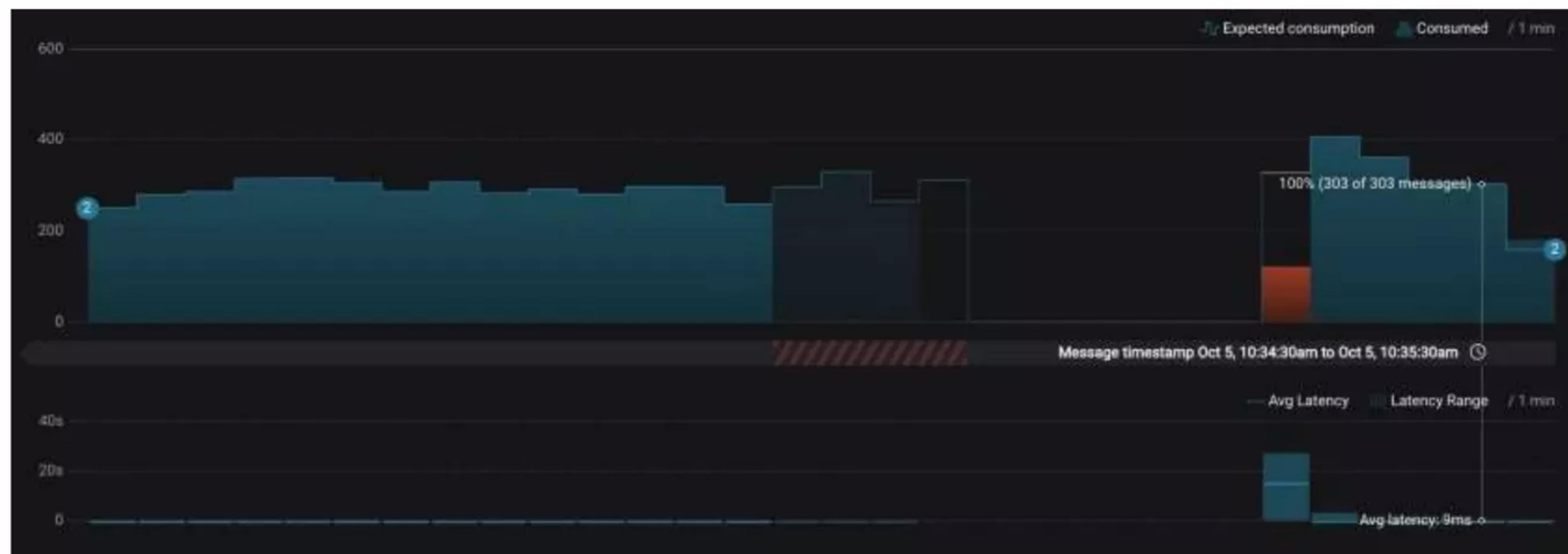
The infamous LinkedIn “Audit”:

- Count messages when they are produced
- Count messages when they are consumed
- Check timestamps when they are consumed
- Compare the results



Under Consumption

- Reasons for under consumption:
 - Producers not handling errors and retried correctly
 - Misbehaving consumers, perhaps the consumer did not follow shutdown sequence
 - Real-time apps intentionally skipping messages

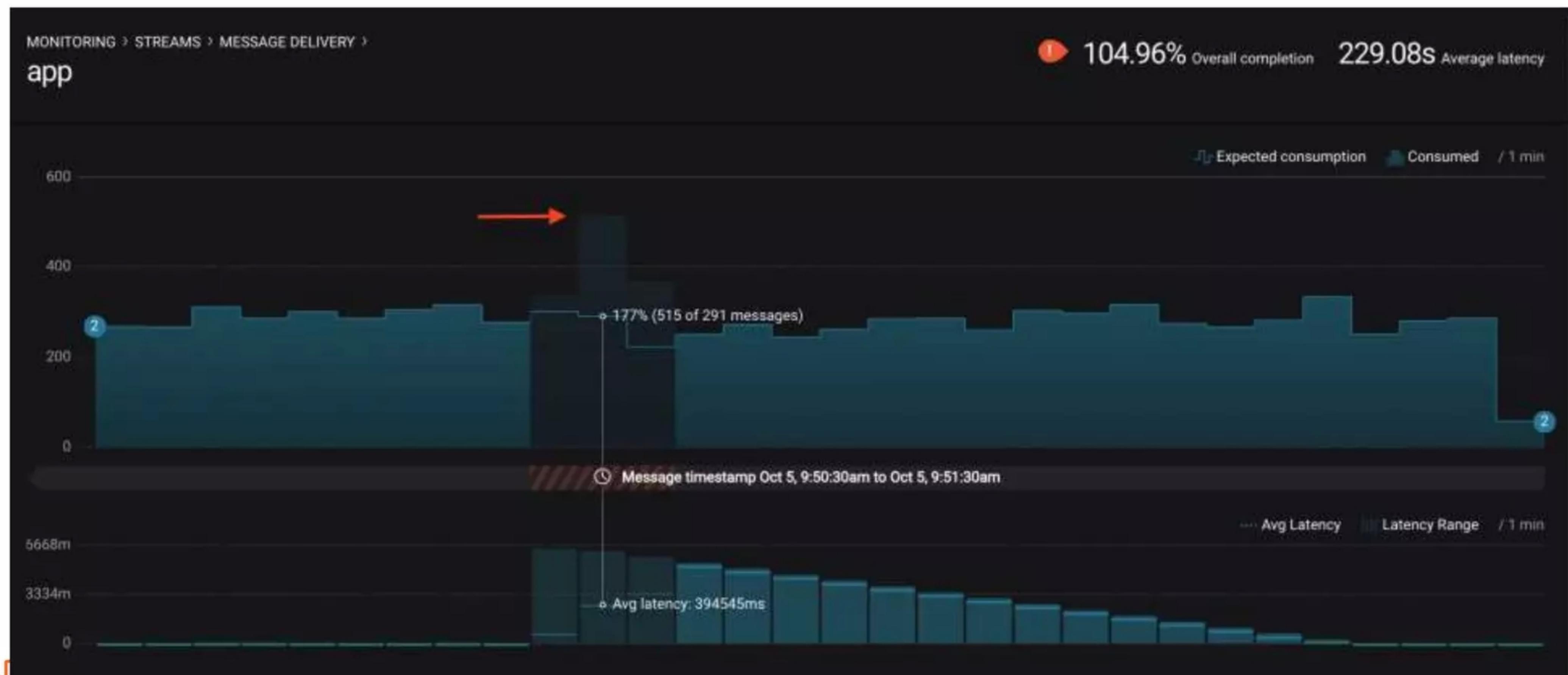


#5 Kafka Monitoring Life Tips:

- **producer.close();**
- **retries > 0**
- **Handle send() exceptions.**
- Use new consumer (0.10 and up)

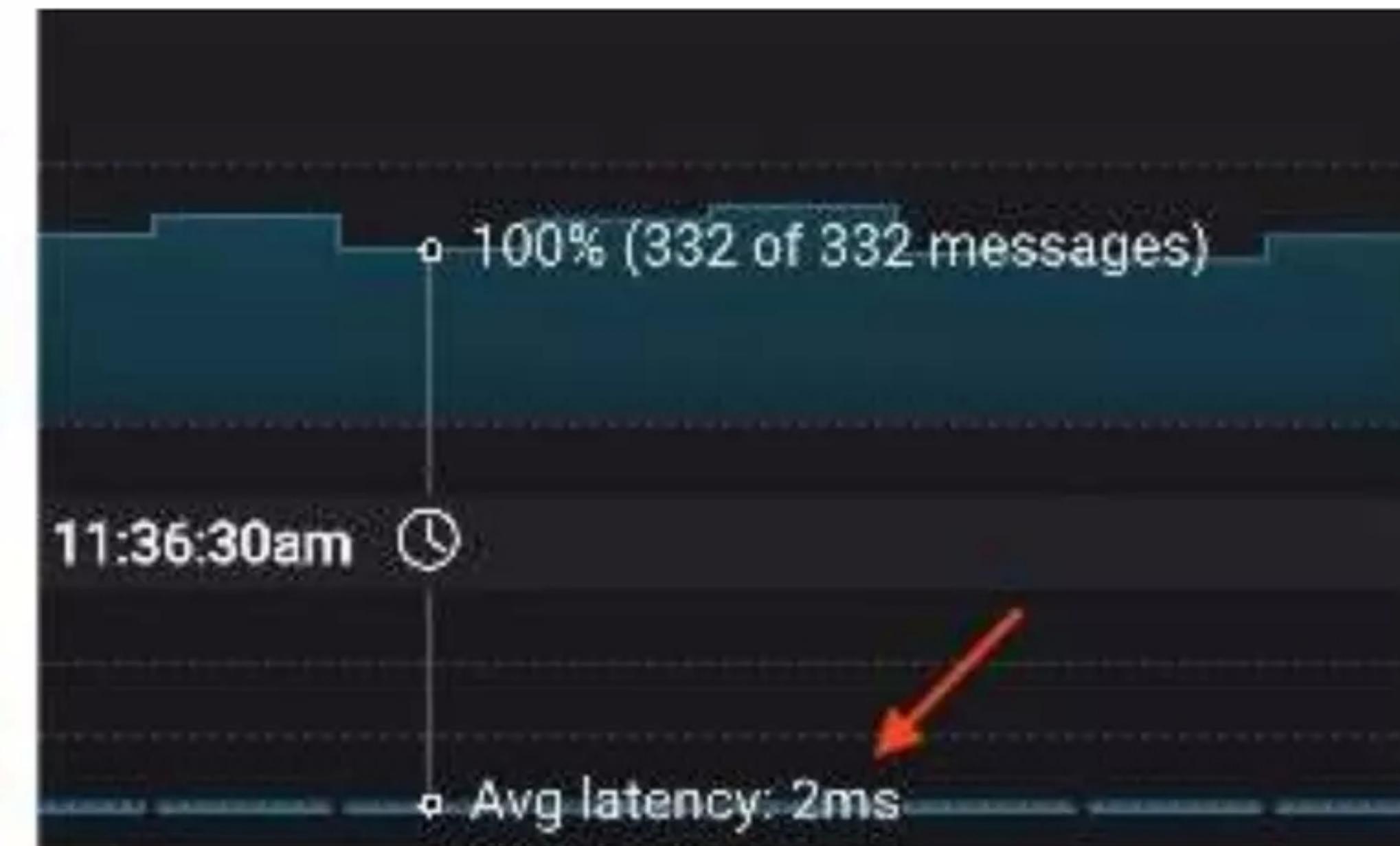
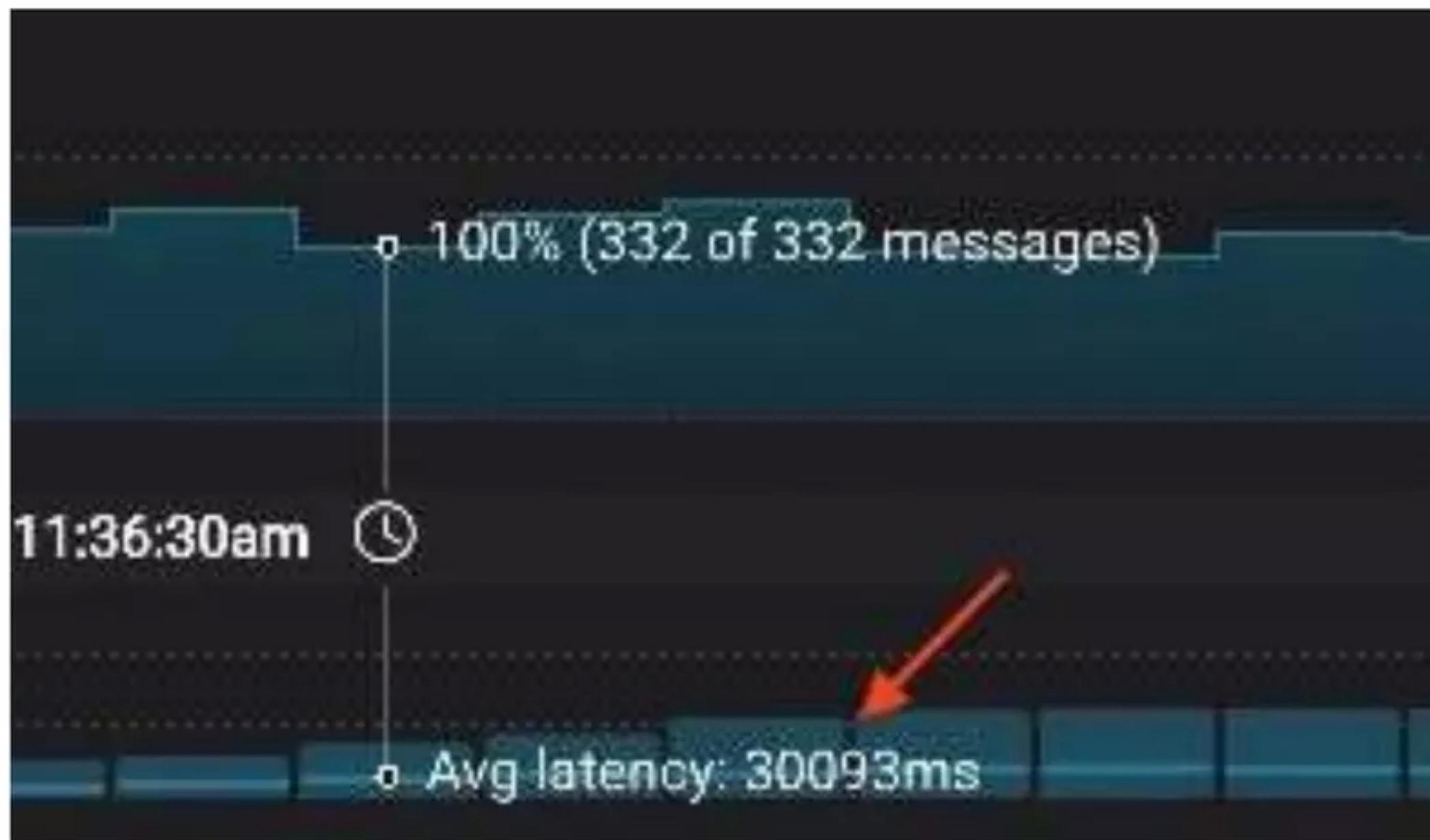
Over Consumption

- Reasons for over consumption
 - Consumers may be processing a set of messages more than once
- Latency may be higher



Slow Consumers

- Identify consumers and consumer groups that are not keeping up with data production
- Compare a slow, lagging consumer (left) to a good consumer (right)



Other important client metrics

- Producer retries
- Producer errors
- Consumer message lag – especially trends

Performance Tuning

#5 Kafka Monitoring Tip: Tune the bottlenecks



Wrong:

“OMG! Log Flush Time is 14ms”

Is this high?

How often we flush logs?

Is it blocking?

Who cares?

Right:

“OMG! We only process 50,000 requests per second. We need 10 times this”

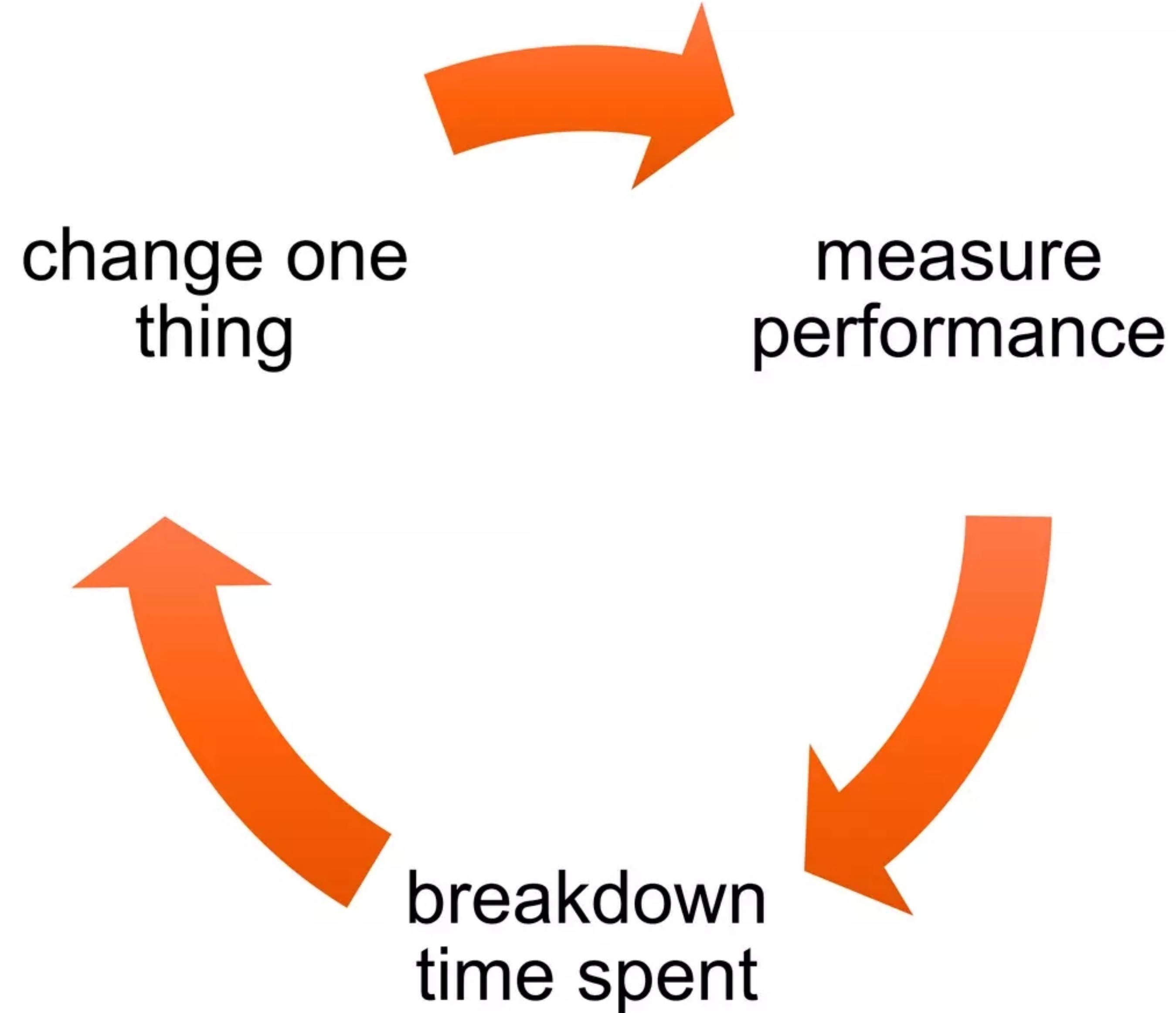
Why? IO threads are busy.

Why? Waiting for flush.

Why are we flushing so often?

Log segments keep rotating.

Lets configure larger segments!



Lifecycle of a request

- Client sends request to broker
- Network thread gets request and puts on queue
- IO thread / handler picks up request and processes
 - Read / Write from/to local “disk”
- Wait for other brokers to ack messages
- Put response on queue
- Network thread sends response to client

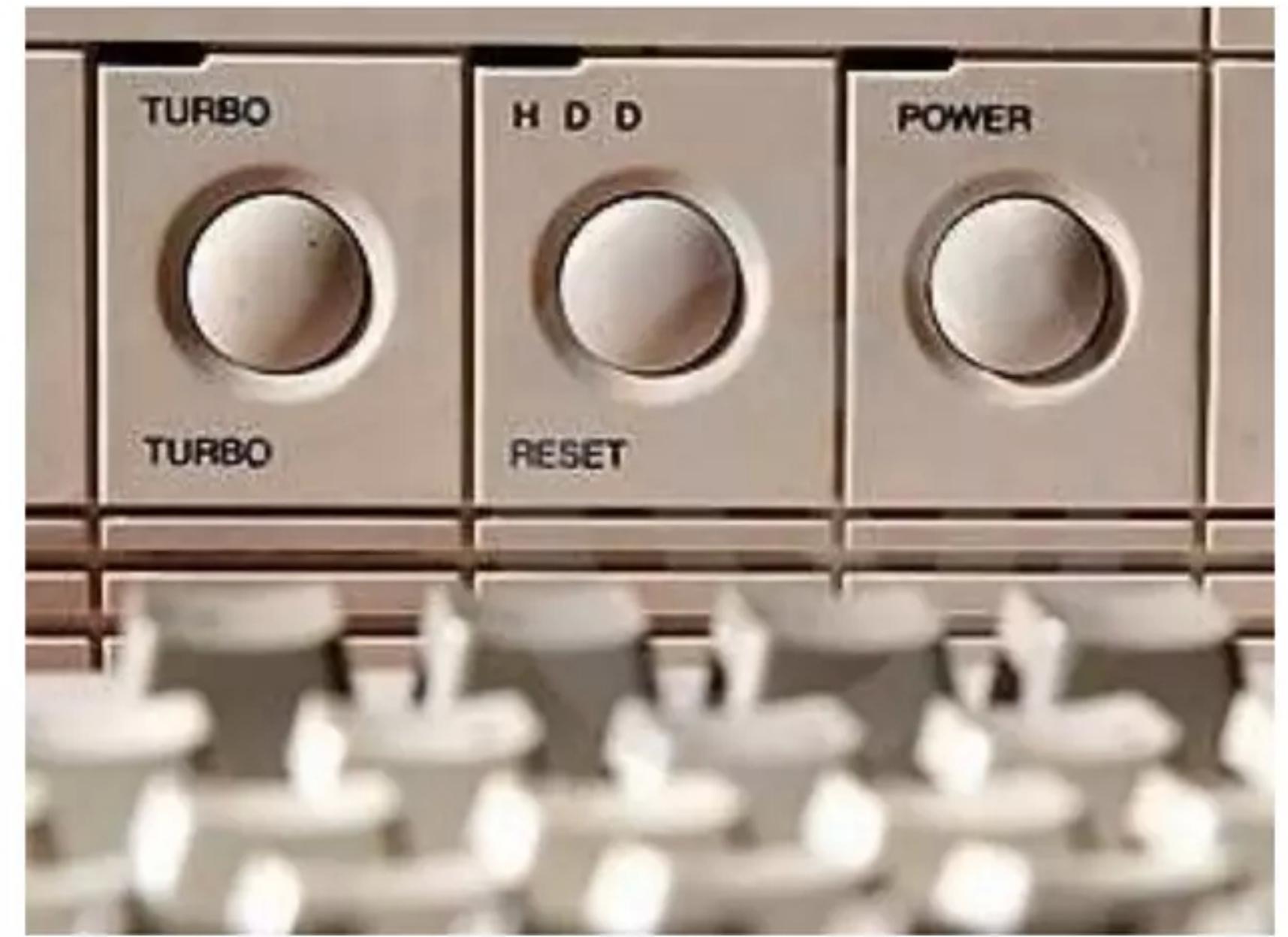
Produce and Fetch Request Latencies

- Breakdown produce and fetch latencies through the entire request lifecycle
- Each time segment correspond to a metric



How to make it faster?

- Are you network, cpu, disk bound?
- Do you need more threads?
- Where do the threads spend their time?



Capacity Planning

Key metrics that indicate a cluster is near capacity:

- CPU
- Network and thread pool usage
- Request latencies
- Network utilization
- Disk utilization

#6 Kafka Monitoring Tip:

By the time you reach 90% utilization.
It is too late.

10



"TOO LATE," said the RABBIT.

Summary:

Please Monitor Kafka.

Few things to remember...

- Alert on what's important: Under-Replicated Partitions is a good start
- DON'T JUST FIDDLE WITH STUFF
- AND DON'T RESTART KAFKA FOR LOLS
- If you don't know what you are doing, it is ok. There's support (and Cloud) for that.

We are hiring.



Thank You!
