CONFLUENT

Tips and Tricks for Apache Kafka™ Ops

Jen Snipes, Senior Customer Success Architect
https://www.linkedin.com/in/jennifersnipes/

Kat Grigg, Senior Customer Success Architect @katgrigg

June 2020

Agenda

- What is Apache Kafka
- Kafka Operations
  - Configuring
  - Deploying
  - Maintaining
  - Monitoring
  - Debugging

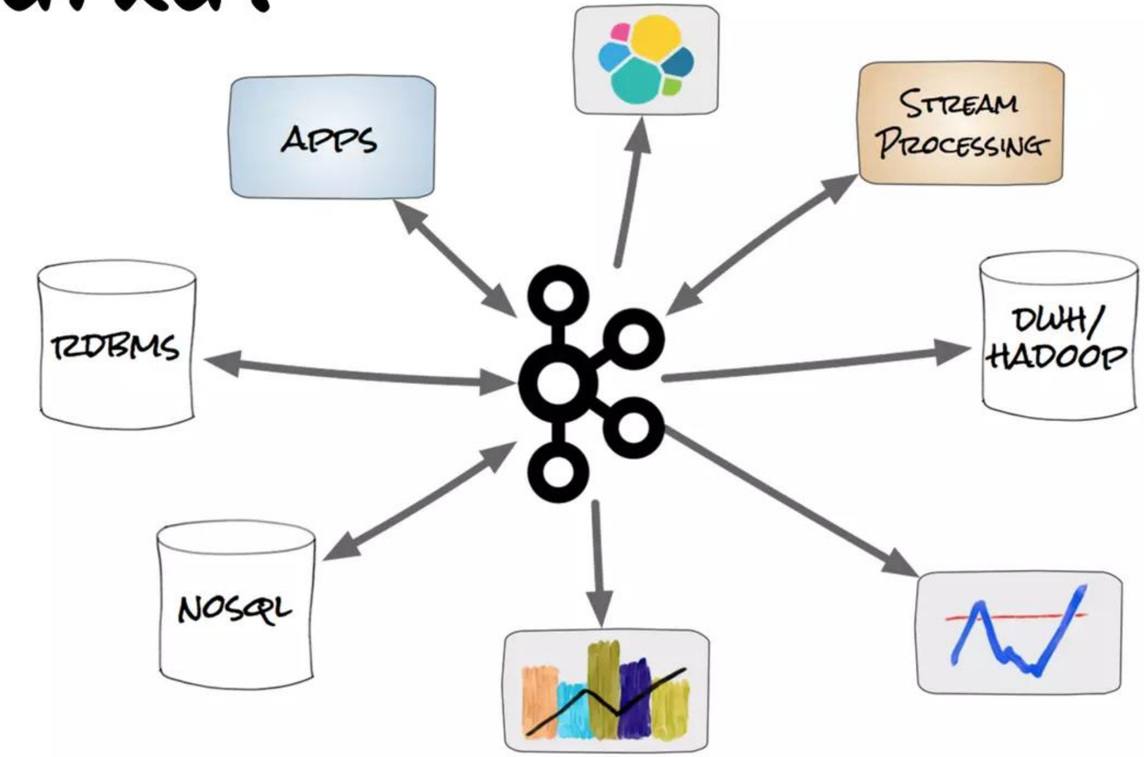
What is Apache Kafka?

Open Source

Real time Event Streaming Platform

Scalable, Distributed Log

Fault tolerant storage



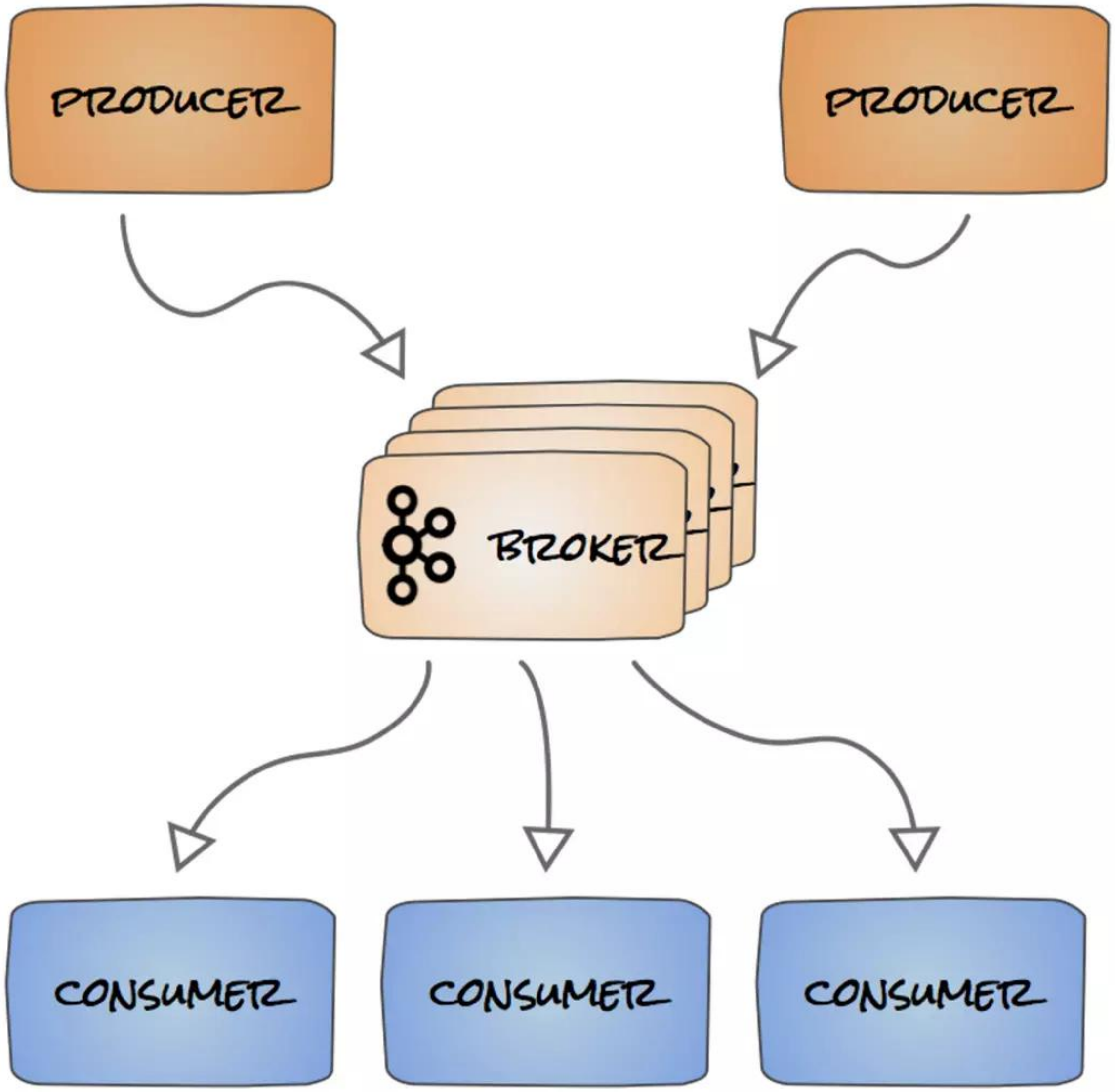
Moving Data

Producers

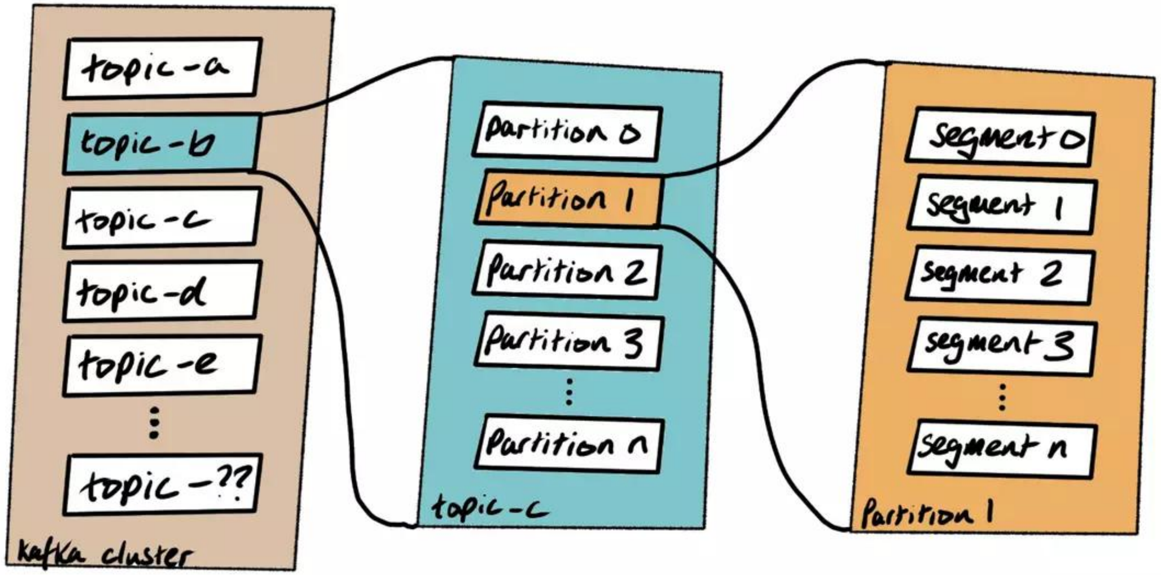
Consumers

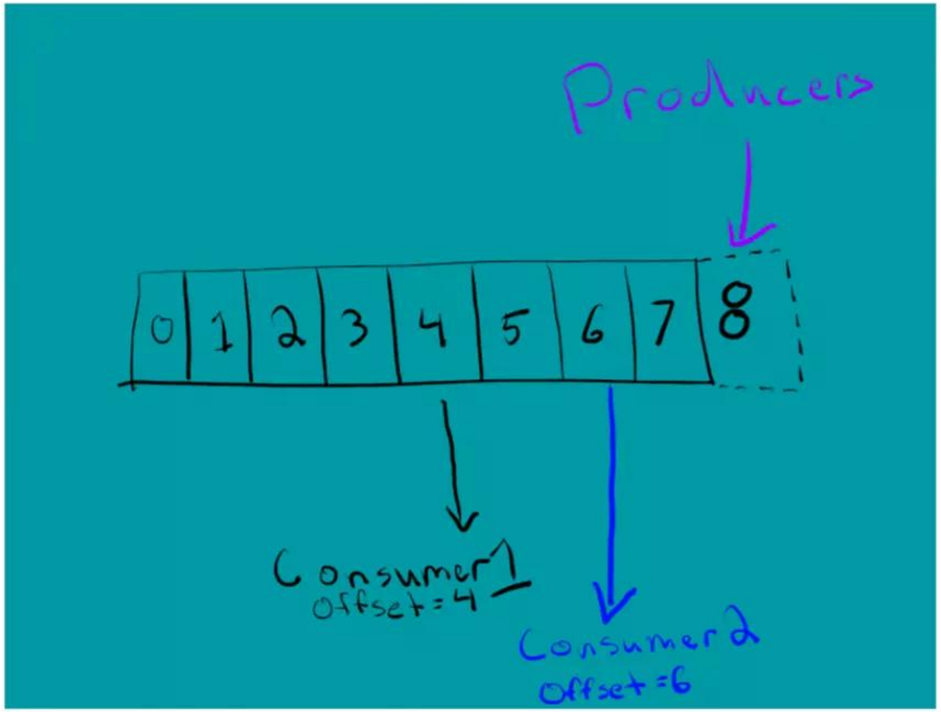
Connect

Streams



Log Structured Data Flow





Kafka Operations

5 Categories of Ops

Configuring

Deploying

Maintaining

Monitoring

Debugging

Configuring Kafka

I DONT KNOW WHAT CONFIGURATION TO CHANGE
AND AT THIS POINT I'M TOO AFRAID TO ASK
makeameme.org

Data Retention

- Disk space available?
- SLA on consuming?
- Keyed data? Cardinality?

Security

- Start with plaintext
- SSL and SASL? Choose one first
- Console clients are great for testing

Optimization

- Throughput
  - increase partitions, batch size, linger.ms
  - use acks 1
  - compression
  - parallelize consumers
- Latency
  - no compression
  - balance number of partitions
  - increase fetcher threads
  - producer linger.ms=0, consumer fetch.min.bytes=1
- Durability
  - acks = all, RF = 3, minISR = 2
  - leverage retries
  - unclean leader election set to false
  - disable auto commit

- Availability
  - minISR 1
  - unclean leader election true
  - increase recovery threads
  - low consumer session.timeout
  - optimize number of partitions

Deploying Kafka

Start Easy

- How do you run your database(s)?
- Do you know Docker well?
- Do you know Kubernetes well?
- If no, then don’t use them
- Cloud an option?

One does not simply start Brokers

- JMX access (metrics reporter)
- Environment Variables
- Logging capacity
- Don’t forget ZooKeeper
- File descriptors (ulimit)
- Disk deployment style

Maintaining Kafka

Updates and Upgrades

Rolling upgrade Tips

- Gwen Shapira's Kafka Summit Talk 2019
- Set protocols and message formats
- Be explicit
- Go slow, too fast and ISR sadness
- Upgrade often, point releases in prod

Rebalancing

- Lose/Add Broker
- Throttle early, adjust later
- Throttles don’t affect in sync replicas
- Not sure? Batch the reassign
- See Monitoring

Updating Configs

Brokers

- Lots of dynamic config options!
- Sync config across the cluster

Topics

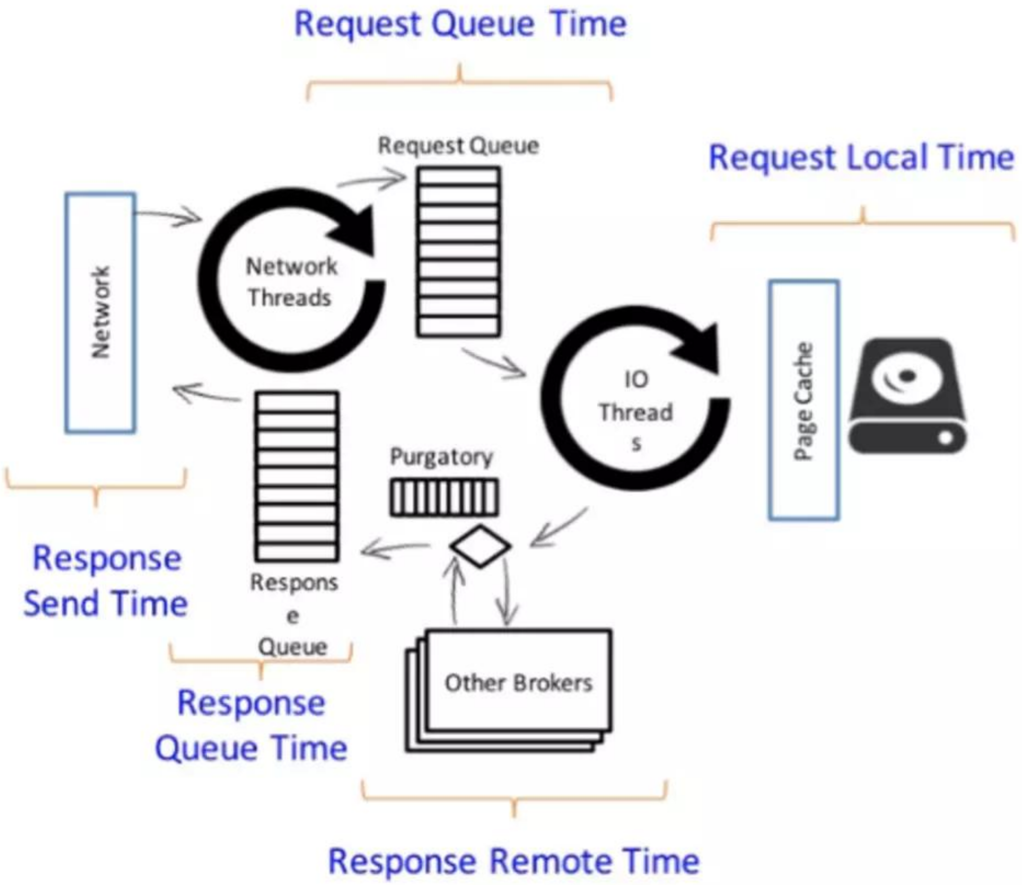
- Overrides are your friend
- Topic level unclean leader election

Monitoring Kafka

What do you watch?

Kafka Thread Model

- Monitor all of these hops with JMX
- Transparent way to see activity
- Especially watch request queue size
- Don't blindly bump thread count



Log Cleaner / Others

- Watch your log cleaner with JMX
- Bytes In/Out just makes sense
- Don’t need to dashboard everything
- Alert don’t automate (at least at first)

Broker Metrics

Top 5 Broker JMX metrics you should be watching

- UnderReplicatedPartitions
- RequestHandlerAvgIdlePercent
- NetworkProcessorAvgIdlePercent
- RequestQueueSize
- TotalTimeMs

Clients / Consumers

- batch sizes
- failed-authentication-rate
- io-ratio
- io-wait-ratio
- io-wait-time-ns-avg

Debugging Kafka

When there’s trouble…

Logs about the Log

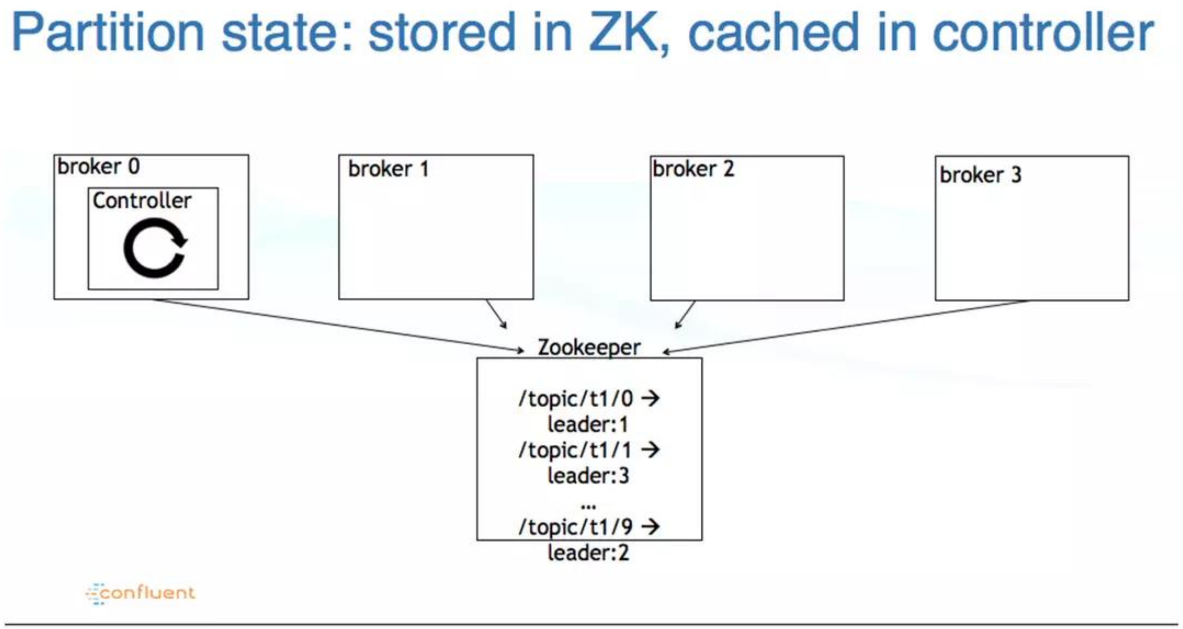
- Debugging? Head to the logs!
- Defaults separates files reasonably
- Splunk, ELK, etc.? filter by appender
- Metadata files are important too

- controller.log -- written to by active controller
- state-change.log -- updates received from and sent to controller
- server.log -- basic broker operations (segment rolls, replica fetching, etc.)
- log-cleaner.log -- cleaner thread (related to compaction)
- kafka-authorizer.log -- secure connections only (set to DEBUG if you want successful connections)
- kafka-request.log -- super verbose, will talk about more
- /var/lib/recovery-point-offset-checkpoint – last offset flushed to disk
- /var/lib/cleaner-offset-checkpoint – offset up to which the cleaner has cleaned (compacted topics only)
- /var/lib/replication-offset-checkpoint – last committed offset

Request logging

- kafka-request.log
- All information about all requests
- Verbose, not great for live debugging
- Use when no other options for root cause
- Rogue clients, single request slowdowns, etc.

Controller Issues



Partition state: stored in ZK, cached in controller

- Kafka Controller Deep Dive
- Kafka Needs No Keeper
- ISR updates aren't happening
- Replication stopped for no reason
- Broker refusing to become leader for extended period
- rmr /controller

The Offsets Topic

- __consumer_offsets → Stores offsets and group metadata
- ***bin/kafka-run-class kafka.tools.DumpLogSegments --offsets-decoder***
- offsets.retention.minutes default 1 day (7 days on 2.0)
- Growing huge? Check the log cleaner health
- High Memory Usage on Brokers? Offsets are cached
- High CPU Usage on Brokers? Make sure this topic isn't huge
- Please replication factor 3

Restarting? Think about it...

- Restarts in distributed systems are tricky
- Have a reason
- Understand startup time
- Clean shutdowns, limited scope

References

- https://www.confluent.io/resources/kafka-the-definitive-guide/
- https://www.confluent.io/kafka-summit-san-francisco-2019/please-upgrade-apache-kafka-now
- https://conferences.oreilly.com/strata/strata-ny-2018/public/schedule/detail/68957
- https://www.confluent.io/kafka-summit-san-francisco-2019/kafka-needs-no-keeper
- https://events.confluent.io/meetups

BLACK
LIVES
MATTER