AWS Kinesis

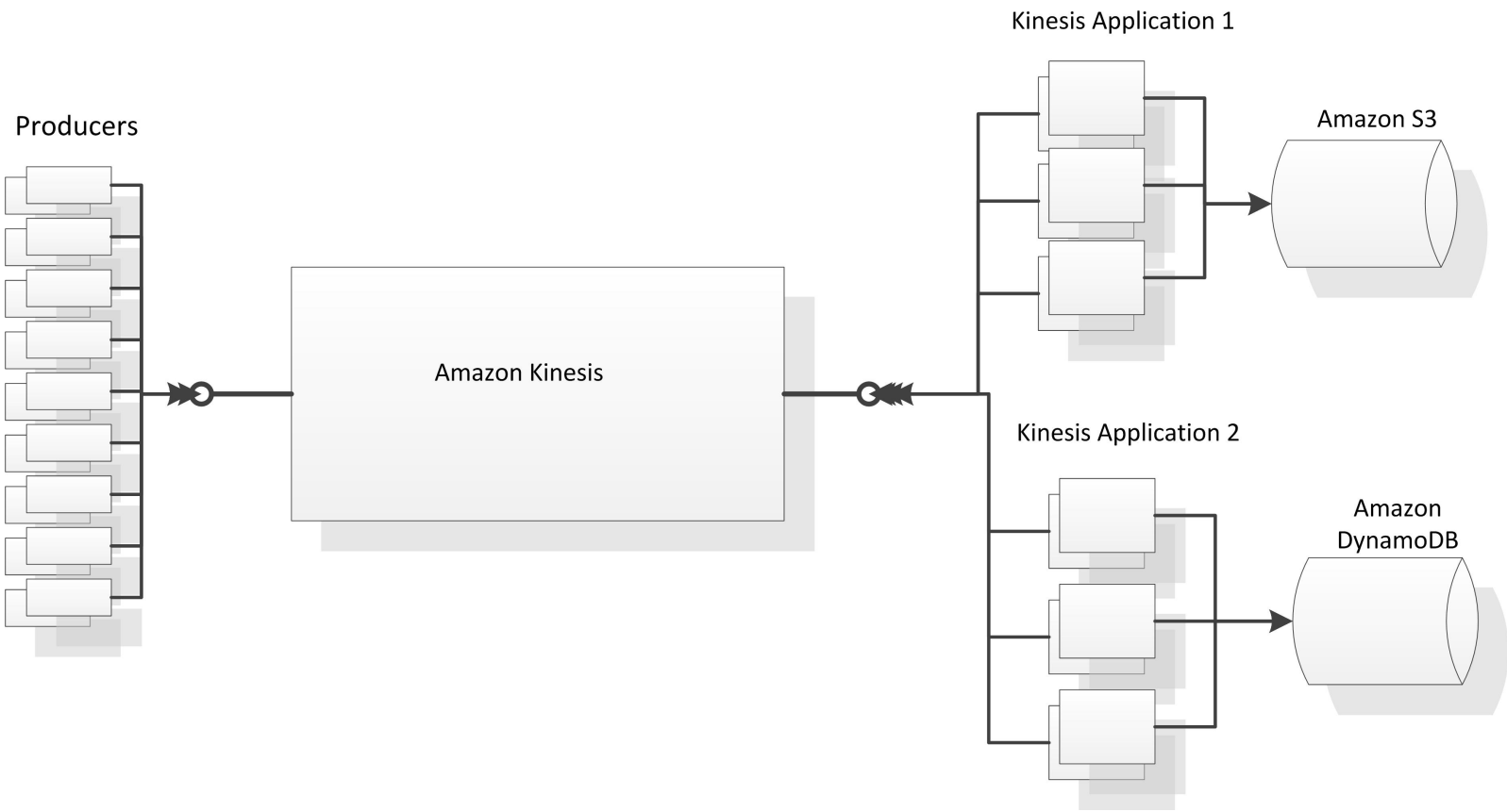
Ken Chen

2015/09/14

Agenda

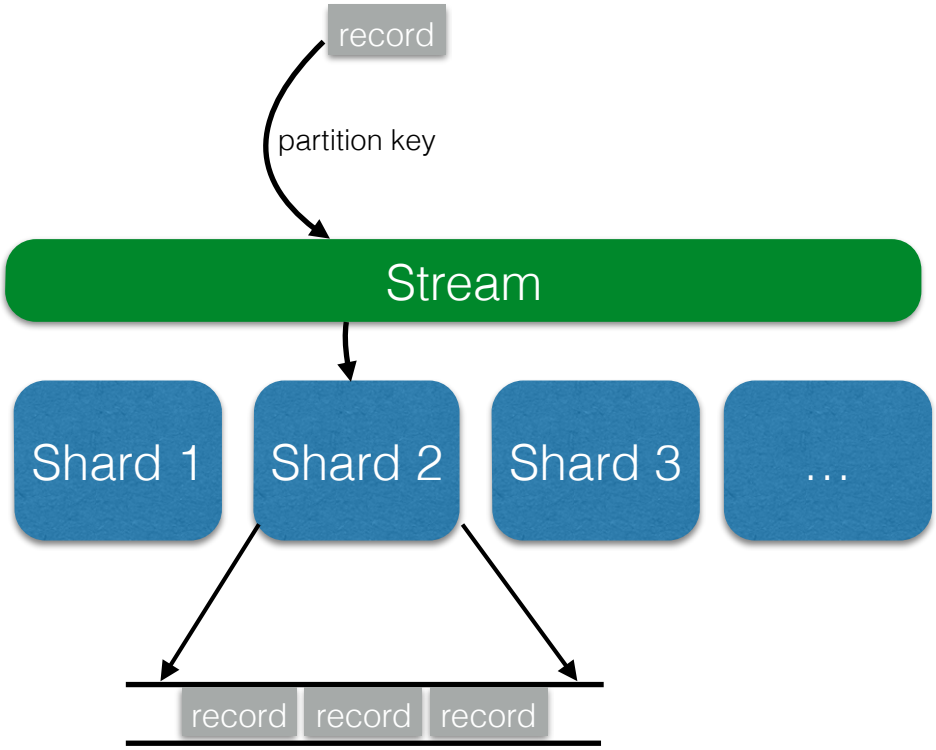
- Architecture
- Key concepts
- Features
- Limits
- Kinesis vs Kafka

Architecture



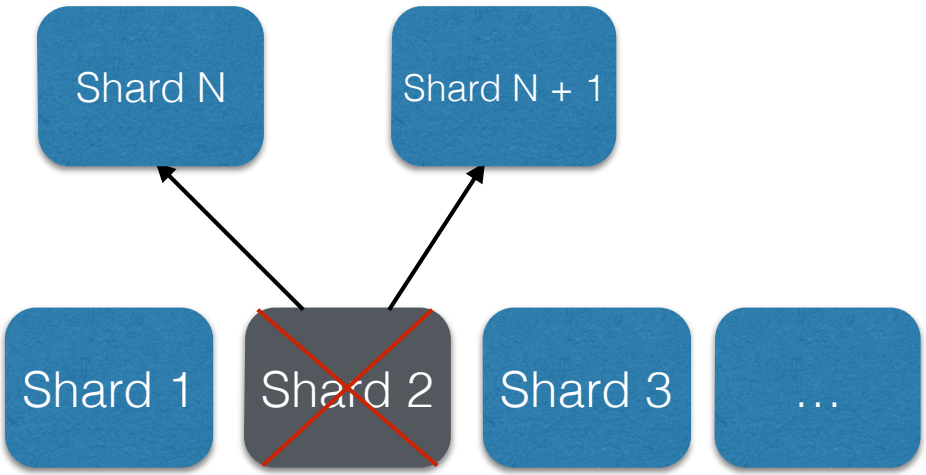
Key concepts

- Stream
- Shard
- Data record
  - UID



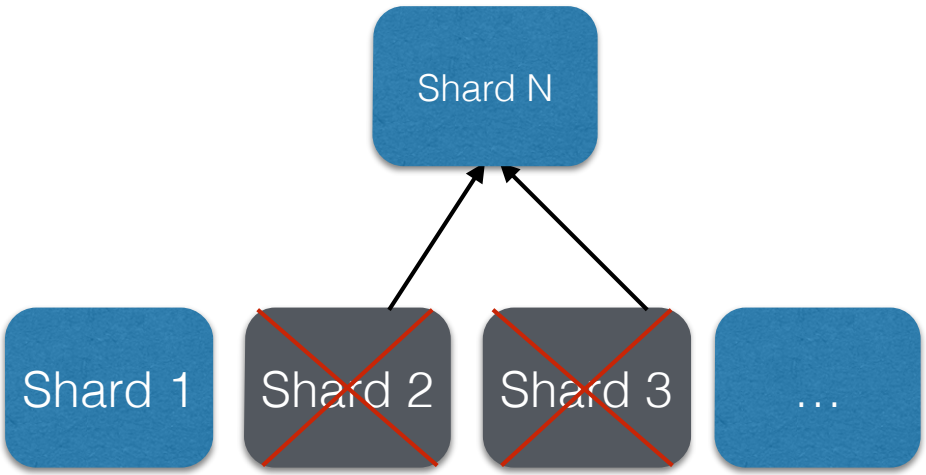
Features - Scale up

- Add more shards (shard split)
  - How producer/consumer handles shard split ? (Kinesis Client Library, KCL)



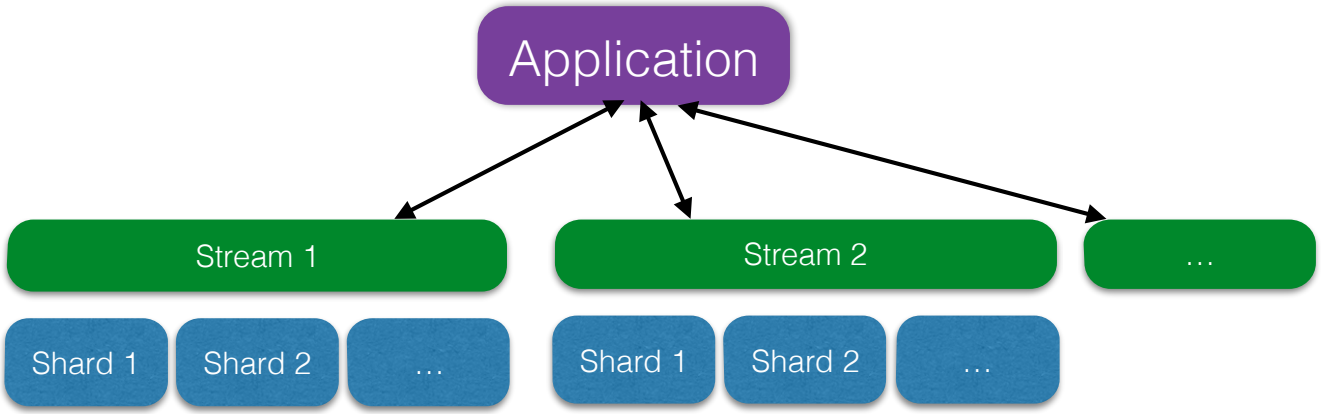
Features - Scale in

- Add more shards (shard merge)
  - How producer/consumer handles shard merge ? (Kinesis Client Library, KCL)



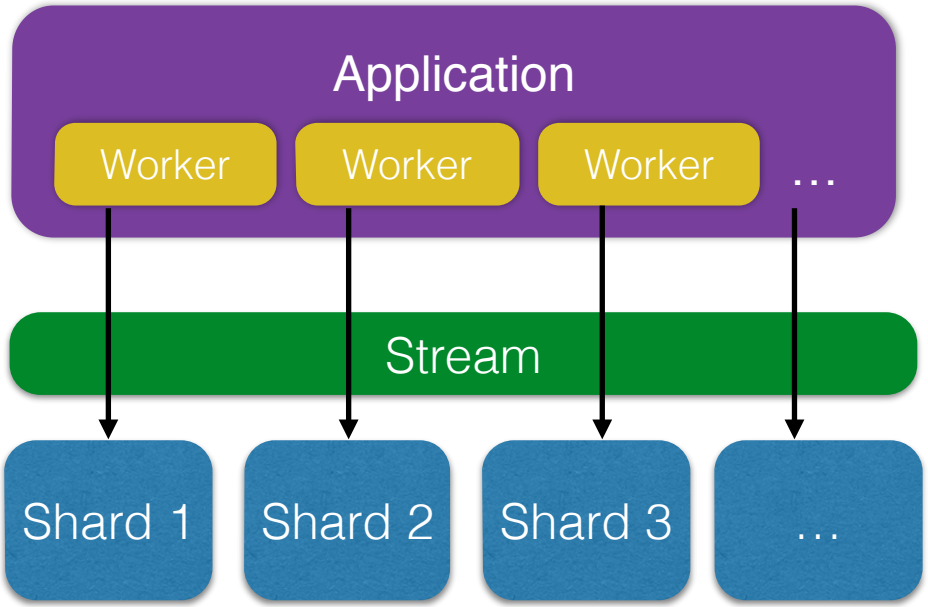
Features - Scale out

- Create more streams
  - Each stream can have up to 10 shards max



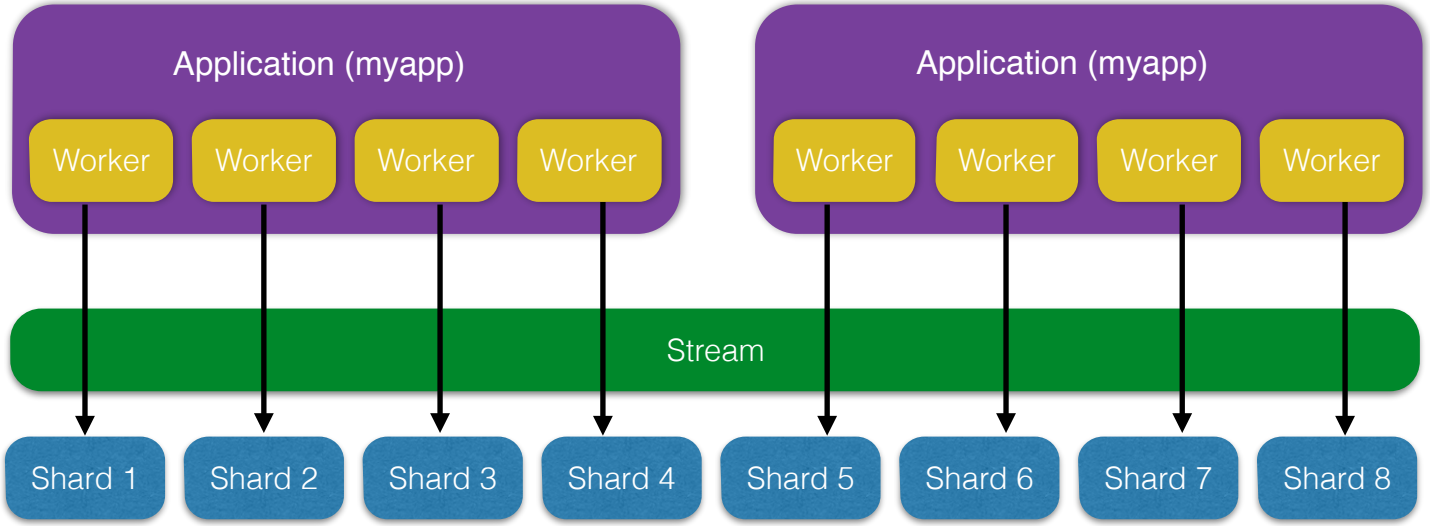
Features - Application Scale up

- Multiple-threading (KCL)
  - Consume the records in shards in parallel in one app



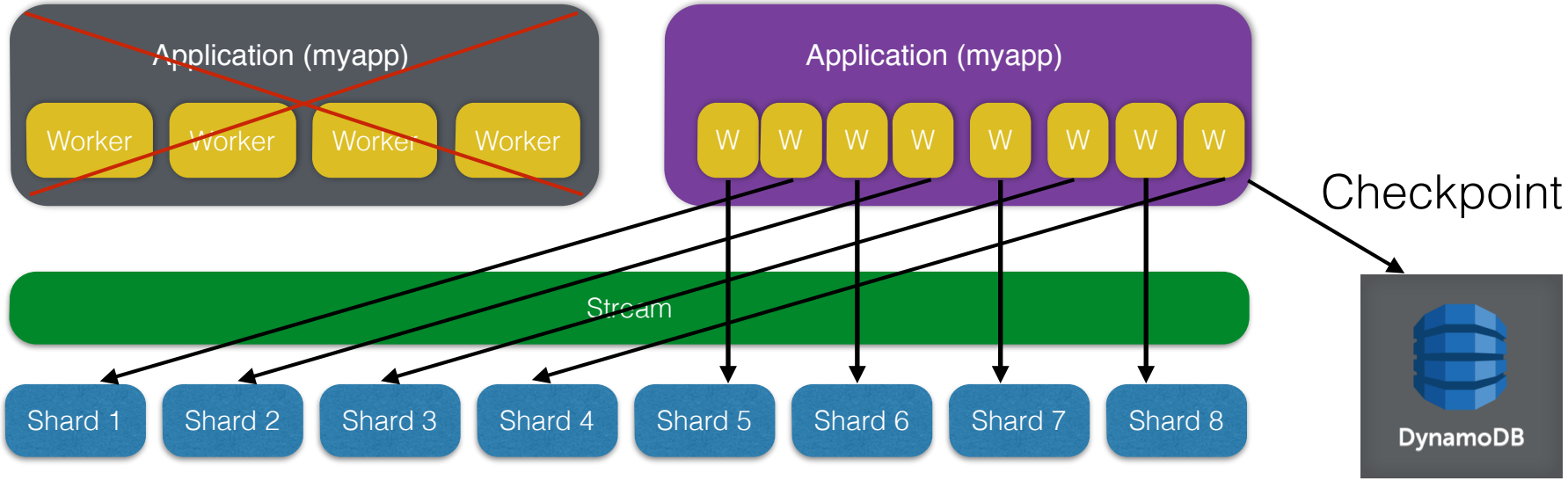
Features - Application Scale out

- Multiple-instances
  - Consume the records in shards in parallel in multi-instances (KCL auto load balance)



Features - Application Fault Tolerant

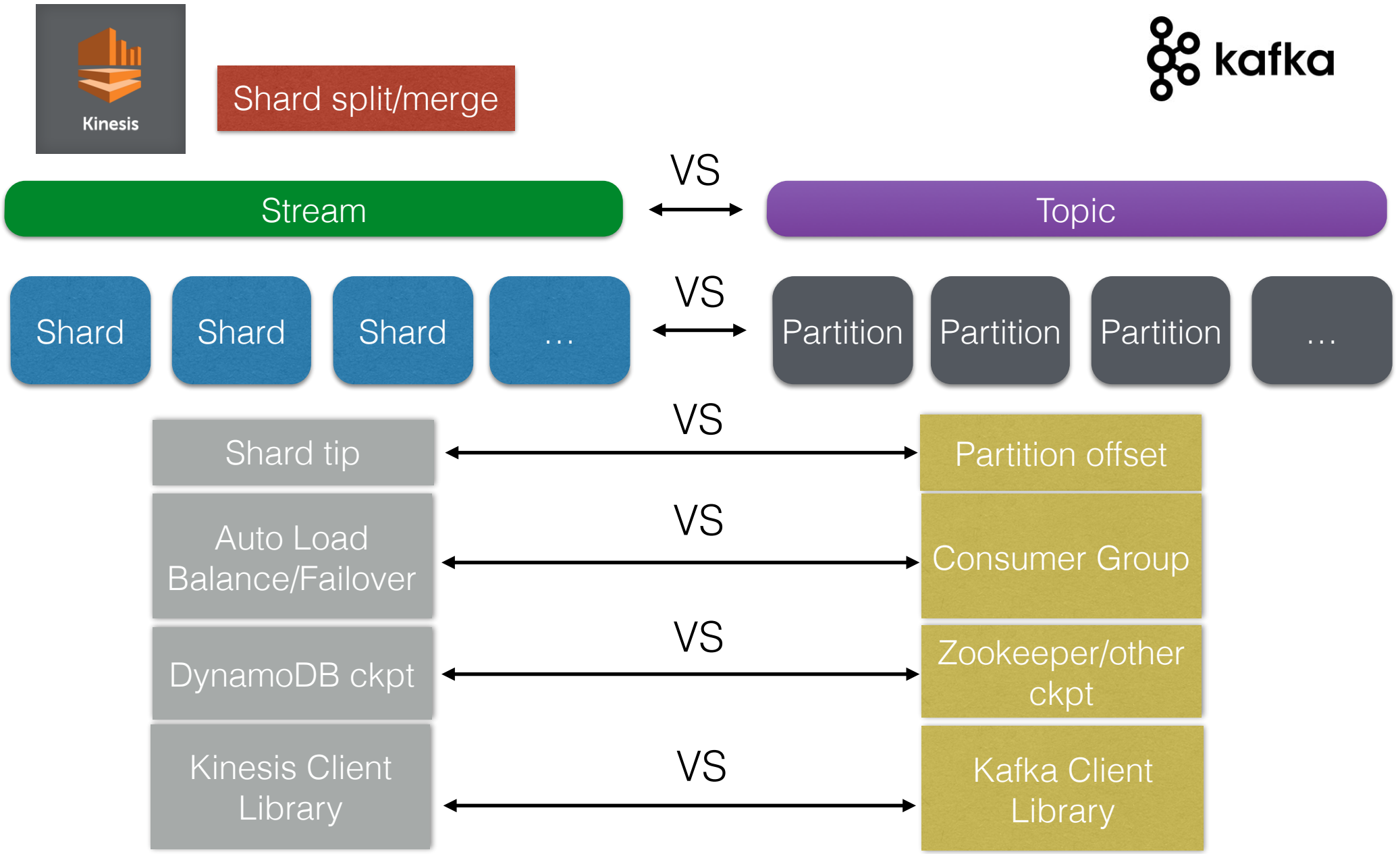
- Multiple-instances
  - Consume the records in shards in parallel in multi-instances (KCL auto load balance/failover)



Keys Limits

- 10 shards/account/region
- Data retention is 24 hours
- 5 reads/second/shard, max 2 MB/second/read
- 1000 writes/second/shard, max 1MB/second/write
- Data in and then out latency is ~5 seconds

AWS Kinesis VS Kafka



AWS Kinesis VS Kafka

Solutions	Multi-tanent	End to end Msg in then out Latency	Perf	Scale	Limits
Kinesis	Naturally support	~5s	Max of 50 MB/s in and 100 MB/s out per stream	1. Scale poorly for multiple consumers/stream (Due to 5 reads/s per shard) 2. Scale out very nicely	Max 50 shards/stream Max 5 reads/s per shard (consumers need throttle themselves)
Kafka	supported,home grown	Usually < 100 ms	Easily hit 200 MB/s in and 300 MB/s out per topic	1. Multiple consumers/topic is not a problem at all. 2. Scale out very nicely	Operation, reliabilities, monitoring, security, cross site DR etc We make the bed, then we are all on our own.