Kinesis

If you are studying for AWS Developer Associate Exam, this guide will help you with quick revision before the exam. it can use as study notes for your preparation.

Dashboard   Other Certification Notes

Kinesis

Kinesis is a managed service service which makes it easy to collect and analyze data and video streams in real time. It is considered a managed alternative for Apache Kafka.

- Kinesis Streams: low latency streaming ingest at scale
- Kinesis Analytics: real-time analytics on streams using SQL
- Kinesis Firehose: load streams into S3, Redshift, Elastic etc.

Overview

- Streams are divided in shards/partitions.
- Data is ordered in shards
- Data can be replayed
- Multiple applications can consume the same stream
- Data in Kinesis is immutable

Shards

- A stream can contain one or more shards
- Write capacity: 1MB/s or 1000 messages per shard
- Read capacity: 2MB/s per shard
- Billing is per shard
- Records are ordered per shard

Kines API

- PutRecord API: requires a partition key which gets hashed.
- Partition key should be highly distributed (otherwise a shard can become overwhelmed)
- PutRecord accepts batches, cost can be reduced
- PutRecord throws ProvisionedThroughputExceeded in case of capacity is exceeded. Solution: exponential back-off, more shards, ensure that partition key is highly distributed

Kinesis Client Library (KCL)

- It is a Java library
- Each shard should be read by only one KCL instance!
- Progress checkpoint is written to DynamoDB
- Records are read in order at the shard level

Security

- IAM policies
- Encryption at flight using HTTPS
- Encryption at rest using KMS
- Client side encryption/decryption (should be done manually)
- VPC Endpoints available for Kinesis

Kinesis Data Analytics

- Real-time analytics on Kinesis Streams using SQL
- Pay for actual consumption, no need to pre-provision anything
- Can create streams from query results

Kinesis Firehose

Kinesis Firehose

- Used for ETL, can load data into Redshift, S3, ElasticSearch etc.
- Fully managed, automatic scaling
- Low latency (60 seconds)
- Pay for the amount of data which goes through Firehose

SQS vs SNS vs Kinesis

SQS	SNS	Kinesis
Consumer pulls data	Data is pushed to many subscribers	Consumer pulls data
Data is deleted after consumed	Up to 10 million subscribers	Can have as many consumer as we want
Can have as many consumer as we want	Data os not persisted	Can replay data
No need to provision throughput	Pub/Sub	Used for big data analytics and ETL
No ordering (except FIFO)	Up to 100k topics	Data expires after X days
Individual message delay capability	No need to provision throughput	Must provision throughput
	Integrates with SQS for fan-out architecture	Ordering at the shard level

Made with 💙 by **Nirav Kanani**   **Contact Us**