AIrflow

Orchestration Service

JERIN SAM

Table of Contents

[Kafka 4](#_Toc181447447)

[Test Kafka executing below commands 4](#_Toc181447448)

[Create new topic 4](#_Toc181447449)

[List kafka topic 4](#_Toc181447450)

[Console kafka producer - Starts console producer 4](#_Toc181447451)

[Console kafka Consumer - Starts console consumer 5](#_Toc181447452)

[Kafka Architecture 5](#_Toc181447453)

[Topics, partitions, and offsets in Kafka 6](#_Toc181447454)

[Kafka Cluster 8](#_Toc181447455)

[Topic with Replication in Multiple Broker Kafka Cluster 10](#_Toc181447456)

[Unclean Leader Assignment Intuition 14](#_Toc181447457)

[In-depth Intuition on Kafka Rack Awareness 17](#_Toc181447458)

[Kafka Log Segments 19](#_Toc181447459)

[Kafka Producer Key & Message Acknowledgements 22](#_Toc181447460)

[Producer Keys 22](#_Toc181447461)

[Message Acknowledgements 23](#_Toc181447462)

[Minimum In Sync Replica 24](#_Toc181447463)

[Kafka Producer Internals 27](#_Toc181447464)

[Kafka Producer Properties 29](#_Toc181447465)

[Buffer and Batch 29](#_Toc181447466)

[Retry Mechanism 30](#_Toc181447467)

[Primary methods of sending messages 33](#_Toc181447468)

[1. Fire and Forget 33](#_Toc181447469)

[2. Synchronous Send 33](#_Toc181447470)

[3. Asynchronous Send 34](#_Toc181447471)

[Kafka Topic Partitioning Strategy when Key is NULL 34](#_Toc181447472)

[Consumer & Consumer Group 35](#_Toc181447473)

[Consumer Offset Commit 38](#_Toc181447474)

[Kafka Consumer Internals 41](#_Toc181447475)

[Consumer - Manual Offset Commits & At Least Once Processing 44](#_Toc181447476)

[Consumer - Manual Offset Commits & Exactly Once Processing 45](#_Toc181447477)

[Consumer - Manual Offset Commits & At-Most Once Processing 47](#_Toc181447478)

[Manual Offset Commit Python– Exactly Once Processing Approach 49](#_Toc181447479)

[Exactly Once Processing 49](#_Toc181447480)

[Consumer Rebalancing 49](#_Toc181447481)

[Kafka Partition Assignment Strategies Across Multiple Consumers within same Consumer Group 50](#_Toc181447482)

[RangePartitionAssignor 50](#_Toc181447483)

[RoundRobinPartitionAssignor 51](#_Toc181447484)

[Python Code 52](#_Toc181447485)

[Consumer Lag Analysis 53](#_Toc181447486)

[Why don’t Kafka allow multiple consumers within Same Group to consume messages from same partition? 55](#_Toc181447487)

[Kafka Log Directory - .index and .timeindex files 56](#_Toc181447488)

[Index Files 56](#_Toc181447489)

[Timeindex File 59](#_Toc181447490)

[Multiple log, Index & timestamp File 60](#_Toc181447491)

[Offset Store in Index File 61](#_Toc181447492)

[Log Retention and Topic Log Compaction 62](#_Toc181447493)

[Log Retention 62](#_Toc181447494)

[Log Compaction 63](#_Toc181447495)

[Schema Registry in Kafka 65](#_Toc181447496)

[Need to Validate Message’s schema 65](#_Toc181447497)

[Schema Registry 66](#_Toc181447498)

[Components of Schema Registry: 66](#_Toc181447499)

[Strategy​ 69](#_Toc181447500)

[Schema Registry Architecture: 70](#_Toc181447501)

[Schema Evolution: 71](#_Toc181447502)

[Schema Registry Python Script 76](#_Toc181447503)

[Kafka Connect 76](#_Toc181447504)

[Connector Configuration Intuition 77](#_Toc181447505)

[Architecture of Kafka Connect 78](#_Toc181447506)

[Kafka Connect Hands-On 80](#_Toc181447507)

[Kafka Rebalance Listener 81](#_Toc181447508)

[Rebalance Listener Implementation 82](#_Toc181447509)

[Error Handling in Kafka Producer 83](#_Toc181447510)

[Handling Message Errors using Dead Letter Queues 86](#_Toc181447511)

[Confluent Rest Proxy 88](#_Toc181447512)

[REST Proxy Implementation 89](#_Toc181447513)

[Kafka Stream Processing in Python using Faust 92](#_Toc181447514)

[Appendix 102](#_Toc181447515)

[Kafka API – Python Code 102](#_Toc181447516)

[Kafka Python References 103](#_Toc181447517)

[Producer API - Python 104](#_Toc181447518)

[Consumer API - Python 105](#_Toc181447519)

[Errors 107](#_Toc181447520)

[Kafka Listener and Advertised Listener 107](#_Toc181447521)

[Kafka Advertised Listener Host IP 109](#_Toc181447522)

# Airflow

Airflow is an orchestrator, not a processing framework. Process your gigabytes of data outside of Airflow (i.e. You have a Spark cluster, you use an airflow operator to execute a Spark job, and the data is processed in Spark).

Components of Airflow:

1. ***DAG*** is a data pipeline.
2. ***Operator*** is a task.
3. ***Executor*** defines how your tasks are executed.
4. ***Worker*** is a process executing your task.
5. ***Scheduler*** schedules your tasks
6. ***Web Server*** serves the UI
7. ***Database*** stores the metadata of Airflow.