WHAT IS APACHE KAFKA?

May 10, 2017

Share Tweet

Like 49 Share

What is Kafka?

Kafka’s growth is exploding (http://www.techrepublic.com/article/an-inside-look-at-why-apache-kafka-adoption-is-exploding/), more than $^1/_3$ of all Fortune 500 companies use Kafka. These companies includes the top ten travel companies, 7 of top ten banks, 8 of top ten insurance companies, 9 of top ten telecom companies, and much more. LinkedIn, Microsoft and Netflix process four comma messages a day with Kafka (1,000,000,000,000). Kafka is used for real-time streams of data, used to collect big data or to do real time analysis or both). Kafka is used with in-memory microservices to provide durability and it can be used to feed events to CEP (https://en.wikipedia.org/wiki/Complex_event_processing) (complex event streaming systems), and IOT/IFTTT style automation systems.

> Cloudurable provides Kafka training (http://cloudurable.com/kafka-training/index.html), Kafka consulting (http://cloudurable.com/kafka-aws-consulting/index.html), Kafka support (http://cloudurable.com/subscription_support/index.html) and helps setting up Kafka clusters in AWS (http://cloudurable.com/services/index.html).

Why Kafka?

Kafka often gets used in the real-time streaming data architectures to provide real-time analytics. Since Kafka is a fast, scalable, durable, and fault-tolerant publish-subscribe messaging system, Kafka is used in use cases where JMS, RabbitMQ, and AMQP may not even be considered due to volume and responsiveness. Kafka has higher throughput, reliability and replication characteristics which make it applicable for things like tracking service calls (tracks every call) or track IOT sensors data where a traditional MOM might not be considered.

Kafka (http://cloudurable.com/kafka-training/index.html) can works with Flume/Flafka, Spark Streaming (http://cloudurable.com/spark-aws-emr-training/index.html), Storm, HBase, Flink and Spark for real-time ingesting, analysis and processing of streaming data. Kafka is a data stream used to feed Hadoop BigData lakes. Kafka brokers support massive message streams for low-latency follow-up analysis in Hadoop or Spark. Also, Kafka Streaming (https://speakerdeck.com/aseigneurin/kafka-summit-introduction-to-kafka-streams-with-a-real-life-example) (a subproject) can be used for real-time analytics.

Kafka use cases

In short, Kafka gets used for stream processing, website activity tracking, metrics collection and monitoring, log aggregation, real-time analytics, CEP, ingesting data into Spark, ingesting data into Hadoop, CQRS (https://en.wikipedia.org/wiki/Command%E2%80%93query_separation#Command_query_responsibility_segregation), replay messages, error recovery, and guaranteed distributed commit log for in-memory computing (microservices).

Who uses Kafka?

A lot of large companies (https://cwiki.apache.org/confluence/display/KAFKA/Powered+By) who handle a lot of data use Kafka. LinkedIn, where it originated, uses it to track activity data and operational metrics. Twitter uses it as part of Storm to provide a stream processing infrastructure. Square uses Kafka as a bus to move all system events to various Square data centers (logs, custom events, metrics, and so on), outputs to Splunk, Graphite (dashboards), and to implement an Esper-like/CEP (http://www.espertech.com/esper/) alerting systems. It gets used by other companies too like Spotify, Uber, Tumbler, Goldman Sachs, PayPal, Box, Cisco, CloudFlare, NetFlix, and much more.

Why is Kafka so popular?

Kafka has operational simplicity. Kafka is to set up and use, and it is easy to reason how Kafka works. However, the main reason Kafka is very popular is its excellent performance. It has other characteristics as well, but so do other messaging systems. Kafka has great performance, and it is stable, provides reliable durability, has a flexible publish-subscribe/queue that scales well with N-number of consumer groups, has robust replication, provides Producers with tunable consistency guarantees, and it provides preserved ordering at shard level (Kafka Topic Partition). In addition, Kafka works well with systems that have data streams to process and enables those systems to aggregate, transform & load into other stores. But none of those characteristics would matter if Kafka was slow. The most important reason Kafka is popular is Kafka's exceptional performance.

> Cloudurable provides Kafka training (http://cloudurable.com/kafka-training/index.html), Kafka consulting (http://cloudurable.com/kafka-aws-consulting/index.html), Kafka support (http://cloudurable.com/subscription_support/index.html) and helps setting up Kafka clusters in AWS (http://cloudurable.com/services/index.html).

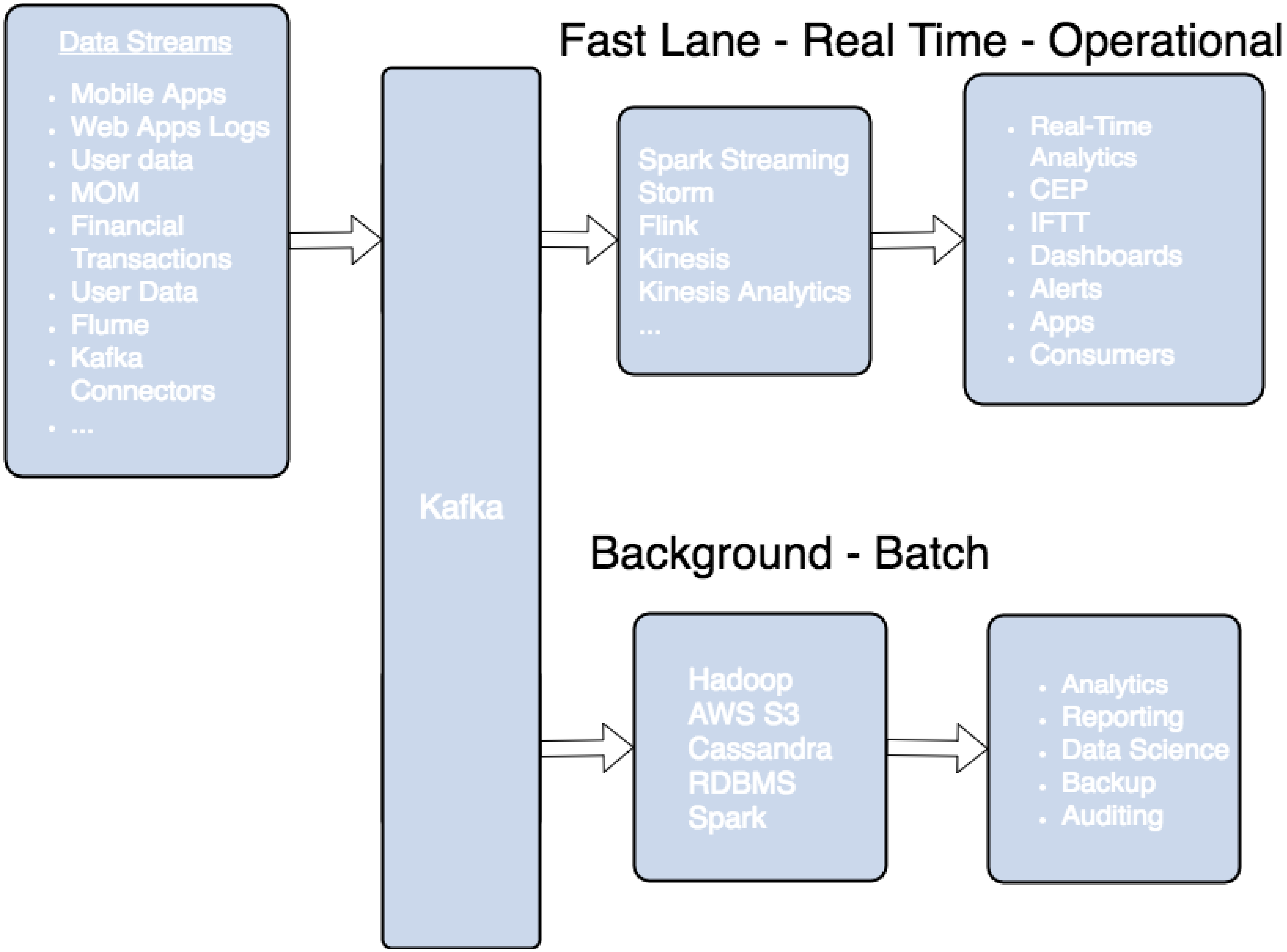
Why is Kafka so Fast?

Kafka relies heavily on the OS kernel to move data around quickly. It relies on the principals of *Zero Copy (https://en.wikipedia.org/wiki/Zero-copy)*. Kafka enables you to batch data records into chunks. These batches of data can be seen end to end from Producer to file system (Kafka Topic Log) to the Consumer. Batching allows for more efficient data compression and reduces I/O latency. Kafka writes to the immutable commit log to the disk sequential; thus, avoids random disk access, slow disk seeking. Kafka provides horizontal Scale through sharding. It shards a Topic Log into hundreds potentially thousands of partitions to thousands of servers. This sharding allows Kafka to handle massive load.

Kafka: Streaming Architecture

Kafka gets used most often for real-time streaming of data into other systems. Kafka is a middle layer to decouple your real-time data pipelines. Kafka core is not good for direct computations such as data aggregations, or CEP. *Kafka Streaming* which is part of the Kafka ecosystem does provide the ability to do real-time analytics. Kafka can be used to feed fast lane systems (real-time, and operational data systems) like Storm, Flink, Spark Streaming and your services and CEP systems. Kafka is also used to stream data for batch data analysis. Kafka feeds Hadoop. It streams data into your BigData platform or into RDBMS, Cassandra, Spark, or even S3 for some future data analysis. These data stores often support data analysis, reporting, data science crunching, compliance auditing, and backups.

Kafka Streaming Architecture Diagram

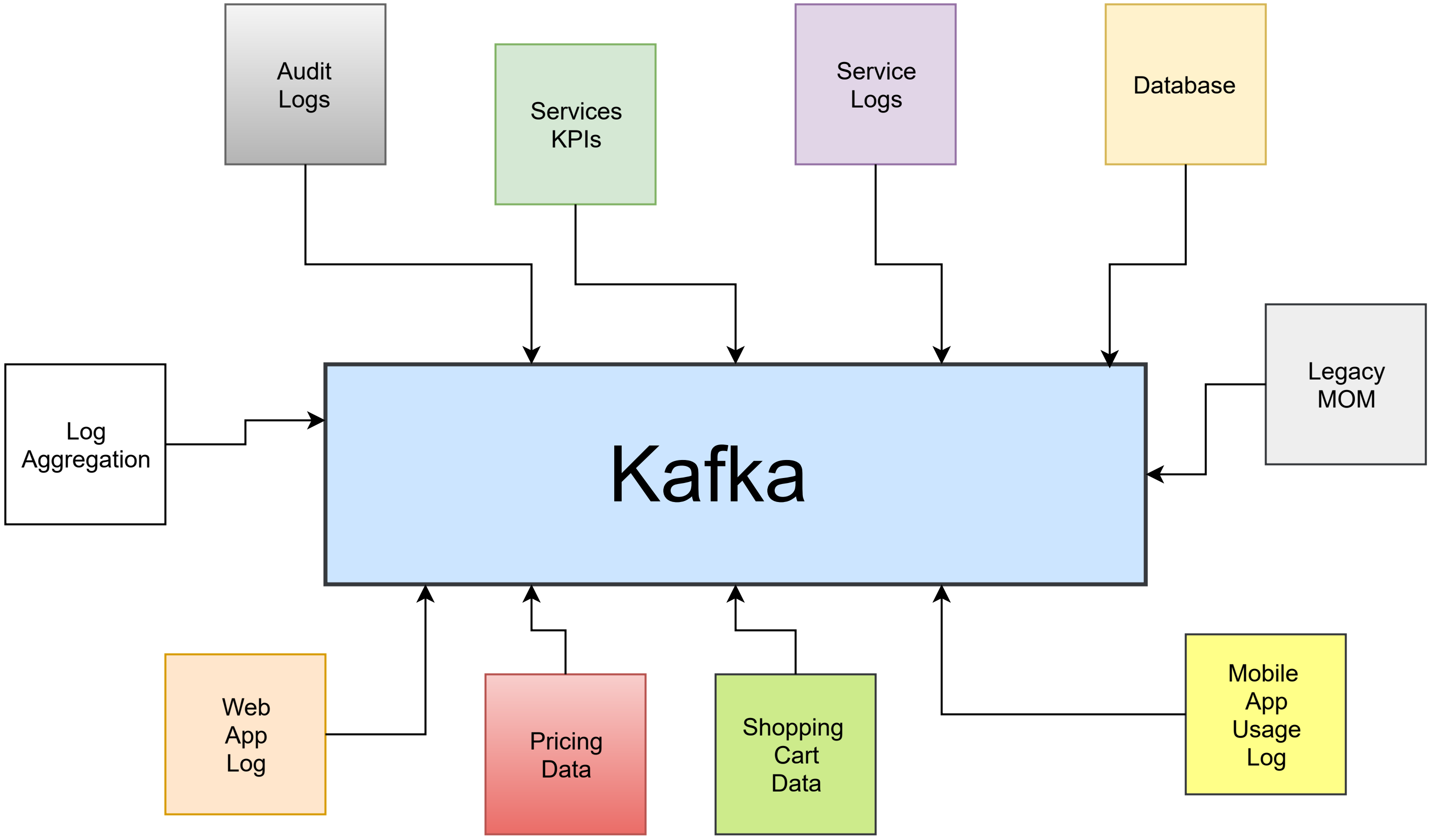


Now let's truly answer the question.

Kafka is…

Kafka is a distributed streaming platform that is used publish and subscribe to streams of records. Kafka gets used for fault tolerant storage. Kafka replicates topic log partitions to multiple servers. Kafka is designed to allow your apps to process records as they occur. Kafka is fast, uses IO efficiently by batching, compressing records. Kafka gets used for decoupling data streams. Kafka is used to stream data into data lakes, applications and real-time stream analytics systems.

Kafka Decoupling Data Streams



Kafka is Polyglot

Kafka communication from clients and servers uses a wire protocol over TCP that is versioned and documented. Kafka promises to maintain backwards compatibility with older clients, and many languages are supported. There are clients in C#, Java, C, Python, Ruby and many more languages. The Kafka ecosystem also provides REST proxy allows easy integration via HTTP and JSON, which makes integration even easier. Kafka also supports Avro schemas (http://cloudurable.com/blog/avro/index.html) via the Confluent Schema Registry for Kafka (http://cloudurable.com/blog/kafka-avro-schema-registry/index.html). Avro and the Schema Registry allows complex records to be produced and read by clients in many programming languages and allows for the evolution of the records. Kafka is truly polyglot.

Kafka is useful - Kafka Usage

Kafka allows you to build real-time streaming data pipe-lines. Kafka enable in-memory microservices (actors, Akka (http://cloudurable.com/akka-training/index.html), Baratine.io (http://baratine.io/), QBit (http://advantageous.github.io/qbit/), reactors, reactive (http://cloudurable.com/blog/reactive-microservices/index.html), Vert.x (http://vertx.io/), RxJava (http://reactivex.io/), Spring Reactor (https://spring.io/blog/2013/05/13/reactor-a-foundation-for-asynchronous-applications-on-the-jvm)) Kafka allows you to build real-time streaming applications that react to streams to do real-time data analytics, transform, react, aggregate, join real-time data flows and perform CEP (complex event processing) (https://flink.apache.org/news/2016/04/06/cep-monitoring.html).

You can use Kafka to aid in gathering Metrics/KPIs, aggregate statistics from many sources implement event sourcing (https://martinfowler.com/eaaDev/EventSourcing.html), use it with microservices (in-memory) and actor systems to implement in-memory services (external commit log for distributed systems).

You can use Kafka to replicate data between nodes, to re-sync for nodes, to restore state. While it is true, Kafka used for real-time data analytics and stream processing, you can also use it for log aggregation, messaging, click-stream tracking, audit trails, and much more.

In a world where data science and analytics is a big deal, then capturing data to feed into your data lakes and real-time analytics systems is a big deal, and since Kafka can hold up to these kinds of strenuous use cases, Kafka is a big deal.

Kafka is a scalable message storage

Kafka is a good storage system for records/messages. Kafka acts like high-speed file system for commit log storage and replication. These characteristics make Kafka useful for all manners of applications. Records written to Kafka topics are persisted to disk and replicated to other servers for fault-tolerance. Since modern drives are fast and quite large, this fits well and is very useful. Kafka Producers can wait on acknowledgment, so messages are durable as the producer write not complete until the message replicates. The Kafka disk structure scales well. Modern disk drives have very high throughput when writing in large streaming batches. Also, Kafka Clients/Consumers can control read position (offset) which allows for use cases like replaying the log if there was a critical bug (fix the bug and the replay). And since offsets are tracked per consumer group, which we talk about in the Kafka Architecture (/blog/kafka-architecture/index.html) article, the consumers can be quite flexible (e.g., replaying the log).

Kafka Record Retention

Kafka cluster retains all published records and if you don’t set a limit, it will keep records until it runs out of disk space. You can set time-based limits (configurable retention period), size-based limits (configurable based on size), or use compaction (keeps the latest version of record using key). You can, for example, set a retention policy of three days or two weeks or a month. The records in the topic log are available for consumption until discarded by time, size or compaction. The consumption speed not impacted by size as Kafka always writes to the end of the topic log.

> Cloudurable provides Kafka training (http://cloudurable.com/kafka-training/index.html), Kafka consulting (http://cloudurable.com/kafka-aws-consulting/index.html), Kafka support (http://cloudurable.com/subscription_support/index.html) and helps setting up Kafka clusters in AWS (http://cloudurable.com/services/index.html).

Let's Review

Why is Kafka so fast?

Kafka is fast because it avoids copying buffers in-memory (Zero Copy), and streams data to immutable logs instead of using random access.

How fast is Kafka usage growing?

When you consider Kafka is six years old, and over $^1/_3$ of fortune 500 companies use Kafka, then the only answer is fast, very fast.

How is Kafka getting used?

Kafka is used to feed data lakes like Hadoop, and to feed real-time analytics systems like Flink, Storm and Spark Streaming.

Where does Kafka fit in the Big Data Architecture?

Kafka is a data stream that fills up Big Data's data lakes.

How does Kafka relate to real-time analytics?

Kafka feeds data to real-time analytics systems like Storm, Spark Streaming, Flink, and Kafka Streaming.

Who uses Kafka?

The top ten travel companies, 7 of top ten banks, 8 of top ten insurance companies, 9 of top ten telecom companies, LinkedIn, Microsoft, Netflix and many more companies.

How does Kafka decouple streams of data?

It decouple streams of data by allowing multiple consumer groups that can each control where in the topic partition they are. The producers don't know about the consumers. Since the Kafka broker delegates the log partition offset (where the consumer is in the record stream) to the clients (Consumers), the message consumption is flexible. This allows you to feed your high-latency daily or hourly data analysis in Spark and Hadoop and the same time you are feeding microservices real-time messages, sending events to your CEP system and feeding data to your real-time analytic systems.

What are some common use cases for Kafka?

Kafka feeds data to real-time analytics systems like Storm, Spark Streaming, Flink, and Kafka Streaming. It also gets used for log aggregation, feeding events to CEP systems, and commit log for in-memory microservices (http://cloudurable.com/blog/high-speed-microservices/index.html).

What are some use cases for Kafka where you work?

Not sure yet. Then give us a call (http://cloudurable.com/contact/index.html).

Related content

- What is Kafka? (http://cloudurable.com/blog/what-is-kafka/index.html)
- Kafka Architecture (http://cloudurable.com/blog/kafka-architecture/index.html)
- Kafka Topic Architecture (http://cloudurable.com/blog/kafka-architecture-topics/index.html)
- Kafka Consumer Architecture (http://cloudurable.com/blog/kafka-architecture-consumers/index.html)
- Kafka Producer Architecture (http://cloudurable.com/blog/kafka-architecture-producers/index.html)
- Kafka Architecture and low level design (http://cloudurable.com/blog/kafka-architecture-low-level/index.html)
- Kafka and Schema Registry (http://cloudurable.com/blog/kafka-avro-schema-registry/index.html)
- Kafka and Avro (http://cloudurable.com/blog/avro/index.html)
- Kafka Ecosystem (http://cloudurable.com/blog/kafka-ecosystem/index.html)
- Kafka vs. JMS (http://cloudurable.com/blog/kafka-vs-jms/index.html)
- Kafka versus Kinesis (http://cloudurable.com/blog/kinesis-vs-kafka/index.html)
- Kafka Tutorial: Using Kafka from the command line (http://cloudurable.com/blog/kafka-tutorial-kafka-from-command-line/index.html)
- Kafka Tutorial: Kafka Broker Failover and Consumer Failover (http://cloudurable.com/blog/kafka-tutorial-kafka-failover-kafka-cluster/index.html)
- Kafka Tutorial (http://cloudurable.com/ppt/kafka-tutorial-cloudruable-v2.pdf)
- Kafka Tutorial: Writing a Kafka Producer example in Java (http://cloudurable.com/blog/kafka-tutorial-kafka-producer/index.html)
- Kafka Tutorial: Writing a Kafka Consumer example in Java (http://cloudurable.com/blog/kafka-tutorial-kafka-consumer/index.html)
- Kafka Architecture: Log Compaction (http://cloudurable.com/blog/kafka-architecture-log-compaction/index.html)

About Cloudurable

We hope you enjoyed this article. Please provide feedback (http://cloudurable.com/contact/index.html). Cloudurable provides Kafka training (http://cloudurable.com/kafka-training/index.html), Kafka consulting (http://cloudurable.com/kafka-aws-consulting/index.html), Kafka support (http://cloudurable.com/subscription_support/index.html) and helps setting up Kafka clusters in AWS (http://cloudurable.com/services/index.html).

Check out our new GoLang course. We provide onsite Go Lang training which is instructor led (http://cloudurable.com/golang-onsite-instructor-led-training/index.html).

in Share Tweet

Like 49 Share

SEARCH

Search

SHARE

Tweet

Share

facebook

49

Like

Share

FOLLOW

Follow @cloudurable

Follow

64

facebook

CATEGORIES

- amazon-ebs (1) (http://cloudurable.com/categories/amazon-ebs/index.html)
- amazon-ec2 (1) (http://cloudurable.com/categories/amazon-ec2/index.html)
- amazon-vpc (1) (http://cloudurable.com/categories/amazon-vpc/index.html)
- ansible (4) (http://cloudurable.com/categories/ansible/index.html)
- avro (2) (http://cloudurable.com/categories/avro/index.html)

aws (4) (http://cloudurable.com/categories/aws/index.html)

aws-cassandra (6) (http://cloudurable.com/categories/aws-cassandra/index.html)

aws-command-line (1) (http://cloudurable.com/categories/aws-command-line/index.html)

cassandra (12) (http://cloudurable.com/categories/cassandra/index.html)

cassandra-aws (3) (http://cloudurable.com/categories/cassandra-aws/index.html)

cassandra-cluster (1) (http://cloudurable.com/categories/cassandra-cluster/index.html)

cassandra-database (2) (http://cloudurable.com/categories/cassandra-database/index.html)

cassandra-training (5) (http://cloudurable.com/categories/cassandra-training/index.html)

cassandra-tutorial (5) (http://cloudurable.com/categories/cassandra-tutorial/index.html)

cloud (4) (http://cloudurable.com/categories/cloud/index.html)

cloudformation (1) (http://cloudurable.com/categories/cloudformation/index.html)

cloudurable (15) (http://cloudurable.com/categories/cloudurable/index.html)

cluster (1) (http://cloudurable.com/categories/cluster/index.html)

devops (16) (http://cloudurable.com/categories/devops/index.html)

ebs (3) (http://cloudurable.com/categories/ebs/index.html)

ec2 (1) (http://cloudurable.com/categories/ec2/index.html)

kafka (13) (http://cloudurable.com/categories/kafka/index.html) kafka-advanced-consumers (1) (http://cloudurable.com/categories/kafka-advanced-consumers/index.html)

kafka-architecture (13) (http://cloudurable.com/categories/kafka-architecture/index.html)

kafka-avro-serialization (1) (http://cloudurable.com/categories/kafka-avro-serialization/index.html)

kafka-consulting (2) (http://cloudurable.com/categories/kafka-consulting/index.html)

kafka-consumer (1) (http://cloudurable.com/categories/kafka-consumer/index.html)

kafka-consumers (1) (http://cloudurable.com/categories/kafka-consumers/index.html)

kafka-ecosystem (1) (http://cloudurable.com/categories/kafka-ecosystem/index.html)

kafka-schema-registry (1) (http://cloudurable.com/categories/kafka-schema-registry/index.html)

kafka-training (23) (http://cloudurable.com/categories/kafka-training/index.html)

kafka-tutorial (20) (http://cloudurable.com/categories/kafka-tutorial/index.html)

kaka-replication (1) (http://cloudurable.com/categories/kaka-replication/index.html)

kinesis (1) (http://cloudurable.com/categories/kinesis/index.html)

kinesis-consulting (1) (http://cloudurable.com/categories/kinesis-consulting/index.html)

linux (1) (http://cloudurable.com/categories/linux/index.html)

metricsd (1) (http://cloudurable.com/categories/metricsd/index.html)

microservices (3) (http://cloudurable.com/categories/microservices/index.html)

nodetool (1) (http://cloudurable.com/categories/nodetool/index.html) schema-registry (1) (http://cloudurable.com/categories/schema-registry/index.html)

smack (3) (http://cloudurable.com/categories/smack/index.html)

spark (3) (http://cloudurable.com/categories/spark/index.html)

spark--cassandra (1) (http://cloudurable.com/categories/spark--cassandra/index.html)

spark--kafka (1) (http://cloudurable.com/categories/spark–kafka/index.html)

spark-training (3) (http://cloudurable.com/categories/spark-training/index.html)

spark-tutorial (3) (http://cloudurable.com/categories/spark-tutorial/index.html)

ssh (1) (http://cloudurable.com/categories/ssh/index.html)

ssh-config (1) (http://cloudurable.com/categories/ssh-config/index.html)

ssl (1) (http://cloudurable.com/categories/ssl/index.html)

systemd (1) (http://cloudurable.com/categories/systemd/index.html)

tls (1) (http://cloudurable.com/categories/tls/index.html)

vagrant (5) (http://cloudurable.com/categories/vagrant/index.html)

TAGS

AKKA (HTTP://CLOUDURABLE.COM/TAGS/AKKA/INDEX.HTML)

AKKA-CONSULTING (HTTP://CLOUDURABLE.COM/TAGS/AKKA-CONSULTING/INDEX.HTML)

AMAZON-EBS (HTTP://CLOUDURABLE.COM/TAGS/AMAZON-EBS/INDEX.HTML)

AMAZON-EC2 (HTTP://CLOUDURABLE.COM/TAGS/AMAZON-EC2/INDEX.HTML)

AMI (HTTP://CLOUDURABLE.COM/TAGS/AMI/INDEX.HTML)

ANSIBLE (HTTP://CLOUDURABLE.COM/TAGS/ANSIBLE/INDEX.HTML)

AVRO (HTTP://CLOUDURABLE.COM/TAGS/AVRO/INDEX.HTML)

AVRO-KAFKA (HTTP://CLOUDURABLE.COM/TAGS/AVRO-KAFKA/INDEX.HTML)

AWS (HTTP://CLOUDURABLE.COM/TAGS/AWS/INDEX.HTML)

AWS-CASSANDRA (HTTP://CLOUDURABLE.COM/TAGS/AWS-CASSANDRA/INDEX.HTML)

AWS-COMMAND-LINE (HTTP://CLOUDURABLE.COM/TAGS/AWS-COMMAND-LINE/INDEX.HTML)

CASSANDRA (HTTP://CLOUDURABLE.COM/TAGS/CASSANDRA/INDEX.HTML)

CASSANDRA-ARCHITECTURE (HTTP://CLOUDURABLE.COM/TAGS/CASSANDRA-ARCHITECTURE/INDEX.HTML)

CASSANDRA-AWS (HTTP://CLOUDURABLE.COM/TAGS/CASSANDRA-AWS/INDEX.HTML)

CASSANDRA-CLOUD (HTTP://CLOUDURABLE.COM/TAGS/CASSANDRA-CLOUD/INDEX.HTML)

CASSANDRA-CLUSTER (HTTP://CLOUDURABLE.COM/TAGS/CASSANDRA-CLUSTER/INDEX.HTML)

CASSANDRA-DATABASE (HTTP://CLOUDURABLE.COM/TAGS/CASSANDRA-DATABASE/INDEX.HTML)

CASSANDRA-DBA (HTTP://CLOUDURABLE.COM/TAGS/CASSANDRA-DBA/INDEX.HTML)

CASSANDRA-DEVOPS (HTTP://CLOUDURABLE.COM/TAGS/CASSANDRA-DEVOPS/INDEX.HTML)

CASSANDRA-OS-SYSTEM-MEMORY (HTTP://CLOUDURABLE.COM/TAGS/CASSANDRA-OS-SYSTEM-MEMORY/INDEX.HTML)

CASSANDRA-TRAINING (HTTP://CLOUDURABLE.COM/TAGS/CASSANDRA-TRAINING/INDEX.HTML)

CASSANDRA-TUTORIAL (HTTP://CLOUDURABLE.COM/TAGS/CASSANDRA-TUTORIAL/INDEX.HTML)

CLOUD (HTTP://CLOUDURABLE.COM/TAGS/CLOUD/INDEX.HTML)

CLOUDFORMATION (HTTP://CLOUDURABLE.COM/TAGS/CLOUDFORMATION/INDEX.HTML)

CLOUDFORMATION-TUTORIAL (HTTP://CLOUDURABLE.COM/TAGS/CLOUDFORMATION-TUTORIAL/INDEX.HTML)

CLOUDURABLE (HTTP://CLOUDURABLE.COM/TAGS/CLOUDURABLE/INDEX.HTML)

CLUSTER (HTTP://CLOUDURABLE.COM/TAGS/CLUSTER/INDEX.HTML)

COMPUTE (HTTP://CLOUDURABLE.COM/TAGS/COMPUTE/INDEX.HTML)

CONSUMERS (HTTP://CLOUDURABLE.COM/TAGS/CONSUMERS/INDEX.HTML)

DBA (HTTP://CLOUDURABLE.COM/TAGS/DBA/INDEX.HTML)

DEVOPS (HTTP://CLOUDURABLE.COM/TAGS/DEVOPS/INDEX.HTML)

EBS (HTTP://CLOUDURABLE.COM/TAGS/EBS/INDEX.HTML)

EC2 (HTTP://CLOUDURABLE.COM/TAGS/EC2/INDEX.HTML)

EC2-INSTANCE-STORE (HTTP://CLOUDURABLE.COM/TAGS/EC2-INSTANCE-STORE/INDEX.HTML)

ECU (HTTP://CLOUDURABLE.COM/TAGS/ECU/INDEX.HTML)

FAILOVER (HTTP://CLOUDURABLE.COM/TAGS/FAILOVER/INDEX.HTML)

IMAGES (HTTP://CLOUDURABLE.COM/TAGS/IMAGES/INDEX.HTML)

INSTANCES (HTTP://CLOUDURABLE.COM/TAGS/INSTANCES/INDEX.HTML)

JMS (HTTP://CLOUDURABLE.COM/TAGS/JMS/INDEX.HTML)

KAFKA (HTTP://CLOUDURABLE.COM/TAGS/KAFKA/INDEX.HTML)

KAFKA-ADVANCED-CONSUMERS (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-ADVANCED-CONSUMERS/INDEX.HTML)

KAFKA-ADVANCED-PRODUCERS (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-ADVANCED-PRODUCERS/INDEX.HTML)

KAFKA-ARCHITECTURE (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-ARCHITECTURE/INDEX.HTML)

KAFKA-AVRO (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-AVRO/INDEX.HTML)

KAFKA-CONNECT (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-CONNECT/INDEX.HTML)

KAFKA-CONSULTING (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-CONSULTING/INDEX.HTML)

KAFKA-CONSUMERS (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-CONSUMERS/INDEX.HTML)

KAFKA-CONSUMERS-ADVANCED (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-CONSUMERS-ADVANCED/INDEX.HTML)

KAFKA-ECOSYSTEM (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-ECOSYSTEM/INDEX.HTML)

KAFKA-LOG-COMPACTION (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-LOG-COMPACTION/INDEX.HTML)

KAFKA-REST-PROXY (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-REST-PROXY/INDEX.HTML)

KAFKA-STREAMS (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-STREAMS/INDEX.HTML)

KAFKA-TRAINING (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-TRAINING/INDEX.HTML)

KAFKA-TUTORIAL (HTTP://CLOUDURABLE.COM/TAGS/KAFKA-TUTORIAL/INDEX.HTML)

KINESIS (HTTP://CLOUDURABLE.COM/TAGS/KINESIS/INDEX.HTML)

KINESIS-CONSULTING (HTTP://CLOUDURABLE.COM/TAGS/KINESIS-CONSULTING/INDEX.HTML)

LINUX (HTTP://CLOUDURABLE.COM/TAGS/LINUX/INDEX.HTML)

METRICSD (HTTP://CLOUDURABLE.COM/TAGS/METRICSD/INDEX.HTML)

MICROSERVICES (HTTP://CLOUDURABLE.COM/TAGS/MICROSERVICES/INDEX.HTML)

MICROSERVICES-ARCHITECTURE (HTTP://CLOUDURABLE.COM/TAGS/MICROSERVICES-ARCHITECTURE/INDEX.HTML)

NAT (HTTP://CLOUDURABLE.COM/TAGS/NAT/INDEX.HTML)

NODETOOL (HTTP://CLOUDURABLE.COM/TAGS/NODETOOL/INDEX.HTML)

NUMA (HTTP://CLOUDURABLE.COM/TAGS/NUMA/INDEX.HTML)

PRODUCERS (HTTP://CLOUDURABLE.COM/TAGS/PRODUCERS/INDEX.HTML)

QBIT (HTTP://CLOUDURABLE.COM/TAGS/QBIT/INDEX.HTML) RAM (HTTP://CLOUDURABLE.COM/TAGS/RAM/INDEX.HTML)

REAKT (HTTP://CLOUDURABLE.COM/TAGS/REAKT/INDEX.HTML)

REPLICATION (HTTP://CLOUDURABLE.COM/TAGS/REPLICATION/INDEX.HTML)

SCHEMA-REGISTRY (HTTP://CLOUDURABLE.COM/TAGS/SCHEMA-REGISTRY/INDEX.HTML)

SMACK (HTTP://CLOUDURABLE.COM/TAGS/SMACK/INDEX.HTML) SPARK (HTTP://CLOUDURABLE.COM/TAGS/SPARK/INDEX.HTML)

SPARK--CASSANDRA (HTTP://CLOUDURABLE.COM/TAGS/SPARK--CASSANDRA/INDEX.HTML)

SPARK-TRAINING (HTTP://CLOUDURABLE.COM/TAGS/SPARK-TRAINING/INDEX.HTML)

SPARK-TUTORIAL (HTTP://CLOUDURABLE.COM/TAGS/SPARK-TUTORIAL/INDEX.HTML)

SSH (HTTP://CLOUDURABLE.COM/TAGS/SSH/INDEX.HTML) SSH-CONFIG (HTTP://CLOUDURABLE.COM/TAGS/SSH-CONFIG/INDEX.HTML)

SSL (HTTP://CLOUDURABLE.COM/TAGS/SSL/INDEX.HTML) SYSTEMD (HTTP://CLOUDURABLE.COM/TAGS/SYSTEMD/INDEX.HTML)

TLS (HTTP://CLOUDURABLE.COM/TAGS/TLS/INDEX.HTML) VAGRANT (HTTP://CLOUDURABLE.COM/TAGS/VAGRANT/INDEX.HTML)

VCPU (HTTP://CLOUDURABLE.COM/TAGS/VCPU/INDEX.HTML) VPC (HTTP://CLOUDURABLE.COM/TAGS/VPC/INDEX.HTML)

WHAT-IS-KAFKA (HTTP://CLOUDURABLE.COM/TAGS/WHAT-IS-KAFKA/INDEX.HTML)

Apache Spark Training (http://cloudurable.com/spark-training/index.html)
Kafka Tutorial (http://cloudurable.com/blog/kafka-tutorial/index.html)
Akka Consulting (http://cloudurable.com/akka-consulting/index.html)
Cassandra Training (http://cloudurable.com/cassandra-course/index.html)
AWS Cassandra Database Support (http://cloudurable.com/subscription_support_benefits_cassandra/index.html)
Kafka Support Pricing (http://cloudurable.com/subscription_support/index.html?q=kafka)
Cassandra Database Support Pricing (http://cloudurable.com/subscription_support/index.html?q=cassandra)
Non-stop Cassandra (http://cloudurable.com/cloudurable-cassandra-watchdog/index.html?q=cassandra)
Watchdog (http://cloudurable.com/cloudurable-cassandra-watchdog/index.html?q=watchdog)

Advantages of using Cloudurable™ (http://cloudurable.com/advantages/index.html)
Cassandra Consulting (http://cloudurable.com/service-quick-start-mentoring-cassandra-or-kafka-aws-ec2/index.html)
Cloudurable™| Guide to AWS Cassandra Deploy (http://cloudurable.com/ppt/amazon-cassandra.pdf)
Cloudurable™| AWS Cassandra Guidelines and Notes (http://cloudurable.com/ppt/amazon-cassandra-notes.pdf)
Free guide to deploying Cassandra on AWS (http://cloudurable.com/cassandra-aws-consulting/index.html)
Kafka Training (http://cloudurable.com/kafka-training/index.html)
Kafka Consulting (http://cloudurable.com/kafka-aws-consulting/index.html)
DynamoDB Training (http://cloudurable.com/dynamodb-training/index.html)
DynamoDB Consulting (http://cloudurable.com/dynamodb-consulting/index.html)
Kinesis Training (http://cloudurable.com/kinesis-training/index.html)
Kinesis Consulting (http://cloudurable.com/kinesis-consulting/index.html)
Kafka Tutorial PDF (http://cloudurable.com/blog/kafka-tutorial-v1/index.html)
Redis Consulting (http://cloudurable.com/redis-consulting/index.html)
Redis Training (http://cloudurable.com/redis-onsite-instructor-led-training/index.html)
ElasticSearch / ELK Consulting (http://cloudurable.com/elk-consulting/index.html)
ElasticSearch Training (http://cloudurable.com/elasticsearch-onsite-instructor-led-training/index.html)
InfluxDB/TICK Training (http://cloudurable.com/influxdb-onsite-instructor-led-training/index.html) TICK Consulting (http://cloudurable.com/tick-consulting/index.html)

ABOUT US

Cloudurable™: Leader in AWS cloud computing for Kafka™, Cassandra™ Database, Apache Spark, AWS CloudFormation™ DevOps. We do **Cassandra training**, **Apache Spark**, **Kafka training**, **Kafka consulting** and **cassandra consulting** with a focus on AWS and data engineering. (FAQ (http://cloudurable.com/faq/index.html))

FOLLOW CLOUDURABLE™

facebook page (https://www.facebook.com/cloudurable)
google plus (https://plus.google.com/116648719730180908239)
twitter (https://twitter.com/cloudurable)
linkedin (https://www.linkedin.com/company/17964258/)

Why Cloudurable™?
Advantage of using Cloudurable™ (http://cloudurable.com/advantages/index.html)
About Cloudurable™?

About Cloudurable™ (http://cloudurable.com/faq/index.html)

What are the benefits of using subscription support?

Benefits of Subscription Cassandra Support (http://cloudurable.com/subscription_support_benefits_cassandra/index.html)

RECENT POSTS

KAFKA CONSUMER: ADVANCED CONSUMERS (HTTP://CLOUDURABLE.COM/BLOG/KAFKA-ADVANCED-CONSUMER-1/INDEX.HTML)

KAFKA TUTORIAL (HTTP://CLOUDURABLE.COM/BLOG/KAFKA-TUTORIAL/INDEX.HTML)

m/blog/kafka-

KAFKA TUTORIAL: CREATING ADVANCED KAFKA PRODUCERS IN JAVA (HTTP://CLOUDURABLE.COM/BLOG/KAFKA-TUTORIAL-KAFKA-PRODUCER-ADVANCED-JAVA-EXAMPLES/INDEX.HTML)

m/blog/kafka-

KAFKA CONSULTING (HTTP://CLOUDURABLE.COM/BLOG/KAFKA-CONSULTING/INDEX.HTML)

m/blog/kafka-

CONTACT

java-
http://cloudurable.com/blog/kafka-
examples/index.html)

Cloudurable Tech
101 California Street
San Francisco
CA 94111
USA
America
(415) 758-1113 (tel:14157581113)

GO TO CONTACT PAGE (/CONTACT/INDEX.HTML)

Template by DevCows (https://github.com/devcows/hugo-universal-theme)