

Explicações simplificadas sobre Apache Kafka e componentes

Comparação simplificada: Apache Kafka vs Apache Spark

- Apache Kafka:

É um sistema de "correio rápido" de mensagens. Ele transporta e armazena temporariamente eventos/dados (ex: pedidos feitos, cliques no site, logs).

Não processa os dados - apenas garante que eles cheguem rapidamente e com segurança.

Exemplo: um clique no site gera uma mensagem que o Kafka armazena e entrega para sistemas interessados.

- Apache Spark:

É um "cérebro de processamento de dados". Ele lê dados (ex: do Kafka ou de bancos) e faz processamento pesado: soma, média, detecção de fraude, geração de relatórios.

Exemplo: Spark lê do Kafka todos os pedidos do site e gera um gráfico de vendas em tempo real.

Resumo:

- Kafka = transporte e armazenamento temporário de eventos.
- Spark = processamento e análise de dados.
- São complementares e muitas arquiteturas modernas usam ambos juntos.

Explicação simplificada: Schema Registry

O Schema Registry funciona como um "cartório de formatos de dados" no Kafka.

- Ele guarda os formatos (schemas) que definem como as mensagens devem ser estruturadas (ex:

um pedido tem nome, idade, valor, etc).

- Antes de enviar ou ler mensagens no Kafka, os sistemas consultam o Schema Registry para garantir que estão usando o formato correto.

Problema que resolve:

Sem o Schema Registry, mudanças no formato das mensagens poderiam quebrar os sistemas que consomem esses dados.

Com ele, há um controle de versões e validação automática, garantindo compatibilidade e evitando erros.

Resumo:

Schema Registry garante que produtores e consumidores do Kafka falem a mesma "língua", mesmo com evolução dos dados.

Explicação simplificada: KRaft (Kafka Raft)

KRaft é uma nova arquitetura do Kafka que elimina a dependência do Apache ZooKeeper.

Antes:

- Kafka precisava do ZooKeeper para coordenar seu cluster (saber quem são os brokers, quais tópicos existem, etc). Isso tornava a arquitetura mais complexa e sujeita a falhas.

Com KRaft:

- Kafka passa a se gerenciar sozinho. Usa o protocolo Raft para garantir que todos os brokers tenham a mesma visão atualizada do cluster.
- Não precisa mais do ZooKeeper.

- Fica mais simples, mais resiliente e mais performático.

Resumo:

KRaft = Kafka autogerenciado, com menos complexidade e maior robustez.