



Streams com Kafka

Facef 2020



Agenda

- Introdução
- Casos de usos
- Preparação Ambiente(instalação)
 - Instalação
 - Manda e recebe via CLI
- Mensageria com Kafka
- Streams API
- Connectors



Introdução

- O que é event streaming
 - Prática de capturar dados em real-time de fonte de eventos como banco de dados, sensores e etc, para depois serem recuperados, manipulados e processados. Garantindo um fluxo contínuo de interpretação das informações.
- Cenários de utilização
 - Processamento de transações de pagamentos em real-time
 - Captura e análise de dados de sensores de IOT
 - Conectar, armazenar e tornar disponíveis dados produzidos por diferentes sistemas em um único repositório.
 - Fundação para **plataforma de dados, arquitetura event-driven e microserviços.**



Introdução

- História do Kafka
 - Desenvolvido em Scala + Java em 2011 pelo LinkedIn.
 - Doado para comunidade open-source
- Kafka como plataforma de eventos
 - **Pub/Sub** para escrever e ler streams de eventos a partir de outros sistemas.
 - **Armazenamento** dos streams de eventos de maneira **durável e confiável**.
 - **Processamento** de streams de eventos em tempo real ou **retrospectivamente**.
- Como o Kafka funciona
- Conceitos e termos
- Kafka Apis



Introdução

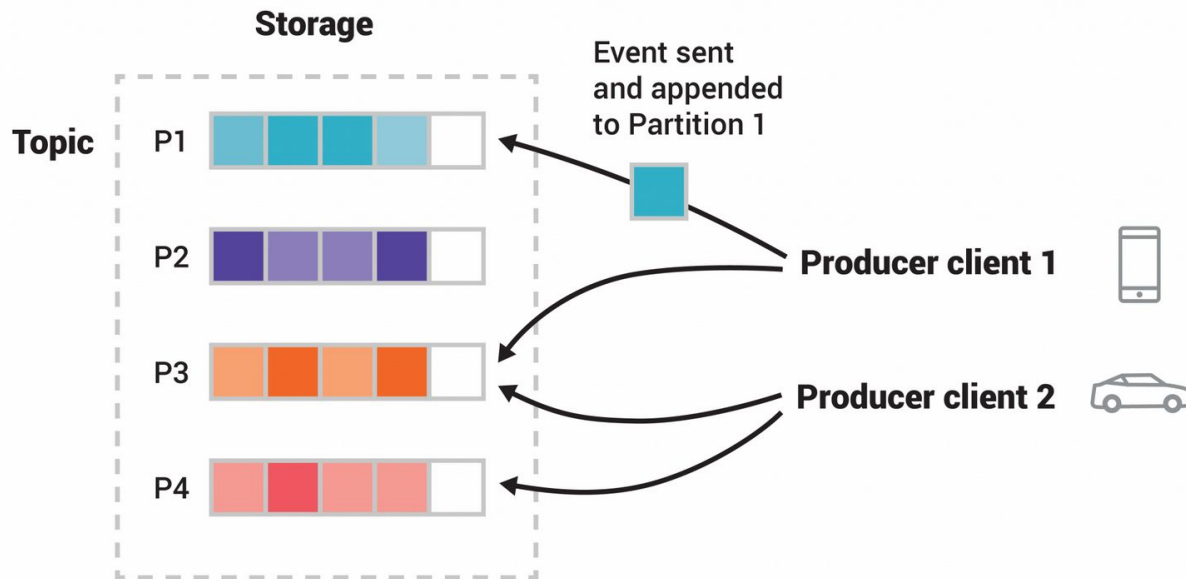
- História do Kafka
 - Desenvolvido em Scala + Java em 2011 pelo LinkedIn.
 - Doado para comunidade open-source
- Kafka como plataforma de eventos
 - **Pub/Sub** para escrever e ler streams de eventos a partir de outros sistemas.
 - **Armazenamento** dos streams de eventos de maneira **durável e confiável**.
 - **Processamento** de streams de eventos em tempo real ou **retrospectivamente**.
- Como o Kafka funciona
 - Sistema distribuído de servidores e clientes que se comunicam através de protocolo TCP.
 - Servidores podem ser executados como clusters
 - Clientes permitem que sejam desenvolvidos aplicações de maneira distribuída.



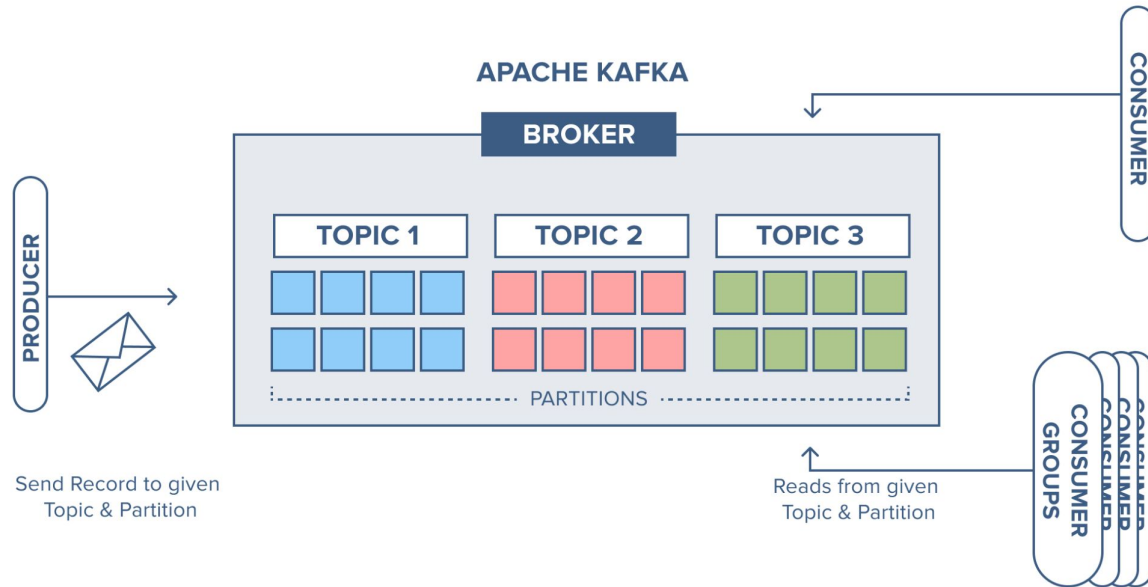
Introdução

- Conceitos e termos
 - Operações de leitura ou escrita no kafka, esse processo é realizado na forma de eventos, conceitualmente um evento tem uma chave, valor, timestamp e headers de metadados opcionais. Exemplo de um evento:
 - **Chave do evento:** "ID do pedido 8391238198"
 - **Valor do evento:** "Pedido no valor de R\$1000,00"
 - **Timestamp do evento:** "2020-09-07T12:00:00"
 - **Producers:** aplicações que publicam os eventos no Kafka.
 - **Consumers:** aplicações que consomem eventos do Kafka.
 - **Tópicos:** organizam e armazenam os eventos, permitindo múltiplos producers e consumers.
 - **Tópicos são particionados:** permitindo assim consumidores lerem as partições

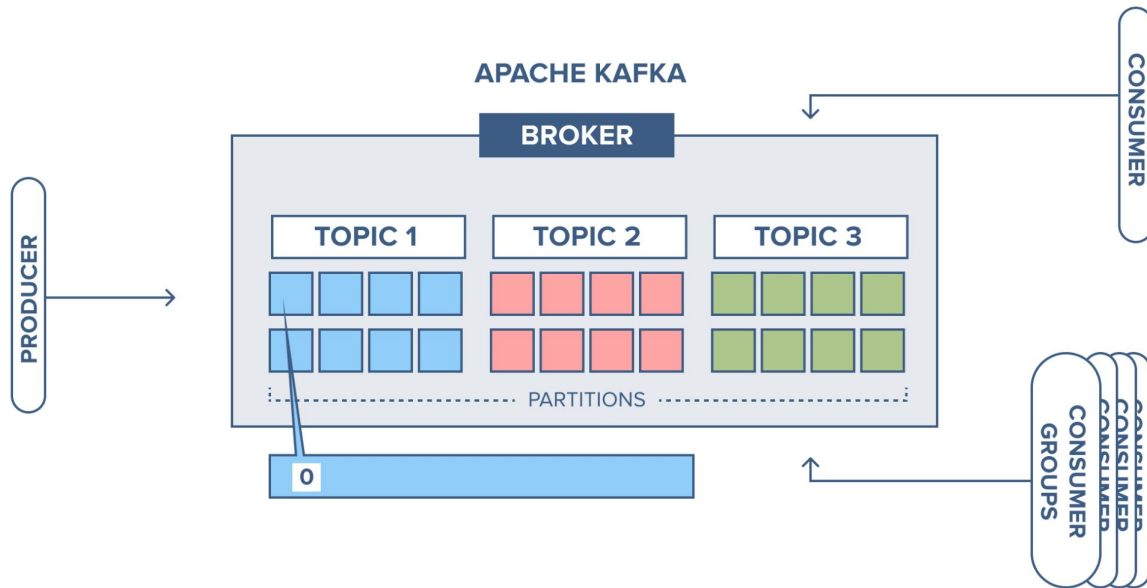
Introdução



Fluxo de mensagens

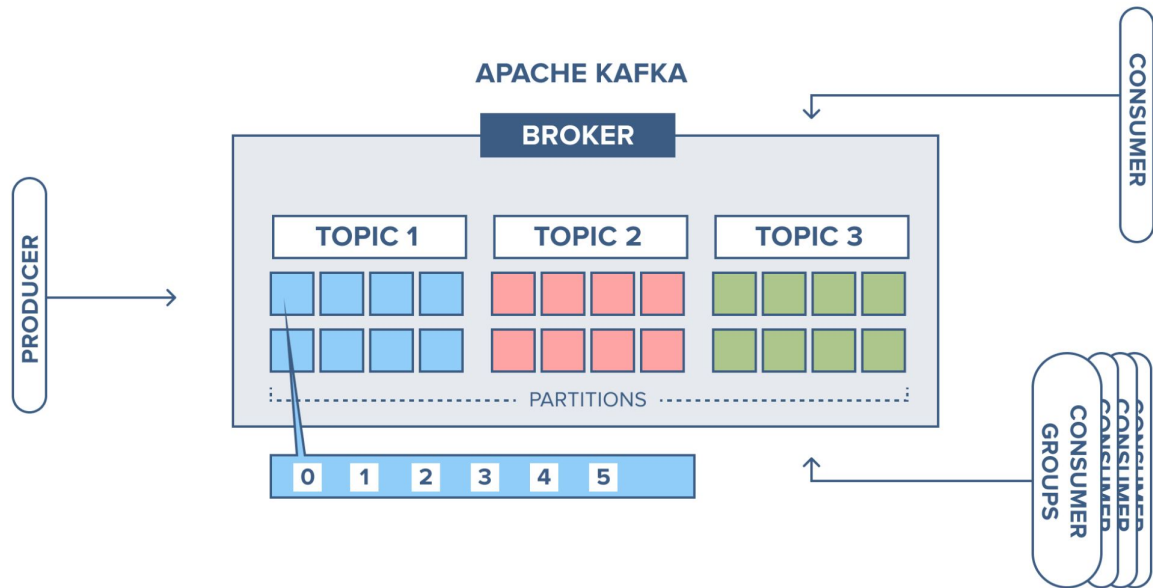


Fluxo de mensagens



Fluxo de mensagens

Commit log termo utilizado no Kafka onde cada mensagem enviada ao tópico é adicionada ao commit log e não existe uma maneira de alterar os registros já existentes no log. Este também será o mesmo Offset no qual os consumers irão utilizar para iniciar o processo de leitura das mensagens presentes em cada partição.





Casos de uso

- Mensageria
- Rastreamento de atividades em Sites
- Métricas
- Agregação de Logs
- Processamento em Stream (Fluxo)
- Event Sourcing

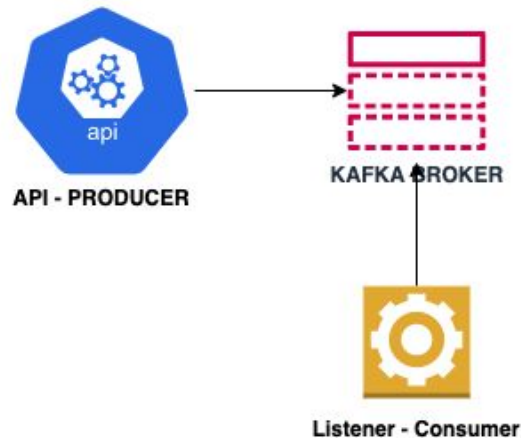


Preparação do ambiente

- Primeiramente devemos realizar a instalação do [docker-compose](#)

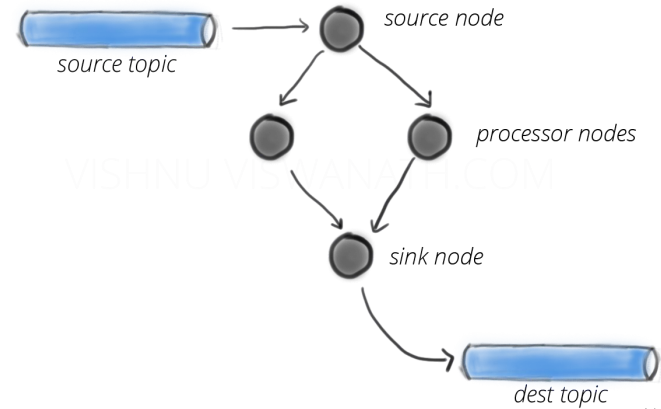
Mensageria com Kafka - Exercício 1

- Vamos construir uma aplicação que irá receber um ID de pedido via API e irá serializar esse pedido no Kafka
- Vamos criar um consumidor para consumir o ID do pedido do Kafka
- Vamos alterar nossa aplicação para possibilitar o recebimento de um Objeto via JSON na requisição.
- Vamos alterar o produtor e o consumidor para poder trabalhar com um objeto complexo do tipo Pedido



Stream Processing

- Ferramenta Big Data
- Consulta constantes
- Detecção de condições em períodos de tempos
- Análise em tempo Real





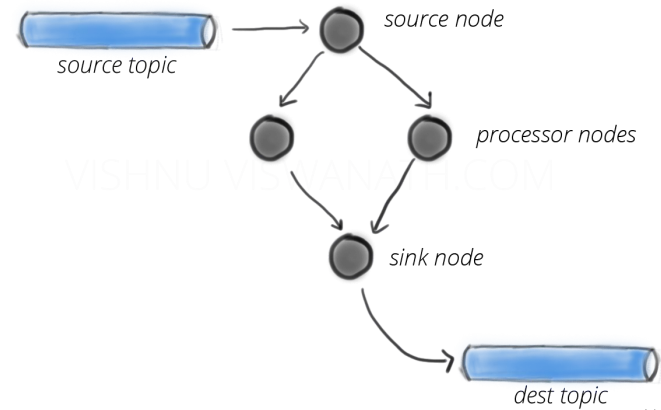
Stream Processing

Razões para usar:

- Eventos naturalmente sem fim;
- Redução de processamento em segundo plano;
- Volume de dados tão grande que não é possível armazenar;
- Eventos já existentes(Transações bancárias, fechamento de pedidos, etc)

E como fazemos Stream Processing?

1. Aplicação que envia um evento para um Broker(Apache Kafka, RabbitMQ)
2. Aplicação que recebe que processa e devolve para broker



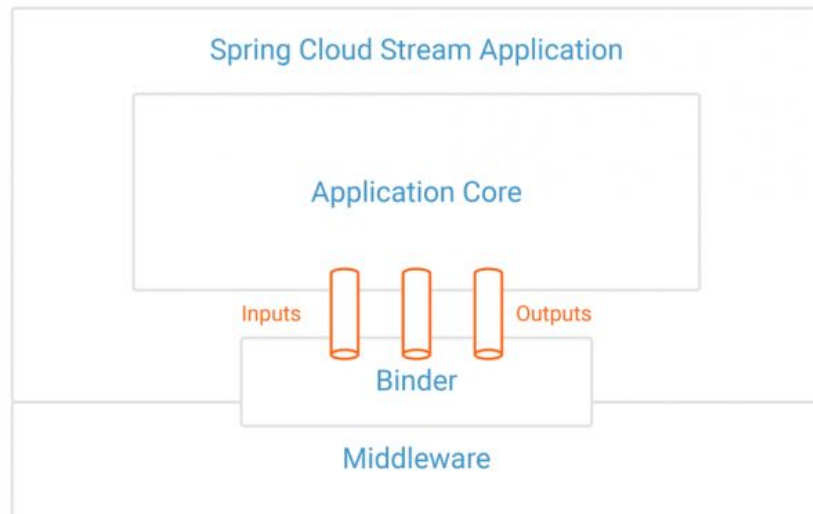


Kafka Stream API

- Processamento Event-at-a-time com latência na caso dos milisegundos;
- Processamento Stateful incluindo agregações e uniões distribuídas;
- Ótimo ferramental;
- Processamento distribuído, tolerância a falhas com "fast failover";
- Capacidade de retro processamento, caso o código seja alterado você pode recalcular;
- No-Downtime deployments;
- Elástico;
- Altamente escalável;

Spring Cloud Stream

- Camada de abstração;
- Suporta vários sistemas de mensageria;
 - RabbitMQ, Kafka, etc;
- Compor seu microservice com o pipeline de Stream;
- Integrado com a suíte do Spring, configurações, monitoria, rastreamento, logs, etc.





Exercício 3

- Vamos construir uma aplicação para analisar um fluxo de texto e fazer a contagem da quantidade de palavras únicas
- Passo a passo

<https://github.com/diegofernandes/kafka-facef/blob/master/contapalavras/Readme.MD>

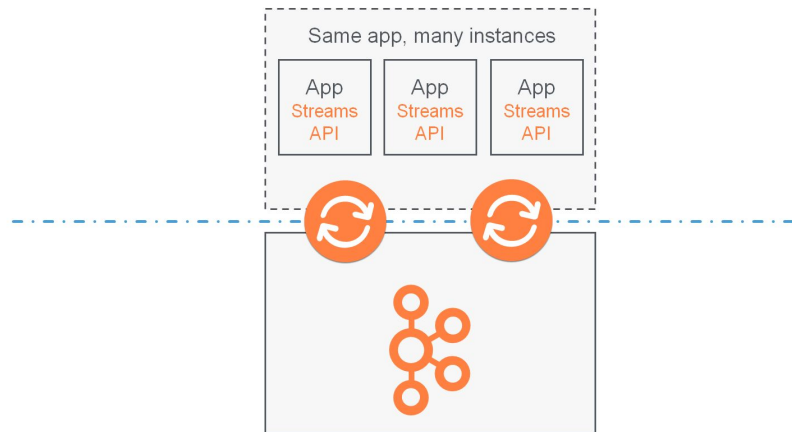


Streams Conceitos - Stream

- Stream
 - Representa um fluxo contínuo de dados imensuráveis;
- Stream partition
 - Sequência de dados imutáveis;
 - Ordenável;
 - Replayable - reprocessável;
 - Tolerante a falha;
 - Chave Valor;

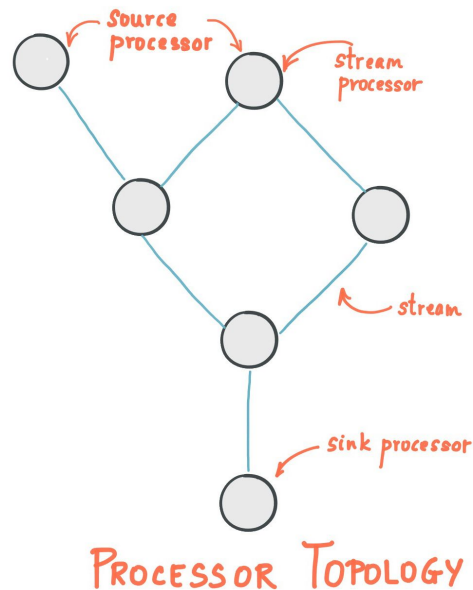
Streams Conceitos - Stream Processing Application

- Stream Processing Application - Qualquer aplicação que faça uso das API do Kafka Streams(seu aplicacao).
- Application instance - Instâncias da sua aplicação, Vertical ou Horizontal;



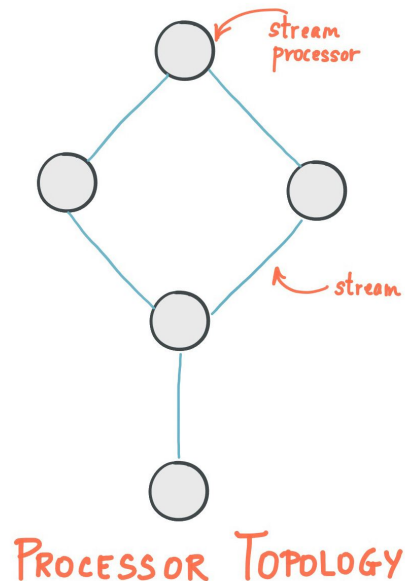
Streams Conceitos - Processor Topology

- Topology - define o processamento lógico da sua aplicação;
 - Como o que entra vira o que sai;
- Source Processor - Consome de um ou mais tópicos e encaminha para os demais processors;
 - Ponto de entrada;
- Sink Processor - Envia para os dados para um ou mais tópicos;
 - Ponto de saída;

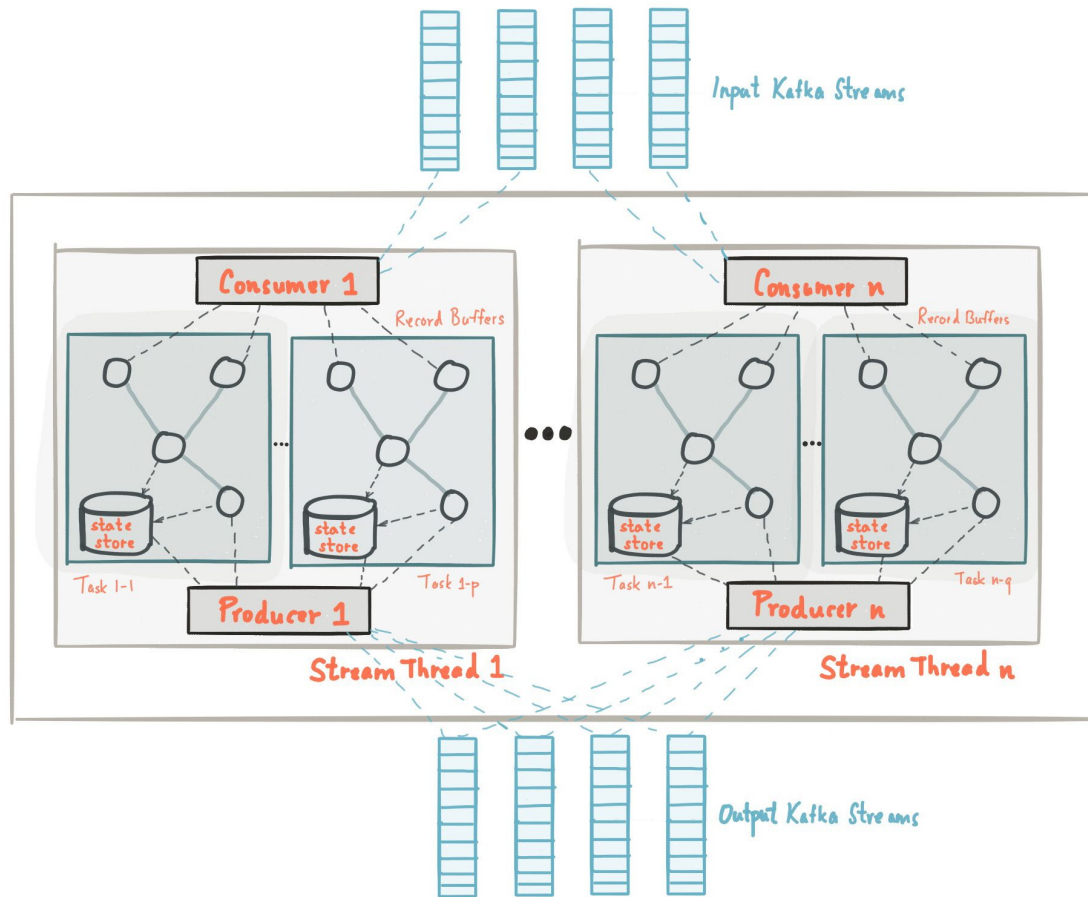


Streams Conceitos - Stream Processor

- Representação de um passo na topologia;
- Transformação dos dados;
 - map, filter, joins e aggregations;
- Emite um ou mais eventos para os próximos processors;
- APIs - temos duas opções de APIs;
 - Declarativa e funcional - Cobre a maioria dos casos(ex map, filter)
 - Imperativa, baixo nível - Maior flexibilidade, porém mais código;
- Stateless - Processa um evento por vez;
- Stateful - Processamento é impactados por outros eventos;
 - joins, aggregate ou window



Kafka Stream API - Arquitetura





Exercício 4

- Vamos construir uma aplicação que além de analisar um fluxo de texto e fazer a contagem da quantidade de palavras únicas, categoriza as palavras e direciona o resultado para tópicos específicos;
- <https://github.com/diegofernandes/kafka-facef/blob/master/categorizadorpalavras/Readme.MD>

Kafka Connectors

- Kafka Connect - componente OpenSource do Kafka.
- Framework de conexão de sistema externos ao Kafka
 - Banco de dados;
 - Chaves-Valor;
 - File System;
- Entrada ou Saída;
- [Lista de Connectors](#)

