



# Semantix

**Confluent**

Aula 5

# Quem sou eu?

Eu sou Rodrigo Augusto Rebouças.

Engenheiro de dados da Semantix  
Instrutor do Semantix Mentoring Academy

Você pode me encontrar em:  
[rodrigo.augusto@semantix.com.br](mailto:rodrigo.augusto@semantix.com.br)





# Kafka Connect



# Kafka Connect

- Componente open-source do Kafka
- Estrutura para conectar o Kafka a sistemas externos
  - Bancos de dados
  - Índices de pesquisa
  - Sistemas de arquivo
- Principais tipos de conectores
  - **Source Connector:** Enviar dados do sistema externo para os tópicos do Kafka
  - **Sink Connector:** Enviar os dados do tópico Kafka para o sistema externo
- Execução
  - Processo autônomo para executar tarefas em uma única máquina
  - Serviço distribuído, escalável e tolerante a falhas



# Confluent Hub

- Repositório de conectores da Confluent
  - <https://www.confluent.io/hub/>

Results (176)

## Kafka Connect S3

SINK CONNECTOR

The S3 connector, currently available as a sink, allows you to export data from Kafka topics to S3 objects in either Avro or JSON formats

 Available fully-managed on Confluent Cloud

**Enterprise support:**

Confluent supported

**Verification:**

Confluent built

**License:**

Free

**Installation:**

Confluent Hub CLI, Download

**Author:**

Confluent, Inc.

**Version:**

5.5.2



- Instalar componentes
  - docker exec -it connect bash
  - confluent-hub install <componente>



Version 5.5.2

Download

Available fully-  
managed on  
[Confluent Cloud](#)

Plugin type: ⓘ  
Sink

Enterprise support: ⓘ  
Confluent supported

## Kafka Connect S3

The S3 connector, currently available as a sink, allows you to export data from Kafka topics to S3 objects in either Avro or JSON formats. In addition, for certain data layouts, S3 connector exports data by guaranteeing exactly-once delivery semantics to consumers of the S3 objects it produces.

Show more ▾

## Installation

### Confluent Hub CLI installation

Use the [Confluent Hub client](#) to install this connector with:

```
confluent-hub install confluentinc/kafka-connect-s3:5.5.2
```

[Copy](#)

### Download installation

Or download the ZIP file and extract it into one of the directories that is listed on the Connect worker's plugin.path configuration properties. This must be done on each of the installations where Connect will be run.

Download

# Conectores Control Center

## ○ Conectores pela Plataforma Confluent

The screenshot displays the Confluent Control Center interface. At the top, a dark blue header bar contains the 'CONFLUENT' logo on the left and a notification bell icon on the right. Below the header, a left-hand navigation sidebar is visible, featuring a grid icon at the top, followed by 'Cluster 1' (highlighted in teal), and a list of navigation items: Overview, Brokers, Topics, Connect (highlighted in light blue), ksqldb, Consumers, Replicators, and Cluster settings. The main content area is a grid of connector cards. Each card displays the connector name, its role (Source or Sink), and a 'Connect' button. The connectors shown are: ActiveMQSource Connector (Source), ElasticsearchSink Connector (Sink), GcsSinkConnector (Sink), IbmMQSourceConnector (Source), JdbcSinkConnector (Sink), JdbcSourceConnector (Source), JmsSourceConnector (Source), S3SinkConnector (Sink), SchemaSourceConnector (Source), DatagenConnector (Source), FileStreamSinkConnector (Sink), FileStreamSource Connector (Source), MirrorCheckpoint Connector (Source), MirrorHeartbeatConnect... (Source), and MirrorSourceConnector (Source). The bottom of the interface features a series of small, light blue dots arranged in a horizontal line.

Connector Name	Role	Action
ActiveMQSource Connector	Source	Connect
ElasticsearchSink Connector	Sink	Connect
GcsSinkConnector	Sink	Connect
IbmMQSourceConnector	Source	Connect
JdbcSinkConnector	Sink	Connect
JdbcSourceConnector	Source	Connect
JmsSourceConnector	Source	Connect
S3SinkConnector	Sink	Connect
SchemaSourceConnector	Source	Connect
DatagenConnector	Source	Connect
FileStreamSinkConnector	Sink	Connect
FileStreamSource Connector	Source	Connect
MirrorCheckpoint Connector	Source	Connect
MirrorHeartbeatConnect...	Source	Connect
MirrorSourceConnector	Source	Connect



## Adicionar Conector





# Adicionar Conector

CONTROLCENTER.CLUSTER

Overview

Brokers

Topics

Connect

ksq|DB

Consumers

Replicators

Cluster settings

CONNECT CLUSTERS > CONNECT-DEFAULT >

## Connectors

### Connectors

0

Total

0

Running

0

Degraded

0

Failed



You haven't added any connectors yet

Bring data into or out of your cluster

Add connector

Learn more

# Adicionar Conector

CONTROLCENTER.CLUSTER

Overview

Brokers

Topics

Connect

ksqlDB

Consumers

Replicators

Cluster settings

ActiveMQSource  
Connector  
Source

Connect

ElasticsearchSink  
Connector  
Sink

Connect

GcsSinkConnector  
Sink

Connect

IbmMQSourceConnector  
Source

Connect

JdbcSinkConnector  
Sink

Connect

JdbcSourceConnector  
Source

Connect

JmsSourceConnector  
Source

Connect

S3SinkConnector  
Sink

Connect

SchemaSourceConnector  
Source

Connect

DatagenConnector  
Source

Connect

FileStreamSinkConnector  
Sink

Connect

FileStreamSource  
Connector  
Source

Connect

# Adicionar Conector

01 SETUP CONNECTION

02 TEST AND VERIFY

How should we connect to your data?

Connector class

io.confluent.kafka.connect.datagen.DatagenConnector

name

Datagen-connect

General

kafka.topic\* ⓘ

topic-users

quickstart ⓘ

users

random.seed ⓘ

Additional Properties

Add a Property

No additional properties

- Propriedades obrigatórias



# Adicionar Conector

CONTROLCENTER.CLUSTER

Overview

Brokers

Topics

Connect

ksqlDB

Consumers

Replicators

Cluster settings

CONNECT CLUSTERS > CONNECT-DEFAULT > CONNECTORS > SOURCES >

## Add Connector

01 SETUP CONNECTION

02 TEST AND VERIFY

```
{
  "name": "Datagen-connect",
  "connector.class": "io.confluent.kafka.connect.datagen.DatagenConnector",
  "kafka.topic": "topic-users",
  "quickstart": "users"
}
```

Launch

Back

[Download connector config file](#)



# Exercícios Kafka Connect

1. Adicionar o conector do Datagen pela Control Center, com as seguintes configurações:
  - Nome do conector: GeracaoDados-Datagen
  - Nome do tópico: topic-datagen
  - Quickstart: pageviews
2. Visualizar as mensagens no tópico através do Control Center

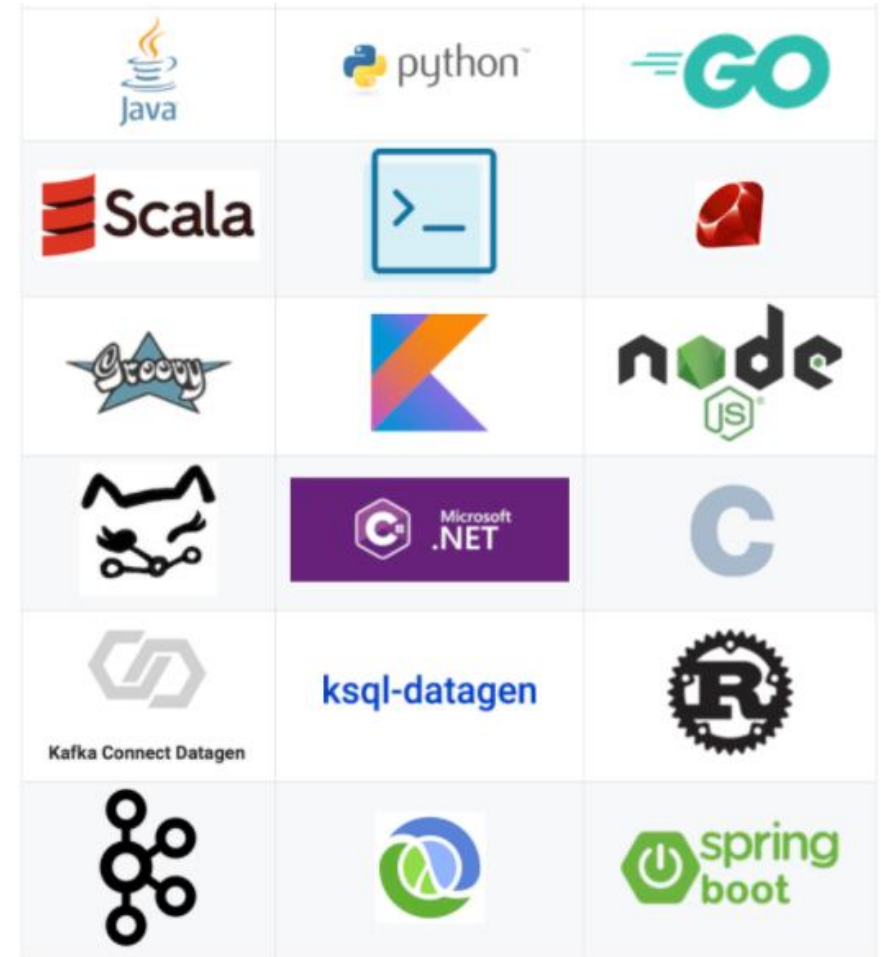


## Kafka Clients



# Kafka Clients

- Bibliotecas de clientes para várias linguagens
  - C/C++
  - Go
  - Java
  - .NET
  - Python
- Fornecem acesso de baixo nível ao Apache Kafka
- Processamento stream em alto nível
- Visualização das características para cada linguagem:
  - <https://docs.confluent.io/5.2.2/clients/>
- Exemplo de outras linguagens
  - <https://docs.confluent.io/5.5.2/tutorials/examples/clients/docs/clients-all-examples.html#clients-all-examples>








# Confluent Cloud





 CONFLUENT

PRODUCTCLOUDDEVELOPERSBLOGDOCS




DOWNLOAD

CONFLUENT CLOUD

# Apache Kafka<sup>®</sup>

## Re-engineered for the Cloud

Focus on building apps and not managing clusters with a scalable, resilient and secure event streaming platform. Event streaming with Kafka made simple on AWS, Azure and GCP clouds.









TRY FREE

# Preços Confluent Cloud

STANDARD Consumption-based	DEDICATED Custom pricing
-------------------------------	-----------------------------

- ✓ Only pay for what you use
- ✓ No minimum fees
- ✓ No upfront cost
- ✓ No termination fees

CONSUMPTION METRICS			
 <b>Data in</b> \$ / GB WRITE	\$0.11	\$0.13	\$0.22
 <b>Data out</b> \$ / GB READ	\$0.11	\$0.13	\$0.24
 <b>Data stored</b> \$ / GB RETAINED	\$0.10	\$0.10	\$0.10

Prices vary by cloud region.

# Exemplos Confluent Cloud

- Custo mensal?
  - Cluster – Exemplo 1
    - Entrada de dados: 100 GB
    - Saída de dados: 200 GB
    - Armazenamento: 500 GB
    - Nuvem: Google Cloud
  - Cluster – Exemplo 2
    - Entrada de dados: Taxa de transferência 1 MB/s
    - Saída de dados: Taxa de transferência 1 MB/s
    - Armazenamento: Política de retenção de 7 dias
    - Nuvem: Google Cloud

# Exemplos Confluent Cloud

## ○ Cluster – Exemplo 1

- Entrada de dados: 100 GB |  $100\text{GB} * 0,11/\text{GB} = \$11,00$
- Saída de dados: 200 GB |  $200\text{GB} * 0,11/\text{GB} = \$22,00$
- Armazenamento: 500 GB |  $500\text{GB} * 0,10/\text{GB} = \$50,00$
- Nuvem: Google Cloud | Total = \$83,00

## ○ Cluster – Exemplo 2

- Taxa de transferência In: 1 MB/s |  $1\text{MB/s} * 86.400 \text{ s/dia} * 30 \text{ dias} = 2531\text{GB} * 0,11/\text{GB} = \$278,41$
- Taxa de transferência Out: 1 MB/s |  $1\text{MB/s} * 86.400 \text{ s/dia} * 30 \text{ dias} = 2531\text{GB} * 0,11/\text{GB} = \$278,41$
- Política de retenção de 7 dias |  $1\text{MB/s} * 86.400 \text{ s/dia} * 7 \text{ dias} * 3\text{rep} = 1772\text{GB} * 0,10/\text{GB} = \$177,20$
- Nuvem: Google Cloud | Total = \$734.02





## Principais Parâmetros



# Kafka Consumidor – Parâmetros importantes

- group.id – Nome do grupo de consumo
- auto.offset.reset – Indica o que o Kafka fará quando não temos um offset inicial (padrão latest)
  - earliest
  - latest
- enable.auto.commit – Indica se o commit do offset será automático. (padrão true)
- max.poll.interval.ms – Intervalo máximo de busca de dados. (padrão 5 minutos)
- max.poll.records – Máximo de mensagens que são retornadas no poll (padrão 500)
- fetch.max.bytes – Quantidade de dados que “capturadas” em cada poll (padrão 52 MB)

# Kafka Produtor – Parâmetros importantes

- `linger.ms` – Tempo de envio (padrão 0)
- `acks` – Confirmação de gravação (padrão 1)
- `retries` – Tentativas (padrão 2147483647)
- `compression.type` – Tipo de compressão (padrão none)
- `max.in.flight.requests.per.connection` – Mensagens em voo (padrão 5)



# Otimização





# Kafka Tipos de otimização

Throughput

Latência

Durabilidade

Disponibilidade

<https://www.confluent.io/wp-content/uploads/Optimizing-Your-Apache-Kafka-Deployment-1.pdf>

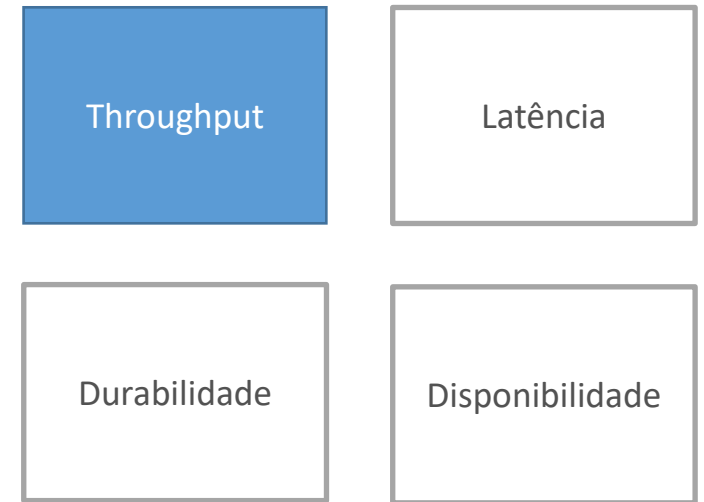
# Kafka Otimizando o throughput

- Produtor

- `linger.ms`: aumente para 10 - 100 (padrão 0)
- `compression.type=lz4` (padrão none)
- `acks=1` (padrão 1)
- `retries=0` (padrão 0)
- `batch.size`: aumente para 100000 - 200000 (padrão 16384)

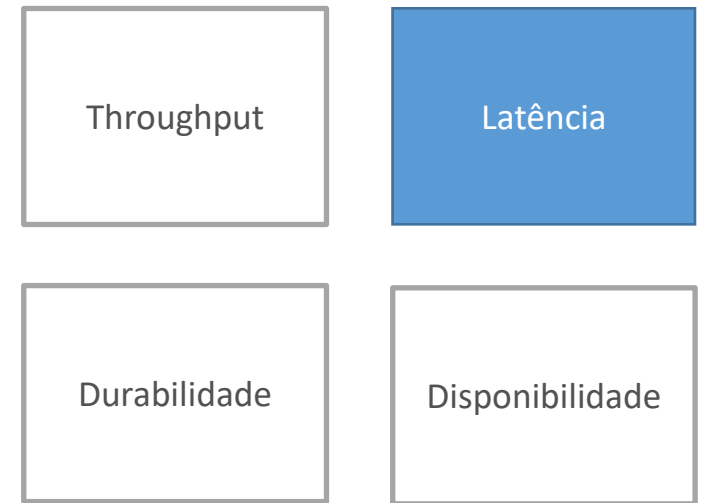
- Consumidor

- `fetch.min.bytes`: aumente para ~100000 (default 1)



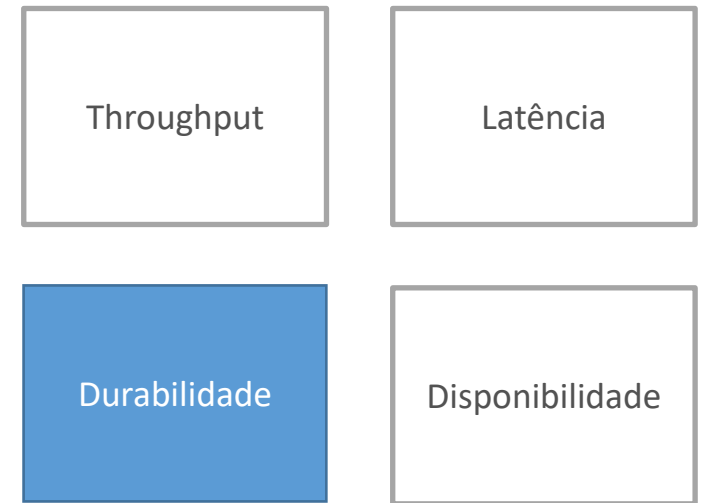
# Kafka Otimizando a latência

- Produtor
  - `linger.ms=0` (padrão 0)
  - `compression.type=none` (padrão none)
  - `acks=1` (padrão 1)
- Consumidor
  - `fetch.min.bytes=1` (padrão 1)



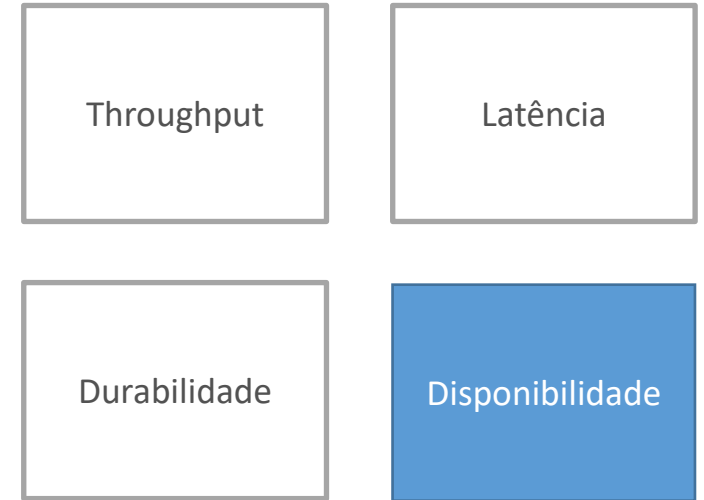
# Kafka Otimizando a durabilidade

- Produtor
  - replication.factor: 3,
  - acks=all (padrão 1)
  - retries: 1 ou mais (padrão 0)
  - max.in.flight.requests.per.connection=1 (padrão 5)
    - Prevenir mensagens fora de ordem
- Consumidor
  - enable.auto.commit=false (padrão true)
- Broker
  - default.replication.factor=3 (padrão 1)
  - auto.create.topics.enable=false (padrão true)
  - min.insync.replicas=2 (default 1);



# Kafka Otimizando a disponibilidade

- Broker
  - `unclean.leader.election.enable=true` (default true);
  - `min.insync.replicas=1` (default 1);
  - `default.replication.factor=3` (padrão 1)
- Consumidor
  - `session.timeout.ms`: Baixar o quanto possivel(default 10000)

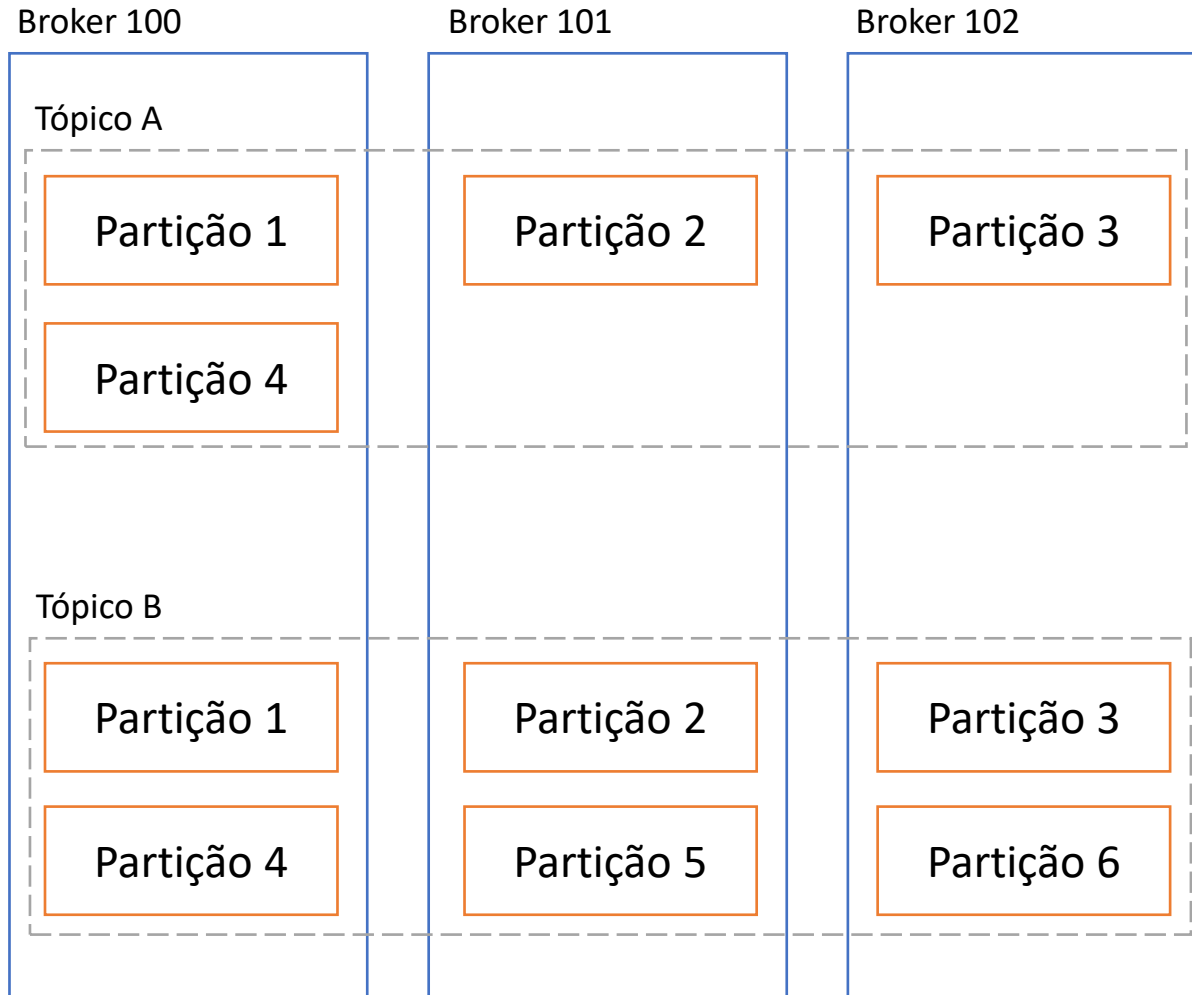




## Melhores Práticas



# Kafka Número de partições

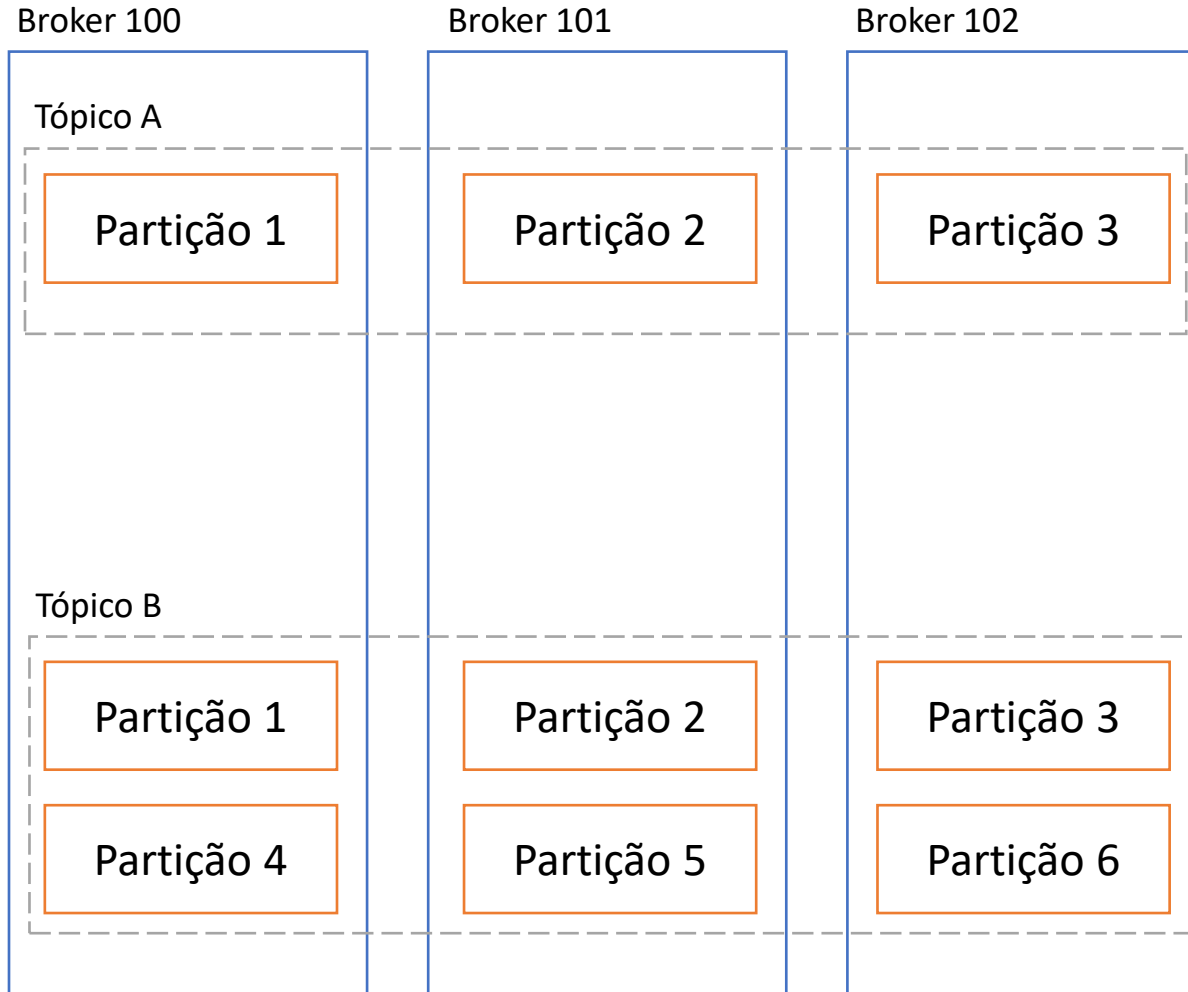


- Partições múltiplo do número de brokers



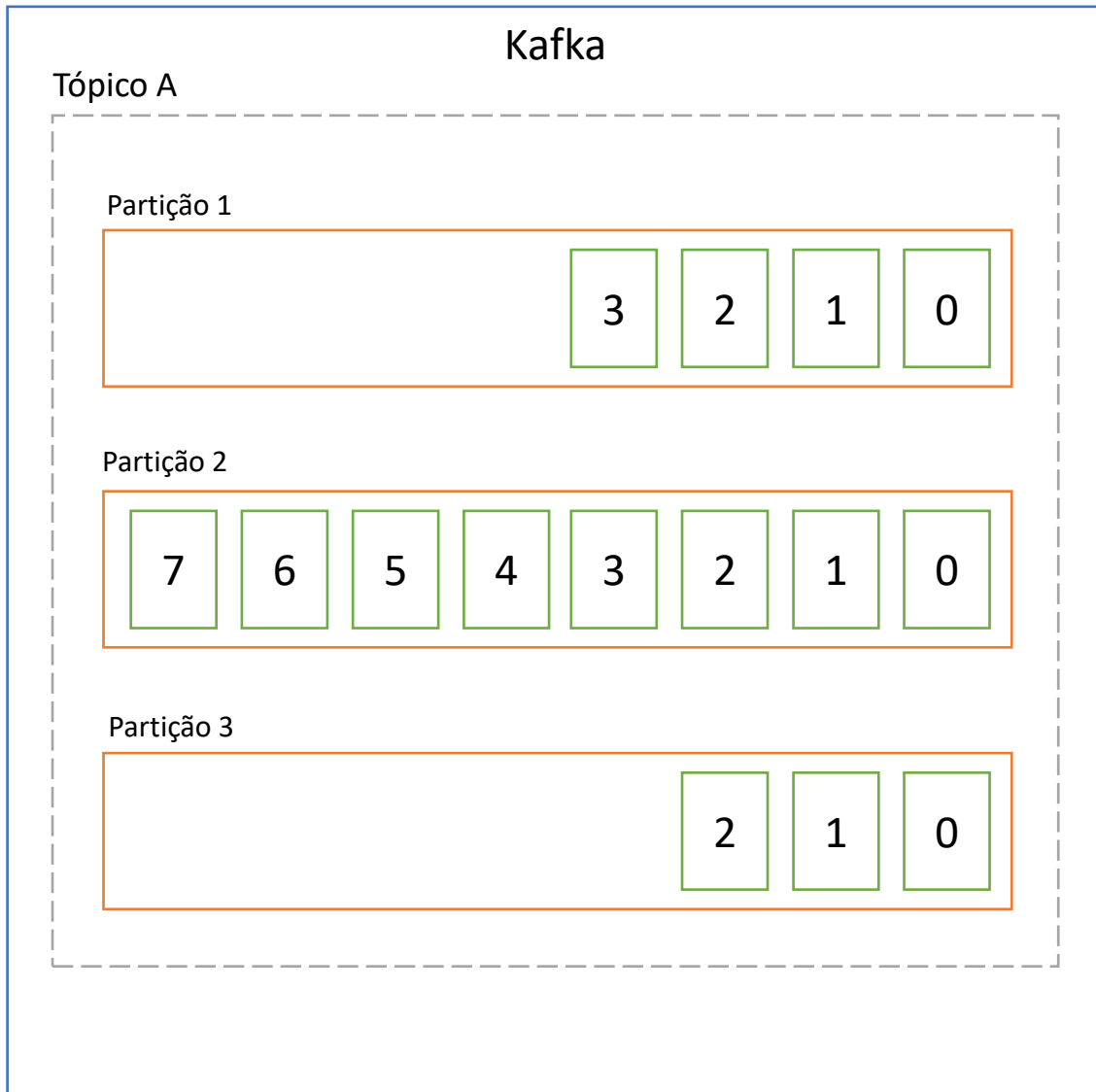


# Kafka Quantidade de partições



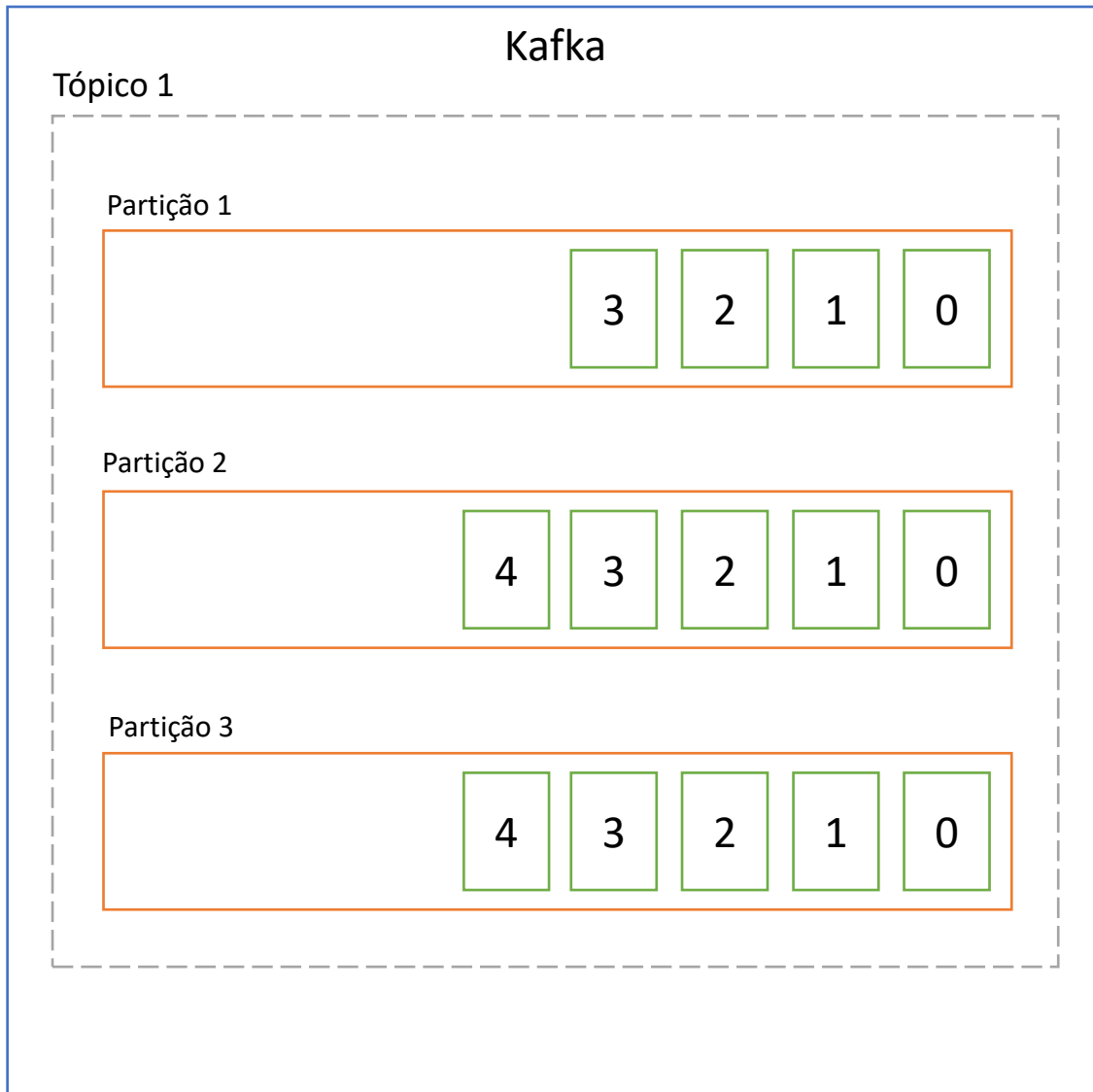
- Para determinar o número de partições
  - Throughput você quer atingir
  - Quanto cada partição entrega
- Comandos
  - `kafka-consumer-perf-test.sh`
  - `kafka-producer-perf-test.sh`
  - `kafka-run-class.sh kafka.tools.TestEndToEndLatency`

# Kafka Balanceamento das partições



- Importante entender o comportamento dos seus dados
  - Manter uma uniformidade de crescimento entre as partições.

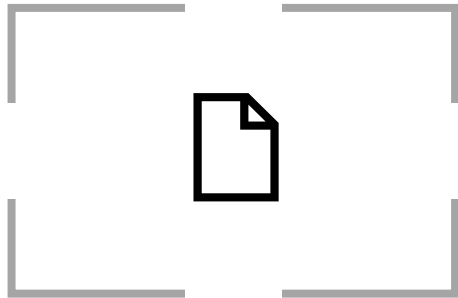
# Kafka Tempo de retenção



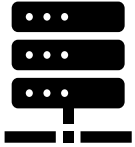
- Manter partições com 25GB
  - Facilitar o gerenciamento
- O tempo de retenção do Kafka
  - SLA da aplicação
  - Necessidade de reprocessamento
  - Regra de negócio / Compliance

# Kafka Criptografia e compressão

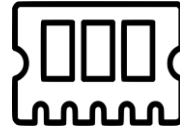
- Não aplicar compressão ou criptografia no disco do broker
- Aplique a compressão ou criptografia na mensagem



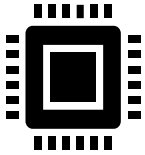
# Kafka Monitoramento



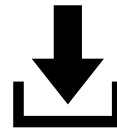
Utilização de Disco



Utilização de Memória



Utilização de CPU



I/O de Disco



I/O de Rede



Arquivos abertos



## Estudo Adicional



# Melhorar o aprendizado

- Leitura do Livro
  - Kafka: The Definitive Guide
- Link
  - <https://www.confluent.io/resources/kafka-the-definitive-guide>



CONFLUENT

PRODUCT CLOUD DEVELOPERS BLOG DOCS DOWNLOAD

## Kafka: The Definitive Guide



Learn how to take full advantage of Apache Kafka®, the distributed, publish-subscribe queue for handling real-time data feeds. With this comprehensive book, you'll understand how Kafka works and how it's designed.

Authors Neha Narkhede, Gwen Shapira, and Todd Palino show you how to deploy production Kafka clusters; secure, tune, and monitor them; write rock-solid applications that use Kafka; and build scalable stream-processing applications.

**Download the Book**

First Name

Last Name

Email

By clicking "Download Now" you agree to receive occasional marketing emails from Confluent. You also agree that your personal data will be processed in accordance with our [Privacy Policy](#).


**DOWNLOAD NOW**



# Melhorar o aprendizado

- Github da Confluent
  - <https://github.com/confluentinc>

github.com/confluentinc



**Confluent Inc.**  
Real-time streams powered by Apache Kafka®  
Mountain View, CA <https://confluent.io> [contact@confluent.io](mailto:contact@confluent.io) Verified


Repositories 158

Packages

People 147


Projects

Dismiss




**Grow your team on GitHub**  
GitHub is home to over 40 million developers working together. Join them to grow your own development teams, manage permissions, and collaborate on projects.

Sign up




Pinned repositories



**ksql**

The event streaming database purpose-built for stream processing applications


Java ★ 3.3k 🍴 622



**examples**

Apache Kafka and Confluent Platform examples and demos


TSQL ★ 573 🍴 398



**schema-registry**


Confluent Schema Registry for Kafka

Java ★ 1.2k 🍴 720




**cp-docker-images**

Docker images for Confluent Platform.



**kafka-streams-examples**


Demo applications and code examples for Apache Kafka's Streams API.



**kafka-rest**

Confluent REST Proxy for Kafka

39

 Semantix



## Types of Courses

Confluent offers instructor-led courses in both traditional and virtual classroom formats.  
Or we can bring the training to your team.



### Classroom Training

Available in a classroom or live, online with the same experience. View our schedule below.

View schedule



### On Demand Training

Self-paced delivery of our classroom curriculum. Includes lecture and online labs.

View Schedule



### Private Training

Train a large number of your professionals at the same time, at your location.

Contact Us

## Confluent Training Offerings



Confluent Fundamentals for Apache Kafka

Get Started



Apache Kafka® Administration by Confluent

Get Started



Confluent Developer Skills for Building Apache Kafka

Get Started



Confluent Stream Processing using Apache Kafka® Streams & KSQL

Get Started



Confluent Advanced Skills for Optimizing Apache Kafka

Get Started



# Semantix

## Obrigado!

Alguma pergunta?



Você pode me encontrar em:  
[rodrigo.augusto@semantix.com.br](mailto:rodrigo.augusto@semantix.com.br)

**GET SMARTER**