



Semantix

Kafka - Básico

Aula 2

Quem sou eu?

Eu sou Rodrigo Augusto Rebouças.

Engenheiro de dados da Semantix
Instrutor do Semantix Academy

Você pode me encontrar em:
rodrigo.augusto@semantix.com.br

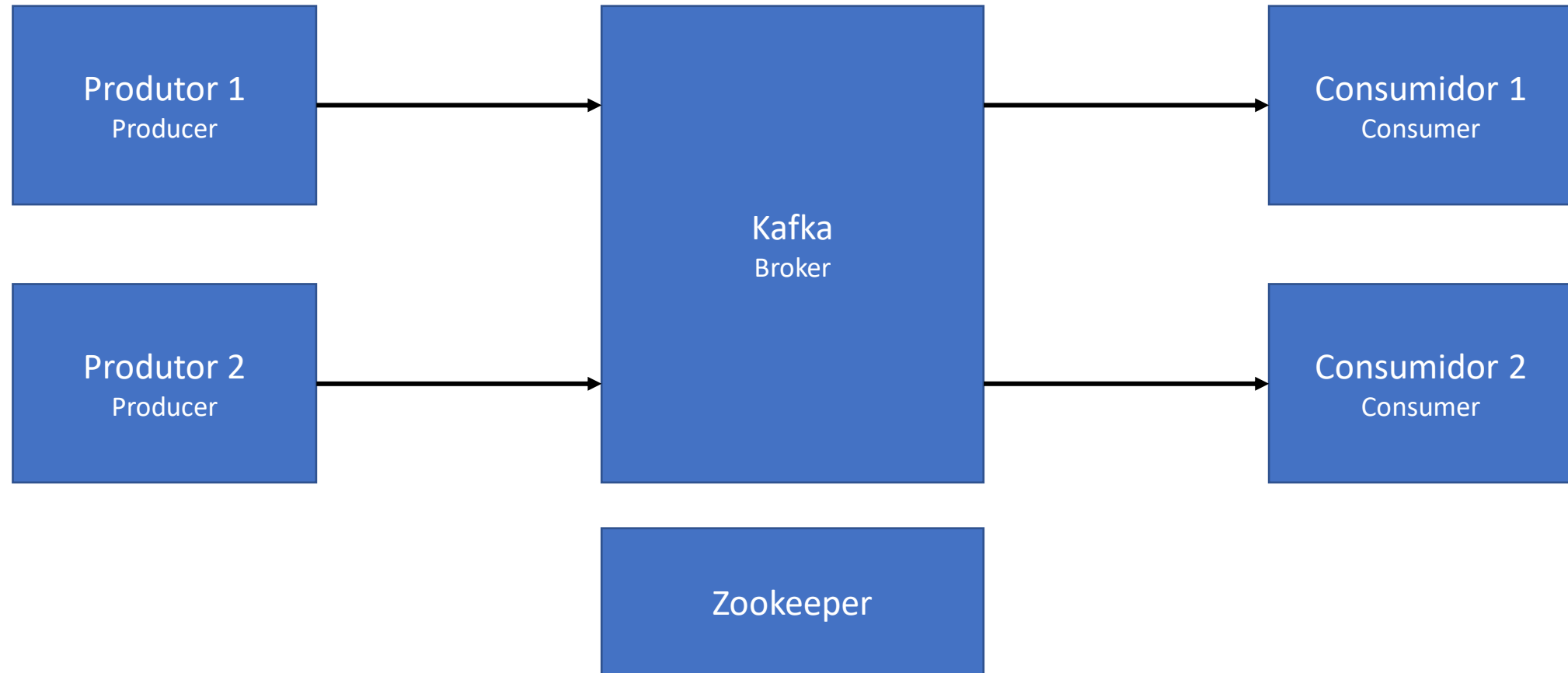




Arquitetura

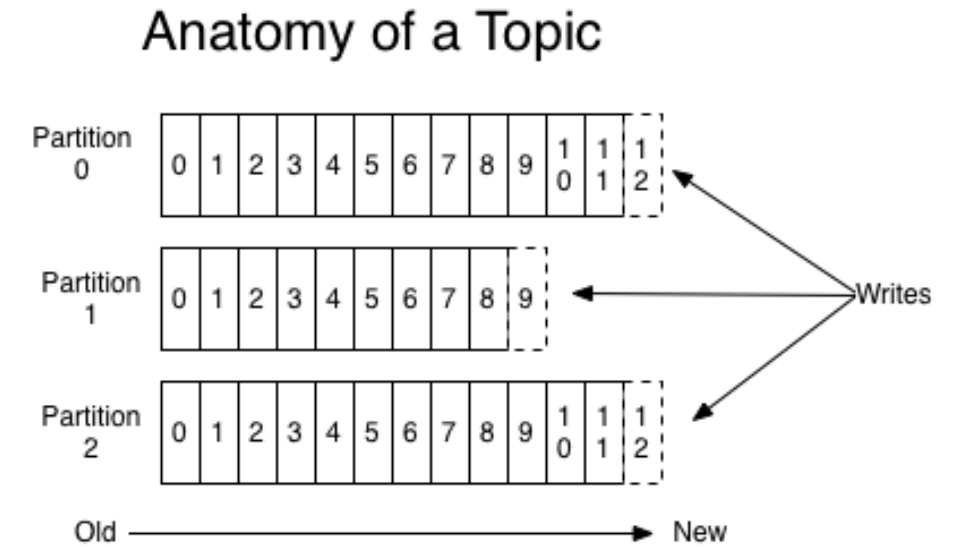


Arquitetura



Tópicos

- Fluxo de registros
- Similar a uma Tabela (SQL)
- Divididos em partições
 - Local onde as mensagens são gravadas
 - Sequência ordenada e imutável de registro
 - Cada registro na partição é atribuído a um id sequencial (offset)
 - Exclusivamente do registro na partição
- Multi-assinantes
 - 0, 1 ou muitos consumidores acessam os dados gravados

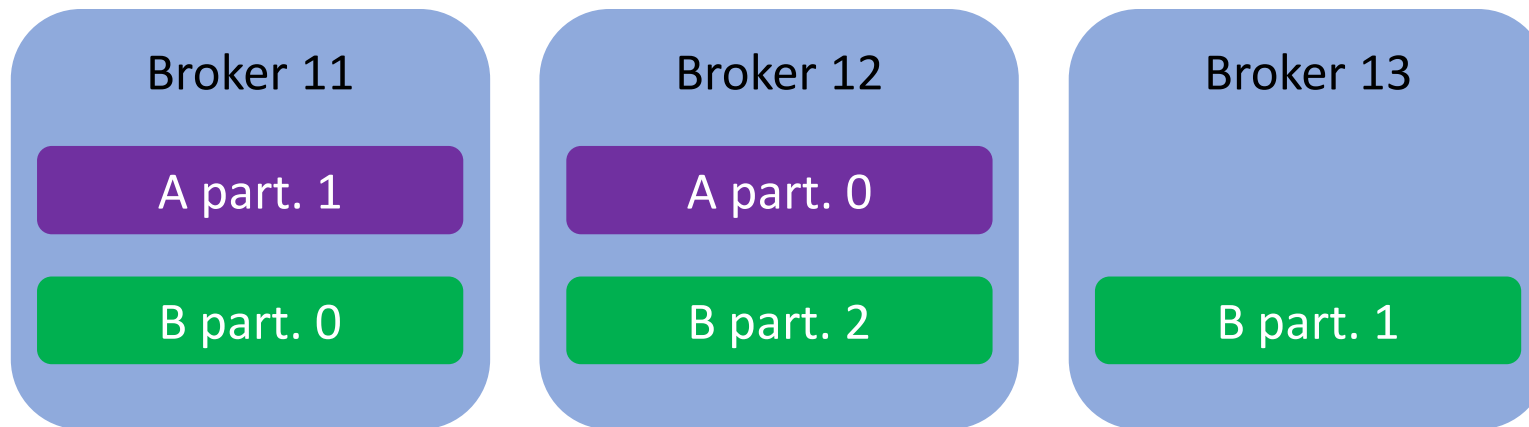


Brokers Conceitos

- Corretores
- Armazenam os tópicos
- Cluster Kafka é composto por múltiplos corretores (servidores)
 - Ambientes de produção ter no mínimo 3
- Corretor é identificado por um id

Brokers Exemplo

- Cluster Kafka
 - 3 brokers
 - Tópico A com 2 partições
 - Tópico B com 3 partições

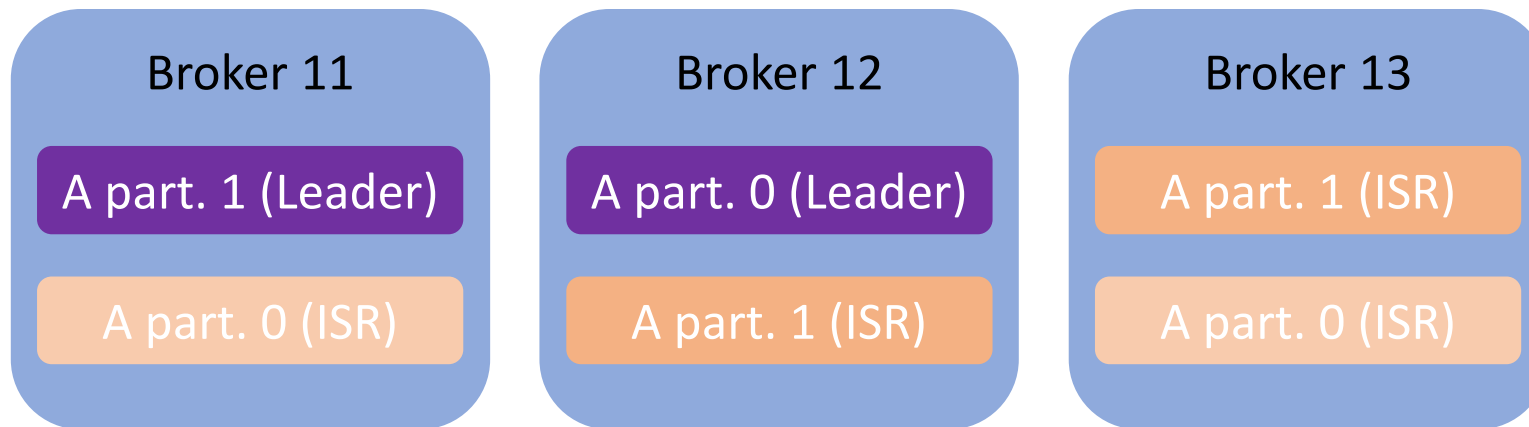


Replicação dos tópicos

- Boa prática para cada partição
 - 1 corretor líder (Leader)
 - Receber os dados
 - 2 corretores de réplica do líder (ISR – in-sync replica)
 - Sincronizar os dados

Replicação dos tópicos - Exemplo

- Cluster Kafka
 - 3 brokers
 - Tópico A
 - 2 partições
 - 3 réplicas



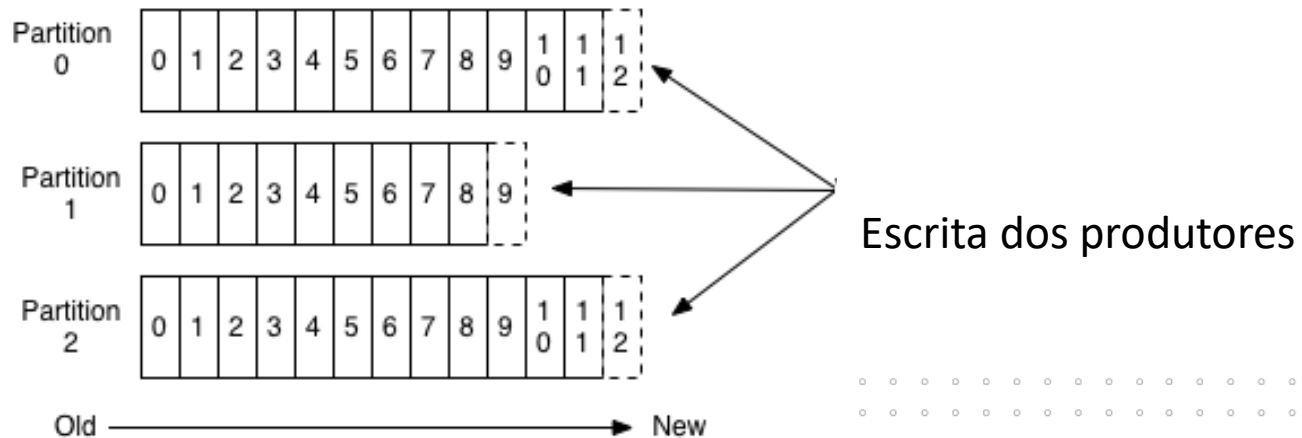
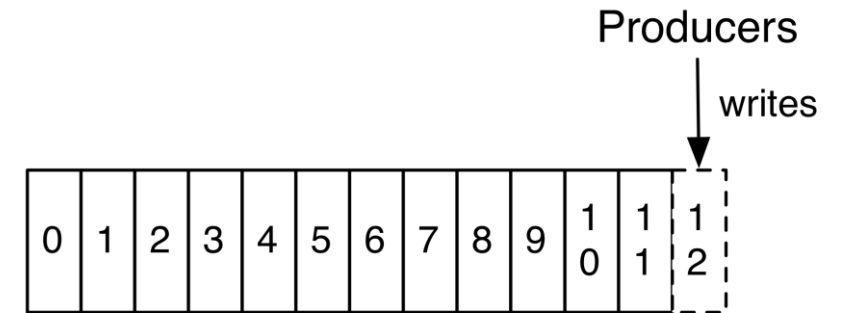


Producers



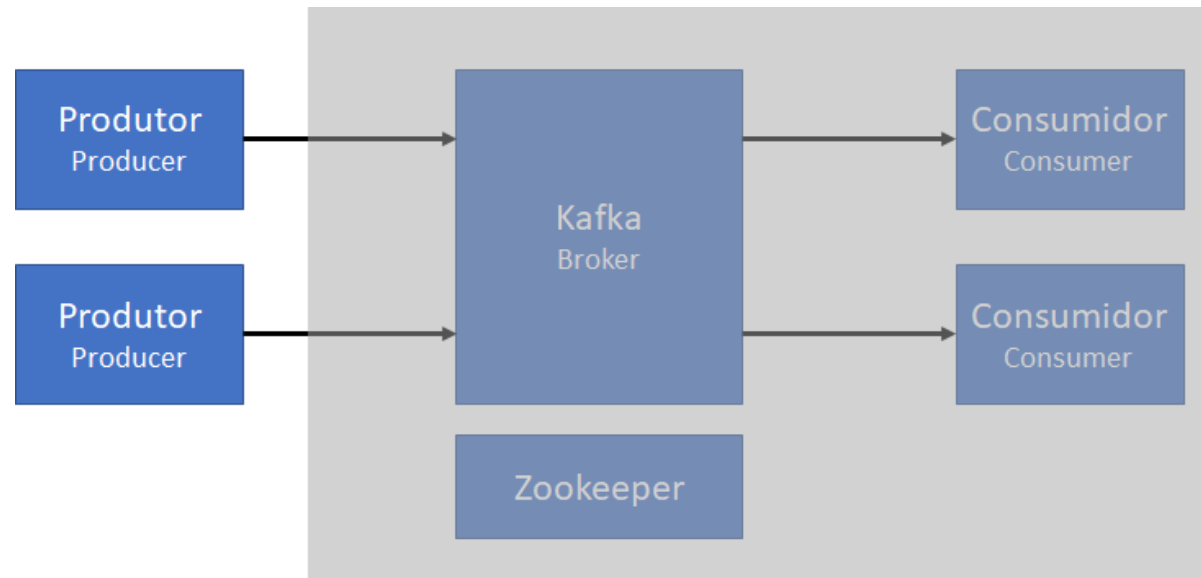
Producers Conceitos

- Produtores
 - Enviar os dados
- Publicar dados nos tópicos de sua escolha
- Escolher qual registro atribuir a qual partição dentro do tópico
 - Balancear a carga
 - Chave no registro



Producers Confirmação de escrita

- Existem 3 tipos de confirmação de escrita (acks) para o produtor
 - 0 – Sem confirmação de escrita
 - 1 – Confirmação de escrita no líder (Padrão)
 - All – Confirmação de escrita no líder e nas replicas (ISR)



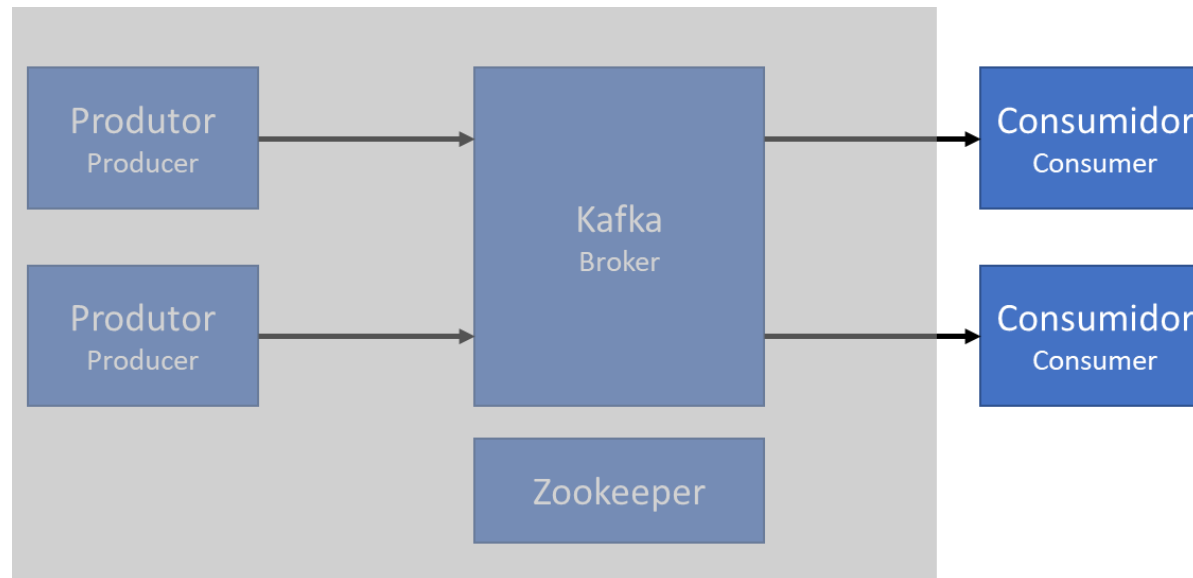
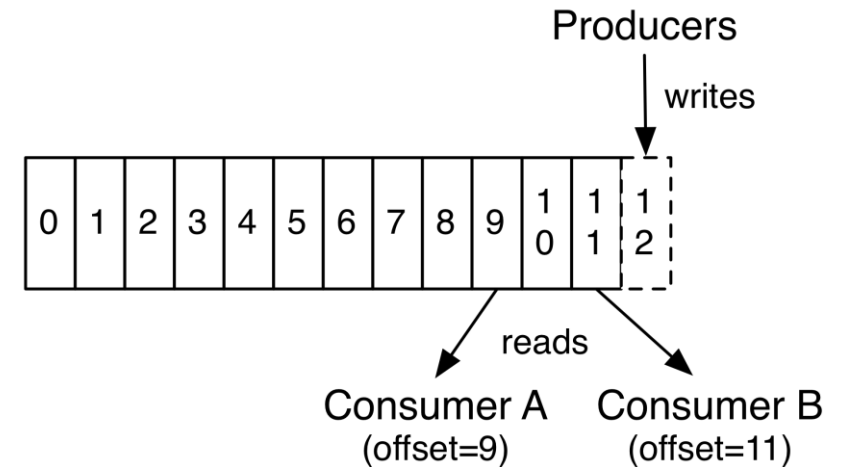


Consumers



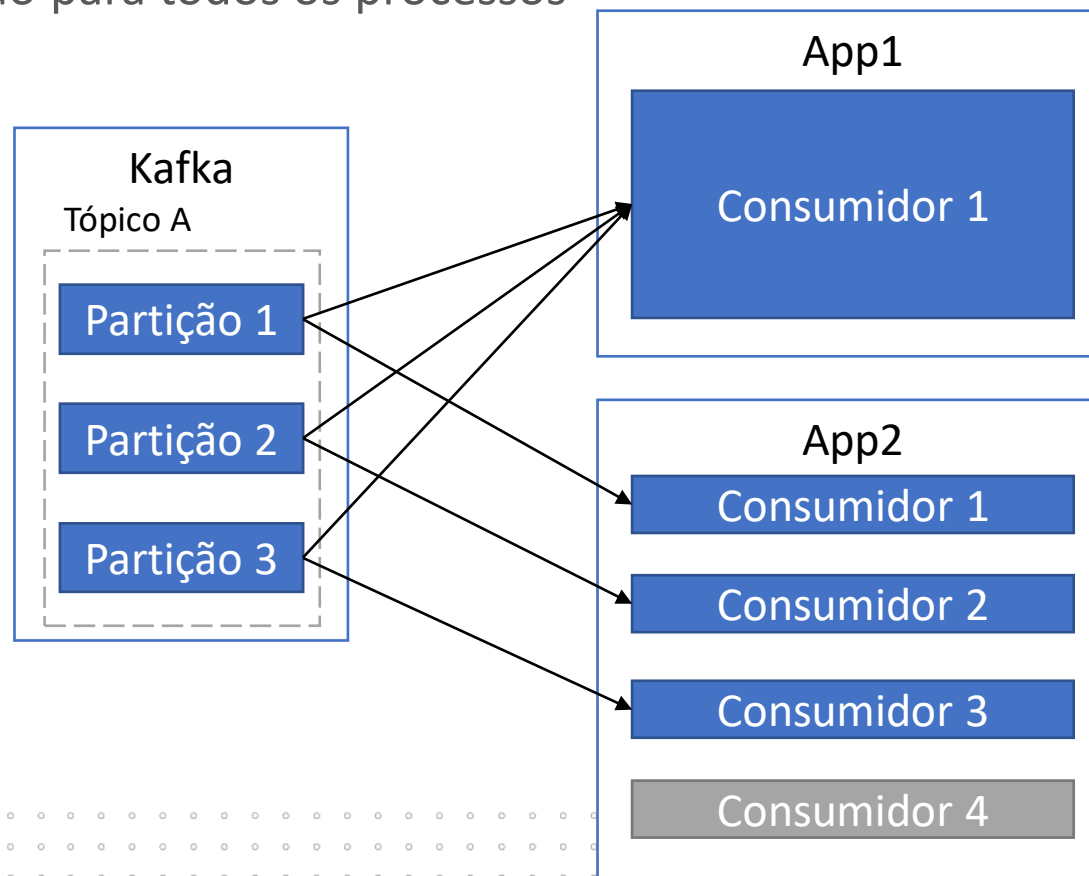
Consumers Conceitos

- Consumidores
 - Receber os dados
- Cada registro publicado em um tópico
 - Entregue aos consumidores dentro de grupo de consumidores



Consumers Grupo de consumidores

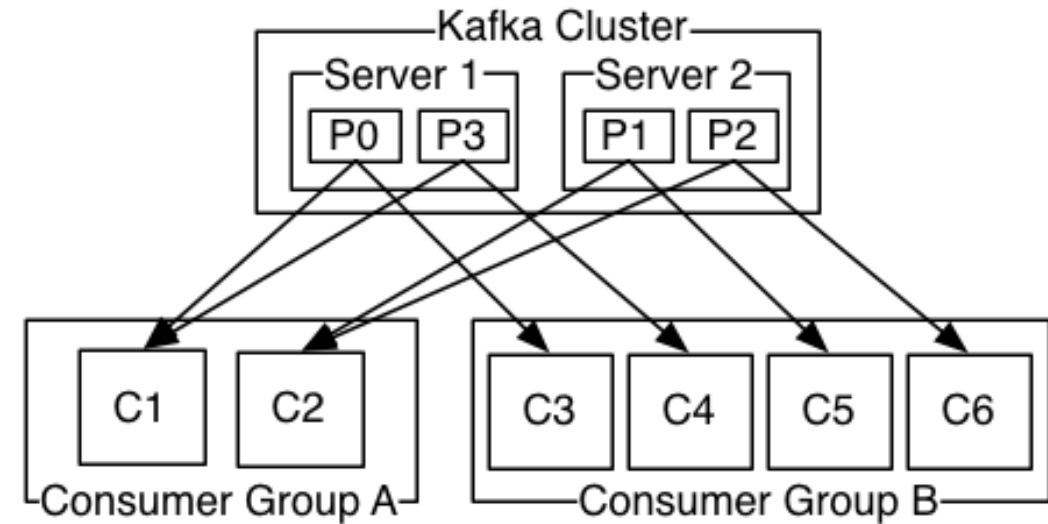
- Se todas as instâncias do consumidor tiverem no mesmo grupo de consumidores
 - registros serão balanceados por carga
- Se todas as instâncias do consumidor tiverem em grupos de consumidores diferentes
 - Cada registro será transmitido para todos os processos



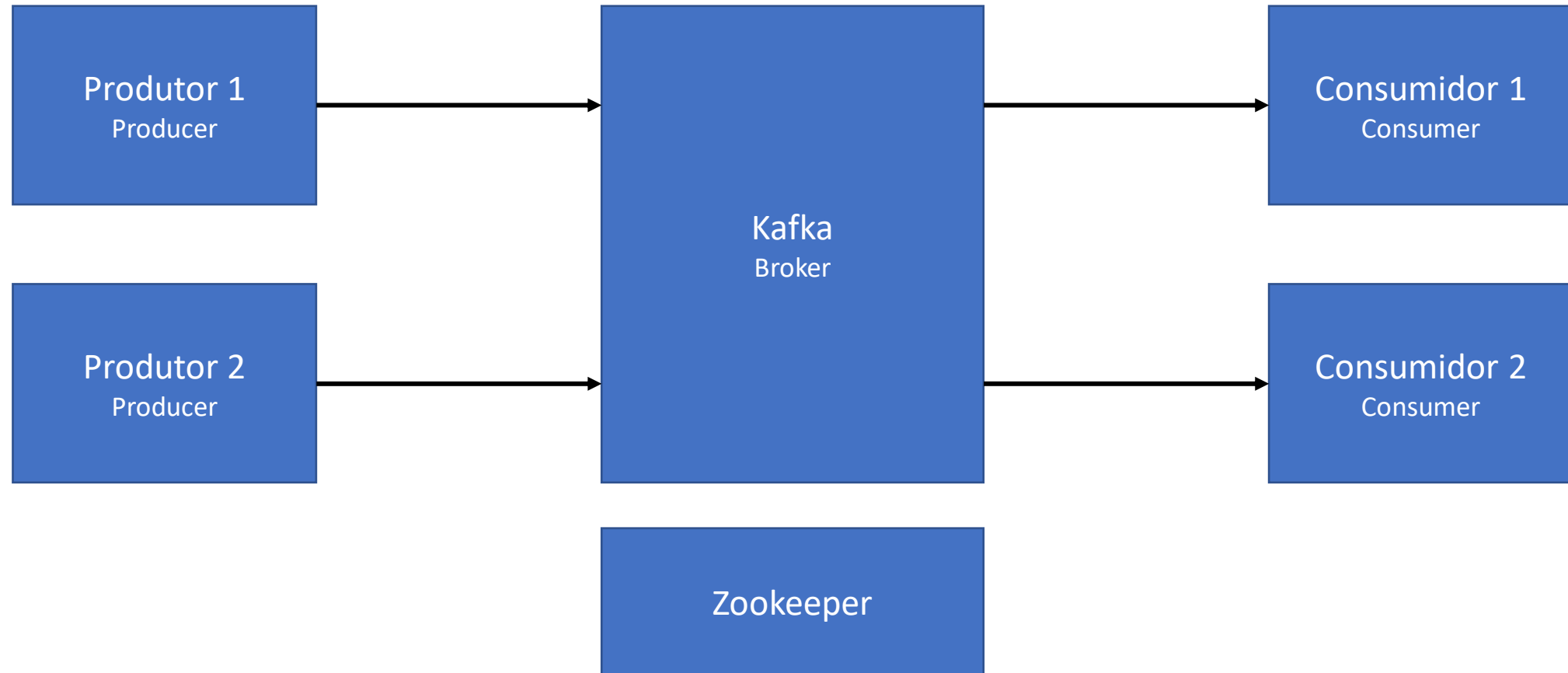
Grupo de consumidores Exemplo

○ Cluster Kafka

- 2 servidores (brokers)
- 4 partições
- 2 grupos de consumidores
 - A – 2 instâncias de consumidor
 - B – 2 instâncias de consumidor



Arquitetura





Gerenciar tópicos

Linha de Comando



Acessar Broker - Container

- Comandos para acessar o Servidor Kafka
 - `docker exec -it broker bash`

```
semantix@NTBSTX7158:~/treinamentos/kafka
$ docker-compose start
Starting zookeeper      ... done
Starting broker         ... done
Starting schema-registry ... done
Starting connect        ... done
Starting ksqldb-server  ... done
Starting control-center ... done
Starting ksqldb-cli     ... done
Starting ksql-datagen   ... done
Starting rest-proxy     ... done
semantix@NTBSTX7158:~/treinamentos/kafka
$ docker exec -it broker bash
root@broker:/# kafka-topics --version
5.5.2-ce (Commit:417a2e7a085d90a7)
```

Tópicos Comandos Básicos

- Listar tópicos

kafka-topics --bootstrap-server localhost:9092 --list

ou

kafka-topics --zookeeper localhost:2181 --list

- Criar tópico

kafka-topics --bootstrap-server localhost:9092 --topic <nomeTópico> --create \

--partitions **3** --replication-factor **1**

- Descrever tópico

kafka-topics --bootstrap-server localhost:9092 --topic <nomeTópico> --describe

- Deletar tópico

kafka-topics --bootstrap-server localhost:9092 --topic <nomeTópico> --delete



Producer

Linha de Comando



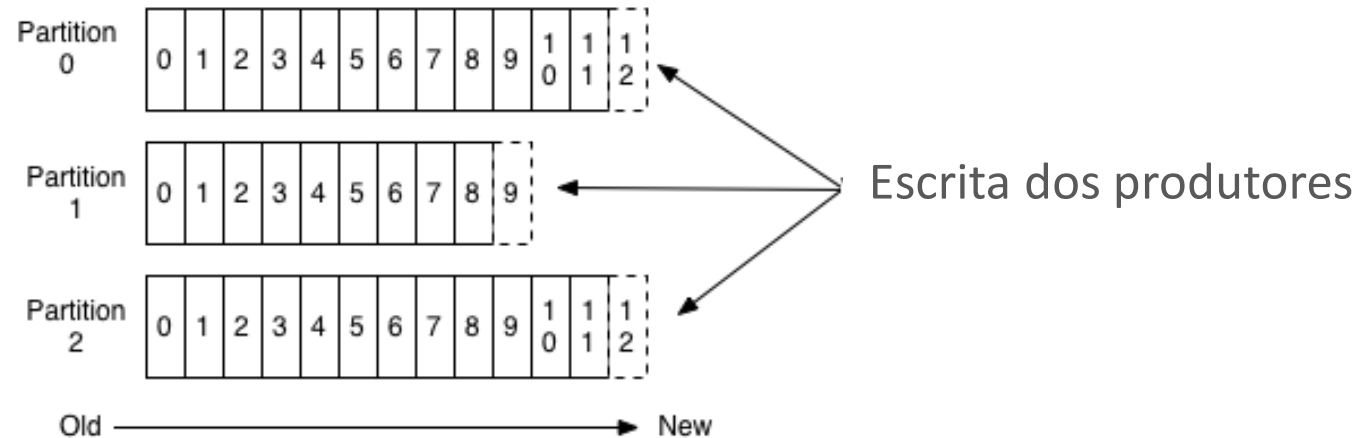
Produtor Console

- Enviar dados

```
kafka-console-producer --broker-list localhost:9092 --topic <nomeTópico>
```

- Enviar dados para todos reconhecerem (Leader e ISR)

```
kafka-console-producer --broker-list localhost:9092 --topic <nomeTópico> --producer-property acks=all
```





Consumer

Linha de Comando

Consumidor Console

- Receber mensagens em tempo real

kafka-console-consumer --bootstrap-server localhost:9092 --topic <nomeTópico>

- Receber mensagens desde a criação do tópico

kafka-console-consumer --bootstrap-server localhost:9092 --topic <nomeTópico> --from-beginning

- Criar grupo de consumidores

kafka-console-consumer --bootstrap-server localhost:9092 --topic <nomeTópico> --group <nomeGrupo>

Grupos de consumidores - Console

- Listar grupos

```
kafka-consumer-groups --bootstrap-server localhost:9092 --list
```

- Descrever grupo

```
kafka-consumer-groups --bootstrap-server localhost:9092 --describe --group <nomeGrupo>
```

- Redefinir o deslocamento do mais antigo

```
kafka-consumer-groups --bootstrap-server localhost:9092 --group <nomeGrupo> \  
--reset-offsets --to-earliest --execute --topic <nomeTópico>
```

- Alterar o deslocamento

- kafka-consumer-groups --bootstrap-server localhost:9092 \
--group <nomeGrupo> --reset-offsets --shift-by 2 --execute --topic <nomeTópico>

Exercícios Kafka

1. Criar o tópico msg-cli com 2 partições e 1 réplica
2. Descrever o tópico msg-cli
3. Criar o consumidor do grupo app-cli
4. Enviar as seguintes mensagens do produtor
 - a) Msg 1
 - b) Msg 2
5. Criar outro consumidor do grupo app-cli
6. Enviar as seguintes mensagens do produtor
 - a) Msg 4
 - b) Msg 5
 - c) Msg 6
 - d) Msg 7
7. Criar outro consumidor do grupo app2-cli
8. Enviar as seguintes mensagens do produtor
 - a) Msg 8
 - b) Msg 9
 - c) Msg 10
 - d) Msg 11
9. Parar o app-cli
10. Definir o deslocamento para -2 offsets do app-cli
11. Descrever grupo
12. Iniciar o app-cli
13. Redefinir o deslocamento o app-cli
14. Iniciar o app-cli
15. Listar grupo



Semantix

Obrigado!

Alguma pergunta?



Você pode me encontrar em:
rodrigo.augusto@semantix.com.br

GET SMARTER