



Try **FREE**
confluent.io/cloud



**\$50 Free
each month**



**3 Months
from signup**



Fundamentals for Apache Kafka®

How Apache Kafka Works

Session Schedule



- Session 1: Benefits of Stream Processing and Apache Kafka Use Cases
- Session 2: Apache Kafka Architecture & Fundamentals Explained
- **Session 3: How Apache Kafka Works**
- Session 4: Integrating Apache Kafka into your Environment

Learning Objectives



After this module you will be able to:



- Give a high level description of the programming logic in Kafka producer and consumer clients
- Explain how EOS works to an interested lay person
- List the means with which Kafka provides durability and HA
- Illustrate on a high level, how you can secure your Kafka cluster

Development: A Basic Producer in Java



```
BasicProducer.java x
1 package clients;
2
3 import java.util.Properties;
4 import org.apache.kafka.clients.producer.KafkaProducer;
5 import org.apache.kafka.clients.producer.ProducerRecord;
6
7 public class BasicProducer {
8     public static void main(String[] args) {
9         System.out.println("## Starting Basic Producer ##");
10
11         Properties settings = new Properties();
12         settings.put("client.id", "basic-producer-v0.1.0");
13         settings.put("bootstrap.servers", "kafka-1:9092,kafka-2:9092");
14         settings.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer");
15         settings.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer");
16
17         final KafkaProducer<String, String> producer = new KafkaProducer<>(settings);
18
19         Runtime.getRuntime().addShutdownHook(new Thread(() -> {
20             System.out.println("## Stopping Basic Producer ##");
21             producer.close();
22         }));
23
24         final String topic = "hello_world_topic";
25         for(int i=1; i<=5; i++){
26             final String key = "key-" + i;
27             final String value = "value-" + i;
28             final ProducerRecord<String, String> record = new ProducerRecord<>(topic, key, value);
29             producer.send(record);
30         }
31     }
32 }
```

configuration

create producer

shutdown behaviour

sending data

Development: A Basic Consumer in .NET/C#

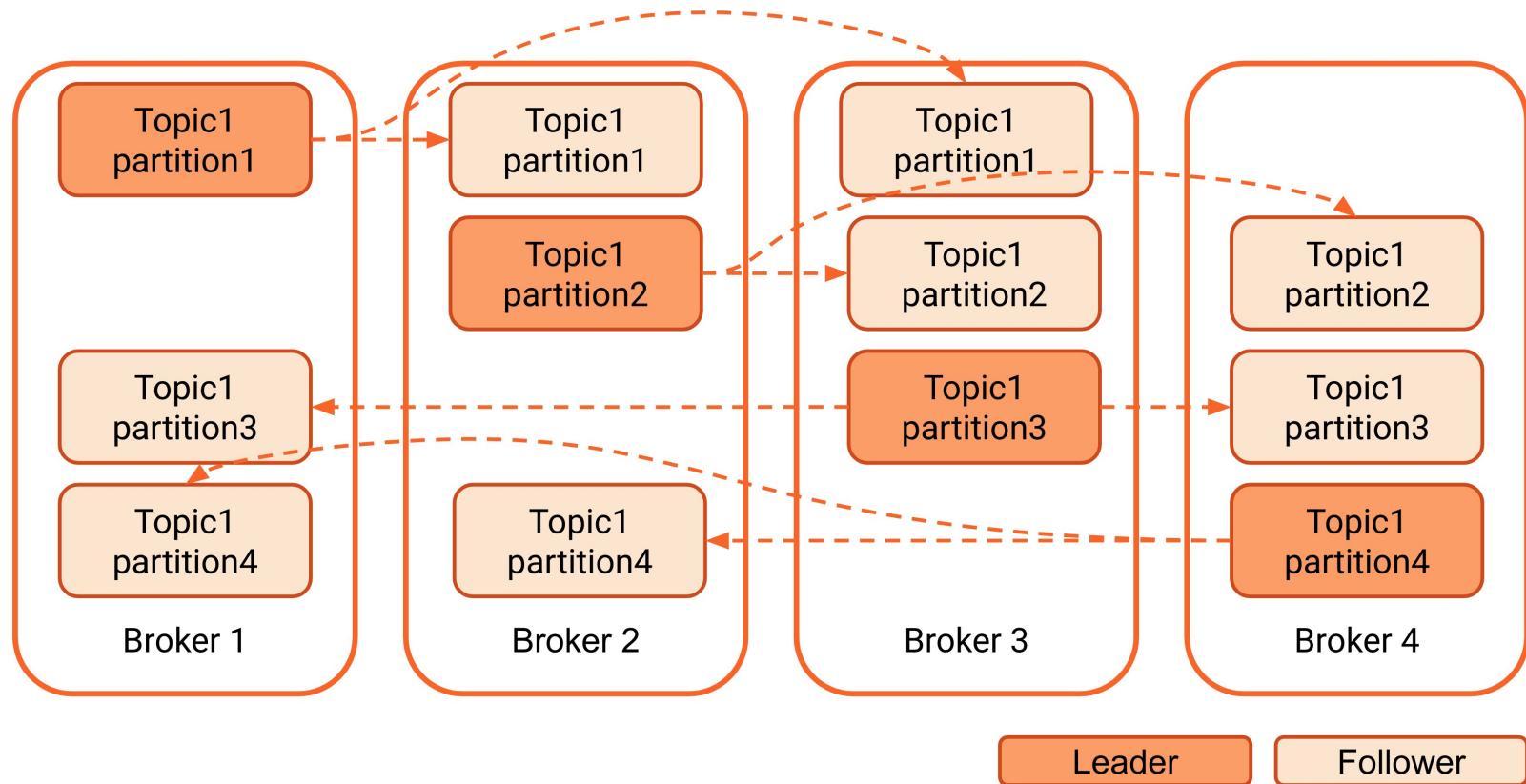


```
8  namespace consumer_net {
9    0 references
10   class Program {
11     0 references
12     static void Main (string[] args) {
13       Console.WriteLine ("Starting Consumer!");
14       var config = new Dictionary<string, object> {
15         { "group.id", "dotnet-consumer-group" },
16         { "bootstrap.servers", "kafka-1:9092" },
17         { "auto.commit.interval.ms", 5000 },
18         { "auto.offset.reset", "earliest" }
19       };
20
21       var deserializer = new StringDeserializer (Encoding.UTF8);
22       using (var consumer = new Consumer<string, string> (config, deserializer, deserializer)) {
23         consumer.OnMessage += (_, msg) =>
24           Console.WriteLine ($"Read ('{msg.Key}', '{msg.Value}') from: {msg.TopicPartitionOffset}");
25
26         consumer.OnError += (_, error) =>
27           Console.WriteLine ($"Error: {error}");
28
29         consumer.OnConsumeError += (_, msg) =>
30           Console.WriteLine ($"Consume error ({msg.TopicPartitionOffset}): {msg.Error}");
31
32         consumer.Subscribe ("hello_world_topic");
33
34         while (true) {
35           consumer.Poll (TimeSpan.FromMilliseconds (100));
36         }
37     }
38 }
```

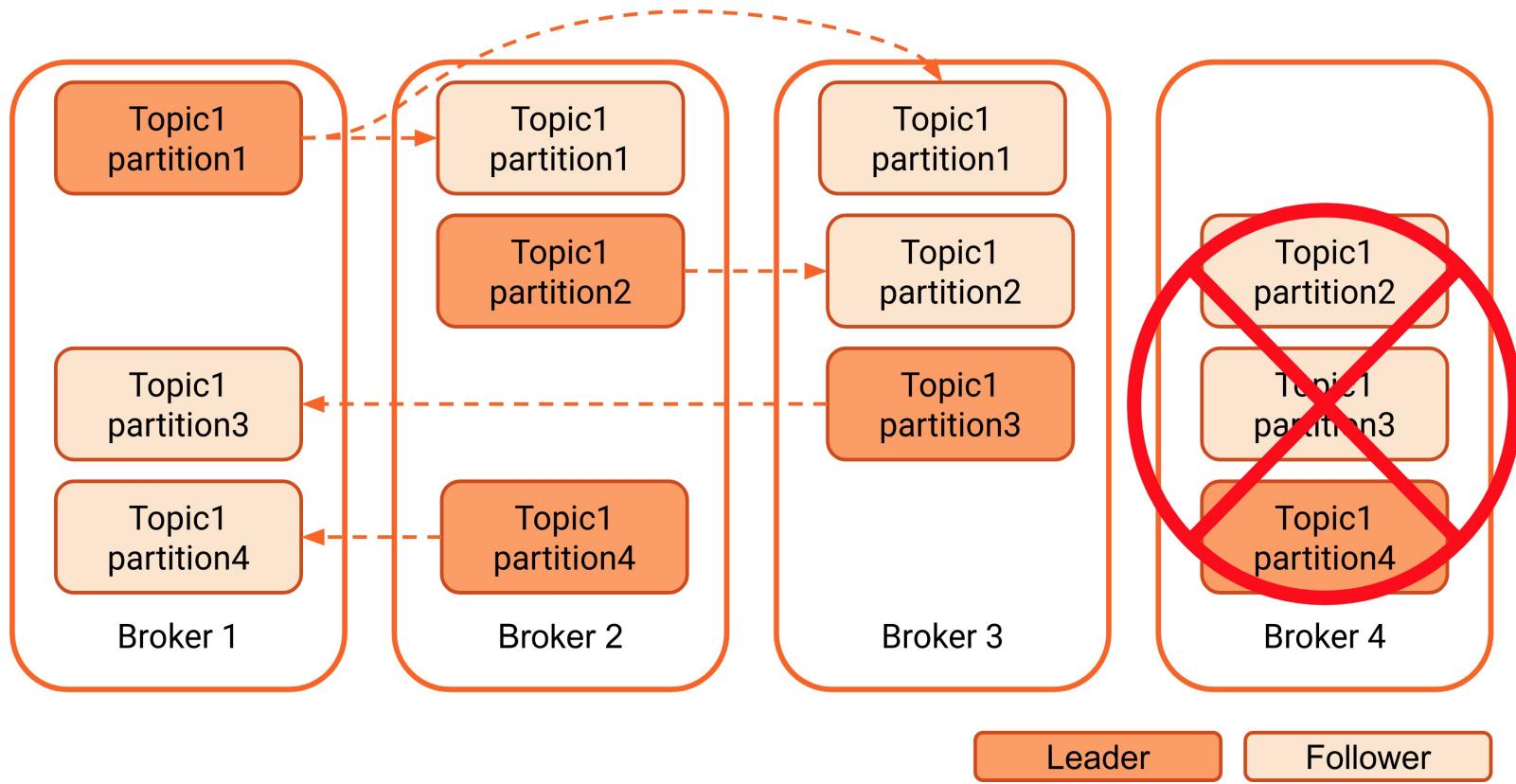
Handwritten annotations with arrows pointing to specific code sections:

- A red box highlights the configuration block (lines 13-18), with a red arrow labeled "configuration" pointing to it.
- A yellow box highlights the consumer.OnMessage event handler (line 21), with a red arrow labeled "message handling" pointing to it.
- A yellow box highlights the consumer.OnError event handler (line 26), with a red arrow labeled "error handling" pointing to it.
- A green box highlights the consumer.OnConsumeError event handler (line 29), with a red arrow labeled "error handling" pointing to it.
- A green box highlights the consumer.Subscribe call (line 32), with a red arrow labeled "subscribing to topic" pointing to it.
- A green box highlights the consumer.Poll loop (lines 34-35), with a red arrow labeled "polling data" pointing to it.

Partition Leadership & Replication



Partition Leadership & Replication

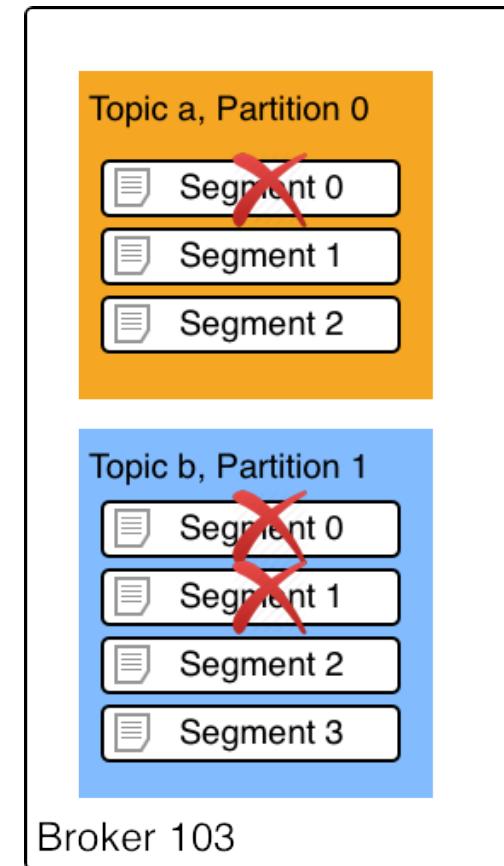




Data Retention Policy

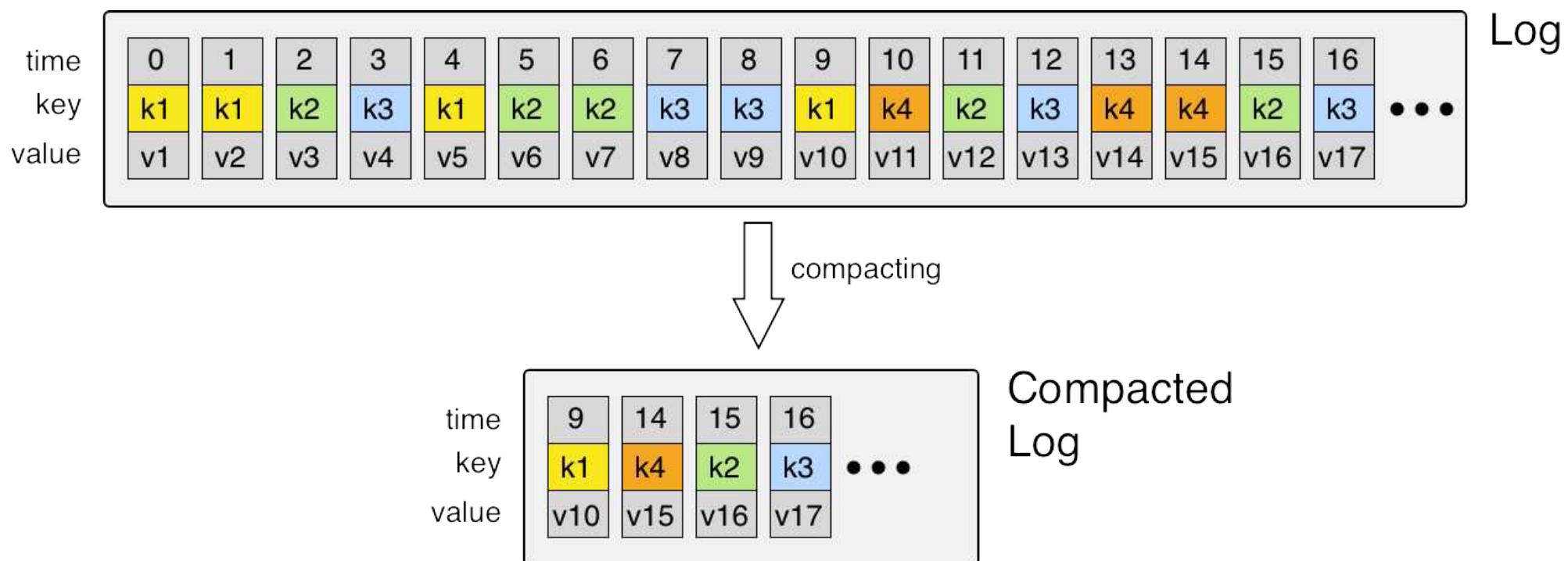
How long do I want or can I store my data?

- How long (default: **one week**)
- Set **globally** or **per topic**
- Business decision
- Cost factor
- Compliance factor (e.g., GDPR)

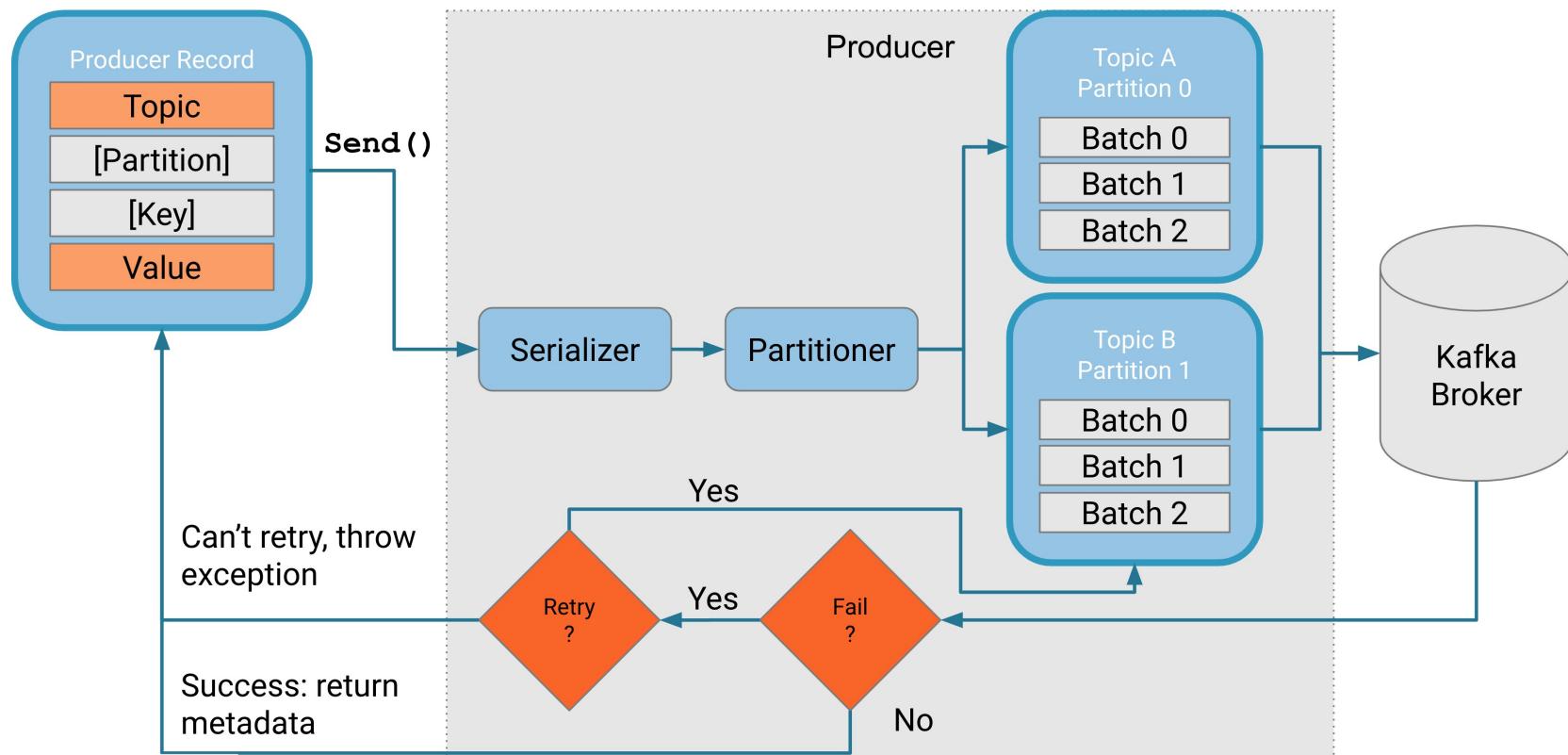


Retention Policy

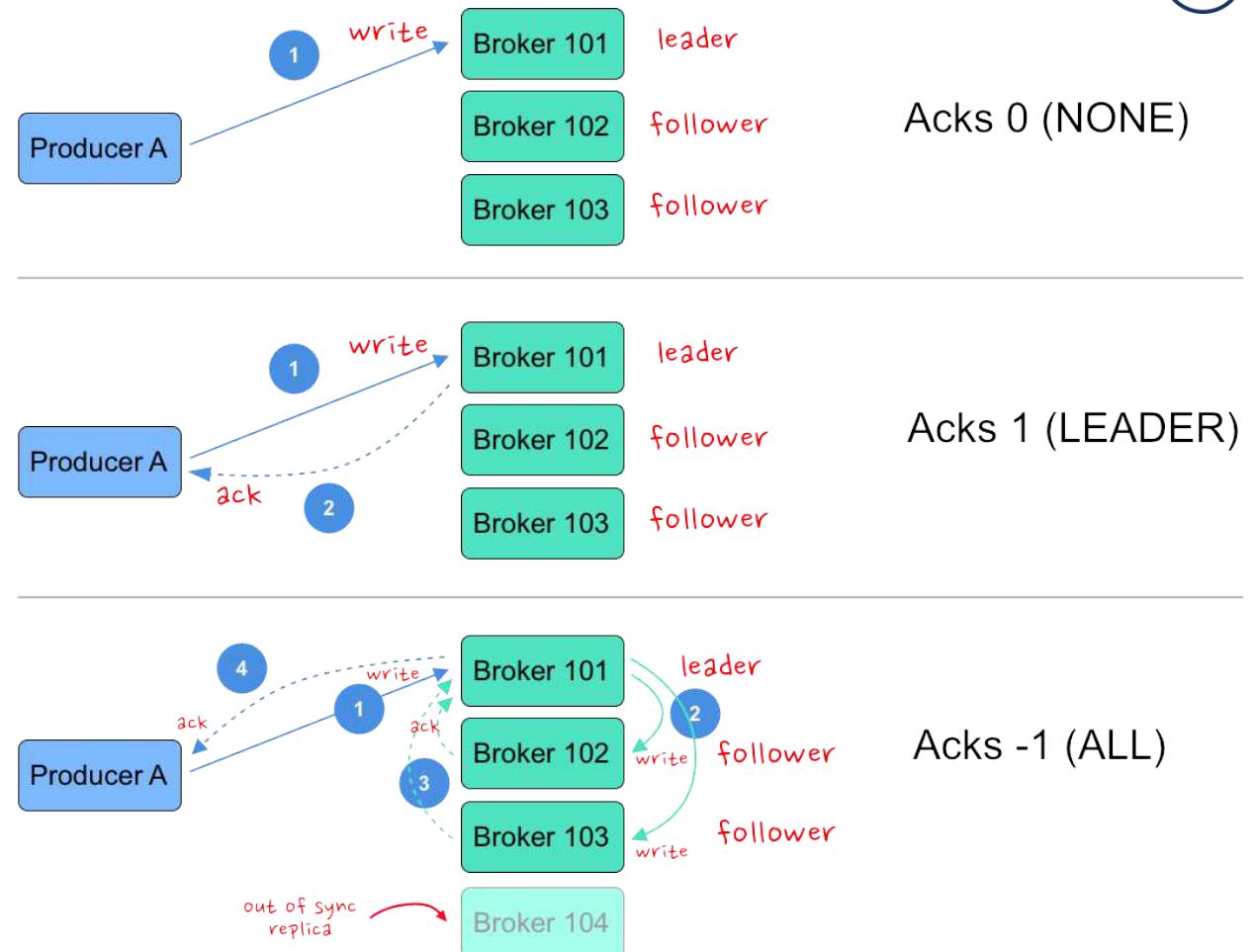
Compacted Topics



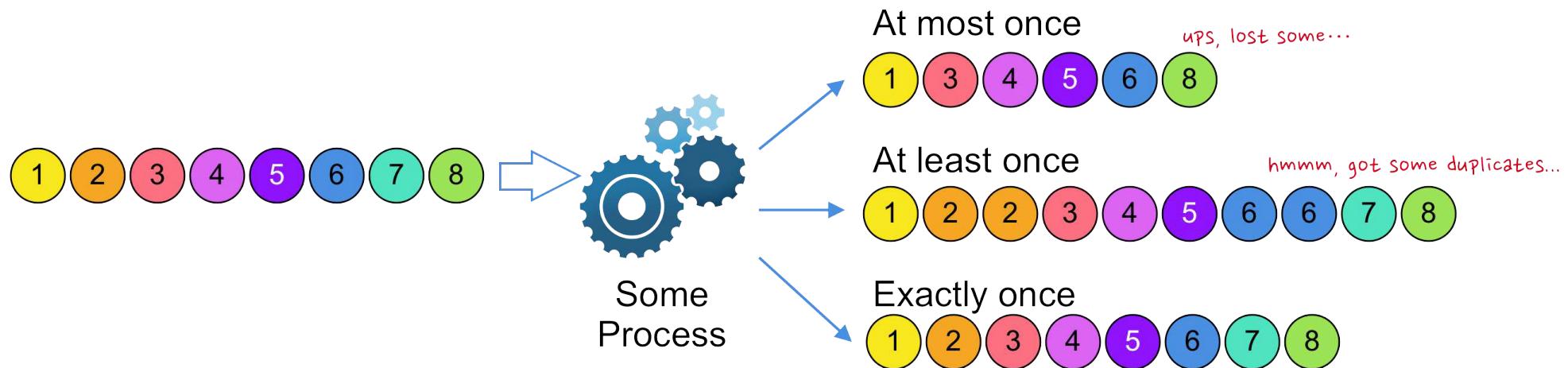
Producer Design



Producer Guarantees



Delivery Guarantees



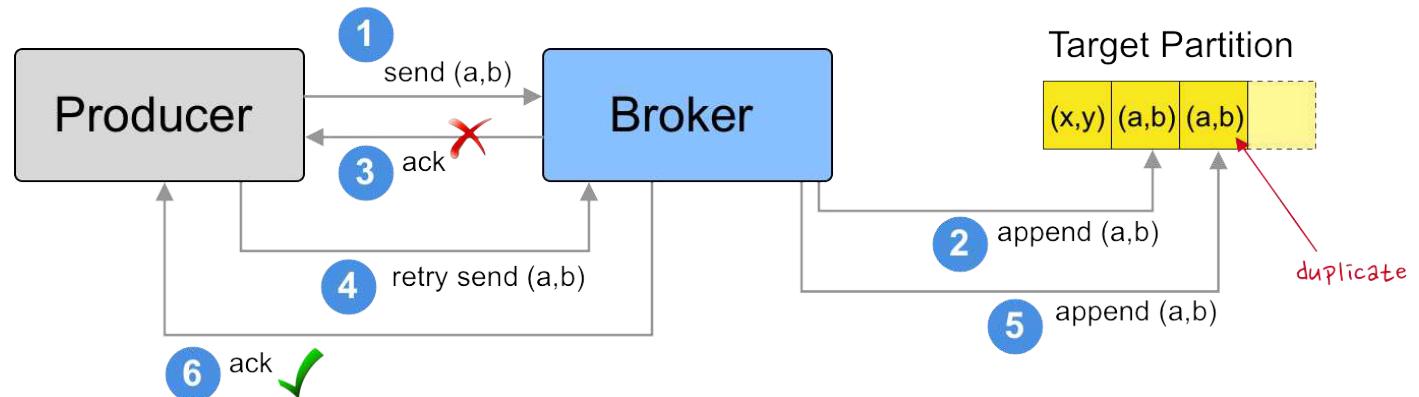
Idempotent Producers



GOOD



BAD



Exactly Once Semantics (EOS)



What is it?

- Strong **transactional guarantees** for Kafka
- Prevents clients from processing duplicate messages
- Handles failures gracefully

Use Cases

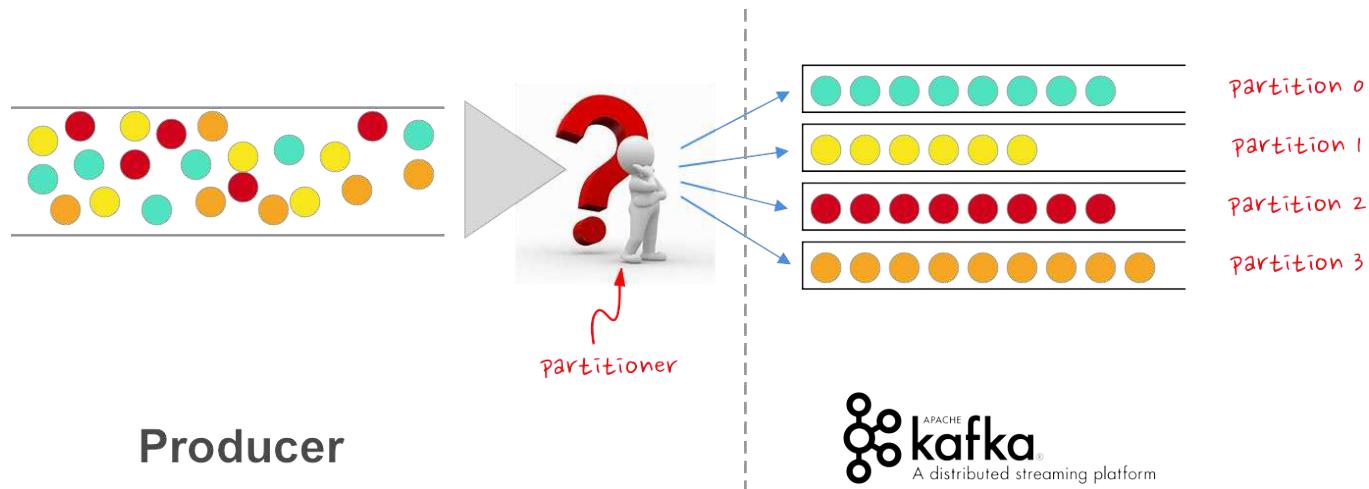
- Tracking ad views
- Processing financial transactions
- Stream processing



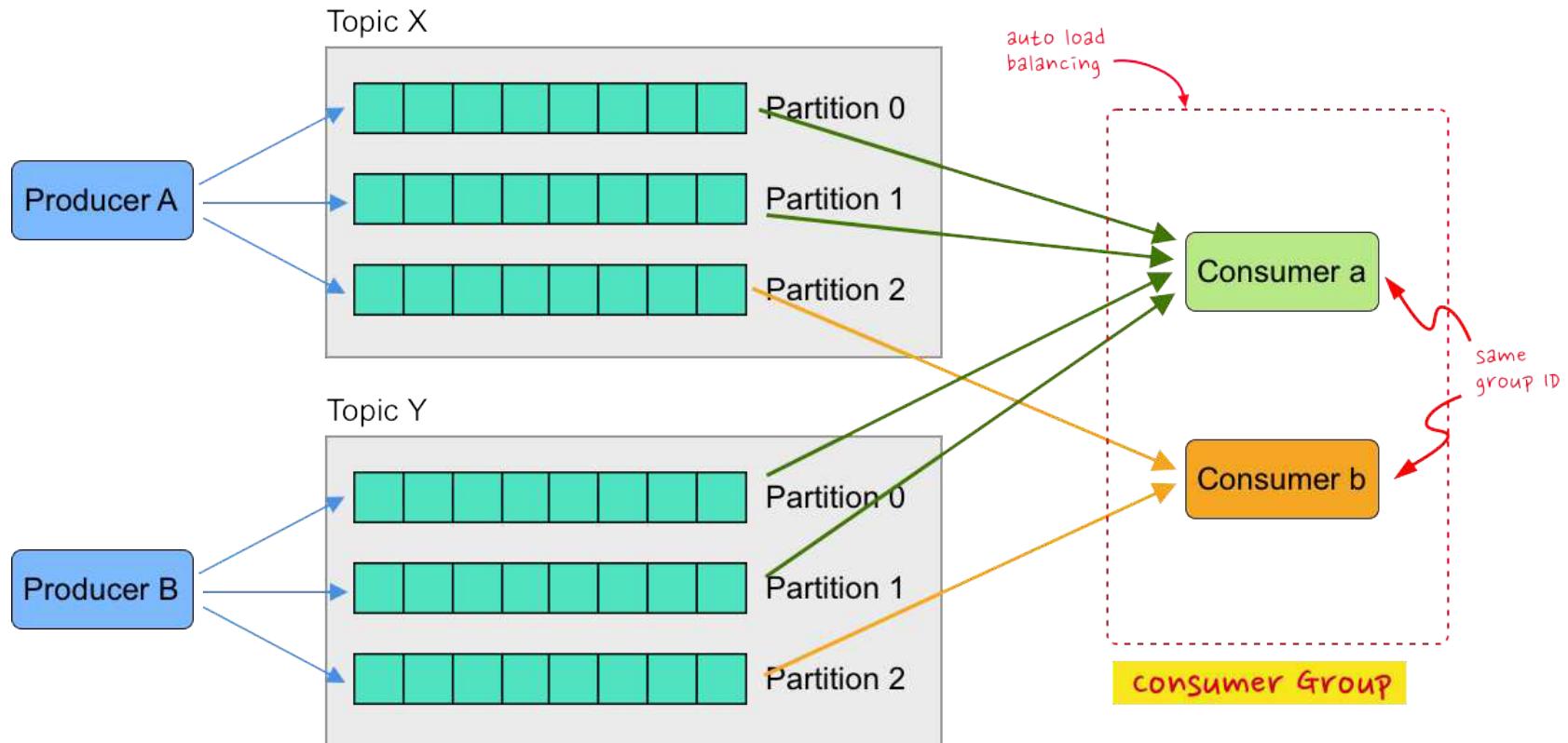
Partitioning Strategies

Why partitioning?

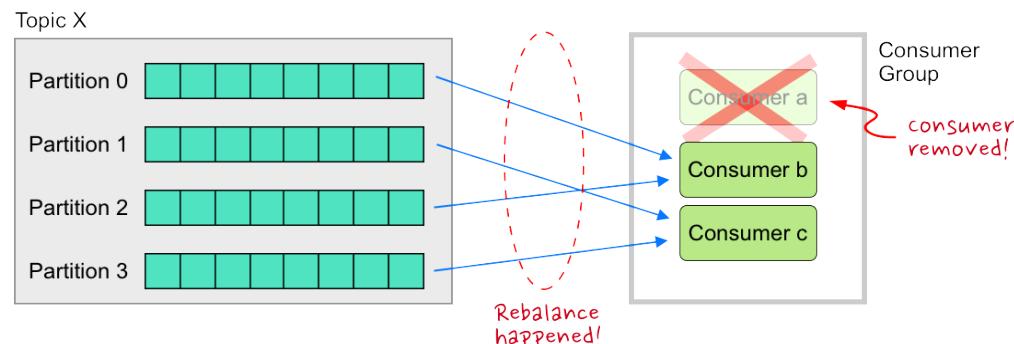
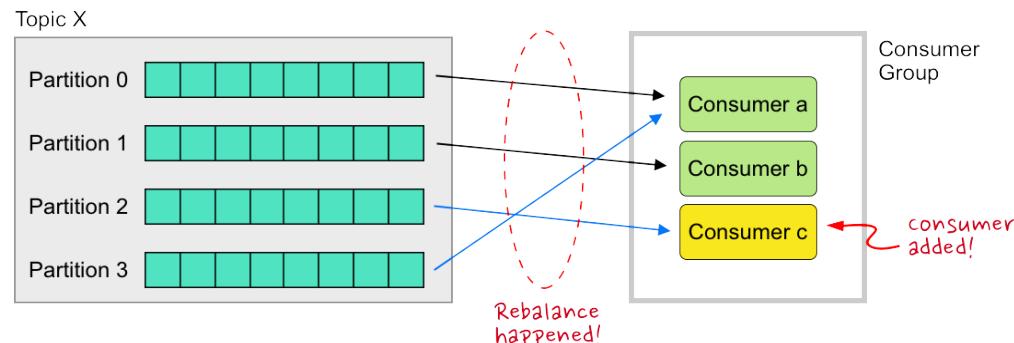
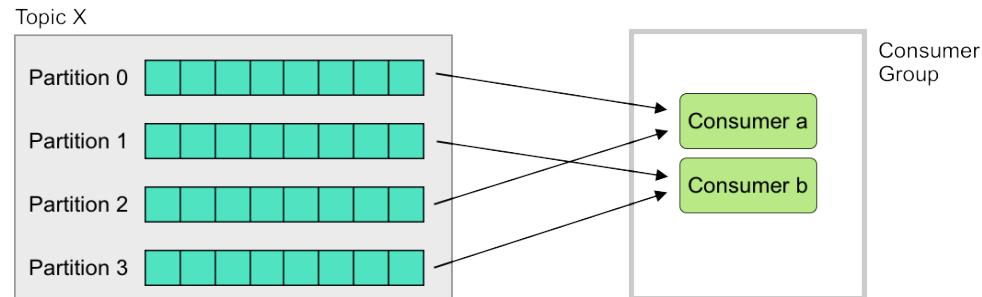
- Consumers need to **aggregate** or **join** by some key
- Consumers need **ordering guarantee**
- Concentrate data for **storage efficiency** and/or **indexing**



Consumer Groups



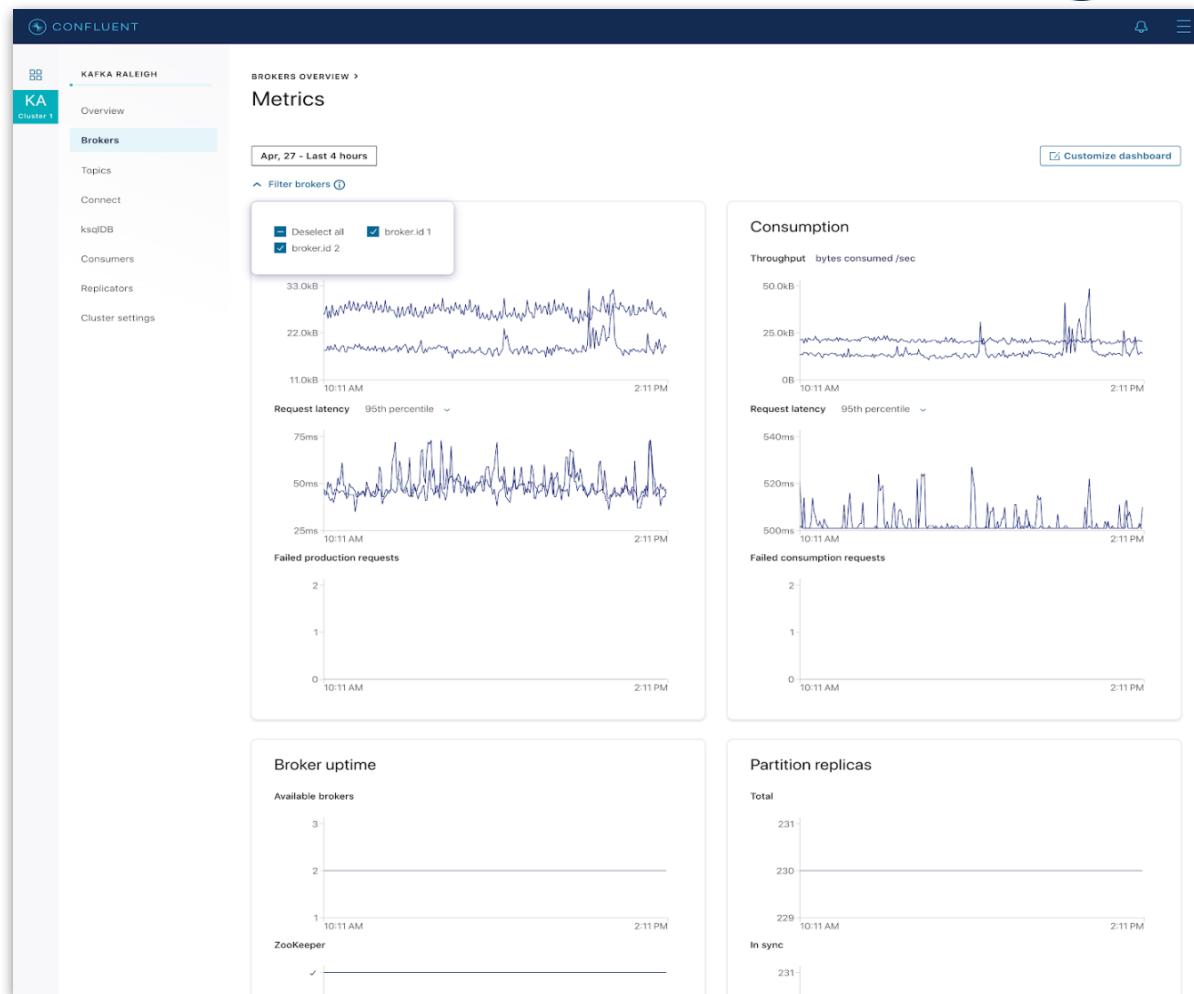
Consumer Rebalances



Troubleshooting



- Confluent Control Center
- Log files
- Special config settings
 - SSL Logging
 - Authorizer debugging



Security Overview



- Kafka supports Encryption in Transit
- Kafka supports Authorization and Authentication
- No Encryption at Rest out of the box
- Clients can be mixed with & without Encryption & Authentication



Client Side Security Features

- Encryption of Data in Transit
- Client Authentication
- Client Authorization



Encryption
in transit
SSL



authn & authz
authn: SASL or SSL
authz: ACLs

Continue your Apache Kafka Education!



- Apache Kafka Administration by Confluent
- Confluent Developer Skills for Building Apache Kafka
- Confluent Stream Processing using Apache Kafka Streams and ksqlDB
- Confluent Advanced Skills for Optimizing Apache Kafka

For more details, see <http://confluent.io/training>



Certifications

Confluent Certified Developer for Apache Kafka

(aligns to Confluent Developer Skills for Building Apache Kafka course)

Confluent Certified Administrator for Apache Kafka

(aligns to Confluent Operations Skills for Apache Kafka)



What you Need to Know

- **Qualifications:** 6-to-9 months hands-on experience
- **Duration:** 90 mins
- **Availability:** Live, online 24/7
- **Cost:** \$150
- **Register online:**
www.confluent.io/certification



Stay in touch!



Confluent Blog
cnfl.io/blog



Community Slack
cnfl.io/slack



Confluent Cloud
cnfl.io/confluent-cloud

