

Kafka UseCase:

Use case

Let's suppose we have a very simple scenario: a service responsible for creating new accounts in a banking system which needs to communicate to another service which is responsible for sending a confirmation email to the user, after the creation.

In a monolith system we would probably have all this logic in the same codebase, in a synchronous way. But in the shiny world of microservices, we have decoupled these responsibilities in two different projects and now we need to let the email service know that a confirmation mail must be sent.

Solution) Kafka is a great fit for many use cases, mostly for website activity tracking, log aggregation, operational metrics, stream processing and, in this post, for messaging.

This platform provides us an elegant way to create a data pipeline where we can connect *producers* (New Account Service) which will be creating new records in the data pipeline and *consumers* (Email Service) which will be listening for new records.

If you come from a RDBMS world you can imagine that a **topic** is a table, a **producer** is someone who will be inserting new data to this table and the **consumer** is an application which will be querying your database to find new entries.

