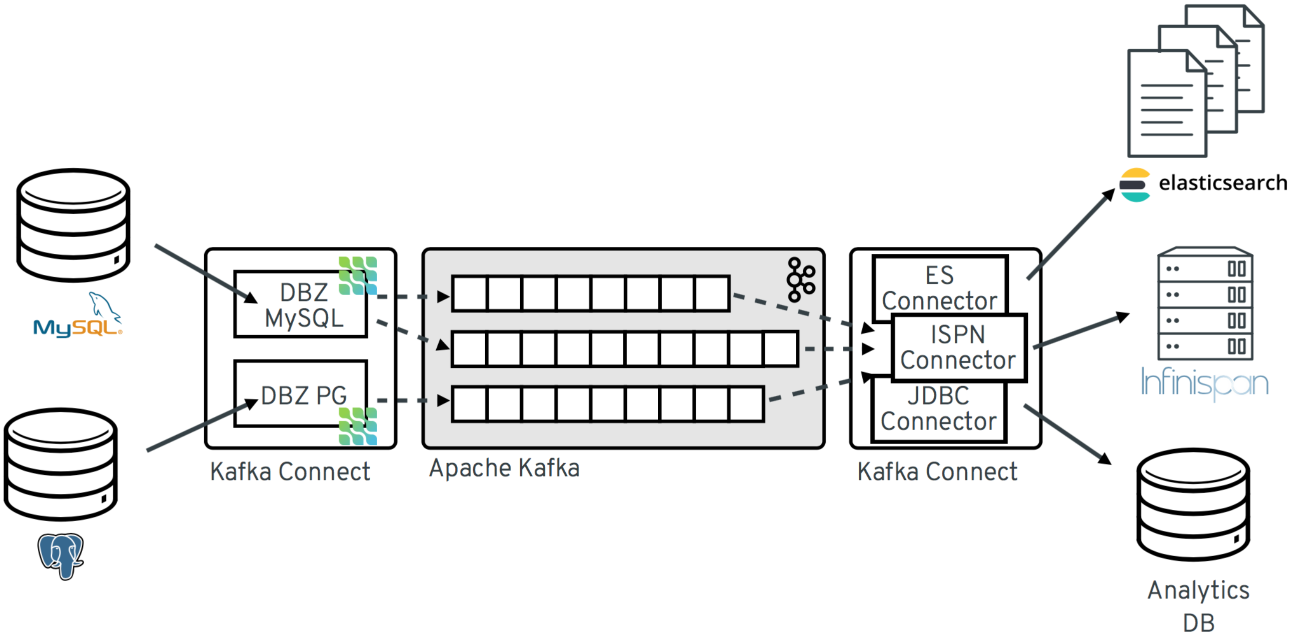
Debezium Architecture

Most commonly, Debezium is deployed via Apache Kafka Connect. Kafka Connect is a framework and runtime for implementing and operating

- source connectors such as Debezium, which ingest data into Kafka and
- sink connectors, which propagate data from Kafka topics into other systems.

The following image shows the architecture of a CDC pipeline based on Debezium:



Kafka Connect is operated as a separate service besides the Kafka broker itself. The Debezium connectors for MySQL and Postgres are deployed to capture the changes out of these two databases. For this purpose, the two connectors establish a connection to the two source databases using a client library for accessing the binlog in case of MySQL and reading from a logical replication stream in case of Postgres.

By default, the changes from one capture table are written to a corresponding Kafka topic. If needed, the topic name can be adjusted with help of Debezium's topic routing SMT, e.g. to use topic names that deviate from the captured tables names or to stream changes from multiple tables into a single topic.

can be used to stream the changes to other systems and databases such as Elasticsearch, data warehouses and analytics systems or caches such as Infinispan. Depending on the chosen sink connector, it may be needed to apply Debezium’s new record state extration SMT, which will only propagate the "after" structure from Debezium’s event envelope to the sink connector.

Embedded Engine

An alternative way for using the Debezium connectors is the embedded engine. In this case, Debezium won’t be run via Kafka Connect, but as a library embedded into your custom Java applications. This can be useful for either consuming change events within your application itself, without the needed for deploying complete Kafka and Kafka Connect clusters, or for streaming changes to alternative messaging brokers such as Amazon Kinesis. You can find an example for the latter in the examples repository.