

Streaming Database Changes with Debezium

redhat.



Gunnar Morling
@gunnarmorling

Agenda

- Use cases for change data streams
- How to create a change data stream?
- Change Data Capture with Kafka (Connect)
- Introducing Debezium
- Demo

Gunnar Morling

- Open source software engineer at Red Hat
 - **Debezium**
 - Hibernate
- **Spec Lead** for Bean Validation 2.0 - *Thursday, 13:35*
- Other projects: ModiTect, MapStruct - *Thursday, 13:10*

✉️ gunnar@hibernate.org

🐦 [@gunnarmorling](https://twitter.com/gunnarmorling)

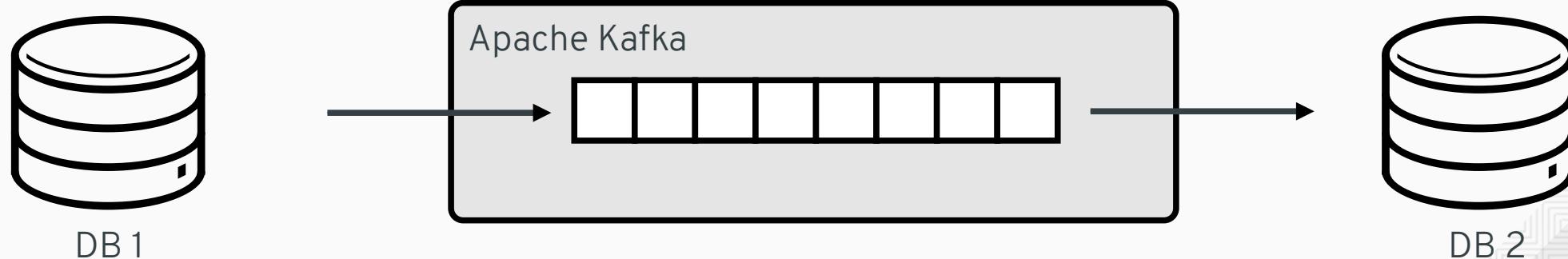
🌐 <http://in.relation.to/gunnar-morling/>



Change Data Capture

What is it about?

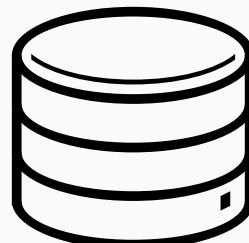
- Get an event stream with all **data and schema changes** in your DB



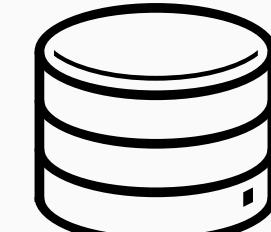
CDC Use Cases

Data Replication

- Replicate data to other DB
- Feed analytics system or DWH
- Feed data to other teams



DB 1



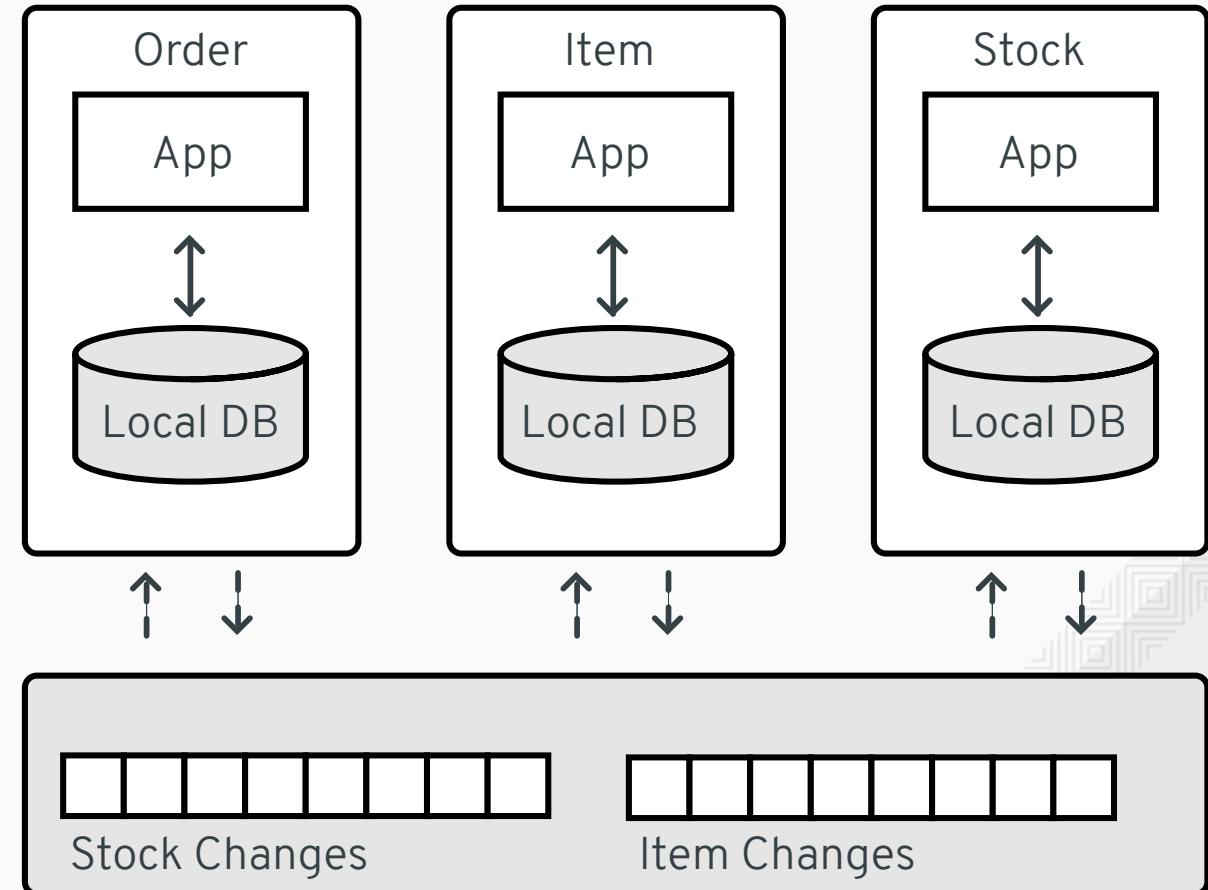
DB 2



CDC Use Cases

Microservice Architectures

- Propagate data between different services **without coupling**
- Each service keeps **optimised views locally**



CDC Use Cases

Others

- Update or invalidate **caches**
- Enable **full-text search** via Elasticsearch, Solr etc.
- Update **CQRS** read models

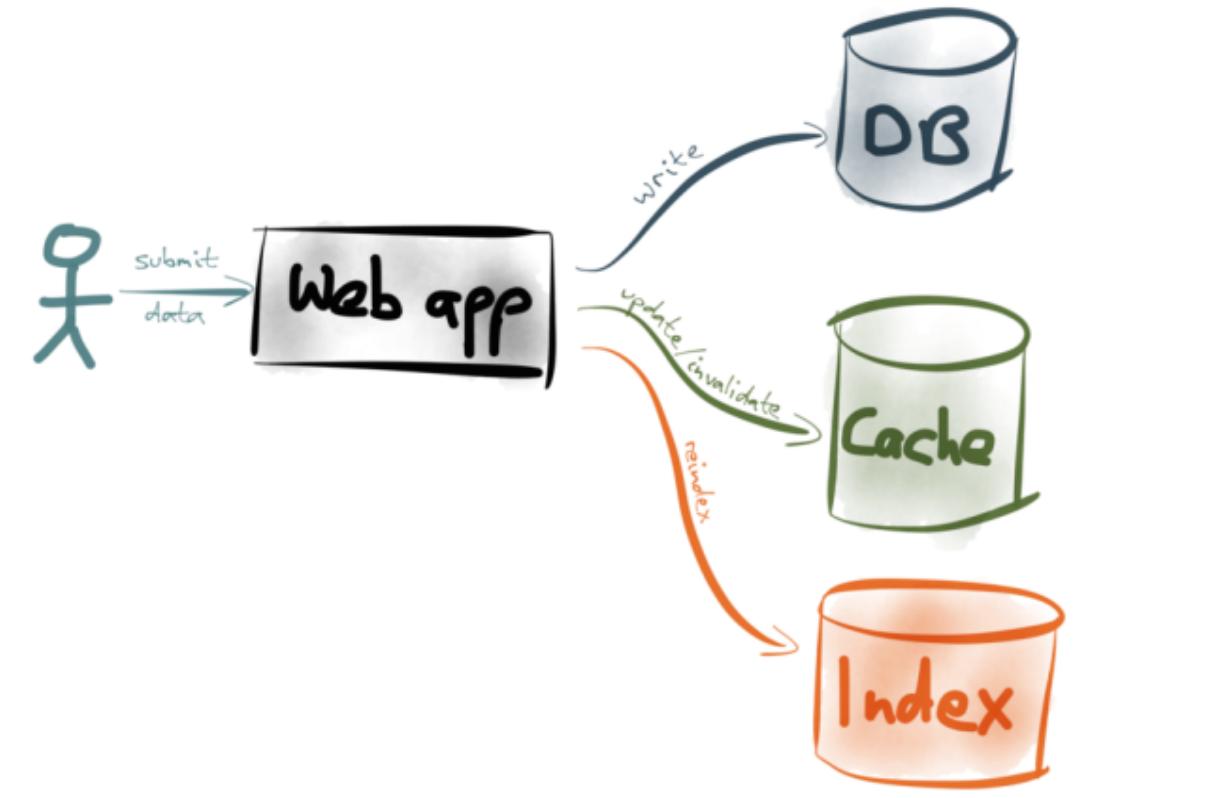
A wide-angle photograph of a mountainous landscape. In the foreground, a deep blue lake stretches across the frame. On either side of the lake are steep, rugged mountains made of light-colored rock. The sky above is a clear, pale blue with a few wispy white clouds.

How to Capture Changes?

How to Capture Changes?

Possible approaches

- Dual writes
 - Failure handling?
 - Prone to **race conditions**
- Polling for changes
 - How to find changed rows?
 - How to handle deleted rows



<https://www.confluent.io/blog/using-logs-to-build-a-solid-data-infrastructure-or-why-dual-writes-are-a-bad-idea/>

How to Capture Changes!

Monitoring the DB

- Reading the database log
- Failures only cause delays but no inconsistencies
- Transparent to upstream applications

Database Logs

A closer look

- Apps write to the DB
- DB records changes in log files, then updates tables
- Logs used for TX recovery, replication etc.
 - MySQL: binlog
 - Postgres: write-ahead log
 - MongoDB op log
- Let's use that for CDC!

Using Apache Kafka as Basis

Well suited for CDC

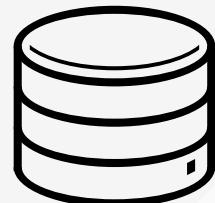
- Messages have a key
- Guaranteed ordering (per partition)
- Pull-based
- Supports compaction
- Scales horizontally



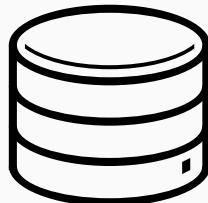
Kafka Connect

- A framework for **source** and **sink** connectors
- **Track offsets**
- Schema support
- Clustering
- Rich **eco-system** of connectors

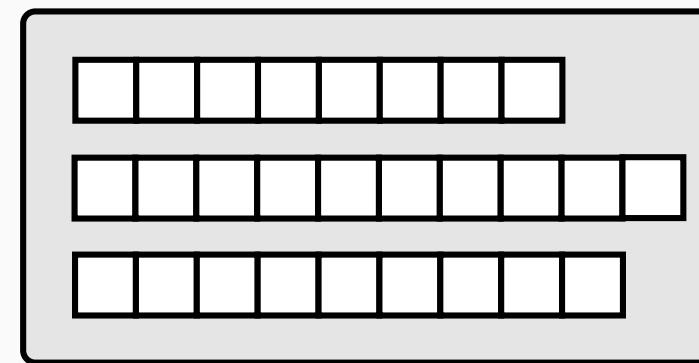
CDC Topology with Kafka Connect



MySQL

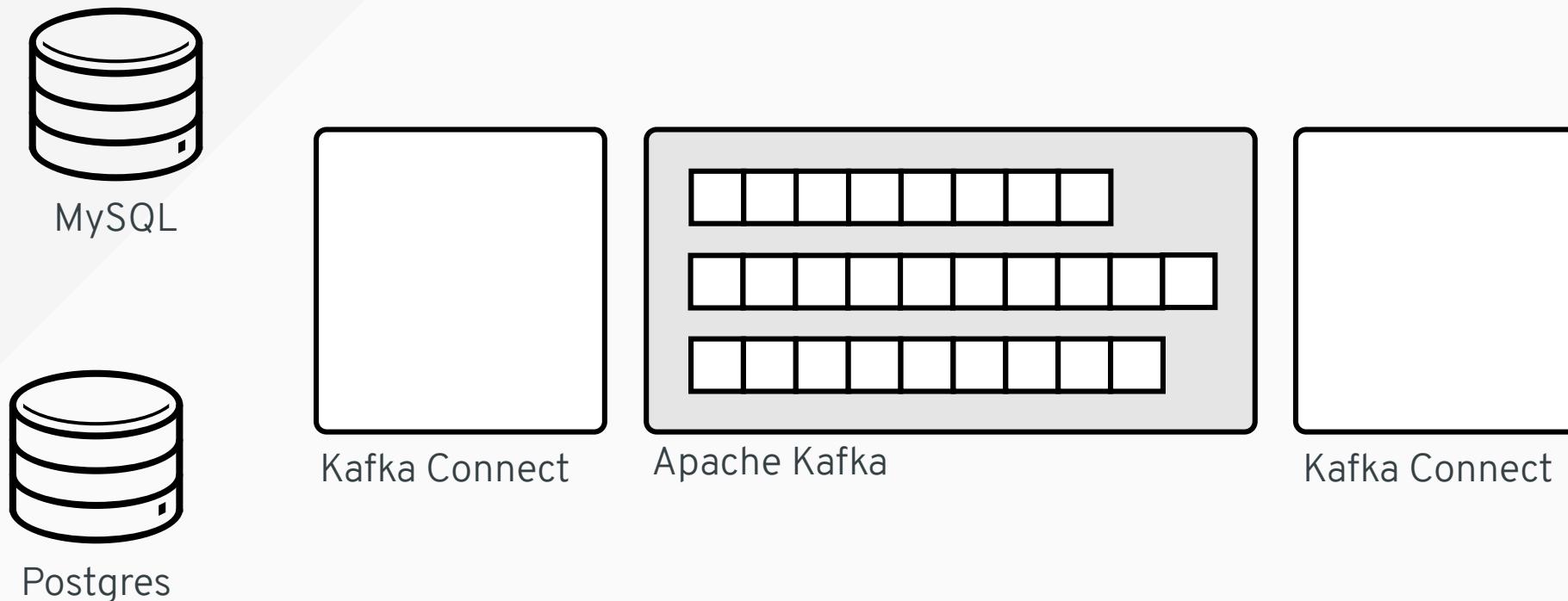


Postgres

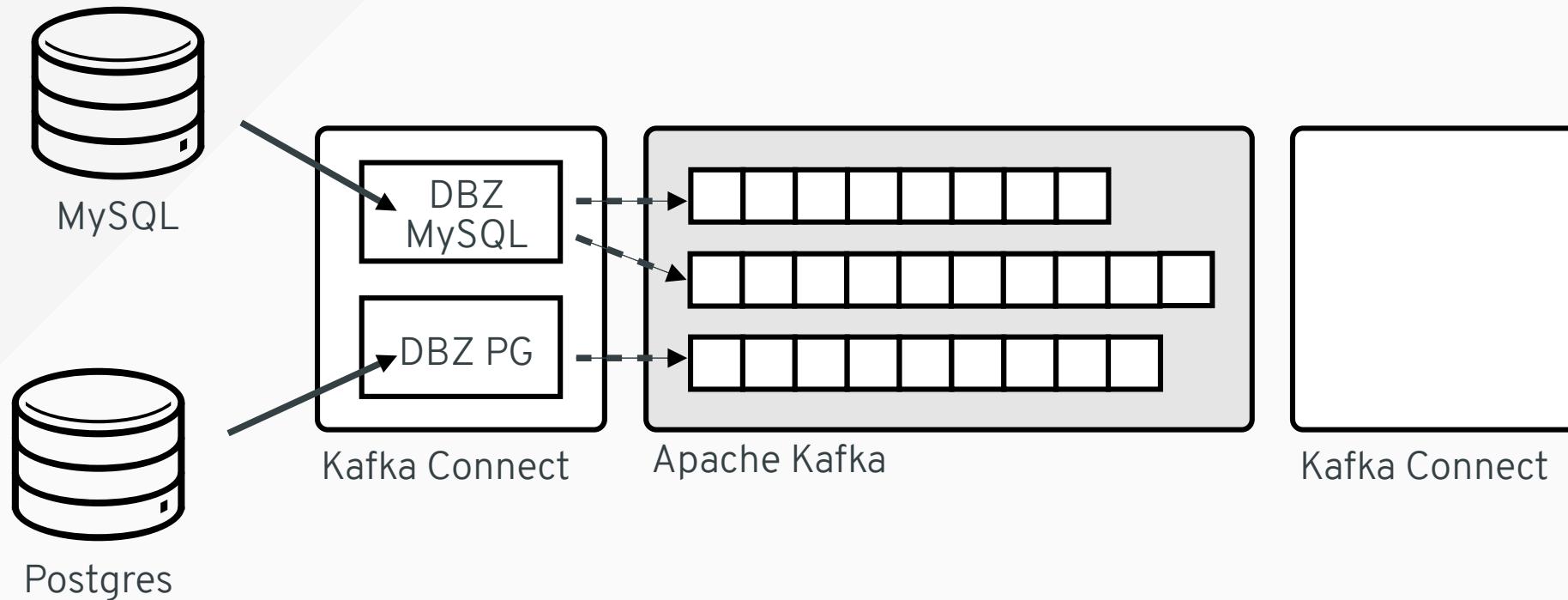


Apache Kafka

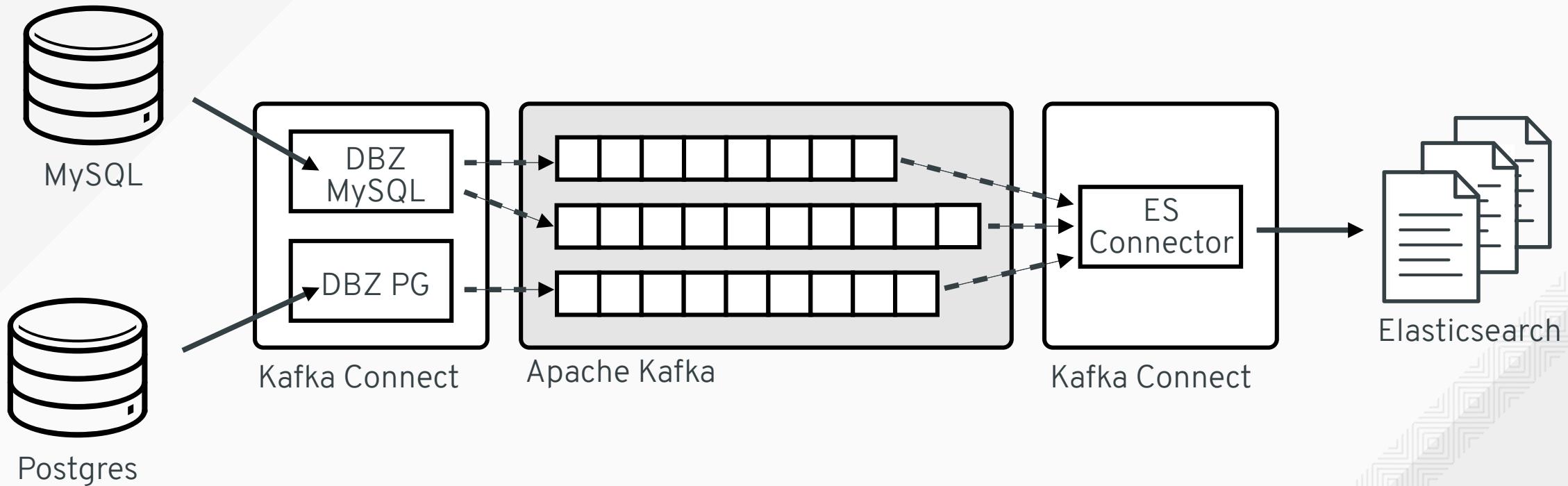
CDC Topology with Kafka Connect



CDC Topology with Kafka Connect



CDC Topology with Kafka Connect



CDC Message Structure

- Key (PK of table) and Value
- Payload: **Before** state, **After** state, **Source** info
- **JSON**
 - Human-readable
 - Verbose
 - Optional: schema in each message
- **Avro**
 - Compact binary representation
 - Using Confluent Schema Registry

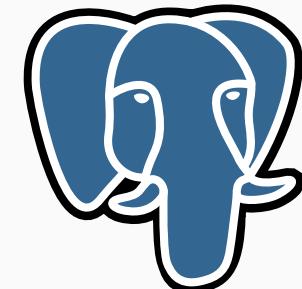
```
{  
  "schema": {  
    ...  
  },  
  "payload": {  
    "before": null,  
    "after": {  
      "id": 1004,  
      "first_name": "Anne",  
      "last_name": "Kretchmar",  
      "email": "annek@noanswer.org"  
    },  
    "source": {  
      "name": "dbserver1",  
      "server_id": 0,  
      "ts_sec": 0,  
      "file": "mysql-bin.000003",  
      "pos": 154,  
      "row": 0,  
      "snapshot": true,  
      "db": "inventory",  
      "table": "customers"  
    },  
    "op": "c",  
    "ts_ms": 1486500577691  
  }  
}
```

Initial Snapshots

- TX logs may have been purged when starting a connector
- Take a snapshot of tables (sending INSERT events)
- Switch to log reading afterwards

Debezium Connectors

- MySQL
- Postgres
- MongoDB (uses patch format)
- Coming next
 - Oracle
 - SQL Server
 - Cassandra?
 - MariaDB?



Debezium Connectors

- Common event format as far as possible
- Common options
 - Filters
 - Snapshotting mode
- Monitoring via JMX
 - Snapshot progress
 - Seconds behind master
 - Last event, last offset etc.

Operating a Connector

Using Kafka Connect REST API

- Set up
- Stopping, e.g. to
 - Reconfigure
 - Update to new version
- Starting up again
 - Take snapshots as needed
- Outage of components may cause delays but no inconsistencies



Demo

Message Transformations

Altering individual messages via SMTs

- Use cases:
 - Extraction
 - Conversion
 - Routing
- Apply on **source** or **sink** side
- Provided by Debezium:
 - Logical table router
 - Event flattening SMT



Demo

Trying It Out Yourself

- Docker images for everything

- Extensive **tutorial**

<http://debezium.io/docs/tutorial/>

- **Docker Compose** files

<https://github.com/debezium/debezium-examples/>

- **OpenShift instructions**

<http://debezium.io/docs/openshift/>

```
1 version: '2'  
2 services:  
3   zookeeper:  
4     image: debezium/zookeeper  
5     ports:  
6       - 2181:2181  
7       - 2888:2888  
8       - 3888:3888  
9   kafka:  
10    image: debezium/kafka:  
11    ports:  
12      - 9092:9092  
13    links:  
14      - zookeeper  
15    environment:  
16      - ZOOKEEPER_CONNECT=zookeeper:2181,zookeeper:2888,zookeeper:3888
```



OPENSHIFT

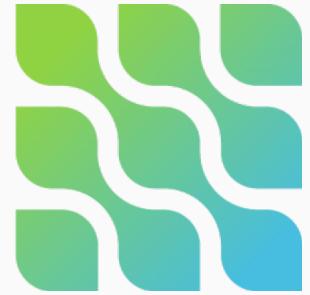
Outlook



- **Debezium 0.7**
 - Based on Kafka 1.0
 - Support for Postgres on Amazon RDS via wal2json
- **Debezium 0.x**
 - More Connectors
 - Support for Infinispan as source and sink
- **Debezium 1.x**
 - Support for event aggregation
 - Building blocks for declarative CQRS support

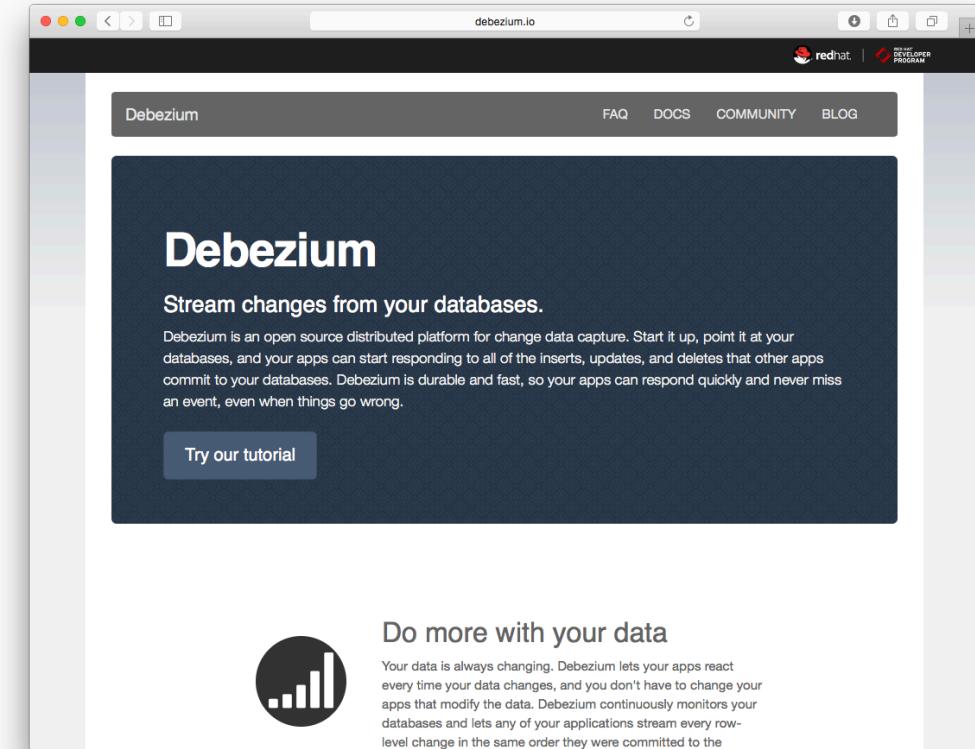
Summary

- Debezium brings **CDC** for growing number of databases
- Transparently set up change data event streams
- **Works reliably** also in case of failures
 - No dual write issues
 - consumers wait until connectors are up again
- Used in production at several sites already
- Everything is **open source** (Apache License v2)
- **Contributions welcome!**



Resources

- Website
<http://debezium.io/>
- Source code, examples, Compose files etc.
<https://github.com/debezium>
- Discussion group
<https://groups.google.com/forum/#!forum/debezium>
- Latest news
 @debezium





redhat®