

Change Data Capture

Suranjan Kumar

Friday 10 June - 9:30am - 10am PT

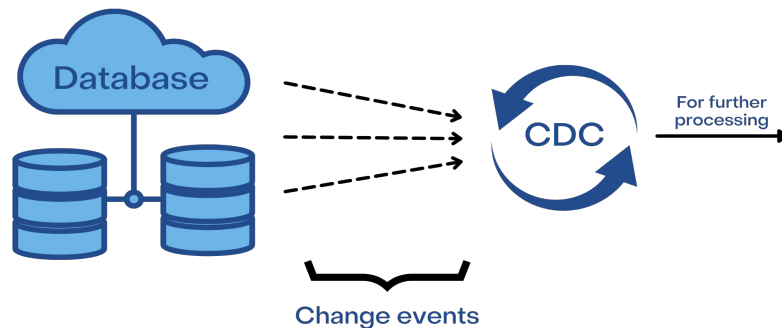


Agenda

- CDC requirements
- Design choices
- Design/Architecture
- User touch points/Demo

CDC

- Process of identifying and tracking the changes made to data in a database
- **Liberates** your data
 - Data Warehouse
 - Analytics dashboards
 - Connecting Microservices
 - Data replication across different data storage
 - Auditing and compliance
 - Full-text search



CDC requirements

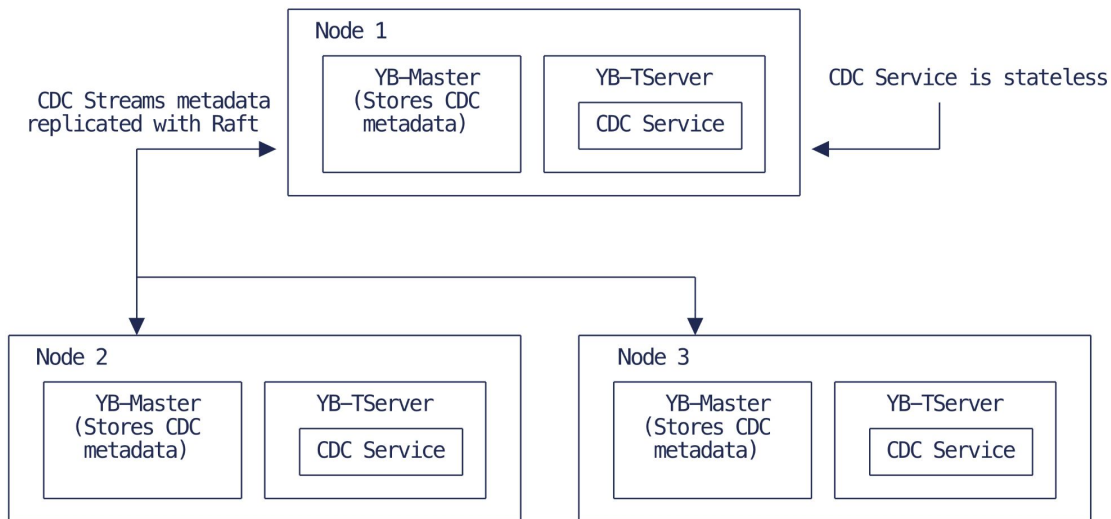
- Resilient to Failures
- Horizontally Scalable
- Supporting variety of use cases
 - Analytics
 - Microservice Architecture
 - Support for multiple formats
 - JSON/AVRO/PROTOBUF
- Multiple Subscribers/Consumers
- Schema Evolution

Design choices

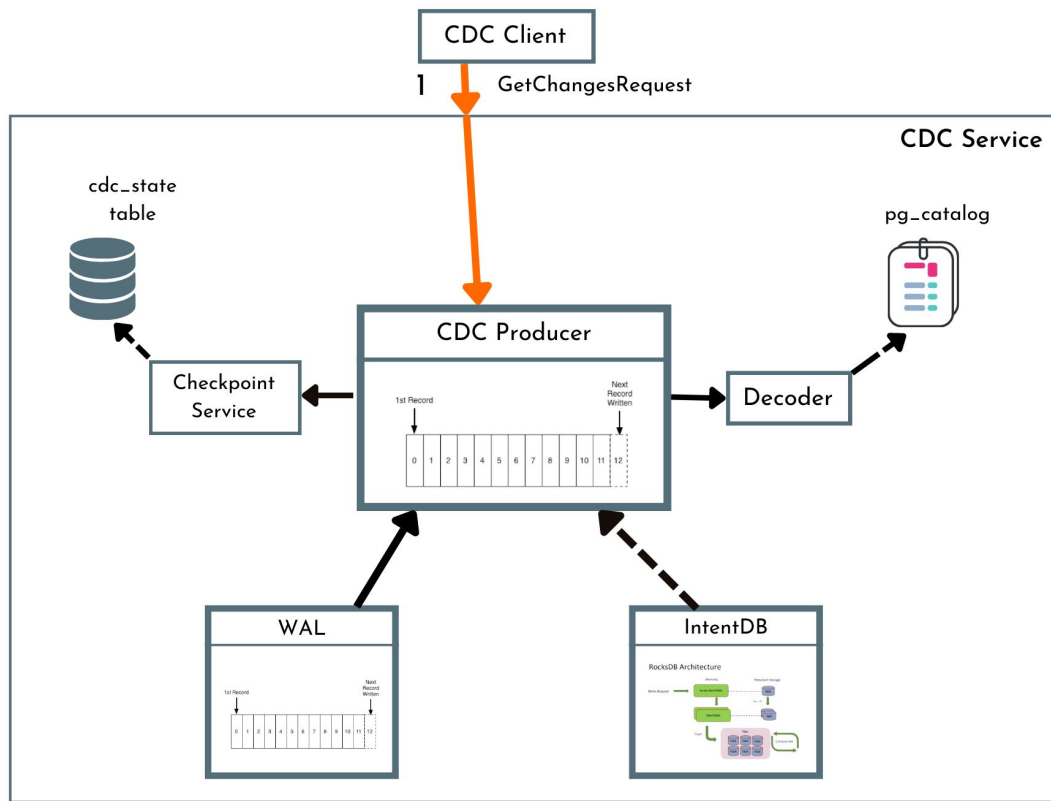
	Options Chosen	Options Considered
CDC Capture	Log Reader	Trigger, Query
CDC Delivery	Stream (micro batch)	Transactional, Aggregated(Batch)
Ordering Guarantees	Per-tablet ordered	Total order
Delivery (Normal operating scenarios)	exactly-once	at-least-once, at-most-once
Delivery (Failure scenarios)	at-least-once	exactly-once, at-most-once
Consistency Guarantees	Timeline consistent (row level and shard level)	
CDC Format	JSON, AVRO (In works) 	pg_output, wal2json
Initial Snapshot	Automatic	Manual
Schema Discovery/Evolution	Automatic	
Retention	Time based, Disk-Size based	

CDC Architecture - process model

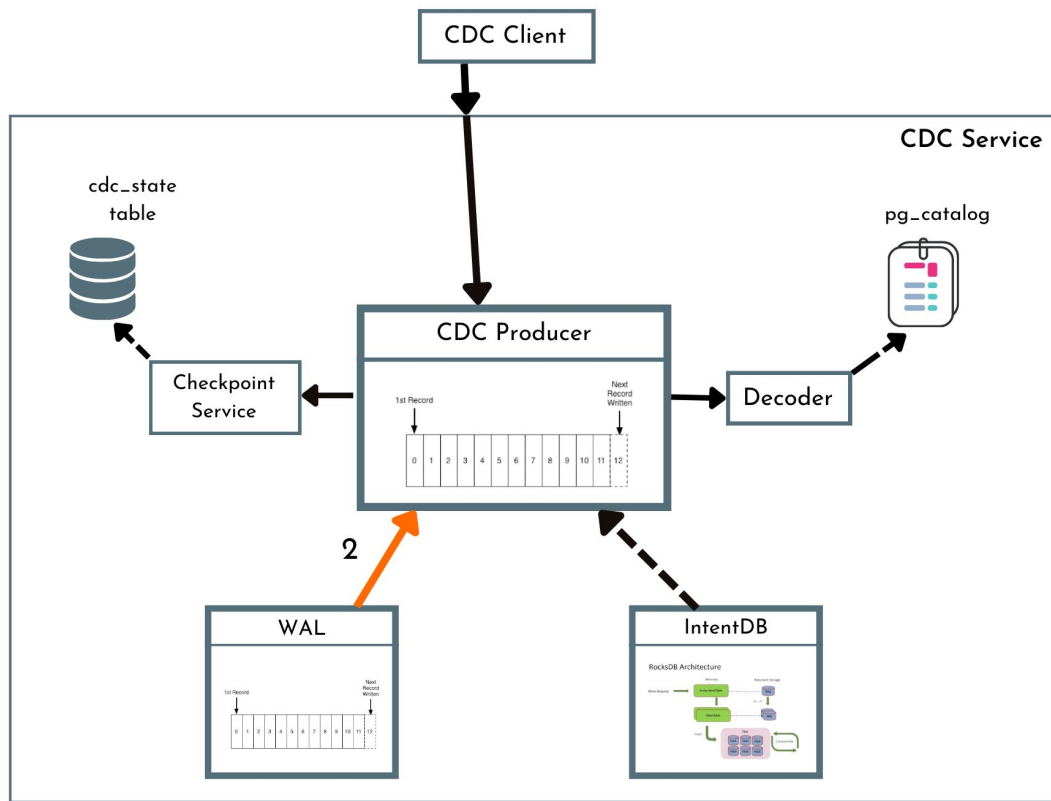
- CDC Administration/Metadata
- CDC Service
- CDC Client/Connector



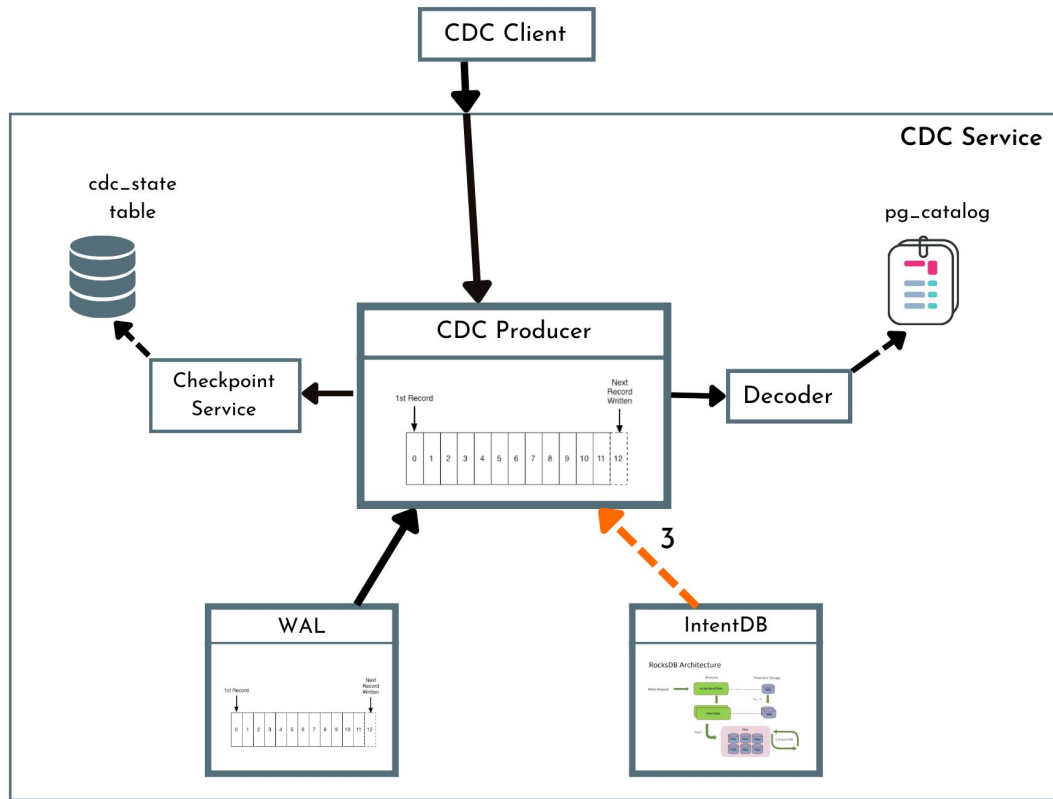
CDC Architecture - CDCService



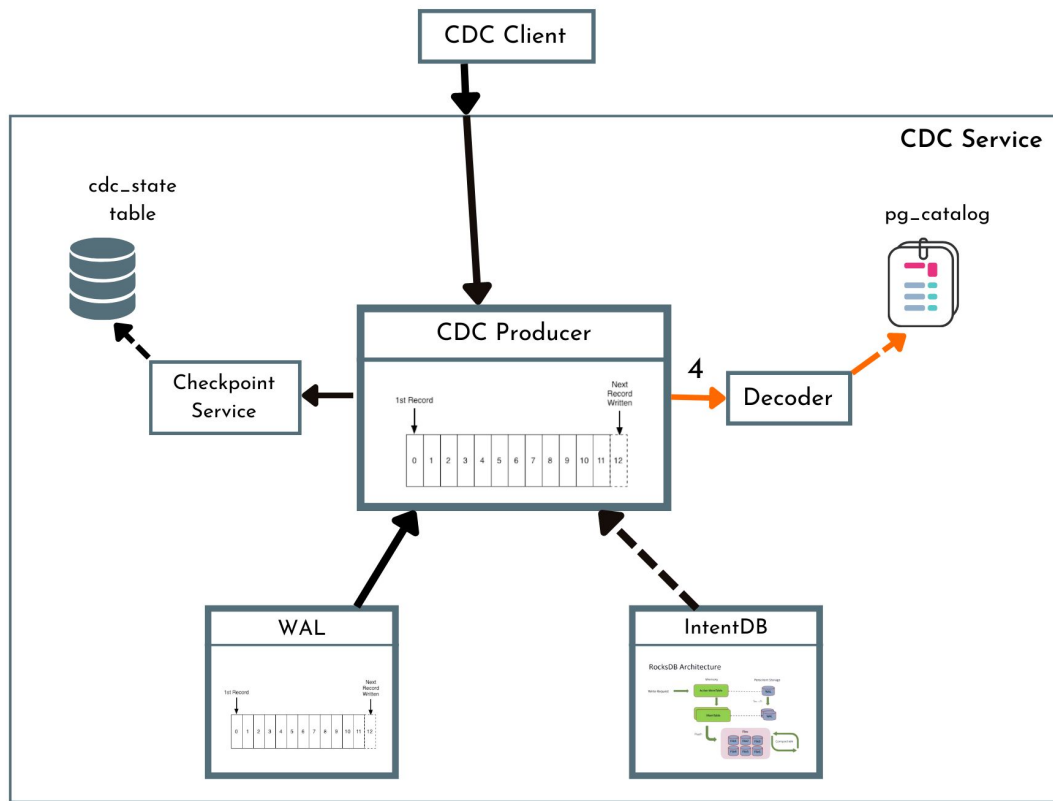
CDC Service architecture



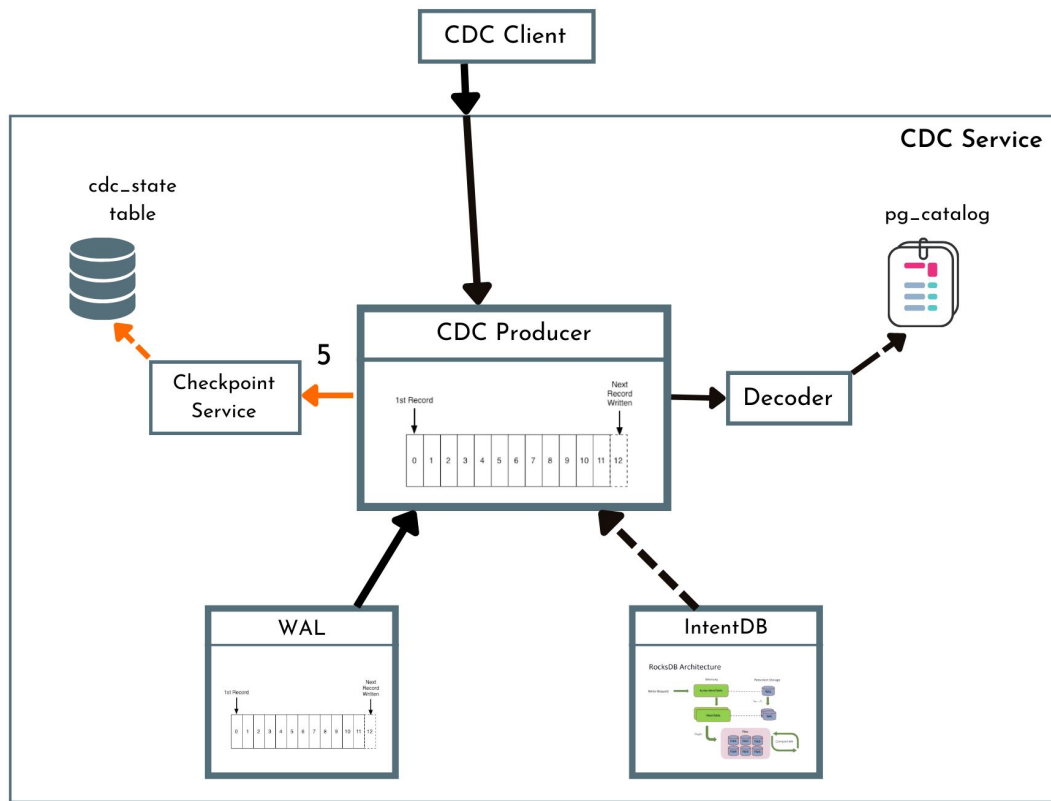
CDC Service architecture



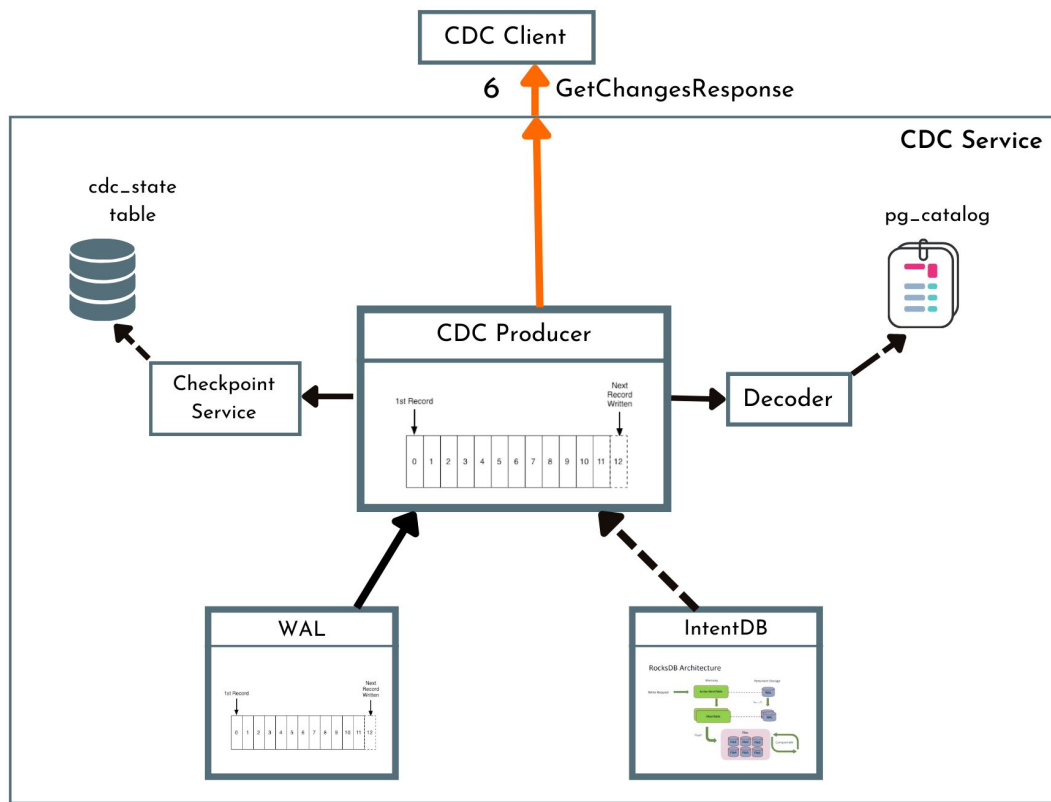
CDC Service architecture



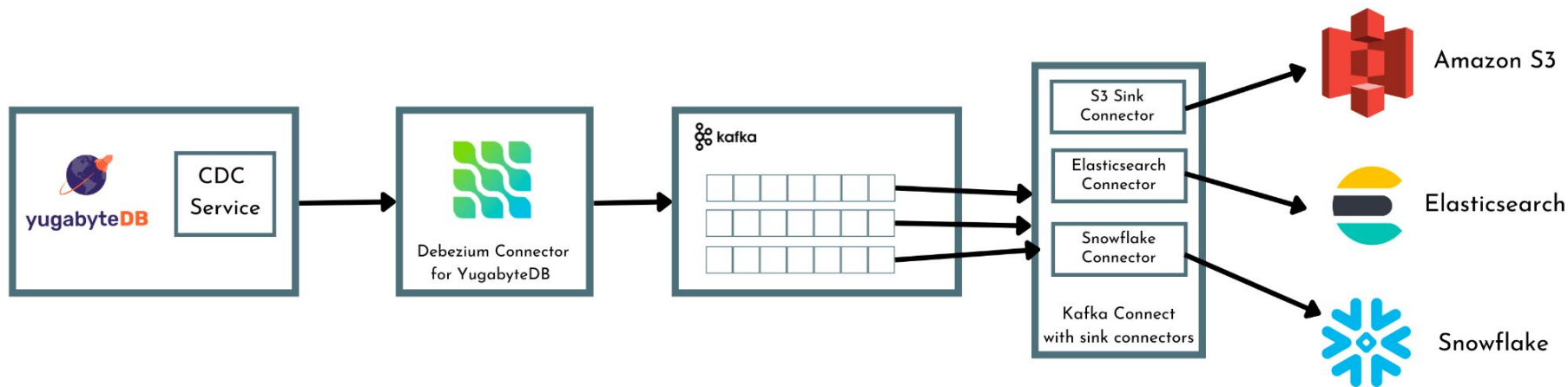
CDC Service architecture



CDC Service architecture

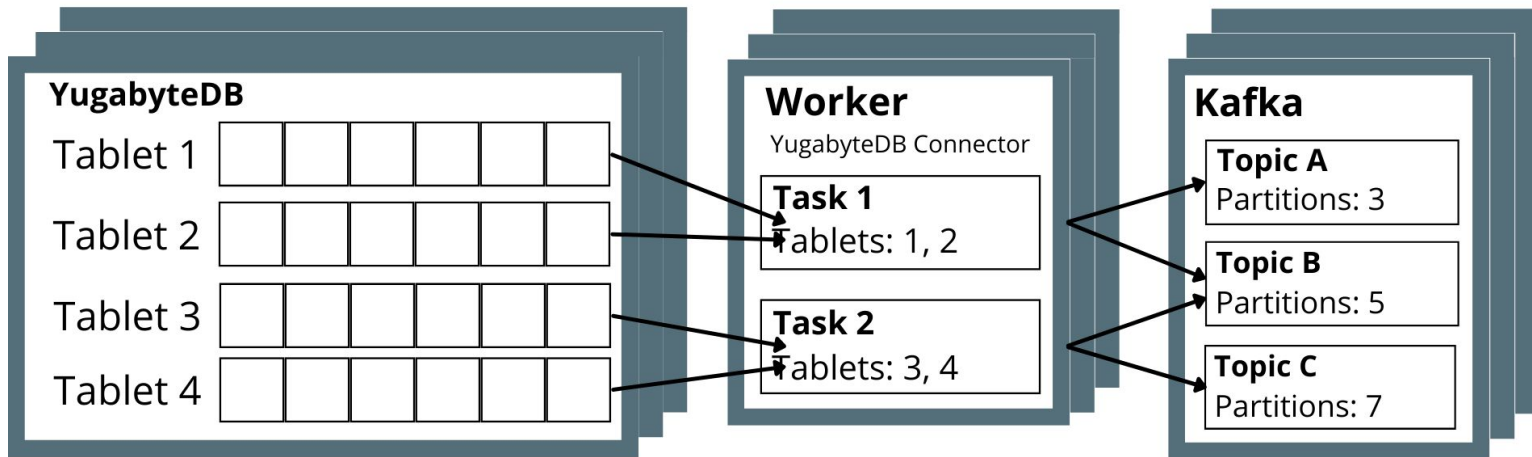


CDC Architecture (CDC pipeline)



CDC Architecture (Client/Connector)

- Scalable at each stage of the pipeline



CDC Administration

```
yb-admin create_change_data_stream <namespace> [change_structure <change-structure>]
```

```
yb-admin get_change_data_stream_info <db_stream_id>
```

```
yb-admin delete_change_data_stream <db_stream_id>
```

```
yb-admin list_change_data_streams [namespace]
```

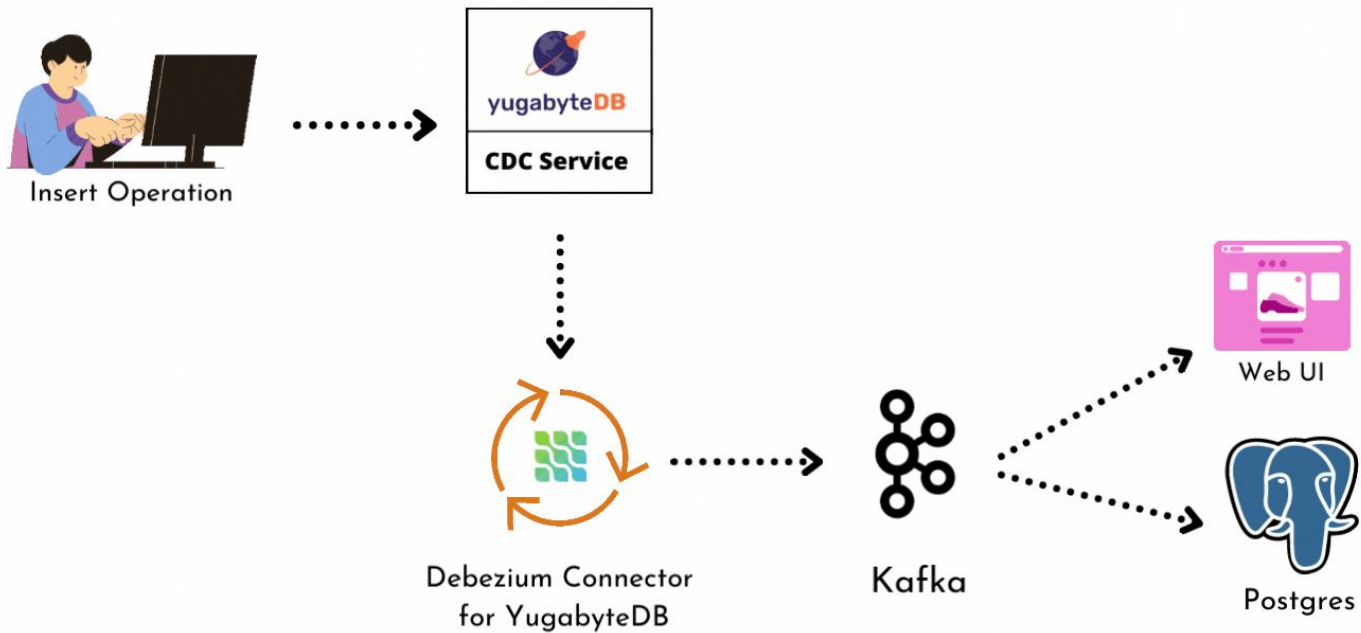
Connector Configuration

- Deploy the connector configuration

```
curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json" localhost:8083/connectors/ -d
```

```
{  
  "name": "ybconnector",  
  "config": {  
    "connector.class": "io.debezium.connector.yugabytedb.YugabyteDBConnector",  
    "database.hostname": "127.0.0.1",  
    "database.port": "5433",  
    "database.master.addresses": "127.0.0.1:7100,127.0.0.2:7100",  
    "database.streamid": "d540f5e4890c4d3b812933cbfd703ed3",  
    "database.user": "yugabyte",  
    "database.password": "yugabyte",  
    "database.dbname": "yugabyte",  
    "database.server.name": "dbserver1",  
    "table.include.list": "public.test1,public.test2"  
  }  
}
```

Demo



CDC - Real world use case

Use Case 1 (Zero code Setup)

- YugabyteDB to MySQL
- 100+ Tables
- 25 nodes cluster across 3 regions (RF=5)
- 1 billion rows
- 1K+ Business TPS
 - ~10K rows per sec

Use Case 2 (Zero code Setup)

- 200m+ rows trade data streamed daily
- Sink through Confluent Cloud
 - ElasticSearch
 - Snowflake

Future work

- Managed Service
- Support for YCQL
- Support for UDT in YSQL
- Support for Tablet Splitting

References

- <https://docs.yugabyte.com/preview/explore/change-data-capture/#yb-admin-commands>
- <https://docs.yugabyte.com/preview/explore/change-data-capture/debezium-connector-yugabytedb/>
- <https://docs.yugabyte.com/preview/integrations/cdc/debezium/>
- <https://debezium.io/>



Thank You

Join us on Slack: [#yftt channel](https://yugabyte.com/slack)

Star us on Github: github.com/yugabyte/yugabyte-db



YFTT
YUGABYTEDB
FRIDAY
TECH TALKS

