debezium

Debezium

CDC - CHANGE DATA CAPTURE

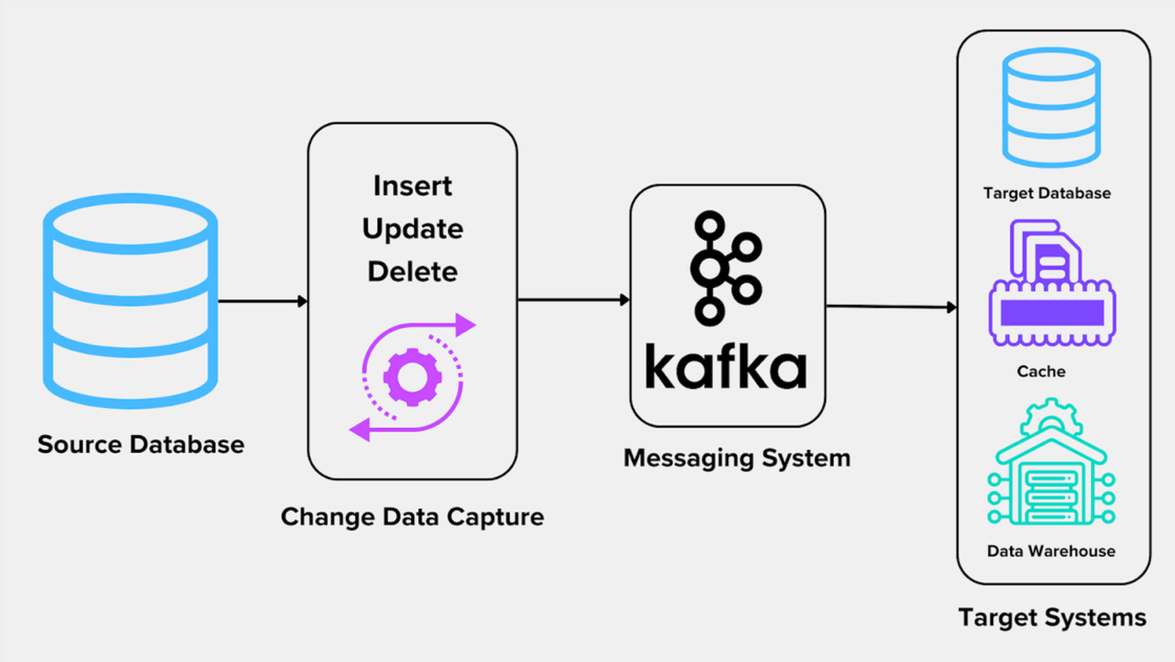
Luka Veličković 2101

S A D R Ž A J

ŠTA JE CDC

debezium

- Tehnika koja omogućava praćenje promena nad podacima u bazi podataka
- Najčešće podrazumeva da se promene nad podacima objavljuju korišćenjem nekog sistema za razmenu poruka (najčešće Kafka)

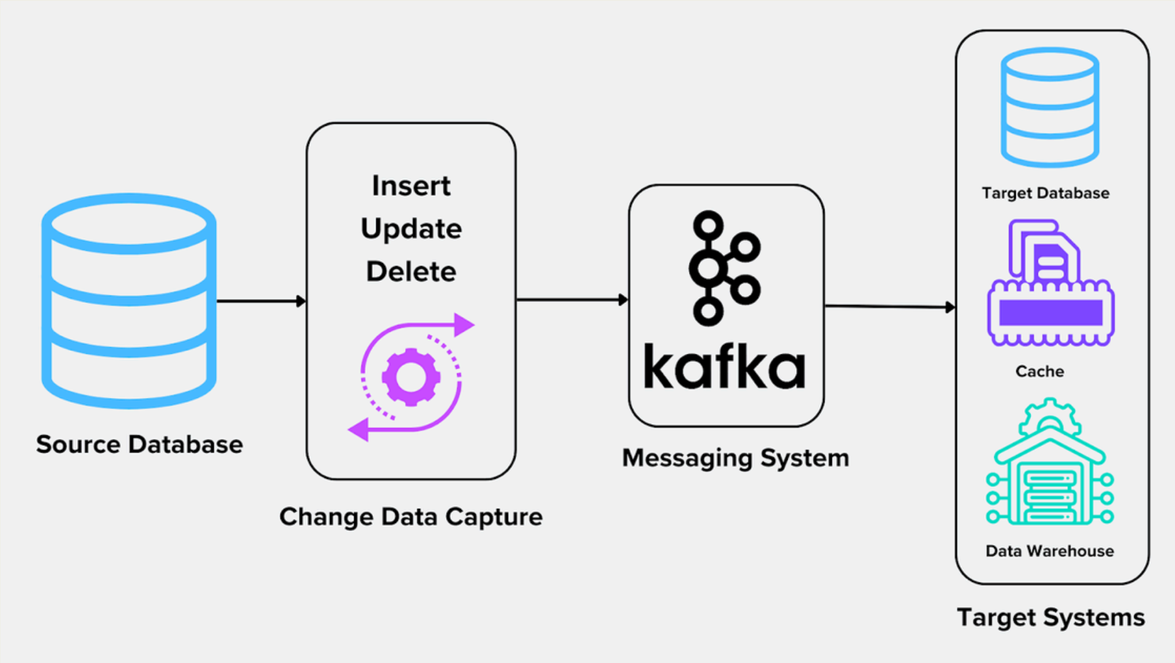


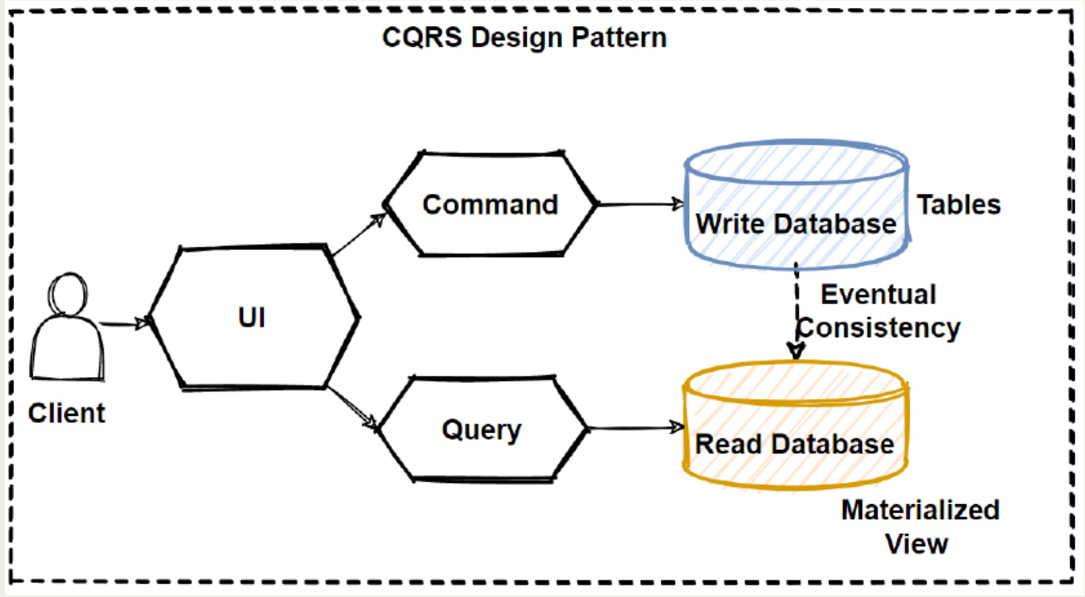
TIPIČNI CDC USE CASES

debezium

- Replikacija podataka
- Upis u veći broj baza podataka
- Održavanje keš koherencije
- CQRS pattern

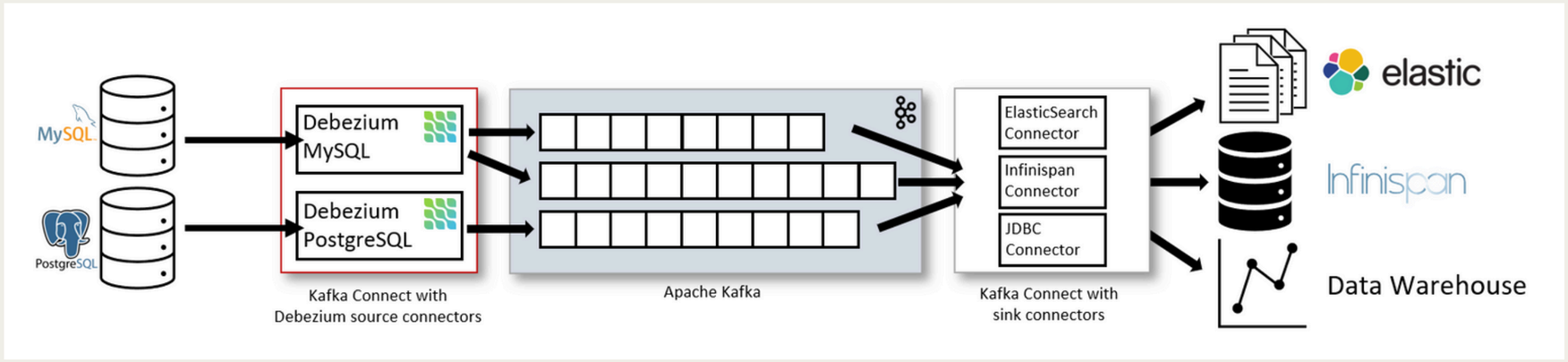
} EVENTUAL CONSISTENCY





ŠTA JE DEBEZIUM

- Open Source platforma koja se koristi za implementaciju CDC-a
- Koristi Apache Kafku
- Koristi KafkaConnect
- Source Connectors - povezuje se na izvor i streamuje promene nad podacima
  - Podrška za PostgreSQL, Oracle, MySQL, MariaDB, SQL Server, Db2, Cassandra...
- Sink Connectors - konzumira evente sa Kafke i upisuje u odredište
  - Podrška za bilo koju bazu za koju postoji JDBC plugin, MongoDB



KONKURENTNI PRISTUPI CDC-U

debezium

- Trigger based CDC

```
CREATE TABLE users_cdc (
    cdc_id BIGSERIAL PRIMARY KEY,
    operation CHAR(1), -- C, U, D
    customer_id INT,
    old_data JSONB,
    new_data JSONB,
    changed_at TIMESTAMP DEFAULT now()
);
```

```
CREATE TRIGGER users_cdc_trg
AFTER INSERT OR UPDATE OR DELETE
ON users
FOR EACH ROW
EXECUTE FUNCTION users_cdc_trigger();
```

```
CREATE OR REPLACE FUNCTION users_cdc_trigger()
RETURNS TRIGGER AS $$
BEGIN
    IF TG_OP = 'INSERT' THEN
        INSERT INTO users_cdc (operation, customer_id, old_data, new_data)
        VALUES ('C', NEW.id, NULL, to_jsonb(NEW));
        RETURN NEW;

    ELSIF TG_OP = 'UPDATE' THEN
        INSERT INTO users_cdc (operation, customer_id, old_data, new_data)
        VALUES ('U', NEW.id, to_jsonb(OLD), to_jsonb(NEW));
        RETURN NEW;

    ELSIF TG_OP = 'DELETE' THEN
        INSERT INTO users_cdc (operation, customer_id, old_data, new_data)
        VALUES ('D', OLD.id, to_jsonb(OLD), NULL);
        RETURN OLD;
    END IF;
END;
$$ LANGUAGE plpgsql;
```

KAKO DEBEZIUM IMPLEMENTIRA CDC

debezium

- Log based CDC
- Specijalni fajlovi u koje baza beleži sve promene koje se dešavaju nad podacima
- Baza prvo upisuje promenu u log, a tek onda u samu tabelu
- Osnovna namena - oporavak, A i D iz ACID
  - PostgreSQL - Write Ahead Log (WAL)
  - MySQL - Binary Log (binlog)
  - Oracle - Redo Log
  - SQL Server - Transaction Log

DEBEZIUM SOURCE CONNECTORS- SNAPSHOTTING

debezium

- Radi se nakon uspostavljanja inicijalne konekcije (konfigurabilno)
  - Pokreće se read only transakcija
  - Čitaju se svi podaci iz odgovarajućih tabela
  - Za svaki pročitani red šalje se event na Kafku
- Ad hoc snapshots
  - Paralelni mod - streaming i snapshotting
- Inkrementalni snapshot
  - Korisnik ima mogućnost da pokrene snapshot u bilo kom trenutku

DEBEZIUM SOURCE CONNECTORS- STREAMING

debezium

- Ime topic-a se po default-u formira kao <ime baze>.<ime šeme>.<tabela> (konfigurabilno)
  - Sve promene na jedan topic:

```
"transforms": "Reroute",
"transforms.Reroute.topic.regex": "dbz\\.public\\..*",
"transforms.Reroute.topic.replacement": "CDC_1",
"transforms.Reroute.type": "io.debezium.transforms.ByLogicalTableRouter"
```

- Eventi mogu da se emituju u različitim formatima (JSON, Avro)
  - before - vrednost koju je red imao pre (null, ako je operacija insert)
  - after - nova vrednost reda (null, ako je operacija delete)
  - source - izvor odakle dolazi event
  - op - operacija
    - c - create
    - u - update
    - d - delete
    - r - read (u snapshot modu)
    - t - truncate
    - m - message (postgres periodično u WAL upisuje stvari koje se ne odnose na podatke u bazi, message op se odnosi na ovakvu aktivnost nad WAL-om)
  - ostali meta podaci, kao što je id transakcije koja je izvršila promenu, vreme izvršenja i slično.

DEBEZIUM SOURCE CONNECTORS- STREAMING


```
{
  "before": null,
  "after": {
    "user_id": 1,
    "attribute_name": "height",
    "attribute_value": "182"
  },
  "source": {
    "version": "2.7.3.Final",
    "connector": "postgresql",
    "name": "dbz",
    "ts_ms": 1767813361449,
    "snapshot": "false",
    "db": "postgres",
    "sequence": "[\"25185064\",\"25185120\"]",
    "ts_us": 1767813361449509,
    "ts_ns": 1767813361449509000,
    "schema": "public",
    "table": "user_attributes",
    "txId": 770,
    "lsn": 25185120,
    "xmin": null
  },
  "transaction": null,
  "op": "c",
  "ts_ms": 1767813361911,
  "ts_us": 1767813361911197,
  "ts_ns": 1767813361911197727
}
```



```
{
  "before": {
    "user_id": 1,
    "attribute_name": "height",
    "attribute_value": "182"
  },
  "after": {
    "user_id": 1,
    "attribute_name": "height",
    "attribute_value": "183"
  },
  "source": {
    "version": "2.7.3.Final",
    "connector": "postgresql",
    "name": "dbz",
    "ts_ms": 1767813524088,
    "snapshot": "false",
    "db": "postgres",
    "sequence": "[\"25185520\",\"25186272\"]",
    "ts_us": 1767813524088716,
    "ts_ns": 1767813524088716000,
    "schema": "public",
    "table": "user_attributes",
    "txId": 771,
    "lsn": 25186272,
    "xmin": null
  },
  "transaction": null,
  "op": "u",
  "ts_ms": 1767813524338,
  "ts_us": 1767813524338060,
  "ts_ns": 1767813524338060199
}
```



```
{
  "before": {
    "user_id": 1,
    "attribute_name": "height",
    "attribute_value": "183"
  },
  "after": null,
  "source": {
    "version": "2.7.3.Final",
    "connector": "postgresql",
    "name": "dbz",
    "ts_ms": 1767813555180,
    "snapshot": "false",
    "db": "postgres",
    "sequence": "[\"25186424\",\"25186480\"]",
    "ts_us": 1767813555180271,
    "ts_ns": 1767813555180271000,
    "schema": "public",
    "table": "user_attributes",
    "txId": 772,
    "lsn": 25186480,
    "xmin": null
  },
  "transaction": null,
  "op": "d",
  "ts_ms": 1767813555297,
  "ts_us": 1767813555297637,
  "ts_ns": 1767813555297637410
}
```

DEBEZIUM SOURCE CONNECTORS- STREAMING


```
{
  "before": null,
  "after": {
    "user_id": 1,
    "attribute_name": "height",
    "attribute_value": "182"
  },
  "source": {
    "version": "2.7.3.Final",
    "connector": "postgresql",
    "name": "dbz",
    "ts_ms": 1767813361449,
    "snapshot": "false",
    "db": "postgres",
    "sequence": "[\"25185064\",\"25185120\"]",
    "ts_us": 1767813361449509,
    "ts_ns": 1767813361449509000,
    "schema": "public",
    "table": "user_attributes",
    "txId": 770,
    "lsn": 25185120,
    "xmin": null
  },
  "transaction": null,
  "op": "c",
  "ts_ms": 1767813361911,
  "ts_us": 1767813361911197,
  "ts_ns": 1767813361911197727
}
```



```
{
  "schema": {
    "type": "struct",
    "fields": [
      {
        "type": "struct",
        "fields": [
          {
            "type": "int32",
            "optional": false,
            "field": "user_id"
          },
          {
            "type": "string",
            "optional": false,
            "field": "attribute_name"
          },
          {
            "type": "string",
            "optional": true,
            "field": "attribute_value"
          }
        ],
        "optional": true,
        "name": "CDC_1.Value",
        "field": "before"
      },
      {
        "type": "struct",
        "fields": [
          {
            "type": "int32",
            "optional": false,
            "field": "user_id"
          },
          {
            "type": "string",
            "optional": false,
            "field": "attribute_name"
          },
          {
            "type": "string",
            "optional": true,
            "field": "attribute_value"
          }
        ],
        "optional": true,
        "name": "CDC_1.Value",
        "field": "after"
      },
      {
        "type": "struct",
        "fields": [
          {
            "type": "string",
            "optional": false,
            "field": "version"
          },
          {
            "type": "string",
            "optional": false,
            "field": "connector"
          },
          {
            "type": "string",
            "optional": false,
            "field": "name"
          },
```

DEBEZIUM SINK CONNECTORS

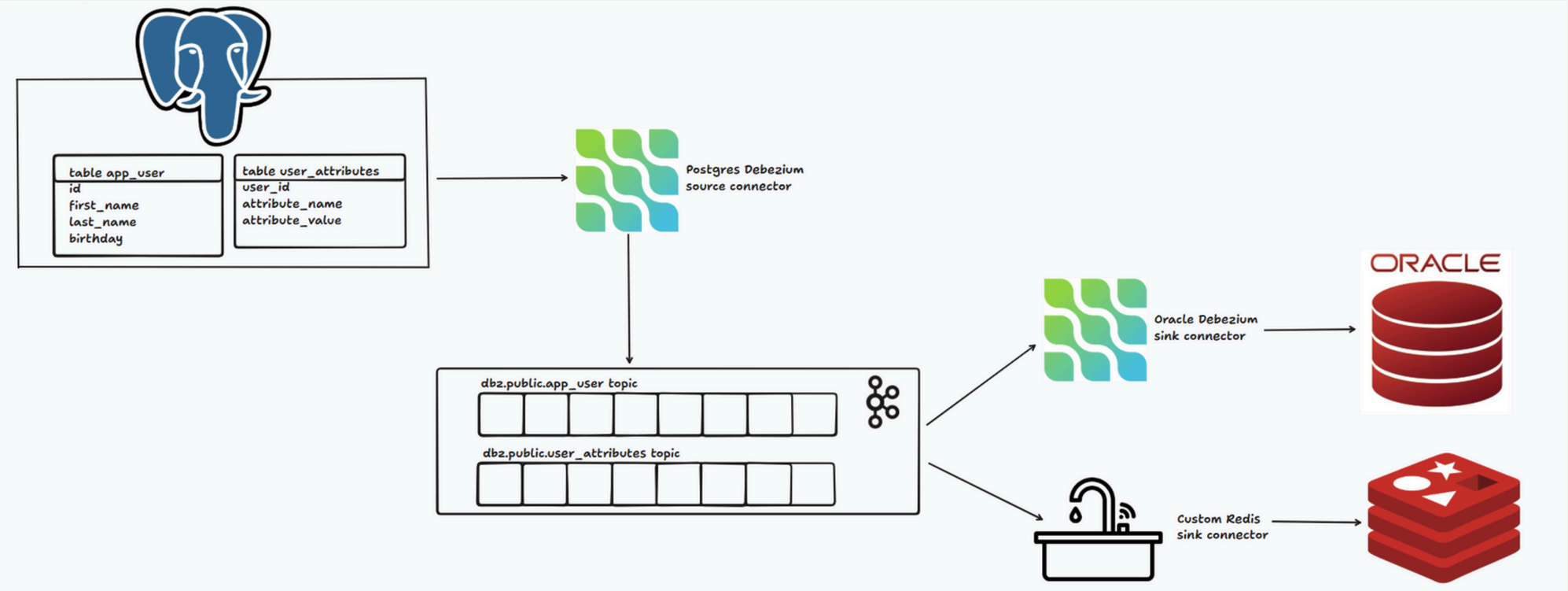
- KafkaConnect sink konektori, koji konzumiraju evente sa Kafka topic-a i upisuju u neki drugi sistem
  - Podrška za bilo koju bazu podataka za koju postoji JDBC driver
  - MongoDB
- JDBC konektori nude:
  - At-least-once delivery - svaki event koji se pričita sa Kafke, biće i obrađen
  - Automatsko mapiranje tipova podataka i tipova kolona
  - Idempotentni upisi - nije podrazumevano podešavanje, moguće je podesiti insert.mode atribut konektora
  - Schema evolution - osnovna podrška za detektovanje promena u šemi izvorne tabele

CUSTOM SINK CONNECTORS

- Generalno, Debezium se najčešće koristi kako bi se promene nad bazom podataka emitovale na Kafku u obliku evenata
- Custom Sink Connectori nisu ništa više do obični Kafka consumer-i koji se povezuju na Kafku i mogu da vrše čitanje evenat-a
- Specijalizovani su, tako da se može izostaviti deo koji se odnosi na šemu
- Neophodno je biti pažljiv, pošto su podaci u eventima predstavljeni korišćenjem custom Debezium tipova podataka (naročito specijalizovani numerički tipovi i vremenski tipovi)

PROJEKAT - DIAGRAM

debezium



PROJEKAT - DOCKER COMPOSE

debezium

- 1 Kafka broker, koristi KRaft protokol
- Kafka UI, na portu 8070
- PostgreSQL
- DebeziumConnect
- Oracle
- Redis

```
lule@ASUS-LAPTOP:~/debezium_swe$ docker compose up -d
[+] Running 7/7
 ✔Network debezium_swe_default   Created                                    0.0s
 ✔Container kafka                Started                                    1.1s
 ✔Container redis                Started                                    1.1s
 ✔Container postgres             Started                                    1.2s
 ✔Container oracle               Started                                    1.1s
 ✔Container kafka-ui             Started                                    1.3s
 ✔Container debezium-connect     Started                                    1.3s
```

PROJEKAT - POSTGRESQL SETUP

debezium

```
create table app_user (
    id int generated always as identity primary key,
    first_name varchar(50) not null,
    last_name varchar(50) not null,
    birthday date not null
);

create table user_attributes (
    user_id int not null references app_user(id),
    attribute_name varchar(50) not null,
    attribute_value varchar(50),
    primary key (user_id, attribute_name)
);

alter table app_user replica identity full;
alter table user_attributes replica identity full;
```

```
create role debezium with login password 'debezium' replication;

grant connect on database postgres to debezium;

grant usage on schema public to debezium;

grant select on table public.app_user, public.user_attributes to debezium;

create publication dbz_publication for table public.app_user, public.user_attributes;

alter publication dbz_publication owner to debezium;

select pg_create_logical_replication_slot(
    'debezium_slot',
    'pgoutput'
);
```

PROJEKAT - DEBEZIUM SOURCE CONNECTOR

debezium

```
{
  "name": "postgres-cdc",
  "config": {
    "connector.class": "io.debezium.connector.postgresql.PostgresConnector",

    "database.hostname": "postgres",
    "database.port": "5432",
    "database.user": "debezium",
    "database.password": "debezium",
    "database.dbname": "postgres",

    "plugin.name": "pgoutput",
    "slot.name": "debezium_slot",
    "publication.name": "dbz_publication",

    "snapshot.mode": "initial",

    "key.converter": "org.apache.kafka.connect.json.JsonConverter",
    "value.converter": "org.apache.kafka.connect.json.JsonConverter",

    "topic.prefix": "dbz"
  }
}
```

```
lule@ASUS-LAPTOP:~/debezium_swe$ curl -X POST \
  -H "Content-Type: application/json" \
  --data @postgres-connector.json \
  http://localhost:8083/connectors | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1140  100   558  100   582  16005  16693 --:--:-- --:--:-- --:--:-- 33529
{
  "name": "postgres-cdc",
  "config": {
    "connector.class": "io.debezium.connector.postgresql.PostgresConnector",
    "database.hostname": "postgres",
    "database.port": "5432",
    "database.user": "debezium",
    "database.password": "debezium",
    "database.dbname": "postgres",
    "plugin.name": "pgoutput",
    "slot.name": "debezium_slot",
    "publication.name": "dbz_publication",
    "snapshot.mode": "initial",
    "key.converter": "org.apache.kafka.connect.json.JsonConverter",
    "value.converter": "org.apache.kafka.connect.json.JsonConverter",
    "topic.prefix": "dbz",
    "name": "postgres-cdc"
  },
  "tasks": [],
  "type": "source"
}
```

PROJEKAT - DEBEZIUM SINK CONNECTOR

debezium

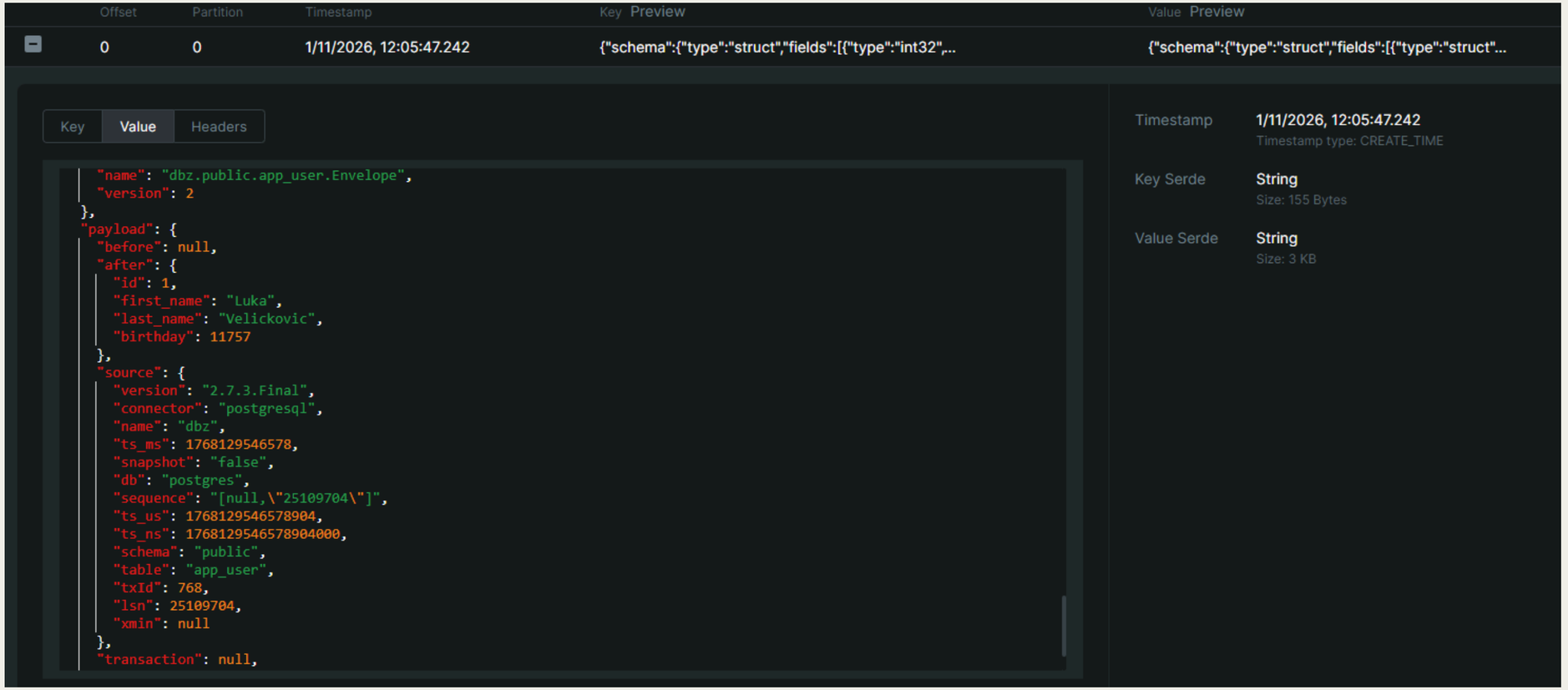
```
{
  "name": "oracle-jdbc-sink",
  "config": {
    "connector.class": "io.debezium.connector.jdbc.JdbcSinkConnector",
    "hibernate.dialect": "org.hibernate.dialect.OracleDialect",
    "tasks.max": "1",

    "connection.url": "jdbc:oracle:thin:@//oracle:1521/FREEPDB1",
    "connection.username": "system",
    "connection.password": "oracle",

    "insert.mode": "upsert",
    "delete.enabled": "true",

    "primary.key.mode": "record_key",

    "schema.evolution": "none",
    "use.time.zone": "UTC",

    "topics": "dbz.public.app_user,dbz.public.user_attributes",

    "collection.name.format": "${source.table}",
    "table.name.format": "${source.table}"
  }
}
```

```
lule@ASUS-LAPTOP:~/debezium_swe$ curl -X POST \
  -H "Content-Type: application/json" \
  --data @oracle-sink.json \
  http://localhost:8083/connectors | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1259  100   616  100   643  19940  20814 --:--:-- --:--:-- --:--:-- 41966
{
  "name": "oracle-jdbc-sink",
  "config": {
    "connector.class": "io.debezium.connector.jdbc.JdbcSinkConnector",
    "hibernate.dialect": "org.hibernate.dialect.OracleDialect",
    "tasks.max": "1",
    "connection.url": "jdbc:oracle:thin:@//oracle:1521/FREEPDB1",
    "connection.username": "system",
    "connection.password": "oracle",
    "insert.mode": "upsert",
    "delete.enabled": "true",
    "primary.key.mode": "record_key",
    "schema.evolution": "none",
    "use.time.zone": "UTC",
    "topics": "dbz.public.app_user,dbz.public.user_attributes",
    "collection.name.format": "${source.table}",
    "table.name.format": "${source.table}",
    "name": "oracle-jdbc-sink"
  },
  "tasks": [],
  "type": "sink"
}
```

PROJEKAT - KAFKA EVENTS

debezium

```
insert into app_user(first_name, last_name, birthday)
        values ('Luka', 'Velickovic', '03.11.2002.');
```



PROJEKAT - KAFKA EVENTS

debezium

```
select * from app_user;
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.086 seconds

	ID	FIRST_NAME	LAST_NAME	BIRTHDAY
1	1	Luka	Velickovic	11-MAR-02

HASH public-app_user:1

104 B Length: 4 TTL: No limit

< 1 min

Show TTL

Field	Value	TTL	
id	1	No Limit	
first_name	Luka	No Limit	
last_name	Velickovic	No Limit	
birthday	11757	No Limit	