**Debezium (Local Docker) with AWS RDS MySQL (WIP…)**

Create MySQL database on RDS. Enable public accessibility, select a backup retention of greater than 0, and create and apply a custom parameter group in which the logbin\_format = ROW. Reboot server and proceed with the steps below (using MySQL version 5.7.22)

**Download Debezium Docker repo**

[**https://github.com/debezium/debezium-examples/tree/master/unwrap-smt**](https://github.com/debezium/debezium-examples/tree/master/unwrap-smt)

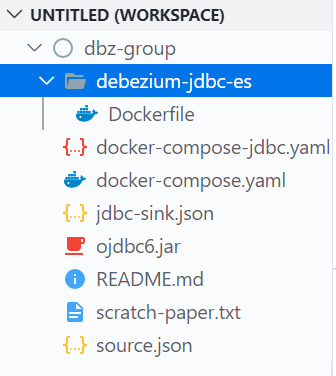
*The whole repo will be installed after running a git clone, but you can just remove all of the unnecessary files. Much of the contents of the Docker images can be removed as well.*

**Download ojdbc6.jar**

[**https://download.oracle.com/otn/utilities\_drivers/jdbc/121010/ojdbc6.jar**](https://download.oracle.com/otn/utilities_drivers/jdbc/121010/ojdbc6.jar)

**from**

[**https://www.oracle.com/database/technologies/jdbc-drivers-12c-downloads.html**](https://www.oracle.com/database/technologies/jdbc-drivers-12c-downloads.html)

**Place ojdbc6.jar in the same directory as your Docker yaml files**

**Modify jdbc-sink.json**

{

    "name": "jdbc-sink",

    "config": {

        "connector.class": "io.confluent.connect.jdbc.JdbcSinkConnector",

        "tasks.max": "1",

        "topics": "topics",

        "connection.url": "jdbc:oracle:thin:@hostname:1521:dbz2",

        "connection.user": "user",

        "connection.password": "password",

        "transforms": "unwrap",

        "transforms.unwrap.type": "io.debezium.transforms.ExtractNewRecordState",

        "transforms.unwrap.drop.tombstones": "false",

        "auto.create": "true",

        "insert.mode": "upsert",

        "delete.enabled": "true",

        "pk.fields": "id",

        "pk.mode": "record\_key"

    }

}

**Modify source.json**

{

    "name": "inventory-connector",

    "config": {

        "connector.class": "io.debezium.connector.mysql.MySqlConnector",

        "tasks.max": "1",

        "database.hostname": "hostname",

        "database.port": "3306",

        "database.user": "user",

        "database.password": "password",

        "database.server.id": "184054",

        "database.server.name": "dbz",

        "table.whitelist": "tables",

        "database.history.kafka.bootstrap.servers": "kafka:9092",

        "database.history.kafka.topic": "schema-changes.dbz",

        "transforms": "route",

        "transforms.route.type": "org.apache.kafka.connect.transforms.RegexRouter",

        "transforms.route.regex": "([^.]+)\\.([^.]+)\\.([^.]+)",

        "transforms.route.replacement": "$3"

    }

}

**Start the application**

export DEBEZIUM\_VERSION=1.0

docker-compose -f docker-compose-jdbc.yaml up

**Place ojdbc6.jar in kafka connect container plugin directory**

docker cp ojdbc6.jar dbz-group\_connect\_1:/kafka/libs

*Assumes you’ve named the folder containing your Docker yamls “dbz-group”*

**Restart kafka connect container such that it loads new oracle jar**

docker restart dbz-group\_connect\_1

*This can also be done in the Docker Desktop dashboard.*

**Start Sink connector**

curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json" http://localhost:8083/connectors/ -d @jdbc-sink.json

**Start Source connector**

curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json" http://localhost:8083/connectors/ -d @source.json

**View topics**

* docker exec -it dbz-group\_kafka\_1 ./bin/kafka-topics.sh --list --zookeeper zookeeper:2181

**View messages**

* docker exec -it dbz-group\_kafka\_1 ./bin/kafka-console-consumer.sh --bootstrap-server dbz-group\_kafka\_1:9092 --topic country --from-beginning