

Step 2: Verifying MySQL Binary Log

FeatureStore Thesis Documentation

May 5, 2025

Purpose

Debezium's change data capture requires MySQL's binary log (`log_bin`) to be enabled. This guide walks through verifying that `log_bin` is set to `ON` in your MySQL container.

1 Identify the MySQL Container

Run:

```
docker ps --filter ancestor=quay.io/debezium/example-mysql:1.9
```

Note the container **Name** (e.g. `configs-mysql-1`).

2 Confirm Root Password Environment Variable

Verify the password is injected:

```
docker exec -it configs-mysql-1 env \
| grep MYSQL_ROOT_PASSWORD
```

Expected output:

```
MYSQL_ROOT_PASSWORD=<your-root-password>
```

3 Check `log_bin` Setting

Execute:

```
docker exec -it configs-mysql-1 \
mysql -uroot -p"$MYSQL_ROOT_PASSWORD" \
-e "SHOW VARIABLES LIKE 'log_bin';"
```

Expected SQL Output

Variable_name	Value
log_bin	ON

4 Troubleshooting

- **No such container:** Ensure you use the exact container name from `docker ps`.
- **Authentication errors:** Verify `MYSQL_ROOT_PASSWORD` in your `docker-compose.yml` matches the container's env var.
- **log_bin = OFF:**

1. Create a local `my.cnf`:

```
[ mysqld ]
log_bin=ON
binlog_format=ROW
```

2. Mount it in `docker-compose.yml`:

```
services:
  mysql:
    image: quay.io/debezium/example-mysql:1.9
    volumes:
      - ./my.cnf:/etc/mysql/conf.d/my.cnf:ro
    environment:
      - MYSQL_ROOT_PASSWORD=debezium
```

3. Restart MySQL:

```
docker compose restart mysql
```

4. Re-run the verification command above.