

MYSQL REPLICATION

Architecture

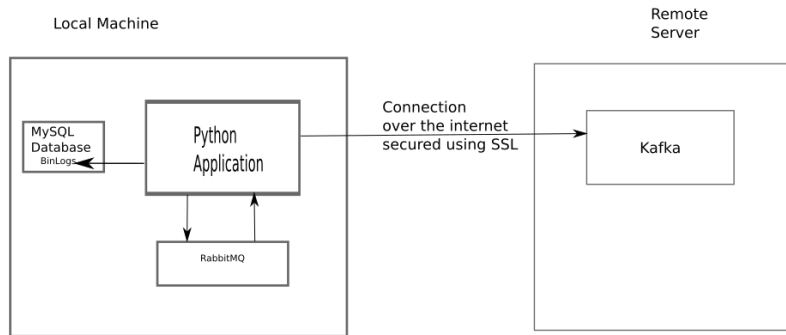


Figure 1: Architecture

The application consists of two background tasks. One task periodically checks the MySQL binlog for changes and writes those changes to a RabbitMQ queue. The other background task checks whether Kafka is accessible (online) and sends logs in the RabbitMQ queue to Kafka.

The application requires a kafka installation on remote server that can be accessed from the computer with MySQL that one intends to backup. Communication between the local machine and remote server should be encrypted using SSL.

The default setup require communication to Kafka be encrypted using SSL and connections of clients (local machines) to Kafka to authenticate using self-signed SSL certificates. One can ensure that data from one local machine cannot be accessed from another machine backing up to the same Kafka instance using Kafka ACL. In such a setup, Kafka ACL are setup such that each user can only access Kafka topics whose name match the username (eg pc1-*).

Configuring MySQL

Check whether the log-bin option is already on:

```
// for MySql 5.x
mysql> SELECT variable_value as "BINARY LOGGING STATUS (log-bin) ::"
FROM information_schema.global_variables WHERE variable_name='log_bin';
// for MySql 8.x
mysql> SELECT variable_value as "BINARY LOGGING STATUS (log-bin) ::"
FROM performance_schema.global_variables WHERE variable_name='log_bin';
```

Enable bin log and GTID in mysql configuration if not enabled in my.cnf

```
# enable bin log
server-id      = 223344
log_bin        = mysql-bin
binlog_format  = ROW
binlog_row_image = FULL
expire_logs_days = 30
binlog_rows_query_log_events=ON
```

```
# enable gtid (optional)
gtid_mode=ON
enforce-gtid-consistency=ON
## Create MySQL user for replication
```

Create a MySQL user with replication rights. Replace 'replication-user' and 'replication-password' with appropriate values.

```
CREATE USER 'replication-user'@'%' IDENTIFIED BY 'replication-password';
GRANT SELECT,EVENT, RELOAD, SHOW DATABASES, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO
FLUSH PRIVILEGES;
```

Installation

Set up RabbitMQ

```
sudo apt-get install rabbitmq-server
```

Create a RabbitMQ user, a virtual host and allow that user access to that virtual host

```
sudo rabbitmqctl add_user myuser mypassword
sudo rabbitmqctl add_vhost replication-vhost
sudo rabbitmqctl set_permissions -p replication-vhost myuser ".*" ".*" ".*"
```

Substitute in appropriate values for myuser, mypassword and myvhost above.

Install python application

Edit config.toml to match the settings of: - Remote kafka including adding ssl certificates for connecting to remote Kafka. - Local MySQL database - Computer name containing MySQL database to be backed up. This should be unique

```
make install
sudo systemctl start mysql-binlog
sudo systemctl start mysql-kafka-upload
```

Enable python application to start automatically

```
sudo systemctl enable mysql-binlog
sudo systemctl enable mysql-kafka-upload
```

Recovery

list snapshots

Get dump

```
python admin.py getdump kenbo-pc0266_2024-03-07_11-41-03_snapshot > /tmp/dump.sql
```

Get logs