



# Онлайн образование



Проверить, идет ли запись

# Меня хорошо видно && слышно?



Тема вебинара

# MySQL. Backup + Репликация



**Прусов Василий**

e-mail: [vasiliyqa@gmail.com](mailto:vasiliyqa@gmail.com)



# Преподаватель



## Василий Прусов

Более 8 лет в IT индустрии, из них более 3-х на должности системного инженера

Ранее занимался тестированием разнообразных продуктов, занимался автоматизацией тестирования

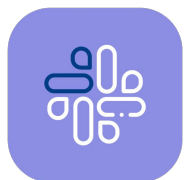
Ведущий системный инженер в “Тета Дата Солюшнс”



# Правила вебинара



Активно  
участвуем



Off-topic обсуждаем  
в Slack  
#webinars-2021-07  
или #general



Задаем вопрос  
в чат или голосом



Вопросы вижу в чате,  
могу ответить не сразу

## Условные обозначения



Индивидуально



Время, необходимое  
на активность



Пишем в чат



Говорим голосом



Документ



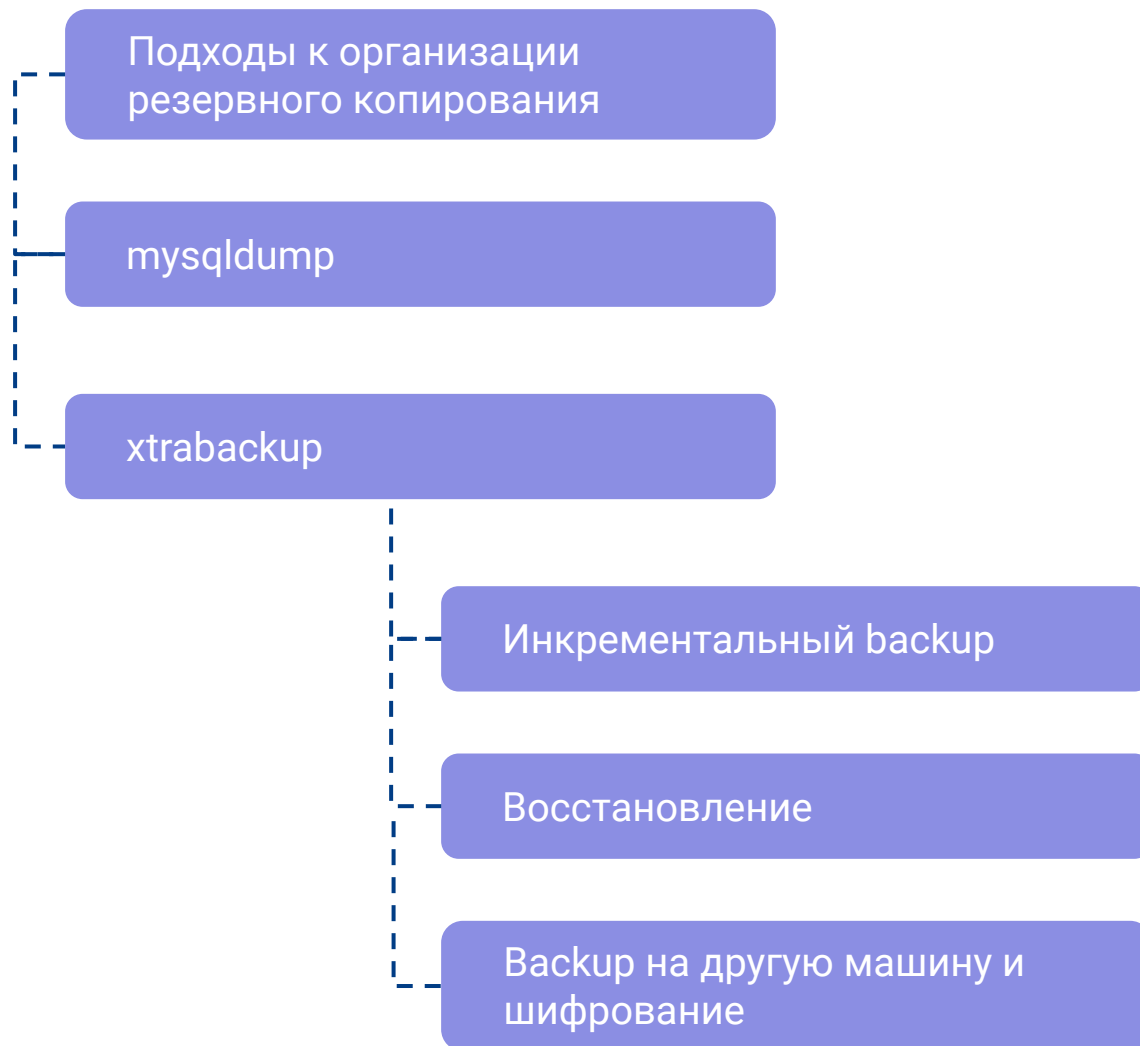
Ответьте себе или  
задайте вопрос

# Цели вебинара

1. Разобраться что такое Бэкап и Репликация
2. Научиться делать Бэкап и восстанавливать его
3. Настраивать репликацию

# Backup

# Сегодня мы разберем





# Подходы к организации резервного копирования

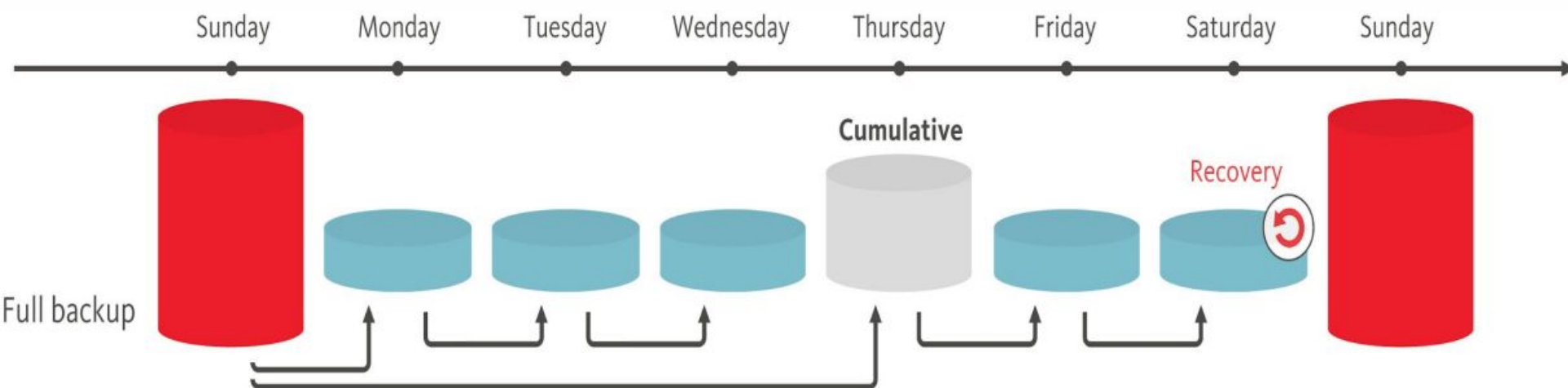
- Холодный и горячий бэкап
- Полный и инкрементальный бэкап
- Логический бэкап
  - mysqldump
  - mysqlpump
- Физический бэкап
  - бэкап средствами файловой системы (zfs, lvm, snapshots)
  - xtrabackup

# Холодный и горячий бэкапы

- Холодный
  - консистентная копия
  - требует полной остановки базы
- Горячий
  - специальные утилиты: mysqlhotbackup, xtrabackup
  - возможно использование снимков файловой системы

# Полный и инкрементальный бэкапы

- Полный позволяет восстановить полную копию на момент снятия бэкапа
- Для инкрементального нужны включенные binary\_logs



# Логический бэкап

- Варианты
  - mysqldump
  - mysqlpump
  - SELECT ..... INTO OUTFILE;
- Перенос структуры
- Полная или частичная миграция данных
- Позволяет миграцию данных между разными версиями
- Файл дампа будет значительно превышать физический размер данных
- Гораздо более медленный чем физический бэкап
- Не включает в себя логов и конфиг файлов

# mysqldump - интересные опции

```
--tab
# выходной текстовый файл с данными и sql со структурой
--single-transaction
# FLUSH, фиксирование позиции, освобождение блокировки и дамп
консистентой копии
--flush-logs
# ротация бинари логов
--master-data
# фиксирование информации SHOW MASTER STATUS;

--events
# события планировщика
--routines
# хранимые процедуры и функции
--triggers
# триггеры
```

# Преимущества xtrabackup

- Онлайн бэкап без прерывания работы БД (InnoDB)
- Инкрементальные бэкапы
- Поточковый бэкап на другой сервер
- Компрессия шифрование
- Перемещение таблиц между серверами online
- Легкое создание новых слэйвов

# xtrabackup

# Полный бэкап

```
xtrabackup --backup --target-dir=/path/to/directory/base
```

# 1й инкрементальный бэкап с момента полного

```
xtrabackup --backup --target-dir=/path/to/directory/inc1  
--incremental-basedir=/path/to/directory/base
```

# 2й инкрементальный бэкап с момента первого

```
xtrabackup --backup --target-dir=/path/to/directory/inc2  
--incremental-basedir=/path/to/directory/inc1
```

# Restore

```
# prepare подготовка для восстановления
# опция apply-log-only обязательна для инкрементального восстановления
# подготавливаем для восстановления базовый бэкап и оба инкремента
xtrabackup --prepare --apply-log-only --target-dir=/path/to/directory/base

xtrabackup --prepare --apply-log-only --target-dir=/path/to/directory/base
--incremental-dir=/path/to/directory/inc1

xtrabackup --prepare --apply-log-only --target-dir=/path/to/directory/base
--incremental-dir=/path/to/directory/inc2

xtrabackup --copy-back --target-dir=/path/to/directory/base
```



# Point-In-Time recovery

```
# необходимо включить binary logs восстановление по бинари логам  
производится после восстановления базы и выставления нужных логов  
# находим стартовую позицию для восстановления  
cat /path/to/directory/base/xtrabackup_info  
  
# стартуем восстановление до нужной точки  
mysqlbinlog /path/to/datadir/binlog.file1 /path/to/datadir/binlog.file2  
--start-position=<number> --stop-datetime="<datetime>" > mybinlog.sql
```

# Partial backups

```
# бэкап конкретных таблиц
```

```
xtrabackup --backup --datadir=/var/lib/mysql  
--target-dir=/path/to/directory/base --tables="^base_name[.]*"
```

```
xtrabackup --backup --tables-file=/tmp/tables.txt
```

```
mysqldump -d <base_name> <table_name> > <base_name>_<table_name>.sql
```

```
# бэкап конкретных БД (схем)
```

```
xtrabackup --databases="mysql sys ..."
```

```
# подготовка для восстановления, опция --export
```

```
xtrabackup --prepare --export --target-dir=/path/to/directory/base
```

# Partial восстановление

- Необходимо знать структура таблиц (dump)
- если восстанавливаем БД, то с начала CREATE DATABASE
- создаем нужную таблицу
- ALTER TABLE <table\_name> DISCARD TABLESPACE
- копируем файлы таблицы
- ALTER TABLE <table\_name> IMPORT TABLESPACE

# Stream

```
# режим STREAM направляет поток в STDOUT в формате xstream, а STDOUT уже  
можно перенаправить куда угодно
```

```
# направляем поток в файл
```

```
xtrabackup --backup --stream=xstream --target-dir=./ > backup.xstream
```

```
# с компрессией
```

```
xtrabackup --backup --stream=xstream --compress --target-dir=./ >  
backup.xstream
```

```
# с шифрованием
```

```
xtrabackup --backup --stream=xstream | gzip - | openssl des3 -salt -k  
"password" > backup.xstream.gz.des3
```

```
# расшифровка
```

```
openssl des3 -salt -k "password" -d -in backup.xstream.gz.des3 -out  
backup.xstream.gz
```

```
gzip -d backup.xstream.gz
```

```
# распаковка файла xstream
```

```
xstream -x < backup.xstream
```

# Stream на другой сервер

```
xtrabackup --backup --compress --stream=xbstream --target-dir=./ | ssh  
xtrabackup@<ip_address> "xbstream -x -C backup"
```

```
# слушаем на приемнике
```

```
nc -l 9999 | cat - > /data/backup/backup.xbstream
```

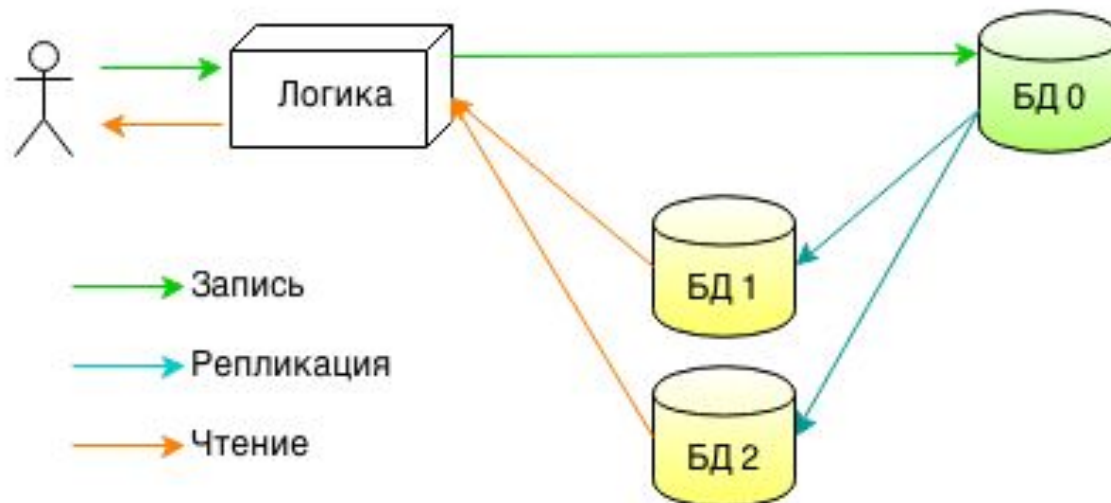
```
# Отсылаем с сервера базы
```

```
xtrabackup --backup --stream=xbstream ./ | nc desthost 9999
```

# Репликация

# Репликация

Это процесс изменения одного набора данных, называемого репликой, в ответ на изменение другого набора данных, называемого основным.



# Задачи репликации



# Зачем нужна репликация?

- Высокая доступность. Бэкап хорошо, но нужно время на его развертывание.

# Зачем нужна репликация?

- Высокая доступность. Бэкап хорошо, но нужно время на его развертывание.
- Что делать, когда закончились физические ядра и память у сервера? Горизонтально масштабировать.

# Зачем нужна репликация?

- Высокая доступность. Бэкап хорошо, но нужно время на его развертывание.
- Что делать, когда закончились физические ядра и память у сервера? Горизонтально масштабировать.
- Бэкап лучше делать с реплики, а не с мастера.

# Зачем нужна репликация?

- Высокая доступность. Бэкап хорошо, но нужно время на его развертывание.
- Что делать, когда закончились физические ядра и память у сервера? Горизонтально масштабировать.
- Бэкап лучше делать с реплики, а не с мастера.
- Геораспределение нагрузки

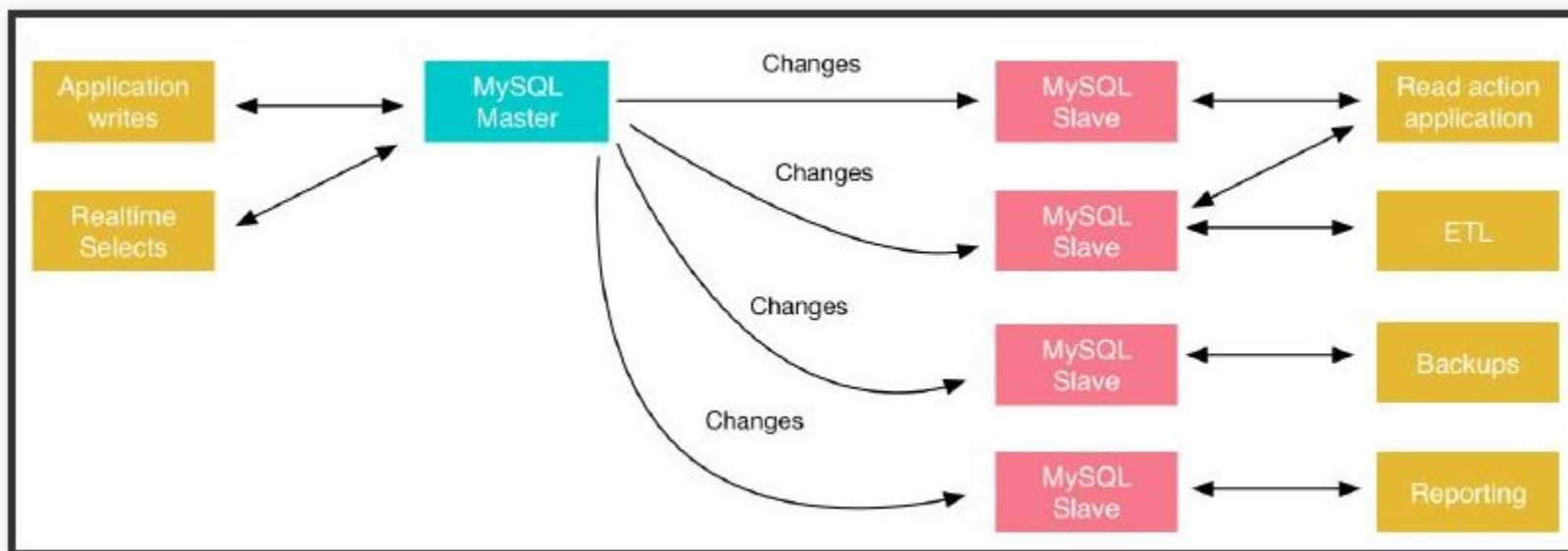
# Зачем нужна репликация?

- Высокая доступность. Бэкап хорошо, но нужно время на его развертывание.
- Что делать, когда закончились физические ядра и память у сервера? Горизонтально масштабировать.
- Бэкап лучше делать с реплики, а не с мастера.
- Геораспределение нагрузки
- Нагрузку по чтению, отчетам, etc. можно переложить на реплику.

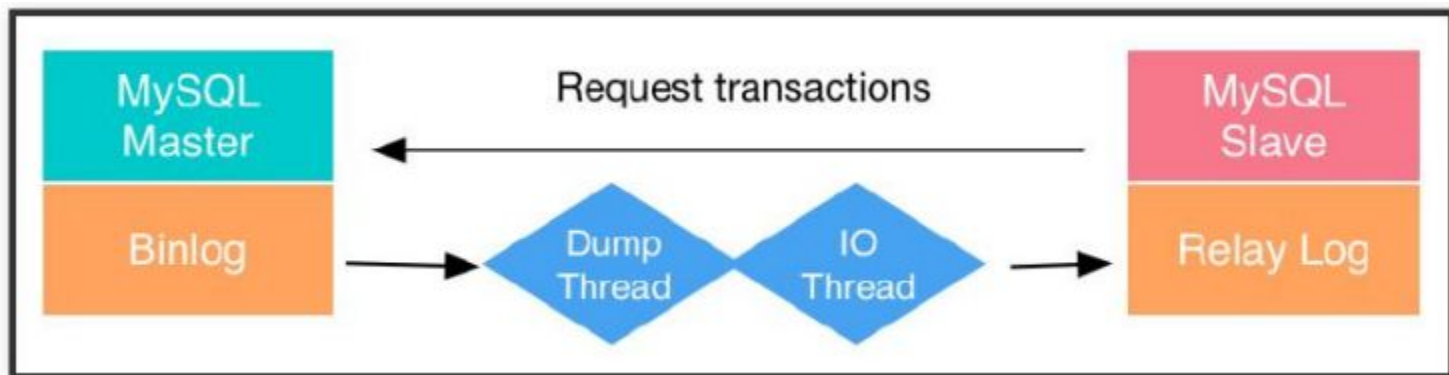
# Виды репликации

# Master Slave репликация

Один сервер (master) является источником информации для многих серверов (slave)



# Процессы Master Slave репликации



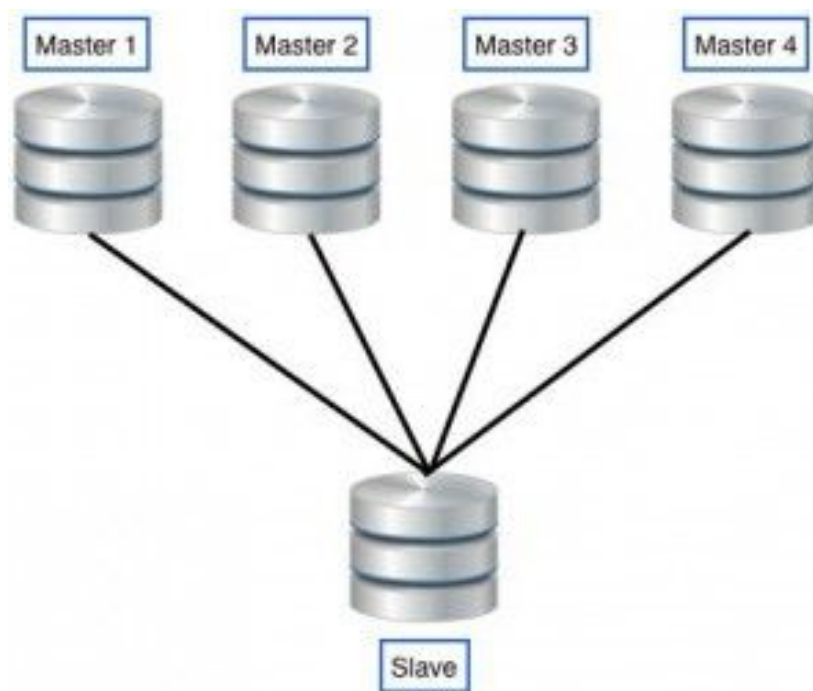
- Бинлог мастера трансферится по запросу через потоки дампа и ИО в релейлог
- Релейлог проигрывается заново на слэйве
- Слэйв пишет свой бинлог

<https://dev.mysql.com/doc/refman/8.0/en/replica-logs-relaylog.html>



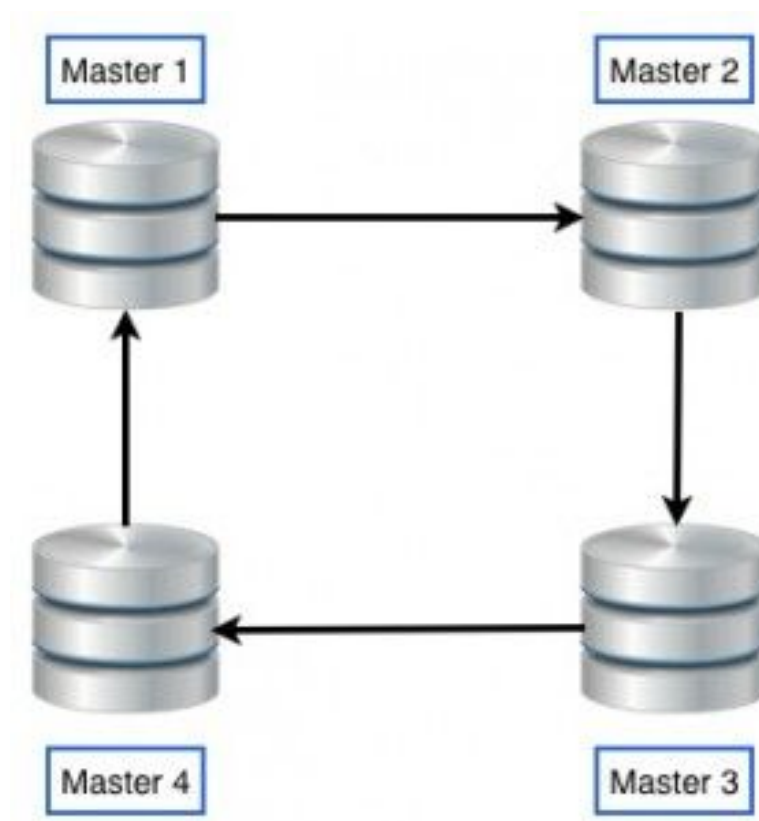
# Multi source репликация

Реплика получает данные непосредственно из нескольких мастеров.

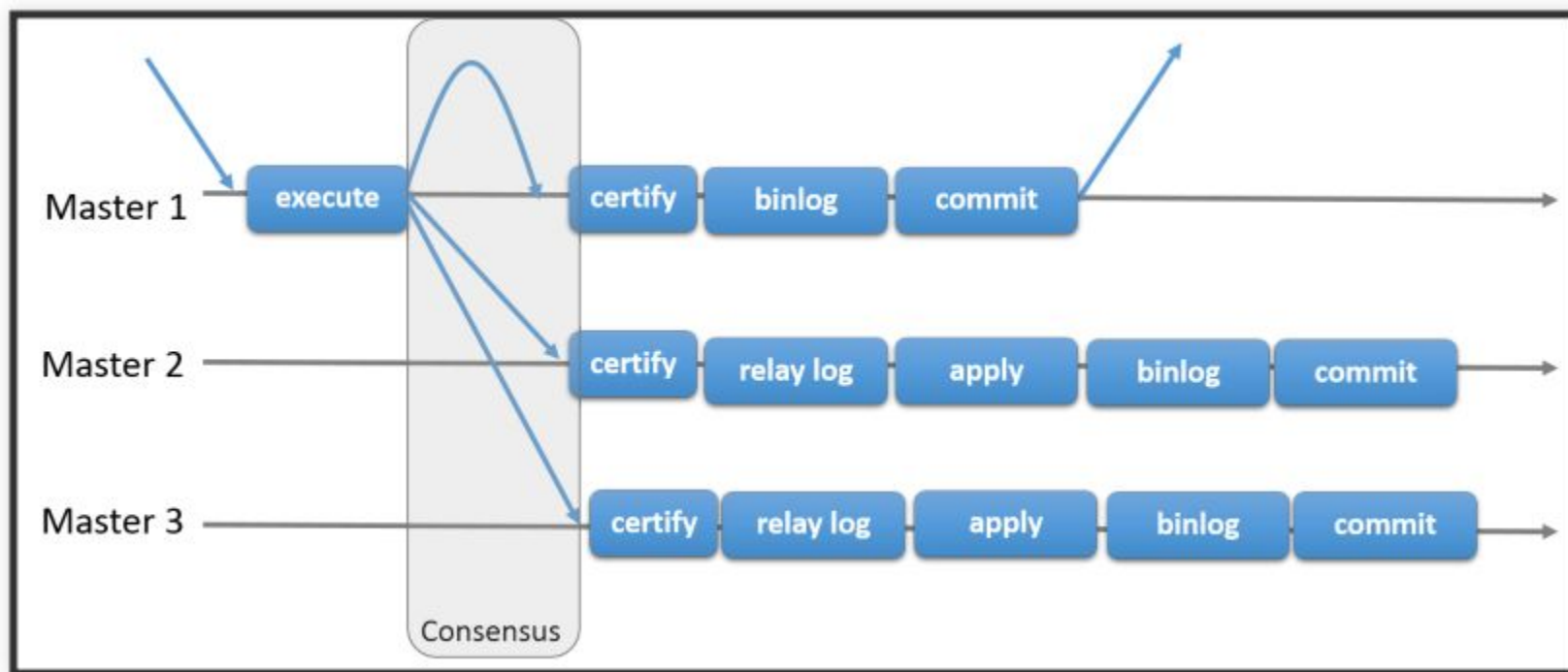


# Master-master репликация

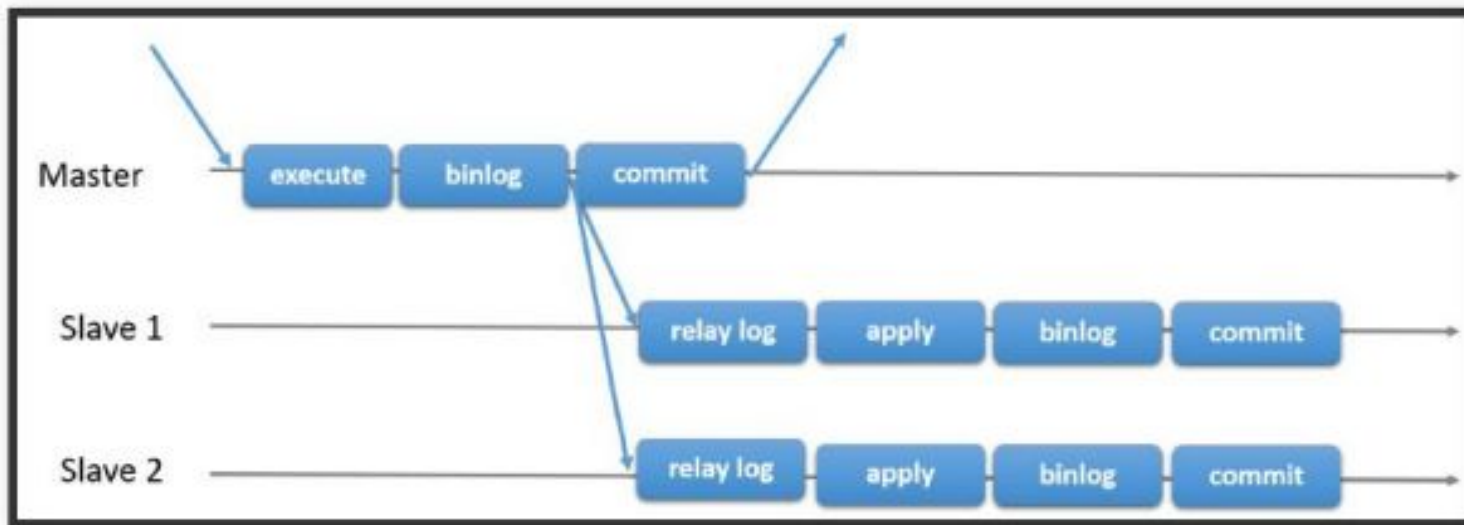
Каждый из серверов используется как для чтения так и для записи.



# Групповая репликация



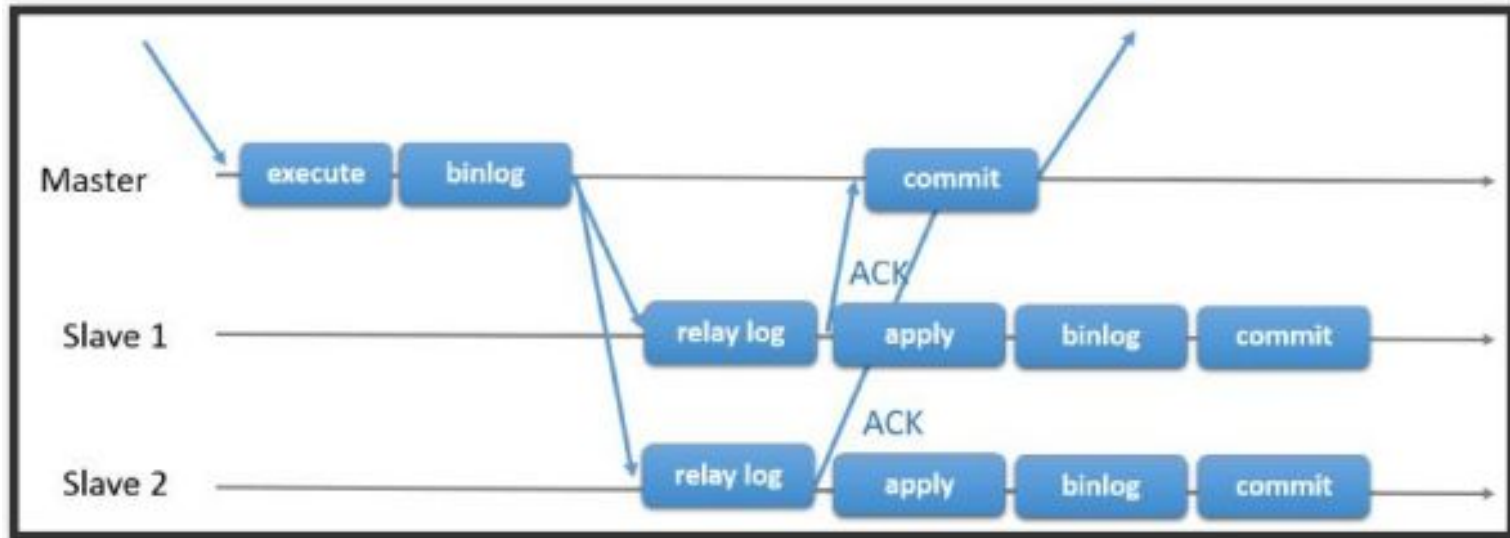
# Асинхронная репликация



- задержки на слэйвах непредсказуемы, обычно ~1 сек, но есть нюансы
- + скорость работы, не ждет слэйва

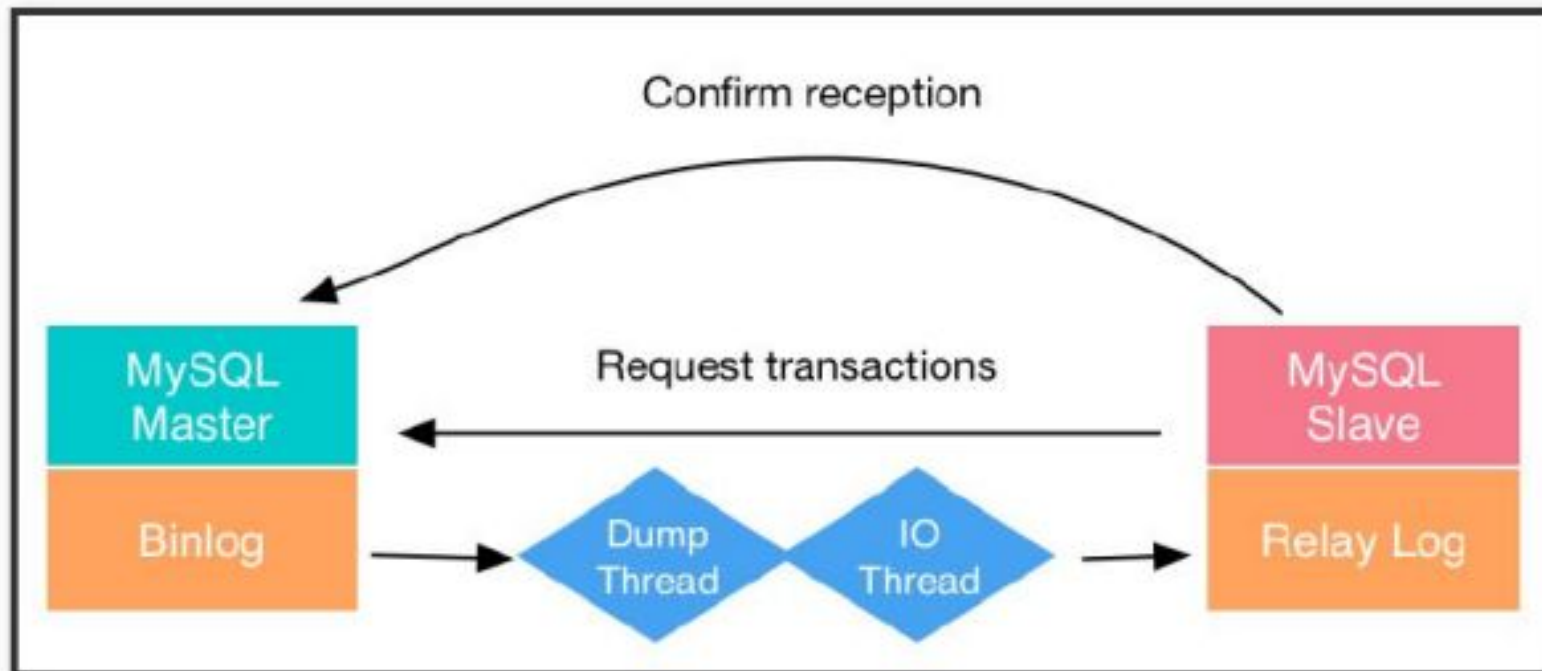
<https://blog.monyog.com/monitoring-master-slave-replication-mysql-8/>

# Полусинхронная репликация



- только после acknowledge доставки relay log, произойдет коммит - меньше скорость мастера
- задержки тоже есть
- + реплики меньше отстают от мастера

# Полусинхронная репликация



# Формат бинарного лога

➤ Документация:

<https://dev.mysql.com/doc/refman/8.0/en/replication-sbr-rbr.html>

➤ Statement-based

- в лог пишутся SQL

➤ row-based

- в лог пишутся строки
- получаются логи большего объема

➤ mixed

- для safe стейтментов пишутся SQL
- для unsafe - строки

# Опции для управления бинарными логами

```
# размер кэша в памяти
binlog_cache_size

# формат
binlog_format = ROW/STATEMENT/MIXED

# шифрование
binlog_encryption

# реакция на ошибку записи
binlog_error_action = ABORT_SERVER/IGNORE_ERROR

# таймаут хранения в секундах
binlog_expire_logs_seconds
```



# Binary Log File Position Based Replication

- включаем binary logs на мастере
- снимаем копию с мастера
- проверяем позицию бинлога
- создаем пользователя для репликации
- поднимаем копию на слэйве
- меняем server\_id на слэйве
- запускаем команду CHANGE MASTER

# On master

```
create user repl@<ip_address> IDENTIFIED BY 'password';  
GRANT REPLICATION SLAVE ON *.* TO repl@<ip_address>;  
  
SHOW MASTER STATUS;
```

# On slave

```
CHANGE MASTER TO MASTER_HOST='<ip_address>',  
MASTER_USER='repl', MASTER_PASSWORD='password',  
MASTER_LOG_FILE='binlog.file', MASTER_LOG_POSITION=<number>;  
  
START SLAVE;
```

# Команды для управления состоянием

```
STOP SLAVE;  
START SLAVE;  
RESET SLAVE;  
SHOW SLAVE STATUS;  
  
RESET MASTER;  
SHOW MASTER STATUS;  
SHOW BINLOG EVENTS;
```

# GDIT

- Global transaction identifier
- Имеет формат: GDIT = source\_id:transaction\_id
  - 3E11FA47-71CA-11E1-9E33-C80AA9429562:2
- Позволяет убедиться, что транзакция принадлежит только одному серверу
- Позволяет убедиться, что транзакция применена только 1 раз в системе

# GDIT жизненный цикл

- транзакция выполняется и коммитится на мастере
  - назначается GDIT, записывается в binlog
- slave считывает GDIT с релэй лога и прописывает его в gdit\_next
- slave производит проверку что такой gdit не используется
- для записи в бинлог slave использует gdit\_next
- после коммита транзакция добавляется в gdit-executed

# Включаем GDIT

```
# включаем режим GDIT
gdit-mode = on

# разрешаем выполнение только безопасных транзакций
enforce-gdit-consistency=true

# на слэйве
CHANGE MASTER TO MASTER_HOST='<ip_address>', MASTER_USER='repl',
MASTER_PASSWORD='password', MASTER_AUTO_POSITION=1;
START SLAVE;
```

# Мониторим GDIT

```
SELECT @@gdit_executed;  
SELECT @@gdit_purged;
```



# Мониторинг репликации

```
# таблицы в Performance Schema

replication_applier_configuration
replication_applier_filters
replication_applier_global_filters
replication_applier_status
replication_applier_status_by_coordinator
replication_applier_status_by_worker
replication_connection_configuration
replication_connection_status
replication_group_member_stats
replication_group_members
```

# Полусинхронная репликация

```
# на мастере и слейвах
INSTALL PLUGIN rpl_semi_sync_master SONAME 'semisync_master.so';

# включение режима репликации
rpl_semi_sync_master_enabled

# таймаут ожидания ответа в мс
rpl_semi_sync_master_timeout

# включение дебага
rpl_semi_sync_master_trace_level

# ожидание подтверждения от заданного кол-ва слейвов
rpl_semi_sync_master_wait_for_slave_count

# коммит на мастере после синхронизации бинлогов на слейв, либо
коммит и дожидается ответа от слейва
rpl_semi_sync_master_wait_point = AFTER_SYNC/AFTER_COMMIT
```

# Фильтры на слэйве

```
CHANGE REPLICATION FILTER filter[, filter]  
    [, ...] [FOR CHANNEL channel]
```

filter:

```
    REPLICATE_DO_DB = (db_list)  
| REPLICATE_IGNORE_DB = (db_list)  
| REPLICATE_DO_TABLE = (tbl_list)  
| REPLICATE_IGNORE_TABLE = (tbl_list)  
| REPLICATE_WILD_DO_TABLE = (wild_tbl_list)  
| REPLICATE_WILD_IGNORE_TABLE = (wild_tbl_list)  
| REPLICATE_REWRITE_DB = (db_pair_list)
```

**Заполните, пожалуйста,  
опрос о занятии  
по ссылке в чате**

Спасибо за внимание!

# Приходите на следующие вебинары



**Прусов Василий**

e-mail: [vasiliyqa@gmail.com](mailto:vasiliyqa@gmail.com)

