MySQL Point-in-Time-Recovery (PiTR)

DOAG K + A, 2022, Nürnberg (D)

Oli Sennhauser

CTO, FromDual GmbH

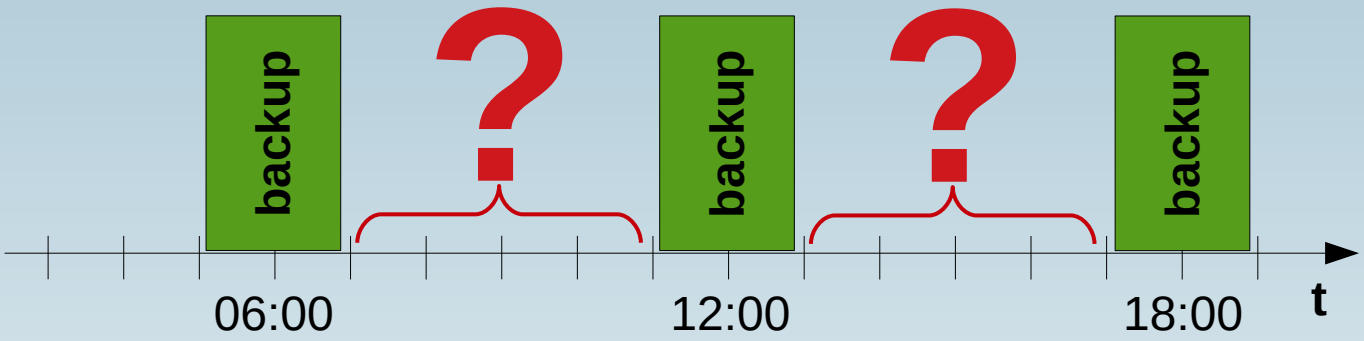
https://www.fromdual.com/presentations

How are you doing backups?

- Most of us do backups? Why?
  - Hardware gets broken
    → You are too slow to fix hardware! This can take hours to days to recover! → HA
  - Logical errors
    → Most common error type!
    e.g. `DROP TABLE ...`
- How are we doing backup?
  - Logical backup: `mysqldump`
  - Physical backup: `xtrabackup`

Point-in-Time-Recovery (PITR)

- Some of us do several backups a day (pulsed backup):



- How many hours between the backups?
- But what is in between?
- This is where PiTR comes into play!
  - Recover ALL changes since LAST full backup! By replaying the binary logs...

Binary Logs

- We must record all the DB changes after the last backup
  - Or better: All the changes at all
- This is done with enabling the "Binary Logs" = Change Log, Journal of the DB

```
# /etc/mysql/conf.d/99-fromdual.cnf

[mysqld]

log_bin       = /var/log/mysql/binary-log
binlog_format = ROW
```

- Enabling Binary Logs requires DB restart!
- MySQL 8.0: Enabled by default!

Logical Backup for PiTR

- We need a consistent Full Backup of the DB
  - Partial Backup (Schema, Table) is possible
- Our recommendation for Logical Backup:

```
# for only InnoDB tables!!!
mysqldump --user=root --all-databases --flush-logs \
--single-transaction --master-data=1 --flush-privileges \
--quick --triggers --routines --events \
--hex-blob > /var/backup/full_dump.sql
```

- `--master-data` → `--source-data`!!!

Restore Logical Backup

- **Restore:**

```
zcat /var/backup/full_dump.sql.gz | mysql --user=root
```

Physical Backup for PiTR

- We need a consistent Full Backup of the DB
- Our recommendation for Physical Backup:

```
xtrabackup --user=root --backup --target-dir=/var/backup
xtrabackup --user=root --prepare --target-dir=/var/backup
```

- You need physical access to the DB files!
- `xtrabackup` (Percona) vs. MySQL Enterprise Backup (Oracle)
- MySQL 8.0.30 has broken compatibility and thus also the tool!
  - → Wait for new version or wait with upgrade!
  - Downgrade in MySQL 8.0 is NOT possible any more!

* Wish to the developers: `--backup-and-prepare` Option!!!

Restore Physical Backup

- **Restore:**

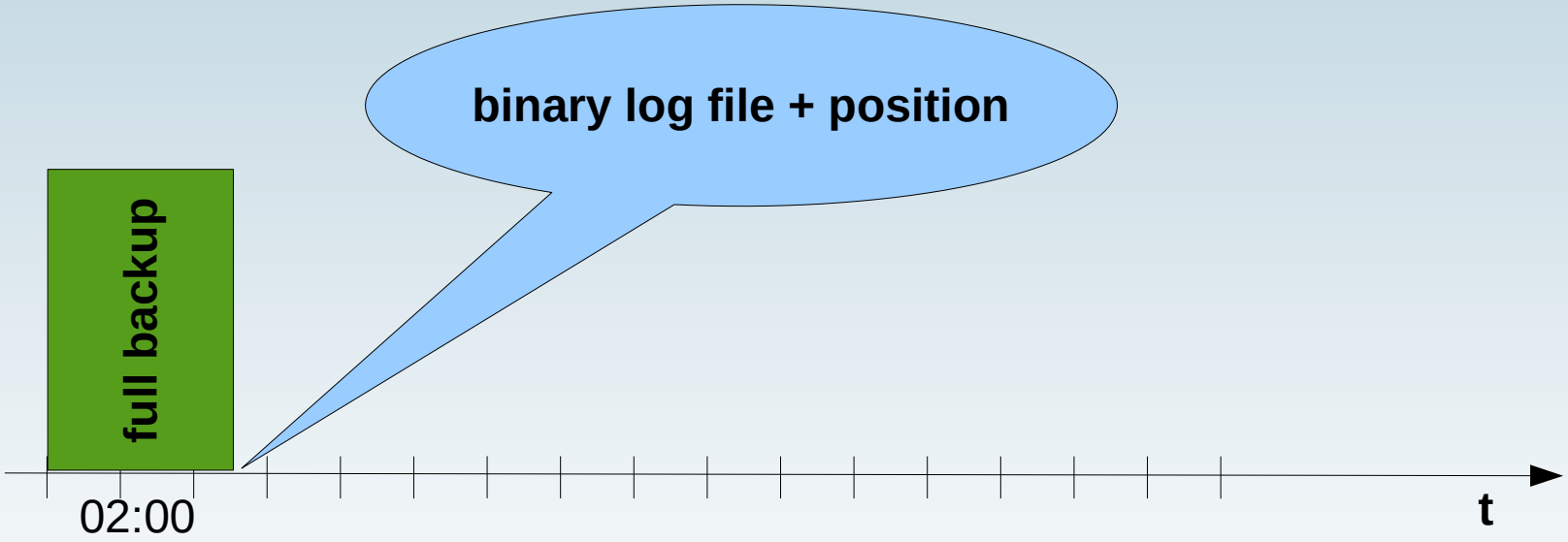
```
systemctl stop mysql

rm -rf /var/lib/mysql/*

xtrabackup --copy-back \
--datadir=/var/lib/mysql --target-dir=/var/backup

chown -R mysql: /var/lib/mysql

systemctl start mysql
```

*** Wish to the developers: remember ownership of the files?**

Point-in-Time-Recovery

- We need **exact** point in time of the backup in the Binary Log stream

  → Binary Log File and Position



Starting point for PiTR?

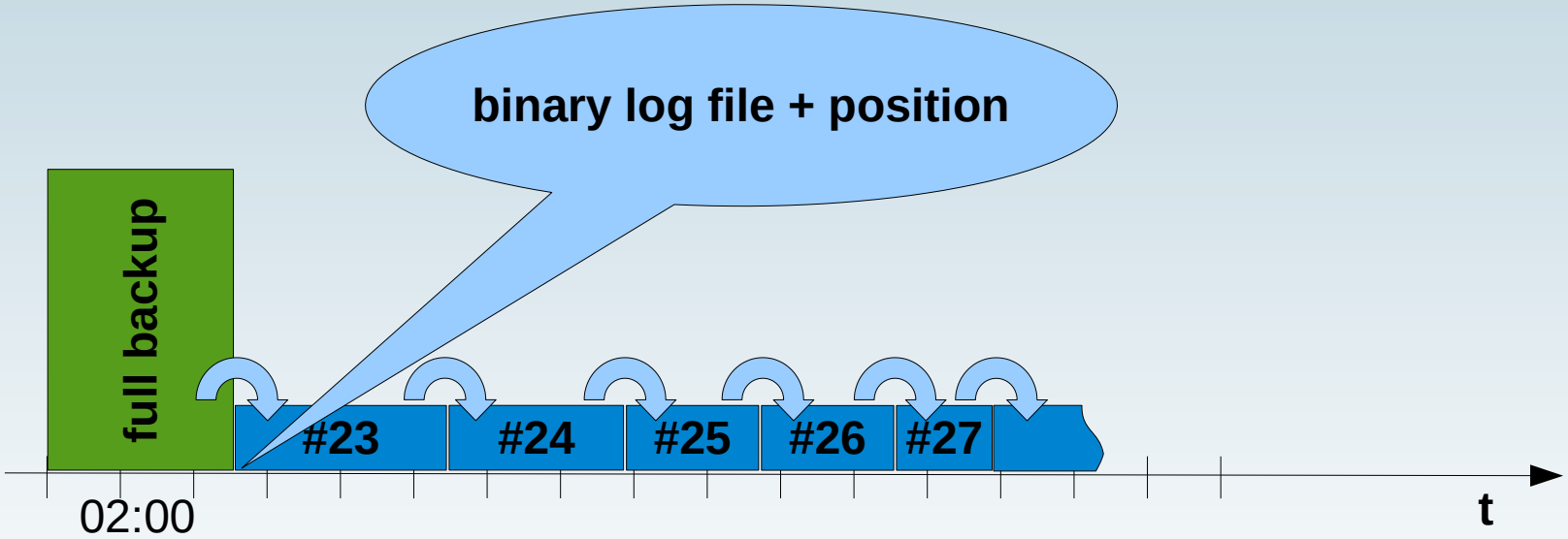
- **mysqldump:**

```
head -n 25 /var/backup/full_dump.sql
18 --
19 -- Position to start replication or point-in-time recovery from
20 --
21
22 CHANGE MASTER TO MASTER_LOG_FILE='binary-log.000023', MASTER_LOG_POS=157;
```

- **xtrabackup:**

```
cat /var/backup/xtrabackup_binlog_info
binary-log.000023  157
```

Point-in-Time-Recovery



Point-in-Time-Recovery

- How to do a PiTR:

```
mysqlbinlog --disable-log-bin \
--start-position=157 \
--stop-datetime='2022-09-01 13:59:00' \
binary-log.000023 binary-log.000024 \
  … binary-log.000027 | mysql --user=root
```

- To be more precise: `--stop-position` can be used!
  - Needs `mysqlbinog` utility to find exact stop position (--verbose).
- First try against `less` to check if you got it right!

Advanced: GTID PiTR

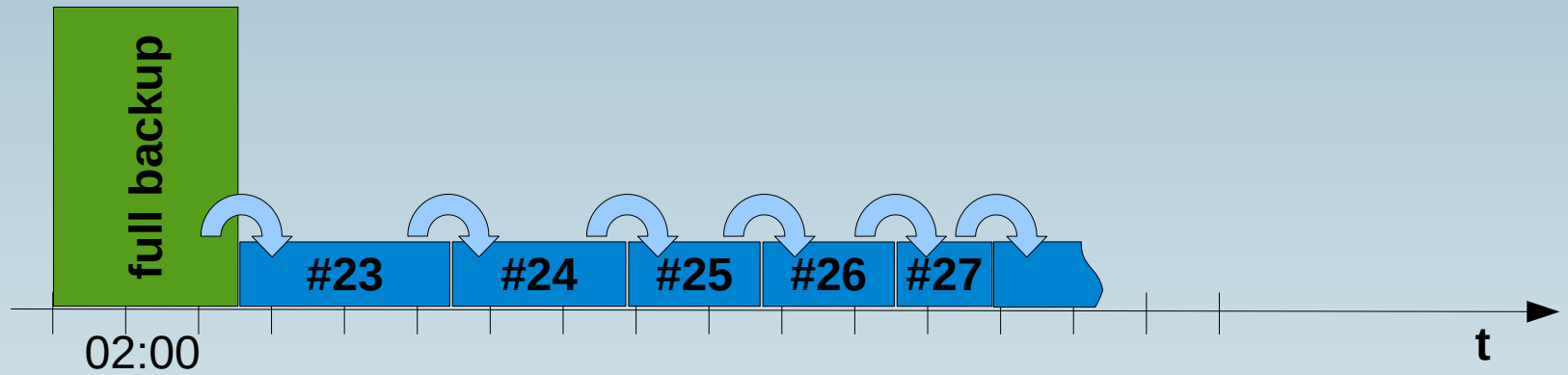
- Do NOT forget to backup auto.cnf
  - To avoid a new sever_uuid

```
shell> mysqldump --set-gtid-purged --source-data=1 ...

23 -- GTID state at the beginning of the backup
26 SET @@GLOBAL.GTID_PURGED=/*!80000 '+'*/ \
   '84fde136-367f-11ed-a479-acfdcee57bd5:1-6310';

shell> mysqlbinlog --stop-datetime='2022-09-01 13:59:00' \
chef_mysql-80_binlog.0000* | mysql --binary-mode
```

- Buggy? GTID_PURGED is too high after restore of phyiscal backup (xtrabackup)!?! This prevents us from applying transactions from binary log…
- Why: Somehow MySQL reads gtid_executed from present binary logs!!!
  - Wether this is a bug or works as designed is unclear…
  - Solution: Move binary logs somewhere else...

Advanced: Flashback

- Instead of doing all this:



- What about just fixing the logical error?
- Binary Log Flashback → Reverse DML operation (needs `binlog_format = ROW`)

```
mariadb-binlog --flashback --start-position=348622898 \
--stop-position=349828120 binary-log.000080 \
| mysql --user=root test --force
```

Test your restore!!!

- Personal advice: Test your restore!!!
- Why should you test your restore?

- Your restore might not work at all!?!
  - And you find it out much, much later...
- Restore can take a very long time.....
  - Verify your MTTR or RTO/RPO!
    - MTTR - Mean Time to Repair
    - RTO - Recovery Time Objective
    - RPO - Recovery Point Objective
- Getting experience!

Thank you!

Questions ?

Discussion?

We have some time for a personal talk...

FromDual provides neutral and independent:

- **Consulting**
- **remote-DBA**
- **Support for MySQL and Galera Cluster**
- **Training**

www.fromdual.com/presentations