

MySQL Yedek & Kurtarma Referansı

mysqldump, MySQL Shell, XtraBackup, binlog arşivi, PITR ve felaket senaryoları için production rehberi



RPO

Ne kadar veri? · kabul edilebilir kayıp



RTO

Ne kadar sürede? · geri dönüş süresi



KURAL

3-2-1-1-0 · 3 kopya, 2 ortam, 1 off-site



KURAL

imm

01 Temel Kavramlar & Strateji



KAVRAM	AÇIKLAMA
Mantıksal	SQL / nesne seviyesinde döküm. Sürüm & platform bağımsız, seçici restore kolay; büyük veride yavaş restore (indeksler yeniden kurulur).
Fiziksel	Veri dosyalarının blok kopyası. Hızlı restore , TB ölçeğe uygun; aynı ana sürüm + platform gerekir.
Snapshot	LVM / ZFS / EBS / SAN anlık görüntüsü. Crash-consistent; WAL/redo aynı volume'deyse ya da backup modu kullanılırsa güvenli.
Full / Incr / Diff	Tamamı / son yedekten beri / son full'den beri değişenler. Incremental zinciri uzadıkça restore süresi ve risk artar .
Hot / Warm / Cold	Okuma+yazma açık / sadece okuma / sunucu kapalı. Production'da hedef: hot .
Sürekli Arşiv	Full yedek + kesintisiz binlog akışı → PITR : herhangi bir ana geri dönüş. RPO'yu saniyelere indirir.

YÖNTEM SEÇİMİ

SENARYO	ÖNERİ
< 100 GB, sürüm geçişi, tek şema	<code>util.dumpSchemas</code> / <code>mydumper</code>
TB ölçek, kısa RTO	XtraBackup / MEB + binlog
Saniyelik RPO	Semi-sync replika + binlog stream
"Yanlışlıkla sildim" koruması	PITR + <code>SOURCE_DELAY</code>

ARAÇ HARİTASI

Mantıksal	<code>mysqldump</code> · MySQL Shell <code>util.*</code> · <code>mydumper</code>
Fiziksel	XtraBackup · MEB (<code>mysqlbackup</code>) · CLONE
Binlog	<code>mysqlbinlog</code> · <code>SHOW BINARY LOGS</code>
Yalnızca InnoDB	Hot fiziksel yedek ve tutarlı dump yalnızca InnoDB'de garanti

Replikasyon yedek değildir: `DROP TABLE` veya hatalı `UPDATE` milisaniyeler içinde tüm replikalara da gider. HA ≠ DR.

02 Mantıksal Yedek

ARAÇ	PARALEL	NOT
<code>mysqldump</code>	X	Her yerde var; tek thread, küçük/orta veri
<code>mysqlpump</code>	✓	8.0.34'te deprecated, 8.4'te kaldırıldı — kullanma
Shell <code>util.*</code>	✓	<code>dumpInstance</code> / <code>dumpSchemas</code> / <code>dumpTables</code> + <code>loadDump</code> ; <code>zstd</code> , chunking, resume
<code>mydumper</code>	✓	Açık kaynak, <code>myloader</code> ile paralel restore

MYSQLDUMP — PRODUCTION BAYRAKLARI

```
$ mysqldump --single-transaction --source-data=2 \
--routines --events --triggers --hex-blob \
--set-gtid-purged=ON --all-databases \
| zstd -T0 > /bk/full_$(date +%F).sql.zst
# --source-data=2 → binlog konumu yorum olarak yazılır
# (8.0.26 öncesi: --master-data=2)
```

MYSQL SHELL DUMP & LOAD (JS)

```
util.dumpInstance('/bk/inst_2026-10-13',
  {threads: 8, compression: 'zstd', consistent: true})
util.dumpSchemas(['shop', '/bk/shop', {threads: 8})
// hedefte local_infile=ON gerekir
util.loadDump('/bk/inst_2026-10-13',
  {threads: 8, deferTableIndexes: 'all'})
```

RESTORE

```
$ zstd -dc full.sql.zst | mysql -u root -p
$ myloader -d /bk/mydumper -t 8 --overwrite-tables
```

Dikkat: `--single-transaction` yalnızca **InnoDB** için tutarlıdır; MyISAM tabloları tutarsız kalır. Dump sırasında DDL (`ALTER/TRUNCATE`) çalıştırma — tutarlılık bozuluşu

Hız: Shell `util.loadDump` ve `myloader`, düz SQL restore'a göre çok daha hızlıdır. paralel yükler ve indeksleri en sona bırakır.

04 Binlog & Point-in-Time Recovery KRİTİK



PARAMETRE	ÖNERİLEN
log_bin	ON (8.0+ varsayılan)
binlog_format	ROW
sync_binlog	1 — her commit'te diske
innodb_flush_log_at_trx_commit	1 — tam dayanıklılık
binlog_expire_logs_seconds	≥ en eski yedeğin yaşı
gtid_mode	ON + enforce_gtid_consistency

🔧 CANLI BINLOG YEDEĞİ (AYRI SUNUCUDA)

```
$ mysqlbinlog --read-from-remote-server --raw \
--stop-never --host=db1 --user=bkp -p \
--result-file=/binlogs/ binlog.000001
```

📖 ADIM ADIM PITR

- 1 Full yedeği geri yükle (dump / XtraBackup / CLONE).
- 2 Başlangıç konumunu bul: dump başlığı veya xtrabackup_binlog_info.
- 3 Hatalı olayın konumunu bul (aşağıda).
- 4 Binlog'ları **tek komutla** olaydan hemen öncesine kadar oynat.

🔍 OLAYI BUL & OYNAT

```
$ mysqlbinlog -v --base64-output=DECODE-ROWS \
--start-datetime="2026-10-13 09:50:00" binlog.000043 \
| grep -n -B5 -i "DROP TABLE" # → end_log_pos
$ mysqlbinlog --start-position=157 --stop-position=48211 \
binlog.000042 binlog.000043 | mysql -u root -p
# GTID ile: hatalı işlemi atla, gerisini oynat
$ mysqlbinlog --exclude-gtids='3E11FA47-...:1234' \
binlog.000042 binlog.000043 | mysql -u root -p
```

⚠️ **Tuzak:** Binlog dosyalarını **ayrı ayrı mysqlbinlog | mysql** ile oynatma — temporary table'lar ve çok dosyaya yayılan transaction'lar kırılır. Hepsini tek çağırda ver.

05 Doğrulama, Test & İzleme

KONTROL	ARAÇ / YÖNTEM
Yedek Bütünlüğü	xtrabackup --prepare → "completed OK!", MEB validate
Veri Bütünlüğü	CHECK TABLE, mysqlcheck, CHECKSUM TABLE
Replika Tutarlılığı	pt-table-checksum → fark varsa pt-table-sync
Binlog Sağlığı	binlog stream süreci, SHOW BINARY LOGS, gtid_purged
Restore Testi	Otomatik, en az haftalık; ayrı sunucuda geri yükle → sorgu ile doğrula → RTO'yu ölç ve kaydet.

💖 BINLOG SAĞLIK KONTROLÜ

```
SHOW BINARY LOGS;
SHOW BINARY LOG STATUS; -- 8.4 (8.0: SHOW MASTER STATUS)
SELECT @@gtid_executed, @@gtid_purged,
@@binlog_expire_logs_seconds;
```

🔧 BÜTÜNLÜK KONTROLLERİ

```
$ mysqlcheck --all-databases --check -u root -p
$ mysqlbackup --backup-image=bk/full.mbi validate
$ pt-table-checksum --replicate=percona.checksums h=db1
CHECKSUM TABLE shop.orders;
```

🚨 ALARM KUR

Son başarılı yedeğin yaşı > 26 saat	KRİTİK
Binlog stream süreci durdu / gecikme > RPO	KRİTİK
Binlog, sonraki yedekten önce purge edildi	KRİTİK
Yedek boyutunda ani ±%30 değişim	ÖNEMLİ
Yedek disk doluluğu > %80	ÖNEMLİ

📌 **MySQL 8.4:** SHOW MASTER STATUS kaldırıldı → SHOW BINARY LOG STATUS kullan (8.2+). İzleme script'lerini sürüm yükseltmeden önce güncelle.