

MySQL Major Version Upgrade Checklist

A replicated environment upgrade guide

PHASE 1: Upgrade Testing

5 steps for safe testing in isolated environment

PHASE 2: Upgrade Preparation

2 steps to resolve issues and prepare infrastructure

PHASE 3: Cutover

3 steps to execute the production migration

PHASE 4: Rollback Plan

2 steps for contingency and rollback procedures

Source: kedar.nitty-witty.com

PHASE 1: UPGRADE TESTING

STEP 1: Prerequisites & Setup

- Provision two identical test servers (test-mysql-1, test-mysql-2)
Production-grade resources only if load testing planned
- Install current MySQL version on both test servers
- Restore production data backup to both instances
- Configure chained replication: production → test-mysql-1 → test-mysql-2
Both instances should be read-only at this stage

STEP 2: MySQL Upgrade Checker

- Install MySQL Shell on test-mysql-2
- Run `checkForServerUpgrade()` against the production database
- Document all compatibility issues, warnings, and deprecation notices
- Fix all identified compatibility issues before proceeding

STEP 3: Prepare for Test Upgrades

- Apply `checkForServerUpgrade()` fixes on test-mysql-2
- Upgrade MySQL to target version on test-mysql-2
- Configure both servers with equivalent settings (except version-specific differences)
- Disconnect replication — both servers now hold independent identical datasets
- Log all upgrade learnings: my.cnf changes, steps taken, issues encountered

STEP 4: Performance Testing (pt-upgrade)

- Collect at least 24 hours of slow query logs from production-mysql
Ensure logs cover all typical query patterns throughout the day
- Run pt-upgrade tool in read-only mode against both instances
- Enable read-write on both instances and re-run pt-upgrade for write workloads
- Analyse pt-upgrade output thoroughly
- Log issues: time differences, plan changes, result differences, errors, reserved keywords

STEP 5: Application Compatibility Testing

- Test application against the target version on test-mysql-2
- Document any application-level issues or performance degradation
- Create an application update plan for all identified incompatibilities

PHASE 2: UPGRADE PREPARATION

STEP 1: Issue Resolution

- Apply all fixes identified during checkForServerUpgrade and pt-upgrade
- Implement all required database-level changes (config, authentication, etc.)
- Obtain go-ahead sign-off from the application team for the target version
- Re-test all fixes to confirm they fully resolve the issues (if required)

STEP 2: Infrastructure & Cutover Preparation

- Provision production-grade candidate servers: candidate-prod-1, candidate-prod-2
Existing test servers may be repurposed if resources are suitable
- Restore production data and configure chained replication to candidate servers
Topology: production → candidate-prod-1 → candidate-prod-2
- Upgrade candidate servers to target version using Phase 1 learnings
- Configure backups, monitoring, and validate restoration from backups
- Document the new environment and all cutover requirements

PHASE 3: CUTOVER

STEP 1: Pre-Cutover Activities

- Verify replication is functioning properly with minimal lag and no errors
- Ensure latest full backups are available and validated
- Confirm all monitoring systems are active and functioning
- Notify all stakeholders of the upcoming maintenance window

STEP 2: Cutover Execution

- Set current production database to read-only mode
- Verify replication has fully caught up — zero lag confirmed
- Disconnect replication from production to candidate-prod-1 (primary)
- Update application connection strings to point to candidate-prod-1 (writer)
- Enable read-write on candidate-prod-1 and start routing application traffic
- Run validation tests and make formal Go / No-Go decision

STEP 3: Post-Cutover Activities

- Monitor application performance and error rates continuously
- Monitor database performance metrics in real-time
- Keep original environment available for potential rollback
- Collect performance metrics and compare with pre-upgrade baseline
- Notify stakeholders of successful upgrade completion
- Document any issues encountered and their resolutions

PHASE 4: ROLLBACK PLAN

STEP 1: Rollback Preparation

- Assess whether reverse replication from candidate back to original is viable
- Establish application tolerance for possible data loss or restore delay
- Identify manual actions required if reverse replication is not possible
- Define specific triggers that would initiate a rollback decision
- Establish and communicate decision-making authority for rollback
- Prepare and test rollback scripts and procedures

STEP 2: Rollback Execution (If Needed)

- Update application connection config to point back to original production
- Return original production database to read-write mode
- Verify application functionality against the original database
- Notify stakeholders of the rollback and expected resolution timeline
- Document the specific issues that triggered the rollback

Source: kedar.nitty-witty.com