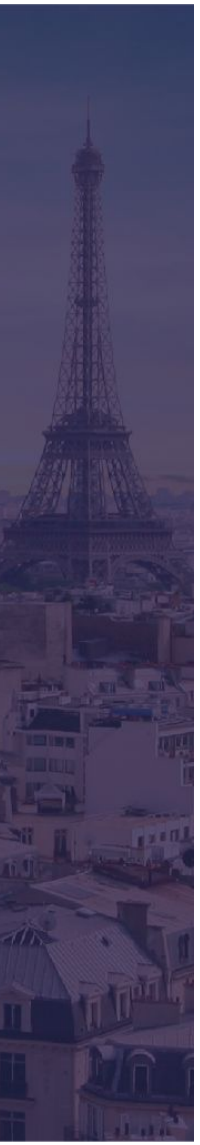
Top reasons to upgrade to MySQL 8.0 and problems you might encounter in the process

Dmitry Lenev
Principal Software Engineer
Percona

Paris, France 2024
PERCONA UNIVERSITY

FerretDB

BlaBlaCar



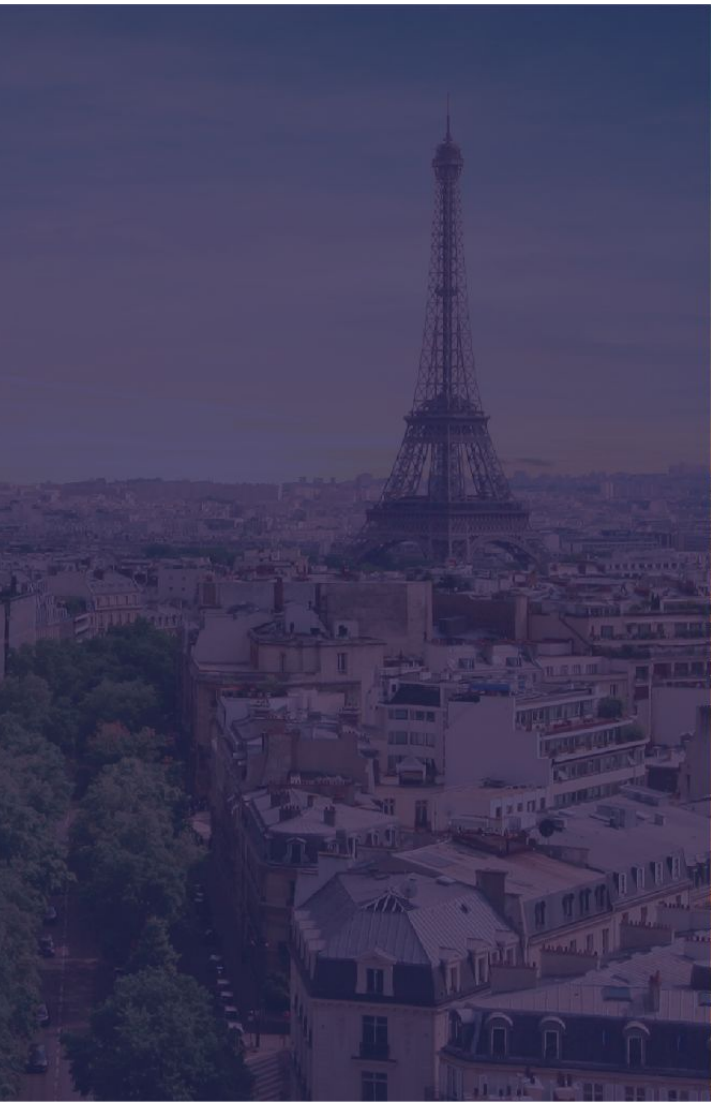
Why are we talking about this in 2024?

Many people still use MySQL 5.7:

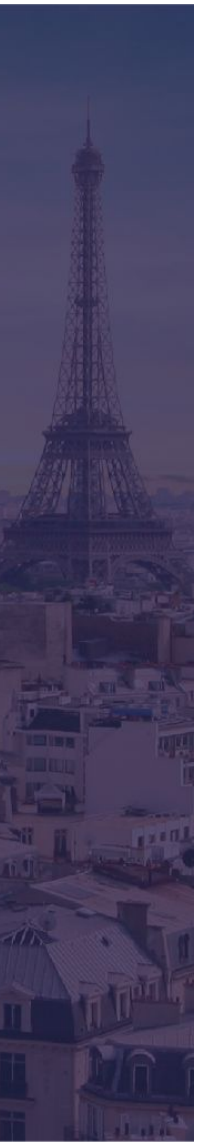
- PMM statistics:
  - 5.7 share is declining but still above 30%!
- New Percona Telemetry:
  - Around 26% of new PS deployments
  - Around 23% of new PXC deployments

People are upgrading from 5.7 to 8.0:

- Hit and report issues like PS-9186, PS-9207 in the process.



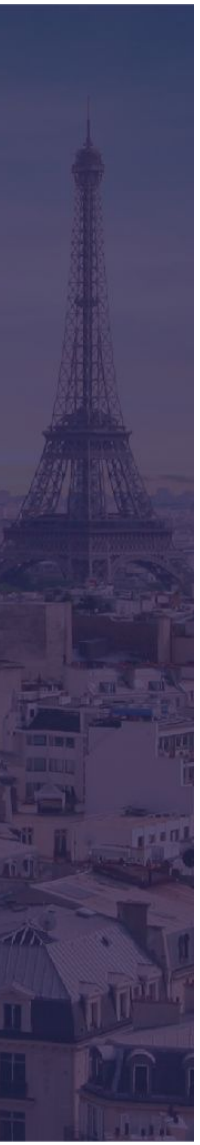
Reasons to upgrade to MySQL 8.0



Reason #1

MySQL 5.7 has reached End-of-Life in October 2023:

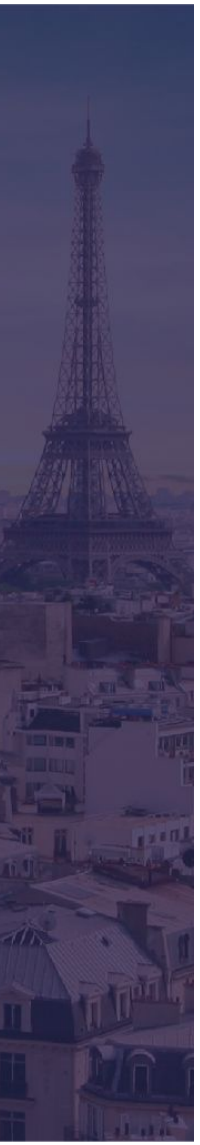
- 5.7.44 from 2023-10-25 is final release
- no new fixes
- no even security/CVE fixes
  - Few fixes in Percona Server 5.7.44-49 from 2024-03-15 (part of Percona Post-EOL Support)



Reason #2

MySQL 8.0 is well matured:

- 8.0.37 (2024-04-30) is latest
- GA since April 2018
- Switched from CDM to bug-fix only releases from 8.0.34 (2023-07-18)
- End-of-life is scheduled for April 2026!
  - New LTS series 8.4 started with 8.4.0 in April 2024
  - No direct upgrade from 5.7 to 8.4



Reasons #3, #4, …

New SQL features:

- Common Table Expressions
- LATERAL tables
- Window functions

- CHECK constraints

- JSON_TABLE

PERCONA

Reasons #7, #8, …

Query Optimizer and Executor Improvements:

- EXPLAIN ANALYZE
- INVISIBLE indexes

- Descending indexes
- Functional indexes

- HASH JOIN

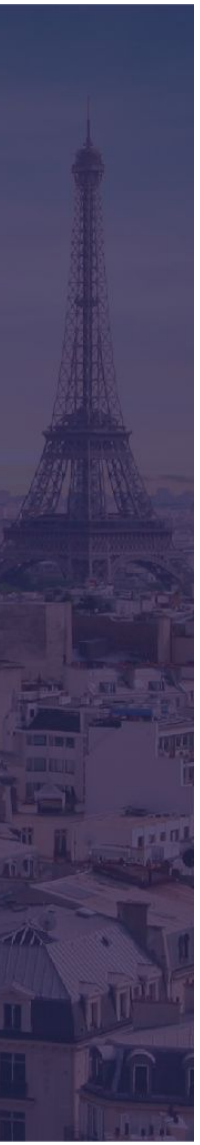
- Histograms

Reason #13

UTF8MB4 improvements:

- UTF8MB4 is new default
- New UCA 9.0 collations
- Performance optimizations

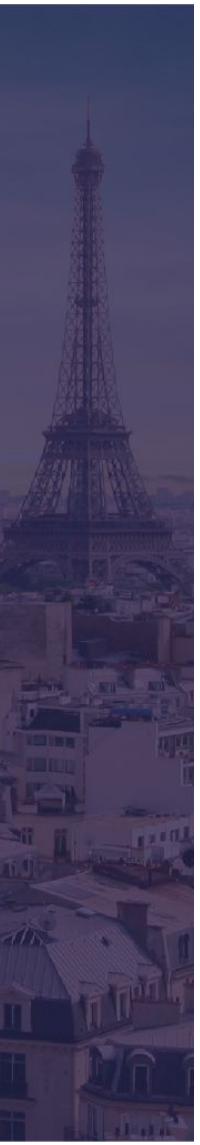
PERCONA



Reason #14

ALTER TABLE .. ALGORITHM=INSTANT

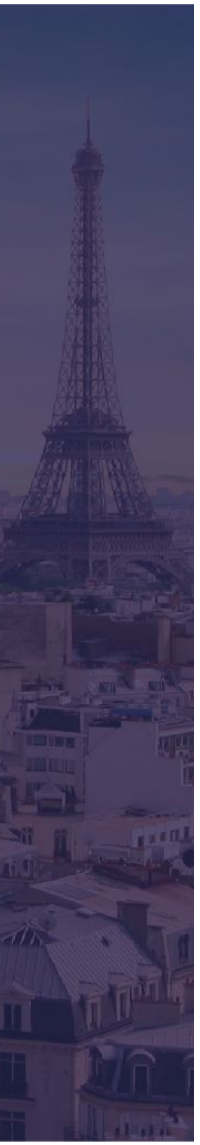
- ADD COLUMN
- DROP COLUMN
- …



Reasons #15, #16, …

Performance and Reliability improvements:

- New Data-Dictionary
  - Faster INFORMATION_SCHEMA queries
  - Atomic DDL
- InnoDB performance at scale improvements
  - latch-free REDO log
  - LOB infrastructure improvements
  - …
- Replication performance improvements
  - WRITESET
- JSON in-place update optimizations

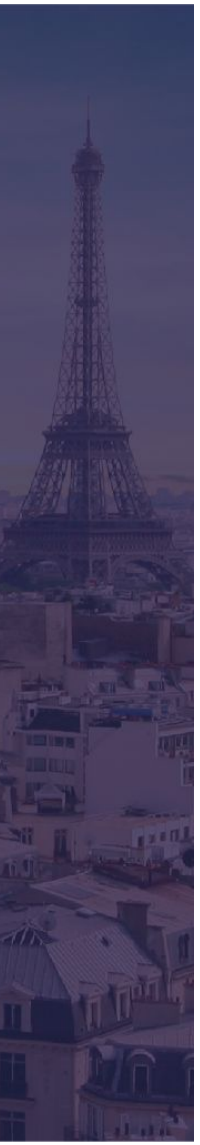


Reasons #19, #20, …

New security features:

- ROLES
- Dynamic privileges
- Partial revokes

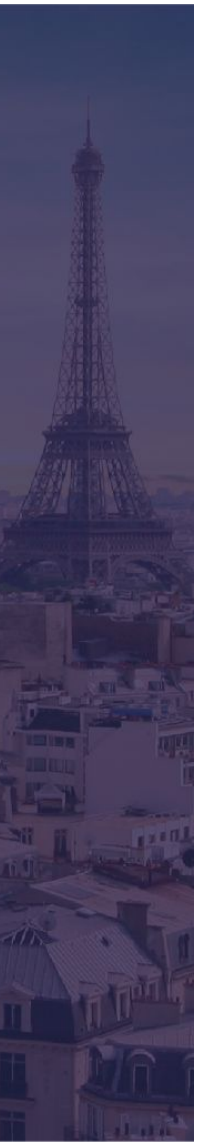
- cached_sha2_password authentication
- Password rotation



Reason #24

Group Replication changes:

- High-availability solution
- Introduced in 5.7
- Lots of improvements in 8.0 to make it mature and reliable



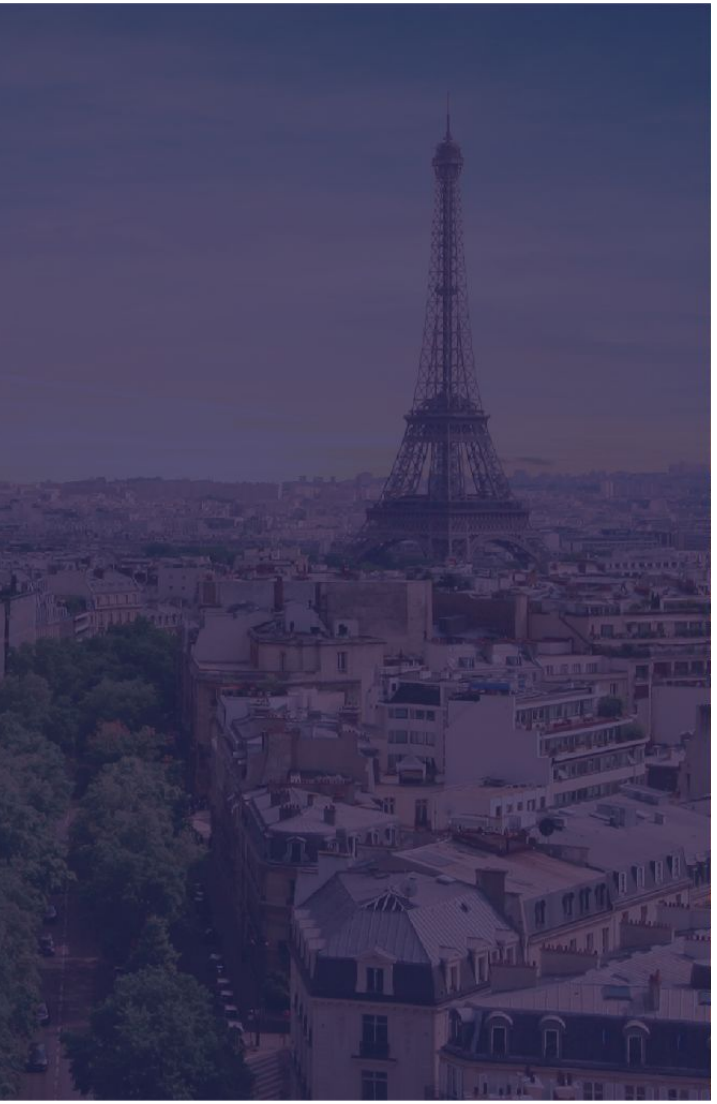
Even more reasons

The preceding list is far from being complete! Read on:

- What Is New in MySQL 8.0

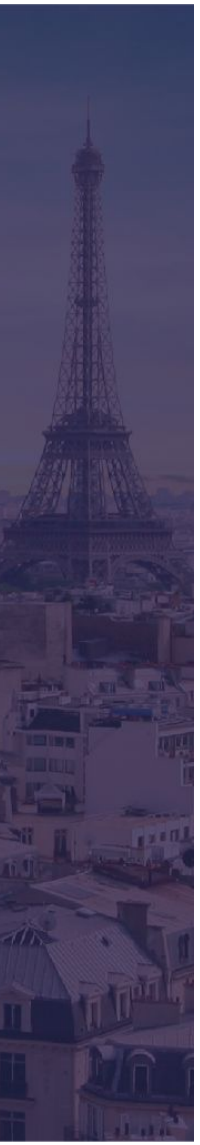
  (https://dev.mysql.com/doc/refman/8.0/en/mysql-nutshell.html)
- The complete list of new features in MySQL 8.0

  (https://dev.mysql.com/blog-archive/the-complete-list-of-new-features-in-mysql-8-0/)
  Mentions 300+ features… Misses changes after January 2019!



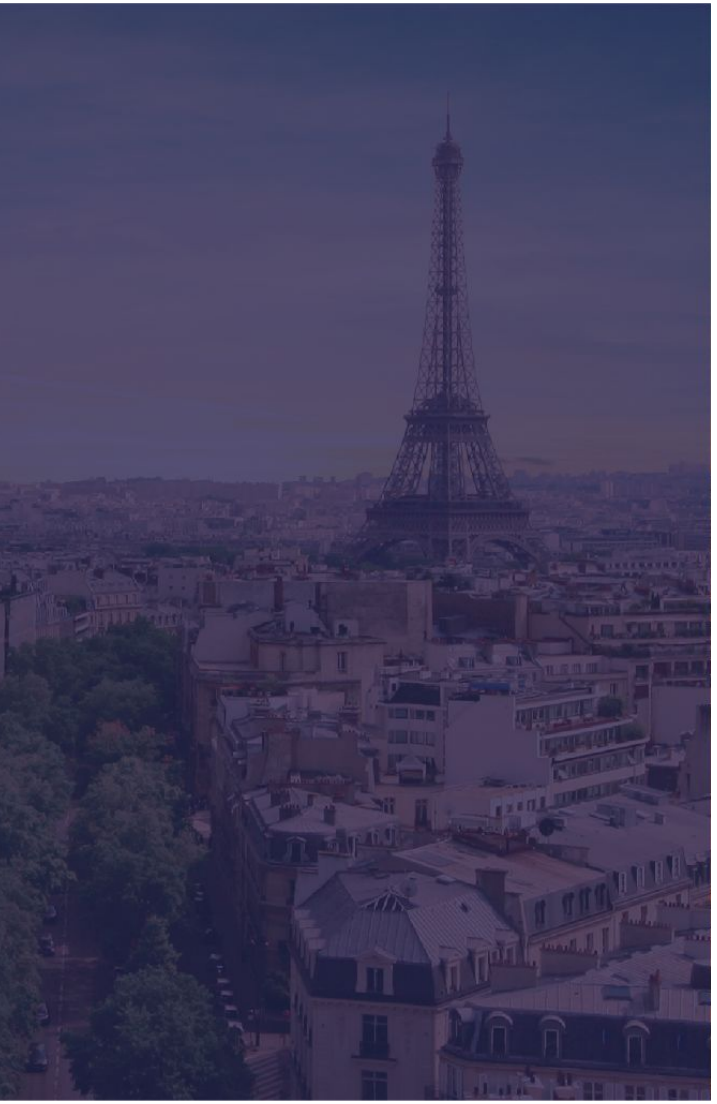
Where is the catch?

(Why some people have not upgraded to MySQL 8.0 yet?)

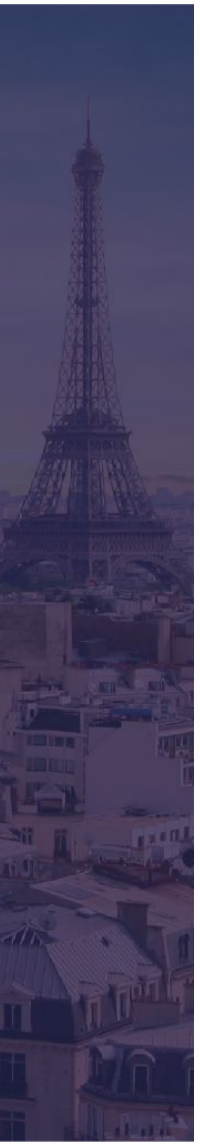


Why people linger with upgrading to MySQL 8.0?

- One way street
- Incompatible changes
  - 20+ new reserved words
- Regressions in execution plans
- Other performance regressions
  - Low concurrency workloads on small servers
  - Cases when 8.0 replicas can’t keep up with primary
  - DDL atomicity comes at a cost

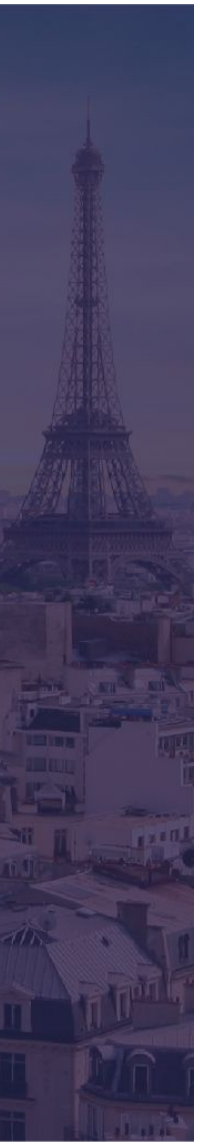


Upgrade problems and ways to solve them



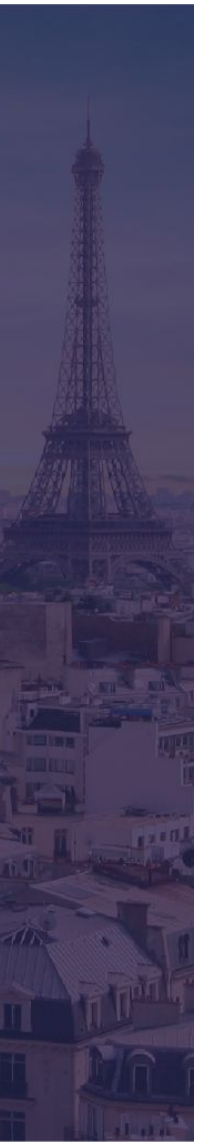
Two types of upgrade

- Logical upgrade
  - Uses SQL dump/restore
  - Not affected by some of upgrade problems
  - Backup as by-product
  - Large storage and CPU overhead
  - More downtime if done in naive way
  - Impractical for medium and large installations due to time required
- In-place upgrade
  - Updates existing data-directory
  - Affected by more upgrade problems (e.g. legacy storage types)
  - Needs separate backup
  - More storage and CPU efficient



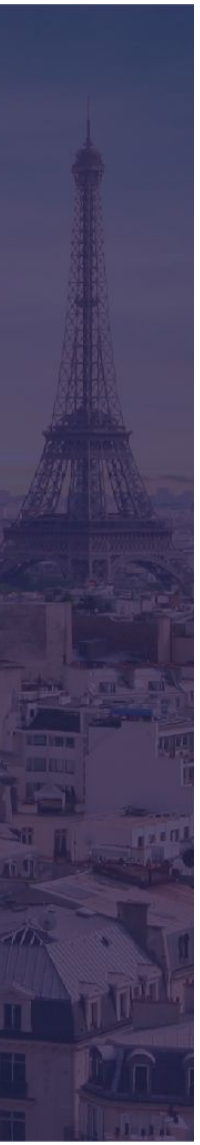
Why upgrade is One Way Street?

- Downgrade is not supported
  - Unlike for minor version upgrade in 5.7 series
  - Downgrade using replication to 5.7 is not officially supported and has problems
- In-place upgrade
  - Converts existing data-directory to use 8.0 data-dictionary
  - Tries to rollback changes on detecting errors so you can start 5.7 again
  - Still a few crashing bugs
    - Backup!!!



Upgrade problems with metadata and data

- Detected during upgrade process
- Some examples:
  - Usage of legacy datetime column types
  - Use of reserved words for table names or in routines
  - Non-native partitioning for tables
  - Orphan entries in InnoDB DD
- Can be detected and corrected before attempting upgrade
  - MySQL Shell Upgrade Checker utility to the rescue
  - Unfortunately not all cases are covered
- Further reading
  - 3.6 Preparing Your Installation for Upgrade (https://dev.mysql.com/doc/refman/8.0/en/upgrade-prerequisites.html)
  - Upgrading to MySQL 8.0? Here is what you need to know… (https://dev.mysql.com/blog-archive/upgrading-to-mysql-8-0-here-is-what-you-need-to-know/)



Post upgrade problems #1

- Changes in options/defaults between 5.7 and 8.0
- Can affect statement semantics and performance
- To some extent can be predicted by reading documentation

Post upgrade problems #2

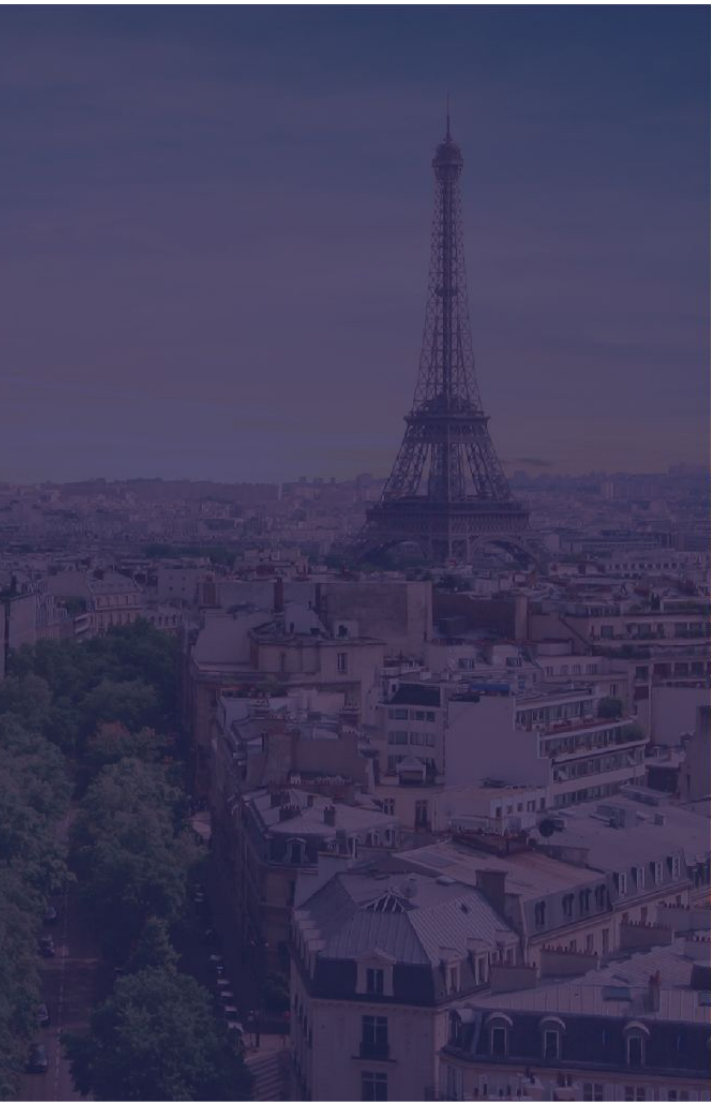
- Problem with execution of statements after upgrade
  - Failures due to incompatible changes
  - Changes in execution plans
  - Performance regressions
- Impossible or hard to predict beforehand

- Detected by using test/benchmark environment
  - Upgrade copy or subset of production system
  - Run real world queries / workload against it

- Ideally run 5.7 and 8.0 in parallel for some time

Post upgrade problems - useful tools from Percona

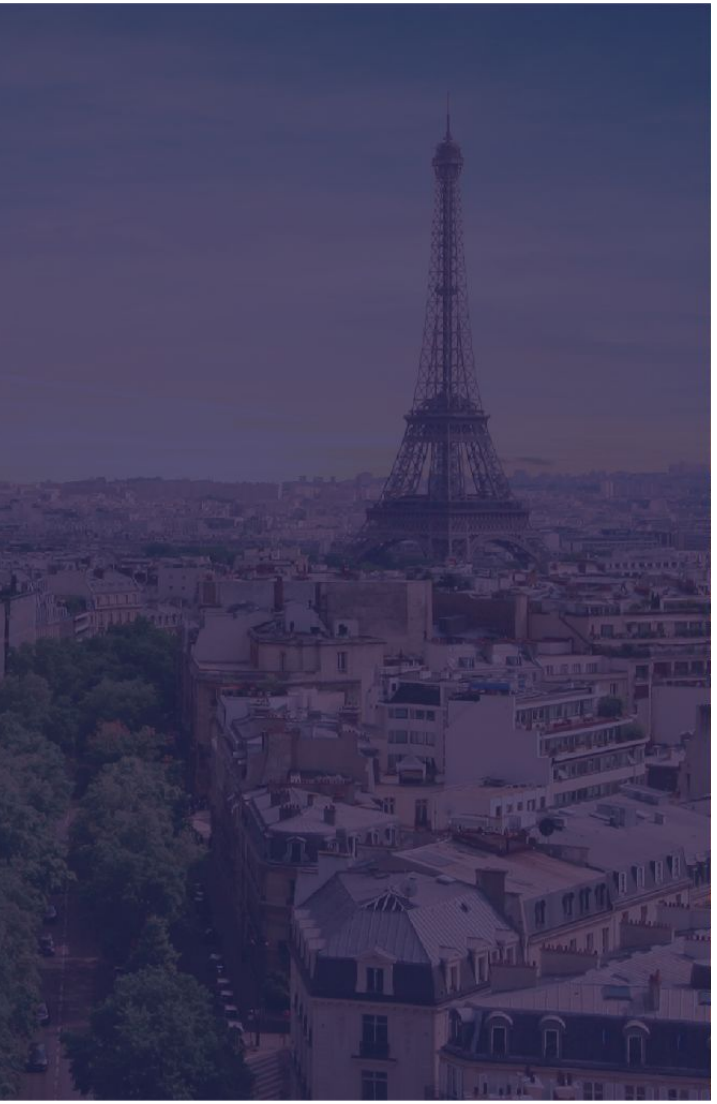
- pt-upgrade - runs statements against different versions and compares results and execution times
- pt-query-digest - useful to gather queries to test before upgrade
- pt-config-diff - can be used to compare configurations before and after upgrade



Upgrade Best Practices

Recap of Upgrade Best Practices

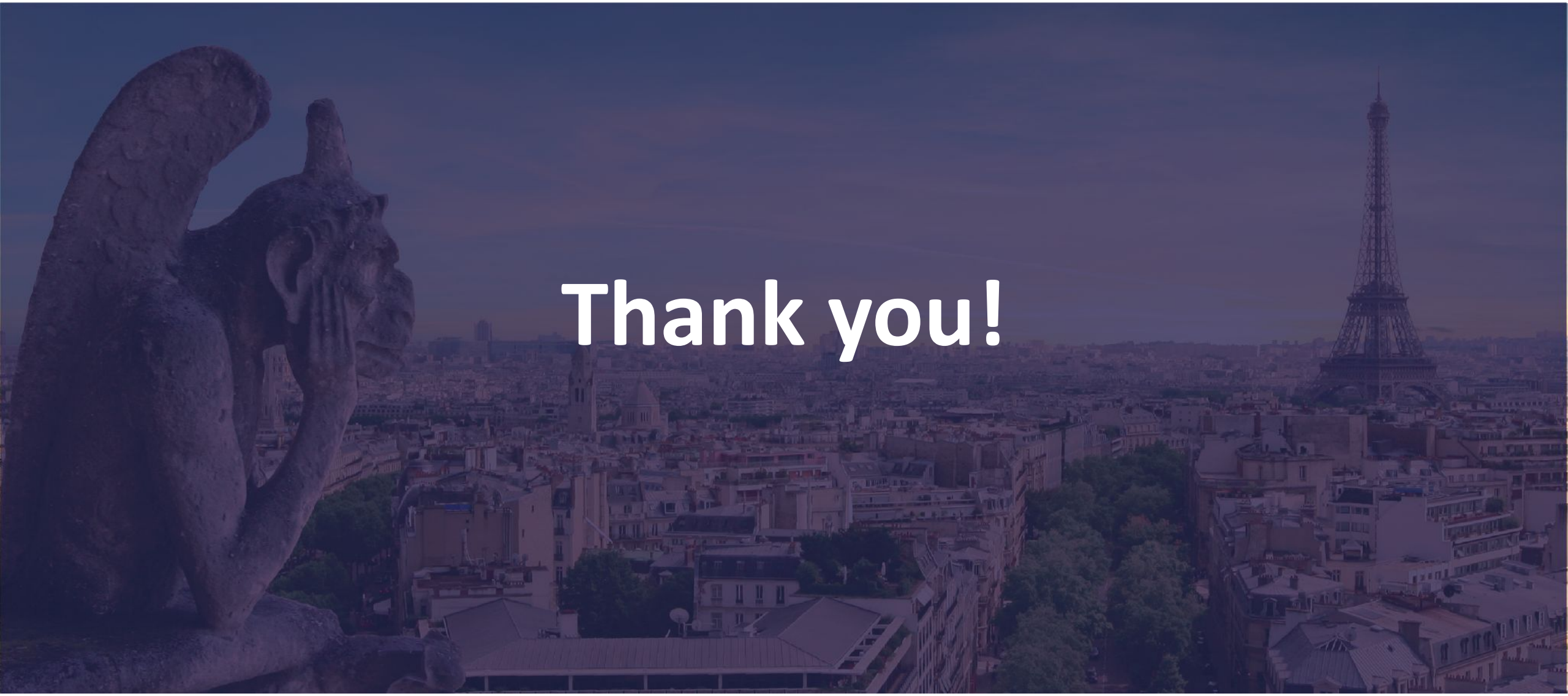
- Plan upgrade (including scenario for the failure case)
- Prepare for upgrade by running Upgrade Checker
- Run application against test environment
- Check statement performance/do benchmarks
- Ideally run old and new versions in parallel for some time
- BACKUP!



Percona University Online has a free course about upgrade:

https://www.percona.com/blog/how-to-upgrade-to-mysql-8-0-free-course-at-percona-university-online/

(also covers replication and rolling upgrades!)



FerretDB

BlaBlaCar