Database requirements for Tiki27

Decision: MariaDB: 10.5+ and MySQL: 8.0+

Tiki27 requires PHP 8.1+ (Released 2021-11). Now, we need to decide minimum versions for MySQL and MariaDB.

MySQL version	Release date	EOL	Notes
~~5.1 LTS~~	~~2008-12~~	~~December 2013~~	
~~5.5 LTS~~	~~2010-12~~	~~December 2018~~	
~~5.6 LTS~~	~~2013-02~~	~~February 2021~~	
5.7 LTS	2015-10	October, 2023	Tiki 21+ Lowest Requirement, Tiki 27 seems to operate here
8.0 LTS	2018-04	Apr 2026	
8.4 LTS	Not Released Yet		

MariaDB version	Release date	EOL	Notes
~~5.5 LTS~~	~~2013-01~~	~~April 2020~~	~~Tiki 21+ Requirement, Does not work with Tiki 27 at least 10.2.2 required~~
~~10.0~~	~~2014-06~~	~~March 2019~~	
~~10.1~~	~~2016-09~~	~~October 2020~~	
~~10.2~~	~~2017-05~~	~~May 2022~~	
~~10.3~~	~~2018-05~~	~~May 2023~~	
10.4	2019-06	June 2024	Tiki 27 seems to operate here
10.5	2020-06	June 2025	
10.6 LTS	2021-07	July 2026	
10.11 LTS	2023-02	February 2028	

OS	Release date	MySQL	MariaDB	PHP
Debian 10	2019-07		10.3.15	7.3
Debian 11	2021-08		10.5.11	7.4
Debian 12	2023-06		10.11.3	8.2
Ubuntu 20.04	2020-04	8.0.19	10.3	7.4
Ubuntu 22.04	2022-04	8.0.36	10.6	8.1
Ubuntu 24.04	2024-04	8.0.36	10.11.6	8.3
RHEL/CentOS 7.x	2014-06		5.5.68	5.4
RHEL 8.x	2019-05	8.0.32	10.5.16	8.0
RHEL 9.x	2022-05	8.0.32	10.5.16	8.1

Related links

- https://dev.mysql.com/blog-archive/introducing-mysql-innovation-and-long-term-support-lts-versions/

- https://mariadb.com/kb/en/mariadb-vs-mysql-compatibility/
- Requirements
- https://gitlab.com/tikiwiki/tiki-manager/-/merge_requests/456#note_1807201425
- https://gitlab.com/tikiwiki/tiki-manager/-/blob/master/config/tiki_requirements.yml
- https://distrowatch.com/table.php?distribution=ubuntu
- https://distrowatch.com/table.php?distribution=debian
- https://distrowatch.com/table.php?distribution=redhat
- https://gitlab.com/tikiwiki/tiki/-/blob/master/.gitlab-ci.yml
- MariaDB vs MySQL strategy
- Database independence
- Database independence

Proposals

Marc Laporte

- Based on the information above, I propose to set minimums to MySQL 8.0.x and MariaDB 10.5. Ex.: Debian 11 with a more recent PHP via https://deb.sury.org/ Anyone on Debian 10 or Ubuntu 20.04 can use Tiki 24x LTS.

Drsassafras

When choosing a version I think of the following factors:

1. How the version impacts what systems Tiki can be run on & how easy it is to get it running. (Shared hosting support, different os support etc.) Direction: **MySQL 5.7 MariaDB 10.4**

2. How the database version impacts programming. eg. If we set the version too low, it may discourage new features in the language from being used, as not to "break" the requirements. It might also complicate code if separate code is needed to handle different versions. Direction: **MySQL 8.0 MariaDB 10.11**

3. How the decision impacts support. eg. there is a regression discovered that only affects an old db version. Direction: **MySQL 8.0 MariaDB 10.11**

4. LTS cycle. It's best that version requirement changes are made early in the cycle. This is difficult, but if done well results a more stable LTS and also allows new Tiki projects staring with non-LTS and eventually staying on the upcoming LTS an easy transition. Direction: **MySQL 5.7 MariaDB 10.4**

5. Security. Simplicity of code often means more secure code. Additionally encouraging the use of software that is no longer receiving security updates may cause people more trouble than wide language support is worth. "It sounded like a good idea at the time" We will support tiki 27 until 2029, with is farther in the future than any current LTS version of MariaDB or MySQL. So I will pick security recommendations based on at least 1 year of DB security support from the time we release Tiki 27. Direction: **MySQL 8.0 MariaDB 10.5**

Database options, including the choice of database and which version is run is largely out of the hands of anyone running Tiki on shared hosting. For this community reason I think we should not remove database versions that might otherwise work without good reason.

I have updated the charts above to include EOL info & some notes of consideration. I would also advocate for Tiki 28 to only support the most up to date LTS version of MySQL and MariaDB, that way by the time our next LTS comes around we can hopefully keep the same database requirements and also have a version that is not totally out of support. Although we are now picking a DB version at the time of LTS release and none of them will be fully supported throughout our lifecycle, which is partly why I think we

need to pick only the latest versions in our first post-LTS release. In addition our programmers will have had a few years of coding to the syntax that our LTS will be using and will have had the opportunity to add better code during that time, during refactors & new features. It provides an easy update train for those moving into a LTS version model. I am of split mind between thinking we should take the security recommendation of 8.0 & 10.5, and keeping the basic requirement of 5.7 & 10.4 (using 10.4 as 10.2.2 is not an LTS), but my mind is firm on 28 using the most recent release only. I think it will result in a more stable code, with less effort, and we probably have to do it do it anyway when we come around to the next LTS.