



# MariaDB Catalogs

**October, 2023**

**Michael Widenius**

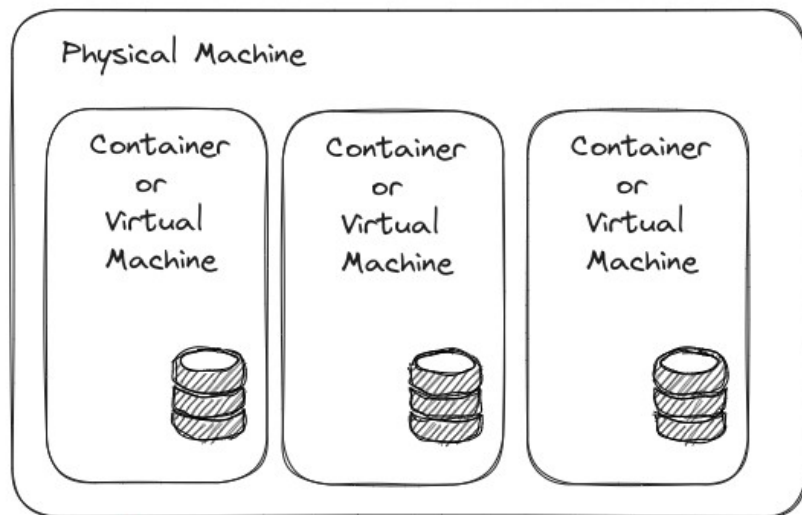
**Creator of MySQL and MariaDB**

**CTO @ MariaDB**

# Database scaling for SaaS providers

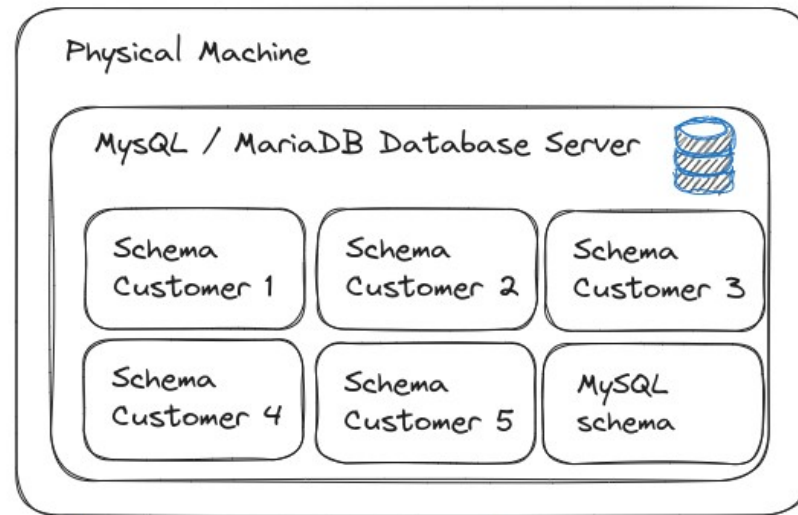
- SaaS providers want to optimize the number of customers they can host on their hardware.
- When offering managed services on MySQL/MariaDB, SaaS providers have the following options:
  - Create separate VMs for each customer.
    - Large overhead – costly (~100 servers/tenants per server)
    - A typical “idle” MariaDB Server requires ~1GB of memory
    - Best user experience
  - Use shared instance
    - Force user restrictions, only grant limited database access.
      - cPanel has this model, many other database management systems share it
    - Very limited overhead
    - Poor user experience, artificial limitations enforced on users
      - Typically one user, one database. Tenant cannot choose the names.
    - Affected by “noisy neighbour”, hard to track

### Container / VM Approach



1 GB minimum per VM

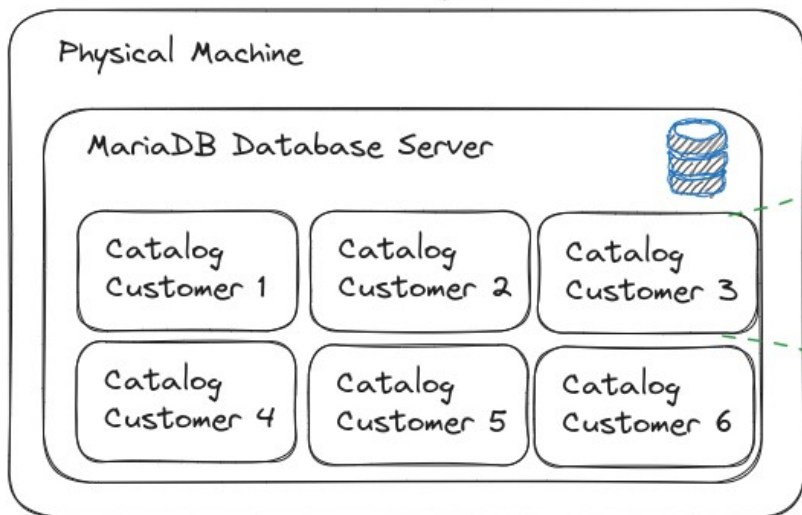
### Shared Server / Separate Schema



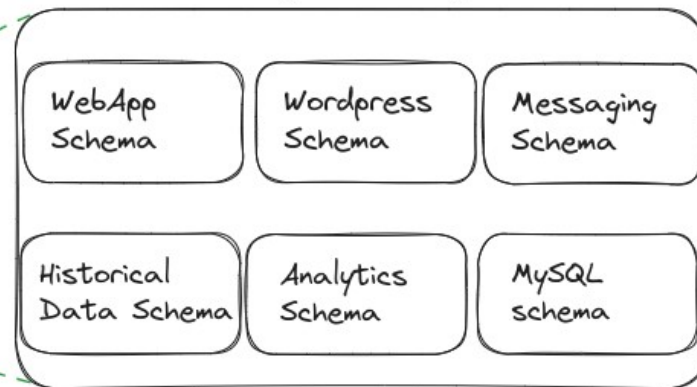
Customers have limited schemas



### Catalogs



### Catalog For a Single Customer



# New catalogs feature for MariaDB

- Catalogs provide the best of both worlds.
  - Shared instance
  - A catalog/tenant that looks like a normal MariaDB Server
  - No user limitations
    - Each tenant has full control over their catalog
    - Root/super access on the catalog
    - Dedicated general and slow query log
- Catalogs still have the problem of noisy neighbour
  - However, statistics are now collected for each catalog
  - So it's trivial to detect the problematic catalog
  - Move the problematic tenant to a bigger machine, increase costs
- Catalogs can also enforce quotas
  - Each catalog have their own configuration file with their own limits
  - Force users to stick to certain performance limitations.
- Sysadmins have control over all catalogs.

# Catalogs feature

- Catalogs enable hyper-scaling for SaaS providers.
  - For basic users, one can now host up-to 100x more users on the same machines.
- In MariaDB, this feature is still under development
  - If you would like to steer the roadmap for this, now is the time!
    - Tooling
    - Performance optimizations
    - Specific functionality

# Accessing a catalog

- Each tenant is assigned one catalog
- Connecting to a catalog can be done with:
  - `mariadb -catalog ...`
  - Connect with catalog as part of database: **`mariadb catalog.test`**
  - Connect by hostname: **`mariadb -hostname=unique_ip`**
  - Connect by port: **`mariadb --hostname=xxx -port=xxx`**
  - Options can store the options in their `my.cnf` file.
- Users can use `mariadb-dump` to copy all their databases and tables (including the 'mysql' database) to a catalog or from a catalog.
- A tenant can only see objects (databases, tables, users etc) from their catalog. They cannot access or move to other catalogs.
- A tenant can ask the SAS provider to replicate:
  - On premise MariaDB server → Catalog
  - Catalog → Catalog or from
  - Catalog → On premise MariaDB.

# Commands only available for “catalog root user”

The ‘catalog root user’ is the super user of the ‘def’ catalog

- SHUTDOWN command
- Setting up replication (MASTER and SLAVE commands)
- SHOW SERVER STATUS will show status data for the whole server
- SET CATALOG catalog\_name; (used to change to another catalog)
- DROP CATALOG
- Load plugins & User defined functions.

# Limitations with catalogs

- Catalog names are limited to 65 ASCII characters (for now)
- Errors from background task (like write error) will be logged into the system error log, not the catalog error log.

Tables that are only read or used from the 'def.mysql' schema:

- plugin
- help\_\* tables
- time\_zone\* tables
- gtid\_slave\_pos (replication state)
- innodb\_index\_stats (innodb internal, to be moved to catalogs)
- servers (federated)
- transaction\_registry (innodb internal)
- func (udf)
- performance\_schema (will filter out other catalogs)



# Current state of catalogs

- <https://mariadb.com/kb/en/catalogs-overview>
- MDEV-31542 Add multi-tenancy catalogs to MariaDB
  - Contains all open tasks for catalogs (~20 tasks)

# Starting with catalogs

Catalogs are created with

- `mariadb_install_db -catalogs="cat1 cat2 dog1 dog2"`
  - This creates the 'def' catalog with base tables and each catalog with their own 'mysql' database.
- One can add new catalogs later with the same command.
  - When adding new catalogs, `mariadb_install_db` will connect to the running server and create the new catalogs

MariaDB server will automatically notice if catalogs exists and start in 'catalog mode' if needed.

# Catalog commands

- `USE CATALOG catalog_name;`
- `CREATE CATALOG`
- `DROP CATALOG` (Not yet fully implemented)
- `ALTER CATALOG`
- `SHOW CATALOGS` (and also `information_schema.catalogs`)
- `SHOW CREATE CATALOG catalog_name`
- `SELECT CATALOG()`

# Catalog status commands

- Show the status for the whole server. Only for the “catalog super user”:
  - SHOW SERVER STATUS
- Show the status for the current catalog:
  - SHOW GLOBAL STATUS;
  - SHOW CATALOG STATUS;
- Show the status for the current connection:
  - SHOW [SESSION] STATUS;
- When the MariaDB server is not configured for catalogs, the following commands are equivalent:
  - SHOW GLOBAL STATUS
  - SHOW SERVER STATUS
  - SHOW CATALOG STATUS

# Implemented features

- Support for catalog in server connect protocol
- Support for catalogs in connector C:
  - `mysql_optionsv(mysql, MARIADB_OPT_CATALOG, catalog);`
- mariadb client now supports `-catalog` option
- mariadb-binlog now has `USE CATALOG=xxx` commands in the output
- `thd_catalog_service()` (for plugins)
- Per catalog status variables
- Removed 'global system tables' from catalog/mysql database
- mysql privilege tables per catalog

# Implemented features

Added support for catalogs to:

- mariadb-hotcopy
- ddl-log (crash recovery of DDL's: ddl\_recovery\_log)
- Events
- Backup DDL log (ddl.log)
- Storage engines: InnoDB, MyISAM, Aria, connect
- MDL locks
- Query cache
- mysql-test-run (mtr) to work with catalogs:
  - mtr –catalogs now works! (running all tests within the 'def' catalog)
  - We still need to add more tests for catalogs itself (suite/catalogs)

# Notable not implemented features

- System variables per catalog
  - Each catalog will have their own 'my.cnf' file in the catalog directory
- Testing (we need catalog specific tests for most important features)
- Catalogs for rocksdb
- Catalog specific tmdir & tmpdir statistics
- Catalogs in all performance schema
- Galera
- Update all standard mariadb clients to support -catalog option
  
- Next stage for catalogs:
  - Add more support for constraining tenants to not cause problems for others.
  - cgroups, constraining memory, disk space etc



---

**Thank you**