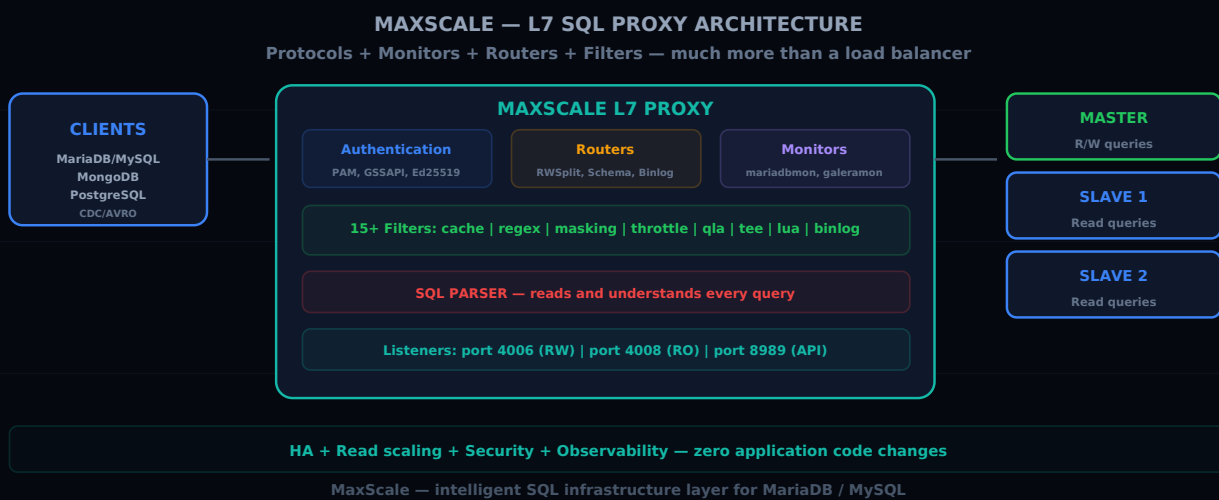


# MaxScale : bien plus qu'un reverse proxy SQL (partie 1)

Sylvain ARBAUDIE · 31 juillet 2025

MAXSCALE MARIADB PROXY ARCHITECTURE



## Pas un simple proxy

Quand on parle de MaxScale, le réflexe est de le comparer à HAProxy ou à un load balancer classique. C'est une erreur fondamentale. MaxScale n'est pas un proxy de niveau 4 (TCP). C'est un **proxy de niveau 7** qui comprend le protocole SQL. Il parse les requêtes, analyse leur type, et prend des décisions de routage intelligentes.

La différence est comparable à celle entre un facteur qui distribue le courrier par adresse (L4) et un assistant qui lit vos emails, les trie par priorité, filtre le spam, et ne vous montre que ce qui est pertinent (L7). MaxScale lit le SQL.

## Authentification : au-delà du mot de passe

MaxScale supporte une gamme complète de mécanismes d'authentification :

- **MariaDBAuth** : l'authentification native MariaDB / MySQL avec cache local des credentials
- **PAM** : intégration avec les annuaires d'entreprise (LDAP, Active Directory) via Pluggable Authentication Modules

- **GSSAPI/Kerberos** : authentification SSO pour les environnements Windows / Active Directory
- **Ed25519** : l'authentification par clé publique de MariaDB, plus sécurisée que le hash SHA classique

L'authentification est gérée au niveau du listener. On peut donc avoir différentes méthodes d'authentification sur différents ports : PAM pour les applications internes, Ed25519 pour les connexions administratives, MariaDBAuth pour la compatibilité legacy.

## Protocoles : pas seulement SQL

---

MaxScale est polyglotte. Il supporte plusieurs protocoles d'entrée et de sortie :

### MariaDBClient / MariaDBBackend

Le protocole natif MariaDB / MySQL. C'est le cas d'usage principal : les applications se connectent à MaxScale comme à un serveur de base de données standard.

### CDC / AVRO

Le protocole Change Data Capture permet de streamer les modifications du binlog en format AVRO. C'est l'outil idéal pour construire des pipelines de données en temps réel, alimenter des data lakes ou synchroniser des systèmes externes.

### NoSQL / MongoDB

MaxScale peut exposer une interface compatible MongoDB. Les applications qui parlent le protocole MongoDB peuvent interagir avec une base MariaDB via MaxScale. C'est une fonctionnalité de niche mais puissante pour les migrations.

### PostgreSQL (expérimental)

Le support du protocole PostgreSQL en entrée est en développement. L'objectif est de permettre à des applications PostgreSQL de se connecter à des backends MariaDB.

## Moniteurs : l'intelligence de la topologie

---

Les moniteurs sont les yeux de MaxScale. Ils interrogent régulièrement les serveurs backend pour détecter automatiquement la topologie et l'état de santé du cluster.

## **mariadbmon**

Le moniteur principal pour les topologies de réplication MariaDB / MySQL. Il détecte le master, les slaves, les relais, et l'état de la réplication. Il gère le failover automatique : si le master tombe, un slave est promu, et les autres slaves sont reconfigurés pour répliquer depuis le nouveau master.

## **galeramon**

Le moniteur dédié aux clusters Galera. Il détecte l'état du cluster (Primary, Non-Primary, Disconnected), le nombre de nœuds, le state UUID, et gère le routage en conséquence.

## **Routeurs : l'intelligence du routage**

---

Les routeurs sont le cerveau de MaxScale. Ils décident où envoyer chaque requête en fonction de son type et de la topologie détectée.

### **readwritesplit**

Le routeur le plus utilisé. Les requêtes en écriture ( `INSERT` , `UPDATE` , `DELETE` , `CREATE` , etc.) sont envoyées au master. Les requêtes en lecture ( `SELECT` ) sont distribuées sur les slaves. Les transactions sont envoyées intégralement au master.

### **readconnroute**

Un routeur plus simple qui distribue les connexions (pas les requêtes) sur les serveurs disponibles. Utile pour les lectures intensives qui ne nécessitent pas de séparation lecture/écriture au niveau de la requête.

### **schemarouter**

Route les requêtes vers le serveur qui héberge le schéma demandé. Idéal pour le sharding par base de données : `client_europe` sur le serveur A, `client_asia` sur le serveur B.

### **binlogrouter**

Transforme MaxScale en relais de réplication. MaxScale se comporte comme un slave du master, et les vrais slaves se connectent à MaxScale plutôt qu'au master. Cela réduit la charge sur le master et permet de centraliser la distribution du binlog.

### **kafkarouter (CDC)**

Envoie les événements du binlog vers Apache Kafka. Chaque modification de la base est publiée comme un message Kafka, permettant aux consommateurs de réagir en temps réel.

## Filtres : plus de 15 modules

---

Les filtres interceptent et transforment le flux SQL entre le client et le serveur. MaxScale en propose plus de 15 :

- **qlafilter** : journalisation complète des requêtes (audit)
- **regexfilter** : réécriture de requêtes par expression régulière
- **cache** : cache de requêtes avec invalidation automatique
- **throttlefilter** : limitation du débit de requêtes par utilisateur
- **masking** : masquage dynamique des données sensibles (emails, numéros de carte)
- **topfilter** : collecte des requêtes les plus lentes
- **commentfilter** : injection de commentaires SQL pour le tracing
- **tee** : duplication du flux vers un second backend (test en shadow)
- **namedserverfilter** : routage vers un serveur spécifique basé sur des règles
- **hintfilter** : interprétation de hints SQL pour forcer le routage
- **luafilter** : exécution de scripts Lua pour une logique personnalisée
- **binlogfilter** : filtrage des événements binlog par schéma ou table

## Déploiement : Docker et Kubernetes ready

---

MaxScale est disponible en image Docker officielle :

```
docker run -d --name maxscale \  
-v /path/to/maxscale.cnf:/etc/maxscale.cnf \  
-p 4006:4006 -p 8989:8989 \  
mariadb/maxscale:latest
```

Pour Kubernetes, MaxScale se déploie comme un StatefulSet ou un Deployment selon le cas d'usage. L'API REST facilite l'intégration avec les health checks et les probes de readiness.

## Valeur métier

---

Au-delà des fonctionnalités techniques, MaxScale apporte une valeur métier concrète :

- **Haute disponibilité** : failover automatique en quelques secondes, transparent pour l'application
- **Scalabilité en lecture** : ajout de slaves sans modifier le code applicatif
- **Sécurité** : filtrage SQL, masquage de données, authentification centralisée
- **Observabilité** : journalisation, métriques, API REST pour le monitoring
- **Migration facilitée** : support multi-protocoles pour les transitions technologiques

MaxScale n'est pas un simple reverse proxy. C'est une couche d'infrastructure intelligente qui s'interpose entre les applications et les bases MariaDB / MySQL, apportant résilience, sécurité et flexibilité sans modifier une seule ligne de code applicatif.

---

Cet article a été initialement publié sur [Medium](#).