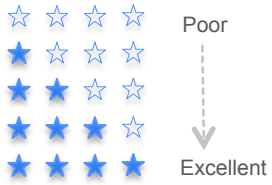
Competitive Scorecard



MySQL clustering for HA, DR, multimaster and geo-scaling

Top prioritized use cases	Continuent Clustering	AWS Aurora with Read Replica	Google Cloud SQL HA	Orchestrator + Proxy	Galera Cluster
Local HA	Yes, automatic failover for primary and secondaries ★★★★	Yes, 30 seconds with DNS change ★★★☆	Just one replica, at least 60 seconds for failover ★☆☆☆	Yes – no auto-recovery of old master in switch ★★☆☆	Only supported with separate HA proxy ★☆☆☆
DR (local/ remote)	Yes, local, remote and global DR are supported ★★★★	Failover only, takes "a few minutes," replication breaks ★★☆☆	Not supported ☆☆☆☆	Yes – no auto-recovery of old master in switch ★★☆☆	Support with async but without management ★☆☆☆
Load balancing	Yes, automatic and transparent ★★★★	Supported but requires application awareness ★★☆☆	Supported but requires application awareness ★★☆☆	Yes, automatic, transparent ★★★★	Automatically to multiple primaries ★★★☆
Zero downtime maintenance	Enables both database, operating system and hardware changes ★★★★	Requires maintenance window, downtime ★☆☆☆	Requires maintenance window, downtime ★☆☆☆	Almost, needs manual reconfiguration ★★☆☆	Changes are supported, but can be costly and time consuming ★★☆☆
Performance	High, built-in load balancing and read/write splitting ★★★★	High performance but read/write splitting requires app aware ★★★☆	Requires read replicas, app awareness, replication lag ★★★☆	High performance ★★★★	Synchronous nature implies additional overhead during writes ★☆☆☆
Transparency	Yes, automatic failover for primary and secondaries ★★★★	Automatic failover but applications disconnect ★★☆☆	Automatic failover but applications disconnect ★★☆☆	Manual reconfiguration needed, disconnections ★★☆☆	Requires cluster and app made aware of changes ☆☆☆☆

Competitive Scorecard

MySQL clustering for HA, DR, multimaster and geo-scaling

Top prioritized use cases	Continuent Clustering	AWS Aurora	Google Cloud SQL HA	Orchestrator + Proxy	Galera Cluster
Deployment	Easy to deploy ★★★★	Easy to deploy ★★★★	Easy to deploy ★★★★	Globally easy ★★★☆	Complex, easier with 3rd party ClusterControl ★☆☆☆
Recovery	Fast and simple, local and cross region ★★★★	Local recovery simple, cross region takes time ★★★★	Local recovery simple ★★★★	Simple as long as automation works ★★★☆	Difficult or time consuming ★☆☆☆
Monitoring	Full monitoring via web-based GUI ★★★★	Full monitoring, Enhanced Monitoring is extra cost ★★★☆	Easy to use basic monitoring ★★★☆	Full monitoring on web interface ★★★☆	Only with 3rd party tools like ClusterControl ★☆☆☆
Geo-Scale	Across regions, availability zones ★★★★	Only basic read replicas in AWS regions ★☆☆☆	Not supported ☆☆☆☆	Yes ★★★★	Async but without management layer ★☆☆☆
Replication	To most popular high-performance NoSQL and data analytics engines ★★★★	Basic MySQL replication only, manual setup (5.5 or later) ★☆☆☆	Basic MySQL replication only, manual setup (5.5 or later) ★☆☆☆	Basic ★☆☆☆	Synchronous locally, async globally, manual setup ★★☆☆
Support	24/7 with <5 min. initial response to urgent support tickets ★★★★	Basic support > 12 hours, critical support extremely expensive ★☆☆☆	Not included, critical support extremely expensive ☆☆☆☆	Only through mailing list/ forums ★☆☆☆	24/7 through Percona ★★★☆