MySQL Cluster Database FAQ - General

1. What is MySQL Cluster?

A: MySQL Cluster is built on the NDB storage engine and provides a highly scalable, real-time, ACID-compliant transactional database, combining 99.999% availability with the low TCO of open source. Designed around a distributed, multi-master architecture with no single point of failure, MySQL Cluster scales horizontally on commodity hardware to serve read and write intensive workloads, accessed via SQL and NoSQL interfaces.

MySQL Cluster's real-time design delivers predictable, millisecond response times with the ability to service millions of operations per second. Support for in-memory and disk-based data, automatic data partitioning (sharding) with load balancing and the ability to add nodes to a running cluster with zero downtime allows linear database scalability to handle the most unpredictable web-based workloads.

2. What is MySQL Cluster Carrier Grade Edition?

A: MySQL Cluster Carrier Grade Edition (CGE) includes tools for the management, monitoring security and auditing of the MySQL Cluster database, coupled with access to Oracle Premier Support. MySQL Cluster CGE is available under a choice of subscription or commercial license and support.

- Learn more about MySQL Cluster CGE

3. Who are the customer references for MySQL Cluster?

A: See https://www.mysql.com/customers/cluster/.

4. What is the current version of MySQL Cluster?

A: The current GA version is MySQL Cluster 7.5. MySQL 5.7 is integrated and bundled with MySQL Cluster.

5. Does MySQL Cluster require any special hardware or software?

A: No, MySQL Cluster is designed to run on commodity hardware. Using specialized hardware such as Infiniband network interconnects one can achieve even higher levels of performance, especially over large clusters with many nodes.

6. What are the system requirements for MySQL Cluster?

	OS:	See current list of Supported Platforms »
	CPU:	Intel Xeon E5-2600 v4 (20+ cores/socket)
A:	Memory:	64GB RAM
	Storage:	512GB SSD
	Network:	1+ nodes (Gigabit Ethernet - TCP/IP)

7. How do I qualify a workload as being a good fit for MySQL Cluster?

A: If you answer "YES" to any of the following questions, then you should consider MySQL Cluster as an option for your application's database:

- Do you need to shard your database to meet growing volumes of write (UPDATE, INSERT, DELETE) operations?

- Do you need to ensure results from SELECT operations are consistent, regardless of which node they are returned from?
- Would a failure of your database result in application downtime that would cause business disruption, including loss of revenue, loss of reputation, etc?
- Would data loss (even if just several seconds worth) during a failover cause business disruption?
- Is your user experience sensitive to response times?
- Do you need to replicate your database across geographic regions, with each region serving both read and write operations?
- Are you running a diverse range of applications that would benefit from direct access to data, without always relying on SQL (ie JavaScript with node.js, memcached API, Java and JPA applications, HTTP/REST web services and C++ apps)?
- Does your application primarily consist of "short" transactions (ie 10s of operations per transaction versus 1000s) executed in parallel?
- Does your application mainly consist of:
    - Primary key database access, with some JOIN operations
    - versus
    - Regular execution of full table scans and JOINs returning 10s of thousands of rows?

Please also review our Evaluation Guide to learn more about MySQL Cluster.

8. What are the ideal applications for MySQL Cluster?

A: Ideal applications include:

- High volume OLTP
- Real time analytics
- Ecommerce and financial trading with fraud detection
- Mobile and micro-payments
- Session management & caching
- Feed streaming, analysis and recommendations
- Content management and delivery
- Massively Multiplayer Online Games
- Communications and presence services
- Subscriber/user profile management and entitlements

See a full list of MySQL Cluster user case studies and applications.

9. Is MySQL Cluster supported on Virtual Machine environments?

A: Yes. MySQL Cluster is tested and certified on Oracle VM.

10. What are the typical performance metrics for MySQL Cluster?

A:

- Availability
    - 99.999% (<5 min downtime/year)
- Performance
    - Response Time: sub 5 millisecond (with synchronous replication and access via SQL). Faster response times can be achieved using one of the NoSQL access methods.
    - Throughput of 600,000+ replicated UPDATE operations/sec on a dual-socket Intel server equipped with 64GB RAM. 1 Billion UPDATE operations per minute across a cluster of 30 x Intel Servers. See full benchmarks.
- Failover

        - Sub-second failover enables you to deliver service without interruption
- Scalability
    - Scale out, scale up and scale dynamically
    - For cost-effective scale-out:
        - Add more application and data nodes per cluster, or
        - Add more CPU threads (16, 32, 64, etc.) or
        - Add more Memory (32GB, 64GB, etc.) per data node

11. How many physical servers are needed to create a minimum Cluster configuration?

A: For evaluation and evelopment purposes, you can run all nodes on a single host. For full redundancy and fault tolerance, you would need a minimum 6 x physical hosts:

- 2 x data nodes
- 2 x SQL/NoSQL Application Nodes
- 2 x Management Nodes

Many users co-locate the Management and Application nodes which reduces the number of nodes to four.

12. Can Data Nodes be split geographically?

A: Yes, as long as your network has the characteristics discussed here

MySQL Cluster has long offered Geographic Replication, distributing clusters to remote data centers to reduce the affects of geographic latency by pushing data closer to the user, as well as providing a capability for disaster recovery.

Geographic replication is asynchronous and can be implemented as an Active / Active or Active / Passive cluster.

Geographic replication is the recommended deployment model for cross data center deployments.

13. What data access APIs exist for MySQL Cluster?

A: Applications can be developed using any MySQL Connectors. MySQL Cluster additionally provides native NoSQL connectivity via JavaScript, Memcached, C++, Java, JPA and HTTP/REST.

14. Are the interfaces for 32-bit applications different from 64-bit?

A: No, the interfaces are the same.

15. Is MySQL Cluster suitable as an embedded database?

A: Yes, MySQL Cluster is often used as an embedded database by ISVs and Network Equipment Providers (NEPs). Customer List »

16. What is Geographic Replication for MySQL Cluster?

A: Geographic replication enables asynchronous replication across active / active geographically separate clusters. This is often used for the scale-out of global services, data locality and disaster recovery.

17. Is Replication bi-directional?

A: Yes, unidirectional and bi-directional replication are supported in MySQL Cluster. Transaction conflict detection and resolution is provided when using bi-directional geographic replication.

18. When using MySQL Cluster as an in-memory database, does MySQL Cluster risk losing data?

A: MySQL Cluster configurations will typically have at least 2 copies of all data, held on different hosts. To cover total system failure, transaction logs and checkpoint files are persisted on disk with configurable frequency. Additionally, non-indexed data may be stored on disk.

19. Does MySQL Cluster include a diskless option?

A: MySQL Cluster has a diskless option as well as a no-logging option.

For the Diskless option the following restrictions apply:

1. No disk data
2. Loss of data in case of Cluster failure
3. No backup

For the no logging option, Cluster will still create log files, but data will not be checkpointed to disk.

MySQL Cluster Manager FAQ

20. Is MySQL Cluster Manager open source software?

A: No. MySQL Cluster Manager is available only as a part of the commercial MySQL Cluster Carrier Grade Edition (CGE) database. To purchase subscriptions or licenses for MySQL Cluster CGE, please contact the MySQL Sales Team.

21. What is MySQL Cluster Manager?

A: MySQL Cluster Manager is software which simplifies the creation and management of the MySQL Cluster database by automating common management tasks.

22. What are the benefits of MySQL Cluster Manager?

A: By using MySQL Cluster Manager, Database Administrators (DBAs) and Systems Administrator are more productive, enabling them to focus on strategic IT initiatives and respond more quickly to changing user requirements. At the same time, risks of database downtime that previously resulted from manual configuration errors, are significantly reduced.

23. Can you give me a practical example of where MySQL Cluster Manager would help with productivity and reduce risk of downtime?

A: As an example, management operations requiring rolling restarts of a MySQL Cluster database that previously demanded 46 manual commands[1] and which consumed 2.5 hours of DBA time[2] can now be performed with a single command, and are fully automated with MySQL Cluster Manager, serving to reduce:

- Management complexity and overhead;
- Risk of downtime through the automation of configuration and change management processes;
- Custom scripting of management commands or developing and maintaining in-house management tools.

24. What sort of management functionality does MySQL Cluster Manager provide?

A: Administrators are able to create and delete entire clusters and start, stop and restart the cluster with a single command, as well as add nodes on-line. As a result, administrators no longer need to manually restart each data node in turn, in the correct sequence, or to create custom scripts to automate the process.

MySQL Cluster Manager automates on-line management operations, including the upgrade, downgrade and reconfiguration of running clusters, without interrupting applications or clients accessing the database. Administrators no longer need to manually edit configuration files and distribute them to all other cluster nodes, or to determine if rolling restarts are required. MySQL Cluster Manager handles all of these tasks, thereby enforcing best practices and making on-line operations significantly simpler, faster and less error-prone.

25. Does MySQL Cluster Manager manage the entire cluster or just individual nodes within it?

A: It can do both. MySQL Cluster Manager provides the ability to control the entire cluster as a single entity, while also supporting very granular control down to individual processes within the cluster itself.

26. What sort of monitoring functionality does MySQL Cluster Manager provide?

A: MySQL Cluster Manager is able to monitor cluster health at both an Operating System and per-process level by automatically polling each node in the cluster. It can detect if a process or server host is alive, dead or has hung, allowing for faster problem detection, resolution and recovery.

27. Many of the capabilities of MySQL Cluster Manager are already available or can be scripted, so what is the benefit?

A: MySQL Cluster Manager integrates and extends existing management functionality by automating tasks that were previously performed manually by an administrator. As demonstrated in the example above, a process that required 46 manual commands is now reduced to a single command, with each process step being fully automated.

In terms of scripting or even developing a custom management system, it is time consuming, costly and potentially error-prone to manually develop, test and maintain such projects. For many maintenance activities, the need for this type of activity is eliminated with MySQL Cluster Manager.

Through automation, MySQL Cluster Manager simplifies cluster management, while reducing cost, risk and effort.

28. Can MySQL Cluster Manager recover failed nodes in a cluster?

A: Yes. MySQL Cluster itself has the capability to self-heal from failures by automatically restarting failed Data Nodes, without manual intervention. MySQL Cluster Manager extends this functionality by also monitoring and automatically recovering SQL and Management Nodes. This supports a more seamless and complete self-healing of the Cluster to fully restore operations and capacity to applications.

29. So MySQL Cluster Manager can manage, monitor and recover all nodes within a cluster?

A: Yes, with the exception of application nodes using the native NDB API (i.e nodes accessing the Cluster via the C++, Cluster Connector for Java, OpenLDAP, etc direct interfaces).

30. Will the failure of a MySQL Cluster Manager agent impact the availability of the MySQL Cluster database?

A: No. To ensure high availability operation, MySQL Cluster Manager is decoupled from the actual database processes, so if a management agent stops or is upgraded, it does not impact the running database in any way. MySQL Cluster Manager continues to operate across surviving nodes when any given agent or the associated host is not available.

31. How is MySQL Cluster Manager implemented with the MySQL Cluster database?

A: MySQL Cluster Manager is implemented as a set of agents – one running on each physical host that will contain MySQL Cluster nodes (processes) to be managed. The administrator connects the regular mysql client to any one of these agents and then the agents each communicate and work with each other to perform operations across the nodes making up the Cluster.

32. How does MySQL Cluster Manager impact previous approaches to managing MySQL Cluster?

A: When using MySQL Cluster Manager to manage a MySQL Cluster deployment, the administrator no longer edits the configuration files (for example config.ini and my.cnf); instead, these files are created and maintained by the agents. In fact, if those files are manually edited, the changes will be overwritten by the configuration information which is held within the agents.

All processes making up the MySQL Cluster deployment are started, restarted and stopped by MySQL Cluster Manager. This includes the data nodes, management nodes and MySQL Server nodes.

Similarly when using MySQL Cluster Manager, management actions must not be performed by the administrator using the ndb_mgm command (which directly connects to the management node meaning that the agents themselves would not have visibility of any operations performed with it).

33. Do I still need management nodes within my cluster?

A: The introduction of MySQL Cluster manager does not remove the need for management nodes; in particular they continue to perform a number of critical roles:

- When data nodes start up (or are restarted) they connect to the management node(s) to retrieve their configuration data (the management node in turn fetches that data from the configuration files created by the agents);
- When stopping or restarting a data node through MySQL Cluster Manager, the state change is actually performed by the management node;
- The management node(s) can continue to act as arbitrators (avoiding a split-brain scenario). For this reason, it is important to continue to run those processes on separate hosts from the data nodes;
- Some reporting information (for example, memory usage) is not yet available in MySQL Cluster Manager and can still be performed using the ndb_mgm tool.

34. Can MySQL Cluster Manager automatically restart failed agents?

A: There is no angel process for the agents themselves and so for the highest levels of availability, the administrator may choose to use a process monitor to detect the failure of an agent and automatically restart it; for example by creating a script in /etc/init.d

35. Can recovered MySQL Cluster Manager agents automatically resynchronize with other agents?

A: Yes. As management agents restart, they are automatically re-synchronized with the other running management agents to ensure configuration consistency across the entire cluster, without administrator intervention.

36. Can MySQL Cluster Manager persist configuration data across restarts?

A: Yes. All MySQL Cluster configuration information and process identifiers are persisted to disk, enabling them to survive system failures or re-starts of the MySQL Cluster Manager.

37. How does MySQL Cluster Manager ensure the cluster configuration remains consistent across all nodes in the cluster?

A: MySQL Cluster Manager supports asynchronous communication between each management agent in order to reliably propagate reconfiguration requests. As a result, configurations remain consistent across all nodes in the cluster.

Any changes are only committed when all nodes confirm they have received the re-configuration request. If one or more nodes fail to receive the request, then an error is reported back to the client. By automating the coordination of re-configuration requests, opportunities for errors resulting from manually distributing configuration files are eliminated.

38. Which platforms are supported by MySQL Cluster Manager?

A: Please refer to the Supported Platforms page.

39. Which releases of the MySQL Cluster database are supported by MySQL Cluster Manager?

A: MySQL Cluster 6.3 and above.

40. Where can I learn more about MySQL Cluster Manager?

A: From the resources as follows:

- MySQL Cluster Manager whitepaper