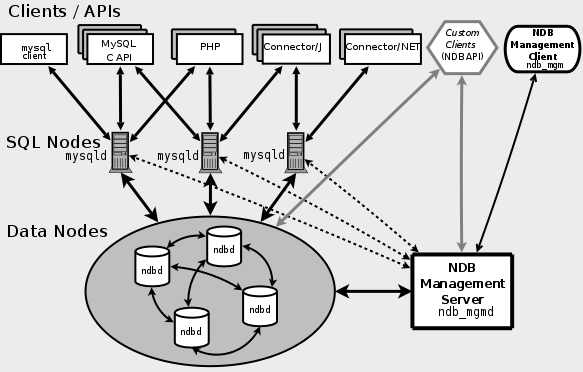
**MySQL NDB Cluster 8.0**

* MySQL NDB Cluster is a high-availability, high-redundancy version of MySQL adapted for the distributed computing environment.
* NDB Cluster is a technology that enables clustering of in-memory databases in a shared-nothing system. The shared-nothing architecture enables the system to work with very inexpensive hardware, and with a minimum of specific requirements for hardware or software.
* NDB Cluster is designed not to have any single point of failure. In a shared-nothing system, each component is expected to have its own memory and disk, and the use of shared storage mechanisms such as network shares, network file systems, and SANs is not recommended or supported.
* NDB Cluster integrates the standard MySQL server with an in-memory clustered storage engine called [NDB](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster.html) (which stands for “*N*etwork *D*ata*B*ase”). In our documentation, the term [NDB](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster.html) refers to the part of the setup that is specific to the storage engine, whereas “MySQL NDB Cluster” refers to the combination of one or more MySQL servers with the [NDB](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster.html) storage engine.
* An NDB Cluster consists of a set of computers, known as hosts, each running one or more processes. These processes, known as nodes, may include MySQL servers (for access to NDB data), data nodes (for storage of the data), one or more management servers, and possibly other specialized data access programs. The relationship of these components in an NDB Cluster is shown here:



All these programs work together to form an NDB Cluster (see [Section 22.4, “NDB Cluster Programs”](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster-programs.html). When data is stored by the [NDB](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster.html) storage engine, the tables (and table data) are stored in the data nodes. Such tables are directly accessible from all other MySQL servers (SQL nodes) in the cluster. Thus, in a payroll application storing data in a cluster, if one application updates the salary of an employee, all other MySQL servers that query this data can see this change immediately.

In addition, a MySQL server that is not connected to an NDB Cluster cannot use the [NDB](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster.html) storage engine and cannot access any NDB Cluster data.

There are three types of cluster nodes, and in a minimal NDB Cluster configuration, there will be at least three nodes, one of each of these types:

* Management node: The role of this type of node is to manage the other nodes within the NDB Cluster, performing such functions as providing configuration data, starting and stopping nodes, and running backups. Because this node type manages the configuration of the other nodes, a node of this type should be started first, before any other node. An MGM node is started with the command [**ndb\_mgmd**](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster-programs-ndb-mgmd.html).
* Data node: This type of node stores cluster data. There are as many data nodes as there are replicas, times the number of fragments (see [Section 22.1.2, “NDB Cluster Nodes, Node Groups, Replicas, and Partitions”](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster-nodes-groups.html)). For example, with two replicas, each having two fragments, you need four data nodes. One replica is sufficient for data storage, but provides no redundancy; therefore, it is recommended to have 2 (or more) replicas to provide redundancy, and thus high availability. A data node is started with the command [**ndbd**](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster-programs-ndbd.html) (see [Section 22.4.1, “**ndbd** — The NDB Cluster Data Node Daemon”](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster-programs-ndbd.html)) or [**ndbmtd**](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster-programs-ndbmtd.html) (see [Section 22.4.3, “**ndbmtd** — The NDB Cluster Data Node Daemon (Multi-Threaded)”](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster-programs-ndbmtd.html)).

NDB Cluster tables are normally stored completely in memory rather than on disk (this is why we refer to NDB Cluster as an in-memory database). However, some NDB Cluster data can be stored on disk; see [Section 22.5.13, “NDB Cluster Disk Data Tables”](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster-disk-data.html), for more information.

* SQL node: This is a node that accesses the cluster data. In the case of NDB Cluster, an SQL node is a traditional MySQL server that uses the [NDBCLUSTER](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster.html) storage engine. An SQL node is a [**mysqld**](https://dev.mysql.com/doc/refman/8.0/en/mysqld.html) process started with the [--ndbcluster](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster-options-variables.html#option_mysqld_ndbcluster) and --ndb-connectstring options, which are explained elsewhere in this chapter, possibly with additional MySQL server options as well.

For a brief introduction to the relationships between nodes, node groups, replicas, and partitions in NDB Cluster, see [Section 22.1.2, “NDB Cluster Nodes, Node Groups, Replicas, and Partitions”](https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster-nodes-groups.html).