SET_VAR Hint Applies	No
Type	Boolean
Default Value	ON

Let ALTER TABLE and other DDL statements use copying operations on NDB tables. Set to OFF to keep this from happening; doing so may improve performance of critical applications.

- --ndb-batch-size=#

Command-Line Format	--ndb-batch-size
System Variable	ndb_batch_size
Scope	Global
Dynamic	No
SET_VAR Hint Applies	No
Type	Integer
Default Value	32768
Minimum Value	0
Maximum Value	31536000

This sets the size in bytes that is used for NDB transaction batches.

- --ndb-cluster-connection-pool=#

Command-Line Format	--ndb-cluster-connection-pool
System Variable	ndb_cluster_connection_pool
System Variable	ndb_cluster_connection_pool
Scope	Global
Scope	Global
Dynamic	No
Dynamic	No
SET_VAR Hint Applies	No
SET_VAR Hint Applies	No
Type	Integer
Default Value	1
Minimum Value	1
Maximum Value	63

By setting this option to a value greater than 1 (the default), a mysqld process can use multiple connections to the cluster, effectively mimicking several SQL nodes. Each connection requires its own [api] or [mysqld] section in the cluster configuration (config.ini) file, and counts against the maximum number of API connections supported by the cluster.

Suppose that you have 2 cluster host computers, each running an SQL node whose mysqld process was started with --ndb-cluster-connection-pool=4; this means that the cluster must have 8 API slots available for these connections (instead of 2). All of these connections are set up when the SQL node connects to the cluster, and are allocated to threads in a round-robin fashion.