

another startup script). This should be similar to what is shown here, where `password` is the MySQL `root` password for a given MySQL server instance:

```
shell> mysqladmin -uroot -ppassword shutdown
081208 20:19:56 mysqld_safe mysqld from pid file
/usr/local/mysql/var/tonfisk.pid ended
shell> mysqld_safe --ndbcluster --ndb-connectstring=198.51.100.10 &
081208 20:20:06 mysqld_safe Logging to '/usr/local/mysql/var/tonfisk.err'.
081208 20:20:06 mysqld_safe Starting mysqld daemon with databases
from /usr/local/mysql/var
```

Of course, the exact input and output depend on how and where MySQL is installed on the system, as well as which options you choose to start it (and whether or not some or all of these options are specified in a `my.cnf` file).

Step 5: Perform an initial start of the new data nodes. From a system shell on each of the hosts for the new data nodes, start the data nodes as shown here, using the `--initial` option:

```
shell> ndbd -c 198.51.100.10 --initial
```



Note

Unlike the case with restarting the existing data nodes, you can start the new data nodes concurrently; you do not need to wait for one to finish starting before starting the other.

Wait until both of the new data nodes have started before proceeding with the next step. Once the new data nodes have started, you can see in the output of the management client `SHOW` command that they do not yet belong to any node group (as indicated with bold type here):

```
ndb_mgm> SHOW
Connected to Management Server at: 198.51.100.10:1186
Cluster Configuration
-----
[ndbd(NDB)]      2 node(s)
id=1      @198.51.100.1  (8.0.26-ndb-8.0.26, Nodegroup: 0, *)
id=2      @198.51.100.2  (8.0.26-ndb-8.0.26, Nodegroup: 0)
id=3      @198.51.100.3  (8.0.26-ndb-8.0.26, no nodegroup)
id=4      @198.51.100.4  (8.0.26-ndb-8.0.26, no nodegroup)

[ndb_mgmd(MGM)]  1 node(s)
id=10     @198.51.100.10 (8.0.26-ndb-8.0.26)

[mysqld(API)]    2 node(s)
id=20     @198.51.100.20 (8.0.26-ndb-8.0.26)
id=21     @198.51.100.21 (8.0.26-ndb-8.0.26)
```

Step 6: Create a new node group. You can do this by issuing a `CREATE NODEGROUP` command in the cluster management client. This command takes as its argument a comma-separated list of the node IDs of the data nodes to be included in the new node group, as shown here:

```
ndb_mgm> CREATE NODEGROUP 3,4
Nodegroup 1 created
```

By issuing `SHOW` again, you can verify that data nodes 3 and 4 have joined the new node group (again indicated in bold type):

```
ndb_mgm> SHOW
Connected to Management Server at: 198.51.100.10:1186
Cluster Configuration
-----
[ndbd(NDB)]      2 node(s)
```