MySQL Cluster Quick Start Guide – SOLARIS

This guide is intended to help the reader get a simple MySQL Cluster database up and running on a single SOLARIS server. Note that for a live deployment multiple hosts should be used to provide redundancy but a single host can be used to gain familiarity with MySQL Cluster; please refer to the final section for links to material that will help turn this into a production system.

Note that there are two alternate tools-based approaches to automatically configuring and running MySQL Cluster:

1. MySQL Cluster Manager is a commercial tool that is available for a 30 day free trial and it allows you to deploy your first, single host, experimental MySQL Cluster with a single command: http://www.clusterdb.com/mysql-cluster/mysql-cluster-manager-1-1-2-creating-a-cluster-is-now-trivial/
2. A preview release of a browser-based MySQL Auto-Installer is available, this hides the complexity of setting up a multi-host Cluster and uses best practices to appropriately configure the available platform resources and supplied hints about your application requirements: http://www.clusterdb.com/mysql-cluster/auto-installer-labs-release/

The rest of this guide focuses on getting your first MySQL Cluster up and running manually.

1   Get the software

For Generally Available (GA), supported versions of the software, download from
http://www.mysql.com/downloads/cluster/

Make sure that you select the correct platform – in this case, "SOLARIS" and then the correct architecture.

If you want to try out a pre-GA version then check
http://dev.mysql.com/downloads/cluster/

For commercial versions of the MySQL Cluster download from
https://edelivery.oracle.com/

Note: Only use MySQL Server executables (mysqlds) that come with the MySQL Cluster installation.
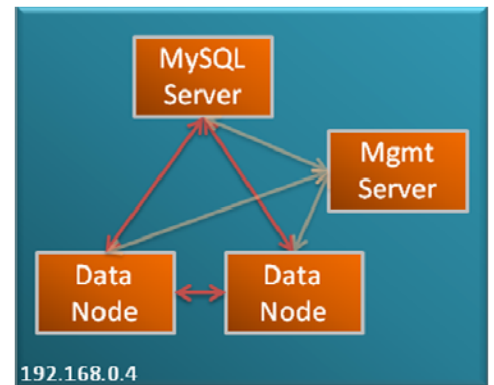
2   Install

Locate the tar ball that you've downloaded, extract it and then create a link to it:

```
[~]$ gunzip Desktop/Downloads/msql-cluster-gpl-7.1.3-solaris10-i386.tar.gz
[~]$ tar xvf Desktop/Downloads/msql-cluster-gpl-7.1.3-solaris10-i386.tar
[~]$ ln –s msql-cluster-gpl-7.1.3-solaris10-i386 mysqlc
```

Optionally, you could add ~/mysqlc/bin to your path to avoid needing the full path when running the processes.

3   Configure

For a first Cluster, start with a single MySQL Server (mysqld), a pair of Data Nodes (ndbd) and a single management node (ndb_mgmd) – all running on the same server.

Create folders to store the configuration files and the data files:

```
[~]$ mkdir my_cluster my_cluster/ndb_data my_cluster/mysqld_data my_cluster/conf
```

In the conf folder, create 2 files:

my.cnf:

```
[mysqld]
ndbcluster
datadir=/home/user1/my_cluster/mysqld_data
basedir=/home/user1/mysqlc
port=5000
```

ORACLE®

config.ini:

```
[ndb_mgmd]
hostname=localhost
datadir=/home/user1/my_cluster/ndb_data
NodeId=1

[ndbd default]
noofreplicas=2
datadir=/home/user1/my_cluster/ndb_data

[ndbd]
hostname=localhost
NodeId=3

[ndbd]
hostname=localhost
NodeId=4

[mysqld]
NodeId=50
```

Note that in a production system there are other parameters that you would set to tune the configuration.

Just like any other MySQL Server, the mysqld process requires a 'mysql' database to be created and populated with essential system data:

```
[~]$ cd mysqlc
[~/mysqlc]$ scripts/mysql_install_db --no-defaults --datadir=/home/user1/my_cluster/mysqld_data/
```

4 Run

The processes should be started in the order of management node, data nodes & then MySQL Server:

```
[~/mysqlc]$ cd ../my_cluster
[~/my_cluster]$ $HOME/mysqlc/bin/ndb_mgmd -f conf/config.ini --initial –configdir=$HOME/my_cluster/conf/
[~/my_cluster]$ $HOME/mysqlc/bin/ndbd -c localhost:1186
[~/my_cluster]$ $HOME/mysqlc/bin/ndbd -c localhost:1186
```

Check the status of the Cluster and wait for the Data Nodes to finish starting before starting the MySQL Server:

```
[~/my_cluster]$ $HOME/mysqlc/bin/ndb_mgm -e show

        Connected to Management Server at: localhost:1186
        Cluster Configuration
        ---------------------
        [ndbd(NDB)]     2 node(s)
        id=3    @127.0.0.1  (mysql-5.1.44 ndb-7.1.3, Nodegroup: 0, Master)
        id=4    @127.0.0.1  (mysql-5.1.44 ndb-7.1.3, Nodegroup: 0)

        [ndb_mgmd(MGM)] 1 node(s)
        id=1    @127.0.0.1  (mysql-5.1.44 ndb-7.1.3)

        [mysqld(API)]   1 node(s)
        id=50 (not connected, accepting connect from any host)

[~/my_cluster]$ $HOME/mysqlc/bin/mysqld --defaults-file=conf/my.cnf &
```

5 Test

Connect to the MySQL Server and confirm that a table can be created that uses the ndb (MySQL Cluster) storage engine:

```
[~/my_cluster]$ $HOME/mysqlc/bin/mysql -h 127.0.0.1 -P 5000 -u root
mysql> create database clusterdb;use clusterdb;
mysql> create table simples (id int not null primary key) engine=ndb;
mysql> insert into simples values (1),(2),(3),(4);
mysql> select * from simples;
```

ORACLE®

```
+----+
| id |
+----+
|  3 |
|  1 |
|  2 |
|  4 |
+----+
```

6 Safely shut down

The MySQL Server must be shut down manually but then the other Cluster nodes can be stopped using the ndb_mgm tool:

```
[~/my_cluster]$ $HOME/mysqlc/bin/mysqladmin -u root -h 127.0.0.1 -P 5000 shutdown
[~/my_cluster]$ $HOME/mysqlc/bin/ndb_mgm -e shutdown
```

7 For further information

MySQL Cluster Evaluation Guide (http://www.mysql.com/why-mysql/white-papers/mysql_cluster_eval_guide.php) In this whitepaper learn the fundamentals of how to design and select the proper components for a successful MySQL Cluster evaluation.

MySQL Cluster Performance Optimization Guide (http://www.mysql.com/why-mysql/white-papers/mysql_wp_cluster_perfomance.php) In this guide, learn how to tune and optimize the MySQL Cluster database to handle diverse workload requirements.

MySQL Cluster Documentation (http://dev.mysql.com/doc/index-cluster.html)