



**Setting Up MySQL™ Cluster Software
Using Solaris™ Zones
Partitioning Technology**

Hashamkha Pathan

August 2008

Sun Microsystems, Inc.

Copyright © 2008 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms. This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd. X/Open is a registered trademark of X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, BluePrints, MySQL, Solaris, and SunSolve are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the United States and other countries.

This product is covered and controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Table of Contents

Introduction.....	4
Steps for Setting Up MySQL Cluster Software With Solaris Zones.....	5
1. Create Solaris Zones.....	5
1.1 Create the Zones Using the Command Line.....	5
1.2 Create the Zones Using a Script.....	6
2. Install MySQL Cluster Software.....	7
2.1 Download and Install MySQL Cluster Software for the Solaris OS for x64 Platforms.....	7
2.2 Set Up the Configuration for the MySQL Server (my.cnf).....	8
2.3 Verify Access to the MySQL Server.....	8
2.4 Modify root User Environment (.profile).....	8
3. Configure and Test MySQL Cluster Software.....	9
3.1 Configure the Management Node.....	9
3.2 Configure the Data and SQL Nodes.....	9
3.3 Start the Cluster.....	10
3.4 Test the Cluster Operation.....	11
For More Information.....	12
Acknowledgments.....	14
About the Author.....	14

Introduction

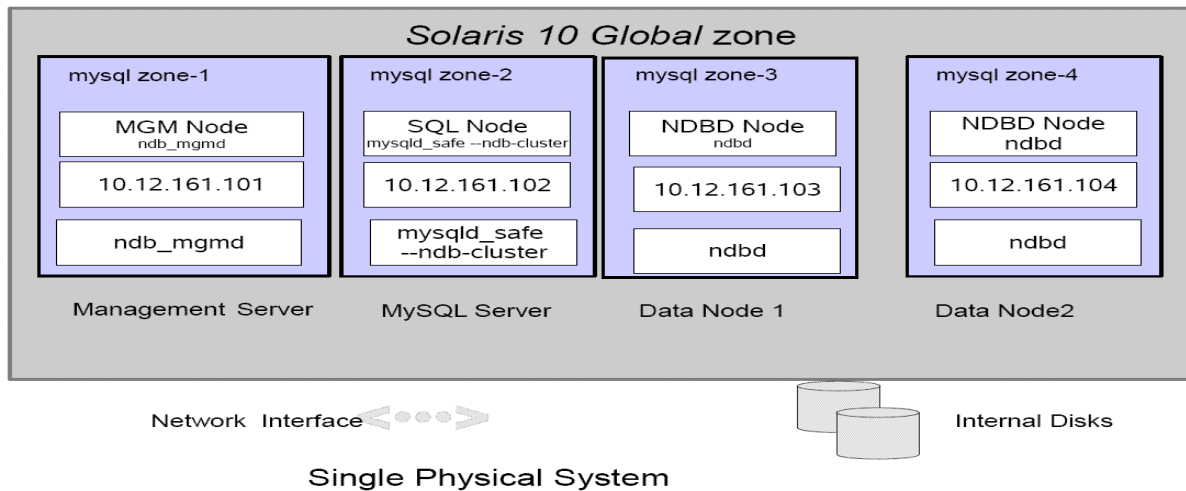
This document describes the steps needed to set up MySQL™ Cluster software to run with Solaris™ Zones partitioning technology.

Solaris Zones is a key feature of the Solaris 10 Operating System. This feature is OS virtualization instrumented inside the Solaris kernel, which enables multiple virtualized secured runtime environments for applications, and it requires very low or no overhead. Unlike some virtualization offerings, Solaris Zones do not require special or dedicated hardware. Some other interesting features of Solaris Zones include cloning and migrations, which can be used to good effect as your rollback strategy for the upgrade and patching process of your applications.

This article covers the steps for getting the MySQL Cluster software running in a Solaris Zones environment as if it were running on independent physical servers. This ability could be useful for replicating an environment in-house without using multiple physical systems. It is possible to extend the setup to use Solaris Zones on different physical systems.

Before we look at the steps involved (install, configure, test) in detail, see Figure 1 to understand the deployment topology. The main components of MySQL Cluster software, which are the MGM node (management server), the SQL node (MySQL server), and the data nodes, are deployed on dedicated zones. A typical production deployment will have these components running on dedicated physical systems.

Figure 1: Deployment Topology



Terms used in this document:

- *Zone or non-global zone:* A zone is a virtualized OS environment created within a single instance of the Solaris OS.
- *Global zone:* The global zone is both the default zone for the system and the zone used for system-wide administrative control.
- *MGM node:* This node manages (configures, starts, stops, and so on) the other nodes within the MySQL Cluster software.

- *SQL node*: The node allows access to cluster data.
- *Data nodes*: This type of node store cluster data.

Steps for Setting Up MySQL Cluster Software With Solaris Zones

1. [Create Solaris Zones](#)
2. [Install MySQL Cluster Software](#)
3. [Configure and Test MySQL Cluster Software](#)

1. Create Solaris Zones

An example of creating a zone using the command line is given in Section 1.1.

When dealing with zones, you mainly use two Solaris commands: `zonecfg`, for configuring, modifying, and deleting zone configurations, and `zoneadm`, for installing, uninstalling, booting, and halting zones.

If you would like speed up things and *not* type the commands for creating all four zones, you can skip Section 1.1 and use the script provided in Section 1.2 to create multiple zones.

1.1 [Create the Zones Using the Command Line](#)

1.2 [Create the Zones Using a Script](#)

1.1 Create the Zones Using the Command Line

1. Configure the zone using the following steps:

```
# zonecfg -z mysql
mysql: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:mysql> create
zonecfg:mysql> set zonepath=/zones/mysql <== Replace this with the directory where the zone files should be placed
zonecfg:mysql> set autoboot=true <== Zone will autoboot with system reboot
zonecfg:mysql> add net
zonecfg:mysql:net> set physical=bge0 <== Replace this with the interface name your system has (Use #ifconfig -a
to verify the interface name)
zonecfg:mysql:net> set address=10.12.161.101/24 <= Replace this with the IP address that you want to assign for
this zone; 24 is the subnet mask)
zonecfg:mysql:net> end
zonecfg:mysql> verify
zonecfg:mysql> commit
zonecfg:mysql> ^D
```

2. Verify the zone configuration using the following command:

```
#zoneadm list -icv <==You should see newly configured zone entry and global zone
```

3. Install the configured zone:

```
# zoneadm -z mysql install
```

This step could take between 15 and 30 minutes depending on the speed of your hard disks and your processor speed.

4. Complete the post-install tasks.

```
#zlogin -C mysql <== From one terminal window, connect to Console; then open a second terminal window and
initiate a boot command as follows
# zoneadm -z mysql boot <== After issuing this command, switch to the Console terminal window and answer the
first-time system configuration questions. Once the zone configuration is complete, the system reboots. Observe this from the Console
window.
```

5. Access the zone:

<== A new command, `zlogin`, is available to access the zone from the global zone without supplying a user name and password. You need to be the `root` user in the global zone. Other access methods, such as `telnet`, `ssh`, and `rlogin`, will work as well.

```
#zlogin <zonename> <== Example: zlogin mysql
```

1.2 Create the Zones Using a Script

The *Solaris Containers-Resource Management and Solaris Zones Guide* contains a script for configuring multiple zones (see “Script to Configure Multiple Zones” at <http://docs.sun.com/app/docs/doc/817-1592/6mhahup04?l=en&a=view#indexterm-224>).

1. Save the script to a file called `zone-create.sh` and run the script to create the four zones needed for the MySQL Cluster software.

```
usage: ./zone-create.sh <#-of-zones> <zonename-prefix> <basedir>
-bash-3.00# ./zone-create.sh 4 mysql /export/home <== Make sure there is no trailing / at the end of the
basedir argument
configuring mysql1
Truncated output
.....
.....
booting mysql4
```

The zones are created and booted.

2. Verify that all zones are configured and running by using the following command:

```
-bash-3.00# zoneadm list -iv
```

3. Modify the zone configuration so all the zones use the correct network interface and IP address.

By default, the zone creation script doesn't assign any network interface or IP address to zones. Use the following commands for *all* the zones:

```
# zonecfg -z mysql1
zonecfg:mysql1> add net
zonecfg:mysql1:net> set physical=bge0
zonecfg:mysql1:net> set address=10.12.161.101/24
zonecfg:mysql1:net> end
zonecfg:mysql1> commit
zonecfg:mysql1> exit
# zoneadm -z mysql1 reboot
```

4. Reset the `root` password for the zones that were created by using the script.

The script sets the `root` user password, which is passed in encrypted form at the time of zone creation. Remember the `zlogin` command described earlier? You need to log in using `zlogin` and reset the `root` password on all the zones, as shown below. You might also want to reset the default shell for the `root` user to be `/usr/bin/bash`.

```
# zlogin mysql1
[Connected to zone 'mysql1' pts/4]
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
# zonename
mysql1
# passwd root
```

```

New Password:
Re-enter new Password:
# usermod -s /usr/bin/bash root
UX: usermod: root is currently logged in, some changes may not take effect until next
login.
# ^D
Reboot the zone using the following command:
#zoneadm -z mysql1 reboot

```

5. Repeat the previous steps for the remaining three zones. Note that you must use a unique IP address for each zone.

You should now have the four zones shown in Table 1, which are required for the MySQL Cluster software to be ready for use.

Table 1: Four Zones That Were Created

MySQL Cluster Component	Zone Name	IP Address
MGM Node	mysql1	10.12.161.101
SQL Node	mysql2	10.12.161.102
Data Node 1	mysql3	10.12.161.103
Data Node 2	mysql4	10.12.161.104

2. Install MySQL Cluster Software

2.1 [Download and Install MySQL Cluster Software for the Solaris OS for x64 Platforms](#)

2.2 [Set Up the Configuration for the MySQL Server \(my.cnf\)](#)

2.3 [Verify Access to the MySQL Server](#)

2.4 [Modify root User Environment \(.profile\)](#)

2.1 Download and Install MySQL Cluster Software for the Solaris OS for x64 Platforms

1. Download the MySQL Cluster software for the Solaris OS for x64 platforms, which is available at http://dev.mysql.com/downloads/cluster/#solaris_tar.

2. Transfer the downloaded binary to all the zones.

You can transfer the file to the individual zones using FTP by using the following commands.

Caution: According to the installation instructions, due to the way MySQL binaries are packaged, you need to create a `/usr/local/mysql` link to the directory where the MySQL binaries are located. Since the `/usr` file system is read-only from the non-global zone for sparse root zone configuration, use the following workaround. You can achieve the same results by playing with the `mysql_install_db` script.

Ensure that you are in the global zone, and then perform the following commands:

```

bash-3.00# zonename
global
bash-3.00# mkdir /mysql-cluster-gpl-6.2.15-solaris10-x86_64
bash-3.00# ln -s /mysql-cluster-gpl-6.2.15-solaris10-x86_64 /usr/local/mysql

```

<== Now we can proceed with the installation in the zone. Verify that you are logged in to the required zone.

```

bash-3.00# zonename
mysql1
bash-3.00# pwd
/
bash-3.00# ls my*          <== MySQL binaries transferred under "/" directory of zone
mysql-cluster-gpl-6.2.15-solaris10-x86_64.tar.gz
bash-3.00# gzip -d mysql-cluster-gpl-6.2.15-solaris10-x86_64.tar.gz
bash-3.00# tar -xf mysql-cluster-gpl-6.2.15-solaris10-x86_64.tar
bash-3.00# ls -ltd /usr/local/my*
lrwxrwxrwx 1 root root 42 Jul 1 19:09 /usr/local/mysql -> /mysql-cluster-gpl-6.2.15-
solaris10-x86_64
bash-3.00# groupadd mysql
bash-3.00# useradd -g mysql mysql
bash-3.00# chown -R mysql:mysql /mysql-cluster-gpl-6.2.15-solaris10-x86_64
bash-3.00# cd /usr/local/mysql
bash-3.00# pwd
/usr/local/mysql
bash-3.00# ./scripts/mysql_install_db --user=mysql
Installing MySQL system tables...
OK
Truncated output
.....
<== You can test the MySQL daemon with mysql-test-run.pl
cd ./mysql-test ; perl mysql-test-run.pl

```

2.2 Set Up the Configuration for the MySQL Server (my.cnf)

Copy the sample my.cnf file that is shipped with MySQL so you can use it as the configuration file for your environment:

```

root@mysql1:[/] # cd /usr/local/mysql/support-files/
root@mysql1:[/usr/local/mysql/support-files] # cp my-small.cnf /etc/my.cnf
bash-3.00# ./bin/mysqld_safe --user=mysql &
[1] 26530
080701 19:24:28 mysqld_safe Logging to '/usr/local/mysql/data/mysql3.err'.
080701 19:24:28 mysqld_safe Starting mysqld daemon with databases from
/usr/local/mysql/data

```

2.3 Verify Access to the MySQL Server

Run the following commands to verify that you have access to the MySQL server:

```

bash-3.00# ./bin/mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.23-ndb-6.2.15-cluster-gpl MySQL Cluster Server (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>mysql> exit
Bye
bash-3.00# pwd
/usr/local/mysql
bash-3.00# ./bin/mysqladmin shutdown

```

2.4 Modify root User Environment (.profile)

Edit the .profile of the root user and add the following entries. Note that this step assumes that the default shell for the root user is set to bash. You might need to correct the syntax based on which shell is used. Changing the default shell for the root user is explained in Section 1.2, Step 4.

```

PATH=$PATH:/usr/local/mysql/bin;export PATH          <== Path to MySQL binaries added
PS1="\u@\h:[\w] # ";export PS1                      <== Default prompt changed

```

Note: Repeat the steps in sections 2.1 through 2.4 on other three zones before proceeding further. After completion, all four zones will have the MySQL Cluster package installed and ready to be configured in cluster mode.

3. Configure and Test MySQL Cluster Software

[3.1 Configure the Management Node](#)

[3.2 Configure the Data and SQL Nodes](#)

[3.3 Start and Stop the Cluster](#)

[3.4 Test the Cluster Operation](#)

3.1 Configure the Management Node

Use the following steps to configure the management (MGM) node:

```
root@mysql1:[/] # mkdir /var/lib/mysql-cluster
root@mysql1:[/] # cp /usr/local/mysql/support-files/config.small.ini /var/lib/mysql-
cluster/config.ini
root@mysql1:[/] # cd /var/lib/mysql-cluster/
root@mysql1:[/var/lib/mysql-cluster] # vi config.ini <== Edit the file and make the changes shown below
[NDBD DEFAULT]
NoOfReplicas: 2
DataDir: /usr/local/mysql/data
FileSystemPath: /usr/local/mysql/data
# Data Memory, Index Memory, and String Memory
DataMemory: 600M
IndexMemory: 100M
BackupMemory: 64M
[MGM DEFAULT]
PortNumber: 1186
DataDir: /usr/local/mysql/data
[NDB_MGMD]
Id: 1
HostName: 10.12.161.101 <== Replace localhost entry with IP address of MGM node
ArbitrationRank: 1
[mysqld]
Id: 2
HostName: 10.12.161.102 <== Replace localhost entry with IP address of SQL node
[NDBD]
Id: 3
HostName: 10.12.161.103 <== Replace localhost entry with IP address of Data node-1
[NDBD]
Id: 4
HostName: 10.12.161.104 <== Replace localhost entry with IP address of Data node-2
```

Before you start the MGM node processes, finish the configuration on SQL and data nodes, as described next.

3.2 Configure the Data and SQL Nodes

To configure the SQL and data nodes, edit `/etc/my.cnf` and add the entries shown below. These entries configure the SQL and data nodes to run in cluster mode, and they provide information regarding the management node of the cluster.

```
# Cluster configuration information: Node is running in cluster mode and 10.12.161.101 is the IP address of the MGM server
[mysqld]
ndbcluster
ndb-connectstring=10.12.161.101
[mysql_cluster]
ndb-connectstring=10.12.161.101
```

3.3 Start the Cluster

1. On the management node, run the following commands. If you are not already logged in to the management node zone, log in using the following command from the global zone.

```
global# zlogin mysql1
[Connected to zone 'mysql1' pts/4]
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
root@mysql1:[/] # cd /var/lib/mysql-cluster
root@mysql1:[/var/lib/mysql-cluster] # ndb_mgmd -f /var/lib/mysql-cluster/config.ini
```

2. On the SQL node, run the following commands. If you are not already logged in to the SQL node zone, log in using the following command from the global zone.

```
global# zlogin mysql2
[Connected to zone 'mysql1' pts/4]
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
root@mysql2:[/] # mysqld_safe --user=mysql &
```

3. On the data node, run the following commands. If you are not already logged in to the data node-1 zone, log in using the following command from the global zone.

```
global# zlogin mysql3
[Connected to zone 'mysql1' pts/4]
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
root@mysql3:[/] # ndbd
root@mysql3:[/] #
```

4. On the other data node, run the following commands. If you are not already logged in to the data node-2 zone, log in using the following command from the global zone.

```
global# zlogin mysql4
[Connected to zone 'mysql1' pts/4]
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
root@mysql4:[/] # ndbd
root@mysql4:[/] #
```

5. Verify the cluster status by running the following commands from the management node:

```
root@mysql1:[/] # ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> show
Cluster Configuration
-----
<>[ndbd(NDB)] 2 node(s)
id=3 @10.12.161.103 (mysql-5.1.23 ndb-6.2.15, Nodegroup: 0, Master)
id=4 @10.12.161.104 (mysql-5.1.23 ndb-6.2.15, Nodegroup: 0)
[ndb_mgmd(MGM)] 1 node(s)
id=1 @10.12.161.101 (mysql-5.1.23 ndb-6.2.15)
[mysqld(API)] 6 node(s)
id=2 @10.12.161.102 (mysql-5.1.23 ndb-6.2.15)
id=5 (not connected, accepting connect from 10.12.161.101)
id=6 (not connected, accepting connect from 10.12.161.101)
id=7 (not connected, accepting connect from any host)
id=8 (not connected, accepting connect from any host)
id=9 (not connected, accepting connect from any host)
ndb_mgm>
```

The SQL node and both data nodes are reported in the configuration information and they are connected. You can now move to testing the cluster configuration.

Note: To shut down the entire cluster, use `ndb_mgm -e shutdown`, as follows, from the management node. (Remember that you would rarely shut down the entire cluster in real deployment scenarios.)

```
root@mysql1:[/] # ndb_mgm -e shutdown
```

3.4 Test the Cluster Operation

1. Connect to the cluster database. You need to access the cluster database through the SQL node. Log in to the SQL node zone and connect to the database and create a sample table.

Later, you'll force failure of one of the data nodes to see if this affects the database availability.

```
root@mysql2:[/] # mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.23-ndb-6.2.15-cluster-gpl MySQL Cluster Server (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>use test;
Database changed
mysql> CREATE TABLE SetupDetails
(ZoneName varchar(15),
Mcomponent varchar(30),
IP int(15)) ENGINE=NDBCLUSTER;
Query OK, 0 rows affected (0.78 sec)
mysql> desc SetupDetails;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ZoneName       | varchar(15) | YES |     | NULL    |      |
| Mcomponent     | varchar(30) | YES |     | NULL    |      |
| IP             | int(15)    | YES |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
mysql>
```

2. Induce the failure of one of the data nodes.

a. On the MGM node, verify the cluster status before inducing the failure.

```
ndb_mgm> show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=3 @10.12.161.103(mysql-5.1.23 ndb-6.2.15, Nodegroup: 0, Master)
id=4 @10.12.161.104(mysql-5.1.23 ndb-6.2.15, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @10.12.161.101(mysql-5.1.23 ndb-6.2.15)

[mysqld(API)] 6 node(s)
id=2 @10.12.161.102 (mysql-5.1.23 ndb-6.2.15)
id=5 (not connected, accepting connect from 10.12.161.101)
id=6 (not connected, accepting connect from 10.12.161.101)
id=7 (not connected, accepting connect from any host)
Truncated output
```

b. Start a session to the database from the SQL node:

```
root@mysql2:[/] # mysql
mysql> use test;
Database changed
mysql> desc SetupDetails;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ZoneName       | varchar(15) | YES |     | NULL    |      |
| Mcomponent     | varchar(30) | YES |     | NULL    |      |
| IP             | int(15)    | YES |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+

```

```
3 rows in set (0.01 sec)
mysql>
```

c. Reboot one of the data nodes:

```
global# zoneadm -z mysql4 reboot
```

d. Verify the status of the cluster from the MGM node. Note that the status of 12.12.161.104 is not connected.

```
ndb_mgm> show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=3 @10.12.161.103 (mysql-5.1.23 ndb-6.2.15, Nodegroup: 0, Master)
id=4 (not connected, accepting connect from 10.12.161.104)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @10.12.161.101 (mysql-5.1.23 ndb-6.2.15)

[mysqld(API)] 6 node(s)
id=2@10.12.161.102(mysql-5.1.23 ndb-6.2.15)
id=5 (not connected, accepting connect from 10.12.161.101)
id=6 (not connected, accepting connect from 10.12.161.101)
id=7 (not connected, accepting connect from any host)
Truncated output
```

e. To determine whether rebooting the data node affected data availability, check from the SQL node. You will notice rebooting didn't affect the availability of the data.

```
mysql> desc SetupDetails;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ZoneName       | varchar(15)   | YES  |     | NULL    |      |
| Mcomponent     | varchar(30)   | YES  |     | NULL    |      |
| IP             | int(15)       | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
mysql>
```

What if we reboot the other remaining data node as well?

Now you are ready to start using the MySQL Cluster software with zones. You can simulate failures on the cluster nodes. In this section, we saw a scenario of a “failed data node,” while we were accessing the database from the SQL node. Since we are running the MySQL Cluster software with Solaris Zones, we now have the ability to test various failure scenarios as if we were using different physical systems.

For More Information

Here are additional resources:

- Documentation:
 - MySQL documentation site: <http://dev.mysql.com/doc/>
 - *System Administration Guide: Solaris Containers-Resource Management and Solaris Zones* on docs.sun.com: <http://docs.sun.com/app/docs/doc/817-1592>

- MySQL download sites:
 - <http://www.sun.com/software/products/mysql/getit.jsp>
 - <http://dev.mysql.com/downloads/>
- MySQL training courses: <http://www.sun.com/software/products/mysql/training.jsp>
- Support:
 - Register your Sun gear: <https://inventory.sun.com/inventory/>
 - Services: <http://www.sun.com/service/>
 - SunSolveSM Online: <http://sunsolve.sun.com>
- MySQL forums: <http://forums.mysql.com/>
- Other web sites and resources:
 - MySQL site on sun.com: <http://www.sun.com/software/products/mysql/index.jsp>
 - mysql.com site: <http://www.mysql.com/>
 - Features & Benefits - MySQL Cluster: <http://www.sun.com/software/products/mysql/cluster.jsp>
 - MySQL and MySQL Cluster FAQs: <http://www.sun.com/software/products/mysql/faqs.jsp>
 - Sun BluePrints™ article: Optimize MySQL Server on Sun x64 Servers and Storage <http://wikis.sun.com/display/BluePrints/Optimize+MySQL+Server+on+Sun+x64+Servers+and+Storage>
 - Resources on BigAdmin:
 - MySQL Resources for System Administrators: <http://www.sun.com/bigadmin/topics/mysql/>
 - Creating a Script for Scheduled MySQL Backups (community submission): http://www.sun.com/bigadmin/content/submitted/sched_mysql_backup.jsp
 - Database Resource Collection: <http://www.sun.com/bigadmin/collections/database.html>
 - Resources on Sun Developer Network (SDN):
 - Using MySQL and Memcached on the GlassFish Application Server: http://developers.sun.com/appserver/reference/techart/mysql_qf/
 - Sun and MySQL: How It Stacks Up for Developers: http://java.sun.com/developer/technicalArticles/mysql_acq/index.html
 - Tech Days Events blog: MySQL and Java DB: http://blogs.sun.com/TechDaysEvents/entry/mysql_and_java_db
- Events of interest to users of Sun products:
 - Sun and MySQL Around the World--Meetup Mashup: <http://www.sun.com/events/mysqltour/index.jsp>
 - Find a MySQL Meetup Group near you: <http://mysql.meetup.com/>
 - Worldwide developer events: <http://developers.sun.com/events/>
 - Current events: <http://www.sun.com/events/index.jsp>

Acknowledgments

Thanks to Ritu Kamboj for reviewing this document.

About the Author

Hasham is staff engineer in the ISV engineering group at Sun Microsystems, which is focused on ISV adoption of Sun technologies. For the past four years, his focus has been on systems and application performance. He has been in the IT industry for more than 13 years and has worked extensively on operating systems, storage, and databases. He is an Oracle Certified Professional for Oracle 10g (OCP), Sun Certified Security Administrator for the Solaris 10 Operating System, Sun Certified System Administrator for the Solaris 2.6, 2.8 and 10 Operating System, and Certified HP-UX Administrator. He also is certified in IT Service Management (ISEB).

Licensing Information

Unless otherwise specified, the use of this software is authorized pursuant to the terms of the license found at http://www.sun.com/bigadmin/common/berkeley_license.html.