

Oracle® Solaris Cluster Data Service for MySQL Guide

ORACLE®

Part No: E63545
June 2017

Part No: E63545

Copyright © 2000, 2017, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Référence: E63545

Copyright © 2000, 2017, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou ce matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> si vous êtes malentendant.

Contents

Using This Documentation	13
1 Installing and Configuring HA for MySQL	15
Installing and Configuring HA for MySQL	15
HA for MySQL Overview	16
Planning the HA for MySQL Installation and Configuration	16
MySQL and Oracle Solaris Zones	16
Configuration Restrictions	17
Configuration Requirements	21
Installing and Configuring MySQL	28
Enabling MySQL to Run in a Global Zone Configuration	29
▼ How to Install and Configure MySQL in a Global Zone	31
Enabling MySQL to Run in a Zone Configuration	34
▼ How to Install and Configure MySQL in a Zone	36
▼ How to Enable MySQL to run in an HA Zone Configuration	39
▼ How to Install and Configure MySQL in an HA Zone	41
Verifying the Installation and Configuration of MySQL	43
▼ How to Verify the Installation and Configuration of MySQL	43
Installing the HA for MySQL Package	44
▼ How to Install the HA for MySQL Package	44
Registering and Configuring HA for MySQL	45
▼ How to Register and Configure HA for MySQL as a Failover Service in a Global Zone Configuration	46
▼ How to Register and Configure HA for MySQL as a Failover Service in a Zone Configuration	52
▼ How to Register and Configure HA for MySQL as a Failover Service in an HA Zone Configuration	58
▼ How to Modify Parameters in the HA for MySQL Manifest	64
▼ How to Remove an HA for MySQL Resource From an HA Zone	65

▼ How to Add an HA for MySQL Resource in a Scalable or Multiple-Master Configuration	65
Verifying the HA for MySQL Installation and Configuration	73
▼ How to Verify the HA for MySQL Installation and Configuration	73
Understanding the HA for MySQL Fault Monitor	73
Resource Properties	73
Probing Algorithm and Functionality	74
Debugging the HA for MySQL	74
▼ How to Activate Debugging for HA for MySQL	74
A Deployment Example: Installing MySQL in the Global Zone	77
Target Cluster Configuration	77
Software Configuration	77
Assumptions	78
Installing and Configuring MySQL on Local Storage in the Global Zone	78
▼ Example: Preparing the Cluster for MySQL	78
▼ Example: Configuring Cluster Resources for MySQL	79
▼ Example: Installing and Bootstrapping the MySQL Software on Local Storage	79
▼ Example: Modifying the MySQL Configuration Files	81
▼ Example: Enabling the MySQL Software to Run in the Cluster	82
B Deployment Example: Installing MySQL in the Non-Global HA Zone	85
Target Cluster Configuration	85
Software Configuration	85
Assumptions	86
Installing and Configuring MySQL on Local Storage in the HA Zone	86
▼ Example: Preparing the Cluster for MySQL	86
▼ Example: Configuring Cluster Resources for MySQL	87
▼ Example: Configuring the HA Container	87
▼ Example: Installing and Bootstrapping the MySQL Software on Local Storage	89
▼ Example: Modifying the MySQL Configuration Files	91
▼ Example: Enabling the MySQL Software to Run in the Cluster	92
C Deployment Example: Installing MySQL in a Scalable or Multiple-Master Configuration	95

Target Cluster Configuration	95
Software Configuration	95
Assumptions	96
Installing and Configuring MySQL on Local Storage in the Global Zone	96
▼ Example: How to Prepare the Cluster for MySQL	96
▼ Example: How to Configure MySQL in a Scalable Resource Group	97
▼ Example: How to Configure MySQL in a Multiple-Master Resource Group	98
▼ Example: How to Install and Bootstrap the MySQL Software on Local Storage	98
▼ Example: How to Modify the MySQL Configuration Files	100
▼ Example: How to Enable the MySQL Software to Run in the Cluster	101
Index	103

Tables

TABLE 1	Task Map: Installing and Configuring HA for MySQL	15
TABLE 2	Protection of Components	16
TABLE 3	Dependencies Between HA for MySQL Components in Failover Configurations	22

Examples

EXAMPLE 1	MySQL Instances with Cluster File Systems	18
EXAMPLE 2	MySQL Instances with Highly Available Local File Systems	20
EXAMPLE 3	The <code>ha_mysql_config</code> File	22
EXAMPLE 4	The <code>mysql_config</code> File	25
EXAMPLE 5	Editing the Debug Config File	75

Using This Documentation

- **Overview** – Explains how to install and configure the HA for MySQL data service
- **Audience** – Technicians, system administrators, and authorized service providers
- **Required knowledge** – Advanced experience troubleshooting and replacing hardware

Product Documentation Library

Documentation and resources for this product and related products are available at <http://www.oracle.com/pls/topic/lookup?ctx=E56676-01>.

Feedback

Provide feedback about this documentation at <http://www.oracle.com/goto/docfeedback>.

◆◆◆ CHAPTER 1

Installing and Configuring HA for MySQL

This chapter explains how to install and configure HA for MySQL.

This chapter contains the following sections.

- [“Installing and Configuring HA for MySQL” on page 15](#)
- [“HA for MySQL Overview” on page 16](#)
- [“Planning the HA for MySQL Installation and Configuration” on page 16](#)
- [“Installing and Configuring MySQL” on page 28](#)
- [“Verifying the Installation and Configuration of MySQL” on page 43](#)
- [“Installing the HA for MySQL Package” on page 44](#)
- [“Registering and Configuring HA for MySQL” on page 45](#)
- [“Verifying the HA for MySQL Installation and Configuration” on page 73](#)
- [“Understanding the HA for MySQL Fault Monitor” on page 73](#)
- [“Debugging the HA for MySQL” on page 74](#)

Installing and Configuring HA for MySQL

[Table 1, “Task Map: Installing and Configuring HA for MySQL,” on page 15](#) lists the tasks for installing and configuring HA for MySQL. Perform these tasks in the order that they are listed.

TABLE 1 Task Map: Installing and Configuring HA for MySQL

Task	For Instructions, Go To
1. Plan the installation.	“HA for MySQL Overview” on page 16 “Planning the HA for MySQL Installation and Configuration” on page 16
2. Install and configure MySQL.	“How to Install and Configure MySQL in a Global Zone” on page 31

Task	For Instructions, Go To
3. Verify installation and configuration.	“How to Verify the Installation and Configuration of MySQL” on page 43
4. Install HA for MySQL Packages.	“Installing the HA for MySQL Package” on page 44
5. Register and Configure HA for MySQL.	“How to Register and Configure HA for MySQL as a Failover Service in a Global Zone Configuration” on page 46
6. Verify HA for MySQL Installation and Configuration.	“How to Verify the HA for MySQL Installation and Configuration” on page 73
7. Understand HA for MySQL fault monitor.	“Understanding the HA for MySQL Fault Monitor” on page 73
8. Debug HA for MySQL.	“Debugging the HA for MySQL” on page 74

HA for MySQL Overview

The MySQL software delivers a fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software.

MySQL is freely available under the GNU General Public License. You can download it from <http://www.mysql.com>.

The HA for MySQL data service provides a mechanism for orderly startup and shutdown, fault monitoring and automatic failover of the MySQL service. The following MySQL components are protected by the HA for MySQL data service.

TABLE 2 Protection of Components

Component	Protected by
MySQL server	HA for MySQL

Planning the HA for MySQL Installation and Configuration

This section contains the information you need to plan your HA for MySQL installation and configuration.

MySQL and Oracle Solaris Zones

HA for MySQL is supported in Oracle Solaris Zones. Oracle Solaris Cluster offers the following concepts for Oracle Solaris Zones.

- Zone clusters are containers that are running after a reboot of the node. These containers form virtual clusters with their own node names.
- High-availability zones (HA Zones), also called failover zones, are managed by the Oracle Solaris Cluster HA for Oracle Solaris Zones data service, and are represented by a resource of a resource group.

Configuration Restrictions

This section provides a list of software and hardware configuration restrictions that apply to HA for MySQL only.

For restrictions that apply to all data services, see the [Oracle Solaris Cluster 4.3 Release Notes](#).



Caution - Your data service configuration might not be supported if you do not observe these restrictions.

Restriction for the HA for MySQL Data Service Configuration

A single MySQL Server is to be configured as a failover resource only. A MySQL server is configured within a MySQL cluster, can be configured as a failover, scalable or a multiple-master resource. Scalable or multiple-master data services are typically deployed when you configure the MySQL server for MySQL Cluster.

Restrictions on the MySQL Configuration File

The MySQL configuration file, `my.cnf`, should be placed only in the MySQL Database directory. If `my.cnf` has to be placed in local file system, and then create a symbolic link from the MySQL Database directory. The configuration file `my.cnf` should not be placed in the `/etc` directory because it will override command line options.

Restrictions for the MySQL Configurations

The following configurations are supported with the HA for MySQL data service.

- Single/Multiple MySQL instances in master configuration
- Single/Multiple MySQL instances in slave configuration

Restrictions on the MySQL Database Directory

Regardless of which MySQL delivery method you have chosen, whether from <http://www.mysql.com> or from other source, each MySQL instance must have a unique Database directory. You can mount this Database directory as either a highly available local file system or cluster file system.

Note - It is always a good practice to mount the cluster file systems with the `/global` prefix and to mount the highly available local file systems with the `/local` prefix.

Restriction for the MySQL smf Service Name in an HA Zone

The MySQL configuration in an HA Zone uses the smf component of Oracle Solaris Cluster HA for Oracle Solaris Zones. The registration of the MySQL data service in an HA Zone defines an smf service to control the MySQL database. The name of this smf service is generated in this naming scheme: `svc:/application/sczone-agents:resource-name`. No other smf service with exactly this name can exist.

The associated smf manifest is automatically created during the registration process in this location and naming scheme: `/var/svc/manifest/application/sczone-agents/resource-name.xml`. No other manifest can coexist with this name.

Examples for the File Systems Layout

EXAMPLE 1 MySQL Instances with Cluster File Systems

The following example shows MySQL installed onto a cluster file system with two MySQL instances (`mysql-data-1` and `mysql-data-2`). The final output shows a subset of the `/etc/vfstab` entries for a MySQL deployment.

```
# ls -l /usr/local
total 4
drwxrwxrwx  2 root    other    512 Oct  1 16:44 bin
lrwxrwxrwx  1 root    other      13 Oct 11 11:20 mysql -> /global/mysql
#
# ls -l /global/mysql
total 10432
drwxr-xr-x 13 mysql   mysql    512 Mar 16 00:03 .
drwxrwxrwx  7 root    other    2048 Apr 11 09:53 ..
-rw-r--r--  1 mysql   mysql   19106 Mar 15 23:29 COPYING
-rw-r--r--  1 mysql   mysql   28003 Mar 15 23:29 COPYING.LIB
```

```

-rw-r--r-- 1 mysql mysql 126466 Mar 15 16:47 ChangeLog
-rw-r--r-- 1 mysql mysql 6811 Mar 15 23:29 INSTALL-BINARY
-rw-r--r-- 1 mysql mysql 1937 Mar 15 16:47 README
drwxr-xr-x 2 mysql mysql 1536 Mar 16 00:03 bin
-rwxr-xr-x 1 mysql mysql 773 Mar 16 00:03 configure
drwxr-x-- 4 mysql mysql 512 Mar 16 00:03 data
drwxr-xr-x 2 mysql mysql 1024 Mar 16 00:03 include
drwxr-xr-x 2 mysql mysql 512 Mar 16 00:03 lib
drwxr-xr-x 3 mysql mysql 512 Mar 16 00:03 man
-rw-r--r-- 1 mysql mysql 2676944 Mar 15 23:23 manual.html
-rw-r--r-- 1 mysql mysql 2329252 Mar 15 23:23 manual.txt
-rw-r--r-- 1 mysql mysql 98233 Mar 15 23:23 manual_toc.html
drwxr-xr-x 6 mysql mysql 512 Mar 16 00:03 mysql-test
drwxr-xr-x 2 mysql mysql 512 Mar 16 00:03 scripts
drwxr-xr-x 3 mysql mysql 512 Mar 16 00:03 share
drwxr-xr-x 7 mysql mysql 1024 Mar 16 00:03 sql-bench
drwxr-xr-x 2 mysql mysql 512 Mar 16 00:03 support-files
drwxr-xr-x 2 mysql mysql 512 Mar 16 00:03 tests
# ls -l /global/mysql-data-1
Total 30
drwxrwxrwx 9 mysql mysql 512 Apr 15 12:06 .
drwxrwxrwx 20 root root 1024 Apr 10 12:41 ..
drwxr-xr-x 2 mysql mysql 512 Apr 15 12:00 BDB
drwxrwxrwx 2 mysql mysql 512 Apr 15 11:59 innodb
drwxrwxrwx 2 mysql mysql 2048 Apr 15 14:47 logs
-rw-r--r-- 1 mysql mysql 1432 Apr 15 11:58 my.cnf
drwx----- 2 mysql mysql 512 Apr 15 11:59 mysql
-rw-rw---- 1 mysql mysql 5 Apr 15 14:47 mysqld.pid
drwx----- 2 mysql mysql 512 Apr 15 14:53 sc3_test_database
drwx----- 2 mysql mysql 512 Apr 15 11:58 test
drwx----- 2 mysql mysql 512 Apr 15 12:00 testdb
#
# ls -l /global/mysql-data-2
total 32
drwxrwxrwx 9 mysql mysql 512 Apr 15 07:49 .
drwxrwxrwx 20 root root 1024 Apr 10 12:41 ..
drwxr-xr-x 2 mysql mysql 512 Apr 14 11:16 BDB
drwxr-xr-x 2 mysql mysql 512 Apr 14 11:14 innodb
drwxr-xr-x 2 mysql mysql 2560 Apr 15 10:15 logs
-rw-r--r-- 1 mysql mysql 1459 Apr 14 11:13 my.cnf
drwx----- 2 mysql mysql 512 Apr 14 11:14 mysql
-rw-rw---- 1 mysql mysql 5 Apr 15 10:10 mysqld.pid
drwx----- 2 mysql mysql 512 Apr 15 10:10 sc3_test_database
drwx----- 2 mysql mysql 512 Apr 14 11:14 test
drwx----- 2 mysql mysql 512 Apr 14 11:16 testdb
# more /etc/vfstab Subset of the output

```

```

/dev/vx/dsk/dg1/vol01 /dev/vx/rdisk/dg1/vol01 /global/mysql ufs 2
yes global,logging
/dev/vx/dsk/dg2/vol01 /dev/vx/rdisk/dg2/vol01 /global/mysql-data-1 ufs 2
yes global,logging
/dev/vx/dsk/dg2/vol01 /dev/vx/rdisk/dg2/vol01 /global/mysql-data-2 ufs 2
yes global,logging

#

```

Note - In the above example the *Database directory* for the MySQL instance 1 is `/global/mysql-data-1`, whereas the *Database directory* for the MySQL instance 2 is `/global/mysql-data-2`.

EXAMPLE 2 MySQL Instances with Highly Available Local File Systems

The following example shows MySQL installed on local file systems and two MySQL instances, `mysql-data-1` and `mysql-data-2`, on highly available local file systems. The final output shows a subset of the `/etc/vfstab` entries for a MySQL deployment.

```

# ls -l /usr/local/mysql
total 10432
drwxr-xr-x 13 mysql  mysql      512 Mar 16 00:03 .
drwxrwxrwx  7 root   other     2048 Apr 11 09:53 ..
-rw-r--r--  1 mysql  mysql    19106 Mar 15 23:29 COPYING
-rw-r--r--  1 mysql  mysql    28003 Mar 15 23:29 COPYING.LIB
-rw-r--r--  1 mysql  mysql   126466 Mar 15 16:47 ChangeLog
-rw-r--r--  1 mysql  mysql    6811 Mar 15 23:29 INSTALL-BINARY
-rw-r--r--  1 mysql  mysql    1937 Mar 15 16:47 README
drwxr-xr-x  2 mysql  mysql    1536 Mar 16 00:03 bin
-rwxr-xr-x  1 mysql  mysql    773 Mar 16 00:03 configure
drwxr-x---  4 mysql  mysql    512 Mar 16 00:03 data
drwxr-xr-x  2 mysql  mysql   1024 Mar 16 00:03 include
drwxr-xr-x  2 mysql  mysql    512 Mar 16 00:03 lib
drwxr-xr-x  3 mysql  mysql    512 Mar 16 00:03 man
-rw-r--r--  1 mysql  mysql   2676944 Mar 15 23:23 manual.html
-rw-r--r--  1 mysql  mysql   2329252 Mar 15 23:23 manual.txt
-rw-r--r--  1 mysql  mysql   98233 Mar 15 23:23 manual_toc.html
drwxr-xr-x  6 mysql  mysql    512 Mar 16 00:03 mysql-test
drwxr-xr-x  2 mysql  mysql    512 Mar 16 00:03 scripts
drwxr-xr-x  3 mysql  mysql    512 Mar 16 00:03 share
drwxr-xr-x  7 mysql  mysql   1024 Mar 16 00:03 sql-bench
drwxr-xr-x  2 mysql  mysql    512 Mar 16 00:03 support-files
drwxr-xr-x  2 mysql  mysql    512 Mar 16 00:03 tests
# ls -l /local/mysql-data-1
Total 30
drwxrwxrwx  9 mysql  mysql    512 Apr 15 12:06 .

```

```

drwxrwxrwx 20 root    root    1024 Apr 10 12:41 ..
drwxr-xr-x  2 mysql  mysql   512 Apr 15 12:00 BDB
drwxrwxrwx  2 mysql  mysql   512 Apr 15 11:59 innodb
drwxrwxrwx  2 mysql  mysql  2048 Apr 15 14:47 logs
-rw-r--r--  1 mysql  mysql  1432 Apr 15 11:58 my.cnf
drwx----- 2 mysql  mysql   512 Apr 15 11:59 mysql
-rw-rw----  1 mysql  mysql    5 Apr 15 14:47 mysqld.pid
drwx----- 2 mysql  mysql   512 Apr 15 14:53 sc3_test_database
drwx----- 2 mysql  mysql   512 Apr 15 11:58 test
drwx----- 2 mysql  mysql   512 Apr 15 12:00 testdb
#

```

```
#ls -l /local/mysql-data-2
```

```

total 32
drwxrwxrwx  9 mysql  mysql   512 Apr 15 07:49 .
drwxrwxrwx 20 root    root    1024 Apr 10 12:41 ..
drwxr-xr-x  2 mysql  mysql   512 Apr 14 11:16 BDB
drwxr-xr-x  2 mysql  mysql   512 Apr 14 11:14 innodb
drwxr-xr-x  2 mysql  mysql  2560 Apr 15 10:15 logs
-rw-r--r--  1 mysql  mysql  1459 Apr 14 11:13 my.cnf
drwx----- 2 mysql  mysql   512 Apr 14 11:14 mysql
-rw-rw----  1 mysql  mysql    5 Apr 15 10:10 mysqld.pid
drwx----- 2 mysql  mysql   512 Apr 15 10:10 sc3_test_database
drwx----- 2 mysql  mysql   512 Apr 14 11:14 test
drwx----- 2 mysql  mysql   512 Apr 14 11:16 testdb

```

```
# more /etc/vfstab Subset of the output
```

```

/dev/vx/dsk/dg2/vol01 /dev/vx/rdisk/dg2/vol01 /local/mysql-data-1 ufs 2
yes logging
/dev/vx/dsk/dg2/vol01 /dev/vx/rdisk/dg2/vol01 /local/mysql-data-2 ufs 2
yes logging
#

```

Note - In the above example the *Database directory* for the MySQL instance 1 is /local/mysql-data-1, whereas the *Database directory* for the MySQL instance 2 is /local/mysql-data-2.

Configuration Requirements

The requirements in this section apply to HA for MySQL only. You must meet these requirements before you proceed with your HA for MySQL installation and configuration.



Caution - Your data service configuration might not be supported if you do not adhere to these requirements.

Components and Their Dependencies for HA for MySQL

MySQL components and their dependencies – You can configure HA for MySQL data service to protect a MySQL instance and its respective components. The components, and their dependencies between each other, are briefly described below.

Note - In the above example the *Database directory* for the *MySQL* instance 1 is `/local/mysql-data-1`, whereas the *Database directory* for the *MySQL* instance 2 is `/local/mysql-data-2`.

TABLE 3 Dependencies Between HA for MySQL Components in Failover Configurations

Component	Dependency
MySQL resource in a global or non-global zones.	<p>SUNW.HAStoragePlus – This dependency is required only if the configuration uses a failover file system or file systems in a non-global zones. It is a <code>resource_offline_restart_dependency</code>.</p> <p>SUNW.LogicalHostName</p> <p>SUNW.SharedAddress</p>
MySQL resource in an HA Zone	<p>HA for Oracle Solaris Zones boot resource.</p> <p>SUNW.HAStoragePlus – This dependency is a <code>resource_offline_restart_dependency</code>.</p> <p>SUNW.LogicalHostName – This dependency is required only if the zone's boot resource does not manage the zone's IP address.</p>

Registration and Configuration File for HA for MySQL

The MySQL component has two configuration and registration files in `/opt/SUNWscmys/util`. These files enable you to register the MySQL component with Oracle Solaris Cluster and prepare a MySQL instance to be registered.

Within these files, the appropriate dependencies have been applied.

EXAMPLE 3 The `ha_mysql_config` File

```
# cd /opt/SUNWscmys
```

```

# more util/*config
:::::::::::::
ha_mysql_config
:::::::::::::
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the License).
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/CDDL.txt
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/CDDL.txt.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets [] replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2006, 2012, Oracle and/or its affiliates. All rights reserved.
#
#ident "ha_mysql_config.ksh 1.9 12/08/20"

# This file will be sourced in by ha_mysql_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
# RS - Name of the resource for the application
# RG - Name of the resource group containing RS
# SCALABLE - Flag to indicate a scalable resource creation.
#           The default is no, so any entry here triggers a scalable resource.
# LB_POLICY - Set the loadbalancing policy for a scalable MySQL service.
#           Use the values defined for the standard resource property
#           Load_balancing_policy. If you do not specify it, the defaults are
#           used.
# RS_PROP - Additional resource properties in the format for clresource create.
#           Example: "-p start_timeout=600".
#           To have the MySQL agent local zone aware, 4 variables are needed:

```

```

#     ZONE - The zone name where the MySQL Database should run
#           Optional
#     ZONEBT - The resource name which controls the zone.
#           Optional
#     PROJECT - A project in the zone that will be used for this service.
#              Specify it if you have an su - in the start stop or probe,
#              or to define the smf credentials. If the variable is not set,
#              it will be translated as :default for the sm and default
#              for the zsh component.
#           Optional
#     ZUSER - A user in the zone which is used for the smf method
#            credentials. Your smf service will run under this user.
#           Optional
#
# MySQL specific variables
#
#     BASEDIR - Name of the MySQL bin directory
#     DATADIR - Name of the MySQL Data directory
#     MYSQLUSER - Name of the user MySQL should be started of
#               LH - Name of the LogicalHostname SC resource
#     MYSQLHOST - Name of the host in /etc/hosts. For scalable/multiple-master \
#                resources, leave it empty.
#     FMUSER - Name of the MySQL fault monitor user
#     FMPASS - Name of the MySQL fault monitor user password
#             If you do not want to store the password as readable in the file, \
#             leave the FMPASS variable empty.
#     LOGDIR - Name of the directory where mysqld should store its logfile.
#     CHECK - Should HA-MySQL check MyISAM index files before start YES/NO.
#     HAS_RS - Name of the MySQL HAStoragePlus SC resource
#     NDB_CHECK - Is MySQL Cluster installed?
#               Any entry here triggers the ndb engine check. If no MySQL \
#               cluster should be checked, leave it empty.
#
#     The following examples illustrate sample parameters
#     for MySQL
#
#     BASEDIR=/usr/local/mysql
#     DATADIR=/global/mysqldata
#     MYSQLUSER=mysql
#     LH=mysqlhh
#     RS_PROP=
#     SCALABLE=
#     LB_POLICY=
#     MYSQLHOST=mysqlhh
#     FMUSER=fmuser
#     FMPASS=fmuser
#     LOGDIR=/global/mysqldata/logs
#     CHECK=YES

```



```

#      NDB_CHECK=
#

RS=
RG=
PORT=
LH=
SCALABLE=
LB_POLICY=
RS_PROP=
HAS_RS=

# local zone specific options

ZONE=
ZONE_BT=
PROJECT=

# mysql specifications

BASEDIR=
DATADIR=
MYSQLUSER=
MYSQLHOST=
FMUSER=
FMPASS=
LOGDIR=
CHECK=
NDB_CHECK=

```

EXAMPLE 4 The mysql_config File

```

cd /opt/SUNWscmys
# more util/*config
::::::::::::
mysql_config
::::::::::::
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the License).
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/CDDL.txt
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions

```

```
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/CDDL.txt.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets [] replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 2006, 2012, Oracle and/or its affiliates. All rights reserved.
#
#

#ident "@(#)mysql_config.ksh 1.11 12/08/20"

# This file will be sourced in by mysql_register and the parameters
# listed below will be used.
#

# Where is mysql installed (BASEDIR)
MYSQL_BASE=

# Mysql admin-user for localhost (Default is root)
MYSQL_USER=

# Password for mysql admin user, if you do not want to disclose this password in
# a file, leave the MYSQL_PASSWD variable empty, or take it out.
MYSQL_PASSWD=

# Configured logicalhost. For scalable or multiple master resources leave it empty.
MYSQL_HOST=

# Specify a username for a faultmonitor user
FMUSER=

# Pick a password for that faultmonitor user, if you do not want to disclose this
# password in a file, leave the FMPASS variable empty, or take it out.
FMPASS=

# Socket name for mysqld ( Should be /tmp/<logical-host>.sock )
MYSQL_SOCKET=

# Specify the physical hostname for the physical NIC that this logicalhostname
# belongs to for every node in the cluster this Resource group can get located on.
# If you use the mysql_geocontrol features to implement the MySQL replication as
```

```
# the replication protocol in Oracle Solaris Cluster geographic edition, specify all
# physical nodes of all clusters, specify at least all the nodes on both sites
# where the mysql databases can be hosted.
# IE: The logicalhost lh1 belongs to hme1 for physical-node phys-1 and
# hme3 for physical-node phys-2. The hostname for hme1 is phys-1-hme0 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="phys-1-hme0 phys-2-hme3"
# IE: If two clusters are tied together by the mysql_geocontrol features, assuming the
# mysql database on cluster one belongs to cl1-phys1-hme0 and cl1-phys2-hme3, the
# mysql database on cluster two belongs to cl2-phys1-hme2 and cl2-phys2-hme4. Then the
# MYSQL_NIC_HOSTNAME variable needs to be set to:
# MYSQL_NIC_HOSTNAME="cl1-phys1-hme0 cl1-phys2-hme3 cl2-phys1-hme2 cl2-phys2-hme4"

MYSQL_NIC_HOSTNAME=

# Where are your databases installed, (location of my.cnf)
MYSQL_DATADIR=

# Is MySQL Cluster installed?
# Any entry here triggers the ndb engine check preparation. If no MySQL cluster should
# be
# checked, leave it empty.
NDB_CHECK=
```

Explanation of the my.cnf File

The my.cnf file – The HA for MySQL data service provides two sample my.cnf files, one sample file for a master configuration and one for a slave configuration. However, ensure that at least the following parameters are set.

Note - The my.cnf file is an important file within MySQL. Refer to the MySQL documentation for complete configuration information on the parameters that follow.

The MySQL my.cnf file in [mysqld] section in a master configuration:

- The bind address parameter must be set to the defined logical host's IP name.

Note - Some of the MySQL parameters do not work with bind address if the name of the logical host is set. In these cases do not set the bind address parameter, or use the absolute address of the logical hostname.

- The **socket** parameter is defined as `/tmp/logical-host-ip-name.sock` in case of a failover deployment. If you deploy a scalable or multiple-master configuration, specify `/tmp/physical-hostname.sock`.
- The `binlog-ignore-db` parameter contains `sc3_test_database`, if `log-bin` option is being used.

The MySQL `my.cnf` file in `[mysqld]` section in a slave configuration:

- The `bind address` parameter must be set to the defined logical host's IP name.

Note - Some MySQL builds do not work with names in the `bind address` parameter. In these cases do not set the `bind address` parameter, or use the absolute IP address of the logical hostname.

- The `socket` parameter is defined as `/tmp/logical-host-IP name.sock`.
- The `binlog-ignore-db` parameter contains `sc3_test_database` if `log-bin` option is being used.
- The `master-host` parameter is the hostname where the master instance resides.
- The `master-user` parameter is the username the slave will use for identification to the master.
- The `master-password` parameter is the password the slave will use for identification to the master.

Note - Beginning in MySQL server 5.0, the use of `master-host`, `master-user` and `master-password` in the `my.cnf` file is deprecated. Configure this information in the MySQL client with the `change_master` statement.

- The `master-info-file` parameter is the location of the file that remembers where MySQL left off on the master during the replication process. This file must be placed on a global file system or a failover file system.

Installing and Configuring MySQL

This section contains the procedures you need to install and configure MySQL.

References will be made to certain directories for MySQL. The following list shows common path names for these references. For more information, refer to the [“Configuration Restrictions” on page 17](#) section.

- MySQL installed from <http://www.mysql.com> on a cluster file system, with a MySQL instance on a cluster file system (Example 1, “MySQL Instances with Cluster File Systems,” on page 18):
 - **MySQL Base directory** – /global/mysql
 - **MySQL Database directory** – /global/mysqldata
- MySQL installed on a local file system, with mysql instances on a highly available local file system (Example 2, “MySQL Instances with Highly Available Local File Systems,” on page 20):
 - **MySQL Base directory** – /usr/local/mysql
 - **MySQL Database directory** – /local/mysqldata

Determine how MySQL will be deployed in an Oracle Solaris Cluster configuration.

- Determine how many MySQL instances will be deployed.
- Determine which cluster file system will be used by each MySQL instance.
- Determine the type of the target zones where you will install MySQL. Valid zones types are the global zones or the HA Zone.

To install and configure MySQL in a global-zones configuration, complete the following tasks:

- “[Enabling MySQL to Run in a Global Zone Configuration](#)” on page 29
- “[How to Install and Configure MySQL in a Global Zone](#)” on page 31

To install and configure MySQL in an HA Zone configuration, complete the following tasks:

- “[How to Enable MySQL to run in an HA Zone Configuration](#)” on page 39
- “[How to Install and Configure MySQL in an HA Zone](#)” on page 41

You will find installation examples for each zones type in the following appendixes:

- [Appendix A, “Deployment Example: Installing MySQL in the Global Zone”](#)
- [Appendix B, “Deployment Example: Installing MySQL in the Non-Global HA Zone”](#)
- [Appendix C, “Deployment Example: Installing MySQL in a Scalable or Multiple-Master Configuration”](#)

Enabling MySQL to Run in a Global Zone Configuration

Determine whether you need to install MySQL in a failover, scalable, or multiple-master configuration and follow the appropriate procedure:

- [“How to Enable MySQL for a Failover Resource” on page 30](#)
- [“How to Enable MySQL to Run in a Scalable Configuration” on page 30](#)
- [“How to Enable MySQL to Run in a Multiple-Master Configuration” on page 31](#)

▼ How to Enable MySQL for a Failover Resource

1. On one of the nodes in the cluster that will host MySQL, become an administrator that provides `solaris.cluster.admin` authorization.
2. Register the `SUNW.gds` and `SUNW.HASStoragePlus` resource type.

```
# clresourcetype register SUNW.gds SUNW.HASStoragePlus
```

3. Create a failover resource group.

```
# clresourcegroup create MySQL-failover-resource-group
```

4. Create a resource for the MySQL disk storage.

```
# clresource create \  
-g MySQL-failover-resource-group \  
-t SUNW.HASStoragePlus \  
-p FilesystemMountPoints=MySQL-instance-mount-points MySQL-has-resource
```

5. Create a resource for the MySQL logical hostname.

```
# clreslogicalhostname \  
-g MySQL-failover-resource-group \  
-h MySQL-logical-hostname \  
MySQL-lh-resource
```

6. Enable the failover resource group that now includes the MySQL disk storage and Logical Hostname resources.

```
# clresourcegroup online -M -n current-node MySQL-failover-resource-group
```

▼ How to Enable MySQL to Run in a Scalable Configuration

1. On one of the nodes in the cluster that will host MySQL, become an administrator that provides `solaris.cluster.admin` authorization.
2. Register the `SUNW.gds` resource type.

```
# clresourcetype register SUNW.gds
```

3. Create a failover resource group.

```
# clresource create SharedAddress-failover-resource-group
```

4. Create the shared address resource.

```
# clressharedaddress create \  
-g SharedAddress-failover-resource-group \  
-h MySQL-logical-hostname \  
MySQL-lh-resource
```

5. Enable the failover resource group that now includes the shared address resources.

```
# clresourcegroup online -M -n current-node MySQL-failover-resource-group
```

6. Create a resource group to run on at minimum two nodes.

```
# clresourcegroup create -p Maximum primaries=2 \  
-p desired primaries=2 MySQL-scalable-resource-group
```

▼ How to Enable MySQL to Run in a Multiple-Master Configuration

- 1. Become an administrator that provides `solaris.cluster.admin` authorization.**
- 2. On one of the nodes in the cluster that will host MySQL, register the `SUNW.gds` resource type.**

```
# clresourcetype register SUNW.gds
```

3. Create a resource group to run on at least two nodes.

```
# clresourcegroup create -p Maximum primaries=2 \  
-p desired primaries=2 MySQL-scalable-resource-group
```

▼ How to Install and Configure MySQL in a Global Zone

Before You Begin If the HA for MySQL package (`ha-cluster/data-service/mysql`) was not installed during your initial Oracle Solaris Cluster installation, proceed to [“Installing the HA for MySQL Package” on page 44](#) to install it on your cluster. Return here to continue the installation and configuration of MySQL.

1. Ensure that you are on the node where you have enabled your resource group.

2. Install MySQL onto all nodes within Oracle Solaris Cluster.

MySQL should be installed onto a cluster file system. For a discussion of the advantages and disadvantages of installing the software on local versus cluster file systems, see “[Determining the Location of the Application Binaries](#)” in *Oracle Solaris Cluster 4.3 Data Services Planning and Administration Guide*.

Download MySQL from <http://www.mysql.com>. If you intend to use local disks for the MySQL software, you will need to repeat this step on all nodes within Oracle Solaris Cluster.

3. Create a `mysql-user` and `mysql-group` for MySQL on all nodes in the cluster that will run MySQL.

Create an entry in `/etc/group` on all nodes with Oracle Solaris Cluster.

```
# groupadd -g 1000 mysql
```

Create an entry in `/etc/passwd` on all nodes within Oracle Solaris Cluster. This user should have a locked password.

```
# useradd -u 1000 -g 1000 -d /global/mysql -s /bin/sh mysql
```

4. Change the owner and the group for MySQL binaries.

If MySQL binaries are on all nodes, then repeat this step on every node.

```
# chown -R mysql:mysql /global/mysql
```

5. Create your MySQL Database directory for your MySQL instance or instances.

```
# mkdir MySQL-Database-directory
```

Note - Refer to “[Configuration Restrictions](#)” on page 17 for a description of the *MySQL Database directory* and to “[Installing and Configuring MySQL](#)” on page 28 for a list of common path names. For a multiple-master operating system scalable deployment, perform the remaining steps of this procedure on all nodes of the cluster that are about to host the MySQL database.

The following listing shows one MySQL instance. MySQL is installed from <http://www.mysql.com> in `/global/mysql`, which is mounted as a cluster file system. The *MySQL Database Directory* for the MySQL instance is `/global/mysql-data`.

```
# cd /global/mysql
#
# ls -l
-rw-r--r--  1 mysql  mysql    19106 Dec 10 14:52 COPYING
```



```

-rw-r--r-- 1 mysql mysql 28003 Dec 10 14:52 COPYING.LIB
-rw-r--r-- 1 mysql mysql 44577 Dec 5 10:37 ChangeLog
-rw-r--r-- 1 mysql mysql 6811 Dec 10 14:53 INSTALL-BINARY
-rw-r--r-- 1 mysql mysql 1976 Dec 5 10:37 README
drwxr-xr-x 2 mysql mysql 1024 Dec 13 18:05 bin
-rwxr-xr-x 1 mysql mysql 773 Dec 10 15:34 configure
drwxr-x-- 3 mysql mysql 512 Apr 3 12:23 data
drwxr-xr-x 2 mysql mysql 1024 Dec 10 15:35 include
drwxr-xr-x 2 mysql mysql 512 Dec 10 15:35 lib
drwxr-xr-x 2 mysql mysql 512 Dec 10 15:35 man
-rw-r--r-- 1 mysql mysql 2582089 Dec 10 14:47 manual.html
-rw-r--r-- 1 mysql mysql 2239278 Dec 10 14:47 manual.txt
-rw-r--r-- 1 mysql mysql 94600 Dec 10 14:47 manual_toc.html
drwxr-xr-x 6 mysql mysql 512 Dec 10 15:35 mysql-test
drwxr-xr-x 2 mysql mysql 512 Dec 10 15:35 scripts
drwxr-xr-x 3 mysql mysql 512 Dec 10 15:35 share
drwxr-xr-x 7 mysql mysql 1024 Dec 10 15:35 sql-bench
drwxr-xr-x 2 mysql mysql 512 Dec 10 15:35 support-files
drwxr-xr-x 2 mysql mysql 512 Dec 10 15:35 tests
#

```

6. Create the MySQL `my.cnf` file.

The HA for MySQL data service provides three sample `my.cnf` files for MySQL. One sample configuration file is for a master configuration, one sample file is for a slave configuration, and one is for the server configuration in a MySQL cluster.

The content of `/opt/SUNWscmys/etc/my.cnf_sample_[master|slave|mysql_d_cluster]` provides a sample MySQL configuration file that you can use to create your MySQL instance `MySQL-Database-directory/my.cnf`. You must still edit that file to reflect your configuration values.

```
# cp /opt/SUNWscmys/etc/my.cnf_sample_master \
MySQL-Database-directory/my.cnf
```

Note - If you are about to configure a multiple-master or a scalable configuration, set the socket directive in the `my.cnf` files to `socket=/tmp/physical-host.sock`.

7. Bootstrap the MySQL instance.

This creates the privilege tables `db`, `host`, `user`, `tables_priv`, and `columns_priv` in the `mysql` database, as well as the `func` table.

```
# cd MySQL-Base-directory
# ./bin/mysqld --initialize --datadir=MySQL-Database-directory
```

8. Create a log file directory in MySQL Database Directory.

```
# mkdir MySQL-Database-Directory/logs
```

9. Create directories for your storage engines.

```
# mkdir MySQL-Database-Directory/innodb
# mkdir MySQL-Database-Directory/BDB
```

10. Change owner and group for MySQL-Database-Directory.

```
# chown -R mysql:mysql MySQL Database Directory
```

11. Change file permission for MySQL-Database-Directory/my.cnf.

```
# chmod 644 MySQL-Database-Directory/my.cnf
```

Enabling MySQL to Run in a Zone Configuration

Determine whether to install MySQL in a failover, scalable, or multiple-master configuration and follow the appropriate procedure:

- [“How to Enable MySQL for a Failover Resource” on page 34](#)
- [“How to Enable MySQL to Run in a Scalable Configuration” on page 35](#)
- [“How to Enable MySQL to Run in a Multiple-Master Configuration” on page 36](#)

▼ How to Enable MySQL for a Failover Resource

- 1. On one of the nodes in the cluster that will host MySQL, become an administrator that provides `solaris.cluster.admin` authorization.**
- 2. Create and boot your zones `MySQL-zone` on all the nodes to host your MySQL database.**

3. Register the `SUNW.gds` and `SUNW.HAStoragePlus` resource type.

```
# clresource type register SUNW.gds SUNW.HAStoragePlus
```

4. Create a failover resource group.

```
# clresourcegroup create \  
-n node1,node2 \  
MySQL-failover-resource-group
```

5. Create a resource for the MySQL disk storage.

```
# clresource create \
-g MySQL-failover-resource-group \
-t SUNW.HAStoragePlus \
-p FilesystemMountPoints=MySQL-instance-mount-points \
MySQL-has-resource
```

6. Create a resource for the MySQL logical hostname.

```
# clreslogicalhostname \
-g MySQL-failover-resource-group \
-h MySQL-logical-hostname \
MySQL-lh-resource
```

7. Enable the failover resource group that now includes the MySQL disk storage and logical hostname resources.

```
# clresourcegroup online -M -n current-node MySQL-failover-resource-group
```

▼ How to Enable MySQL to Run in a Scalable Configuration

1. On one of the nodes in the cluster that will host MySQL, become an administrator that provides `solaris.cluster.admin` authorization.

2. Register the `SUNW.gds` resource type.

```
# clresourcetype register SUNW.gds
```

3. Create and boot your zones `MySQL-zone` on all the nodes to host your MySQL database.

4. Create a failover resource group.

```
# clresourcegroup create -n node1,node2 MySQL-failover-resourcegroup
```

5. Create the shared address resource.

```
# clressharedaddress create \
-g SharedAddress-failover-resource-group \
-h MySQL-logical-hostname \
MySQL-lh-resource
```

6. Enable the failover resource group that now includes the shared address resources.

```
# clresourcegroup online -M -n current-node MySQL-failover-resource-group
```

7. **Create a resource group to run on at minimum two nodes.**

```
# clresourcegroup create -p Maximum primaries=2 \  
-p desired primaries=2 \  
-n node1,node2 \  
MySQL-scalable-resource-group
```

▼ How to Enable MySQL to Run in a Multiple-Master Configuration

1. **Become an administrator that provides `solaris.cluster.admin` authorization.**
2. **On one of the nodes in the cluster that will host MySQL, register the `SUNW.gds` resource type.**

```
# clresourcetype register SUNW.gds
```

3. **Create a resource group to run on at least two nodes.**

```
# clresourcegroup create \  
-p Maximum primaries=2 \  
-p desired primaries=2 -n node1,node2 \  
MySQL-scalable-resource-group
```

▼ How to Install and Configure MySQL in a Zone

Before You Begin If the HA for MySQL package (`ha-cluster/data-service/mysql`) was not installed during your initial Oracle Solaris Cluster installation, proceed to [“Installing the HA for MySQL Package” on page 44](#) to install it on your cluster. Return here to continue the installation and configuration of MySQL.

1. **Ensure that you are on the node where you have enabled your resource group.**
2. **Log in to your zones.**

```
# zlogin MySQL-zone
```

3. **On one of the nodes in the cluster that will host MySQL, become an administrator that provides `solaris.cluster.admin` authorization.**

4. **Install MySQL on all nodes within Oracle Solaris Cluster.**

MySQL should be installed onto a cluster file system. For a discussion of the advantages and disadvantages of installing the software on local versus cluster file systems, see [“Determining](#)

the Location of the Application Binaries” in *Oracle Solaris Cluster 4.3 Data Services Planning and Administration Guide*.

Download MySQL from <http://www.mysql.com>. If you intend to use local disks for the MySQL software, you will need to repeat this step on all nodes within Oracle Solaris Cluster.

5. Create a `mysql-user` and `mysql-group` for MySQL on all nodes in the cluster that will run MySQL.

a. Create an entry in `/etc/group` on all cluster nodes.

```
# groupadd -g 1000 mysql
```

b. Create an entry in `/etc/passwd` on all cluster nodes.

This user should have a locked password.

```
# useradd -u 1000 -g 1000 -d /global/mysql -s /bin/sh mysql
```

6. Change the owner and group for MySQL binaries.

If MySQL binaries are on all nodes, then repeat this step on every node.

```
# chown -R mysql:mysql /global/mysql
```

7. Create your MySQL Database directory for your MySQL instance or instances.

```
# mkdir MySQL-Database-directory
```

Note - Refer to “[Configuration Restrictions](#)” on page 17 for a description of the *MySQL Database directory* and to “[Installing and Configuring MySQL](#)” on page 28 for a list of common path names.

The following listing shows one MySQL instance. MySQL has been installed from <http://www.mysql.com> in `/global/mysql`, which is mounted as a cluster file system. The *MySQL Database Directory* for the MySQL instance is `/global/mysql-data`.

```
# cd /global/mysql
#
# ls -l
-rw-r--r--  1 mysql  mysql  19106 Dec 10 14:52 COPYING
-rw-r--r--  1 mysql  mysql  28003 Dec 10 14:52 COPYING.LIB
-rw-r--r--  1 mysql  mysql  44577 Dec  5 10:37 ChangeLog
-rw-r--r--  1 mysql  mysql   6811 Dec 10 14:53 INSTALL-BINARY
-rw-r--r--  1 mysql  mysql   1976 Dec  5 10:37 README
drwxr-xr-x  2 mysql  mysql   1024 Dec 13 18:05 bin
-rwxr-xr-x  1 mysql  mysql    773 Dec 10 15:34 configure
```

```

drwxr-x-- 3 mysql mysql      512 Apr  3 12:23 data
drwxr-xr-x 2 mysql mysql     1024 Dec 10 15:35 include
drwxr-xr-x 2 mysql mysql      512 Dec 10 15:35 lib
drwxr-xr-x 2 mysql mysql      512 Dec 10 15:35 man
-rw-r--r-- 1 mysql mysql    2582089 Dec 10 14:47 manual.html
-rw-r--r-- 1 mysql mysql    2239278 Dec 10 14:47 manual.txt
-rw-r--r-- 1 mysql mysql     94600 Dec 10 14:47 manual_toc.html
drwxr-xr-x 6 mysql mysql      512 Dec 10 15:35 mysql-test
drwxr-xr-x 2 mysql mysql      512 Dec 10 15:35 scripts
drwxr-xr-x 3 mysql mysql      512 Dec 10 15:35 share
drwxr-xr-x 7 mysql mysql     1024 Dec 10 15:35 sql-bench
drwxr-xr-x 2 mysql mysql      512 Dec 10 15:35 support-files
drwxr-xr-x 2 mysql mysql      512 Dec 10 15:35 tests
#

```

8. Create the MySQL `my.cnf` file according to your requirements.

The HA for MySQL data service provides three sample `my.cnf` files for MySQL. One sample configuration file is for a master configuration, one is sample file is for a slave configuration, and one for the server configuration in a MySQL cluster.

The contents of `/opt/SUNWscmys/etc/my.cnf_sample_[master|slave|mysqld_cluster]` provide a sample MySQL configuration file that you can use to create your MySQL instance `MySQL-Database-directory/my.cnf`. You must still edit that file to reflect your configuration values.

```
# cp /opt/SUNWscmys/etc/my.cnf_sample_master \
MySQL-Database-directory/my.cnf
```

Note - For a multiple-master operating system scalable deployment, perform the remaining step of this procedure on all zones of the cluster that are about to host the MySQL database. In this case, set the socket directive in the `my.cnf` file to the following value:

```
socket=/tmp/physical-host.sock
```

9. Bootstrap MySQL instance.

This creates the privilege tables `db`, `host`, `user`, `tables_priv` and `columns_priv` in the MySQL database, as well as the `func` table.

```
# cd MySQL-base-directory
# ./scripts/mysql_install_db \
--datadir=MySQL-Database-directory
```

10. Create a logfile directory in `MySQL-Database-Directory`.

```
# mkdir MySQL-Database-Directory/logs
```

11. Create directories for your storage engines.

```
# mkdir MySQL-Database-Directory/innodb
# mkdir MySQL-Database-Directory/BDB
```

12. Change the owner and group for MySQL—Database-Directory.

```
# chown -R mysql:mysql MySQL-Database-Directory
```

13. Change file permission for MySQL-Database-Directory/my.cnf.

```
# chmod 644 MySQL-Database-Directory/my.cnf
```

▼ How to Enable MySQL to run in an HA Zone Configuration

Before You Begin Ensure that the `/etc/netmasks` file has IP-address subnet and netmask entries for all logical hostnames. If necessary, edit the `/etc/netmasks` file to add any missing entries.

- 1. On one of the nodes in the cluster that will host MySQL, become an administrator that provides `solaris.cluster.admin` authorization.**
- 2. As the root role, register the `SUNW.HASStoragePlus` and the `SUNW.gds` resource types.**

```
# clresourcetype register SUNW.HASStoragePlus SUNW.gds
```

- 3. Create a failover resource group.**

```
# clresourcegroup create MySQL-resource-group
```

- 4. Create a resource for the MySQL zone's disk storage.**

```
# clresource create -t SUNW.HASStoragePlus \
-p FileSystemMountPoints=MySQL-instance-mount-points \
MySQL-has-resource
```

- 5. (Optional) If you want the protection against a total adapter failure for your public network, create a resource for the MySQL's logical hostname.**

```
# clreslogicalhostname create -g MySQL-resource-group \
-h logical-hostname \
MySQL-logical-hostname-resource-name
```

- 6. Place the resource group in the managed state.**

```
# clresourcegroup online -M MySQL-resource-group
```

7. Install the zones.

Install the zones according to the Oracle Solaris Cluster HA for Solaris Containers agent documentation, assuming that the resource name is *MySQL-zone-rs* and that the zone's name is *MySQL-zone*.

8. Verify the zone's installation.

```
# zoneadm -z MySQL-zone boot
# zoneadm -z MySQL-zone halt
```

9. Register the zone's boot component.

a. Copy the container resource boot component configuration file.

```
# cp /opt/SUNWsczones/sczbt/util/sczbt_config zones-target-configuration-file
```

b. In the configuration file, set the following variables:

```
RS=MySQL-zone-rs
RG=MySQL-resource-group
PARAMETERDIR=MySQL-zone-parameter-directory
SC_NETWORK=true|false
SC_LH=MySQL-logical-hostname-resource-name
FAILOVER=true|false
HAS_RS=MySQL-has-resource
Zonename=MySQL-zone
Zonebootopt=zone-boot-options
Milestone=zone-boot-milestone
Mounts=
```

c. Create the parameter directory for your zone's resource.

```
# mkdir MySQL-zone-parameter-directory
```

d. Execute the Oracle Solaris Cluster HA for Solaris Zones registration script.

```
# /opt/SUNWsczones/sczbt/util/sczbt_register -f zone's-target-configuration-file
```

e. Enable the zone's resource.

```
# clresource enable MySQL-zone-rs
```

10. Enable the resource group.


```
# clresourcegroup online MySQL-resource-group
```

▼ How to Install and Configure MySQL in an HA Zone

Before You Begin If the HA for MySQL package (ha-cluster/data-service/mysql) was not installed during your initial Oracle Solaris Cluster installation, proceed to [“Installing the HA for MySQL Package” on page 44](#) to install it on your cluster. Return here to continue the Installation and Configuration of MySQL.

1. **Ensure that you are on the node where you have enabled your resource group.**

2. **Log in to your zone.**

```
# zlogin MySQL-zone
```

3. **On one of the nodes in the cluster that will host MySQL, become an administrator that provides `solaris.cluster.admin` authorization.**

4. **Install MySQL.**

MySQL should be installed onto a cluster file system. For a discussion of the advantages and disadvantages of installing the software on local versus cluster file systems, see [“Determining the Location of the Application Binaries” in *Oracle Solaris Cluster 4.3 Data Services Planning and Administration Guide*](#).

Download MySQL from <http://www.mysql.com>. If you intend to use local disks for the MySQL software, you will need to repeat this step on all cluster nodes.

5. **Create a `mysql-user` and `mysql-group` for the MySQL zone that will run MySQL.**

- a. **Create an entry in `/etc/group` in the zone.**

```
# groupadd -g 1000 mysql
```

- b. **Create an entry in `/etc/passwd` on all cluster nodes.**

This user should have a locked password.

```
# useradd -u 1000 -g 1000 -d /global/mysql -s /bin/sh mysql
```

6. **Change the owner and group for MySQL binaries.**

```
# chown -R mysql:mysql /global/mysql
```

7. Create your MySQL Database directory for your MySQL instance or instances.

```
# mkdir MySQL-Database-directory
```

Note - Refer to the “[Configuration Restrictions](#)” on page 17 section for a description of the *MySQL Database directory* and to “[Installing and Configuring MySQL](#)” on page 28 for a list of common path names.

The following listing shows one MySQL instance. MySQL has been installed from <http://www.mysql.com> in /global/mysql, which is mounted as a cluster file system. The *MySQL Database Directory* for the MySQL instance is /global/mysql-data.

```
# cd /global/mysql
#
# ls -l
-rw-r--r--  1 mysql  mysql  19106 Dec 10 14:52 COPYING
-rw-r--r--  1 mysql  mysql  28003 Dec 10 14:52 COPYING.LIB
-rw-r--r--  1 mysql  mysql  44577 Dec  5 10:37 ChangeLog
-rw-r--r--  1 mysql  mysql  6811 Dec 10 14:53 INSTALL-BINARY
-rw-r--r--  1 mysql  mysql  1976 Dec  5 10:37 README
drwxr-xr-x  2 mysql  mysql  1024 Dec 13 18:05 bin
-rwxr-xr-x  1 mysql  mysql   773 Dec 10 15:34 configure
drwxr-x--  3 mysql  mysql   512 Apr  3 12:23 data
drwxr-xr-x  2 mysql  mysql  1024 Dec 10 15:35 include
drwxr-xr-x  2 mysql  mysql   512 Dec 10 15:35 lib
drwxr-xr-x  2 mysql  mysql   512 Dec 10 15:35 man
-rw-r--r--  1 mysql  mysql 2582089 Dec 10 14:47 manual.html
-rw-r--r--  1 mysql  mysql 2239278 Dec 10 14:47 manual.txt
-rw-r--r--  1 mysql  mysql  94600 Dec 10 14:47 manual_toc.html
drwxr-xr-x  6 mysql  mysql   512 Dec 10 15:35 mysql-test
drwxr-xr-x  2 mysql  mysql   512 Dec 10 15:35 scripts
drwxr-xr-x  3 mysql  mysql   512 Dec 10 15:35 share
drwxr-xr-x  7 mysql  mysql  1024 Dec 10 15:35 sql-bench
drwxr-xr-x  2 mysql  mysql   512 Dec 10 15:35 support-files
drwxr-xr-x  2 mysql  mysql   512 Dec 10 15:35 tests
#
```

8. Create the MySQL my.cnf file.

The HA for MySQL data service provides three sample my.cnf files for MySQL. One sample configuration file is for a master configuration, one sample file is for a slave configuration, and one is for the server configuration in a MySQL cluster.

The contents of /opt/SUNWscmys/etc/my.cnf_sample_[master|slave|mysqld_cluster] provides a sample MySQL configuration file that you can use to create your MySQL instance *MySQL-Database-directory/my.cnf*. You must still edit that file to reflect your configuration values.

```
# cp /opt/SUNWscmys/etc/my.cnf_sample_master \
MySQL-Database-directory/my.cnf
```

9. Bootstrap the MySQL instance.

This creates the privilege tables `db`, `host`, `user`, `tables_priv` and `columns_priv` in the MySQL database, as well as the `func` table.

```
# cd MySQL-Base-director

# ./scripts/mysql_install_db \
--datadir=MySQL-Database-directory
```

10. Create a log file directory in *MySQL-Database-Directory*.

```
# mkdir MySQL-Database-Directory/logs
```

11. Create directories for your storage engines.

```
# mkdir MySQL-Database-Directory/innodb
# mkdir MySQL-Database-Directory/BDB
```

12. Change owner and group for *MySQL-Database-Directory*.

```
# chown -R mysql:mysql MySQL-Database-Directory
```

13. Change file permission for *MySQL-Database-Directory/my.cnf*.

```
# chmod 644 MySQL-Database-Directory/my.cnf
```

Verifying the Installation and Configuration of MySQL

This section contains the procedure you need to verify the installation and configuration.

▼ How to Verify the Installation and Configuration of MySQL

This procedure does not verify that your application is highly available because you have not yet installed your data service.

Note - Before verifying the installation and configuration of MySQL, ensure that the logical hostname for the MySQL is available. You will need to complete the appropriate registration and configuration procedure in this document according to your zone type.

1. (Optional) Log in to your target zone.

```
# zlogin mysql-zone
```

2. Start the MySQL Server for this instance.

```
# cd MySQL-Base-directory
# ./bin/mysqld --defaults-file=MySQL-Database-directory/my.cnf \
--basedir=MySQL-Base-directory \
--datadir=MySQL-Database-directory \
--user=mysql --pid-file=MySQL-Database-directory/mysqld.pid &
```

3. Connect to the MySQL instance.

```
# MySQL-Base-directory/bin/mysql -S /tmp/Logical-host.sock -uroot
```

4. Stop the MySQL server instance.

```
# kill -TERM `cat MySQL-Database-directory/mysqld.pid`
```

5. (Optional) Leave the target zone.

Installing the HA for MySQL Package

If you did not install the HA for MySQL package during your initial Oracle Solaris Cluster installation, perform this procedure to install the package.

▼ How to Install the HA for MySQL Package

Perform this procedure on each cluster node where you want the HA for MySQL software to run.

1. On the cluster node where you are installing the data service package, assume the root role.

2. **Ensure that the data service package is available from the configured publisher and that the `solaris` and `ha-cluster` publishers are valid.**

```
# pkg list -a ha-cluster/data-service/mysql
# pkg publisher
PUBLISHER                TYPE    STATUS  P  LOCATION
solaris                   origin  online  F  solaris-repository
ha-cluster                 origin  online  F  ha-cluster-repository
```

For information about setting the `solaris` publisher, see [“Adding, Modifying, or Removing Package Publishers” in *Adding and Updating Software in Oracle Solaris 11.3*](#).

Tip - Use the `-nv` options whenever you install or update to see what changes will be made, such as which versions of which packages will be installed or updated and whether a new BE will be created.

If you do not get any error messages when you use the `-nv` options, run the command again without the `-n` option to actually perform the installation or update. If you do get error messages, run the command again with more `-v` options (for example, `-nvv`) or more of the package FMRI pattern to get more information to help you diagnose and fix the problem. For troubleshooting information, see [Appendix A, “Troubleshooting Package Installation and Update,” in *Adding and Updating Software in Oracle Solaris 11.3*](#).

3. **Install the HA for MySQL software package.**

```
# pkg install ha-cluster/data-service/mysql
```

4. **Verify that the package installed successfully.**

```
$ pkg info ha-cluster/data-service/mysql
```

Installation is successful if output shows that State is Installed.

5. **Perform any necessary updates to the Oracle Solaris Cluster software.**

For instructions on updating your software, see [Chapter 11, “Updating Your Software” in *Oracle Solaris Cluster 4.3 System Administration Guide*](#).

Registering and Configuring HA for MySQL

This section contains the procedures you need to configure HA for MySQL. According to your zone type, you need to complete one of the following tasks.

- [“How to Register and Configure HA for MySQL as a Failover Service in a Global Zone Configuration” on page 46](#)
- [“How to Register and Configure HA for MySQL as a Failover Service in a Zone Configuration” on page 52](#)
- [“How to Register and Configure HA for MySQL as a Failover Service in an HA Zone Configuration” on page 58](#)

▼ How to Register and Configure HA for MySQL as a Failover Service in a Global Zone Configuration

This procedure assumes that you installed the data service packages during your initial Oracle Solaris Cluster installation.

If you did not install the HA for MySQL packages as part of your initial Oracle Solaris Cluster installation, go to [“Installing the HA for MySQL Package” on page 44](#).

1. **On a node in the cluster that hosts MySQL, become an administrator that provides `solaris.cluster.admin` authorization.**
2. **Start the MySQL Server instance manually.**

```
# cd MySQL-Base- directory

# ./bin/mysqld --defaults-file=MySQL-Database-directory/my.cnf \
--basedir=MySQL-Base-directory \
--datadir=MySQL-Database-directory \
--user=mysql \
--pid-file=MySQL-Database-directory/mysqld.pid &
```

3. **Configure the administrator password for the administrative user.**

```
# MySQL-Database-directory/bin/mysqladmin \
-S /tmp/logical-host.sock password 'admin-password'
```

4. **Add the administrative user for locally accessing a MySQL instance with a MySQL `logical-host-ip-name`.**

Note - If you want to access the MySQL instance only through the socket (`localhost`), omit this step.

When bootstrapping MySQL the command `mysql_install_db` creates two administrative users, one belonging to `localhost` and one belonging to the node on which `mysql_install_db` was executed.

Add an administrative user for every global-cluster node that runs this MySQL instance.

Note - If the node name and the hostname for the physical interface are different, use the hostname for the physical interface.

The following is an example for a MySQL instance.

```
# mysql -S /tmp/haostix1.sock -uroot -padmin-password
mysql> use mysql;
mysql> GRANT ALL ON *.* TO 'root'@'clusterix2' IDENTIFIED BY 'rootpasswd';
mysql> GRANT ALL ON *.* TO 'root'@'clusterix1' IDENTIFIED BY 'rootpasswd';
mysql> exit;
```

Note - You must manually add `Grant_priv` to the administrative users. See the MySQL administration documentation.

The following is an example for a MySQL 4.x or 5.0.x instance.

```
# mysql -S /tmp/haostix1.sock -uroot -padmin-password
mysql> use mysql;
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='clusterix1';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='clusterix2';
mysql> exit;
```

Note - If you experience any problems here, refer to the MySQL administration documentation.

5. Copy the MySQL configuration files to your home directory.

```
# cp /opt/SUNWscmys/util/ha_mysql_config /home-dir
# cp /opt/SUNWscmys/util/mysql_config /home-dir
```

6. Create a `faultmonitor-user` and a `test-database` for the MySQL instance.

```
# cd /home-dir
```

Edit the `mysql_config` file and follow the comments within that file.

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
```

```
# Common Development and Distribution License (the License).
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/CDDL.txt
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/CDDL.txt.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets [] replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 2006,2012, Oracle and/or its affiliates. All rights reserved.
#

#ident "@(#)mysql_config.ksh 1.11 12/08/20"

# This file will be sourced in by mysql_register and the parameters
# listed below will be used.
#

# Where is mysql installed (BASEDIR)
MYSQL_BASE=

# MySQL admin-user for localhost (Default is root)
MYSQL_USER=

# Password for mysql admin user, if you do not want to disclose this password in
# a file, leave the MYSQL_PASSWD variable empty, or take it out.
MYSQL_PASSWD=

# Configured logicalhost. For scalable or multiple-master resources, leave it empty.
MYSQL_HOST=

# Specify a username for a faultmonitor user
FMUSER=

# Pick a password for that faultmonitor user, if you do not want to disclose this
# password in a file, leave the FMPASS variable empty, or take it out.
FMPASS=

# Socket name for mysqld ( Should be /tmp/<logical-host>.sock )
```



```

MYSQL_SOCKET=

# Specify the physical hostname for the physical NIC that this logical hostname
# belongs to for every node in the cluster this resource group can be located on.
# If you use the mysql_geocontrol features to implement the MySQL replication as
# the replication protocol in Oracle Solaris Cluster geographic edition, specify all
# physical nodes of all clusters. Specify at least all the nodes on both sites
# where the MySQL databases can be hosted.
# IE: The logicalhost lh1 belongs to hme1 for physical-node phys-1 and
# hme3 for physical-node phys-2. The hostname for hme1 is phys-1-hme0 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="phys-1-hme0 phys-2-hme3"
# IE: If two clusters are tied together by the mysql_geocontrol features, assuming the
# MySQL database on cluster one belongs to cl1-phys1-hme0 and cl1-phys2-hme3, the
# MySQL database on cluster two belongs to cl2-phys1-hme2 and cl2-phys2-hme4. Then the
# MYSQL_NIC_HOSTNAME variable needs to be set to:
# MYSQL_NIC_HOSTNAME="cl1-phys1-hme0 cl1-phys2-hme3 cl2-phys1-hme2 cl2-phys2-hme4"

MYSQL_NIC_HOSTNAME=

# Where are your databases installed? (location of my.cnf)
MYSQL_DATADIR=

# Is MySQL Cluster installed?
# Any entry here triggers the ndb engine check preparation. If no MySQL cluster should
# be
# checked, leave it empty.
NDB_CHECK=

```

The following is an example for a MySQL instance.

```

MYSQL_BASE=global/mysql
MYSQL_USER=root
MYSQL_PASSWD=
MYSQL_HOST=hahostix1
FMUSER=fmuser
FMPASS=
MYSQL_SOCKET=tmp/hahostix1.sock
MYSQL_NIC_HOSTNAME=clusterix1 clusterix2
MYSQL_DATADIR=global/mysql-data

```

After editing `mysql_config`, you must run the `mysql_register` script.

```
# /opt/SUNWscmys/util/mysql_register -f home-dir/mysql_config
```

7. Stop the MySQL Server instance manually.

```
# kill -TERM `cat MySQL-Database-directry/mysqlld.pid`
```

8. Create and register MySQL as a failover data service.

a. Navigate to the *home-dir* directory.

```
# cd home-dir
```

b. Edit the `ha_mysql_config` file and follow the comments within that file.

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the License).
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/CDDL.txt
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/CDDL.txt.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets [] replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2006, 2012, Oracle and/or its affiliates. All rights reserved.
#

#ident "@(#)ha_mysql_config.ksh      1.9      12/08/20"

# This file will be sourced in by ha_mysql_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#       RS - name of the resource for the application
#       RG - name of the resource group containing RS
# SCALABLE - flag to indicate a scalable resource creation.
#           The default is no, so any enty here triggers a scalable resource.
# LB_POLICY - Set the loadbalancing policy for a scalable mysql service.
#           Use the values defined for the standard resource property
#           Load_balancing_policy. If you do not specify it, the defaults are
#           used.
```

```

# RS_PROP - Additional resource properties in the format for clresource create,
#           example "-p start_timeout=600"
#
#           To have the mysql agent local zone aware, 4 Variables are needed:
#           ZONE - the zone name where the Mysql Database should run in
#                 Optional
#           ZONEBT - The resource name which controls the zone.
#                 Optional
#           PROJECT - A project in the zone, that will be used for this service
#                   specify it if you have an su - in the start stop or probe,
#                   or to define the smf credentials. If the variable is not set,
#                   it will be translated as :default for the sm and default
#                   for the zsh component
#                   Optional
#           ZUSER - A user in the the zone which is used for the smf method
#                  credentials. Your smf service will run under this user
#                  Optional
#
# Mysql specific Variables
#
#   BASEDIR - name of the Mysql bin directory
#   DATADIR - name of the Mysql Data directory
#   MYSQLUSER - name of the user Mysql should be started of
#   LH - name of the LogicalHostname SC resource
#   MYSQLHOST - name of the host in /etc/hosts. For scalable/multiple master resources
#               leave it empty
#   FMUSER - name of the Mysql fault monitor user
#   FMPASS - name of the Mysql fault monitor user password
#           If you do not want to store the password readable in a file, leave the
#           FMPASS variable
#           empty.
#   LOGDIR - name of the directory mysqld should store it's logfile.
#   CHECK - should HA-MySQL check MyISAM index files before start YES/NO.
#   HAS_RS - name of the MySQL HAStoragePlus SC resource
#   NDB_CHECK - Is MySQL Cluster installed?
#             Any entry here triggers the ndb engine check, if no MySQL cluster
#             should be checked
#             leave it empty.
#
#

```

The following is an example for a MySQL instance.

```

RS=mysql-res
RG=mysql-rg
BASEDIR=/global/mysql
DATADIR=/global/mysql-data
MYSQLUSER=mysql
LH=hahostix1

```

```
MYSQLHOST=hahostix1
FMUSER=fmuser
FMPASS=
LOGDIR=/global/mysql-data/logs
CHECK=YES
HAS_RS=mysql-has-res
ZONE=
ZONE_BT=
PROJECT=
```

- c. (Optional) If you did not specify the FMPASS variable, call the following script on one node or zone that can host the MySQL database.

```
# /opt/SUNWscmys/util/ha_mysql_register -f home-dir/ha_mysql_config -e
```

- d. Register the MySQL resource.

```
# /opt/SUNWscmys/util/ha_mysql_register -f home-dir/ha_mysql_config
```

9. Enable each MySQL resource.

Repeat this step for each MySQL instance, if multiple instances were created.

```
# clresource status
# clresource enable MySQL-resource
```

▼ How to Register and Configure HA for MySQL as a Failover Service in a Zone Configuration

This procedure assumes that you installed the data service packages during your initial Oracle Solaris Cluster installation.

If you did not install the HA for MySQL packages as part of your initial Oracle Solaris Cluster installation, go to [“Installing the HA for MySQL Package” on page 44](#).

1. Log in to the zone on the node that hosts your *MySQLresource-group*.
2. On the node's zone in the cluster that hosts MySQL, become an administrator that provides `solaris.cluster.admin` authorization.
3. Start the MySQL server instance manually.

```
# cd MySQL-base-directory
```

```
# ./bin/mysqld --defaults-file=MySQL-Database-directory/my.cnf \  
--basedir=MySQL-Base-directory \  
--datadir=MySQL-Database-directory \  
--user=mysql \  
--pid-file=MySQL-Database-directory/mysqld.pid &
```

4. Configure the administrator password for the administrative user.

```
# MySQL-Database-directory/bin/mysqladmin \  
-S /tmp/logical-host.sock password 'admin-password'
```

5. Add the administrative user for accessing locally a MySQL instance with a MySQL logical-host-ip-name.

Note - If you want to access the MySQL instance only through the socket (localhost), omit this step.

When bootstrapping MySQL the command `mysql_install_db` creates two administrative users, one belonging to localhost and one belonging to the node on which `mysql_install_db` was executed.

Add an administrative user for every physical node in the cluster that runs this MySQL instance.

Note - If the node name and the hostname for the physical interface are different, use the hostname for the physical interface.

The following is an example for a MySQL instance.

```
# mysql -S /tmp/hahostix1.sock -uroot -padmin-password  
mysql> use mysql;  
mysql> GRANT ALL ON *.* TO 'root'@'zone1' IDENTIFIED BY 'rootpasswd';  
mysql> GRANT ALL ON *.* TO 'root'@'zone2' IDENTIFIED BY 'rootpasswd';  
mysql> exit;
```

Note - You have to manually add `Grant_priv` to the administrative users. See the MySQL administration documentation.

The following is an example for a MySQL 4.x or 5.0.x instance.

```
# mysql -S /tmp/hahostix1.sock -uroot -padmin-password  
mysql> use mysql;  
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='zone1';  
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='zone2';
```

```
mysql> exit;
```

Note - If you experience any problems here, refer to the MySQL administration documentation.

6. Copy the MySQL configuration file to your home directory.

```
# cp /opt/SUNWscmys/util/mysql_config /home-dir
```

7. Create a fault-monitor user and a test database for the MySQL instance.

```
# cd home-dir
```

Edit the `mysql_config` file and follow the comments within that file.

```
# Copyright (c) 2006, 2012, Oracle and/or its affiliates. All rights reserved.
#
#
#ident "@(#)mysql_config.ksh 1.11 12/08/20"
#
# This file will be sourced in by mysql_register and the parameters
# listed below will be used.
#

# Where is mysql installed (BASEDIR)
MYSQL_BASE=

# MySQL admin-user for localhost (Default is root)
MYSQL_USER=

# Password for MySQL admin user, if you do not want to disclose this password in
# a file, leave the MYSQL_PASSWD variable empty, or take it out.
MYSQL_PASSWD=

# Configured logicalhost
MYSQL_HOST=

# Specify a username for a faultmonitor user
FMUSER=

# Pick a password for that faultmonitor user, if you do not want to disclose this
# password in a file, leave the FMPASS variable empty, or take it out.
FMPASS=

# Socket name for mysqld ( Should be /tmp/<logical-host>.sock )
MYSQL_SOCKET=

# Specify the physical hostname for the physical NIC that this
# logical hostname
```

```
# belongs to for every node in the cluster this resource group can be
# located on.

# IE: The logicalhost lh1 belongs to hme1 for physical-node phys-1 and
# hme3 for physical-node phys-2. The hostname for hme1 is phys-1-hme0 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="phys-1-hme0 phys-2-hme3"

MYSQL_NIC_HOSTNAME=

# Where are your databases installed? (location of my.cnf)

MYSQL_DATADIR=
```

The following is an example for a MySQL instance.

```
MYSQL_BASE=/global/mysql
MYSQL_USER=root
MYSQL_PASSWD=
MYSQL_HOST=hahostix1
FMUSER=fmuser
FMPASS=
MYSQL_SOCKET=/tmp/hahostix1.sock
MYSQL_NIC_HOSTNAME="zone1 zone2"
MYSQL_DATADIR=/global/mysql-data
```

After editing `mysql_config`, you must run the `mysql_register` script.

```
# /opt/SUNWscmys/util/mysql_register -f home-dir/mysql_config
```

8. Stop the MySQL server instance manually.

```
# kill -TERM `cat MySQL-Database-directory/mysqlld.pid`
```

9. Copy the MySQL configuration file to your home directory.

```
# cp /opt/SUNWscmys/util/ha_mysql_config /home-dir
```

10. Create and register MySQL as a failover data service.

a. Navigate to the `home-dir` directory.

```
# cd home-dir
```

b. Edit the `ha_mysql_config` file and follow the comments within that file.

```
#
```

```
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the License).
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/CDDL.txt
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/CDDL.txt.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets [] replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2006, 2012, Oracle and/or its affiliates. All rights reserved.
#

#ident "@(#)ha_mysql_config.ksh 1.9 12/08/20"

# This file will be sourced in by ha_mysql_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#     RS - name of the resource for the application
#     RG - name of the resource group containing RS
# SCALABLE - flag to indicate a scalable resource creation.
#           The default is no, so any enty here triggers a scalable resource.
# LB_POLICY - Set the loadbalancing policy for a scalable mysql service.
#           Use the values defined for the standard resource property
#           Load_balancing_policy. If you do not specify it, the defaults are
#           used.
# RS_PROP - Additional resource properties in the format for clresource create,
#           example "-p start_timeout=600"
#
#           To have the mysql agent local zone aware, 4 Variables are needed:
#     ZONE - the zone name where the Mysql Database should run in
#           Optional
#     ZONEBT - The resource name which controls the zone.
#           Optional
#     PROJECT - A project in the zone, that will be used for this service
```



```

#           specify it if you have an su - in the start stop or probe,
#           or to define the smf credentials. If the variable is not set,
#           it will be translated as :default for the sm and default
#           for the zsh component
#           Optional
#   ZUSER - A user in the the zone which is used for the smf method
#           credentials. Your smf service will run under this user
#           Optional
#
# Mysql specific Variables
##   BASEDIR - name of the Mysql bin directory
#   DATADIR - name of the Mysql Data directory
#   MYSQLUSER - name of the user Mysql should be started of
#           LH - name of the LogicalHostname SC resource
#   MYSQLHOST - name of the host in /etc/hosts. For scalable/multiple master resources
#           leave it empty
#   FMUSER - name of the Mysql fault monitor user
#   FMPASS - name of the Mysql fault monitor user password
#           If you do not want to store the password as readable in a file, leave
#           the FMPASS variable
#           empty.
#   LOGDIR - name of the directory mysqld should store it's logfile.
#   CHECK - should HA-MySQL check MyISAM index files before start YES/NO.
#   HAS_RS - name of the MySQL HAStoragePlus SC resource
#   NDB_CHECK - Is MySQL Cluster installed?
#           Any entry here triggers the ndb engine check, if no MySQL cluster
#           should be checked
#           leave it empty.
#
#

```

The following is an example for a MySQL instance.

```

RS=mysql-res
RG=mysql-rg
BASEDIR=/global/mysql
DATADIR=/global/mysql-data
MYSQLUSER=mysql
LH=hahostix1
MYSQLHOST=hahostix1
FMUSER=fmuser
FMPASS=
LOGDIR=/global/mysql-data/logs
CHECK=YES
HAS_RS=mysql-has-res
ZONE=
ZONE_BT=
PROJECT=

```

- c. (Optional) If you did not specify the FMPASS variable, call the following script in one zone that can host the MySQL database.

```
# /opt/SUNWscmys/util/ha_mysql_register -f home-dir/ha_mysql_config -e
```

- d. Register the MySQL resource.

```
# /opt/SUNWscmys/util/ha_mysql_register -f home-dir/ha_mysql_config
```

11. Enable each MySQL resource.

Repeat this step for each MySQL instance, if multiple instances were created.

```
# clresource status
# clresource enable MySQL-resource
```

▼ How to Register and Configure HA for MySQL as a Failover Service in an HA Zone Configuration

This procedure assumes that you installed the data service packages during your initial Oracle Solaris Cluster installation.

If you did not install the HA for MySQL packages as part of your initial Oracle Solaris Cluster installation, go to [“Installing the HA for MySQL Package” on page 44](#).

1. Log in to the zone on the node that hosts your MySQL resource-group.
2. On the node's zone in the cluster that hostsMySQL, become an administrator that provides `solaris.cluster.admin` authorization.
3. Start the MySQL server instance manually.

```
# cd MySQL-base-directory
# ./bin/mysqld --defaults-file=MySQLDatabaseDirectory/my.cnf \
--basedir=MySQL-Base-directory \
--datadir=MySQL-Database-directory \
--user=mysql \
--pid-file=MySQL-Database-directory/mysqld.pid &
```

4. Configure the administrator password for the administrative user.

```
# MySQL-Database-directory/bin/mysqladmin \
```

```
-S /tmp/logical-host.sock password 'admin-password'
```

5. **Add the administrative user for locally accessing a MySQL instance with a MySQL logical-host-ip-name.**

Note - If you want to access the MySQL instance only through the socket (localhost), omit this step.

When bootstrapping MySQL the command `mysql_install_db` creates two administrative users, one belonging to `localhost` and one belonging to the node on which `mysql_install_db` was executed.

Add an administrative user for every global-cluster node that runs this MySQL instance.

Note - If the node name and the hostname for the physical interface are different, use the hostname for the physical interface.

The following is an example for a MySQL instance.

```
# mysql -S /tmp/hahostix1.sock -uroot -padmin-password
mysql> use mysql;
mysql> GRANT ALL ON *.* TO 'root'@'zone1' IDENTIFIED BY 'rootpasswd';
mysql> exit;
```

Note - You have to manually add `Grant_priv` to the administrative users. See the MySQL administration documentation.

The following is an example for a MySQL 4.x or 5.0.x instance.

```
# mysql -S /tmp/hahostix1.sock -uroot -padmin-password
mysql> use mysql;
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='zone';
mysql> exit;
```

Note - If you experience any problems, refer to the MySQL administration documentation.

6. **Copy the MySQL configuration file to your home directory.**

```
# cp /opt/SUNWscmys/util/mysql_config /home-dir
```

7. **Create a fault-monitor user and a test database for the MySQL instance.**

- a. **Navigate to the `home-dir` directory.**

```
# cd home-dir
```

b. Edit the `mysql_config` file and follow the comments within that file.

```
#
# Copyright (c) 2006, 2012, Oracle and/or its affiliates. All rights reserved.
#
#
#ident "@(#)mysql_config.ksh 1.11 12/08/20"
# This file will be sourced in by mysql_register and the parameters
# listed below will be used.
#

# Where is MySQL installed (BASEDIR)
MYSQL_BASE=

# MySQL admin-user for localhost (Default is root)
MYSQL_USER=

# Password for MySQL admin user, if you do not want to disclose this password in
# a file, leave the MYSQL_PASSWD variable empty, or take it out.
MYSQL_PASSWD=

# Configured logicalhost
MYSQL_HOST=

# Specify a username for a faultmonitor user
FMUSER=

# Pick a password for that faultmonitor user, if you do not want to disclose this
# password in a file, leave the FMPASS variable empty, or take it out.
FMPASS=

# Socket name for mysqld ( Should be /tmp/<logical-host>.sock )
MYSQL_SOCKET=

# Specify the physical hostname for the physical NIC that this
# logical hostname

# belongs to for every node in the cluster this resource group can be
# located on.

# IE: The logicalhost lh1 belongs to hme1 for physical-node phys-1 and
# hme3 for physical-node phys-2. The hostname for hme1 is phys-1-hme0 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="phys-1-hme0 phys-2-hme3"
```

```
MYSQL_NIC_HOSTNAME=
```

```
# Where are your databases installed? (location of my.cnf)
```

```
MYSQL_DATADIR=
```

The following is an example for a MySQL instance.

```
MYSQL_BASE=/global/mysql
MYSQL_USER=root
MYSQL_PASSWD=
MYSQL_HOST=hahostix1
FMUSER=fmuser
FMPASS=
MYSQL_SOCKET=/tmp/hahostix1.sock
MYSQL_NIC_HOSTNAME="zone1"
MYSQL_DATADIR=/global/mysql-data
```

- c. After editing `mysql_config`, you must run the `mysql_register` script.**

```
# /opt/SUNWscmys/util/mysql_register -f home-dir/mysql_config
```

- 8. Stop the MySQL server instance manually.**

```
# kill -TERM `cat MySQL-Database-directry/mysql_d.pid`
```

- 9. Leave the zone.**

- 10. In the global zone of the cluster node that hosts MySQL, become an administrator that provides `solaris.cluster.admin` authorization.**

- 11. Copy the MySQL configuration file to your home directory.**

```
# cp /opt/SUNWscmys/util/ha_mysql_config /home-dir
```

- 12. Create and register MySQL as a failover data service.**

- a. Navigate to `home-dir` directory.**

```
# cd home-dir
```

- b. Edit the `ha_mysql_config` file and follow the comments within that file.**

```
#
# CDDL HEADER START
```

```
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the License).
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/CDDL.txt
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/CDDL.txt.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets [] replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 2006, 2012, Oracle and/or its affiliates. All rights reserved.
#

#ident "@(#)ha_mysql_config.ksh      1.9      12/08/20"

# This file will be sourced in by ha_mysql_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#       RS - name of the resource for the application
#       RG - name of the resource group containing RS
# SCALABLE - flag to indicate a scalable resource creation.
#           The default is no, so any enty here triggers a scalable resource.
# LB_POLICY - Set the loadbalancing policy for a scalable mysql service.
#           Use the values defined for the standard resource property
#           Load_balancing_policy. If you do not specify it, the defaults are
#           used.
# RS_PROP - Additional resource properties in the format for clresource create,
#           example "-p start_timeout=600"
#
#       To have the mysql agent local zone aware, 4 Variables are needed:
#       ZONE - the zone name where the Mysql Database should run in
#           Optional
#       ZONEBT - The resource name which controls the zone.
#           Optional
#       PROJECT - A project in the zone, that will be used for this service
#           specify it if you have an su - in the start stop or probe,
```

```
#           or to define the smf credentials. If the variable is not set,
#           it will be translated as :default for the sm and default
#           for the zsh component
#           Optional
#   ZUSER - A user in the the zone which is used for the smf method
#           credentials. Your smf service will run under this user
#           Optional
#
# Mysql specific Variables
#
#   BASEDIR - name of the Mysql bin directory
#   DATADIR - name of the Mysql Data directory
#   MYSQLUSER - name of the user Mysql should be started of
#   LH - name of the LogicalHostname SC resource
#   MYSQLHOST - name of the host in /etc/hosts. For scalable/multiple master resources
#               leave it empty
#   FMUSER - name of the Mysql fault monitor user
#   FMPASS - name of the Mysql fault monitor user password
#           If you do not want to store the password as readable in a file, leave
#           the FMPASS variable
#           empty.
#   LOGDIR - name of the directory mysqld should store it's logfile.
#   CHECK - should HA-MySQL check MyISAM index files before start YES/NO.
#   HAS_RS - name of the MySQL HAStoragePlus SC resource
#   NDB_CHECK - Is MySQL Cluster installed?
#           Any entry here triggers the ndb engine check, if no MySQL cluster
#           should be checked
#           leave it empty.
#
#
```

The following is an example for a MySQL instance.

```
RS=mysql-res
RG=mysql-rg
BASEDIR=/global/mysql
DATADIR=/global/mysql-data
MYSQLUSER=mysql
LH=hahostix1
MYSQLHOST=hahostix1
FMUSER=fmuser
FMPASS=
LOGDIR=/global/mysql-data/logs
CHECK=YES
HAS_RS=mysql-has-res
ZONE=zone1
ZONE_BT=zone1-rs
PROJECT=MySQL-project
```

- c. (Optional) If you did not specify the FMPASS variable, make the `ha_mysql_config` file available in the zone and call the following script in the zone that can host the MySQL database.

```
# /opt/SUNWscmys/util/ha_mysql_register -f home-dir/ha_mysql_config -e
```

- d. Register the MySQL resource.

```
# /opt/SUNWscmys/util/ha_mysql_register -f home-dir/ha_mysql_config
```

13. Enable each MySQL resource.

Repeat this step for each MySQL instance, if multiple instances were created.

```
# clresource status
# clresource enable MySQL-resource
```

▼ How to Modify Parameters in the HA for MySQL Manifest

Perform this task to change parameters in the HA for MySQL manifest and to validate the parameters in the HA Zone. Parameters for the HA for MySQL manifest are stored as properties of the SMF service. To modify parameters in the manifest, change the related properties in the SMF service then validate the parameter changes.

1. On the zone's console, become an administrator that provides `solaris.cluster.modify` and `solaris.cluster.admin` authorizations.
2. Change the Solaris Service Management Facility (SMF) properties for the HA for MySQL manifest.

```
# svccfg svc:/application/sczone-agents:resource
```

For more information, see the [svccfg\(1M\)](#) man page.

3. Validate the parameter changes.

```
# /opt/SUNWscmys/bin/control_mysql validate resource
```

Messages for this command are stored in the `/var/adm/messages/` directory of the HA Zone.

4. Disconnect from the HA Zone console.

▼ How to Remove an HA for MySQL Resource From an HA Zone

1. Become an administrator that provides `solaris.cluster.modify` and `solaris.cluster.admin` authorizations.
2. Disable and remove the resource that is used by the HA for MySQL data service.

```
# clresource disable resource
# clresource delete resource
```

3. Log in as the `root` role to the HA Zone's console.
4. Unregister HA for MySQL from the SMF service.

```
# /opt/SUNWscmys/util/ha_mysql_smf_remove -f filename
```

`-f` Specifies the configuration file name.

`filename` The name of the configuration file that you used to register HA for MySQL with the SMF service.

Note - If you no longer have the configuration file that you used to register HA for MySQL with the SMF service, create a replacement configuration file:

1. Make a copy of the default file, `/opt/SUNWscmys/util/ha_mysql_config`.
2. Set the `ZONE` and `RS` parameters with the values that are used by the data service.
3. Run the `ha_mysql_smf_remove` command and use the `-f` option to specify this configuration file.

-
5. Disconnect from the HA Zone's console.

▼ How to Add an HA for MySQL Resource in a Scalable or Multiple-Master Configuration

Before You Begin Ensure that you installed the data service packages during your initial Oracle Solaris Cluster installation. If you did not install the HA for MySQL packages as part of your initial Oracle Solaris Cluster installation, go to [“Installing the HA for MySQL Package” on page 44](#).

Note - Perform the following steps, including stopping the MySQL server on every node or zone that hosts the MySQL server.

1. (Optional) Log in to the target zone.

```
# zlogin zone-name
```

Note - If you install MySQL in a zone cluster, you are in the target zone already.

2. On the cluster node that hosts MySQL, become an administrator that provides `solaris.cluster.admin` RBAC authorization.

3. Start the MySQL Server instance manually.

```
# cd MySQL-Base-directory
# ./bin/mysqld --defaults-file=MySQL-Database-directory/my.cnf \
--basedir=MySQL-Base-directory \
--datadir=MySQL-Database-directory --user=mysql \
--pid-file=MySQL-Database-directory/mysqld.pid &
```

4. Configure the administrator password for the administrative user.

```
# MySQL-Database-directory/bin/mysqladmin \
-S /tmp/Logicalhost.sock password 'admin-password'
```

5. Add the administrative user for locally accessing a MySQL instance with a `MySQL-logical-host-ip-name`.

Note - If you want to access the MySQL instance only through the socket (`localhost`), omit this step. When bootstrapping MySQL, the command `mysql_install_db` creates two administrative users, one belonging to `localhost` and one belonging to the node on which `mysql_install_db` was executed.

Add an administrative user for every global-cluster node that runs this MySQL instance.

Note - If the node name and the hostname for the physical interface are different, use the hostname for the physical interface.

The following is an example for a MySQL instance.

```
# mysql -S /tmp/clusterix2.sock -uroot -padmin-password
mysql> use mysql;
mysql> GRANT ALL ON *.* TO 'root'@'clusterix2' IDENTIFIED BY 'rootpasswd';
```

```
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='clusterix2';
mysql> exit;
```

Note - If you experience any problems here, refer to the MySQL administration documentation.

6. Copy the MySQL configuration files to your home directory.

```
# cp /opt/SUNWscmys/util/ha_mysql_config /home-dir
# cp /opt/SUNWscmys/util/mysql_config /home-dir
```

7. Create a fault-monitor user and a test database for the MySQL instance.

```
# cd home-dir
```

8. Edit the `mysql_config` file and follow the comments within that file.

```
##
CDDL HEADER START
##
#The contents of this file are subject to the terms of the
# Common Development and Distribution License (the License).
# You may not use this file except in compliance with the License.
##
#You can obtain a copy of the license at usr/src/CDDL.txt
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
##
#When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/CDDL.txt.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets [] replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
##
CDDL HEADER END
#
##
#Copyright 2012 Oracle Corporation. All rights reserved.
# Use is subject to license terms.
#
#ident "@(#)ds_template.sgm 1.41 12/10/12"
# This file will be sourced in by mysql_register and the parameters
# listed below will be used.
#
# Where is MySQL installed (BASEDIR)
MYSQL_BASE=
# MySQL admin-user for localhost (Default is root)
MYSQL_USER=
```

```
# Password for MySQL admin user
MYSQL_PASSWD=
# Configured logicalhost. For scalable or multiple-master resources leave it empty.
MYSQL_HOST=
# Specify a username for a faultmonitor user
FMUSER=
# Pick a password for that faultmonitor user
FMPASS=
# Socket name for mysqld ( Should be /tmp/logical-host.sock )
MYSQL_SOCKET=
# Specify the physical hostname for the physical NIC that this logicalhostname
# belongs to for every node in the cluster this resource group can be located on.
# If you use the mysql_geocontrol features to implement the MySQL replication as
# the replication protocol in Oracle Solaris Cluster geographic edition, specify all
# physical nodes of all clusters. Specify at least all the nodes on both sites
# where the MySQL databases can be hosted.
# IE: The logicalhost lh1 belongs to hme1 for physical-node phys-1 and
# hme3 for physical-node phys-2. The hostname for hme1 is phys-1-hme0 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="phys-1-hme0 phys-2-hme3"
# IE: If two clusters are tied together by the mysql_geocontrol features, assuming the
# MySQL database on cluster one belongs to cl1-phys1-hme0 and cl1-phys2-hme3, the
# MySQL database on cluster two belongs to cl2-phys1-hme2 and cl2-phys2-hme4. Then the
# MYSQL_NIC_HOSTNAME variable needs to be set to:
# MYSQL_NIC_HOSTNAME="cl1-phys1-hme0 cl1-phys2-hme3 cl2-phys1-hme2 cl2-phys2-
#hme4"
MYSQL_NIC_HOSTNAME=
# Where are your databases installed, (location of my.cnf)
MYSQL_DATADIR=
# Is MySQL Cluster installed?
# Any entry here triggers the ndb engine check preparation. If no MySQL cluster should
# be
# checked, leave it empty.
NDB_CHECK=
```

The following is an example for a MySQL instance.

```
MYSQL_BASE=/global/mysql
MYSQL_USER=root MYSQL_PASSWD=root
MYSQL_HOST=hahostix1
FMUSER=fmuser
FMPASS=
MYSQL_SOCKET=/tmp/hahostix1.sock
MYSQL_NIC_HOSTNAME="clusterix1 clusterix2"
MYSQL_DATADIR=/global/mysql-data
NDB_CHECK=
```

If you want to monitor the ndb tables of a MySQL cluster, set NDB_CHECK to yes.

If you are about to prepare the database servers for a scalable or multiple-master configuration, set `MYSQL_SOCKET=hostname.sock`.

9. **After editing `mysql_config`, run the `mysql_register` script.**

```
# /opt/SUNWscmys/util/mysql_register -f home-dir/mysql_config
```

10. **Stop the MySQL Server instance manually.**

```
# kill -TERM `cat MySQL-Database-directory/mysqld.pid`
```

11. **After finishing the previous steps on every node or zone in the cluster, continue in the global zone unless you configure the MySQL server resource in a zone cluster.**

12. **Create and register MySQL as a scalable or multiple-master data service.**

- a. **Navigate to the `home-dir` directory.**

```
# cd home-dir
```

- b. **Edit the `ha_mysql_config` file and follow the comments within that file.**

```
##
CDDL HEADER START
##
#The contents of this file are subject to the terms of the
# Common Development and Distribution License (the License).
# You may not use this file except in compliance with the License.
##
#You can obtain a copy of the license at usr/src/CDDL.txt
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
##
#When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/CDDL.txt.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets [] replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
##
CDDL HEADER END
#
##
#
# Copyright (c) 2006, 2012, Oracle and/or its affiliates. All rights reserved.
#
```

```
#ident "@(#)ha_mysql_config.ksh      1.9      12/08/20"
##
# These parameters can be customized in (key=value) form
##
# RS - name of the resource for the application
# RG - name of the resource group containing RS
# SCALABLE - flag to indicate a scalable resource creation.
# The default is no, so any entry here triggers a scalable resource.
# LB_POLICY - Set the loadbalancing policy for a scalable MySQL service.
# Use the values defined for the standard resource property
# Load_balancing_policy. If you do not specify it, the defaults are
# used.
# RS_PROP - Additional resource properties in the format for clresource create.
# Example: "-p start_timeout=600"
##
# To have the MySQL agent local zone aware, 4 variables are needed:
# ZONE - The zone name where the MySQL Database should run.
# Optional
# ZONEBT - The resource name which controls the zone.
# Optional
# PROJECT - A project in the zone that will be used for this service.
# Specify it if you have an su - in the start stop or probe,
# or to define the smf credentials. If the variable is not set,
# it will be translated as :default for the sm and default
# for the zsh component.
# Optional
# ZUSER - A user in the the zone which is used for the smf method
# credentials. Your smf service will run under this user.
# Optional
##
MySQL specific variables
##
# BASEDIR - name of the MySQL bin directory
# DATADIR - name of the MySQL Data directory
# MYSQLUSER - name of the user MySQL should be started with or from
# LH - name of the LogicalHostname SC resource
# MYSQLHOST - name of the host in /etc/hosts. For scalable/multiple-master
resources, leave it
# empty.
# FMUSER - name of the MySQL fault monitor user
# FMPASS - name of the MySQL fault monitor user password.
# If you do not want to store the password as readable in a file, leave the FMPASS
# variable empty.
# LOGDIR - name of the directory where mysqld should store its logfile.
# CHECK - should HA-MySQL check MyISAM index files before start YES/NO.
# HAS_RS - name of the MySQL HAStoragePlus SC resource
# NDB_CHECK - Is MySQL Cluster installed?
```

```
# Any entry here triggers the ndb engine check, if no MySQL cluster should be
  checked
# leave it empty.
##
#The following examples illustrate sample parameters
# for MySQL
##
# BASEDIR=/usr/local/mysql
# DATADIR=/global/mysqldata
# MYSQLUSER=mysql
# LH=mysqlhh
# RS_PROP=
# SCALABLE=
# LB_POLICY=
# MYSQLHOST=mysqlhh
# FMUSER=fmuser
# FMPASS=fmuser
# LOGDIR=/global/mysqldata/logs
# CHECK=YES
# NDB_CHECK=
#
RS=
RG=
PORT=
LH=
SCALABLE=
LB_POLICY=
HAS_RS=
# local zone specific options
ZONE=
ZONE_BT=
PROJECT=
# mysql specifications
BASEDIR=
DATADIR=
MYSQLUSER=
MYSQLHOST=
FMUSER=
FMPASS=
LOGDIR=
CHECK=
NDB_CHECK=
```

The following is an example for a MySQL instance.

Note - If you want to register a multiple-master resource, do not set the variables SCALABLE and MYSQLHOST.

```
SCALABLE=  
MYSQLHOST=  
RS=mysql-res  
RG=mysql-rg  
BASEDIR=/global/mysql  
DATADIR=/global/mysql-data  
MYSQLUSER=mysql  
LH=hahostix1  
SCALABLE=yes  
LB_POLICY=  
RS_PROP=  
MYSQLHOST=hahostix1  
FMUSER=fmuser  
FMPASS=  
LOGDIR=/global/mysql-data/logs  
CHECK=YES  
NDB_CHECK=  
HAS_RS=  
ZONE=  
ZONE_BT=  
PROJECT=
```

If you want to monitor the ndb tables of a MySQL cluster, set NDB_CHECK to yes.

- c. **(Optional)** If you did not specify the FMPASS variable, make the `ha_mysql_config` file available in one host or zone and call the following script in one host or zone that can host the MySQL database.

```
# /opt/SUNWscmys/util/ha_mysql_register -f home-dir/ha_mysql_config -e
```

- d. **Register the MySQL resource.**

```
# /opt/SUNWscmys/util/ha_mysql_register -f home-dir/ha_mysql_config
```

13. **Switch the resource group to a managed online state.**

```
# clresourcegroup online -M MySQL-Scalable-resource-group
```


Verifying the HA for MySQL Installation and Configuration

This section contains the procedure you need to verify that you installed and configured your data service correctly.

▼ How to Verify the HA for MySQL Installation and Configuration

1. Assume the `root` role on one of the nodes in the cluster that will host MySQL.
2. Ensure all the MySQL resources are online.

```
# cluster status
```

For each MySQL resource that is not online, use the `clresource` command as follows.

```
# clresource enable MySQL-resource
```

3. Run the `clresourcegroup` command to switch the MySQL resource group to another cluster node, such as `node2`.

```
# clresourcegroup -h node2 MySQL-failover-resource-group
```

Understanding the HA for MySQL Fault Monitor

This section describes the HA for MySQL fault monitor's probing algorithm or functionality, states the conditions, messages, and recovery actions associated with unsuccessful probing.

For conceptual information on fault monitors, see [“Data Services Fault Monitors” in Oracle Solaris Cluster 4.3 Concepts Guide](#).

Resource Properties

The HA for MySQL fault monitor uses the same resource properties as resource type `SUNW.gds`. Refer to the [SUNW.gds\(5\)](#) man page for a complete list of resource properties used.

Probing Algorithm and Functionality

The following are the MySQL probing steps:

- Sleeps for `Thorough_probe_interval`.
- Tries to connect to the MySQL instance, with `mysqladmin` command with argument `ping`, with the defined fault-monitor user `fault_fmuser`. If this fails, then the probe will restart the MySQL resource.
- Every 300 seconds, the probe also checks the following:
 - If the MySQL instance is a slave configuration, the probe checks whether the MySQL instance is connected to its master. If the Slave is not connected, the probe will write an error message to `syslog` and sets a status message.
 - Verifies that the probe can list all databases and tables, not the contents. If the probe receives any errors the probe writes an error message to `syslog`.
 - Conducts function tests on the defined test-database, Create Table, Insert into Table, Update Table, Delete from Table and Drop Table. If any of those operations fails, then the probe restarts the MySQL resource.
- If all MySQL processes have died, `pmf` interrupts the probe to immediately restart the MySQL resource.
- If the MySQL resource is repeatedly restarted and subsequently exhausts the `Retry_count` within the `Retry_interval` then a failover is initiated for the Resource Group onto another node if `Failover_enabled` is set to `TRUE`.

Debugging the HA for MySQL

HA for MySQL has a file named `config` that enables you to activate debugging for MySQL resources. This file is in the `/opt/SUNWscmys/etc` directory.

▼ How to Activate Debugging for HA for MySQL

1. Determine whether you are in a global zone or in an HA Zone configuration.
2. Determine whether debugging for HA for MySQL is active.

```
# grep daemon /etc/syslog.conf
*.err;kern.debug;daemon.notice;mail.crit      /var/adm/messages
```

```
*.alert;kern.err;daemon.err          operator
#
```

If debugging is inactive, `daemon.notice` is set in the file `/etc/syslog.conf` of the appropriate zone.

3. **If debugging is inactive, edit the `/etc/syslog.conf` file in the appropriate zone to change `daemon.notice` to `daemon.debug`.**

4. **Confirm that debugging for HA for MySQL is active.**

If debugging is active, `daemon.debug` is set in the file `/etc/syslog.conf`.

```
# grep daemon /etc/syslog.conf
*.err;kern.debug;daemon.debug;mail.crit      /var/adm/messages
*.alert;kern.err;daemon.err                  operator
#
```

5. **Restart the `syslogd` daemon in the appropriate zone.**

```
# svcadm restart system-log
```

6. **Edit the appropriate `/opt/SUNWscmys/etc/config` file to change the `DEBUG=` parameter.**

The value of `component-name` can be `ndb_mgmd`, `ndbd`, `ndbd_shutdown`.

- To debug all resources, use `DEBUG=ALL`.
- To debug specific resources, use `DEBUG=resource name , resource name [, ...]`.

Note - To deactivate debugging, repeat this procedure after you change `daemon.debug` to `daemon.notice` and the `DEBUG` variable to `DEBUG=`.

Example 5 Editing the Debug Config File

```
# cat /opt/SUNWscmys/etc/config
#
# Copyright 2012 OracleCorporation. All rights reserved.
# Use is subject to license terms.
#
# Usage:
#     DEBUG=<RESOURCE_NAME> or ALL
#
DEBUG=ALL
#
```


◆◆◆ APPENDIX A

Deployment Example: Installing MySQL in the Global Zone

This appendix presents a complete example of how to install and configure the MySQL application and data service in the global zone. It presents a simple two-node cluster configuration. If you need to install the application in any other configuration, refer to the general purpose procedures presented elsewhere in this manual. For an example of MySQL installation in a non-global zone, see [Appendix B, “Deployment Example: Installing MySQL in the Non-Global HA Zone”](#) or [Appendix C, “Deployment Example: Installing MySQL in a Scalable or Multiple-Master Configuration”](#) according to your zone type.

Target Cluster Configuration

This example uses a two-node cluster with the following node names:

- `phys-schost-1` (a physical node, which owns the file system)
- `phys-schost-2` (a physical node)

This configuration also uses the logical host name `ha-host-1`.

Software Configuration

This deployment example uses the following software products and versions:

- Oracle Solaris 11 software for SPARC or x86 platforms
- Oracle Solaris Cluster 4.3 core software
- Oracle HA for MySQL
- MySQL - Max version 5.0.22

This example assumes that you have already installed and established your cluster. It illustrates installation and configuration of the data service application only.

Assumptions

The instructions in this example were developed with the following assumptions:

- **Shell environment:** All commands and the environment setup in this example are for the bash shell environment. If you use a different shell, replace any bash shell-specific information or instructions with the appropriate information for your preferred shell environment.
- **User login:** Unless otherwise specified, perform all procedures by assuming a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` RBAC authorization.

Installing and Configuring MySQL on Local Storage in the Global Zone

The tasks you must perform to install and configure MySQL in the global zone are as follows:

- [“Example: Preparing the Cluster for MySQL” on page 78](#)
- [“Example: Configuring Cluster Resources for MySQL” on page 79](#)
- [“Example: Installing and Bootstrapping the MySQL Software on Local Storage” on page 79](#)
- [“Example: Modifying the MySQL Configuration Files” on page 81](#)
- [“Example: Enabling the MySQL Software to Run in the Cluster” on page 82](#)

▼ Example: Preparing the Cluster for MySQL

1. **Install and configure the cluster as instructed in the [Oracle Solaris Cluster 4.3 Software Installation Guide](#).**

Install the following cluster software components on both nodes.

- Oracle Solaris Cluster core software
- Oracle Solaris Cluster data service for MySQL

2. **Beginning on the node that owns the file system, add the `mysql` user.**

```

phys-schost-1# groupadd -g 1000 mysql
phys-schost-1# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
phys-schost-2# groupadd -g 1000 mysql
phys-schost-2# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql

```

▼ Example: Configuring Cluster Resources for MySQL

1. Register the necessary data types on one of the nodes.

```
phys-schost-1# clresourcetype register SUNW.gds SUNW.HAStoragePlus
```

2. Create the MySQL resource group.

```
phys-schost-1# clresourcegroup create RG-MYS
```

3. Create the logical host.

```
phys-schost-1# clreslogicalhostname create -g RG-MYS ha-host-1
```

4. Create the HAStoragePlus resource in the RG-MYS resource group.

```
phys-schost-1# clresource create -g RG-MYS -t SUNW.HAStoragePlus -p AffinityOn=TRUE \
-p FilesystemMountPoints=/global/mnt3,/global/mnt4 RS-MYS-HAS
```

5. Enable the resource group.

```
phys-schost-1# clresourcegroup online -M RG-MYS
```

▼ Example: Installing and Bootstrapping the MySQL Software on Local Storage

These steps illustrate how to install the MySQL software in the default directory `/usr/local/mysql`. If only one node is mentioned, it needs to be the node where your resource group is online.

1. Install the MySQL binaries on both nodes.

```

phys-schost-1# cd /usr/local
phys-schost-1# tar xvf mysql-max-5.0.22-solaris10-architecture_64.tar.gz

```

```
phys-schost-1# ln -s mysql-max-5.0.22-solaris10-architecture_64 mysql
```

2. Change the ownership of the MySQL binaries on both nodes.

```
phys-schost-1# chown -R mysql:mysql /usr/local/mysql
```

3. Create your database directories.

```
phys-schost-1# mkdir -p /global/mnt3/mysql-data/logs
phys-schost-1# mkdir /global/mnt3/mysql-data/innodb
phys-schost-1# mkdir /global/mnt3/mysql-data/BDB
```

4. Bootstrap MySQL.

```
phys-schost-1# cd MySQL-Base-directory
# ./bin/mysqld --initialize --datadir=MySQL-Database-directory
```

5. Create your my.cnf config-file in /global/mnt3/mysql-data.

```
phys-schost-1# cat > /global/mnt3/mysql-data/my.cnf << EOF
[mysqld]
server-id=1
#port=3306
# 10.18.5.1 is the address of the logical host
bind-address=10.18.5.1 # this is the address of the logical host
socket=/tmp/hahostix1.sock
log=/global/mnt3/mysql-data/logs/log1
log-bin=/global/mnt3/mysql-data/logs/bin-log
binlog-ignore-db=sc3_test_database
log-slow-queries=/global/mnt3/mysql-data/logs/log-slow-queries
#log-update=/global/mnt3/mysql-data/logs/log-update

# Innodb
#skip-innodb
innodb_data_home_dir = /global/mnt3/mysql-data/innodb
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = /global/mnt3/mysql-data/innodb
innodb_log_arch_dir = /global/mnt3/mysql-data/innodb
# You can set ..buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
set-variable = innodb_buffer_pool_size=50M
set-variable = innodb_additional_mem_pool_size=20M
# Set ..log_file_size to 25 % of buffer pool size
set-variable = innodb_log_file_size=12M
set-variable = innodb_log_buffer_size=4M
innodb_flush_log_at_trx_commit=1
set-variable = innodb_lock_wait_timeout=50
```



```

# BDB
# uncomment the skip-bdb if you used a binary download.
# binary downloads come very often without the bdb support.
#skip-bdb
bdb-home=/global/mnt3/mysql-data
bdb-no-recover
bdb-lock-detect=DEFAULT
bdb-logdir=/global/mnt3/mysql-data/BDB
bdb-tmpdir=/global/mnt3/mysql-data/BDB
#bdb_max_lock=10000

# Replicating Slave
#server-id=2
#master-host=administerix
#master-user=repl
#master-password=repl
#master-info-file=/global/mnt3/mysql-data/logs/master.info

# MySQL 4.x
#relay-log=/global/mnt3/mysql-data/logs/slave-bin.log
#relay-log-info-file=/global/mnt3/mysql-data/logs/slave-info

```

6. Change the ownership of the MySQL data directory.

```
phys-schost-1# chown -R mysql:mysql /global/mnt3/mysql-data
```

7. Change the permission of the my.cnf file.

```
phys-schost-1# chmod 644 /global/mnt3/mysql-data/my.cnf
```

▼ Example: Modifying the MySQL Configuration Files

1. Copy the MySQL configuration file from the agent directory to its deployment location.

```
phys-schost-1# cp /opt/SUNWscmys/util/mysql_config /global/mnt3
phys-schost-1# cp /opt/SUNWscmys/util/ha_mysql_config /global/mnt3
```

2. Add this cluster's information to the mysql_config configuration file.

The following listing shows the relevant file entries and the values to assign to each entry.

```

...
MYSQL_BASE=/usr/local/mysql

```

```
MYSQL_USER=root
MYSQL_PASSWD=
MYSQL_HOST=ha-host-1
FMUSER=fmuser
FMPASS=
MYSQL_SOCKET=/tmp/ha-host-1.sock
MYSQL_NIC_HOSTNAME="phys-schost-1 phys-schost-2"
MYSQL_DATADIR=/global/mnt3/mysql-data
NDB_CHECK=
```

3. **Add this cluster's information to the `ha_mysql_config` configuration file.**

The following listing shows the relevant file entries and the values to assign to each entry.

```
...
RS=RS-PGS
RG=RG-MYS
PORT=5432
LH=hahostix1
HAS_RS=RS-MYS-HAS
...
BASEDIR=/usr/local/mysql
DATADIR=/global/mnt3/mysql-data
MYSQLUSER=mysql
MYSQLHOST=ha-host-1
FMUSER=fmuser
FMPASS=
LOGDIR=/global/mnt3/mys-data/logs
CHECK=YES
LB_POLICY=
SCALABLE=
```

4. **Save and close the file.**

▼ Example: Enabling the MySQL Software to Run in the Cluster

1. **Start the MySQL database manually on the node where the resource group is online.**

```
phys-schost-1# cd /usr/local/mysql
phys-schost-1# ./bin/mysqld --defaults-file=/global/mnt3/mysql-data/my.cnf \
--basedir=/usr/local/mysql --datadir=/global/mnt3/mysql-data \
--pid-file=/global/mnt3/mysql-data/mysqld.pid \
--user=mysql >> /global/mnt3/mysql-data/logs/ha-host-1.log 2>&1 &
```

2. Set the password for localhost in MySQL to root.

```
phys-schost-1# /usr/local/mysql/bin/mysqladmin -S /tmp/ha-host-1.sock -uroot \
password 'root'
```

3. Add an administrative user in the MySQL database for the logical host.

```
phys-schost-1# /usr/local/mysql/bin/mysql -S /tmp/ha-host-1.sock -uroot -proot
mysql> use mysql;
mysql> GRANT ALL ON *.* TO 'root'@'phys-schost-1' IDENTIFIED BY 'root';
mysql> GRANT ALL ON *.* TO 'root'@'phys-schost-2' IDENTIFIED BY 'root';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='phys-schost-1';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='phys-schost-2';
mysql> exit
```

4. Prepare the Oracle Solaris Cluster specific test database.

```
phys-schost-1# ksh /opt/SUNWscmys/util/mysql_register \
-f /global/mnt3/mysql_config
```

a. Enter root when you are asked for the MySQL admin password.

You set it to root in [Step 3](#).

b. Enter a password when you are prompted for the fault monitor user password.

5. Stop the MySQL database.

```
phys-schost-1# kill -TERM `cat /global/mnt3/mysql-data/mysqld.pid`
```

6. Make the /global/mnt3/ha_mysql_config file available on all nodes to run the MySQL database.

7. Encrypt the fault monitor user password on one node to run the MySQL database.

```
phys-schost-1# ksh /opt/SUNWscmys/util/ha_mysql_register \
-f /global/mnt3/ha_mysql_config -e
```

Enter the same password you provided in [Step 4](#) when you are prompted for the fault monitoring user password.

8. Run the ha_mysql_register script to register the resource.

```
phys-schost-1# ksh /opt/SUNWscmys/util/ha_mysql_register \
-f /global/mnt3/ha_mysql_config
```

9. Enable the resource.

```
phys-schost-1# clresource enable RS-MYS
```

Deployment Example: Installing MySQL in the Non-Global HA Zone

This appendix presents a complete example of how to install and configure the MySQL application and data service in a non-global high-availability zone. It presents a simple two-node cluster configuration. If you need to install the application in any other configuration, refer to the general-purpose procedures presented elsewhere in this manual. For an example of MySQL installation in a global zone, see [Appendix A, “Deployment Example: Installing MySQL in the Global Zone”](#).

Target Cluster Configuration

This example uses a two-node cluster with the following node names:

- `phys-schost-1` (a physical node, which owns the file system)
- `phys-schost-2` (a physical node)
- `clu1` the zone to be failed over

This configuration also uses the logical host name `ha-host-1`.

Software Configuration

This deployment example uses the following software products and versions:

- Oracle Solaris 11 software for SPARC or x86 platforms
- Oracle Solaris Cluster 4.3 core software
- Oracle HA for MySQL
- Oracle Solaris Cluster HA for Solaris Zones
- MySQL - Max version 5.0.22

This example assumes that you have already installed and established your cluster. It illustrates installation and configuration of the data service application only.

Assumptions

The instructions in this example were developed with the following assumptions:

- **Shell environment:** All commands and the environment setup in this example are for the bash shell environment. If you use a different shell, replace any bash shell-specific information or instructions with the appropriate information for your preferred shell environment.
- **User login:** Unless otherwise specified, perform all procedures by assuming a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` RBAC authorization.

Installing and Configuring MySQL on Local Storage in the HA Zone

The tasks you must perform to install and configure MySQL in the HA Zone are as follows:

- [“Example: Preparing the Cluster for MySQL” on page 86](#)
- [“Example: Configuring Cluster Resources for MySQL” on page 87](#)
- [“Example: Configuring the HA Container” on page 87](#)
- [“Example: Installing and Bootstrapping the MySQL Software on Local Storage” on page 89](#)
- [“Example: Modifying the MySQL Configuration Files” on page 91](#)
- [“Example: Enabling the MySQL Software to Run in the Cluster” on page 92](#)

▼ Example: Preparing the Cluster for MySQL

1. **Install and configure the cluster as instructed in the [Oracle Solaris Cluster 4.3 Software Installation Guide](#).**

Install the following cluster software components on both nodes.

- Oracle Solaris Cluster core software
- Oracle Solaris Cluster data service for MySQL
- Oracle Solaris Cluster data service for Solaris zones

2. **Beginning on the node that owns the file system, add the `mysql` user.**

```
phys-schost-1# groupadd -g 1000 mysql
phys-schost-1# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
phys-schost-2# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
```

▼ Example: Configuring Cluster Resources for MySQL

1. **Register the necessary data types on one of the nodes.**

```
phys-schost-1# clresourcetype register SUNW.gds SUNW.HASStoragePlus
```

2. **Create the MySQL resource group.**

```
phys-schost-1# clresourcegroup create RG-MYS
```

3. **Create the HASStoragePlus resource in the RG-MYS resource group.**

```
phys-schost-1# clresource create -g RG-MYS -t SUNW.HASStoragePlus -p AffinityOn=TRUE \
-p FilesystemMountPoints=/global/mnt3,/global/mnt4 RS-MYS-HAS
```

4. **Enable the resource group.**

```
phys-schost-1# clresourcegroup online -M RG-MYS
```

▼ Example: Configuring the HA Container

1. **On shared cluster storage, create a directory for the HA container root path.**

This example presents a sparse root zone. You can use a whole root zone if that type better suits your configuration.

```
phys-schost-1# mkdir /global/mnt3/zones
```

2. **Create a temporary file, for example `/tmp/x`, and include the following entries:**

```
create -b
set zonepath=/global/mnt3/zones/clu1
set autoboot=false
set pool=pool_default
add inherit-pkg-dir
set dir=/lib
end
```

```
add inherit-pkg-dir
set dir=/platform
end
add inherit-pkg-dir
set dir=/sbin
end
add inherit-pkg-dir
set dir=/usr
end
add net
set address=ha-host-1
set physical=hme0
end
add attr
set name=comment
set type=string
set value="MySQL cluster zone"      Put your desired zone name between the quotes here.
end
```

3. Configure the HA container, using the file you created.

```
phys-schost-1# zonecfg -z clu1 -f /tmp/x
```

4. Install the zone.

```
phys-schost-1# zoneadm -z clu1 install
```

5. Log in to the zone.

```
phys-schost-1# zlogin -C clu1
```

6. Open a new window to the same node and boot the zone.

```
phys-schost-1a# zoneadm -z clu1 boot
```

7. Close this terminal window and disconnect from the zone console.

```
phys-schost-1# ~~.
```

8. Copy the container's configuration file to a temporary location.

```
phys-schost-1# cp /opt/SUNWsczone/sczbt/util/sczbt_config /tmp/sczbt_config
```

9. Edit the /tmp/sczbt_config file and set variable values as shown:

```
RS=RS-MYS-ZONE
RG=RG-MYS
PARAMETERDIR=/global/mnt3/zonepar
SC_NETWORK=false
```



```
SC_LH=  
FAILOVER=true  
HAS_RS=RS-MYS-HAS
```

```
Zonename=clu1  
Zonebootopt=  
Milestone=multi-user-server  
Mounts=
```

Note - For more information about creating a zone, see the [Oracle Solaris Cluster Data Service for Oracle Solaris Zones Guide](#).

10. Register the zone resource.

```
phys-schost-1# ksh /opt/SUNWsczone/sczbt/util/sczbt_register -f /tmp/sczbt_config
```

11. Enable the zone resource.

```
phys-schost-1# clresource enable RS-MYS-ZONE
```

▼ Example: Installing and Bootstrapping the MySQL Software on Local Storage

These steps illustrate how to install the MySQL software in the default directory `/usr/local/MySQL`. If only one node is mentioned, it needs to be the node where your resource group is online.

Note - This procedure assumes that you inherited `/usr`, so you cannot write to `/usr/local` in the zones. If you linked `/usr/local` to a local directory, start at step 4.

1. Add the mysql user.

```
phys-schost-1# groupadd -g 1000 mysql  
phys-schost-1# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
```

2. Install the MySQL binaries on both nodes.

```
phys-schost-1# cd /usr/local  
phys-schost-1# tar xvf mysql-max-5.0.22-solaris10-architecture_64.tar.gz  
phys-schost-1# ln -s mysql-max-5.0.22-solaris10-x86_64 mysql
```

3. Change the ownership of the MySQL binaries on both nodes.

```
phys-schost-1# chown -R mysql:mysql /usr/local/mysql
```

4. Log in to the zone.

```
phys-schost-1# zlogin clu1
```

5. Create the parent for the mysql home directory.

```
zone# mkdir -p /global/mnt3
```

6. Add the mysql user.

```
zone# groupadd -g 1000 mysql
zone# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
```

7. Create your database directories.

```
zone# mkdir -p /global/mnt3/mysql-data/logs
zone# mkdir /global/mnt3/mysql-data/innodb
zone# mkdir /global/mnt3/mysql-data/BDB
```

8. Bootstrap MySQL.

```
zone# cd MySQL-Base-directory
# ./bin/mysqld --initialize --datadir=MySQL-Database-directory
```

9. Create your my.cnf configuration file in /global/mnt3/mysql-data.

```
zone# cat > /global/mnt3/mysql-data/my.cnf << EOF
[mysqld]
server-id=1
#port=3306
# 10.18.5.1 is the address of the logical host
bind-address=10.18.5.1 # this is the address of the logical host
socket=/tmp/ha-host-1.sock
log=/global/mnt3/mysql-data/logs/log1
log-bin=/global/mnt3/mysql-data/logs/bin-log
binlog-ignore-db=sc3_test_database
log-slow-queries=/global/mnt3/mysql-data/logs/log-slow-queries
#log-update=/global/mnt3/mysql-data/logs/log-update

# Innodb
#skip-innodb
innodb_data_home_dir = /global/mnt3/mysql-data/innodb
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = /global/mnt3/mysql-data/innodb
innodb_log_arch_dir = /global/mnt3/mysql-data/innodb
# You can set .._buffer_pool_size up to 50 - 80 %
```

```

# of RAM but beware of setting memory usage too high
set-variable = innodb_buffer_pool_size=50M
set-variable = innodb_additional_mem_pool_size=20M
# Set ..log_file_size to 25 % of buffer pool size
set-variable = innodb_log_file_size=12M
set-variable = innodb_log_buffer_size=4M
innodb_flush_log_at_trx_commit=1
set-variable = innodb_lock_wait_timeout=50

# BDB
# uncomment the skip-bdb if you used a binary download.
# binary downloads come very often without the bdb support.
#skip-bdb
bdb-home=/global/mnt3/mysql-data
bdb-no-recover
bdb-lock-detect=DEFAULT
bdb-logdir=/global/mnt3/mysql-data/BDB
bdb-tmpdir=/global/mnt3/mysql-data/BDB
#bdb_max_lock=10000

# Replication of Slave
#server-id=2
#master-host=administerix
#master-user=repl
#master-password=repl
#master-info-file=/global/mnt3/mysql-data/logs/master.info

# MySQL 4.x
#relay-log=/global/mnt3/mysql-data/logs/slave-bin.log
#relay-log-info-file=/global/mnt3/mysql-data/logs/slave-info

```

10. Change the ownership of the MySQL data directory.

```
zone# chown -R mysql:mysql /global/mnt3/mysql-data
```

11. Change the permission of the my.cnf file.

```
zone# chmod 644 /global/mnt3/mysql-data/my.cnf
```

▼ Example: Modifying the MySQL Configuration Files

1. Copy the MySQL database configuration file from the agent directory to its deployment location.

```
phys-schost-1# cp /opt/SUNWscmys/util/mysql_config /config-files
```

2. Add this cluster's information to the `mysql_config` configuration file.

The following listing shows the relevant file entries and the values to assign to each entry.

```
..MYSQL_BASE=/usr/local/mysql
MYSQL_USER=root
MYSQL_PASSWD=
MYSQL_HOST=ha-host-1
FMUSER=fmuser
FMPASS=
MYSQL_SOCKET=/tmp/ha-host-1.sock
MYSQL_NIC_HOSTNAME="ha-host-1"
MYSQL_DATADIR=/global/mnt3/mys-data
NDB_CHECK=
```

3. Save and close the file.

▼ Example: Enabling the MySQL Software to Run in the Cluster

1. Start the MySQL database manually on the node where the resource group is online.

```
zone# cd /usr/local/mysql
zone# ./bin/mysqld --defaults-file=/global/mnt3/mysql-data/my.cnf \
--basedir=/usr/local/mysql --datadir=/global/mnt3/mysql-data \
--pid-file=/global/mnt3/mysql-data/mysqld.pid \
--user=mysql >> /global/mnt3/mysql-data/logs/ha-host-1.log 2>&1 &
```

2. Set the password for `localhost` in MySQL to `root`.

```
zone# /usr/local/mysql/bin/mysqladmin -S /tmp/ha-host-1.sock -uroot \
password 'root'
```

3. Add an administrative user in the MySQL database for the logical host.

```
zone# /usr/local/mysql/bin/mysql -S /tmp/ha-host-1.sock -uroot -proot
mysql> use mysql;
mysql> GRANT ALL ON *.* TO 'root'@'ha-host-1' IDENTIFIED BY 'root';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='ha-host-1';
mysql> exit
```

4. Prepare the Oracle Solaris Cluster specific test database.

```
zone# ksh /opt/SUNWscmys/util/mysql_register -f /config-files/mysql_config
```

- a. Enter root after you are prompted for the MySQL administrative password.

You set this to root in [Step 3](#).

- b. Enter a password when you are prompted for the fault monitor user password.

5. Stop the MySQL database.

```
zone# kill -TERM `cat /global/mnt3/mysql-data/mysqld.pid`
```

6. Leave the zone.

7. Copy the MySQL database configuration file from the agent directory to its deployment location.

```
phys-schost-1# cp /opt/SUNWscmys/util/ha_mysql_config /global/mnt3
```

8. Add this cluster's information to the ha_mysql_config configuration file.

The following listing shows the relevant file entries and the values to assign to each entry.

```
...
RS=RS-MYS
RG=RG-MYS
PORT=5432
LH=ha-host-1
LB_POLICY=
SCALABLE=
HAS_RS=RS-MYS-HAS
...
ZONE=clu1
ZONE_BT=RS-MYS-ZONE
PROJECT=
...
BASEDIR=/usr/local/mysql
DATADIR=/global/mnt3/mysql-data
MYSQLUSER=mysql
MYSQLHOST=ha-host-1
FMUSER=fmuser
FMPASS=
LOGDIR=/global/mnt3/mys-data/logs
CHECK=YES
NDB_CHECK=
```

9. **Save and close the file.**
10. **Make the `/global/mnt3/ha_mysql_config` file available on the zone to run the MySQL database.**
11. **Enter the zone.**
12. **Encrypt the fault monitor user password on one node to run the MySQL database.**

```
zone# ksh /opt/SUNWscmys/util/ha_mysql_register \  
-f /global/mnt3/ha_mysql_config -e
```

Enter the same password you provided in [Step 4](#) when you are prompted for the fault monitoring user password.

13. **Leave the zone.**
14. **Run the `ha_mysql_register` script to register the resource.**

```
phys-schost-1# ksh /opt/SUNWscmys/util/ha_mysql_register \  
-f /global/mnt3/ha_mysql_config
```

15. **Enable the resource.**

```
phys-schost-1# clresource enable RS-MYS
```

Deployment Example: Installing MySQL in a Scalable or Multiple-Master Configuration

This appendix presents a complete example of how to install and configure the MySQL application and data service in a scalable or multiple-master configuration in the global zone. It presents a simple two-node cluster configuration. If you need to install the application in any other configuration, refer to the general-purpose procedures presented elsewhere in this manual.

For an example of MySQL installation in a non-global zone, see [Appendix A, “Deployment Example: Installing MySQL in the Global Zone”](#) or [Appendix B, “Deployment Example: Installing MySQL in the Non-Global HA Zone”](#), according to your zone type.

Target Cluster Configuration

This example uses a two-node cluster with the following node names:

- phys-schost-1 (a physical node)
- phys-schost-2 (a physical node)

The scalable configuration also uses the shared address host name ha-host-1.

Software Configuration

This deployment example uses the following software products and versions:

- Oracle Solaris 11 software for SPARC or x86 platforms
- Oracle Solaris Cluster 4.3 core software
- HA for MySQL
- MySQL - Max version 5.0.22

This example assumes that you have already installed and established your cluster. It illustrates installation and configuration of the data service application only.

Assumptions

The instructions in this example make the following assumptions:

- Shell environment: All commands and the environment setup in this example are for the bash shell environment. If you use a different shell, replace any bash shell-specific information or instructions with the appropriate information for your preferred shell environment.
- User login: Unless otherwise specified, perform all procedures by assuming a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` RBAC authorization.

Installing and Configuring MySQL on Local Storage in the Global Zone

These instructions assume that you are installing the MySQL software as the `mysql` user on the local storage in the global zone.

The tasks you must perform to install and configure MySQL in the zone are as follows:

- [“Example: How to Prepare the Cluster for MySQL” on page 96](#)
- [“Example: How to Configure MySQL in a Scalable Resource Group” on page 97](#)
- [“Example: How to Configure MySQL in a Multiple–Master Resource Group” on page 98](#)
- [“Example: How to Install and Bootstrap the MySQL Software on Local Storage” on page 98](#)
- [“Example: How to Modify the MySQL Configuration Files” on page 100](#)
- [“Example: How to Enable the MySQL Software to Run in the Cluster” on page 101](#)

▼ Example: How to Prepare the Cluster for MySQL

Before You Begin Install and configure the cluster as instructed in the [Oracle Solaris Cluster 4.3 Software Installation Guide](#).

1. Install the following cluster software components on both nodes.

- Oracle Solaris Cluster core software
- Oracle Solaris Cluster data service for MySQL

2. Add the `mysql` group and user.

```
phys-schost-1# groupadd -g 1000 mysql
phys-schost-1# useradd -g 1000 -d /local/mysql -m -s /bin/ksh mysql
phys-schost-2# groupadd -g 1000 mysql
phys-schost-2# useradd -g 1000 -d /local/mysql -m -s /bin/ksh mysql
```

▼ Example: How to Configure MySQL in a Scalable Resource Group

This procedure describes how to configure MySQL in a scalable resource group. If you need to configure MySQL in a multiple-master group, see [“Example: How to Configure MySQL in a Multiple–Master Resource Group”](#) on page 98.

1. Register the necessary data types on one of the nodes.

```
phys-schost-1# clresourcetype register SUNW.gds
```

2. Create the failover resource group for the shared address resource.

```
phys-schost-1# clresourcegroup create RG-ACCESS
```

3. Create the shared address resource.

```
phys-schost-1# clressharedaddress create ha-host-1
```

4. Create the MySQL resource group.

```
phys-schost-1# clresourcegroup create RG-MYSQL
```

5. Enable the resource groups.

```
phys-schost-1# clresourcegroup online -M RG-MYSQL
phys-schost-1# clresourcegroup online -M RG-ACCESS
```

▼ Example: How to Configure MySQL in a Multiple-Master Resource Group

This procedure describes how to configure MySQL in a multiple-master resource group. If you need to configure MySQL in a scalable group, see [“Example: How to Configure MySQL in a Scalable Resource Group” on page 97](#).

1. **Register the necessary data types on one of the nodes.**

```
phys-schost-1# clresourcetype register SUNW.gds
```

2. **Create the MySQL resource group.**

```
phys-schost-1# clresourcegroup create -p Maximum primaries=2 \  
-p desired primaries=2 RG-MYS
```

3. **Enable the resource groups.**

```
phys-schost-1# clresourcegroup online -M RG-MYS
```

▼ Example: How to Install and Bootstrap the MySQL Software on Local Storage

These steps illustrate how to install the MySQL software in the default directory `/usr/local/mysql`.

1. **Install the MySQL binaries on both nodes.**

```
phys-schost-1# cd /usr/local  
phys-schost-1# tar xvf mysql-max-5.0.22-solaris10-architecture_64.tar.gz  
phys-schost-1# ln -s mysql-max-5.0.22-solaris10-architecture_64 mysql
```

2. **Change the ownership of the MySQL binaries on both nodes.**

```
phys-schost-1# chown -R mysql:mysql /usr/local/mysql
```

3. **Create your database directories.**

```
phys-schost-1# mkdir -p /local/mysql-data/logs  
phys-schost-1# mkdir /local/mysql-data/innodb  
phys-schost-1# mkdir /local/mysql-data/BDB
```

4. Bootstrap MySQL on both nodes.

```
phys-schost-1# cd MySQL-Base-directory
# ./bin/mysqld --initialize --datadir=MySQL-Database-directory
```

5. Create your my.cnf config-file in /local/mysql-data.

```
phys-schost-1# cat > /local/mysql-data/my.cnf << EOF
[mysqld]
server-id=1
#port=3306
# 10.18.5.1 is the address of the logical host
#bind-address=10.18.5.1 #
socket=/tmp/phys-schost-1.sock
log=/local/mysql-data/logs/log1
log-bin=/local/mysql-data/logs/bin-log
binlog-ignore-db=sc3_test_database
log-slow-queries=/local/mysql-data/logs/log-slow-queries
#log-update=/local/mysql-data/logs/log-update
# InnoDB
#skip-innodb
innodb_data_home_dir = /local/mysql-data/innodb
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = /local/mysql-data/innodb
innodb_log_arch_dir = /local/mysql-data/innodb
# You can set ..buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
set-variable = innodb_buffer_pool_size=50M
set-variable = innodb_additional_mem_pool_size=20M
# Set ..log_file_size to 25 % of buffer pool size
set-variable = innodb_log_file_size=12M
set-variable = innodb_log_buffer_size=4M
innodb_flush_log_at_trx_commit=1
set-variable = innodb_lock_wait_timeout=50
# BDB
# uncomment the skip-bdb if you used a binary download.
# binary downloads come very often without the bdb support.
#skip-bdb
bdb-home=/local/mysql-data
bdb-no-recover
bdb-lock-detect=DEFAULT
bdb-logdir=/local/mysql-data/BDB
bdb-tmpdir=/local/mysql-data/BDB
#bdb_max_lock=10000
# Replication of Slave
#server-id=2
#master-host=administerix
#master-user=repl
```

```
#master-password=repl
#master-info-file=/local/mysql-data/logs/master.info
# MySQL 4.x
#relay-log=/local/mysql-data/logs/slave-bin.log
#relay-log-info-file=/local/mysql-data/logs/slave-info
```

Note - If you are running multiple slaves, please ensure that the parameters `server-id` and `socket` differ for all the MySQL resources.

6. Change the ownership of the MySQL data directory.

```
phys-schost-1# chown -R mysql:mysql /local/mysql-data
```

7. Change the permission of the `my.cnf` file.

```
phys-schost-1# chmod 644 /local/mysql-data/my.cnf
```

▼ Example: How to Modify the MySQL Configuration Files

1. Copy the MySQL configuration file from the agent directory to its deployment location on both nodes.

```
phys-schost-1# cp /opt/SUNWscmys/util/mysql_config /local
phys-schost-1# cp /opt/SUNWscmys/util/ha_mysql_config /local
```

2. Add this cluster's information to the `mysql_config` configuration file on both nodes.

The following listing shows the relevant file entries for a scalable service and the values to assign to each entry.

```
...
MYSQL_BASE=/usr/local/mysql
MYSQL_USER=root
MYSQL_PASSWD=
MYSQL_HOST= phys-schost-1
FMUSER=fmuser
FMPASS=
MYSQL_SOCKET=/tmp/phys-schost-1.sock
MYSQL_NIC_HOSTNAME="phys-schost-1"
MYSQL_DATADIR=/local/mysql-data
NDB_CHECK=
```

Note - Make sure that the parameter `MYSQL_HOST` reflects the physical hostname on both nodes.

3. Add this cluster's information to the `ha_mysql_config` configuration file.

The following listing shows the relevant file entries and the values to assign to each entry.

Note - For a multiple-master resource leave the variables `SCALABLE` and `LH` empty.

```
...
RS=RS-MYSQL
RG=RG-MYSQL
PORT=3306
LH=ha-host-1
SCALABLE=yes
LB_POLICY=
HAS_RS=
...
BASEDIR=/usr/local/mysql
DATADIR=/local/mysql-data
MYSQLUSER=mysql
MYSQLHOST=
FMUSER=fmuser
FMPASS=
LOGDIR=/local/mys-data/logs
CHECK=YES
NDB_CHECK=
```

4. Save and close the file.

▼ Example: How to Enable the MySQL Software to Run in the Cluster

1. Start the MySQL database manually on both nodes.

```
phys-schost-1# cd /usr/local/mysql
phys-schost-1# ./bin/mysqld --defaults-file=/local/mysql-data/my.cnf \
--basedir=/usr/local/mysql --datadir=/local/mysql-data \
--pid-file=/local/mysql-data/mysqld.pid \
--user=mysql >> /local/mysql-data/logs/phys-schost-1.log 2>&1
```

Note - Make sure to change `phys-schost-1` to `phys-schost-2` on the second node.

2. Set the password for localhost in MySQL to root on both nodes.

```
phys-schost-1# /usr/local/mysql/bin/mysqladmin -S /tmp/phys-schost-1.sock -uroot \  
password 'root'
```

Note - Make sure to change phys-schost-1 to phys-schost-2 on the second node.

3. Add an administrative user in the MySQL database for the physical host on both nodes.

```
phys-schost-1# /usr/local/mysql/bin/mysql -S /tmp/phys-schost-1.sock -uroot -proot  
mysql> use mysql;  
mysql> GRANT ALL ON *.* TO 'root'@'phys-schost-1' IDENTIFIED BY 'root';  
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='phys-schost-1';  
mysql> exit
```

Note - Make sure to change phys-schost-1 to phys-schost-2 on the second node.

4. Prepare the Oracle Solaris Cluster specific test database on both nodes.

```
phys-schost-1# ksh /opt/SUNWscmys/util/mysql_register -f /local/mysql_config
```

5. Stop the MySQL database.

```
phys-schost-1# kill -TERM `cat /local/mysql-data/mysqld.pid`
```

6. Make the /global/mnt3/ha_mysql_config file available on all nodes to run the MySQL database.

7. Encrypt the fault monitor user password on one node to run the MySQL database.

```
phys-schost-1# ksh /opt/SUNWscmys/util/ha_mysql_register \  
-f /global/mnt3/ha_mysql_config -e
```

Enter the same password you provided in [Step 4](#) when you are prompted for the fault monitoring user password.

8. Run the ha_mysql_register script to register the resource on one node.

```
phys-schost-1# ksh /opt/SUNWscmys/util/ha_mysql_register \  
-f /local/ha_mysql_config
```

9. Enable the resource.

```
phys-schost-1# clresource enable RS-MYSQL
```

Index

A

administrative user, 59
 setting the password, 47

B

Bootstrap, 38

C

commands
 chmod, 39, 43
 chown, 32, 37, 39, 43
 clresourcegroup, 40
 clresourcetype, 30
 groupadd, 32
 useradd, 32
configuration
 multi-master, 36
 MySQL, 28
 registering Oracle Solaris Cluster HA for
 MySQL, 45
 requirements, 21
 restrictions, 17
 scalable, 35
 verify, 43

D

debugging, 74
deployment examples
 MySQL in a non-global zone, 85

MySQL in a scalable or multiple-master
configuration, 95
MySQL in the global zone, 77

E

enabling resource group, 40
examples
 MySQL in a non-global zone, 85
 MySQL in a scalable or multiple-master
 configuration, 95
 MySQL in the global zone, 77

F

failover resource, 34
fault monitor, 73
 probing algorithm and functionality, 74
 resource properties, 73
 setting the password, 22

H

HA for MySQL
 installing, 44
 software package, installing, 44

I

installing, 28
 HA for MySQL, 44

M

multi-master configuration, 36
MySQL software overview, 16

O

Oracle Solaris Cluster software
publisher, 45, 45

P

package, 44
password
for the fault monitor, 22
publisher
Oracle Solaris Cluster software, 45, 45

R

resource group
enabling, 40
resource types
SUNW.gds, 30, 34

S

scalable configuration, 35
software package, 44