Oracle® Solaris Cluster Data Service for MySQL Guide

Part No: 821–1528–10
September 2010, Revision A

110714@25097

Contents

Preface

Oracle Solaris Cluster Data Service for MySQL Guide explains how to install and configure HA for MySQL.

Note – This Oracle Solaris Cluster release supports systems that use the SPARC and x86 families of processor architectures: UltraSPARC, SPARC64, AMD64, and Intel 64. In this document, x86 refers to the larger family of 64-bit x86 compatible products. Information in this document pertains to all platforms unless otherwise specified.

This document is intended for system administrators with extensive knowledge of Oracle software and hardware. Do not use this document as a planning or presales guide. Before reading this document, you should have already determined your system requirements and purchased the appropriate equipment and software.

The instructions in this book assume knowledge of the Oracle Solaris Operating System and expertise with the volume-manager software that is used with Oracle Solaris Cluster software.

Using UNIX Commands

This document contains information about commands that are specific to installing and configuring Oracle Solaris Cluster data services. The document does *not* contain comprehensive information about basic UNIX commands and procedures, such as shutting down the system, booting the system, and configuring devices. Information about basic UNIX commands and procedures is available from the following sources:

- Online documentation for the Oracle Solaris Operating System
- Oracle Solaris Operating System man pages
- Other software documentation that you received with your system

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P–2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`
Bash shell, Korn shell, and Bourne shell for superuser	`#`
C shell	`machine_name%`
C shell for superuser	`machine_name#`

Related Documentation

Information about related Oracle Solaris Cluster topics is available in the documentation that is listed in the following table. All Oracle Solaris Cluster documentation is available at `http://docs.sun.com`.

Topic	Documentation
Data service administration	*Oracle Solaris Cluster Data Services Planning and Administration Guide* Individual data service guides
Concepts	*Oracle Solaris Cluster Concepts Guide*
Overview	*Oracle Solaris Cluster Overview*
Software installation	*Oracle Solaris Cluster Software Installation Guide*
System administration	*Oracle Solaris Cluster System Administration Guide*
Hardware administration	*Oracle Solaris Cluster 3.3 Hardware Administration Manual* Individual hardware administration guides
Data service development	*Oracle Solaris Cluster Data Services Developer's Guide*
Error messages	*Oracle Solaris Cluster Error Messages Guide*
Command and function reference	*Oracle Solaris Cluster Reference Manual*

For a complete list of Oracle Solaris Cluster documentation, see the release notes for your release of Oracle Solaris Cluster at `http://docs.sun.com`.

Related Third-Party Web Site References

Third-party URLs that are referenced in this document provide additional related information.

Note – Oracle is not responsible for the availability of third-party web sites mentioned in this document. Oracle does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Oracle will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

See the following web sites for additional resources:

- Documentation (`http://docs.sun.com`)
- Support (`http://www.oracle.com/us/support/systems/index.html`)
- Training (`http://education.oracle.com`) – Click the Sun link in the left navigation bar.

Oracle Welcomes Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of its documentation. If you find any errors or have any other suggestions for improvement, go to `http://docs.sun.com` and click Feedback. Indicate the title and part number of the documentation along with the chapter, section, and page number, if available. Please let us know if you want a reply.

Oracle Technology Network (`http://www.oracle.com/technetwork/index.html`) offers a range of resources related to Oracle software:

- Discuss technical problems and solutions on the Discussion Forums (`http://forums.oracle.com`).
- Get hands-on step-by-step tutorials with Oracle By Example (`http://www.oracle.com/technology/obe/start/index.html`).
- Download Sample Code (`http://www.oracle.com/technology/sample_code/index.html`).

Getting Help

If you have problems installing or using Oracle Solaris Cluster, contact your service provider and provide the following information:

- Your name and email address (if available)
- Your company name, address, and phone number
- The model number and serial number of your systems
- The release number of the Oracle Solaris Operating System (for example, Oracle Solaris 10)
- The release number of Oracle Solaris Cluster (for example, Oracle Solaris Cluster 3.3)

Use the following commands to gather information about each node on your system for your service provider.

Command	Function
`prtconf -v`	Displays the size of the system memory and reports information about peripheral devices
`psrinfo -v`	Displays information about processors
`showrev –p`	Reports which patches are installed
`prtdiag -v`	Displays system diagnostic information
`/usr/cluster/bin/clnode show-rev`	Displays Oracle Solaris Cluster release and package version information

Also have available the contents of the `/var/adm/messages` file.

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

Table 1–1 lists the tasks for installing and configuring HA for MySQL. Perform these tasks in the order that they are listed.

TABLE 1–1 Task Map: Installing and Configuring HA for MySQL

Task	For Instructions, Go To
1. Plan the installation.	"HA for MySQL Overview" on page 14 "Planning the HA for MySQL Installation and Configuration" on page 15

TABLE 1–1 Task Map: Installing and Configuring HA for MySQL *(Continued)*

Task	For Instructions, Go To
2. Install and configure MySQL.	"How to Install and Configure MySQL in a Global Zone" on page 26
3. Verify installation and configuration.	"How to Verify the Installation and Configuration of MySQL" on page 37
4. Install HA for MySQL Packages.	"Installing the HA for MySQL Packages" on page 37
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 40
6. Verify HA for MySQL Installation and Configuration.	"How to Verify the HA for MySQL Installation and Configuration" on page 60
7. Understand HA for MySQL fault monitor.	"Understanding the HA for MySQL Fault Monitor" on page 61
8. Debug HA for MySQL.	"Debugging the HA for MySQL" on page 62
9. Upgrade to Oracle Solaris Cluster 3.3 when using Oracle Solaris Cluster HA for MySQL.	"Upgrade to Oracle Solaris Cluster 3.3 When Using HA for MySQL" on page 63
10. Upgrade to MySQL 4.0.15 from 3.23.54 when using HA for MySQL.	"Upgrade to MySQL 4.x.x From 3.23.54 When Using HA for MySQL" on page 66

HA for MySQL Overview

The MySQL software delivers a fast, multithreaded, 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 a trademark of MySQL AB.

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 1–2 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 Solaris Containers

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

- Zones are containers which are running after a reboot of the node. These containers, combined with resource groups having the node name *nodename:zonename* as a valid "node name" in the resource groups node name list.
- HA Containers are managed by the Solaris Container agent, 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 3.3 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 Container

The MySQL configuration in an HA container uses the `smf` component of Oracle Solaris Cluster HA for Solaris Containers. The registration of the MySQLdata service in an HA container 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 1–1 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 MySQL deployed using Veritas Volume Manager.

```
# 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 ..
```

EXAMPLE 1–1 MySQL Instances with Cluster File Systems *(Continued)*

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

EXAMPLE 1–1 MySQL Instances with Cluster File Systems *(Continued)*

```
#
```

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 1–2 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 MySQL deployed using Veritas Volume Manager.

```
# 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
#
```

EXAMPLE 1–2 MySQL Instances with Highly Available Local File Systems *(Continued)*

```
#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/rdsk/dg2/vol01  /local/mysql-data-1    ufs     2
yes logging
/dev/vx/dsk/dg2/vol01   /dev/vx/rdsk/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 1–3 Dependencies Between HA for MySQL Components in Failover Configurations

Component	Dependency

MySQL resource in a global or non-global zone.	`SUNW.HAStoragePlus` This dependency is required only if the configuration uses a failover file system or file systems in a non-global zone.
	`SUNW.LogicalHostName`
	`SUNW.SharedAddress`

MySQL resource in an HA container	Oracle Solaris Cluster HA for Solaris Container boot resource. `SUNW.HAStoragePlus` `SUNW.LogicalHostName` — This dependency is required only if the zone's boot resource does not manage the zone's IP address.

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 1–3 MySQL Configuration and Registration File for Oracle Solaris Cluster

```
# 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
```

EXAMPLE 1–3 MySQL Configuration and Registration File for Oracle Solaris Cluster *(Continued)*

```
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

#ident  "@(#)ds_template.sgm 1.30     10/08/18 SMI"

# 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.
#
#        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
#    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.
```

EXAMPLE 1–3 MySQL Configuration and Registration File for Oracle Solaris Cluster *(Continued)*

```
#
#       The following examples illustrate sample parameters
#       for MySQL
#
#       BASEDIR=/usr/local/mysql
#       DATADIR=/global/mysqldata
#       MYSQLUSER=mysql
#       LH=mysqllh
#       SCALABLE=
#       LB_POLICY=
#       MYSQLHOST=mysqllh
#       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=
```

Explanation of the `my.cnf` File

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 a important file within MySQL. Refer to the MySQL documentation for complete configuration information on the parameters that follow.

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

- bind address 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.

- **socket** 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.
- binlog-ignore-db contains sc3_test_database, if log-bin option is being used.

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

- bind address 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.

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

  Note – Beginning inMySQL 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.

- master-info-file 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 15 section.

- MySQL installed from `http://www.mysql.com` on a cluster file system, with a MySQL instance on a cluster file system (Example 1–1)
  - *MySQL Basedirectory*— `/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 1–2)
  - *MySQL Basedirectory*— `/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 zone where you will install MySQL. Valid zone types are, the global zone, the HA container, or a non-global zone.

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

- "Enabling MySQL to Run in a Global Zone Configuration" on page 25
- "How to Install and Configure MySQL in a Global Zone" on page 26

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

- "Enabling MySQL to Run in a Zone Configuration" on page 28
- "How to Install and Configure MySQL in a Zone" on page 30

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

- "How to Enable MySQL to run in an HA Container Configuration" on page 33
- "How to Install and Configure MySQL in an HA Container" on page 34

You will find installation examples for each zone 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 Container"
- Appendix C, "Deployment Example: Installing MySQL in a Non-Global Zone"
- Appendix D, "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 25
- "How to Enable MySQL to Run in a Scalable Configuration" on page 25
- "How to Enable MySQL to Run in a Multiple-Master Configuration" on page 26

▼ How to Enable MySQL for a Failover Resource

1 **Become superuser or assume a role that provides `solaris.cluster.admin`RBAC authorization on one of the nodes in the cluster that will host MySQL.**

2 **Register the `SUNW.gds` and `SUNW.HAStoragePlus` resource type.**

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

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.HAStoragePlus \
-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 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization on one of the nodes in the cluster that will host MySQL.**

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 -eM -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 superuser or assume a role that provides `solaris.cluster.admin` RBAC 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 (`SUNWscmys`) was not installed during your initial Oracle Solaris Cluster installation, proceed to "Installing the HA for MySQL Packages" on page 37 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 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 15 for a description of the *MySQL Database directory* and to "Installing and Configuring MySQL" on page 24 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|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
```

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
# ./scripts/mysql_install_db  --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 29

- "How to Enable MySQL to Run in a Scalable Configuration" on page 29
- "How to Enable MySQL to Run in a Multiple-Master Configuration" on page 30

▼ How to Enable MySQL for a Failover Resource

1 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization on one of the nodes in the cluster that will host MySQL.**

2 **Create and boot your zone *MySQL-zone* on all the nodes to host your MySQL database.**

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

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

4 **Create a failover resource group.**

```
# clresourcegroup create \
-n node1:MySQL-zone,node2:MySQL-zone \
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 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization on one of the nodes in the cluster that will host MySQL.**

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

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

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

4 **Create a failover resource group.**

```
# clresource create -n node1:zone1,node2:zone2 SharedAddress-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 -eM -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:zone1,node2:zone2 \
MySQL-scalable-resource-group
```

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

1 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC 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:zone1,node2:zone2 \
MySQL-scalable-resource-group
```

▼ How to Install and Configure MySQL in a Zone

Before You Begin If the HA for MySQL package (`SUNWscmys`) was not installed during your initial Oracle Solaris Cluster installation, proceed to "Installing the HA for MySQL Packages" on page 37 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 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization on one of the nodes in the cluster that will host MySQL.**

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

IMySQL 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 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:zones` 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 15 for a description of the *MySQL Database directory* and to "Installing and Configuring MySQL" on page 24 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 Container Configuration

1 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization on one of the nodes in the cluster that will host MySQL.**

2 **As superuser register the `SUNW.HAStoragePlus` and the `SUNW.gds` resource types.**

```
# clresourcetype register SUNW.HAStoragePlus 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.HAStoragePlus \
-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 zone.**

Install the zone 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 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/SUNWsczone/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 Container registration script.**

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

e. **Enable the Solaris Container 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 Container

Before You Begin If the HA for MySQL package (SUNWscmys) was not installed during your initial Oracle Solaris Cluster installation, proceed to “Installing the HA for MySQL Packages” on page 37 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 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization on one of the nodes in the cluster that will host MySQL.**

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 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 15 section for a description of the *MySQL Database directory* and to "Installing and Configuring MySQL" on page 24 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 Packages

If you did not install the HA for MySQL packages during your initial Oracle Solaris Cluster installation, perform this procedure to install the packages. To install the packages, use the `installer` program.

Note – You need to install the HA for MySQL packages in the global cluster and not in the zone cluster.

How to Install the HA for MySQL Packages

Perform this procedure on each cluster node where you are installing the HA for MySQL packages.

You can run the `installer` program with a command-line interface (CLI) or with a graphical user interface (GUI). The content and sequence of instructions in the CLI and the GUI are similar.

Note – Even if you plan to configure this data service to run in non-global zones, install the packages for this data service in the global zone. The packages are propagated to any existing non-global zones and to any non-global zones that are created after you install the packages.

Before You Begin Ensure that you have the Oracle Solaris Cluster installation media.

If you intend to run the `installer` program with a GUI, ensure that your `DISPLAY` environment variable is set.

1. **On the cluster node where you are installing the data service packages, become superuser.**

2. **Load the Oracle Solaris Cluster installation media into the DVD-ROM drive.**

   If the Volume Management daemon `vold`(1M) is running and configured to manage DVD-ROM devices, the daemon automatically mounts the DVD-ROM on the `/cdrom` directory.

3. **Change to the installation wizard directory of the DVD-ROM.**

   - **If you are installing the data service packages on the SPARC platform, type the following command:**

     ```
     # cd /cdrom/cdrom0/Solaris_sparc
     ```

   - **If you are installing the data service packages on the x86 platform, type the following command:**

     ```
     # cd /cdrom/cdrom0/Solaris_x86
     ```

4. **Start the installation wizard.**

   ```
   # ./installer
   ```

5. **When you are prompted, accept the license agreement.**

6 **From the list of Oracle Solaris Cluster agents under Availability Services, select the data service for MySQL.**

7 **If you require support for languages other than English, select the option to install multilingual packages.**

English language support is always installed.

8 **When prompted whether to configure the data service now or later, choose Configure Later.**

Choose Configure Later to perform the configuration after the installation.

9 **Follow the instructions on the screen to install the data service packages on the node.**

The installation wizard displays the status of the installation. When the installation is complete, the wizard displays an installation summary and the installation logs.

10 **(GUI only) If you do not want to register the product and receive product updates, deselect the Product Registration option.**

The Product Registration option is not available with the CLI. If you are running the installation wizard with the CLI, omit this step.

11 **Exit the installation wizard.**

12 **Unload the installation media from the DVD-ROM drive.**

a. **To ensure that the DVD-ROM is not being used, change to a directory that does *not* reside on the DVD-ROM.**

b. **Eject the DVD-ROM.**

```
# eject cdrom
```

Next Steps See "Registering and Configuring HA for MySQL" on page 39to register HA for MySQL and to configure the cluster for the data service.

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 40
- "How to Register and Configure HA for MySQL as a Failover Service in a Zone Configuration" on page 44
- "How to Register and Configure HA for MySQL as a Failover Service in an HA Container Configuration" on page 49

▼ 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 Packages" on page 37.

1 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization on the node in the cluster that host MySQL.**

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/hahostix1.sock -uroot -p'admin-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/hahostix1.sock -uroot -p'admin-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, i.e.

```
#
# 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 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
```

```
#ident  "@(#)ds_template.sgm 1.30     10/08/18 SMI"

# 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_SOCK=

# 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=root
```

```
MYSQL_HOST=hahostix1
FMUSER=fmuser
FMPASS=fmuser
MYSQL_SOCK=/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/mysqld.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.**

```
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

#ident   "@(#)ha_mysql_config.ksh 1.8     09/09/01 SMI"

# 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
#
#        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 Logical Hostname Oracle Solaris Cluster resource
# MYSQLHOST - Name of the host in /etc/hosts
#    FMUSER - Name of the MySQL fault monitor user
#    FMPASS - Name of the MySQL fault monitor user password
#    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 Oracle Solaris Cluster resource
#
```

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

c. **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 Packages" on page 37.

1 **Log in to the zone on the node that hosts your*MySQLresource-group*.**

2 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization on the node's zone in the cluster that hosts MySQL.**

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 -p'adminpassword'
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 -p'adminpassword'
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 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

#ident   "@(#)mysql_config.ksh 1.3     06/03/08 SMI"

# 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
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_SOCK=

# 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=root
MYSQL_HOST=hahostix1
FMUSER=fmuser
FMPASS=fmuser
MYSQL_SOCK=/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/mysqld.pid
```

9 **Leave the zone.**

10 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization in the global zone of the cluster node that hosts MySQL.**

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 the *home-dir* directory.**

```
# cd home-dir
```

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

```
#
# Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

#ident   "@(#)ha_mysql_config.ksh 1.3     06/03/08 SMI"

# 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
```

```
#
#        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 Logical Hostname Oracle Solaris Cluster resource
# MYSQLHOST - Name of the host in /etc/hosts
#    FMUSER - Name of the Mysql fault monitor user
#    FMPASS - Name of the Mysql fault monitor user password
#    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 Oracle Solaris Cluster resource
#
```

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

c. 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 Register and Configure HA for MySQL as a Failover Service in an HA Container 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 Packages" on page 37.

1 **Log in to the zone on the node that hosts your MySQL resource-group.**

2 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization on the node's zone in the cluster that hostsMySQL.**

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 -p'admin-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 -p'admin-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 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.**

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

```
# cd home-dir
```

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

```
#
# Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

#ident   "@(#)mysql_config.ksh 1.3     06/03/08 SMI"

# 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
```

```
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_SOCK=

# 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=root
MYSQL_HOST=hahostix1
FMUSER=fmuser
FMPASS=fmuser
MYSQL_SOCK=/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/mysqld.pid
```

9 **Leave the zone.**

10 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization in the global zone of the cluster node that hosts MySQL.**

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.

```
#
# Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

#ident   "@(#)ha_mysql_config.ksh 1.3     06/03/08 SMI"

# 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
#
#        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 Logical Hostname Oracle Solaris Cluster resource
# MYSQLHOST - Name of the host in /etc/hosts
#    FMUSER - Name of the MySQL fault monitor user
#    FMPASS - Name of the MySQL fault monitor user password
#    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 Oracle Solaris Cluster resource
#
```

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=fmuser
LOGDIR=/global/mysql-data/logs
CHECK=YES
HAS_RS=mysql-has-res
ZONE=zone1
ZONE_BT=zone1-rs
PROJECT=MySQL-project
```

c. **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 container. 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 **Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations on the zones console.**

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 container.

4 **Disconnect from the HA container's console.**

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

1 **Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC 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 superuser to the HA container's console.**

4 **Unregister HA for MySQL from the Solaris Service Management Facility (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:

a. Make a copy of the default file, `/opt/SUNWscmys/util/ha_mysql_config`.

b. Set the `ZONE` and `RS` parameters with the values that are used by the data service.

c. Run the `ha_mysql_smf_remove` command and use the `-f` option to specify this configuration file.

---

5 **Disconnect from the HA container'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 Packages" on page 37.

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 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization on the cluster node that hosts MySQL.**

3 **Start the MySQL Server instance manually.**

```
# cd MySQL-Base-directory
# ./bin/mysqld --defaults-file=MySQL-Database-directry/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 admiistrative 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 -p'admin-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 `Link: 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 2009 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
#ident "@(#)ds_template.sgm 1.30 10/08/18 SMI"
# 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_SOCK=
# 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=fmuser
MYSQL_SOCK=/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_SOCK=`*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-directry/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 2009 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
#ident "@(#)ds_template.sgm 1.30 10/08/18 SMI"
# 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 sclable 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.
##
# 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
# 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=mysqllh
# SCALABLE=
# LB_POLICY=
# MYSQLHOST=mysqllh
# 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=
MYSQLHOST=hahostix1
FMUSER=fmuser
FMPASS=fmuser
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. **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 -eM 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 **Become superuser 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 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

- **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 will also check the following:
    - If the MySQL instance is a slave configuration, the probe will check whether the MySQL instance is connected to its master. If the Slave is not connected, the probe will write an error message to syslog.
    - Verify that the probe can list all databases and tables, not the contents. If the probe receives any errors the probe will write an error message to syslog.
    - Conduct an function test 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 will restart the MySQL resource.

- If all MySQL processes have died, pmf will interrupt 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 container configuration.**

If your operating system is Solaris 10 and your MySQL resource is dependent on a Solaris Container boot component resource, you are in an HA container configuration. In any other case, especially on a Solaris 9 system, you are in a global 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.**

- Solaris 9:

  ```
  # pkill -1 syslogd
  ```

- Solaris 10:

```
# svcadm refresh svc:/system/system-log:default
```

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, changing `daemon.debug` to `daemon.notice` and changing the `DEBUG` variable to `DEBUG=`.

Example 1–4 Editing the Debug `Config` File

```
# cat /opt/SUNWscmys/etc/config
#
# Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Usage:
#       DEBUG=<RESOURCE_NAME> or ALL
#
DEBUG=ALL
#
```

Upgrade to Oracle Solaris Cluster 3.3 When Using HA for MySQL

Use the information in this section to understand how to upgrade to Oracle Solaris Cluster 3.3 when using HA for MySQL.

Note – This procedure will not describe how to upgrade to Oracle Solaris Cluster 3.3. It includes only the steps to upgrade HA for MySQL to Oracle Solaris Cluster 3.3.

▼ Upgrade From Sun Cluster 3.1 8/05 to Oracle Solaris Cluster 3.3 When Using HA for MySQL

Use this procedure to upgrade HA for MySQL from Sun Cluster 3.1 8/05 to Oracle Solaris Cluster3.3.

1 **Shut down the HA for MySQL resource.**

```
# clresource disableMySQL-resource
```

2 Upgrade the nodes to Oracle Solaris Cluster 3.3 according to *Oracle Solaris Cluster Upgrade Guide*.

3 Start the MySQL Server manually on Oracle Solaris Cluster 3.3.

```
#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 Access the MySQL instance from the local node with the socket option.

```
# cd MySQL-Base-directory
/bin/mysql -S MySQL-socket -uroot \
 -pAdminpassword \
```

The following is an example for a MySQL instance.

```
# mysql -s /tmp/hahostix1.sock -uroot -proot
mysql>
```

5 Drop the HA for MySQL test database `sc3_test_database`.

```
# mysql -s /tmp/hahostix1.sock -uroot -proot
mysql> DROP DATABASE sc3_test_database;
Query OK, 0 rows affected (0.03 sec)
```

6 Delete all entries in `db-table` in `mysql-database` that contain `User='`*MySQL-Faultmonitor-user*`'`.

```
# mysql -s /tmp/hahostix1.sock -uroot -proot
mysql> use mysql;
Database changed
mysql> DELETE FROM db WHERE User='fmuser';
Query OK, 1 row affected (0.03 sec)
```

7 Delete all entries in `user-table` in `mysql-database` that contain `User='`*<MySQL Faultmonitor user>*`'`.

```
# mysql -s /tmp/hahostix1.sock -uroot -proot
mysql> use mysql;
Database changed
mysql> DELETE FROM user WHERE User='fmuser';
Query OK, 1 row affected (0.03 sec)
```

8 Add a fault-monitor user and a test database to MySQL.

a. Navigate to the `util` directory.

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

b. **Copy the `mysql_config` to *home-dir* and edit the `mysql_config` file and follow the comments within that file.**

```
# Where is MySQL installed (BASEDIR)
MYSQL_BASE=

# MySQL admin-user for localhost (Should be root)
MYSQL_USER=

# Password for MySQL admin user
MYSQL_PASSWD=

# Configured logicalhost
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_SOCK=

# 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=

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

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=fmuser
MYSQL_SOCK=/tmp/hahostix1.sock
MYSQL_NIC_HOSTNAME="clusterix1 clusterix2"
MYSQL_DATADIR=/global/mysql-data
NDB_CHECK=
```

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

```
# ./mysql_register -f home-dir/mysql_config
```

10 **Stop the MySQL Server manually.**

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

11 **Start the HA for MySQL resource.**

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

Upgrade to MySQL 4.x.x From 3.23.54 When Using HA for MySQL

Use the information in this section to understand how to upgrade to 4.0.12 when using HA for MySQL.

Note – The procedure will not describe how to upgrade to version 4.x.x. It includes only the steps to upgrade HA for MySQL to version 4.x.x. These steps assume that the new MySQL binaries are installed in the same place. If the new binaries are installed in a new directory, you must reregister the MySQL resource with a new MySQL Base directory.

▼ Upgrade to MySQL Version 4.x.x From Version 3.23.54

Use this procedure to upgrade from MySQL version 3.23.54 to MySQL version 4.x.x.

1 **Shut down the HA for MySQL resource.**

```
# clresource disable MySQL-resource
```

2 **Become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization on one of the nodes in the cluster that will host MySQL.**

3 **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 files systems, see “Determining the Location of the Application Binaries” in *Oracle Solaris Cluster Data Services Planning and*

Administration Guide?Determining the Location of the Application Binaries? in Oracle Solaris Cluster 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.

4 **Start up the HA for MySQL resource.**

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

5 **Shut down the HA for MySQL fault monitor.**

```
# clresource unmonitor MySQL-resource
```

6 **Follow the instructions in the MySQL Documentation to upgrade the MySQL database.**

7 **Start up the HA for MySQL fault monitor.**

```
# clresource monitor MySQL-resource
```

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 Container," or Appendix C, "Deployment Example: Installing MySQL in a Non-Global 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, 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 10 10/09 software for SPARC or x86 platforms
- Oracle Solaris Cluster 3.3 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.

Note – The steps for installing MySQL in a cluster that runs on the Solaris 9 OS are identical to the steps in this example.

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 Korn shell environment. If you use a different shell, replace any Korn shell-specific information or instructions with the appropriate information for you preferred shell environment.
- **User login:** Unless otherwise specified, perform all procedures as superuser or assume 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 70
- "Example: Configuring Cluster Resources for MySQL" on page 71
- "Example: Installing and Bootstrapping the MySQL Software on Local Storage" on page 71
- "Example: Modifying the MySQL Configuration Files" on page 73
- "Example: Enabling the MySQL Software to Run in the Cluster" on page 74

▼ Example: Preparing the Cluster for MySQL

1 **Install and configure the cluster as instructed in *Oracle Solaris Cluster 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-architcture_64.tar.gz
phys-schost-1# ln -s mysql-max-5.0.22-solaris10-architcture_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 /usr/local/mysql
phys-schost-1# ./scripts/* --datadir=/global/mnt3/mysql-data
```

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=root
MYSQL_HOST=ha-host-1
FMUSER=fmuser
FMPASS=fmuser
MYSQL_SOCK=/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=fmuser
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
```

5. **Stop the MySQL database.**

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

6. **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
```

7. **Enable the resource.**

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

APPENDIX B

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

This appendix presents a complete example of how to install and configure the MySQL application and data service in a non-global high-availability container. 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." For an installation in a zone, see Appendix C, "Deployment Example: Installing MySQL in a Non-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 10 10/09 software for SPARC or x86 platforms
- Oracle Solaris Cluster 3.3 core software
- Oracle HA for MySQL
- Oracle Solaris Cluster HA for Solaris Container
- 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 Korn shell environment. If you use a different shell, replace any Korn shell-specific information or instructions with the appropriate information for you preferred shell environment.
- **User login:** Unless otherwise specified, perform all procedures as superuser or assume 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 Container

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

- "Example: Preparing the Cluster for MySQL" on page 76
- "Example: Configuring Cluster Resources for MySQL" on page 77
- "Example: Configuring the HA Container" on page 77
- "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 81

▼ Example: Preparing the Cluster for MySQL

1. **Install and configure the cluster as instructed in *Oracle Solaris Cluster 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 Containers

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.HAStoragePlus
   ```

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

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

3. **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
   ```

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 *Oracle Solaris Cluster Data Service for Solaris Containers 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-architcture_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 /usr/local/mysql
zone# ./scripts/* --datadir=/global/mnt3/mysql-data
```

9 **Create your `my.cnf` `config-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=root
MYSQL_HOST=ha-host-1
FMUSER=fmuser
FMPASS=fmuser
MYSQL_SOCK=/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
```

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=fmuser
LOGDIR=/global/mnt3/mys-data/logs
CHECK=YES
NDB_CHECK=
```

9 **Save and close the file.**

10 **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
```

11 **Enable the resource.**

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

APPENDIX C

Deployment Example: Installing MySQL in a Non-Global Zone

This appendix presents a complete example of how to install and configure the MySQL application and data service in a non-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 in the global zone, see Appendix A, "Deployment Example: Installing MySQL in the Global Zone." For installation in an HA container, see Appendix B, "Deployment Example: Installing MySQL in the Non-Global HA Container."

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

Software Configuration

This deployment example uses the following software products and versions:

- Oracle Solaris 10 10/09 software for SPARC or x86 platforms
- Oracle Solaris Cluster 3.3 core software
- Oracle Solaris Cluster Data Service for PostgreSQL
- MySQL version 5.0.22 tar file

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 Korn shell environment. If you use a different shell, replace any Korn shell-specific information or instructions with the appropriate information for you preferred shell environment.
- **User login:** Unless otherwise specified, perform all procedures as superuser or assume a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` RBAC authorization.

Installing and Configuring MySQL on Local Storage in a Non-Global Zone

These instructions assume that you are installing the MySQL software as the `mysql` user in a local directory.

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

- "Example: Preparing the Cluster for MySQL" on page 84
- "Example: Configuring the Zone" on page 84
- "Example: Configuring Cluster Resources for MySQL" on page 86
- "Example: Installing the MySQL Software on Local Storage" on page 86
- "Example: Bootstrapping the MySQL Software on Local Storage" on page 87
- "Example: Modifying the MySQL Configuration File" on page 88
- "Example: Enabling the MySQL Software to Run in the Cluster" on page 89

▼ Example: Preparing the Cluster for MySQL

● **Install and configure the cluster as instructed in *Oracle Solaris Cluster Software Installation Guide*.**

Install the following cluster software components on both nodes.

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

▼ Example: Configuring the Zone

In this task you will install the Solaris Container on `phys-schost-1` and `phys-schost-2`. Perform this procedure on both hosts.

1 **On local storage of both cluster nodes, create a directory for the zone 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 /zones
```

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

```
create -b
set zonepath=/zones/clu1
set autoboot=true
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=zone-1      Choose a different addtress (zone-2) on the second node.
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-1# zoneadm -z clu1 boot
```

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

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

▼ Example: Configuring Cluster Resources for MySQL

1 **Register the HAStoragePlus resource type.**

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

2 **Create the MySQL resource group.**

```
phys-schost-1# clresourcegroup create -n phys-host-1:clu1,phys-host-2:clu1 RG-MYS
```

3 **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
```

4 **Enable the resource group.**

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

▼ Example: Installing 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 the `/usr` directory, 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
phys-schost-2# groupadd -g 1000 mysql
phys-schost-2# 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-architcture_64.tar.gz
phys-schost-1# ln -s mysql-max-5.0.22-solaris10-architcture_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
phys-schost-2# zlogin clu1
```

5 **Add the `mysql` group and user.**

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

6 **Leave the zone.**

▼ Example: Bootstrapping the MySQL Software on Local Storage

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

1 **Log in to the zone.**

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

2 **Create your database directories.**

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

3 **Bootstrap MySQL.**

```
zone-1# cd /usr/local/mysql
zone-1# ./scripts/* --datadir=/global/mnt3/mysql-dat
```

4 **Create your `my.cnf config-file` in `/global/mnt3/mysql-data`.**

```
zone-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/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
```

5. **Change the ownership of the MySQL data directory.**

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

6. **Change the permission of the `my.cnf` file.**

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

▼ Example: Modifying the MySQL Configuration File

In this task, you will create the configuration file to prepare the MySQL database. It is assumed that you are still logged in to this zone.

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

```
zone-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=root
MYSQL_HOST=ha-host-1
FMUSER=fmuser
FMPASS=fmuser
MYSQL_SOCK=/tmp/ha-host-1.sock
MYSQL_NIC_HOSTNAME="zone-1 zone-2"
MYSQL_DATADIR=/global/mnt3/mys-data
```

3 **Save and close the file.**

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

This task will initialize and prepare your database. It is essential that you perform it on one node only. It is assumed that you are still logged in to this zone.

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

```
zone-1# cd /usr/local/mysql
zone-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.**

```
zone-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.**

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

4 **Prepare the Oracle Solaris Cluster specific test database.**

```
zone-1# ksh /opt/SUNWscmys/util/mysql_register -f /global/mnt3/mysql_config
```

5 **Stop the MySQL database.**

```
zone-1# 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
HAS_RS=RS-MYS-HAS
...
BASEDIR=/usr/local/mysql
DATADIR=/global/mnt3/mysql-data
MYSQLUSER=mysql
MYSQLHOST=ha-host-1
FMUSER=fmuser
FMPASS=fmuser
LOGDIR=/global/mnt3/mys-data/logs
CHECK=YES
```

9 **Save and close the file.**

10 **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
```

11 **Enable the resource.**

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

APPENDIX D

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," Appendix B, "Deployment Example: Installing MySQL in the Non-Global HA Container," or Appendix C, "Deployment Example: Installing MySQL in a Non-Global 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 10 10/09 software for SPARC or x86 platforms
- Oracle Solaris Cluster 3.3 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 Korn shell environment. If you use a different shell, replace any Korn shell-specific information or instructions with the appropriate information for you preferred shell environment.
- User login: Unless otherwise specified, perform all procedures as superuser or assume 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 92
- "Example: How to Configure MySQL in a Scalable Resource Group" on page 93
- "Example: How to Configure MySQL in a Multiple—Master Resource Group" on page 93
- "Example: How to Install and Bootstrap the MySQL Software on Local Storage " on page 94
- "Example: How to Modify the MySQL Configuration Files" on page 95
- "Example: How to Enable the MySQL Software to Run in the Cluster " on page 96

▼ Example: How to Prepare the Cluster for MySQL

Before You Begin Install and configure the cluster as instructed in *Oracle Solaris Cluster 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 93.

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 -eM RG-MYSQL
   phys-schost-1# clresourcegroup online -eM 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 93.

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 Mximum_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 /usr/local/mysql
phys-schost-1# ./scripts/* --datadir=/local/mysql-data
```

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=root
MYSQL_HOST= phys-schost-1
FMUSER=fmuser
FMPASS=fmuser
```

```
MYSQL_SOCK=/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=fmuser
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 **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
```

7 **Enable the resource.**

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

Index