



# **SteelEye Protection Suite for Linux: Apache/MySQL**

## **Evaluation Guide**

This document and the information herein is the property of SIOS Technology Corp. Any unauthorized use and reproduction is prohibited. SIOS Technology Corp. makes no warranties with respect to the contents of this document. SIOS Technology Corp. reserves the right to revise this publication and make changes to the products described herein without prior notification.

---

SteelEye Technology, LifeKeeper and DataKeeper are registered trademarks, and SteelEye is a trademark of SIOS Technology Corp.

DB2 Universal Database and IBM Director are trademarks of International Business Machines Corporation.

IBM is a registered trademark of International Business Machines Corporation in the U.S. and other countries.

Microsoft, Windows, Windows Server 2008, Windows Server 2003, Windows 2000, Windows NT, SQL Server and Exchange are trademarks or registered trademarks of Microsoft in the U.S. and other countries.

Other brand and product names used herein are for identification purposes only and may be trademarks of their respective companies.

It is the policy of SIOS Technology Corp. to improve products as new technology, components and software become available. SIOS Technology Corp. therefore, reserves the right to change specifications without prior notice.

To maintain the quality of our publications, we welcome your comments on the accuracy, clarity, organization and value of this document.

Email correspondence to:

[evalsupport@us.sios.com](mailto:evalsupport@us.sios.com)

Copyright © 2007-2012

By SIOS Technology Corp.

San Mateo, CA U.S.A.

All Rights Reserved Worldwide

© by SIOS Technology Corp. All rights reserved worldwide.

# SteelEye Protection Suite for Linux: Evaluation Guide

## Table of Contents

Objective .....	5
Terms to Know.....	5
The Evaluation Process.....	7
Phase 1 – Prepare to Install.....	8
Hardware Requirements .....	8
Primary and Secondary Servers.....	8
Client.....	8
Software Requirements.....	8
Primary Server and Secondary Server .....	8
Network Requirements .....	10
Primary Server and Secondary Servers.....	11
Client.....	11
Phase 2 – Configure Storage.....	11
Before You Begin .....	11
Partition local storage for use with SteelEye Data Replication .....	11
Primary Server .....	11
Secondary Server.....	13
Configure iSCSI initiator, discover and login to iSCSI target .....	13
Phase 3 – Install and Configure Apache and PHP .....	15
Before You Begin .....	15
Primary Server .....	15
Secondary Server.....	16
Phase 4 – Install, Configure, and Start MySQL .....	16
Primary Server .....	16
Secondary Server.....	18

Phase 5 – Install SteelEye Protection Suite for Linux.....	19
Download Software .....	19
Run the SteelEye Protection Suite Installer Script.....	19
Install the Evaluation License Keys .....	21
Start the SteelEye Protection Suite for Linux .....	21
Phase 6 - Configure the Cluster .....	21
Primary Server .....	21
Access the LifeKeeper GUI.....	22
Create Communication (Comm) Paths .....	23
Verify the Communications Paths .....	28
Create the LifeKeeper Hierarchy .....	29
Create and Extend an IP Resource.....	29
Create a Second IP Resource .....	34
Create a Mirror and Begin Data Replication.....	35
Create the Apache Hierarchy.....	39
Create the Shared Filesystem Resource Hierarchy.....	41
Create the MySQL Resource Hierarchy .....	43
Create the MySQL IP Address Dependency .....	44
Phase 7 –Test Your Environment.....	45
1. Manual Switchover of the MySQL Hierarchy to Secondary Server.....	45
2. Manual Switchover of the Apache Hierarchy to Secondary Server .....	47
3. Manual Switchover of the MySQL Hierarchy back to Primary Server .....	51
4. Manual Switchover of the Apache Hierarchy back to the Primary Server .....	52
5. Simulate a network failure on the Primary Server by failing the IP resource.....	55
6. Hard failover of the resource from the Secondary Server back to the Primary Server .....	56
7. Bring Failed Server back on line.....	57
8. Verify Local Recovery of MySQL Server .....	58

## Objective

This document is intended to aid you in installing, configuring and using the SteelEye Protection Suite for Linux evaluation product, to make Apache and MySQL highly available. If Apache and MySQL are not already installed, please allocate some time to install it on your servers. Once this task has been completed, you may install and configure SteelEye Protection Suite for Linux.

There are five phases in this process:

- Phase 1 – Prepare to Install
- Phase 2 – Configure Storage
- Phase 3 – Install and Configure Apache/PHP
- Phase 4 – Install and Configure MySQL
- Phase 5 – Install SteelEye Protection Suite for Linux
- Phase 6 – Configure your LifeKeeper Cluster
- Phase 7 – Test Your Environment

## Terms to Know

The following terms are used throughout this document and, while some may be familiar to you, it may be helpful to review how SteelEye defines and uses these terms.

### Network Communication Terms

**Crossover cable** – A cable used to directly connect computing devices together, instead of being connected to a network switch, hub or router. This cable creates an isolated, private network to allow cluster-related and data replication traffic to flow between systems.

### Types of LifeKeeper Servers

**Server** – A computer system dedicated to running software application programs.

**Active Server** – This is the server where the resource hierarchy is currently running (IN SERVICE).

**Standby Server** – This is the server where the resource hierarchy is defined, but is not currently running. This server is available to bring the resource hierarchy into service should something happen to the resource hierarchy on the Active Server.

**Primary Server** – This is the server in a LifeKeeper configuration with the highest priority for a given resource hierarchy. It is the server that provides services for the resource hierarchy under normal circumstances.

**Secondary Server** – This is the server in a LifeKeeper configuration with the 2nd highest priority for a given resource hierarchy.

**Source Server** - In a LifeKeeper cluster, using data replication, this is the Active Server. It is where the resource hierarchy is currently running and the replicated partition (Source Partition) is accessible for writes.

**Target Server** – In a LifeKeeper cluster, using data replication, this is the Standby Server. The replicated partition (Target Partition) is updated with writes from the Source Partition by the SteelEye Data Replication system. This partition should not be accessed/modified manually.

## SteelEye Data Replication Terms

**Replication** – Transferring data from one partition to another via a sector-by-sector copy. During replication, the target partition should not be accessed or modified assuring your data integrity.

**Synchronous** – A replication scheme in which the data is confirmed written and valid on the target before the write operation occurs on the source disk through a series of information exchanges. Synchronous mirrors should only be implemented on high speed (100Mbps+) networks due to the network overhead involved.

**Asynchronous** – A replication scheme in which the data is released for writing on the source immediately and is sent to the target(s) simultaneously for writing as fast as the data can get there and can be written on them.

**Rate of Change** – A measure of the amount of data which is changing over a set period of time.

**Compression** – An algorithm which is optionally implemented to reduce the amount of traffic between source and target nodes. Nine levels of compression are offered. Compression is turned off by default.

**Throttling** – An optionally implemented mechanism to limit the bandwidth used for replication.

## LifeKeeper Product Terms

**Communications Path** – A mechanism supporting communication between nodes in a LifeKeeper cluster. SIOS highly recommends implementing multiple communication paths between all servers in the cluster to eliminate a single point of failure.

**Heartbeat** – A periodic message exchanged between nodes in a LifeKeeper cluster that provides server health monitoring. A heartbeat message is one type of inter-node cluster communication sent over a communications path.

**Split Brain** – A situation in which all communications paths between cluster members fail, but all servers remain up and running. In this situation, both systems believe the other has failed and both believe they should keep or bring resources into service.

**Failover** – The unplanned migration of a resource hierarchy to the Standby Server because of a system or resource failure on the Active Server.

**Switchover** – The planned migration of a resource hierarchy from the Active Server to the Standby Server.

**Switchback** – The setting that governs the recovery behavior of the server where the resource was in service when it failed. If the setting is “intelligent”, the server acts as a possible backup for the given

resource. If the setting is “Automatic”, the server actively attempts to re-acquire the resource without further notice.

**Resource** – A system asset that can be protected by LifeKeeper. Resources can be used to represent disk partitions, virtual IP addresses, applications, etc.

**Extend a Resource** – Create or define an already configured LifeKeeper resource onto another server in the cluster and build an equivalency relationship that prevents the resource from coming in service on both systems simultaneously.

**Resource Hierarchy** – A grouping of resources, in a predetermined order, from high to low. This may also be referred to as simply a Hierarchy.

**Shared Storage** – One or more logical disk partitions that are physically attached to all nodes in a cluster. LifeKeeper ensures that the volume is only accessible by one server at a time. This is formally called I/O fencing.

**Data Replication (Disk Mirroring)** – The replication of logical disk partitions to separate physical hard disks in real time to ensure continuous availability, currency and accuracy of data.

**Source** – The partition on the source server used for replication. The “gold” copy of the data.

**Target** – The partition on the target server used for replication.

**Switchable IP Address** – A unique IP address that may be moved between systems in the cluster. Client systems connect to this address and the system where the virtual IP resource is active will respond to requests.

## The Evaluation Process

SIOS strongly recommends performing your evaluation of SteelEye Protection Suite for Linux within a test lab environment. SIOS is not responsible and cannot provide support for evaluation software installed in a production environment.

All questions during the evaluation period should be directed to [evalsupport@us.sios.com](mailto:evalsupport@us.sios.com) or your local Pre-Sales Engineering contact. Pre-sales support will contact you by the next business day to answer questions. Once you are a licensed customer on software maintenance, you will have access to 24 X 7 post-sales technical support.

---

---

**Important**

Your evaluation license is valid for a limited period of time from the day you receive the SIOS product evaluation package and licenses via email from the SIOS sales team.

---

---

## Phase 1 – Prepare to Install

### Hardware Requirements

#### Primary and Secondary Servers

- Systems must meet the minimum requirements for the Linux distribution to be used during the evaluation.
- 512MB RAM minimum; 1GB RAM recommended.
- 2GB of available hard disk space recommended.
- Multiple Network Interface Cards (NIC's) are recommended.
- Configure one or more additional partitions to be used for data replication. On the primary server, these will become the source partitions. On the secondary server(s), these will become the target partitions. In this evaluation example we will be replicating Apache website data (which will be located at /var/www)
- For replicated partitions, a target partition's size must equal to or larger than the size of its source partition.
- The system ( / ) and boot (/boot) partitions are not eligible for replication.

**Note:** You may use more than one partition for replicated data, allowing for separation of multiple database files and/or log directories for performance reasons.

#### Client

This system is not required but is recommended for testing the cluster environment.

- A standard web browser can be used to test the Apache/MySQL configuration.

### Software Requirements

#### Primary Server and Secondary Server

- Linux Distribution:
  - RedHat Enterprise Linux 5 (5.4+ recommended) or 6
  - CentOS Linux 5 (5.4+ recommended) or 6
  - Oracle Enterprise Linux 5 (5.4+ recommended)
  - SuSE Linux Enterprise Server 10 or 11 (11 recommended)
  - See <http://docs.us.sios.com/Linux/8.1/LK4L/ReleaseNotes/> for a full list of supported Operating Systems
- Current patches / security updates are recommended.
- On 64-bit Linux systems, ensure that the appropriate 32bit compatibility packages are installed, for example compat-libstdc++

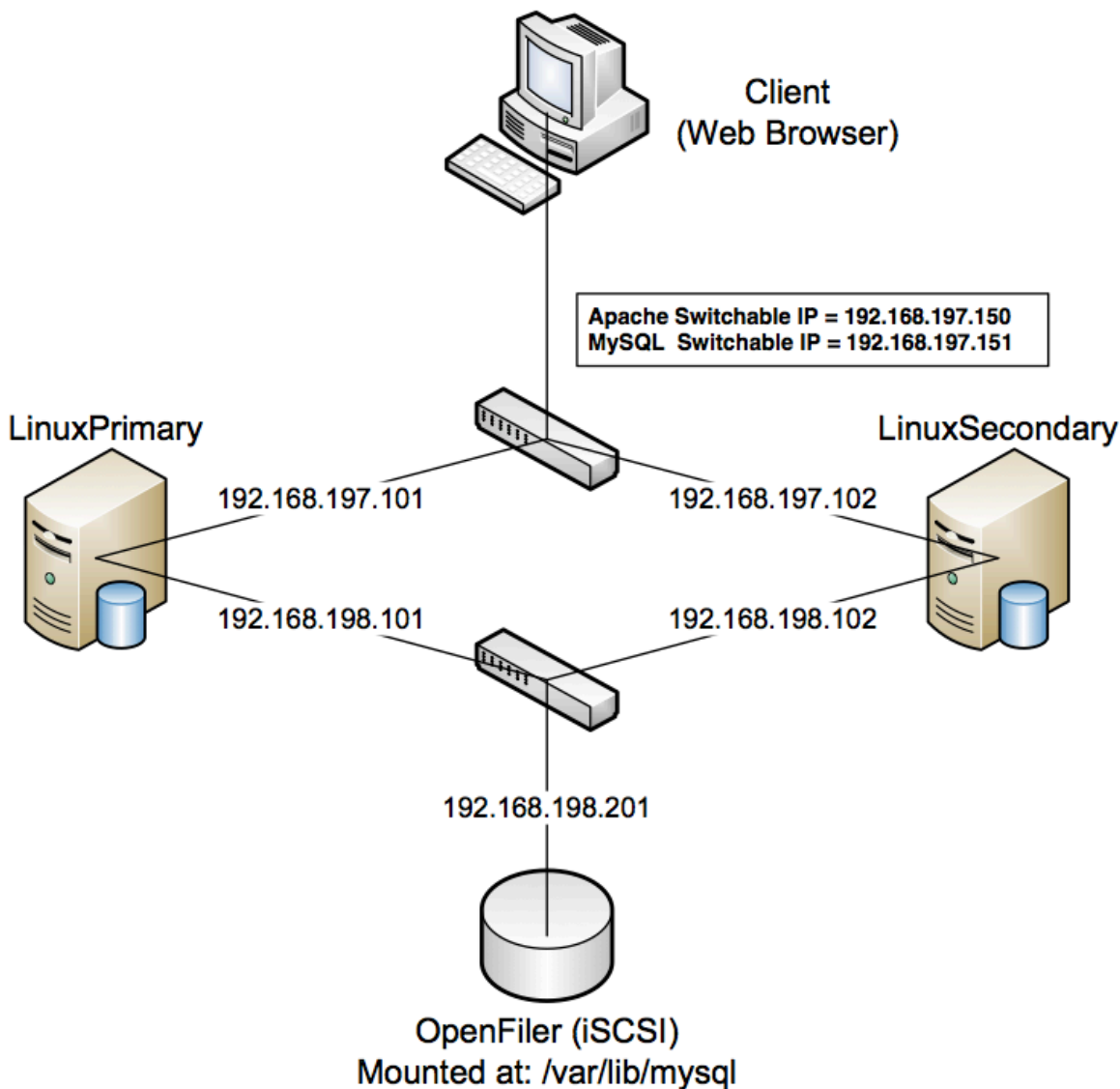
- Its recommended that IPTables is disabled
  - # /etc/init.d/iptables off
  - # chkconfig iptables off
  - See [http://docs.us.sios.com/Linux/8.1/LK4L/TechDoc/Content/user\\_guide/using\\_lifekeeper\\_linux/maintenance\\_tasks/running\\_lifekeeper\\_firewall.htm](http://docs.us.sios.com/Linux/8.1/LK4L/TechDoc/Content/user_guide/using_lifekeeper_linux/maintenance_tasks/running_lifekeeper_firewall.htm) for information regarding the ports SteelEye Protection Suite for Linux uses.
- It's recommended that SELinux is disabled:
  - Edit /etc/selinux/config
  - Set SELINUX=disabled (note: permissive mode is also acceptable)
- Check the configuration of your /etc/hosts file
  - localhost.localdomain and localhost are the only entries that can be on 127.0.0.1
  - create a separate entry for your hostname with a static address

## Network Requirements

For your evaluation, we recommend configuring your machines similarly to the following example.

LinuxPrimary and LinuxSecondary are multi-homed, between two LAN segments (the second NIC in each server could even be connected via a cross-over cable if a second physical network is not available). The second NIC is optional in this configuration, but highly recommended in production environments to avoid a single point of failure.

In this example evaluation scenario we will be leveraging both local, replicated storage (with the Apache configuration) as well as Shared Storage (iSCSI, for the MySQL configuration). OpenFiler is a storage appliance server which will serve an iSCSI target to LinuxPrimary and LinuxSecondary.



## Network Configuration Example

### Primary Server and Secondary Servers

- Configure the Host file with entries for all LifeKeeper protected servers. This is typically /etc/hosts.

Example:

```
192.168.197.101          LinuxPrimary
192.168.197.102          LinuxSecondary
```

- See your Network Administrator to obtain an unused IP Address to be used as the switchable IP Address. This switchable IP Address will be created later in the configuration process.
- Public Network connection(s) configured with:
  - Static IP address
  - Correct subnet mask
  - Correct gateway address
  - Correct DNS server address(es)
- Private Network connection(s) configured with:
  - Static IP address (on a different subnet from the public network)
  - Correct network mask
  - No gateway IP address
  - No DNS server addresses

### Client

Must be able to communicate on the same subnet/network as the servers Public interface addresses. In our example, this is the 192.168.197.0/24 network.

## Phase 2 – Configure Storage

### Before You Begin

Ensure the following:

- You have an extra volume/partition on both servers that can be used for data replication. A target volume's size must equal to or larger than the size of its source volume.
- Shared storage is available. This can either be Fiber Channel SAN, iSCSI, NAS, etc. In this example we will review configuration of an iSCSI target for use as our MySQL database storage repository.

## Partition local storage for use with SteelEye Data Replication

### Primary Server

On your Primary server, perform the following actions:

1. Identify an existing free, unused disk partition to use as our Apache repository. Alternatively, create a new partition. Use the “fdisk” utility to partition your disk appropriately. In this example /dev/sdb is an unused disk where we will create a single partition
  - a. fdisk /dev/sdb
  - b. Press “n” to create a new partition
  - c. Press “p” to create a primary partition
  - d. This example uses a new disk, so we will use all default values (Partition 1, entire disk)  
Hit Enter twice to confirm these parameters
  - e. Press “w” to write the partition table and exit fdisk

---

```
[root@LinuxPrimary ~]# fdisk /dev/sdb

Command (m for help): n
Command action
  e  extended
  p  primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-256, default 1): <enter>
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-256, default 256): <enter>
Using default value 256

Example

Command (m for help): w
The partition table has been altered! Calling ioctl() to re-read partition table.
Syncing disks.

[root@LinuxPrimary ~]#

[root@LinuxPrimary ~]# df /var/www
Filesystem      1K-blocks  Used Available Use% Mounted on
/dev/sdb1        253855    11083  229666  5% /var/www
[root@LinuxPrimary ~]# ls /var/www
cgi-bin error html icons lost+found manual usage
```

---

2. Format the newly created disk partition

```
# mkfs.ext3 /dev/sdb1
```

3. Mount the partition temporarily at /mnt

```
# mount /dev/sdb1 /mnt
```

4. Move any existing data from /var/www/ into this new disk partition (assumes a default apache configuration)

```
# cd /var/www  
# mv * /mnt
```

5. Remount /dev/sdb1 at /var/www

```
# cd /root  
# umount /mnt  
# mount /dev/sdb1 /var/www
```

6. Note: there is no need to add this partition to /etc/fstab. Lifekeeper will take care of mounting this automatically.

---

---

<b>Result</b>	<pre>[root@LinuxPrimary ~]# df /var/www Filesystem      1K-blocks  Used Available Use% Mounted on /dev/sdb1       253855    11083  229666  5% /var/www [root@LinuxPrimary ~]# ls /var/www cgi-bin error html icons lost+found manual usage</pre>
---------------	--

---

---

## Secondary Server

7. On your Secondary server, only perform Step #1 above, where you partition the disk. The size of the Target volume needs to be the same size, or greater, than our Source volume.

## Configure iSCSI initiator, discover and login to iSCSI target

This Evaluation guide will not cover how to setup an iSCSI Target Server. It is assumed that the shared storage already exists in your environment. If you don't have shared storage and wish to configure it, a simple solution is to use OpenFiler ( <http://www.openfiler.com/> ), an Open Source storage management appliance, which can be run on physical hardware or as a virtual machine.

On both Primary and Secondary servers, perform the following functions:

1. If not already installed, ensure that the **iscsi-initiator-utils** rpm package is installed:

```
# yum install iscsi-initiator-utils
```

2. Start the iscsid service and enable it to automatically start when the system boots

```
# service iscsid start  
# chkconfig iscsid on
```

3. Configure the iscsi service to automatically start, which logs into iSCSI targets needed at system start up.

```
# chkconfig iscsi on
```

4. Use the iscsiadm command to discover all available targets on the network storage server (OpenFiler)

```
# iscsiadm -m discovery -t sendtargets -p <name or IP of iSCSI server>
```

---

---

```
[root@LinuxPrimary init.d]# iscsiadm -m discovery -t sendtargets -p 192.168.198.201
```

**Example** `iqn.2006-01.com.openfiler:tsn.mysql`

---

---

5. Manually Login to the iSCSI Target

```
# iscsiadm -m node -T iqn.2006-01.com.openfiler:tsn.mysql -p 192.168.198.201 --login
```

6. Configure Automatic Login

```
# iscsiadm -m node -T iqn.2006-01.com.openfiler:tsn.mysql -p 192.168.198.201 --op  
update -n node.startup -v automatic
```

7. Use the “fdisk” command to format your iSCSI LUN, if needed

```
# fdisk /dev/sdc
```

8. Create a filesystem on your new iSCSI LUN Partition, sdc1

```
# mkfs.ext3 /dev/sdc1
```

9. Mount your iSCSI LUN at /var/lib/mysql (assuming a default mysql configuration). If data already exists in this directory, make sure to move it into the shared iSCSI LUN

```
# mount /dev/sdc1 /var/lib/mysql
```

10. At this point you now have a local partition, /dev/sdb1 mounted at /var/www and an iSCSI shared LUN, /dev/sdc1, mounted at /var/lib/mysql. Our disk layout now look as follows (example):

---

---

**Example** `[root@LinuxPrimary mysql]# df`

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
------------	-----------	------	-----------	------	------------

---

---

---

---

/dev/sda2	25967432	3683016	1976400	66%	/
/dev/sda1	101086	24659	71208	26%	/boot
tmpfs	517552	0	517552	0%	/dev/shm
<b>/dev/sdb1</b>	<b>253855</b>	<b>11132</b>	<b>229617</b>	<b>5%</b>	<b>/var/www</b>
<b>/dev/sdc1</b>	<b>966644</b>	<b>38944</b>	<b>878596</b>	<b>5%</b>	<b>/var/lib/mysql</b>

---

---

## Phase 3 – Install and Configure Apache and PHP

### Before You Begin

Ensure the following:

1. If you are not familiar with installing and configuring Apache, please refer to the Apache documentation
2. Apache should not be running at the time you attempt to protect it with LifeKeeper.

---

---

#### **Important**

If Apache and PHP are already installed, skip to step #2 in this section below. You will need to verify the Apache configuration and/or relocate the location of the website data.

---

---

### Primary Server

1. Install Apache.
  - a. This example assumes that you are running a RHEL or CentOS 5.X based Linux distribution. For other Linux distros, please refer to your apache documentation for syntax differences.
  - b. If the Apache and PHP packages are not already installed, from the command line, , run: “yum install httpd php”
  - c. A number of dependencies will most likely be discovered. Install those as well.
2. Apache should be configured so that it will not automatically start when the server boots. LifeKeeper will control the start/stop of the webserver once its protected
  - a. Check the status of the webserver: “/etc/init.d/httpd status
  - b. If running, please stop of: /etc/init.d/httpd stop
  - c. Disable automatic startup: chkconfig httpd off
3. Apache Configuration. In this example, we will assume default Apache settings and directory locations, specifically:

- a. Validate the following settings in `/etc/httpd/conf/httpd.conf`:
    - i. `ServerRoot "/etc/httpd"`
    - ii. `DocumentRoot "/var/www/html"`
    - iii. `Listen 192.168.197.150:80` (note 192.168.197.150 is the Apache Switchable IP we will configure later in the LifeKeeper user interface)
  - b. Edit the Listen parameter in the `/etc/httpd/conf.d/ssl.conf` configuration file
    - i. `Listen 192.168.197.150:443`
4. Create a sample Index page:
- a. `vi /var/www/html/index.php`
  - b. Insert the following single line of code into this file:  
`<? phpinfo(); ?>`

## Secondary Server

1. Install Apache/PHP exactly as you did on the primary server, making all of the same configuration changes.
2. There is no need to perform Step #4 in which you create a sample index page. All data in `/var/www` will be replicated from LinuxPrimary to LinuxSecondary
3. Stop all Apache services on the secondary server.

## Phase 4 – Install, Configure, and Start MySQL

### Primary Server

On your Primary server, perform the following actions:

1. Install both the "mysql" and "mysql-server" rpm packages if they do not exist on your system. Apply any required dependencies as well  
`# yum install mysql mysql-server`
2. Verify that your Shared iSCSI LUN is still mounted at `/var/lib/mysql` via the "df" command
3. If this is a fresh MySQL install, initialize a sample MySQL database:  
`# /usr/bin/mysql_install_db --datadir="/var/lib/mysql" --user=mysql`
4. Ensure that all files in your MySQL data directory (`/var/lib/mysql`) have correct permissions and ownership  
`# chown -R mysql:mysql /var/lib/mysql`  
`# chmod 755 /var/lib/mysql`

5. Finally, manually start the MySQL daemon from the command line. Note: **Do Not** start it via the “service” command, or the /etc/init.d/ scripts  
# mysqld\_safe --user=root --socket=/var/lib/mysql/mysql.sock --port=3306 --datadir=/var/lib/mysql --log &

6. Verify MySQL is running by connecting with the mysql client:

```
[root@LinuxPrimary mysql]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.0.77-log Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> exit
Bye
[root@LinuxPrimary mysql]#
```

7. Update the root password for your mysql configuration. In this example we set the MySQL root password to “SteelEye”

```
# echo "update user set Password=PASSWORD('SteelEye') where User='root'; flush
privileges" | mysql mysql
```

8. Verify your new password:

```
# mysql mysql -u root -p
(Enter “SteelEye” as the password)
#exit
```

9. Create a MySQL configuration file. We will place this in the same shared directory (/var/lib/mysql/my.cnf)

```
# vi /var/lib/mysql/my.cnf
```

---

---

**Example**

```
# cat /var/lib/mysql/my.cnf

[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
pid-file=/var/lib/mysql/mysqld.pid
user=root
port=3306
# Default to using old password format for compatibility with mysql 3.x
# clients (those using the mysqlclient10 compatibility package).
old_passwords=1
```

---

---

```
# Disabling symbolic-links is recommended to prevent assorted security risks;
# to do so, uncomment this line:
# symbolic-links=0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

[client]
user=root
password=SteelEye
```

---

---

10. Delete the original MySQL configuration file, located in /etc  
# rm /etc/my.cnf

## Secondary Server

On your Secondary Server:

1. Install both the “mysql” and “mysql-server” rpm packages if they do not exist on your system.  
Apply any required dependencies as well  
# yum install mysql mysql-server
2. Ensure that all files in your MySQL data directory (/var/lib/mysql) have correct permissions and ownership  
# chown -R mysql:mysql /var/lib/mysql  
# chmod 755 /var/lib/mysql
3. There is no need to perform any of the additional steps taken on the Primary Server

## Phase 5 – Install SteelEye Protection Suite for Linux

For the ease of installation, SIOS has provided the SteelEye Protection Suite for Linux with an installation script. After the installation script has been run, the optional Application Recovery Kits (ARKs) should be installed. ARKs are provided as standard RPMs. Instructions for how to obtain the software is described in an email with your evaluation license keys. The software will be installed to the following locations:

SPS for Linux Component	Install Location
LifeKeeper Software	/opt/LifeKeeper
LifeKeeper Config File	/etc/default/LifeKeeper

Perform the following actions on **both** Primary and Secondary server.

### Download Software

1. Open the SteelEye Protection Suite evaluation email you received from SIOS.
2. Download the SteelEye Protection Suite Software from the link provided in your email. It is generally easiest to use “wget” to recursively download all files. Example:
  - a. # cd /root
  - b. # wget -r <URL>
  - c. After successful download you will have downloaded contents similar to the follow directory listing:

```
[root@LinuxPrimary ~]# ls -l <directory>
total 63680
-rw-r--r-- 1 root root 23163 May 30 14:03 EULA.pdf
-rw-r--r-- 1 root root 536 May 30 14:03 readme.txt
-rw-r--r-- 1 root root 65179648 May 30 14:03 sps.img
```
3. Download your Evaluation license key from the link specified in your evaluation email. Save the license file to an easy to remember location on both servers.

### Run the SteelEye Protection Suite Installer Script

1. Loopback mount the sps.img file previously downloaded, which is an ISO9660 image file. Run the “setup” script inside:

```
# mount -o loop sps.img /mnt
# cd /mnt
# ./setup
```

```
Welcome to the SteelEye Protection Suite for Linux v8.0.0-5104 Setup

A log of this setup will be written to: /var/log/LK_install.log

This script will set up and install SPS for Linux on your system.

Do you wish to continue (answering "no" will abort setup) (y/n) [y] ? █
```

2. During this procedure, you will hit Enter in most cases to accept default values and continue to the next screen. Note the following exceptions:
  - a. On the screen titled "High AvailabilityNFS" you may select "n" as in this particular eval guide we will not be creating a highly available NFS server cluster configuration.
  - b. If you have plans to create a highly available NFS service, adjust your response accordingly.
3. Towards the end of the setup script, you can choose to install a trial license key now, or later. We will install the license key in the next step, so you can safely select "n" at this point
4. In the final screen of the "setup" select the ARKs you wish to install from the list displayed on the screen.
5. The following RPMs should be installed:
  - a. steeleye-lkAPA-<version>.noarch.rpm
  - b. steeleye-lkSQL-<version>.noarch.rpm
  - c. steeleye-lkDR-<version>.noarch.rpm
6. Un-mount the Distribution Enabling disk image:

```
# cd /root
# umount /mnt
```

## Install the Evaluation License Keys

The last phase of the setup process installs the licensing keys. You must install the evaluation license key file (".lic") that you downloaded with your evaluation software before starting the SteelEye Protection Suite for Linux.

1. To install your trial license key, run the "lkkeyins" command on both Primary and Secondary Server. This command is located at /opt/LifeKeeper/bin/lkkeyins . Example:

```
# /opt/LifeKeeper/bin/lkkeyins <path_to_license/<filename>.lic
```

2. Validate your license keys were installed via the /opt/LifeKeeper/bin/lklicmgr command

```
# /opt/LifeKeeper/bin/lklicmgr
License File: 20101230.lic
Product          Type  Expiry          Other
LifeKeeper for Linux      Eval  27 Mar 2011 (87 days)
Apache Recovery Kit      Eval  27 Mar 2011 (87 days)
SteelEye Data Replication ARK Eval  27 Mar 2011 (87 days)
MySQL Recovery Kit       Eval  27 Mar 2011 (87 days)
...
```

## Start the SteelEye Protection Suite for Linux

1. Start:

```
# /opt/LifeKeeper/bin/lkstart
```

## Phase 6 - Configure the Cluster

### Primary Server

Complete the following steps on the primary server to configure the cluster:

- Create TCP Communication (Comm) Path(s)
- Verify the Communication (Comm) Path(s)

Before you begin, SIOS recommends at least two TCP communications paths between each server within the cluster to each remote server for heartbeat redundancy.

---

---

<b>Important</b>	Supported configurations require that you define redundant comm paths, so that the failure of a single communication line will not cause a split brain where resource hierarchies may come in-service on multiple servers simultaneously.
------------------	---

---

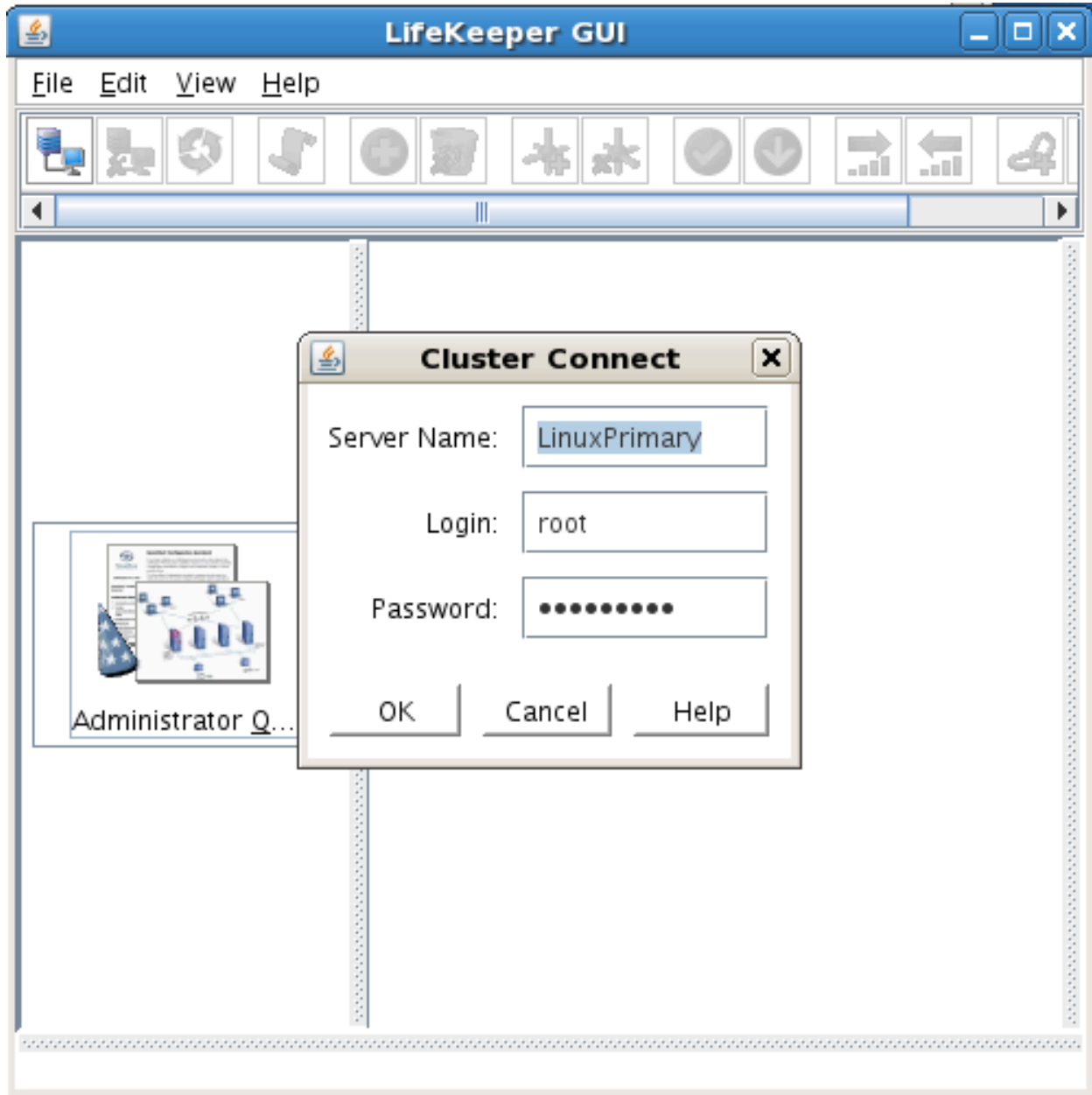
---

### Access the LifeKeeper GUI

The LifeKeeper Graphical User Interface (GUI) is a Java based application that can be run as a native Linux application, or as an applet within your Java-Enabled Web Browser.

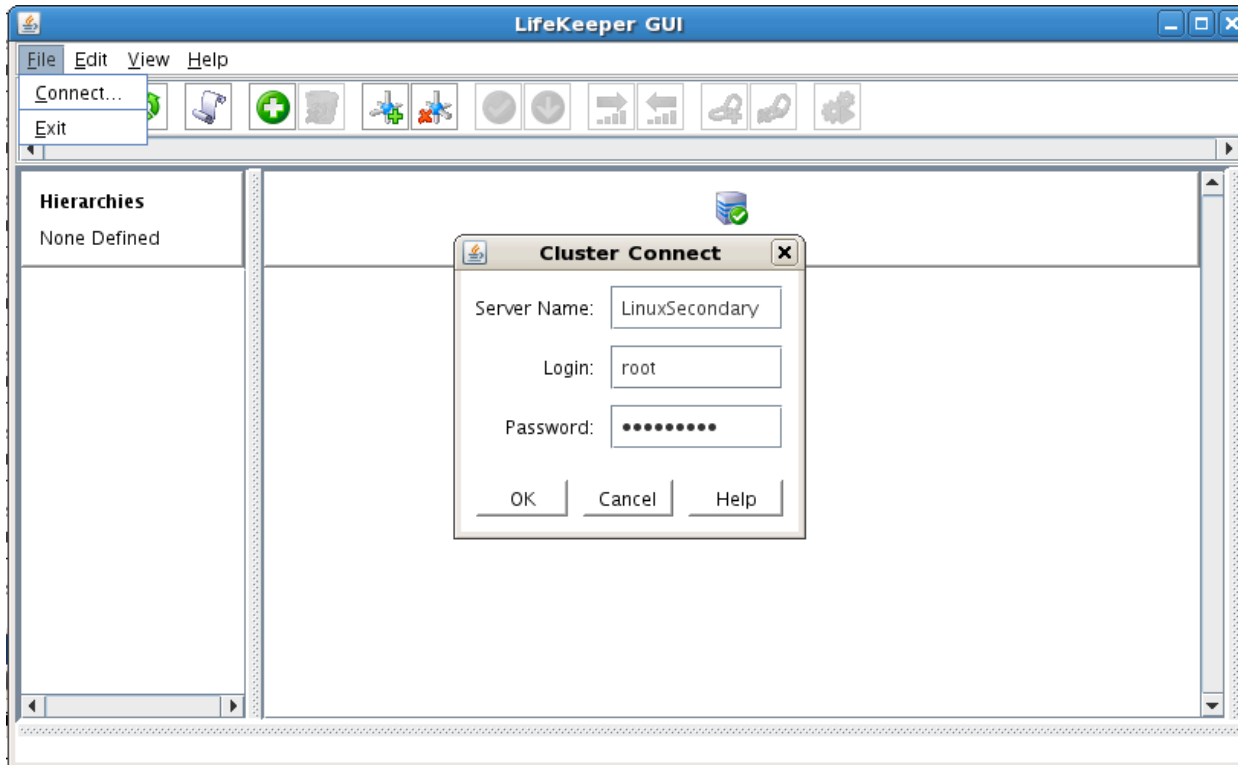
The LifeKeeper GUI is based on Java RMI with callbacks. Hostnames must be resolvable or you may receive a Java 115 or 116 error.

1. To start the LifeKeeper Linux GUI Application:
  - a. `/opt/LifeKeeper/bin/lkGUIapp &`
2. To Connect to the LifeKeeper GUI Applet from a Web Browser, go to:
  - a. <http://<hostname>:81>
3. Enter the name of the server you wish to connect to (this field will be populated with the name of the server you are on, if you are running the GUI from a server with LifeKeeper installed) along with your root credentials and click OK.

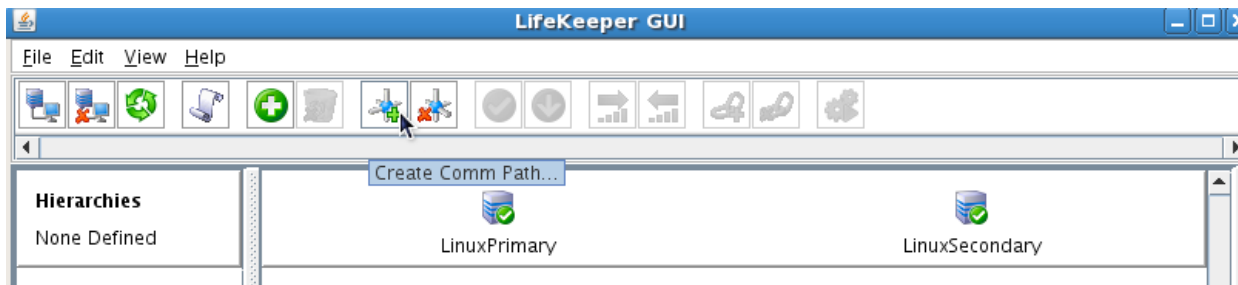


### Create Communication (Comm) Paths

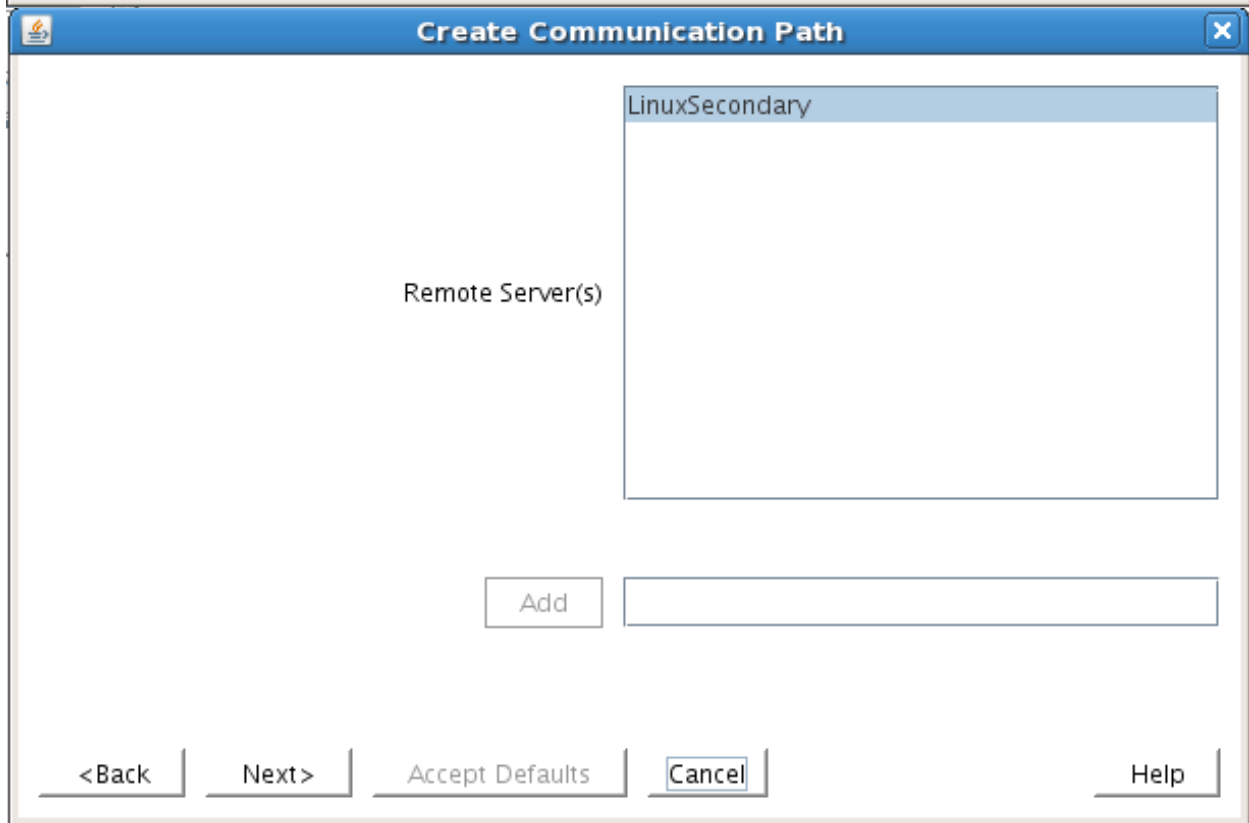
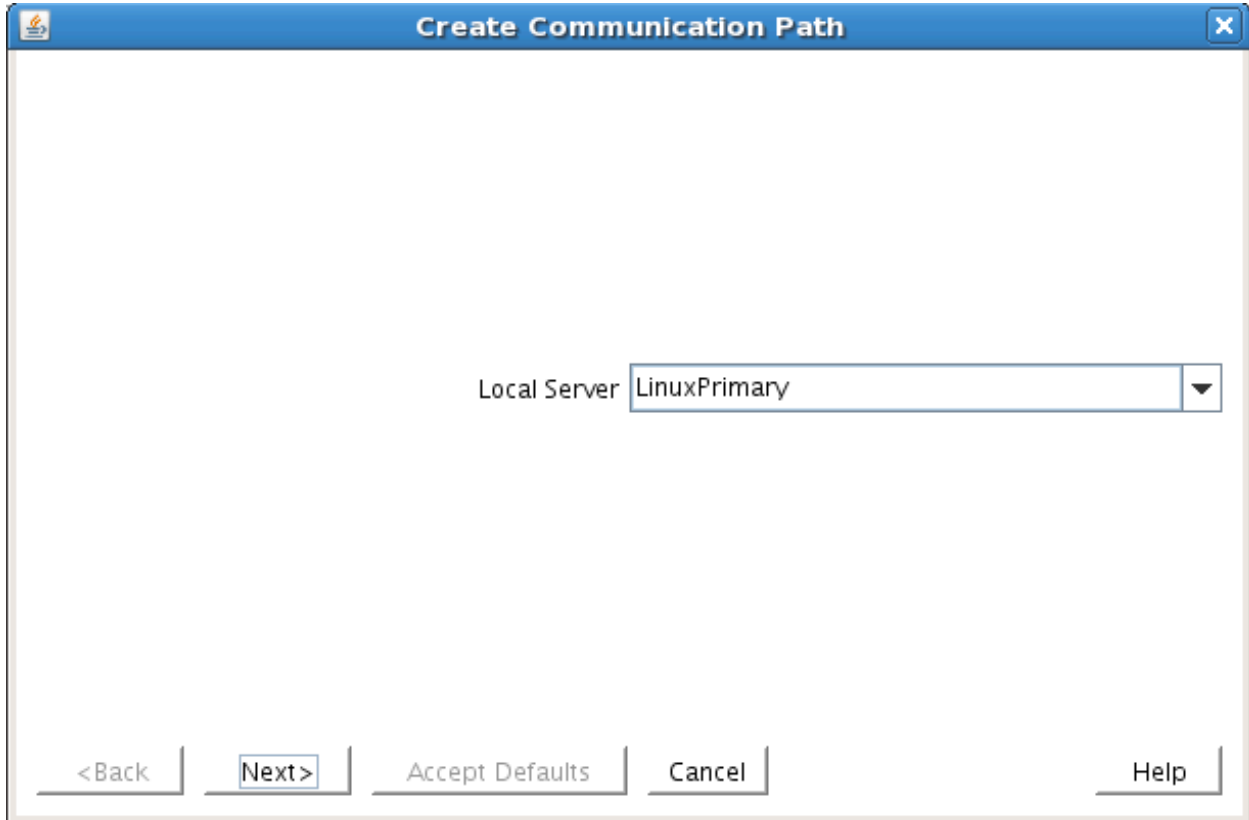
4. Within the LifeKeeper GUI, from the File menu, select Connect. Enter the name of your Secondary server, login and password when the Cluster Connect window displays.



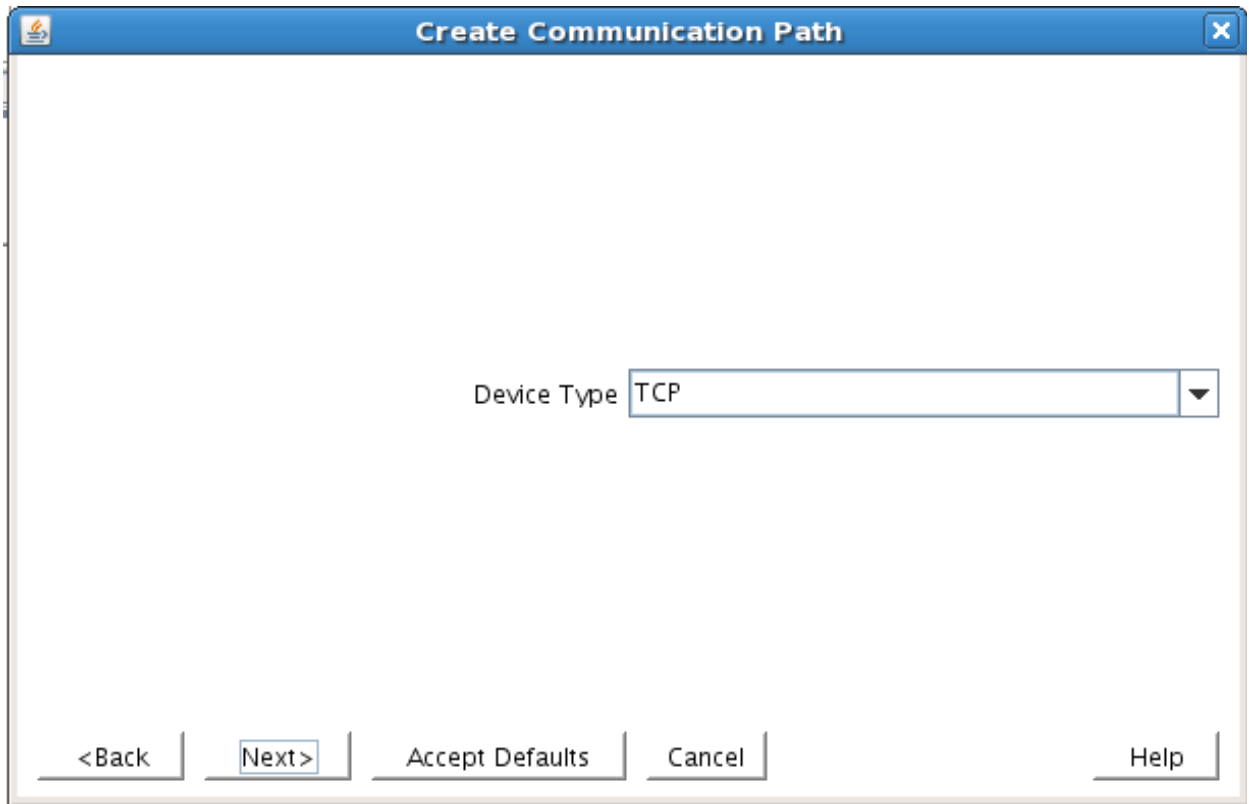
5. Within the LifeKeeper GUI, click the Create Comm Path button on the toolbar. You can also right click one of the servers and click Create Comm Path from the pop-up menu as well.



6. Select your Local and Remote Server(s) from the list box. If a server does not appear in the list box, you may enter it by typing its name and clicking the Add Server button. When using the Add Server procedure, you must make sure that the computer names for both network interfaces on the servers respond correctly when you ping them (from all of the partner server(s)) using the **ping -a IP ADDRESS** syntax. If they do not, this must be corrected prior to continuing. Click Next.

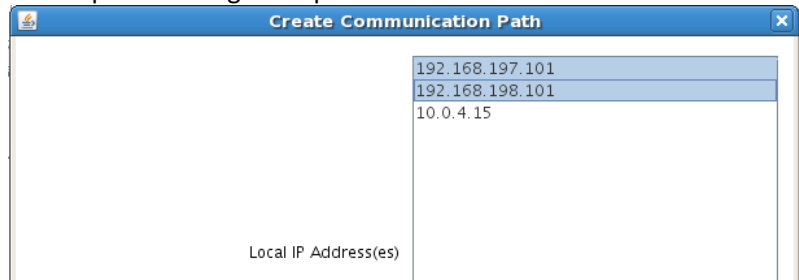


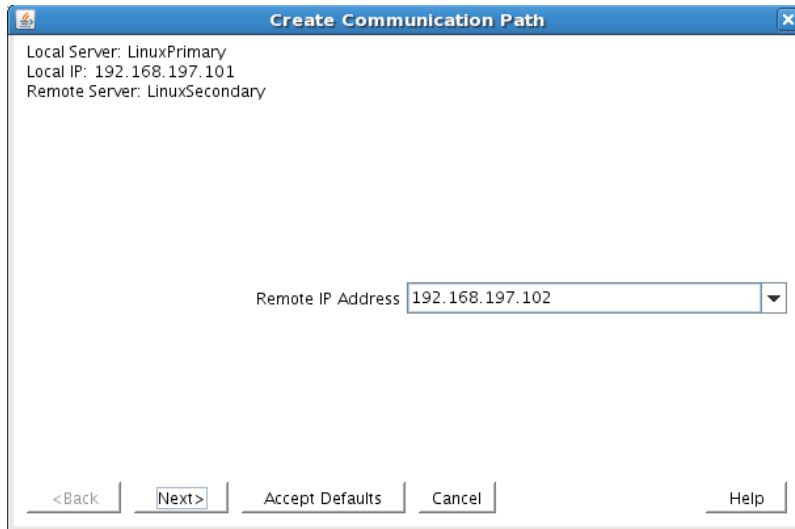
7. Select TCP for Device Type and Click Next.



8. Provide all the required information and click Next for the following series of dialog boxes. For each field in the dialog box you can click Help for further information or refer to the table below for an explanation or recommendation.

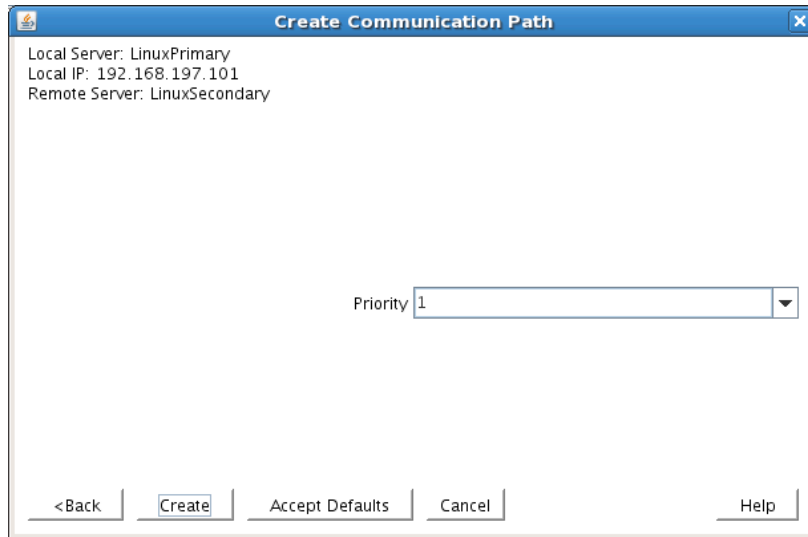
Field	Tips
<i>For TCP/IP Comm Path...</i>	
Local IP Address	Choose the IP address to be used by the local server for this comm path. Select both interfaces so that the wizard creates multiple comm. paths during this operation
Remote IP Address	Choose the IP address to be used by the remote server for this comm path



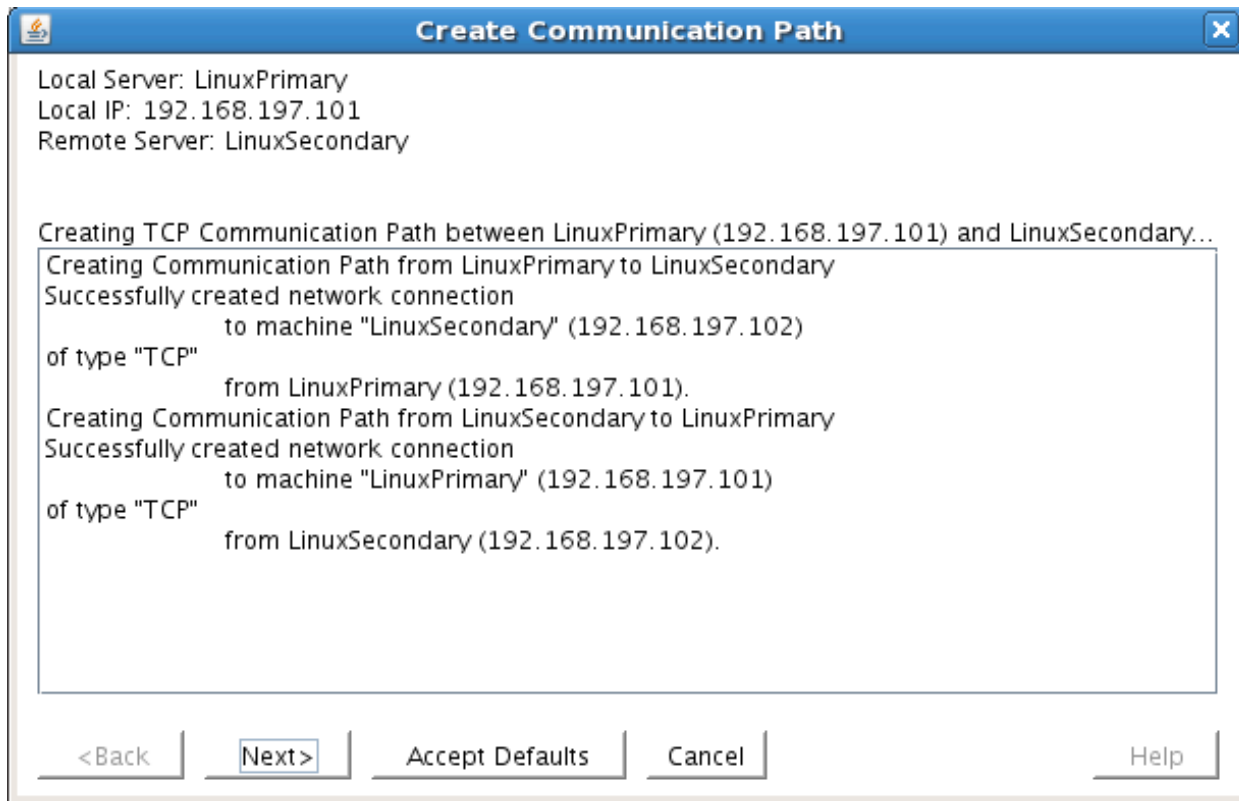


Enter the priority for the comm path on the local server. The priority will be used to determine the order that the comm paths between two servers will be used. Priority 1 is the highest; 99 is the lowest.

Priority



9. After entering data in all the required fields, select Create. A message will display indicating the network communication path is successfully created. Click Next.



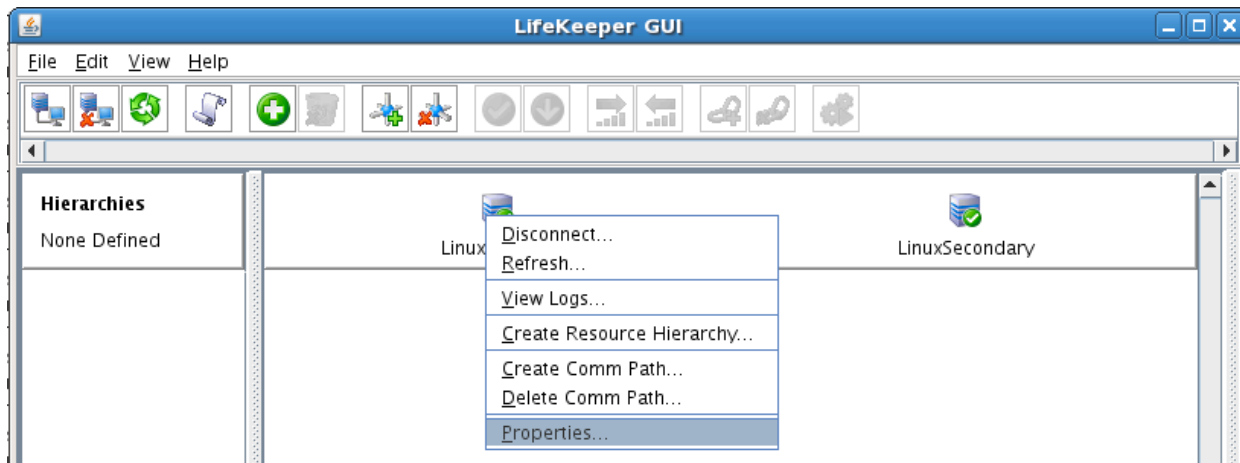
If you selected multiple Local IP Addresses or multiple Remote Servers and the Device Type was set to TCP, then the procedure will return you to the setup wizard the next Comm Path.

10. Select Done in the last dialog box.

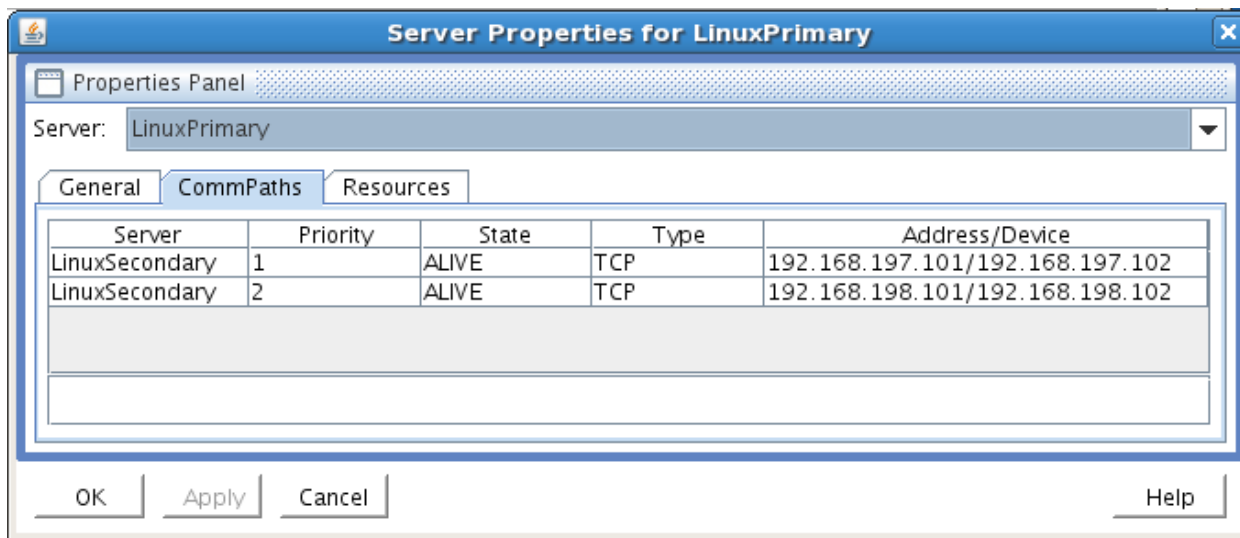
Repeat this process until you have defined all the communication paths you plan to use. SteelEye strongly recommends that you define at least two communication paths for redundancy.

### Verify the Communications Paths

1. Verify that the communications paths are configured properly by viewing the Server Properties dialog box. From the LK GUI, select Edit, Server, Properties and then the Comm Paths tab.



2. Note the State displayed is ALIVE. You can also check the server icon in the right, main pane of the GUI. If only one comm path has been created, the server icon shows a yellow warning icon on the server icon, indicating that one comm. path is ALIVE, but there is no redundant comm path. The server icon will display a green heartbeat checkmark when there are at least two comm paths configured and ALIVE.

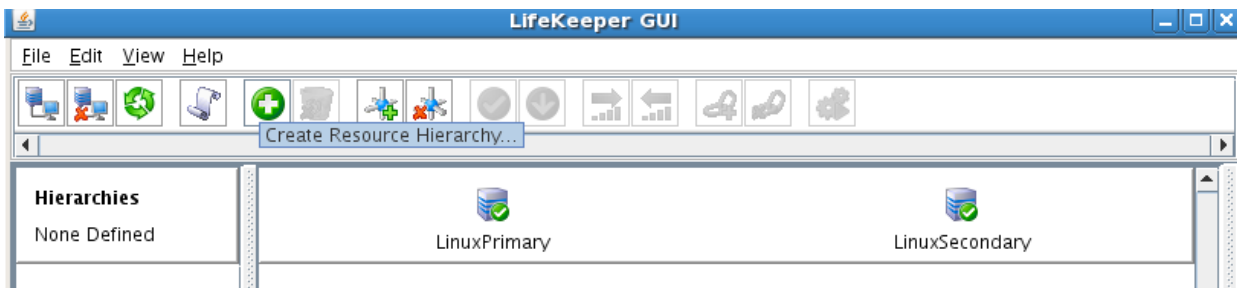


## Create the LifeKeeper Hierarchy

### Create and Extend an IP Resource

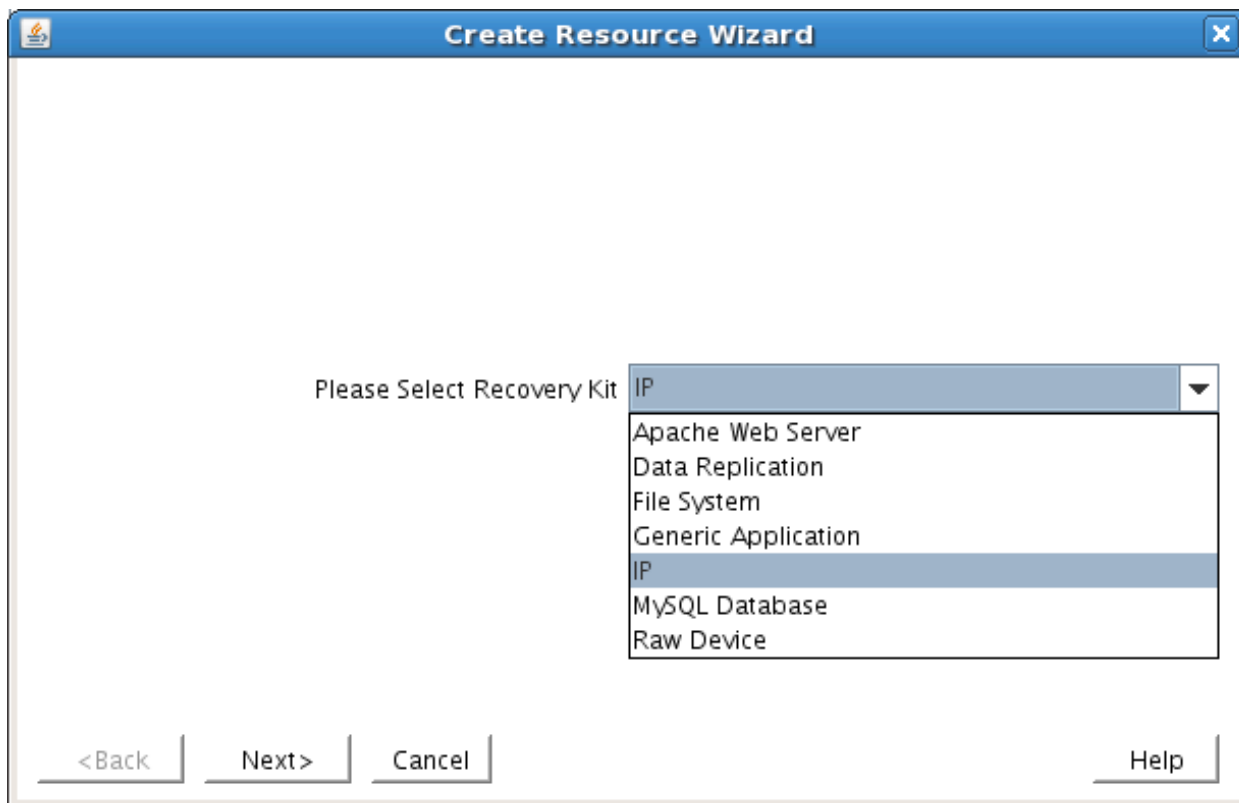
In LifeKeeper, create an IP resource and extend it to the secondary server by completing the following steps. This Virtual IP will have the ability to move between cluster nodes along the application that depends on it.

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.



The Create Resource Wizard dialog box will appear with a drop down list box displaying all recognized Recovery Kits installed within the cluster.

2. Select IP Address and click Next.



3. Enter the appropriate information for your configuration. The table below contains a list of the fields that display and additional information to assist you as you complete this procedure. Recommended values are also show below. You can also click the Help button for further

information. Press Next to continue after entering the required information.

## IP Creation Field Definitions

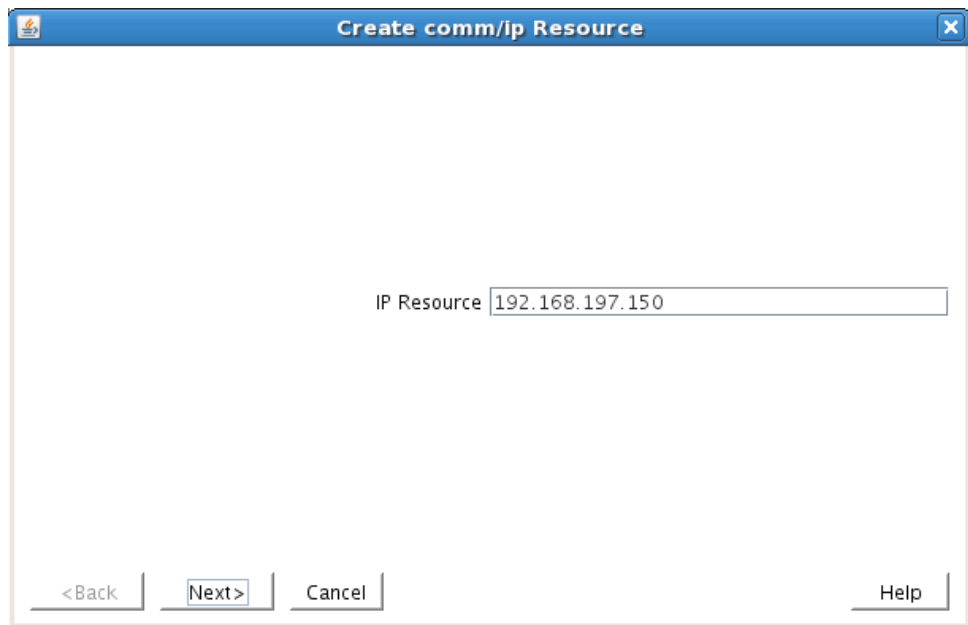
Field	Tips
Resource Type	Select IP Address as the resource type and click Next.
Switchback Type	Select Intelligent and click Next.
Server	Select the Server where the IP resource will be created. Select your Primary server and click Next.
IP Resource	Enter the virtual IP information and click Next

Example: **192.168.197.150**

**Note:** This is an IP address that is not currently in use anywhere on your network. This is the address that all clients will use to connect to the protected resources.

In this configuration example, we will be protecting two (2) virtual IPs.

First we will protect 192.168.197.150, which our Apache webserver will use.

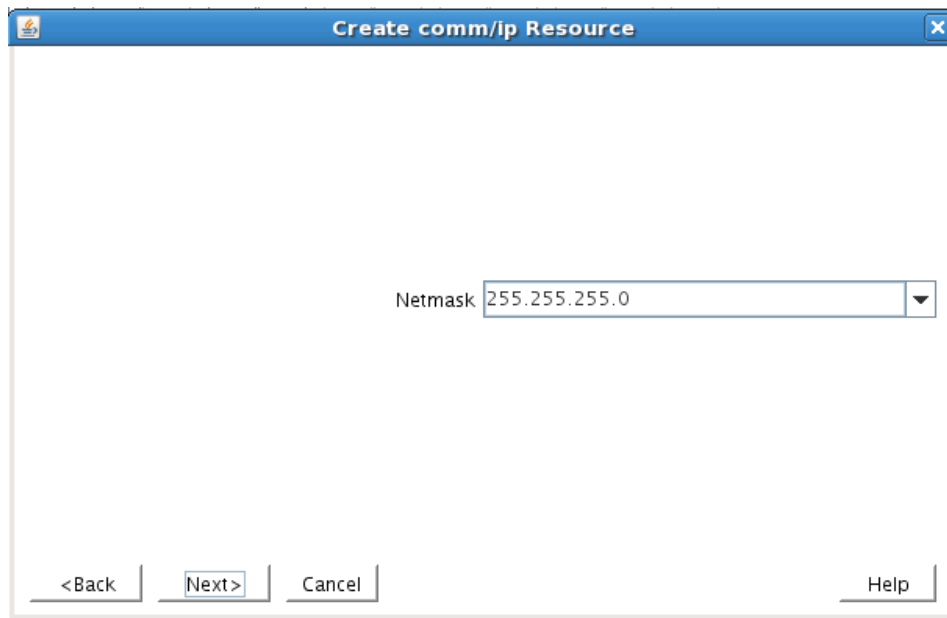


The second time through this wizard we will protect 192.168.197.151, which will be used by MySQL

**Netmask** The IP subnet mask that your TCP/IP resource will use on the target server. Any standard netmask for the class of the specific TCP/IP resource address is valid.

In our sample configuration 255.255.255.0 is used for a subnet mask on both networks.

**Note:** The subnet mask you choose, combined with the IP address, determines the subnet that will be used by the TCP/IP resource and should be consistent with the network configuration.



Network  
Connection

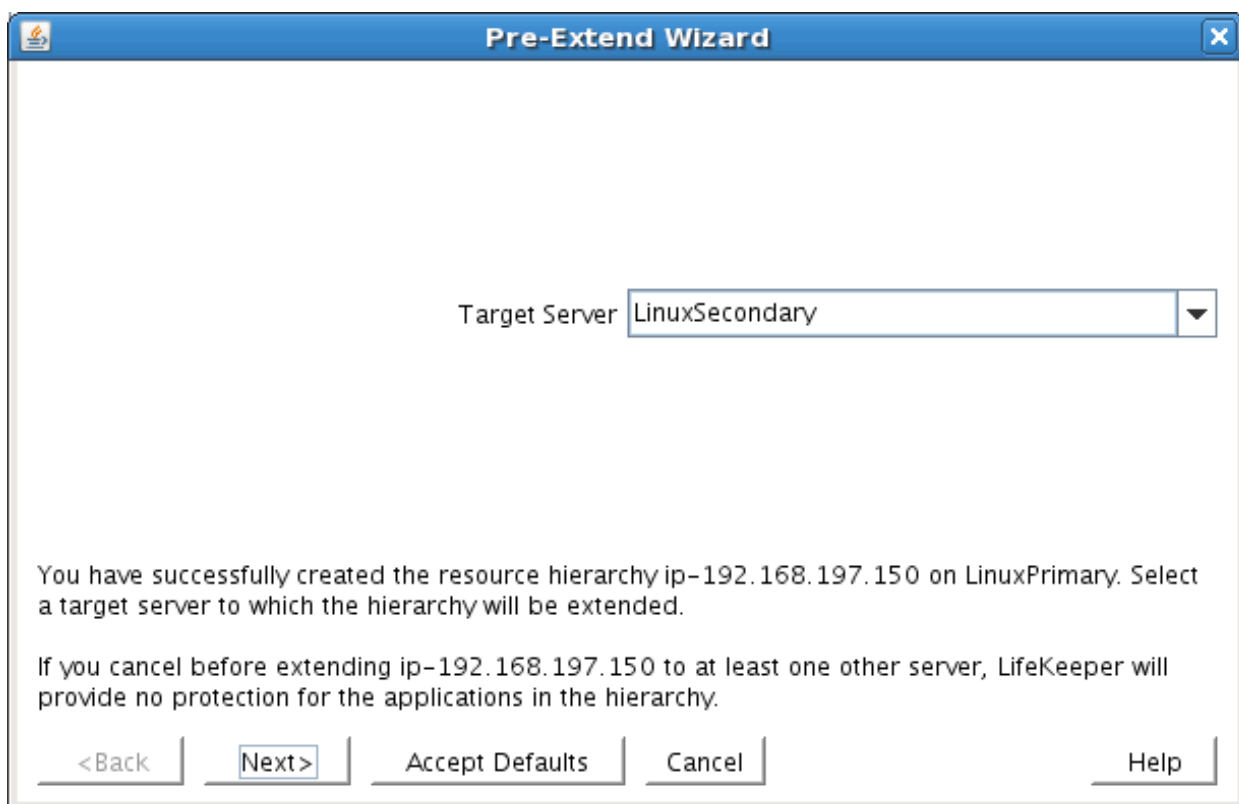
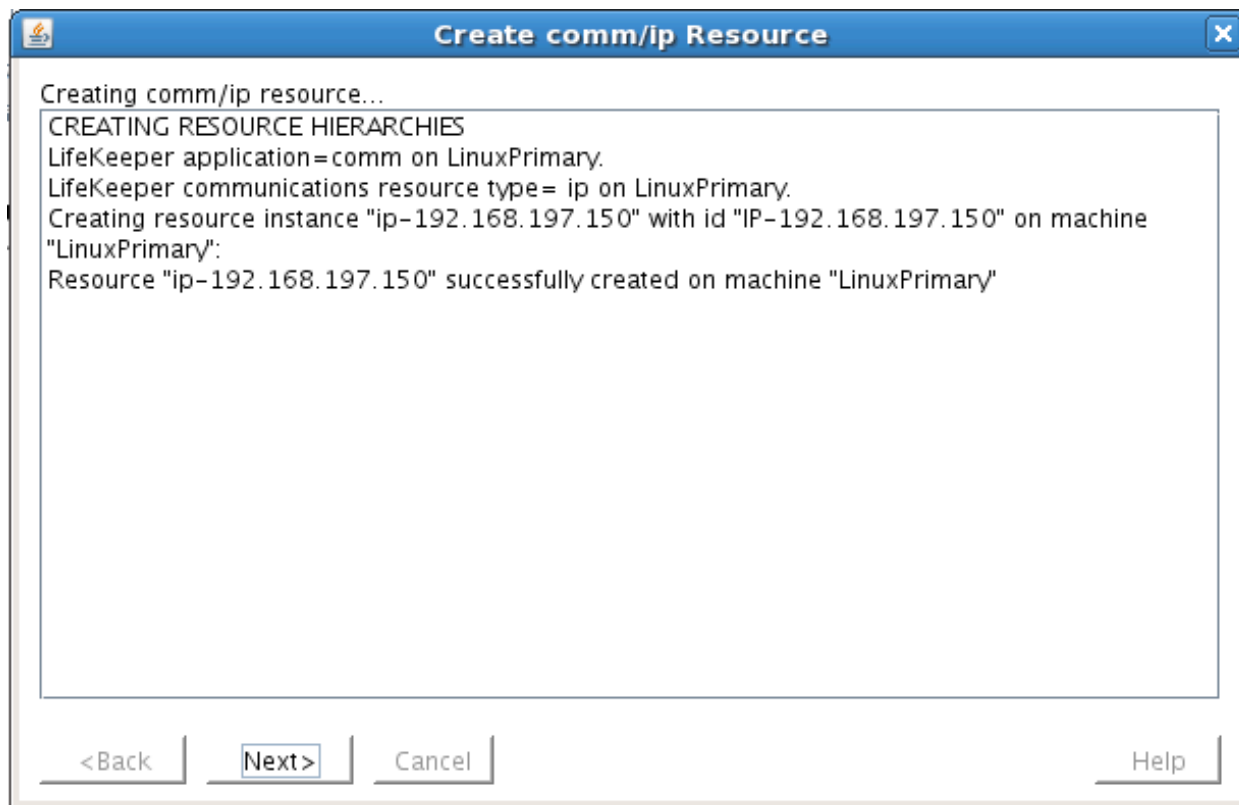
This is the physical Ethernet card that the IP address is interfacing with. Chose the network connection that will allow your virtual IP address to be routable. Select the correct NIC and click Next.

Backup Interface  
IP Resource Tag

Select "none" and click Next

Accept the default value and click Next. This value only affects how the IP is displayed in the GUI. The IP resource will be created on our Primary server.

4. LifeKeeper will create and validate your resource. After receiving the message that the resource has been created successfully, click Next when the following dialog box appears so that you can complete the process of Extending the IP Resource to our Secondary server, below.



Extending the IP resource will start automatically after you have finished creating an IP address resource if you clicked Next in the dialog box displayed above. You can also start this from an existing IP address resource by right clicking on the active resource and selecting Extend Resource Hierarchy.

Refer to the table below to complete the Extend IP Resource procedure.

Field	Recommended Entries or Notes
Switchback Type	Leave as "intelligent" and click Next
Template Priority	Leave as default (1)
Target Priority	Leave as default (10)
Network Interface	This is the physical Ethernet card that the IP address is interfacing with. Chose the network connection that will allow your virtual IP address to be routable. The correct physical NIC should be selected by default. Please verify and then click Next.
Backup Interface	Leave as "none"
IP Resource Tag	Leave as default.
Target Restore Mode	Select Enable and click Next.
Target Local Recovery	Select Yes to enable Local Recovery for the SQL resource on the Target server.
Backup Priority	Accept the default value.

- After receiving the message Hierarchy extend operations completed, click Finish and then click Done
- Your first IP resource (192.168.197.150) is now fully protected and has the ability to "float" between cluster nodes as needed. Looking at the LifeKeeper GUI you will notice that the IP resource is Active on the Primary cluster node and Standby on the Secondary cluster node

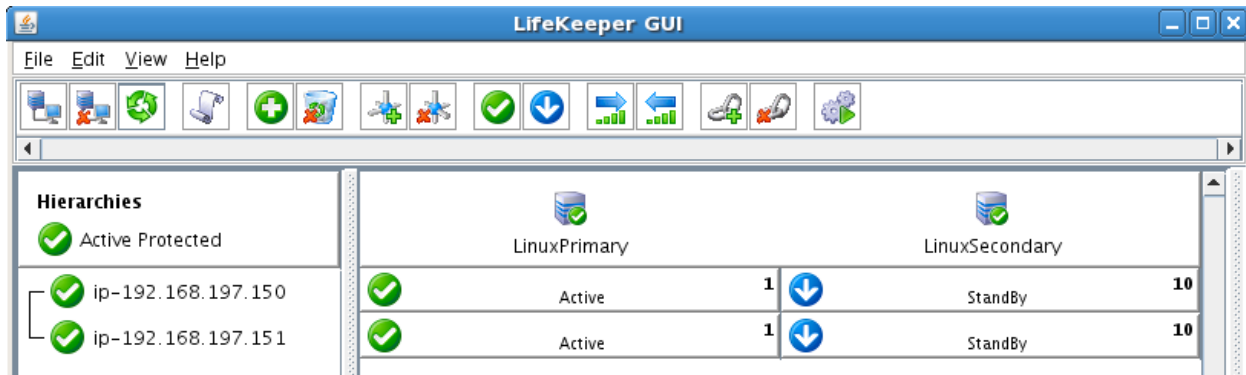


### Create a Second IP Resource

Repeat the procedure above to protect a 2<sup>nd</sup> IP resource.

This second time, protect 192.168.197.151, which is the IP address our MySQL database will later use.

As a result, your LifeKeeper GUI will display as follows, with both IP resources Active and protected on the Primary cluster node:



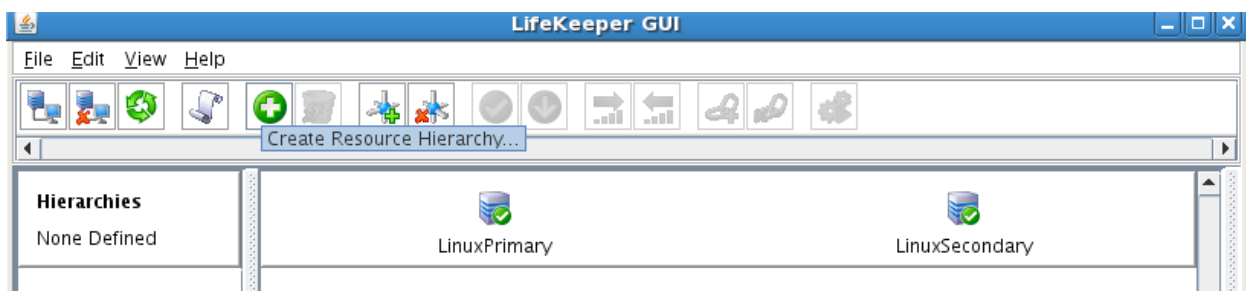
### Create a Mirror and Begin Data Replication

In this section we will setup and configure the Data Replication resource, which be used to synchronize our Apache Webserver's data between cluster nodes. The data we will replicate resides in the /var/www partition on our Primary cluster node

Please note:

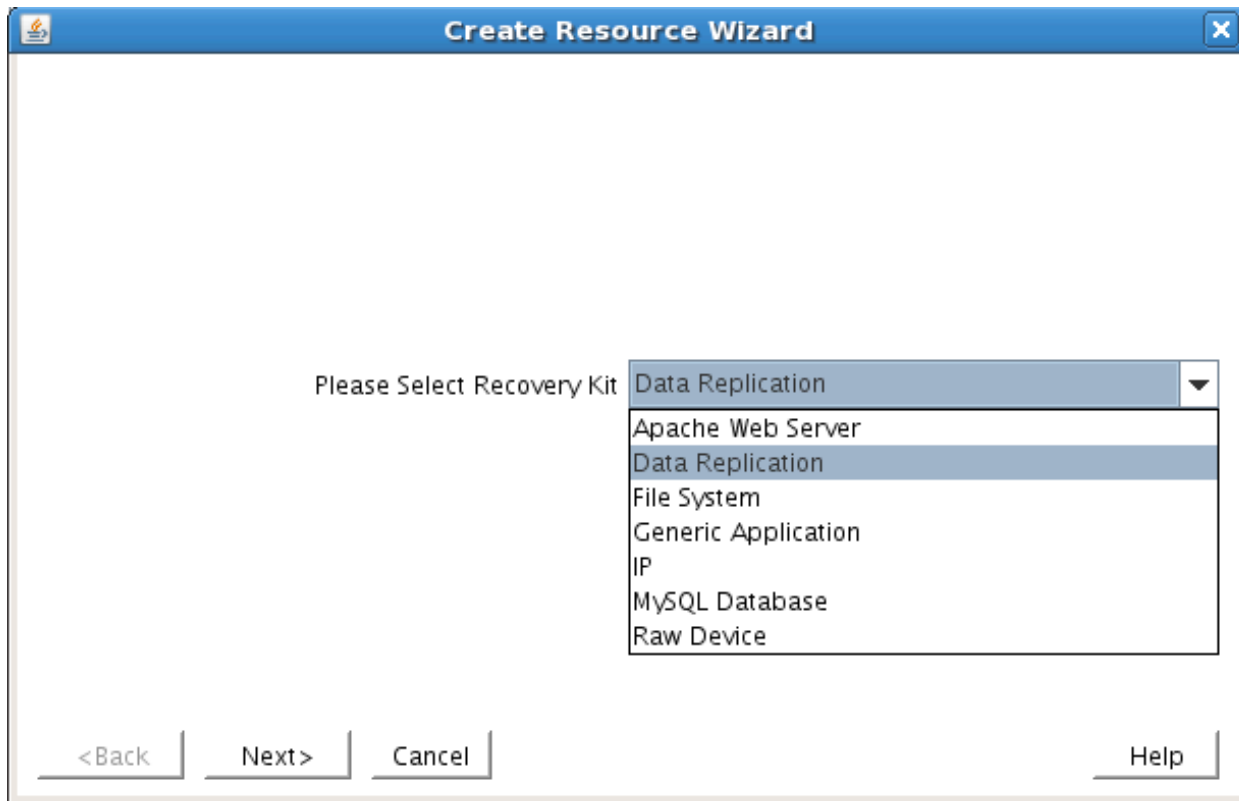
- The source volume to be replicated must be mounted on the Primary server
- The target volume, which will received replicated data, must **NOT** be mounted on the Secondary server.
- The target volume's size must equal to or larger than the size of its source volume.

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.



The Create Resource Wizard dialog box will appear with a drop down list box displaying all recognized Recovery Kits installed within the cluster.

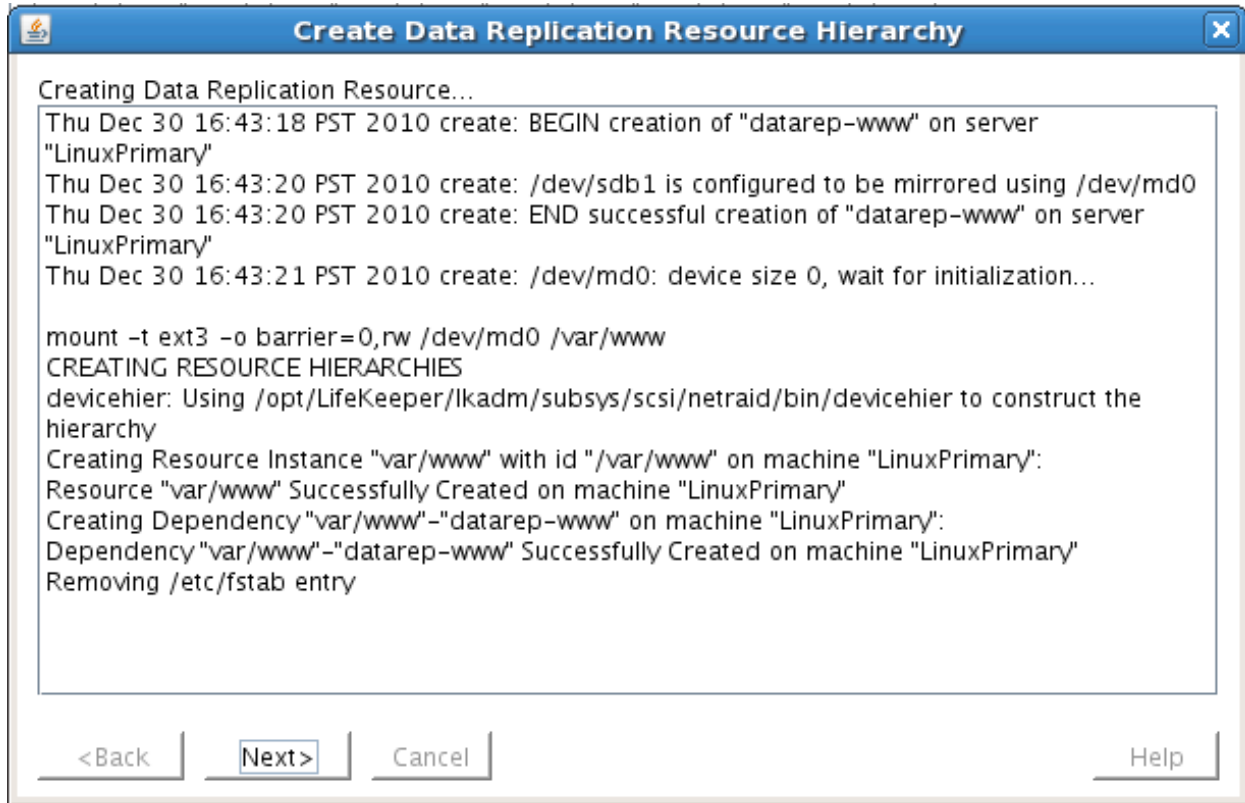
2. Select Data Replication and click Next.



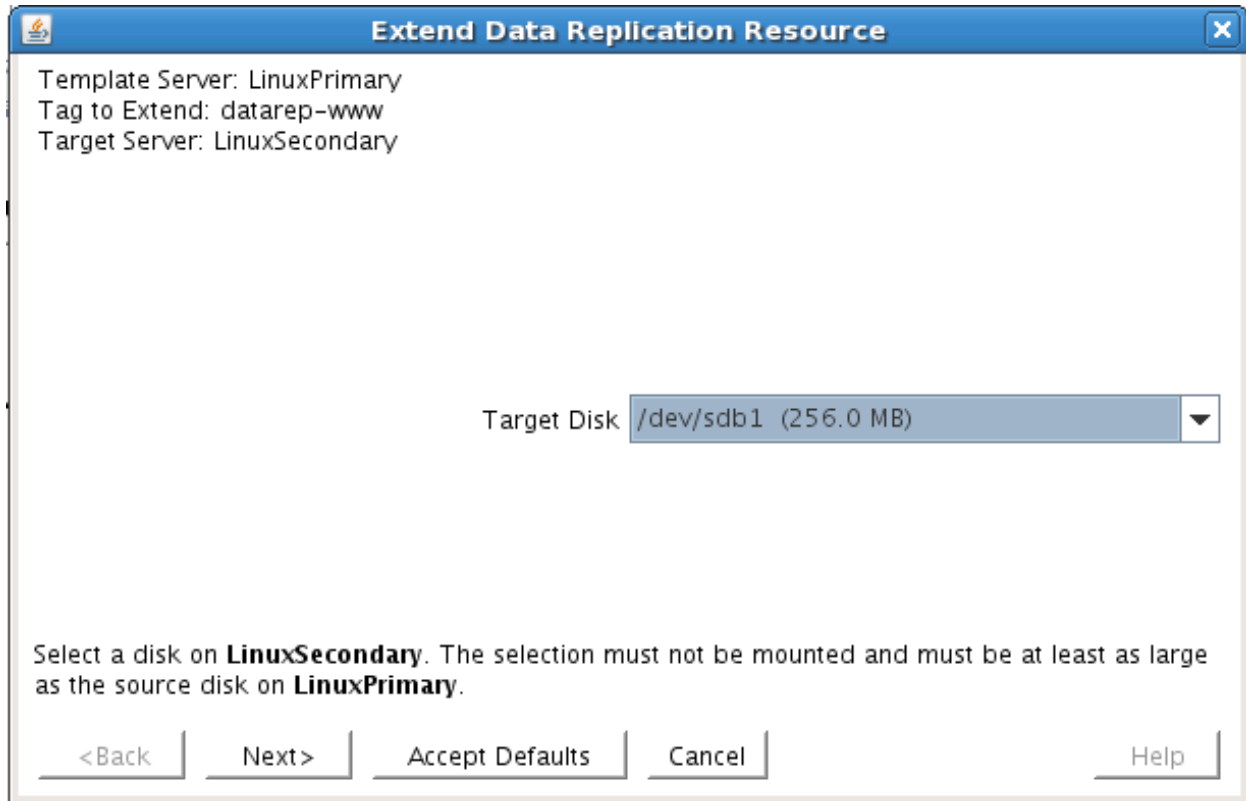
3. Follow the Data Replication wizard, and enter the following values:

Field	Recommended Entries or Notes
Switchback Type Server	Intelligent LinuxPrimary (Primary Cluster Node, i.e. Mirror Source)
Hierarchy Type Existing Mount Point	Select: <b>“Replicate Existing Filesystem”</b> At this step you will select the mounted partition to replicate. In our example, select <b>“/var/www”</b>
Data Replication Resource Tag	Leave as default
File System Resource Tag	Leave as default
Bitmap File	Leave as default
Enable Asynchronous Replication	Leave as default (Yes)

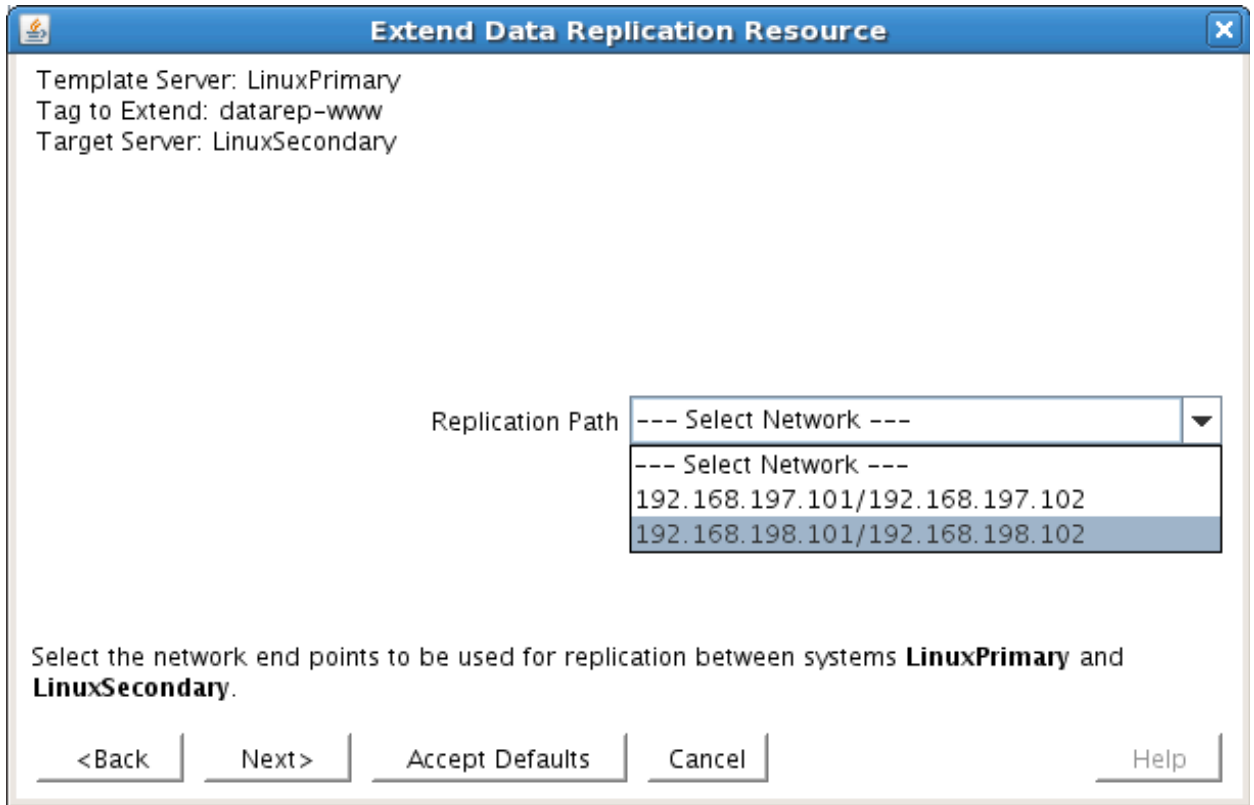
4. Click Next to begin creation of the Data Replication resource hierarchy. Status will be displayed in the GUI as follows:



5. Click Next to begin the process to Extend the Data Replication Resource. Select all default settings. When it asks for the target disk, select a free partition on your Target server which is the same size (or greater) than the Source Volume we are replicating. This partition should NOT be mounted on the Target system.



6. Continue through the wizard, and you will be prompted to select the network you would like replication to take place over. In general, it's a best practice to separate your user/application and your replication traffic. In our example setup we will replicate over our backend network, 192.168.198.X



7. Click Next and continue through the wizard. Once completed, your resource hierarchy will look as follows

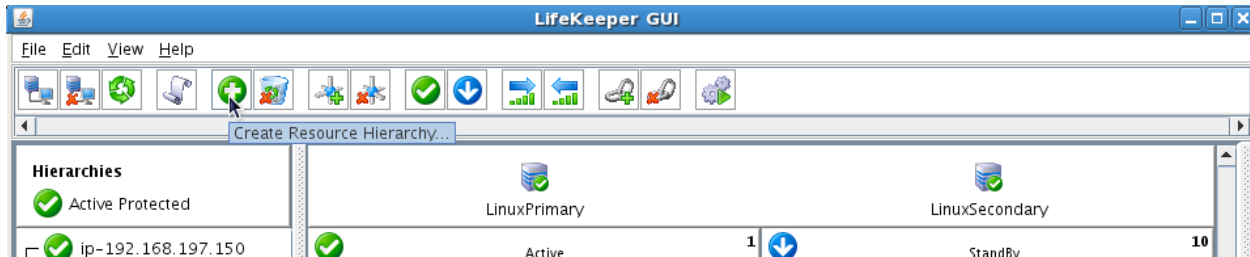


### Create the Apache Hierarchy

In this section we will create an Apache resource hierarchy on the primary server and extend it to the backup server. This step will create a dependency on the IP resource created in previous the step.

**Important** The Apache web server should not be running at this time.

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.



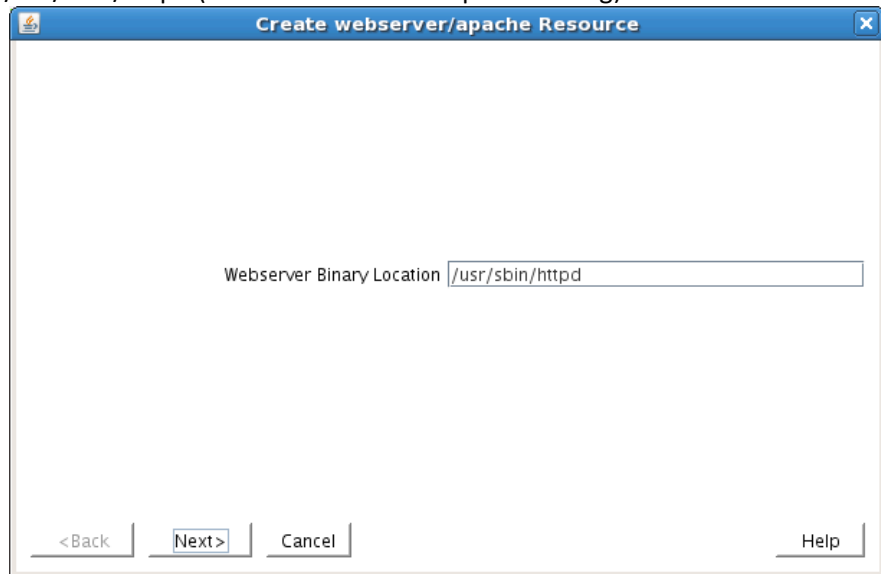
The Create Resource Wizard dialog box will appear with a drop down list box displaying all recognized Recovery Kits installed within the cluster.

2. Select Apache Web Server and click Next.
3. Proceed Through the Resource Creation wizard, providing the following values

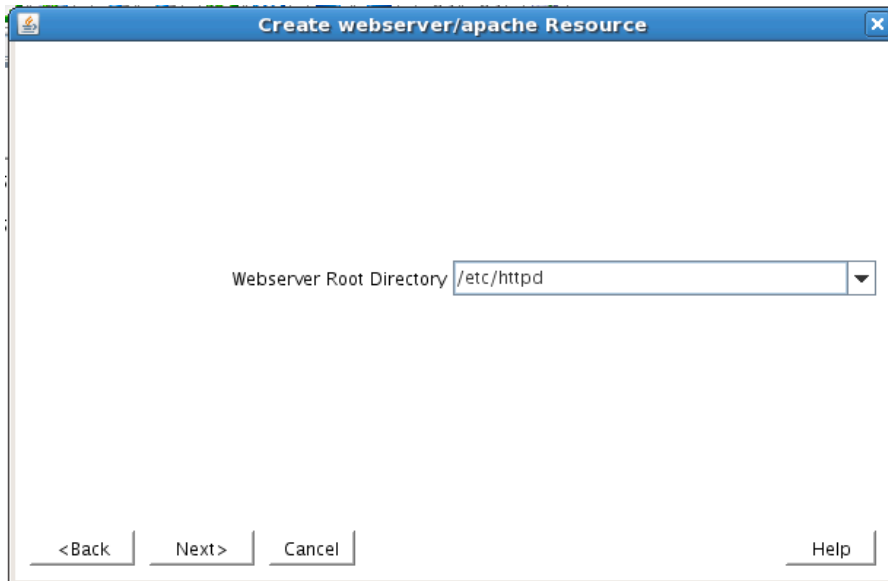
**Field**  
Switchback Type  
Server  
Webserver Binary  
Location

**Recommended Entries or Notes**

Intelligent  
LinuxPrimary (Primary Cluster Node, i.e. Mirror Source)  
/usr/sbin/httpd (assumes a standard apache config)

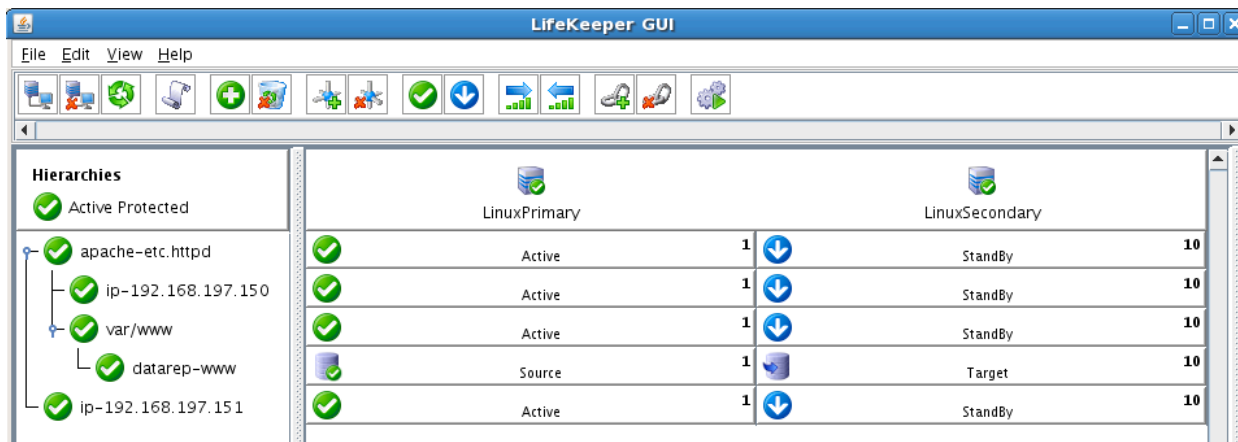


Webserver Root Directory /etc/httpd (assumes a standard apache config)



Root Tag                      Leave as default

4. Click "Create" to begin resource hierarchy creation on the primary server. Once complete, click "Next" to extend this resource to the secondary server.
5. During the Extend Resource wizard, leave all settings as default.
6. Note: during the resource creation process, LifeKeeper extracted the existing configuration of the existing Apache webservice, and identified that it depends on the IP resource (192.168.197.150) and the Data Replication resource (/var/www) that were created in previous steps. These resources now appear underneath the newly created Apache resource, to indicate the dependency relationship.



### Create the Shared Filesystem Resource Hierarchy

Create a Filesystem resource to protect the shared iSCSI filesystem and make it high available between cluster nodes. SPS for Linux leverages SCSI Persistent Group Reservations (PGR) to lock the LUN, ensuring that only the active cluster node for the storage resource can access it.

---



---

**Important**

At this point, the shared iSCSI LUN needs to already be mounted on the Primary Server. It should NOT be mounted on the Secondary Server. See section titled “Configure iSCSI initiator, discover and login to iSCSI target” above to review the steps involved.

---

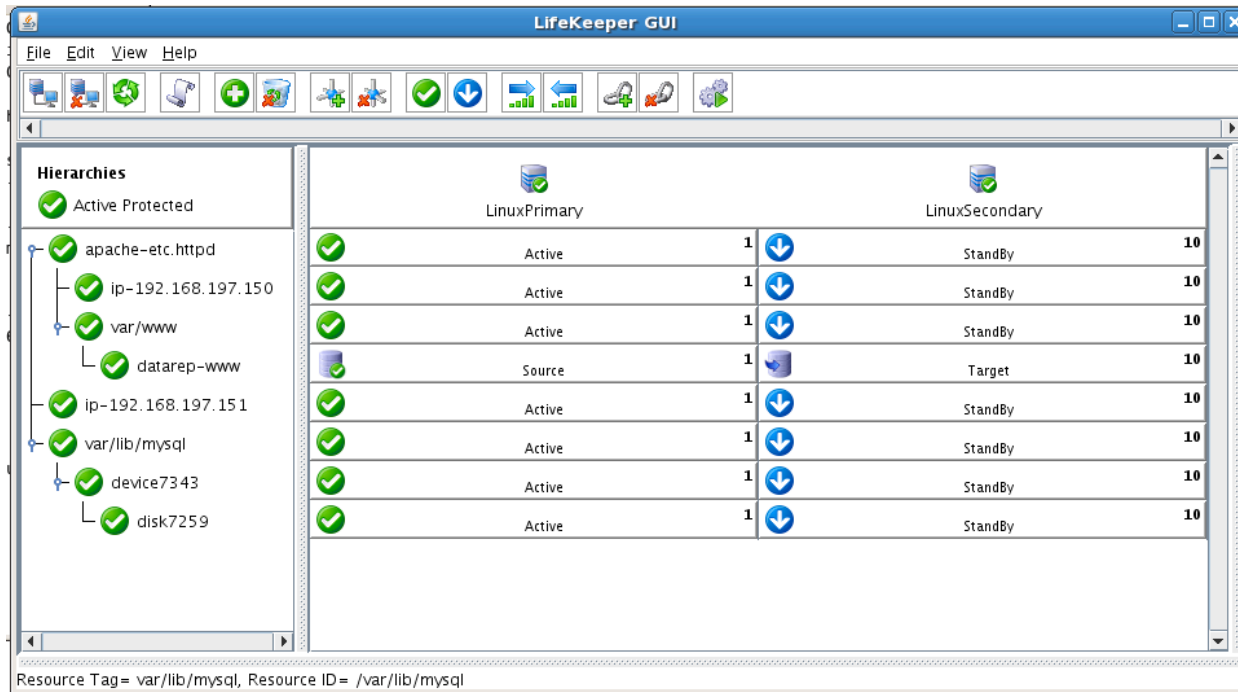


---

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.
2. Select File System and click Next.
3. Proceed Through the Resource Creation wizard, providing the following values

Field	Recommended Entries or Notes
Switchback Type	Intelligent
Server	LinuxPrimary (Primary Cluster Node)
Mount Point	Select /var/lib/mysql . Note that LifeKeeper scans the system for LUNS that are sharable between cluster nodes. The list of possible shared LUNS is presented automatically in this step of the wizard.

4. Select Create Instance to define this resource hierarchy on the Primary Server
5. Click Next to Extend the File System Resource to the Secondary Server
6. In the Extend Wizard, select “Accept Defaults”
7. As a result the File System resource is now protected on both cluster nodes. Click Finish to exit the Extend wizard.
8. Your resource hierarchy should look as follows:



### Create the MySQL Resource Hierarchy

Create a MySQL resource to protect the MySQL database and make it high available between cluster nodes.

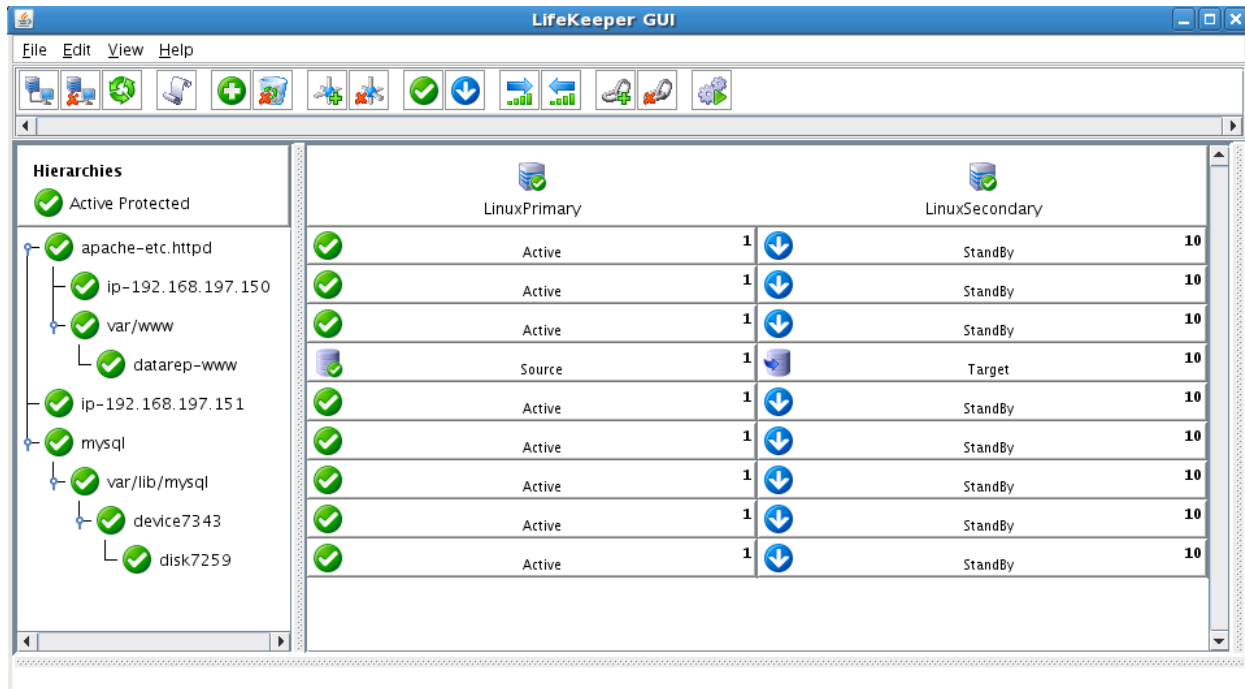
**Important**

At this point, MySQL needs to be already running on the Primary Server. It should NOT be running on the Secondary Server. See section titled “Install, Configure, and Start MySQL” above to review the process to configure and start MySQL as needed.

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.
2. Select **MySQL Database** and click Next.
3. Proceed Through the Resource Creation wizard, providing the following values

Field	Recommended Entries or Notes
Switchback Type	Intelligent
Server	LinuxPrimary (Primary Cluster Node)
Location of my.cnf	Enter “/var/lib/mysql”. Note that earlier in the MySQL configuration process we created a my.cnf file in this directory.
Location of MySQL executables	Leave as default (/usr/bin) since we are using a standard MySQL install/configuration in this example
Database tag	Leave as default

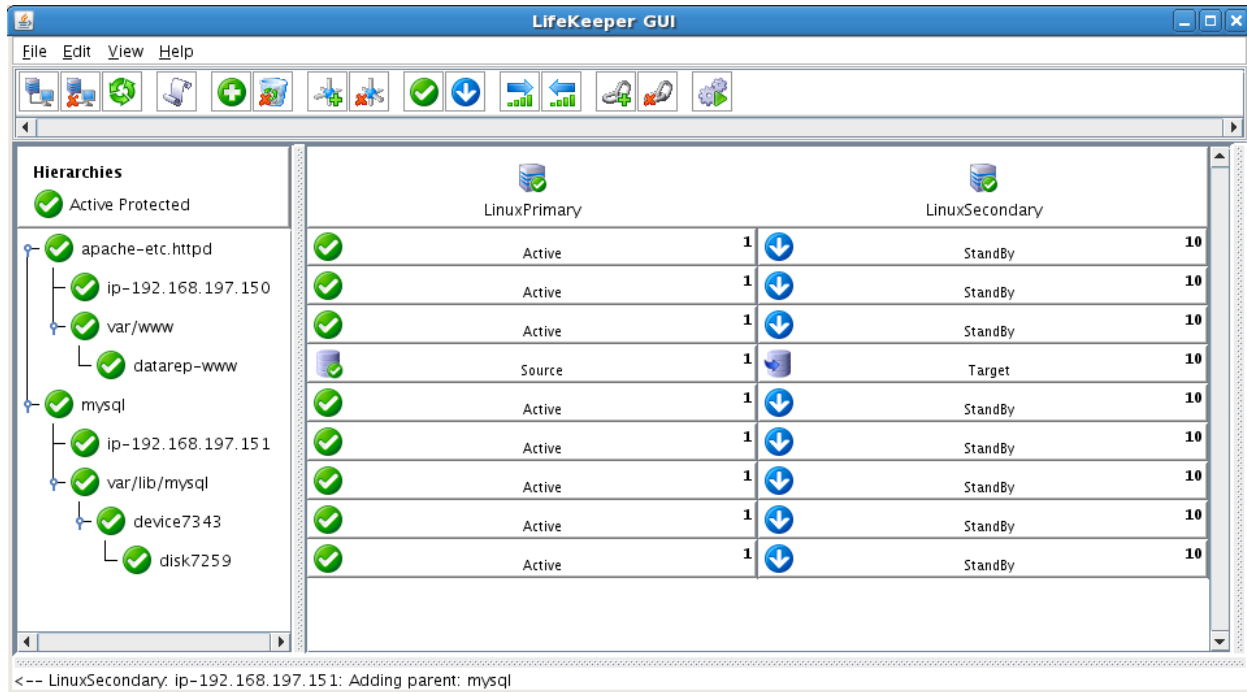
4. Select Create to define the MySQL resource hierarchy on the Primary Server
5. Click Next to Extend the File System Resource to the Secondary Server
6. In the Extend Wizard, select “Accept Defaults”
7. As a result the MySQL resource is now protected on both cluster nodes. Click Finish to exit the Extend wizard.
8. Note: LifeKeeper will automatically identify that the MySQL resource has a dependency on the FileSystem resource (/var/lib/mysql). The Filesystem Resource will appear underneath the MySQL resource in the GUI
9. Your resource hierarchy should look as follows:



### Create the MySQL IP Address Dependency

In this step will define an additional dependency: that MySQL depends on a Virtual IP (192.168.197.151) so that the IP address follows the MySQL database should it move. This IP (.151) is the IP the webserver will use to access the MySQL database.

1. From the LifeKeeper GUI toolbar, right-click on the “mysql” resource
2. Select “Create Dependency” from the right-click context menu
3. In the Child Resource Tag dropdown menu, select “ip-192.168.197.151”
4. Click Next
5. Click Create Dependency
6. Click Done
7. The Virtual IP address resource (192.168.197.151) will now appear underneath the MySQL resource in the LifeKeeper user interface. This ensures that resources move together, and are started/stopped in the proper order.
8. Your resource hierarchy should look as follows



At this point in the Evaluation, we have fully protected Apache, MySQL, and their dependent resources: IP addresses, and Storage, both shared and replicated.

## Phase 7 –Test Your Environment

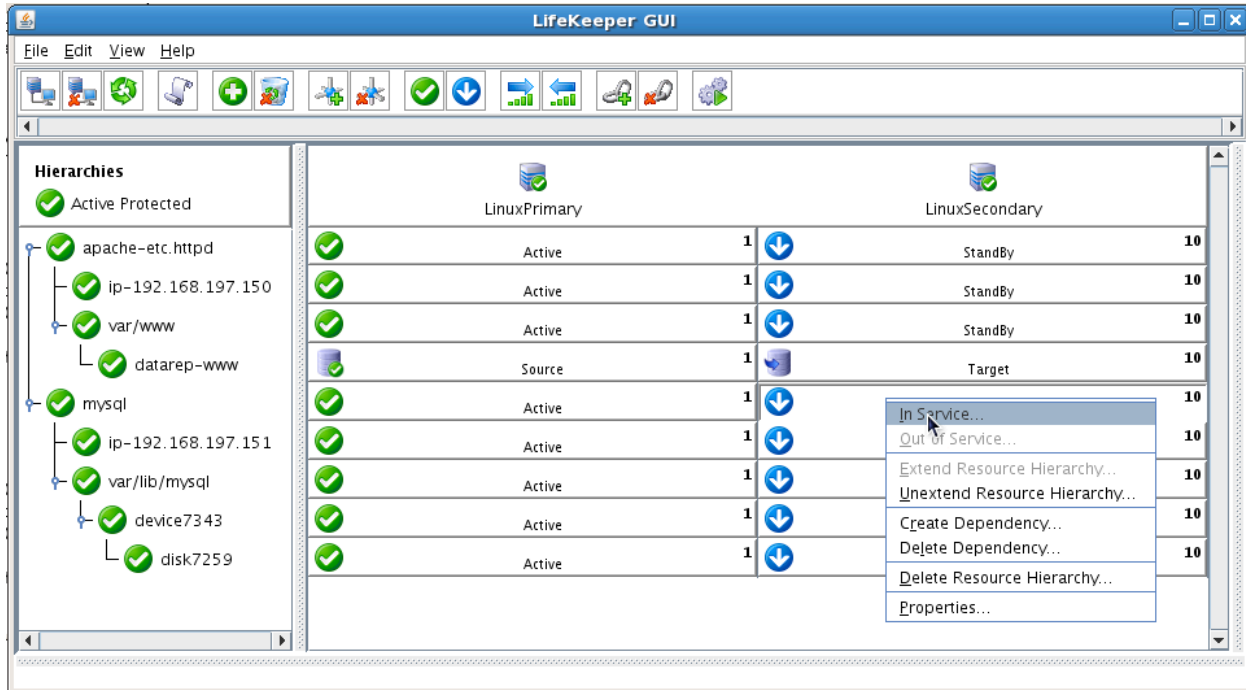
The following test scenarios have been included to guide you as you get started evaluating SteelEye Protection Suite for Linux. **Before beginning these tests, make sure the data replication resources are in the mirroring state.**

Note: For these test examples, the Primary Server is referred to as LINUXPRIMARY. The Backup or Secondary Server is referred to as LINUXSECONDARY.

### 1. Manual Switchover of the MySQL Hierarchy to Secondary Server

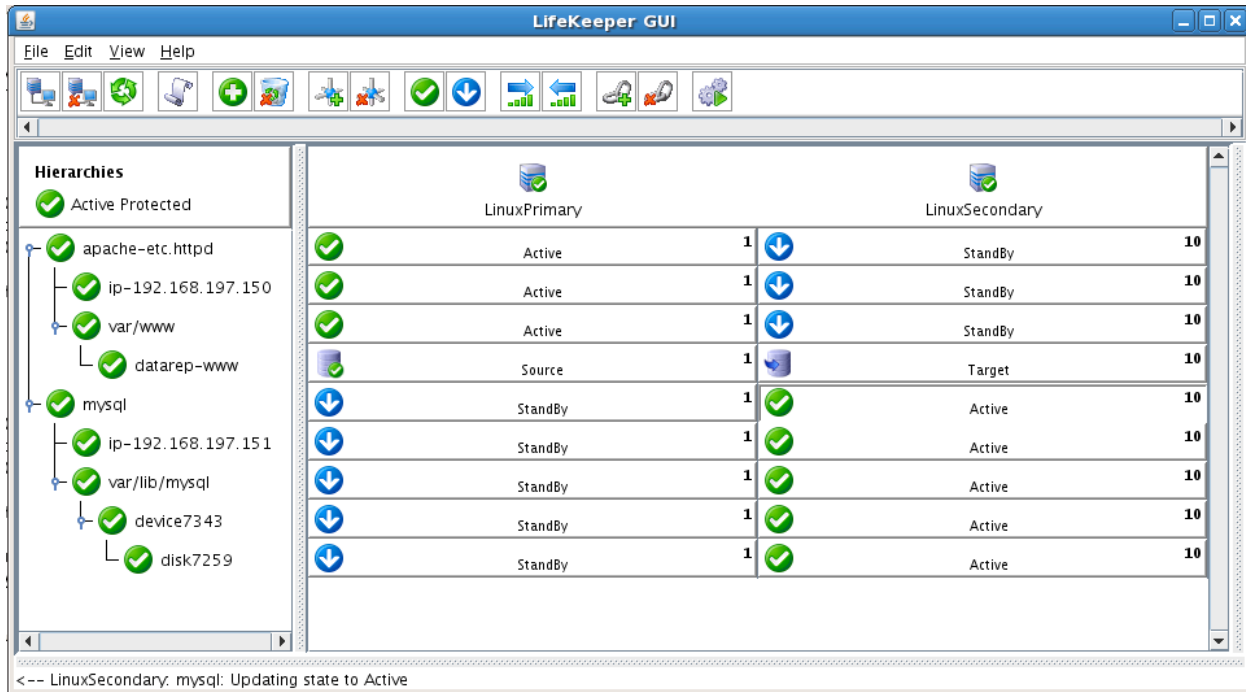
#### Procedure:

- From the LifeKeeper GUI, right click on the MySQL resource on the Secondary Server (LINUXSECONDARY) and choose IN SERVICE.
- Click “In Service” in the window that pops up



**Expected Result:**

- Beginning with the MySQL resource, all resources will be removed from service on the Active Server (LINUXPRIMARY).
- Beginning with the dependent resources (IP and Shared Volume), all resources will be brought in service on LINUXSECONDARY
- At this point, we now have an “Active/Active” cluster because both cluster nodes are actively running resources.



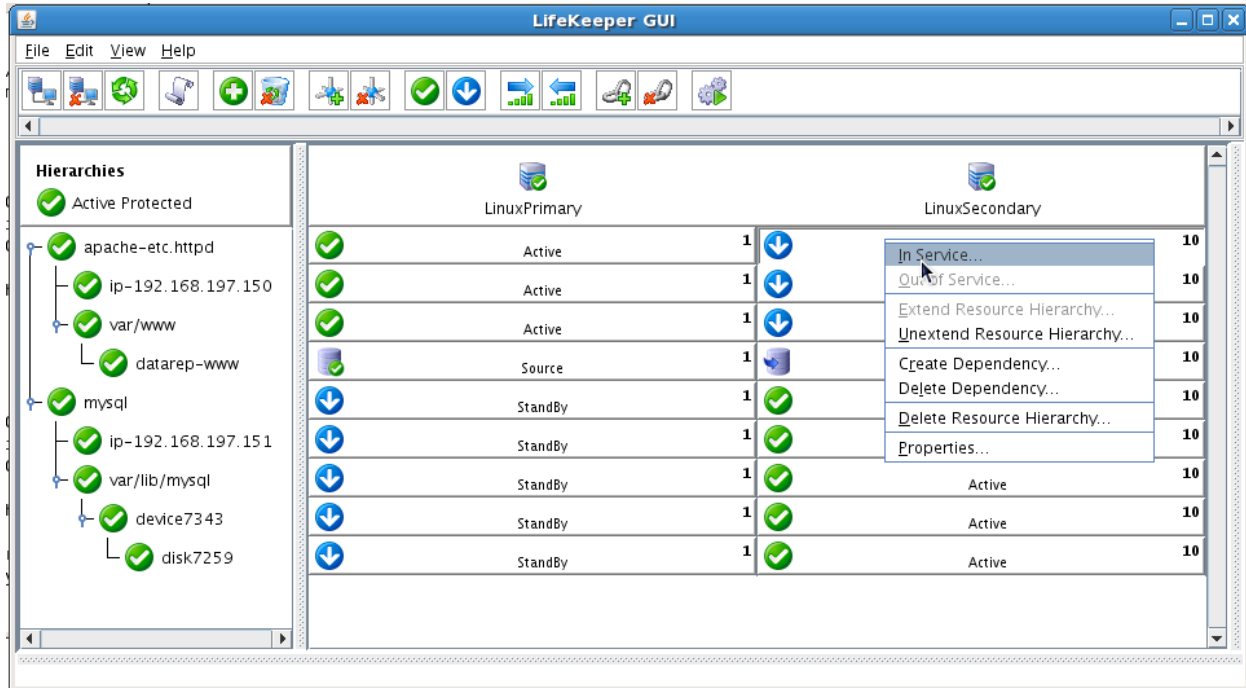
### Tests/Verification:

- Using the LifeKeeper GUI, verify that the MySQL and dependent resources are active on LINUXSECONDARY.
- Run “ifconfig -a” on LINUXSECONDARY to validate that the IP Address 192.168.197.151 is active on LINUXSECONDARY
- Run “df -h” to verify that the /var/lib/mysql shared iSCSI filesystem is mounted on LINUXSECONDARY
- Verify the MySQL services are running on LINUXSECONDARY by running “ps -ef | grep -i mysql”
- On LINUXSECONDARY run the following command to verify client connectivity to the MySQL database:
  - # mysql -S /var/lib/mysql/mysql.sock -u root -p
  - (enter password “SteelEye”)
- Verify that the SCSI reservation has properly locked the iSCSI LUN to the currently active cluster node. On LINUXPRIMARY, run “mount /dev/sdc1 /var/lib/mysql”. This should FAIL because LINUXPRIMARY does not own the SCSI reservation on this LUN.

## 2. Manual Switchover of the Apache Hierarchy to Secondary Server

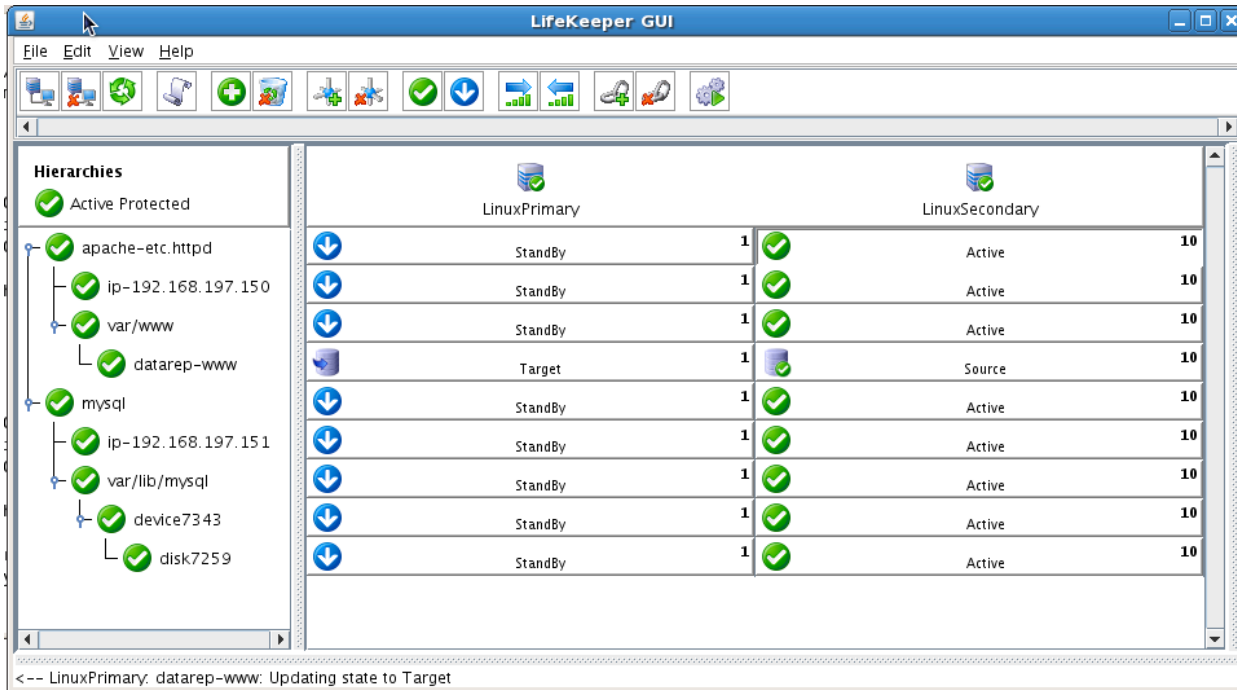
### Procedure:

- From the LifeKeeper GUI, right click on the Apache resource on the Secondary Server (LINUXSECONDARY) and choose IN SERVICE.
- Click “In Service” in the window that pops up



**Expected Result:**

- Beginning with the Apache resource, all resources will be removed from service on the Active Server (LINUXPRIMARY).
- Beginning with the dependent resources (IP and Replicated Volume), all resources will be brought in service on LINUXSECONDARY
- During this process, the direction of the mirror reversed. Data is now transmitting from LINUXSECONDARY -> LINUXPRIMARY
- At this point, we now back to an "Active/Passive" cluster because all services are now actively running on LINUXSECONDARY



**Tests/Verification:**

- Using the LifeKeeper GUI, verify that the Apache and dependent resources are active on LINUXSECONDARY.
- Using the LifeKeeper GUI, verify the mirror is now reversed and mirroring in the opposite direction. Right click on the "datarep-www" resource and select Properties



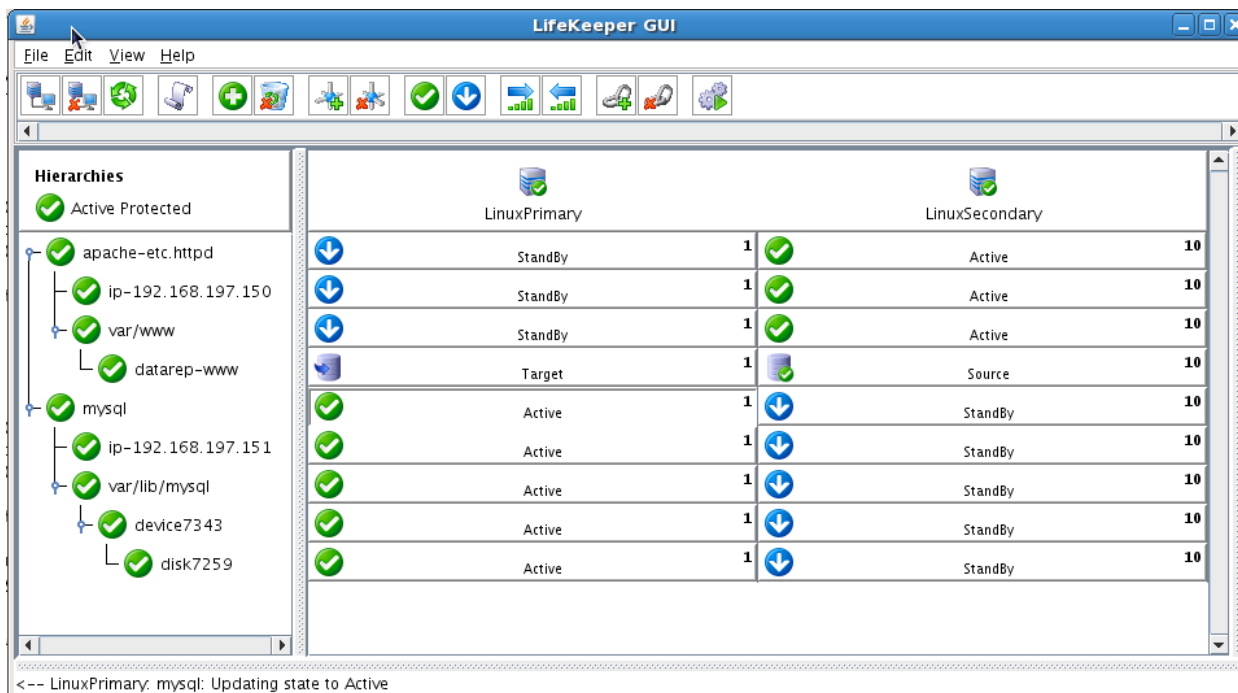
### 3. Manual Switchover of the MySQL Hierarchy back to Primary Server

**Procedure:**

- From the LifeKeeper GUI, right click on the MySQL resource on the Primary Server (LINUXPRIMARY) and choose IN SERVICE.
- Click “In Service” in the window that pops up

**Expected Result:**

- Beginning with the MySQL resource, all resources will be removed from service on the Active Server (LINUXSECONDARY).
- Beginning with the dependent resources (IP and Shared Volume), all resources will be brought in service on LINUXPRIMARY



**Tests/Verification:**

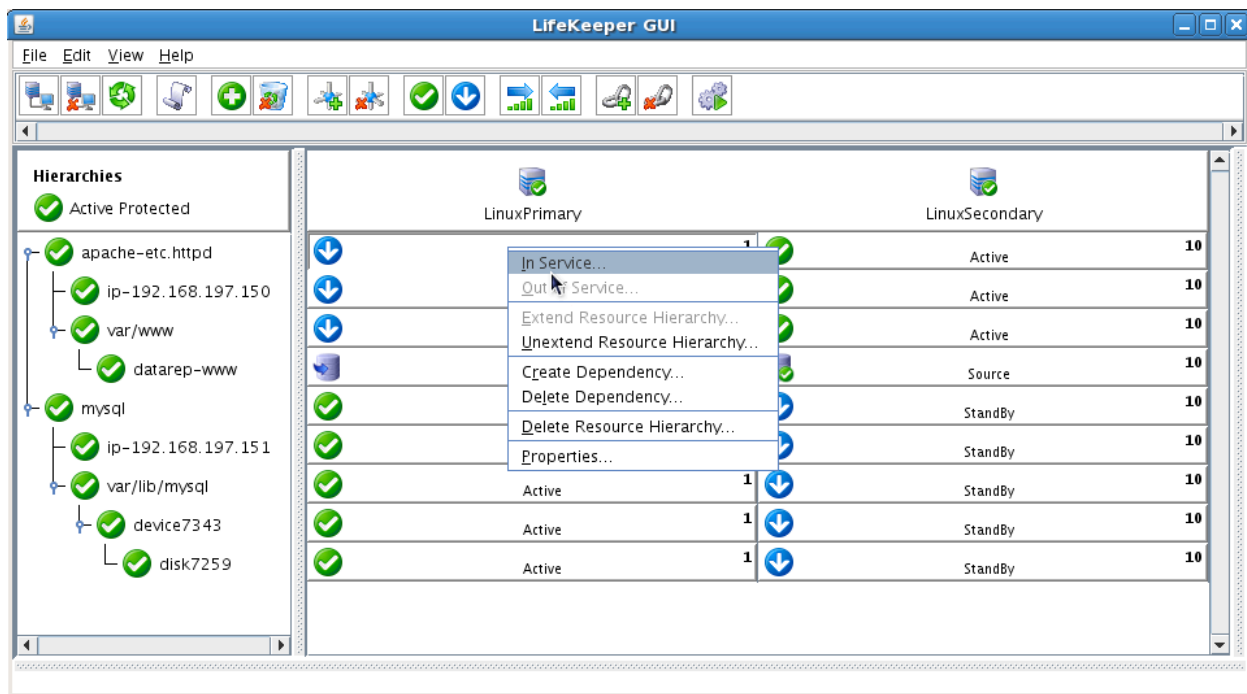
- Using the LifeKeeper GUI, verify that the MySQL and dependent resources are active on LINUXPRIMARY.
- Run “ifconfig -a” on LINUXPRIMARY to validate that the IP Address 192.168.197.151 is active on LINUXPRIMARY
- Run “df -h” to verify that the /var/lib/mysql shared iSCSI filesystem is mounted on LINUXPRIMARY
- Verify the MySQL services are running on LINUXPRIMARY by running “ps -ef | grep -i mysql”
- On LINUXPRIMARY run the following command to verify client connectivity to the MySQL database:
  - # mysql -S /var/lib/mysql/mysql.sock -u root -p

- (enter password “SteelEye”)
- Verify that the SCSI reservation has properly locked the iSCSI LUN to the currently active cluster node. On LINUXSECONDARY, run “mount /dev/sdc1 /var/lib/mysql”. This should FAIL because LINUXSECONDARY does not own the SCSI reservation on this LUN.

## 4. Manual Switchover of the Apache Hierarchy back to the Primary Server

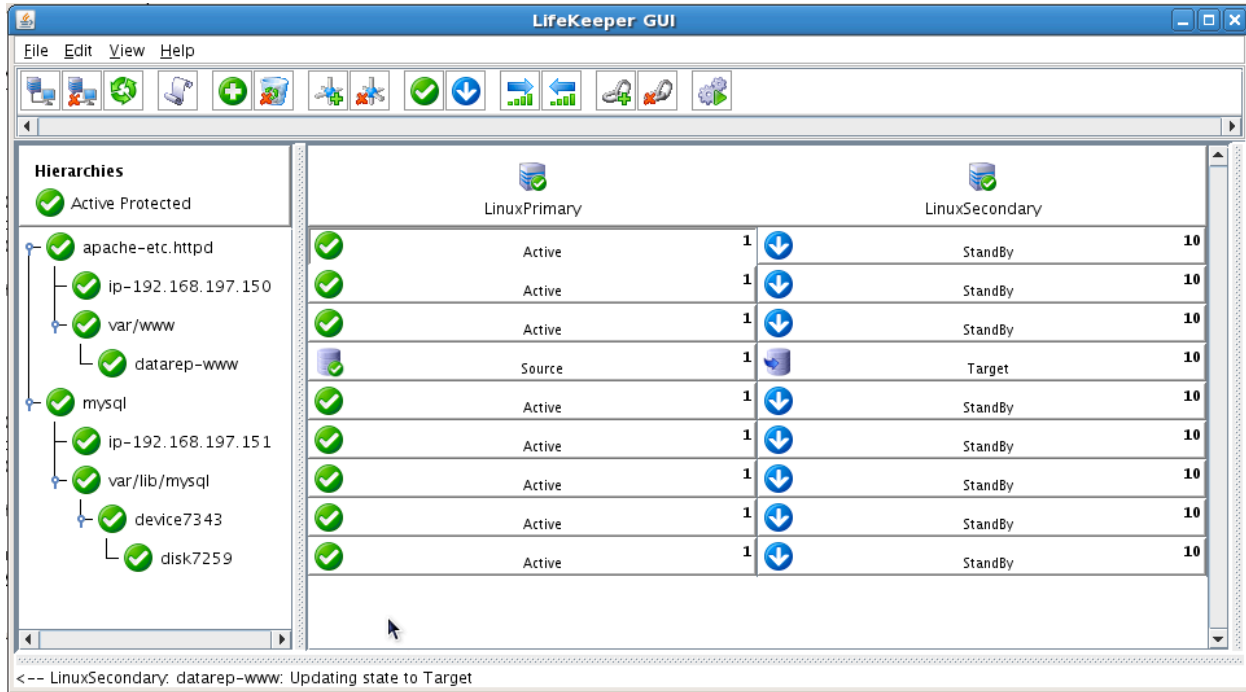
### Procedure:

- From the LifeKeeper GUI, right click on the Apache resource on the Primary Server (LINUXPRIMARY) and choose IN SERVICE.
- Click “In Service” in the window that pops up



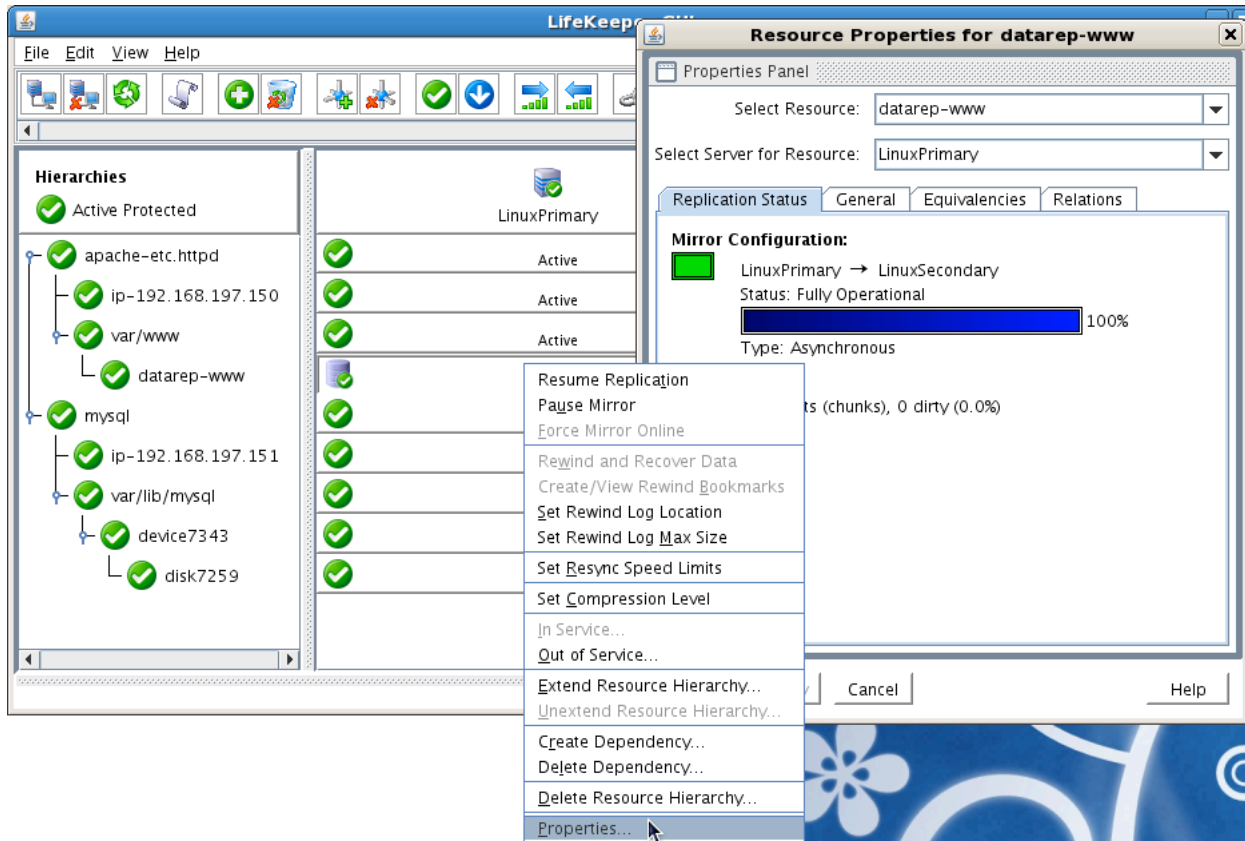
### Expected Result:

- Beginning with the Apache resource, all resources will be removed from service on the Active Server (LINUXSECONDARY).
- Beginning with the dependent resources (IP and Replicated Volume), all resources will be brought in service on LINUXPRIMARY
- During this process, the direction of the mirror reversed. Data is now transmitting from LINUXPRIMARY -> LINUXSECONDARY



**Tests/Verification:**

- Using the LifeKeeper GUI, verify that the Apache and dependent resources are active on LINUXSECONDARY.
- Using the LifeKeeper GUI, verify the mirror is now reversed and mirroring in the opposite direction. Right click on the “datarep-www” resource and select Properties



- Run “ifconfig –a” on LINUXPRIMARY to validate that the IP Address 192.168.197.150 is active on LINUXPRIMARY
- Run “df –h” to verify that the /var/www replicated filesystem is mounted as an “md” device (example: /dev/md0”) on LINUXPRIMARY
- Verify the Apache services are running on LINUXPRIMARY by running “ps –ef | grep –i httpd”
- Open a Web Browser to <http://192.168.197.150> and verify that it can successfully connect. The PHPInfo output should indicate that the system name is “LinuxPrimary”

**PHP Version 5.1.6**


<b>System</b>	Linux LinuxPrimary 2.6.18-194.26.1.el5 #1 SMP Tue Nov 9 12:54:40 EST 2010 i686
<b>Build Date</b>	Nov 29 2010 16:49:03
<b>Configure</b>	'./configure' '--build=i686-redhat-linux-gnu' '--host=i686-redhat-linux-gnu' '--target=i386-

## 5. Simulate a network failure on the Primary Server by failing the IP resource

**IMPORTANT NOTE: Only perform this test if you have more than one communications path configured.**

If you perform this test and have only one communications path configured, your system will enter a split-brain scenario as described in the LifeKeeper Administration Guide found [here](#). Refer to this document for more information or contact SteelEye pre-sales technical support for assistance in resolving this condition.

### Procedure:

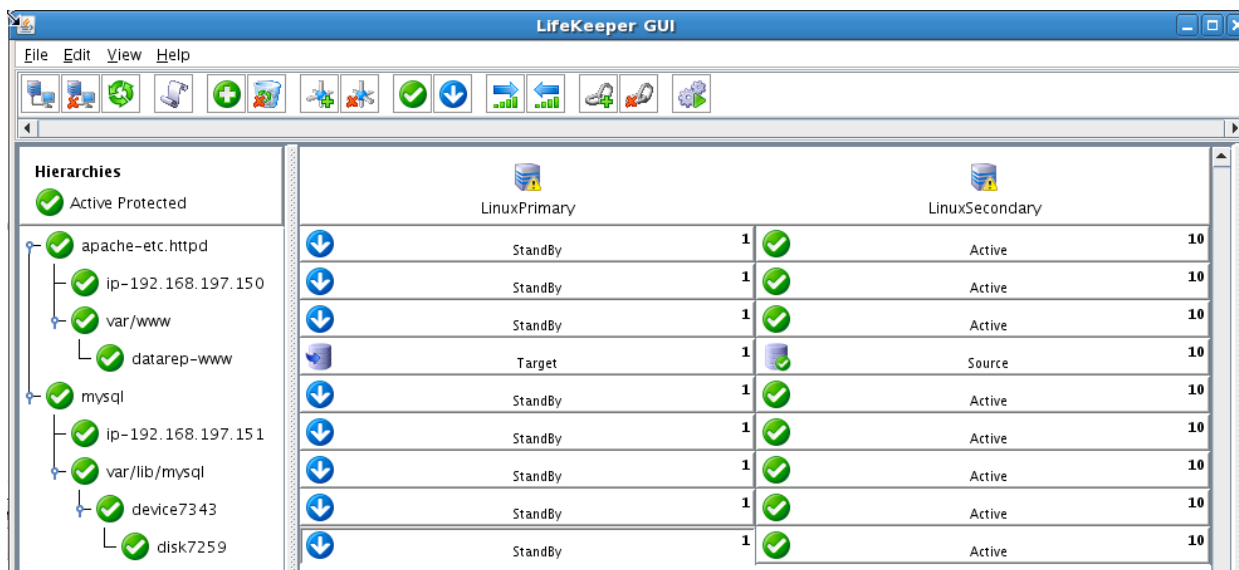
- On LINUXPRIMARY, pull the network cable attached to the NIC that the virtual IP address is configured on

### Expected Result:

- The IP Resource should fail first.
- The entire hierarchy should failover to LINUXSECONDARY

### Tests/Verification:

- Check the LifeKeeper Log to verify the IP resource failed – “/opt/LifeKeeper/bin/lk\_log log”
- Using the LifeKeeper GUI, verify the MySQL and Apache resource hierarchies fail over successfully to LINUXSECONDARY



## 6. Hard failover of the resource from the Secondary Server back to the Primary Server

### Procedure:

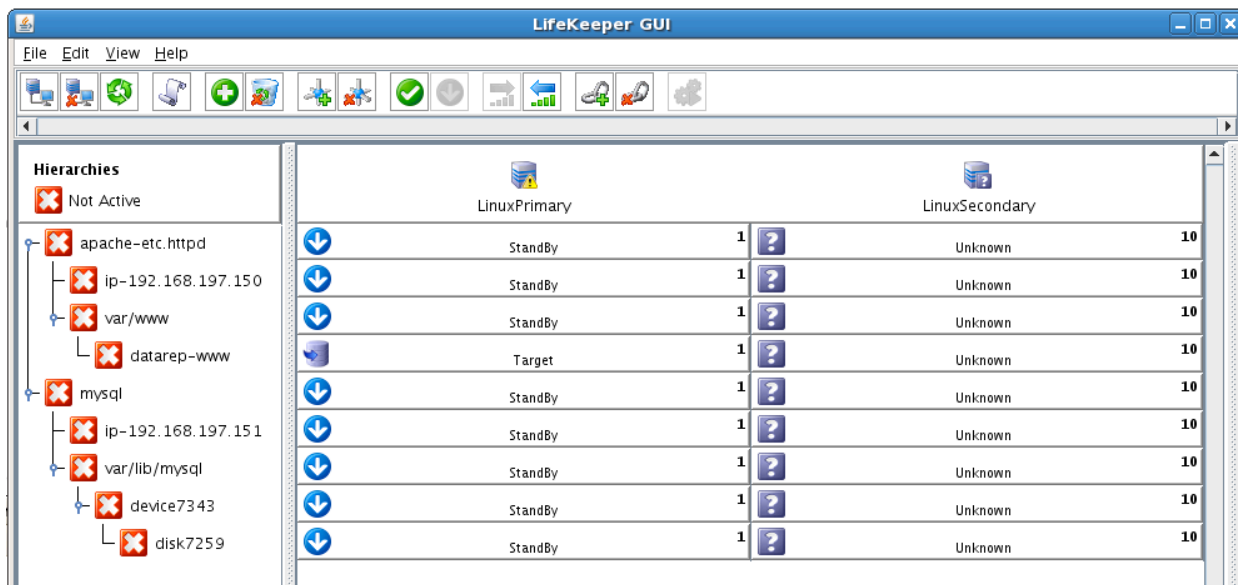
- Pull the power cord on LINUXSECONDARY, as this is the server with all resources currently In Service.

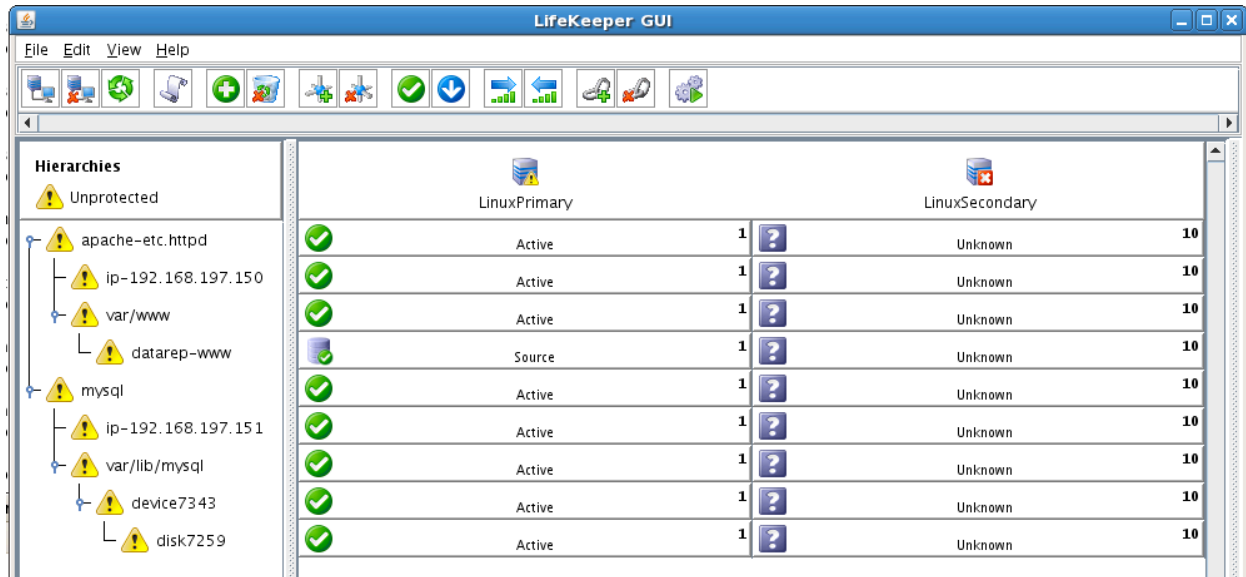
### Expected Result:

- After failure has been detected, beginning with the dependent resources (IP and Volume), all resources will be brought in service on LINUXPRIMARY.

### Tests/Verification:

- Using the LifeKeeper GUI, verify the mirror has reversed and is in a Resync Pending state waiting for LINUXSECONDARY to come back on line.
- Verify the Apache and MySQL Server services are running on LINUXPRIMARY.
- Verify that the client can still connect to the Webserver and database running on LINUXPRIMARY.
- Verify you can write data to the replicated volume, /var/www on LINUXPRIMARY.





## 7. Bring Failed Server back on line

### Procedure:

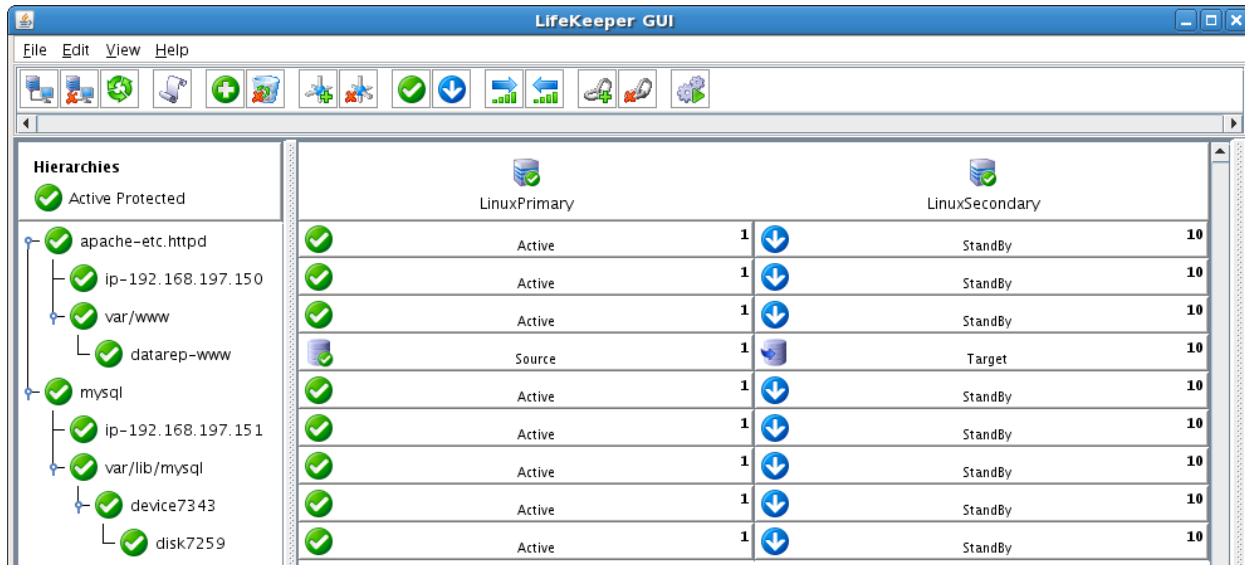
- Plug the power cord back into LINUXSECONDARY and boot it up.

### Expected Result:

- Using the LifeKeeper GUI, verify that LINUXSECONDARY is coming back up and has become the Standby Server.

### Tests/Verification:

- Verify the mirror performs a quick partial resync and moves to the Mirroring state
- Verify the Apache and MySQL Hierarchy are in service on LINUXPRIMARY and standby on LINUXSECONDARY.



## 8. Verify Local Recovery of MySQL Server

### Procedure:

- Kill the MySQL processes via the command line:
  - # ps -ef | grep sql
  - # killall mysqld mysqld\_safe
  - run "ps -ef | grep sql" once again to verify that the processes no longer exist

### Expected Result: (Assumes Local Recovery for SQL resource is set to YES)

- The MySQL Server service should stop.
- The MySQL quickcheck process will automatically restart the MySQL Server Service when it runs periodically.
- No failure of MySQL should occur.

### Tests/Verification:

- Execute "ps -ef | grep sql" once again to verify that the mysql processes have been restored locally on LINUXPRIMARY.
- Verify connectivity to the MySQL database by running:
  - # mysql -S /var/lib/mysql/mysql.sock -u root -p
  - (Enter password "SteelEye")
- If you inspect the LifeKeeper logs, you will see information indicating that LifeKeeper detected the failure of the MySQL service and recovered it locally. Run /opt/LifeKeeper/bin/lk\_log log for more information.