

PowerGres Plus V9.1 向け LifeKeeper による HA クラスタ構成 導入手順書

バージョン 1.0.1

2014 年 4 月 4 日



SRA OSS, INC.

SRA OSS, Inc. 日本支社

改訂履歴

バージョン	変更日	変更内容
1.0.1	2014 年 4 月 4 日	<ul style="list-style-type: none">• 実行例のファイル名に含まれるバージョンを LifeKeeper for Linux v8.2.1、PowerGres Plus (Linux 版) V9.1 Update 1 に合わせて更新• 「6.1. データベースクラスタの作成」にパラメータの設定を変更する手順と、lc_messages パラメータは C 以外に設定してはいけないという記述を追加• 「9. リソース依存関係の作成」でデータレプリケーションリソースの子として IP リソースの依存関係を作成するように手順を変更
1.0	2014 年 3 月 6 日	<ul style="list-style-type: none">• 新規作成

著作権

本書の著作権は SRA OSS, Inc. 日本支社に帰属します。本書の一部または全部を SRA OSS, Inc. 日本支社の許諾なしに複製、転載および改変などを行うことは禁止されています。

Copyright © 2014 SRA OSS, Inc. Japan All rights reserved.

免責事項

本書の内容は予告なしに変更されることがあります。SRA OSS, Inc. 日本支社は、本書の内容について万全を期していますが、その内容の正確性を保証するものではありません。SRA OSS, Inc. 日本支社は本書を利用することで発生したいかなる損害について一切の責任を負いません。

商標

PowerGres は株式会社 SRA の登録商標です。

SteelEye Technology、LifeKeeper および DataKeeper は SIOS Technology Corp. の登録商標です。SteelEye は SIOS Technology Corp. の商標です。

Linux は Linux Torvalds 氏の日本およびその他の国における登録商標または商標です。

PostgreSQL は PostgreSQL Global Development Group の米国およびその他の国における登録商標または商標です。

その他、本書に記載の社名および商品名などはそれぞれの所有者に所有権が帰属する商標または登録商標が含まれています。

目次

1.はじめに.....	1
1.1.目的.....	1
1.2.対象読書.....	1
1.3.関連文書.....	1
2.システム構成.....	2
3.インストール.....	4
3.1.LifeKeeper のインストール.....	4
3.2.PowerGres Plus のインストール.....	5
3.3.PowerGres Recovery Kit のインストール.....	7
4.コミュニケーションパスの作成.....	8
5.データレプリケーションリソース階層の作成.....	18
6.PowerGres Plus の設定.....	36
6.1.データベースクラスタの作成.....	36
6.2.パスワードファイルの作成.....	40
6.3.クライアント認証の設定.....	40
6.4.オンラインバックアップの設定.....	43
6.5.データ暗号化の設定.....	50
7.PowerGres リソース階層の作成.....	65
8.IP リソース階層の作成.....	73
9.リソース依存関係の作成.....	81

1. はじめに

PowerGres Plus は、オープンソースデータベースの PostgreSQL をベースに、トランザクションログの二重化やデータ暗号化など、独自に機能を拡張したデータベース製品です。HA (High Availability; 高可用性) クラスタソフトウェアの LifeKeeper によって HA クラスタ構成を構築することにより、シングルサーバでは実現できない高可用性を実現できます。

1.1. 目的

本書では LifeKeeper によって PowerGres Plus の HA クラスタ構成を導入する手順について説明します。また、PowerGres Plus 独自の機能を HA クラスタ構成で使用するのに必要な手順についても説明します。

1.2. 対象読書

本書は PowerGres Plus の LifeKeeper による HA クラスタ構成を設計、導入するエンジニアを対象にしています。なお、HA クラスタ構成の概念、Linux (Unix 系 OS) の基本的な操作について理解していることが前提になります。

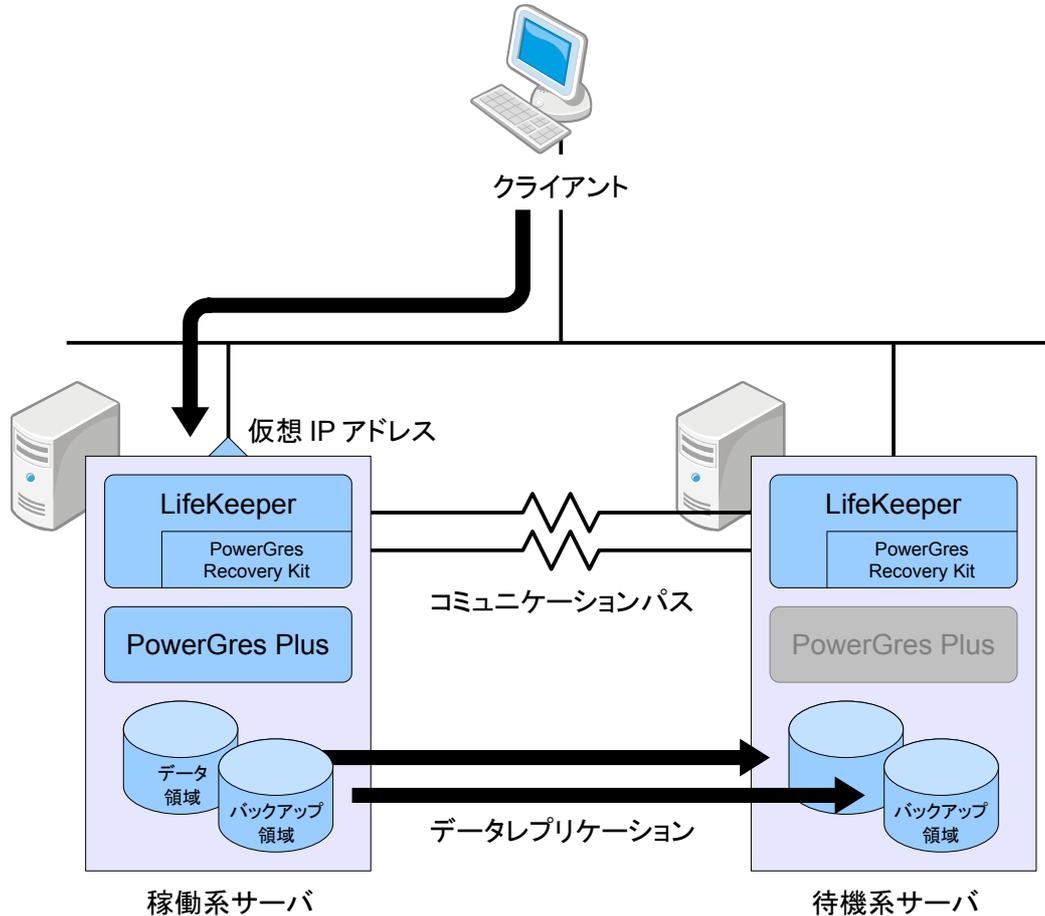
1.3. 関連文書

PowerGres Plus および LifeKeeper の詳細については以下の文書を参照してください。

- [PowerGres Plus \(Linux 版\) マニュアル](#)
- [PowerGres Plus Manager マニュアル](#)
- [SteelEye Protection Suite for Linux インストールガイド](#)
- [SteelEye Protection Suite for Linux テクニカルドキュメンテーション](#)

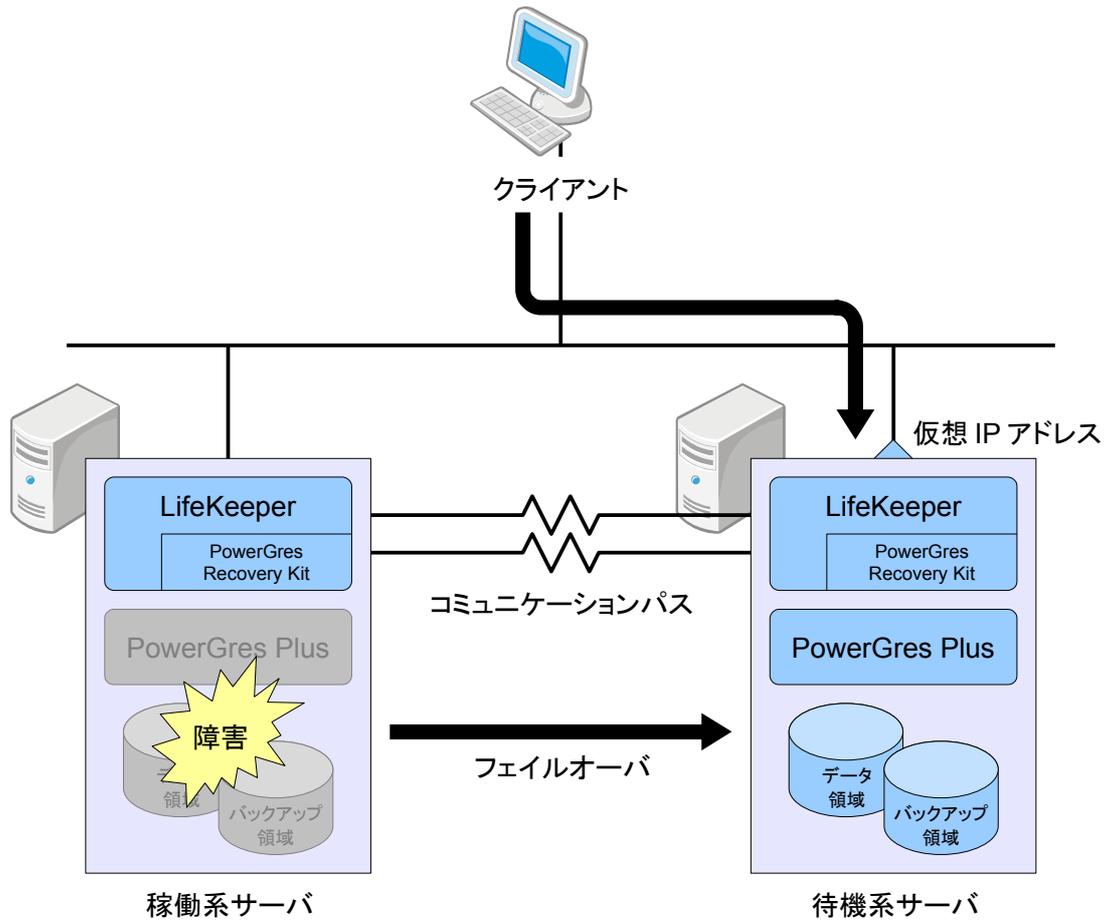
2. システム構成

本書では以下のようなアクティブ・スタンバイ型の HA クラスタ構成を対象にしています。



HA クラスタ構成では通常は稼働系サーバで PowerGres Plus のサービスが起動しています。クライアントは仮想 IP アドレスを通してデータベースに接続します。データ領域とバックアップ領域のデータは DataKeeper によって待機系サーバにレプリケーションされます。

PowerGres Plus、データ領域とバックアップ領域の配置されたディスク、仮想 IP アドレスの状態は LifeKeeper によって監視されます。また、サーバの状態はコミュニケーションパスを通して相互に監視されます。稼働系サーバで障害が検知されると、ローカルサーバでサービスの復旧が開始されます。ローカルサーバでサービスを復旧できない場合には待機系サーバへの切り替え(フェイルオーバー)が開始されます。



待機系サーバへのサービスの切り替えが完了すると、待機系サーバで PowerGres Plus のサービスが起動されます。データ領域とバックアップ領域の配置されたディスクと仮想 IP アドレスも待機系サーバに切り替わります。データベースへの接続は仮想 IP アドレスを通して行うため、クライアントは待機系サーバにサービスが切り替わったことを意識する必要がありません。

3. インストール

以下のソフトウェアをすべてのサーバにインストールします。

- LifeKeeper for Linux v8
- PowerGres Plus (Linux 版) V9.1
- PowerGres Recovery Kit V9.1

3.1. LifeKeeper のインストール

LifeKeeper をすべてのサーバにインストールします。詳細については「[SteelEye Protection Suite for Linux インストールガイド](#)」を参照してください。

インストールは以下の手順に従って root ユーザで行います。

1. LifeKeeper のメディアを CD ドライブにセットします。

自動マウントの設定が有効な場合には自動的にマウントされます。マウント先のディレクトリはデフォルトでは /media/LKL_V821_031414 になります。ディレクトリ名はバージョンによって異なる場合があります。

自動的にマウントされない場合には手動でマウントします。

```
# mkdir /mnt/cdrom
# mount /dev/cdrom /mnt/cdrom
```

CD ドライブのデバイスは環境によって異なる場合があります。マウント先のディレクトリはあらかじめ作成しておく必要があります。

2. メディアがマウントされたディレクトリに移動します。

```
# cd /media/LKL_V821_031414
```

3. セットアップイメージをマウントします。

```
# mkdir /mnt/sps
# mount -o loop sps_821.img /mnt/sps
```

セットアップイメージのファイル名はバージョンによって異なる場合があります。マウント先のディレクトリはあらかじめ作成しておく必要があります。

4. セットアップイメージをマウントしたディレクトリに移動します。

```
# cd /mnt/sps
```

5. セットアップスクリプトを実行します。

```
# ./setup
```

セットアップスクリプトでは表示される指示に従って対話的に操作していきます。基本的に Enter キーを押してデフォルトの設定のままです。

6. セットアップスクリプト内でライセンスキーをインストールしていない場合にはインストールします。

```
$ sudo /opt/LifeKeeper/bin/lkkeyins
```

```
Please enter the complete pathname of the file containing your license key:
(ライセンスキーファイルの絶対パスを入力)
```

```
LifeKeeper license key installation was successful!
```

7. セットアップスクリプト内で DataKeeper をインストールしていない場合にはインストールします。

```
# rpm -ivh kits/steeleye-lkDR-8.2.1-6353.noarch.rpm
準備中... ##### [100%]
  1:steeleye-lkDR ##### [100%]
%]
```

DataKeeper は LifeKeeper と組み合わせてデータをレプリケーションするためのソフトウェアです。

DataKeeper を含むオプションの Recovery Kit のパッケージは kits ディレクトリ内に格納されています。DataKeeper のパッケージ名は steeleye-lkDR になります。パッケージのファイル名はバージョンによって異なる場合があります。

8. コマンドとオンラインマニュアルのパスの設定を行います。

root ユーザのログインシェルが bash の場合には初期化ファイル ~/.bash_profile の最後に以下の内容を追加します。

```
# for LifeKeeper

PATH=$PATH:/opt/LifeKeeper/bin
MANPATH=$MANPATH:/opt/LifeKeeper/man

export PATH MANPATH
```

初期化ファイルはログイン時に読み込まれます。ただちに設定を反映するには以下のコマンドを実行します。

```
# . ~/.bash_profile
```

9. セットアップイメージ、メディアをアンマウントします。

```
# cd
# umount /mnt/sps
# rmdir /mnt/sps
# umount /media/LKL_V821_031414
```

3.2. PowerGres Plus のインストール

PowerGres Plus をすべてのサーバにインストールします。詳細については「[PowerGres Plus \(Linux 版\) マニュアル](#)」の「[インストールとアンインストール](#)」を参照してください。

インストールは以下の手順に従って root ユーザで行います。

1. PowerGres Plus のメディアを CD ドライブにセットします。

自動マウントの設定が有効な場合には自動的にマウントされます。マウント先のディレクトリはデフォルトでは /media/POWERGRESPLUS91 になります。

自動的にマウントされない場合には手動でマウントします。

```
# mkdir /mnt/cdrom
# mount /dev/cdrom /mnt/cdrom
```

CDドライブのデバイスは環境によって異なる場合があります。マウント先のディレクトリはあらかじめ作成しておく必要があります。

2. メディアがマウントされたディレクトリに移動します。

```
# cd /media/POWERGRESPLUS91
```

3. インストールスクリプトを実行します。

```
# ./install.sh
```

インストールスクリプトでは表示される指示に従って対話的に操作していきます。基本的に Enter キーを押してデフォルトの設定のままですべての問題ありません。

インストールスクリプトの途中でライセンスキーの入力が求められます。PowerGres Plus の使用許諾契約書に記載のライセンスキーを入力します。

4. PowerGres Plus のスーパーユーザ postgres の設定を行います。

インストールスクリプトの実行時に postgres ユーザが存在しない場合には自動的に作成されます。

- 1) postgres ユーザが存在することを確認します。

```
# id postgres
uid=26(postgres) gid=26(postgres) 所属グループ=26(postgres)
```

postgres ユーザが存在しない場合には手動で作成します。

```
# groupadd -g 26 -o -r postgres
# useradd -M -n -g postgres -o -r -d /var/lib/pgsql -s /bin/bash -c "PostgreSQL Server" -u 26 postgres
```

UIDとGIDが26でない場合には削除して作成しなおします。

- 2) パスワードの設定を行います。

```
# passwd postgres
ユーザー postgres のパスワードを変更。
新しいパスワード: (パスワードを入力)
新しいパスワードを再入力してください: (パスワードを入力)
passwd: 全ての認証トークンが正しく更新できました。
```

- 3) ライブラリ、コマンドおよびオンラインマニュアルのパスの設定を行います。

postgres ユーザの bash の初期化ファイル~postgres/.bash_profile の最後に以下の内容を追加します。インストールスクリプト内で postgres ユーザが作成された場合にはすでに設定が行われています。

```
# for PowerGres Plus V9.1

LD_LIBRARY_PATH=/opt/powergresplus91/lib:$LD_LIBRARY_PATH
PATH=/opt/powergresplus91/bin:$PATH
MANPATH=/opt/powergresplus91/share/man:$MANPATH
```

```
export LD_LIBRARY_PATH PATH MANPATH
```

5. メディアをアンマウントします。

```
# cd
# umount /media/POWERGRESPLUS91
```

3.3. PowerGres Recovery Kit のインストール

PowerGres Recovery Kit をすべてのサーバにインストールします。

PowerGres Recovery Kit は PowerGres on Linux および PowerGres Plus のサービスを LifeKeeper の保護対象として管理するためのソフトウェアです。

インストールは以下の手順に従って root ユーザで行います。

1. PowerGres Recovery Kit のメディアを CD ドライブにセットします。

自動マウントの設定が有効な場合には自動的にマウントされます。マウント先のディレクトリはデフォルトでは /media/POWERGRES91 になります。

自動的にマウントされない場合には手動でマウントします。

```
# mkdir /mnt/cdrom
# mount /dev/cdrom /mnt/cdrom
```

CD ドライブのデバイスは環境によって異なる場合があります。マウント先のディレクトリはあらかじめ作成しておく必要があります。

2. メディアがマウントされたディレクトリに移動します。

```
# cd /media/POWERGRES91
```

3. PowerGres Recovery Kit をインストールします。

```
# rpm -ivh LKRK/powergres-LKRK-9.1.0-1.noarch.rpm
準備中... ##### [100%]
  1:powergres-LKRK ##### [100%]
%]
```

PowerGres Recovery Kit のパッケージは LKRK ディレクトリ内に格納されています。パッケージ名は powergres-LKRK になります。パッケージのファイル名はバージョンによって異なる場合があります。

4. メディアをアンマウントします。

```
# cd
# umount /media/POWERGRES91
```

4. コミュニケーションパスの作成

コミュニケーションパスを作成します。詳細については「[SteelEye Protection Suite for Linux テクニカルドキュメンテーション](#)」の「[コミュニケーションパスの作成](#)」を参照してください。

コミュニケーションパスはサーバの状態を監視するのに使用する通信経路です。すべてのコミュニケーションパスが切断されると、サーバの状態を相互に監視できなくなります。そうすると、両方のサーバでサービスが起動され、サービスを正常に提供できない状態になる場合があります。そのため、コミュニケーションパスは 2 本以上作成することを推奨します。

ここではコミュニケーションパスを 3 本作成します。コミュニケーションパスの設定は以下のとおりです。背景色が青灰色(■)の部分はデフォルトの設定のままであることを表します。

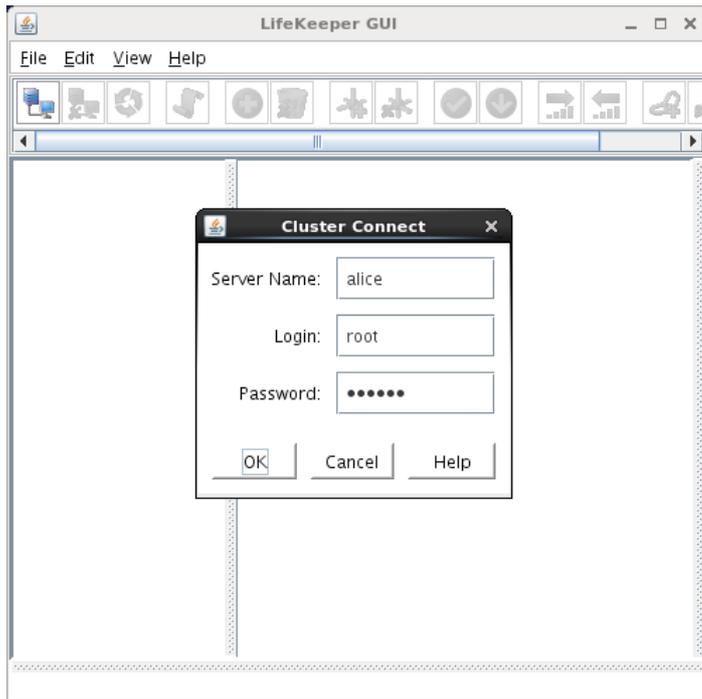
	1 本目	2 本目	3 本目
Local Server	alice	alice	alice
Remote Server(s)	bob	bob	bob
Device Type	TCP	TCP	TCP
Device Type が TCP の場合			
Local IP Address(es)	192.168.137.2	192.168.1.2	192.168.2.2
Remote IP Address	192.168.137.3	192.168.1.3	192.168.2.3
Priority	1	2	3
Device Type が TTY の場合			
Local TTY Device	-	-	-
Remote TTY Device	-	-	-
Baud Rate	-	-	-

コミュニケーションパスの作成は以下の手順に従って稼働系サーバにおいて root ユーザで行います。

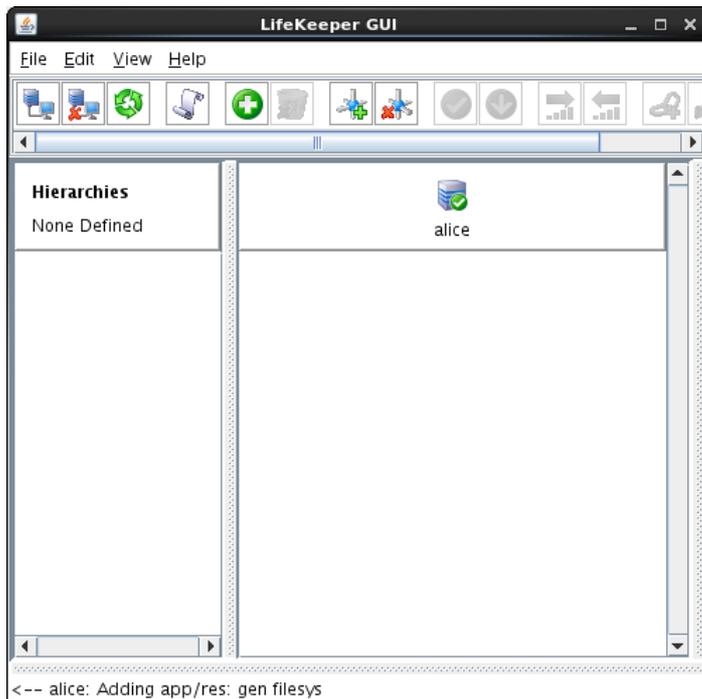
1. LifeKeeper GUI を起動します。

```
# lkGUIapp
```

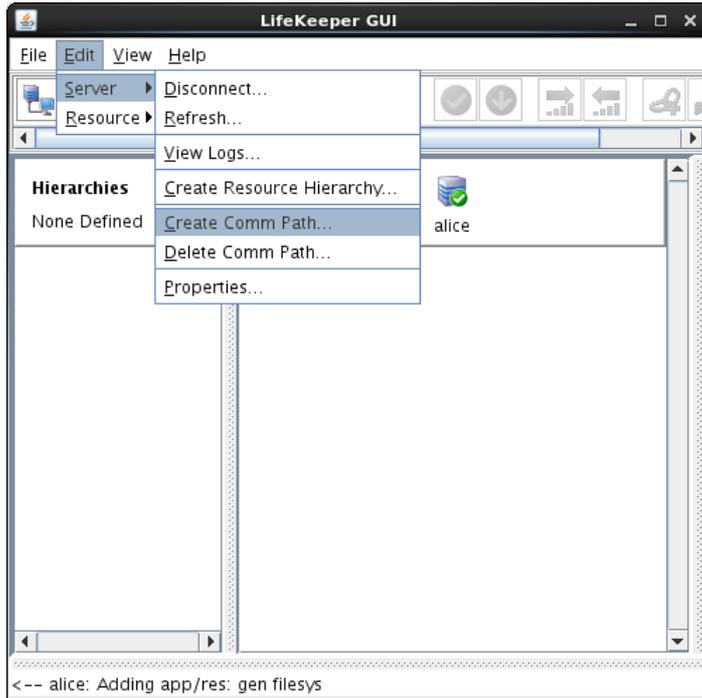
2. LifeKeeper GUI に接続します。



Server Name に接続先のサーバ名、**Login** にユーザ名として root、**Password** に root ユーザのパスワードを入力し、**OK** ボタンを押します。



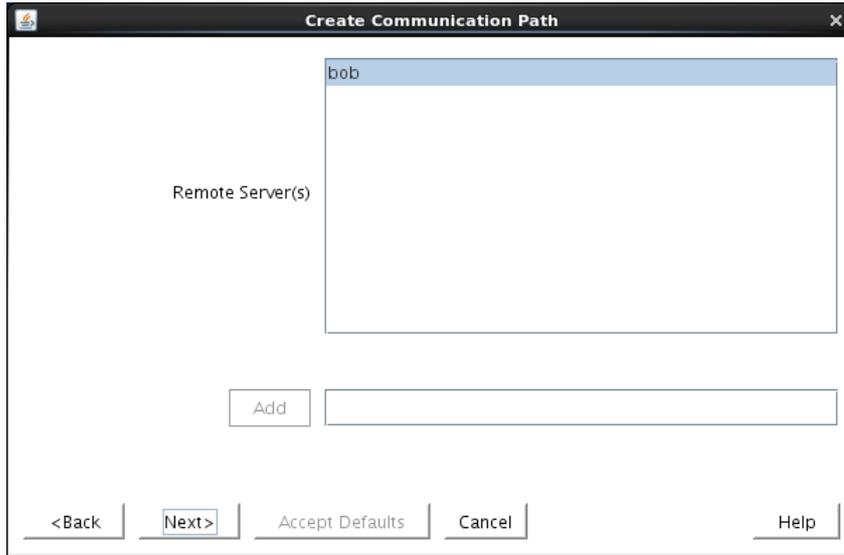
3. **Edit** メニューの **Server** ▶ **Create Comm Path...** を選択します。



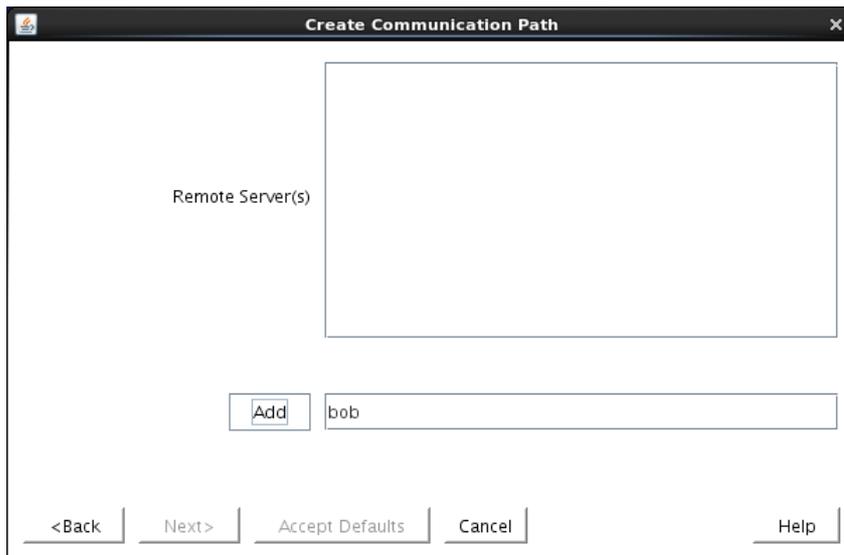
4. **Local Server** から稼働系サーバを選択し、**Next>**ボタンを押します。



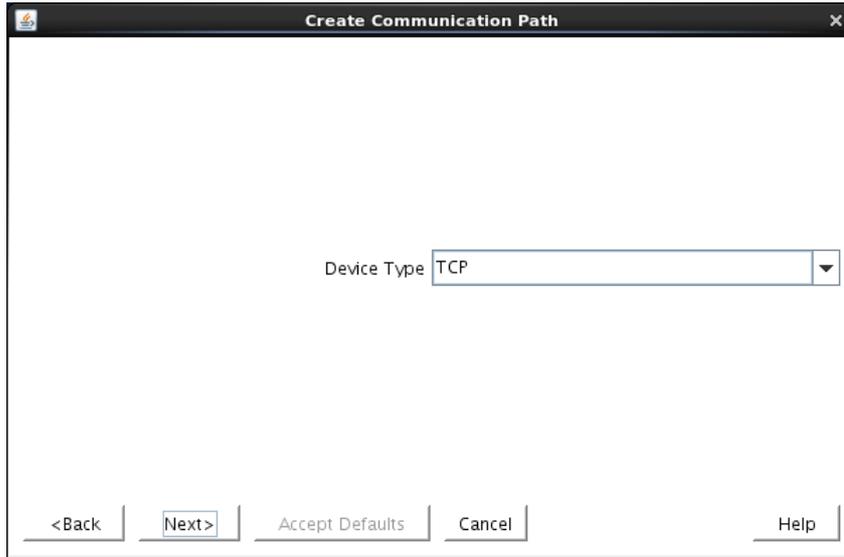
5. **Remote Server(s)**から待機系サーバを選択し、**Next>**ボタンを押します。



Remote Server(s)に待機系サーバが表示されていない場合には、**Add** ボタンの右に待機系サーバのホスト名を入力し、**Add** ボタンを押します。そうすると、**Remote Server(s)**に待機系サーバが追加されます。



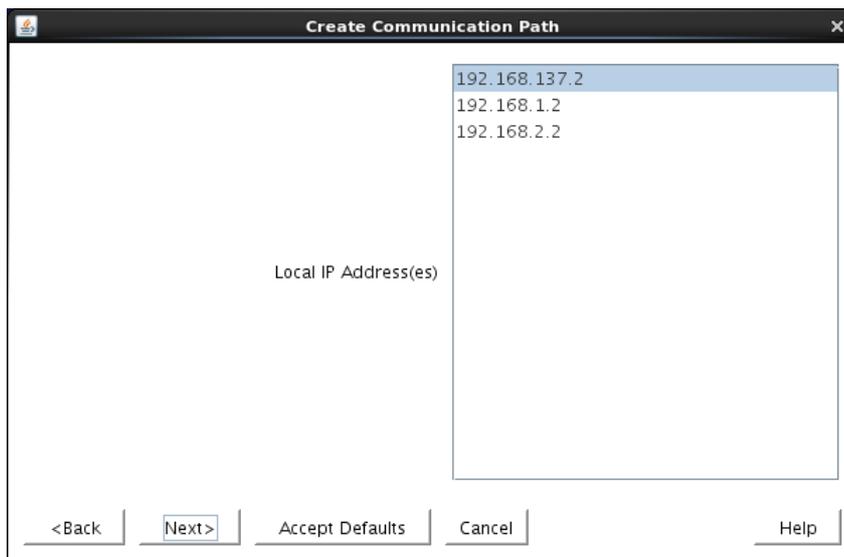
6. **Device Type** からコミュニケーションパスの通信方式として TCP または TTY を選択し、**Next>** ボタンを押します。



TCP は TCP/IP 通信、TTY はシリアル通信を表します。シリアル通信によるコミュニケーションパスはサーバ間に 1 本のみ作成できます。

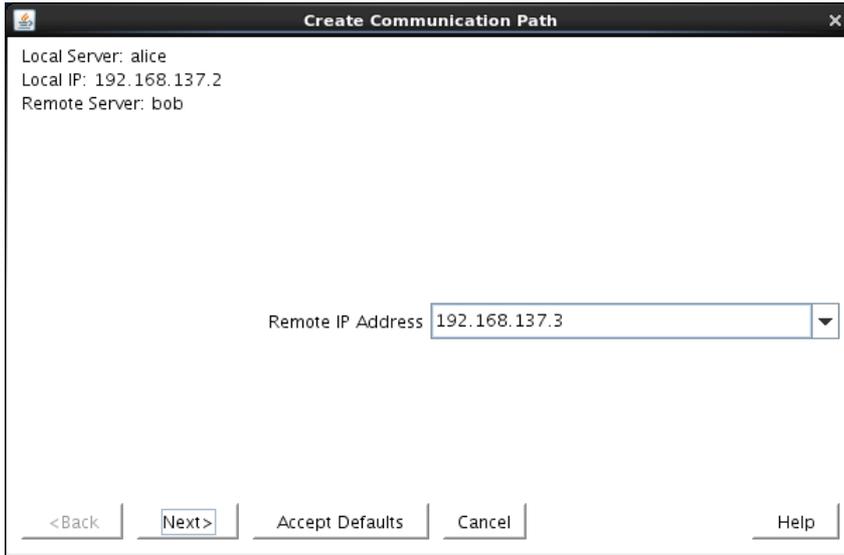
7. Device Type が TCP の場合

1) **Local IP Address(es)**から稼働系サーバの IP アドレスを選択します。

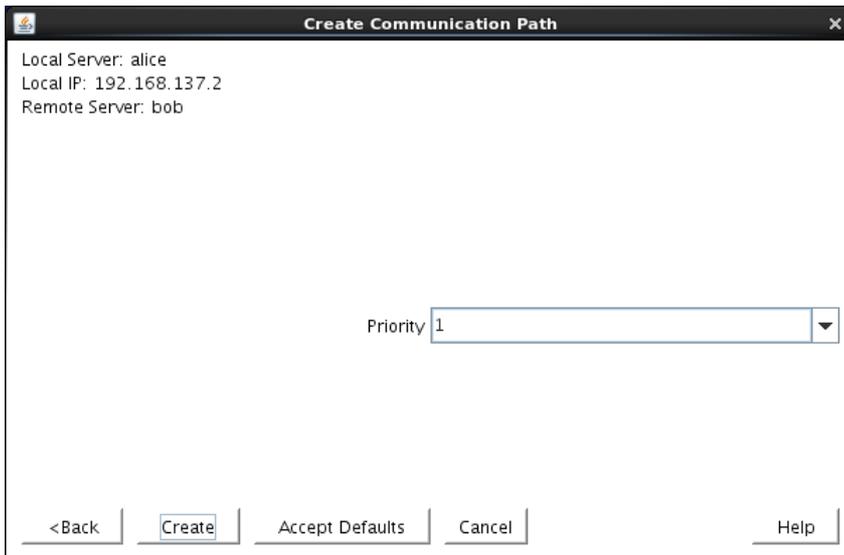


これ以降の設定を確認せずにデフォルトの設定のまま問題ない場合には **Accept Defaults** ボタンを押します。そうでない場合には **Next>** ボタンを押します。ここでは **Next>** ボタンを押した場合について説明します。

2) **Remote IP Address** から待機系サーバの IP アドレスを選択し、**Next>** ボタンを押します。

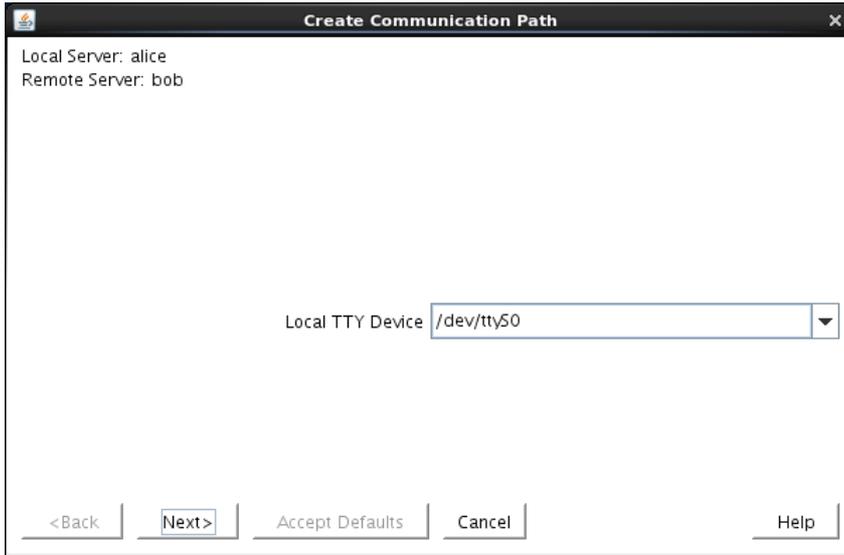


- 3) **Priority** に TCP/IP 通信によるコミュニケーションパスの優先順位を入力し、**Create** ボタンを押します。

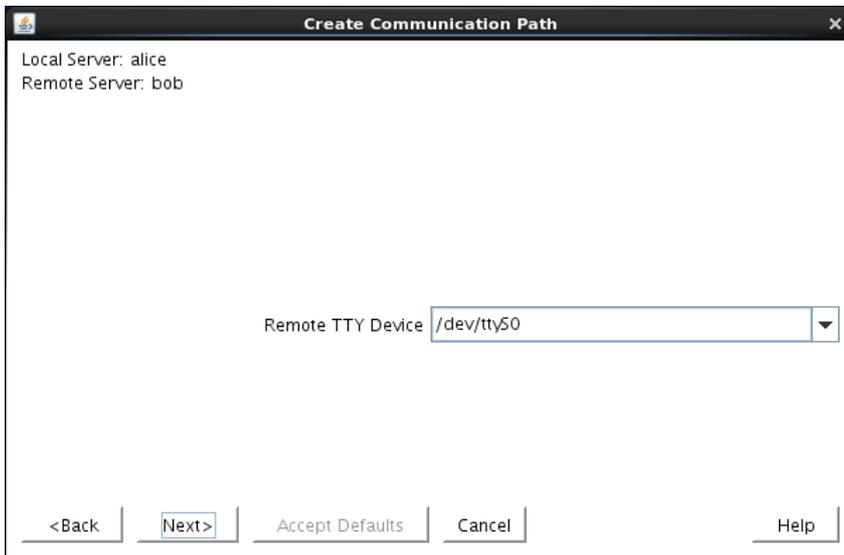


8. Device Type が TTY の場合

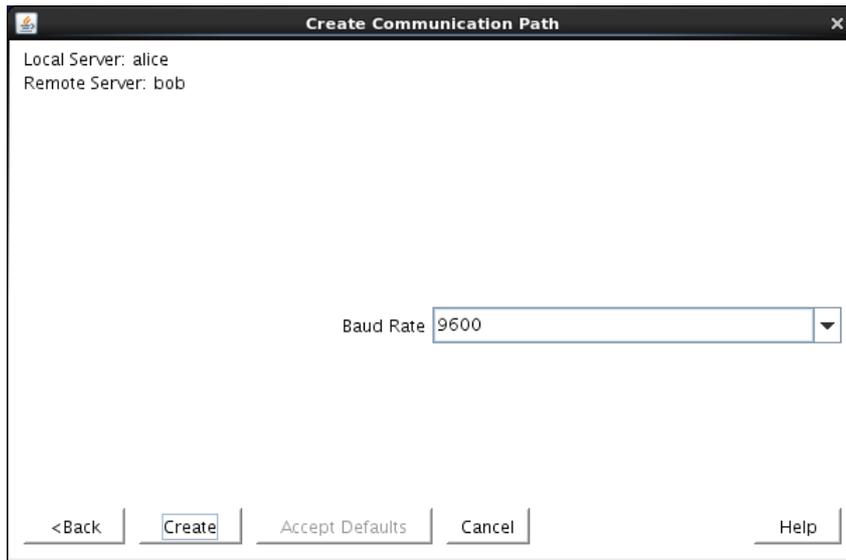
- 1) **Local TTY Device** から稼働系サーバの TTY デバイスを選択し、**Next>** ボタンを押します。



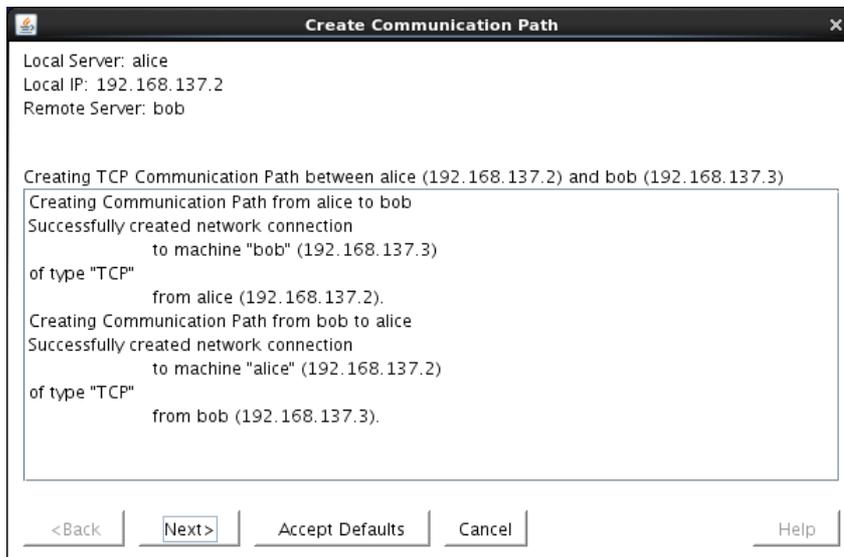
- 2) **Remote TTY Device** から待機系サーバの TTY デバイスを選択し、**Next>** ボタンを押します。



- 3) **Baud Rate** からシリアル通信のボーレート(変調速度)を選択し、**Create** ボタンを押します。

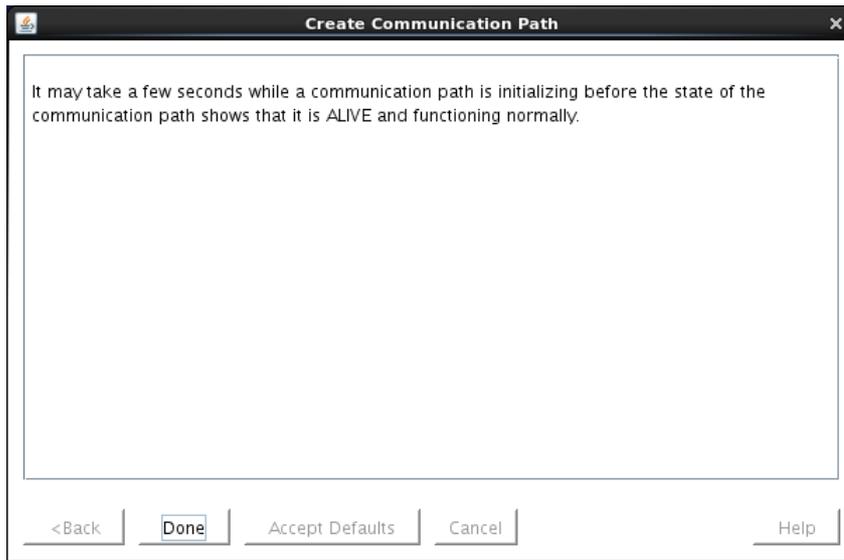


9. コミュニケーションパスが作成されます。

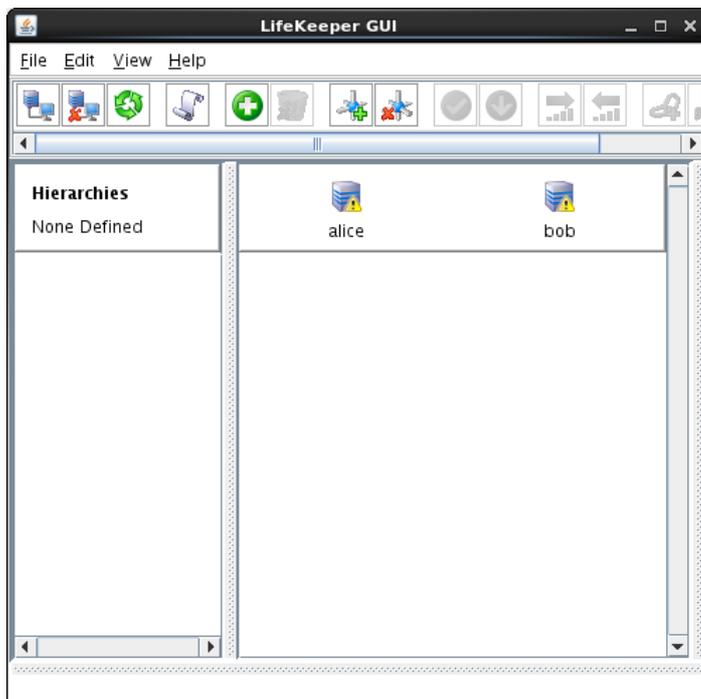


コミュニケーションパスの作成が完了したら、**Next>**ボタンを押します。

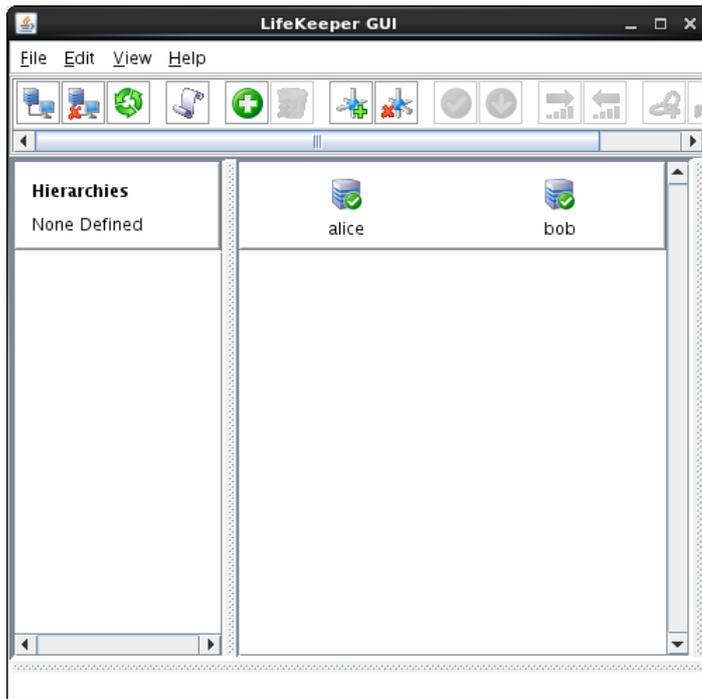
10. **Done** ボタンを押します。



11. コミュニケーションパスを1本のみ作成すると、サーバの状態が警告(🚨)になります。



1本目と同じように2本目以降のコミュニケーションパスを作成します。コミュニケーションを2本以上作成すると、サーバの状態が正常(🟢)になります。



12. **File** メニューの **Exit** を選択し、LifeKeeper GUI を終了します。

5. データレプリケーションリソース階層の作成

データレプリケーションリソース階層を作成します。詳細については「[SteelEye Protection Suite for Linux テクニカルドキュメンテーション](#)」の「[DataKeeper リソース階層の作成](#)」を参照してください。

データレプリケーションリソース階層は DataKeeper によってサーバ間でデータをレプリケーションするためのリソース階層です。

レプリケーションは PowerGres Plus のデータ領域とバックアップ領域に対して行います。データ領域にはデータベースクラスタ、バックアップ領域には二重化されたトランザクションログ、オンラインバックアップの設定が有効な場合にはベースバックアップとアーカイブログが格納されます。

片方のディスクに障害が発生してもデータを復旧できるように、データ領域とバックアップ領域は異なるディスク上に配置し、それぞれに対してデータレプリケーションリソース階層を作成することを推奨します。

ここではデータ領域用とバックアップ領域用のデータレプリケーションリソース階層を作成します。データレプリケーションリソース階層の設定は以下のとおりです。背景色が青灰色(■)の部分はデフォルトの設定のままであることを表します。

	データ領域用	バックアップ領域用
リソース階層の作成		
Please Select Recovery Kit	Data Replication	Data Replication
Switchback Type	intelligent	intelligent
Server	alice	alice
Hierarchy Type	Replicate New Filesystem	Replicate New Filesystem
Source Disk	/dev/sdb1	/dev/sdb2
New Mount Point	/mnt/disk1	/mnt/disk2
New Filesystem Type	ext3	ext3
Data Replication Resource Tag	datarep-disk1	datarep-disk2
File System Resource Tag	/mnt/disk1	/mnt/disk2
Bitmap File	/opt/LifeKeeper/bitmap__mnt_disk1	/opt/LifeKeeper/bitmap__mnt_disk2
Enable Asynchronous Replication?	no	no
リソース階層の拡張		
Target Server	bob	bob
Switchback Type	intelligent	intelligent
Template Priority	1	1
Target Priority	10	10
Target Disk	/dev/sdb1	/dev/sdb2
Data Replication Resource Tag	datarep-disk1	datarep-disk2
Bitmap File	/opt/LifeKeeper/bitmap__m	/opt/LifeKeeper/bitmap__m

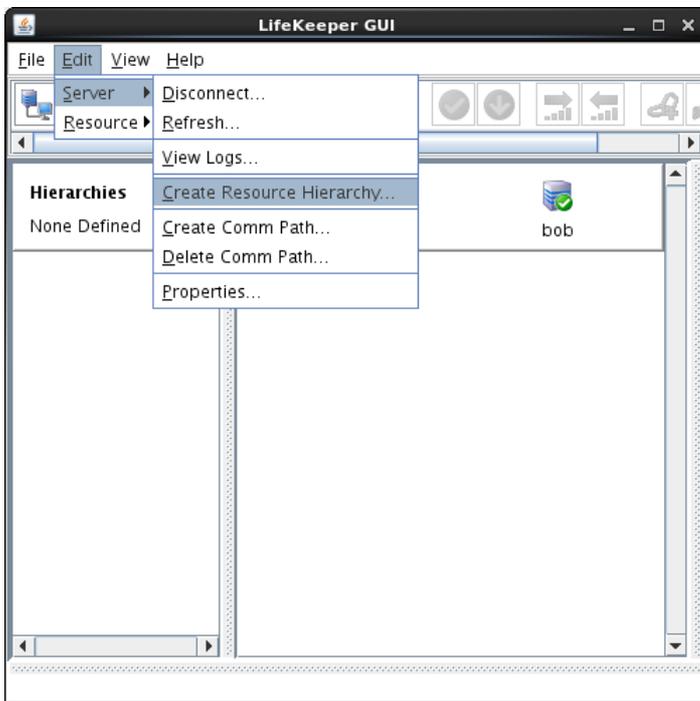
	データ領域用	バックアップ領域用
	nt_disk1	nt_disk2
Replication Path	192.168.1.2/192.168.1.3	192.168.2.2/192.168.2.3
Mount Point	/mnt/disk1	/mnt/disk2
Root Tag	/mnt/disk1	/mnt/disk2

データレプリケーションリソース階層の作成は以下の手順に従って稼働系サーバにおいて root ユーザで行います。

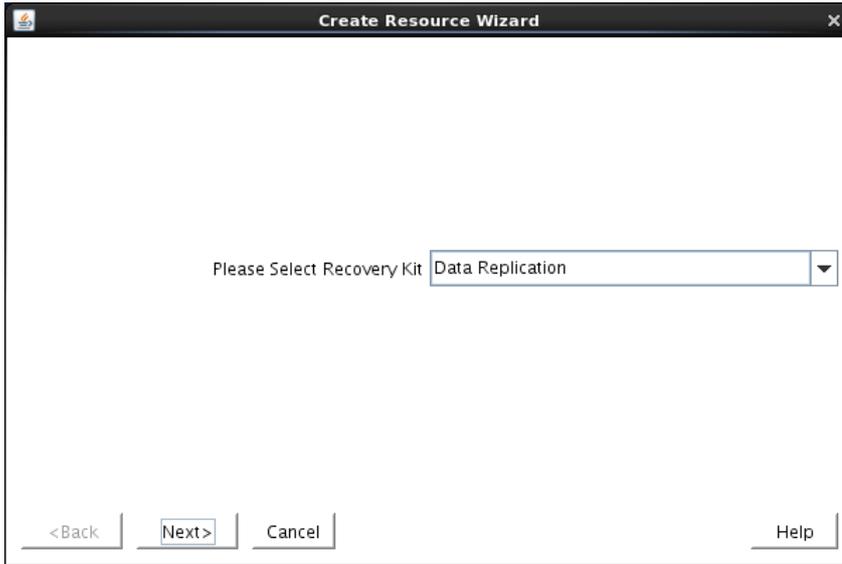
1. LifeKeeper GUI を起動し、接続します。

```
# lkGUIapp
```

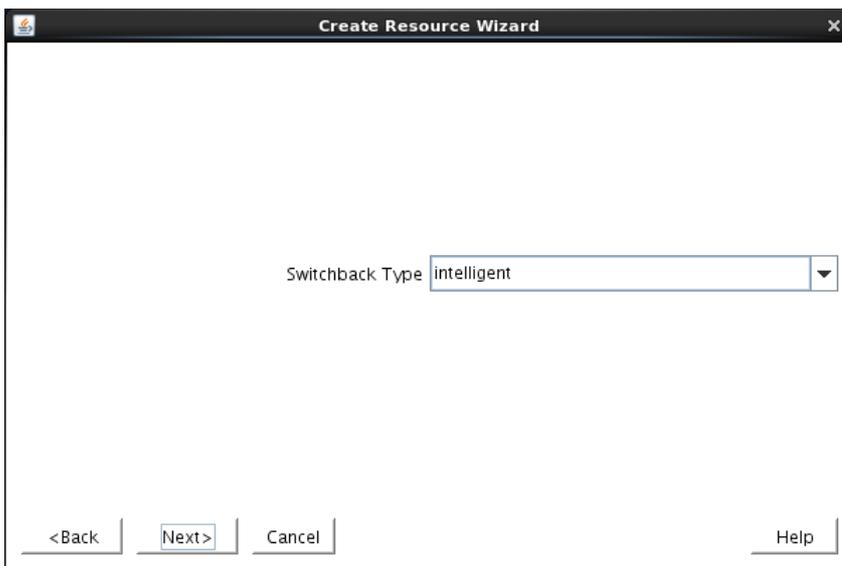
2. **Edit** メニューの **Server** ▶ **Create Resource Hierarchy...** を選択します。



3. **Please Select Recovery Kit** から Recovery Kit として Data Replication を選択し、**Next>** ボタンを押します。

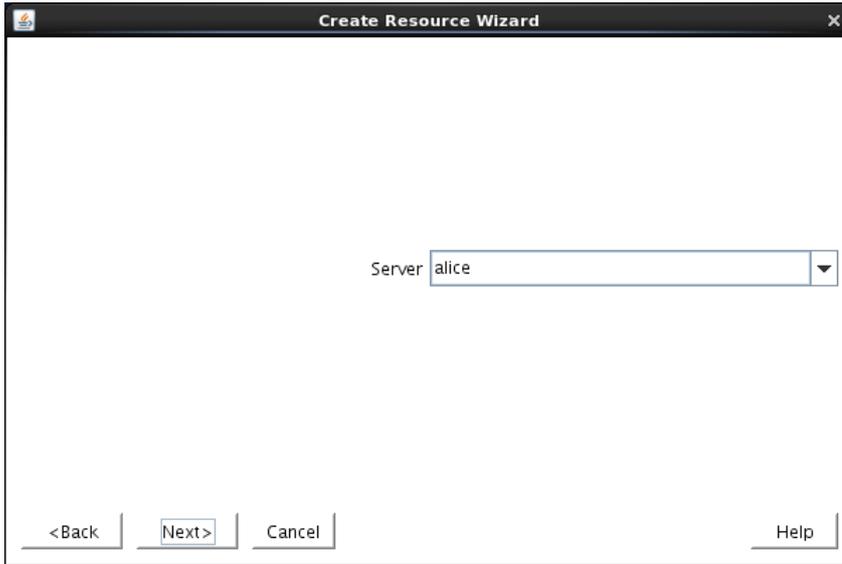


4. **Switchback Type** からスイッチバックの動作として **intelligent** または **automatic** を選択し、**Next>**ボタンを押します。

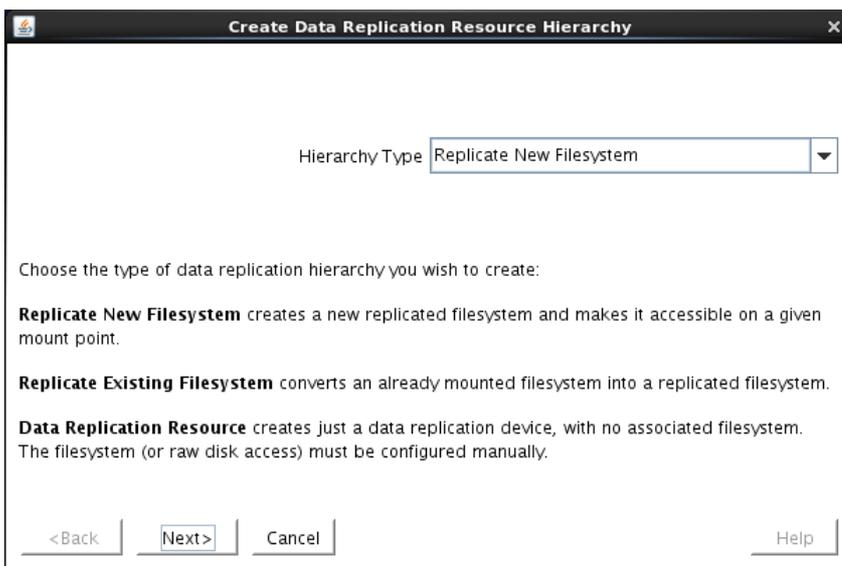


スイッチオーバーはフェイルオーバーで切り替わったリソースのサービスをもとのサーバに切り戻すことを表します。スイッチバックを手動で行う場合には **intelligent**、自動的に行う場合には **automatic** を選択します。一般的にサーバが復旧しても障害の原因が解決するまでリソースのサービスを切り戻すべきではありません。そのため、**intelligent** を選択することを推奨します。

5. **Server** からリソース階層の作成先のサーバとして稼働系サーバを選択し、**Next>**ボタンを押します。



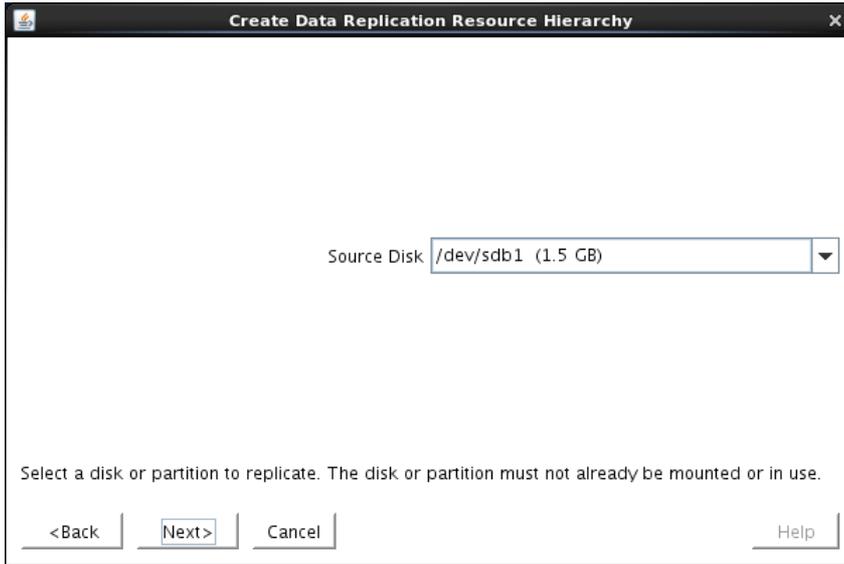
6. **Hierarchy Type** からリソース階層の作成方法として Replicate New Filesystem、Replicate Existing Filesystem または Data Replication Resource を選択し、**Next>** ボタンを押します。



レプリケーション用のファイルシステムを新規に作成する場合には Replicate New Filesystem、マウント済みのファイルシステムをレプリケーション用に変換する場合には Replicate Existing Filesystem を選択します。

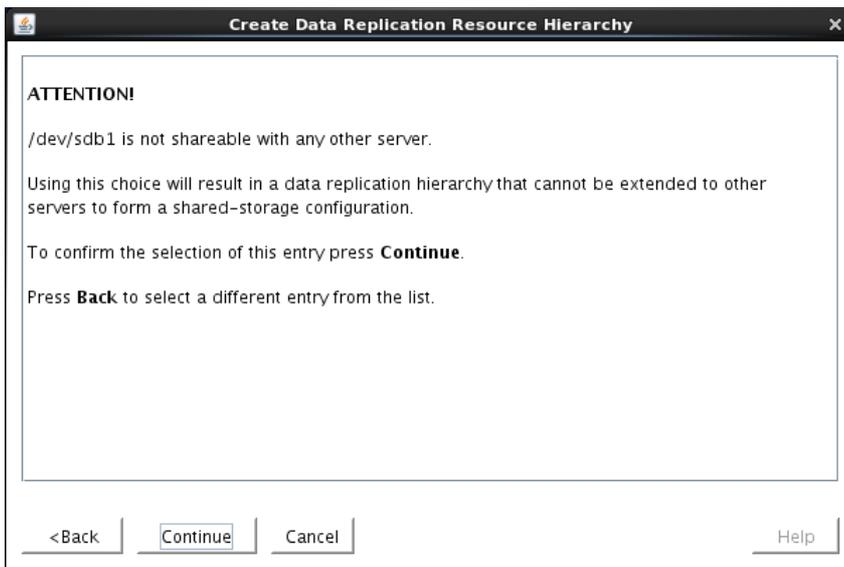
ファイルシステムを作成せずにレプリケーション用のデバイスのみを作成する場合には Data Replication Resource を選択します。

7. **Hierarchy Type** が Replicate New Filesystem の場合
- 1) **Source Disk** からレプリケーション元のディスクを選択し、**Next>** ボタンを押します。

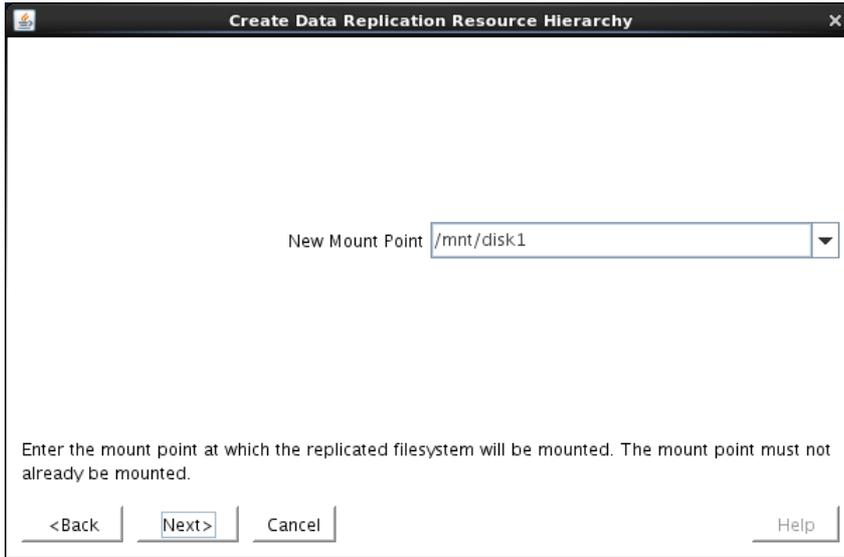


マウント済みのディスクは **Source Disk** に表示されません。選択したディスク上のデータはすべて削除されます。

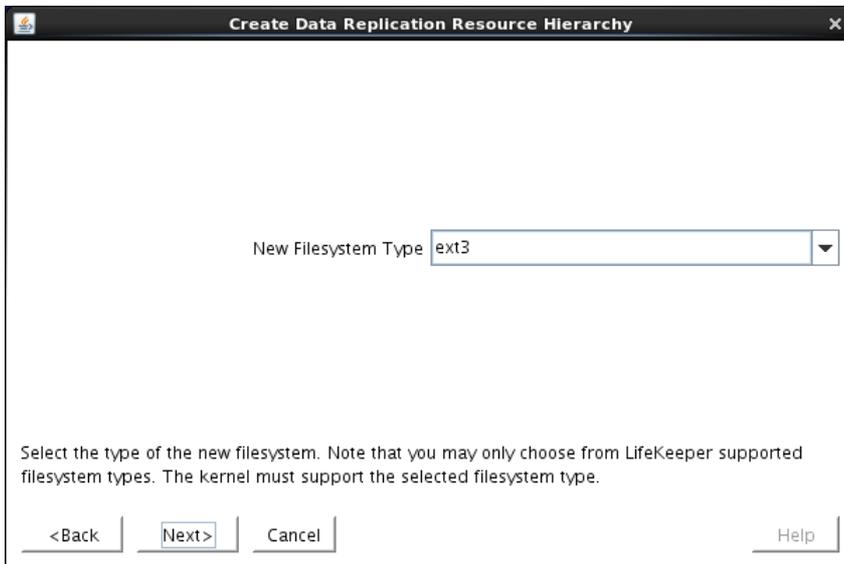
- 2) レプリケーション元のディスクが共有ディスクでない場合には、共有ディスク用にリソース階層を拡張できないという注意が表示されます。マルチサイトクラスタでない場合には無視して問題ありません。**Continue** を押します。



- 3) **New Mount Point** にマウント先のディレクトリのパスを入力し、**Next>** ボタンを押します。

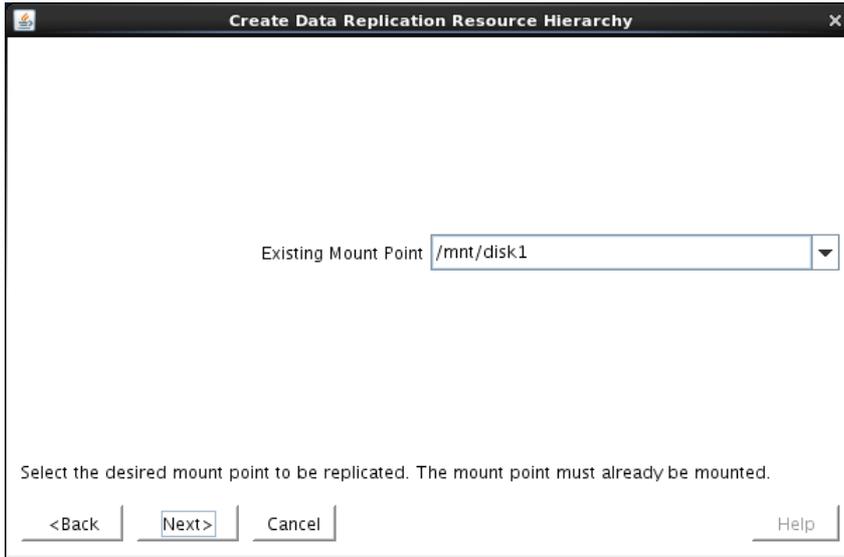


- 4) **New Filesystem Type** からファイルシステムタイプを選択し、**Next>**ボタンを押します。

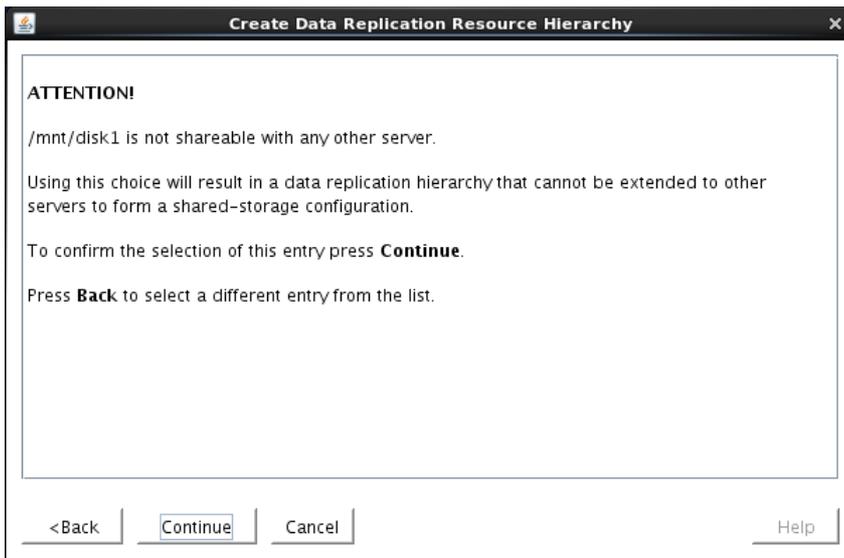


8. **Hierarchy Type** が **Replicate Existing Filesystem** の場合

- 1) **Existing Mount Point** からレプリケーション用に変換するファイルシステムのマウントされたディレクトリを選択します。



- 2) レプリケーション用に変換するファイルシステムの作成されたディスクが共有ディスクでない場合には、共有ディスク用にリソース階層を拡張できないという注意が表示されます。マルチサイトクラスタでない場合には無視して問題ありません。**Continue** ボタンを押します。



9. **Data Replication Resource Tag** にデータレプリケーションリソースのタグ名を入力し、**Next>** ボタンを押します。



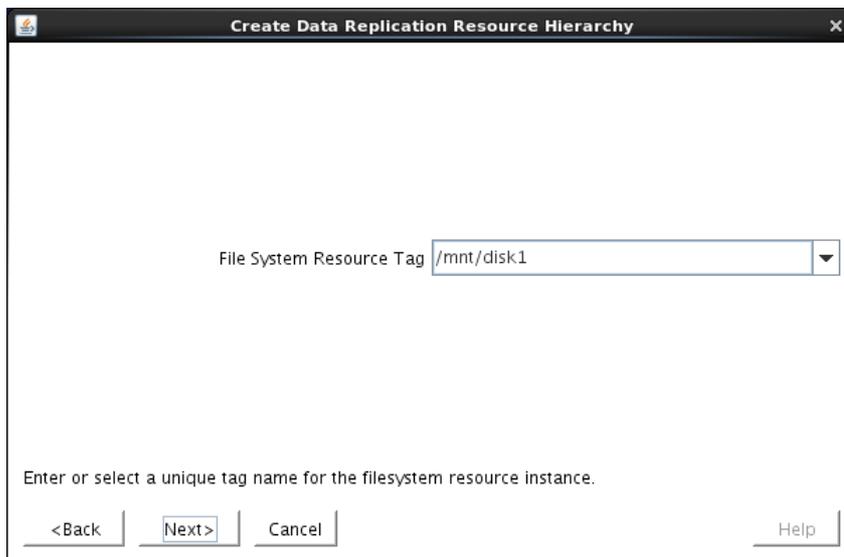
Create Data Replication Resource Hierarchy

Data Replication Resource Tag

Enter or select a unique tag name for the data replication resource instance.

<Back Next> Cancel Help

10. **File System Resource Tag** にファイルシステムリソースのタグ名を入力し、**Next>**ボタンを押します。



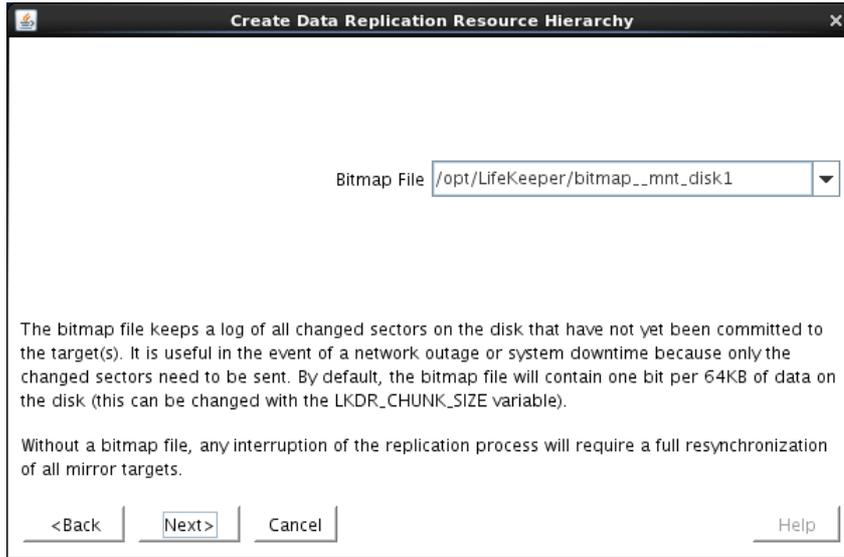
Create Data Replication Resource Hierarchy

File System Resource Tag

Enter or select a unique tag name for the filesystem resource instance.

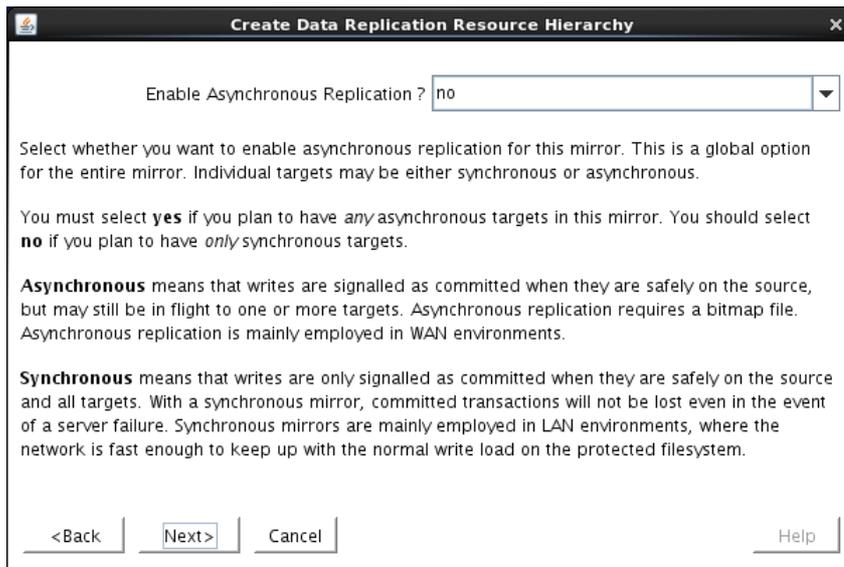
<Back Next> Cancel Help

11. **Bitmap File** にビットマップファイルのパスを入力し、**Next>**ボタンを押します。



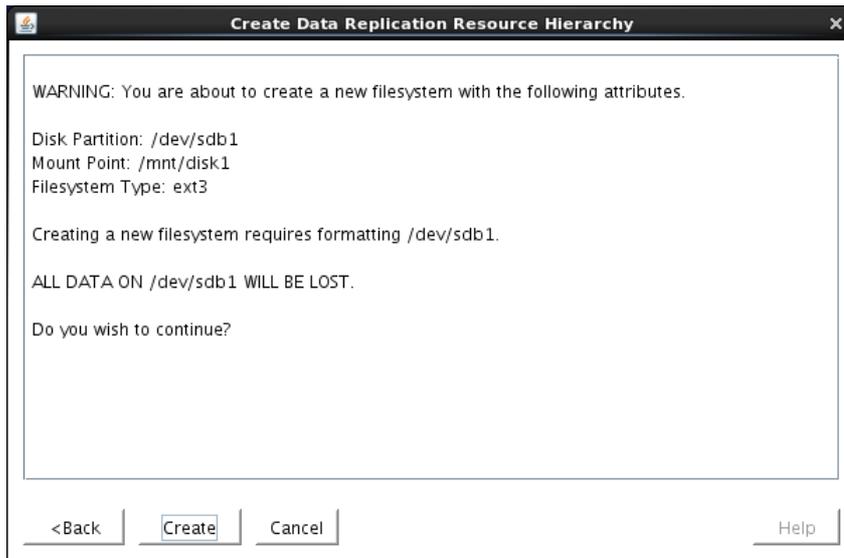
ビットマップファイルはサーバ間で同期されていないデータブロックを記録するファイルです。

12. **Enable Asynchronous Replication?**から非同期レプリケーションを有効にするかの質問に対して **no** または **yes** を選択し、**Next>** ボタンを押します。

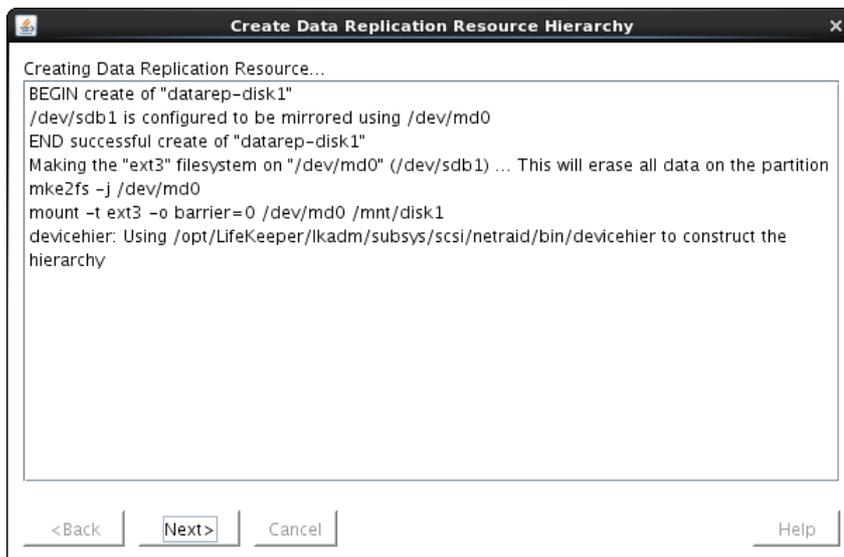


非同期レプリケーションはレプリケーション先のディスクへの書き込みが完了するまで待機しないことを表します。非同期レプリケーションを有効にすると、性能が向上する代わりに信頼性が低下します。そのため、**no** を選択することを推奨します。

13. **Hierarchy Type** が **Replicate New Filesystem** の場合には、選択したディスク上のデータがすべて削除されるという警告が表示されます。問題ない場合には **Create** ボタンを押します。

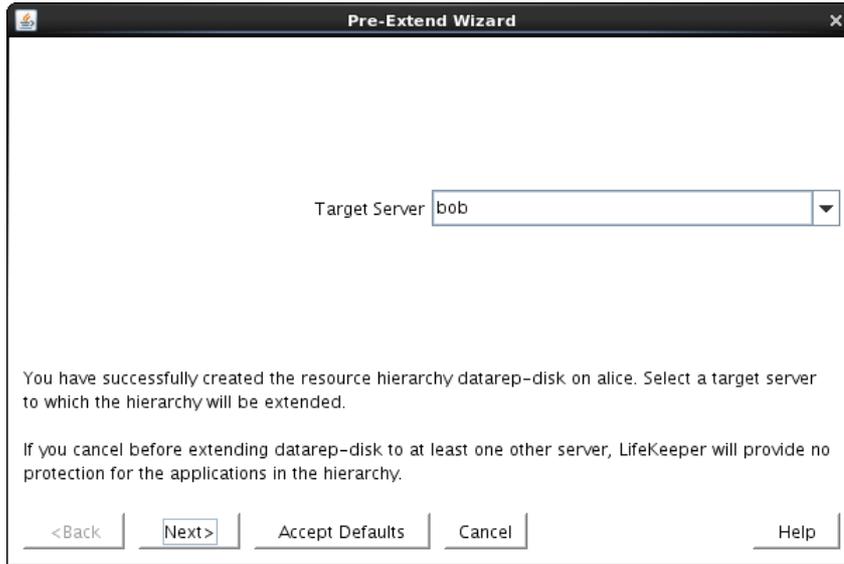


14. リソース階層が作成されます。



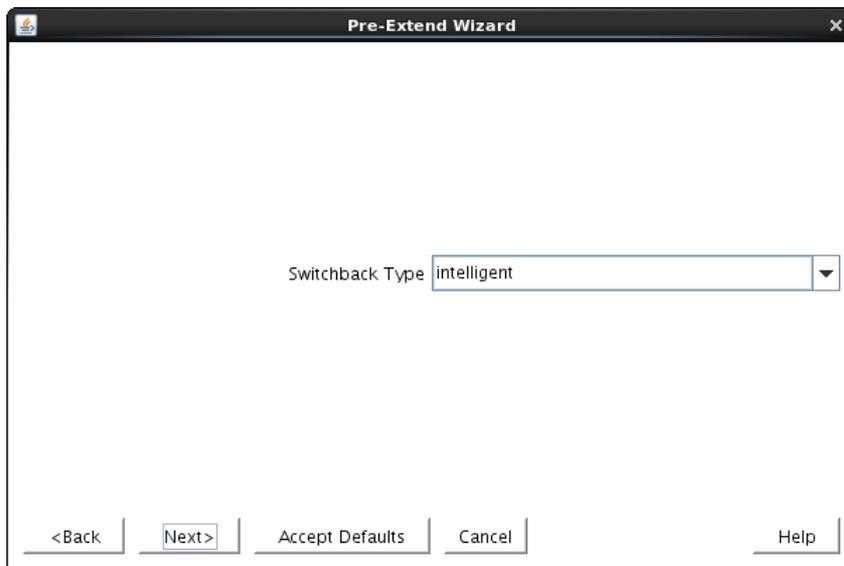
リソース階層の作成が完了したら、**Next>**ボタンを押します。

15. **Target Server** からリソース階層の拡張先のサーバとして待機系サーバを選択します。

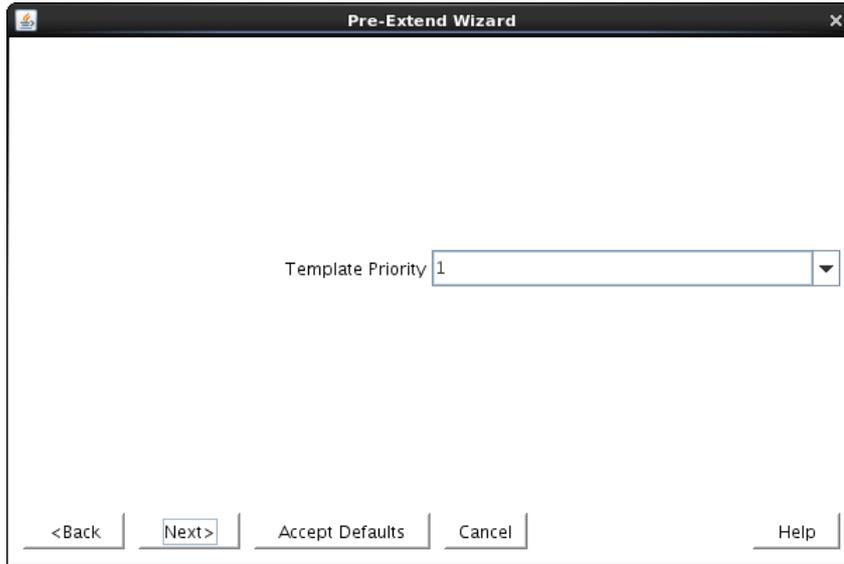


これ以降の設定を確認せずにデフォルトの設定のままで問題ない場合には **Accept Defaults** ボタンを押します。そうでない場合には **Next>** ボタンを押します。ここでは **Next>** ボタンを押した場合について説明します。

16. **Switchback Type** からスイッチバックの動作として **intelligent** または **automatic** を選択し、**Next>** ボタンを押します。

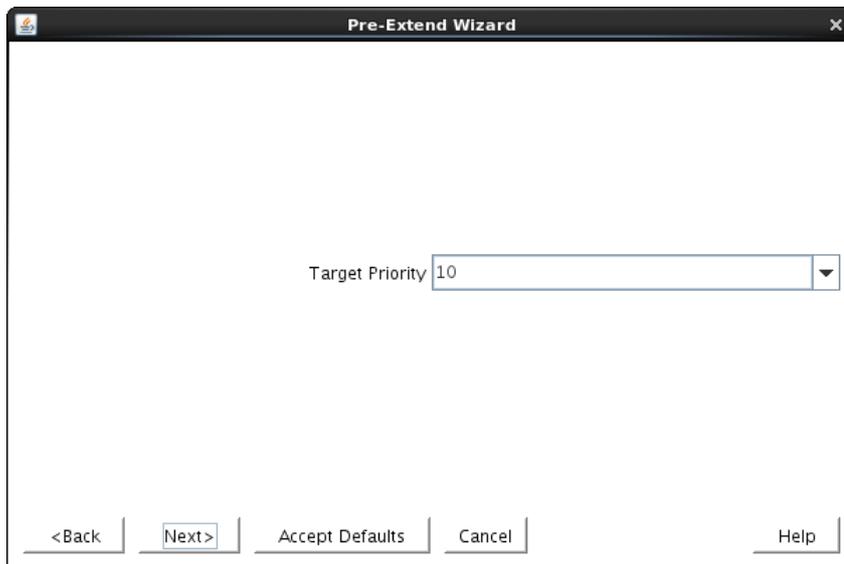


17. **Template Priority** に拡張元のリソース階層の優先順位を入力し、**Next>** ボタンを押します。



The screenshot shows a window titled "Pre-Extend Wizard". Inside the window, there is a text label "Template Priority" followed by a text input field containing the number "1". Below the input field is a small downward-pointing arrow. At the bottom of the window, there are five buttons: "<Back", "Next>", "Accept Defaults", "Cancel", and "Help". The "Next>" button is highlighted with a blue border.

18. **Target Priority** に拡張先のリソース階層の優先順位を入力し、**Next>**ボタンを押します。



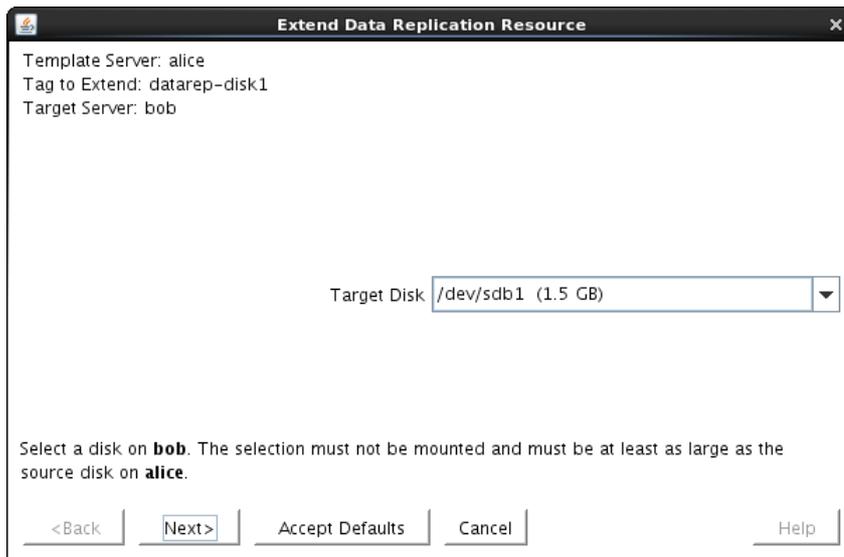
The screenshot shows the same "Pre-Extend Wizard" window. The text label is now "Target Priority" and the input field contains the number "10". The "Next>" button remains highlighted.

19. 拡張元のリソース階層が検査されます。



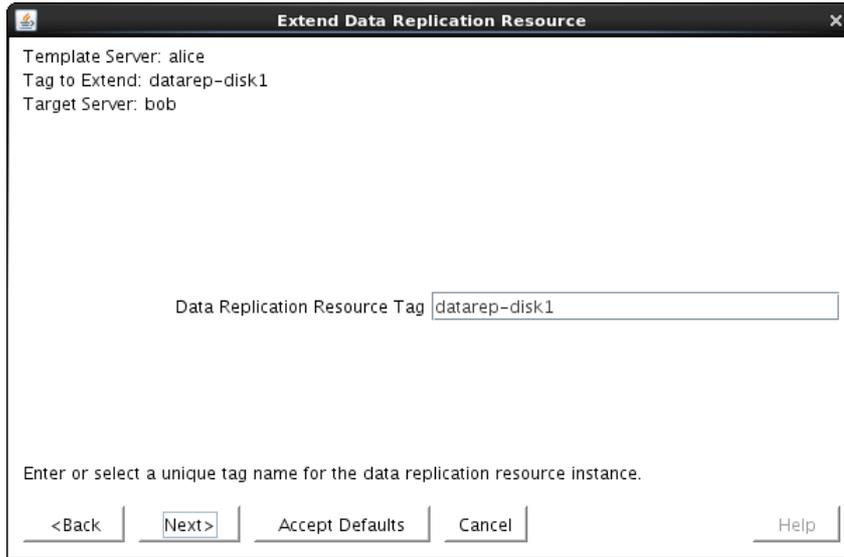
拡張元のリソース階層の検査が完了したら、**Next>**ボタンを押します。

20. **Target Disk** からレプリケーション先のディスクを選択し、**Next>**ボタンを押します。



マウント済みのディスクは **Target Disk** に表示されません。ディスクのサイズはレプリケーション元のディスク以上の必要があります。選択したディスク上のデータはすべて削除されます。

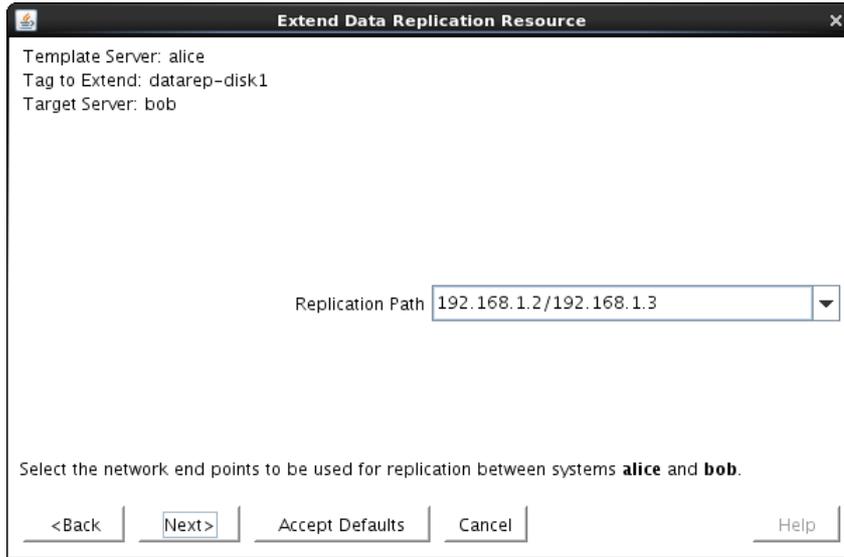
21. **Data Replication Resource Tag** にデータレプリケーションリソースのタグ名を入力し、**Next>**ボタンを押します。



22. **Bitmap File** にビットマップファイルのパスを入力し、**Next>**ボタンを押します

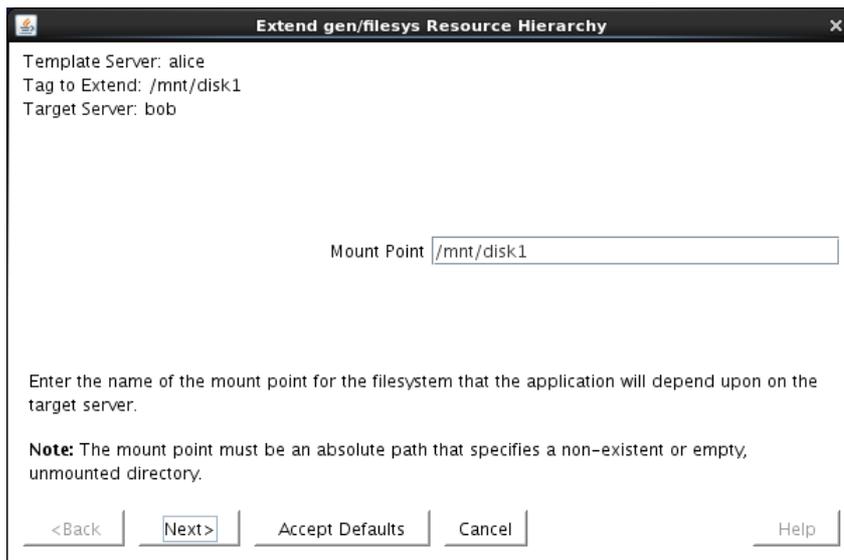


23. **Replication Path** からレプリケーションパスを選択し、**Next>**ボタンを押します。



レプリケーションパスはデータのレプリケーションに使用する通信経路です。コミュニケーションパスの中から選択します。専用のコミュニケーションパスを選択することを推奨します。

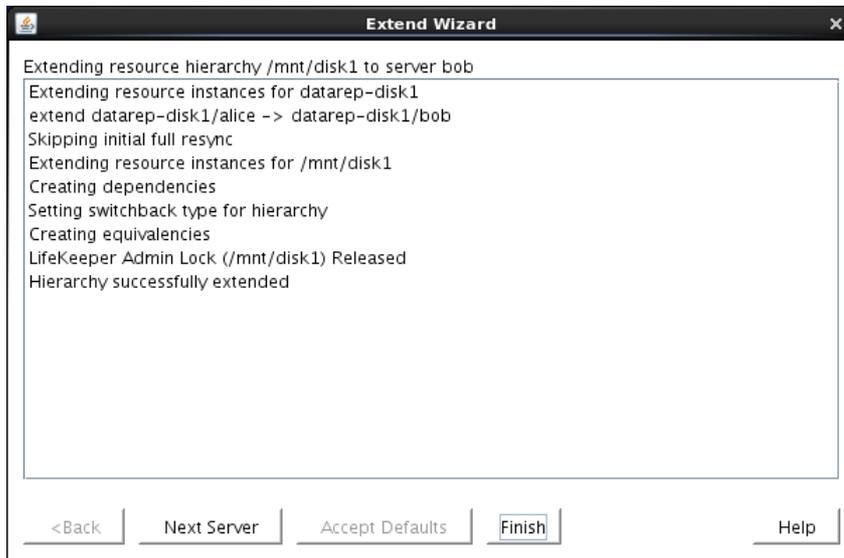
24. **Mount Point** にマウント先のディレクトリのパスを入力し、**Next>** ボタンを押します。



25. **Root Tag** にファイルシステムリソースのタグ名を入力し、**Next>** ボタンを押します。

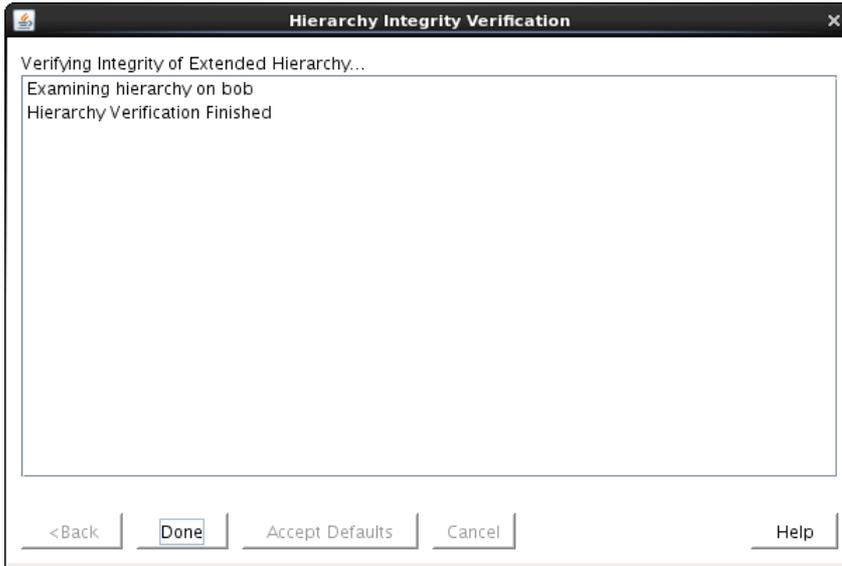


26. リソース階層が拡張されます。

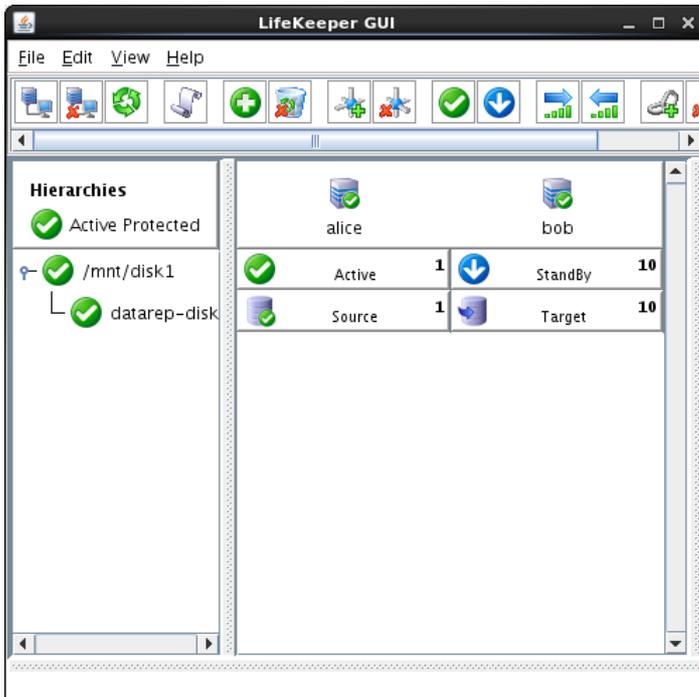


リソース階層の拡張が完了したら、**Finish** ボタンを押します。

27. **Done** ボタンを押します。

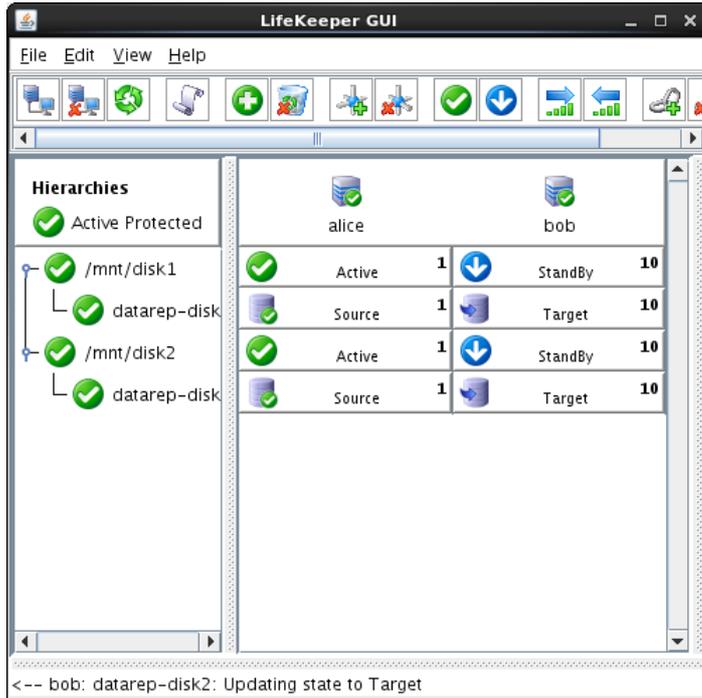


28. リソース階層ツリーにデータレプリケーションリソース階層が追加されます。



データレプリケーションリソース階層の状態は稼働系サーバがサービス起動中(☑Active)、待機系サーバがサービス停止中(⬇StandBy)になります。また、データレプリケーションリソースの状態は稼働系サーバがレプリケーション元(📁Source)、待機系サーバがレプリケーション先(📁Target)になります。

データ領域用と同じようにバックアップ領域用のデータレプリケーションリソース階層を作成します。



29. **File** メニューの **Exit** を選択し、LifeKeeper GUI を終了します。

6. PowerGres Plus の設定

PowerGres Plus の設定では、データベースクラスタの作成、パスワードファイルの作成、クライアント認証の設定、オンラインバックアップの設定、データ暗号化の設定を行います。

6.1. データベースクラスタの作成

データベースクラスタを作成します。詳細については「[PowerGres Plus Manager マニュアル](#)」の「[データベースクラスタの新規作成](#)」を参照してください。

データベースクラスタは PowerGres Plus のデータを格納する領域です。データベースクラスタはデータベースディレクトリに作成します。また、データベースクラスタの作成時にバックアップディレクトリを設定を行います。バックアップディレクトリは PowerGres Plus のバックアップの格納先のディレクトリです。データベースディレクトリとバックアップディレクトリはデータレプリケーションリソース階層でレプリケーションされたディスク上に作成します。

ここでは、データ領域用のディスク上にデータベースディレクトリ、バックアップ領域用のディスク上にバックアップディレクトリを配置し、データベースクラスタを作成します。データベースクラスタの設定は以下のとおりです。背景色が青灰色(■)の部分はデフォルトの設定のままであることを表します。

ラベル	powergresplus
ポート	5432
サーバの所在	ローカルのサーバを管理
データベースディレクトリ	/mnt/disk1/data
バックアップディレクトリ	/mnt/disk2/backups
サーバの種類	新規データベースクラスタを作成
文字エンコーディング	UTF8
スーパーユーザ名	postgres
パスワード	password
パラメータチューニングも実施する	(選択しない)

データベースクラスタの作成は以下の手順に従って稼働系サーバで行います。

1. root ユーザでデータベースディレクトリとバックアップディレクトリを作成し、postgres ユーザのみがアクセスできるようにアクセス権と所有者の設定を行います。

```
# mkdir /mnt/disk1/data
# chmod 0700 /mnt/disk1/data
# chown postgres:postgres /mnt/disk1/data
# mkdir /mnt/disk2/backups
# chmod 0700 /mnt/disk2/backups
# chown postgres:postgres /mnt/disk2/backups
```

2. postgres ユーザで PowerGres Plus Manager を起動します。

```
$ powergres-mgr
```

3. サーバメニューのサーバを登録...を選択します。



- ラベルにラベル名、ポートにデータベースサーバのポート番号を入力し、ローカルのサーバを管理を選択し、進むボタンを押します。



- データベースディレクトリにデータベースディレクトリのパス、バックアップディレクトリにバックアップディレクトリのパスを入力し、新規データベースクラスタを作成を選択し、進むボタンを押します。

サーバを登録

新規サーバ登録

データベースディレクトリ:
/mnt/disk1/data 選択...

バックアップディレクトリ:
/mnt/disk2/backups 選択...

サーバの種類を選択します。

新規データベースクラスタを作成
 既存のデータベースクラスタを使用

戻る(B) 進む(E) キャンセル(C)

6. 文字エンコーディングにデフォルトの文字エンコーディング、スーパーユーザ名に PowerGres Plus のスーパーユーザ名、パスワードにスーパーユーザのパスワードを入力し、進むボタンを押します。

サーバを登録

新規サーバ登録

新規に作成するデータベースクラスタの情報を入力します。

文字エンコーディング: UTF8

スーパーユーザ名: postgres

パスワード: パスワード(再確認):

パラメータチューニングも実施する

チューニング

合計メモリ容量(MB) 988 最大接続数: 100

データベースの用途: Web OLTP データウェアハウス

戻る(B) 進む(E) キャンセル(C)

7. OK ボタンを押すと、データベースクラスタが作成されます。

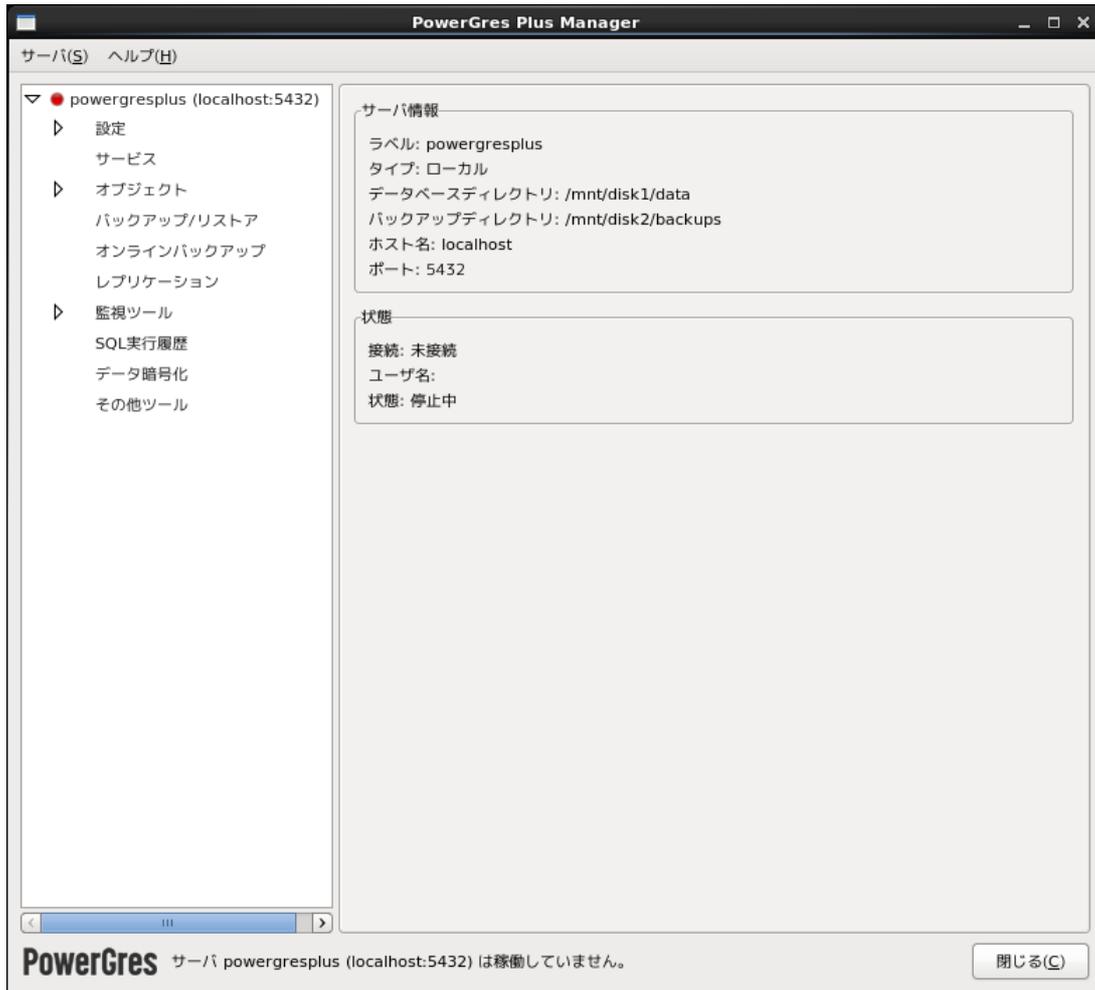
サーバを登録

新規サーバ登録

OKボタンを押して登録を完了します。

戻る(B) OK(O) キャンセル(C)

8. ツリーメニューにサーバが追加されます。



パラメータの設定を変更する場合にはツリーメニューの **設定 ▶ 基本(postgresql.conf)** を選択します。詳細については「[PowerGres Plus Manager マニュアル](#)」の「[基本\(postgresql.conf\)](#)」を参照してください。なお、ログファイルに出力されるメッセージのロケールを指定する `lc_messages` パラメータは C 以外に設定しないでください。

9. **サーバメニューの終了**を選択し、PowerGres Plus Manager を終了します。
10. フェイルオーバー後に待機系サーバでも PowerGres Plus Manager を起動できるように、PowerGres Plus Manager の設定ファイルを待機系サーバにコピーします。

```
$ rsync -av ~/.powergresplus91 bob:~
postgres@bob's password: (パスワードを入力)
sending incremental file list
.powergresplus91/
.powergresplus91/cron.txt
.powergresplus91/powergres-mgr91.ini
.powergresplus91/powergres-mgr91.ini.bak
.powergresplus91/powergres-mgr91.log

sent 1778 bytes  received 92 bytes  534.29 bytes/sec
total size is 1452  speedup is 0.78
```

6.2. パスワードファイルの作成

パスワードファイルを作成します。詳細については「[PostgreSQL 文書](#)」の「[31.14. パスワードファイル](#)」を参照してください。

PowerGres Recovery Kit では、データベースサーバが正常に起動していることを監視するため、ローカルホストから template1 データベースへの接続を試みます。クライアント認証の設定でデータベースサーバへの接続時にパスワードの入力が求められると監視に失敗してしまいます。そのため、パスワードを入力せずにデータベースサーバに接続できるようにパスワードファイルを作成します。

パスワードファイルの作成は以下の手順に従って postgres ユーザで行います。

- 稼働系サーバでパスワードファイル `~/ .pgpass` を作成し、所有者のみが読み取れるようにアクセス権の設定を行います。

```
$ touch ~/ .pgpass
$ chmod 0600 ~/ .pgpass
```

- パスワードファイルに以下の内容を記述します。

```
# hostname:port:database:username:password
localhost:5432:template1:postgres:password
```

パスワードファイルはホスト名、ポート番号、データベース名、ユーザ名、パスワードを `:` で区切って記述します。データベースサーバへの接続時に一致するエントリが存在すると指定したパスワードで認証されます。ポート番号、ユーザ名およびパスワードはデータベースクラスタの作成時に指定したものを記述します。

- フェイルオーバー後に待機系サーバでもパスワードを入力せずにデータベースサーバに接続できるように、パスワードファイルを待機系サーバにコピーします。

```
$ scp ~/ .pgpass bob:~
postgres@bob's password:
. pgpass                                100%   86    0.1KB/s   00:00
```

6.3. クライアント認証の設定

クライアント認証の設定を行います。詳細については「[PowerGres Plus Manager マニュアル](#)」の「[接続認証 \(pg_hba.conf\)](#)」を参照してください。

クライアント認証はクライアントからデータベースサーバへの接続を認証する機能です。デフォルトではローカルホストからの接続を MD5 暗号化パスワードで認証するように設定されています。リモートホストからの接続を許可するにはクライアント認証の設定にエントリを追加します。

ここではネットワークアドレス `192.168.137.0/24` からの接続を MD5 暗号化パスワードで認証するように設定を行います。クライアント認証の設定は以下のとおりです。

接続タイプ	host
データベース名	all
ユーザ名	all
アドレス	192.168.137.0/24
認証方法	md5

オプション

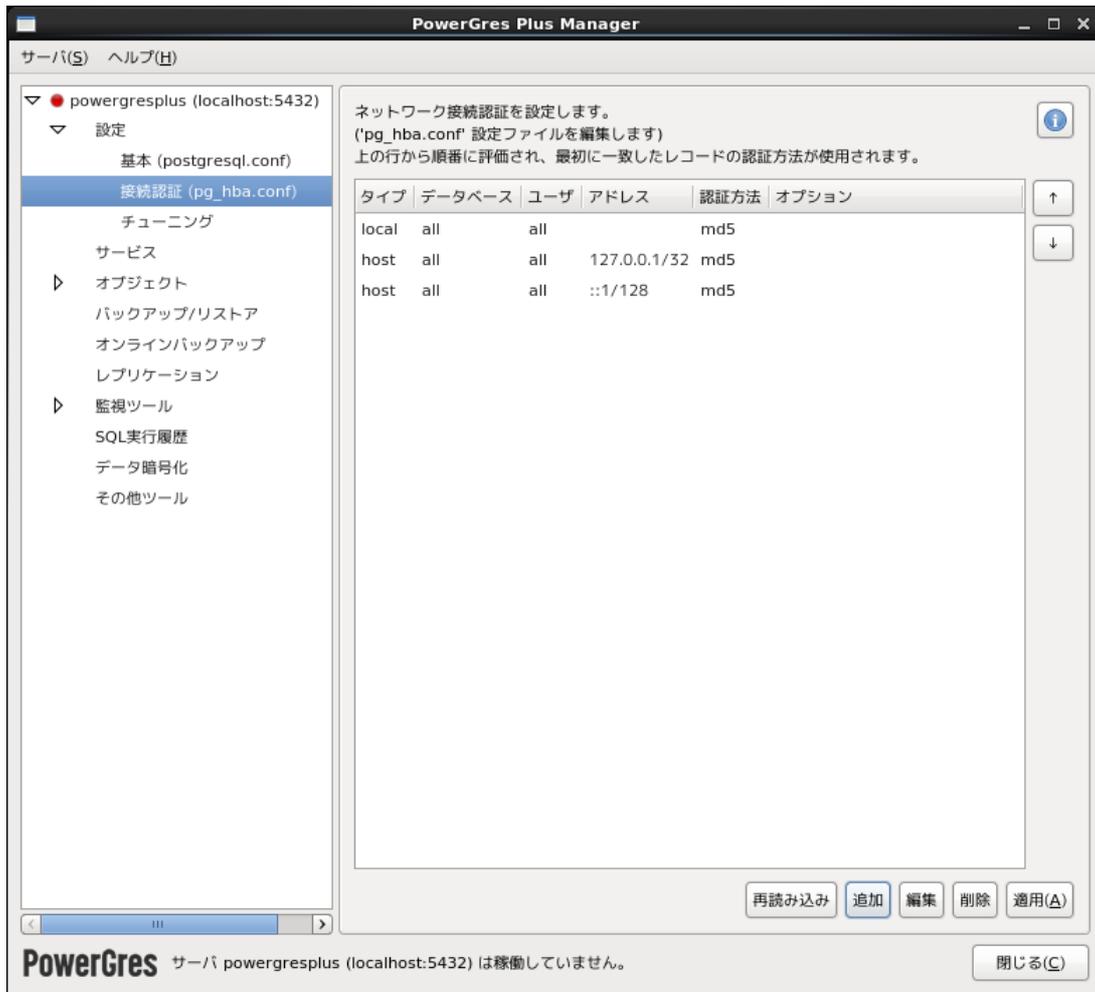
(入力しない)

クライアント認証の設定は以下の手順に従って postgres ユーザで行います。

- 稼働系サーバで PowerGres Plus Manager を起動します。

```
$ powergres-mgr
```

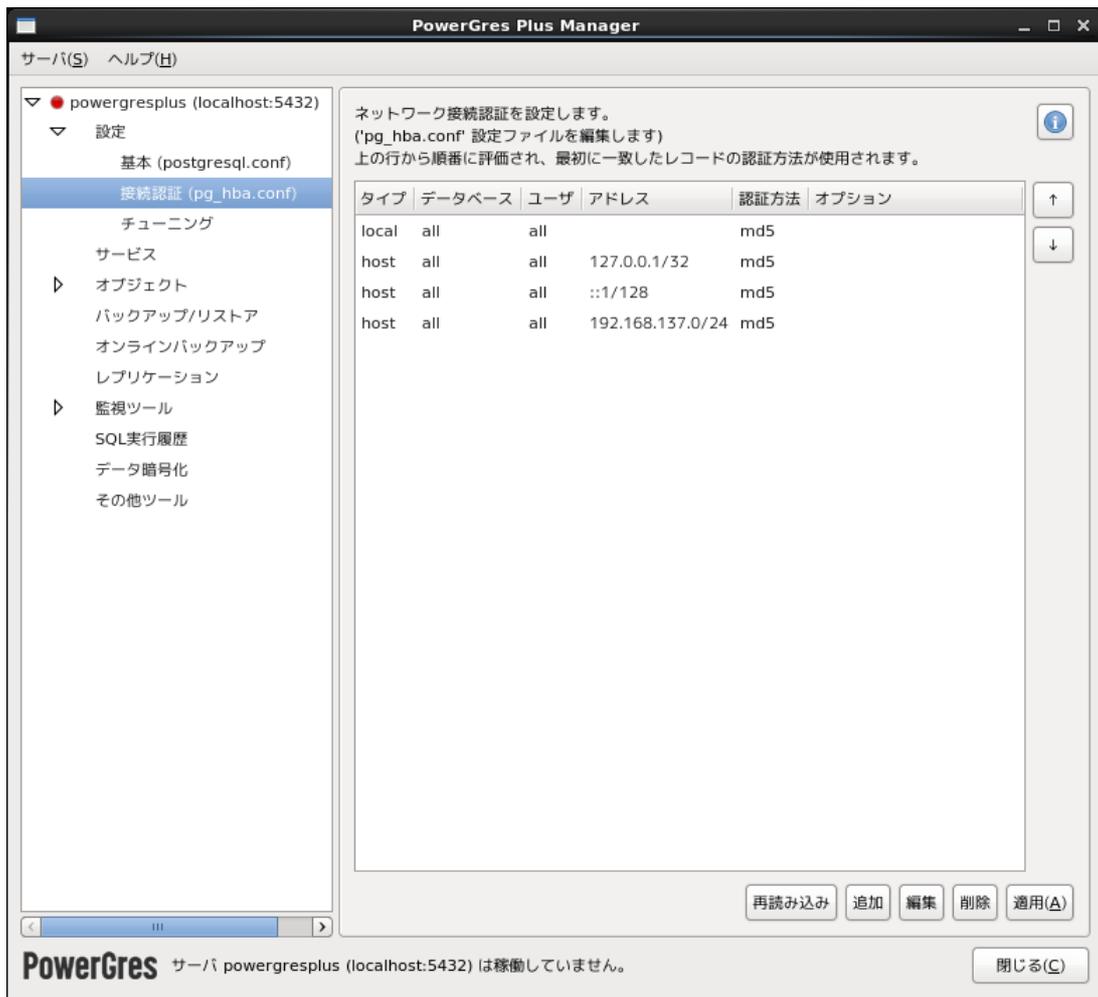
- ツリーメニューの **設定** ▶ **接続認証 (pg_hba.conf)** を選択し、**追加** ボタンを押します。



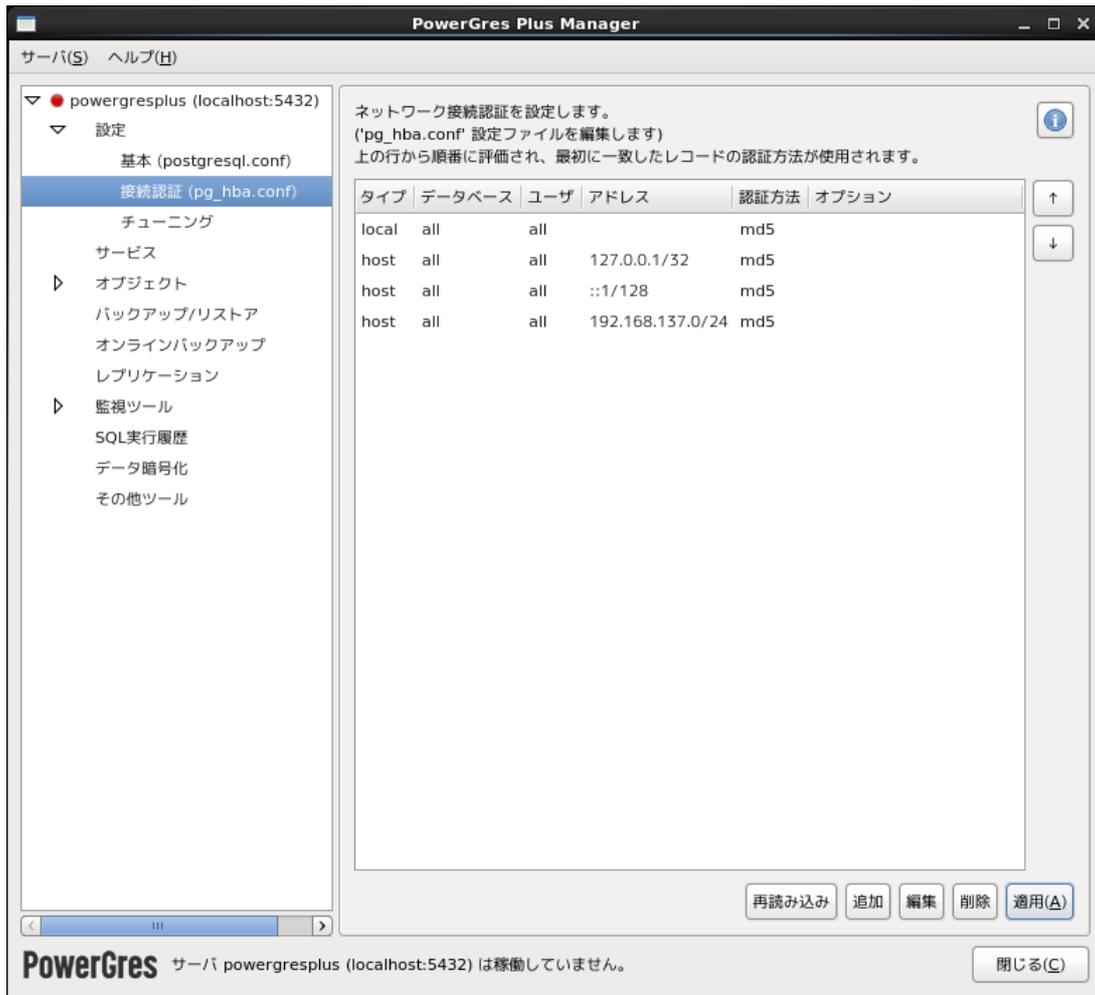
- 接続タイプ**から接続タイプとして local または host、hostssl、hostnossl を選択します。次に **データベース名**にデータベース名、**ユーザ名**にユーザ名を入力します。**接続タイプ**が local でない場合には **アドレス**に接続元の IP アドレスの範囲を入力します。**認証方法**から認証方法を選択します。最後に **OK** ボタンを押します。



クライアント認証の設定の最後にエントリが追加されます。



4. 適用ボタンを押します。



クライアント認証の設定をファイルに保存するかを確認するダイアログが表示されます。OK ボタンを押します。



5. サーバメニューの終了を選択し、PowerGres Plus Manager を終了します。

6.4. オンラインバックアップの設定

オンラインバックアップの設定を行います。詳細については「[PowerGres Plus Manager マニュアル](#)」の「[オンラインバックアップ](#)」を参照してください。

オンラインバックアップは、データベースクラスタのスナップショットをベースバックアップとして保存し、データの変更で発生するトランザクションログをアーカイブログとして保存していくバックアップ方式です。データの変更が随時バックアップされるため、障害が発生しても直前の状態までデータを復旧できます。

PowerGres Plus ではオンラインバックアップの設定が有効になっています。そのため、定期的にベースバック

アップを作成して不要なアーカイブログを削除する必要があります。

オンラインバックアップの機能を使用しない場合にはアーカイブモードを無効にします。アーカイブモードを無効にするには、PowerGres Plus Manager を起動し、ツリーメニューの**オンラインバックアップ**を選択し、**全般タブ**を開きます。次に**アーカイブモードを有効にする**のチェックを外し、**適用ボタン**を押します。最後にサービスが起動している場合にはサービスを再起動します。オンラインバックアップの機能を使用する場合にはベースバックアップの自動作成の設定を行うことを推奨します。

ここでは毎日 0:00 に自動的にベースバックアップを作成するように設定を行います。ベースバックアップの自動作成の設定は以下のとおりです。

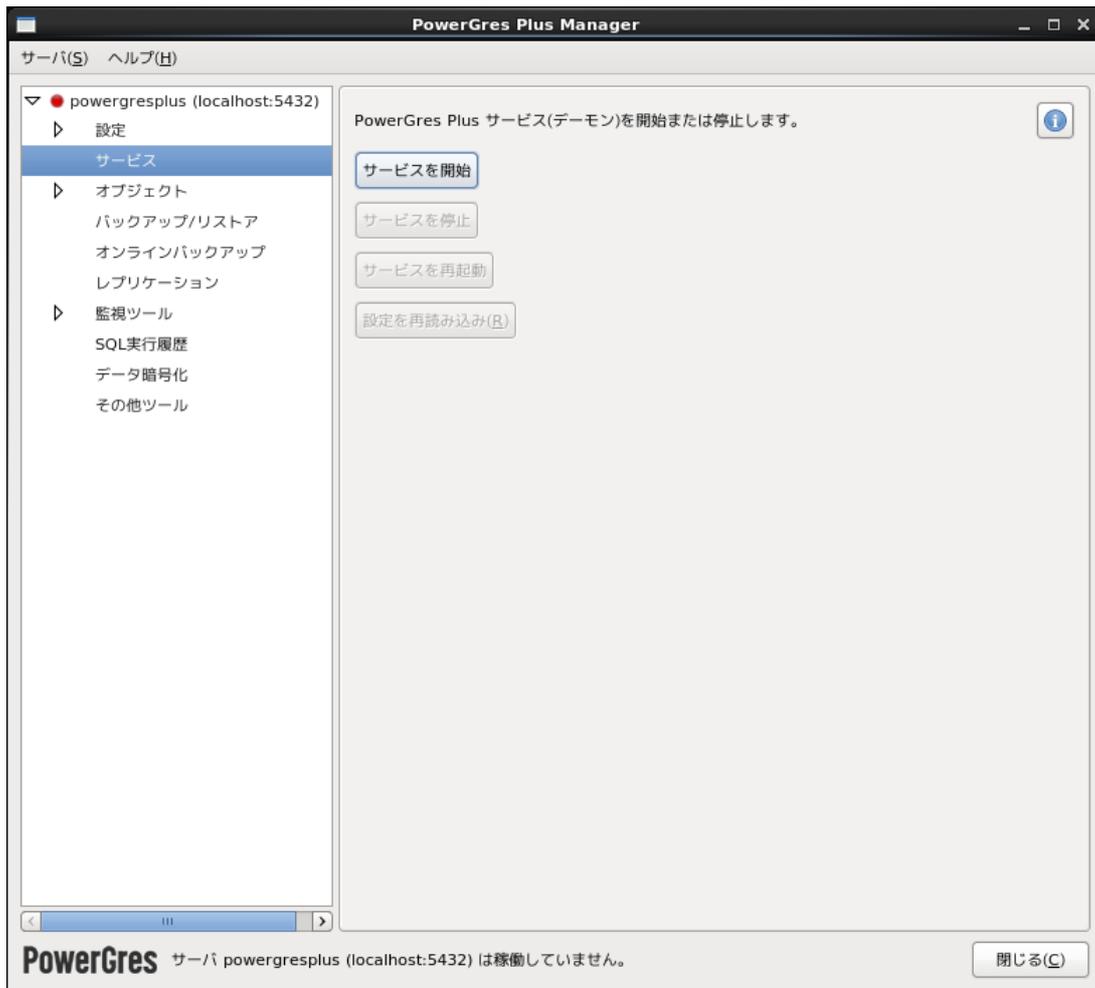
バックアップ設定ファイル	/var/lib/pgsql/basebackup.sh
バックアップ頻度	毎日
時刻	0:00

ベースバックアップの自動作成の設定は以下の手順に従って postgres ユーザで行います。

- 稼働系サーバで PowerGres Plus Manager を起動します。

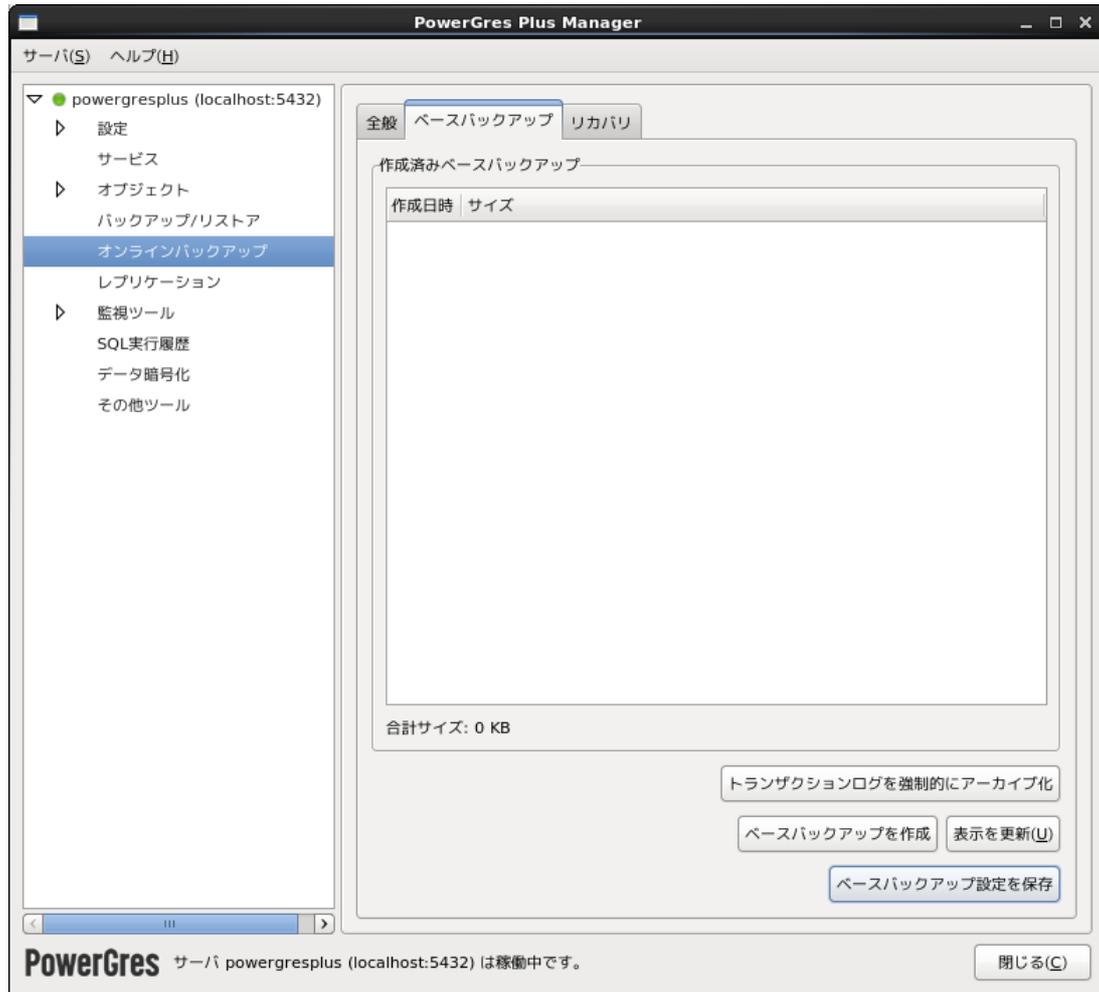
```
$ powergres-mgr
```

- ツリーメニューの**サービス**を選択し、**サービスを開始**ボタンを押します。

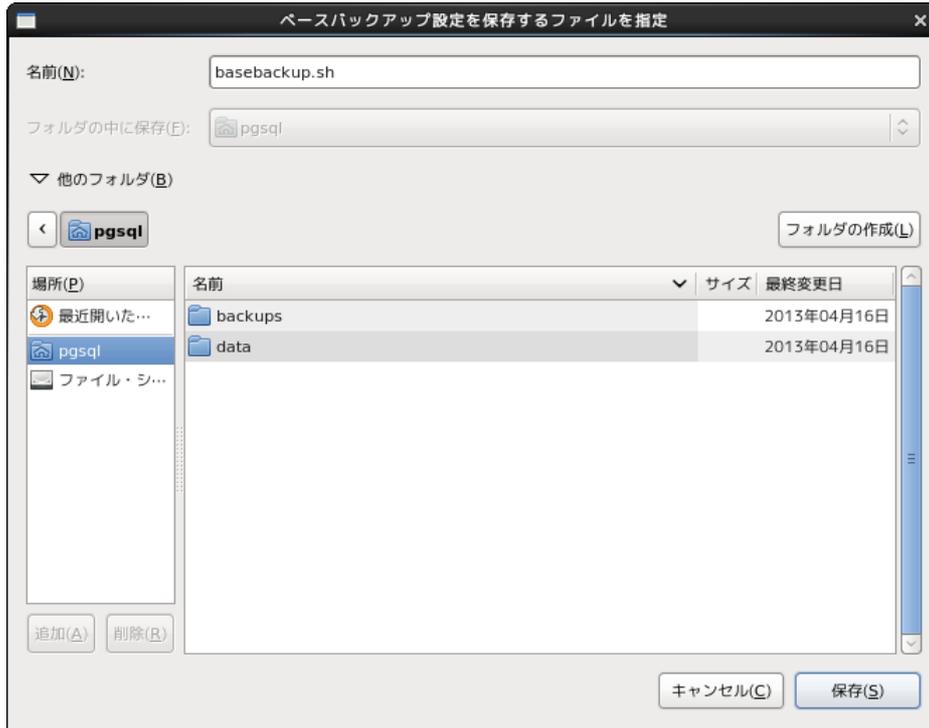


- ツリーメニューの**オンラインバックアップ**を選択し、**ベースバックアップ**タブを開き、**ベースバックアップ設**

定を保存ボタンを押します。



4. バックアップ設定ファイルの保存先のフォルダを選択し、名前にファイル名を入力し、保存ボタンを押します。



ファイルに保存したというダイアログが表示されます。OK ボタンを押します。



バックアップ設定ファイルの内容は以下のようになります。

```
#!/bin/sh
export PATH=/opt/powergresplus91/bin:$PATH
export PGHOME=/opt/powergresplus91
"/opt/powergresplus91/bin/"powergres-mgr --basebackup "service=powergresplus
password=password user=postgres"
```

上記のバックアップ設定ファイルでもベースバックアップは作成できますが、ベースバックアップの作成に成功した場合にもログメッセージが出力され、サービスが起動していない場合にはベースバックアップの作成に失敗してしまいます。

サービスが起動している場合のみベースバックアップを作成し、ベースバックアップの作成に失敗した場合のみエラーメッセージが出力されるようにするには、バックアップ設定ファイルを以下のように修正します。

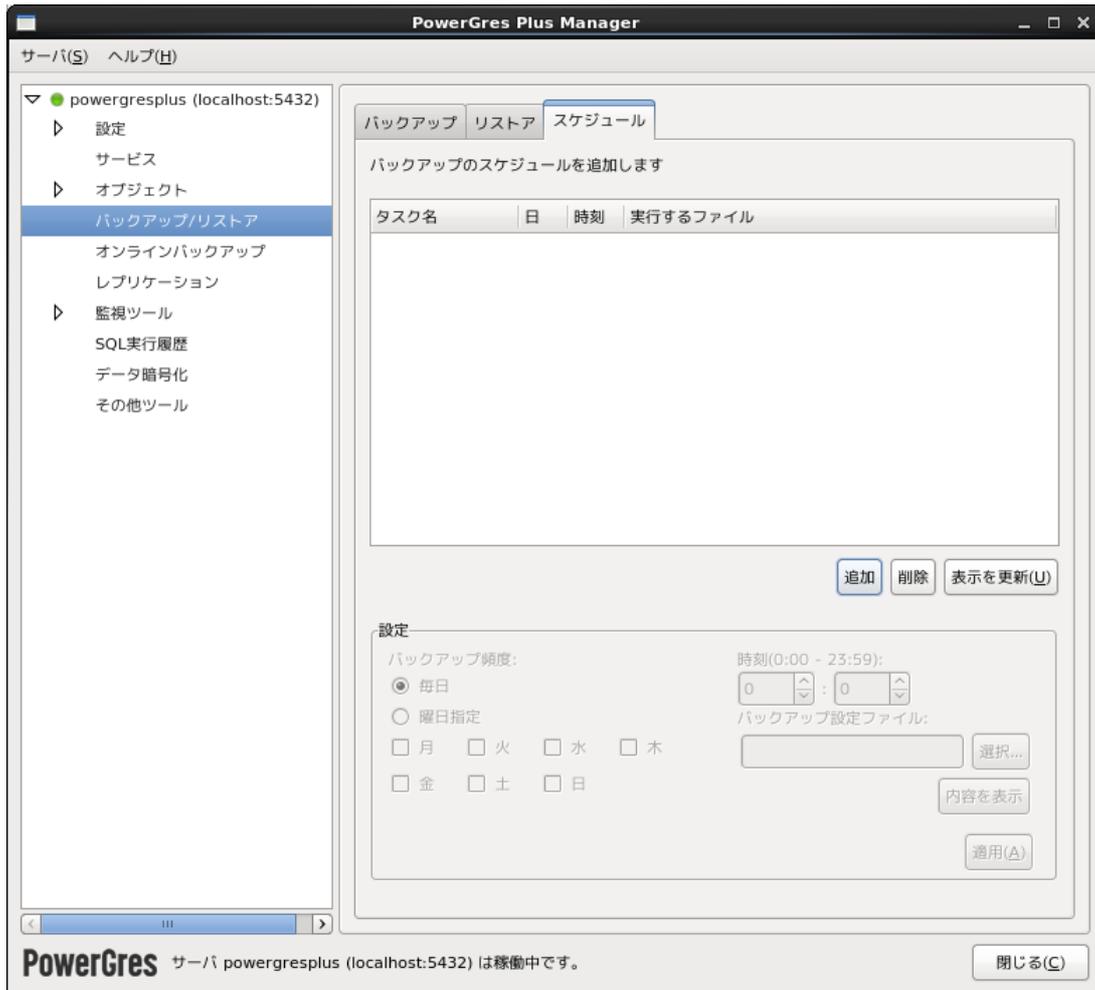
```
#!/bin/sh
export PATH=/opt/powergresplus91/bin:$PATH
export PGHOME=/opt/powergresplus91
export PGDATA=/mnt/disk1/data
if [ ! -f $PGDATA/postmaster.pid ]; then
    exit 0
fi
```

```

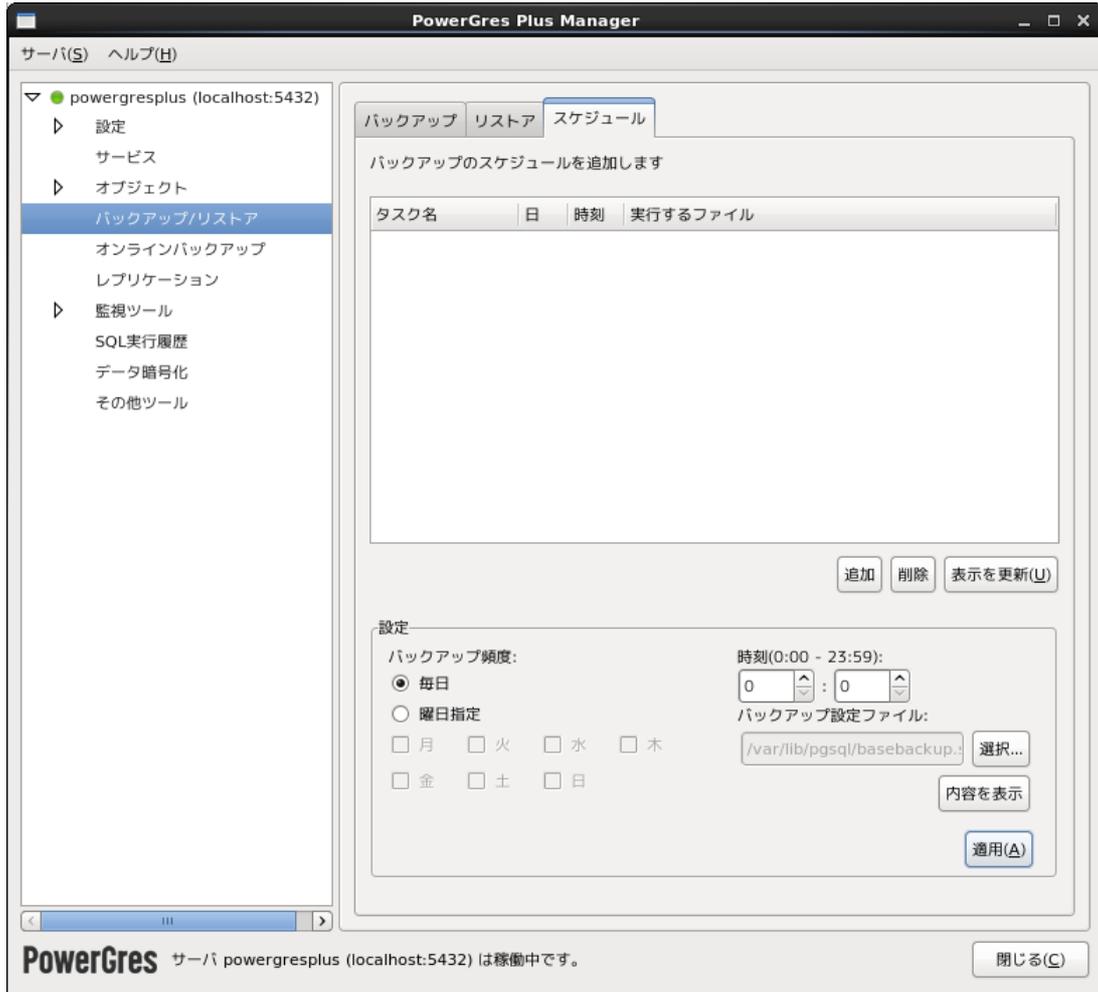
output=`"/opt/powergresplus91/bin/"powergres-mgr --basebackup "service=power
gresplus password=password user=postgres" 2>&1`
exit_status=$?
if [ $exit_status -ne 0 ]; then
    echo -e "$output"
    exit $exit_status
fi
exit 0

```

5. ツリーメニューのバックアップ/リストアを選択し、スケジュールタブを開き、追加ボタンを押します。



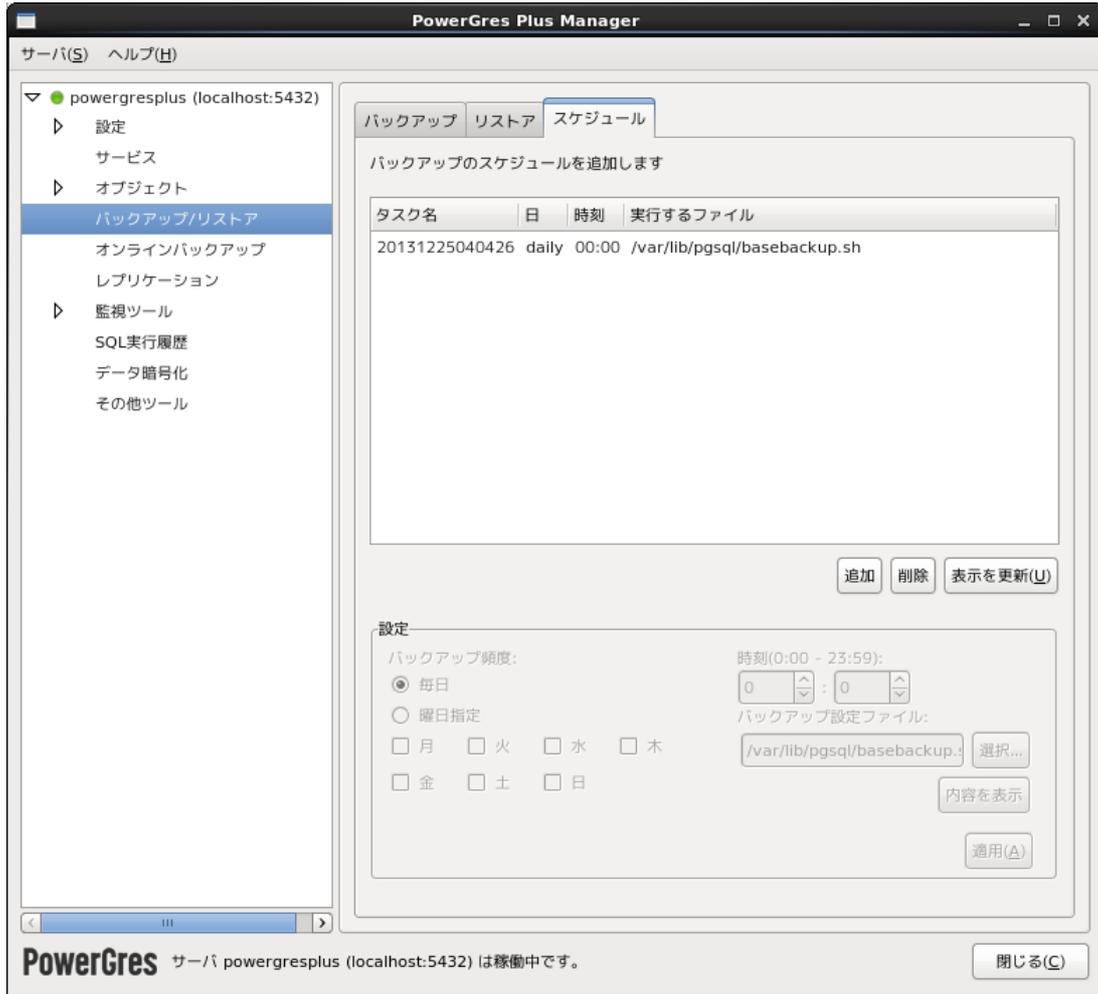
6. 設定のバックアップ頻度ではバックアップを毎日実行する場合には**毎日**を選択し、特定の曜日に実行する場合には**曜日指定**を選択して月～日の曜日をチェックします。次に**時刻**にバックアップを実行する時刻を入力し、**バックアップ設定ファイル**の**選択**ボタンを押してバックアップ設定ファイルを選択します。最後に**適用**ボタンを押します。



スケジュールを追加したというダイアログが表示されます。**OK** ボタンを押します。



7. スケジュールの一覧にバックアップのスケジュールが追加されます。



8. サーバメニューの終了を選択し、PowerGres Plus Manager を終了します。
9. フェイルオーバー後に待機系サーバでも自動的にベースバックアップを作成できるように、バックアップ設定ファイルを待機系サーバにコピーします。

```
$ scp ~/basebackup.sh bob:~
postgres@bob's password:
basebackup.sh                               100% 198    0.2KB/s   00:00
```

稼働系サーバと同じように待機系サーバでもベースバックアップの自動作成の設定を行います。

待機系サーバで PowerGres Plus Manager を起動すると、設定ファイル postgresql.conf を読み込めないというエラーのダイアログが表示されます。データベースクラスタの存在するディスクがマウントされていないためです。ベースバックアップの自動作成の設定には影響ないため、無視して問題ありません。OK を押します。



6.5. データ暗号化の設定

データ暗号化の機能を使用する場合にはデータ暗号化の設定を行います。詳細については「[PowerGres Plus \(Linux 版\) マニュアル](#)」の「[透過的データ暗号化](#)」、「[PowerGres Plus Manager マニュアル](#)」の「[データ暗号化](#)」を参照してください。

データ暗号化は暗号化テーブルスペース内に作成されたテーブルおよびインデックスを暗号化する機能です。暗号化テーブルスペースの格納先のディレクトリはデータレプリケーションリソース階層でレプリケーションされたディスク上に作成します。

ここでは暗号化テーブルスペース encrypted を作成します。データ暗号化の設定は以下のとおりです。

キーストア格納ディレクトリ	/var/lib/pgsql/keystore
パスフレーズ	passphrase
テーブルスペース名	encrypted
テーブルスペースの場所	/mnt/disk1/encrypted_tblspc
テーブルスペースの所有者	postgres
データ暗号化アルゴリズム	AES256

データ暗号化の設定は以下の手順に従って行います。

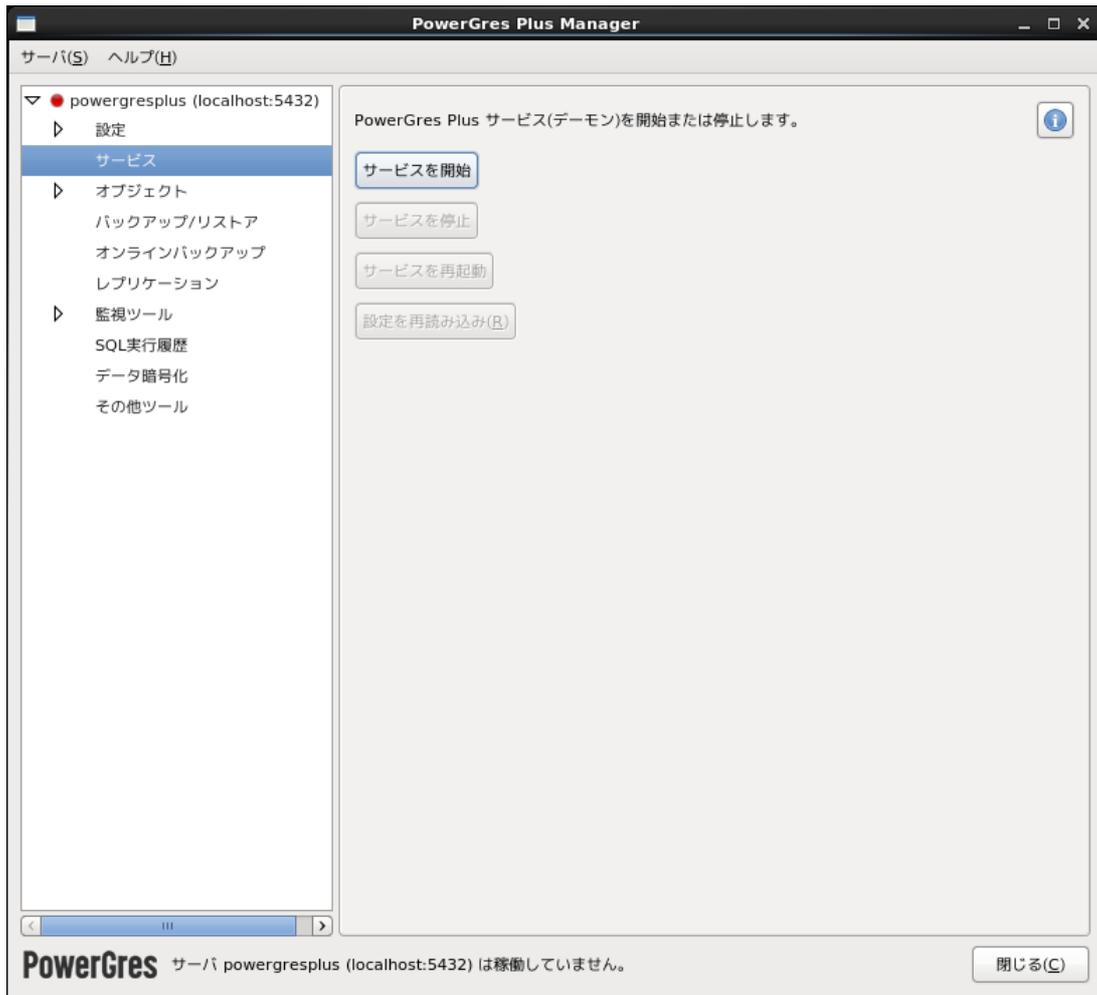
- 稼働系サーバにおいて root ユーザでテーブルスペースの場所に暗号化テーブルスペースの格納先のディレクトリを作成し、PowerGres Plus のスーパーユーザのみがアクセスできるようにアクセス権と所有者の設定を行います。

```
# mkdir /mnt/disk1/encrypted_tblspc
# chmod 0700 /mnt/disk1/encrypted_tblspc
# chown postgres:postgres /mnt/disk1/encrypted_tblspc
```

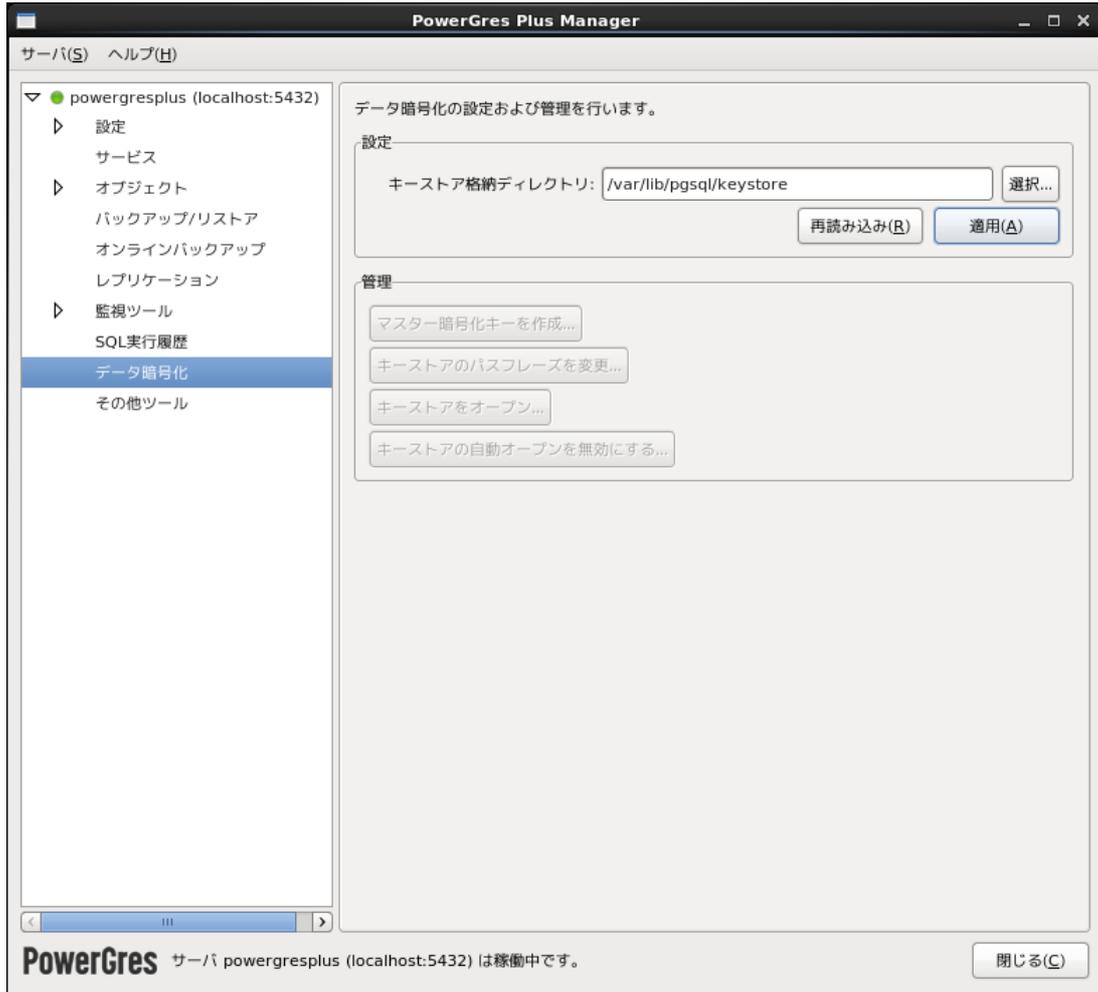
- 稼働系サーバにおいて postgres ユーザで PowerGres Plus Manager を起動します。

```
$ powergres-mgr
```

- サービスが起動していない場合には、ツリーメニューのサービスを選択し、サービスを開始ボタンを押します。

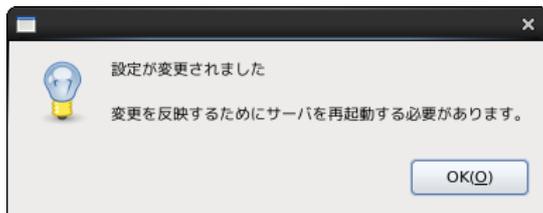


4. ツリーメニューの**データ暗号化**を選択し、**設定のキーストア格納ディレクトリ**にキーストア格納ディレクトリのパスを入力し、**適用ボタン**を押します。

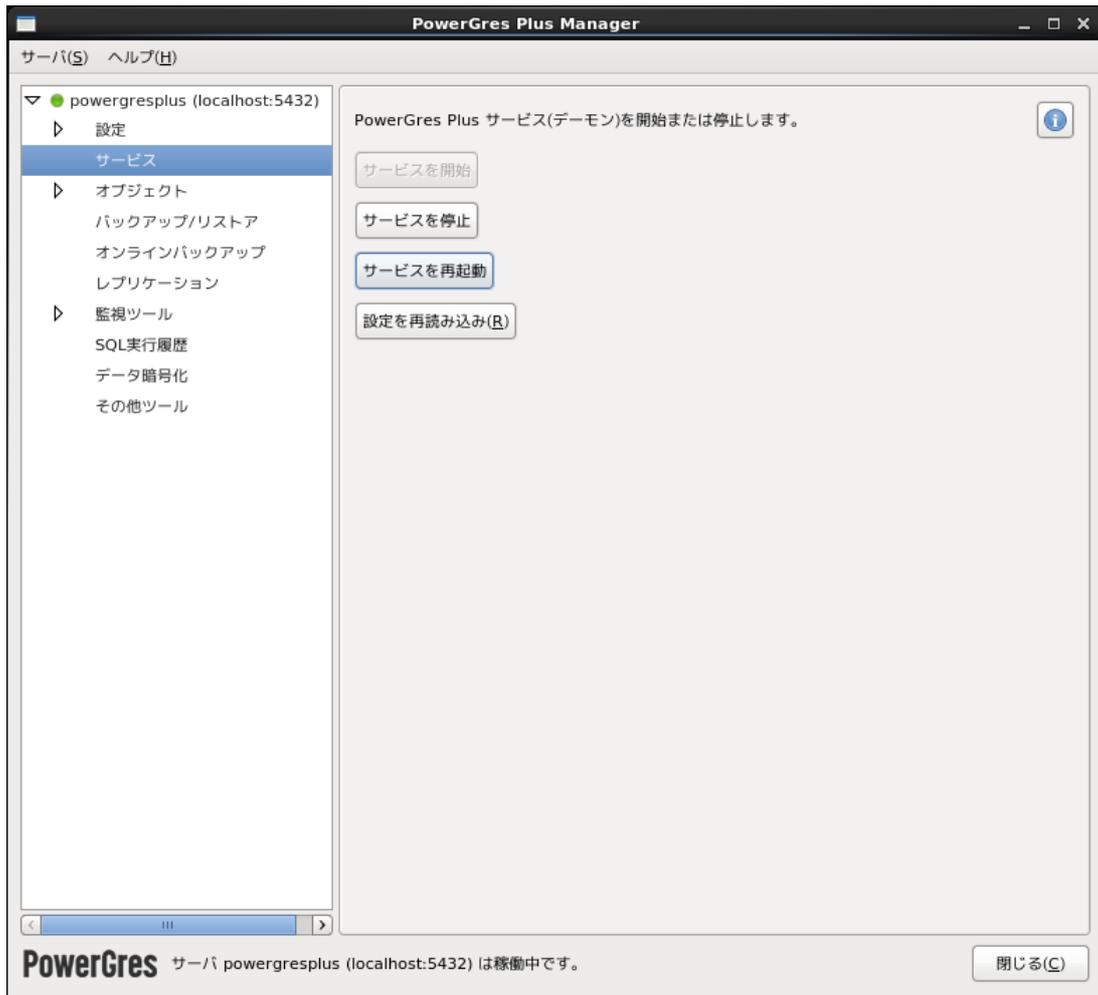


キーストア格納ディレクトリにはレプリケーションされたディスクではなく通常のディスク上のパスを指定します。キーストアは暗号化されたマスター暗号化キーが格納されるファイルです。

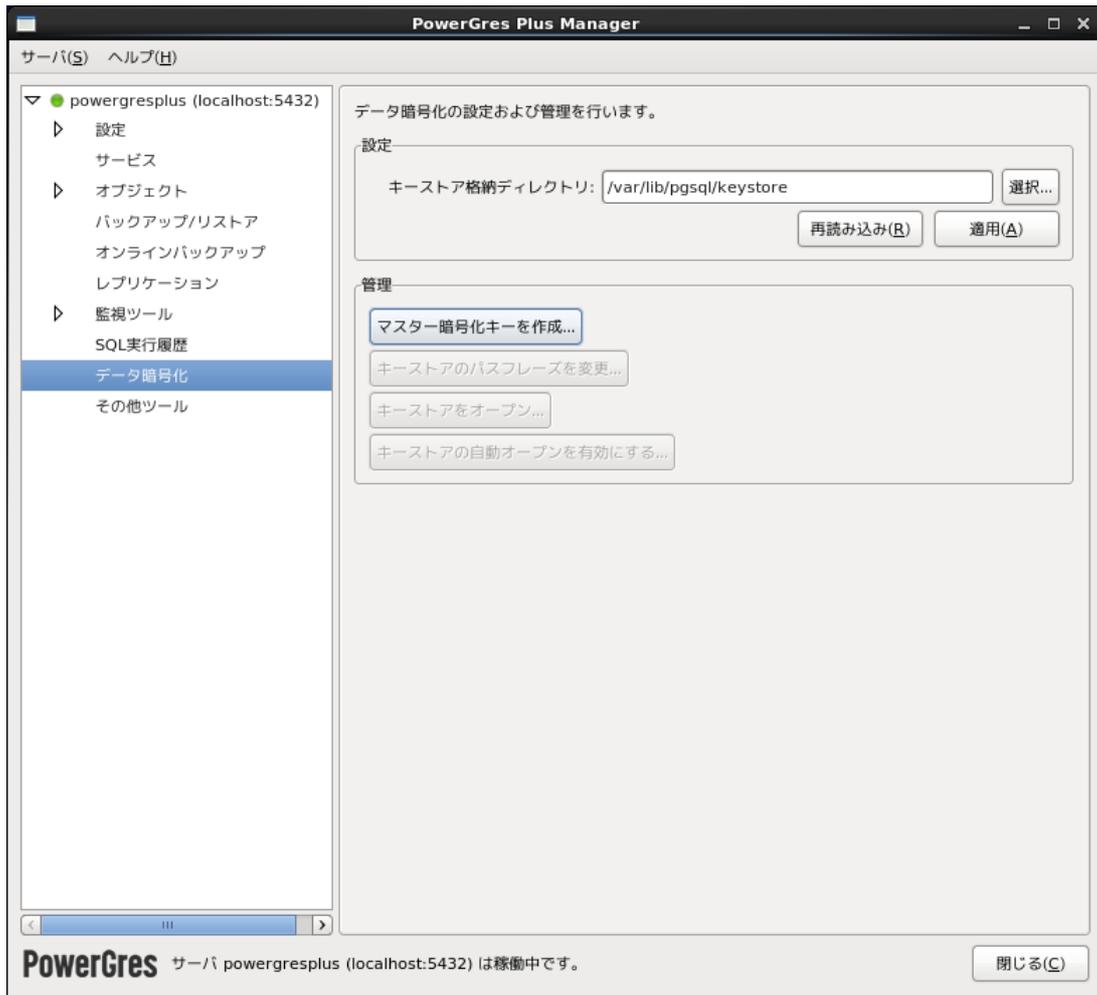
設定を反映するにはサービスの再起動が必要であるというダイアログが表示されます。**OK** ボタンを押します。



5. ツリーメニューの**サービス**を選択し、**サービスを再起動**ボタンを押します。



6. ツリーメニューの**データ暗号化**を選択し、**管理のマスター暗号化キーを作成...**ボタンを押します。



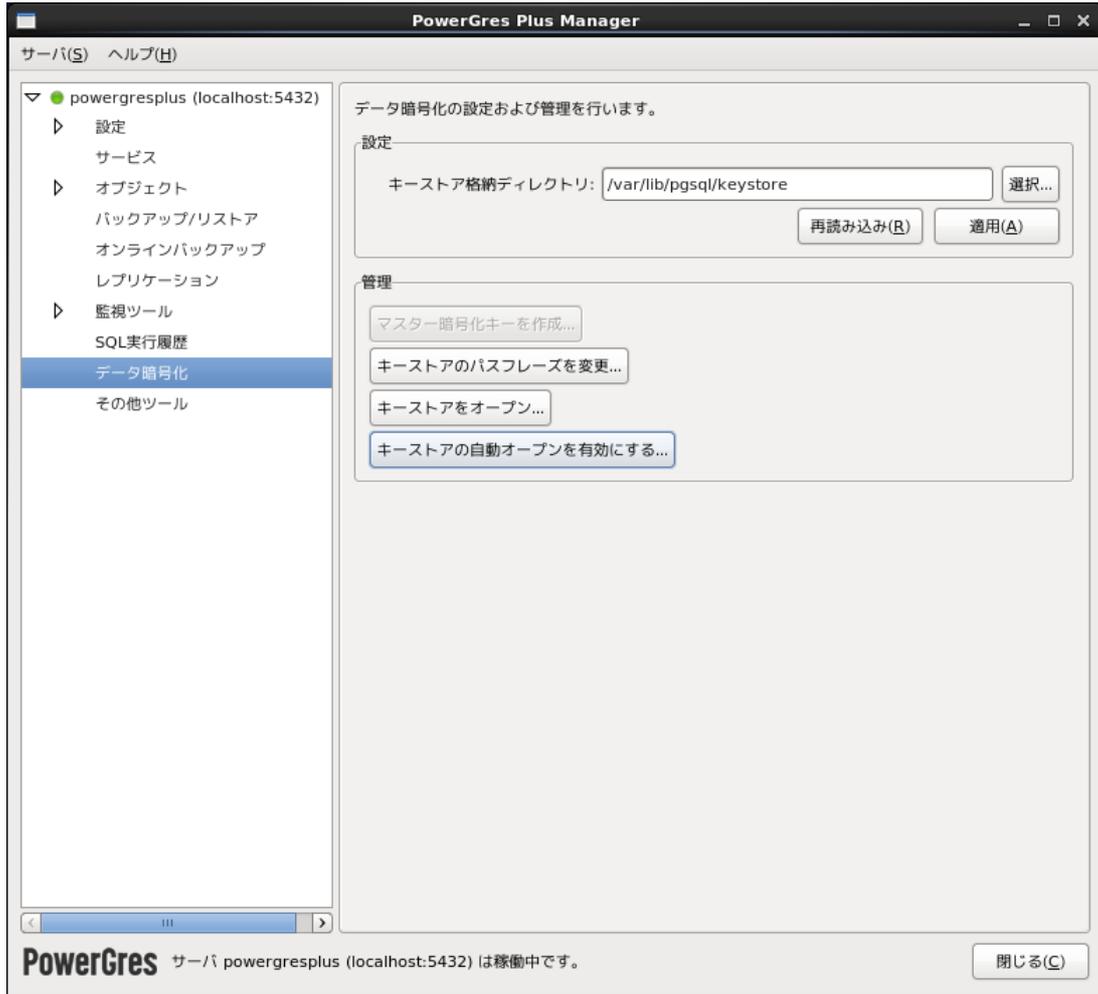
7. パスフレーズを入力するダイアログが表示されます。パスフレーズを入力し、**OK** ボタンを押します。



ランダムなビット列からなるマスター暗号化キーが作成されます。マスター暗号化キーは、指定したパスフレーズで暗号化され、キーストアに格納されます。マスター暗号化キーはデータの暗号化および復号に使用されます。

データの暗号化および復号にはキーストアをオープンする必要があります。サービスの起動時に自動的にキーストアをオープンするにはキーストアの自動オープンを有効にします。

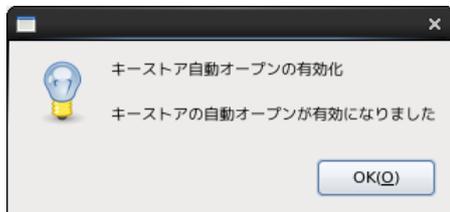
8. ツリーメニューの**データ暗号化**を選択し、**管理のキーストアの自動オープンを有効にする...**ボタンを押します。



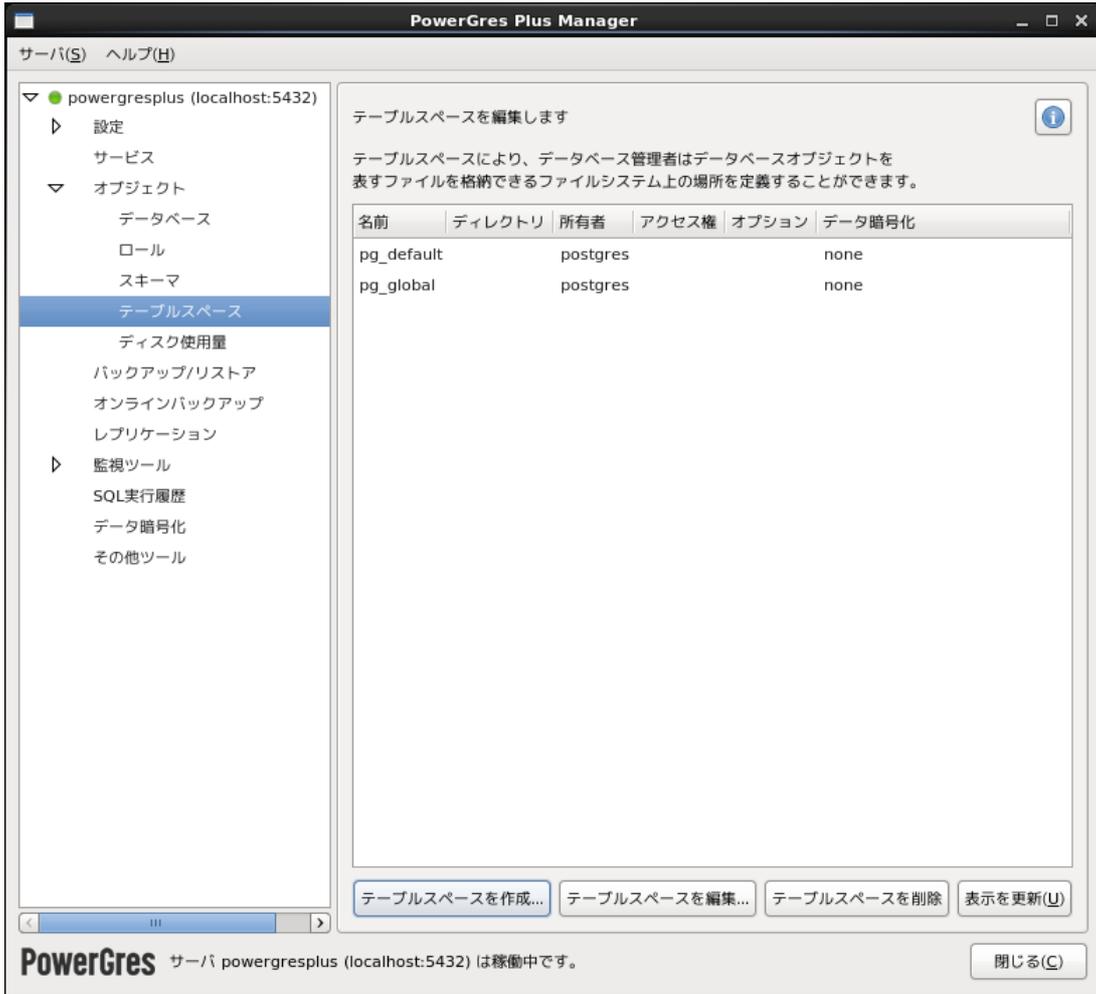
9. パスフレーズを入力するダイアログが表示されます。マスター暗号化キーの作成時に指定したパスフレーズを入力し、**OK** ボタンを押します。



キーストアの自動オープンの有効になったというダイアログが表示されます。**OK** ボタンを押します。



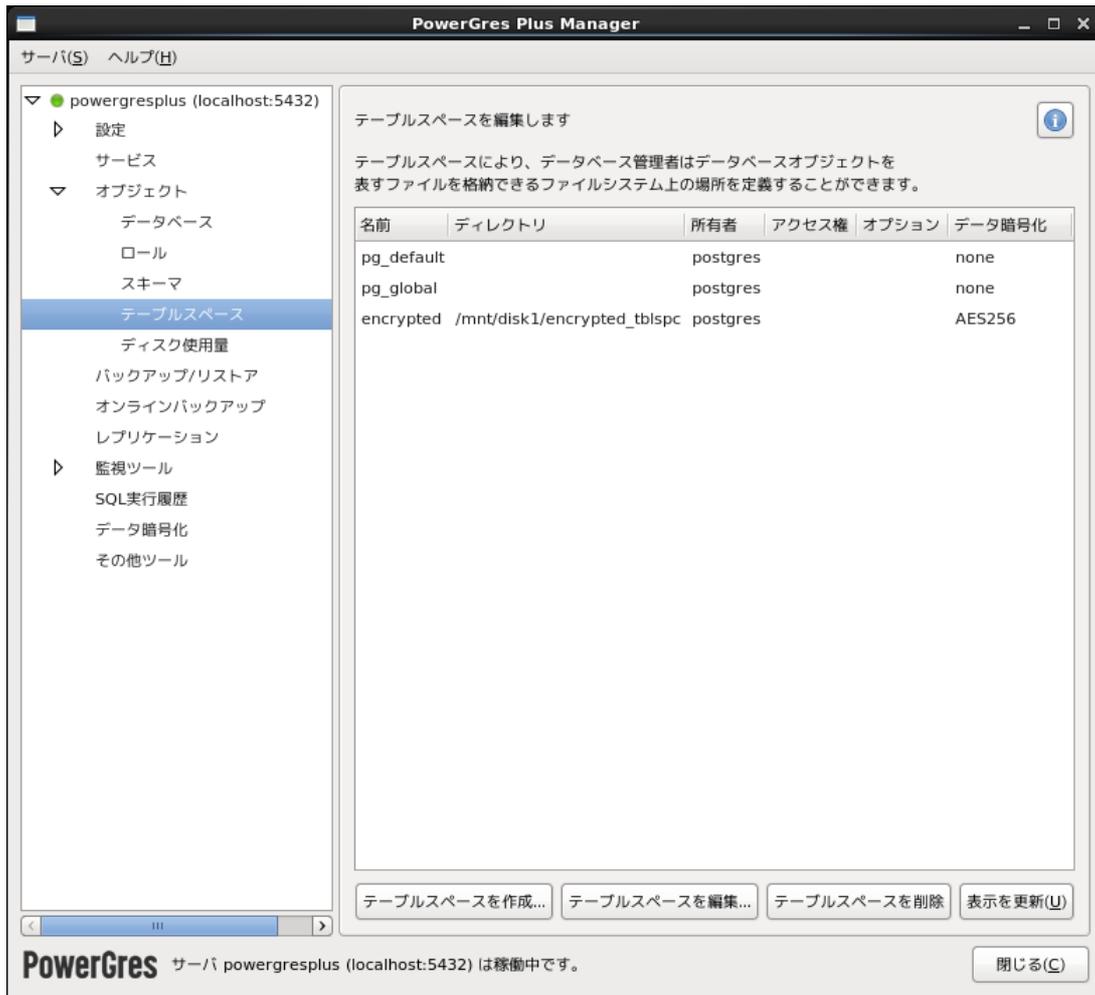
10. ツリーメニューのオブジェクト ▶ テーブルスペースを選択し、テーブルスペースを作成...ボタンを押します。



11. テーブルスペース名にテーブルスペース名、**テーブルスペースの場所**に暗号化テーブルスペースの格納先のディレクトリのパスを入力します。次に**テーブルスペースの所有者**からテーブルスペースの所有者、**データ暗号化アルゴリズム**からデータ暗号化アルゴリズムとして AES128 または AES256 を選択します。最後に **OK** ボタンを押します。



12. テーブルスペースの一覧に暗号化テーブルスペースが追加されます。



暗号化テーブルスペースは**データ暗号化列**にデータ暗号化アルゴリズムとして AES128 または AES256 と表示されます。

- ツリーメニューの**サービス**を選択し、**サービスを停止**ボタンを押します。
- サーバ**メニューの**終了**を選択し、PowerGres Plus Manager を終了します。
- フェイルオーバー後に待機系サーバでもキーストアをオープンできるように、キーストア格納ディレクトリを待機系サーバにコピーします。

```
$ rsync -av ~/keystore bob::~
postgres@bob's password: (パスワードを入力)
sending incremental file list
keystore/
keystore/keystore.aks
keystore/keystore.ks

sent 2044 bytes  received 54 bytes  599.43 bytes/sec
total size is 1856  speedup is 0.88
```

稼働系サーバと同じように待機系サーバでもキーストアの自動オープンの設定を行います。

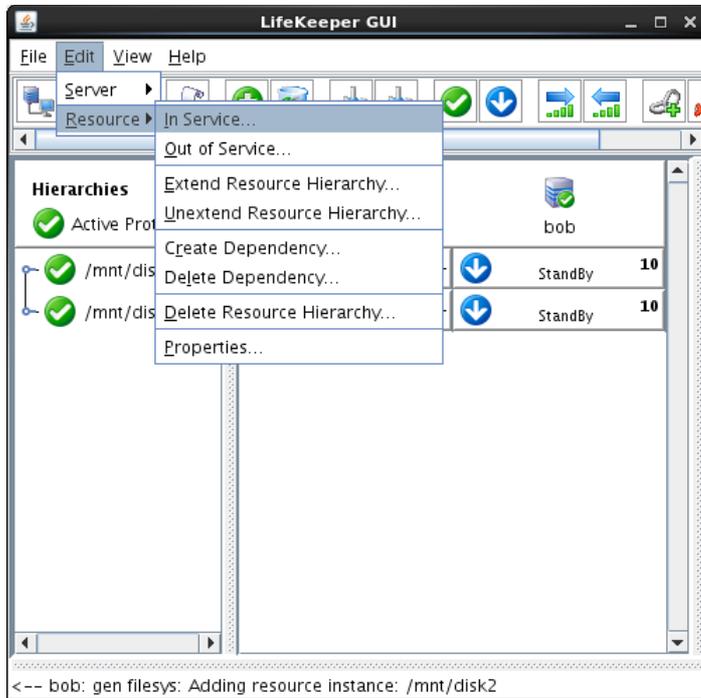
キーストアの自動オープンの設定を行うにはデータベースクラスタの存在するディスクがマウントされている必要があります。そのため、データレプリケーションリソース階層のサービスを待機系サーバに切

り替えます。

16. 待機系サーバにおいて root ユーザで LifeKeeper GUI を起動し、接続します。

```
# lkGUIapp
```

17. **Edit** メニューの **Resource** ▶ **In Service...** を選択します。



18. **Server** から待機系サーバを選択し、**Next>** ボタンを押します。



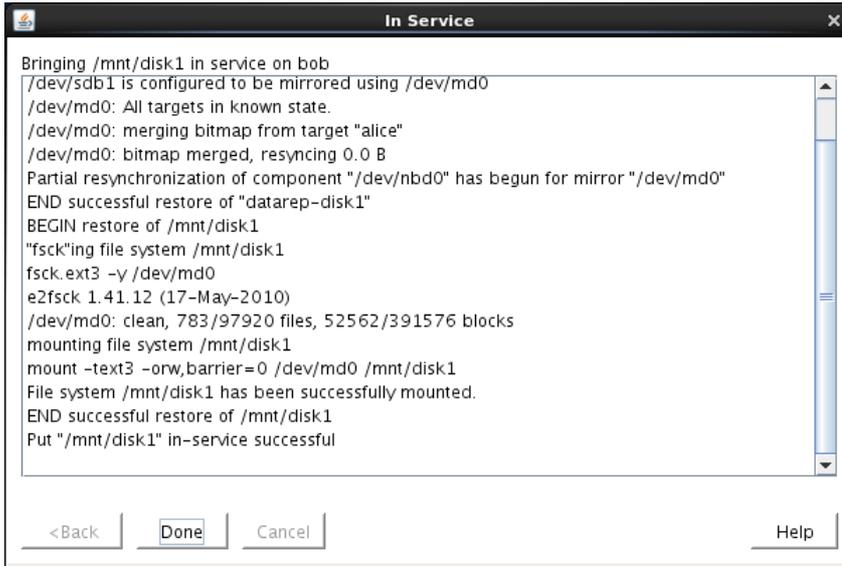
19. **Resource(s)** から待機系サーバでサービスを起動するリソースを選択し、**Next>** ボタンを押します。



20. **In Service** ボタンを押します。

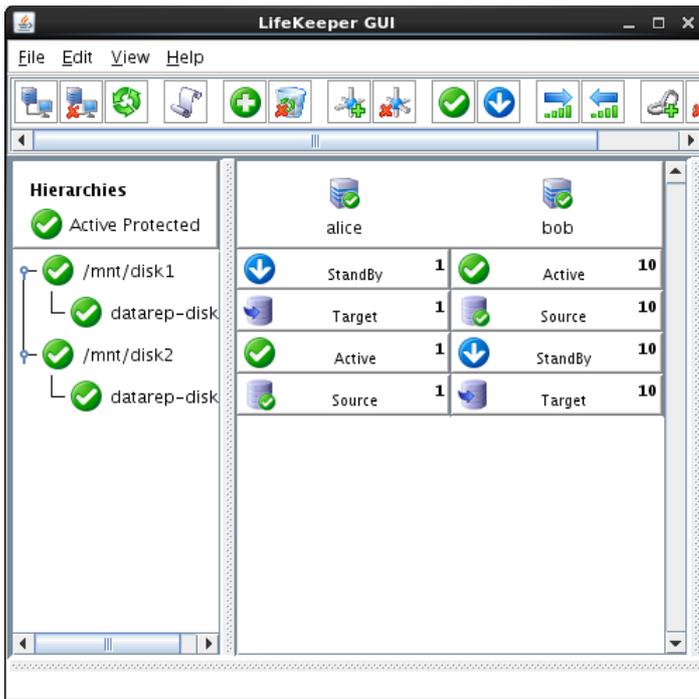


21. 待機系サーバでリソースのサービスが起動されます。



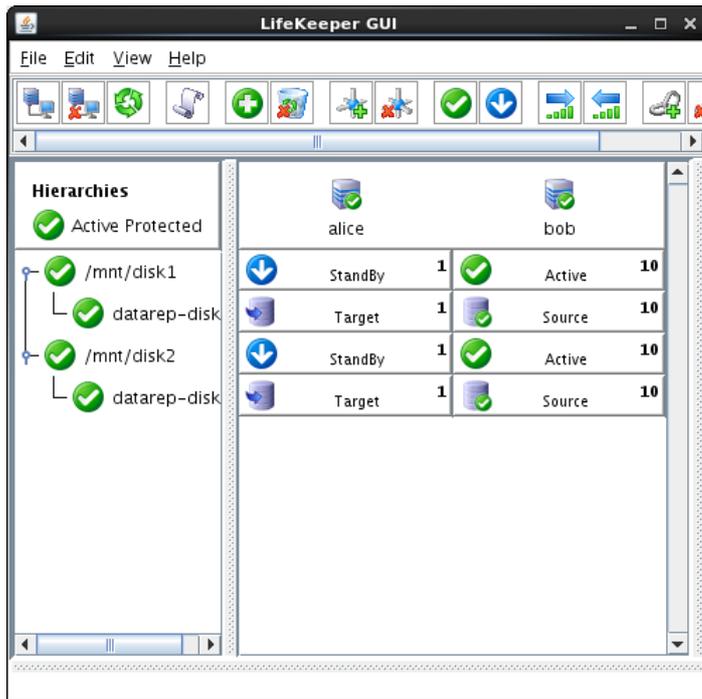
リソースのサービスが起動したら、**Done** ボタンを押します。

22. データレプリケーションリソース階層のサービスが待機系サーバに切り替わります。



データレプリケーションリソース階層の状態は稼働系サーバがサービス停止中(StandBy)、待機系サーバがサービス起動中(Active)になります。また、データレプリケーションリソースの状態は稼働系サーバがレプリケーション先(Target)、待機系サーバがレプリケーション元(Source)になります。

データ領域用と同じようにバックアップ領域用のデータレプリケーションリソース階層のサービスを待機系サーバに切り替えます。



23. **File** メニューの **Exit** を選択し、LifeKeeper GUI を終了します。

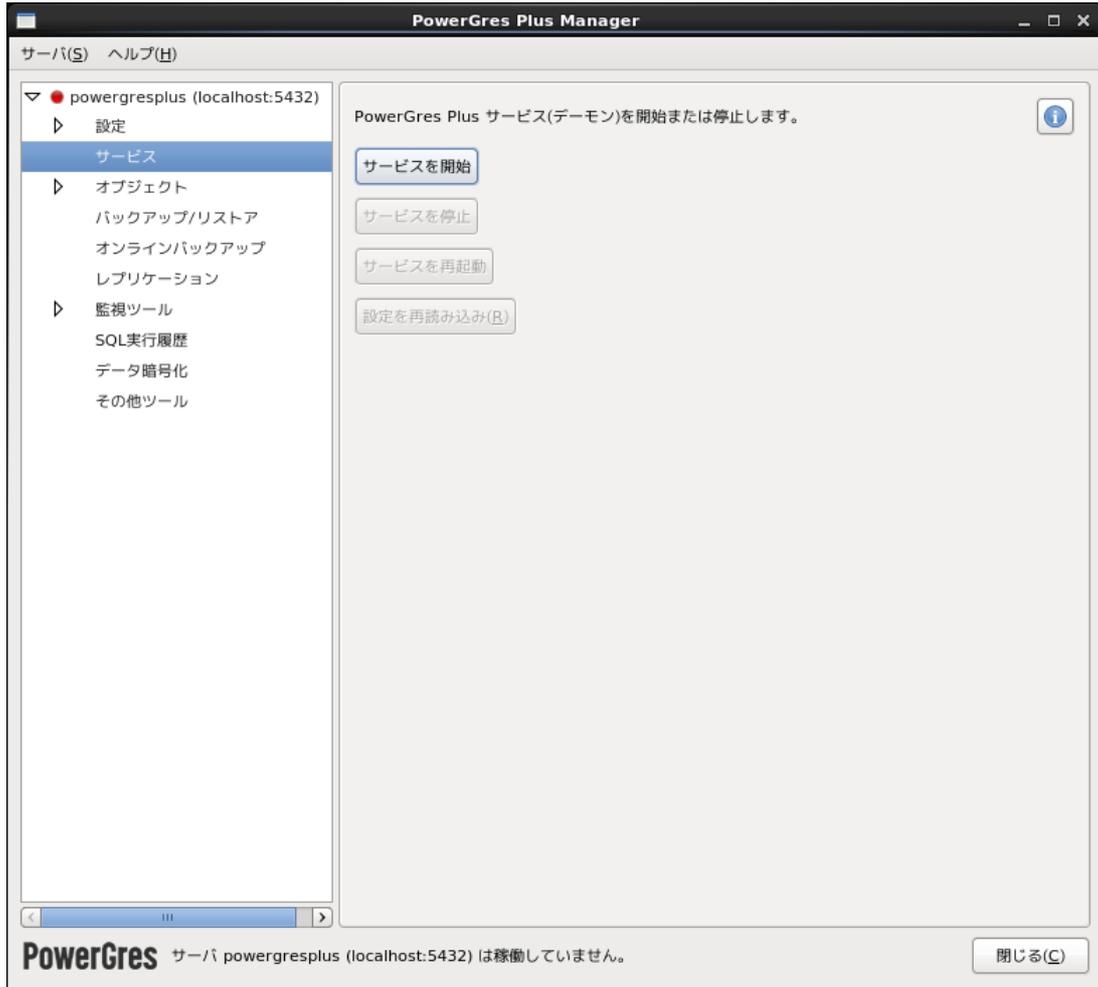
24. 待機系サーバにおいて postgres ユーザで自動オープンキーストアを削除します。

```
$ rm -f ~/keystore/keystore.aks
```

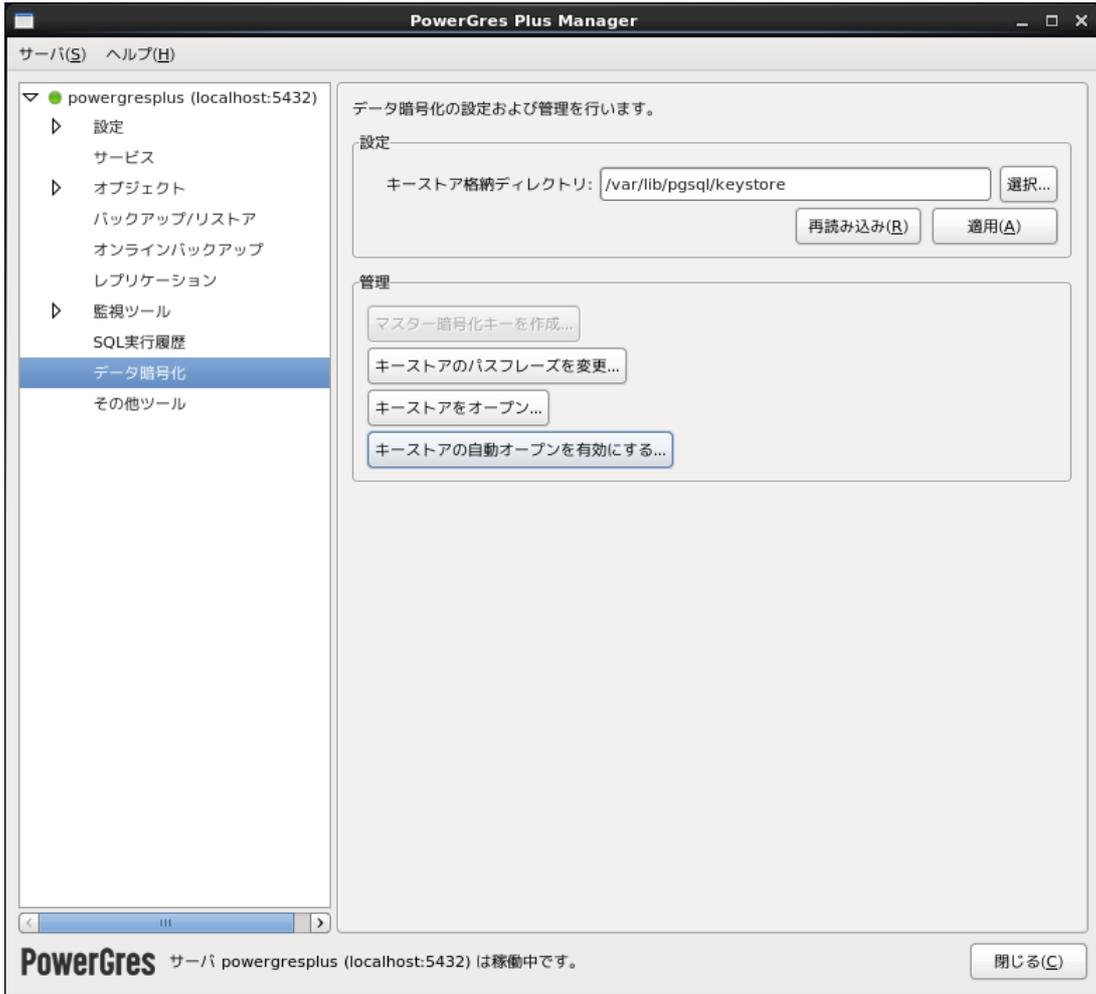
25. 待機系サーバにおいて postgres ユーザで PowerGres Plus Manager を起動します。

```
$ powergres-mgr
```

26. ツリーメニューの **サービス** を選択し、**サービスを開始** ボタンを押します。



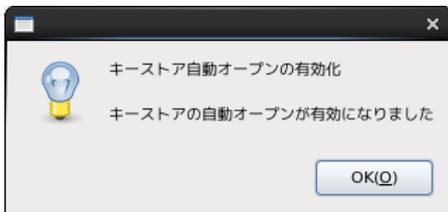
27. ツリーメニューのデータ暗号化を選択し、管理のキーストアの自動オープンを有効にする...ボタンを押します。



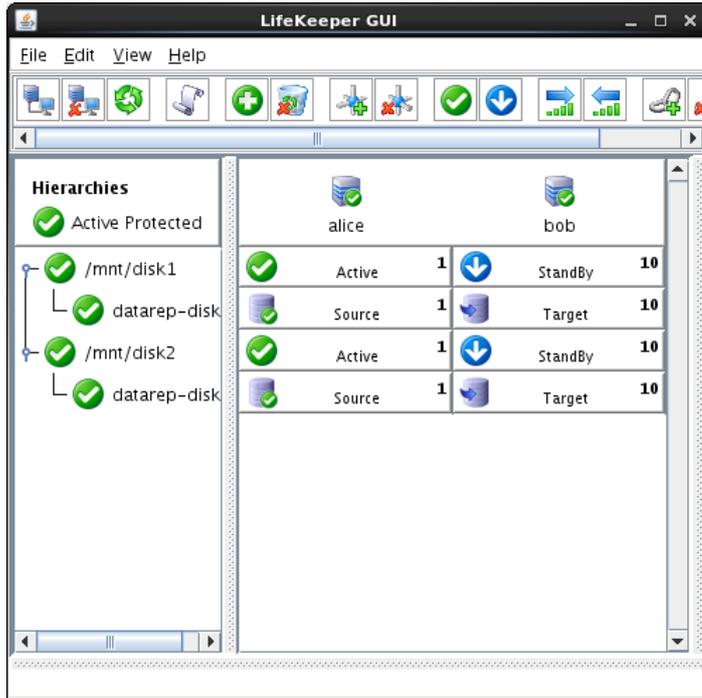
28. パスフレーズを入力するダイアログが表示されます。マスター暗号化キーの作成時に指定したパスフレーズを入力し、**OK** ボタンを押します。



キーストアの自動オープンの有効になったというダイアログが表示されます。**OK** ボタンを押します。



29. ツリーメニューの**サービス**を選択し、**サービスを停止**ボタンを押します。
30. **サーバ**メニューの**終了**を選択し、PowerGres Plus Manager を終了します。
31. データレプリケーションリソース階層のサービスを待機系サーバに切り替えたときと同じようにサービスを稼働系サーバに切り戻します。



7. PowerGres リソース階層の作成

PowerGres リソース階層を作成します。

PowerGres リソース階層は PowerGres を LifeKeeper で保護するためのリソース階層です。

ここではデータベースディレクトリ/mnt/disk1/data に作成したデータベースクラスタに対して PowerGres リソース階層を作成します。PowerGres リソース階層の設定は以下のとおりです。背景色が青灰色(■)の部分はデフォルトの設定のままであることを表します。

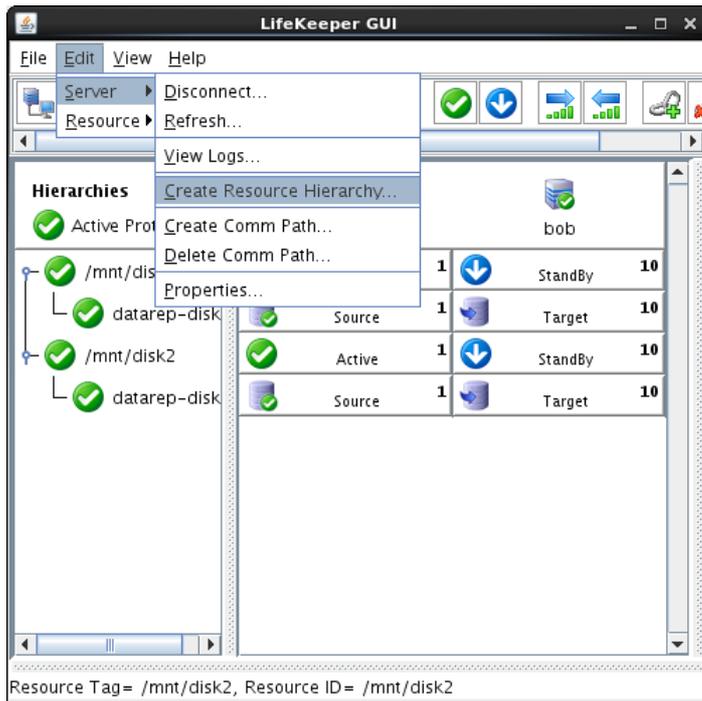
リソース階層の作成	
Please Select Recovery Kit	PowerGres Database
Switchback Type	intelligent
Server	alice
PowerGres data Directory(PGDATA)	/mnt/disk1/data
Location of PowerGres Executables	/opt/powergresplus91/bin
Port number for PowerGres service	5432
Quick check user for PowerGres service	postgres
Root Tag	powergres-/mnt/disk1/data
リソース階層の拡張	
Target Server	bob
Switchback Type	intelligent
Template Priority	1
Target Priority	10
PowerGres data Directory(PGDATA)	/mnt/disk1/data
Location of PowerGres Executables	/opt/powergresplus91/bin
Port number for PowerGres service	5432
Quick check user for PowerGres service	postgres
Root Tag	powergres-/mnt/disk1/data

PowerGres リソース階層の作成は以下の手順に従って稼働系サーバにおいて root ユーザで行います。

1. LifeKeeper GUI を起動し、接続します。

```
# lkGUIapp
```

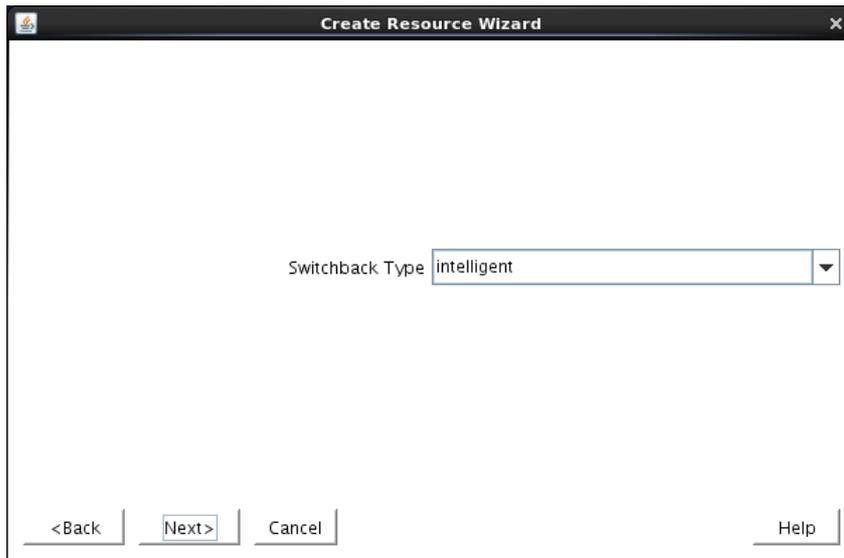
2. **Edit** メニューの **Server ▶ Create Resource Hierarchy...** を選択します。



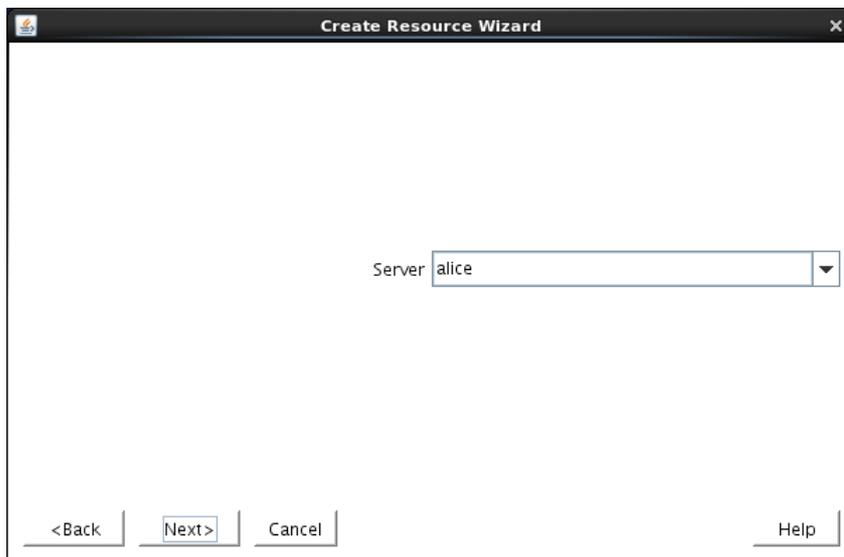
3. **Please Select Recovery Kit** から Recovery Kit として PowerGres Database を選択し、**Next>** ボタンを押します。



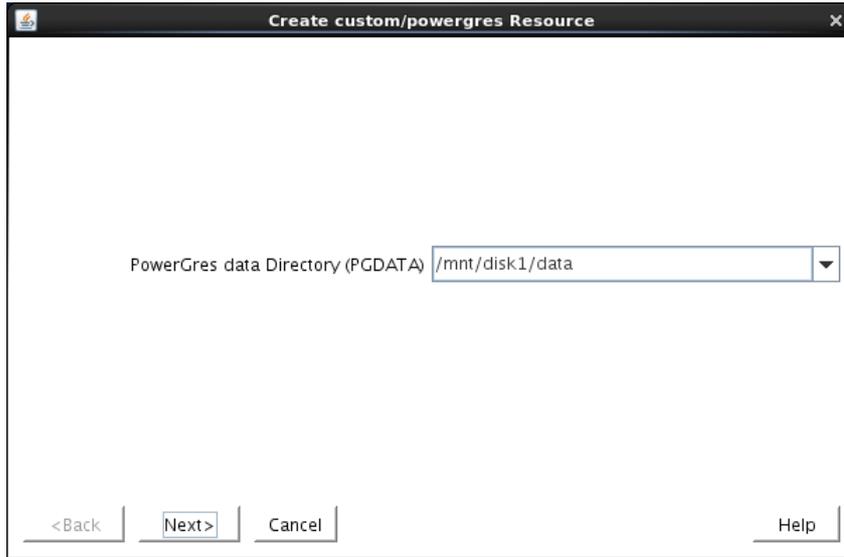
4. **Switchback Type** からスイッチバックの動作として intelligent または automatic を選択し、**Next>** ボタンを押します。



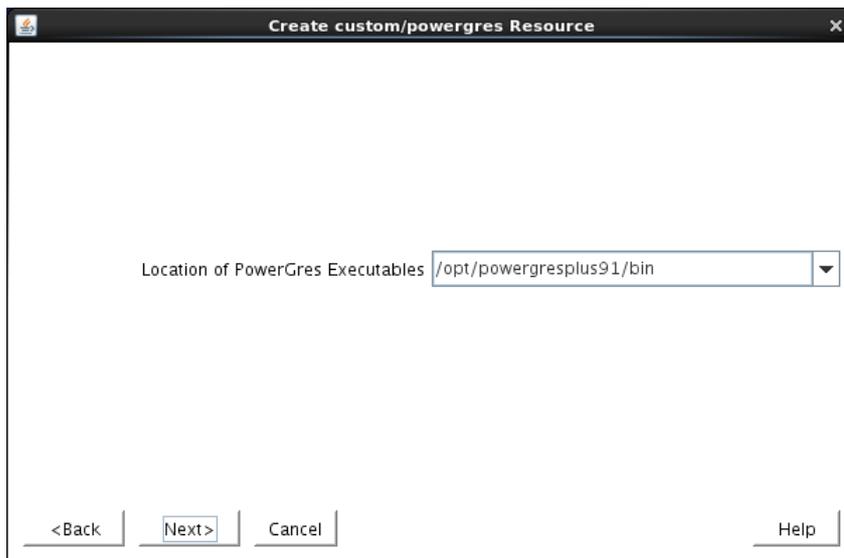
5. **Server** からリソース階層の作成先のサーバとして稼働系サーバを選択し、**Next>**ボタンを押します。



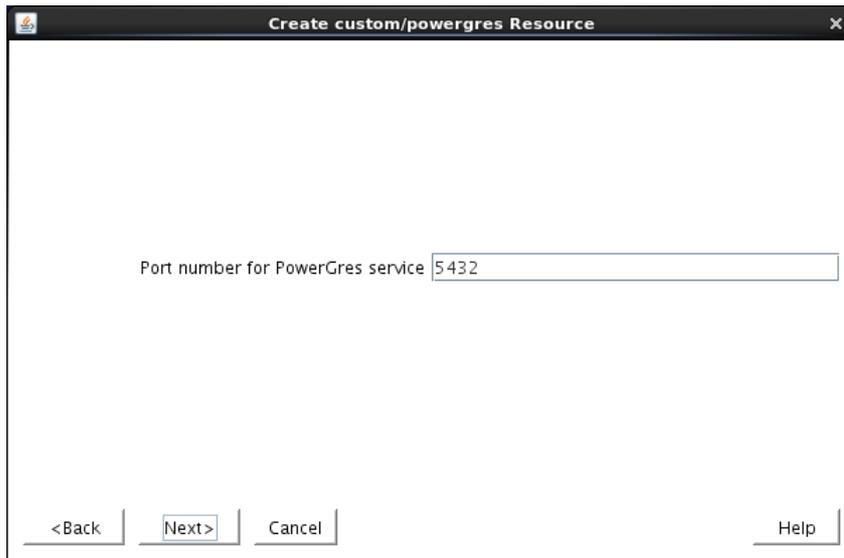
6. **PowerGres data Directory(PGDATA)**にデータベースディレクトリ(データベースクラスタの作成先のディレクトリ)のパスを入力し、**Next>**ボタンを押します。



7. **Location of PowerGres Executables** に PowerGres Plus の実行プログラムの格納先のディレクトリのパスとして /opt/powergresplus91/bin と入力し、**Next>** ボタンを押します。



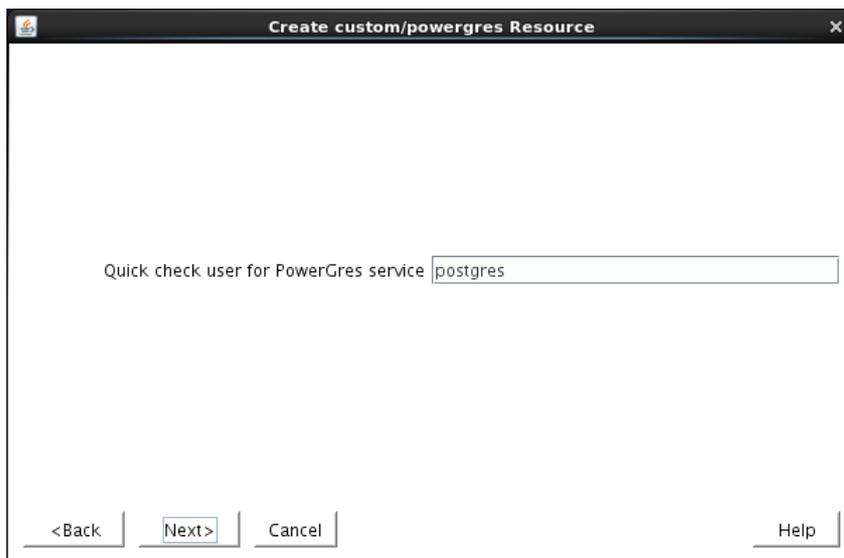
8. **Port number for PowerGres service** にデータベースサーバのポート番号を入力し、**Next>** ボタンを押します。



Port number for PowerGres service 5432

<Back Next> Cancel Help

9. **Quick check user for PowerGres service** にデータベースサーバの監視時に接続に使用するユーザ名を入力し、**Next>**ボタンを押します。



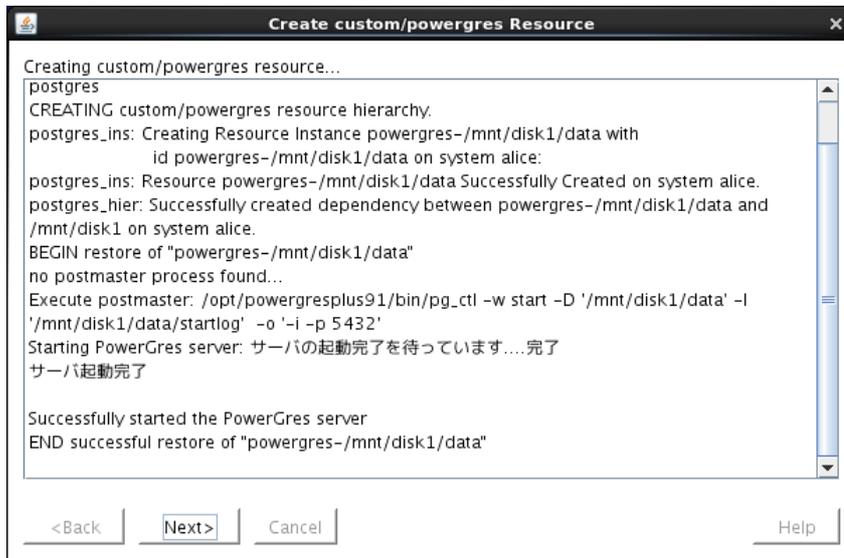
Quick check user for PowerGres service postgres

<Back Next> Cancel Help

10. **Root Tag** に PowerGres リソースのタグ名を入力し、**Create** ボタンを押します。

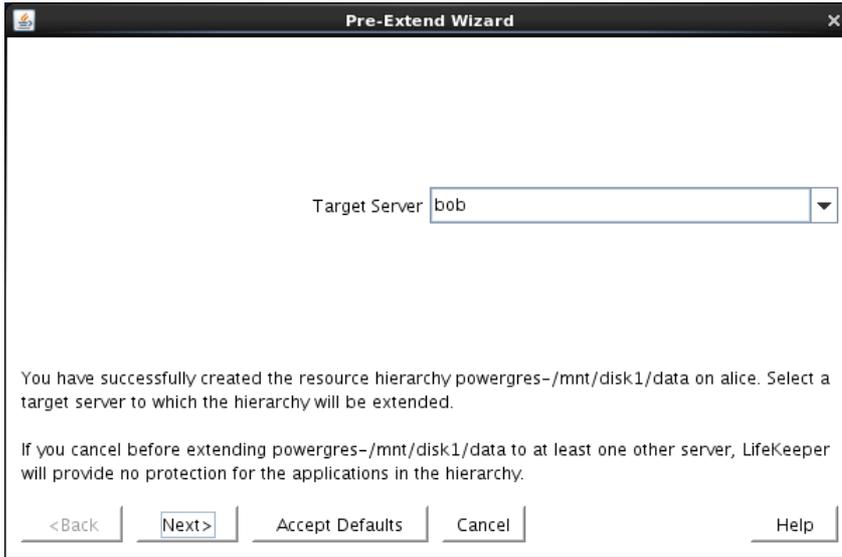


11. リソース階層が作成されます。



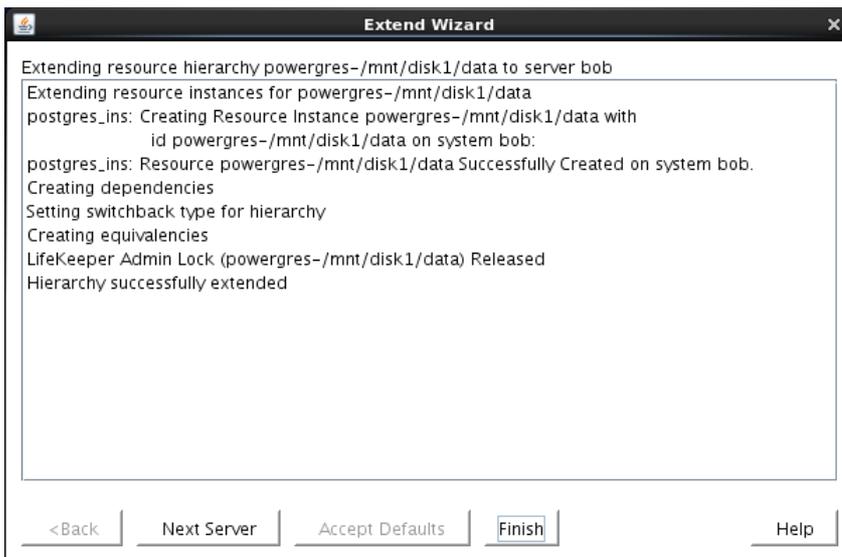
リソース階層の作成が完了したら、**Next>**ボタンを押します。

12. **Target Server** からリソース階層の拡張先のサーバとして待機系サーバを選択します。



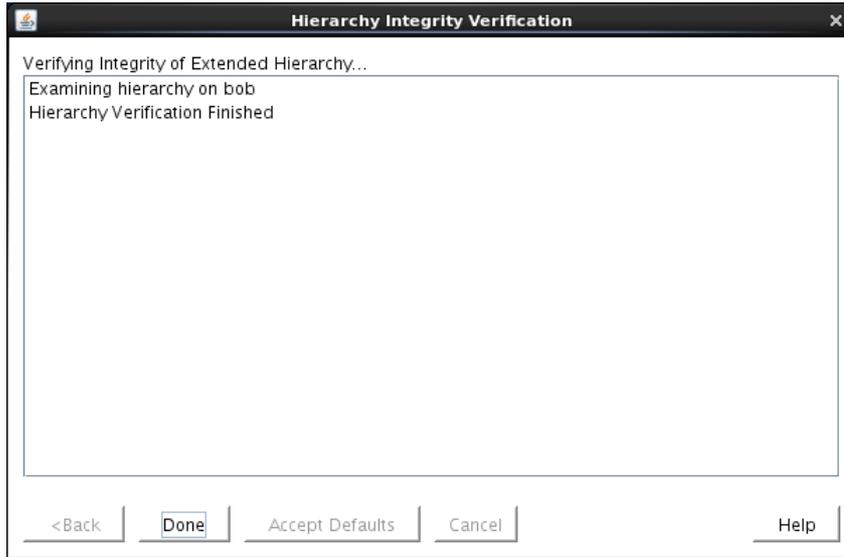
これ以降の設定を確認せずにデフォルトの設定のままで問題ない場合には **Accept Defaults** ボタンを押します。そうでない場合には **Next>** ボタンを押します。ここでは **Accept Defaults** ボタンを押した場合について説明します。

13. リソース階層が拡張されます。

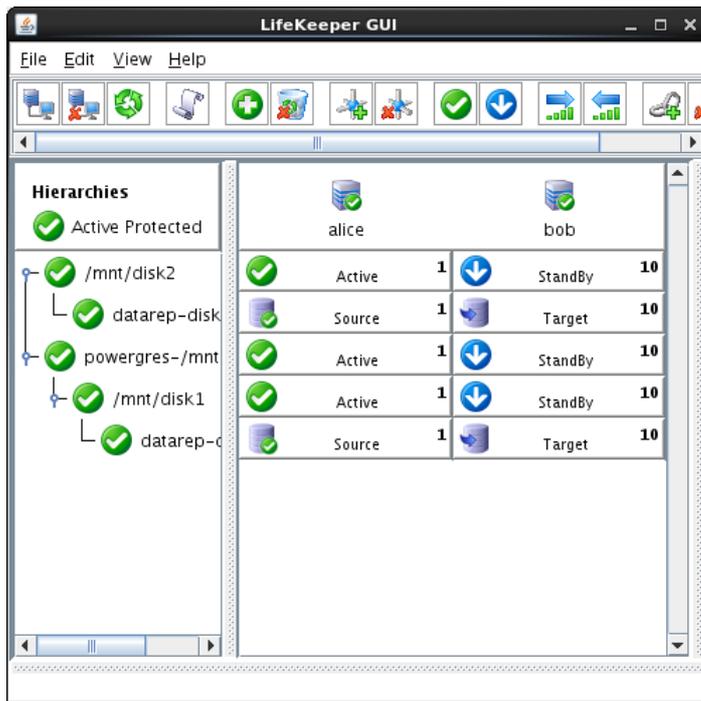


リソース階層の拡張が完了したら、**Finish** ボタンを押します。

14. **Done** ボタンを押します。



15. リソース階層ツリーに PowerGres リソース階層が追加されます。また、PowerGres リソースとデータ領域用のデータレプリケーションリソース階層のファイルシステムリソースとの依存関係が自動的に作成されます。



PowerGres リソース階層の状態は稼働系サーバがサービス起動中(✓Active)、待機系サーバがサービス停止中(↓StandBy)になります。

16. **File** メニューの **Exit** を選択し、LifeKeeper GUI を終了します。

8. IP リソース階層の作成

IP リソース階層を作成します。詳細については「[IP Recovery Kit Technical Documentation](#)」の「[Creating an IP Resource Hierarchy](#)」を参照してください。

IP リソース階層は仮想 IP アドレスを LifeKeeper で保護するためのリソース階層です。仮想 IP アドレスを通してデータベースに接続することにより、稼働系サーバと待機系サーバのどちらで起動しているかを意識する必要がなくなります。

ここでは仮想 IP アドレス 192.168.137.100 に対して IP リソース階層を作成します。IP リソース階層の設定は以下のとおりです。背景色が青灰色(■)の部分はデフォルトの設定のままであることを表します。

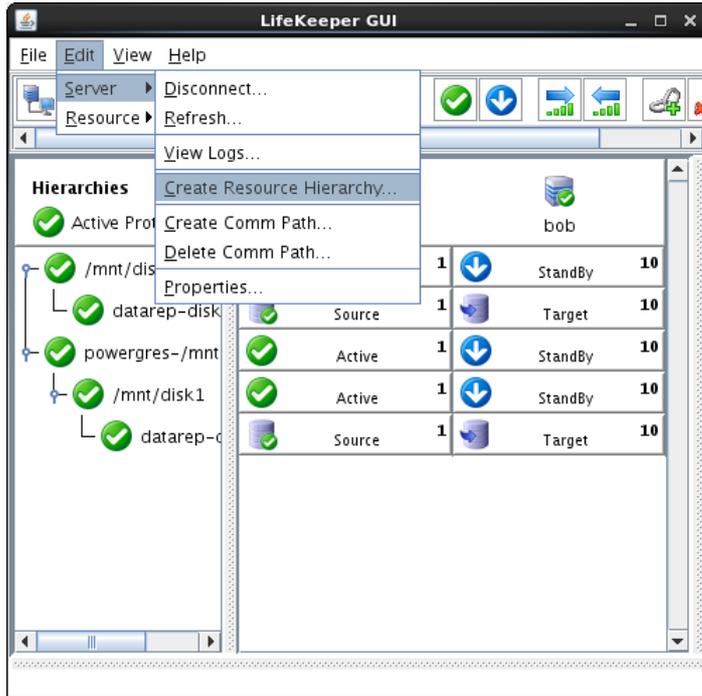
リソース階層の作成	
Please Select Recovery Kit	IP
Switchback Type	intelligent
Server	alice
IP Resource	192.168.137.100
Netmask	255.255.255.0
Network Interface	eth0
IP Resource Tag	ip-192.168.137.100
リソース階層の拡張	
Target Server	bob
Switchback Type	intelligent
Template Priority	1
Target Priority	10
IP Resource	192.168.137.100
Netmask	255.255.255.0
Network Interface	eth0
IP Resource Tag	ip-192.168.137.100

IP リソース階層の作成は以下の手順に従って稼働系サーバにおいて root ユーザで行います。

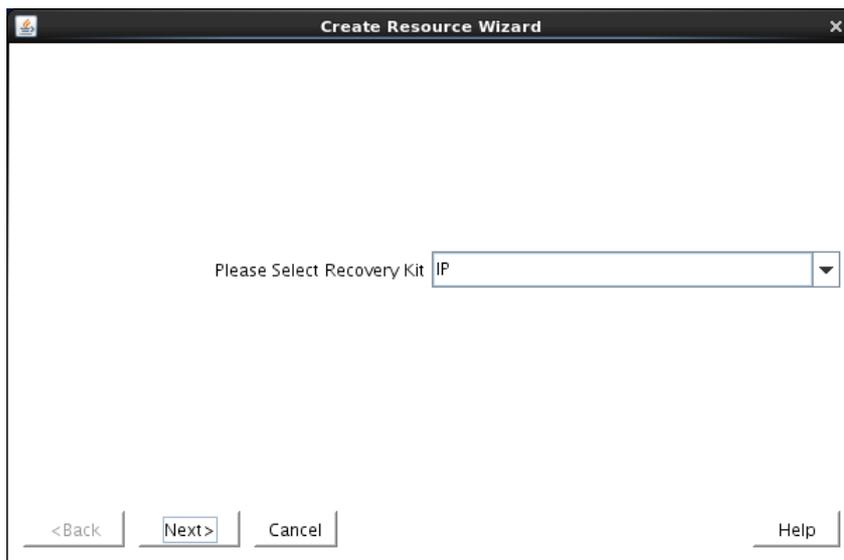
1. LifeKeeper GUI を起動し、接続します。

```
# lkGUIapp
```

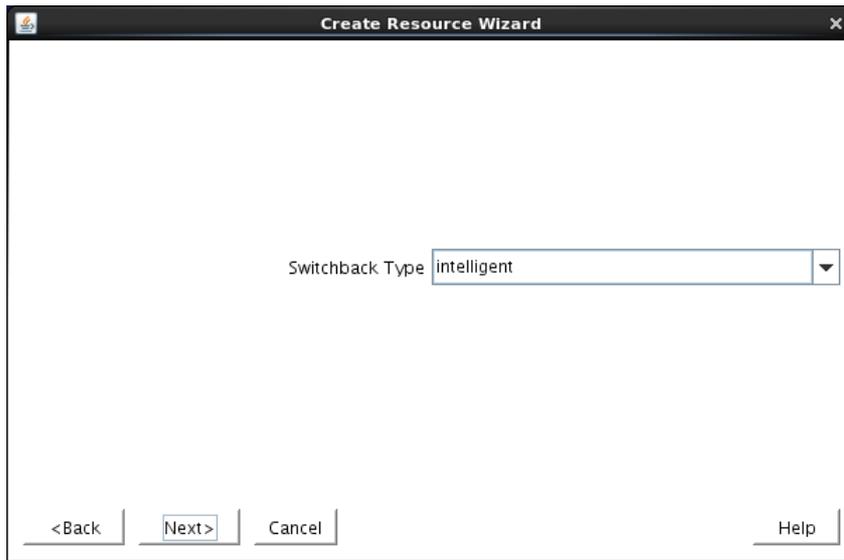
2. **Edit** メニューの **Server** ▶ **Create Resource Hierarchy...** を選択します。



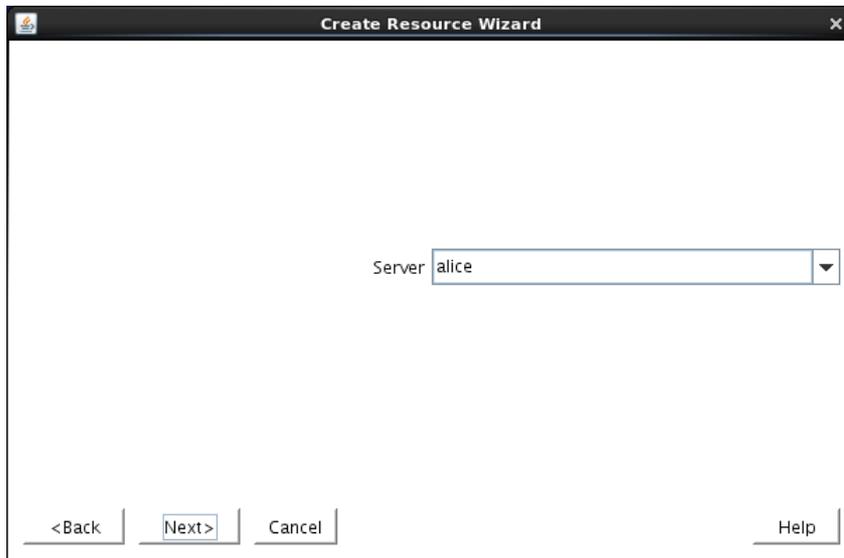
3. Please Select Recovery Kit から Recovery Kit として IP を選択し、Next>ボタンを押します。



4. Switchback Type からスイッチバックの動作として intelligent または automatic を選択し、Next>ボタンを押します。



5. **Server** からリソース階層の作成先のサーバとして稼働系サーバを選択し、**Next>**ボタンを押します。



6. **IP Resource** に仮想 IP アドレスを入力し、**Next>**ボタンを押します。

IP Resource 192.168.137.100

Enter the IP address or symbolic name to be switched by LifeKeeper. This is used by client applications to login into the parent application over a specific network interface. If a symbolic name is used, it must exist in the local /etc/hosts file or be accessible via a Domain Name Server (DNS). Any valid hosts file entry, including aliases, is acceptable. If the address cannot be determined or if it is found to be already in use, it will be rejected. If a symbolic name is given, it is used for translation to an IP address and is not retained by LifeKeeper. Both IPv4 and IPv6 style addresses are supported.

<Back Next> Cancel Help

7. **Netmask** から仮想 IP アドレスのネットマスクを選択し、**Next>**ボタンを押します。

Netmask 255.255.255.0

Enter or select a network mask for the IP resource. Any standard network mask for the class of the specified IP resource address is valid (IPv4 or IPv6 style addresses).
Note: The choice of netmask, combined with the address, determines the subnet to be used by the IP resource and should be consistent with the network configuration.

<Back Next> Cancel Help

8. **Network Interface** から仮想 IP アドレスのネットワークインタフェースを選択し、**Next>**ボタンを押します。

Network Interface

Enter or select the network interface that will be used for the IP resource being placed under LifeKeeper protection. The network interface must support the class of the IP address being protected (IPv4 or IPv6 style addresses). The default value is the first valid network interface that LifeKeeper finds on the target server that supports the class of the address being protected. Valid choices will depend on the existing network configuration and the values chosen for the IP resource address and netmask.

<Back Next> Cancel Help

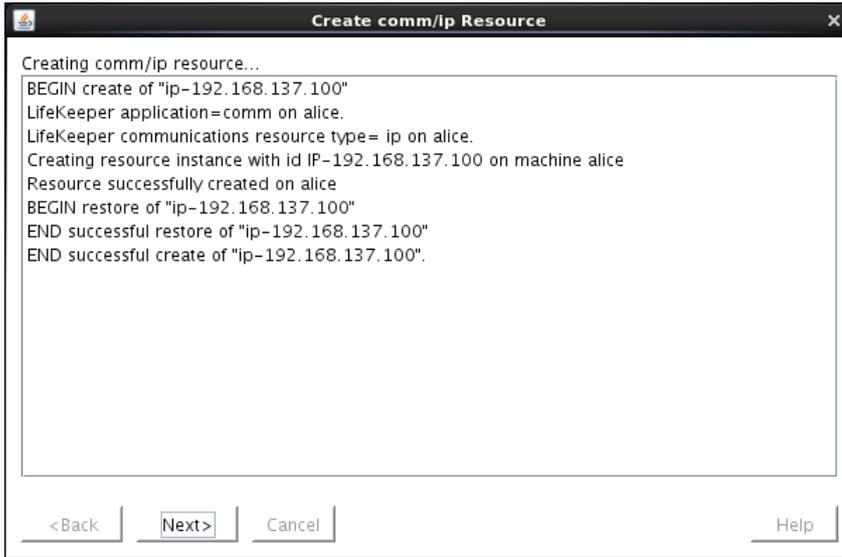
9. IP Resource Tag に IP リソースのタグ名を入力し、Create ボタンを押します。

IP Resource Tag

Enter a unique name that will be used to identify this IP resource instance on *alice*. The default tag includes the protected IP address. The valid characters allowed for the tag are letters, digits, and the following special characters:
- . /

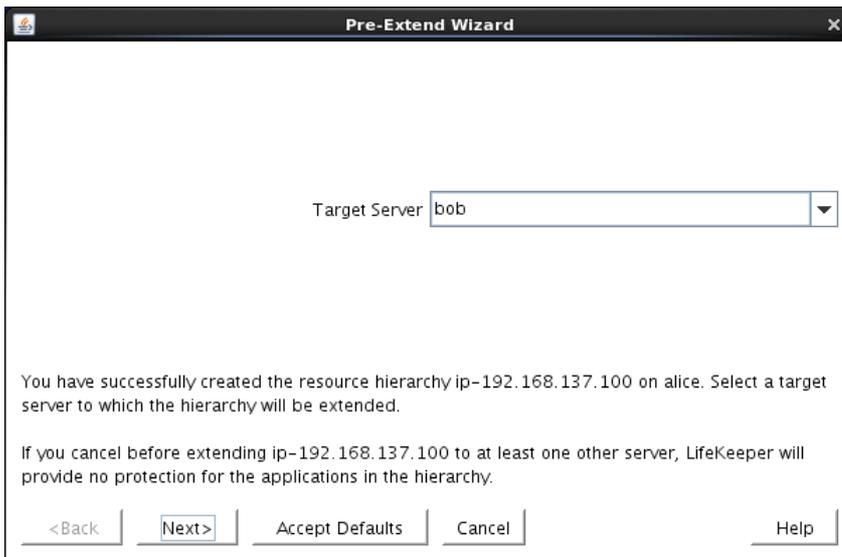
<Back Create Cancel Help

10. リソース階層が作成されます。



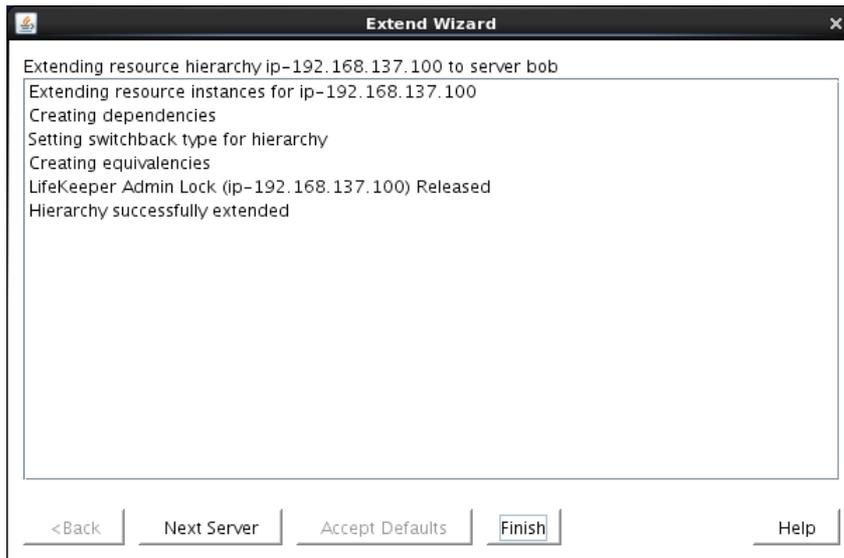
リソース階層の作成が完了したら、**Next>**ボタンを押します。

11. **Target Server** からリソース階層の拡張先のサーバとして待機系サーバを選択します。



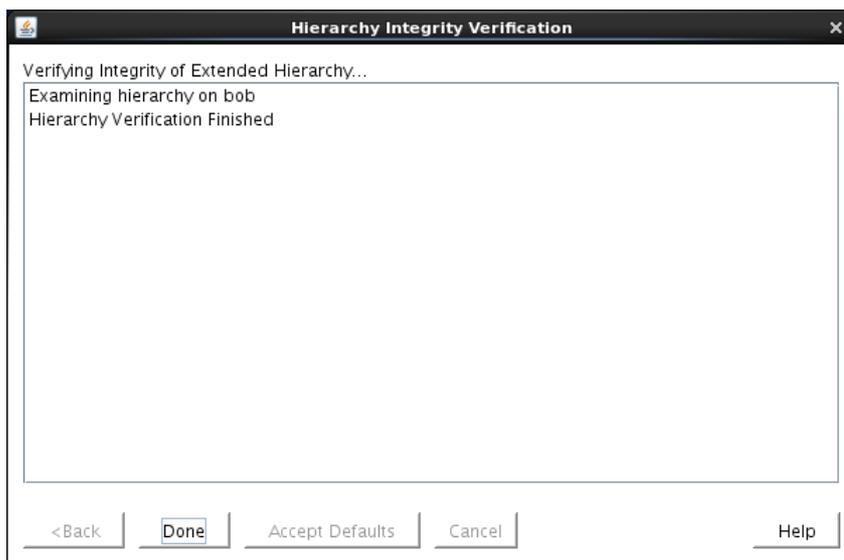
これ以降の設定を確認せずにデフォルトの設定のままで問題ない場合には **Accept Defaults** ボタンを押します。そうでない場合には **Next>**ボタンを押します。ここでは **Accept Defaults** ボタンを押した場合について説明します。

12. リソース階層が拡張されます。

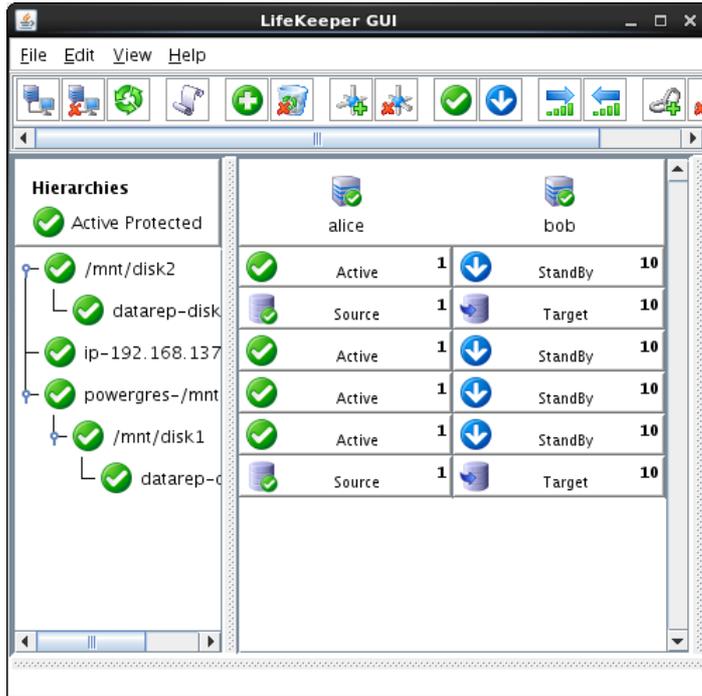


リソース階層の拡張が完了したら、**Finish** ボタンを押します。

13. **Done** ボタンを押します。



14. リソース階層ツリーに IP リソース階層が追加されます。



IP リソース階層の状態は稼働系サーバがサービス起動中(✓Active)、待機系サーバがサービス停止中(↓StandBy)になります。

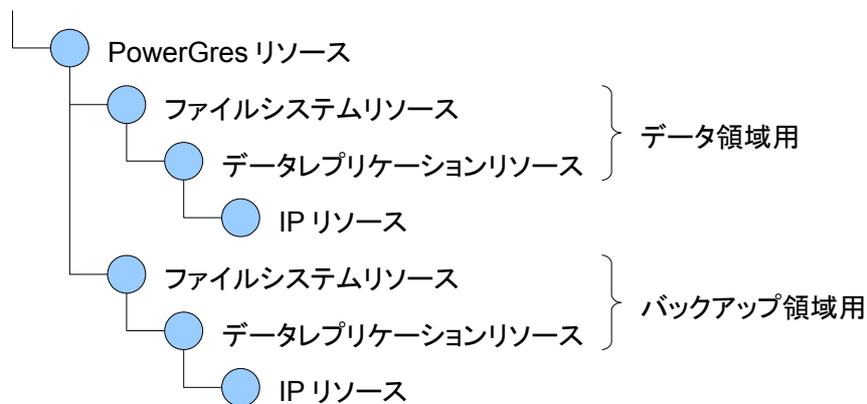
- File メニューの Exit を選択し、LifeKeeper GUI を終了します。

9. リソース依存関係の作成

リソース依存関係を作成します。詳細については「[SteelEye Protection Suite for Linux テクニカルドキュメンテーション](#)」の「[リソース依存関係の作成](#)」を参照してください。

リソースの依存関係はリソース間の親子関係です。リソースの起動時には親より子が先に起動され、リソースの停止時には子より親が先に停止されます。

ここでは、PowerGres リソースの子としてバックアップ領域用のファイルシステムリソース、データ領域用とバックアップ領域用のデータレプリケーションリソースの子として IP リソースの依存関係を作成します。これにより、IP リソースがデータレプリケーションリソースより先に起動されるようになり、スプリットブレインの発生時に IP アドレスの衝突によって両方のサーバでデータレプリケーションリソースが同時にレプリケーション元になることを防止します。



PowerGres リソースとデータ領域用のファイルシステムリソースとの依存関係は PowerGres リソースの作成時に自動的に作成されています。また、データ領域用とバックアップ領域用のデータレプリケーションリソースの子としてそれぞれ IP リソースとの依存関係がありますが、1 つの子に対して複数の親をもつ依存関係を作成できるため、データレプリケーションリソースごとに IP リソースを作成する必要はありません。

リソース依存関係の設定は以下のとおりです。

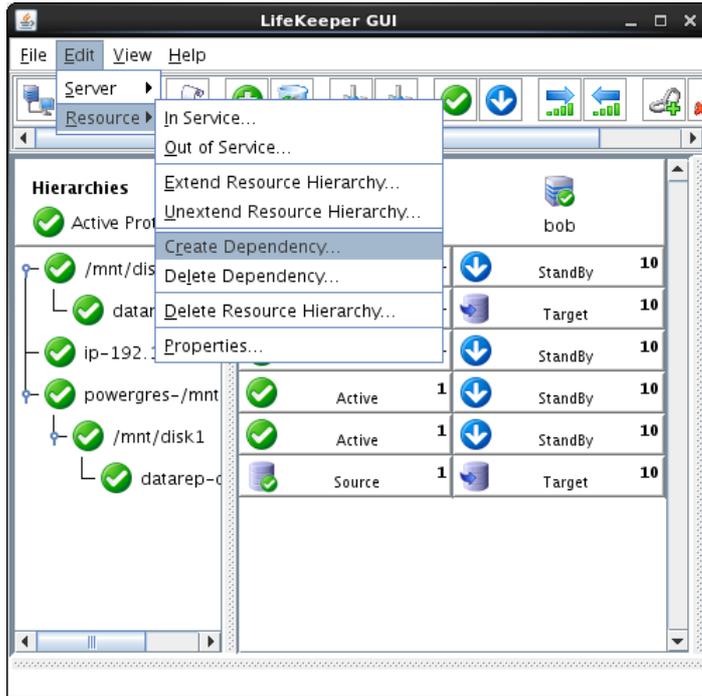
	PowerGres リソース	ファイルシステムリソース	
		データ領域用	バックアップ領域用
Server	alice	alice	alice
Parent Resource Tag	powergres- /mnt/disk1/data	/mnt/disk1	/mnt/disk2
Child Resource Tag	/mnt/disk2	ip-192.168.137.100	ip-192.168.137.100

リソース依存関係の作成は以下の手順に従って稼働系サーバにおいて root ユーザで行います。

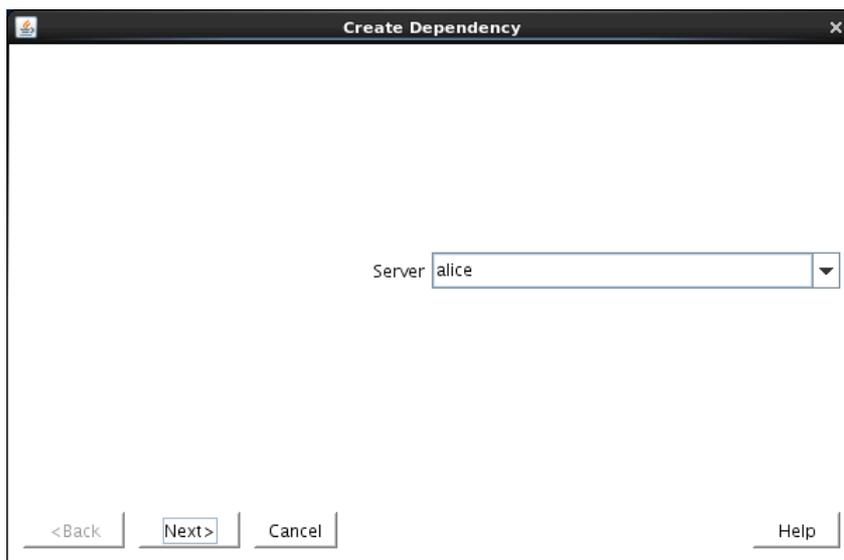
1. LifeKeeper GUI を起動し、接続します。

```
# lkGUIapp
```

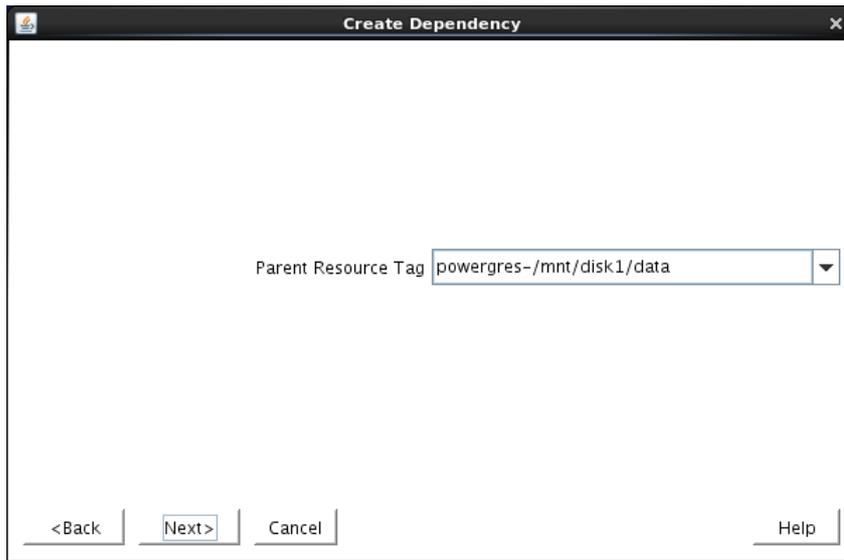
2. **Edit** メニューの **Resource ▶ Create Dependency...** を選択します。



3. **Server** からリソースの存在するサーバとして稼働系サーバを選択し、**Next>**ボタンを押します。



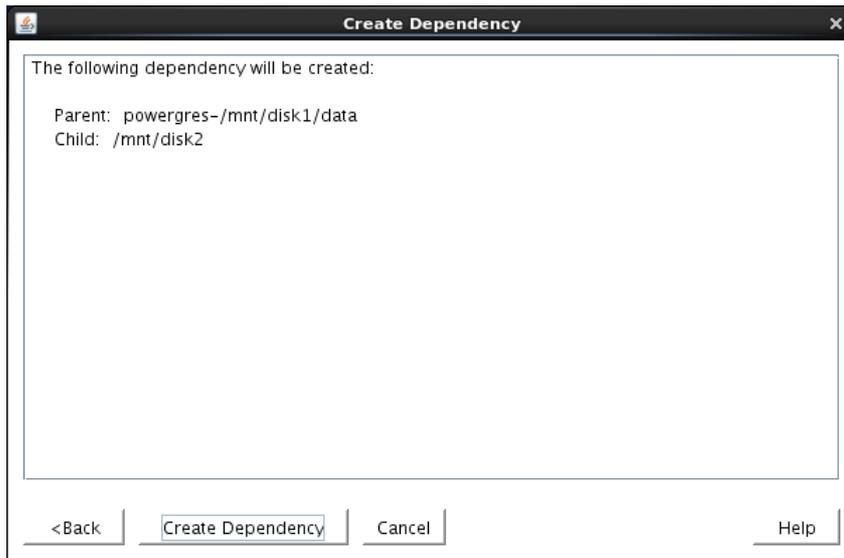
4. **Parent Resource Tag** から親のリソースを選択し、**Next>**ボタンを押します。



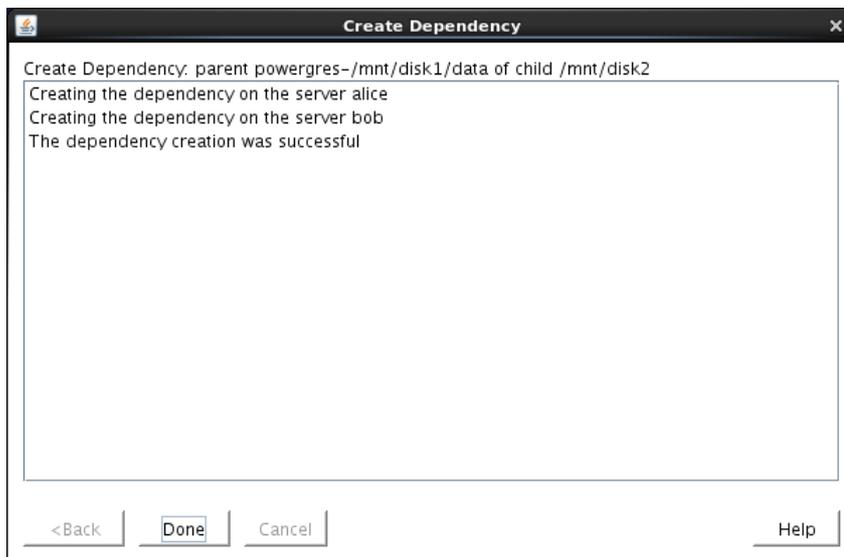
5. **Child Resource Tag** から子のリソースを選択し、**Next>**ボタンを押します。



6. **Create Dependency** ボタンを押します。

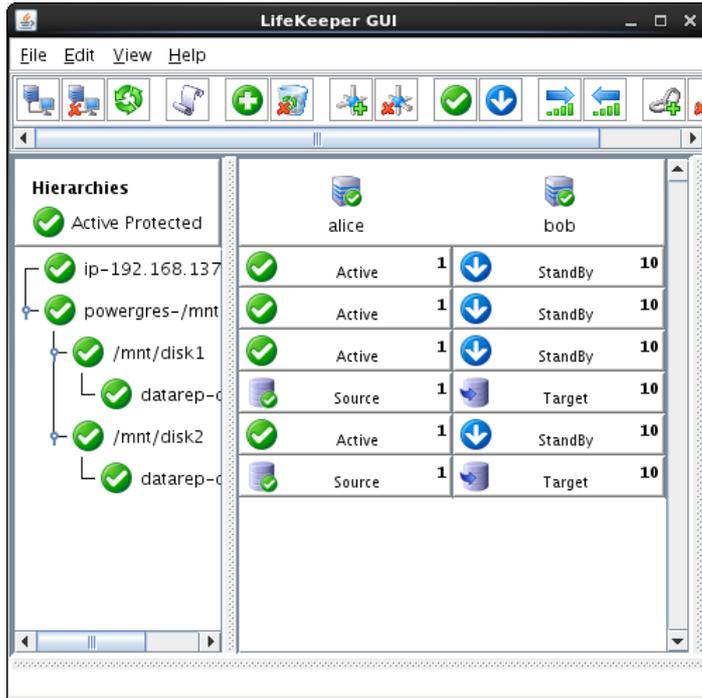


7. リソースの依存関係が作成されます。

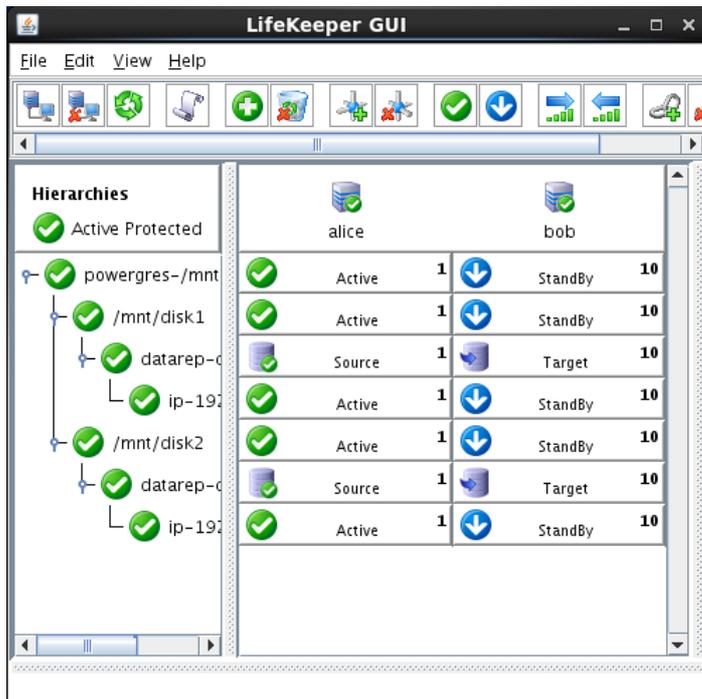


リソースの依存関係の作成が完了したら、**Done** ボタンを押します。

8. PowerGres リソースとバックアップ領域用のファイルシステムリソースとの依存関係が作成されます。



PowerGres リソースと同じようにデータ領域用とバックアップ領域用のデータレプリケーションリソースと IP リソースとの依存関係を作成します。



9. **File** メニューの **Exit** を選択し、LifeKeeper GUI を終了します。