PowerGres V13 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



2022-01-24 SRA OSS, Inc.日本支社

# 改定履歴

日付	版番号	変更内容
2022/1/24	1.0	初版

# 一 目 次 一

1	本プ	文書について	4 -
	1.1	対象読者	4 -
	1.2	関連文書	4 -
	1.3	著作権と商標	4 -
	1.4	免責事項	4 -
2	シフ	ステム構成	5 -
	2.1	構成概要	5 -
	2.2	本手順書における動作環境	6 -
3	イン	ンストール	7 -
	3.1	LIFEKEEPER のインストール	7 -
	3.2	VMDK AS SHARED STORAGE RECOVERY KIT のインストールと ESXI ホスト登録設定	8 -
	3.3	PostgreSQL Recovery Kit のインストール	9 -
	3.4	PowerGres V13 のインストール	10 -
4	VMD	)K 共有ディスクの設定	13 -
5	POW	ÆRGRES の設定	16 -
	5.1	データベースクラスタの作成	16 -
	5.2	死活監視用ユーザとパスワードファイルの作成	19 -
	5.3	スタンバイサーバの設定	20 -
6		ミュニケーションパスの作成	24 -
7	VMD	Ж リソース階層の作成	31 -
8	POW	ÆRGRES リソース階層の作成	38 -
a	ТР		- 17 -
1	0 リン	ソース依存関係の作成	52 -
1	1 <b>動作</b>	乍確認	55 -
	11.1	スイッチオーバの確認	55 -
	11.2	フェイルオーバの確認	57 -

#### 1 本文書について

本文書は、LifeKeeper によって PowerGres の HA クラスタ構成を導入する手順について説明します。

#### 1.1 対象読者

本書は PowerGres 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。

#### 1.2 関連文書

PowerGres 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。

- <u>PowerGres on Linux マニュアル</u>
- PowerGres on Linux 管理ツールマニュアル
- SIOS Protection Suite インストレーションガイド
- SIOS Protection Suite for Linux テクニカルドキュメンテーション

#### 1.3 著作権と商標

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。

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。

#### 1.4 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。

#### 2 システム構成

#### 2.1 構成概要

本書は以下図 1 に示すアクティブ・スタンバイ型の HA クラスタ構成を対象とします。



#### 図 1 構築する HA クラスタ構成

HA クラスタ構成では通常は稼働系サーバで PowerGres 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。データ領域のデータは VMDK 共有ディスクに よって待機系サーバと共有されます。

PowerGres、データ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)が開始されます。



#### 図 2 フェイルオーバ動作

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 のサービスが起動 されます。データ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。

#### 2.2 本手順書における動作環境

本手順書では以下のプラットフォーム上で構築を行うものとしています。

- Redhat Enterprise Linux 8.4 (x86\_64)
- VMware ESXi 6.0.0 Update 3

#### 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 v9.6
- VMDK as Shared Storage Recovery Kit v9.6
- PostgreSQL Recovery Kit v9.6
- PowerGres on Linux V13 Update 4

#### 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「<u>SIOS Protection Suite イ</u> ンストレーションガイド」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。

LifeKeeper のイメージメディアをマウントします。
 ここでは、イメージメディア(LKL\_V960\_110421.iso)が/media 配下に在るものとします。

# mkdi	r /mnt/I	S0						
# moun	t -t iso	9660 -о	loop	/media/LKL	_V960_11042	1.iso	/mnt/ISO	

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. イメージ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 /mnt/ISO

3. セットアップイメージをマウントします。

# # mkdir /mnt/IMG # mount -t iso9660 -o loop sps\_960.img /mnt/IMG

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/IMG

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問題ありません。  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. 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<sup>~</sup>/.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

# 3.2 VMDK as Shared Storage Recovery Kit のインストールと ESXi ホスト登録設定

VMDK as Shared Storage Recovery Kit をすべてのサーバにインストールします。 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. setup スクリプトを使用し、以下をインストールします。

<pre># cd /mnt/IMG/kits</pre>										
# ./setup										
(IkVMDK LifeKeeper	VMDK	as	Shared	Storage	Recovery	Kit	を選択し、	Enter	を押す)	

以下のメッセージが表示されたら、インストールは完了です。

Install / Upgrade was successful.

2. ESXi ホストの登録

VMDK リソースを作成する前に、ESXi ホストの情報を登録する必要があるため、以下の操作を行いま す。詳細については「VMDK as Shared Storage Recovery Kit 管理ガイド」を参照してください。

クラスタを構成する各 ESXi ホストについて以下コマンドで登録を行います。

#### # /opt/LifeKeeper/Ikadm/subsys/scsi/vmdk/bin/esxi\_register -a <ESXi ホスト名>

コマンドを実行すると、対話的にユーザ名およびパスワードを問い合わせてくるので、ESXi ホストに

ログインすることのできるユーザ・パスワードを入力します。正しくログインできなかった場合はエラ ーになり登録できません。

以下のメッセージが表示されたら、登録完了です。

#### Login successful. Registered this server.

すべての ESXi ホストが登録できたら、以下のコマンドで正しく登録できていることを確認してくだ さい。登録済みのホスト一覧が出力されます。

# /opt/LifeKeeper/Ikadm/subsys/scsi/vmdk/bin/esxi\_register -I

3. 仮想マシンオプションの設定変更をします。

VMの「設定の編集」-「仮想マシン オプション」-「詳細」-「構成パラメータ」-「構成の編集」で、 編集ダイアログを開いて下記パラメータを追加します。

+-	值
disk.enableUUID	TRUE

以下に ESXi 設定画面の例を示します。

106		v el of	(Q. 秋常 ~
IRC	(5x-90atto x (5x-90atto     x-90atto     x-90atto	9 v In	(Q.888
	+- V/(3/-90-810) sched scsi1.0 throughputCap	~ @ of	÷
	sched.scsi1.0.throughputCap	off	
	tools mast desidos autolocia		
	rood geest deskipp animous	faise	
	nvram	LK-test1.nvram	
Ininghoppereller.	pc@ridge0.present	true	
	svga present	true	
パリに入りシブ ファイル北地地します。 デールストア	pcBridge4 present	true	
所に指定したデータストアにスワップファイル未施防しま 有限マシンと目むディルクトリにスワップファイル考核	pcBridge4 virtualDev	pcieRootPort	
IDIRストに認識されないデータストアを使用すると、その Whoten パフォーマンスが低下する可能性があります。	pciBridge4 functions	8	
86088			70 アイテム
0			
			OK キャンセル
	80085 88 \$1200	RE \$7200	80082 0 87 \$720/

🥖 構成パラメータ			🥖 構成パラメータ			
◆パラメータの追加 ×パラメータの削除		(Q. 検索	◆パラメータの追加 X パラメータの削り	18	(Q. 桃弗	
キー 2/(ラメータの追加	~ 億	*	+-	~	e	~
sched.scsi1:0.throughputCap	off	0	guestinfo vmtools buildNumber		16036546	
tools guest desktop autolock	false		guestinfo appinfo		{ "version"."1", "updateCounter"."11", "publishTime"."20	È.c.
nvram	LK-test1.nvram		vmware.tools.internalversion		2147483647	
pciBridge0.present	true		vmware.tools.requiredversion		10309	
svga.present	true		migrate.hostLogState		none	
pciBridge4.present	true		migrate migrationId		0	
pciBridge4.virtualDev	pcieRootPort		migrate.hostLog		/LK-test1-b2327372.hlog	
pciBridge4.functions	8		disk.enableUUID		TRUE	0
		70 アイテム			70 ምብ	7.L
		OK キャンセル			OK +773	セル

# 3.3 PostgreSQL Recovery Kit のインストール

PostgreSQL Recovery Kit をすべてのサーバにインストールします。PostgreSQL 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. 以下をインストール (setup スクリプトより)

<pre># cd /mnt/</pre>	IMG/kits								
# ./setup									
(IkPGSQL	LifeKeeper	PostgreSQL	RDBMS	Recovery	Kit	を選択し、	Enter	を押す)	

以下のメッセージが表示されたら、インストールは完了です。

Install / Upgrade was successful.

2. セットアップイメージ、メディアをアンマウントします。

Π	C C C	
π	ou	

- # umount /mnt/IMG
- # rmdir /mnt/IMG
- # umount /media/LKL\_V960\_110421.iso

#### 3.4 PowerGres V13 のインストール

PowerGres V13 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 V13 (Linux <u>版) マニュアル</u>」の「<u>インストールとアンインストール</u>」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. ダウンロードした製品を展開します。

# tar zxvf /media/powergres-13update4-linux.tar.gz

インストールスクリプトを実行します。
 (ア)展開して作成されたディレクトリへ移動して実行します。

# cd powergres-13update4-linux

(イ)インストールスクリプトを実行

# ./install.sh

(ウ)使用許諾契約書の内容が表示されます。確認のうえ「yes」と入力します。

Do you agree to the license terms (yes/no): yes

(エ) ライセンスキーを入力します。 ライセンスキーは別途連絡されます。

#### Please enter the license key (XXXX-XXXX-XXXX): XXXX-XXXX-XXXX-XXXX

評価版の場合はライセンスキー入力箇所に Evaluation と入力してください。評価版の試用期間は 60 日となっており、試用期間を過ぎるとデータベースサーバを起動できなくなります。

(オ) PowerGres Administration Tool にログインするユーザ名、パスワードを入力します。

Please enter username and password to login to administration tool.

Username [admin]: Password: (パスワードを入力) Confirm password: (もう一度同じパスワードを入力)

(カ) PowerGres Administration Tool で使用するポート番号を入力します。

Please enter port numbers used for administration tool.

HTTP port number [80]: HTTPS port number [443]:

(キ)インストールが開始されます。 インストール時に postgres ユーザが存在しなければ、自動的に作成されます。

Installing packages for database server... ...(省略) Complete! Installing packages for administration tool... ...(省略) Complete!

(ク) ライセンスキーがインストールされます。

Installing license key... Done.

(ケ) PowerGres Administration Tool のユーザとサーバプロファイルが作成されます。

Changing configuration for administration tool... Done.

Creating user for administration tool... Done.

Creating server profile for administration tool... Done.

(⊐) PowerGres Administration Tool が起動されます。

Starting administration tool... Done.

(サ)最後に以下のメッセージが表示されれば、インストールは完了です。

PowerGres on Linux 13 installation completed.

3. PowerGres 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。

(ア) 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) groups=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。

(イ)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。

(ウ)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 V13 LD\_LIBRARY\_PATH=/opt/powergres13/lib:\$LD\_LIBRARY\_PATH PATH=/opt/powergres13/bin:\$PATH MANPATH=/opt/powergres13/share/man:\$MANPATH export LD\_LIBRARY\_PATH PATH MANPATH

#### 4 VMDK 共有ディスクの設定

VMDK 共有ディスクの設定を行います。VMDK とは、VMware 仮想マシン(VM)のハードディスクの内容 を格納する仮想ディスクであり、今回は VMDK を共有ディスクとして設定します。

- 1. 設定前に仮想マシンを停止します。
- 2. 仮想マシン1台目(test1)の設定を行います。
  - (ア) VM の「設定の編集」-「その他のデバイスの追加」-「SCSI コントローラ」をクリックし、 「SCSI パスの共有」は「物理」を選びます。

	Jシ		
- ハードディスクの追加 📧 ネッ	ットワーク アダプタの追加	📑 その他のデバイスの追加	
CPU	1 ~	iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii	
🌃 メモリ	1024 MB		
💷 ハードディスク1	20 GB	■ パラレル ボート	0
SCSI コントローラ 0	LSI Logic Parallel	USB コントローラ	0
📴 USB コントローラ 1		<ul> <li>サウンドコントローラ</li> </ul>	0
■ ネットワーク アダプタ1	VM Network	Net FIGA	ft 📀
🎟 ネットワーク アダプタ 2	dummyNW		-3 <sup>£</sup> ©
🞯 CD/DVD ドライブ 1	データストア ISO フ	アイル ~ □	接続
from 1.2			
9定の編集: LK-test1 (ESXi 6.0 仮想	<b>!マ</b> シン)		保存(キャン
後定の編集: LK-test1 (ESXi 6.0 仮想 認想ハードウェア 仮想マシンオ コードディスクの追加 ma ネッ	<b>マシン)</b> ブシ ットワーク アダブタの追加	🖻 その他のデバイスの追加	保存
&走の編集: LK-test1 (ESXi 6.0 仮想 恐想ハードウェア 仮想マシンオ ユードディスクの追加 ■ ネッ ロ CPU	マシン) ブシ ットワーク アダプタの追加 1 ~ ①	📑 その他のデバイスの追加	保存
&定の編集: LK-test1 (ESXi 6.0 仮想 想/\ードウェア 仮想マシンオ 】 八ードディスクの追加 ■ ネ・ ■ CPU ■ メモリ	<b>マシン)</b> プシ ットワーク アダプタの追加 1 ~ 1024 MB	その他のデバイスの追加	保存
&定の編集: LK-test1 (ESXi 6.0 仮想 想)/ードウェア 仮想マシンオ 】 ハードディスクの追加 ■ ネ・ ■ CPU ■ メモリ ■ ハードディスク1	Rマシン) ブシ リトワーク アダブタの追加 1 v 1024 MB 20 GB	<ul> <li>その他のデバイスの追加</li> <li>×</li> <li>×</li> </ul>	<b>保存</b> キャン
<ul> <li>(またの編集: LK-test1 (ESXi 6.0 仮想</li> <li>(思)ハードウェア 仮想マシンオ</li> <li>(ハードディスクの追加 ■ネ*</li> <li>(こ) CPU</li> <li>(二) メモリ</li> <li>(二) ハードディスク1</li> <li>(ご) SCSI コントローラ 0</li> </ul>	マシン) プシ フトワーク アダプタの追加 1 ~ ① 1024 MB 20 GB LSI Logic Parallel	<ul> <li>■ その他のデバイスの追加</li> <li>×</li> <li>×</li></ul>	保存 キャン
<ul> <li>(ESX) 6.0 仮想</li> <li>(ESX) 6.0 仮想</li> <li>(現) - ドウェア 仮想マシンオ</li> <li>(ハードディスクの追加 ■ ネック・</li> <li>CPU</li> <li>メモリ</li> <li>ハードディスク1</li> <li>SCSI コントローラ 0</li> <li>(回) 新規 SCSI コントローラ</li> </ul>	Rマシン) ブシ リトワーク アダプタの追加 1 ~ ① 1024 MB 20 GB LSI Logic Parallel LSI Logic Parallel	<ul> <li>その他のデバイスの追加</li> <li>、</li> <li>、</li></ul>	
<ul> <li>●定の編集: LK-test1 (ESXi 6.0 仮想 想)/ードウェア 仮想マシンオ</li> <li>■ パードディスクの追加 ■ネ:</li> <li>■ CPU</li> <li>■ メモリ</li> <li>■ パードディスク1</li> <li>© SCSI コントローラ 0</li> <li>**</li> <li>**</li></ul>	マシン) プシ フシ フシ コーク アダプタの追加 1024 MB 20 GB LSI Logic Parallel LSI Logic Parallel 物理	<ul> <li>■ その他のデバイスの追加</li> <li></li> <li>&lt;</li></ul>	
CPU     CPU     ズリードディスクの追加 ■ ネ・     CPU     ズリ     ハードディスクの追加 ■ ネ・     CPU     ズリ     ハードディスク1     CSSIコントローラ 0     SSSIノントローラ     SSSIパスの共有     USBコントローラ1	Rマシン) ブシ ットワーク アダプタの追加 1 ~ ① 1024 MB 20 GB LSI Logic Parallel LSI Logic Parallel 物理 USB 2.0	<ul> <li>その他のデバイスの追加</li> <li>、</li> <li>、</li></ul>	
<ul> <li>         ・ ・・・・・・・・・・・・・・・・・・・・・・・・・・・</li></ul>	Rマシン) プシ クトワークアダプタの追加 1 、 ① 1 、 ① 1024 MB 20 GB LSI Logic Parallel LSI Logic Parallel 初理 USB 2.0 VM Network	<ul> <li>この他のデバイスの追加</li> <li>、</li> <li>、</li></ul>	保存     キャン       ○     ○       ○     ○       扱続     ○

(イ)「ハードディスクの追加」をクリックし、「新規標準ハードディスク」を選び、以下を指定 し保存します。

項目	設定値
ディスクプロビジョニング	シックプロビジョニング(Eager Zeroed)
コントローラの場所	(ア)で作成した SCSI コントローラを指定
ディスクモード	独立型:通常

共有

マルチライターの共有

はパートウエア 収退マシンパン	۶			
ハードディスクの追加 飅 ネット	、ワーク アダプタの追加 🛛 📱 その他のデ	「イスの追加		
🌡 新規標準ハード ディスク 👌 🎫	⊈#л-F∓+27 V ()			
既存のハードディスク				
新規の永続的なメモリディスク	1024 MB ~			
新規 Raw ディスク	GB ~~			0
SCSI コントローラ 0	LSI Logic Parallel	Ŷ		0
🚱 新規 SCSI コントローラ	LSI Logic Parallel	~		0
SCSIバスの共有	物理	~		
😢 USB コントローラ 1		~		0
■ ネットワーク アダプタ 1	VM Network	~	☑ 接続	0
■ ☆… L ローク マグブクク 定の編集: LK-test1 (ESXi 6.0 仮想マ	う ( シン)		<b>保</b>	存 キャン
■ ☆… L ローム アダイクタ 定の編集: LK-test1 (ESXi 6.0 仮想マ □ 新規八ード ディスク	8 GB v		保	存 キャン
■ ☆… L ローム アグイクム 定の編集: LK-test1 (ESXi 6.0 仮想マ ■ 新規パード ディスク 最大サイズ	<ul> <li>S&gt;)</li> <li>8 GB ✓</li> <li>637.57 GB</li> </ul>		保	存 キャン ()
■ ☆… L ロ _ A マパオクタ 定の編集: LK-test1 (ESXi 6.0 仮想マ ■ 新規リ、ード ディスク 最大サイズ 場所	多ン) 8 GB マ 637.57 GB [detastore3rd] LK-test1/		<b>保</b> 参照	存 (キャン ③
<ul> <li>★・・・・ロームマグラクム</li> <li>定の編集: LK-test1 (ESXi 6.0 仮想マ</li> <li>新規ハードディスク</li> <li>最大サイズ</li> <li>場所</li> <li>ディスクプロビジョニング</li> </ul>	<ul> <li>(シン)</li> <li>8</li> <li>637.57 GB</li> <li>(datastore3rd] LK-test1/</li> <li>シンプロビジョニング</li> <li>シックプロビジョニング (Lazy Zerod)</li> <li>・シックプロビジョニング (Lazy Zerod)</li> <li>・シックプロビジョニング (Lager Zerod)</li> </ul>	ad) bed)	保 参照	₽
<b>本小ドロームアパゴクム</b> 定の編集: LK-test1 (ESXi 6.0 仮想マ 新規パードディスク 最大サイズ 場所 ディスクプロビジョニング シェア	<ul> <li>(シン)</li> <li>8 GB ×</li> <li>637.57 GB</li> <li>[datastore3rd] LK-test1/</li> <li>シンプロビジョニング</li> <li>シックプロビジョニング (Lazy Zeror</li> <li>● シックプロビジョニング (Eager Zero</li> <li>● シックプロビジョニング (Eager Zero</li> <li>正常 × 10</li> </ul>	ad) sed) 00 ~	保	₽ ][ +v>
<ul> <li>キャルトロ・A アダイオタ</li> <li>定の編集: LK-test1 (ESXi 6.0 仮想マ</li> <li>新規ハードディスク</li> <li>最大サイズ</li> <li>場所</li> <li>ディスクプロビジョニング</li> <li>シェア</li> <li>制限 - IOPs</li> </ul>	8     GB     く       637.57 GB     [datastore3rd] LK-test1/       ③ンプロビジョニング     シックプロビジョニング (Lazy Zeror ③ シックプロビジョニング (Lazy Zeror ④ シックプロビジョニング (Lager Zeror 正常       正常     ✓       初限なし     ✓	ad) bed)	保 参照	₽     +v>       ⊗
たの編集: LK-test1 (ESXi 6.0 仮想マ 単 新規ハードディスク 最大サイズ 場所 ディスクプロビジョニング シェア 制限 - IOPs コントローラの場所	8     GB     、       637.57 GB     637.57 GB       [datastore3rd] LK-teet1/       シンプロビジョニング       シックプロビジョニング (Lazy Zerod)       ● シックプロビジョニング (Eager Zerod)       正常     10       制限なし     、       SCSI コントローラ1     、	ad) sed) 00 ~ SI (1:0)	· · · ·	₽
	8     GB     、       637.57 GB     [datastore3rd] LK-test1/       ⑤シンプロビジョニング     ・       ⑤シックプロビジョニング (Lazy Zeror ⑥ シックプロビジョニング (Lazy Zeror ⑦ シックプロビジョニング (Lazy Zeror ⑦ シックプロビジョニング (Eager Zeror ⑦ シックプロビジョニング (Seger Zeror ⑧ ホートローラーク (Seger Zeror)       正常     ✓       10     制限なし       家CSI コントローラ1     、       SCSI コントローラ1     、       第位型: 通常     ✓	ad) ped) 00 ~ SI (1:0)	₩	₽     +v>       ⊗

3. 仮想マシン2台目(test2)の設定を行います。

(ア) VM の「設定の編集」-「その他のデバイスの追加」-「SCSI コントローラ」をクリックし、「SCSI パスの共有」は「物理」を選択します。

(イ)「ハードディスクの追加」をクリックし、「既存のハードディスク」を選び、手順2で作成 した仮想マシン1台目(test1)の追加ディスクを選択後、以下を指定し保存します。

項目	設定値
ディスクプロビジョニング	シックプロビジョニング(Eager Zeroed)
コントローラの場所	(ア)で作成した SCSI コントローラを指定
ディスクモード	独立型:通常
共有	マルチライターの共有

4. 仮想マシン2台を起動します。マルチライターなどの設定に誤りがあると、ここでの起動操作に 失敗します。 仮想マシン1台目(test1)の追加ディスクのマウント操作を行います。
 (ア)追加したディスク(/dev/sdb)を認識している事を確認します。

# # fdisk −l

(イ)ディスクをフォーマットします。

#### # mkfs -t ext4 /dev/sdb

(ウ) 共有ディレクトリにマウントします。(ここでは共有ディレクトリは /share とする)

# mount /dev/sdb /share

仮想マシン2台目(test2)の追加ディスクの認識確認を行います。
 追加したディスク(/dev/sdb)を認識している事を確認します。

## # fdisk −l

ディスクのフォーマットとマウントは、1台目で実施済みのため必要ありません。

#### 5 PowerGres の設定

PowerGres 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。

# 5.1 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「<u>PowerGres Administration Tool マニュアル</u>」 の「<u>サーバ管理</u>」を参照してください。

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 共有ディスクがマウントされているディレクトリに、データベースディレクトリ 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 /share/data	
# chmod 0700 /share/data	
# chown postgres:postgres /share/data	

- PowerGres Administration Tool にログインするため、ウェブブラウザで PowerGres Administration Tool が起動するサーバにアクセスします。
- PowerGres Administration Tool が起動するサーバにアクセスすると、「ログイン」画面が表示 されます。 インストール時に指定したユーザ名、パスワードを入力し、「ログイン」ボタンをク リックします。

PowerGres Administration Tool
ログイン
コグイン
1-54
admin
ドーマスプ
2021 SRA OSS, Inc. Japan

4. 「新しいサーバ」ボタンをクリックします。

サーバ	0				
新しいサーバ 0 件中 0 - 0 件を表	<del>.</del>				c 3
ID	サーバ名	サーバプロファイル	ポート	状態	操作
0件中0-0件を表	<b></b> <del>7</del>				¢ 3

5. 「新しいサーバ」ボタンをクリックすると、「新しいサーバ」画面が表示されます。

6. 作成するサーバの設定を入力し、「次へ」ボタンをクリックします。

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。

項目	設定値
サーバ名	postgres
ポート番号	5432
データディレクトリ	/share/data
バックアップディレクトリ(オプション)	(記入なし)
プロファイル名	PowerGres on Linux 13
作成方法	新しいデータベースクラスタを作成

# 新しいサーバ

(戻る) 次へ) キャンセル

一覧へ戻る
~

7. 「新しいデータベースクラスタ」画面が表示され、作成するデータベースクラスタの設定を入力 し、「完了」ボタンをクリックします。

項	目	設定値
スーパーユーザ ユーザ名		postgres
パスワード		<postgres ユーザのパスワード=""></postgres>
デフォルトエンコーディング		UTF8

デフォルトロケール	C
データチェックサムを有効にする	(チェックなし)
性能チューニングを行うか	有効(チェックを入れる)

新しいサーバ 新しいデータベースクラスタ

	一覧へ戻る
スーパーユーザ	
ユーザ名	
postgres	
バスワード	
バスワードの確認	
デフォルトエンコーディング	
UTF8	
デフォルトロケール	
C	
〕 データチェックサムを有効にする	
2 性能チューニングを行うか	
合計メモリ	
807MB	
<b>衰</b> 続数	
100	
データベースタイプ	
) ウェブアプリケーション	
) ジータウェアハウス	
(戻る) 完了, キャンセル	

8. サーバの作成が完了すると、完了したことを表すメッセージが表示され、作成したサーバが一覧 に追加されます。

新しいサーバが正常に作成されました。							×
ホーム	パーセーバ						
サ-	-バ	0					
新しい 1件中1	サーバ - 1 件を表示						٤ ،
	ID	サーバ名	サーバプロファイル	ボート	状態	操作	
0	8	postgres	PowerGres on Linux 13	5432	停止	【起動】 その他 ▼	
1件中1	- 1 件を表示						(; )

9. データベースクラスタを起動するため、「操作」の起動ボタンを押します。

-	+ "						
1件中1	- 1 件を表示						
	ID	サーバ名	サーバブロファイル	ボート	状態	操作	
	8	postgres	PowerGres on Linux 13	5432	停止	記動しその他・	
1 件中 1	- 1 件を表示						( )

サーバが正常に起動したメッセージが表示されます。

₩-,	くが正常に起動しました。			×

# 5.2 死活監視用ユーザとパスワードファイルの作成

PostgreSQL on Linux リカバリーキットでは PowerGres on Linux が起動していることを確認するため、サービス稼働する OS ユーザ (本構成では postgres ユーザ) にて、template1 データベースに対し て接続試験を行います。その際のパスワード入力を自動化するため、postgres のホームディレクトリに パスワードファイルを作成しておきます。

1. PowerGres on Linux データベーススーパーユーザで データベースに接続します。

# psql -U postgres

2. 死活監視用のデータベースの一般ユーザを作成します。 ここでは、死活監視用ユーザとして dbmonitor を作成します。

postgres=# CREATE ROLE dbmonitor LOGIN; CREATE ROLE

3. dbmonitor のパスワードを設定します。

```
postgres=# ¥password dbmonitor
Enter new password: (パスワードを入力)
Enter it again: (パスワードを入力)
```

 PowerGres on Linux のサービス実行 0S ユーザ (postgres) のホームディレクトリにパスワード ファイル (.pgpass) を作成します。

# su - postgres
\$ touch ~/.pgpass

5. エディタで.pgpass を以下の書式で編集を行います。

各項目には以下の設定を行います。

- 《hostname》 『localhost』を記述してください。
- 《port》

PowerGres Administration Tool で設定したポート番号を記述します。

- 《database》 『template1』を記述してください。
  - «username»

死活監視用ユーザ名を記述します。

《password》
 死活監視用ユーザのパスワードを記述します。

ここでは、以下のような記述をします。ここでは、パスワードは powergres とします。

# localhost:5432:tempalte1:dbmonitor:powergres

\$ chmod 0600 ~/.pgpass

なお、pgpass ファイルの詳細は 以下の URL を参照してください。 https://www.postgresql.jp/document/13/html/libpq-pgpass.html

## 5.3 スタンバイサーバの設定

ここでは、スタンバイサーバの PowerGres Administration Tool の設定を行います。

 スタンバイサーバの設定を行う前に、アクティブサーバにて、「5.1 データベースクラスタの作成」でマウントした共有ディスクをルートユーザでアンマウントします。 アクティブサーバの PowerGres Administration Tool にて、作成したデータベースクラスタの「操作」の 停止ボタンを押して データベースクラスタ停止後、共有ディスクをアンマウントします。

# umount /share

2. 次にスタンバイサーバにて、共有ディスクを root ユーザでマウントします。

<sup>6.</sup> ファイルのアクセス権限を設定します。

- スタンバイサーバで PowerGres Administration Tool を起動してデータベースクラスタを登録 します。
- 4. 「新しいサーバ」ボタンをクリックします。

サーバ	0				
新しいサーバ 0件中 0 - 0件を表	<b></b> <del>7</del>				x >
ID	サーバ名	サーバブロファイル	ボート	状態	操作
0件中0-0件を表	ज				¢ >

- 5. 「新しいサーバ」ボタンをクリックすると、「新しいサーバ」画面が表示されます。
- 作成するサーバの設定を入力し、「次へ」ボタンをクリックします。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。

項目	設定値
サーバ名	postgres
ポート番号	5432
データディレクトリ	/share/data
バックアップディレクトリ(オプション)	(記入なし)
プロファイル名	PowerGres on Linux 13
作成方法	既存のデータベースクラスタを追加

新しいサーバ

	一覧へ戻る
サーバ名	
postgres	
ボート番号	
5432	
データディレクトリ	
/share/data	
バックアップディレクトリ (オプション)	
プロファイル名	
PowerGres on Linux 13	~
作成方法	
○ 新しい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追加 ○ ペースパックマップからデータペースクラスタを復加	
○ ハースパックアップからテータハースクラスタを復旧 ○ レプリケーションのスタンパイサーバをセットアップ	
(戻る) びへっ キャンセル	

7. サーバの作成が完了すると、完了したことを表すメッセージが表示され、作成したサーバが一覧 に追加されます。

ホーム	ノサーバ						
サ-	-バ						
新しい	パーゼ						
新しい 1 件中 1	サーバ - 1 件を表示						<i>t. 3</i>
新しい 1 件中 1	サーバ - 1 件を表示 ID	サーバ名	サーバプロファイル	ボート	状態	操作	t y
新しい 1 件中 1 〇	サーバ - 1 件を表示 ID 3	サーバ名 postgres	サーバプロファイル PowerGres on Linux 13	<b>ポート</b> 5432	<del>状態</del> 停止	<b>操作</b> 記動 その他 ▼	6 3

8. スタンバイサーバにて共有ディスクをアンマウントします。

# umount /share	

9. スタンバイサーバでも同様に PowerGres on Linux のサービス実行 OS ユーザ (postgres) のホ ームディレクトリにパスワードファイル (.pgpass) を作成します。

#	su – postgres
\$	touch ~/.pgpass

**10.** エディタで.pgpass の編集を行います。アクティブサーバと同様に、以下のような記述をします。 localhost:5432:tempalte1:dbmonitor:powergres

11. ファイルのアクセス権限を設定します。

\$ chmod 0600 ~/.pgpass

以上でスタンバイサーバの設定が完了しました。

スタンバイサーバの設定完了後、アクティブサーバにて共有ディスクをマウントします。

	/share	/dev/sdb	rw	-0	ext4	-t	mount	#
--	--------	----------	----	----	------	----	-------	---

その後、アクティブサーバにて データベースクラスタを起動するため、PowerGres Administration Tool にログインし、「操作」の 起動ボタンを押します。

サー	-バ(						
新しい 1 件中 1	サーバ						( )
	ID	サーバ名	サーバブロファイル	オート	状態	操作	
	8	postgres	PowerGres on Linux 13	5432	停止	起動 長その他 -	
1 件中 1	- 1 件を表示						( )

サーバが正常に起動したメッセージが表示されます。

×

サーバが正常に起動しました。

### 6 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「<u>SteelEye Protection Suite for Linux</u> テ <u>クニカルドキュメンテーション</u>」の「<u>コミュニケーションパスの作成</u>」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。そうなると、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 2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。背景色が青色(■)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
Local Server	test1	test1
Remote Server(s)	test2	test2
Device Type	TCP	TCP
Device Type が TCP の場合		
Local IP Address(es)	133. 137. 175. 2	10. 10. 10. 102
Remote IP Address	133. 137. 175. 3	10. 10. 10. 103
Priority	1	2
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。

LifeKeeper GUI	LifeKeeper GUI ×
Eile Edit View Help	<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>H</u> elp
****	
Cluster Connect ×	Hierarchies
Server Name: test1	
Login: root	
Password:	
OK Cancel Help	
	C tort1 Adding app/rog surtem powergree

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。

	LifeKeeper GU		×
<u>F</u> ile <u>Edit</u> <u>V</u> iew <u>H</u> el	p		
<u>Server</u> <u>R</u> esource	Disconnect Refresh	👍 📥	0
Hierarchies None Defined	<u>V</u> iew Logs <u>C</u> reate Resource Hierarchy		Î
	<u>Create Comm Path</u> <u>D</u> elete Comm Path	_	1
	Froberties	_	
<			-
test1: Adding app/	res: custom powergres		

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



test2 Remote Server(s)	Cr	eate Communication Path	,
	Remote Server(s)	test2	
Add			

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。

Create Communication	Path >
Remote Server(s)	
Add test2	

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。

Create Comm	unication Path	×
Device Type	ТСР	×
<back next=""> Accept Defaults</back>	Cancel	Help

7. Device Type が TCP の場合

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

	Create Communication Path		×
		133.137.175.2	
		10.10.10.102	
	Local IP Address(es)		
<back next=""  =""></back>	Accept Defaults	Cancel	Help

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。

(イ)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。

Create Con	nmunication Path	×
Local Server: test1 Local IP: 133.137.175.2 Remote Server: test2		
Remote IP Addr	ang 133,137,175,3	~

(ウ)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。

Create Com	munication Path	×
Local Server: test1		
Local IP: 133.137.175.2		
Remote Server: test2		
Priorit	y 1	~
Priorit	y <u>1</u>	~

8. Device Type が TTY の場合

(ア)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。

Create Comm	unication Path	,
Local Server: test1 Remote Server: test2		
Local TTY Device	/dev/ttyS0	~

(イ) 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。

	Create Communication Path	×
Local Server: test1 Remote Server: test2		
R	emote TTY Device /dev/ttyS0	•

(ウ)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。

Create Comm	unication Path	×
Local Server: test1 Remote Server: test2		
Baud Rate	9600	~

9. コミュニケーションパスが作成されます。

	Create Comr	nunication Path		×
Local Server: test1				
Local IP: 133.137.175.2				
Remote Server: test2				
Creating TCP Communication P	ath between test1	L (133.137.175.2) and t	est2 (133.137.175.3)	
<b>Creating Communication Path</b>	from test1 to test.	2		
Successfully created network of	onnection			
to machine "tes	st2" (133.137.175.	3)		
of type "TCP"				
from test1 (133	.137.175.2).			
<b>Creating Communication Path</b>	from test2 to test	1		
Successfully created network of	onnection			
to machine "te	st1" (133.137.175.	2)		
of type "TCP"				
from test2 (133	.137.175.3).			
Deals Chronication	Annual Defeulte	Cancal	114	ala

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。

	Create Communication Path	×
It may take a few secon communication path sho	nds while a communication path is initializing before the state ows that it is ALIVE and functioning normally.	of the

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。

	LifeKeeper G	SUI	×
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>H</u> elp			
🛃 🧞 🥥	☞   ⊙	🔳   👍 🏄	0
<		Í.	>
Hierarchies			î
None Defined	test1	test2	
			- 88
			- 11
			- 11
			- 88
c tost3: Undating conversion	a taraliya		Y

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

# 7 VMDK リソース階層の作成

VMDK リソース階層を作成します。詳細については「<u>VMDK as Shared Storage Recovery Kit 管理ガイ</u>」の「<u>リソース階層の作成</u>」を参照してください。

VMDK リソース階層を作成することで、「4 VMDK 共有ディスクの設定」で作成したディスクが LifeKeeper for Linux の保護下に置かれます。ここではデータ領域用の VMDK リソース階層を作成しま す。VMDK 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	File System	
Switchback Type	intelligent	
Server	test1	
Mount Point	/share	
Root Tag	/share	
リソース階層の拡張		
Target Server	test2	
Switchback Type	intelligent	
Template Priority	1	
Target Priority	10	
Mount Point	/share	
Root Tag	/share	

1. LifeKeeper GUI を起動し、接続します。

#	lkGIII	ann
#	INUUI	app

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 File System を選択し、Next>ボタン を押します。

Create Reso	urce Wizard	×
Please Select Recovery Kit	File System	
<back next=""> Cancel</back>		Help

 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。

Create Reso	urce Wizard	×
Switchback Type	intelligent	~
<back next=""> Cancel</back>		Help

スイッチオーバはフェイルオーバ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. VMDK (File System) Resource Hierarchy で保護する Mount Point パスを選択し、Next>ボタン を押します。

Create gen/fil	esys Resource	×
Mount Point	/share	~
Select the name of the mount point for the filesyst Note: In order for a mount point to appear in the o mounted.	em that the application will depend upon. hoice list, the mount point must be currently	y

すべての「ローカル」・ファイルシステム(つまり、共有ストレージを使用するファイルシステム) お よび VMDK Recovery Kit で扱える仮想ハードディスクのマウントポイントがリストされます。ドロッ プダウン・リストから目的のマウントポイントを選択します。

7. Root Tag に、ファイルシステムリソースのタグ名を入力し、Create Instance ボタンを押します。

Create gen/filesys Resource		×
Root Tag	/share	~
Enter a unique name for the resource instance on t	the target server.	
Enter a unique name for the resource instance on t	the target server. s. digits. and the following special characters:	
Enter a unique name for the resource instance on 1 The valid characters allowed for the tag are letters /	the target server. 5, digits, and the following special characters:	

8. VMDK リソース階層の作成に成功したことを示す情報ボックスが表示されます。リソース階層を

LifeKeeper で保護するには、クラスタ内の別のサーバにリソース階層を 拡張する必要があるため、Next> をクリックします。

		Create gen/filesys Resource	>
Creating gen/1	ilesys resource	/share on test1	
/opt/LifeKeep devicehier: Us Creating reso "disk1479896 Locking SCSI scsisdev -l /de	er/lkadm/subs ing /opt/LifeKe urce instance " " successfully c /dev/sdb.	/s/gen/filesys/bin/creFShier test1/share /share int eper/lkadm/subsys/scsi/disk/bin/devicehier to cons disk1479896" with id "36000c29000b516e9ea8b83/ reated	elligent struct the hierarchy c95ca4f280":
SCSI /dev/sdł	successfully lo	ked.	

9. Pre-Extend Wizard が立ち上がります。拡張先サーバとして Target Server に待機系サーバを 選択し、Next>ボタンを押します。

Pr	e-Extend Wizard	×
Targe	t Server test2	
You have successfully created the resource the hierarchy will be extended.	hierarchy /share on test1. Select a ta	arget server to which
If you cancel before extending /share to at protection for the applications in the hiera	least one other server, LifeKeeper wil rchy.	ll provide no
<back next=""> Accept Defa</back>	ults Cancel	Help

10. 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。

Pre-Exten	d Wizard	×
Switchback Type	intelligent	~
<back next=""> Accept Defaults</back>	Cancel	Help

11. 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。

	Pre-Exten	d Wizard	×
	Template Priority	1	×
<back< th=""><th>Next&gt; Accept Defaults</th><th>Cancel</th><th>Help</th></back<>	Next> Accept Defaults	Cancel	Help

12. 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。

Pre-Exten	d Wizard	×
		1
Target Priority	10	
<back accept="" defaults<="" td=""><td>Cancel</td><td>Help</td></back>	Cancel	Help

13. 環境がチェックされ、このリソースを拡張するのに必要な要件がすべて満たされていることを示 す情報ボックスが表示されます。チェックが完了したら、Next>ボタンを押します。

	Pre-Extend Wizard	×
Executing the pre-extend sci	ript	
Building independent resou	rce list	
Checking existence of exter	nd and canextend scripts	
Checking extendability for /	share	
Pre Extend checks were suc	ccessful	
		111

満たされていない要件がある場合、 Next>ボタンが使用できなくなり、 Back ボタンが有効になります。

14. VMDK (File System) Resource Hierarchy で保護する Mount Point パスを選択し、Next>ボタン を押します。

Extend gen/filesys	Resource Hierarchy	×
Template Server: test1		
Tag to Extend: /share		
Target Server: test2		
Mount Point	/share	
Enter the name of the mount point for the filesyste	m that the application will depend upon on the	,
target server.		
Note: The mount point must be an absolute path the directory.	hat specifies a non-existent or empty, unmoun	ted
<back next=""> Accept Defaults</back>	Cancel He	lp
		1

15. 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。

Extend gen/filesys	Resource Hierarchy	×
Template Server: test1		
Tag to Extend: /share		
Target Server: test2		
Root Tag	/share	
	[ / - · · · · ·	
Enter a unique name for the resource instance on	the target server	
enter a anque name for the resource instance of	the target server.	
The valid characters allowed for the tag are letter	s, digits, and the following special characters:	
/		

16. 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。

	Extend Wizard	×
Extending resource hierarc	hy /share to server test2	
Extending resource instand	ces for /share	
Creating dependencies		
Setting switchback type fo	r hierarchy	
Creating equivalencies		
LifeKeeper Admin Lock (/sl	hare) Released	
Hierarchy successfully exte	ended	

17. [Done] をクリックして、 [リソース階層の拡張] メニューを終了します。



18. リソース階層ツリーに VMDK リソース階層が追加されます。

	LifeKeeper GUI		×
<u>E</u> ile <u>E</u> dit <u>V</u> iew <u>H</u> elp			
🗄 🧞 🚳	🕼 🚺 🗿	👍 🗼	0
<			>
Hierarchies			î
Active Protected	test1	test2	- 11
🕶 📀 /share 🛛 📀	Active 1	StandBy	10
🔗 disk1479896	Active 1	StandBy	10
			- 1
			- 1
			- 1
			- 1
			- 1
			- 1
( >			
			1150

VMDK 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。

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

# 8 PowerGres リソース階層の作成

LifeKeeper for Linux でリソースを作成します。

PowerGres は PostgreSQL と同様に LifeKeeper のリソースを構成することができます。リソースの作 成により PowerGres on Linux が LifeKeeper for Linux の保護下に置かれます。

リソース作成は root ユーザで行います。

詳細は「<u>PostgreSQL Recovery Kit</u> 管理ガイド」の「<u>PostgreSQL リソース階層の作成</u>」を参照してく ださい。

PowerGres (PostgreSQL)の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	PostgreSQL Database
Switchback Type	intelligent
Server	test1
PostgreSQL Executable Location	/opt/powergres13/bin
PostgreSQL Client Executable Location	/opt/powergres13/bin/psql
PostgreSQL Administration Executable Location	/opt/powergres13/bin/pg_ctl
PostgreSQL Data Directory	/share/data
PostgreSQL Port	5432
PostgreSQL Socket Path	/var/run/postgresql/.s.PGSQL.5432
PostgreSQL Database Administrator User	dbmonitor
PostgreSQL Logfile	/tmp/pgsql-5432.lk.log
PostgreSQL Database Tag	pgsql-5432
リソース階層の拡張	
Target Server	Test2
Switchback Type	intelligent
Template Priority	1
Target Priority	10
PostgreSQL Executable Location	/opt/powergres13/bin
PostgreSQL Database Tag	pgsqI-5432

1. LifeKeeper GUI を起動し、接続します。

# IkGUIapp

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

	LifeKeeper GUI		×
Eile Edit View Help			
E Server >	<u>D</u> isconnect <u>R</u> efresh	👍 🍌	C
<	<u>V</u> iew Logs		1
Hierarchies	<u>C</u> reate Resource Hierarchy.	test2	
/share	<u>C</u> reate Comm Path <u>D</u> elete Comm Path	StandBy	10
o disk1479896	Properties	StandBy	10

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

Create Resource Wizard	×
Please Select Recovery Kit PostgreSQL Database	·

 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。

Create Reso	ource Wizard	×
Server	test1	~
<back next=""> Cancel</back>		Help

6. PostgreSQL Executable Location に PowerGres の実行可能ファイルを含むディレクトリパスを 指定し、Next>ボタンを押します。

ipgsqi nesource	,
/opt/powergres13/bin	·
reSQL utilities pg_ctl and psql.	
	/opt/powergres13/bin

 PostgreSQL Client Executable Location に PowerGresの psql コマンドのフルパスを指定し、 Next>ボタンを押します。



 PostgreSQL Administration Executable Location に PowerGres の pg\_ctl コマンドのフルパス を指定し、Next>ボタンを押します。

Create database	/pgsql Resource	>
PostereSQL Administration Executable Location	/opt/powergres13/bin/og_ct1	~
	, - p + p	1000
Specify the full path to the PostgreSQL utility pg_c	а.	

9. PostgreSQL Data Directory に LifeKeeper の保護下に置かれる PostgreSQL データディレク トリ (datadir) の場所を指定し、Next>ボタンを押します。指定されたディレクトリは、共有フ ァイルシステムに存在し、常駐している必要があります。

		Create database	/pgsql Resource	×
	Pos	tgreSQL Data Directory	/share/data	~
Specify the	location of the	PostgreSQL data direct	ory (datadir) that will be placed	l under LifeKeeper
protection.	The specified d	latadir must exist and mi	ust be located on a shared files	ystem.

10. PostgreSQL Port 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。

(	Create database	/pgsql Resource	×
	PostgreSQL Port	5432	~
Specify the TCP/IP port number	on which the postr	master daemon is listening or	n for connections
rom client applications.			
<back next=""> Car</back>	ncel		Help

11. PostgreSQL Socket Pathに ポストマスタデーモンがクライアントアプリケーションからの接続 をリスニングする UNIX ドメインソケットへのフルパスを指定し、Next>ボタンを押します。

	/pgsql Resource	>
PostgreSQL Socket Path	/var/run/postgresql/.s.PGSQL.5432	~
Snerify the full nath of the Unix-domain socket on	which nostmaster is listening for connecti	ions from
Specify the full path of the Unix-domain socket on client applications. The default is normally /tmp/.s.	which postmaster is listening for connecti PGSQL.5432.	ions from

12. PostgreSQL Database Administrator User に データベースインスタンスへの接続テストで使われる PostgreSQL データベースユーザ名を入力し、Next>ボタンを押します。

ここでは、「5.2 死活監視用ユーザとパスワードファイルの作成」で作成した dbmonitor を指 定します。

Enter Database Administrator User dbmonitor Enter a PostgreSQL Database Administrator User for the instance using datadir / <b>share/data</b> . This			Create database/pgsql Resour	ce >
Enter a PostgreSQL Database Administrator User for the instance using datadir /share/data. This				
Enter a PostgreSQL Database Administrator User for the instance using datadir <b>/share/data</b> . This			dbmonitor	r ]
Enter a PostgreSQL Database Administrator User for the instance using datadir / <b>share/data</b> . This				
Enter a PostgreSQL Database Administrator User for the instance using datadir /share/data. This				
Enter a PostgreSQL Database Administrator User for the instance using datadir /share/data. This				
THEAT ACCOUNT TO HET INCLUDES SUPERIO DECEMPTING FROM THE THEAT ACTIVATES INCLUDED	Enter a Pos	tgreSQL Data	base Administrator User for the instance	using datadir <i>/share/data</i> . This

13. PostgreSQL Logfile に PostgreSQL の起動や停止に使用する pg\_ctl の -1 オプションで使わ れるログファイルパスを指定し、Next>ボタンを押します。

Create database	/pgsql Resource	×
PostgreSQL Logfile	/tmp/pgsql-5432.lk.log	~

14. PostgreSQL Database Tag に PowerGres リソースのタグ名を入力し、Create ボタンを押しま す。

Create database	/pgsql Resource	×
PostgreSQL Database Tag	pgsql-5432	

15. PowerGres リソース階層の作成に成功したことを示す情報ボックスが表示されます。リソース階 層を LifeKeeper で保護するには、クラスタ内の別のサーバにリソース階層を拡張する必要があ るため、Next>をクリックします。

Creating database/PostgreSQL resource BEGIN rerate of "psql-5432" on server "test1" Information: The path /opt/powergres13/bin will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device. Information: The path /ntm/posgresQl will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device. Information: The path /ntm/posgreSQLklog will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device. EfGIN restore of "psgql-5432" on server "test1" END successful restore of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1"		Create database/pgsql Resource	
BEGIN rerate of "psgl-5432" on server "test1"         Information: The path /opt/powergres13/bin will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device.         Information: The path /nav/run/postgresql will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device.         Information: The path /nm/psgl=5432.lk.log will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device.         BEGIN restore of "psgl=5432" on server "test1"         END successful restore of "psgl=5432" on server "test1"         END successful create of "psgl=5432" on server "test1"         END successful create of "psgl=5432" on server "test1"	Creating database/Post	reSQL resource	
Information: The path /pot/powergres13/bin will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device. Information: The path /var/nur/postgresql will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device. Information: The path /tmp/pgsql-5432? Ik.log will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device. BEGIN restore of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1"	BEGIN create of "pgsql-	i432" on server "test1"	
located on a shared filesystem or shared device. Information: The path /var/run/postgresql/will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device. Information: The path /tm/pgsql-5432! Nog will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device. BEGIN restore of "pgsql-5432" on server "test1" END successful restore of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1"	Information: The path /	pt/powergres13/bin will not be protected by LifeKeeper because it is	not
Information: The path /var/run/postgresql will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device. Information: The path /tmp/pgsql-5432.lk.log will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device. BEGIN restore of "pgsql-5432" on server "test1" END successful restore of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1"	located on a shared files	ystem or shared device.	
located on a shared filesystem or shared device. Information: The path /tmp/pgsql-5432.lk0g will not be protected by LifeKeeper because it is not located on a shared filesystem or shared device. BEGIN restore of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1"	Information: The path /	ar/run/postgresql will not be protected by LifeKeeper because it is no	,t
Information: The path / tmp/pgsql-5432.lk.log will not be protected by LifeKeeper because it is not located on a shared diveve. BEGIN restore of "pgsql-5432" on server "test1" END successful restore of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1"	located on a shared file:	ystem or shared device.	
located on a shared filesystem or shared device. BEGIN restore of "pgsql-5432" on server "test1" END successful restore of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1" - Back	Information: The path /1	mp/pgsql-5432.lk.log will not be protected by LifeKeeper because it is	not
BEGIN restore of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1" END successful create of "pgsql-5432" on server "test1"	located on a shared files	ystem or shared device.	
END successful create of "pgsql-S432" on server "test1" END successful create of "pgsql-S432" on server "test1"	BEGIN restore of "pgsqi	5432" on server "test1"	
<back cancel="" help<="" td=""><td>END successful restore</td><td>if "pgsql-5432" on server "test1"</td><td></td></back>	END successful restore	if "pgsql-5432" on server "test1"	
<back next=""> Cancel Help</back>	END SUCCESSIUL Create o	pgsql-5432" on server "test1"	
<back next=""> Cancel Help</back>			
<back next=""> Cancel Help</back>			
<back next=""> Cancel Help</back>			
<back next=""> Cancel Help</back>			
<back next=""> Cancel Help</back>			
<back next=""> Cancel Help</back>			
	<back next=""></back>	Cancel	Help

16. Pre-Extend Wizard が立ち上がります。拡張先サーバとして Target Server に待機系サーバを 選択します。

		Pre-Exter	nd Wizard	>
		Target Server	test2	
You have suc which the hie	cessfully creat rarchy will be	ed the resource hierarc extended.	hy pgsql-5432 on test1. Select a ta	rget server to
You have suc which the hie If you cancel protection fo	cessfully creat rarchy will be before extend r the applicatio	ed the resource hierarc extended. ing pgsql-5432 to at lea: ons in the hierarchy.	hy pgsql-5432 on test1. Select a ta st one other server, LifeKeeper wil	rget server to I provide no

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。

17. 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。



18. 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。

	Pre-Extend Wizard	>
	Template Priority 1	~
		A
<back ne<="" td=""><td>xt&gt; Accept Defaults Cancel</td><td>Help</td></back>	xt> Accept Defaults Cancel	Help

19. 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。

Pre-Exten	d Wizard	×
Tourse Believille	10	
Target Phoney	10	, in the second se
<back next=""> Accept Defaults</back>	Cancel	Help

20. 環境がチェックされ、このリソースを拡張するのに必要な要件がすべて満たされていることを示 す情報ボックスが表示されます。チェックが完了したら、Next>ボタンを押します。

	Pre-Extend Wizard	د
Executing the pre-extend s	cript	
Building independent reso	urce list	
Checking existence of ext	and and canextend scripts	
Checking extendability for	pgsql-5432	
The path /var/lib/pgsql/.p	zpass is not a valid file.	
The /var/lib/pgsql/.pgpass	file does not contain required entries for database of	connectivity. Verify the
file contains proper entrie	s for all password protected instances.	
Pre Extend checks were s	uccessful	
De als I responses	Annual Defaultes   Connect	Itala
<back next=""></back>	Accept Defaults Cancel	Help

21. PostgreSQL Executable Location に PostgreSQL 実行可能ファイルを含むディレクトリパスを 指定し、Next>ボタンを押します。

Extend database	e/pgsql Resource	×
Template Server: test1		
Tag to Extend: pgsql-5432		
Target Server: test2		
PostgreSQL Executable Location	/opt/powergres13/bin	~
Specify the directory path that contains the Postgr	eSQL utilities pg_ctl and psql.	
<back next=""> Accept Defaults</back>	Cancel	Help

22. PostgreSQL Database Tag に PowerGres リソースのタグ名を入力し、Next>ボタンを押します。

Extend database	/pgsql Resource	×
Template Server: test1		
Tag to Extend: pgsql-5432		
Target Server: test2		
Portero COL Databaco Tag	perel 5422	
PostgresQL Database rag	hR2di-2425	
Enter a unique name for the recourse on test? The	unlid characters allowed for th	o namo avo
letters, digits, and the following special characters:	valid characters allowed for th	e name are

23. 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。

	Extend Wizard	×
Extending resource hierarc	hy pgsql-5432 to server test2	
Extending resource instan	ces for pgsql-5432	
Creating dependencies		
Setting switchback type for	r hierarchy	
Creating equivalencies		
LifeKeeper Admin Lock (p	sql-5432) Released	
Hierarchy successfully ext	ended	
<back next="" serv<="" td=""><td>er Accept Defaults Finish</td><td>Help</td></back>	er Accept Defaults Finish	Help
		- <b>K</b>

24. [Done] をクリックして、 [リソース階層の拡張] メニューを終了します。



25. リソース階層ツリーに PowerGres リソース階層が追加されます。

LifeKeeper GUI				
<u>E</u> ile <u>E</u> dit ⊻iew <u>H</u> elp				
🐌 🛃 🚳	S   O			0
<				
Hierarchies				
Active Protected	test1		test2	
🗸 📀 pgsql-5432	Active	1	StandBy	10
🕶 📀 /share 🛛 🔇	Active	1	StandBy	10
🕑 disk1479896 🄇	Active	1	StandBy	10
				_
· · · · · · · · · · · · · · · · · · ·				

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

#### 9 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「<u>IP Recovery Kit 管理ガイド</u>」の「<u>IP リソース階</u> <u>層の作成</u>」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 10.10.10.110 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	test1	
IP Resource	10. 10. 10. 100	
Netmask	255. 255. 255. 0	
Netmask Interface	ens34	
IP Resource Tag	ip-10. 10. 10. 100	
リソース階層の拡張		
Target Server	test2	
Switchback Type	intelligent	
Template Priority	1	
Target Priority	10	
IP Resource	10. 10. 10. 100	
Netmask	255. 255. 255. 0	
Netmask Interface	ens34	
IP Resource Tag	ip-10. 10. 10. 100	

1. LifeKeeper GUI を起動し、接続します。

#	IkGUI	app
---	-------	-----

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。

	Create Resource Wizard	
	Please Select Recovery Kit IP	
<back n<="" td=""><td>ext&gt; Cancel</td><td>Help</td></back>	ext> Cancel	Help

 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。



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。



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。



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。



10. リソース階層が作成されます。

Create comm/ip Resource	>
Creating comm/ip resource	
BEGIN create of "ip-10.10.10.100"	
LifeKeeper application=comm on test1.	
LifeKeeper communications resource type= ip on test1.	
Creating resource instance with id IP-10.10.10.100 on machine test1	
Resource successfully created on test1	
BEGIN restore of "ip-10.10.10.100"	
END successful restore of "ip-10.10.10.100"	
END successful create of "ip-10.10.10.100".	
Parts   margared formal	
<back cancel<="" next2="" td=""><td>негр</td></back>	негр

- 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。

Pre-Exter	id Wizard	×
Target Server	test2	~
You have successfully created the resource hierarch to which the hierarchy will be extended.	ıy ip-10.10.10.100 on test1. S	elect a target server
If you cancel before extending ip-10.10.10.100 to at	least one other server, LifeK	eeper will provide no
protection for the applications in the hierarchy.		

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。

Verifying Integrity of Extended Hierarchy Examining hierarchy on test2 Hierarchy Verification Finished		Hierarchy Integrity Verification	×
Examining hierarchy on test2 Hierarchy Verification Finished	Verifying Integrity	of Extended Hierarchy	
Hierarchy Verification Finished	Examining hierarc	ny on test2	
	Hierarchy Verificat	ion Finished	
<back accept="" cancel="" defaults="" done="" help<="" td=""><td><back d<="" td=""><td>one Accept Defaults Cancel</td><td>Help</td></back></td></back>	<back d<="" td=""><td>one Accept Defaults Cancel</td><td>Help</td></back>	one Accept Defaults Cancel	Help

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。

	LifeKeeper G	IUI		×
<u>E</u> ile <u>E</u> dit <u>V</u> iew <u>H</u> elp				
1. L O	S 0		* *	
<				>
Hierarchies	1		1	î
Active Protected	test1		test2	
🧭 ip-10.10.10.100	Active	1	StandBy	10
🕶 📀 pgsql-5432 🛛  🤇	Active	1	StandBy	10
🕶 🚫 /share 🛛 🦉	Active	1	StandBy	10
🧭 disk1479896	Active	1	StandBy	10
				- 1
				- 1
				- 1
<>				~

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

#### 10 リソース依存関係の作成

PowerGres リソースと IP リソースの関連づけるため、リソース依存関係を作成します。

詳細については「<u>SIOS Protection Suite for Linux テクニカルドキュメンテーション</u>」の「<u>リソー</u> ス依存関係の作成」を参照してください。

リソース依存関係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。

リソース依存関係の作成		
Server	test1	
Parent Resource Tag	pgsq1-5432	
Child Resource Tag	ip-10. 10. 10. 100	

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。

ile Edit View Help		
Server >		
Resource >	In Service	our
	Out of Service	
Hierarchies	Extend Resource Hierarchy	1
Active Protectec	Unextend Resource Hierarchy	test2
🧭 ip-10.10.10.100		StandBy 10
Desal-5432	Create Dependency	StandBy 10
- O /chara	Delete Dependency	10
• O /snare	Delete Resource Hierarchy	StandBy
🕑 disk14798	Properties	StandBy 10
	Eroperticum	

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。

ここでは PowerGres リソースのタグを指定します。

	Create De	pendency	×
	Parent Resource Tag	pgsql-5432	
<back next=""></back>	Cancel		Help

- 5. Child Resource Tag にはリソースを関連付けるときに子になるリソースのタグを選択し、Next> ボタンを押します。
  - ここでは IP リソースのタグを指定します。

Create De	pendency	×
Child Resource Tag	ip-10.10.10.100	
<back next=""> Cancel</back>		Help

6. 確認画面が表示され、Create Dependency ボタンを押します。

Create Dependency	×
The following dependency will be created:	
Parent: pgsql-5432	
Child: ip-10.10.10.100	
-Rack Create Dependency Cancel	Holp
STRUE A LEGATE DELEDUESD A ST A RUD EL	1.17711.2

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. IP リソースが PowerGres リソースに関連付けられたことを確認してください。

	🖓   🗿	2	* *	
Hierarchies	test1		test2	
🔗 pgsql-5432	Active	1	StandBy	10
- 🖉 /share	Active	1	StandBy	10
🔗 disk1479896	Active	1	StandBy	10
🧭 ip-10.10.10.100 🚺	Active	1	StandBy	10

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

#### 11 動作確認

初期設定が完了したことによって PowerGres on Linux が実際に LifeKeeper for Linux の保護下に 置かれていることを確認します。

### 11.1 スイッチオーバの確認

1. PowerGres on Linux が起動していることを確認します。

PowerGres on Linux が起動しているサーバでデータベースデーモン (postgres) が起動していることを確認します。以下のようにプロセスが存在することを確認してください。

<pre># ps -C postgres</pre>	
PID TTY	TIME CMD
1280702 ?	00:00:00 postgres
1280705 ?	00:00:00 postgres
1280707 ?	00:00:00 postgres

PowerGres on Linux をスイッチオーバします。
 LifeKeeper GUI を起動し、接続します。

# lkGUIapp

Standby 側の PowerGres リソースにて 右クリック後、In Service を押します。



対象となる Server と Resource が表示され、In Service ボタンを押すと PowerGres on Linux がス イッチオーバします。

In Service	>
Confirm in service action for	
Server: test2	
Resource: pgsql-5432 (test1.pgsql-5432)	
<back cancel<="" in="" service="" td=""><td>Help</td></back>	Help

詳細は「<u>PostgreSQL Recovery Kit</u>管理ガイド」の「<u>LifeKeeper GUI からの手動スイッチオーバの</u> 実行」を参照してください。

スイッチオーバが完了したら、Done ボタンを押します。

×
^

3. スイッチオーバされたことを確認します。

全てのリソースが先ほどまで Standby だったサーバへ移っていることが確認できます。

LifeKeeper GUI					;
ile <u>E</u> dit <u>V</u> iew <u>H</u> elp					
P. P. O	1210	) 🗟			10
	1 4 1		1 01		
Hierarchies					
Active Protected	test1			test2	
ogsql-5432	StandB	sy 1	0	Active	10
🕶 🧭 /share	StandB	sy 1	0	Active	10
odisk1479896	StandB	sy 1	0	Active	10
🧭 ip-10.10.10.100	StandB	sy 1	0	Active	10

また、先ほどまで Active だったサーバではデータベースデーモンがなくなっていることを確認しま す。

以下のようにプロセスがなくなっていることを確認してください。

# ps -C postgres	
PID TTY	TIME CMD

さらに、スイッチオーバされたサーバにてデータベースデーモンが起動していることを確認します。 以下のようにプロセスが存在することを確認してください。

 # ps -C postgres

 PID TTY
 TIME CMD

 92075 ?
 00:00:00 postgres

 92077 ?
 00:00:00 postgres

 92103 ?
 00:00:00 postgres

# 11.2 フェイルオーバの確認

1. Active サーバの PowerGres on Linux が実際に起動していることを確認します。

# ps −C postgres	
PID TTY	TIME CMD
1315968 ?	00:00:00 postgres
1316019 ?	00:00:00 postgres
1316041 ?	00:00:00 postgres

2. PowerGres on Linux をフェイルオーバします。

PowerGres on Linux が起動しているサーバでデータベースデーモンを強制的に停止します。

なお、pid にはデータベースデーモンのプロセス ID を指定します。

# kill <pid></pid>	
<pre># ps -C postgres</pre>	
PID TTY	TIME CMD

データベースデーモンを強制的に停止しても、データベースデーモンがローカルリカバリされ、 しばらくすると自動的に復帰することが確認できます。

# ps −C postgre	es
PID TTY	TIME CMD
1321389 ?	00:00:00 postgres
1321403 ?	00:00:00 postgres
1321411 ?	00:00:00 postgres

そこで、ローカルリカバリできないようデータベースデーモンのファイル名を一時的に変更し、再び データベースデーモンを強制的に停止します。

# mv /opt/powergres13/bin/postgres /opt/powergres13/bin/postgres~ # kill <pid>

しばらくすると PowerGres on Linux が優先度の高いサーバへフェイルオーバします。

3. LifeKeeper GUI を起動し、接続します。

#### # IkGUIapp

4. フェイルオーバされたことを確認します。

LifeKeeper GUI にて Active サーバのリソースが Standby へ変更され、Standby サーバのリソース が Active に移り変わることが確認できます。

	LifeKeeper 0	GUI		×
<u>E</u> ile <u>E</u> dit ⊻iew <u>H</u> elp				
<b>L L</b> Ø			*	0
<		1		,
Hierarchies	1			î
Active Protected	test1		test2	
🕶 📀 pgsql-5432	StandBy	1 ⊘	Active	10
🕶 📀 /share 🛛	StandBy	1	Active	10
odisk1479896	StandBy	1	Active	10
🧭 ip-10.10.10.100	StandBy	1	Active	10
				- 1
				- 1
				- 1
· · · · ·				Y

また、Active に移り変わったサーバではデータベースデーモンの起動が確認できます。

# ps −C po	ostgres
PID TTY	TIME CMD
95448 ?	00:00:00 postgres
95450 ?	00:00:00 postgres
95473 ?	00:00:00 postgres

なお、フェイルオーバ確認のために変更したファイル名は必ず元に戻してください。

# mv /opt/powergres13/bin/postgres~ /opt/powergres13/bin/postgres