

AWS 上 PowerGres Plus/LifeKeeper 導入手順書

バージョン 1.0



2023-05-16

SRA OSS 合同会社

改定履歴

日付	版番号	変更内容
2023/5/16	1.0	初版

目次

1	本文書について	- 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	オンプレミスとクラウドの違い	- 5 -
3	システム構成	- 6 -
3.1	構成概要	- 6 -
3.2	本手順書における動作環境	- 6 -
3.3	AWS 上の設定	- 6 -
4	インストール	- 7 -
4.1	LifeKeeper のインストール	- 7 -
4.2	PowerGres Plus V13 のインストール	- 10 -
5	コミュニケーションパスの作成	- 14 -
6	DataKeeper リソースの作成	- 20 -
7	PowerGres Plus の設定	- 31 -
7.1	データベースクラスタの作成	- 31 -
7.2	死活監視用ユーザとパスワードファイルの作成	- 34 -
7.3	待機系サーバの設定	- 35 -
7.4	データ暗号化の設定	- 39 -
8	PowerGres Plus リソース階層の作成	- 45 -
9	IP リソース階層の作成	- 56 -
10	EC2 リソース階層の作成 (Elastic IP シナリオ)	- 57 -
10.1	EC2 リソース作成の事前設定	- 57 -
10.2	EC2 リソースの作成	- 57 -
11	リソース依存関係の作成	- 63 -
12	動作確認	- 66 -
12.1	スイッチオーバーの確認	- 66 -
12.2	フェイルオーバーの確認	- 68 -

1 本文書について

本文書は、AWS 上の 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。

1.1 対象読者

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。

1.2 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 \(Linux 版\) V13 マニュアル](#)
- [PowerGres 管理ツールマニュアル](#)
- [LifeKeeper for Linux インストレーションガイド](#)
- [LifeKeeper 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 合同会社に帰属します。本書の一部または全部を SRA OSS 合同会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 合同会社は、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 合同会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 オンプレミスとクラウドの違い

オンプレミスとは、自社内にサーバー室、ハードウェアを保有し、自社にて設備調達から行い、配線、ネットワーク設計、システム設計・運用、サーバー保守まで自社で行うことを指します。その反対で、インターネットなどのネットワークを経由して、自社にハードウェアを置かなくても、いつでもどこでもサービスを使いたければ、接続してデータやソフト等を利用できることはクラウドになります。

オンプレミス環境にワークロードが配置されている場合、通常これらのワークロードの配置場所は、確立されたデータセンターの場所によって定義されます。多くの場合、ワークロードをホストするために別の場所を選ぶという選択肢はありません。しかし、パブリッククラウドでは、幅広い地域と可用性ゾーンを選択することができます。

PowerGres Plus と LifeKeeper を用いる場合、オンプレミス環境ではクラスターノード間のデータ共有に共有ストレージが使われることが一般的ですが、物理的な共有ストレージが使えないパブリッククラウド環境では、「LifeKeeper/DataKeeper」を組み合わせることでオンプレミスと同様にクラウド環境でも HA クラスターを構築できます。

PowerGres Plus + LifeKeeper/DataKeeper on AWS の利点は下記となります。

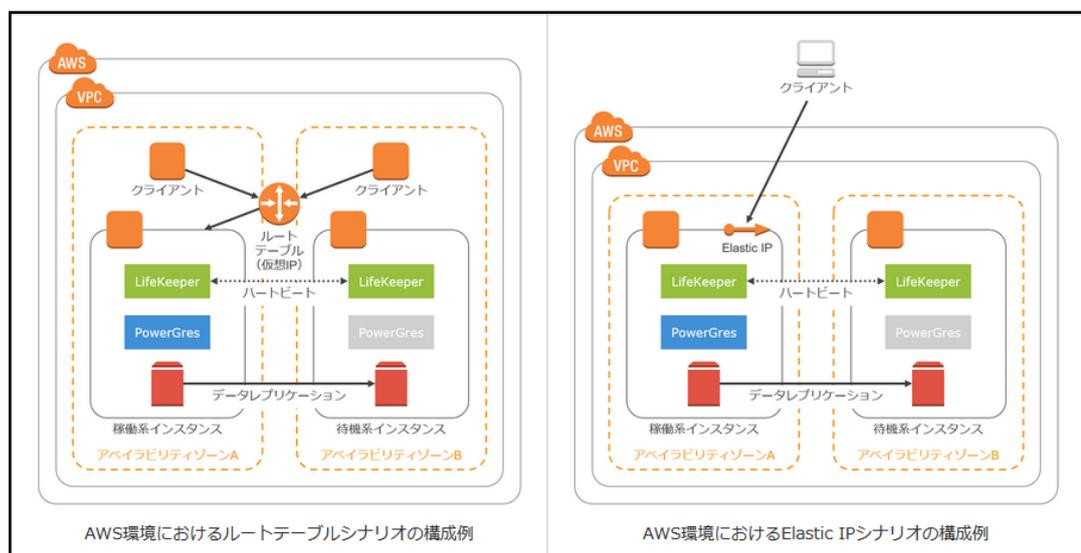
- EC2 インスタンス内のアプリケーションの監視と障害時のフェイルオーバーが可能
- 多彩なオプション製品 (Application Recovery Kit) により、代表的なアプリケーションはユーザー様側での作り込みを必要とせずに、短期間でクラスター環境の構築が可能
- 分かり易く使い易い GUI を提供し、簡単で直観的な HA クラスター構築および管理を実現
- PowerGres Plus は透過的データ暗号化 (TDE) とトランザクションログの二重化で PostgreSQL にはないセキュリティと信頼性を実現
- 万一、システムが稼働中の AZ (Availability Zone) が停止した場合、他の AZ 上の待機系へ自動的にフェイルオーバーすることによりサービスの継続が可能

3 システム構成

3.1 構成概要

クラウド環境では、仮想環境で多くのレイヤーが抽象化されているため、ARP を使用してアクティブノードを特定することはできません。特定のクラウド環境で使用されているネットワークインフラに基づいた代替方法が必要になる場合があります。

稼働系インスタンスへのルーティングをどのように制御するかにより、おもに 2 つのシナリオがあります。ルートテーブルシナリオは、ルートテーブルの仮想 IP を稼働系インスタンスから待機系インスタンスにつけ替えてルーティングを制御します。Elastic IP シナリオは、Elastic IP を稼働系インスタンスから待機系インスタンスの ENI につけ替えてルーティングを制御します。



本環境の構成では、AWS Elastic IP シナリオを利用しています。

3.2 本手順書における動作環境

LifeKeeper for Linux のサポートマトリックスについては「[LifeKeeper for Linux サポートマトリックス](#)」を参照してください。

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 8.6.0 (x86_64)

3.3 AWS 上の設定

AWS でゼロからインスタンスを作成する詳細手順については、「[AWS でゼロからインスタンスを作成する](#)」を参照してください。

4 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 v9.6.2
- LifeKeeper Recovery kit for EC2
- DataKeeper
- PostgreSQL Recovery Kit v9.6
- PowerGres Plus Linux V13 Update 6

4.1 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「[LifeKeeper 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 (LLKL_V962_092122.iso) が /media 配下に在るもの
とします。

```
# mkdir /mnt/ISO
# mount -t iso9660 -o loop /media/LKL_V962_092122.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_962.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。また、実行する前に SELinux を無効にする必要があります。

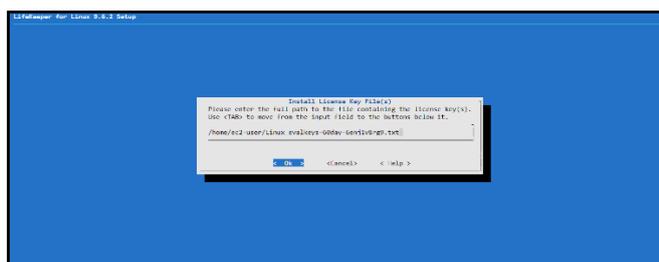
```
# ./setup
```

6. インストールのメイン画面で設定を行います。

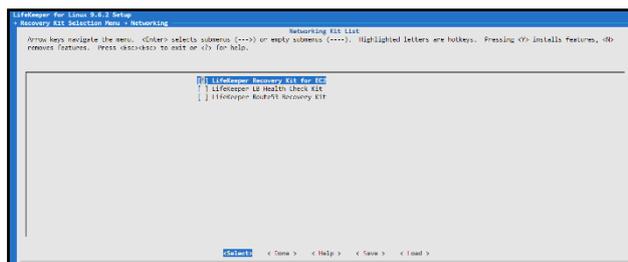
(ア) LifeKeeper Authentication を設定します。本手順書はデフォルトのまま進めます。Done で戻ります。



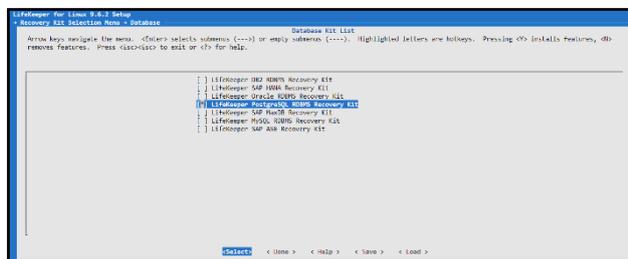
(イ) Install License Key File(s)を指定します。ファイル名を入力できるダイアログに切り替わるので、フルパス名を入力します。スペース区切りで複数のライセンスファイルを指定することが出来ます。本手順書では、以下のように /home/ec2-user/Linux_evalkeys-60day-6enjlv8rg9.txt というライセンスキーファイルを指定します。Done で戻ります。



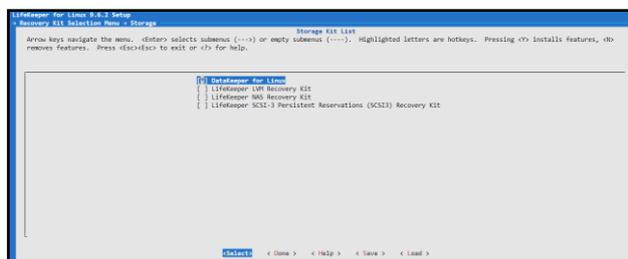
(ウ) Recovery Kit Selection Menu -> Networking -> LifeKeeper Recovery kit for EC2 をインストールします。Done で戻ります。



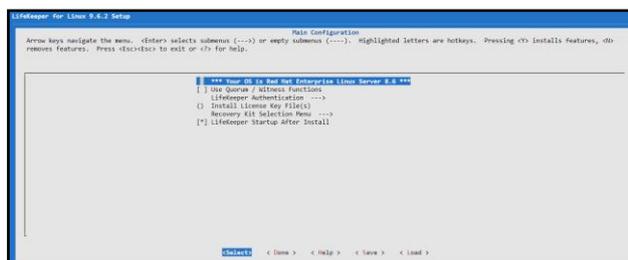
(エ) Recovery Kit Selection Menu → Database → LifeKeeper PostgreSQL RDBMS のリカバリーキットをインストールします。Done で戻ります。



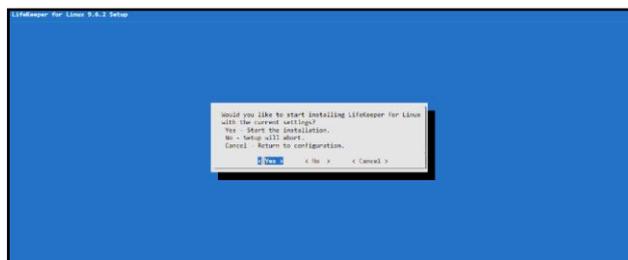
(オ) Recovery Kit Selection Menu → Storage → DataKeeper のリカバリーキットをインストールします。Done で戻ります。



(カ) LifeKeeper Startup After Install を選択します。



(キ) インストールの設定が完了したら、Yes で進めます。正常に完了した場合は、“Setup complete” が出力されます。



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 ~/.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
```

4.2 PowerGres Plus V13 のインストール

PowerGres Plus 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 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。

2. ダウンロードした製品を展開します。

```
# tar zxvf /media/powergresplus-13update6-linux.tar.gz
```

3. 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 powergresplus-13update6-linux
```

(イ) インストールスクリプトを実行

```
# bash 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-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.
```

(コ) 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 Plus (Linux) 13 installation completed.
```

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。

(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。

```
# for PowerGres Plus V13
LD_LIBRARY_PATH=/opt/powergresplus13/lib:$LD_LIBRARY_PATH
PATH=/opt/powergresplus13/bin:$PATH
MANPATH=/opt/powergresplus13/share/man:$MANPATH
export LD_LIBRARY_PATH PATH MANPATH
```

5 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「[LifeKeeper 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。ただし、クラウド環境では、基盤となるネットワークレイヤーが抽象化されているため、ノード間のコミュニケーションパスは 1 つで十分な場合があります。

本環境ではコミュニケーションパスを 1 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。また、事前に両ノードの名前を node-a と node-b に変更しています。

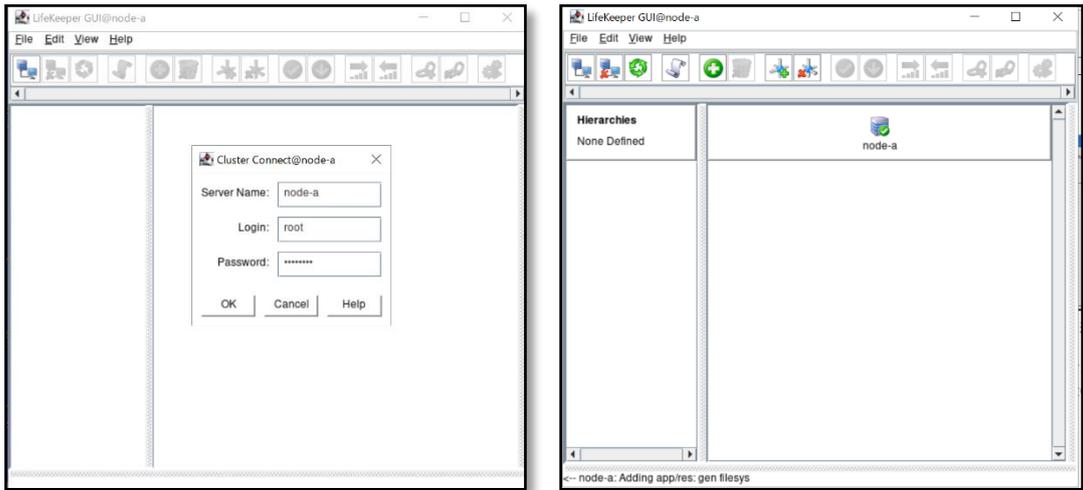
フィールド	値
Local Server	node-a
Remote Server(s)	node-b
Device Type	TCP
Local IP Address(es)	172.31.26.89
Remote IP Address	172.31.30.175
Priority	1
Local TTY Device	-
Remote TTY Device	-
Baud Rate	-

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。

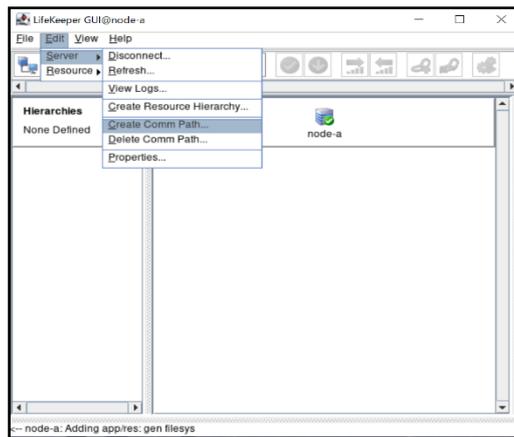
```
# lkGUIapp &
```

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。

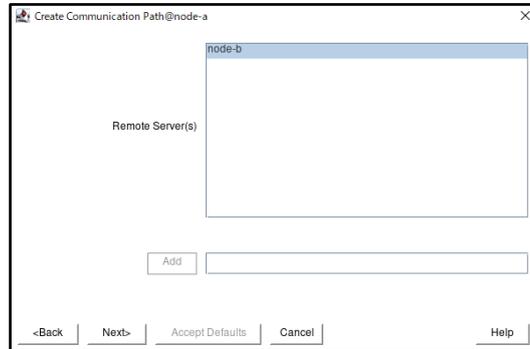
3. Edit メニューの Server > Create Comm Path... を選択します。



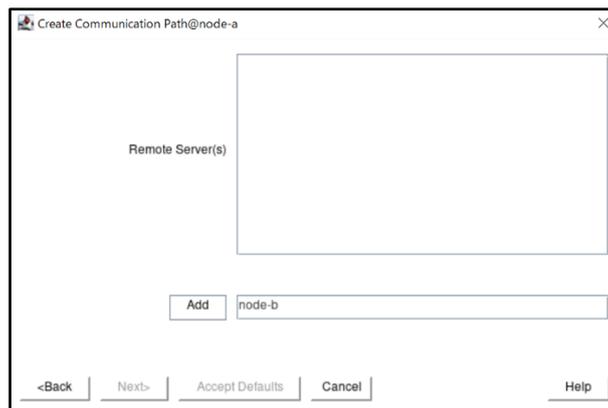
4. Local Server から稼働系サーバを選択し、Next>ボタンを押します。



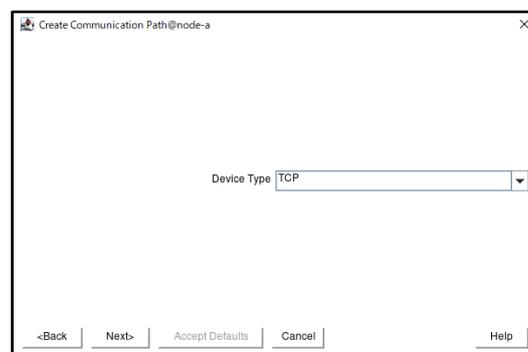
5. Remote Server(s)から待機系サーバを選択し、Next>ボタンを押します。



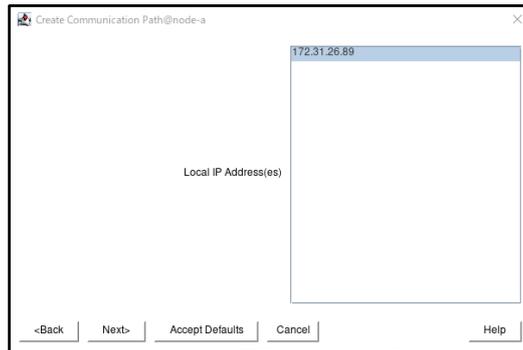
Remote Server(s)に待機系サーバが表示されていない場合には、Add ボタンの右に待機系サーバのホスト名を入力し、Add ボタンを押します。そうすると、Remote Server(s)に待機系サーバが追加されます。



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 を選択し、Next>ボタンを押します。

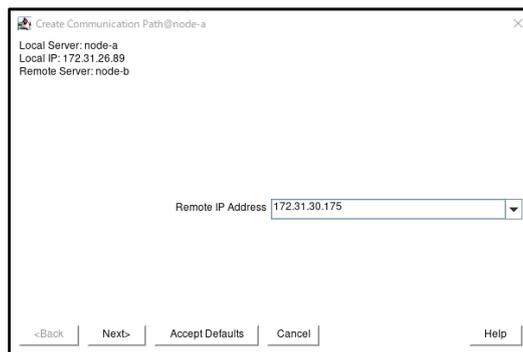


7. Local IP Address(es)から稼働系サーバの IP アドレスを選択します。

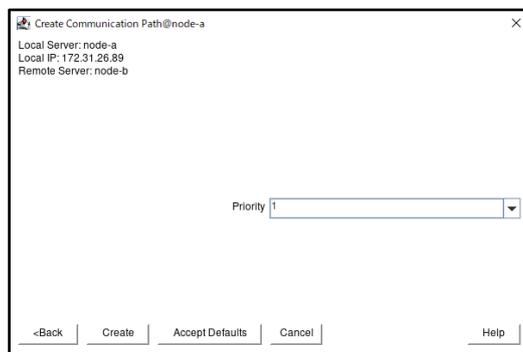


これ以降の設定を確認せずにデフォルトの設定のままで問題ない場合には `Accept Defaults` ボタンを押します。そうでない場合には `Next>` ボタンを押します。ここでは `Next>` ボタンを押した場合について説明します。

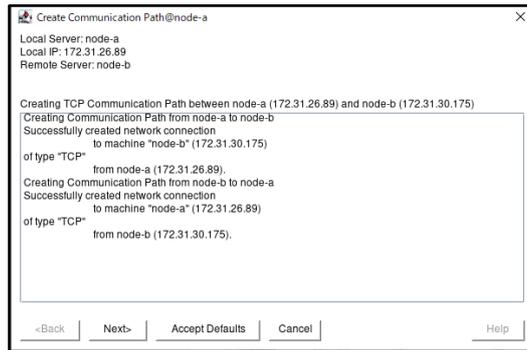
8. `Remote IP Address` から待機系サーバの IP アドレスを選択し、`Next>` ボタンを押します。



9. `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。

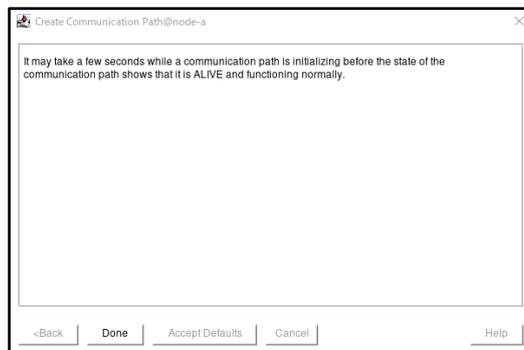


10. コミュニケーションパスが作成されます。

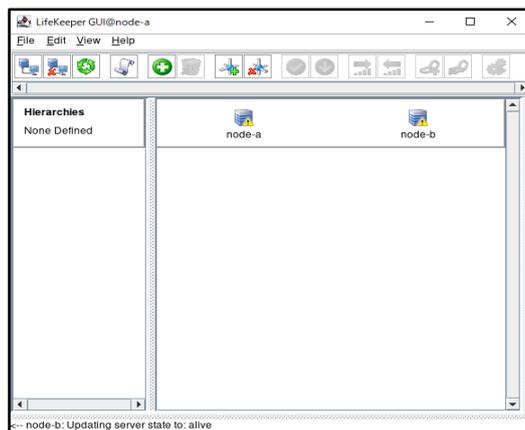


コミュニケーションパスの作成が完了したら、Next>ボタンを押します。

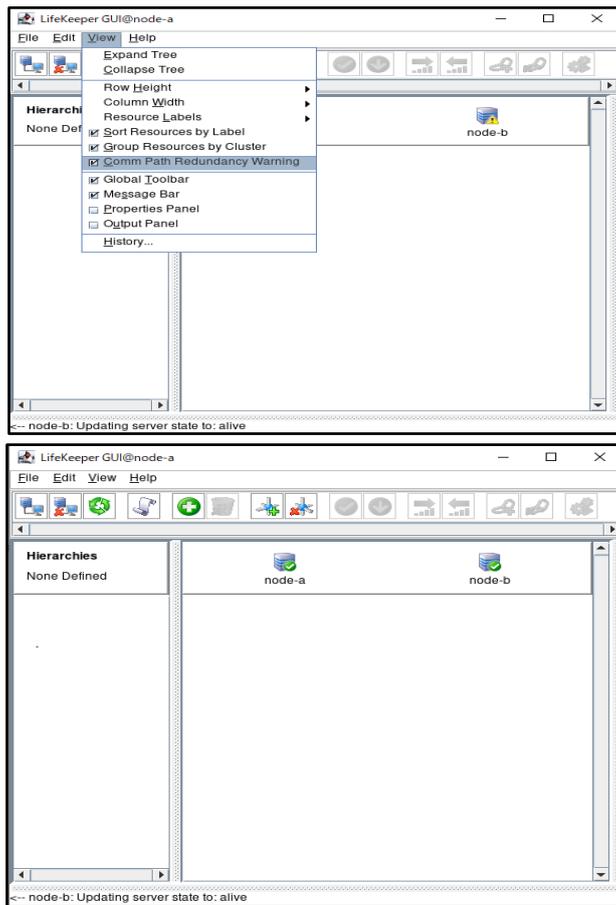
11. Done ボタンを押します。



12. 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。



警告サインは、LifeKeeper GUI の[View] メニューの [Comm Path Redundancy Warning] のチェックボックスのチェックを外すことでオフにできます。



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

6 DataKeeper リソースの作成

DataKeeper リソースを作成します。詳細については「[LifeKeeper for Linux テクニカルドキュメンテーション](#)」の「[DataKeeper for Linux 評価ガイド](#)」を参照してください。

本環境では、LifeKeeperGUI を使用した、DataKeeper リソースを作成します。

プライマリノード側のリソース設定は以下のとおりです。また、事前に両サーバのディスクにパーティションか LVM を作成する必要があります。

フィールド	値	設定内容
Select Recovery Kit	Data Replication	冗長化対象のアプリケーション
Switchback Type	intelligent	自動フェイルバックの有無
Server	node-a	稼働系ノード名
Hierarchy Type	Replicate New Filesystem	レプリケーションタイプ
Source Disk	/dev/mapper/vgroup1-lv1	同期元のディスク
New Mount Point	/data	マウントポイントの指定
New Filesystem Type	ext4	ファイルシステムの選択
Data Replication Resource Tag	datarep-data	LifeKeeper 管理上のタグ名
File System Resource Tag	/data	LifeKeeper 管理上のタグ名
Bitmap File	/opt/Lifekeeper/bitmap_data	差分情報の保存場所
Enable Asynchronous Replication	no	同期モードの選択

バックアップノードへリソースを拡張するための Pre-Extend ウィザードの設定は以下のとおりです。

フィールド	値	設定内容
Target Server	node-b	待機系ノード名
Switchback Type	intelligent	自動フェイルバックの有無
Template Priority	1	稼働系サーバのプライオリティ
Target Priority	10	待機系サーバのプライオリティ

バックアップノード側のリソース設定は以下のとおりです。

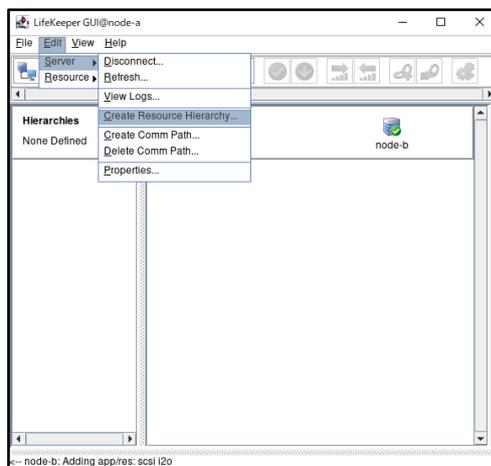
フィールド	値	設定内容
Mount Point	/data	マウントポイントの指定
Root Tag	/data	LifeKeeper 管理上のタグ名
Target Disk	/dev/mapper/vgroup1-lv1	同期先のディスク
Data Replication Resoure Tag	datarep-data	LifeKeeper 管理上のタグ名
Bitmap File	/opt/Lifekeeper/bitmap_data	差分情報の保存場所
Replication Path	172.31.26.89/172.31.30.175	同期に使用するネットワーク

DataKeeper リソー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。

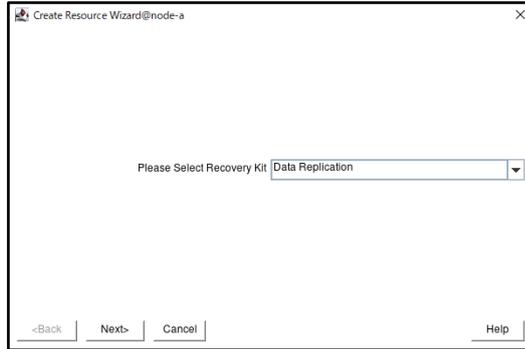
1. LifeKeeper GUI を起動し、接続します。

```
# lkGUIapp &
```

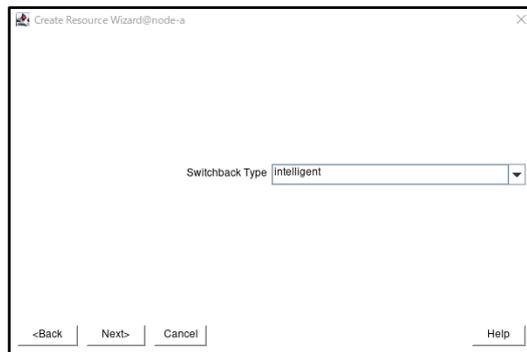
2. Edit メニューの Server > Create Resource Hierarchy... を選択します。



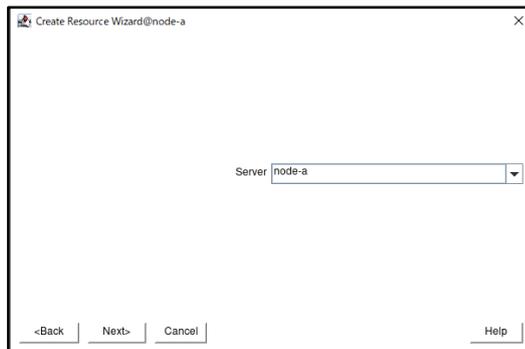
3. Please Select Recovery Kit から Recovery Kit として Data Replication を選択し、Next>ボタンを押します。



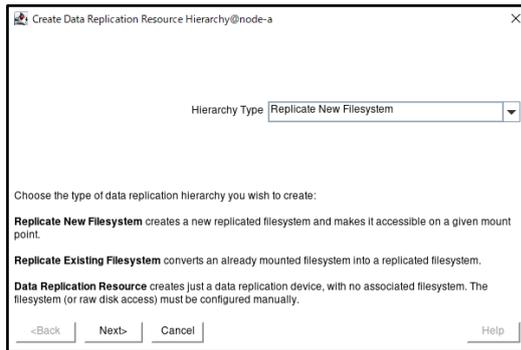
4. Switchback Type からスイッチバックの動作として intelligent または automatic を選択し、Next>ボタンを押します。



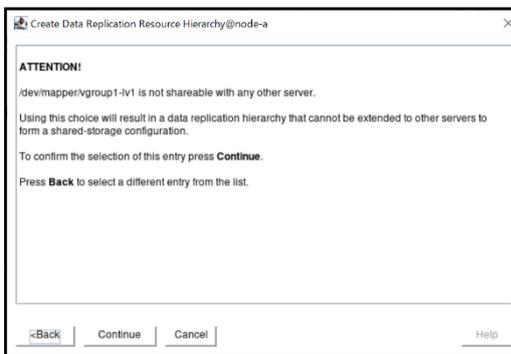
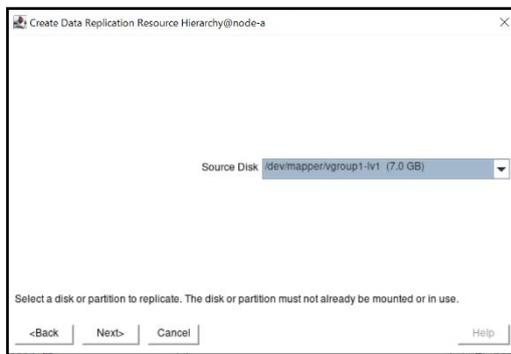
5. Server からリソース階層の作成先のサーバとして稼働系サーバを選択し、Next>ボタンを押します。



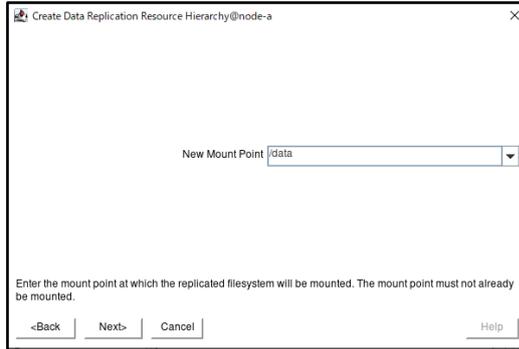
6. Hierarchy Type からレプリケーションタイプとして、Replicate New Filesystem を選択し、Next>ボタンを押します。



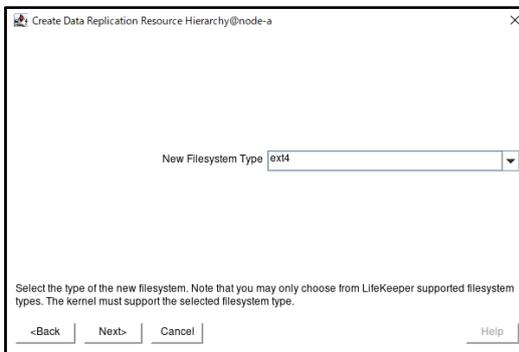
7. Source Disk から同期のディスクとして、対象ディスクを選択し、Next>ボタンを押します。この後 Attention メッセージが表示されますが、Continue ボタンを押します。



8. New Mount Point からマウントポイントとして、パスを指定し、Next>ボタンを押します。



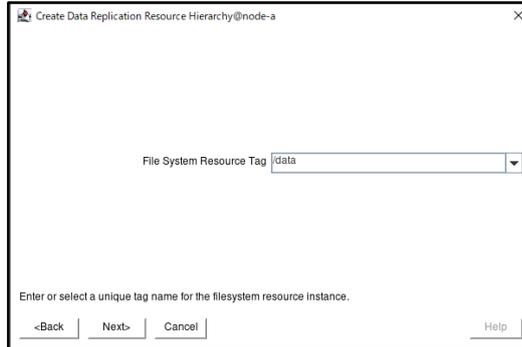
9. New Filesystem Type からファイルシステムタイプとして、ext4 を指定し、Next> ボタンを押します。



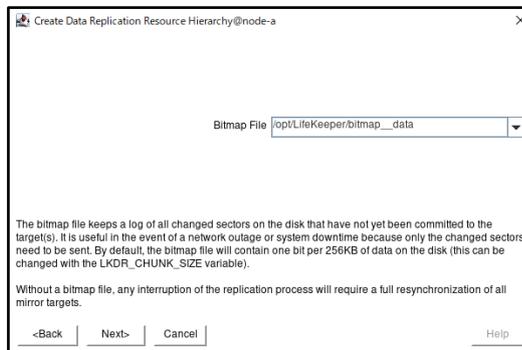
10. Data Replication Resource Tag から LifeKeeper 管理上のタグ名として、タグを指定し、Next>ボタンを押します。



11. File System Resource Tag から LifeKeeper 管理上のタグ名として、タグを指定し、Next>ボタンを押します。

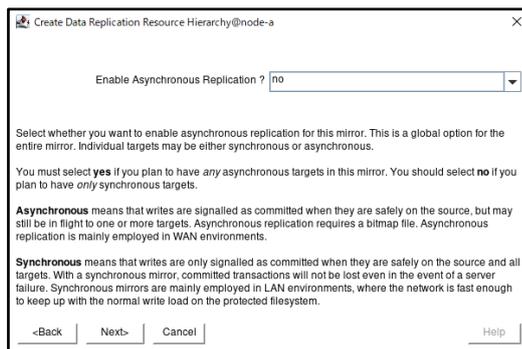


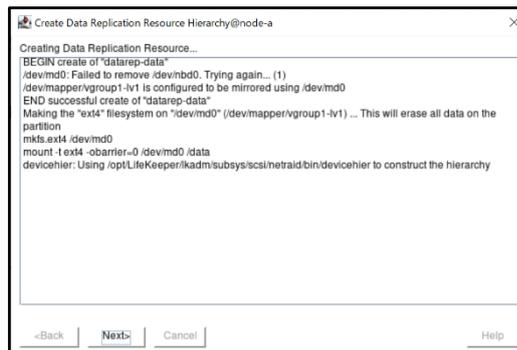
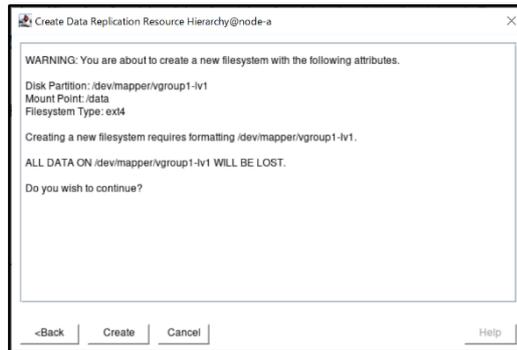
12. Bitmap File から差分情報の保存場所として、パスを指定し、Next>ボタンを押します。



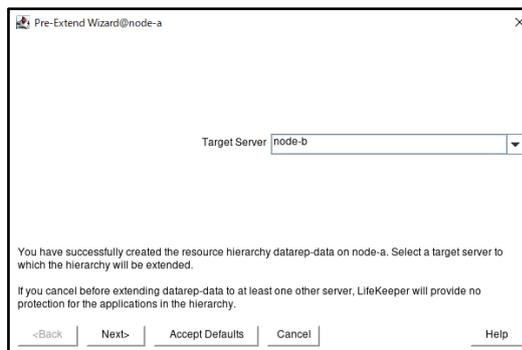
13. Enable Asynchronous Replication から同期モードを選択し、Next>ボタンを押します。この後 Warning メッセージが出力されますが、Create ボタンを押します。作成完了されたら、Next>ボタンを押します。

※ RHEL 8.6 カーネルの問題によりデータレプリケーション構成の非同期モードはサポートされません。

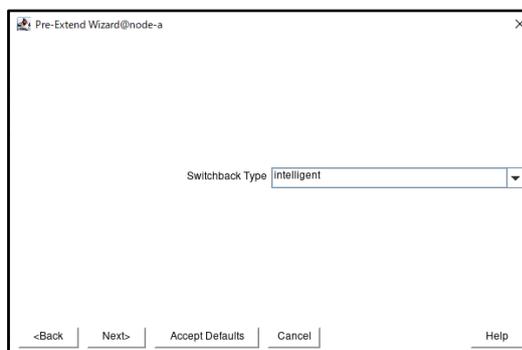




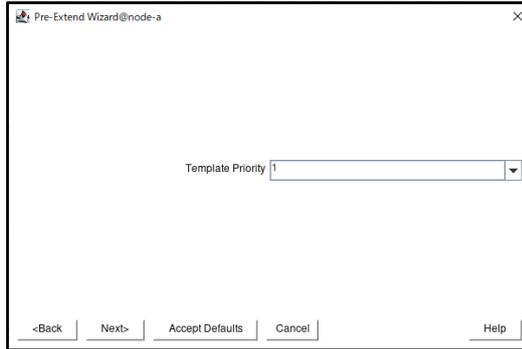
14. Target Server から待機系サーバを選択し、Next>ボタンを押します。



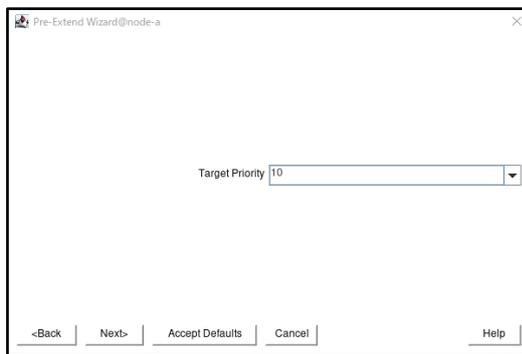
15. Switchback Type からスイッチバックの動作として intelligent または automatic を選択し、Next>ボタンを押します。



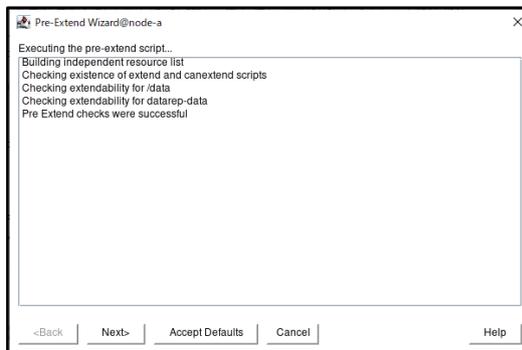
16. Template Priority から 1 を選択し、Next>ボタンを押します。



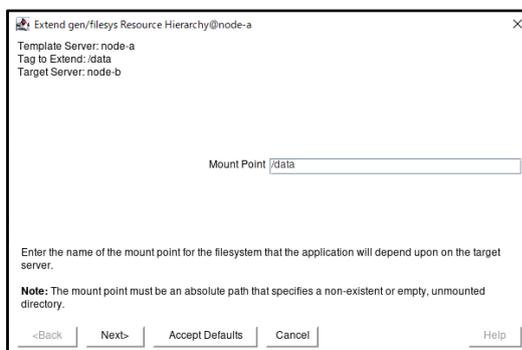
17. Target Priority から 10 を選択し、Next>ボタンを押します。



18. Pre-Extend のスクリプトが実行完了されたら、Next>ボタンを押します。

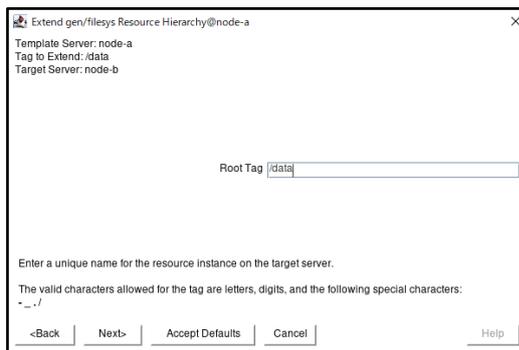


19. Mount Point からマウントポイントとしてパスを指定し、Next>ボタンを押します。



20. Root Tag から LifeKeeper 管理上のタグ名としてタグを指定し、Next>ボタンを押

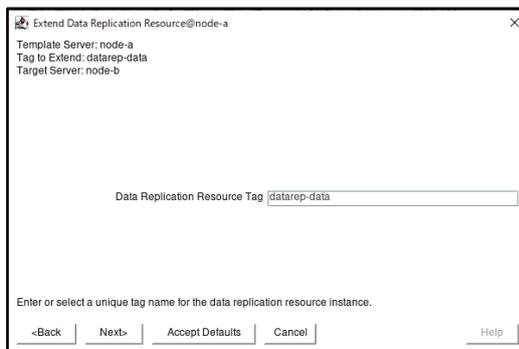
します。



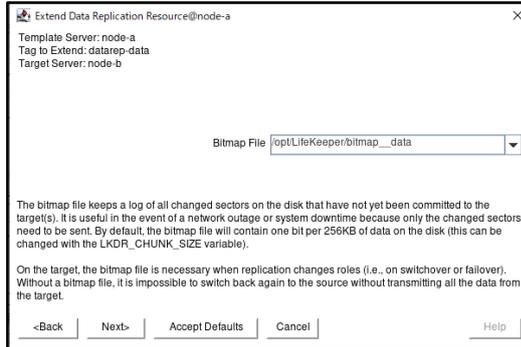
21. Target Disk から同期先のディスクを指定し、Next>ボタンを押します。



22. Data Replication Resource Tag から LifeKeeper 管理上のタグ名として、タグを指定し、Next>ボタンを押します。



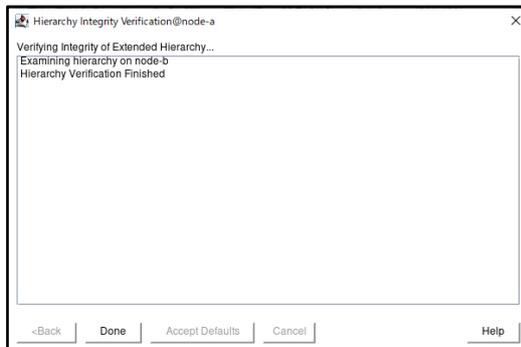
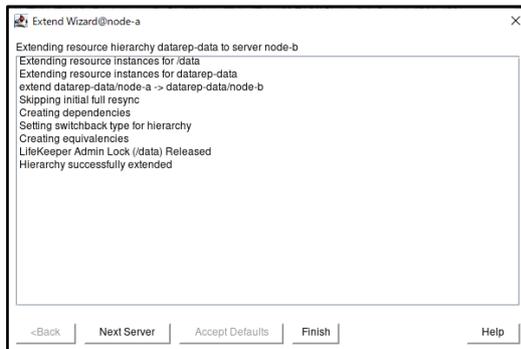
23. Bitmap File から差分情報の保存場所として、パスを指定し、Next>ボタンを押します。



24. Replication Path から同期に使用するネットワークとして、IPアドレスを選択し、Next>ボタンを押します。

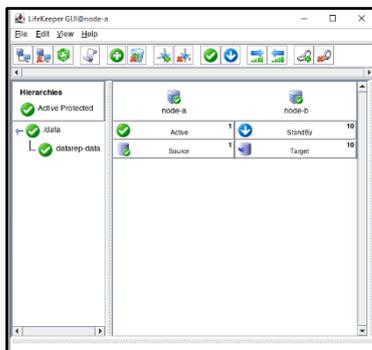


25. Hierarchy successfully extended が出力されたら、Finish ボタンを押します。Hierarchy Verification Finished が出力されたら、Done ボタンを押します。



26. リソース階層ツリーに DataKeeper 階層が追加されます。

DataKeeper 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。



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

7 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、データ暗号化の設定を行います。

7.1 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 Administration Tool マニュアル](#)」の「[サーバ管理](#)」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 DataKeeper 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 -p /data/pgsql/data
# chmod 0700 /data/pgsql/data
# chown postgres:postgres /data/pgsql/data
```

2. 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。
3. 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。



4. 「新しいサーバ」ボタンをクリックします。



- 「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。

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	/data/pgsql/data
バックアップディレクトリ (オプション)	(記入なし)
プロフィール名	PowerGres Plus (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>
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

パスワード

パスワードの確認

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

性能チューニングを行うか

合計メモリ
801MB

接続数
100

データベースタイプ

- ウェブアプリケーション
- OLTP システム
- データウェアハウス

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 / サーバ

サーバ ①

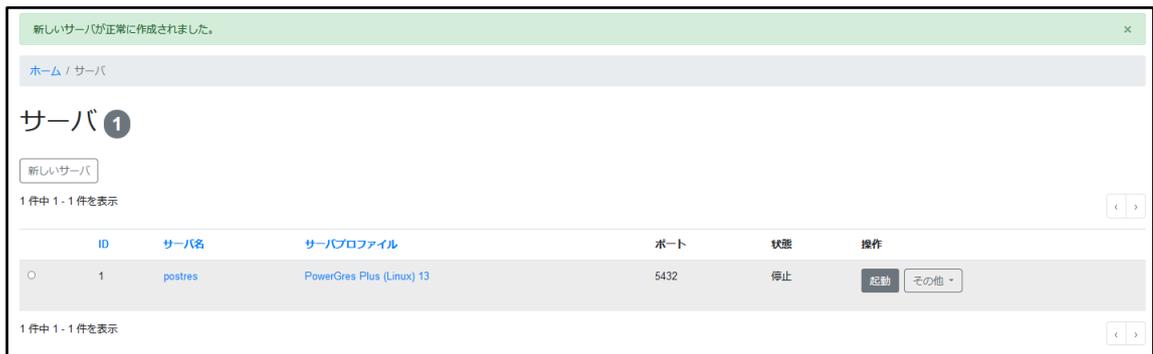
新しいサーバ

1件中1 - 1件を表示

ID	サーバ名	サーバプロファイル	ポート	状態	操作
1	postgres	PowerGres Plus (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。



サーバが正常に起動したメッセージが表示されます。



7.2 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 Plus (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 Plus (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 Plus (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
```

- エディタで .pgpass を以下の書式で編集を行います。

```
《hostname》 : 《port》 : 《database》 : 《username》 : 《password》
```

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 password とします。

```
localhost:5432:template1:dbmonitor:password
```

- ファイルのアクセス権限を設定します。

```
$ 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>

7.3 待機系サーバの設定

ここでは、待機系サーバの PowerGres Administration Tool の設定を行います。

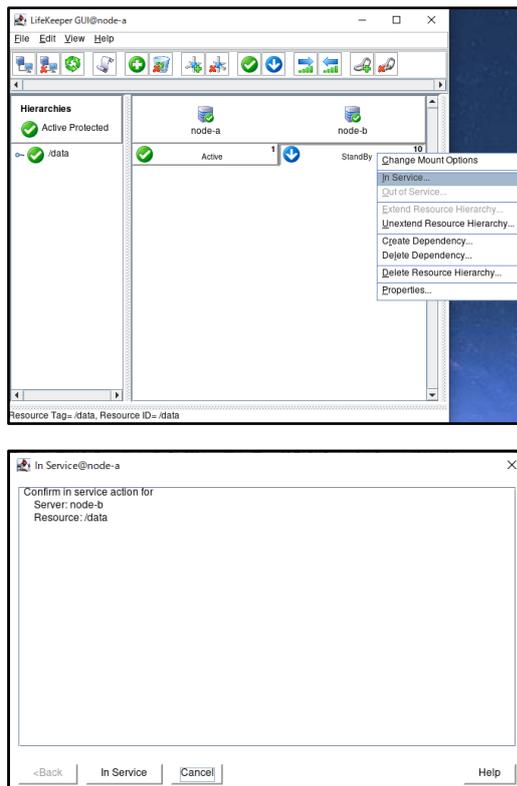
- LifeKeeper GUI を起動し、接続します。

lkGUIapp &

- 稼働系側のデータベースを停止する必要があります。「操作」の 停止ボタンを押します。



- 待機系側の DataKeeper リソースにて右クリック後、In Service を押します。次の確認画面に引き続き In Service ボタンを押します。最後に Done ボタンを押します。



```

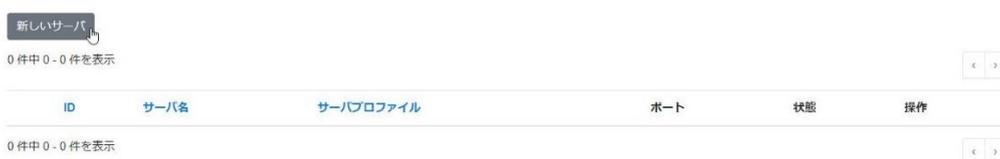
In Service@node-a
Bringing /data in service on node-b
BEGIN restore of "datarep-data"
/dev/vmd0: Failed to remove /dev/nbd0. Trying again... (1)
/dev/vxd01 is configured to be mirrored using /dev/vmd0
Resource "/data" is "OSU". The mirror "datarep-data" will wait to replicate data until all resources in the
hierarchy of type "filesys" are in-service. To replicate data immediately run:
"opt.LifeKeeper/bin/mirror_action datarep-data resume" on "node-b" (see
"LKOR_WAIT_TO_RESYNC" in /etc/default/LifeKeeper).
END successful restore of "datarep-data"
BEGIN restore of /data
"fsck"ing file system /data
fsck.ext4 -y /dev/vmd0
e2fsck 1.45.6 (20-Mar-2020)
/dev/vmd0: clean, 984/524288 files, 68951/2096128 blocks
mounting file system /data
mount -t ext4 -o rw,relatime,nobarrier /dev/vmd0 /data
File system /data has been successfully mounted.
END successful restore of /data
Put "/data" in-service successful

```

4. 待機系サーバで PowerGres Administration Tool を起動してデータベースクラスタを登録します。

5. 「新しいサーバ」ボタンをクリックします。

サーバ 0



6. 「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。

7. 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。

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	/data/pgsql/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 Plus (Linux) 13
作成方法	既存のデータベースクラスタを作成

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。

ID	サーバ名	サーバプロファイル	ポート	状態	操作
1	postgres	PowerGres Plus (Linux) 13	5432	停止	起動 その他

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。サーバが正常に起動したメッセージが表示されます。

10. 待機系サーバでも同様に 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
```

11. 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:template1:dbmonitor:password
```

12. 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。

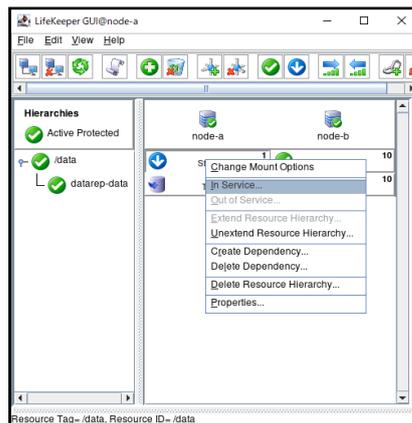
7.4 データ暗号化の設定

透過的データ暗号化は PowerGres Plus のみで使用できます。透過的データ暗号化の詳細については「[PowerGres Plus \(Linux 版\) マニュアル](#)」の「[透過的データ暗号化](#)」を参照してください。

1. 待機系サーバで PowerGres Administration Tool をログインします。「操作」の停止ボタンを押します。



2. 稼働系側の DataKeeper リソースにて右クリック後、In Service を押します。次の確認画面に引き続き In Service ボタンを押します。最後に Done ボタンを押します。



```

In Service@node-a
Bringing /data in service on node-a
BEGIN restore of "datarep-data"
/dev/md0: Failed to remove /dev/nbd0. Trying again... (1)
/dev/md0 is configured to be mirrored using /dev/md0
Resource "/data" is "OSU". The mirror "datarep-data" will wait to replicate data until all resources in the
hierarchy of type "filecopy" are in-service. To replicate data immediately run:
"opt/LifeKeeper/bin/mirror_action datarep-data resume" on "node-a" (see
"LKDR_WAIT_TO_RESYNC" in /etc/default/LifeKeeper).
END successful restore of "datarep-data"
BEGIN restore of /data
"fsck" file system /data
fsck.ext4 -y /dev/md0
e2fsck 1.45.6 (20-Mar-2020)
/dev/md0: clean, 982/458752 files, 63645/1835008 blocks
mounting file system /data
mount -tx4 -onv,relatime,nobarrier /dev/md0 /data
File system /data has been successfully mounted.
END successful restore of /data
Put "/data" in-service successful

```

- 稼働系サーバで PowerGres Administration Tool をログインします。「操作」の 起動ボタンを押します。

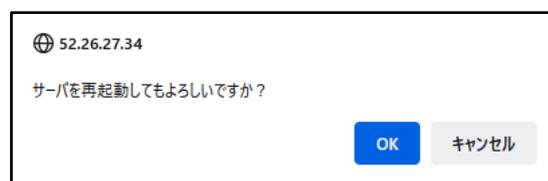


- 「透過的データ暗号化」画面にアクセスするには、メニューの「透過的データ暗号化」をクリックします。



- マスターキーを作成するのに必要な情報を入力し、「作成」ボタンをクリックします。

6. 「作成」ボタンをクリックすると、サーバを再起動するかを確認するダイアログが表示されます。「OK」ボタンをクリックします。



7. マスターキー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ます。



8. マスターキーの作成直後にキーストアはオープンされます。
9. 自動オープンキーストアを有効にするため、「自動オープンキーストアを有効化」ボタンをクリックします。パスワードを入力し、「有効化」ボタンをクリックします。

10. 自動オープンキーストアの有効化が完了すると、完了したことを表すメッセージが表示されます。

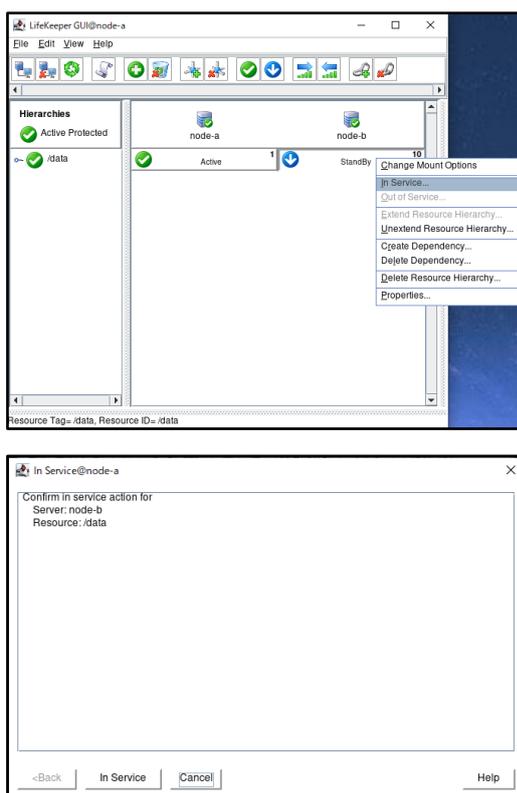


11. フェイルオーバー後に待機系サーバでもキーストアをオープンできるように、キーストア格納ディレクトリ (/var/lib/pgsql/keystore) を待機系サーバにコピーします。

13. 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。
ここでは稼働系データベースサーバを正常に終了してから実行する必要があります。
下記の画面で「操作」の 停止ボタンを押します。



12. 待機系側の DataKeeper リソースにて右クリック後、In Service を押します。次の確認画面に引き続き In Service ボタンを押します。最後に Done ボタンを押します。



```

In Service@node-a
Bringing /data in service on node-b
BEGIN restore of "datarep-data"
/dev/mdb1: Failed to remove /dev/mdb0. Trying again... (1)
/dev/xvdb1 is configured to be mirrored using /dev/mdb0
Resource "/data" is "OSU". The mirror "datarep-data" will wait to replicate data until all resources in the
hierarchy of type "filesys" are in-service. To replicate data immediately run:
"topcli.keeper/bin/mirror_action datarep-data resume" on "node-b" (see
"LKDR_WAIT_TO_RESYNC" in /etc/default/LikeKeeper).
END successful restore of "datarep-data"
BEGIN restore of /data
"fsck"ing file system /data
fsck.ex14 -y /dev/mdb0
e2fsck 1.45.5 (20-Mar-2020)
/dev/mdb0: clean, 984524288 files, 689512096128 blocks
mounting file system /data
mount -text4 -orw,relatime,nobarrier /dev/mdb0 /data
File system /data has been successfully mounted.
END successful restore of /data
Put "/data" in-service successful

```

13. 自動オープンキーストアはそれぞれのサーバで作成しなければならないので、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する必要があります。

```
$ rm /var/lib/pgsql/keystore/keystore.aks
```

14. 待機系サーバで PowerGres Administration Tool をログインします。「操作」の起動ボタンを押します。



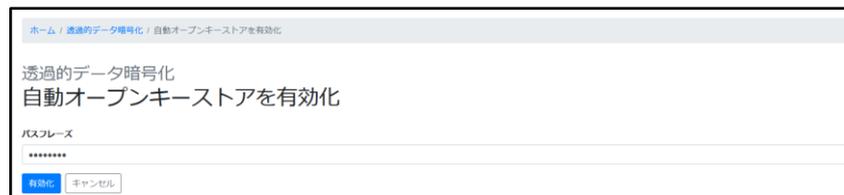
15. 「透過的データ暗号化」画面にアクセスするには、メニューの「透過的データ暗号化」をクリックします。



16. パスフレーズを入力し、「オープン」ボタンをクリックします。キーストアが正常にオープンされたことを表すメッセージが表示されます。



17. 自動オープンキーストアを有効にするため、「自動オープンキーストアを有効化」ボタンをクリックします。パスフレーズを入力し、「有効化」ボタンをクリックします。



18. 自動オープンキーストアの有効化が完了すると、完了したことを表すメッセージが表示されます。



8 PowerGres Plus リソース階層の作成

PowerGres Plus 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 Plus (Linux 版) 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「[PostgreSQL Recovery Kit 管理ガイド](#)」の「[PostgreSQL リソース階層の作成](#)」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。

リソース階層の作成	
Please Select Recovery Kit	PostgreSQL Database
Switchback Type	intelligent
Server	node-a
PostgreSQL Executable Location	/opt/powergresplus13/bin
PostgreSQL Client Executable Location	/opt/powergresplus13/bin/psql
PostgreSQL Administration Executable Location	/opt/powergresplus13/bin/pg_ctl
PostgreSQL Data Directory	/data/pgsql/data
PostgreSQL Port	5432
PostgreSQL Socket Path	/var/run/postgresql/.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	node-b
Switchback Type	intelligent
Template Priority	1
Target Priority	10
PostgreSQL Executable Location	/opt/powergresplus13/bin
PostgreSQL Database Tag	pgsql-5432

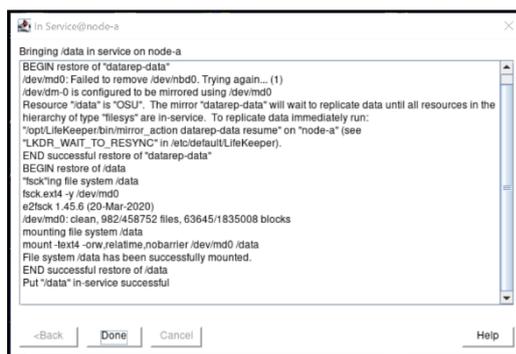
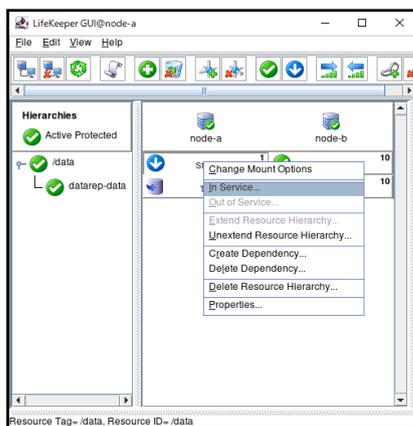
- 稼働系サーバから LifeKeeper GUI を起動し、接続します。

```
# lkGUIapp &
```

- 待機系サーバで PowerGres Administration Tool をログインします。「操作」の停止ボタンを押します。



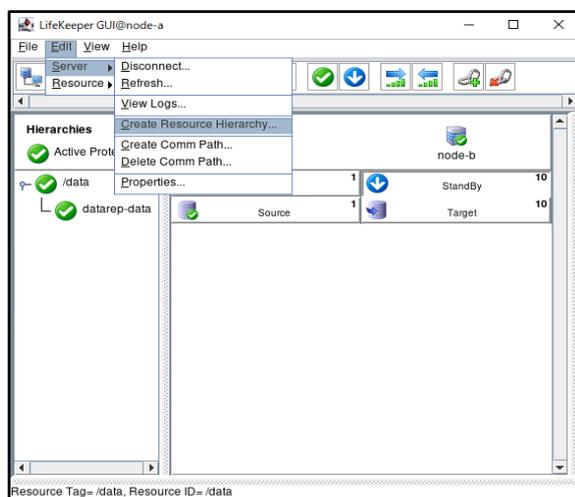
- 稼働系側の DataKeeper リソースにて右クリック後、In Service を押します。次の確認画面に引き続き In Service ボタンを押します。最後に Done ボタンを押します。



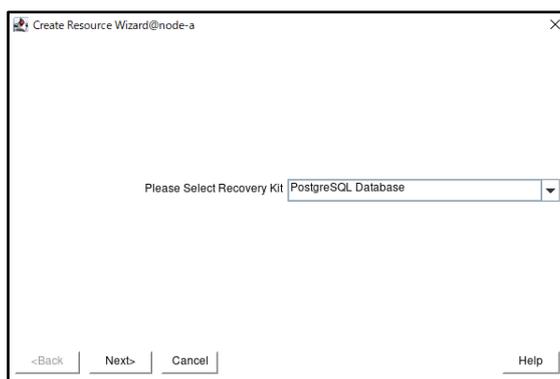
- 稼働系サーバで PowerGres Administration Tool をログインします。「操作」の起動ボタンを押します。



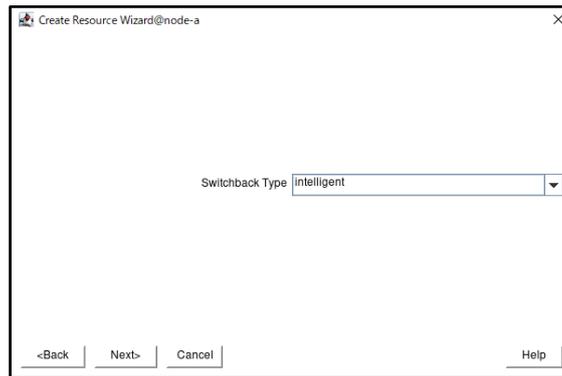
- 稼働系サーバから Edit メニューの Server > Create Resource Hierarchy... を選択します。



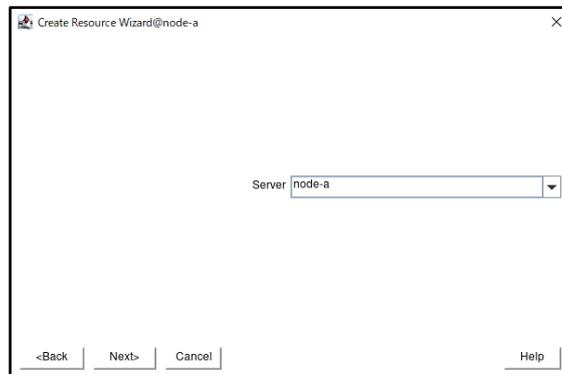
- Please Select Recovery Kit から Recovery Kit として PostgreSQL Database を選択し、Next>ボタンを押します。



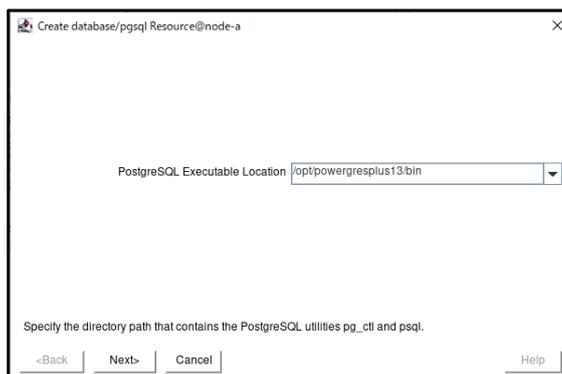
- Switchback Type からスイッチバックの動作として intelligent または automatic を選択し、Next>ボタンを押します。



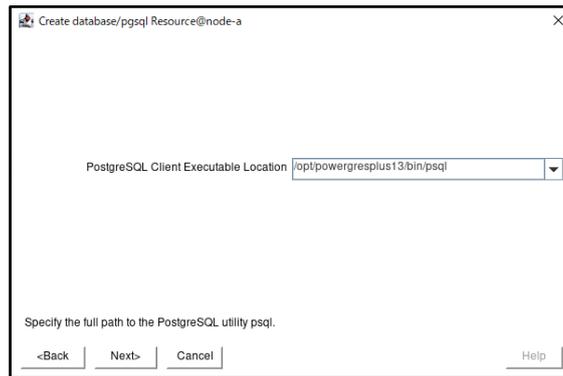
8. Server からリソース階層の作成先のサーバとして稼働系サーバを選択し、Next>ボタンを押します。



9. PostgreSQL Executable Location に PowerGres Plus 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。



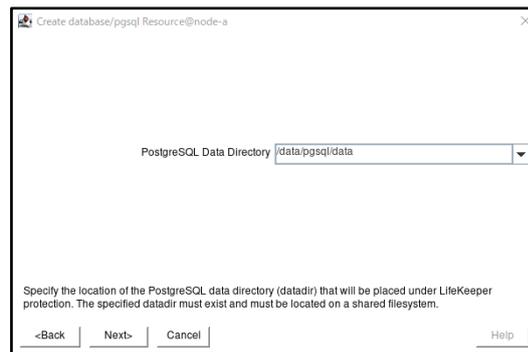
10. PostgreSQL Client Executable Location に PowerGres Plus の psql コマンドのフルパスを指定し、Next>ボタンを押します。



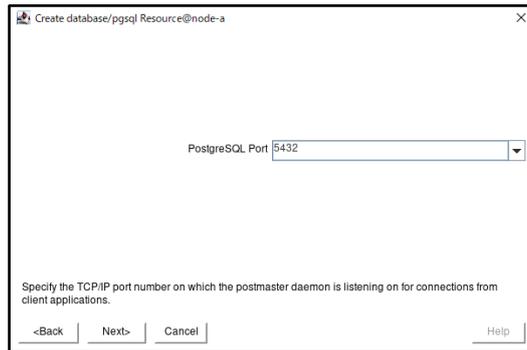
11. PostgreSQL Administration Executable Location に PowerGres Plus の pg_ctl コマンドのフルパスを指定し、Next>ボタンを押します。



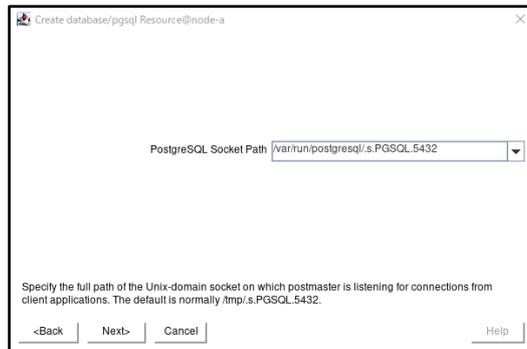
12. PostgreSQL Data Directory に LifeKeeper の保護下に置かれる PostgreSQL データディレクトリ (datadir) の場所を指定し、Next>ボタンを押します。



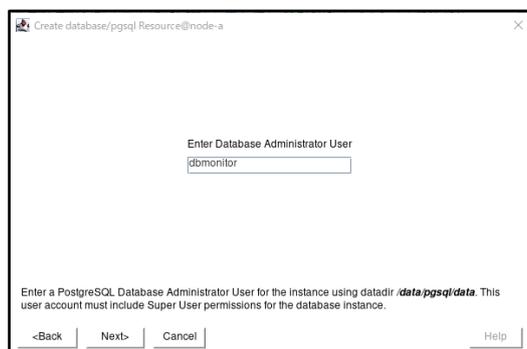
13. PostgreSQL Port に データベースサーバのポート番号を入力し、Next>ボタンを押します。



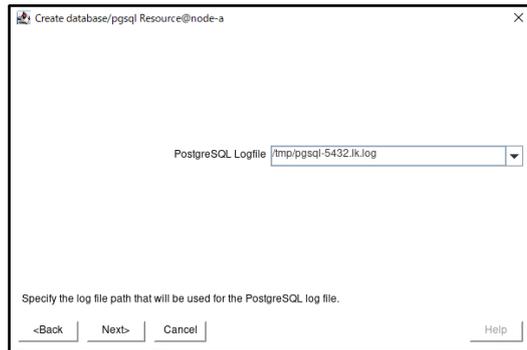
14. PostgreSQL Socket Path に ポストマスターデーモンがクライアントアプリケーションからの接続をリスニングする UNIX ドメインソケットへのフルパスを指定し、Next>ボタンを押します。



15. 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、「[7.2 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。



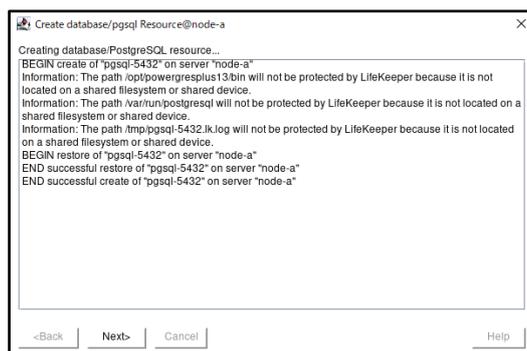
16. PostgreSQL Logfile に PostgreSQL の起動や停止に使用する pg_ctl の -l オプションで使われるログファイルパスを指定し、Next>ボタンを押します。



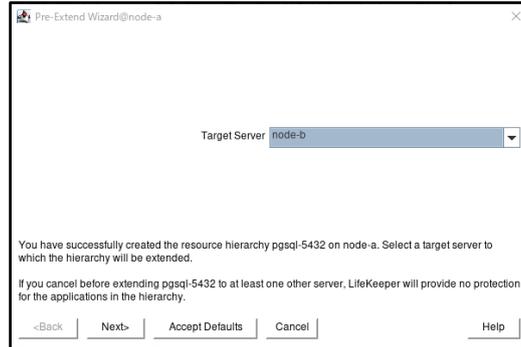
17. PostgreSQL Database Tag に PowerGres Plus リソースのタグ名を入力し、Create ボタンを押します。



18. PowerGres Plus 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。

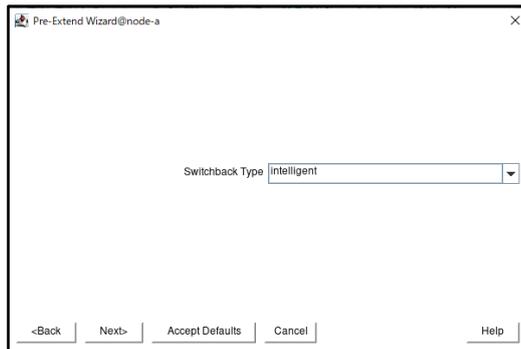


19. Pre-Extend Wizard が立ち上がります。拡張先サーバとして Target Server に待機系サーバを選択します。

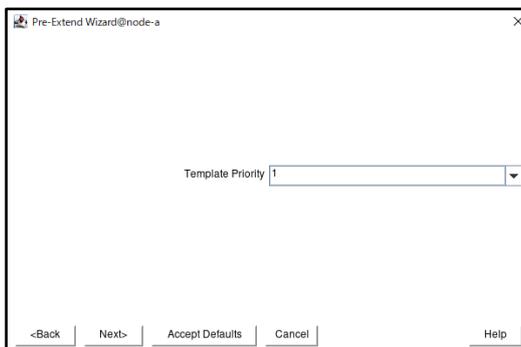


これ以降の設定を確認せずにデフォルトの設定のまま問題ない場合には **Accept Defaults** ボタンを押します。そうでない場合には **Next>**ボタンを押します。ここでは **Next>**ボタンを押した場合について説明します。

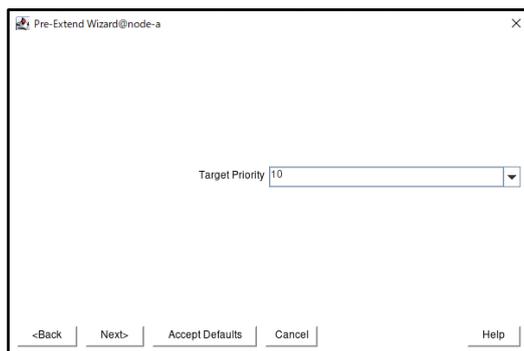
20. **Switchback Type** からスイッチバックの動作として **intelligent** または **automatic** を選択し、**Next>**ボタンを押します。



21. **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。



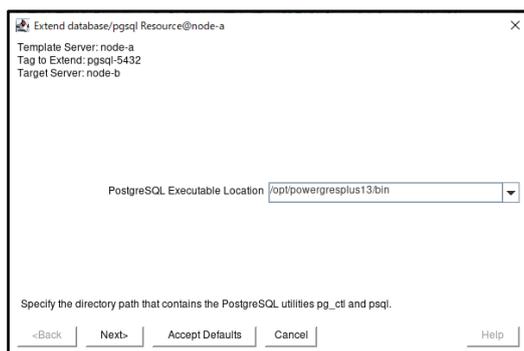
22. **Target Priority** に拡張先のリソース階層の優先順位を入力し、**Next>**ボタンを押します。



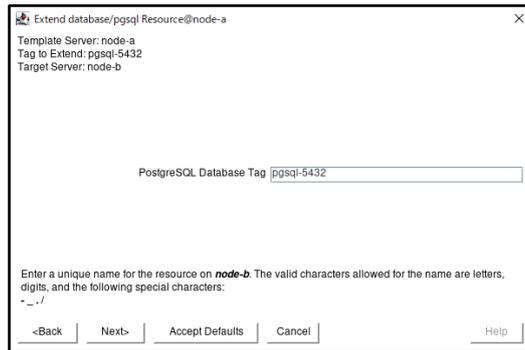
23. 環境がチェックされ、このリソースを拡張するのに必要な要件がすべて満たされていることを示す情報ボックスが表示されます。チェックが完了したら、Next>ボタンを押します。満たされていない要件がある場合、Next>ボタンが使用できなくなり、<Back ボタンが有効になります。



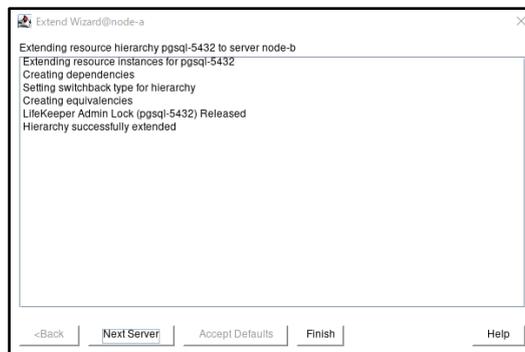
24. PostgreSQL Executable Location に PostgreSQL 実行可能ファイルを含むディレクトリパスを指定し、Next>ボタンを押します。



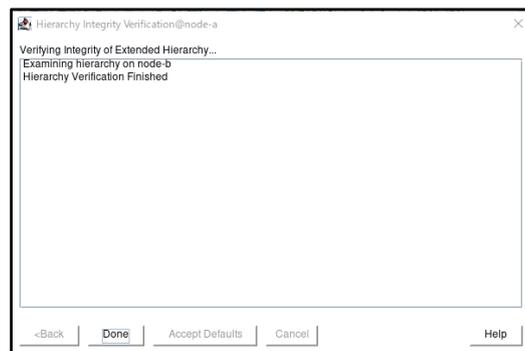
25. PostgreSQL Database Tag に PowerGres Plus リソースのタグ名を入力し、Next>ボタンを押します。



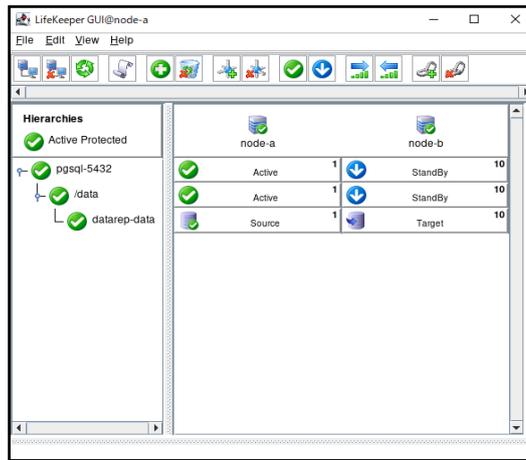
26. リソース階層が拡張されます。リソース階層の拡張が完了したら、Finish ボタンを押します。



27. [Done] 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。



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



29. 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

AWS 上で Elastic IP シナリオを利用する場合、IP リソースを作成する必要はありません。ルートテーブルシナリオを利用する場合、IP リソース階層を作成する必要があります。詳細については「[IP Recovery Kit 管理ガイド](#)」の「[IP リソース階層の作成](#)」を参照してください。

10 EC2 リソース階層の作成 (Elastic IP シナリオ)

EC2 リソース階層 (Elastic IP シナリオ) を作成します。詳細については「[クラウド環境でのノードの切り替え](#)」の「[AWS EC2 リソースの作成 \(Elastic IP シナリオ\)](#)」を参照してください。

10.1 EC2 リソース作成の事前設定

AWS CLI をインストールする必要があります。詳しい手順は、「[AWS CLI の最新バージョンをインストールまたは更新します](#)」に記載されています。CLI をインストールしたら、CLI の場所を `/etc/default/LifeKeeper` ファイルに追加します。

1. 以下のコマンドで、AWS CLI を確認します。

```
# aws --version
```

2. LifeKeeper の設定ファイル `/etc/default/LifeKeeper` の PATH に、AWS CLI 実行ファイルのパス (`/usr/local/bin`) を追加します。
`/etc/default/LifeKeeper` をテキストエディタで開き、AWS CLI 実行ファイルのパスの末尾に「`:/usr/local/bin`」を追記します。
ここまでの設定を両ノードで行います。
3. EC2 Recovery Kit を利用するには、RouteTable のエントリーの更新、ENI (Elastic Network Interface) の再割り当てを許可された役割がインスタンスに必要です。詳しい手順は、「[Recovery Kit for EC2 の要件](#)」に記載されています。

10.2 EC2 リソースの作成

EC2 リソース階層の設定は以下のとおりです。また、Elastic IP を取得する必要があります。以下の例では、Elastic IP アドレスは `52.26.27.34` です。なお、この IP アドレスは一例です。

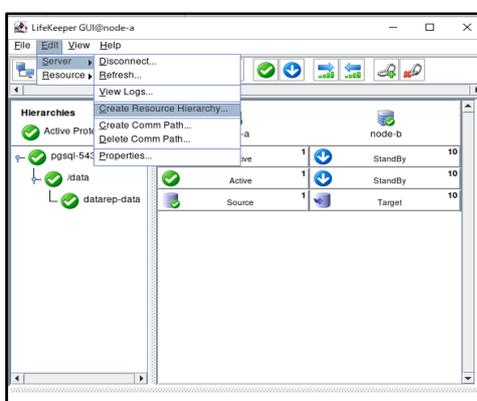
リソース階層の作成	
Please Select Recovery Kit	Amazon EC2
Switchback Type	intelligent
Server	node-a
EC2 Resource Type	Elastic IP(Frontend Cluster)
Network Interface	eth0
Elastic IP	52.26.27.34
EC2 Resource Tag	ec2-52.26.27.34

リソース階層の拡張	
Target Server	node-b
Switchback Type	intelligent
Template Priority	1
Target Priority	10

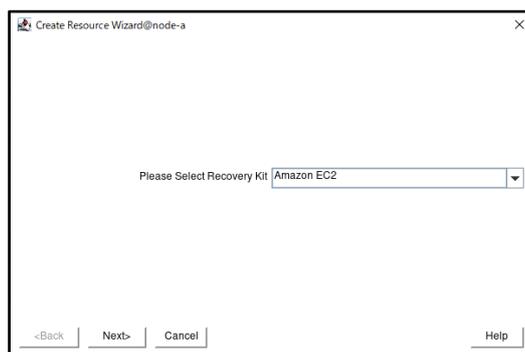
1. LifeKeeper GUI を起動し、接続します。

lkGUIapp &

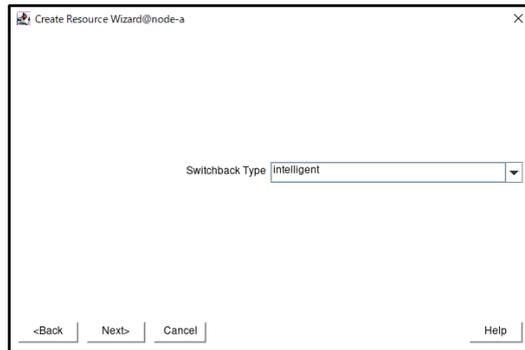
2. Edit メニューの Server > Create Resource Hierarchy... を選択します。



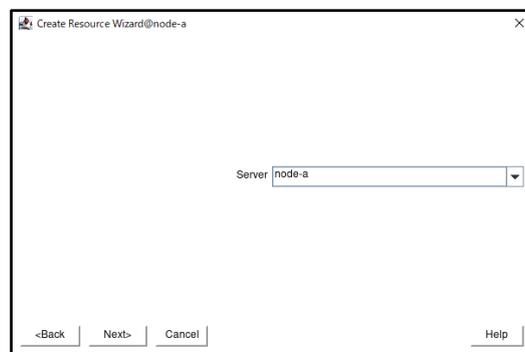
3. Please Select Recovery Kit から Recovery Kit として Amazon EC2 を選択し、Next>ボタンを押します。



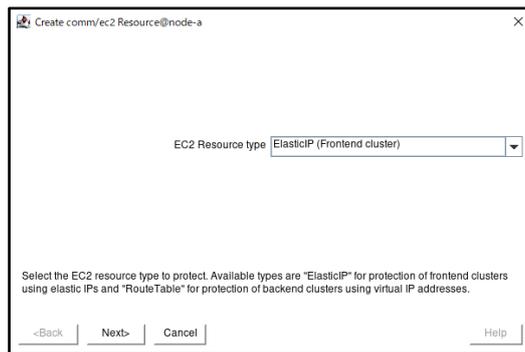
4. Switchback Type からスイッチバックの動作として intelligent または automatic を選択し、Next>ボタンを押します。



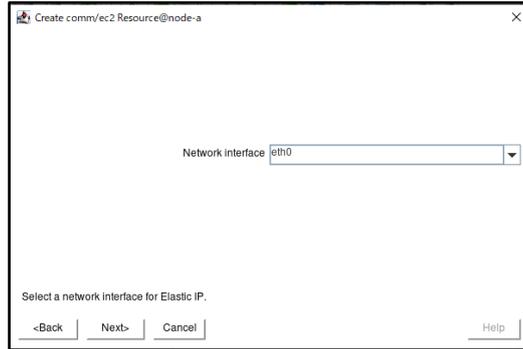
5. Server からリソース階層の作成先のサーバとして稼働系サーバを選択し、Next>ボタンを押します。



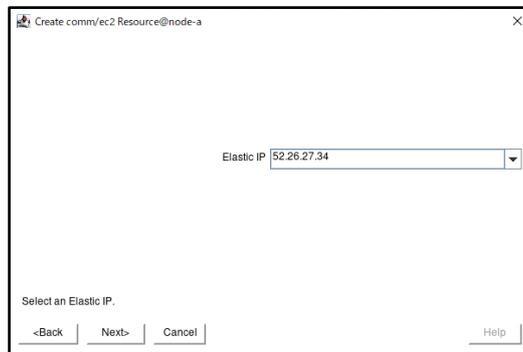
6. EC2 Resource Type から Elastic IP (Frontend Cluster) を選択し、Next>ボタンを押します。



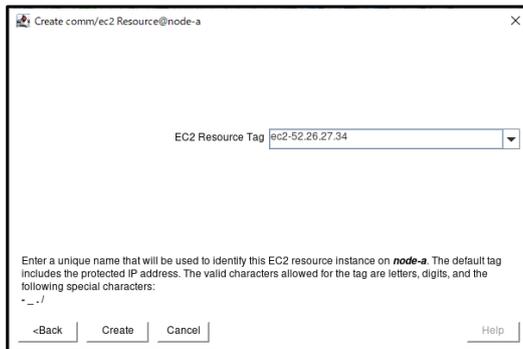
7. Network Interface から eth0 を選択し、Next>ボタンを押します。



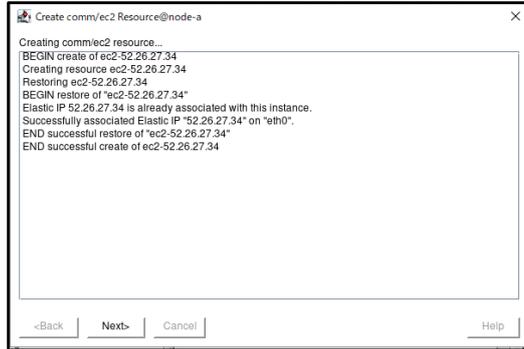
8. Elastic IP から IP アドレスを選択し、Next>ボタンを押します。



9. EC2 Resource Tag に EC2 リソースのタグ名を入力し、Create ボタンを押します。

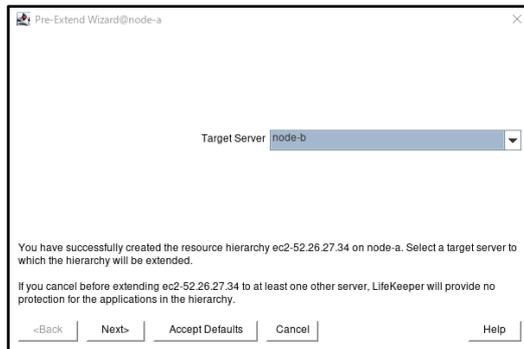


10. リソース階層が作成されます。



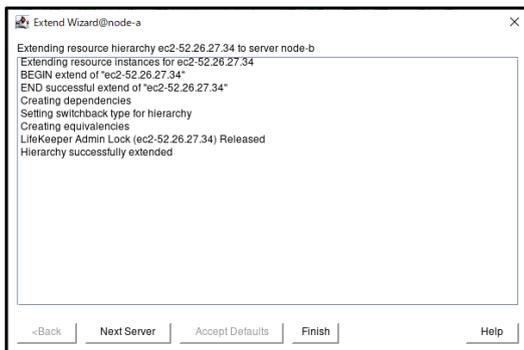
リソース階層の作成が完了したら、Next>ボタンを押します。

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。



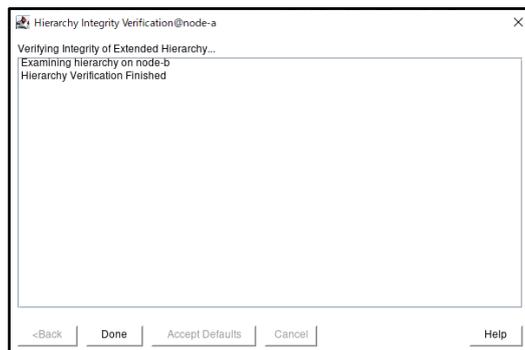
これ以降の設定を確認せずにデフォルトの設定のままで問題ない場合には Accept Defaults ボタンを押します。そうでない場合には Next>ボタンを押します。ここでは Accept Defaults ボタンを押した場合について説明します。

12. リソース階層が拡張されます。



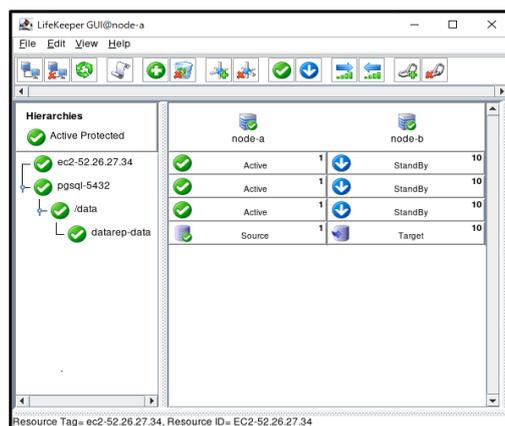
リソース階層の拡張が完了したら、Finish ボタンを押します。

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。



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

11 リソース依存関係の作成

PowerGres Plus リソースと EC2 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「[LifeKeeper 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。

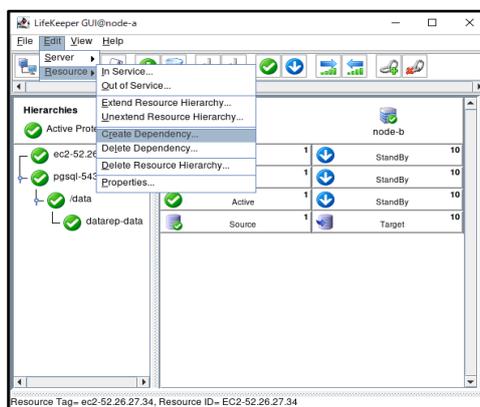
リソース依存関係の設定は以下のとおりです。

リソース依存関係の作成	
Server	node-a
Parent Resource Tag	pgsql-5432
Child Resource Tag	ec2-52.26.27.34

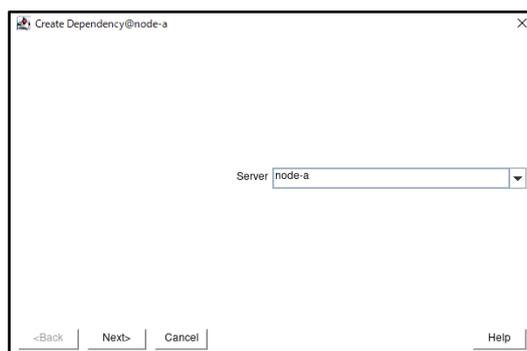
1. LifeKeeper GUI を起動し、接続します。

```
# lkGUIapp &
```

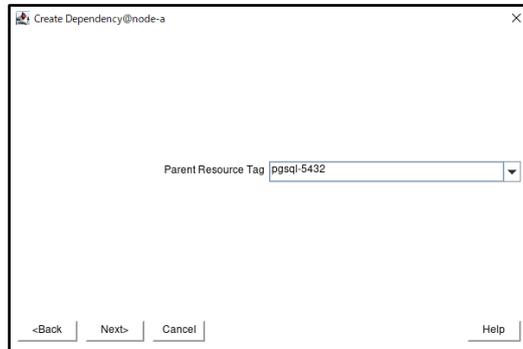
2. Edit メニューの Resource > Create Dependency... を選択します。



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。



- 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 Plus リソースのタグを指定します。



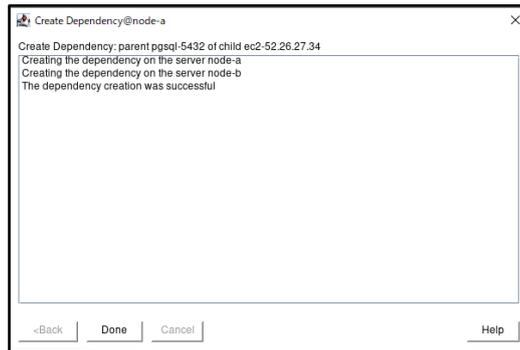
- Child Resource Tag にはリソースを関連付けるときに子になるリソースのタグを選択し、Next>ボタンを押します。ここでは EC2 リソースのタグを指定します。



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

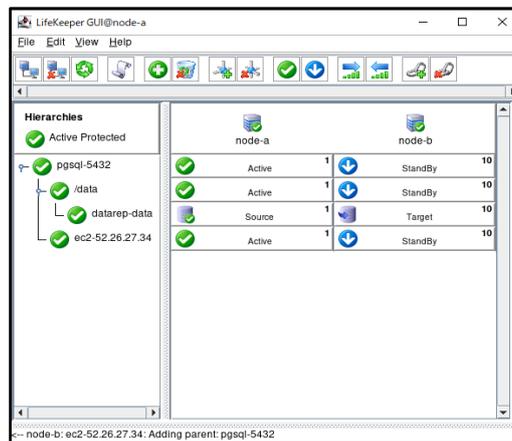


- リソースの依存関係が作成されます。



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. EC2 リソースが PowerGres Plus リソースに関連付けられたことを確認してください。



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

12 動作確認

初期設定が完了したことによって PowerGres Plus(Linux 版) 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。

12.1 スイッチオーバーの確認

1. PowerGres Plus(Linux 版) が起動していることを確認します。

PowerGres Plus(Linux 版) 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。

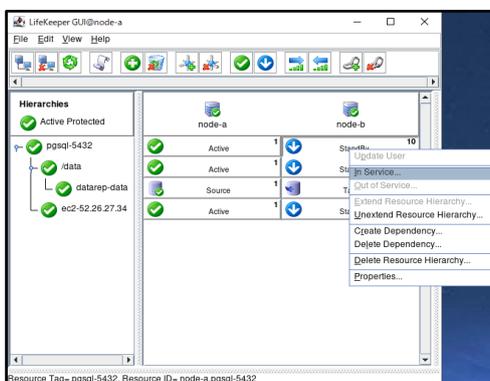
```
# ps -C postgres
  PID TTY          TIME CMD
 2910 ?            00:00:00 postgres
 2911 ?            00:00:00 postgres
 2913 ?            00:00:00 postgres
 2914 ?            00:00:00 postgres
 2915 ?            00:00:00 postgres
 2916 ?            00:00:00 postgres
 2917 ?            00:00:00 postgres
 2918 ?            00:00:00 postgres
```

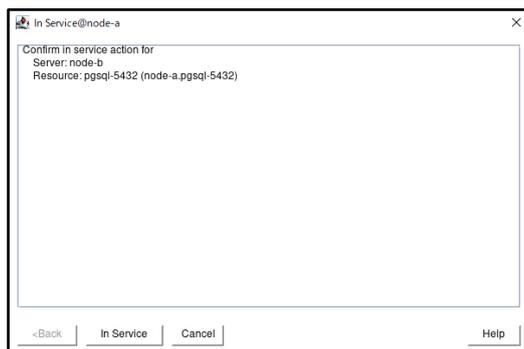
2. PowerGres Plus(Linux 版) をスイッチオーバーします。

LifeKeeper GUI を起動し、接続します。

```
# lkGUIapp &
```

3. 待機系側の PowerGres Plus リソースにて 右クリック後、In Service を押しします。対象となる Server と Resource が表示され、In Service ボタンを押すと PowerGres Plus(Linux 版) がスイッチオーバーします。

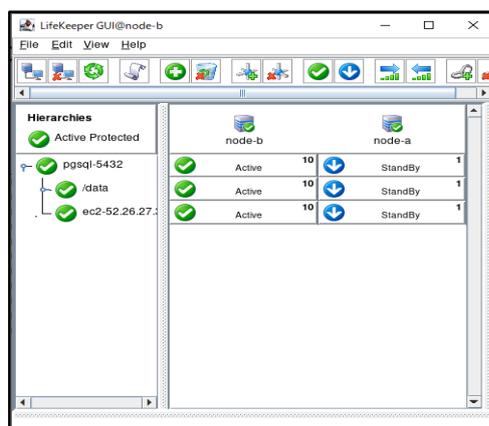




詳細は「[PostgreSQL Recovery Kit 管理ガイド](#)」の「[LifeKeeper GUI からの手動スイッチオーバーの実行](#)」を参照してください。ここで EC2 リソースを切り替えると、LifeKeeper の GUI は自動的に閉じます。

4. スイッチオーバーされたことを確認します。

待機系サーバに接続して、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。



また、スイッチオーバー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
72832 ?            00:00:00 postgres
72833 ?            00:00:00 postgres
72835 ?            00:00:00 postgres
72836 ?            00:00:00 postgres
72837 ?            00:00:00 postgres
72838 ?            00:00:00 postgres
72839 ?            00:00:00 postgres

```

```
72840 ?      00:00:00 postgres
```

12.2 フェイルオーバーの確認

- 1. Active サーバの PowerGres Plus(Linux 版) 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
3552 ?          00:00:00 postgres
3553 ?          00:00:00 postgres
3555 ?          00:00:00 postgres
3556 ?          00:00:00 postgres
3557 ?          00:00:00 postgres
3558 ?          00:00:00 postgres
3559 ?          00:00:00 postgres
3560 ?          00:00:00 postgres
```

- 2. PowerGres Plus(Linux 版) をフェイルオーバーします。

PowerGres Plus(Linux 版) 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>
# ps -C postgres
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 postgres
PID TTY      TIME CMD
66411 ?          00:00:00 postgres
66412 ?          00:00:00 postgres
66414 ?          00:00:00 postgres
66415 ?          00:00:00 postgres
66416 ?          00:00:00 postgres
66417 ?          00:00:00 postgres
66418 ?          00:00:00 postgres
66419 ?          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/powergresplus13/bin/postgres /opt/powergresplus13/bin/postgres~
```

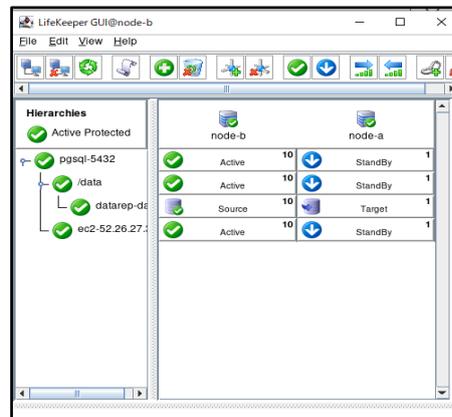
```
# kill <pid>
```

しばらくすると PowerGres Plus (Linux 版) が優先度の高いサーバへフェイルオーバーします。

3. 待機系サーバに LifeKeeper GUI を起動し、接続します。

```
# lkGUIapp &
```

4. フェイルオーバーされたことを確認します。
LifeKeeper GUI にて稼働系サーバのリソースが Standby へ変更され、待機系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。



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 postgres
```

```
PID TTY TIME CMD
38493 ? 00:00:00 postgres
38494 ? 00:00:00 postgres
38496 ? 00:00:00 postgres
38497 ? 00:00:00 postgres
38498 ? 00:00:00 postgres
38499 ? 00:00:00 postgres
38500 ? 00:00:00 postgres
38501 ? 00:00:00 postgres
```

なお、フェイルオーバー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/powergresplus13/bin/postgres~ /opt/powergresplus13/bin/postgres
```