

# LifeKeeper for Linux v9.5.0 スタートアップガイド

初版



サイオステクノロジー株式会社

## 目次

1	LifeKeeper for Linux 概要	3
1.1	LifeKeeper for Linux 概要	3
1.2	本マニュアルについて	4
1.3	対象読者について	4
1.4	サイオステクノロジーについて	4
1.5	サイオステクノロジーへのお問い合わせ	4
1.6	サポートへのお問い合わせ	5
1.7	製品に関する情報	5
2	インストール前の要件	6
2.1	リリースノートの確認	6
2.2	TCP/IP 接続と名前解決の確認	6
2.3	Firewall の確認	6
2.4	SELinux の確認	7
2.5	LifeKeeper for Linux v9.5.0 の動作に必要なパッケージの確認	7
2.6	既知の問題の確認	7
3	LifeKeeper for Linux v9.5.0 のインストール	8
3.1	インストール手順について	8
①	setup スクリプトの実行	8
②	Install Java Runtime (JRE)	10
③	Use Quorum / Witness functions	10
④	LifeKeeper authentication	10
⑤	Install license key file(s)	12
⑥	Recovery Kit Selection	13
⑦	LifeKeeper startup after install	13
⑧	LifeKeeper のインストール	13
⑨	残りのノードへの LifeKeeper のインストール	14
3.2	Quorum/Witness Server Support パッケージについて	15
4	LifeKeeper の起動と停止	16
4.1	LifeKeeper の起動	16
4.2	LifeKeeper の停止	17
4.3	LifeKeeper の GUI サーバーについて	17
•	GUI サーバーの起動	17
•	GUI サーバーの停止	17
5	クラスタシステムの作成	17
5.1	LifeKeeper GUI クライアントでの接続	18
5.2	コミュニケーションパスの作成	20
5.3	リソースの作成	21
5.4	ファイルシステムリソースの作成	21
5.5	ファイルシステムのマウント	22
5.6	ファイルシステムリソースの作成	22
6	その他	32

6.1	製品サポートへお問い合わせいただく際に収集すべき情報について .....	32
6.2	よく使用する LifeKeeper コマンドについて .....	33
6.3	CUI によるリソースの起動、停止およびスイッチオーバー.....	34
6.4	GUI クライアントのステータス表示 .....	35
6.5	LifeKeeper のアンインストール.....	35
7	免責事項.....	36
8	著作権.....	36

## 改訂履歴

2020 年 3 月 6 日初版

## 1 LifeKeeper for Linux 概要

### 1.1 LifeKeeper for Linux 概要

LifeKeeper はシステムの可用性を高めることを目的とした HA クラスタソフトウェアです。

#### ■ HA クラスタとは

あるサーバーで障害を検知した場合に、他のサーバーへ自動的にサービスを切り替えることにより、障害からの復旧速度を向上させます。その結果として、年間のダウンタイムを短縮させ、システムの可用性の向上を目指します。

#### ■ Availability(稼働率・可用性)のレベル

想定されるシステムの年間ダウンタイムの長さによって、可用性レベルが分類できます。LifeKeeper はフォルトレジリエントレベルの可用性を持つシステムを構築できるよう設計されています。

**表 1 Availability(稼働率・可用性)のレベル**

Availability クラス	Availability レベル	年間のダウンタイム
連続処理(Continuous Processing)	100%	0 分
フォルトトレラント(Fault Tolerant)	99.999%	5 分
フォルトレジリエント (Fault Resilient)	99.99%	53 分
一般の商用 Availability	99 - 99.5%	44-87 時間

#### ■ 主な特徴

- ソフトウェアでのフォルトレジリエントソリューションを提供します。
- 統一インターフェースによる HA システム構築を可能とします。
- GUI による簡単で直感的な HA クラスタ管理を提供します。
- ハートビート通信によるノードの死活監視と、リソース単位での監視を行い、障害と判定した場合にサービスを自動的にスタンバイノードへ引き継ぎます。

#### ■ ノードの死活監視

LifeKeeper はコミュニケーションパスを通じたハートビートの応答確認によってサーバーの死活監視を行っています。デフォルトでは、5 秒に一度のハートビートが 3 回連続して失敗するとハートビート断と判定します。ハートビート断が全てのコミュニケーションパスで発生するとフェイルオーバーを行いません。コミュニケーションパスは以下の用途に使用します。

- コミュニケーションパス
  - ✓ サーバーの死活監視を行うためのハートビートの通信経路
  - ✓ LifeKeeper のノード間の情報のやり取り

## ■ リソース単位での監視

LifeKeeper は監視対象となるアプリケーション、ファイルシステム等のサービスをリソースと呼びます。デフォルトでは 120 秒に 1 回の頻度でリソースの状態を監視しています。リソース障害を検知すると、リソース監視が動作しているノード上でリソースの回復処理を行ないます。回復に失敗するとフェイルオーバーを行ないます。

### ➤ リソースの種類

大きく分類して以下の種類のリソースがあります。LifeKeeper は個々のリソースに対して監視を行ないます。

- ✓ アプリケーションリソース(Oracle、PostgreSQL 等)
- ✓ IP リソース(仮想 IP アドレス)
- ✓ ファイルシステムリソース

## 1.2 本マニュアルについて

本マニュアルでは、LifeKeeper for Linux v9.5.0 のインストール手順を解説します。LifeKeeper の使い方や、運用方法に関する情報を提供するものではありません。LifeKeeper の使い方に関する詳しい情報は、ユーザーポータルや SIOS Technology Corp. の Web サイト内のドキュメントを参照してください。

また、記載されている rpm パッケージや画像に記載されているバージョン番号等については、インストールに影響のないものについては実際のバージョンと一致しない場合があります。その場合は実際に利用されているものに読み替えて使用してください。

## 1.3 対象読者について

本マニュアルは、Linux オペレーティングシステムについて基本的な知識を持っている技術者を対象としています。

## 1.4 サイオステクノロジーについて

サイオステクノロジーは、1997 年の創業以来、オープンソースソフトウェアを軸に、Web アプリケーションや OS、IT、システムの開発/基盤構築/運用サポート等の事業を展開し、現在はこれらにクラウド技術を加え、新たな価値創造とそこご提供に取り組んでおります。サイオステクノロジーに関する詳細については、<http://www.sios.com> をご参照ください。

## 1.5 サイオステクノロジーへのお問い合わせ

サイオステクノロジー株式会社

〒106-0047 東京都港区南麻布 2 丁目 12-3 サイオスビル

日本国内および海外の事業所の情報に関しては、弊社の Web サイトをご参照ください。

#### 1.6 サポートへのお問い合わせ

お問い合わせの一次窓口が弊社ではない場合があります。お問い合わせの際はサポート証書よりサポート窓口をご確認ください。サポート窓口が弊社になっている場合は、下記の Web サイトよりお問い合わせください。

<https://sios.jp/products/lkdk/contact/>

#### 1.7 製品に関する情報

製品ドキュメントに関する情報は、下記のリンクよりご参照ください。製品のリリースノートや Recovery Kit の管理ガイドがあります。Japanese Documentation のリンクより日本語マニュアルをご確認いただけます。

<http://docs.us.sios.com/>

Recovery Kit の動作概要、製品の Errata 情報、ライセンスの取得方法などに関する情報は、ユーザーポータルでご確認いただけます。

<https://lkdkuserportal.sios.jp/hc/ja/>

## 2 インストール前の要件

LifeKeeper のインストールを開始する前に、以下の事項を確認します。

### 2.1 リリースノートの確認

インストールを開始する前にリリースノートをよく読みます。リリースノートには重要な情報が含まれています。リリースノートを参照して、サポートされているプラットフォーム、オペレーティングシステム、アプリケーション、ストレージを確認します。リリースノートは SIOS Technology Corp.の以下の Web サイトから参照できます。

<http://docs.us.sios.com/>

### 2.2 TCP/IP 接続と名前解決の確認

GUI の機能を使用するためにクラスタノードの双方で名前解決ができる必要があります。名前解決には、DNS サービスもしくは/etc/hosts を使用します。また、localhost が 127.0.0.1 に解決される必要があります。

### 2.3 Firewall の確認

以下のポートを使用しています。

- コミュニケーションパス(TCP)の通信用:7365/tcp
- GUI サーバーの通信用:81/tcp、82/tcp
- GUI サーバー、クライアント間の RMI 通信用: 1024/tcp 以降の全てのポート
- DataKeeper の同期用 (DataKeeper 使用時):"10001+<mirror number>+<256 \* i>"

---

### 補足

- GUI サーバー、クライアントの通信に使用するポートは LifeKeeper がインストールされるクラスタノードと、GUI クライアントを動作させる全てのシステムでポートが開放されている必要があります。
  - DataKeeper で使用するポートは上記の計算式で割り出すことができます。i の値は 0 から開始し、使用されていないポートを見つけるとそのポートを使用します。例えば mirror number が 0 の DataKeeper リソースが存在している環境で、10001 番ポートが別のアプリケーションによって使用されていた場合は、10257 番ポートが使用されます。
  - GUI サーバー、クライアント間の通信では Java の RMI(Remote Method Invocation) にて 1024 番以降のポートをランダムに使用します。クラスタシステムにアクセス制御などを適用する場合は、これらのポートを考慮し、パケットフィルタリングを行う必要があります。セキュリティー対策の観点で本仕様が問題となる場合は、ssh の X フォワーディングを使用して対応することもできます。設定方法等はテクニカルドキュメントをご参照ください。
-

#### 2.4 SELinux の確認

SELinux の設定が enabled の場合、LifeKeeper はインストールできません。SELinux を disabled にするには OS ディストリビューションのドキュメントを参照してください。SAP 環境で必要な場合を除いて、SELinux の permissive モードを使用することはお勧めしません。クラスタ上で実行されるアプリケーションが permissive モードをサポートしていることを確認してください。SELinux の permissive モードは次の ARK でテストされています。

SAP / SAP MaxDB / Sybase / Oracle / DB2 / NFS / DataKeeper / NAS / EC2 / IP / FileSystem / MQ

#### 2.5 LifeKeeper for Linux v9.5.0 の動作に必要なパッケージの確認

必要なパッケージはテクニカルドキュメント内の Setting Up Environment セクションの Linux Dependencies に記載しています。

#### 注意

---

- ご利用のディストリビューションから提供されている適切なパッケージをインストールしてください。
  - sg3\_utils パッケージは DMMP Recovery Kit、PowerPath Recovery Kit 等のマルチパス用のリカバリキットを使用する環境に必要です。マルチパス用のリカバリキットを使用しない環境には必要ありません。
- 

#### 2.6 既知の問題の確認

既知の問題は、テクニカルドキュメント内の Troubleshooting セクションの Known Issues and Restrictions に記載しています。ご利用の環境に該当する既知の問題がないか確認します。また、最新の情報につきましては、ユーザーポータル「Errata/制限事項」に記載されている場合がございますので、こちらも合わせてご確認ください。



### 3 LifeKeeper for Linux v9.5.0 のインストール

#### 3.1 インストール手順について

LifeKeeper のインストールは、インストール CD に含まれる setup スクリプトを実行することによって行われます。スクリプトは対話形式となっており、各内容に対して選択を行うか、もしくは適切な入力を行います。設定ファイルを用いた非対話でのインストールも可能ですが、ここでは取り扱いません。この章では setup スクリプトによる LifeKeeper のインストール方法を記載しています。

---

#### 補足

以降の手順に掲載されている LifeKeeper の各パッケージ番号は、実際にリリースされているものと異なる場合がありますが操作手順には影響ありません。ご利用のバージョンに読み替えてご利用ください。

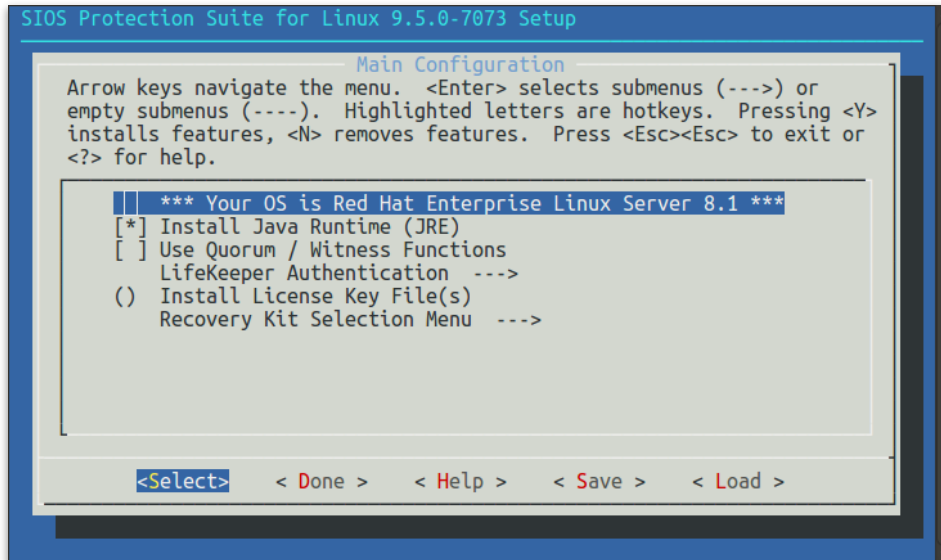
---

#### ① setup スクリプトの実行

製品の CD-ROM を任意のディレクトリにマウントします。続いて sps\_xxx.img (xxx はバージョン番号) を任意のディレクトリにマウントし、setup を実行します。以下は CD イメージをマウントする際の操作例です。

```
# mount /dev/cdrom /media/cdrom
# cd /media/cdrom
# mount sps_xxx.img -t iso9660 -o loop /mnt
# cd /mnt/
# ./setup
```

setup スクリプトを起動すると、以下のメニュー画面が表示されます。「Pre-Install Warning」画面が表示された場合は内容を確認し <Continue> で先に進めてください。



メニューは以下のキーで操作します。

↑↓：選択項目の移動

←→：最下行ボタンの移動

ENTER：サブメニューを開く

Y / N / SPACE：選択項目の ON / OFF / 反転

最下行ボタンは以下の動作を行います

Select：詳細画面を開きます

Done：この画面を閉じて一つ上の画面に戻ります。メイン画面の場合は構成を確定します。

Help：選択中項目のヘルプテキストを表示します

Save：現在の設定を構成ファイルとして保存します。これは非対話インストールで使用します。

Load：保存した構成ファイルを読み込みます。

以降の章でメニューの各項目について説明を行います。

---

② Install Java Runtime (JRE)

LifeKeeper GUI が使用する Java 実行環境をインストールします。

---

**補足**

LifeKeeper で使用している Java は LifeKeeper の設定ファイル(/etc/default/LifeKeeper) を参照しています。OS の環境変数は使用していないため、複数のバージョンの Java を混在させることができます。製品同梱の Java パッケージと他の Java パッケージが混在しても、LifeKeeper の動作に影響はありません。

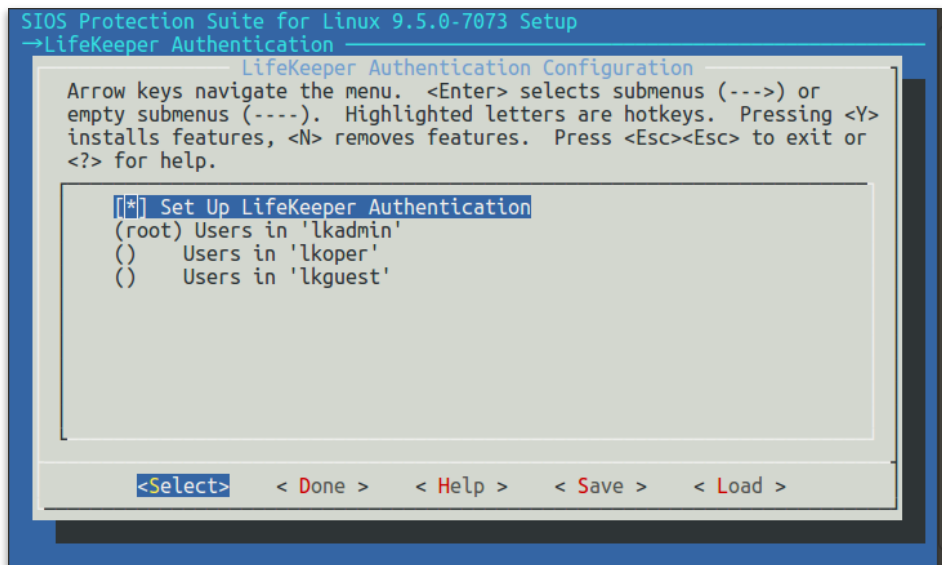
---

③ Use Quorum / Witness functions

Quorum/Witness を使用します。機能の詳細はテクニカルドキュメンテーションの Quorum/Witness を参照してください。

④ LifeKeeper authentication

LifeKeeper GUI のログインに使用するユーザを登録できます。空白で区切ることで複数のユーザを指定できます。詳細はインストレーションガイドの GUI ユーザーの設定を参照してください。



---

### 注意

全ての項目を未設定でインストールを実施した場合は LifeKeeper インストール完了後に、お客様にて以下の設定を行う必要があります。設定が行われていない場合は、LifeKeeper の GUI クライアントにログインできません。n を間違えて選択してしまった場合には以下のように OS の操作でユーザーグループの作成と追加を行なってください。

=>OS のコマンドを使用し、下記のグループを追加

lkguest

lkoper

lkadimin

※任意のグループ ID を指定できます。

=>作成した lkadmin のグループに root ユーザを追加

ユーザ管理に関する情報につきましては、以下の URL にございますユーザーポータルの情報も合わせてご参照ください。

[Linux]GUI 管理画面にアクセスできるユーザの設定方法(v8.1.1 以降)

<https://lkduserportal.sios.jp/hc/ja/articles/360037732471>

---

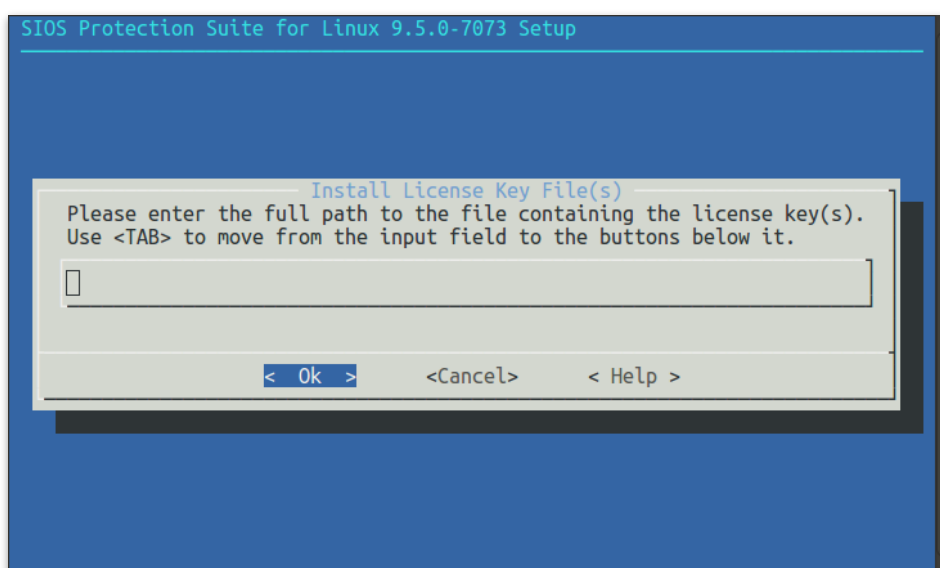
LifeKeeper 関連のユーザ、グループが既に作成されている状態でインストールを実施した場合は、そのユーザ、グループ情報は維持されます。

LifeKeeper 関連のユーザ、グループが既に作成されている状態でそれらユーザを削除した場合は、そのユーザ、グループ情報は削除されます。

---

⑤ Install license key file(s)

インストールするライセンスファイルのパス名を入力します。空白で区切ることで複数のファイルを指定できます。



---

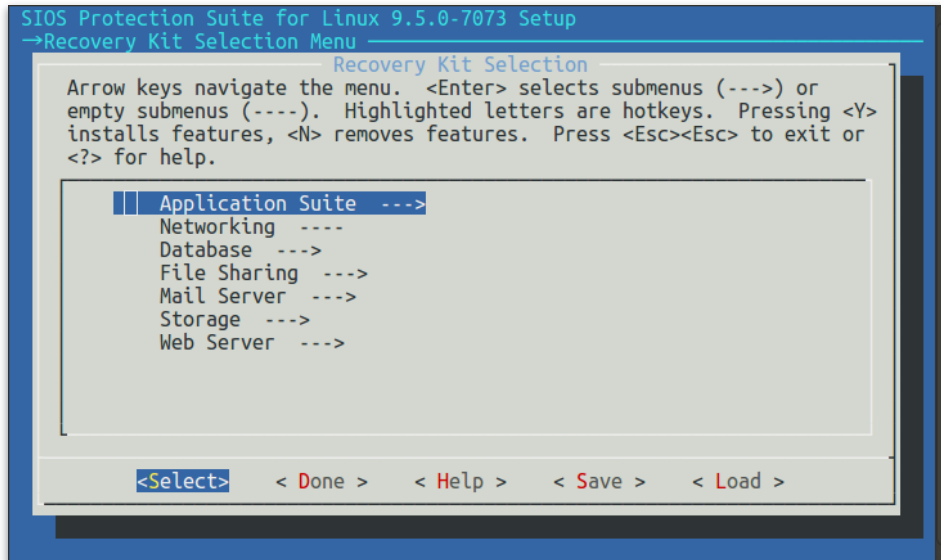
**補足**

ライセンス取得方法とライセンスのインストールの詳細な手順は「ライセンス取得方法解説ページ」<https://lkduserportal.sios.jp/hc/ja/articles/360037843531> をご参照ください。

---

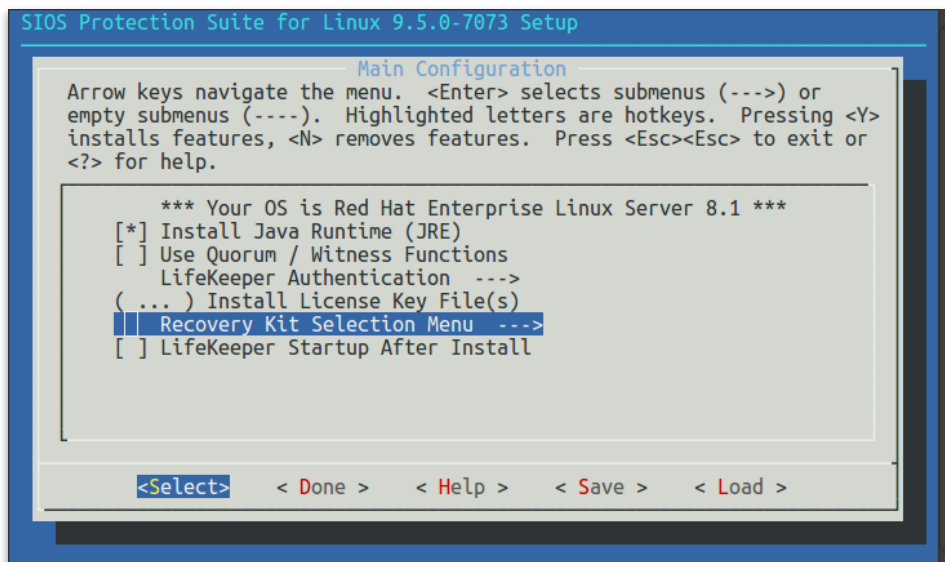
## ⑥ Recovery Kit Selection

使用するリカバリキットを選択します。



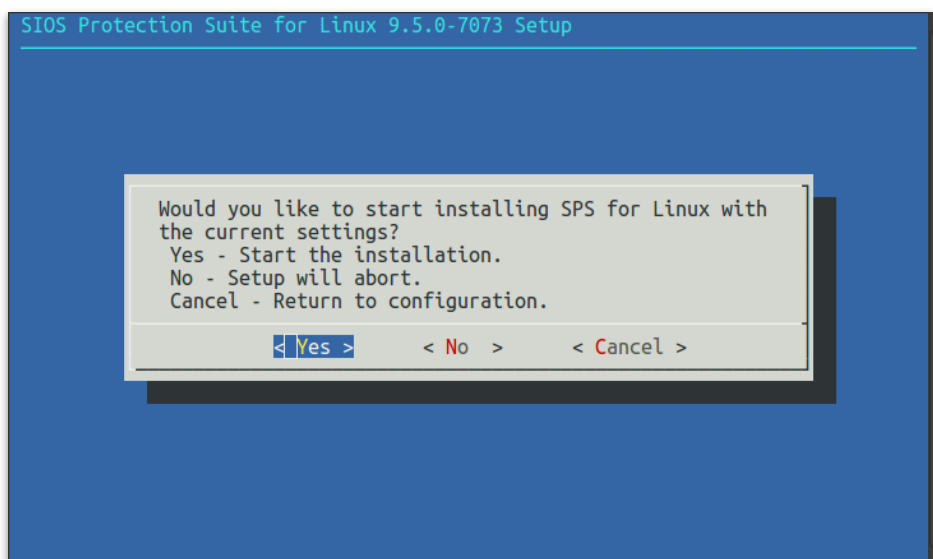
## ⑦ LifeKeeper startup after install

ライセンスを指定した場合、この項目が表示されます。インストール作業の完了時に LifeKeeper を起動する場合はこの項目を選択してください。



## ⑧ LifeKeeper のインストール

全ての項目の選択が完了した場合、メインメニューから **<Done>** を選択してください。以下の画面が表示されます。



問題がなければ**<Yes>**を選択、インストールを実施してください。インストール完了後、「Setup Complete.」が出力されればアップデートは完了です。インストールに失敗した場合は対応したエラーメッセージが出力されます。問題を解決した上で再度インストールを実施してください。

- ⑨ 残りのノードへの LifeKeeper のインストール  
LifeKeeper をインストールする対象になっているノードも同様の手順でパッケージをインストールします。

インストールに関する手順は以上です。

---

#### 補足

- LifeKeeper はデフォルトで/opt/LifeKeeper 以下にインストールされます。シェルの環境変数に以下の行を追加するとパスの設定ができます。

```
例) /root/.bash_profile
# For LifeKeeper
PATH=$PATH:/opt/LifeKeeper/bin
MANPATH=$MANPATH:/opt/LifeKeeper/man
export PATH MANPATH
```

- 下記のグループ、ユーザは LifeKeeper for Linux v9.5.0 に必要なグループ、ユーザです。

グループ

steeleye-lighttpd,lkguest,lkoper,lkadmin

ユーザ

steeleye-lighttpd

---

#### 3.2 Quorum/Witness Server Support パッケージについて

v9.3 で AWS S3 や共有ディスクを用いたディスク Quorum/Witness モードをサポートしました。Quorum/Witness Server 方式の設定方法、動作シナリオの詳細については、以下のオンラインマニュアルをご覧ください、SIOS Protection Suite for Linux についての LifeKeeper 内の「Quorum/Witness」のページをご参照ください。

<http://jpdocs.us.sios.com/>

---

#### 注意

リリースノートにてサポートを表明しているストレージを共有ディスクとして使用する場合、特別な記述がない限りは本パッケージをインストールする必要はありません。後から追加インストールすることもできます。追加インストールするには、クラスタシステムの停止は伴いません。

---



## 4 LifeKeeper の起動と停止

### 4.1 LifeKeeper の起動

LifeKeeper を起動するには、以下のいずれかのコマンドを実行します。

- lkcli start コマンド
- lkstart コマンド
- systemctl start lifekeeper.service (systemd 環境)

systemctl start lifekeeper.service で LifeKeeper を起動した場合、サーバの起動時に LifeKeeper が自動的に起動しません。LifeKeeper の自動起動を有効化するには以下のコマンドを実行します。

- systemctl enable lifekeeper.service (systemd 環境)

```
# lkstart
Created symlink /etc/systemd/system/lifekeeper-graphical.target.requires/lifekeeper.service → /usr/lib/systemd/system/lifekeeper.service.
Created symlink /etc/systemd/system/lifekeeper-multi-user.target.requires/lifekeeper.service → /usr/lib/systemd/system/lifekeeper.service.
```

または

```
# systemctl start lifekeeper.service
# systemctl enable lifekeeper.service
Created symlink /etc/systemd/system/lifekeeper-graphical.target.requires/lifekeeper.service → /usr/lib/systemd/system/lifekeeper.service.
Created symlink /etc/systemd/system/lifekeeper-multi-user.target.requires/lifekeeper.service → /usr/lib/systemd/system/lifekeeper.service.
```

LifeKeeper の起動確認は、lktest コマンドまたは systemctl status lifekeeper.service を実行します。lktest コマンドは、LifeKeeper の主要プロセスの状態をチェックし、ps -cwf コマンドから得られた結果を表示します。稼動しているプロセスによっては複数出力されます。プロセスが正常に起動していることが確認できた場合は、戻り値に 0 が返りません。

```
# lktest
F  S UID      PID    PPID    C   CLS PRI  NI  SZ    STIME    TIME    CMD
4  S root     8986   8009    0   TS   39 -20 6685 15:34    00:00:00 lcm
4  S root     8991     1      0   TS   39 -20 6683 15:34    00:00:00 eventslcm -l
4  S root     8995   8015    0   TS   39 -20 6685 15:34    00:00:00 ttymonlcm
4  S root     8999   8008    0   TS   29 -10 8717 15:34    00:00:00 lcd
```

systemctl status lifekeeper.service は lifekeeper.service のステータスを表示します。正常に起動している場合は、以下の様に表示されます。

```
# systemctl status lifekeeper.service
● lifekeeper.service - SIOS LifeKeeper
   Loaded: loaded (/usr/lib/systemd/system/lifekeeper.service; enabled; vendor preset: enabled)
   Active: active (exited) since Wed 2019-09-18 16:51:24 JST; 5min ago
 Main PID: 32095 (code=exited, status=0/SUCCESS)
    Tasks: 0 (limit: 11517)
   Memory: 2.8M
    CGroup: /system.slice/lifekeeper.service
```

## 4.2 LifeKeeper の停止

LifeKeeper を停止するには、以下のいずれかのコマンドを実行します。起動に使用したコマンドと対応したコマンドを使用してください。

- lkcli stop コマンド
- lkstop コマンド
- systemctl stop lifekeeper.service (systemd 環境)

```
# lkstop
Removed /etc/systemd/system/lifekeeper-graphical.target.requires/lifekeeper.service.
Removed /etc/systemd/system/lifekeeper-multi-user.target.requires/lifekeeper.service.
```

## 4.3 LifeKeeper の GUI サーバーについて

インストール完了後の LifeKeeper の設定は LifeKeeper GUI クライアントを使用しています。LifeKeeper GUI クライアントを操作するためには、LifeKeeper がインストールされているサーバー上で LifeKeeper GUI サーバーが起動している必要があります。通常 GUI サーバーは LifeKeeper の起動、停止と連動して起動と停止が行われますが、GUI サーバーを単独で停止したり起動したりすることも可能です。GUI サーバーが停止していたとしても、障害検知やフェイルオーバーなどの HA クラスタとしての機能には影響しません。クラスタ環境をセットアップ後、GUI 操作が必要でなければ、平時は GUI サーバーを停止しておき、必要な時だけ GUI サーバーを起動するといった運用もできます。

LifeKeeper の GUI サーバーの起動と停止の方法は以下の通りです。

- GUI サーバーの起動

GUI サーバーを起動する場合は、lkGUIserver start コマンドを実行します。

```
# lkGUIserver start
LifeKeeper GUI Server Setup Starting
LifeKeeper GUI Server Enabled
ok: run: /opt/LifeKeeper/etc/service/lkguiserver: (pid 21818) 0s, normally down
LifeKeeper GUI Server Setup Completed
```

- GUI サーバーの停止

GUI サーバーを停止する場合は、lkGUIserver stop コマンドを実行します。

```
# lkGUIserver stop
LifeKeeper GUI Server Setup Starting
LifeKeeper GUI Server Disabled
ok: down: /opt/LifeKeeper/etc/service/lkguiserver: 0s
LifeKeeper GUI Server Setup Completed
```

※オプションとして restart を使用すれば再起動をすることもできます。GUI 表示に問題が生じた場合には、GUI サーバーの再起動で復旧する場合があります。

## 5 クラスタシステムの作成

LifeKeeper でクラスタシステムを作成するには、HA クラスタを構成するノード間に「コミュニケーションパス」を設定する必要があります。その後、保護対象を定義するため「リソース」を作成します。この資料ではコミュニケーションパスの設定とファイルシステムリソースの作成を解説します。

## 5.1 LifeKeeper GUI クライアントでの接続

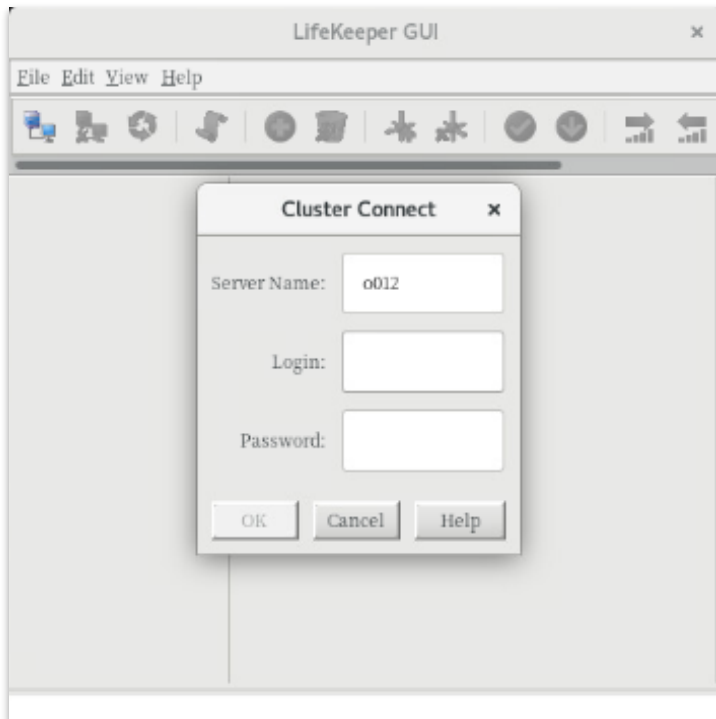
LifeKeeper の設定作業は GUI を使用して行います。

GUI クライアントは lkGUIapp コマンドで起動します。LifeKeeper を起動したあとで、以下のコマンドで LifeKeeper GUI クライアントを起動します。以下はその実行例です。

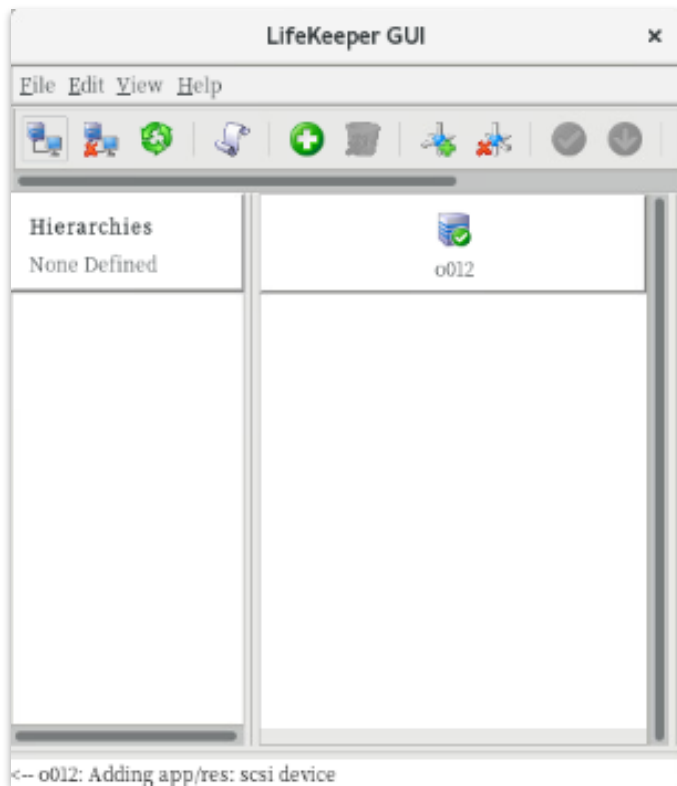
```
# lkGUIapp
openjdk version "1.8.0_232"
OpenJDK Runtime Environment (build 1.8.0_232-b09)
OpenJDK 64-Bit Server VM (build 25.232-b09, mixed mode)
Setting up secure random number generator
Random number setup completed
```

コマンドを実行後、GUI クライアントが起動し、ログイン画面が立ち上がります。

Server Name には実行したサーバー名が入ります。ログインユーザ名とパスワードは LifeKeeper の管理ユーザの情報を入力します。管理ユーザの情報はデフォルトで OS のスーパーユーザ(root)とそのパスワードが引き継がれます。



ログイン後、LifeKeeper が起動しており、LifeKeeper GUIserver との接続に問題がなければ、以下のような画面が表示されます。



#### 補足

Web ブラウザ経由でリモートホストから GUI 操作を行うこともできます。Web ブラウザを使用する場合は、アクセスするリモートホストとクラスタサーバー間で相互に名前解決できる必要があります。Web ブラウザの場合は 81 番ポートを使用します。http://ホスト名:81 若しくは http://IP アドレス:81 のように入力しリモートホストからアクセスします。

なお、ブラウザを使用して LifeKeeper GUI を使用するには、いくつかの必要要件があります。その詳細につきましては、リリースノートやテクニカルドキュメンテーションを合わせてご確認ください。この要件のため、ネットワーク環境によってはブラウザでの操作が難しい場合もあります。そのような場合には、リモートデスクトップや SSH X フォワーディングの利用なども検討するようにしてください。

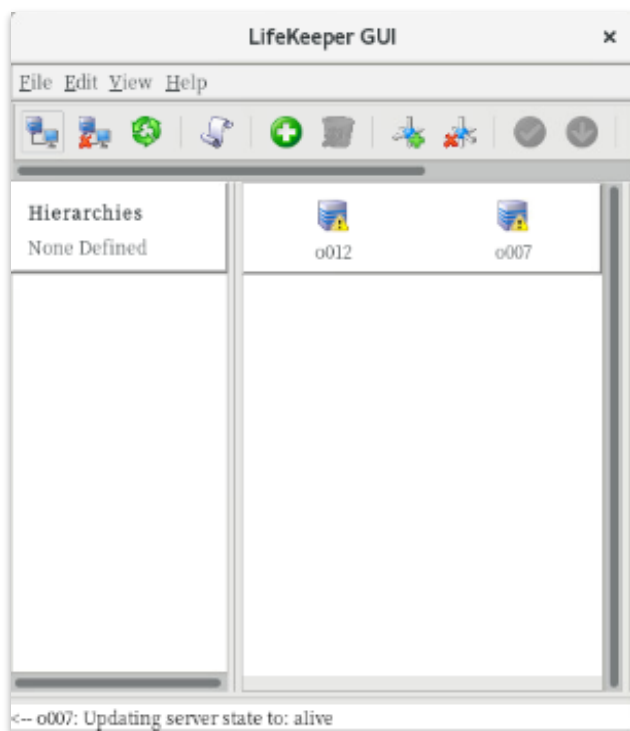
## 5.2 コミュニケーションパスの作成

GUI クライアントでコミュニケーションパスを作成するためのウィザードを起動し、コミュニケーションパスを作成します。GUI クライアントのツールバーの[Edit]を選択し、[Server]、[Create Comm Path...]を実行します。設定は下記の順序で行ないます。最後に[Create]を選択するとコミュニケーションパスの作成が開始されます。

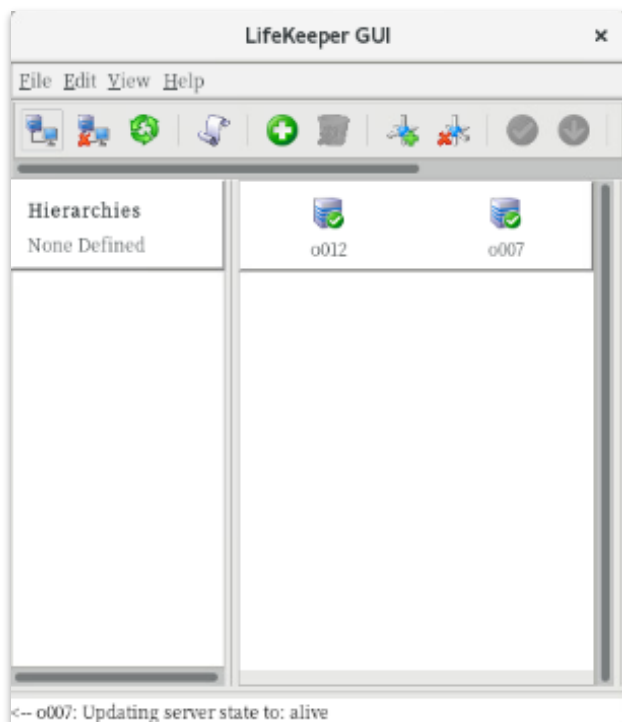
表 2 コミュニケーションパスの設定値

番号	項目	入力もしくは選択する値
1	Local Server	プライマリサーバーを選択
2	Remote Server	バックアップサーバー名を入力もしくは選択
3	Device Type	TCP もしくは TTY を選択
4	Local IP Address	IP アドレスを選択
5	Remote IP Address	IP アドレスを選択
6	Priority	優先順位を設定

コミュニケーションパスを 1 つだけ作成した場合は、GUI クライアントのサーバーアイコンが警告で表示されます。



1つ目のコミュニケーションパスの作成と同様の操作で予備のコミュニケーションパスを作成してください。コミュニケーションパスを2つ以上作成すると下記のようにサーバーアイコンが緑色で表示されます。



### 5.3 リソースの作成

保護するサービスやアプリケーションに対応するリソースを作成します。本ドキュメントでは、ファイルシステムリソースの作成手順を例示します。

---

#### 注意

各 Recovery Kit のリソースの作成手順は、テクニカルドキュメントより参照することができます。Recovery Kit 毎にセットアップ時の要件や注意点がありますので、ご利用になる Recovery Kit 用のマニュアルを参照のうえ、リソースの作成を行なってください。

---

### 5.4 ファイルシステムリソースの作成

ファイルシステムリソースは共有ストレージ上のファイルシステムをクラスタノード間で切り替えることを可能する機能を提供します。ファイルシステムリソースを作成するには、以下の条件を満たしている必要があります。

- 共有ストレージが物理的に接続され、各サーバーから同じ状態で接続できること
- parted などのユーティリティを使い、GUID パーティションテーブル(GPT)で共有ディスクのパーティションが切られていること
- mkfs などのユーティリティを使い、ファイルシステムが作成されていること
- 各サーバーでファイルシステムをマウント、アンマウントができること

ファイルシステムリソースの作成条件を満たしていることが確認できたら次の手順に移ります。

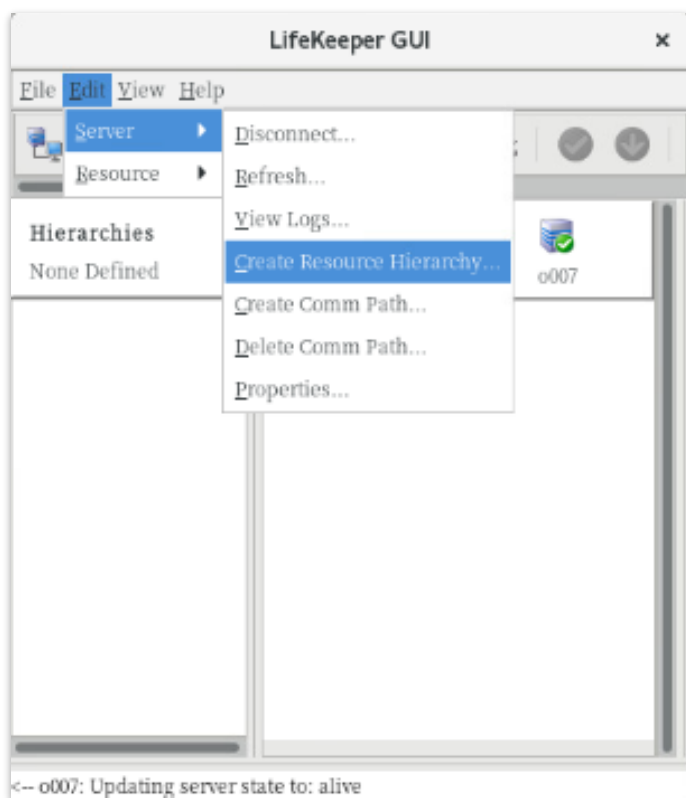
### 5.5 ファイルシステムのマウント

ファイルシステムリソース作成対象のファイルシステムを任意のディレクトリにマウントします。例では/dev/sdb1 を/mnt/fs にマウントしています。

```
# df
ファイルシス      1K-ブロック   使用   使用可  使用%  マウント位置
devtmpfs           921428         0   921428    0% /dev
tmpfs              936752         0   936752    0% /dev/shm
tmpfs              936752   44940   891812    5% /run
tmpfs              936752         0   936752    0% /sys/fs/cgroup
/dev/mapper/rhel-root 14034944 5061184  8973760   37% /
/dev/sdal          1038336  172828   865508   17% /boot
tmpfs              187348         28   187320    1% /run/user/42
/dev/sdb1         16765932  150036 16615896    1% /mnt/fs
```

### 5.6 ファイルシステムリソースの作成

GUI クライアントでファイルシステムリソースを作成するためのウィザードを起動し、ファイルシステムリソースを作成します。GUI クライアントのツールバーの[Edit]を選択し、[Server]、[Create Resource Hierarchy]を実行します。



ファイルシステムリソースの作成ウィザードの内容は以下の表の通りです。

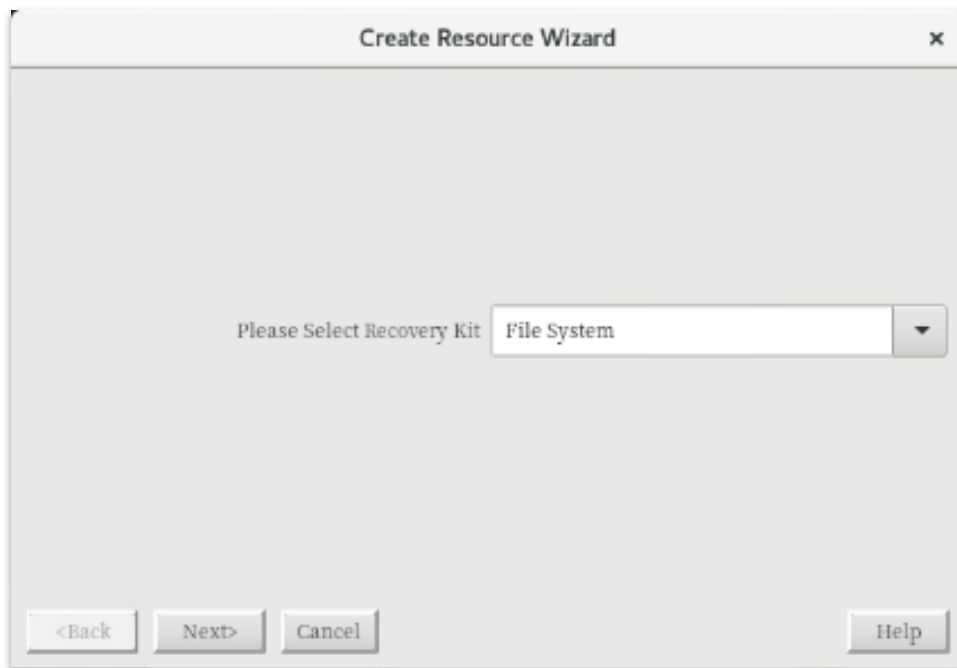
表 3 ファイルシステムリソースの設定値

番号	項目	入力もしくは選択する値
1	Please Select Recovery Kit	File System を選択
2	Switchback Type(プライマリサーバー)	intelligent もしくは automatic を選択
3	Server	プライマリサーバー名を選択
4	Mount Point(プライマリサーバー)	マウントポイントを選択
5	Root Tag(プライマリサーバー)	タグ名を選択もしくは入力
6	Target Server	バックアップサーバー名を選択
7	Switchback Type(バックアップサーバー)	intelligent もしくは automatic を選択
8	Template Priority(プライマリサーバー)	デフォルト値を選択
9	Target Priority(バックアップサーバー)	デフォルト値を選択
10	Mount Point(バックアップサーバー)	デフォルト値を選択
11	Root Tag(バックアップサーバー)	デフォルト値を選択

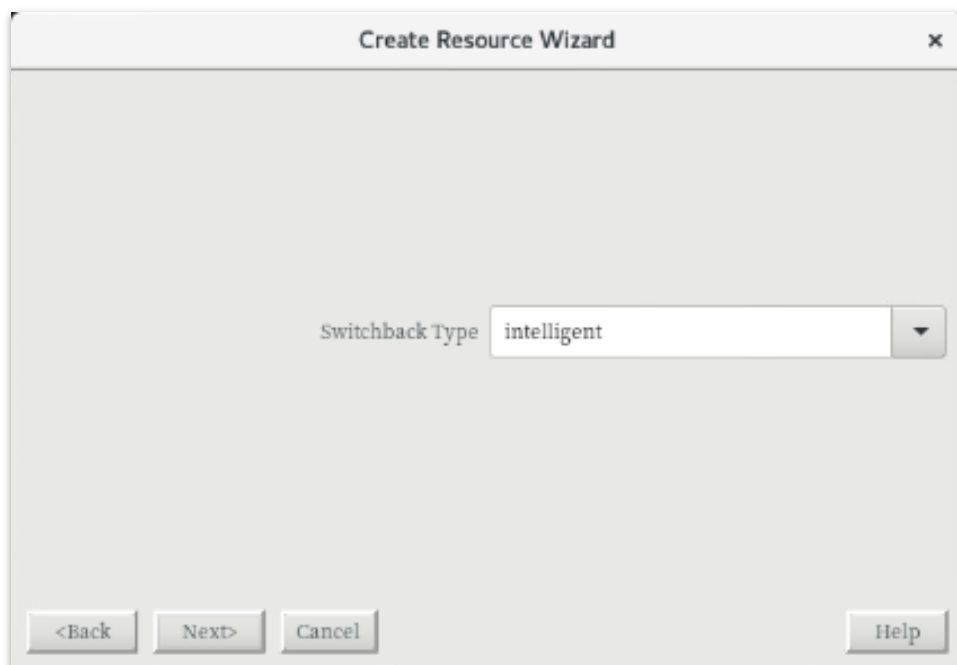
実際のウィザード表示は次の通りです。



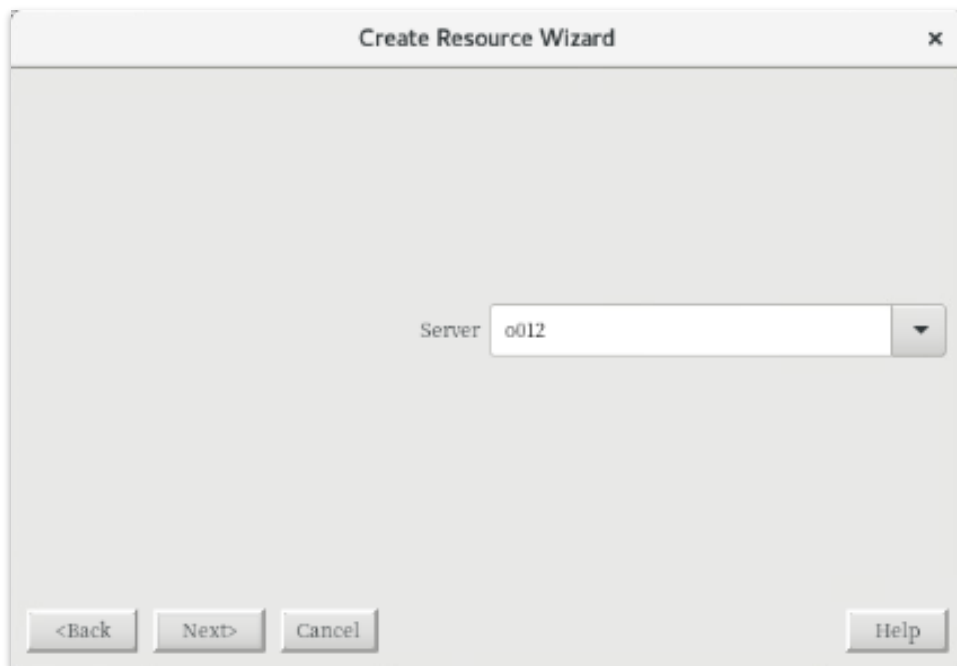
- ① [Please Select Recovery Kit]の[File System]を選択し、[Next]をクリック



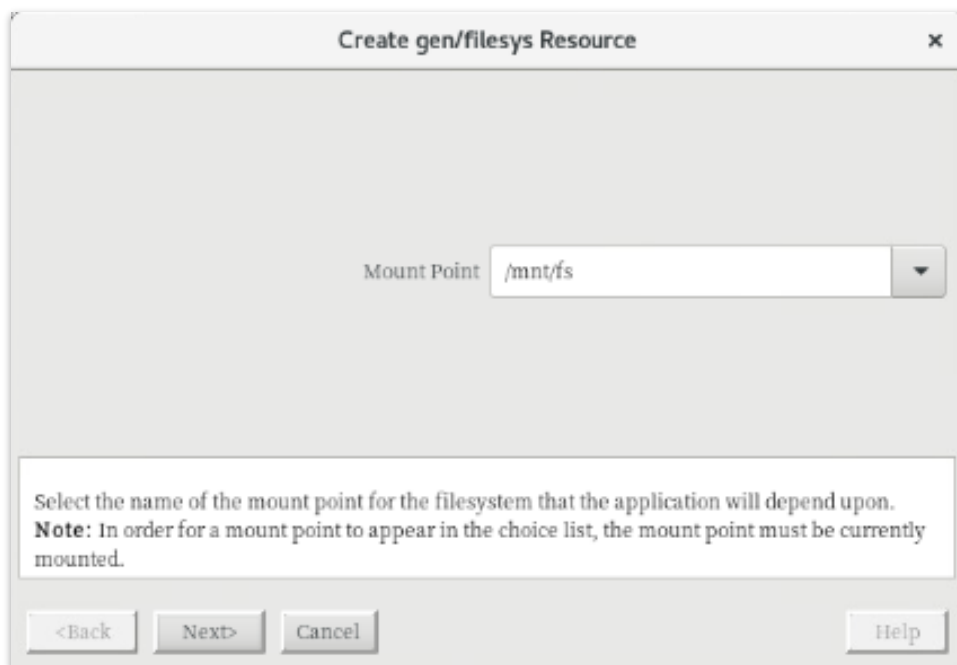
- ② [Switchback Type]の[intelligent]もしくは[automatic]を選択し、[Next]をクリック



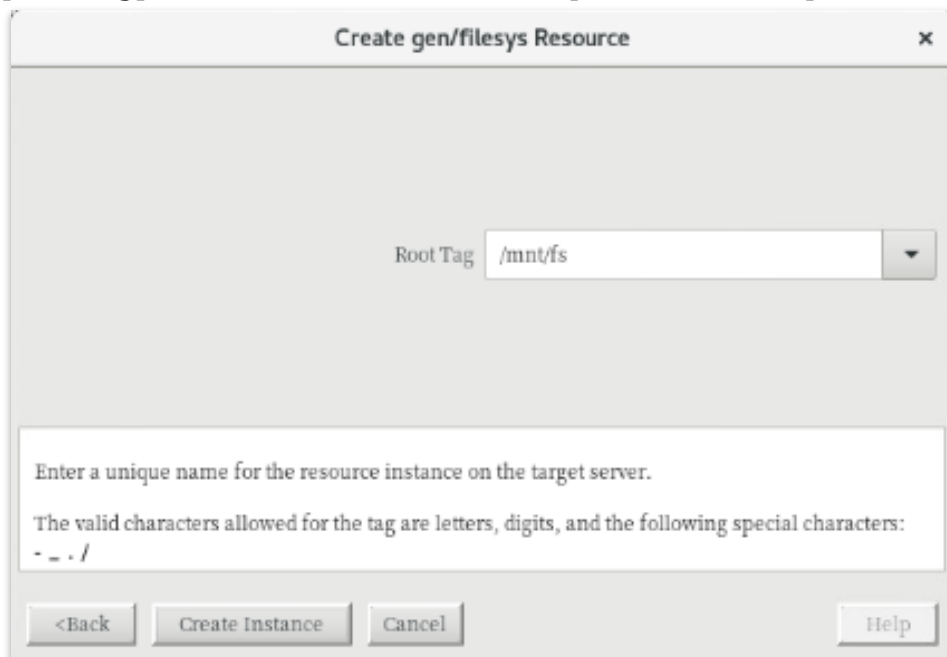
- ③ [Server]からプライマリーサーバー名を選択し、[Next]をクリック



- ④ [Mount Point]を選択し、[Next]をクリック



- ⑤ [Root Tag]のタグ名を選択もしくはは入力し、[Create Instance]をクリック

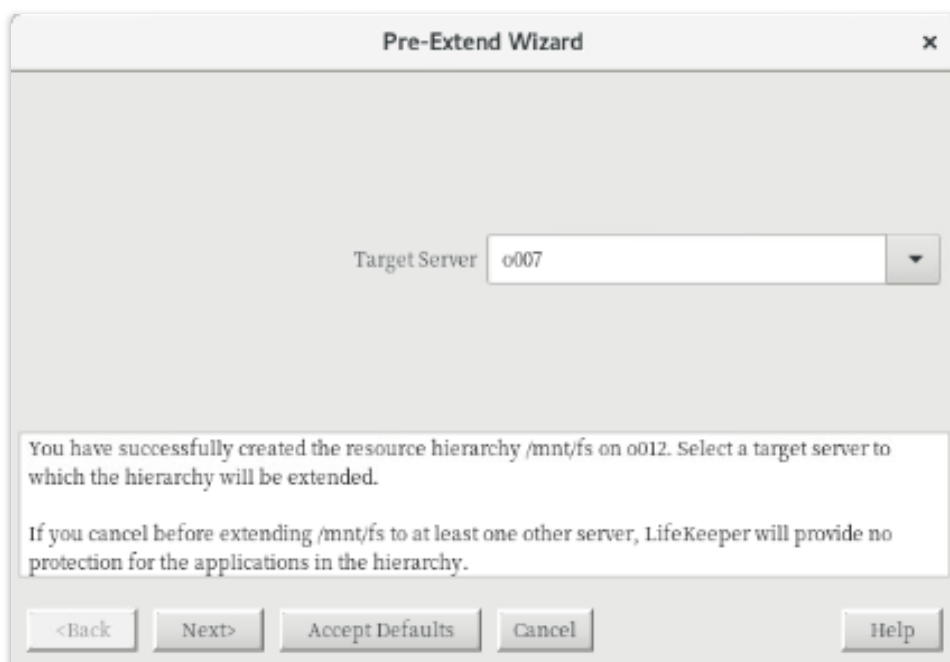


Root Tag /mnt/fs

Enter a unique name for the resource instance on the target server.  
The valid characters allowed for the tag are letters, digits, and the following special characters:  
- \_ . /

<Back Create Instance Cancel Help

- ⑥ [Target Server]からバックアップサーバーを選択し、[Next]をクリック

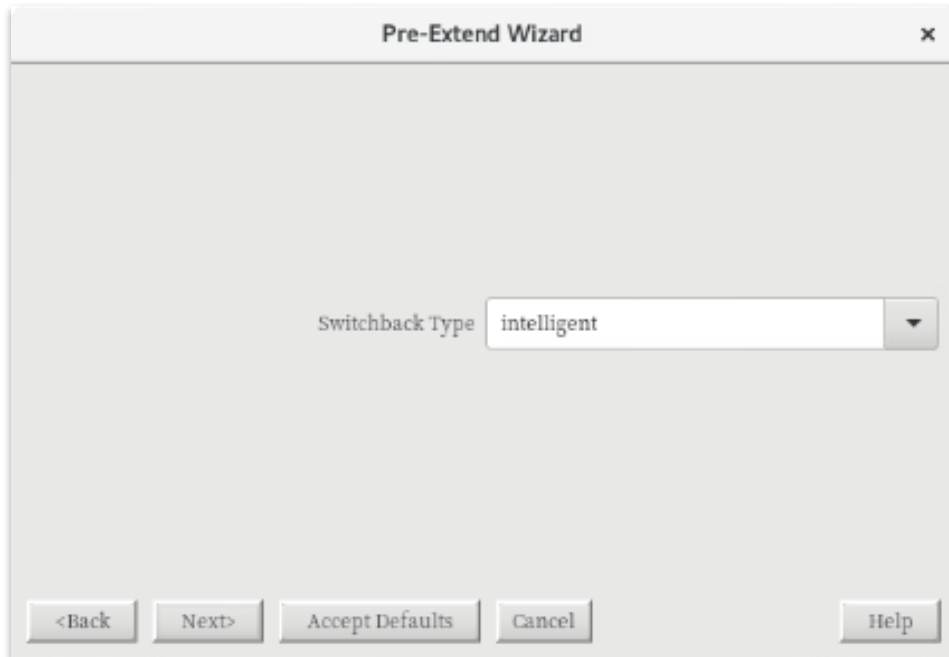


Target Server o007

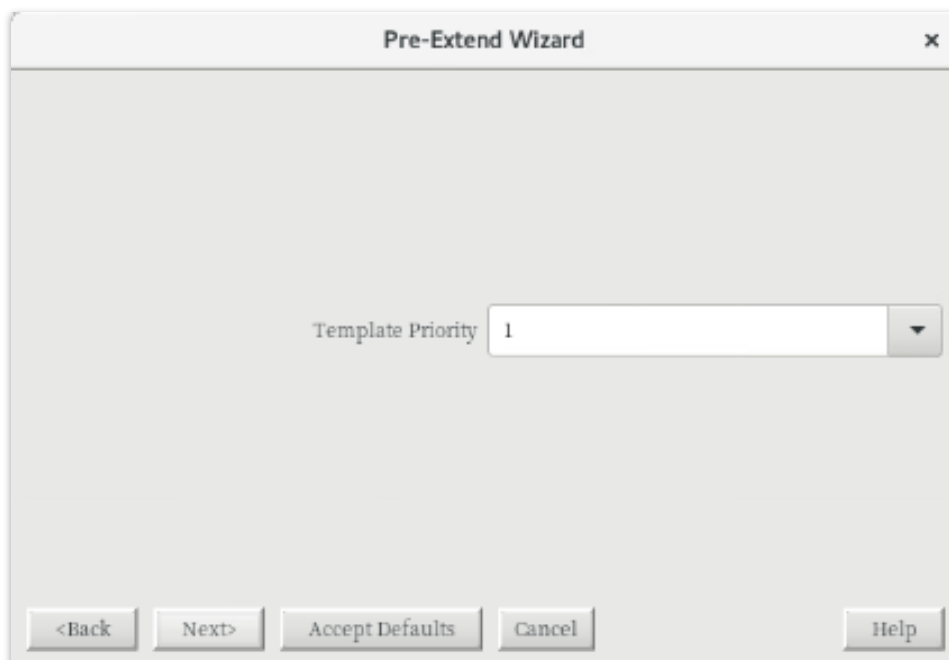
You have successfully created the resource hierarchy /mnt/fs on o012. Select a target server to which the hierarchy will be extended.  
If you cancel before extending /mnt/fs to at least one other server, LifeKeeper will provide no protection for the applications in the hierarchy.

<Back Next> Accept Defaults Cancel Help

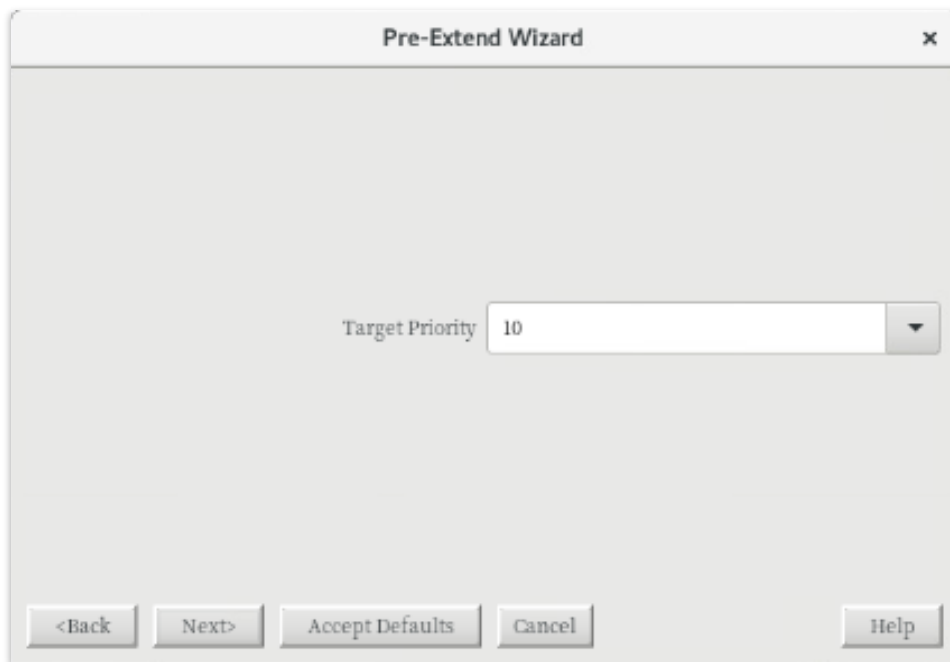
- ⑦ [Switchback Type]の[intelligent]もしくは[automatic]を選択し、[Next]をクリック



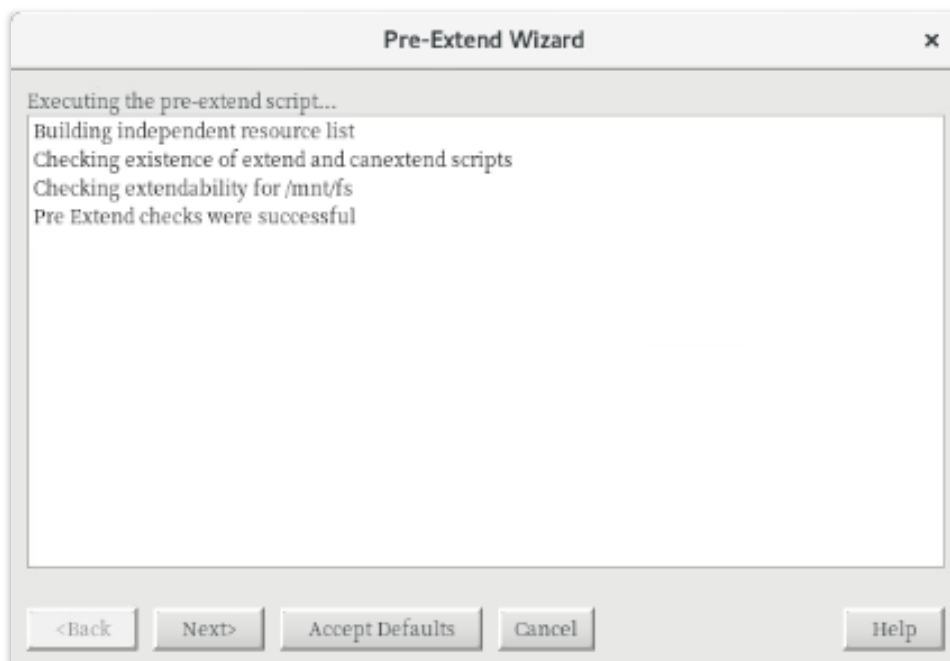
- ⑧ [Template Priority]の[1]を選択し、[Next]をクリック



- ⑨ [Target Priority]の[10]を選択し、[Next]をクリック



- ⑩ [Next]をクリック



- ⑪ [Mount Point]を確認し、[Next]をクリック

Extend gen/filesys Resource Hierarchy

Template Server: o012  
Tag to Extend: /mnt/fs  
Target Server: o007

Mount Point

Enter the name of the mount point for the filesystem that the application will depend upon on the target server.

**Note:** The mount point must be an absolute path that specifies a non-existent or empty, unmounted directory.

<Back Next> Accept Defaults Cancel Help

- ⑫ [Root Tag]のルートタグ名を選択もしくは入力し、[Next]をクリック

Extend gen/filesys Resource Hierarchy

Template Server: o012  
Tag to Extend: /mnt/fs  
Target Server: o007

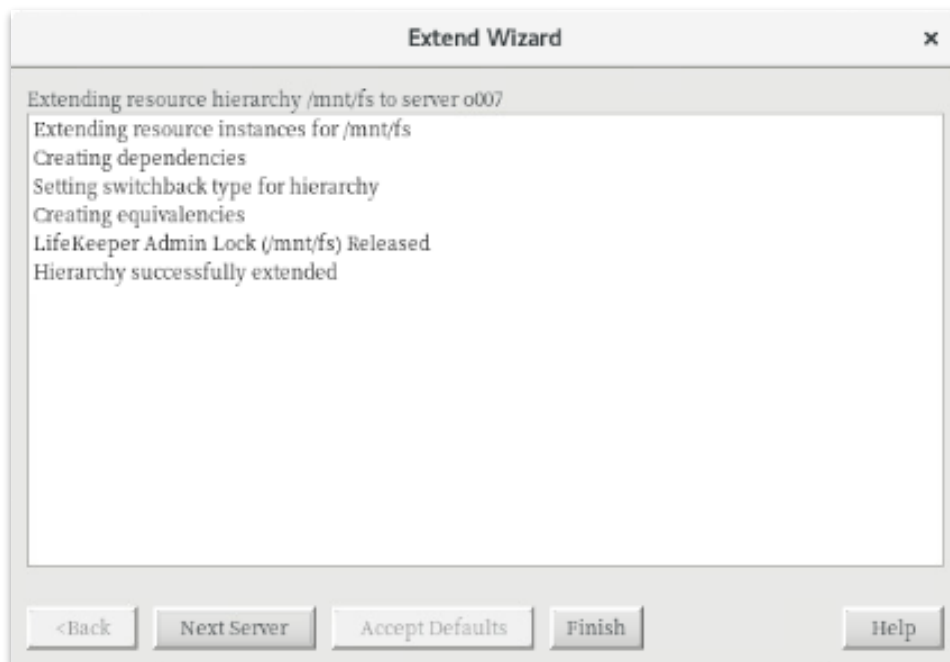
Root Tag

Enter a unique name for the resource instance on the target server.

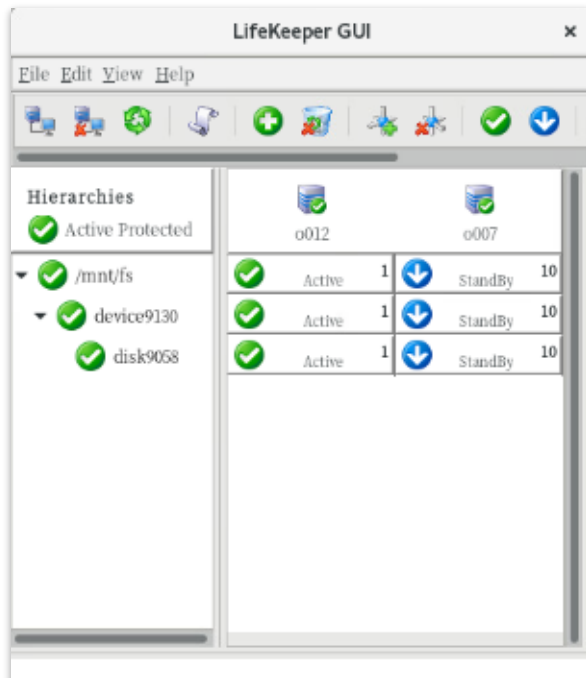
The valid characters allowed for the tag are letters, digits, and the following special characters:  
- \_ . /

<Back Next> Accept Defaults Cancel Help

## ⑬ [Finish]をクリック



ファイルシステムリソースの作成が完了すると、以下のようにリソースが表示されます。



ファイルシステムリソースを作成すると上記のように自動的に 3 つのリソースが依存関係を持った状態で作成されます。これらは、上位からマウントポイント、デバイス、ディスクを管理しており、それぞれ制御する対象が異なっています。これらは 3 つで 1 つの役割を担うため、依存関係の操作などで順序の変更などを行わないようにしてください。ファイルシステムリソースに関する詳細はユーザーポータル以下の URL などを参照してください。  
[Linux]FileSystemRecoveryKit の提供する機能（処理動作概要）を知りたいのですが  
<https://lkdkuserportal.sios.jp/hc/ja/articles/360037348492>

---

#### **注意**

共有ストレージの構成では複数サーバーからの同時アクセスを防ぐために SCSI コマンドを使用して LU 単位の排他制御を行いません。ファイルシステムリソースを作成すると、プライマリサーバーで共有ディスクの排他制御が行われます。バックアップサーバーから共有ディスクへアクセスを試みると、reservation conflict が/var/log/messages に出力されます。

NAS ストレージ構成や DataKeeper によるレプリケーション構成は SCSI コマンドによる排他制御は行なわないため reservation conflict は出力されません。

---

その他のリソース作成の手順につきましては、各 Recovery Kit のマニュアルをご参照ください。



## 6 その他

### 6.1 製品サポートへお問い合わせいただく際に収集すべき情報について

製品サポートにお問い合わせをいただく際は lksupport によるログを可能な限りご提供くださいますようお願いを致します。障害解析やインストレーションに関する調査をご希望の際は、以下の情報をご提供ください。

- 事象発生時刻
- 全クラスタノードの/var/log/messages
- 全クラスタノードの lksupport
- その他、お気づきの点

---

### 補足

lksupport でアーカイブファイルを生成するために以下のコマンドを実行します。

```
#/opt/LifeKeeper/bin/lksupport
```

lksupport の実行に成功すると/tmp 以下に次の命名規則のファイルが生成されます。

```
#/tmp/lksupport/<ホスト名>.lksupport.<タイムスタンプ>.tar.gz
```

※データレプリケーションの構成では nbd デバイスがロードされ、/dev/nbd\*が作成されます。lksupport 実行時や、lvdisplay、vgdisplay コマンド等、/dev/nbd\*を走査する処理が実行された場合は、以下のようなメッセージが出力されますが問題ではありません。

```
Nov  9 01:30:28 lk064 kernel: nbd0: Attempted send on closed socket
Nov  9 01:30:28 lk064 kernel: end_request: I/O error, dev nbd0, sector 0
Nov  9 01:30:28 lk064 kernel: nbd2: Attempted send on closed socket
Nov  9 01:30:28 lk064 kernel: end_request: I/O error, dev nbd2, sector 0
```

---

## 6.2 よく使用する LifeKeeper コマンドについて

コマンドの詳細につきましては、オンラインマニュアルをご参照ください。

### ■ LifeKeeper GUI クライアントの起動

```
# /opt/LifeKeeper/bin/lkGUIapp
```

### ■ LifeKeeper の起動

```
# /opt/LifeKeeper/bin/lkcli start
```

または

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

または

```
# systemctl start lifekeeper.service
```

### ■ LifeKeeper の停止（リソースも停止する）

```
# /opt/LifeKeeper/bin/lkcli stop
```

または

```
# /opt/LifeKeeper/bin/lkstop
```

または

```
# systemctl stop lifekeeper.service
```

### ■ LifeKeeper の停止（リソースは停止しない）

```
# /opt/LifeKeeper/bin/lkcli stop -f
```

または

```
# /opt/LifeKeeper/bin/lkstop -f
```

### ■ LifeKeeper のステータス確認

ステータスを簡易表示するには「-e」オプションを付与します。

```
# /opt/LifeKeeper/bin/lkcli status(もしくは lkcli status -e)
```

または

```
# /opt/LifeKeeper/bin/lcdstatus(もしくは lcdstatus -e)
```

### ■ LifeKeeper のログの確認

/var/log/lifekeeper.log を参照します。リアルタイムにログの出力を確認したい場合には以下のように tail コマンドを使用することもできます。

```
# tail -f /var/log/lifekeeper.log
```

### ■ LifeKeeper の構成情報やログの一括取得

```
# /opt/LifeKeeper/bin/lksupport
```

### ■ LifeKeeper の構成情報のバックアップ・リストア

LifeKeeper の構成情報のバックアップ

```
# /opt/LifeKeeper/bin/lkbackup -c
```

LifeKeeper の構成情報のリストア

```
# /opt/LifeKeeper/bin/lkbackup -x -f archive.<タイムスタンプ>.tar.gz
```

### 6.3 CUI によるリソースの起動、停止およびスイッチオーバ

CUI によるリソースの起動とスイッチオーバは、perform\_action コマンドの [-a] オプションを付与し引数に restore を与えます。

```
# /opt/LifeKeeper/bin/perform_action [-b] -t <タグ名> -a restore
```

---

#### 補足

[-b]オプションを付与した場合は、下位のリソースから指定したリソースまで起動します。  
[-b]オプションを付与しない場合、指定したリソースと依存関係がある全てのリソースが起動します。

---

CUI によるリソースの停止は、perform\_action コマンドの[-a]オプションを付与し引数に remove を与えます。

```
# /opt/LifeKeeper/bin/perform_action -t <タグ名> -a remove
```

---

#### 補足

リソースの停止を実行した場合、指定したリソースとそのリソースの上位に存在するリソースも停止します。

---

#### 6.4 GUI クライアントのステータス表示

- サーバー状態の情報



全てのコミュニケーションパスのステータスが ALIVE となっている。



一部のコミュニケーションパスのステータスが DEAD となっている。



全てのコミュニケーションパスのステータスが DEAD となっている。



サーバーのステータスが不明である。

- リソース状態の情報



Active - リソースの稼働状態



Standby - リソースの停止状態



Fail - リソースの障害状態



UNKNOWN - リソースの状態を確認できない状態

#### 6.5 LifeKeeper のアンインストール

LifeKeeper をアンインストールする場合は、rmlk コマンドを実行してください。

```
# rmlk
```

```
This script will uninstall LifeKeeper on your system.  
All in service resources will be taken out of service.  
This will stop any applications protected by LifeKeeper.
```

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

y を入力するとアンインストールが開始されます。

## 7 免責事項

- 本書に記載された情報は予告なしに変更、削除される場合があります。最新のものをご確認ください。
- 本書に記載された情報は、全て慎重に作成され、記載されていますが、本書をもって、その妥当性や正確性についていかなる種類の保証をするものではありません。
- 本書に含まれた誤りに起因して、本書の利用者に生じた損害については、サイオステクノロジー株式会社は一切の責任を負うものではありません。
- 第三者による本書の記載事項の変更、削除、ホームページ及び本書等に対する不正なアクセス、その他第三者の行為により本書の利用者に応じた一切の損害について、サイオステクノロジー株式会社は一切の責任を負うものではありません。
- システム障害などの原因によりメールフォームからのお問い合わせが届かず、または延着する場合がありますので、あらかじめご了承ください。お問い合わせの不着及び延着に関し、サイオステクノロジー株式会社は一切の責任を負うものではありません。

## 8 著作権

本書に記載されているコンテンツ(情報・資料・画像等種類を問わず)に関する知的財産権は、サイオステクノロジー株式会社に帰属します。その全部、一部を問わず、サイオステクノロジー株式会社の許可なく本書を複製、転用、公衆への送信、販売、翻案その他の二次利用をすることはいずれも禁止されます。またコンテンツの改変、削除についても一切認められません。本書では、製品名、ロゴなど、他社が保有する商標もしくは登録商標を使用しています。