
LifeKeeper for Linux 運用手順書

(対象サーバー名 : [host1.example.co.jp](#) / [host2.example.co.jp](#))

20XX 年 XX 月 XX 日

サイオステクノロジー株式会社

目次

1 はじめに.....	3
2 本書の見方	3
3 LifeKeeper 本体の操作	4
3-1 LifeKeeper をコマンドで操作する際のログインユーザーについて	4
3-2 LifeKeeper の起動と停止について	4
3-3 LifeKeeper の起動	4
3-4 LifeKeeper の停止	5
3-5 ホスト名、もしくは IP アドレス変更時の LifeKeeper 設定変更.....	6
3-6 ローカルリカバリを無効化する	8
3-7 LifeKeeper Web GUI に接続.....	9
3-8 LifeKeeper Web GUI へのログイン	10
3-9 GUI 管理画面によるステータス表示	11
3-10 コマンドラインによるステータス表示	12
3-11 ステータス表示と意味(リソース、サーバーの状態)	13
4 リソースの起動と停止	15
4-1 GUI からのリソース起動 (スイッチオーバー)	15
4-2 GUI からのリソース停止	17
4-3 CUI からのリソース起動(スイッチオーバー)	18
4-4 CUI からのリソース停止	19
5 LifeKeeper の情報確認	21
5-1 LifeKeeper のログの確認.....	21
5-2 LifeKeeper のクラスタシステム情報の取得.....	23
6 参考情報.....	24

1 はじめに

本書は、本システムを対象に LifeKeeper の運用に必要な基本的な操作を説明するものです。前提となる LifeKeeper の基礎や動作等について解説するものではありません。

2 本書の見方

◇ 本文について

italic や**太字**、**赤字**は強調などに使用しています。

◇ ターミナル・コンソール上の表記について

コマンドプロンプト (スーパーユーザ)	#
(一般ユーザー)	\$
コマンド入力および値入力	黒太文字
コメント	青太文字
注意部分	赤太文字

◇ GUI 上の操作について

文章中で[]に囲まれている語は、ユーザーが操作する GUI 上のボタン表記です。

◇ クラスタシステムにおける表記について

本システムは **host1.example.co.jp/host2.example.co.jp** の Active/Standby で構成されています。本資料では、以下のように表記しています。

稼動系 (アクティブサーバー)： リソースステータスが Active であり、サービスが稼動しているサーバーです。通常運用時では **host1.example.co.jp** となります。

待機系 (スタンバイサーバー)： リソースステータスが Standby であり、サービスが稼動しているサーバーの障害発生時に切り替わり先となるサーバーです。通常運用時は **host2.example.co.jp** となります。

リソース： LifeKeeper により保護対象となっているオブジェクトを指します。本システムでは、**IP リソース**、**ファイルシステムリソース(NAS)**、**MySQL リソース**があります。

◇ コマンドの出力例について

コマンド実行時の出力例の日時やファイル名等はお客様の環境と完全に一致するものではありません。例としてご参照ください。

3 LifeKeeper 本体の操作

3-1 LifeKeeper をコマンドで操作する際のログインユーザーについて

LifeKeeper を管理・操作する際には、LifeKeeper GUI を使用するか、LifeKeeper の CUI(コマンドライン)を使用して操作する方法があります。LifeKeeper GUI には LifeKeeper 用のユーザーグループが作成され、操作できる内容がユーザーグループによって異なっています。GUI 管理ユーザーの詳細については以下のオンラインマニュアルの内容を確認してください。

GUI ユーザーの設定

http://jpdocs.us.sios.com/Linux/9.0.2/LK4L/TechDoc/index.htm#user_guide/using_lifecycle_keeper_linux/gui/preparing_run_gui/configuring_gui_users.htm

LifeKeeper をコマンドで操作する場合には、OS の root ユーザーで操作する必要があります。そのため、後述のコマンド実行例では全て root で実行している例を表しています。

3-2 LifeKeeper の起動と停止について

通常クラスタシステムを運用する場合、クラスタの機能が利用できるよう LifeKeeper を起動した状態で利用し、システム再起動後も自動的に起動するよう設定されています。ただし、LifeKeeper の設定変更を行なう場合や、一時的にクラスタ機能を停止したい場合は、以下の方法で LifeKeeper の停止/起動を手動で行うことが可能です。

3-3 LifeKeeper の起動

LifeKeeper の本体プロセスの起動は、以下の何れかの方法で行います。

```
# lkstart
# /etc/init.d/lifekeeper start
```

以下は lkstart コマンドコマンド実行時の出力例です。

```
# lkstart
Starting LifeKeeper... [ OK ]
```

以下は/etc/init.d/lifekeeper start コマンドコマンド実行時の出力例です。

```
# /etc/init.d/lifekeeper start
Starting LifeKeeper... [ OK ]
```

起動コマンドは直ぐに戻ってきますが、実際には LifeKeeper のイニシャライズ処理が行われています。そのため、lkstart コマンドを実行してから保護対象サービスが起動するまでには、時間差があります。

LifeKeeperのプロセスが正しく起動されているかを確認するため lktest コマンドを実行します。以下はコマンド実行時の出力例です。

```
# lktest ; echo $?
```

```
F  S UID      PID  PPID  C  CLS PRI  NI SZ   STIME   TIME  CMD
4 S root    22974 1     0   TS   39 -20 2214 May16  00:04:48 lcm
4 S root    22975 1     0   TS   39 -20 2497 May16  00:00:00 ttymonlcm
4 S root    22976 1     0   TS   34 -10 3048 May16  00:00:25 lcd
0
```

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出力結果と戻り値を返します。

lkstart 実行直後に実行した場合には、イニシャライズ処理の関係上プロセスは一時的に複数出力されます。起動処理が完了し、状態が収束すると上記のように 3 つのプロセスが表示される状態となります。また、通常運用時にも複数のプロセスが表示される場合がありますが、異常ではありません。正常動作が確認できた場合は、戻り値が 0 となります。

3-4 LifeKeeper の停止

LifeKeeper の本体プロセスの起動は、以下の何れかの方法で行います。

```
# /etc/init.d/lifekeeper stop
```

LifeKeeper のリソースを停止しプロセスを停止する。お客様環境では“Shutdown Failover”が設定されている為、スタンバイノードにリソースが切り替わる。

```
# /etc/init.d/lifekeeper stop-nofailover
```

LifeKeeper のリソースを停止しプロセスを停止する。nofailover フラグを作成しリソースの切り替えを抑制する。

```
# /etc/init.d/lifekeeper stop-daemons
```

LifeKeeper のプロセスを停止する。ノードでリソースが起動している場合は、リソースの停

止を行わない。

```
# /etc/init.d/lifekeeper stop-failover
```

LifeKeeper のリソースを停止しプロセスを停止する。また強制的にリソースの切り替えをスタンバイノードで起動する。

```
# lkstop
```

lkstop コマンドは、以下の様にコマンドを実行して LifeKeeper の停止と LifeKeeper の自動起動の無効化を行う。

```
# chkconfig lifekeeper off; /etc/init.d/lifekeeper stop-nofailover
```

* lkstop コマンドは以下のようにオプションを利用する事で挙動を変えることが出来ます。

-f : chkconfig lifekeeper off; /etc/init.d/lifekeeper stop-daemons で停止する。

-n : chkconfig lifekeeper off; /etc/init.d/lifekeeper stop-failover で停止する。

-r : chkconfig lifekeeper off を実行しない。

-r -f : /etc/init.d/lifekeeper stop-daemons で停止する。

3-5 ホスト名、もしくはIPアドレス変更時のLifeKeeper 設定変更

LifeKeeper インストールしリソース設定後にシステムのホスト名もしくはIPアドレスを変更する場合、lk_chg_value コマンドを使用して LifeKeeper の設定情報を書き換える事が可能です。以下はホスト名を変更する手順となっていますが、IP アドレスの変更を行なう場合は、ホスト名を IP アドレスに置き換えて実施してください。

また LifeKeeper はノード間で情報を共有する構成をとりますので、host1.example.co.jp のホスト名のみ変更する場合でも、lk_chg_value コマンドは必ず両ノードで同じコマンドを実行してください。

① 両ノードの LifeKeeper の構成情報をバックアップします。

以下のコマンドでバックアップファイル名を/tmp/lkbackup_archive として保存します。構成情報のバックアップは両ノードで実施します。手順のミスなどで起動が行えなくなった場合、バックアップファイルを使用して元の状態に戻すことが出来ます。

```
# lkbackup -c -f /tmp/lkbackup_archive
```

- ② 両ノードの LifeKeeper を停止します。

LifeKeeper のプロセス、リソースの両方を停止してください。

- ③ OS 上でのホスト名を変更します。

- ④ LifeKeeper で保持しているホスト名の情報を変更します。

lk_chg_value コマンドで、LifeKeeper が持つホスト名の情報を更新します。このコマンドは両ノードで行ってください。

```
# /opt/LifeKeeper/bin/lk_chg_value -o <変更前ホスト名> -n <変更後ホスト名>
```

- ⑤ 両ノードの LifeKeeper を起動します。

LifeKeeper の起動に失敗した場合は、手順のミスが考えられます。以下の手順で元に戻してください。

- ① 両ノードの LifeKeeper を停止します。

LifeKeeper のプロセス、リソースの両方を停止してください。

- ② OS 上でのホスト名を元のホスト名に戻します。

- ③ 以下の lkbackup コマンドでバックアップファイルを使用して、LifeKeeper の設定を変更前に戻します。

```
# lkbackup -x -f /tmp/lkbackup_archive
```

- ④ 両ノードの LifeKeeper を起動します。

3-6 ローカルリカバリを無効化する

LifeKeeper はインストール直後から、デフォルトでローカルリカバリが有効になっています。障害発生時にローカルリカバリを行わず即座にフェイルオーバーする設定に変更したい場合等では、以下の様に `lkpolicy` コマンドを使用して、ローカルリカバリをコントロールしてください。

- <リソースタグ名>リソースのローカルリカバリを無効化にする。
`lkpolicy --set-policy LocalRecovery --off tag=<リソースタグ名>`
- <リソースタグ名>リソースのローカルリカバリを有効化にする。
`lkpolicy --set-policy LocalRecovery --on tag=<リソースタグ名>`
- 全てのリソースのローカルリカバリを無効化する。
`lkpolicy --set-policy LocalRecovery --off`
- 全てのリソースのローカルリカバリを有効化する。
`lkpolicy --set-policy LocalRecovery --on`
- 現在のポリシー設定を確認する。
`lkpolicy --get-policies`

* `lkpolicy` によるポリシー設定は、リソース毎に設定する方法と全体に設定する方法の 2 種類の方法があります。

* リソース毎に設定する場合は、紐づくリソースタグ名を割り当ててください。<リソースタグ名>は、`lcdstatus -e` コマンドで出力される **TAG 列**もしくは GUI 管理画面の左ウインドウ「**Hierarchies**」を指定してください。

* `device`(共有ディスク、NAS、DataKeeper)リソースは `lkpolicy` コマンドによるローカルリカバリのコントロールが行えません。

3-7 LifeKeeper GUI に接続

LifeKeeper GUI に接続する方法は、x window system で GUI を起動する方法と、ブラウザから接続する方法の 2 種類が用意されています。

x window system で GUI を起動する場合は、以下のコマンドを実行します。

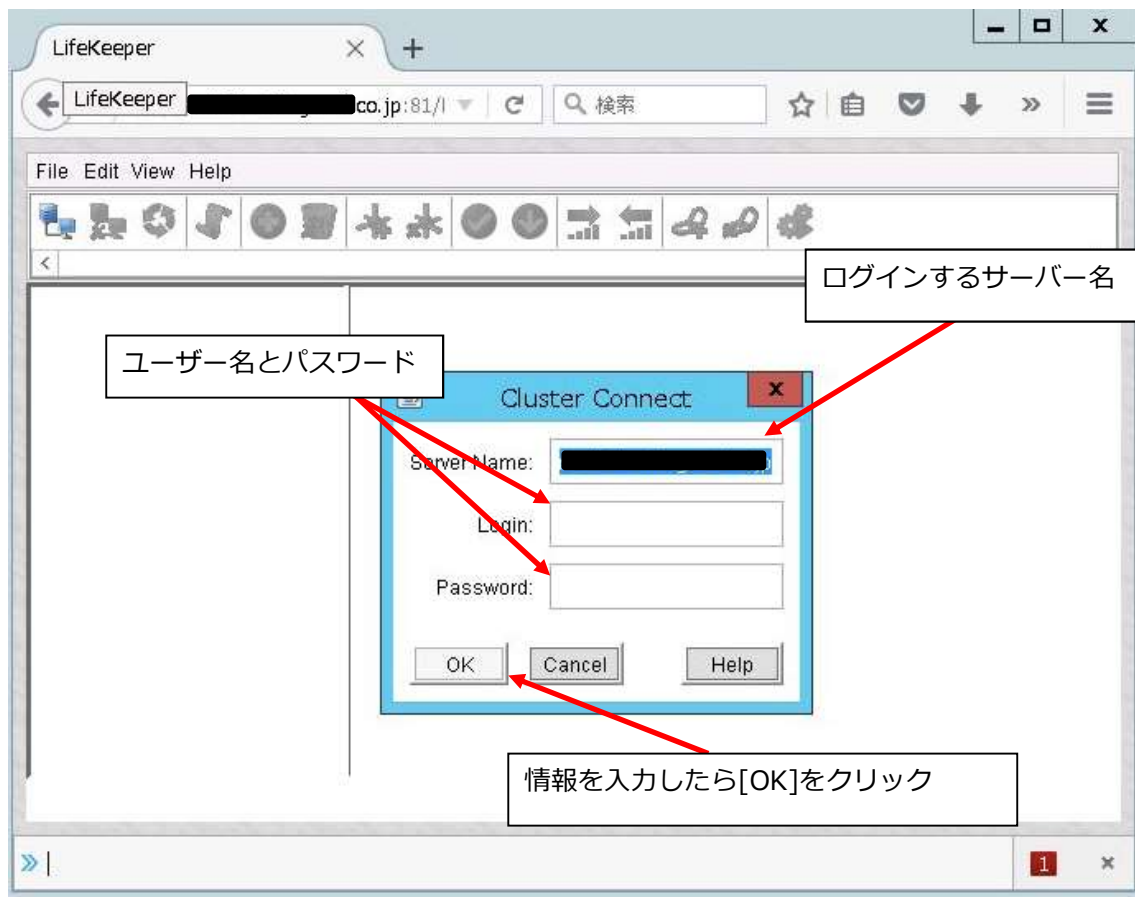
```
# lkGUIapp &
```

ブラウザから接続する場合は、Java プラグインをインストールされている環境のブラウザを使用して、以下のアドレスを入力して LifeKeeper Web GUI に接続します。

<http://host1.example.co.jp:81/>
(または <http://host2.example.co.jp:81/>)

3-8 LifeKeeper GUI へのログイン

ログインするには、以下のようにログインに必要な情報を入力してください。



3-9 GUI 管理画面によるステータス表示

ログインに成功すると以下のような表示となります。以下は通常運用時の状態でリソース階層を全て展開した状態です。

The screenshot displays the LifeKeeper management interface. The left pane, titled 'Hierarchies', shows a tree structure of resources: 'Active Protected' (root), 'mysql-56', '/db', 'device-nfs38678', and 'ip-172.17.0.170'. The right pane shows a table of resource status across two nodes, both labeled 'co.jp'. The table has columns for resource name, status, a numeric value, and a larger numeric value. The status for all resources is 'Active' on the left node and 'StandBy' on the right node. The numeric values are 1 and 10 respectively. A red dashed box highlights the top part of the table, and a red arrow points to it from a text box. Another red arrow points from a text box to the left pane. A third red arrow points from a text box to the table.

各クラスターノードの状態
コミュニケーションパスのステータス

左ペイン：全クラスターのリソースの状態。リソースの依存関係。

右ペイン：ノード毎のリソースの状態

Resource	Status	Value 1	Value 2
mysql-56	Active	1	10
/db	Active	1	10
device-nfs38678	Active	1	10
ip-172.17.0.170	Active	1	10

3-10 コマンドラインによるステータス表示

コマンドを使用してリソースやコミュニケーションパスのステータスを確認することができます。確認には **lcdstatus -q** コマンドを使用します。以下使用した際の出力例です。

```
# /opt/LifeKeeper/bin/lcdstatus -q
```

LOCAL	TAG	ID	STATE	PRIO	PRIMARY
host1.example.co.jp	mysql-56	mysql45088	ISP	1	host1.example.co.jp
host1.example.co.jp	/db	/db	ISP	1	host1.example.co.jp
host1.example.co.jp	device-nfs38678	172.XXX.XXX.40-colon-/vol/rsvdb01	ISP	1	host1.example.co.jp
host1.example.co.jp	ip-172.XXX.XXX.170	IP-172.XXX.XXX.170	ISP	1	host1.example.co.jp
MACHINE	NETWORK ADDRESSES/DEVICE		STATE	PRIO	
host2.example.co.jp	TCP	172.XXX.XXX.171/172.XXX.XXX.172	ALIVE	1	
host2.example.co.jp	TCP	172.XXX.XXX.38/172.XXX.XXX.39	ALIVE	2	

各出力の意味は以下の通りです。

➤ リソースの情報

LOCAL : コマンドを実行しているホスト名です。
TAG : 各リソースを識別する名前です。リソース作成時に定義されます。
ID : LifeKeeper が各リソースを識別するための名前です。
STATE : 自サーバーにおける各リソースの状態を表しています。
PRIO : そのリソースの自サーバー上のプライオリティを表示します。
PRIMARY : そのリソースのプライマリサーバー名です。

➤ コミュニケーションパス情報

MACHINE : ハートビートにおける対向サーバーのホスト名です。
NETWORK : ハートビートの種類です。TCP か TTY のいずれかです。
ADDRESS/DEVICE : ハートビートに使用している IP アドレスのリストです。
 “自サーバーの IP アドレス/相手サーバーの IP アドレス”の書式で表示されます。
STATE : 各ハートビートのステータスを表します。ALIVE(正常)か DEAD(異常)のいずれかです。
PRIO : 各ハートビートに定義された優先順位を表します。





lcdstatus コマンドは実行したサーバー自身のステータスを表示します。前述の例は LOCAL の欄が **host1.example.co.jp** となっていますので、**host1.example.co.jp** 上でコマンドを実行した例となります。

クラスタシステム内の他サーバーのステータスを確認するには、[-d]オプションを使用します。たとえば、**host1.example.co.jp** にログインした状態で **host2.example.co.jp** 上のステータスを確認する場合は以下のように実行します。

```
# /opt/LifeKeeper/bin/lcdstatus -q -d host2.example.co.jp
```

3-11 ステータス表示と意味(リソース、サーバーの状態)





➤ クラスタノードの状態

表示	解説
	全てのコミュニケーションパスのステータスが ALIVE です。(正常)
	一部のコミュニケーションパスのステータスが DEAD です。 コミュニケーションパスに使用しているネットワークの通信状態について確認してください。
	全てのコミュニケーションパスのステータスが DEAD です。 コミュニケーションパスに使用しているネットワークの通信状態、あるいは LifeKeeper 本体や GUI Server が起動しているかを確認してください。
	サーバーのステータスが不明です。LifeKeeper の起動中やネットワークの遅延などによって一時的にこの状態になることがあります。しばらく時間を置いてください。しばらくしても復旧しない場合には、LifeKeeper GUI クライアントや GUI Server の再起動を試みてください。

➤ リソースステータス (LifeKeeper GUI の右ペイン)

表示	解説
 Active	リソースは Active です。lcdstatus コマンドでは ISP (In-Service Protected) と表示されます。リソースは起動されており、保護対象サービスも起動した上体です。障害時に待機系に切り替わることが出来ます。通常運用時の場合の稼働系の表示です。
 Standby	リソースは Standby です。lcdstatus コマンドでは OSU (Out-of-Service Unimpaired)と表示されます。リソースは停止しており、対象サービスも停止した状態です。通常運用時では待機系の表示です。
 Failed	リソースは Failed です。lcdstatus コマンドでは OSF (Out-of-Service Failed) と表示されます。そのノードでは対象サービスは停止されています。直前に障害を検知したか、リソース停止や起動に失敗したなどの状態を表します。このステータスを確認した場合には、フェイルオーバーのトリガーや停止、起動の失敗原因などの調査が必要となる場合があります。
 Unknown	リソースは Unknown です。lcdstatus コマンドでは ILLSTATE と表示されます。LifeKeeper の起動中やネットワークの遅延などによって一時的にこの状態になることがあります。しばらく時間を置いてください。しばらくしても復旧しない場合には、LifekeeperGUI クライアントや GUI Server の再起動を試みてください。

➤ 全クラスターノードのリソースステータス (LifeKeeper GUI の左ペイン)

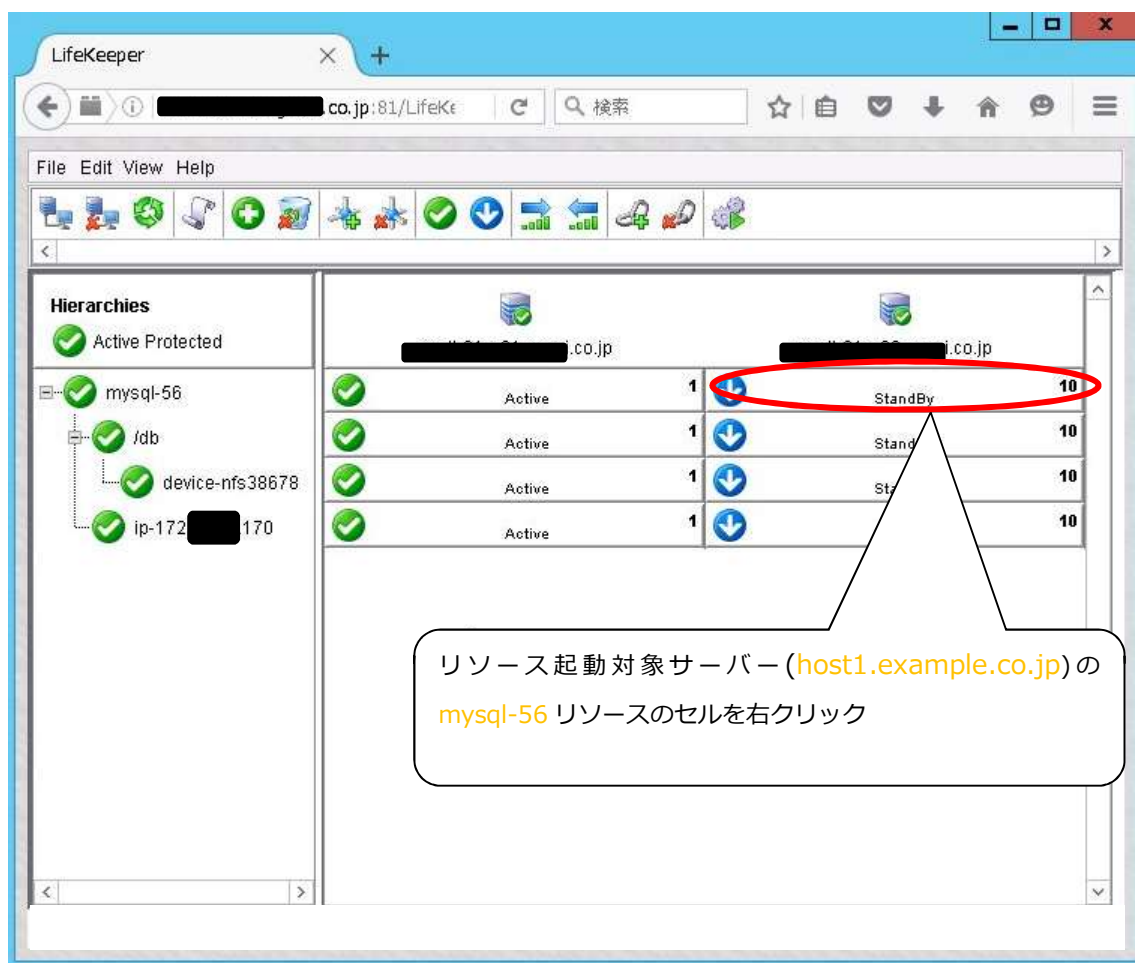
表示	解説
	全クラスターノードが正常に動作しており、いずれかのノードでリソースが Active な状態です。
	リソースは起動していますが、障害発生時に切り替わることが出来ません。待機系ノードが定義されていない可能性があります。
	どのノードでも起動していないリソースが存在し、サービスが提供できていません。
	リソースステータスが取得できていません。 しばらくしても復旧しない場合には、LifekeeperGUI クライアントや GUI Server の再起動を試みてください。

4 リソースの起動と停止

4-1 GUI からのリソース起動（スイッチオーバー）

GUI からリソースの起動やノードの切り替えを行うには、Standby ステータスにあるリソースを、起動させたいサーバー上で **[In Service]** を実行します。これは、リソース他サーバーへ手動で切り替える手順ともなります。以下は mysql-56 リソース以下の全リソースを host1.example.co.jp から host2.example.co.jp へ手動で切り替える際の操作例です。

- ① host2.example.co.jp の Standby ステータスの **mysql-56** リソースのアイコン上で右クリックをします。

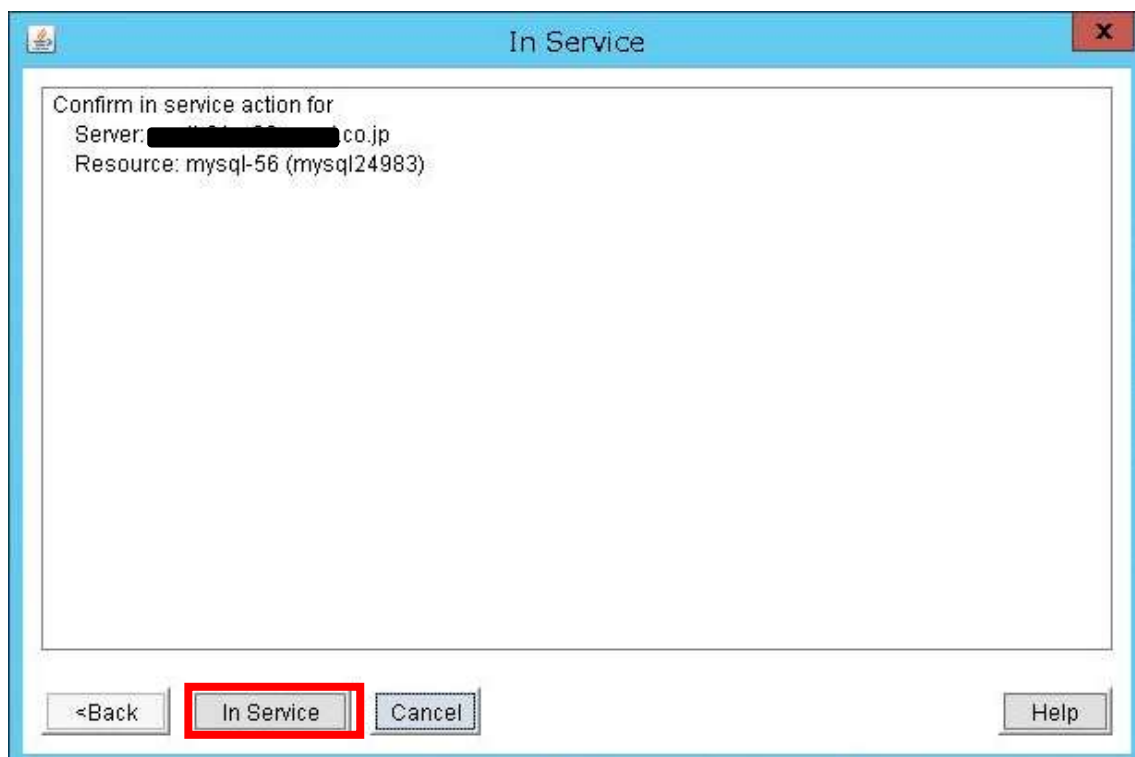


※**mysql-56** リソースは全てのリソース階層の最上位にあり、それ以下のリソースを束ねている状態です。そのため、すべてのリソース階層を切り替える、あるいは起動する場合には **mysql-56** リソースを指定します

- ② 右クリックをすると以下のようなコンテキストメニューが表示されます。



表示されたメニューの **[In-service]** をクリックしてください。その後、別途以下のような in-service の実行について確認するウインドウが開きます。



この画面の**[In-service]**ボタンを押してください。実際にリソース切り替えの処理が開始されます。

- ③ スタンバイノードでリソースの起動が完了したらウインドウ画面の**[Done]**ボタンをクリックして完了します。

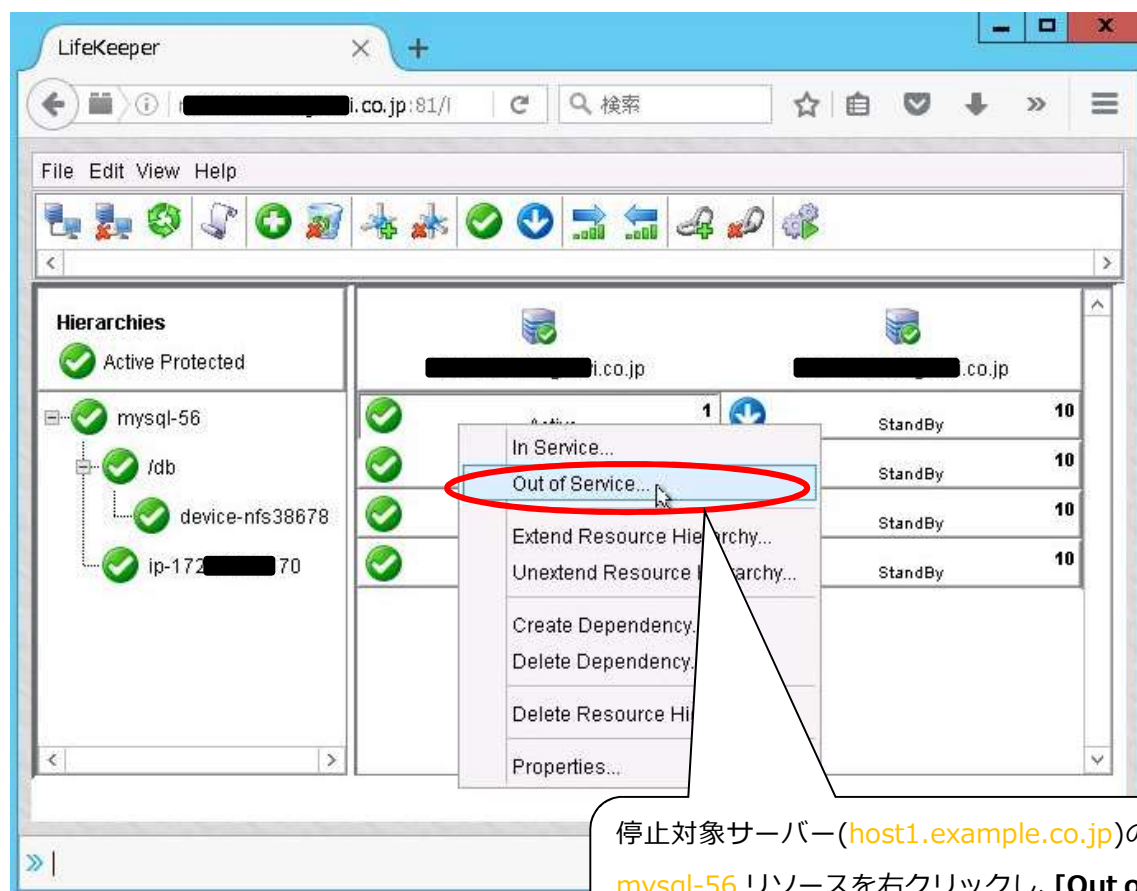
リソースのサービスノードを切り替えるのではなく、同一ノードで再度起動処理を実行したい場合には、起動させたいノード側のリソースアイコンをクリックして、上記①～③と同等の操作を行います。

4-2 GUI からのリソース停止

GUI からリソースの停止を行うには、停止対象リソースのセルを右クリックし、[**Out Of Service**]を実行します。実行後、そのリソースと依存関係を持っているリソース階層について、上位のリソースから順に停止されます。

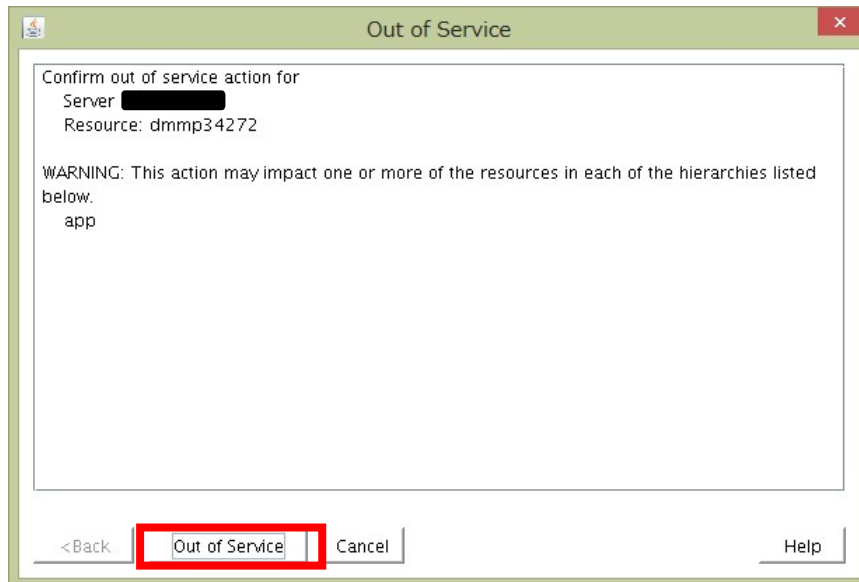
本環境の場合、**mysql-56** リソース以下に並列の関係となっているリソースがあるため、一度の操作でリソースを停止することができません。そのため、数回に分けて停止操作を行う必要があります。以下は、全リソース階層を停止する際の例です。

- ① **host1.example.co.jp** の Active ステータスの **mysql-56** リソースのアイコン上で右クリックをします。



停止対象サーバー(**host1.example.co.jp**)の **mysql-56** リソースを右クリックし、[**Out of Service**]を選択する。

- ② 別ウインドウで以下のような out-of-service の実行について確認するウインドウが開きます。



この画面の[**Out of Service**]ボタンを押してください。リソースの停止処理が開始されます。

Mysql-56 リソースが保護する mysql サービスが停止します。

- ③ 関連リソースの停止が完了したらウインドウ画面の[**Done**]ボタンをクリックして完了します。

*リソースを全て停止する場合は、①から③の手順を参考に ip-172.XXX.XXX.170 リソース、device-nfs38678 リソースを停止してください。リソースの停止は上位のリソースから順番に行われますので、全てのリソースが停止されます。

4-3 CUI からのリソース起動(スイッチオーバー)

CUI からリソースの起動には、perform_action コマンドを使用します。

```
# perform_action [-b] -t $TAG -a restore
```

リソースタグ名[\$TAG]は、lcdstatus -e コマンドで出力される **TAG 列**もしくは、GUI 管理画面の左ウインドウ「**Hierarchies**」で確認可能です。

[-b]オプションを付与しない場合、perform_action コマンドは、指定したリソースの上位と下位のリソースを全て起動します。 [-b]オプションを付与した場合は、指定したリソース及び下位のリソースを全て起動します。

たとえば、host1.example.co.jp で Active ステータスとなっている mysql-56 リソースの下位リソース全てを host2.example.co.jp へコマンドで切り替える場合には次のように実行します。

① host2.example.co.jp に root でログインしてください。

② 以下のようにコマンドを実行してください。

```
# perform_action -t mysql-56 -a restore
```

※ 最上位のリソース mysql-56 を指定しています。これによって全ての階層を切り替えることができます。

※ コマンドを実行すると、リソース切り替えの処理内容が画面上に出力されます。

③ lcdstatus -q コマンドを使用し、リソースステータスが host1.example.co.jp で OSU, host2.example.co.jp で ISP となっていることを確認してください。

4-4 CUI からのリソース停止

CUI からリソースの停止には、perform_action コマンドの [-a] オプションで remove を指定します。

```
# perform_action -t $TAG -a remove
```

たとえば、host1.example.co.jp で Active ステータスとなっている mysql-56 のサービスを停止する場合には次のように実行します。

① host1.example.co.jp に root でログインしてください。

② mysql サービスを保護する mysql-56 リソースを停止します。以下のようにコマンドを実行してください。

```
# perform_action -t mysql-56 -a remove
```

③ `host1.example.co.jp` で `mysql` サービスが停止している事を確認してください。

最上位のリソースから、`mysql-56` で指定したリソースまでが停止します。それより下位のリソースは停止しません。

5 LifeKeeper の情報確認

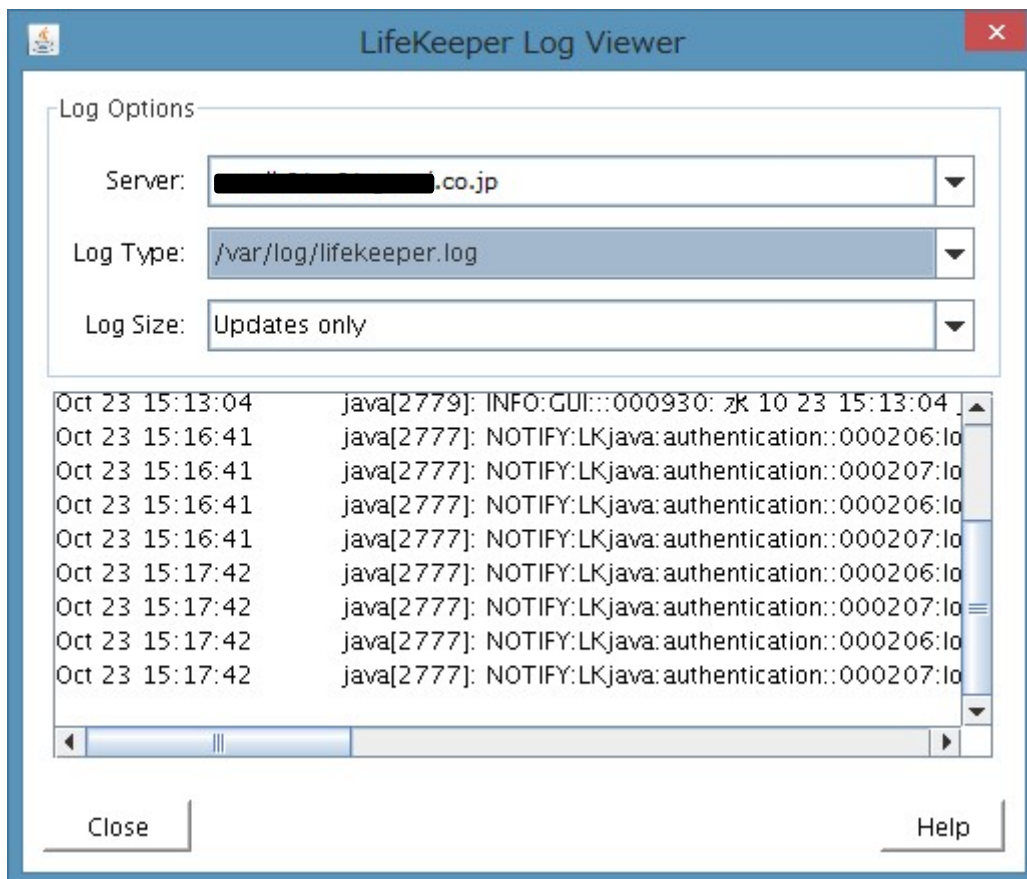
5-1 LifeKeeper のログの確認

➤ GUI を使用したログの確認

GUI 上から LifeKeeper のログを確認する場合 lkGUIapp コマンドで LifeKeeperGUI を起動後、以下のアイコンをクリックしビューアを起動します。LifeKeeper の GUI 管理画面上からの操作を行う場合、その操作毎に専用のログウィンドウが起動されますので、このビューアを明示的に起動する必要はありません。



[View Logs]アイコンをクリックします。すると log viewer 画面が起動します。以下はその表示例です。



GUI のログビューアの選択可能な項目と意味は以下の通りです。

項目	意味
Server	サーバー(host1.example.co.jp か host2.example.co.jp)を選択します。
Log Type	/var/log/lifekeeper.log が表示されます。選択することはできません。
Log Size	Updates only = リアルタイムで表示します。 Last [100 500 1000] lines = 最終ログから 100,500,1000 行を表示します。

➤ コマンドでのログ確認

LifeKeeper for Linux の詳細ログ（LifeKeeper の起動、停止、リソースの障害検知時の LifeKeeper の動作に関するログ）は **/var/log/lifekeeper.log** に出力されます。コマンドを使用して LifeKeeper のログを確認する場合には、/var/log/lifekeeper.log ファイルを開いて確認してください。

LifeKeeper のログフォーマットは以下のようになっています。

Date: Hostname: Process[PID]: Level: Source: Action: Resource Tag: Message-ID: Message

Date :syslog()が提供
Hostname :syslog()が提供
Process[PID] :プロセス名
Level :EMERG / NOTIFY / FATAL / ERROR / WARN / TRACE / INFO / DEBUG
Source :module.routine name(*)
Action :create, delete, extend, depend, restore, remove, quickCheck, etc.(*)
Resource Tag :リソースタグ名(*)
Message-ID :Message ID
Message :Message(*)
(*)には該当するものがない場合は、空になります。

※ パッケージをインストールする時、syslog はすべての LifeKeeper ログ・メッセージのファシリティに「local6」を使うように構成されます。

※ LifeKeeper のエラーログは、以下の URL にございます messages Catalog を参照してください。
<http://jpdocs.us.sios.com/Linux/9.0.2/LK4L/MessageCatalogs/MessageCatalog/>

5-2 LifeKeeper のクラスタシステム情報の取得

サポートに問い合わせを行う際に、ログの送付が必要となる場合があります。LifeKeeper の調査のために必要な情報は lksupport コマンドで一括収集することができます。実行例は以下です。

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

上記のコマンドを実行すると実行したノードの情報を収集し、**/tmp/lksupport/以下に「host 名.lksupportYYMMDDHHMM.tar.gz」**というアーカイブを作成します。これを、関連する全クラスターノードについて取得してください。実行に際して LifeKeeper を停止する必要はありません。

お問い合わせ窓口につきましては、ご契約のお問い合わせ受付窓口をご確認ください。

また、お問い合わせの際には、両ノードの lksupport の他に以下の情報を一緒に収集し、可能な範囲で提供することによってサポート担当が調査を行いやすくなります。

- ・ 事象が発生した日時
- ・ 事象が発生した日時のログを含む両ノードの/var/log/messages
- ・ 事象が発生した前後のオペレーションやシステム上の特殊な状況等の情報

6 参考情報

LifeKeeper/DataKeeper ユーザサイト

<http://lk.sios.com/>

製品に関する FAQ や最新の Errata 情報などを確認することができます。

LifeKeeper 製品マニュアル

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

全マニュアルの URL になります。LifeKeeper の利用に関する詳細については、お客様がご利用の LifeKeeper バージョンの「テクニカルドキュメンテーション」をご利用ください。

また、LifeKeeper のコマンドには man が付属しています。コマンドの詳細を確認するためには man コマンドを利用することも可能です。

LifeKeeper 製品情報

<http://www.sios.com/products/bcp/lkdk/>

LifeKeeper をはじめとした弊社の事業継続ソリューションの紹介、導入事例、セミナースケジュール、製品サポート等の情報を確認することができます。