

日本HP OpenSource/Linux技術文書

LifeKeeper for Linux v7.4  
インストールレーションガイド  
Red Hat Enterprise Linux 6.1  
+ HP P2000 G3 MSA FCマルチパス編  
(第1版)

日本ヒューレット・パッカード株式会社  
2011年12月15日

## 目次

[本ドキュメントについて].....	4
1. 環境 .....	5
2. 設定手順 .....	6
2-1. ノード1 とノード2 に Red Hat Enterprise Linux 6.1 をインストール.....	6
2-2. P2000 G3 MSA FC ストレージ設定 .....	6
2-3. PSP (ProLiant Support Pack) の適用.....	8
2-4. Device Mapper Multipath のインストール・設定 .....	8
2-5. ネットワークの確立.....	12
2-6. /etc/fstab ファイルの編集.....	12
2-7. LifeKeeper Installation Support CD を使った設定 .....	12
2-8. License Key のインストール .....	13
2-9. LifeKeeper for Linux v7.4 のインストール .....	13
2-10. LifeKeeper の起動 .....	14
2-11. LifeKeeper GUI の起動.....	14
2-12. クラスター設定、Volume (P2000 G3 MSA FC) リソースの設定.....	15
3. 注意事項 .....	15

## 図表目次

図 1. サインイン画面 .....	7
図 2. 仮想ディスク作成画面 .....	7

[本ドキュメントについて]

- 本ドキュメントの内容については充分チェックをしておりますが、その正確性を保証するものではありません。また、将来、予告なしに変更することがあります。
- 本ドキュメントの使用で生じるいかなる結果も利用者の責任となります。日本ヒューレット・パカード株式会社は、本ドキュメントの内容に一切の責任を負いません。
- 本ドキュメントの技術情報は、ハードウェア構成、OS、アプリケーションなど使用環境により大幅に数値が変化する場合がありますので、十分なテストを個別に実施されることを強くお勧め致します。
- 本ドキュメント内で表示・記載されている会社名・サービス名・商品名等は各社の商標又は登録商標です。
- 本ドキュメントで提供する資料は、日本の著作権法、条約及び他国の著作権法にいう著作権により保護されています。

本ドキュメントは、共有FCストレージにマルチパス構成のP2000 G3 MSA FCを使用したProLiantサーバーにRed Hat Enterprise Linux 6.1とLifeKeeper for Linux v7.4をインストールするためのガイドです。

注) 当資料が対象とするバージョン以外のOS、Device Mapper Multipath、LifeKeeper等をご使用の場合は、インストールや設定の手順が異なる場合があります。その場合は、ご使用のバージョンのマニュアルやRelease Notes等に記載された手順に従ってください。

## 1.環境

実際に使用した環境は、以下になります。

### H/W環境

サーバー : HP ProLiant DL360 G7

CPU : Xeon X5675 3.06GHz

メモリ : 48GB

RAIDコントローラ : SmartアレイP410iコントローラ

ファイバチャネルアダプタ : HP 82Q PCIe FC HBA Dual Port (AJ764A)

ストレージシステム : HP P2000 G3 MSA FC

ファイバチャネル集線装置 : HP SAN スイッチ 8/24 Base

### S/W環境

O/S : Red Hat Enterprise Linux 6.1 kernel-2.6.32-131.0.15.el6.x86\_64

クラスターソフトウェア : LifeKeeper for Linux v7.4

FCデバイスドライバ : qla2xxxドライバ v 8.03.07.03.06.1-k

トポロジー : Fabric接続

なお、今回は2台のサーバーでクラスターを構成しましたが、各ノードとも、環境は同一です。以下、2台のノードをノード1、ノード2とします。

## 2.設定手順

実際の手順の流れは以下になります。

- 1) ノード1とノード2にRed Hat Enterprise Linux 6.1をインストール
- 2) P2000 G3 MSA FCストレージ設定
- 3) PSP (ProLiant Support Pack) の適用
- 4) Device Mapper Multipathのインストール・設定
- 5) ネットワークの確立
- 6) /etc/fstabファイルの編集
- 7) LifeKeeper Installation Support CDを使った設定
- 8) License Keyのインストール
- 9) LifeKeeper for Linux v7.4のインストール
- 10) LifeKeeperの起動
- 11) LifeKeeper GUIの起動
- 12) クラスタ設定、Volume (P2000 G3 MSA FC) リソースの設定

各項目の具体的な作業内容を以降に記述します。

### 2-1.ノード1とノード2にRed Hat Enterprise Linux 6.1をインストール

ノード1とノード2にRed Hat Enterprise Linux 6.1をインストールしてください。この時、ストレージとの接続およびLifeKeeperに必要な以下のパッケージをインストールしてください。

```
device-mapper-multipath-0.4.9-41.el6.x86_64.rpm  
device-mapper-multipath-libs-0.4.9-41.el6.x86_64.rpm  
compat-libstdc++-296-2.96-144.el6.i686.rpm  
compat-libstdc++-33-3.2.3-69.el6.i686.rpm  
nss-softokn-freebl-3.12.9-3.el6.i686.rpm  
libgcc-4.4.5-6.el6.i686.rpm  
libXau-1.0.5-1.el6.i686.rpm  
libxcb-1.5-1.el6.i686.rpm  
libX11-1.3-2.el6.i686.rpm  
libXext-1.1-3.el6.i686.rpm  
libXi-1.3-3.el6.i686.rpm  
libXtst-1.0.99.2-3.el6.i686.rpm  
sg3_utils-1.28-3.el6.x86_64.rpm
```

### 2-2.P2000 G3 MSA FC ストレージ設定

Storage Management Utility (以下SMU) を使って、P2000 G3 MSA FCストレージ設定を行います。詳細については、「HP P2000 G3 MSAシステム SMU リファレンスガイド」を参照して下さい。今回は、以下の手順で行いました。

A) SMUにWebブラウザ経由でサインインします。

各コントローラには、工場出荷時、次のIP設定が割り当てられています。

- コントローラAのIPアドレス : 10.0.0.2
- コントローラBのIPアドレス : 10.0.0.3
- IPサブネットマスク : 255.255.255.0
- ゲートウェイIPアドレス : 10.0.0.1

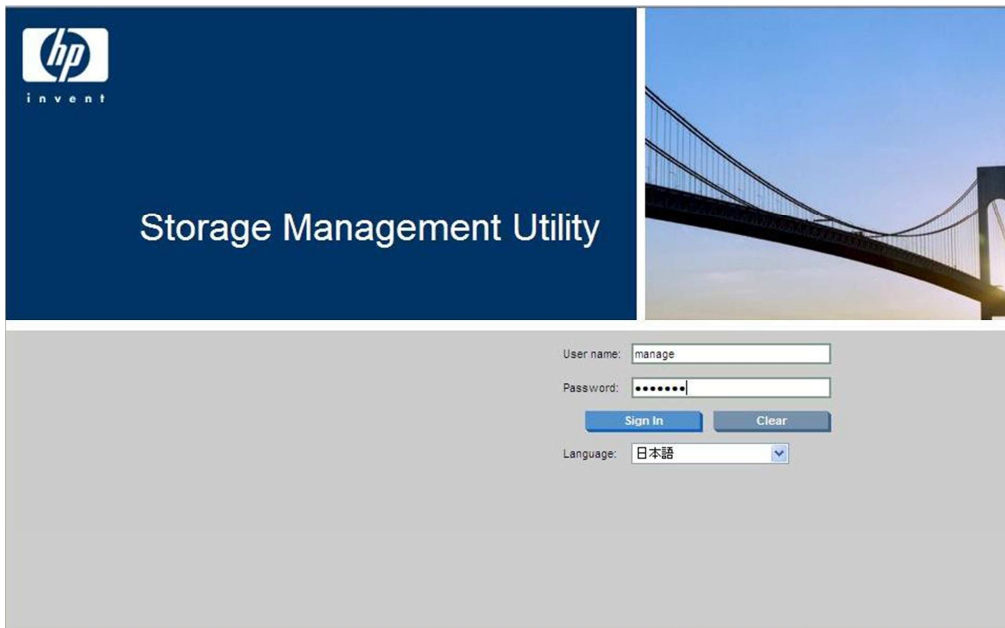


図 1. サインイン画面

B) 仮想ディスクを作成します。

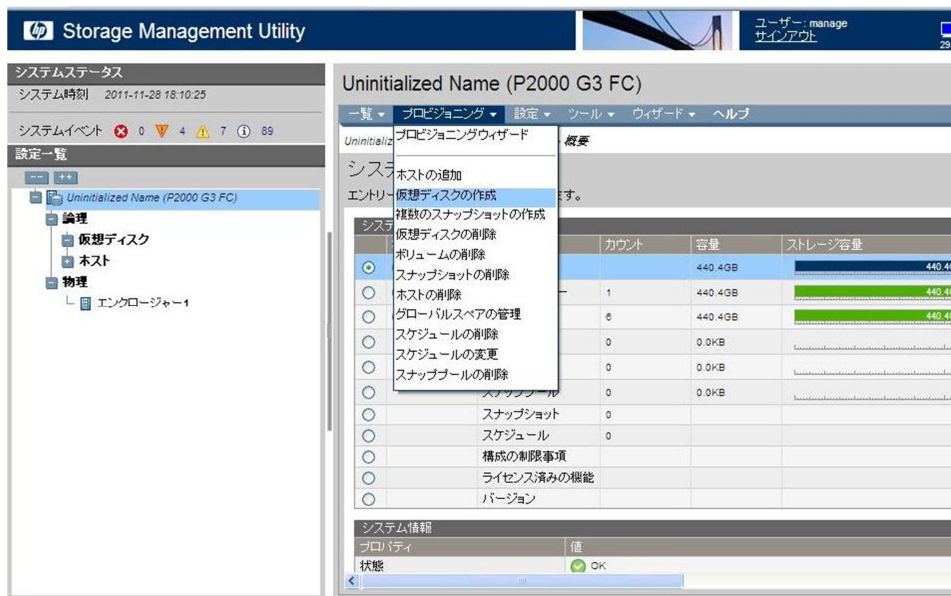


図 2. 仮想ディスク作成画面

- C) グローバルスペアを設定します。
- D) ボリュームを作成します。
- E) ボリュームのマッピングを行います。

### 2-3.PSP (ProLiant Support Pack) の適用

以下のサイトを参考にして、各ノードにPSPを適用してください。この時、hp-fc-enablementキットをインストールしてください。なお、PSPに含まれているドライバよりも新しいバージョンのドライバが個別に提供されている場合は、新しいバージョンのドライバを適用することを推奨します。

<http://h50146.www5.hp.com/products/software/oe/linux/mainstream/support/doc/general/mgmt/index.html#psp>

PSP適用後、O/Sをrebootします。

```
# shutdown -r now
```

O/Sが起動されたら、hp-fc-enablementキットを使って、qla2xxxドライバのパラメーターを変更します。

以下を実行します。

A) ドライバのパラメータ変更

```
# /opt/hp/hp-fc-enablement/set_parm -m
```

B) Initial RAM Diskの更新

```
# /opt/hp/hp-fc-enablement/make_initrd
```

C) O/Sのreboot

```
# shutdown -r now
```

### 2-4.Device Mapper Multipath のインストール・設定

Device Mapper Multipathのインストール・設定を行います。以下を実行してください。

A) ノード1にrootでlogin

B) device-mapper-multipathとdevice-mapper-multipath-libsの2つがインストールされているか確認して、入っていない場合はインストールしてください。

<確認>

```
# rpm -qa | grep multipath
```

```
device-mapper-multipath-0.4.9-41.el6.x86_64
```

```
device-mapper-multipath-libs-0.4.9-41.el6.x86_64
```

<インストール>

```
# rpm -ivh device-mapper-multipath-0.4.9-41.el6.x86_64.rpm
```

```
# rpm -ivh device-mapper-multipath-libs-0.4.9-41.el6.x86_64.rpm
```

C) mpathconfユーティリティを使って、マルチパスを設定します。

```
# mpathconf --enable --find_multipaths y
```



- D) /etc/multipath.confファイルが作成されるので、そのファイルの「devices」セクション内に、以下の様にP2000 G3 MSA用設定を追加します。「Native Linux Device-Mapper Multipath for HP StorageWorks Disk Arrays reference guide」に掲載されているHP arrays parameter valuesを使用してください。

```

devices {
    device {
        vendor            "HP"
        product           "P2000 G3 FC|P2000G3 FC/iSCSI|P2000 G3 SAS|P2000 G3 iSCSI"
        getuid_callout    "/lib/udev/scsi_id --whitelisted --device=/dev/%n"
        prio              alua
        hardware_handler  "0"
        path_selector     "round-robin 0"
        path_grouping_policy group_by_prio
        failback          immediate
        rr_weight         uniform
        rr_min_io         100
        no_path_retry     18
        path_checker      tur
    }
}

```

- E) multipathdデーモンを起動します

```
# service multipathd start
```

- F) P2000 G3 MSA FCに作成したLUN (volume) に対応したmultipath deviceが、/dev/mapper下に生成されている事を確認します。下記はLUNを4個作成した場合の例です。今回の環境では、multipath device名に、システム定義によるuser\_friendly\_name (mpatha、mpathb、mpathc・・・mpath + アルファベットの形式) を使用しています。

```
# ll /dev/mapper/
```

```
合計 0
```

```

crw-rw---- 1 root root 10, 58 11月 29 16:37 2011 control
lrwxrwxrwx 1 root root    7 11月 29 16:45 2011 mpatha -> ../dm-3      ← LUN #1
lrwxrwxrwx 1 root root    7 11月 29 16:45 2011 mpathb -> ../dm-4      ← LUN #2
lrwxrwxrwx 1 root root    7 11月 29 16:45 2011 mpathc -> ../dm-5      ← LUN #3
lrwxrwxrwx 1 root root    7 11月 29 16:45 2011 mpathd -> ../dm-6      ← LUN #4
lrwxrwxrwx 1 root root    7 11月 29 16:38 2011 vg_node2-lv_home -> ../dm-2
lrwxrwxrwx 1 root root    7 11月 29 16:38 2011 vg_node2-lv_root -> ../dm-0
lrwxrwxrwx 1 root root    7 11月 29 16:38 2011 vg_node2-lv_swap -> ../dm-1

```

- G) 各multipath device (/dev/mapper/mpatha等) 毎に、4本のpath、即ち4個のblock device (/dev/sda等) と2つのpath groupが構成されている事を確認します。

```
# multipath -ll
```

```

mpathd (3600c0ff000da6a2e4808d54e01000000) dm-6 HP,P2000 G3 FC      ← LUN #4
size=17G features='1 queue_if_no_path' hwhandler='0' wp=rw
|+- policy='round-robin 0' prio=50 status=active                    ← path group
| |- 3:0:0:4 sdd            8:48 active ready running            ← path
| `-- 2:0:0:4 sdl           8:176 active ready running            ← path
`+- policy='round-robin 0' prio=10 status=enabled                  ← path group
  |- 3:0:1:4 sdh            8:112 active ready running            ← path
  `-- 2:0:1:4 sdp           8:240 active ready running            ← path
mpathc (3600c0ff000da6a2ee907d54e01000000) dm-5 HP,P2000 G3 FC      ← LUN #3

```

```

size=17G features='1 queue_if_no_path' hwhandler='0' wp=rw
|+- policy='round-robin 0' prio=50 status=active
| |- 3:0:0:3 sdc      8:32  active ready running
| `-- 2:0:0:3 sdk      8:160 active ready running
`+- policy='round-robin 0' prio=10 status=enabled
  |- 3:0:1:3 sdg      8:96  active ready running
  `-- 2:0:1:3 sdo      8:224 active ready running
mpathb (3600c0ff000da6a2ea807d54e01000000) dm-4 HP,P2000 G3 FC    ← LUN #2
size=17G features='1 queue_if_no_path' hwhandler='0' wp=rw
|+- policy='round-robin 0' prio=50 status=active
| |- 3:0:0:2 sdb      8:16  active ready running
| `-- 2:0:0:2 sdj      8:144 active ready running
`+- policy='round-robin 0' prio=10 status=enabled
  |- 3:0:1:2 sdf      8:80  active ready running
  `-- 2:0:1:2 sdn      8:208 active ready running
mpatha (3600c0ff000da6a2e5107d54e01000000) dm-3 HP,P2000 G3 FC    ← LUN #1
size=17G features='1 queue_if_no_path' hwhandler='0' wp=rw
|+- policy='round-robin 0' prio=50 status=active
| |- 3:0:0:1 sda      8:0   active ready running
| `-- 2:0:0:1 sdi      8:128 active ready running
`+- policy='round-robin 0' prio=10 status=enabled
  |- 3:0:1:1 sde      8:64  active ready running
  `-- 2:0:1:1 sdm      8:192 active ready running

```

H) 各multipath device1に対してパーティションを作成します。

```

# fdisk /dev/mapper/mpatha
# fdisk /dev/mapper/mpathb
# fdisk /dev/mapper/mpathc
# fdisk /dev/mapper/mpathd

```

I) パーティションが正常に作成されているか確認します。

```

# ll /dev/mapper/
合計 0
crw-rw---- 1 root root 10, 58 11月 29 16:37 2011 control
lrwxrwxrwx 1 root root    7 11月 29 16:53 2011 mpatha -> ../dm-3
lrwxrwxrwx 1 root root    7 11月 29 16:53 2011 mpathap1 -> ../dm-7
lrwxrwxrwx 1 root root    7 11月 29 16:53 2011 mpathb -> ../dm-4
lrwxrwxrwx 1 root root    7 11月 29 16:53 2011 mpathbp1 -> ../dm-8
lrwxrwxrwx 1 root root    7 11月 29 16:53 2011 mpathc -> ../dm-5
lrwxrwxrwx 1 root root    7 11月 29 16:53 2011 mpathcp1 -> ../dm-9
lrwxrwxrwx 1 root root    7 11月 29 16:53 2011 mpathd -> ../dm-6
lrwxrwxrwx 1 root root    8 11月 29 16:53 2011 mpathdp1 -> ../dm-10
lrwxrwxrwx 1 root root    7 11月 29 16:38 2011 vg_node2-lv_home -> ../dm-2
lrwxrwxrwx 1 root root    7 11月 29 16:38 2011 vg_node2-lv_root -> ../dm-0
lrwxrwxrwx 1 root root    7 11月 29 16:38 2011 vg_node2-lv_swap -> ../dm-1

```

J) 該当デバイスに対してファイルシステムを作成します。下記はext4ファイルシステムを作成する例です。

```

# mkfs.ext4 /dev/mapper/mpathap1
# mkfs.ext4 /dev/mapper/mpathbp1
# mkfs.ext4 /dev/mapper/mpathcp1
# mkfs.ext4 /dev/mapper/mpathdp1

```

```

K) マウントポイントを作成します。
# mkdir /mnt1 /mnt2 /mnt3 /mnt4
L) ファイルシステムをmountし、正常にマウント出来る事を確認します。
# mount /dev/mapper/mpathap1 /mnt1
# mount /dev/mapper/mpathbp1 /mnt2
# mount /dev/mapper/mpathcp1 /mnt3
# mount /dev/mapper/mpathdp1 /mnt4
M) 正常にファイルを作成できる事を確認します。
N) 動作確認終了後はアンマウントしておきます。
# umount /mnt1
# umount /mnt2
# umount /mnt3
# umount /mnt4
O) ノード2にrootでログイン
P) device-mapper-multipathとdevice-mapper-multipath-libsの2つがインストールされているか確認して、入っていない場合はインストールしてください。
<確認>
# rpm -qa | grep multipath
device-mapper-multipath-0.4.9-41.el6.x86_64
device-mapper-multipath-libs-0.4.9-41.el6.x86_64
<インストール>
# rpm -ivh device-mapper-multipath-0.4.9-41.el6.x86_64.rpm
# rpm -ivh device-mapper-multipath-libs-0.4.9-41.el6.x86_64.rpm
Q) ノード1の/etc/multipath.confファイルと/etc/multipath.bindingsファイルをノード2にコピーします。以下はscpコマンドでコピーする例です。
# scp ノード1のホスト名:/etc/multipath.conf /etc
# scp ノード1のホスト名:/etc/multipath/bindings /etc/multipath
R) multipathdデーモンを起動します
# service multipathd start
S) システム起動時、multipathdデーモンが自動起動されるように設定します。
# chkconfig multipathd on
T) multipath deviceとパーティションが正しく認識されているか確認します。
# ll /dev/mapper/
合計 0
crw-rw---- 1 root root 10, 58 11月 29 16:39 2011 control
lrwxrwxrwx 1 root root 7 11月 29 17:01 2011 mpatha -> ../dm-3
lrwxrwxrwx 1 root root 7 11月 29 17:01 2011 mpathap1 -> ../dm-9
lrwxrwxrwx 1 root root 7 11月 29 17:01 2011 mpathb -> ../dm-4
lrwxrwxrwx 1 root root 7 11月 29 17:01 2011 mpathbp1 -> ../dm-8
lrwxrwxrwx 1 root root 7 11月 29 17:01 2011 mpathc -> ../dm-5
lrwxrwxrwx 1 root root 8 11月 29 17:01 2011 mpathcp1 -> ../dm-10
lrwxrwxrwx 1 root root 7 11月 29 17:01 2011 mpathd -> ../dm-6
lrwxrwxrwx 1 root root 7 11月 29 17:01 2011 mpathdp1 -> ../dm-7
lrwxrwxrwx 1 root root 7 11月 29 16:39 2011 vg_node1-lv_home -> ../dm-2
lrwxrwxrwx 1 root root 7 11月 29 16:39 2011 vg_node1-lv_root -> ../dm-0
lrwxrwxrwx 1 root root 7 11月 29 16:39 2011 vg_node1-lv_swap -> ../dm-1

```

- U) マウントポイントを作成します。マウントポイント名は、必ずノード1と同一にしてください。
  - # mkdir /mnt1 /mnt2 /mnt3 /mnt4
- V) ファイルシステムをmountし、正常にマウント出来る事を確認します。
  - # mount /dev/mapper/mpathap1 /mnt1
  - # mount /dev/mapper/mpathbp1 /mnt2
  - # mount /dev/mapper/mpathcp1 /mnt3
  - # mount /dev/mapper/mpathdp1 /mnt4
- W) 先ほど、ノード1で作成したファイルが正常に見える事を確認します。
- X) 動作確認終了後はアンマウントします。
  - # umount /mnt1
  - # umount /mnt2
  - # umount /mnt3
  - # umount /mnt4
- Y) 再度、ノード1（プライマリサーバー）からファイルシステムをmountします。
  - # mount /dev/mapper/mpathap1 /mnt1
  - # mount /dev/mapper/mpathbp1 /mnt2
  - # mount /dev/mapper/mpathcp1 /mnt3
  - # mount /dev/mapper/mpathdp1 /mnt4
- Z) 両ノードから、ファイルシステムがマウント可能で、アクセス可能である事が確認できたならば、ファイルシステムをマウントするのは、ノード1（プライマリサーバー）だけにしてください。

## 2-5.ネットワークの確立

両ノードのセットアップが完了したら、それぞれのノードに対して、pingが可能か確認してください。また、それぞれのノードの/etc/hostsファイルに、localhost名のエン트리、切り替え可能なIP アドレス（仮想IPアドレス）とそれぞれのノードのIPアドレスと対応するホスト名（エイリアス）を正しく登録してください。

## 2-6./etc/fstab ファイルの編集

/etc/fstabファイルにラベル名を使用している場合は、ブロックデバイス名に変更します。LifeKeeperでは/etc/fstab内では、ラベル名ではなく、ブロックデバイス名を使用することが推奨されています。

## 2-7.LifeKeeper Installation Support CD を使った設定

LifeKeeper をインストールするために必要な前作業をLifeKeeper Installation Support CD を使って行います。以下の作業を両方のノード上で実行してください。

- A) CDRROM ドライブにLifeKeeper Installation Support CD を挿入します。
- B) mount /media/CDROM の実行（必要ならば）
- C) cd /media/CDROM/Install の実行
- D) sh setup の実行
- E) 画面に表示される質問に、応答してください。

- F) 途中、以下のように unique host ID が表示されます。この ID は、後で License Key を取得する時に必要になるので、正確に記録しておいてください。

The unique host ID for this system is listed below.

00XX7856XXX0

- G) 「Would you like to install a license key now? (y/n) [n] ?」と質問されます。License Key は後でインストールするので、ここではリターン（もしくは n で応答）。
- H) 「Setup has completed successfully.」と表示されます。
- I) システムを reboot します。  
# shutdown -r now
- J) /var/log/LK\_install.log ファイルにログが書かれます。

## 2-8. License Key のインストール

以下の手順で、各ノードごとに License Key をインストールしてください。

- A) 2-7. LifeKeeper Installation Support CD を使った設定の手順 F) で表示された unique host ID と LifeKeeper ソフトウェアに同梱されている Entitlement ID (Authorization Code) で、米国サイオステクノロジー社 (<http://us.sios.com> の LICENSE KEY リンク) から各ノードの License Key を入手してください。
- B) 各ノードに root で login
- C) 以下のコマンドを実行  
# /opt/LifeKeeper/bin/lkkeyins
- F) 画面に表示される質問に、適切に応答してください。
- G) 「LifeKeeper license key installation was successful!」と表示されれば、License Key のインストールは成功です。

## 2-9. LifeKeeper for Linux v7.4 のインストール

LifeKeeper for Linux v7.4 をインストールするために、両ノード上で、以下のことを実行してください。

- A) root で login
- B) CDRROM ドライブに LifeKeeper Core CD を挿入します。
- C) mount /media/CDROM の実行（必要ならば）
- D) cd /media/CDROM/Core の実行
- E) パッケージをインストールします。Core パッケージである steeleye-lk-7.4.0-63.i386.rpm をかならず最初にインストールしてください。  
# rpm -ivh steeleye-lk\*.rpm
- F) インストール中、いくつかメッセージが表示されますが、エラーメッセージが表示されなければ、インストールは成功です。
- G) オプションの DMMP ARK を LifeKeeper Recovery Kits のメディアからインストールしてください。  
# rpm -ivh steeleye-lkDMMP-7.3.0-2.noarch.rpm

H) 以下のコマンドを実行して、パッケージがインストールされたか確認します。

```
# rpm -qa | grep steeleye
steeleye-lkRAW-7.4.0-63.noarch
steeleye-lk-7.4.0-63.i386
steeleye-lkRHAS-7.4.0-26.noarch
steeleye-lkLIC-7.4.0-26.i386
steeleye-lkIP-7.4.0-63.noarch
steeleye-lkHLP-7.4.0-63.noarch
steeleye-lkDMMP-7.3.0-2.noarch
steeleye-lkGUI-7.4.0-63.i386
steeleye-lkMAN-7.4.0-63.noarch
```

I) 環境変数 PATH および MANPATH を以下のように設定しておく便利です。

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

## 2-10.LifeKeeper の起動

LifeKeeper を起動するために、両ノードで以下のことを実行してください。

A) LifeKeeper を起動します。

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

なお、LifeKeeper 起動時、次のメッセージが表示されますが、無視してください。

```
WARNING: All config files need .conf: /etc/modprobe.d/fc-hba.conf.old, it will be
ignored in a future release.
```

B) LifeKeeper デーモンが起動されたか ps コマンドで確認します。

```
# ps -ef | grep LifeKeeper
```

C) 以下のように表示されれば OK。

```
root      15269      1  0 18:23 ?                00:00:00 /opt/LifeKeeper/bin/lk_logmgr
-l/opt/LifeKeeper/out -d/etc/default/LifeKeeper
root      15297      1  0 18:23 ?                00:00:00 /opt/LifeKeeper/bin/lcm
root      15299      1  0 18:23 ?                00:00:00 /opt/LifeKeeper/bin/lcd
root      15302      1  0 18:23 ?                00:00:00 /opt/LifeKeeper/bin/lkcheck
root      15304      1  0 18:23 ?                00:00:00 /opt/LifeKeeper/bin/lkscsid
root      15306      1  0 18:23 ?                00:00:00 /opt/LifeKeeper/bin/ttymonlcm
root      15308      1  0 18:23 ?                00:00:00 /bin/sh /opt/LifeKeeper/bin/runGu
iServer
root      15851  15308  2 18:23 ?                00:00:02 java -Xint -Xss3M -DS_LK=true
-Djava.rmi.server.hostname=node1 -Dcom.steeleye.LifeKeeper.rmiPort=82
-Dcom.steeleye.LifeKeeper.LKROOT=/opt/LifeKeeper -DGUI_RMI_REGISTRY=internal
-DGUI_WEB_PORT=81 com.steeleye.LifeKeeper.beans.S_LK
root      19859  19361  0 18:23 pts/2          00:00:00 grep LifeKeeper
D) LifeKeeper を停止する場合は、以下のコマンドを実行してください。
# /opt/LifeKeeper/bin/lkstop
```

## 2-11.LifeKeeper GUI の起動

LifeKeeper GUI を起動するために、両ノードで、以下のことを実行してください。なお、今回の構成では、LifeKeeper GUI のクライアント/サーバーは同一マシンです。

- A) LifeKeeper GUI パッケージがインストールされているか rpm コマンドで確認します。  
# rpm -qa |grep steeleye-lkGUI
- B) 以下のように表示されれば OK。  
steeleye-lkGUI-7.4.0-63.i386
- C) LifeKeeper GUI サーバーが起動されているか ps コマンドで確認します。  
# ps -ef|grep runG
- D) 以下のように表示されれば OK。  
root 15308 1 0 18:23 ? 00:00:00 /bin/sh /opt/LifeKeeper/bin/runGuiServer
- E) もし、LifeKeeper GUI サーバーが起動されていない場合は、以下のコマンドを実行して起動します。  
# /opt/LifeKeeper/bin/lkGUIserver start
- F) LifeKeeper GUI を起動します。X Window 上から以下のコマンドを実行。  
# /opt/LifeKeeper/bin/lkGUIapp
- G) Cluster Connect Dialog が表示されます。
- H) LifeKeeper GUI サーバーを停止する場合は、以下のコマンドを実行してください。  
# /opt/LifeKeeper/bin/lkGUIserver stop
- I) 初期インストール後、一度、LifeKeeper GUI サーバーを起動すれば、LifeKeeper の起動/停止に伴い、LifeKeeper GUI サーバーも起動/停止されます。
- J) 自ノードではない他のマシンから GUI クライアントを使用して LifeKeeper GUI サーバーへ接続し、通信を行う時、以下のポート番号が使用されるので、この点を考慮して、パケットフィルタリングして下さい。  
81(TCP) : GUI サーバープロセスで使用  
82(TCP) : GUI サーバープロセスで使用  
1024(TCP) ~ : GUI のための RMI 通信で使用

## 2-12. クラスタ設定、Volume (P2000 G3 MSA FC) リソースの設定

LifeKeeper GUI からクラスタの設定を行います。以下の作業は、ノード 1 でのみ行います。

- A) クラスタの構成  
ノード 2 とコミュニケーションパスをはり、クラスタを構築します。  
Edit > Server > Create Comm Path から 2 本のコミュニケーションパスをノード 2 とはり、クラスタを構築します。
- B) Volume リソースの作成  
DMMP 環境でも、DMMP ARK インストール後は、通常の FileSystem リソース作成と同手順で、DMMP の共有ディスクリソースが作成可能です。  
Edit > Server > Create Resource Hierarchy.. を選びます。  
Select Recovery Kit で File system を選びます。正常にインストール及び、マウントされている場合は、Mount Point 項目で、マルチパスデバイスがマウントされているディレクトリが表示され、Wizard を進んでいくと、リソースが作成されます。

## 3. 注意事項

- A) /etc/fstab ファイルの編集  
/etc/fstab ファイルにラベル名を使用している場合は、ブロックデバイス名に変更してください。LifeKeeper では/etc/fstab 内では、ラベル名ではなく、ブロックデバイス名を使用することが推奨されています。

B) LifeKeeper 起動時、表示されるメッセージ

LifeKeeper 起動時、次のメッセージが表示されますが、無視してください。

```
WARNING: All config files need .conf: /etc/modprobe.d/fc-hba.conf.old, it will be
ignored in a future release.
```

C) LifeKeeper GUI が使用するポート

自ノードではない他のマシンから GUI クライアントを使用して LifeKeeper GUI サーバーへ接続し、通信を行う時、以下のポート番号が使用されるので、この点を考慮して、パケットフィルタリングして下さい。

81 (TCP) : GUI サーバープロセスで使用  
82 (TCP) : GUI サーバープロセスで使用  
1024 (TCP) ~ : GUI のための RMI 通信で使用

以上