

CLUSTERPRO MC StorageSaver 1.0 for Linux Linux システムディスクアレイ装置 障害監視構築ガイド

第2版

2013年5月

日本電気株式会社

はじめに

本資料について

本資料は、Linux システムに接続されたディスクアレイ装置などの外部ストレージの障害監視を導入する手順を説明します。

監視の構成

本資料では、次のような構成の監視システムを構築します。

- iStorage StoragePathSavior による I/O パスの冗長化
- StorageSaver による I/O パスの障害監視
- CLUSTERPRO によるサーバの冗長化
- サーバ管理基盤(RENS)を用いた CLUSTERPRO と StorageSaver の連携

注意事項

- 本資料で紹介するのはあくまでも構築事例であり、実際の業務システムへ適用する場合は要件や実際のハードウェア構成に応じてカスタマイズする必要があります。
- 設定ファイルの妥当性は必ず目視で確認してください。
- 作業は、スーパーユーザーアカウント (root 権限)で行ってください。
- rpm パッケージのバージョンは『xxx』で表現しています。

目次

1. システム構成について

- i. ハードウェア構成
- ii. ソフトウェア構成

2. 構築手順

- i. サーバ管理基盤(RENS)セットアップ
- ii. StoragePathSaviorによる I/O パスの冗長化
- iii. 構成情報の確認
- iv. LVM環境構築、ファイルシステム作成
- v. 共有ディスクの設定
- vi. StorageSaver の監視定義
- vii. CLUSTERPRO 連携設定

3. 動作確認

- i. 障害試験事前確認
- ii. FC 抜線による障害発生時のシステム連携確認

4. 付録

- i. サーバ管理基盤(RENS)を利用せず、StorageSaver と CLUSTERPRO を直接連携する方法
- ii. FC 抜線による障害発生時のシステム連携確認

1. システム構成について

i. ハードウェア構成

各 Server から見たディスク構成

Disk 名	Disk1		Disk2	
Disk サイズ	2GB		2GB	
I/O パス	/dev/sda	/dev/sdb	/dev/sdc	/dev/sdd
SPS(※1) デバイス ファイル	/dev/dda		/dev/ddb	
パーティション数	2		0	
パーティション名	dda1(※2)	dda2(※3)	-	
LVM	×		○	
VG 名	- (StorageSaver では "PSEUDO_VG001" という仮想VG名で管理)		VolGroup001	
LV 名	-		lvol0	
LV サイズ	-		1GB	

※1 "SPS"は、"StoragePathSavior"の略称です。

※2 ディスクハートビート用

※3 ディスクリソース用

1. システム構成について

ii. ソフトウェア構成

スケーラブル HA サーバを用いる場合と、SIGMABLADE などのその他一般 Linux サーバを用いる場合ではソフトウェア構成が異なります。
使用環境がスケーラブル HA サーバである場合はサーバ管理基盤(RENS)を使用して CLUSTERPRO と連携します。
その他一般 Linux サーバの場合はサーバ管理基盤(RENS)を使用せず StorageSaver が直接 CLUSTERPRO と連携します。

※ サーバ管理基盤は、以降、RENS(Resource Event Notification Service)と呼びます。

本資料が対象とするソフトウェア構成は次ページ以降に示します。

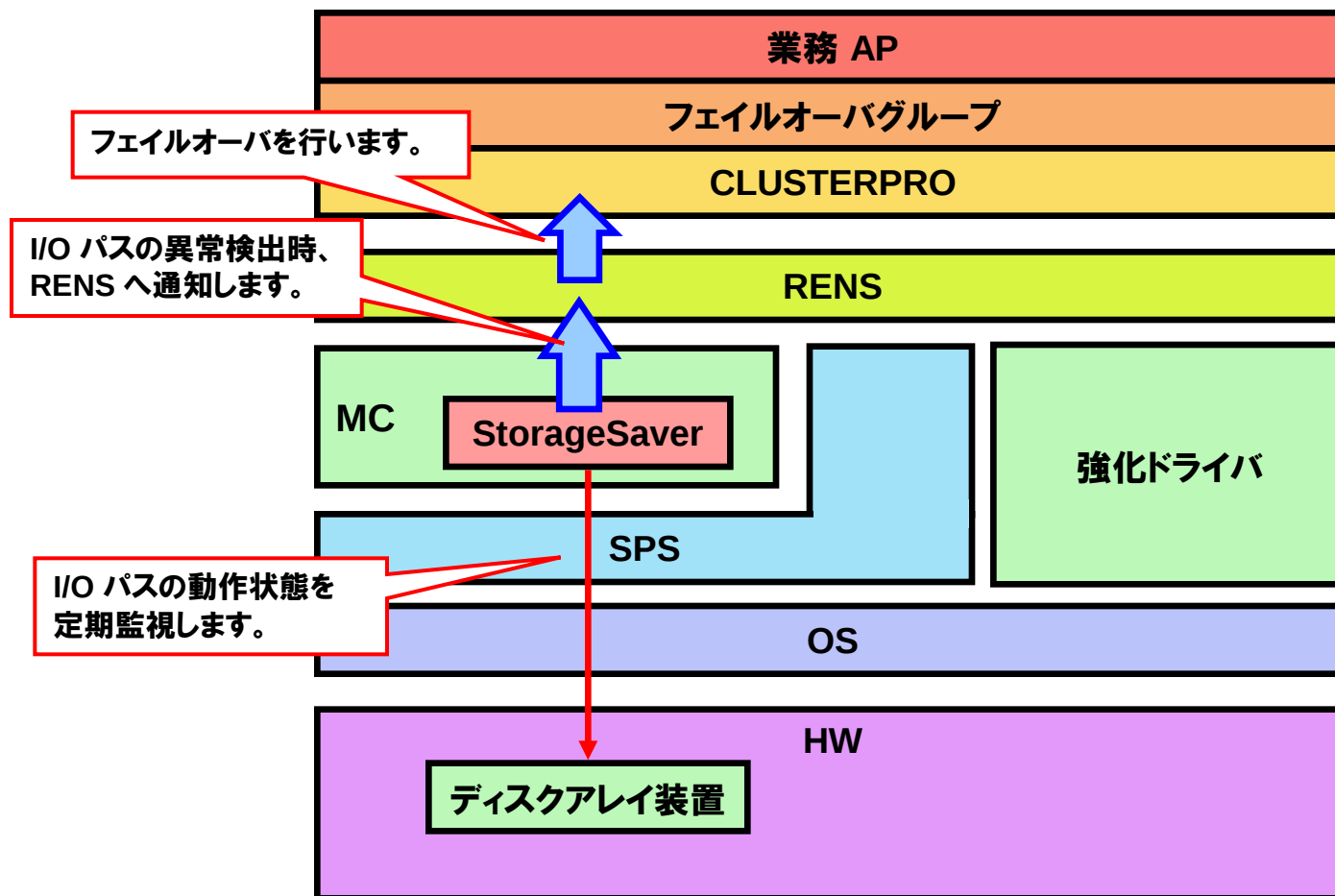
本資料は RENS 連携を行う構成の手順について記載しています。
一般 Linux サーバ環境等、RENS 連携を行わない構成の場合は、付録の『サーバ管理基盤(RENS)を利用せず、StorageSaver と CLUSTERPRO を直接連携する方法』を参照してください。

1. システム構成について

ii. ソフトウェア構成

① スケーラブルHAサーバ構成

RENS を使用して CLUSTERPRO と連携します。

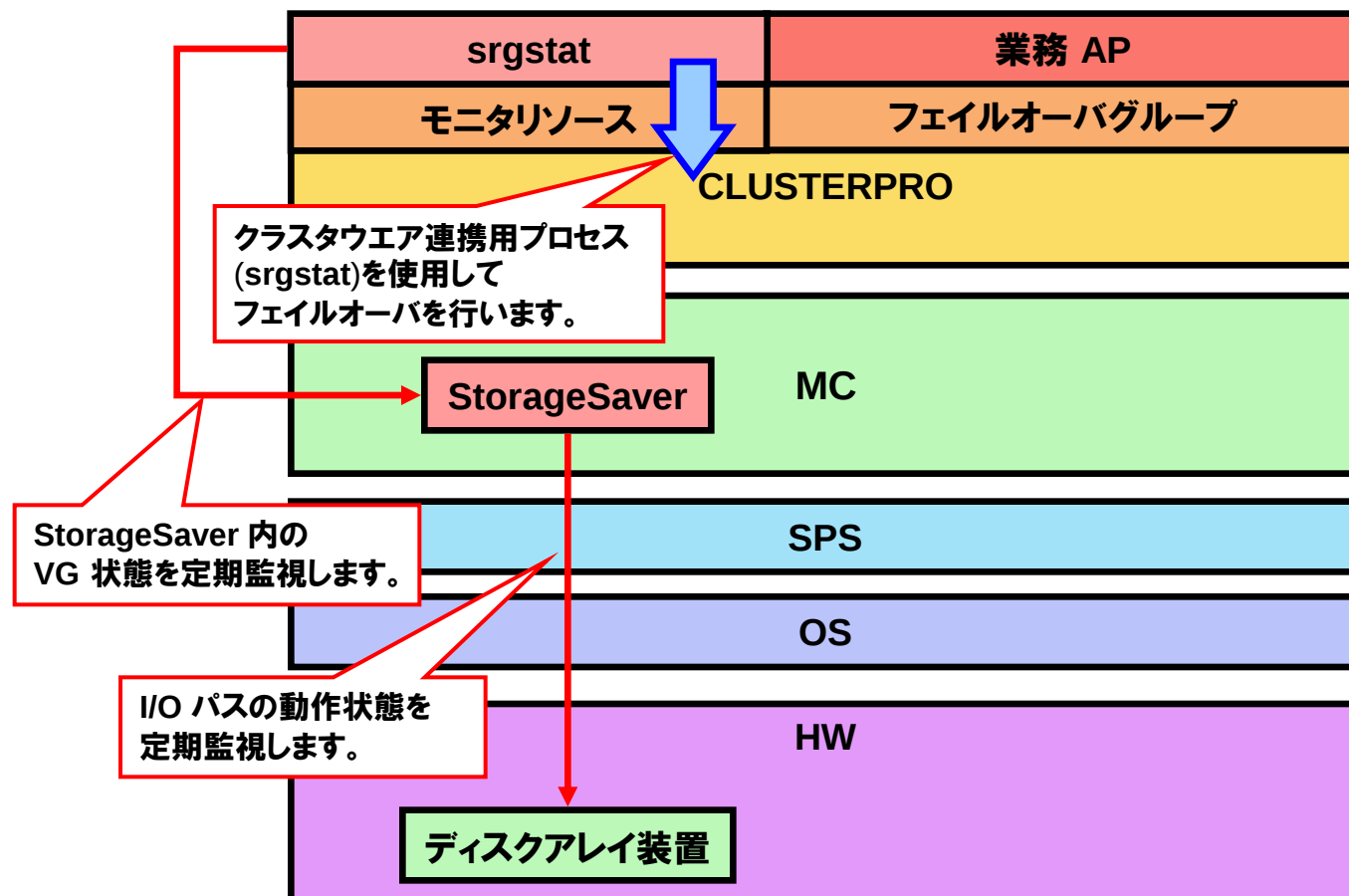


1. システム構成について

ii. ソフトウェア構成

② その他一般 Linux サーバ構成

RENS を使用せずに CLUSTERPRO と連携します。



2. 構築手順

i. サーバ管理基盤(RENS)セットアップ

RENS は、サーバのハードウェア、ソフトウェア障害イベントを捕捉し、ログ出力を行い、クラスタリングソフトウェア、syslog、テキストログへ通知するサービスを提供します。
ここではそのセットアップ手順を記載します。

※ 使用環境が一般 Linux サーバなど、RENS 連携を行わない構成の場合は、
この手順 (サーバ管理基盤(RENS)セットアップ) を行う必要はありません。

2. 構築手順

i. サーバ管理基盤(RENS)セットアップ

① RENS インストール

1. RENS をインストールするために RENS パッケージが含まれる CD-ROM 媒体を mount します。

※ CD-ROM パスが /dev/cdrom, mount 先が /mnt/cdrom の場合

```
# /bin/mount /dev/cdrom /mnt/cdrom
```

2. rpm コマンドにより、RENS カーネルパッケージをインストールします。

```
# /bin/rpm -ivh /mnt/cdrom/RENS-x.x/mcl-rens-kernel-x.x-x.x.rpm
```

3. rpm コマンドにより、RENS ユーザーパッケージをインストールします。

```
# /bin/rpm -ivh /mnt/cdrom/RENS-x.x/mcl-rens-base-x.x-x.x.rpm
```

2. 構築手順

i. サーバ管理基盤(RENS)セットアップ

4. RENS がインストールされていることを確認します。

```
# /bin/rpm -qa | /bin/grep rens  
mcl-rens-kernel-x.x-x  
mcl-rens-base-x.x-x
```

5. 以上で RENS パッケージのインストールは終了です。

2. 構築手順

i. サーバ管理基盤(RENS)セットアップ

② RENS セットアップ

1. リソース情報ファイル(/opt/mcl/rens/conf/resource.conf)を作成します。

```
# /opt/mcl/rens/script/make_resource_bond.sh > /opt/mcl/rens/conf/resource.conf
```

2. 構築手順

i. サーバ管理基盤(RENS)セットアップ

2. モニタプロセス起動スクリプトファイルを編集して初期設定を行います。

※ 今回の例は vi コマンドを使用して編集を行います。

```
# /bin/vi /opt/mcl/rens/script/monitor_run.sh
```

```
#!/bin/sh
```

```
RENS_HOME=/opt/mcl/rens
```

```
RENS_BIN=${RENS_HOME}/bin
```

```
RENS_SCRIPT=${RENS_HOME}/script
```

CLUSTERPRO との連携箇所をコメントアウトします。
本処理は CLUSTERPRO インストール後に適用します。

```
#{RENS_BIN}/targetregclpd -f /opt/mcl/rens/conf/targetclp.conf &
```

```
#{RENS_SCRIPT}/check_target.sh -t clpx
```

```
#{RENS_BIN}/nicmon -n e1000 &  
#{RENS_BIN}/fcmon -n lpfc &
```

本手順書では RENS 連携用 e1000 ドライバを導入しません。
e1000 ドライバとの連携箇所をコメントアウトします。

2. 構築手順

i. サーバ管理基盤(RENS)セットアップ

3. カーネルモジュールの依存関係を更新します。

```
# /sbin/depmod -a `uname -r`
```

4. rc スクリプトを OS に登録します。

```
# /sbin/chkconfig --add rens_lower
```

5. rc スクリプトが正しく登録されたことを確認します。

```
# /sbin/chkconfig --list rens_lower
```

```
rens_lower    0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

2. 構築手順

i. サーバ管理基盤(RENS)セットアップ

6. サービスを開始します。

```
# /sbin/service rens_lower start
```

```
Mounting /sys/kernel/debug: RENS lower layer initialized.
```

7. サービスが正しく開始したことを確認します。

```
# /bin/mount | /bin/grep debugfs
```

```
none on /sys/kernel/debug type debugfs (rw)
```

2. 構築手順

i. サーバ管理基盤(RENS)セットアップ

RENS 連携のために強化ドライバのインストールおよびセットアップを行います。

8. FCドライバのインストール

```
# cd /root
# /bin/cp /mnt/cdrom/DRIVERS-x.x/lpfc-2.6-x.x.x.xx-x.x.x-MCL-bin.tbz2 ./
# /bin/tar jxvf ./lpfc-2.6-x.x.x.xx-x.x.x-MCL-bin.tbz2
lpfc.ko
```

9. 展開したファイルを全て /lib/modules/`uname -r`/extra/mcl にコピーします。

```
# /bin/mkdir -p /lib/modules/`uname -r`/extra/mcl
# /bin/cp lpfc.ko /lib/modules/`uname -r`/extra/mcl
```

※ カレントディレクトリ内の lpfc-2.6-x.x.x.xx-x.x.x-MCL-bin.tbz2 と、lpfc.ko は一時ファイルですので、上記コマンド実行後に削除してください。

10. モジュール依存関係を更新します。

```
# /sbin/depmod -a `uname -r`
```

2. 構築手順

i. サーバ管理基盤(RENS)セットアップ

11. initrd を更新します。

```
# /sbin/mkinitrd initrd-`uname -r`.img `uname -r`
```

12. イメージファイルがカレントディレクトリにできますので /boot にコピーします。

※ RHEL5.3 の場合は、initrd-2.6.18-128.el5.img というファイルが作成されます。

カレントディレクトリに作成されたイメージファイルは一時ファイルですので、下記コマンド実行後に削除してください。

```
# /bin/cp /boot/initrd-2.6.xx-xxx.img /boot/initrd-2.6.xx-xxx.img.bkup  
# /bin/cp initrd-2.6.xx-xxx.img /boot
```

2. 構築手順

i. サーバ管理基盤(RENS)セットアップ

13. 動作中のドライバを削除してインストールしたドライバをロードするために、以下のコマンドを実行します。

※ 失敗する場合はドライバ使用中なので、OS 再起動が必要です。OS 再起動によってドライバが自動的に置き換わります。

```
# /sbin/modprobe -r lpfc  
# /sbin/modprobe lpfc
```

14. 以上で FC ドライバのインストールは終了です。

15. RENS カーネルモジュールがロードされていることを確認します。

```
# /sbin/lsmmod | /bin/grep rens  
rens_kernel_mod      xxxxxx x lpfc
```

※ xxxxxx にはモジュールサイズ、x には使用カウント数が表示されます。

2. 構築手順

i. サーバ管理基盤(RENS)セットアップ

16. RENS を起動させます。

```
# /opt/mcl/rens/script/rens_start.sh  
warning: /etc/inittab saved as /etc/inittab.renssave  
RENS started.
```

※ lpfc 用モニタプロセス(fcmon)登録後の RENS 初回起動時のみ syslog へ以下のメッセージが出力されますが、動作には問題ありませんので無視してください。
--RENS-- WARNING: Cannot read expected data size:
key of RENS(lower) temporary buffer: path=/opt/mcl/rens/conf/lower/tmpbuf/resource/lpfc.key, size=0/10

17. インストール CD-ROM を umount して、媒体を取り出します。

```
# /bin/umount /dev/cdrom
```

18. 以上で RENS セットアップは終了です。

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

StoragePathSavior は、サーバと iStorage ディスクアレイ装置を接続したシステムにおいて、サーバ・ストレージ装置間の I/O パスの冗長化を実現します。

SAN システムにおいてストレージ装置は主要な障害箇所の一つであり、サーバ・ストレージ装置間の物理的な I/O パスの障害が起こるとストレージ装置へのアクセスができなくなります。

StoragePathSavior は、そのような脆弱性を I/O パスの冗長化により排除することができます。ここではその StoragePathSavior の導入手順について記載します。

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

① StoragePathSavior インストール

1. StoragePathSavior をインストールするために StoragePathSavior パッケージが含まれる CD-ROM 媒体を mount します。

※ CD-ROM パスが /dev/cdrom, mount 先が /mnt/cdrom の場合

```
# /bin/mount /dev/cdrom /mnt/cdrom
```

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

2. 使用 OS のカーネルバージョンが StoragePathSavior が動作可能なカーネルバージョンであることを確認します。

※ sps-mcopt-xxx、sps-driver-E-xxx のどちらのファイルでも確認が可能です。
今回の例は sps-driver-E-xxx のファイルを対象にして確認を行った結果を表記しています。

```
# /bin/uname -r
```

2.6.x-xx

ご使用中のカーネルバージョン

```
# /bin/rpm -qip /mnt/cdrom/Express5800_100_NX7700i/RPMS/ < 使用ディストリビューション >  
/sps-driver-E-xxx.rpm
```

Name : sps-driver-E (sps-mcopt) Relocations: (not relocatable)

Version : xxxx Vendor: NEC Corporation

< 省略 >

The driver (dd_mod,sps_mod,sps_mod2) provides the redundant SCSI-path for
NEC iStorage Disk Array System.

This Driver works on Red Hat Enterprise Linux 5 (2.6.x-xx)

StoragePathSavior が動作可能なカーネルバージョン

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

3. rpm コマンドにより、StoragePathSavior パッケージをインストールします。

```
# /bin/rpm -ivh /mnt/cdrom/Express5800_100_NX7700i/RPMS/ <使用ディスト  
リビューション> /sps-util-x.x.x-x.rpm
```

```
# /bin/rpm -ivh /mnt/cdrom/Express5800_100_NX7700i/RPMS/ <使用ディスト  
リビューション> /sps-driver-E-x.x.x-x.x.xx.xxx.xxx.rpm
```

```
# /bin/rpm -ivh /mnt/cdrom/Express5800_100_NX7700i/RPMS/ <使用ディスト  
リビューション> /sps-mcopt-x.x.x-x.rpm
```

4. パッケージがインストールされていることを確認します。

```
# /bin/rpm -qa | /bin/grep sps  
sps-util-x.x.x-x  
sps-driver-E-x.x.x-x.x.xx.xxx.xxx  
sps-mcopt-x.x.x-x
```

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

5. アップデートモジュールの適用を行います。

/mnt/cdrom/Express5800_100_NX7700i/update/ 内のディレクトリを確認し、該当するアップデートモジュールがある場合は、お使いのバージョンに適したモジュールを適用してください。

```
# /bin/ls /mnt/cdrom/Express5800_100_NX7700i/update/
```

※ 詳細は StoragePathSavior のインストールガイドを参照してください。

6. インストール CD-ROM を umount して、媒体を取り出します。

```
# /bin/umount /dev/cdrom
```

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

7. 以下のコマンドを実行して、エラーが表示されないことを確認します。

```
# /sbin/depmmod -a `uname -r`
```

8. StoragePathSavior 自動起動の設定を行うため、OS 起動スクリプトにパッチファイルを適用します。

```
# cd /etc/rc.d  
# /usr/bin/patch -b -p0 < /opt/nec/sps/patch/rc.sysinit.rhel5.diff
```

※ パッチファイル適用時に Unreversed patch detected! Ignore -R? [n] と表示された場合は“n”を、次に表示される Apply anyway? [n] に対しては“y”と入力してください。

※ 手順 9 は、手順 8 にてエラーメッセージが出力される場合のみ行ってください。

9. 手順8 でエラーメッセージが出力された場合、パッチファイルの一行目が “+” で表されている行を `/etc/rc.d/rc.sysinit` に追加してください。
追加する場所は、パッチファイルの内容を参考にしてください。

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

10. モニタプロセス設定ファイルを作成します。

```
# /bin/cp /opt/mcl/rens/conf/monitor.conf.template  
/opt/mcl/rens/conf/sps.conf
```

11. 他のモニタプロセス設定ファイルの shm_key 値を取得します。

```
# /bin/cat /opt/mcl/rens/conf/*.conf | /bin/grep "shm_key"
```

12. sps.conf の shm_key 値を上記コマンドで取得した値と競合しない値に変更してください。

※ 今回の例は vi コマンドを使用して編集を行います。

```
# /bin/vi /opt/mcl/rens/conf/sps.conf
```

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

13. モニタプロセス起動スクリプトファイルを編集して RENS との連携設定を行います。

※ 今回の例は vi コマンドを使用して編集を行います。

```
# /bin/vi /opt/mcl/rens/script/monitor_run.sh
```

```
#!/bin/sh
```

```
RENS_HOME=/opt/mcl/rens
```

```
RENS_BIN=${RENS_HOME}/bin
```

```
RENS_SCRIPT=${RENS_HOME}/script
```

```
#{RENS_BIN}/targetregclpd -f /opt/mcl/rens/conf/targetclp.conf &
```

```
#{RENS_SCRIPT}/check_target.sh -t clpx
```

```
#{RENS_BIN}/nicmon -n e1000 &
```

```
#{RENS_BIN}/fcmon -n lpfc &
```

```
#{RENS_BIN}/spsmon -n sps &
```

RENS と StoragePathSavior の
連携に必要な設定を追記

```
exit 0
```

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

14. パッケージ適用のため OS を再起動します。

```
# /bin/sync
```

```
# /sbin/shutdown -r now
```

※ StoragePathSavior 用モニタプロセス(spsmon)登録後のOS再起動に伴う RENS 初回起動時のみ syslog へ以下のメッセージが出力されますが、動作には問題ありませんので無視してください。
--RENS-- WARNING: Cannot read expected data size:
key of RENS(lower) temporary buffer: path=/opt/mcl/rens/conf/lower/tmpbuf/event/sps.key, size=0/10

15. 以上で StoragePathSavior のインストールは終了です。

2. 構築手順

ii. StoragePathSaviorによるI/Oパスの冗長化

② LVM 設定

StoragePathSavior デバイスを LVM で利用するために以降の手順を実施します。

1. OS に認識されている SCSI ディスクを確認します。

```
# /usr/bin/sg_scan -i /dev/sd*  
/dev/sda: scsi0 channel=0 id=0 lun=0 [em]  
  [ NEC      iStorage xxxx ] yyyy[rmb=0 cmdq=1 pqual=0 pdev=0x0]  
/dev/sdb: scsi0 channel=0 id=0 lun=1 [em]  
  [ NEC      iStorage xxxx ] yyyy[rmb=0 cmdq=1 pqual=0 pdev=0x0]  
      <省略>
```

“NEC”、“iStorage xxxx”と表示されれば iStorage のデバイスと判断できます。

xxxx には iStorage 型番が、yyyy にはデバイス Rev が表示されます。

2. LVM 設定ファイルを変更する前にバックアップを作成します。

```
# /bin/cp -p /etc/lvm/lvm.conf /etc/lvm/lvm.conf.sps
```

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

3. LVM を使用するために LVM 設定ファイルを編集します。

※ 今回の例は vi コマンドを使用して編集を行います。

```
# /bin/vi /etc/lvm/lvm.conf
```

```
# This section allows you to configure which block devices should
```

```
# be used by the LVM system.
```

```
devices {
```

<省略>

```
# By default we accept every block device:
```

```
filter = [ "a|/dev/dd.*|", "r|/dev/.*|" ] ①
```

<省略>

```
# Advanced settings.
```

```
# List of pairs of additional acceptable block device types found
```

```
# in /proc/devices with maximum (non-zero) number of partitions.
```

```
# types = [ "fd", 16 ]
```

```
types = [ "dd", 16 ] ②
```

<省略>

① filter 行の "a" は accept(許可)、"r" は reject(除外)を意味します。
dd デバイスを最大数(256)まで作成できるよう、" a|/dev/dd.*| " で許可し、それ以外のデバイスを対象外にするために、" r|/dev/.*| " で除外します。

ローカルディスクを LVM 領域として使用する場合はそのデバイスも記述する必要があります。

例として sde1 に対して行う許可を行う場合は、" a|^/dev/sde1\$" と記述を行います。

② types 行を追加します。

※ LVM の仕様上、accept より前に、すべてを reject 対象とする "r | .*|" を記述すると、以降のデバイスを accept できないので reject 設定は必ず最後に記述してください。

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

4. StoragePathSavior デバイスが LVM から認識されたことを確認します。

```
# /usr/sbin/lvmdiskscan  
/dev/dda [    2.00 GB]  
/dev/ddb [    2.00 GB]  
2 disks  
0 partitions  
0 LVM physical volume whole disks  
0 LVM physical volumes
```

5. 以上で LVM 設定は終了です。

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

③ 構成変更時の設定

StoragePathSavior インストール後の OS 再起動にて監視ディスク設定ファイル (sps.conf) は自動生成されるため、通常インストール後はこの手順を行う必要はありません。

自動生成に失敗した場合や、構成の変更があった場合のみ、この手順を行ってください。

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

1. パス巡回デーモンの起動を確認します。

```
# /etc/init.d/dd_daemon status  
dd_daemon (pid xxxxx) is running...
```

2. パス巡回デーモンを停止させます。

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

3. パス巡回デーモンが停止していることを確認します。

```
# /etc/init.d/dd_daemon status  
dd_daemon is stopped
```

4. 監視ディスク設定ファイルを新たに作成するために既存のファイルを削除します。

```
# /bin/rm /etc/sps.conf
```

2. 構築手順

ii. StoragePathSavior による I/O パスの冗長化

5. 監視ディスク設定ファイルが削除されていることを確認します。

```
# /bin/ls /etc/sps.conf
```

6. 構成の変更を反映させるため mkdd コマンドを実行します。

```
# /sbin/mkdd
```

7. 新しく監視ディスク設定ファイルが作成されていることを確認します。

```
# /bin/ls /etc/sps.conf
```

8. StoragePathSaviorデーモンを起動します。

```
# /etc/init.d/dd_daemon start
```

9. 以上で構成変更時の設定は終了です。

2. 構築手順

iii. 構成情報の確認

以降の手順を実行する上で必要な構成情報を確認します。

1. udev デバイスパスの情報を確認します。

```
# /bin/ls -l /dev/disk/by-path/
```

```
lrwxrwxrwx xx root root xx xx xx xx: xx pci-0000:13:00.0-fc-0x2100001697120ca7:0x0000000000000000 -> ../../sda
```

```
lrwxrwxrwx xx root root xx xx xx xx: xx pci-0000:15:00.0-fc-0x2100001697120ca7:0x0000000000000000 -> ../../sdb
```

```
lrwxrwxrwx xx root root xx xx xx xx: xx pci-0000:13:00.0-fc-0x2100001697120ca7:0x0001000000000000 -> ../../sdc
```

```
lrwxrwxrwx xx root root xx xx xx xx: xx pci-0000:15:00.0-fc-0x2100001697120ca7:0x0001000000000000 -> ../../sdd
```

2. sg_scan コマンドでデバイス情報を確認します。

※ 以下の例は /dev/sda を対象としています。

```
# /usr/bin/sg_scan /dev/sda
```

```
/dev/sda: scsi0 channel=0 id=0 lun=0 [em]
```

2. 構築手順

iii. 構成情報の確認

3. 手順2 の内容と StoragePathSavior デバイスファイルの内容が一致していることを確認します。

```
# /bin/cat /proc/scsi/sps/dd*
```

```
device:/dev/dda
```

< 省略 >

```
path-info:0 Host:scsi:0 Channel:0 Id:0 Lun:0 Priority:1 Watch:Enable Status:ACT
```

< 省略 >

手順2 で確認した sda のデバイスファイルの内容と一致しているため、/dev/sda が /dev/dda の多重化経路の内の一つであることが確認できます。

2. 構築手順

iv. LVM 環境構築、ファイルシステム作成

システムでの監視対象ディスクを論理ボリュームとして扱うために LVM 環境構築を行います。

この手順を行うことでディスク領域を無駄なく使用することができます。

ここではそのLVM環境の構築手順を、Disk2(/dev/ddb) に対して行う場合について記載します。

※ LVM 環境構築を行わない場合この手順を行う必要はありません。

各 Server から見たディスク構成

Disk名	Disk2	
Disk サイズ	2GB	
I/O パス	/dev/sdc	/dev/sdd
SPS(※1) デバイス ファイル	/dev/ddb	
パーティション数	0	
パーティション名	-	
LVM	○	
VG 名	VolGroup001	
LV 名	lvol0	
LV サイズ	1GB	

※1 "SPS"は、" StoragePathSavior"の略称です。

2. 構築手順

iv. LVM 環境構築、ファイルシステム作成

LVM 環境の構築を行います。

1. PV を作成します。

※ 以下の例では /dev/ddb を対象として PV を作成します。

```
# /usr/sbin/pvcreate -f /dev/ddb
```

```
Physical volume "/dev/ddb" successfully created
```

■ オプション説明

-f : PV を強制的に作成します。

2. 構築手順

iv. LVM 環境構築、ファイルシステム作成

2. 作成した PV 情報の確認を行います。

```
# /usr/sbin/pvdisplay /dev/ddb  
--- Physical volume ---  
PV Name           /dev/ddb  
VG Name  
PV Size           xxx GB  
Allocatable       NO  
PE Size (KByte)   0  
Total PE          0  
Free PE           0  
Allocated PE      0  
PV UUID           XXXXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXXXXX
```

2. 構築手順

iv. LVM 環境構築、ファイルシステム作成

3. 作成した PV を使用して VG を構築します。

※ 以下の例では VG 名は VolGroup001 として構築します。

```
# /usr/sbin/vgcreate -s 4m VolGroup001 /dev/ddb  
Volume group "VolGroup001" successfully created
```

■ オプション説明

-s : 構築する VG の物理エクステントサイズを指定します。
デフォルト値は 4MB です。

4. VG 情報の確認を行います。

```
# /usr/sbin/vgdisplay -v VolGroup001  
Finding volume group "VolGroup001"  
--- Volume group ---  
VG Name          VolGroup001  
  < 省略 >  
--- Physical volumes ---  
PV Name          /dev/ddb  
  < 省略 >
```

2. 構築手順

iv. LVM 環境構築、ファイルシステム作成

5. LV を作成します。

※ 以下の例では LV のサイズは 1GB で作成します。

```
# /usr/sbin/lvcreate -L 1G VolGroup001  
Logical volume "lvol0" created
```

■ オプション説明

-L : 作成する LV のバイトサイズを指定します。
サイズを指定せずにコマンドを実行することはできません。

6. LV 情報を確認します。

```
# /usr/sbin/lvdisplay /dev/VolGroup001/lvol0  
--- Logical volume ---  
LV Name           /dev/VolGroup001/lvol0  
VG Name           VolGroup001  
< 省略 >
```

2. 構築手順

iv. LVM 環境構築、ファイルシステム作成

7. LV にファイルシステムを作成します。

※ 以下の例ではファイルシステムの形式を ext3 に指定します。

```
# /sbin/mkfs -t ext3 /dev/VolGroup001/lvol0
```

■ オプション説明

-t : 作成するファイルシステムの形式を指定します。

8. 以上で LVM 環境構築は終了です。

2. 構築手順

V. 共有ディスクの設定

CLUSTERPRO にディスクハートビートリソースを追加するため、ディスクハートビート用のパーティションとディスクリソース用のパーティションを用意する必要があります。

1. パーティションを作成するためにデバイスを指定して fdisk コマンドを実行します。

※ 以下の例では dda を対象にパーティションを作成します。

```
# /sbin/fdisk /dev/dda
```

2. 新しくパーティションを作成するために”n”と入力します。

コマンド(mでヘルプ): n

“n” と入力し ENTER キーを押します

2. 構築手順

V. 共有ディスクの設定

3. 基本パーティションを作成するために”p”と入力します。

コマンドアクション

e 拡張

p 基本領域 (1-4)

p

“p” と入力し ENTER キーを押します

4. パーティション1を作成するため、“1”と入力します。

※ パーティション 2 の作成時は”2”を入力します。

領域番号 (1-4):1

“1” と入力し ENTER キーを押します

2. 構築手順

V. 共有ディスクの設定

5. 作成するパーティションの始点を決めます。何も入力せずに次へ進めます。

※ 手順 5,6 の xxxx にはディスクのシリンダ数が表示されます。シリンダ数はディスクによって異なります。

最初 シリンダ (1-xxxx, default 1):

何も入力せずに ENTER キーを押します

6. パーティションの終点を決めます。"+10M"と入力します。

これでディスクハートビート用パーティション(dda1)の作成が完了します。

※ パーティション 2 の作成時はディスク終端までを領域として使用するため、終点を指定せずに作成します。

Using default value 1

終点シリンダ または +サイズ または +サイズM または +サイズK (1-xxxx, default xxxx):+10M

"+10M" と入力し ENTER キーを押します

7. 同様に手順 2 ~ 6 までを行い、ディスクリソース用パーティション(dda2)を作成します。

2. 構築手順

V. 共有ディスクの設定

8. パーティションの作成が完了したらパーティション情報を確認します。

コマンド(mでヘルプ): p

“p” と入力し ENTER キーを押します

Disk /dev/dda: xxxx MB, xxxxxxxxxxx bytes

xx heads, xx sectors/track, xx cylinders

Units = シリンダ数 of xxxx * xx = xx bytes

デバイス	Boot	Start	End	Blocks	Id	System
/dev/dda1		1	xx	xx	83	Linux
/dev/dda2		xx	xx	xx	83	Linux

9. 設定を保存し、fdisk コマンドを終了します。

コマンド(mでヘルプ): w

“w” と入力し ENTER キーを押します

2. 構築手順

V. 共有ディスクの設定

10. dda2 にファイルシステムを作成します。

```
# /sbin/mkfs -t ext3 /dev/dda2
```

■ オプション説明

-t : 作成するファイルシステムの形式を指定します。

11. CLUSTERPRO 連携でディスクハートビートとして利用するために /dev/dda1 の RAW デバイス /dev/raw/raw1 を作成してください。

12. 以上で共有ディスクの設定は終了です。

2. 構築手順

vi. StorageSaver の監視定義

StorageSaver は、サーバと iStorage ディスクアレイ装置を接続したシステムにおいて、iStorage ディスクアレイ装置の I/O パスの動作状態を定期監視します。

I/O パスの異常を検出すると障害レポートを通知し、さらに iStorage ディスクアレイ装置へのアクセスができなくなるとクラスタウェアと連携しノードを切り替えることでクラスタシステムでの可用性を向上させます。

ここではその StorageSaver の導入手順を記載します。

2. 構築手順

vi. StorageSaver の監視定義

① StorageSaver のインストール

1. StorageSaver のインストールを行うために、StorageSaver のパッケージが含まれる CD-ROM 媒体を mount します。

※ CD-ROM パスが /dev/cdrom, mount 先が /mnt/cdrom の場合

```
# /bin/mount /dev/cdrom /mnt/cdrom
```

2. rpm コマンドにより、StorageSaver パッケージをインストールします。

```
# /bin/rpm -ivh /mnt/cdrom/Linux/rpm/clusterpro-mc-ss-x.x.x-x.  
xxxx.rpm
```

3. RENS と連携させるために以下パッケージもインストールします。

※ 一般 Linux サーバを使用する場合 RENS との連携を行わないのでこの手順を行う必要はありません。

```
# /bin/rpm -ivh /mnt/cdrom/Linux/rpm/clusterpro-mc-ss-rens-x.x.x-  
x.xxx_xx.rpm
```

2. 構築手順

vi. StorageSaver の監視定義

4. パッケージがインストールされていることを確認します。

※ RENS 連携を行う場合のみ”clusterpro-mc-ss-rens-x.x.x-x”は表示されます。

```
# /bin/rpm -qa | /bin/grep clusterpro-mc-ss  
clusterpro-mc-ss-x.x.x-x  
clusterpro-mc-ss-rens-x.x.x-x
```

5. インストール CD-ROM をumount して、媒体を取り出します。

```
# /bin/umount /dev/cdrom/
```

6. 以上で StorageSaver のインストールは終了です。

2. 構築手順

vi. StorageSaver の監視定義

② StorageSaver の設定

1. ディスク監視に必要な SG ファイルの作成を行います。

以下コマンドで SG ファイルのテンプレートを自動生成します。

※ StorageSaver は基本的にディスクを LVM 構成として内部管理しているため、
LVM 構成ではないディスクは内部的には擬似的な VG 構成(PSEUDO_VG)として管理されます。

```
# /opt/HA/SrG/bin/srgquery -s <格納ディレクトリ>
```

■ オプション説明

-s : SG ファイルを作成するディレクトリを指定します。

指定のない場合、カレントディレクトリに作成します。

※ 指定したディレクトリ内には『srg.map』『srg.rsc』『srg.config』が作成されます。
これらのファイルは一時ファイルですので、実行環境への適用後に削除してください。

2. 構築手順

vi. StorageSaver の監視定義

2. RENS との連携機能を設定するために、システム定義ファイル(srg.config)のパラメタを変更します。
変更するパラメタは、VG 異常検出時のアクションを設定する『VG_FAULT_ACTION』であり、
設定可能な値を下表に示します。

パラメタ	設定値	動作詳細
VG_FAULT_ACTION	RENS_REPORT_ENABLE	RENS との連携を行います
	ACTION_NONE	アクション指定なし
	CLPNM_KILL_ENABLE	clpnm を強制終了させ、ノードを切り替えます
	TOC_EXEC	ダンプを採取し、OS を強制停止させ、 ノードを切り替えます

変更する設定値はスケーラブル HA サーバ構成の場合とその他一般 Linux サーバ構成の場合で異なります。それぞれの設定値を以降に記載します。

iStorage ディスクアレイ装置を使用する構成の場合は TESTIO_FAULT_ACTION パラメタを BLOCK_PATH に設定することで I/O パスの自動閉塞機能を有効にすることができます。
自動閉塞機能を有効にすることにより、間欠的な FC リンクダウン障害に伴う頻繁なパス切り替えによる I/O パフォーマンスの低下や、他のディスク装置への影響を未然に防止することができます。

※ TESTIO_FAULT_ACTION では TestI/O 異常検出時のアクションを設定することができます。

自動閉塞機能の設定手順詳細については
『CLUSTERPRO MC StorageSaver 1.0 for Linux ユーザーズガイド』を参照してください。

2. 構築手順

vi. StorageSaver の監視定義

スケーラブル HA サーバ構成

スケーラブル HA サーバ構成は RENS との連携を行いますのでシステム定義ファイル(srg.config)内のパラメタを” RENS_REPORT_ENABLE”に変更します。

※ 今回の例は vi コマンドを使用して編集を行います。

```
# /bin/vi <SG ファイルの格納ディレクトリ>/srg.config
```

< 省略 >

```
#####
```

```
# User Config Area
```

```
#####
```

< 省略 >

```
VG_FAULT_ACTION
```

```
RENS_REPORT_ENABLE
```

< 省略 >

『ACTION_NONE』
から
『RENS_REPORT_ENABLE』
に変更します。

2. 構築手順

vi. StorageSaver の監視定義

その他一般 Linux サーバ構成

その他一般 Linux サーバ構成は RENS との連携を行いませんのでシステム定義ファイル (srg.config) 内のパラメタを変更する必要はありません。

2. 構築手順

vi. StorageSaver の監視定義

3. SG ファイルの妥当性の確認を行います。

```
# /opt/HA/SrG/bin/srgconfig -c -s <SG ファイルの格納ディレクトリ>  
srgconfig:sg check complete
```

■ オプション説明

- s : 対象となる SG ファイルのディレクトリを指定します。
- c : 指定されたディレクトリにある SG ファイルの妥当性、整合性をチェックします。

4. SG ファイルの実行環境への適用を行います。

```
# /opt/HA/SrG/bin/srgconfig -a -s <SG ファイルの格納ディレクトリ>
```

※ 実行環境への適用完了後、指定したディレクトリ内に作成していた SG ファイルを削除してください。

■ オプション説明

- a : 指定されたディレクトリにある SG ファイルを実行環境に適用します。

2. 構築手順

vi. StorageSaver の監視定義

5. デーモンプロセス(srgd)を起動します。

```
# /etc/init.d/srgctl start
```

6. デーモンプロセス(srgd)が起動していることを確認します。

※ リソース監視モニタ(srgping)、プロセス監視デーモン(srgwatch)についても同様に確認します。
srgping は構成によっては起動までに時間がかかることがあります。

```
# /bin/ps -ef | /bin/grep srg  
/opt/HA/SrG/local/bin/srgwatch  
/opt/HA/SrG/bin/srgd  
srgping
```

7. 以上で StorageSaver の設定は終了です。

2. 構築手順

vi. StorageSaver の監視定義

③ StorageSaver と RENS の連携設定

※ 一般 Linux サーバを使用する場合 RENS との連携を行わないのでこの手順を行う必要はありません。

1. ssdiagd の辞書ファイルを登録します。

```
# /bin/cp /var/opt/HA/SrG/conf/rens/* /opt/mcl/rens/dict/
```

2. RENS SW イベント監視コンポーネント設定ファイルを作成します。

```
# /bin/cp /opt/mcl/rens/conf/lower/buffer/monitor_buf.conf.template  
/opt/mcl/rens/conf/lower/buffer/ssdiagd_buf.conf
```

3. モニタプロセス設定ファイルを作成します。

```
# /bin/cp /opt/mcl/rens/conf/monitor.conf.template  
/opt/mcl/rens/conf/ssdiagd.conf
```

2. 構築手順

vi. StorageSaver の監視定義

4. 他のモニタプロセス設定ファイルの shm_key 値を取得します。

```
# /bin/cat /opt/mcl/rens/conf/*.conf | /bin/grep "shm_key"
```

5. ssdiagd.conf の shm_key 値を上記コマンドで取得した値と競合しない値に変更してください。

※ 今回の例は vi コマンドを使用して編集を行います。

```
# /bin/vi /opt/mcl/rens/conf/ssdiagd.conf
```

2. 構築手順

vi. StorageSaver の監視定義

6. モニタプロセス起動スクリプトファイルを編集して RENS との連携設定を行います。

※ 今回の例は vi コマンドを使用して編集を行います。

```
# /bin/vi /opt/mcl/rens/script/monitor_run.sh

#!/bin/sh
RENS_HOME=/opt/mcl/rens
RENS_BIN=${RENS_HOME}/bin
RENS_SCRIPT=${RENS_HOME}/script

# ${RENS_BIN}/targetregclpd -f /opt/mcl/rens/conf/targetclp.conf &
# ${RENS_SCRIPT}/check_target.sh -t clpx

# ${RENS_BIN}/nicmon -n e1000 &
# ${RENS_BIN}/fcmon -n lpfc &
# ${RENS_BIN}/spsmon -n sps &
# ${RENS_BIN}/hamon -n ssdiagd &

exit 0
```

RENS と StorageSaver の
連携に必要な設定を追記

2. 構築手順

vi. StorageSaver の監視定義

7. ssdiagd を起動します。

```
# /etc/init.d/ssdiagctl start
```

※ ssdiagd 用モニタプロセス(hamon)登録後の ssdiagd 初回起動時のみsyslogへ以下のメッセージが出力されますが、動作には問題ありませんので無視してください。

```
--RENS-- WARNING: Cannot read expected data size: key of RENS(lower) temporary buffer:  
path=/opt/mcl/rens/conf/lower/tmpbuf/event/ssdiagd.key, size=0/10
```

8. RENS 連携用モニタプロセス(ssdiagd)と、モニタプロセス(hamon)が起動していることを確認します。

```
# /bin/ps -ef | /bin/grep ssdiagd  
/opt/HA/SrG/bin/ssdiagd -c 60  
/opt/mcl/rens/bin/hamon -n ssdiagd
```

9. 以上で StorageSaver と RENS の連携設定は終了です。

2. 構築手順

vii. CLUSTERPRO 連携設定

CLUSTERPRO のインストール

※ 以降の手順は、CLUSTERPRO X3.1 における設定手順です。

1. CLUSTERPRO のインストールを行うために、CLUSTERPRO のパッケージが含まれる CD-ROM 媒体を mount します。

※ CD-ROM が” /dev/cdrom”、mount 先が”/mnt/cdrom”の場合

```
# /bin/mount /dev/cdrom /mnt/cdrom
```

2. rpm コマンドにより、CLUSTERPRO パッケージをインストールします。

```
# /bin/rpm -ivh /mnt/cdrom/Linux/3.1/jp/server/clusterpro-xxxx.xxx.rpm
```

2. 構築手順

vii. CLUSTERPRO 連携設定

3. パッケージがインストールされていることを確認します。

```
# /bin/rpm -qa | /bin/grep clusterpro  
clusterpro-xxxx.xxx
```

4. インストール CD-ROM を umount して、媒体を取り出します。

```
# /bin/umount /dev/cdrom
```

2. 構築手順

vii. CLUSTERPRO 連携設定

5. ライセンスファイルを登録します。

※ *filepath* には、ライセンスファイルへのファイルパスを指定してください。

```
# /usr/sbin/clplcnsd -i filepath -p BASE31
```

Command succeeded.

6. ライセンスの登録情報を確認します。

```
# /usr/sbin/clplcnsd -l -p BASE31
```

※ 手順詳細は CLUSTERPRO インストール & 設定ガイドを参照してください。

2. 構築手順

vii. CLUSTERPRO 連携設定

7. ライセンスを有効にするために OS を再起動します。

```
# /sbin/shutdown -r now
```

8. 以上で CLUSTERPRO インストールは終了です。

2. 構築手順

vii. CLUSTERPRO 連携設定

設定ファイル変更

1. モニタプロセス起動スクリプトファイルを編集して RENS との連携設定を行います。

※ 今回の例は vi コマンドを使用して編集を行います。

```
# /bin/vi /opt/mcl/rens/script/monitor_run.sh
```

```
#!/bin/sh
```

```
RENS_HOME=/opt/mcl/rens
```

```
RENS_BIN=${RENS_HOME}/bin
```

```
RENS_SCRIPT=${RENS_HOME}/script
```

```
{RENS_BIN}/targetregclpd -f /opt/mcl/rens/conf/targetclp.conf &
```

```
{RENS_SCRIPT}/check_target.sh -t clpx
```

```
#{RENS_BIN}/nicmon -n e1000 &
```

```
{RENS_BIN}/fcmon -n lpfc &
```

```
{RENS_BIN}/spsmon -n sps &
```

```
{RENS_BIN}/hamon -n ssdiagd &
```

```
exit 0
```

CLUSTERPRO との連携設定を有効にするために行頭の”#”を削除します。

2. 構築手順

vii. CLUSTERPRO 連携設定

2. RENS との連携設定を有効にするために RENS を再起動します。

```
# /opt/mcl/rens/bin/rensadmin stop
```

```
Really? [Y/N] y
```

```
RENS stopped.
```

“y” と入力し ENTER キーを押下

※ 上記コマンド実行後、RENS は自動的に再起動します。

3. 以上で設定ファイル変更は終了です。

2. 構築手順

vii. CLUSTERPRO 連携設定

クラスタ環境構築

クラスタ構成情報の作成は、クラスタ構成情報の作成 / 変更用の機能である

CLUSTERPRO Builder (以下 Builder) を用いて行います。

管理用クライアントからアクセスしたCLUSTERPRO WebManager (以下 WebManager) から Builder を起動し、クラスタ構成情報を作成します。作成したクラスタ構成情報は、Builder からクラスタシステムに反映します。

※ 以降の手順は、両系にて『2 - i サーバ管理基盤(RENS)セットアップ』から『2 - vii CLUSTERPRO 連携設定』の設定ファイル変更までを実施した上で、管理用クライアントにて行ってください。

2. 構築手順

vii. CLUSTERPRO 連携設定

① WebManager の起動

1. Web ブラウザのアドレスバーに CLUSTERPRO Server をインストールしたサーバの実 IP アドレスとポート番号を入力します。

※ 今回の例では Server1(現用系)のアドレスとポート番号を入力します。

http://10.0.0.1:29003/

インストール時に指定した WebManager
のポート番号(既定値 29003)

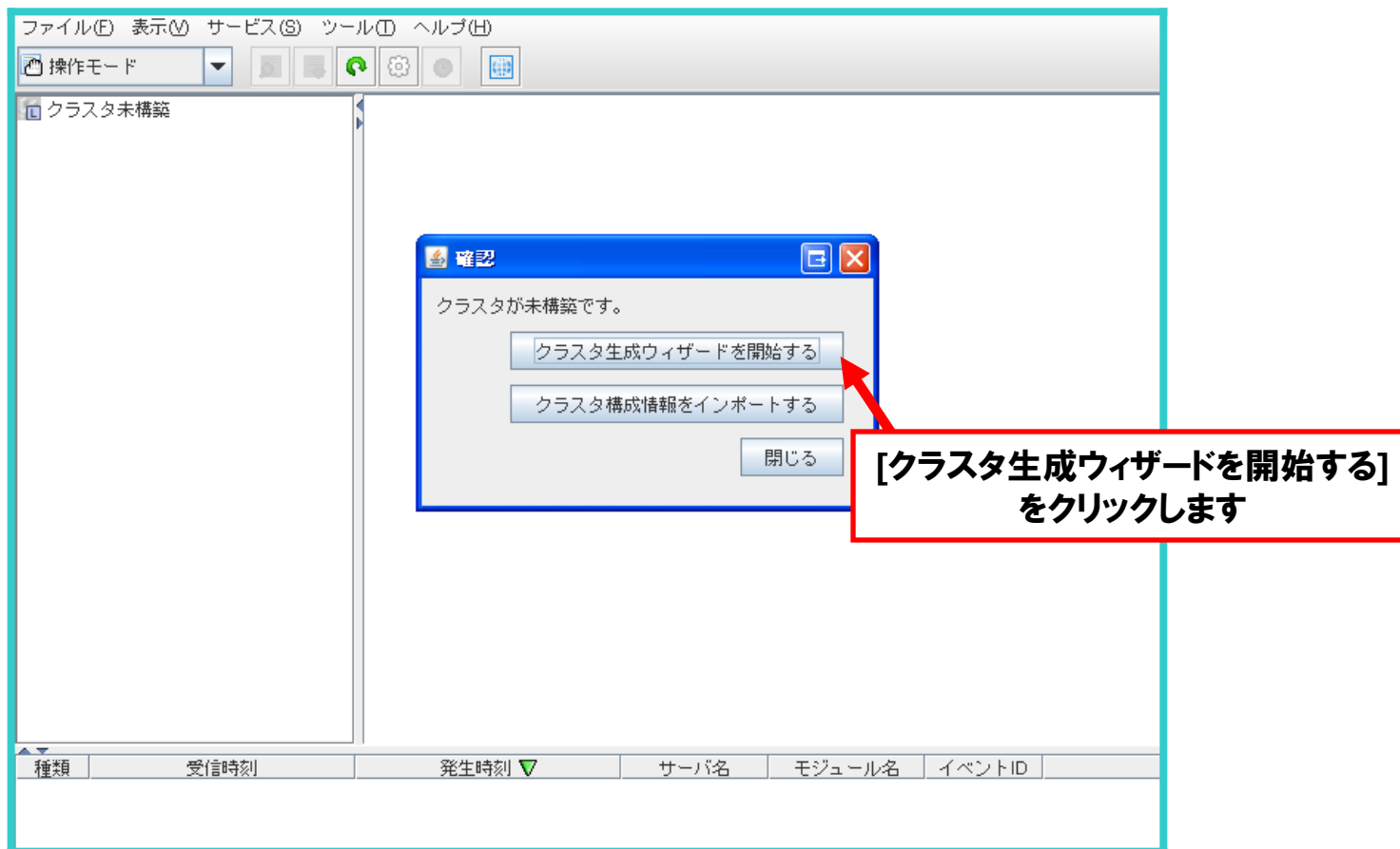
CLUSTERPRO Server をインストールし
たサーバの実 IP アドレス

2. 構築手順

vii. CLUSTERPRO 連携設定

② クラスタの生成

1. WebManager を起動すると Builder の画面に遷移します。
Builder の画面から [クラスタ生成ウィザードを開始する] をクリックします。



2. 構築手順

vii. CLUSTERPRO 連携設定

2. クラスタ生成ウィザードが開始されるのでクラスタ情報を入力します。

[クラスタ名] にクラスタ名 (cluster) を入力します。

[言語] フィールドに WebManager を使用するマシンの OS で使用している言語を選択し、[次へ] をクリックします。

クラスタ生成ウィザード

ステップ

- クラスタ
- サーバ
 - 基本設定
 - インタコネクト
 - NP解決
 - グループ
 - モニタ

クラスタの定義

クラスタ名(M) cluster

コメント(C)

言語(L) 日本語

管理IPアドレス(I)

説明

クラスタの生成を開始します。
クラスタの名前を入力して、WebManagerを動作させる環境の言語（ロケール）を選択してください。
統合WebManagerで複数のクラスタを管理する場合、クラスタ名でクラスタを識別するため、重複しない名前を設定してください。
管理IPアドレスはWebManagerの接続に使用するフローティングIPアドレスです。各サーバのIPアドレスを指定して接続する場合は省略可能です。
継続するには[次へ]をクリックしてください。

< 戻る(B) 次へ(N) > キャンセル

クラスタ名を入力します

OS 使用言語を選択します

[次へ] をクリックします

2. 構築手順

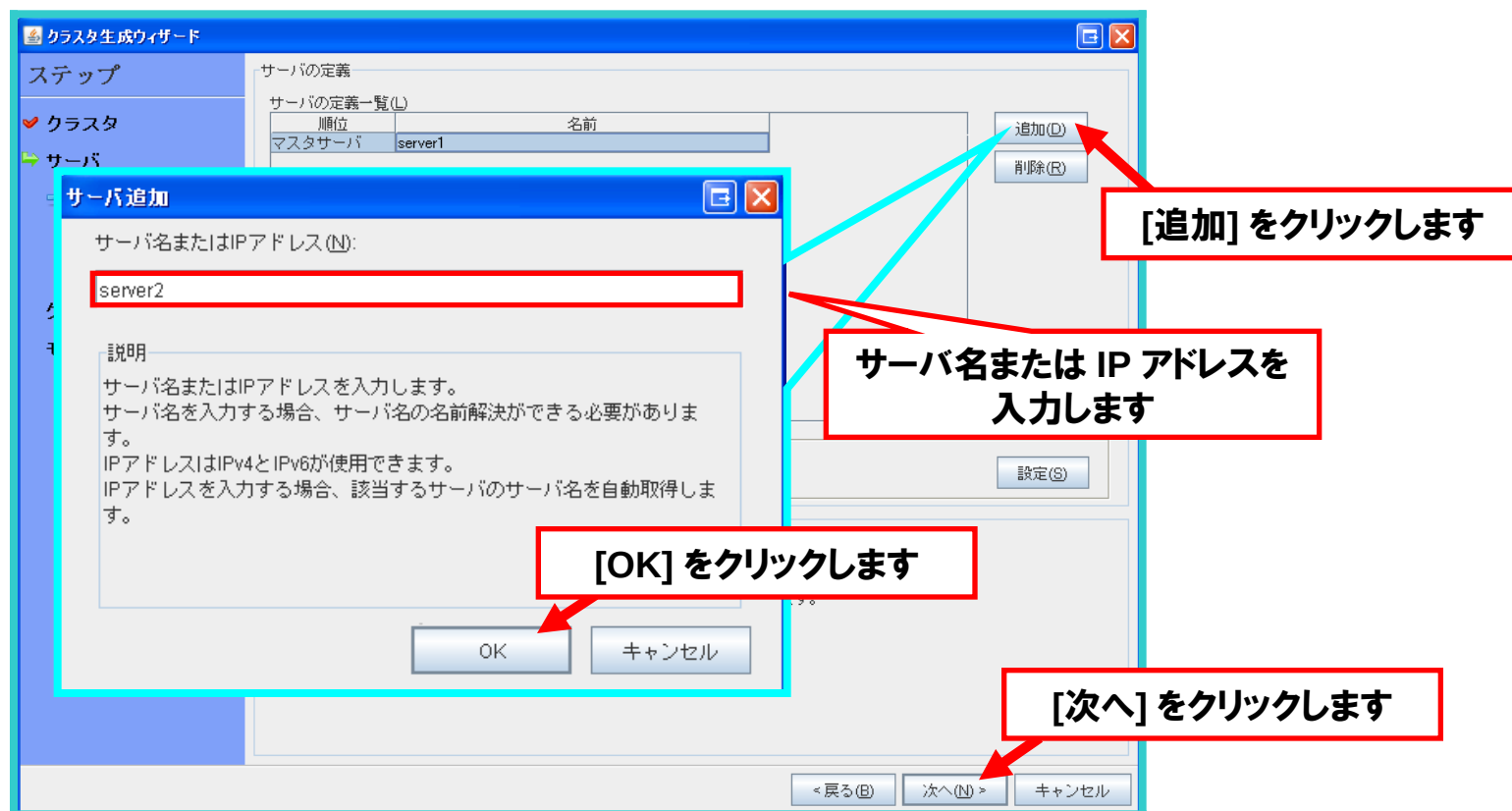
vii. CLUSTERPRO 連携設定

3. サーバ(2 台目)を追加します。

WebManager でアクセスしたサーバ (server1)は自動で登録されます。

[サーバ名またはIPアドレス] に 2 台目のサーバ名 (server2) または IP アドレス (10.0.0.2)を入力し、[OK] をクリックします。
入力したサーバが [サーバの定義一覧] に設定されていることを確認し、[次へ] をクリックします。

※ サーバ名は、実際のサーバのホスト名です。また、大文字と小文字は区別されます。

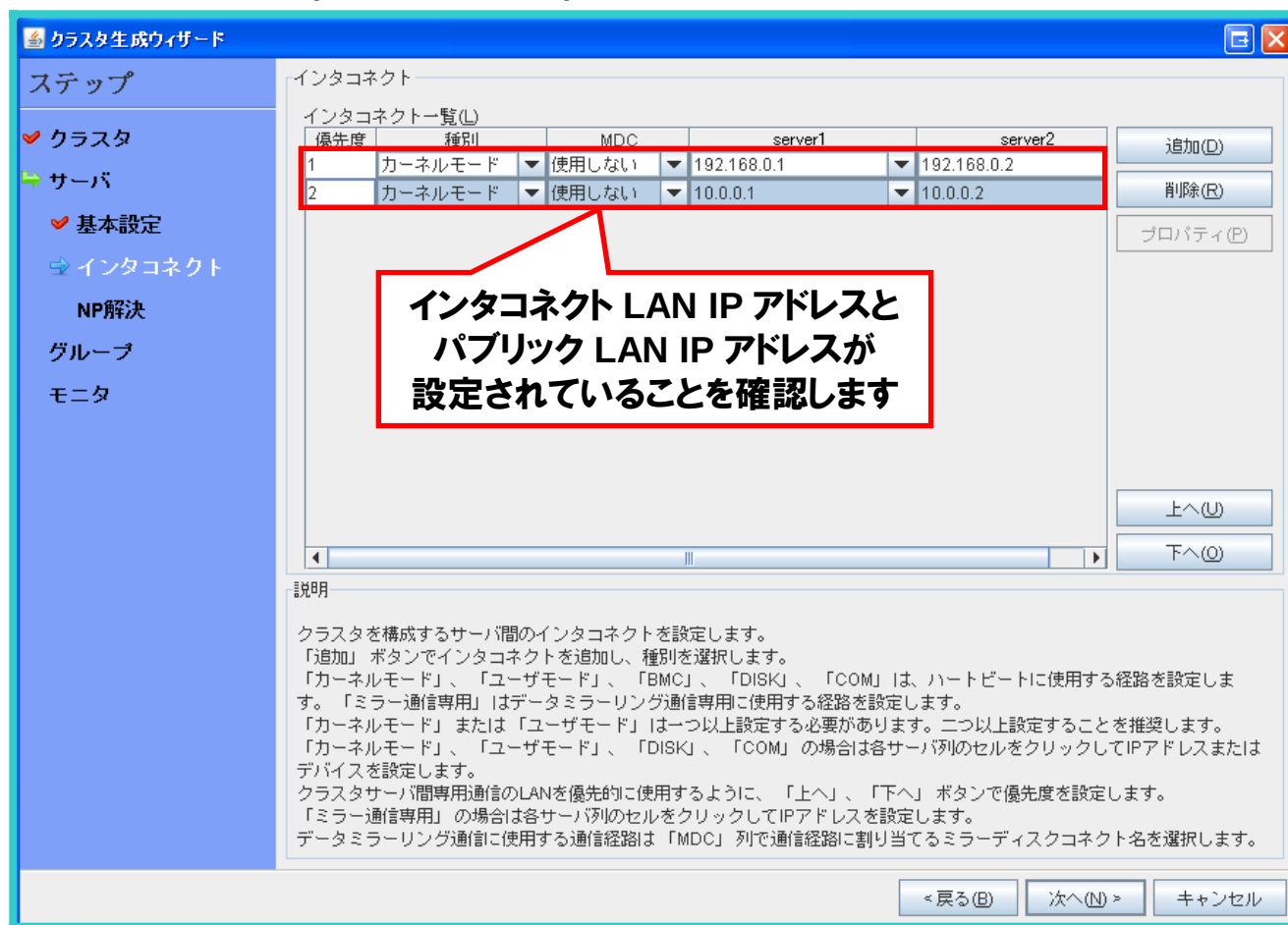


2. 構築手順

vii. CLUSTERPRO 連携設定

4. インタコネクト LAN とパブリック LAN の設定を行います。

[インタコネクト一覧] に、各サーバ (server1, server2) のインタコネクト LAN IP アドレス (192.168.0.1, 192.168.0.2) とパブリック LAN IP アドレス (10.0.0.1, 10.0.0.2) が設定されていることを確認します。



2. 構築手順

vii. CLUSTERPRO 連携設定

5. ディスクハートビートリソースの設定を行います。

[追加] をクリックし、[種別] フィールドに DISK を選択し、各サーバ (server1, server2) のフィールドに実デバイス名 (/dev/dda1) を選択します。

クラスタ生成ウィザード

ステップ

- クラスタ
- サーバ
- 基本設定
- インタコネク
- NP解決
- グループ
- モニタ

インタコネク

インタコネクター一覧(L)

優先度	種別	MDC	server1	server2
1	カーネルモード	使用しない	192.168.0.1	192.168.0.2
2	カーネルモード	使用しない	10.0.0.1	10.0.0.2
3	DISK	使用しない	/dev/dda1	/dev/dda1

追加(D) 削除(R) プロパティ(P)

上へ(U) 下へ(D)

説明

クラスタを構成するサーバ間のインタコネクを設定します。
「追加」ボタンでインタコネクを追加し、種別を選択します。
「カーネルモード」、「ユーザモード」、「BMC」、「DISK」、「COM」は、ハートビートに使用する経路を設定します。
「ミラー通信専用」はデータミラーリング通信専用使用する経路を設定します。
「カーネルモード」または「ユーザモード」は一つ以上設定する必要があります。二つ以上設定することを推奨します。
「カーネルモード」、「ユーザモード」、「DISK」、「COM」の場合は各サーバ列のセルをクリックしてIPアドレスまたはデバイスを設定します。
クラスタサーバ間専用通信のLANを優先的に使用するよう、「上へ」、「下へ」ボタンで優先度を設定します。
「ミラー通信専用」の場合は各サーバ列のセルをクリックしてIPアドレスを設定します。
データミラーリング通信に使用する通信経路は「MDC」列で通信経路に割り当てるミラーディスクコネク名を選択します。

< 戻る(B) 次へ(N) > キャンセル

[追加] をクリックします

DISK を選択します

実デバイス名を選択します

2. 構築手順

vii. CLUSTERPRO 連携設定

[プロパティ] をクリックし、各サーバ (server1, server2) の [Rawデバイス] フィールドに Raw アクセスするためのデバイス名 (/dev/raw/raw1) を選択します。
[OK] をクリックし、[次へ] をクリックします。

クラスタ生成ウィザード

ステップ

- クラスタ
- サーバ
- 基本設定
- インタコネクト
- NP解決
- グループ
- モニタ

インタコネクト

インタコネクター一覧(L)

優先度	種別	MDC	server1	server2
1	カーネルモード	使用しない	192.168.0.1	192.168.0.2
2	カーネルモード	使用しない	10.0.0.1	10.0.0.2
3	DISK	使用しない	/dev/dda1	/dev/dda1

追加(D)
削除(R)
プロパティ(P)
上へ(U)
下へ(D)

DISKハートビートのプロパティ

DISKデバイス一覧(L)

サーバ	デバイス	Rawデバイス
server1	/dev/dda1	/dev/raw/raw1
server2	/dev/dda1	/dev/raw/raw1

RAW アクセス用デバイス名を選択します

OK キャンセル

ビートに使用する経路を設定しま
以上設定することを推奨します。
セルをクリックしてIPアドレスまたはは

ように、「上へ」、「下へ」ボタンで優先度を設定し
クリックしてIPアドレスを設定します。
サーバミラーリング通信に使用する通信経路は「MDC」列で通信経路に割り当てるミラーディスクコネク

< 戻る(B) 次へ(N) > キャンセル

[プロパティ] をクリックします

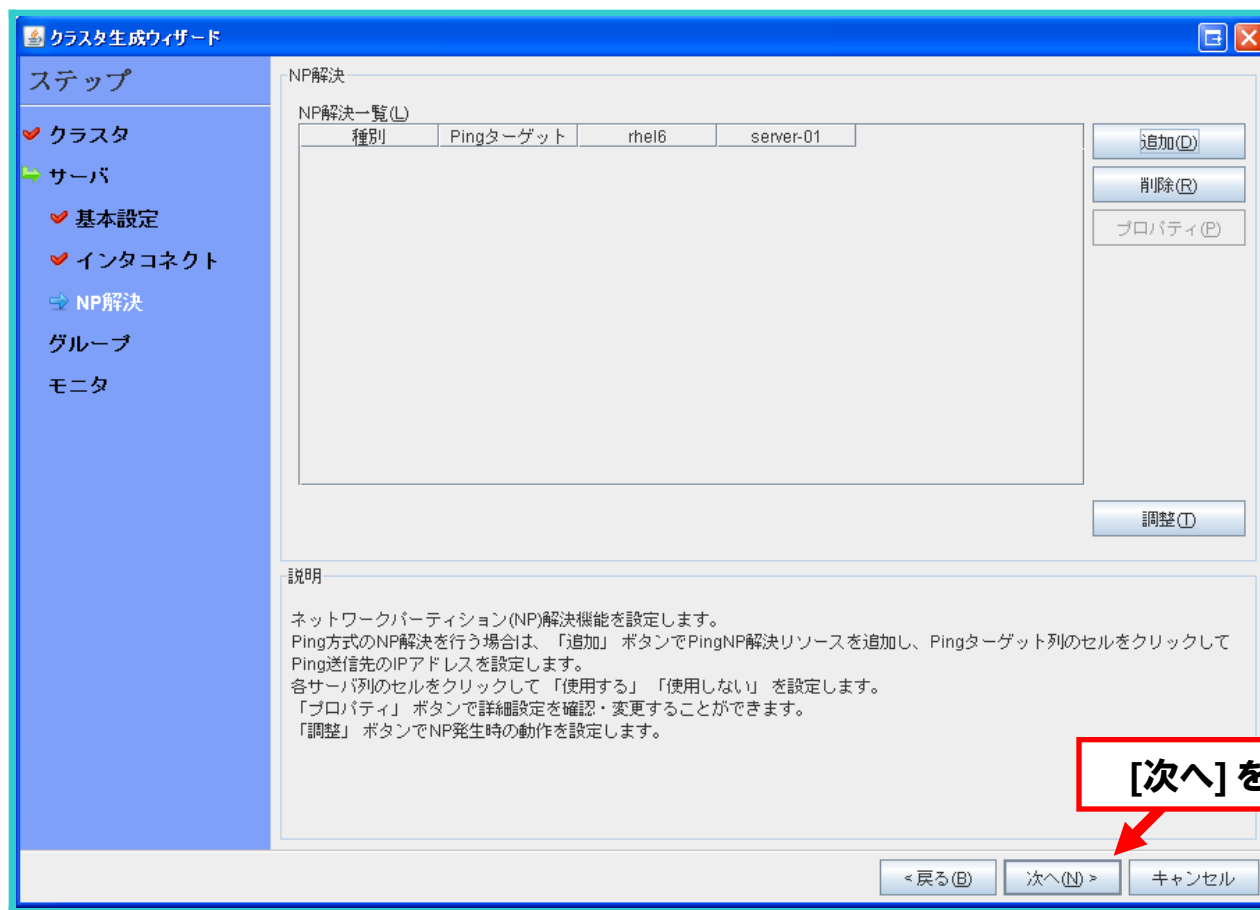
[OK] をクリックします

[次へ] をクリックします

2. 構築手順

vii. CLUSTERPRO 連携設定

6. [NP解決一覧] が表示されます。何も設定せずに、[次へ] をクリックします。



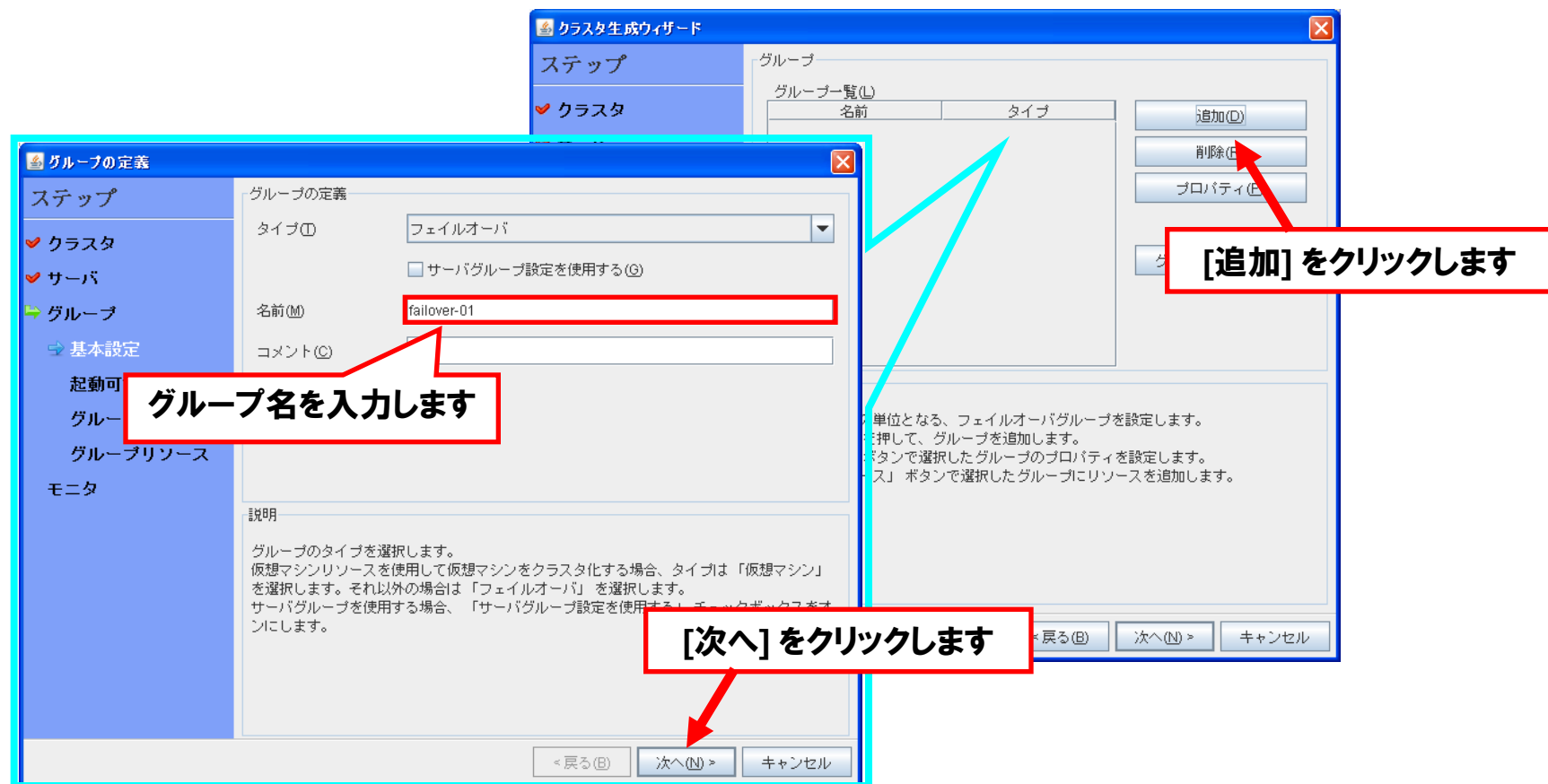
2. 構築手順

vii. CLUSTERPRO 連携設定

③ フェイルオーバーグループの追加

1. クラスタにフェイルオーバーグループを追加します。

障害発生時にフェイルオーバーを行う単位となる、フェイルオーバーグループの設定を行うために [追加] をクリックします。
[グループの定義] ダイアログボックスが開いた後、[名前] にグループ名(failover-01)を入力し、[次へ] をクリックします。



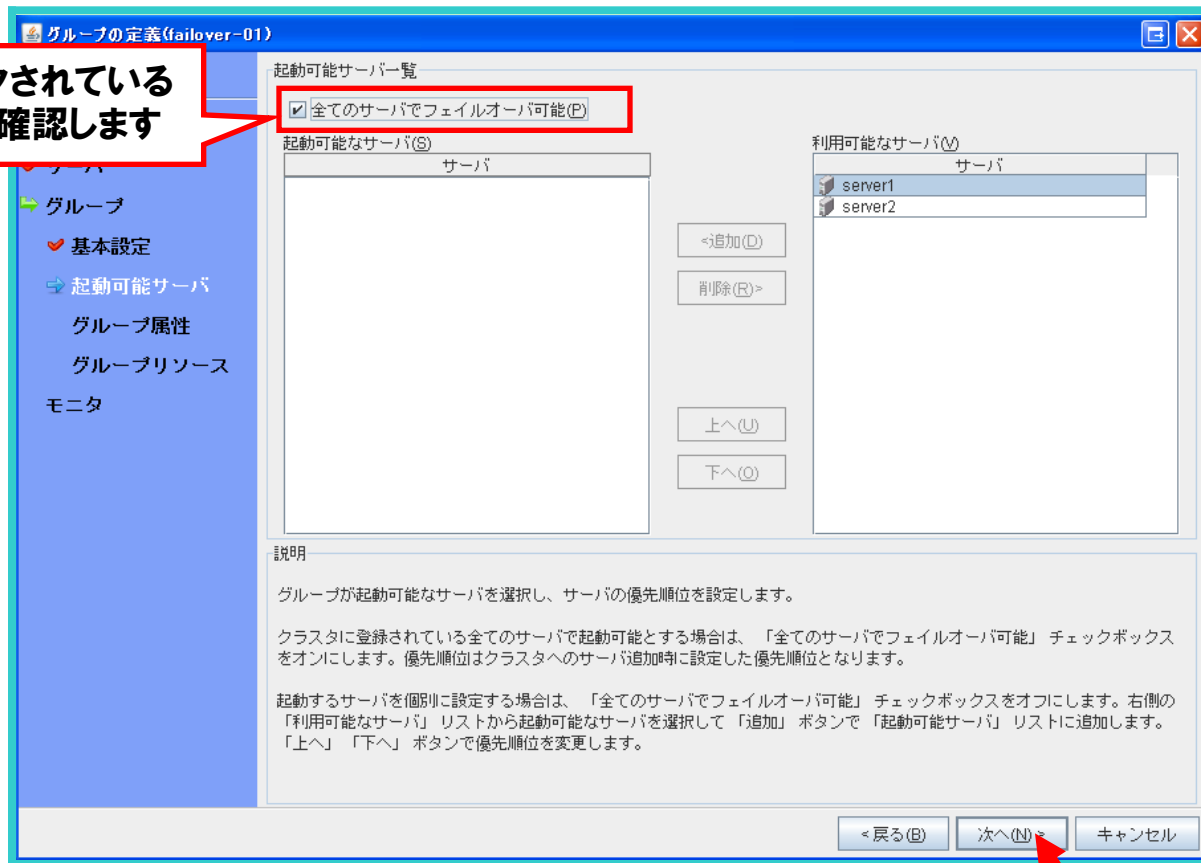
2. 構築手順

vii. CLUSTERPRO連携設定

2. 起動可能サーバの設定を行います。

[全てのサーバでフェイルオーバー可能]がチェックされていることを確認し、[次へ]をクリックします。

チェックされている
ことを確認します



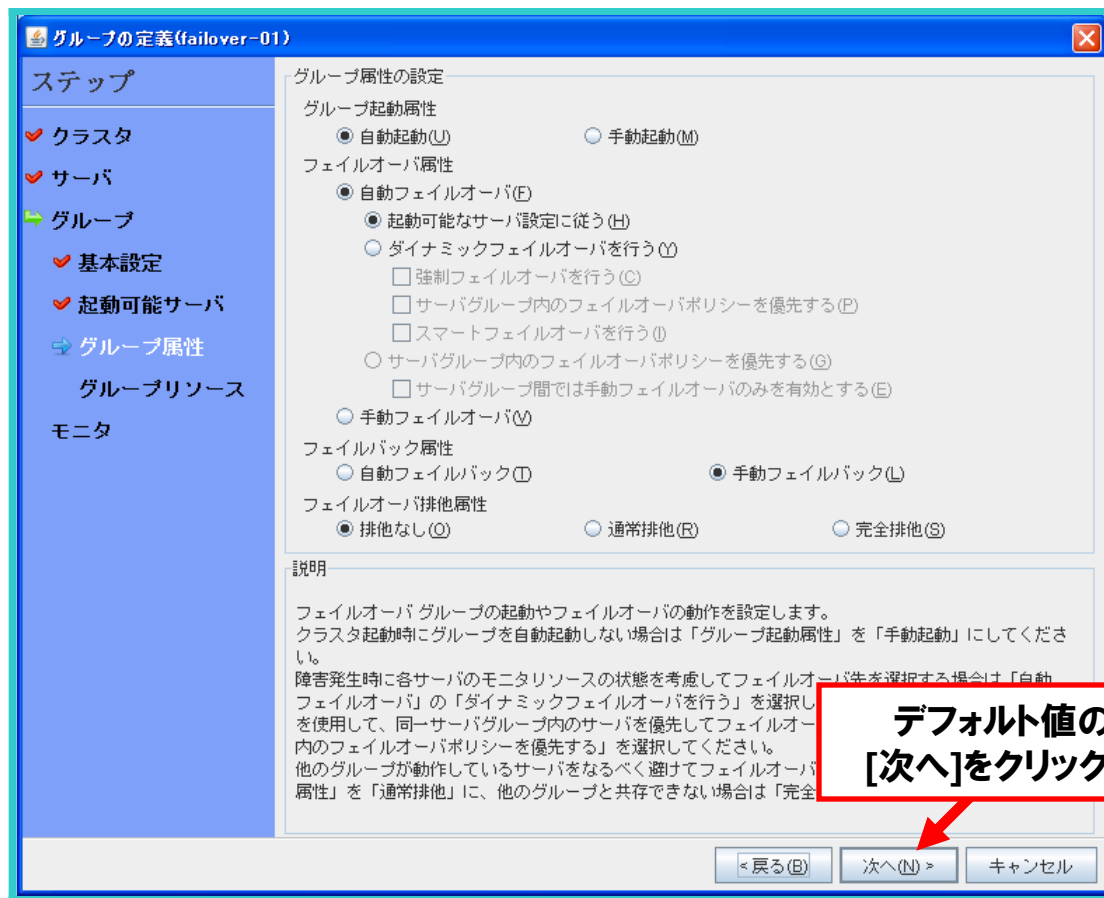
[次へ]をクリックします

2. 構築手順

vii. CLUSTERPRO連携設定

3. グループ属性の設定を行います。

[グループ属性の設定]が表示されます。[次へ]をクリックします。

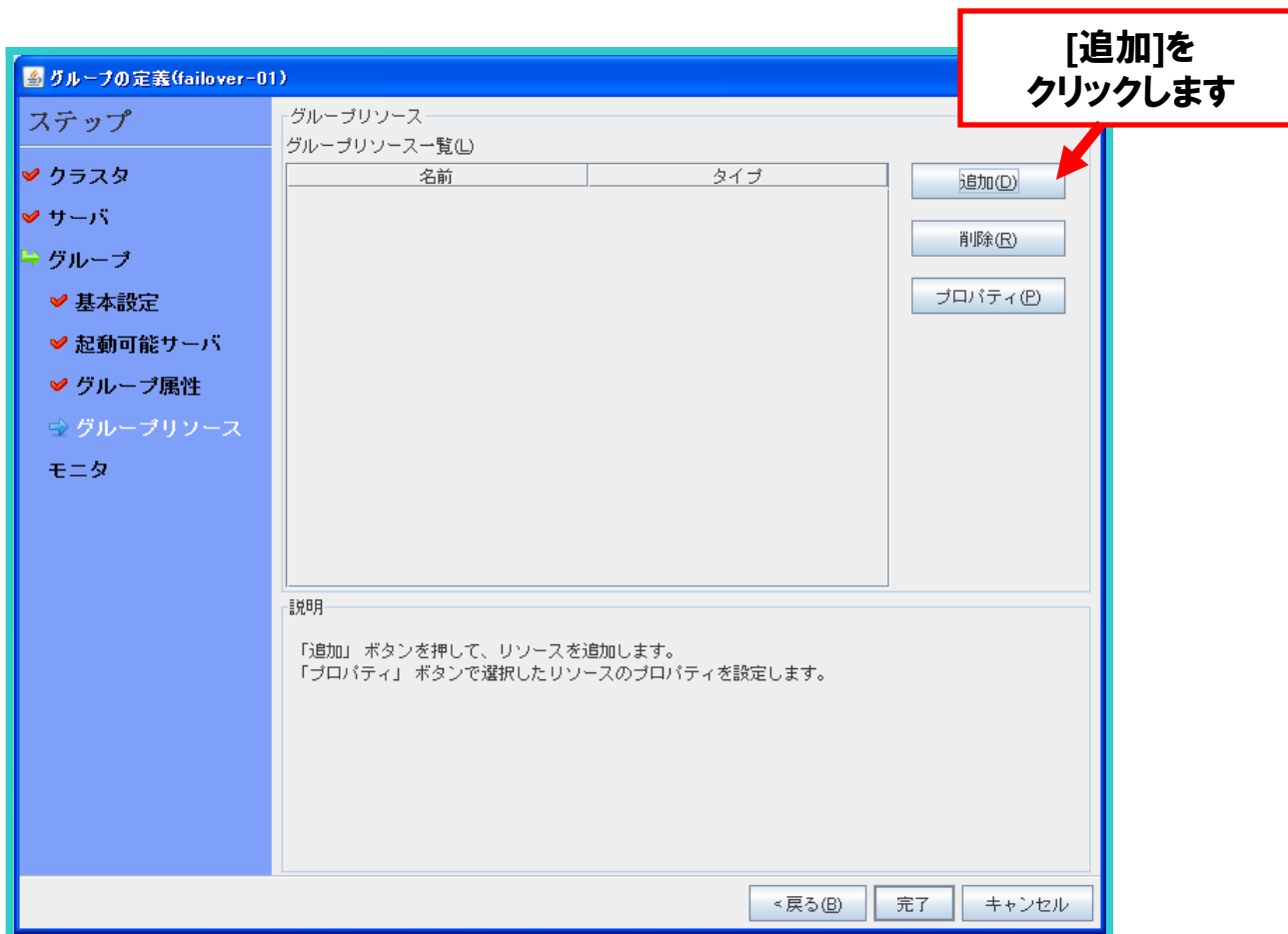


2. 構築手順

vii. CLUSTERPRO連携設定

4. グループリソースの設定を行います。

[グループリソース一覧]が表示されます。[追加]をクリックします。



2. 構築手順

vii. CLUSTERPRO連携設定

5. フェイルオーバーグループ用フローティングIPアドレスを設定します。

[グループのリソース定義]ダイアログボックスが開きます。[タイプ]でグループリソースのタイプ(floating ip resource)を選択し、[名前]にグループリソース名(fip1)を入力します。[次へ]をクリックします。

グループ (failover-01) のリソース定義

ステップ

- ✓ クラスタ
- ✓ サーバ
- ➡ グループ
 - ✓ 基本設定
 - ✓ 起動可能サーバ
 - ✓ グループ属性
 - ➡ グループリソース
 - ➡ 情報
 - 依存関係
 - 復旧動作
 - 詳細
- モニタ

グループリソースの定義

タイプ (T) floating ip resource

名前 (N) fip1

コメント (C)

ライセンス

グループリソース名を入力します

[floating ip resource]を選択します

説明

グループリソースの種類を選択して名前を入力してください。

< 戻る (B) 次へ (N) > キャンセル

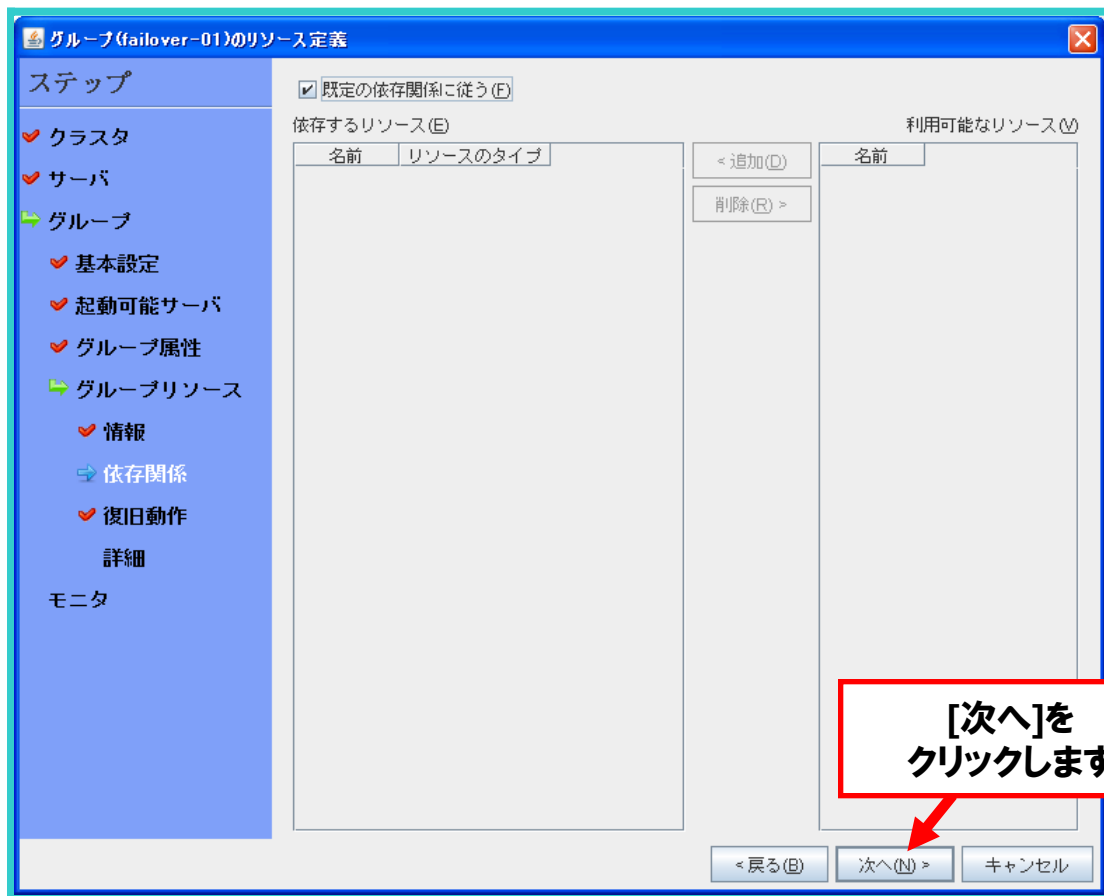
[次へ]をクリックします

2. 構築手順

vii. CLUSTERPRO連携設定

6. 依存関係の設定を行います。

依存関係設定のページが表示されます。何も指定せず[次へ]をクリックします。



2. 構築手順

vii. CLUSTERPRO連携設定

7. 障害発生時の最終動作の設定を行います。

[活性異常検出時の復旧動作]、[非活性異常検出時の復旧動作]が表示されます。[次へ]をクリックします。

グループ (failover-01) のリソース定義

ステップ

- ✓ クラスタ
- ✓ サーバ
- ➡ グループ
 - ✓ 基本設定
 - ✓ 起動可能サーバ
 - ✓ グループ属性
 - ➡ グループリソース
 - ✓ 情報
 - ✓ 依存関係
 - ➡ 復旧動作
 - 詳細
 - モニタ

活性異常検出時の復旧動作

活性リトライしきい値 (R) 5 回

フェイルオーバーしきい値 (I) 1 回

最終動作 (F) 何もしない (次のリソースを活性しない)

☐ 最終動作前にスクリプトを実行する (X) 設定 (S)

非活性異常検出時の復旧動作

非活性リトライしきい値 (E) 0 回

最終動作 (I) クラスタサービス停止と OS シャットダウン

☐ 最終動作前にスクリプトを実行する (C) 設定 (C)

< 戻る (B) 次へ (N) > キャンセル

デフォルト値のまま
[次へ]をクリックします

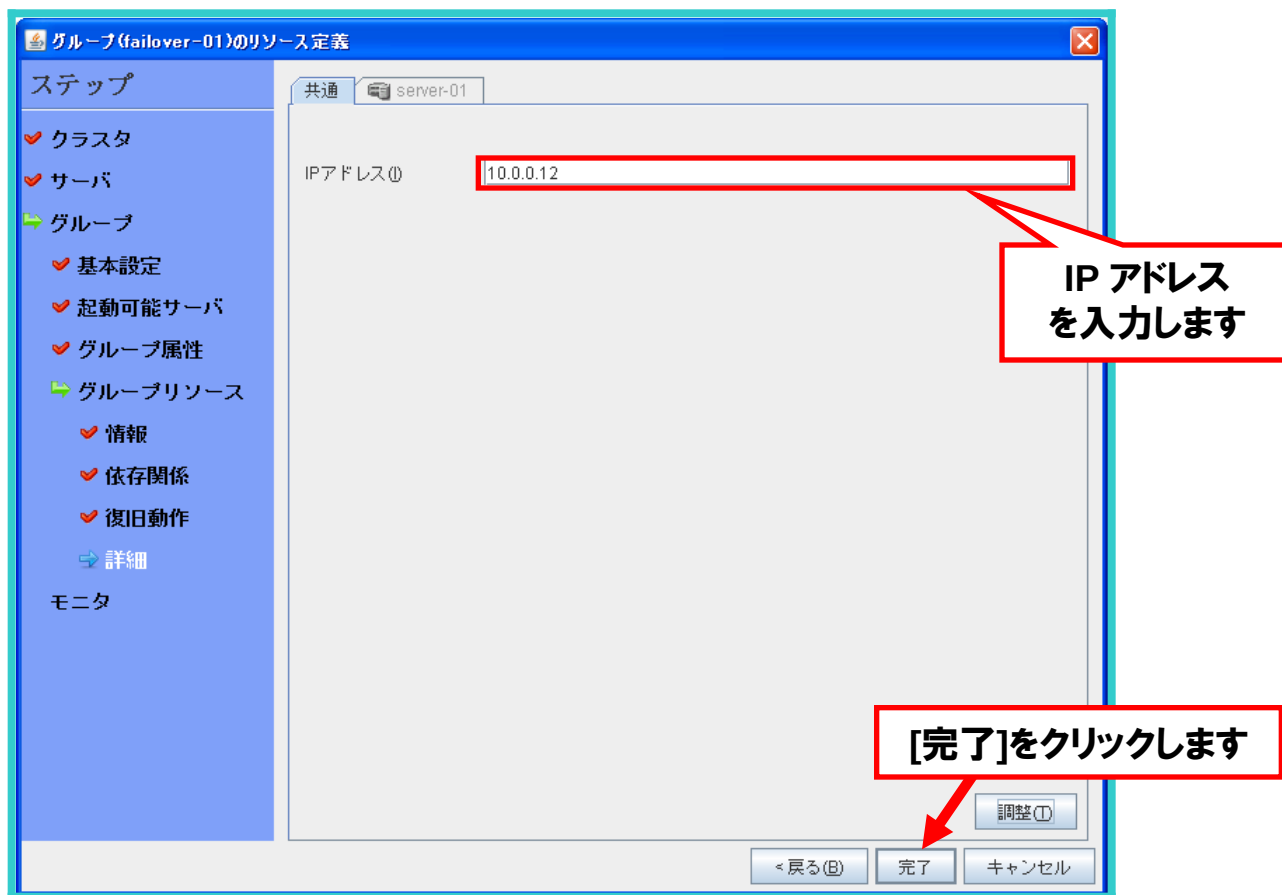
2. 構築手順

vii. CLUSTERPRO連携設定

8. フェイルオーバーグループ用フローティングIPアドレスを入力します。

[IP アドレス]にIPアドレス(10.0.0.12)を入力し[完了]をクリックします。

フローティングIPは、WebManagerへアクセスする際に管理クライアントが使用するIPアドレスです。
これにより、管理用クライアントからは、常にアクティブなサーバにアクセスすることができます。

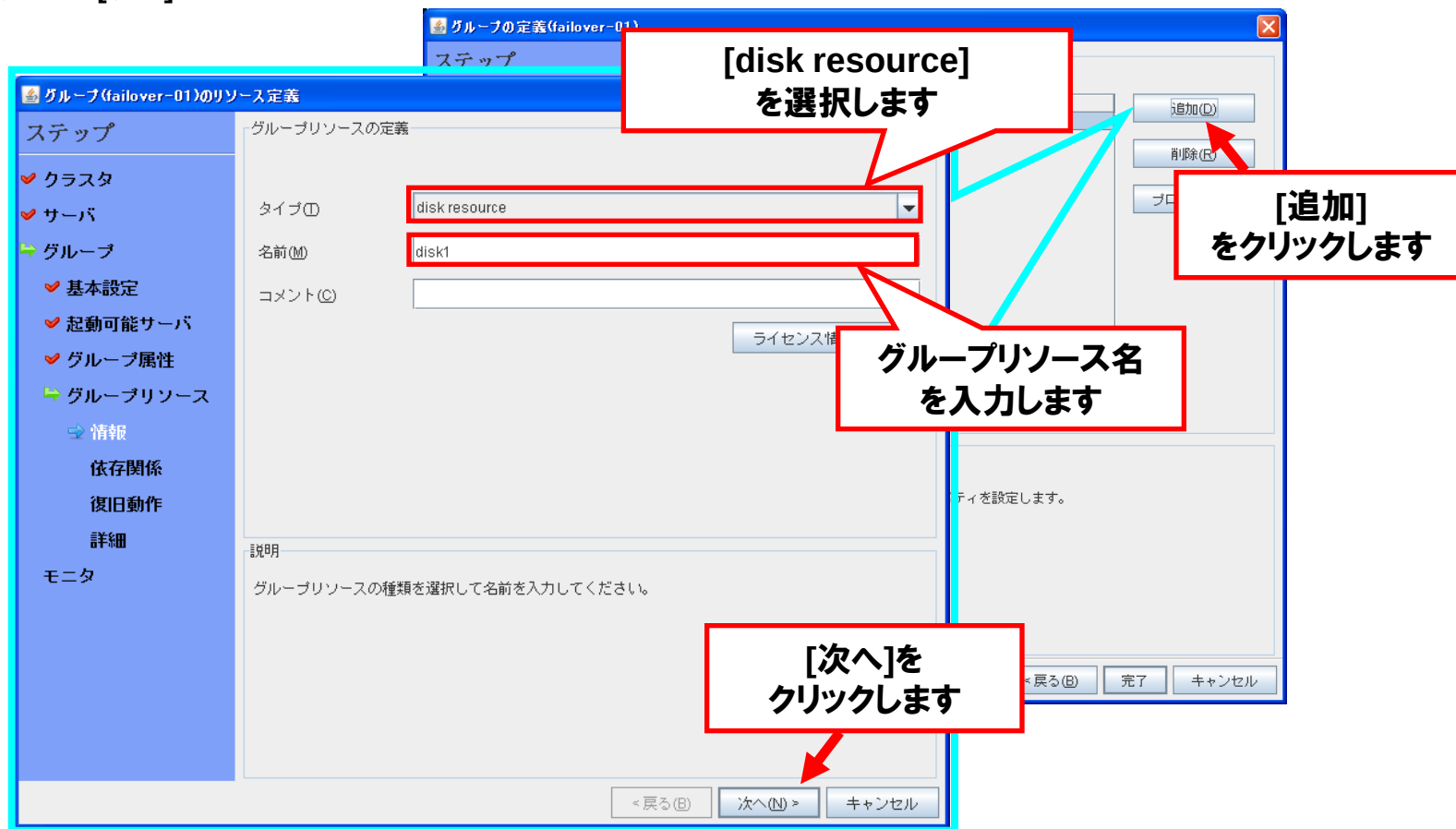


2. 構築手順

vii. CLUSTERPRO連携設定

9. 共有ディスクをグループリソースとして追加します。

[グループリソースの定義一覧]で[追加]をクリックすると[グループのリソース定義]ダイアログボックスが開きます。
[タイプ]でグループリソースのタイプ(disk resource)を選択し、[名前]にグループリソース名(disk1)を入力して、[次へ]をクリックします。

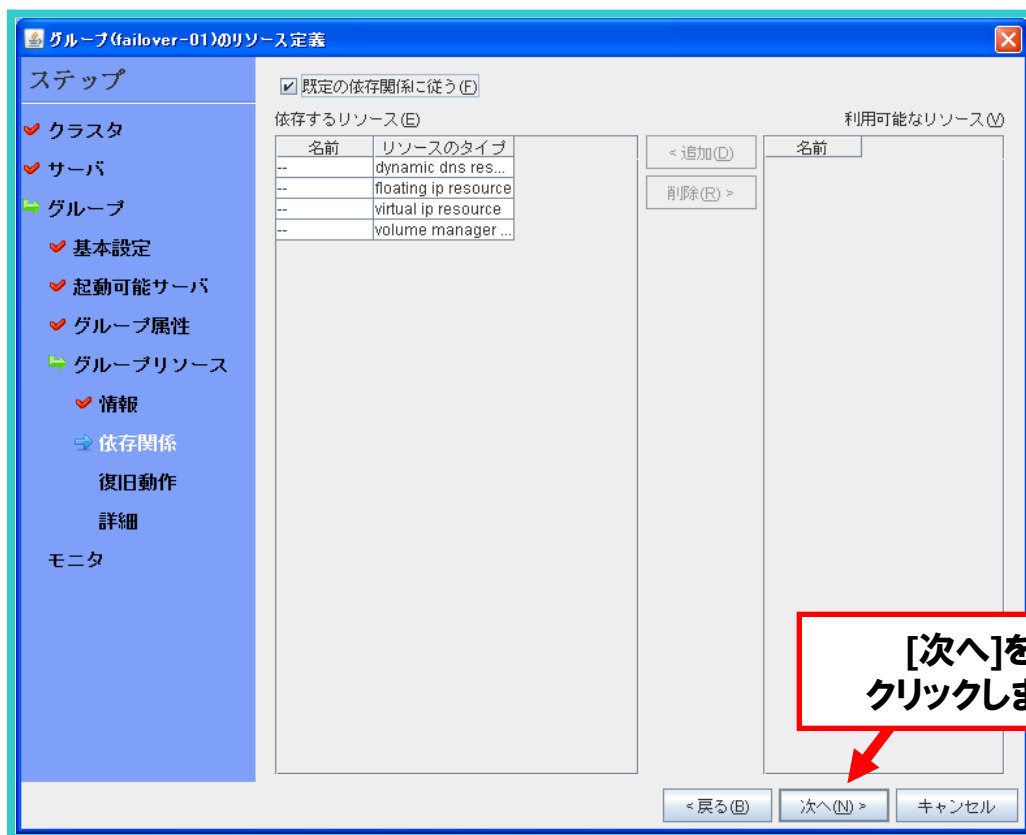


2. 構築手順

vii. CLUSTERPRO連携設定

10. 依存関係の設定を行います。

依存関係設定のページが表示されます。何も指定せず[次へ]をクリックします。



2. 構築手順

vii. CLUSTERPRO連携設定

11. 障害発生時の最終動作を設定します。

[活性異常検出時の復旧動作]、[非活性異常検出時の復旧動作]が表示されます。[次へ]をクリックします。

2. 構築手順

vii. CLUSTERPRO連携設定

12. 共有ディスクの情報を入力します。

デバイス名(/dev/dda2)、マウントポイント(/mnt/dda)をそれぞれのボックスに入力し、[ファイルシステム]でファイルシステム(ext3)、[ディスクのタイプ]でディスクのタイプ(disk)を選択します。[完了]をクリックします。

※マウントポイントには存在するディレクトリを指定してください。

The screenshot shows the 'グループ(failover-01)のリソース定義' (Group Resource Definition) window. The left sidebar lists steps: ステップ, クラスタ, サーバ, グループ, 基本設定, 起動可能サーバ, グループ属性, グループリソース, 情報, 依存関係, 復旧動作, 詳細, モニタ. The main area is titled '共通 server-01' and contains the following fields:

- ディスクのタイプ(D): disk (selected)
- ファイルシステム(F): ext3 (selected)
- デバイス名(E): /dev/dda2
- RAWデバイス名(R):
- マウントポイント(M): /mnt/dda

Red callout boxes with arrows point to the following elements:

- [disk]を選択します (points to the Disk Type dropdown)
- [ext3]を選択します (points to the File System dropdown)
- デバイス名を入力します (points to the Device Name text box)
- マウントポイントを入力します (points to the Mount Point text box)
- [完了]をクリックします (points to the '完了' button at the bottom right)

At the bottom of the window are buttons: < 戻る(B), 完了, キャンセル, and a small 調整(A) button.

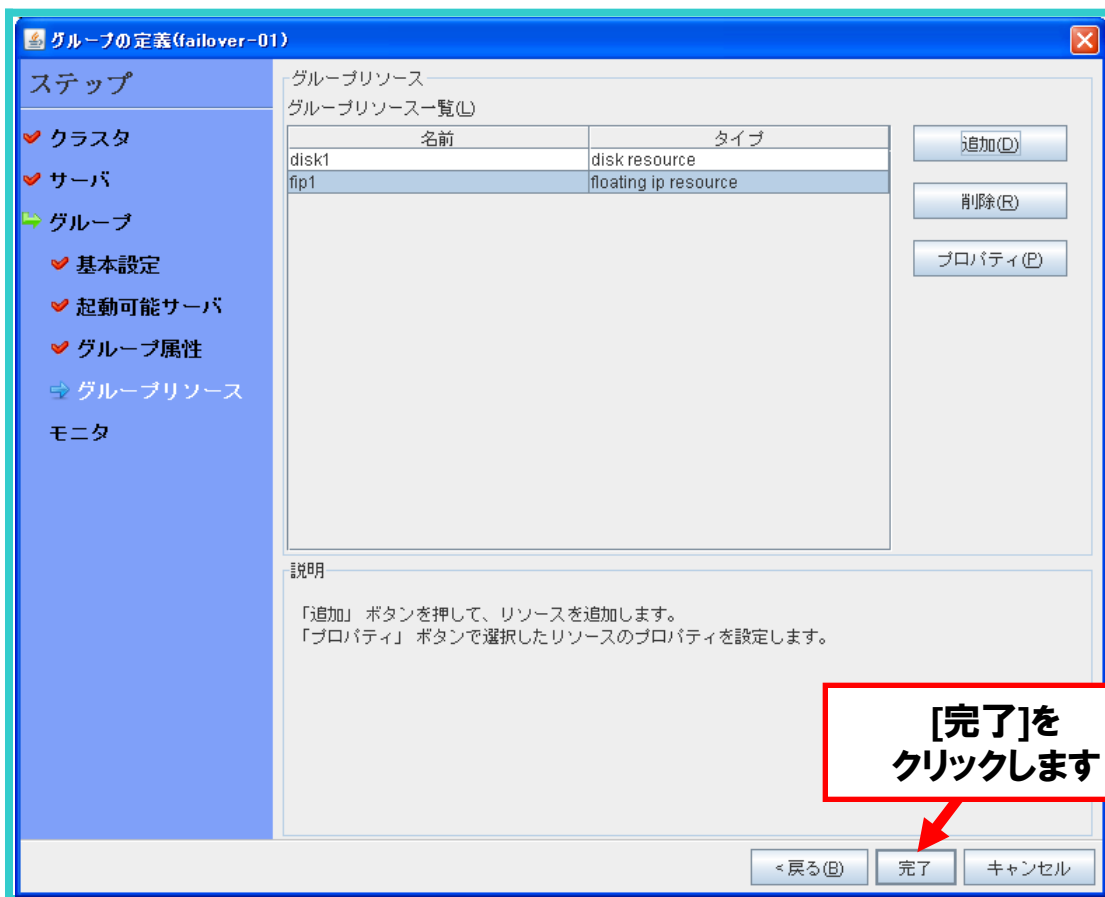
2. 構築手順

vii. CLUSTERPRO連携設定

13. グループリソース一覧が表示されます。

登録情報が正しく表示されていることを確認し、[完了]をクリックします。

以上でフェイルオーバーグループの追加は終了です。



2. 構築手順

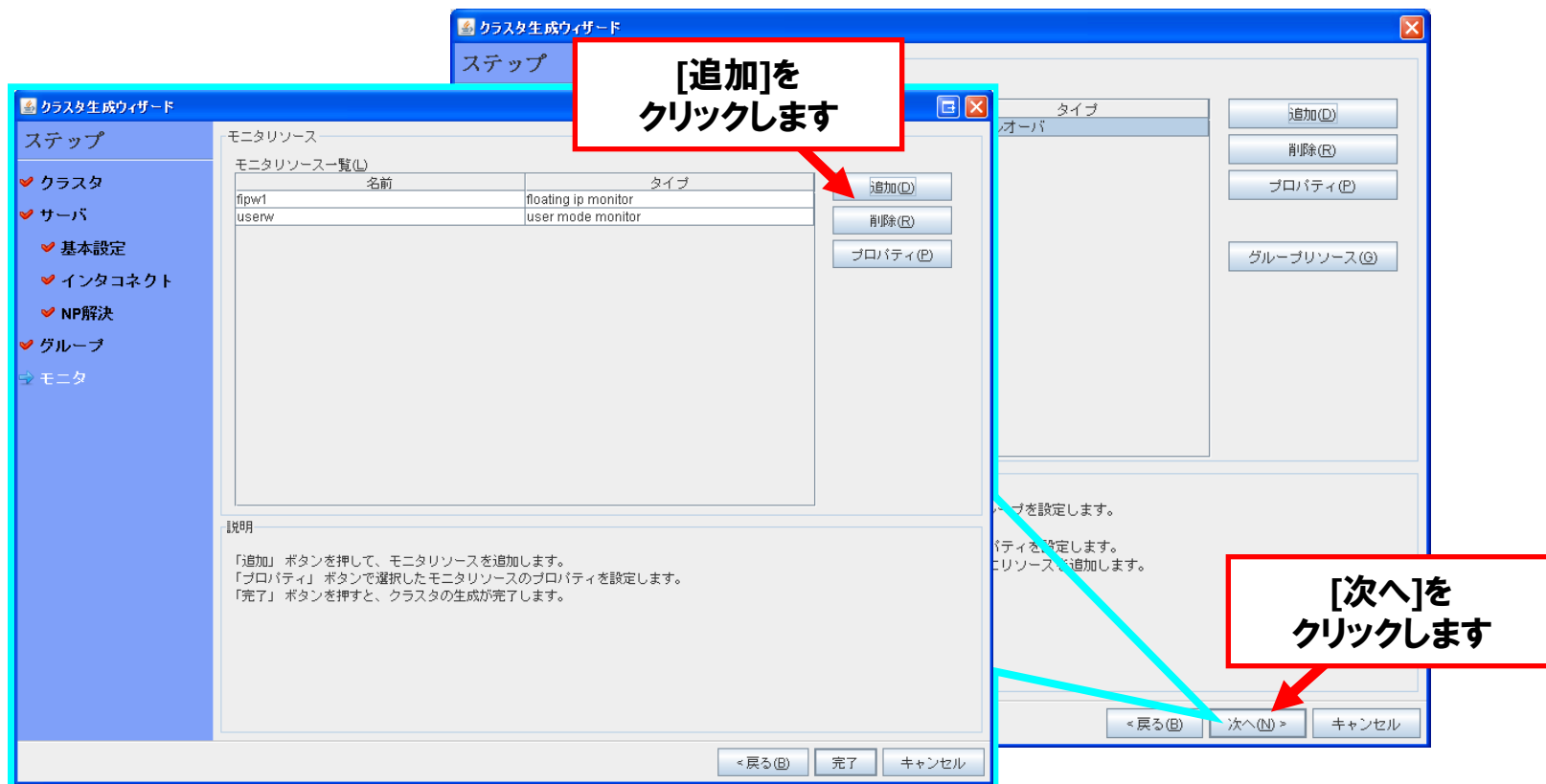
vii. CLUSTERPRO連携設定

⑤ モニタリソースの追加

1. モニタリソースの追加を行います。

[グループの定義一覧]で、[次へ]をクリックします。

[モニタリソースの定義一覧]が表示されます。[追加]をクリックします。



2. 構築手順

vii. CLUSTERPRO連携設定

2. モニタリソースのタイプと名前を設定します。

[モニタリソースの定義]ダイアログボックスが開きます。

[タイプ]でグループリソースのタイプ(message receive monitor)を選択し、

[名前]にグループリソース名(mrw1)を入力して、[次へ]をクリックします。

The screenshot shows the 'Monitor Resource Definition' dialog box. The left sidebar has a tree view with 'ステップ' (Steps) expanded, showing 'クラスター' (Cluster), 'サーバ' (Server), 'グループ' (Group), and 'モニタ' (Monitor). Under 'モニタ', '情報' (Information) is selected, and '監視(共通)' (Common Monitoring) is highlighted. The main area is titled 'モニタリソース定義' (Monitor Resource Definition) and contains three fields: 'タイプ(T)' (Type) with a dropdown menu showing 'message receive monitor', '名前(N)' (Name) with the text 'mrw1', and 'コメント(C)' (Comment). Below these fields is a '説明' (Description) section with the text 'モニタリソースの種類を選択して名前を入力してください。' (Please select the type of monitor resource and enter the name). At the bottom are three buttons: '< 戻る(B)' (Back), '次へ(N) >' (Next), and 'キャンセル' (Cancel). Three red callout boxes with arrows point to the 'message receive monitor' dropdown, the 'mrw1' text field, and the '次へ(N) >' button.

モニタリソースの定義

ステップ

- クラスター
- サーバ
- グループ
- モニタ
 - 情報
 - 監視(共通)
 - 監視(固有)
 - 回復動作

モニタリソース定義

タイプ(T) [message receive monitor]

名前(N) [mrw1]

コメント(C)

説明

モニタリソースの種類を選択して名前を入力してください。

[message receive monitor]を選択します

グループリソース名を入力します

[次へ]をクリックします

< 戻る(B) 次へ(N) > キャンセル

2. 構築手順

vii. CLUSTERPRO連携設定

3. [監視リソース設定画面]が表示されます。[次へ]をクリックします。

モニタリソースの定義

ステップ

- ✓ クラスタ
- ✓ サーバ
- ✓ グループ
- ➡ モニタ
 - ✓ 情報
 - ➡ 監視(共通)
 - 監視(固有)
 - 回復動作

インターバル① 10 秒

タイムアウト① 30 秒

リトライ回数(R) 0 回

監視開始待ち時間(S) 0 秒

監視タイミグ

☒ 常時(L)

☐ 活性時(C)

対象リソース 参照(W)

nice値(E) 19

監視を行うサーバを選択する

サーバ(W)

[次へ]をクリックします

< 戻る(B) 次へ(N) > キャンセル

2. 構築手順

vii. CLUSTERPRO連携設定

4. モニタリソースの監視タイプを設定します。

[カテゴリ]でHA/SSを選択し、[キーワード]に監視対象(PSEUDO_VG001_status)を入力して[次へ]をクリックします。

モニタリソースの定義

ステップ

- ✓ クラスタ
- ✓ サーバ
- ✓ グループ
- ➡ モニタ
 - ✓ 情報
 - ✓ 監視(共通)
 - ➡ 監視(固有)
 - 回復動作

共通 server-01

カテゴリ(C) HA/SS

キーワード(K) PSEUDO_VG001_status

[HA/SS]を選択します

監視対象名を入力します

[次へ]をクリックします

<戻る(B) 次へ(N)> キャンセル

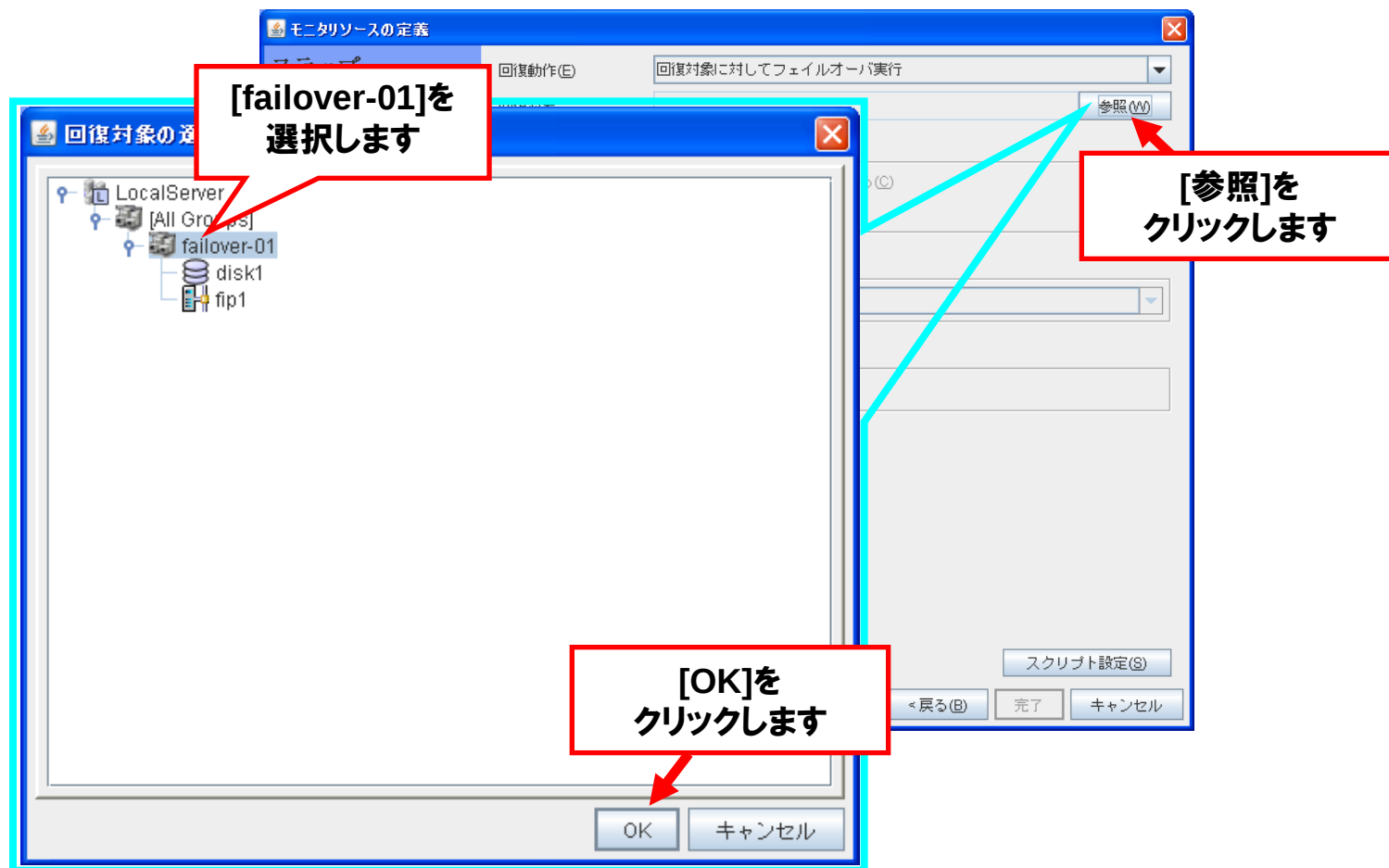
2. 構築手順

vii. CLUSTERPRO連携設定

5. 回復対象の選択を行います。

[参照]をクリックして[回復対象の選択]画面を表示します。

[failover-01]を選択して[OK]をクリックします。



2. 構築手順

vii. CLUSTERPRO連携設定

6. フェイルオーバーが発生した場合の動作である最終動作の選択を行います。

[回復動作]で最終動作を実行を選択し、

[最終動作]でクラスタサービス停止とOSシャットダウンを選択して[完了]をクリックします。

モニタリソースの定義

ステップ

- ✓ クラスタ
- ✓ サーバ
- ✓ グループ
- ➡ モニタ
- ✓ 情報
- ✓ 監視(共通)
- ✓ 監視(固有)
- ➡ 回復動作

回復動作(E) 最終動作を実行

回復対象 failover-01

☐ フェイルオーバー実行前にマイグレーションを実行する(C)

☐ サーバグループ外にフェイルオーバーする(S)

最終動作(E) クラスタサービス停止とOSシャットダウン

☒ 回復動作前にスクリプトを実行する(S)

スクリプト設定(S)

<戻る(B) 完了 キャンセル

[最終動作を実行]を選択します

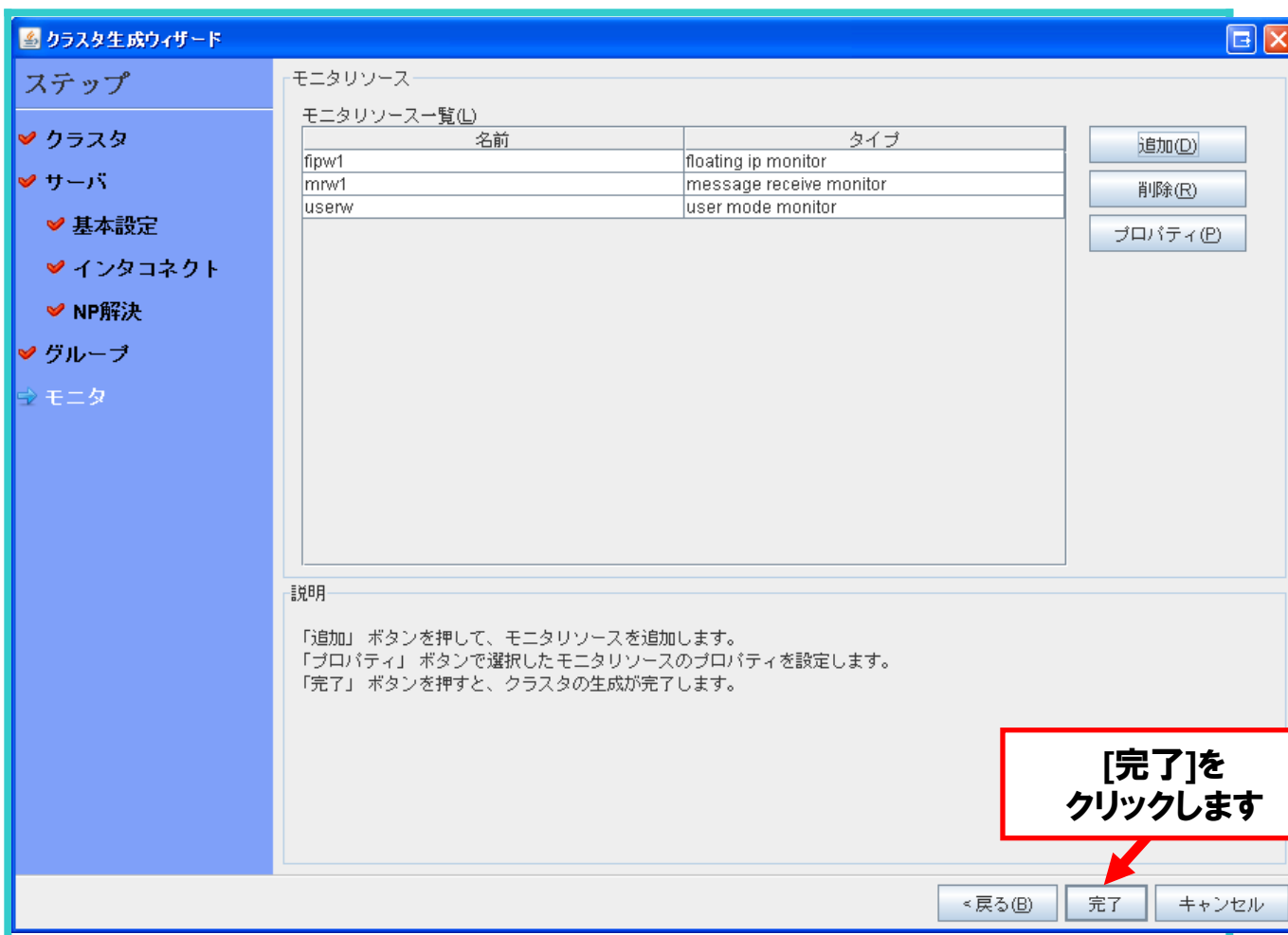
[クラスタサービス停止とOSシャットダウン]を選択します

[完了]をクリックします

2. 構築手順

vii. CLUSTERPRO連携設定

7. モニタリソースの定義一覧が表示されるので[完了]をクリックします。

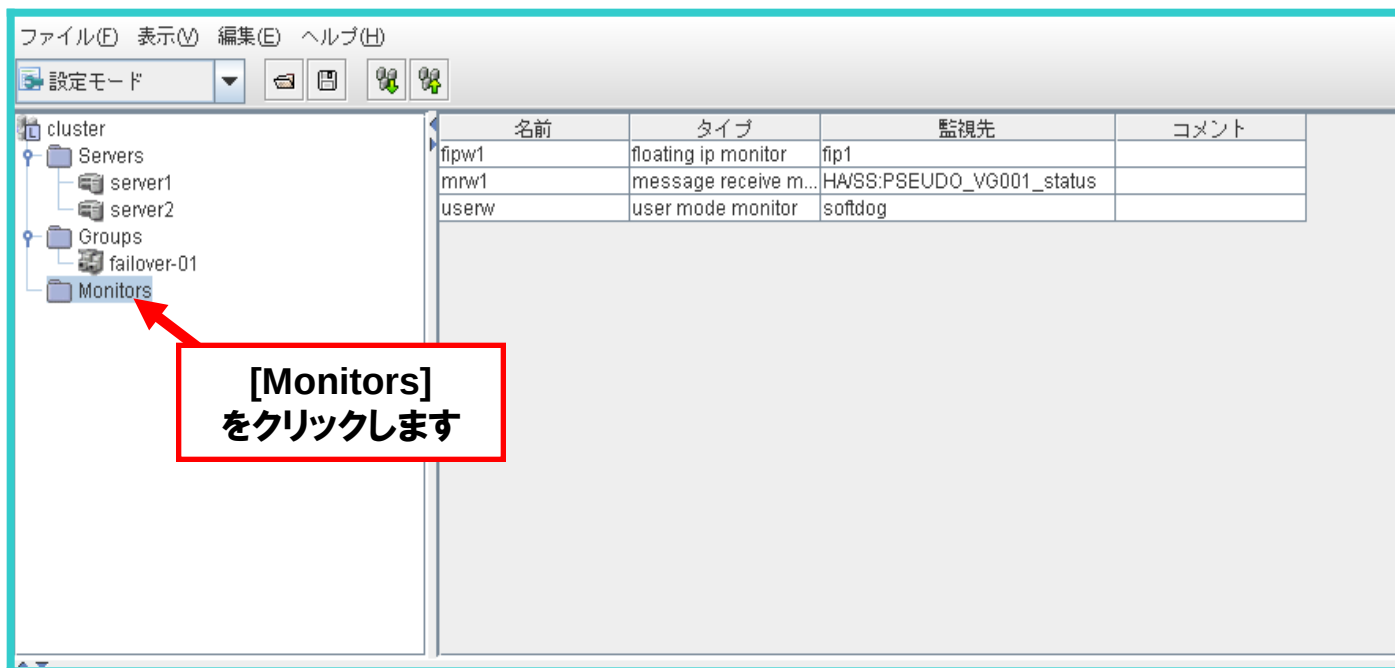


2. 構築手順

vii. CLUSTERPRO連携設定

8. 以上でモニタリソースの追加は終了です。

[Monitors] をクリックし、定義したモニタリソースが表示されることを確認してください。



2. 構築手順

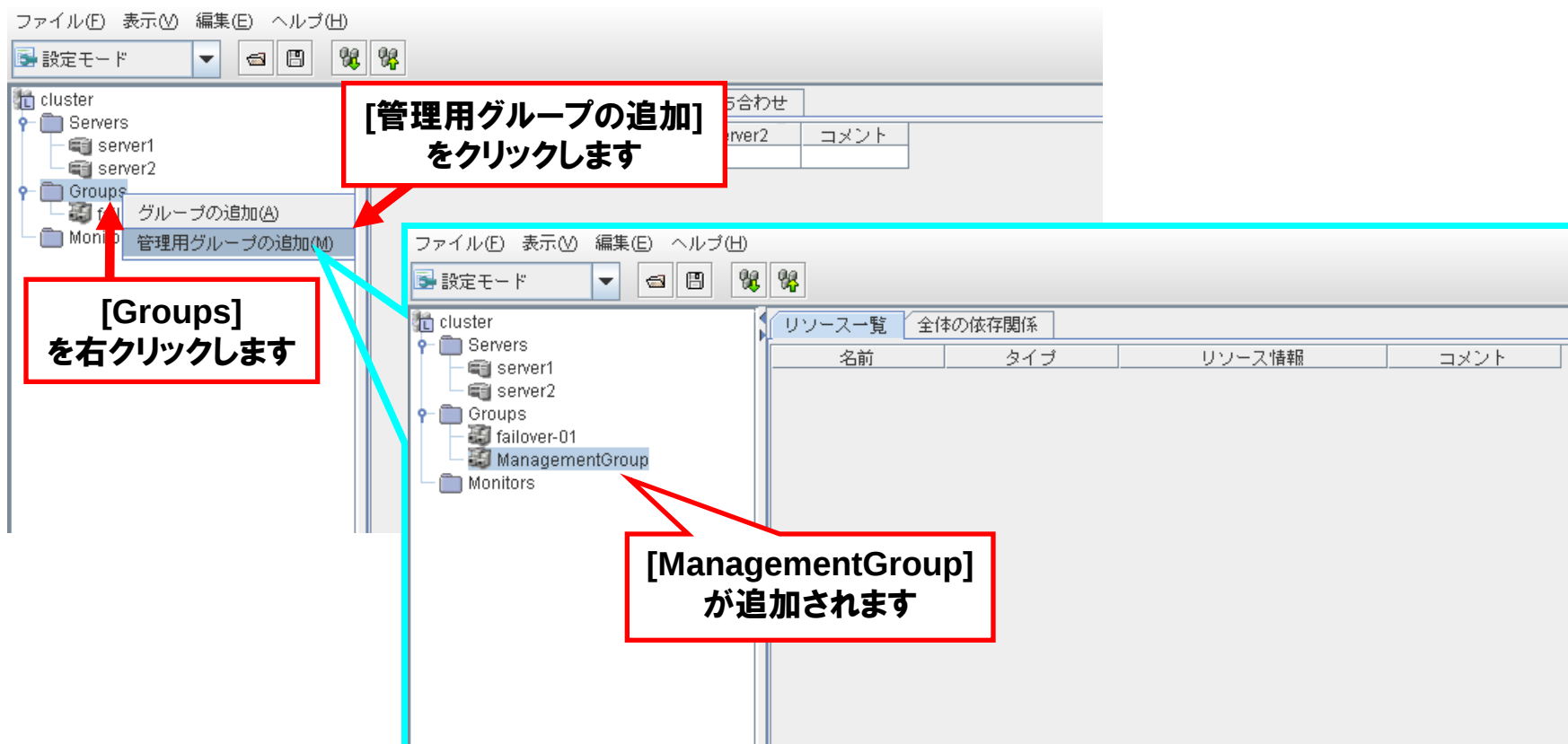
vii. CLUSTERPRO連携設定

⑥ ManagementGroup の追加

1. ManagementGroupの追加を行います。

[Groups]を右クリックし、[管理用グループの追加]をクリックしてください。

[Groups]配下に、[ManagementGroup]が追加され表示されます。

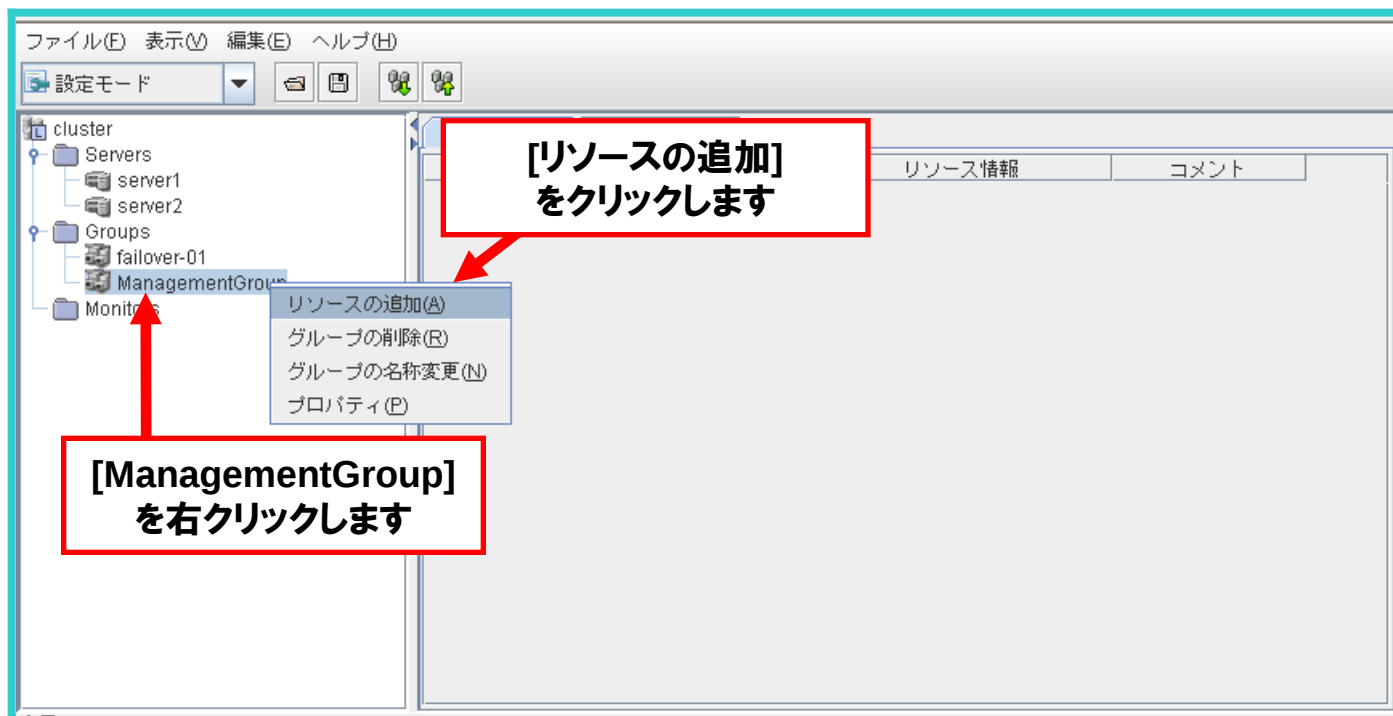


2. 構築手順

vii. CLUSTERPRO連携設定

2. ManagementGroupの追加を行います。

[ManagementGroup]を 右クリックし、[リソースの追加]をクリックしてください。



2. 構築手順

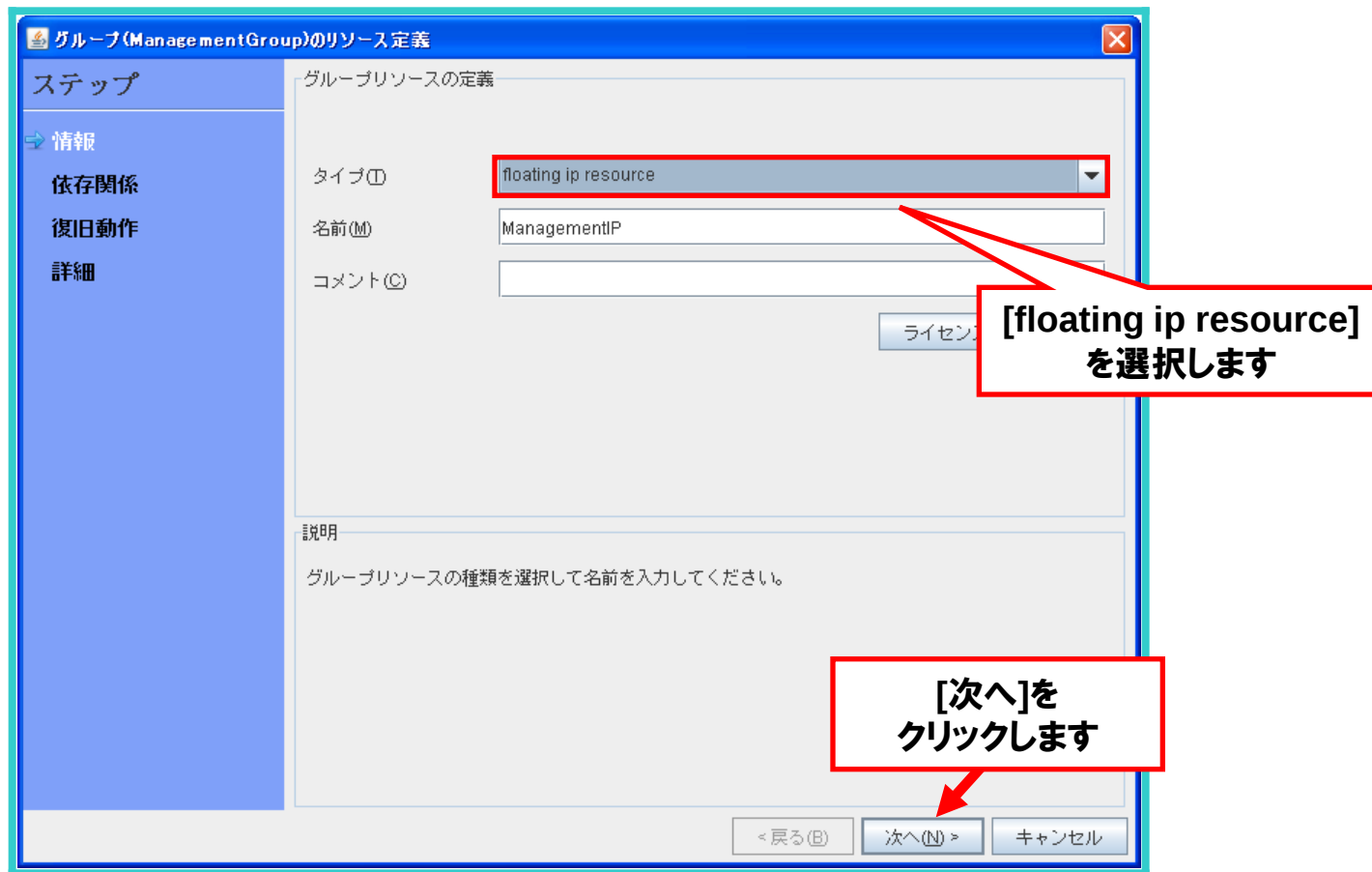
vii. CLUSTERPRO連携設定

3. ManagementGroup用フローティングIPアドレスの設定を行います。

[グループのリソース定義]ダイアログボックスが開きます。

[タイプ]でグループリソースのタイプ(floating ip resource)を選択します。

[名前]にはデフォルトの名前(ManagementIP)が入力されています。[次へ]をクリックします。

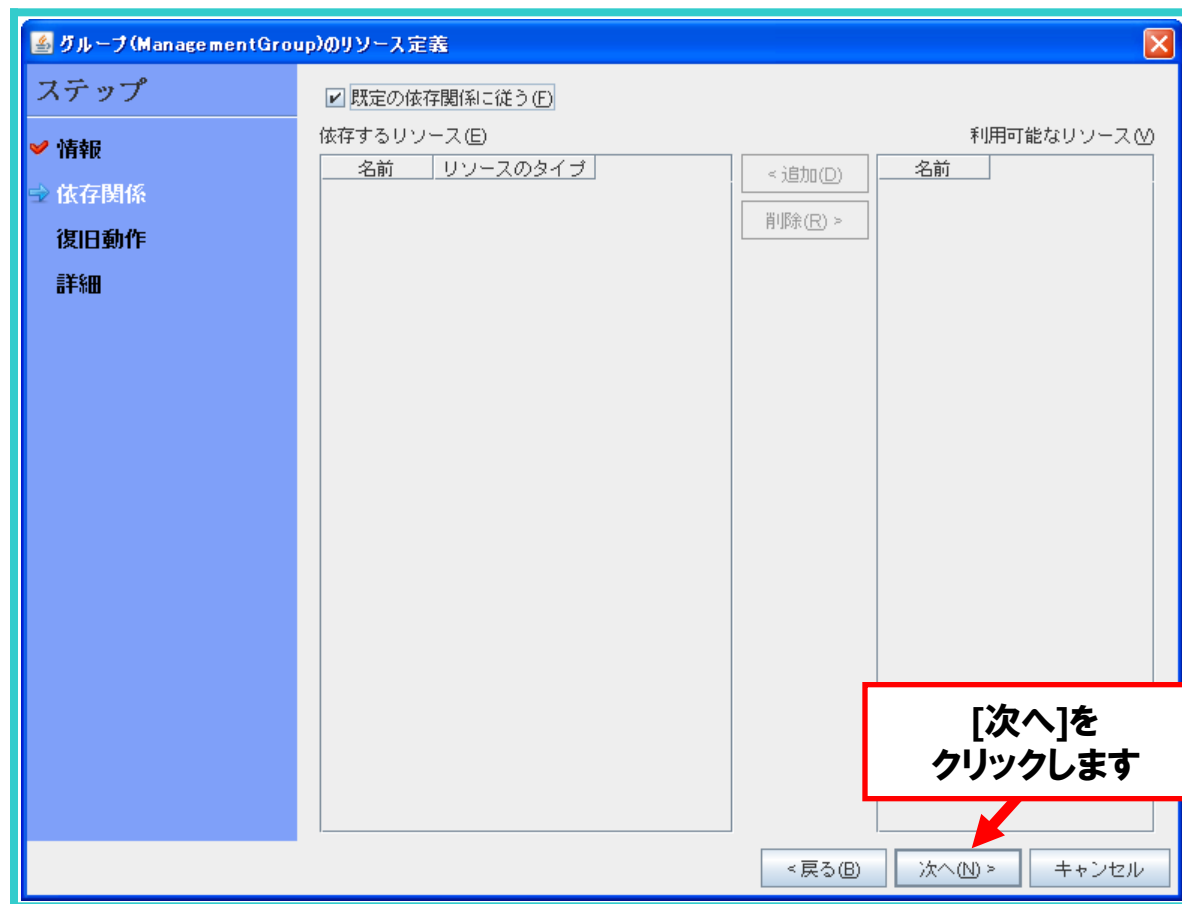


2. 構築手順

vii. CLUSTERPRO連携設定

4. 依存関係の設定を行います。

依存関係設定のページが表示されます。何も指定せず[次へ]をクリックします。



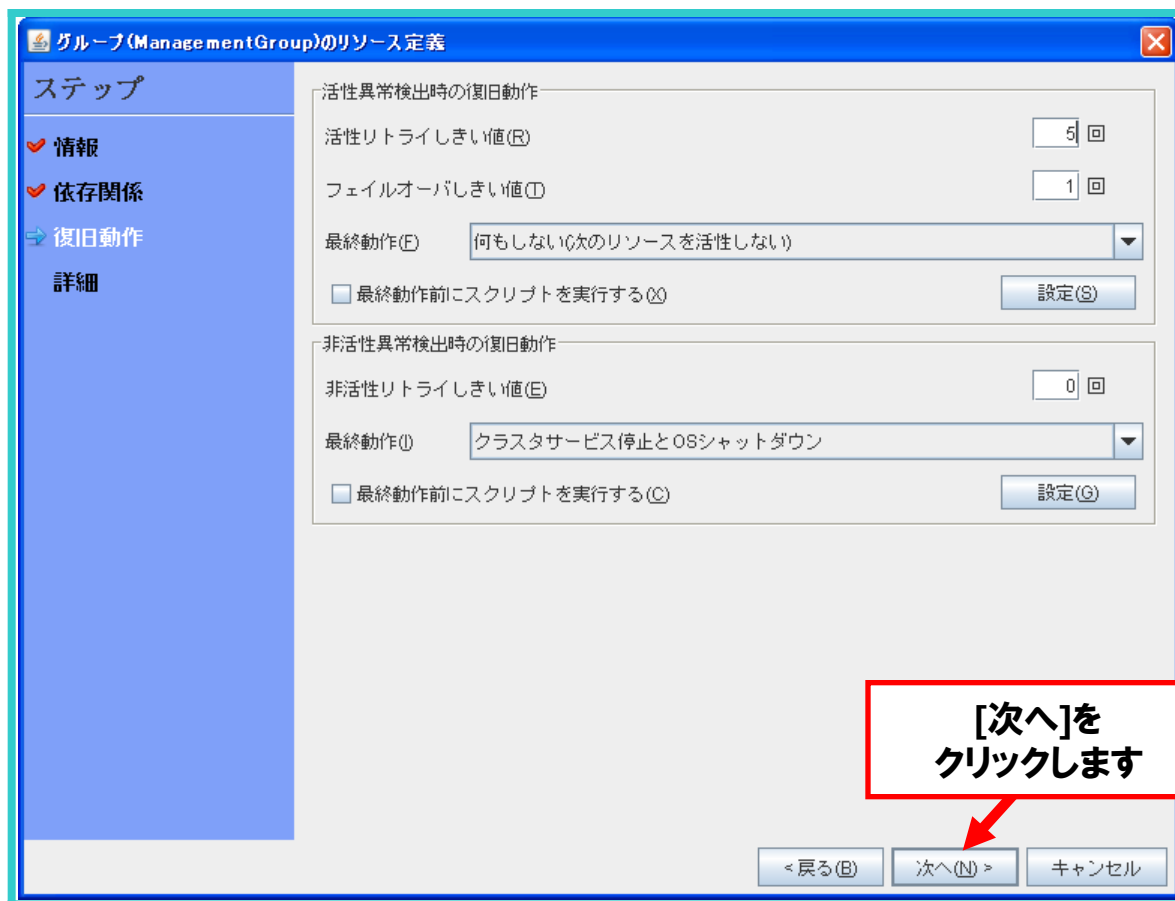
2. 構築手順

vii. CLUSTERPRO連携設定

5. 障害発生時の最終動作を設定します。

[活性異常検出時の復旧動作]、[非活性異常時の復旧動作]が表示されます。

[次へ]をクリックします。



グループ (ManagementGroup) のリソース定義

ステップ

- 情報
- 依存関係
- 復旧動作
- 詳細

活性異常検出時の復旧動作

活性リトライしきい値 (R) 5 回

フェイルオーバーしきい値 (I) 1 回

最終動作 (E) 何もしない (次のリソースを活性しない)

☐ 最終動作前にスクリプトを実行する (S) 設定 (S)

非活性異常検出時の復旧動作

非活性リトライしきい値 (R) 0 回

最終動作 (I) クラスタサービス停止とOSシャットダウン

☐ 最終動作前にスクリプトを実行する (C) 設定 (S)

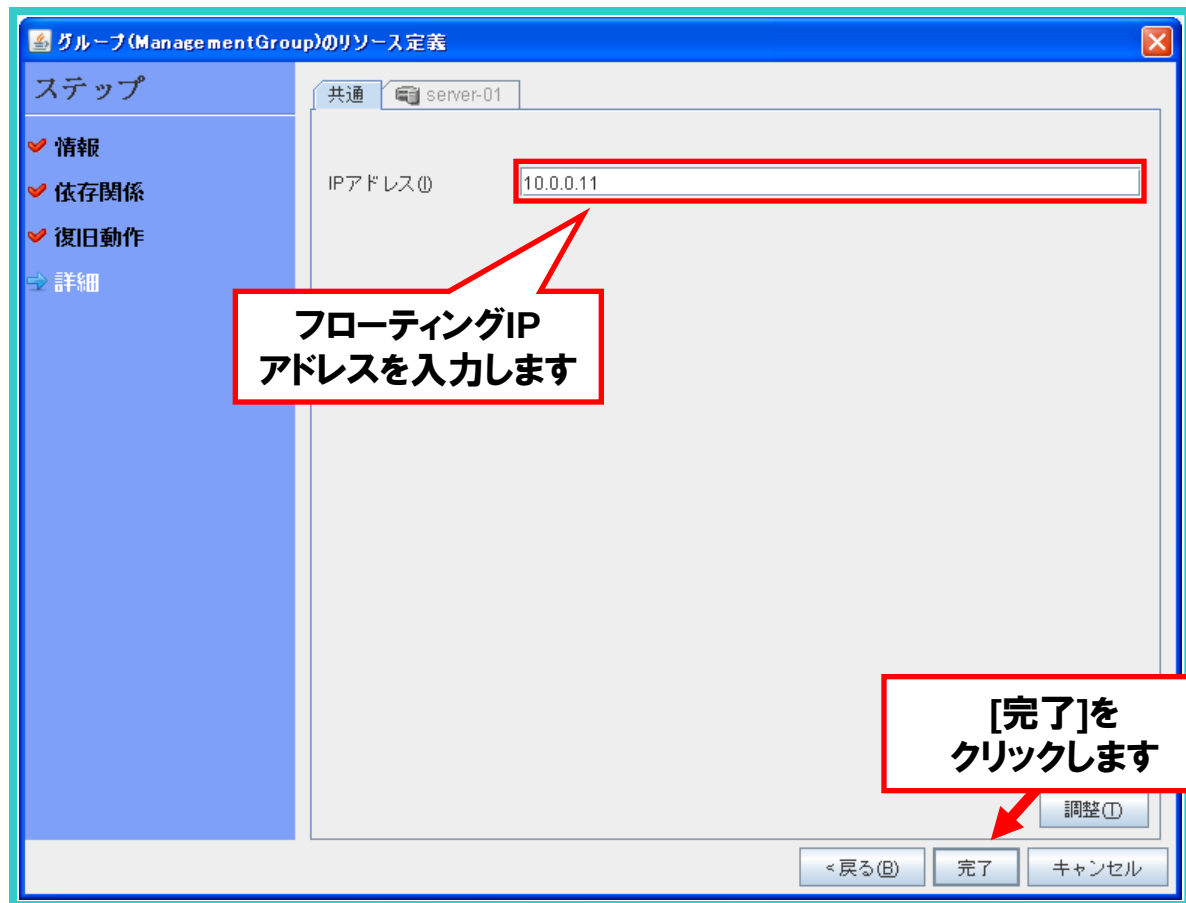
[次へ]をクリックします

< 戻る (B) 次へ (N) > キャンセル

2. 構築手順

vii. CLUSTERPRO連携設定

6. ManagementGroup用フローティングIPアドレスの入力を行います。
[IPアドレス]にフローティングIPアドレス(10.0.0.11)を入力し[完了]をクリックします。

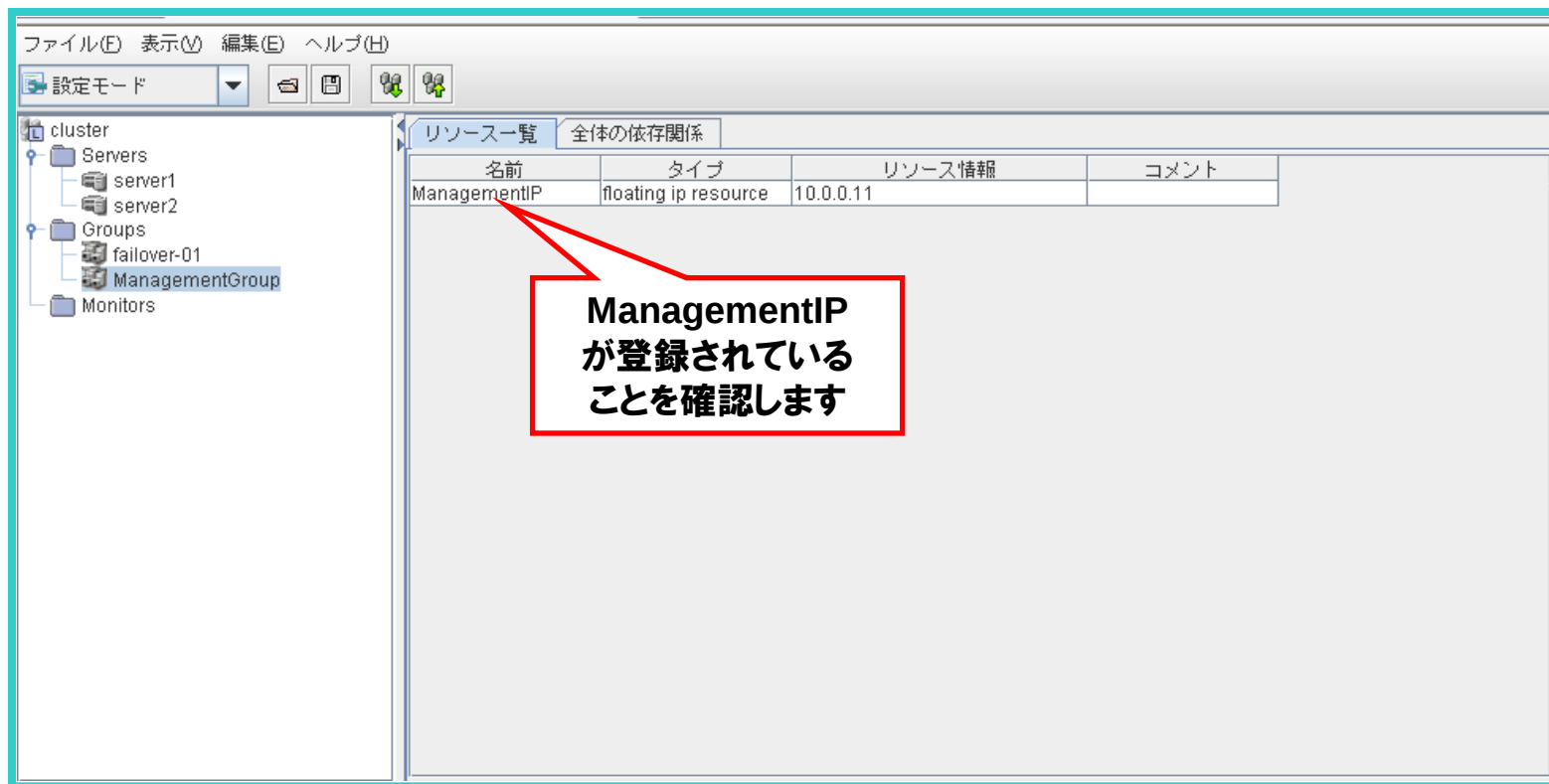


2. 構築手順

vii. CLUSTERPRO連携設定

7. 登録情報の確認を行います。

[リソース一覧]に、ManagementIPが登録されていることを確認します。



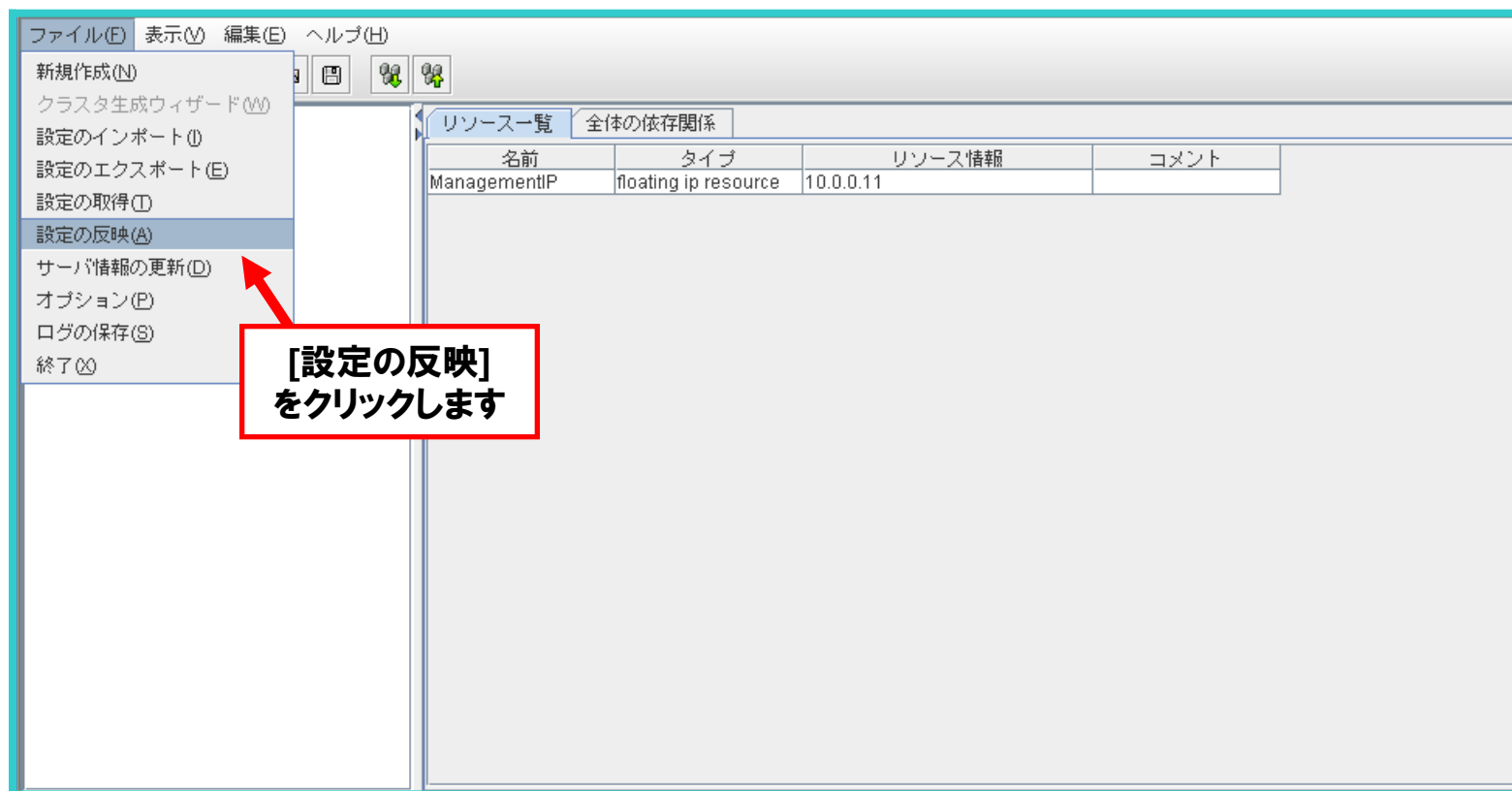
2. 構築手順

vii. CLUSTERPRO連携設定

⑦ クラスタ起動

1. 接続しているサーバにクラスタ構成情報を反映します。

[ファイル]メニューから[設定の反映]を選択してクリックします。



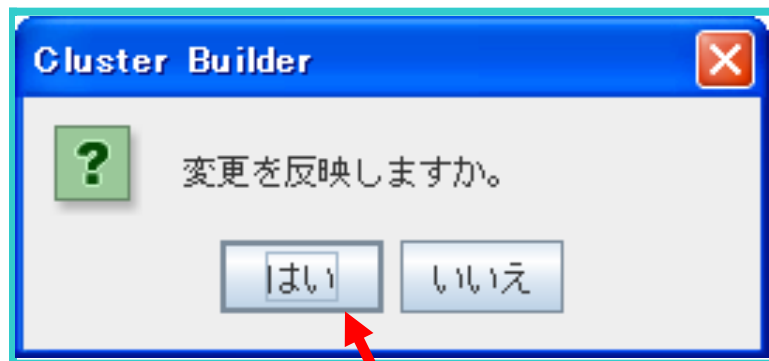
2. 構築手順

vii. CLUSTERPRO連携設定

2. 確認ダイアログが表示されます。 [はい]をクリックします。

反映に成功すると確認ダイアログが表示されます。「了解」をクリックしてダイアログを閉じます。

※ 変更した情報によって表示されるメッセージが異なりますので、
表示されたメッセージにしたがって操作を行ってください。
詳細は、CLUSTERPRO X 3.1 for Linux マニュアルを参照してください。

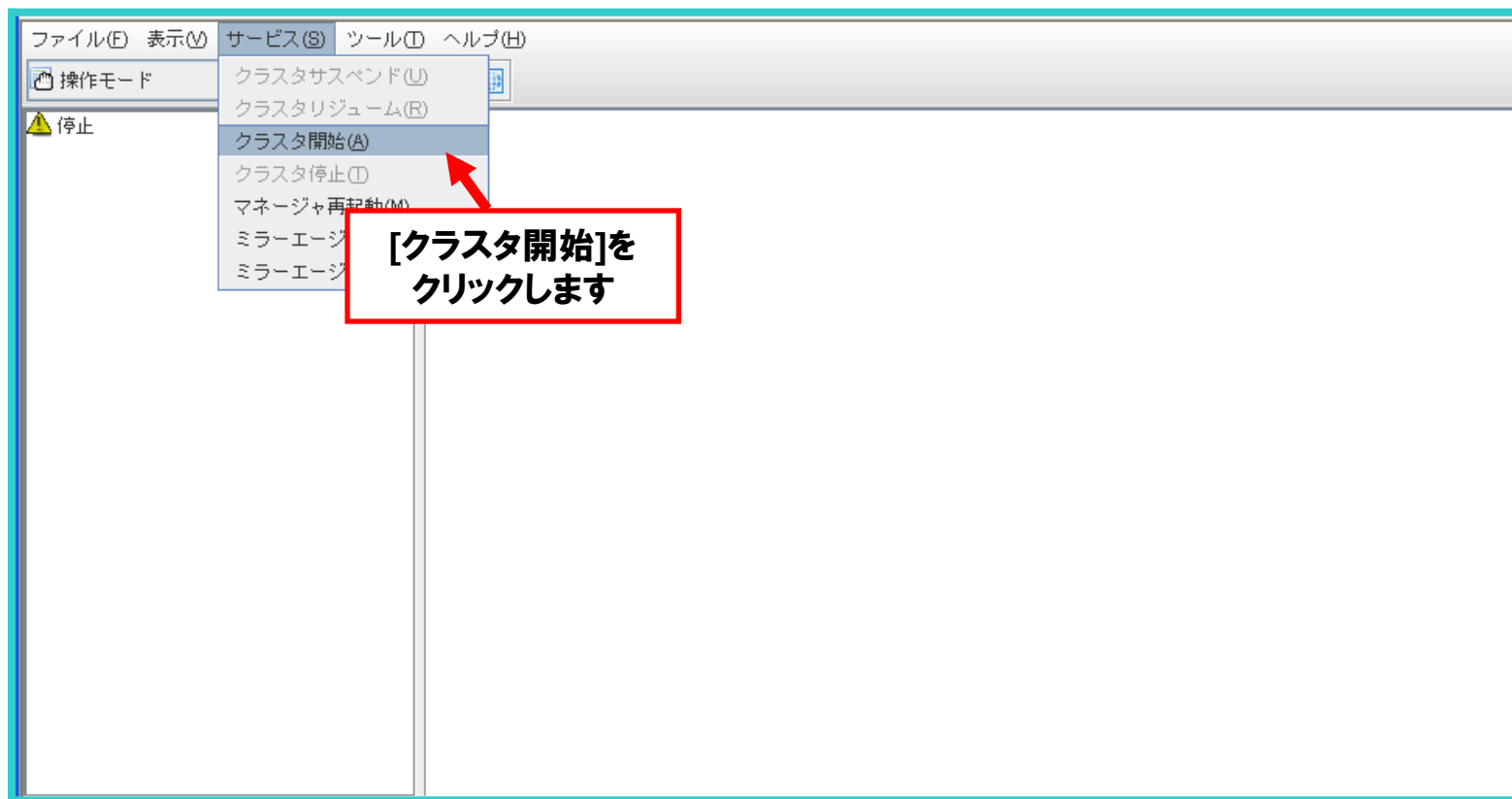


[はい]を
クリックします

2. 構築手順

vii. CLUSTERPRO連携設定

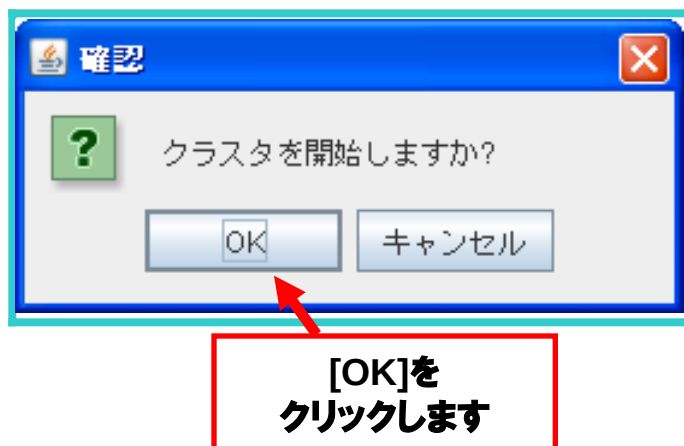
3. WebManager の「表示」メニューより「操作モード」を選択し、
タイトルバーの[サービス]から[クラスタ開始]を選択し、クリックします。



2. 構築手順

vii. CLUSTERPRO連携設定

4. クラスタの開始確認ダイアログが表示されるので[OK]をクリックします。



2. 構築手順

vii. CLUSTERPRO連携設定

5. クラスタが開始されます。

クラスタの情報が WebManager に表示されます。

以上でCLUSTERPROの設定は完了です。

The screenshot displays the CLUSTERPRO WebManager interface. On the left is a tree view with nodes: cluster, Servers (server1, server2), Groups (ManagementGroup, failover-01), and Monitors. The main panel shows the status for the selected 'cluster'. It includes a table for group status and monitoring resource status.

クラスタ: cluster		
	server1	server2
グループステータス		
ManagementGroup	起動済	停止済
failover-01	起動済	停止済
モニタリソースステータス		
mrw1	正常	正常
userw	正常	正常

3. 動作確認

i. 障害試験事前確認

システムが正常に動作していることを確認するために以下の手順を行います。

1. StorageSaverデーモンプロセス(srgd)が起動していることを確認します。

※リソース監視モニタ(srgping)、プロセス監視デーモン(srgwatch)についても同様に確認します。

```
# /bin/ps -ef | /bin/grep srg  
/opt/HA/SrG/local/bin/srgwatch  
/opt/HA/SrG/bin/srgd  
srgping
```

2. RENS連携用プロセス(ssdiagd)が起動していることを確認します。

```
# /bin/ps -ef | /bin/grep ssdiagd  
/opt/HA/SrG/bin/ssdiagd -c 60
```

3. 動作確認

i. 障害試験事前確認

3. StorageSaverが正常に動作していることを確認します。

```
# /opt/HA/SrG/bin/srgadmin
(monitor status = TRUE)
=====
type : device          : HostBusAdapter      : L status : P status : Online status
=====;=====;=====;=====;=====;=====
VG  : PSEUDO_VG001      : ---                : up
PV  : /dev/sda          : pci-0000:15:00.0    : up       : up       : extended
PV  : /dev/sdb          : pci-0000:13:00.0    : up       : up       : extended
VG  : VolGroup001       : ---                : up
PV  : /dev/sdc          : pci-0000:15:00.0    : up       : up       : extended
PV  : /dev/sdd          : pci-0000:13:00.0    : up       : up       : extended
```

4. RENSが起動していることを確認します。

```
# /bin/ps -ef | /bin/grep rens
/opt/mcl/rens/bin/rensd
```

3. 動作確認

i. 障害試験事前確認

5. RENSが正常に動作していることを確認します。

```
# /opt/mcl/rens/bin/rensadmin show
```

ID	ResourceName	Alias	Status	LastUpdateTime	MonitorName
0	NEC:iStorage_XXXX:XXXXXXXXXXXXXXXXXXXX:XXXX	dda	up	XXXX/XX/XX XX:XX:XX	sps
1	0000:15:00.0	hostx	up	XXXX/XX/XX XX:XX:XX	lpfc
2	NEC:iStorage_XXXX:XXXXXXXXXXXXXXXXXXXX:XXXX	ddb	up	XXXX/XX/XX XX:XX:XX	sps
3	0000:13:00.0	hostx	up	XXXX/XX/XX XX:XX:XX	lpfc
4	PSEUDO_VG001_status	PSEUDO_VG001_status	up	XXXX/XX/XX XX:XX:XX	ssdiagd
5	VolGroup001_status	VolGroup001_status	up	XXXX/XX/XX XX:XX:XX	ssdiagd

ID	TargetName	Type	Priority
0	syslog	syslog	middle
1	textlog	textlog	middle
2	clpx	clusterpro	middle low

ID	MonitorName	Pid	Status	Commandline
0	targetregclpd	XXXXX	run	/opt/mcl/rens/bin/targetregclpd -f /opt/mcl/rens/conf/targetclp.conf
1	sps	XXXXX	run	/opt/mcl/rens/bin/spsmon -n sps
2	lpfc	XXXXX	run	/opt/mcl/rens/bin/fcmon -n lpfc
3	ssdiagd	XXXXX	run	/opt/mcl/rens/bin/hamon -n ssdiagd

※Statusがunknownになっているリソースがある場合は下記コマンドを実行して最新の状態を取得してください。

```
# /opt/mcl/rens/bin/rensadmin show -k
```

3. 動作確認

i. 障害試験事前確認

6. StoragePathSaviorの監視機能(パス巡回デーモン)が起動していることを確認します。

```
# /bin/ps -ef | /bin/grep dd_daemon  
dd_daemon
```

7. 各パスの状態が正常(ACT)であることを確認します。

```
# /bin/cat /proc/scsi/sps/dd*  
    <省略>  
path-info:x Host:scsi:0 Channel:0 Id:0 Lun:0 Priority:1 Watch:Enable Status:ACT  
path-info:x Host:scsi:0 Channel:0 Id:1 Lun:0 Priority:2 Watch:Enable Status:ACT
```

3. 動作確認

i. 障害試験事前確認

8. CLUSTERPROのステータス表示コマンドでフェイルオーバーグループが現用系で起動されていることを確認します。

```
# /usr/sbin/clpstat
===== CLUSTER STATUS =====
Cluster : cluster
<server>
*server1 .....: Online
  lanhb1      : Normal    LAN Heartbeat
  lankhb1     : Normal    Kernel Mode LAN Heartbeat
  diskhb1     : Normal    DISK Heartbeat
server2 .....: Online
  lanhb1      : Normal    LAN Heartbeat
  lankhb1     : Normal    Kernel Mode LAN Heartbeat
  diskhb1     : Normal    DISK Heartbeat
<group>
ManagementGroup .: Online
  current      : server1
ManagementIP     : Online
failover-01 .....: Online
  current      : server1
  disk1        : Online
  fip1         : Online
<monitor>
mrw1             : Normal
userw            : Normal    user mode monitor
=====
```

3. 動作確認

i. 障害試験事前確認

9. WebManagerを起動して、正常に動作していることを確認します。

WebブラウザのアドレスバーにCLUSTERPRO Serverをインストールしたサーバの実IPアドレスとポート番号を入力します。 ※今回の例ではServer1(現用系)のアドレスとポート番号を入力します。

cluster

Servers

server1

server2

Groups

ManagementGroup

failover-01

Monitors

クラスタ: cluster

	server1	server2
グループステータス		
ManagementGroup	起動済	停止済
failover-01	起動済	停止済
モニタリソースステータス		
mrw1	正常	正常
userw	正常	正常

10. 以上で擬似障害発生前の事前確認は終了です。

3. 動作確認

ii. FC抜線による障害発生時のシステム連携確認

FC抜線により障害を発生させ、ディスクにアクセスができなくなった際に
フェイルオーバーが発生することを確認します。
ここではその動作確認手順について記載します。

- 1. 『3 - i 障害試験事前確認』の手順を行い、システムが正常に動作していることを確認してください。
- 2. FC抜線を行い、片系障害を発生させます。
今回はFC2を抜線した場合について記載します。
- 3. 約3分後に障害を検出していることを確認します。

```
# /opt/HA/SrG/bin/srgadmin
(monitor status = TRUE)

=====
type : device          : HostBusAdapter    : L status : P status : Online status
=====;=====;=====;=====;=====;=====
VG  : PSEUDO_VG001      : ---              : suspend
PV  : /dev/sda          : pci-0000:15:00.0 : up       : up       : extended
PV  : /dev/sdb          : pci-0000:13:00.0 : down     : down     : extended
VG  : VolGroup001       : ---              : suspend
PV  : /dev/sdc          : pci-0000:15:00.0 : up       : up       : extended
PV  : /dev/sdd          : pci-0000:13:00.0 : down     : down     : extended
```

3. 動作確認

ii. FC抜線による障害発生時のシステム連携確認

4. syslogに各I/Oパスの障害検出ログが出力されていることを確認します。

```
# /bin/view /var/log/messeges
xx xx xx:xx:xx server1 srgd[xxxxx]: PV status change fail .[hwpath = pci-0000:13:00.0-fc-
0x2100001697120ca7:0x0000000000000000 : s.f = /dev/sdb].
xx xx xx:xx:xx server1 srgd[xxxxx]: PV status change fail .[hwpath = pci-0000:13:00.0-fc-
0x2100001697120ca7:0x0001000000000000 : s.f = /dev/sdd].
xx xx xx:xx:xx server1 hamon(ssdiagd)[xxxxx]: RENS detected the CRITICAL event.
<resource=PSEUDO_VG001_status (id=xxx)><event#=3><severity=CRITICAL><summary=
PSEUDO_VG001 : SUSPEND><event seq#=xxx>
```

5. syslogにSPSの障害検出ログが出力されていることを確認します。

```
# /bin/view /var/log/messeges
xx xx xx:xx:xx server1 kernel: sps: Warning: Detect ACT path fail /dev/dda (0) host:1 channel: 0 id: 0 lun:
0
xx xx xx:xx:xx server1 kernel: sps: Warning: Detect ACT path fail /dev/ddb (2) host:1 channel: 0 id: 0 lun:
1
```

3. 動作確認

ii. FC抜線による障害発生時のシステム連携確認

- 6. 続いて、FC1を抜線して両系障害を発生させます。
- 7. 約1分後に障害を検出し、StorageSaverのステータスがdownになることを確認します。
また、CLUSTERPROの設定により、OSのシャットダウンが行われることを確認します。

```
# /opt/HA/SrG/bin/srgadmin
(monitor status = TRUE)

=====
=====
type : device          : HostBusAdapter      : L status : P status : Online status
=====;=====;=====;=====;=====;=====
===
VG  : PSEUDO_VG001      : ---              : down
PV  : /dev/sda          : pci-0000:15:00.0 : down      : down      : extended
PV  : /dev/sdb          : pci-0000:13:00.0 : down      : down      : extended
VG  : VolGroup001       : ---              : down
PV  : /dev/sdc          : pci-0000:15:00.0 : down      : down      : extended
PV  : /dev/sdd          : pci-0000:13:00.0 : down      : down      : extended
```

※ CLUSTERPROの設定により、OSがシャットダウンするため、上記確認はできない場合があります。
ログ等の確認はOS再起動後に行います。

3. 動作確認

ii. FC抜線による障害発生時のシステム連携確認

8. 待機系ノードでフェイルオーバーグループが起動されることを確認します。

※手順8, 9は待機系ノードで行います。

```
# /bin/view /var/log/messages
```

<省略>

```
xx xx xx:xx:xx server2 clusterpro: <type: nm><event: 2> Server server1 has been stopped.
```

<省略>

```
xx xx xx:xx:xx server2 clusterpro: <type: rc><event: 61> Failover group failover-01 has completed.
```

3. 動作確認

ii. FC抜線による障害発生時のシステム連携確認

9. CLUSTERPROのステータス表示コマンドでフェイルオーバーグループが待機系で起動されていることを確認します。

```
# /usr/sbin/clpstat
===== CLUSTER STATUS =====
Cluster : cluster
<server>
server1 .....: Offline
  lanhb1      : Unknown      LAN Heartbeat
  lankhb1     : Unknown      Kernel Mode LAN Heartbeat
  diskhb1     : Unknown      DISK Heartbeat
*server2 .....: Caution
  lanhb1      : Caution      LAN Heartbeat
  lankhb1     : Caution      Kernel Mode LAN Heartbeat
  diskhb1     : Caution      DISK Heartbeat
<group>
ManagementGroup .: Online
  current      : server2
  ManagementIP  : Online
failover-01 .....: Online
  current      : server2
  disk1        : Online
  fip1         : Online
<monitor>
mrw1          : Normal
userw         : Normal      user mode monitor
=====
```

3. 動作確認

ii. FC抜線による障害発生時のシステム連携確認

10. シャットダウンしたserver1に抜線したFCを接続後、起動させてsyslogに以下のメッセージが出力されていることを確認します。

※障害を検出するタイミングや、メッセージを受け取るタイミングによってsyslogメッセージの出力順序は変更になることがあります。

```
# /bin/view /var/log/messages
```

```
xx xx xx:xx:xx server1 srgd[xxxxx]: VG status change down .(vg=PSEUDO_VG001)
```

```
xx xx xx:xx:xx server1 srgd[xxxxx]: VG status change down .(vg=VolGroup001)
```

```
xx xx xx:xx:xx server1 hamon(ssdiagd)[xxxxx]: RENS detected the CRITICAL event.
```

```
<resource=PSEUDO_VG001_status (id=x)><event#=2><severity=CRITICAL>
```

```
<summary= PSEUDO_VG001 : DOWN><event seq#=xxxx>
```

```
xx xx xx:xx:xx server1 hamon(ssdiagd)[xxxxx]: --RENS-- INFO: Succeeded to send notification. Target = clpx.
```

```
xx xx xx:xx:xx server1 clusterpro: <type: mm><event: 903> An error of HA/SS type and PSEUDO_VG001_status  
device has been detected. (PSEUDO_VG001 : DOWN)
```

```
xx xx xx:xx:xx server1 clusterpro: <type: mm><event: 905> An error has been detected in monitoring  
PSEUDO_VG001_status. (-1)
```

```
xx xx xx:xx:xx server1 clusterpro: <type: apisv><event: 11> There was a request to shutdown server from the  
mm(IP=::ffff:xx.xx.xx.xx).
```

① StorageSaverがVG downを検出

② RENSがVG down通知を受け、CLUSTERPROに通知

③ CLUSTERPROがVG down通知を受け、フェイルオーバーを開始

11. 以上でシステム連携の確認は終了です。

4. 付録

i. サーバ管理基盤(RENS)を利用せず、StorageSaverとCLUSTERPROを直接連携する方法

RENSとの連携を行わない場合、CLUSTERPROのカスタムモニタリソースおよびStorageSaverのクラスタウェア連携用コマンド(srgstat)を利用することでStorageSaverとCLUSTERPROを直接連携させることが可能です。ここではその設定手順および動作確認方法を記載します。

※一般Linuxサーバ環境等、RENSを使わず、StorageSaverとCLUSTERPROを直接連携させる場合、RENS以外の設定については変更ありませんので、LVMやStorageSaverの設定などは、本資料2章の構築手順を参照してください。

4. 付録

i. サーバ管理基盤(RENS)を利用せず、StorageSaverとCLUSTERPROを直接連携する方法

① 設定手順

CLUSTERPROのカスタムモニタリソースにStorageSaverのクラスタウェア連携用コマンド(srgstat)起動処理を登録します。ここではその設定手順を記載します。

この方式では待機系も含めて全てのノードでsrgstatが起動し、監視を開始します。

1. StorageSaverのシステム定義ファイル(srg.config)にて

VG_FAULT_ACTIONがACTION_NONEになっていることを確認します。

※VG_FAULT_ACTIONがACTION_NONEになっていない場合のみ以下2～4の手順で設定変更およびデーモンプロセス(srgd)の再起動を行ってください。

```
# /bin/cat /var/opt/HA/SrG/conf/srg.config | /bin/grep "VG_FAULT_ACTION"  
VG_FAULT_ACTION      ACTION_NONE
```

4. 付録

i. サーバ管理基盤(RENS)を利用せず、StorageSaverとCLUSTERPROを直接連携する方法

※VG_FAULT_ACTIONがACTION_NONEになっている場合は以下2～4の手順を行う必要はありません。

2. システム定義ファイル(srg.config)を編集します。

※今回の例はviコマンドを使用して編集を行います。

```
# /bin/vi /var/opt/HA/SrG/conf/srg.config
```

3. システム定義ファイル(srg.config)が正しく変更されていることを確認します。

```
# /bin/cat /var/opt/HA/SrG/conf/srg.config | /bin/grep "VG_FAULT_ACTION"  
VG_FAULT_ACTION      ACTION_NONE
```

4. デーモンプロセス(srgd)を再起動します。

```
# /etc/init.d/srgctl restart
```

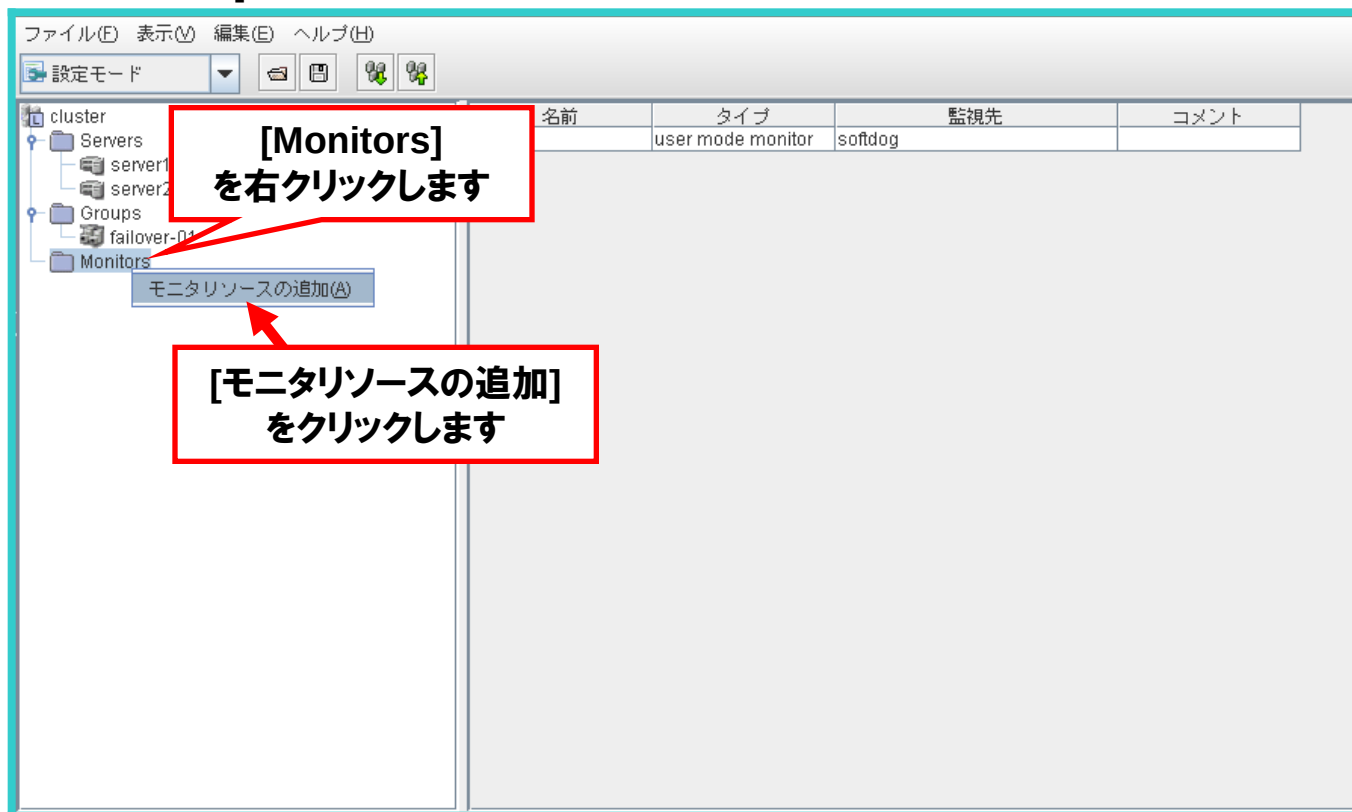
4. 付録

i. サーバ管理基盤(RENS)を利用せず、StorageSaverとCLUSTERPROを直接連携する方法

5. Builderを起動します。

(起動方法については『2 - vii クラスタ環境構築』を参照して下さい。)

6. Builder左部分に表示されているツリービューで[Monitors]を右クリックし、
[モニタリソースの追加]をします。



4. 付録

i. サーバ管理基盤(RENS)を利用せず、StorageSaverとCLUSTERPROを直接連携する方法

7. [タイプ]から "custom monitor" を選択し、
[名前]にカスタムモニタリソース名(srgstat_mon) を入力して、[次へ]をクリックします。

モニタリソースの定義

ステップ

情報

監視(共通)

監視(固有)

回復動作

モニタリソース定義

タイプ(T) custom monitor

名前(N) srgstat_mon

コメント(C)

ライセンス情報取得(L)

説明

モニタリソースの種類を選択して名前を入力してください。

[custom monitor] を選択します

カスタムモニタリソース名を入力します

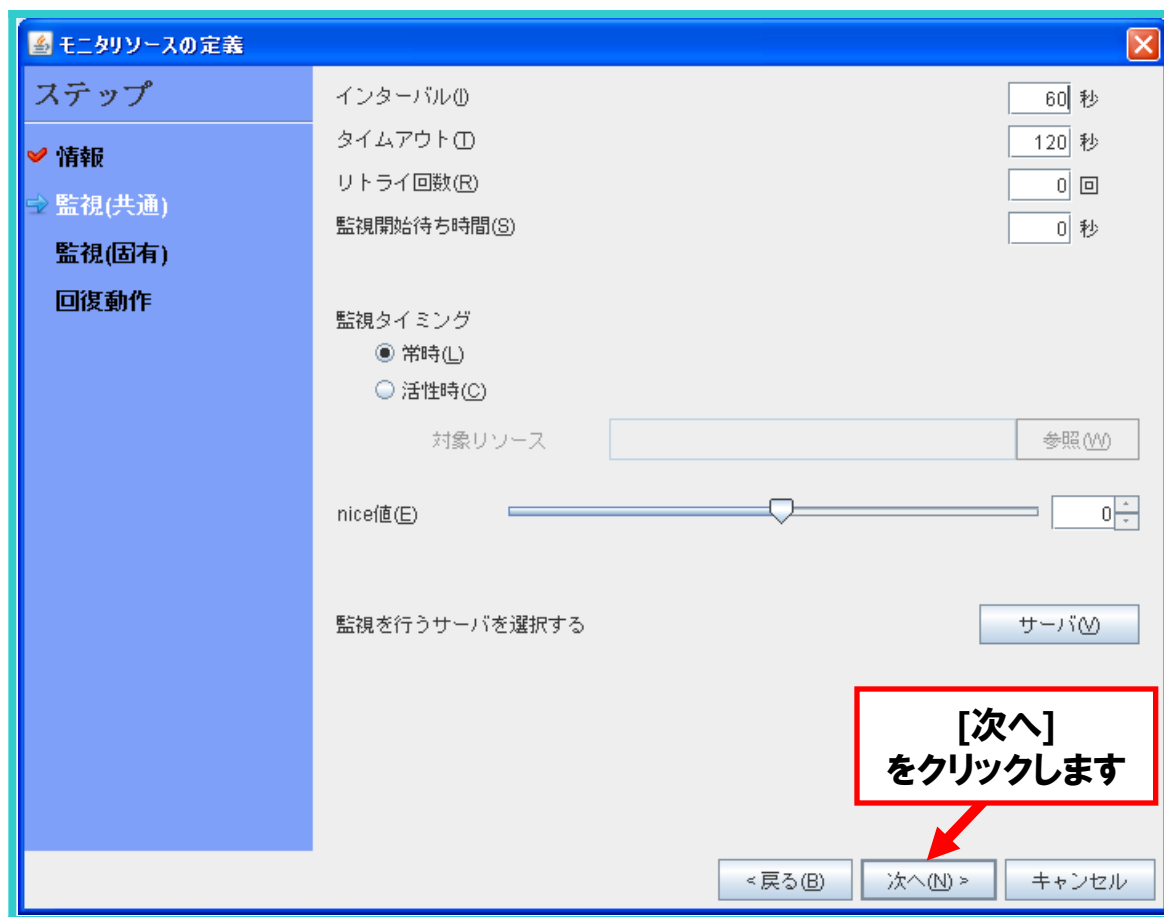
[次へ] をクリックします

< 戻る(B) 次へ(N) > キャンセル

4. 付録

i. サーバ管理基盤(RENS)を利用せず、StorageSaverとCLUSTERPROを直接連携する方法

8. [次へ]をクリックします。



4. 付録

i. サーバ管理基盤(RENS)を利用せず、StorageSaverとCLUSTERPROを直接連携する方法

9. "ユーザアプリケーション" を選択し、[ファイル]に srgstat 起動処理(/opt/HA/SrG/bin/srgstat -w 30)を入力します。
監視タイプは "非同期" を選択します。
設定完了後、[次へ]をクリックします。

The screenshot shows the 'モニタリソースの定義' (Monitoring Source Definition) window. On the left, the 'ステップ' (Steps) pane shows '監視(固有)' (Monitoring (Unique)) is selected. The main area has the following settings:

- ユーザアプリケーション(U)** (User Application) is selected with a radio button. A red box highlights this option with the text "[ユーザアプリケーション]を選択します" (Select [User Application]).
- この製品で作成したスクリプト(W)** (Script created by this product) is unselected.
- ファイル(F)** (File) is set to `/opt/HA/SrG/bin/srgstat -w 30`. A red box highlights the text field with the text "[srgstat 起動処理]を入力します" (Enter [srgstat startup process]).
- 監視タイプ** (Monitoring Type) is set to **非同期(M)** (Asynchronous). A red box highlights the radio button with the text "[非同期]を選択します" (Select [Asynchronous]).
- 同期(S)** (Synchronous) is unselected.
- ログ出力先(L)** (Log output destination) is empty.
- ローテートする(R)** (Rotate) is unchecked.
- ローテートサイズ(S)** (Rotate size) is empty.
- 正常な戻り値(M)** (Normal return value) is set to `0`.

At the bottom, the **次へ(N) >** (Next) button is highlighted with a red box and the text "[次へ]をクリックします" (Click [Next]).

4. 付録

i. サーバ管理基盤(RENS)を利用せず、StorageSaverとCLUSTERPROを直接連携する方法

10. 回復動作の一覧から "最終動作のみ実行" を選択し、回復対象の[参照]をクリックして、"LocalServer" を選択します。
最終動作の一覧から "クラスタサービス停止とOSシャットダウン" を選択します。
設定完了後、[完了]をクリックします。

モニタリソースの定義

ステップ

- 情報
- 監視(共通)
- 監視(固有)
- 回復動作

回復動作(E) 最終動作のみ実行

回復対象 LocalServer 参照(W)

回復スクリプト実行回数(Y) 0 回

☐ 再活性前にスクリプトを実行する(Y)

最大再活性回数(R) 0 回

☐ フェイルオーバー実行前にスクリプトを実行する(O)

☐ フェイルオーバー実行前にマイグレーションを実行する(C)

最大フェイルオーバー回数(I) 0 回

☐ 最終動作前にスクリプトを実行する(O)

最終動作(F) クラスタサービス停止とOSシャットダウン

スクリプト設定

< 戻る(B) 完了 キャンセル

[最終動作のみ実行]を選択し、
[参照]をクリックして、
[LocalServer]を選択します

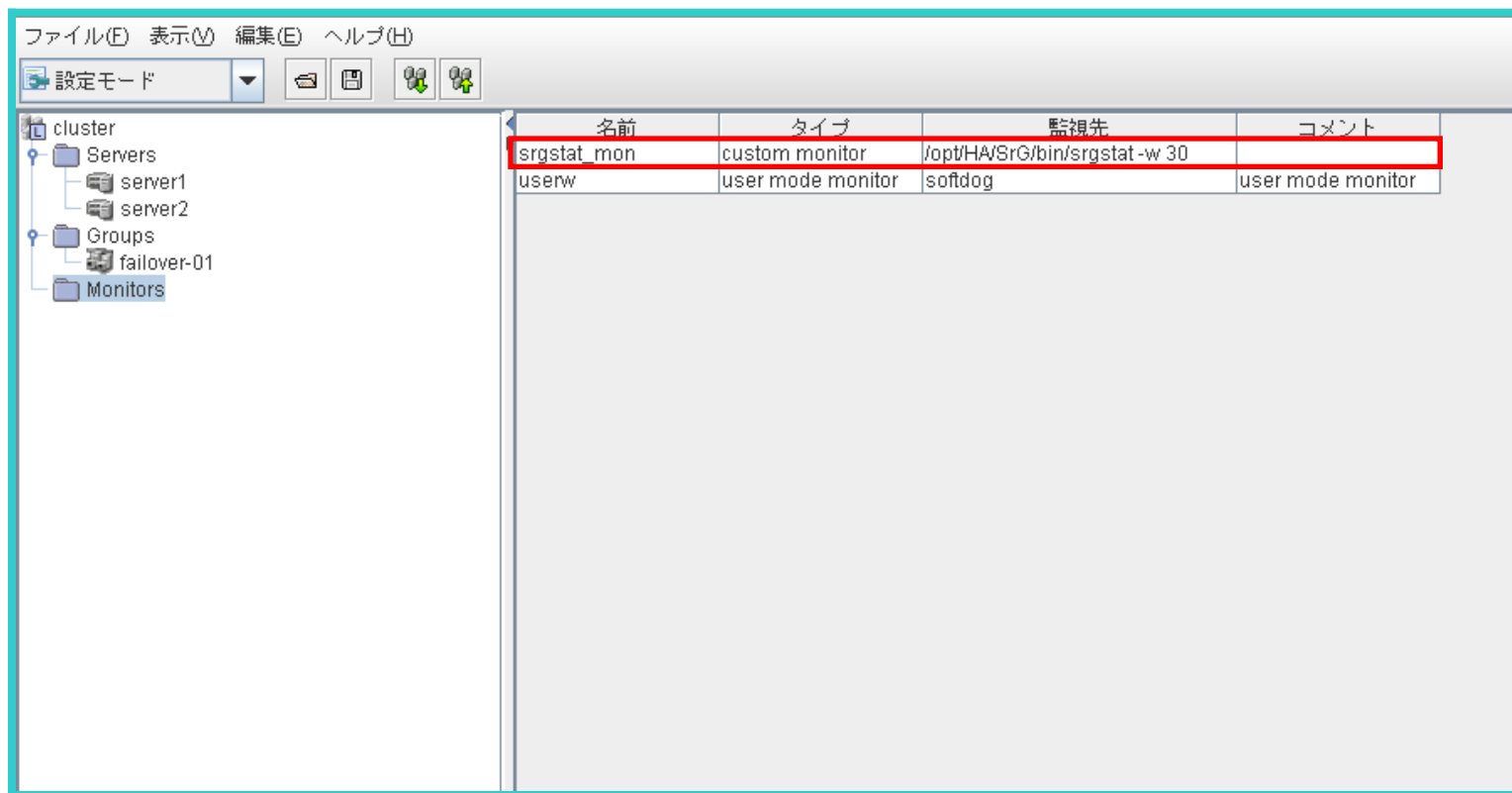
[クラスタサービス停止と
OSシャットダウン]
を選択します

[完了]
をクリックします

4. 付録

i. サーバ管理基盤(RENS)を利用せず、StorageSaverとCLUSTERPROを直接連携する方法

11. [Monitors]をクリックし、“カスタムモニタリソース”が登録されていることを確認します。



4. 付録

i. サーバ管理基盤(RENS)を利用せず、StorageSaverとCLUSTERPROを直接連携する方法

12. 設定ファイルのアップロードを行います。

(アップロードの方法については『 2 - vii ⑥クラスタ起動 』を参照して下さい。)

13. クラスタを起動します。

(クラスタの起動方法については『 2 - vii ⑥クラスタ起動 』を参照して下さい。)

14. エラーが出ていないことを確認します。

15. StorageSaverのクラスタウェア連携用コマンド(srgstat)が起動していることを確認します。

```
# /bin/ps -ef | /bin/grep srgstat  
/opt/HA/SrG/bin/srgstat -w 30
```

16. 以上で設定完了です。

4. 付録

ii. FC抜線による障害発生時のシステム連携確認

② 動作確認

FC 抜線により障害を発生させ、ディスクにアクセスすることができなくなった際に
フェイルオーバが発生することを確認します。
ここではその動作確認手順について記載します。

1. 『 3 - i 障害試験事前確認 』の手順を行い、システムが正常に動作していることを確認してください。
2. FC抜線を行い、片系障害を発生させます。
今回はFC2を抜線した場合について記載します。

4. 付録

ii. FC抜線による障害発生時のシステム連携確認

3. 約3分後に片系障害を検出することを確認します。

```
# /opt/HA/SrG/bin/srgadmin
(monitor status = TRUE)

=====
=====
type : device          : HostBusAdapter    : L status : P status : Online status
=====;=====;=====;=====;=====;=====
=====
VG  : PSEUDO_VG001      : ---          : suspend
PV  : /dev/sda          : pci-0000:15:00.0 : up       : up       : extended
PV  : /dev/sdb          : pci-0000:13:00.0 : down     : down     : extended
VG  : VolGroup001      : ---          : suspend
PV  : /dev/sdc          : pci-0000:15:00.0 : up       : up       : extended
PV  : /dev/sdd          : pci-0000:13:00.0 : down     : down     : extended
```

4. 付録

ii. FC抜線による障害発生時のシステム連携確認

4. syslogに各I/Oパスの障害検出ログが出力されていることを確認します。

```
# /bin/view /var/log/messages
xx xx xx:xx:xx server1 srgd[xxxxx]: PV status change fail .
    [hwpath = pci-0000:13:00.0-fc-0x2900001697120ca7:0x0000000000000000: s.f = /dev/sdb].
xx xx xx:xx:xx server1 srgd[xxxxx]: PV status change fail .
    [hwpath = pci-0000:13:00.0-fc-0x2900001697120ca7:0x0001000000000000: s.f = /dev/sdd].
```

5. syslogにSPSの障害検出ログが出力されていることを確認します。

```
# /bin/view /var/log/messeges
xx xx xx:xx:xx server1 kernel: sps: Warning: Detect ACT path fail /dev/dda (0) host:1 channel: 0 id: 0 lun:
    0
xx xx xx:xx:xx server1 kernel: sps: Warning: Detect ACT path fail /dev/ddb (2) host:1 channel: 0 id: 0 lun:
    1
```

4. 付録

ii. FC抜線による障害発生時のシステム連携確認

- 6. 続いて、FC1を抜線して両系障害を発生させます。
- 7. 約1分後に障害を検出し、StorageSaverのステータスがdownになることを確認します。
また、CLUSTERPROの設定により、OSのシャットダウンが行われることを確認します。

```
# /opt/HA/SrG/bin/srgadmin
(monitor status = TRUE)
=====
=====
type : device          : HostBusAdapter    : L status : P status : Online status
=====;=====;=====;=====;=====;=====
=====
VG  : PSEUDO_VG001      : ---          : down
PV  : /dev/sda          : pci-0000:15:00.0 : down : down : extended
PV  : /dev/sdb          : pci-0000:13:00.0 : down : down : extended
VG  : VolGroup001       : ---          : down
PV  : /dev/sdc          : pci-0000:15:00.0 : down : down : extended
PV  : /dev/sdd          : pci-0000:13:00.0 : down : down : extended
```

※ CLUSTERPROの設定により、OSがシャットダウンするため、上記確認はできない場合があります。
ログ等の確認はOS再起動後に行います。

4. 付録

ii. FC抜線による障害発生時のシステム連携確認

8. 待機系ノードでフェイルオーバーグループが起動されることを確認します。

※手順8,9は待機系ノードで行います。

```
# /bin/view /var/log/messages
```

<省略>

```
xx xx xx:xx:xx server2 clusterpro: <type: nm><event: 2> Server server1 has been stopped.
```

<省略>

```
xx xx xx:xx:xx server2 clusterpro: <type: rc><event: 61> Failover group failover-01 has completed.
```

4. 付録

ii. FC抜線による障害発生時のシステム連携確認

9. CLUSTERPROのステータス表示コマンドでフェイルオーバーグループが待機系で起動されていることを確認します。

```
# /usr/sbin/clpstat
===== CLUSTER STATUS =====
Cluster : cluster
<server>
server1 .....: Offline
  lanhb1      : Unknown      LAN Heartbeat
  lankhb1     : Unknown      Kernel Mode LAN Heartbeat
  diskhb1     : Unknown      DISK Heartbeat
*server2 .....: Caution
  lanhb1      : Caution      LAN Heartbeat
  lankhb1     : Caution      Kernel Mode LAN Heartbeat
  diskhb1     : Caution      DISK Heartbeat
<group>
ManagementGroup .: Online
  current     : server2
ManagementIP   : Online
failover-01 .....: Online
  current     : server2
  disk1       : Online
  fip1        : Online
<monitor>
genw           : Normal
userw          : Normal      user mode monitor
=====
```

4. 付録

ii. FC抜線による障害発生時のシステム連携確認

10. シャットダウンしたserver1に抜線したFCを接続後、起動させてsyslogに以下のメッセージが出力されていることを確認します。

※障害を検出するタイミングや、メッセージを受け取るタイミングによってsyslogメッセージの出力順序は変更になることがあります。

```
# /bin/view /var/log/messages
```

```
xx xx xx:xx:xx server1 srgd[xxxxx]: VG status change down .(vg=PSEUDO_VG001)
```

```
xx xx xx:xx:xx server1 srgd[xxxxx]: VG status change down .(vg=VolGroup001)
```

```
xx xx xx:xx:xx server1 srgstat[xxxxx]: found VG status is down
```

```
xx xx xx:xx:xx server1 clusterpro: <type: rm><event: 9> Detected an error in monitoring genw1.  
(11 : Asynchronous process does not exist. (pid=xxxxx))
```

```
xx xx xx:xx:xx server1 clusterpro: <type: apisv><event: 11> There was a request to shutdown  
server from the clprm process(IP=::ffff:xxx.xxx.xxx.xxx).
```

① StorageSaverがVG downを検出

② srgstatがVG down通知を検出、CLUSTERPROに通知

③ CLUSTERPROがVG down通知を受け、フェイルオーバーを開始

11. 以上で動作確認完了です。

Empowered by Innovation

NEC