

CLUSTERPRO® X *for Linux*

ソフトウェア構築ガイド (MySQL)

2025.8.27
第2.0版

CLUSTERPRO

改版履歴

版	改版日付	章	内 容
1.0	2022/11/17		新規作成
1.1	2023/2/8	3.7	④のmysqld.serviceのユニットファイルの編集に、3.8.1章③を統合、誤記修正。
		3.8.1	• ③を削除し、3.7章の④に統合 ④のmysqld.serviceの起動手順を削除
		3.8.3	⑤のmysqld.serviceの停止手順を追加
2.0	2025/8/27	全体	CLUSTERPRO X5.3、MySQL 8.4対応

免責事項

本書の内容は、予告なしに変更されることがあります。

日本電気株式会社は、本書の技術的もしくは編集上の間違い、欠落について、一切責任をおいません。

また、お客様が期待される効果を得るために、本書に従った導入、使用および使用効果につきましては、お客様の責任とさせていただきます。

本書に記載されている内容の著作権は、日本電気株式会社に帰属します。本書の内容の一部または全部を日本電気株式会社の許諾なしに複製、改変、および翻訳することは禁止されています。

商標情報

CLUSTERPRO® X は日本電気株式会社の登録商標です。

Linuxは、Linus Torvalds氏の米国およびその他の国における、登録商標または商標です。

MySQLは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

本書に記載されたその他の製品名および標語は、各社の商標または登録商標です。

その他のシステム名、社名、製品名等はそれぞれの会社の商標及び登録商標です。

目次

1. はじめに	i
対象読者と目的	i
適用範囲	i
CLUSTERPRO マニュアル体系	ii
本書の表記規則	iii
最新情報の入手先	iv
2. 機能概要	1
2.1 構成概要	1
2.2 機能範囲	3
2.3 動作環境	3
3. 構築手順	4
3.1 前提環境	4
3.2 ハードウェア構成後の手順	5
3.2.1 パーティションの作成	5
3.2.2 ファイアウォールの設定確認	7
3.3 環境構築の流れ	8
3.4 CLUSTERPRO Xソフトウェアのインストール	9
3.5 クラスタの作成	10
3.6 サーバリソース作成	14
3.6.1 フェイルオーバーグループ	14
3.6.2 ディスクリソース	16
3.6.3 フローティングIPリソース	19
3.6.4 設定項目の反映	21
3.7 MySQLソフトウェアのインストール	24
3.8 データベースの作成	25
3.8.1 MySQLサーバの初期化	25
3.8.2 rootアカウントのパスワード設定	26
3.8.3 監視用のアカウント、データベースを作成する	27
3.9 MySQL用のサービスリソース作成	28
3.9.1 execリソースの作成	28
3.9.2 モニタリソースの作成	34
3.9.3 設定項目の反映	39

1. はじめに

対象読者と目的

『CLUSTERPRO® ソフトウェア構築ガイド』は、クラスタシステムに関して、システムを構築する管理者、およびユーザサポートを行うシステムエンジニア、保守員を対象にしています。

本書では、CLUSTERPRO環境下での動作確認が取れたソフトウェアをご紹介します。ここでご紹介するソフトウェアや設定例は、あくまで参考情報としてご提供するものであり、各ソフトウェアの動作保証をするものではありません。

適用範囲

本書は、以下の製品を対象としています。

CLUSTERPRO X 5.3 for Linux
CLUSTERPRO X 5.2 for Linux
CLUSTERPRO X 5.1 for Linux

CLUSTERPRO マニュアル体系

CLUSTERPRO のマニュアルは、以下の 5 つに分類されます。各ガイドのタイトルと役割を以下に示します。

『CLUSTERPRO X スタートアップガイド』 (Getting Started Guide)

すべてのユーザを対象読者とし、製品概要、動作環境、アップデート情報、既知の問題などについて記載します。

『CLUSTERPRO X インストール&設定ガイド』 (Install and Configuration Guide)

CLUSTERPRO を使用したクラスタシステムの導入を行うシステムエンジニアと、クラスタシステム導入後の保守・運用を行うシステム管理者を対象読者とし、CLUSTERPRO を使用したクラスタシステム導入から運用開始前までに必須の事項について説明します。実際にクラスタシステムを導入する際の順番に則して、CLUSTERPRO を使用したクラスタシステムの設計方法、CLUSTERPRO のインストールと設定手順、設定後の確認、運用開始前の評価方法について説明します。

『CLUSTERPRO X リファレンスガイド』 (Reference Guide)

管理者、および CLUSTERPRO を使用したクラスタシステムの導入を行うシステムエンジニアを対象とし、CLUSTERPRO の運用手順、各モジュールの機能説明およびトラブルシューティング情報等を記載します。『インストール&設定ガイド』を補完する役割を持ちます。

『CLUSTERPRO X メンテナンスガイド』 (Maintenance Guide)

管理者、および CLUSTERPRO を使用したクラスタシステム導入後の保守・運用を行うシステム管理者を対象読者とし、CLUSTERPRO のメンテナンス関連情報を記載します。

『CLUSTERPRO X ハードウェア連携ガイド』 (Hardware Feature Guide)

管理者、および CLUSTERPRO を使用したクラスタシステムの導入を行うシステムエンジニアを対象読者とし、特定ハードウェアと連携する機能について記載します。『インストール&設定ガイド』を補完する役割を持ちます。

本書の表記規則

本書では、「注」および「重要」を以下のように表記します。

注：は、重要ではあるがデータ損失やシステムおよび機器の損傷には関連しない情報を表します。

重要：は、データ損失やシステムおよび機器の損傷を回避するために必要な情報を表します。

関連情報：は、参照先の情報の場所を表します。

また、本書では以下の表記法を使用します。

表記	使用方法	例
[] 角かっこ	コマンド名の前後 画面に表示される語 (ダイアログ ボックス、メニューなど) の前後	[スタート] をクリックします。 [プロパティ] ダイアログ ボックス
コマンドライン中の [] 角かっこ	かっこ内の値の指定が省略可能であることを示します。	<code>clpstat -s[-h host_name]</code>
モノスペース フォント (courier)	コマンド ライン、関数、パラメータ	<code>clpstat -s</code>
モノスペース フォント太字 (courier)	ユーザが実際にコマンドプロンプトから入力する値を示します。	以下を入力します。 <code>clpcl -s -a</code>
モノスペース フォント (courier) 斜体	ユーザが有効な値に置き換えて入力する項目	<code>clpstat -s [-h host_name]</code>

最新情報の入手先

最新の製品情報については、以下のWebサイトを参照してください。

<https://jpn.nec.com/clusterpro/>

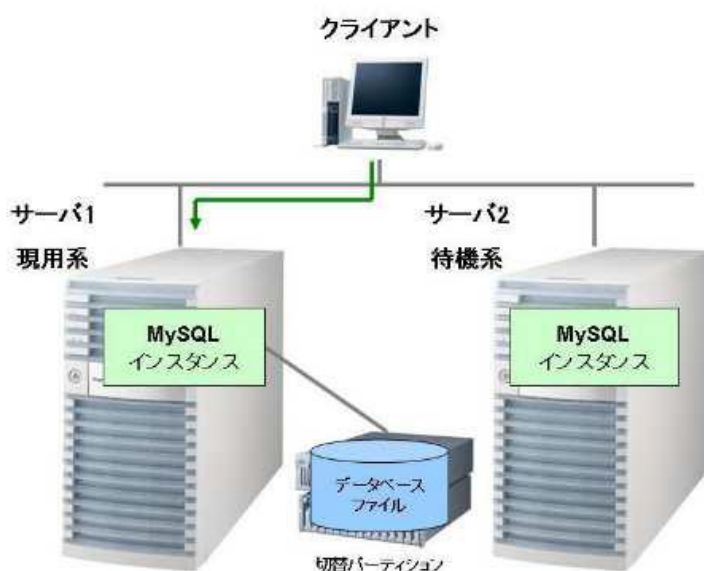
2. 機能概要

2.1 構成概要

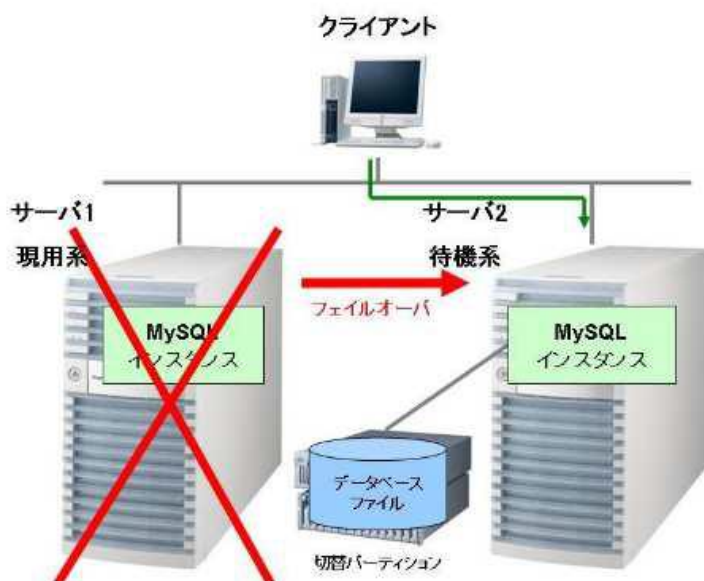
MySQLとCLUSTERPROを連携させることで、以下のクラスタを構成することができます。

【片方向スタンバイ型】

下図は、片方向スタンバイ型の構成例です。サーバ1を現用系、サーバ2を待機系として構成しています。クライアントからは、フローティングIP や仮想コンピュータ名を使用して接続します。

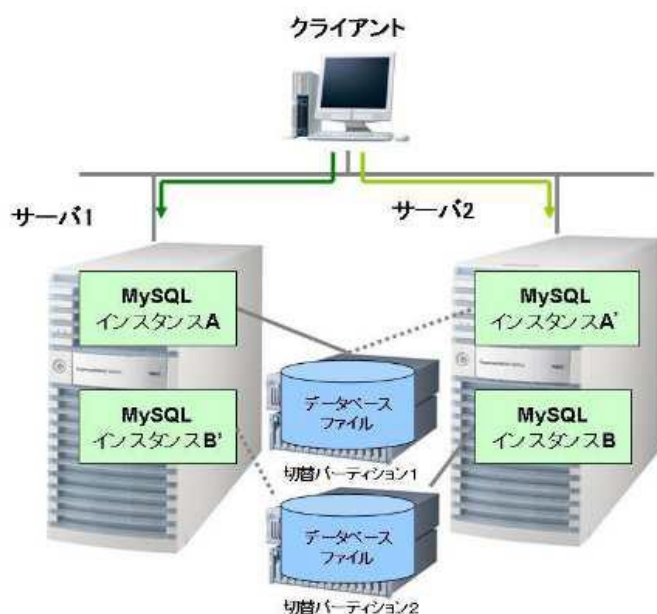


サーバ1で障害が発生すると、切替パーティションがサーバ1からサーバ2に移行します。移行後、サーバ2でMySQL インスタンスが立ち上がり、フェイルオーバーが完了します。フェイルオーバー後も、同一のフローティングIP や仮想コンピュータ名で接続が可能です。

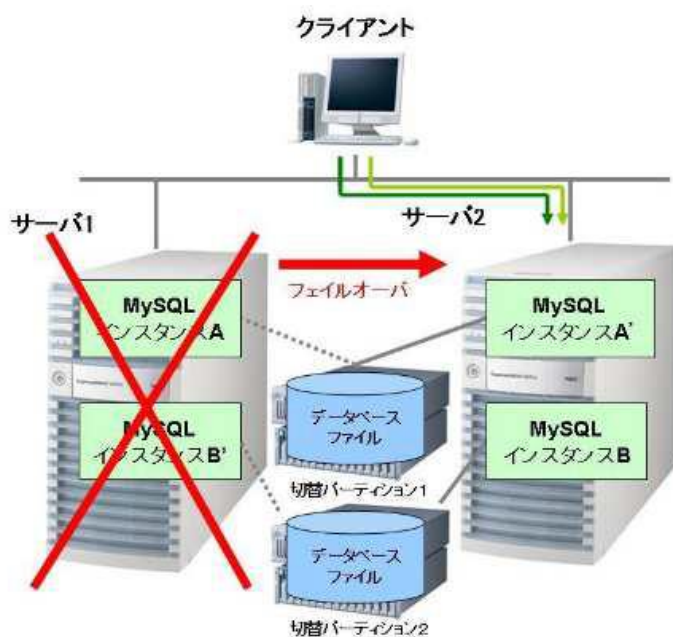


【双方向スタンバイ型】

下図は、双方向スタンバイ型の構成例です。サーバ1はMySQL インスタンスA の現用系であると同時にMySQL インスタンスB の待機系です。サーバ2はMySQL インスタンスA の待機系であると同時にMySQL インスタンスB の現用系です。クライアントからは、フローティングIP や仮想コンピュータ名を使用して接続します。



サーバ1で障害が発生すると、切替パーティション1がサーバ1からサーバ2に移行します。移行後、サーバ2でインスタンスA' が立ち上がり、フェイルオーバーが完了します。この時、サーバ2では2つのMySQL インスタンスが起動している状態になります。フェイルオーバー後も、同一のフローティングIP や仮想コンピュータ名で接続が可能です。



2.2 機能範囲

本書は、CLUSTERPRO X 5.3 for Linuxおよび、MySQL 8.4の組み合わせで検証しています。また、MySQL InnoDB Cluster、MySQL NDB Cluster 構成は対象外となります。

2.3 動作環境

下記のMySQL、CLUSTERPRO、OSのバージョンの組み合わせで動作確認を行っております。これ以外の組み合わせについては、お客様にて十分に動作確認を行ってください。

- ・MySQLのバージョン
8.4.5
- ・CLUSTERPROのバージョン
CLUSTERPRO X 5.3 for Linux（内部バージョン 5.3.0-1）
- ・OSの種類
Red Hat Enterprise Linux 8.4

3. 構築手順

3.1 前提環境

本書では、以下のような 2 サーバ構成のクラスタでの片方向スタンバイ環境を想定し、説明を行います。

- クラスタサーバ環境

	サーバ1(現用系)	サーバ2(待機系)
パブリックIPアドレス	173.8.100.3	173.8.100.4
インタコネクトIPアドレス	192.168.100.3	192.168.100.4
MySQLデータ用共有ディスク (切替パーティション)	/dev/sdb2 (マウントポイント: /var/lib/mysql)	
ハートビート用共有ディスク (切替パーティション)	/dev/sdb1	

- フェイルオーバーグループ情報

フローティングIPアドレス	173.8.100.100
MySQLデータ用共有ディスク (切替パーティション)	/dev/sdb2 (マウントポイント: /var/lib/mysql)
ハートビート用共有ディスク (切替パーティション)	/dev/sdb1

- データベース環境

データディレクトリ	/var/lib/mysql/data
エラーログ	/var/lib/mysql/mysqlld.log
オプションファイル	/var/lib/mysql/my.cnf

3.2 ハードウェア構成後の手順

3.2.1 パーティションの作成

ディスクハートビート用のパーティションと、MySQL データ格納用のパーティションを作成します。

パーティション		用途	備考
デバイス名	フォーマット		
/dev/sdb1	しない	ディスクハートビート	
/dev/sdb2	ext4	ディスクリソース	/var/lib/mysql にマウント

- ① ディスクハートビート用パーティションを作成します。本手順はサーバ1で実行します。

```
(パーティション・テーブルのリストを確認)
# fdisk -l

(パーティションを作成)
# fdisk /dev/sdb

コマンド (m でヘルプ): p                ← 領域テーブルの表示
ディスク /dev/sdb: 32 GiB, 34359738368 バイト, 67108864 セクタ
単位: セクタ (1 * 512 = 512 バイト)
セクタサイズ (論理 / 物理): 512 バイト / 512 バイト
I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト
ディスクラベルのタイプ: dos
ディスク識別子: 0xee5f47dc

コマンド (m でヘルプ): n                ← パーティションの新規作成
パーティションタイプ
  p   基本パーティション (0 プライマリ, 0 拡張, 4 空き)
  e   拡張領域 (論理パーティションが入ります)
選択 (既定値 p): p                    ← 基本パーティションを指定
パーティション番号 (1-4, 既定値 1): 1 ← パーティション番号 1 を指定
最初のセクタ (2048-67108863, 既定値 2048): ← 未指定(既定値を使用)
最終セクタ, +セクタ番号 または +サイズ{K,M,G,T,P} (2048-67108863, 既定値 67108863):
+10M ← サイズ10Mを指定

新しいパーティション 1 をタイプ Linux、サイズ 10 MiB で作成しました。

コマンド (m でヘルプ): w                ← パーティションの確定
パーティション情報が変更されました。
ioctl() を呼び出してパーティション情報を再読み込みします。
ディスクを同期しています。
```

- ② ディスクリソース用のパーティションを作成します。本手順はサーバ1で実行します。

```
(パーティションを作成)
# fdisk /dev/sdb

コマンド (m でヘルプ): p                ← 領域テーブルの表示
ディスク /dev/sdb: 32 GiB, 34359738368 バイト, 67108864 セクタ
単位: セクタ (1 * 512 = 512 バイト)
セクタサイズ (論理 / 物理): 512 バイト / 512 バイト
I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト
ディスクラベルのタイプ: dos
ディスク識別子: 0xee5f47dc

デバイス  起動  開始位置  終了位置  セクタ  サイズ  Id  タイプ
/dev/sdb1          2048    22527   20480    10M  83  Linux

コマンド (m でヘルプ): n                ← パーティションの新規作成
パーティションタイプ
  p   基本パーティション (1 プライマリ, 0 拡張, 3 空き)
  e   拡張領域 (論理パーティションが入ります)
選択 (既定値 p): p                      ← 基本パーティションを指定
パーティション番号 (2-4, 既定値 2): 2   ← パーティション番号 2 を指定
最初のセクタ (22528-67108863, 既定値 22528): ← 未指定(既定値を使用)
最終セクタ, +セクタ番号 または +サイズ {K,M,G,T,P} (22528-67108863, 既定値 67108863): ← 未指定(既定値を使用)

新しいパーティション 2 をタイプ Linux、サイズ 32 GiB で作成しました。

コマンド (m でヘルプ): w                ← パーティションの確定
パーティション情報が変更されました。
ioctl() を呼び出してパーティション情報を再読み込みします。
ディスクを同期しています。
```

- ③ ディスクリソース用のパーティションにext4ファイルシステムを作成します。本手順はサーバ1で実行します。

```
# /sbin/mkfs -t ext4 /dev/sdb2
```

- ④ ディスクリソース用のパーティションをマウントするディレクトリを作成します。ディレクトリの権限はMySQLソフトウェアインストール後に設定します。本手順はサーバ1、サーバ2で実行します。

```
# mkdir /var/lib/mysql
```

- ⑤ サーバを再起動します。本手順はサーバ1、サーバ2で実行します。

```
# systemctl reboot
```


3.2.2 ファイアウォールの設定確認

CLUSTERPRO X 5.3 のインストール&設定ガイドを確認し、必要なポート番号が使用できるようファイアウォールを設定してください。

MySQL は下記起動オプションで指定するポートを使用しますので、ポート番号が使用できるようにファイアウォールを設定してください。

本手順はサーバ1、サーバ2で実行します。

パラメータ名	デフォルト値
port	3306
mysqlx_port	33060
admin_port	33062

3.3 環境構築の流れ

環境構築は以下の手順で行います。

- 【 1 】. CLUSTERPRO X ソフトウェアのインストール
※サーバ 1、サーバ 2 で実行
- 【 2 】. クラスタの作成
- 【 3 】. サーバリソースの作成
- 【 4 】. MySQL ソフトウェアのインストール
※サーバ 1、サーバ 2 で実行
- 【 5 】. データベースの作成
※サーバ 1 で実行
- 【 6 】. MySQL 用のサービスリソースの作成

3.4 CLUSTERPRO X ソフトウェアのインストール

本章では、CLUSTERPRO X ソフトウェアのインストール、ライセンス登録を行います。本手順はサーバ1、サーバ2で実行します。

- ① インストールCD-ROMをmountします。
- ② パッケージファイルが格納されたフォルダに移動し、rpmコマンドを実行してパッケージファイルをインストールします。

```
# rpm -ivh clusterpro-5.3.0-1.x86_64.rpm
```

- ③ SELinux用設定を行うため、clpselctrl.sh スクリプトを実行します。

```
# clpselctrl.sh --add
```

- ④ ライセンスを登録します。
本書で構成する環境では下記ライセンスを登録します。

ライセンス製品名	登録サーバ
CLUSTERPRO X 5.3 for Linux	サーバ1
CLUSTERPRO X Database Agent 5.3 for Linux	サーバ1、サーバ2

ライセンスファイルが格納されたフォルダに移動し、clplcncscコマンドを実行してライセンスを登録します。

【サーバ1で実行】

```
# clplcncsc -i CLPX5.x-Base-lin.key
# clplcncsc -i CLPX5.x-DatabaseAgent-lin.key
```

【サーバ2で実行】

```
# clplcncsc -i CLPX5.x-DatabaseAgent-lin.key
```

- ⑤ サーバを再起動します。

```
# systemctl reboot
```

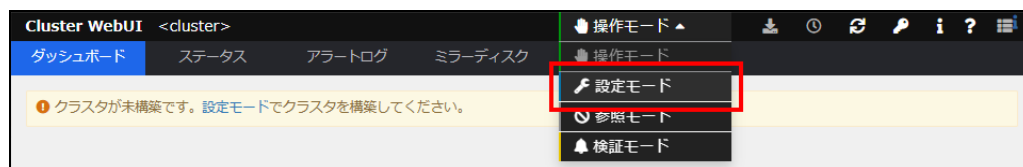
3.5 クラスタの作成

本章では、Cluster WebUI からクラスタとフェイルオーバーグループを作成します。

- ① ブラウザに下記アドレスを入力し、Cluster WebUIを起動します。

`http://173.8.100.3:29003/`

下記画面が表示されますので、[設定モード]に切り替えます。



- ② [クラスタ生成ウィザード]をクリックします。



- ③ クラスタ名を設定し、[次へ]をクリックします。

クラスタ生成ウィザード

サーバ

サーバ

サーバ

クラスタ

→

基本設定

→

インタコネクト

→

フェンシング

→

グループ

→

モニタ

クラスタ名*

cluster

コメント

言語*

日本語 ▼

管理IPアドレス

① 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

次へ ▶

キャンセル

- ④ [追加]をクリックし、サーバ2を追加します。[次へ]をクリックします。

クラスタ生成ウィザード

クラスタ ☒ → **サーバ** → インタコネクト → フェンシング → グループ → モニタ

サーバの定義一覧

順位	名前
マスタサーバ	server1

サーバグループの設定

① 「追加」ボタンを押して、クラスタを構成するサーバを追加します。
 サーバの優先順位は「↑」、「↓」ボタンで変更します。
 サーバグループを使用する場合は「設定」ボタンでサーバグループを設定します。

サーバ追加

サーバ名またはIPアドレス*

server2

① サーバ名またはIPアドレスを入力します。
 サーバ名を入力する場合、サーバ名の名前解決ができる必要があります。
 IPアドレスはIPv4とIPv6が使用できます。
 IPアドレスを入力する場合、該当するサーバのサーバ名を自動取得します。

クラスタ生成ウィザード

クラスタ ☒ → **サーバ** → インタコネクト → フェンシング → グループ → モニタ

サーバの定義一覧

順位	名前
マスタサーバ	server1
1	server2

サーバグループの設定

① 「追加」ボタンを押して、クラスタを構成するサーバを追加します。
 サーバの優先順位は「↑」、「↓」ボタンで変更します。
 サーバグループを使用する場合は「設定」ボタンでサーバグループを設定します。

- ⑤ 使用するインタコネクとディスクハートビート用デバイスを設定し、[次へ]をクリックします。
本書ではパブリック用のIPアドレスを優先度2に登録し、インタコネクを2つにしています。

クラスタ生成ウィザード

クラスタ ☒ → 基本設定 ☒ → インタコネク → フェンシング → グループ → モニタ

プロパティ 追加 削除

インタコネク一覧

優先度	種別	MDC	server1	server2
1	カーネルモード	使用しない	192.168.100.3	192.168.100.4
2	カーネルモード	使用しない	173.8.100.3	173.8.100.4
3	DISK	使用しない	/dev/sdb1	/dev/sdb1

↑ ↓

① 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」、「DISK」、「Witness」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を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」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。
「Witness」の場合は各サーバ列のセルをクリックして「使用する」、「使用しない」を設定し、「プロパティ」ボタンで詳細を設定します。
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に、「↑」、「↓」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 次へ キャンセル

- ⑥ 本書ではネットワークパーティション解決は設定せず [次へ]をクリックします。

クラスタ生成ウィザード

クラスタ ☒ → 基本設定 ☒ → インタコネク ☒ → フェンシング → グループ → モニタ

プロパティ 追加 削除

NP解決一覧

タイプ	ターゲット	server1	server2
NP解決はありません			

調整

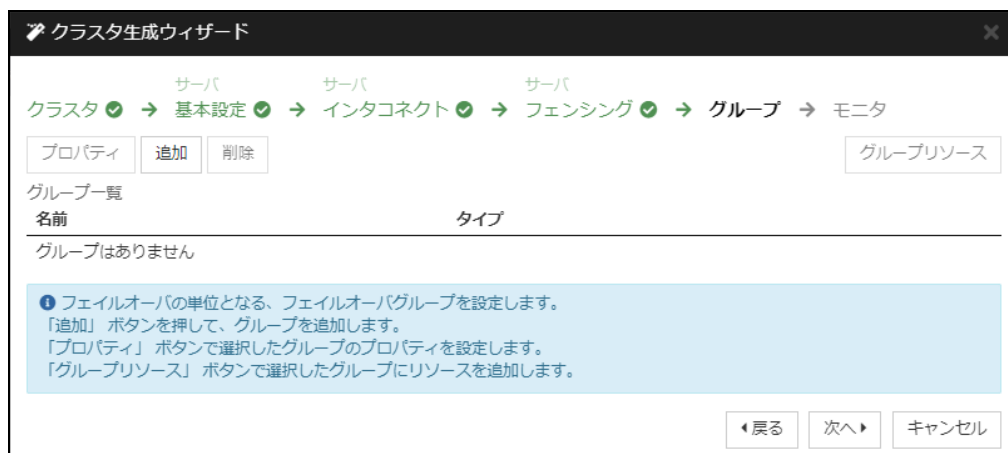
強制停止

タイプ* 使用しない 属性

① フェンシング機能を設定します。
「追加」ボタンでネットワークパーティション(NP)解決リソースを追加し、タイプを選択します。
「Ping」はターゲット列のセルをクリックしてPing送信先のIPアドレスを設定し、各サーバ列のセルをクリックして「使用する」「使用しない」を設定します。
「HTTP」はターゲット列のセルをクリックしてHTTP/パケット送信先を設定し、各サーバ列のセルをクリックして「使用する」「使用しない」を設定します。
「プロパティ」ボタンで詳細設定を確認・変更することができます。
「調整」ボタンでNP発生時の動作を設定します。
強制停止を使用する場合、使用する「タイプ」を選択し、「属性」ボタンで動作を設定します。

戻る 次へ キャンセル

- ⑦ フェイルオーバーグループは後で作成するため、[次へ]をクリックします。



クラスタ生成ウィザード

クラスタ → 基本設定 → インタコネクト → フェンシング → **グループ** → モニタ

グループリソース

グループ一覧

名前	タイプ
グループはありません	

1 フェイルオーバーの単位となる、フェイルオーバーグループを設定します。
「追加」 ボタンを押して、グループを追加します。
「プロパティ」 ボタンで選択したグループのプロパティを設定します。
「グループリソース」 ボタンで選択したグループにリソースを追加します。

戻る 次へ キャンセル

- ⑧ モニタリソースは後で作成するため、[完了]をクリックします。



クラスタ生成ウィザード

クラスタ → 基本設定 → インタコネクト → フェンシング → **グループ** → **モニタ**

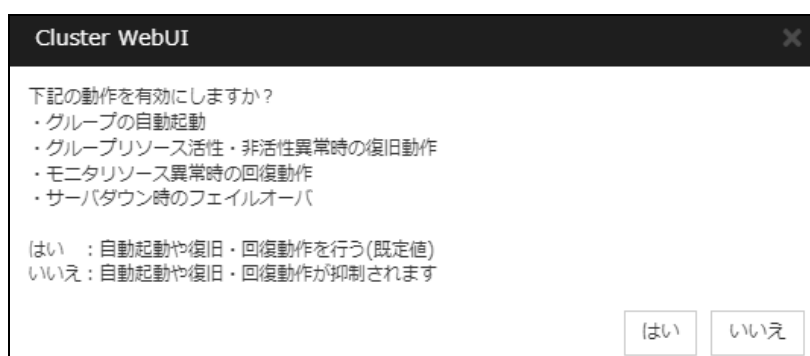
モニタリソース一覧

名前	タイプ
userw	ユーザ空間モニタ

1 「追加」 ボタンを押して、モニタリソースを追加します。
「プロパティ」 ボタンで選択したモニタリソースのプロパティを設定します。
「完了」 ボタンを押すと、クラスタの生成が完了します。

戻る 完了 キャンセル

- ⑨ [はい]をクリックします。



Cluster WebUI

下記の動作を有効にしますか?

- ・グループの自動起動
- ・グループリソース活性・非活性異常時の復旧動作
- ・モニタリソース異常時の回復動作
- ・サーバダウン時のフェイルオーバー

(はい) : 自動起動や復旧・回復動作を行う(既定値)
(いいえ) : 自動起動や復旧・回復動作が抑制されます

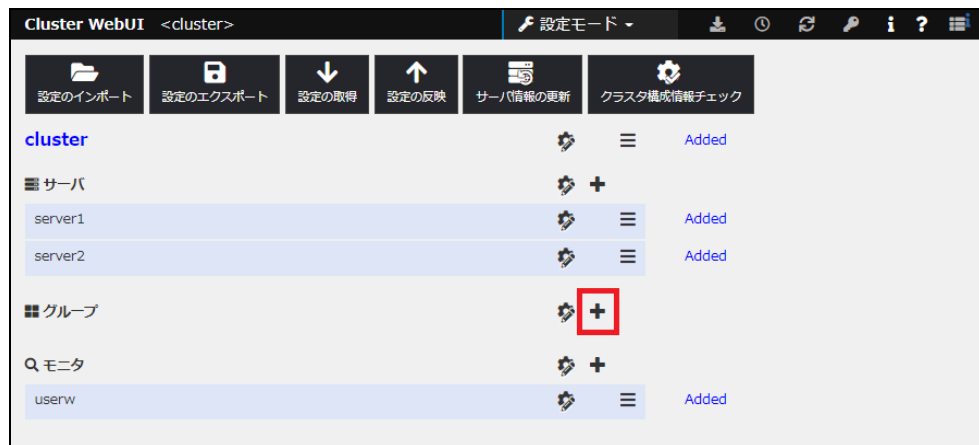
はい いいえ

3.6 サーバリソース作成

本章では、サーバリソースを作成します。

3.6.1 フェイルオーバーグループ

- ① グループから[グループの追加]をクリックします。



- ② フェイルオーバーグループ名を指定し、[次へ]をクリックします。

The screenshot shows the 'グループの定義' (Group Definition) form. The breadcrumb navigation is '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'. The form has the following fields:

- タイプ***: A dropdown menu set to 'フェイルオーバー' (Failover).
- サーバグループ設定を使用する**: A checkbox that is currently unchecked.
- 名前***: A text input field containing 'failover'.
- コメント**: A text input field that is empty.

At the bottom, there is a blue information box with the following text:

① グループのタイプを選択します。
サーバグループを使用する場合、「サーバグループ設定を使用する」チェックボックスをオンにします。

At the bottom right, there are three buttons: '戻る' (Back), '次へ' (Next), and 'キャンセル' (Cancel).

- ③ [全てのサーバでフェイルオーバー可能]にチェックがついていることを確認し、[次へ]をクリックします。

グループの定義 failover ✕

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

全てのサーバでフェイルオーバー可能 ☒

サーバ

server1
server2

① グループが起動可能なサーバを選択し、サーバの優先順位を設定します。

クラスタに登録されている全てのサーバで起動可能とする場合は、「全てのサーバでフェイルオーバー可能」チェックボックスをオンにします。優先順位はクラスタへのサーバ追加時に設定した優先順位となります。

起動するサーバを個別に設定する場合は、「全てのサーバでフェイルオーバー可能」チェックボックスをオフにします。右側の「利用可能なサーバ」リストから起動可能なサーバを選択して「追加」ボタンで「起動可能サーバ」リストに追加します。「↑」「↓」ボタンで優先順位を変更します。

◀ 戻る 次へ ▶ キャンセル

- ④ 要件に合わせて設定し、[次へ]をクリックします。本書ではデフォルトのままにします。

グループの定義 failover ✕

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

グループ起動属性 ☒ 自動起動 ☐ 手動起動

両系活性チェックを行う ☐

タイムアウト 秒

フェイルオーバー属性

☒ 自動フェイルオーバー

- ☒ 起動可能なサーバ設定に従う
- ☐ ダイナミックフェイルオーバーを行う
 - ☐ サーバグループ内のフェイルオーバーポリシーを優先する
 - ☐ スマートフェイルオーバーを行う
- ☐ サーバグループ内のフェイルオーバーポリシーを優先する
 - ☐ サーバグループ間では手動フェイルオーバーのみを有効とする
- ☐ 手動フェイルオーバー

フェイルオーバー属性(拡張)

☐ 指定したモニタリソースで異常を検出しているサーバをフェイルオーバー先から除外する

☐ 全てのサーバで異常を検出している場合、異常を無視してフェイルオーバーを行う

フェイルバック属性

☐ 自動フェイルバック ☒ 手動フェイルバック

① フェイルオーバーグループの起動やフェイルオーバーの動作を設定します。

クラスタ起動時にグループを自動起動しない場合は「グループ起動属性」を「手動起動」にしてください。

障害発生時に各サーバのモニタリソースの状態を考慮してフェイルオーバー先を選択する場合は「自動フェイルオーバー」の「ダイナミックフェイルオーバーを行う」を選択してください。サーバグループ設定を使用して、同一サーバグループ内のサーバを優先してフェイルオーバーする場合は、「サーバグループ内のフェイルオーバーポリシーを優先する」を選択してください。

◀ 戻る 次へ ▶ キャンセル

- ⑤ リソースは後で設定するため、[完了]をクリックします。

3.6.2 ディスクリソース

本章では共有ディスク構成のディスクリソースを作成します。

- ① 作成したフェイルオーバーグループの[リソースの追加]をクリックします。

- ② タイプから[ディスクリソース]を選択し、名前を設定して[次へ]をクリックします。

- ③ デフォルトのまま[次へ]をクリックします。

グループのリソース定義 | failover disk ✕

情報 → 依存関係 → 復旧動作 → 詳細

既定の依存関係に従う ☒

依存するリソース

AWS Elastic IPリソース
AWSセカンダリIPリソース
AWS仮想IPリソース
フローティングIPリソース
ボリュームマネージャリソース
仮想IPリソース

◀ 戻る
次へ ▶
キャンセル

- ④ ディスクリソースの復旧動作を選択し、[次へ]をクリックします。本書ではデフォルトのままにします。

グループのリソース定義 | failover disk ✕

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* 回

フェイルオーバーしきい値* 回

最終動作* 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 設定

非活性異常検出時の復旧動作

非活性リトライしきい値* 回

最終動作* 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 設定

◀ 戻る
次へ ▶
キャンセル

- ⑤ ファイルシステム、デバイス名、マウントポイントを入力し、「完了」をクリックします。

グループのリソース定義 | failover

disk

情報 → 依存関係 → 復旧動作 → 詳細

共通 server1 server2

ディスクタイプ*

ext4

ファイルシステム*

/dev/sdb2

デバイス名*

RAWデバイス名

マウントポイント*

/var/lib/mysql

調整

戻る

完了

キャンセル

3.6.3 フローティングIPリソース

本章ではフローティングIP(fip)リソースを作成します。クライアントはfipリソースを使用してクラスタサーバに接続することができます。また、fipリソースを使用することで、CLUSTERPROがフェイルオーバーまたはフェイルオーバーグループの移動を行っても、クライアントは接続先サーバの切り替えを意識する必要がありません。

- ① 作成したフェイルオーバーグループの[リソースの追加]をクリックします。



- ② タイプから[フローティングIPリソース]を選択し、fipリソース名を設定します。[次へ]をクリックします。



- ③ デフォルトのまま[次へ]をクリックします。



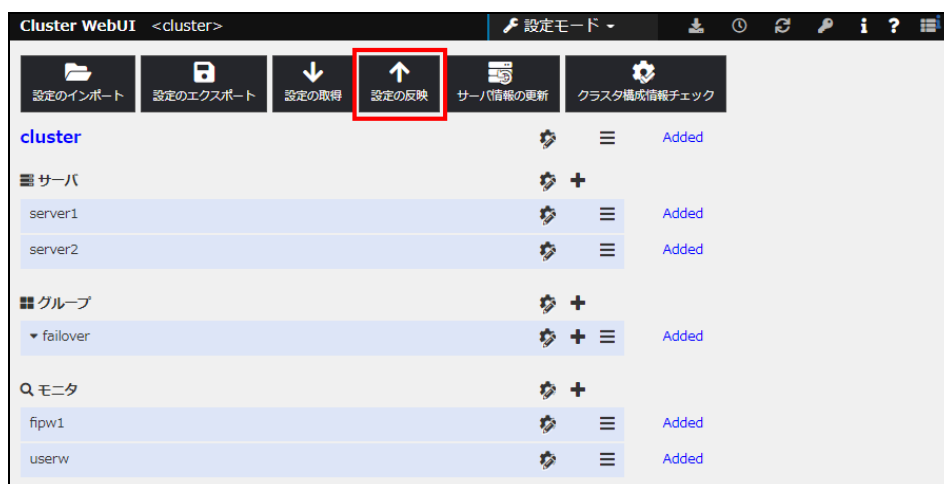
- ④ fipリソースの復旧動作を設定し、[次へ]をクリックします。本書ではデフォルトのままにします。

- ⑤ 使用するfipのアドレスを入力し、[完了]をクリックします。

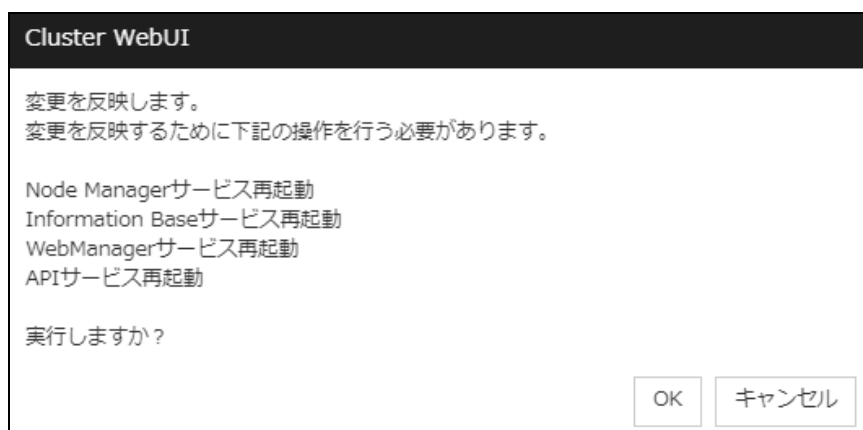
3.6.4 設定項目の反映

ディスクリソース、フローティングIPリソースの設定が完了したら、設定項目を反映します。

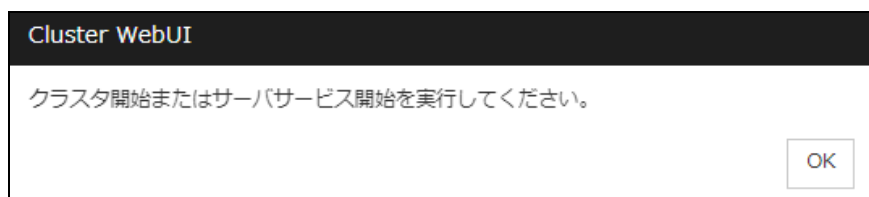
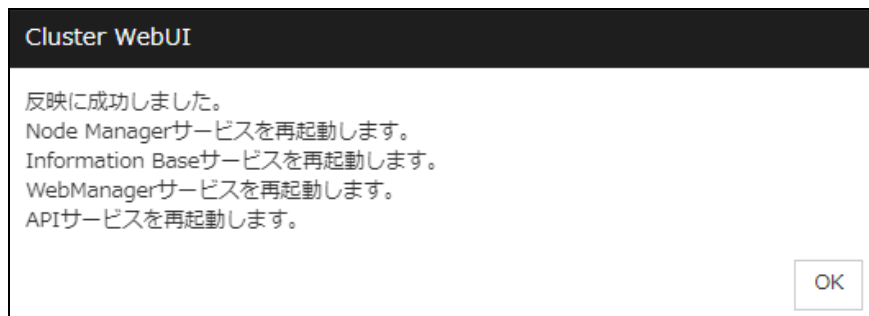
- ① [設定の反映]をクリックします。



- ② [OK]をクリックします。



- ③ ダイアログが表示されるので[OK]をクリックします。



- ④ [操作モード]に切り替えます。



- ⑤ [クラスタ開始]をクリックします。



- ⑥ [開始]をクリックします。クラスタが正常に動作することを確認してください。

クラスタ開始 | cluster

クラスタを開始しますか?

▶ 開始

キャンセル

3.7 MySQL ソフトウェアのインストール

本章では、MySQL ソフトウェアのインストールを行います。本手順はサーバ 1、サーバ 2 で実行します。

- ① ダウンロードしたインストール媒体を解凍します。

```
# unzip p37827121_840_Linux-x86-64.zip
```

- ② rpmパッケージをインストールします。

```
# rpm -Uvh mysql-commercial-server-8.4.5-1.1.el8.x86_64.rpm ¥  
mysql-commercial-client-8.4.5-1.1.el8.x86_64.rpm ¥  
mysql-commercial-client-plugins-8.4.5-1.1.el8.x86_64.rpm ¥  
mysql-commercial-common-8.4.5-1.1.el8.x86_64.rpm ¥  
mysql-commercial-libs-8.4.5-1.1.el8.x86_64.rpm ¥  
mysql-commercial-icu-data-files-8.4.5-1.1.el8.x86_64.rpm
```

- ③ mysqld.serviceの自動起動を無効にします。

```
# systemctl disable mysqld.service
```

- ④ mysqld.serviceのユニットファイルを編集し、環境変数 MYSQLD_OPTS にmy.cnfのパスの設定、自動再起動設定(Restart)の無効化をします。

```
# vi /usr/lib/systemd/system/mysqld.service  
  
(変更内容)  
-----  
# Start main service  
Environment=MYSQLD_OPTS="--defaults-file=/var/lib/mysql/my.cnf"  
ExecStart=/usr/sbin/mysqld $MYSQLD_OPTS  
  
# Use this to switch malloc implementation  
EnvironmentFile=-/etc/sysconfig/mysql  
  
# Sets open_files_limit  
LimitNOFILE = 10000  
Restart=no
```

- ⑤ /var/lib/mysqlディレクトリの権限を設定します。

```
# chown mysql:mysql /var/lib/mysql  
# chmod 750 /var/lib/mysql
```

3.8 データベースの作成

本章では、データベースを作成します。手順実施前に 3.6 章で作成したフェイルオーバーグループがサーバ 1 で起動していることを確認してください。

3.8.1 MySQLサーバの初期化

本手順はサーバ 1 で実行します。

- ① 構成ファイル(/var/lib/mysql/my.cnf)を作成します。本書では下記内容で作成します。

```
# vi /var/lib/mysql/my.cnf

(設定例)
-----
[mysqld]
port=3306
datadir=/var/lib/mysql/data
socket=/var/lib/mysql/data/mysql.sock

log-error=/var/lib/mysql/mysql.log
log_timestamps=SYSTEM
pid-file=/var/run/mysqld/mysqld.pid

[client]
socket=/var/lib/mysql/data/mysql.sock
```

- ② MySQLサーバを初期化します。

```
# /usr/sbin/mysqld --defaults-file=/var/lib/mysql/my.cnf ¥
--initialize --user=mysql
```

3.8.2 rootアカウントのパスワード設定

本手順はサーバ1で実行します。

- ① mysqld.service を起動します。

```
# systemctl start mysqld.service
```

- ② エラーログ(/var/lib/mysql/mysqld.log)からrootユーザの初期パスワードを確認します。下記例では「I00BzS5d-d0M」が初期パスワードです。

```
2025-06-11T16:29:12.044468+09:00 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: I00BzS5d-d0M
```

- ③ 確認した初期パスワードでMySQLにログインします。

```
# mysql --defaults-file=/var/lib/mysql/my.cnf --user=root -p
```

- ④ rootユーザのパスワードを変更します。

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'password';
```

- ⑤ 接続を終了します。

```
mysql> exit
```

3.8.3 監視用のアカウント、データベースを作成する

本手順はサーバ1で実行します。

- ① MySQLにログインします。

```
# mysql --defaults-file=/var/lib/mysql/my.cnf --user=root -p
```

- ② 監視用アカウントを作成します。

```
mysql> CREATE USER 'mysql'@'server1' IDENTIFIED BY '<password>';  
mysql> CREATE USER 'mysql'@'server2' IDENTIFIED BY '<password>';  
mysql> GRANT ALL PRIVILEGES ON watch.* TO 'mysql'@'server1';  
mysql> GRANT ALL PRIVILEGES ON watch.* TO 'mysql'@'server2';
```

- ③ 監視用データベースを作成します。

```
mysql> CREATE DATABASE watch;
```

- ④ モニタリソースを「監視レベル1」で設定する場合、監視用テーブルを作成します。「監視レベル2」以上を指定する場合は作成不要です。

```
mysql> CREATE TABLE watch.mysqlwatch (num INT NOT NULL PRIMARY  
KEY) ENGINE=INNODB;  
mysql> INSERT INTO watch.mysqlwatch VALUES(0);  
mysql> COMMIT;  
mysql> exit;
```

- ⑤ mysqld.service を停止します。

```
# systemctl stop mysqld.service
```

3.9 MySQL 用のサービスリソース作成

本章では、MySQL 用のリソースを登録します。

3.9.1 execリソースの作成

- ① ブラウザに下記アドレスを入力し、Cluster WebUIを起動します。

<http://173.8.100.3:29003/>

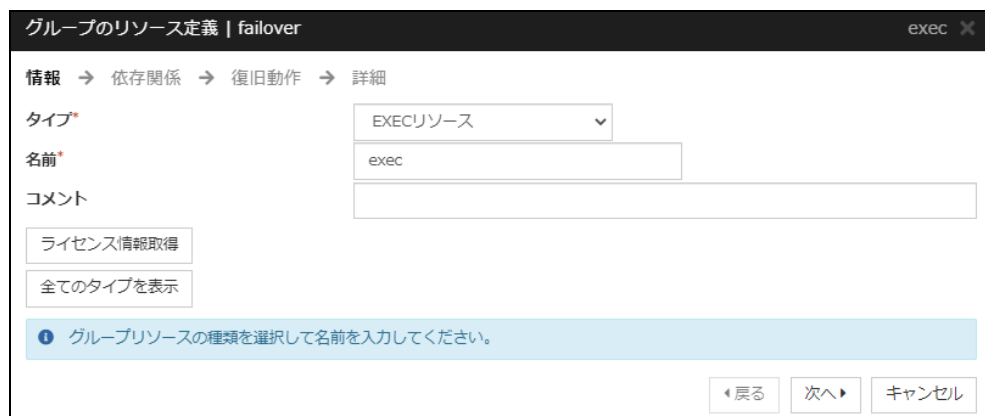
下記画面が表示されますので、[設定モード]を選択してください。



- ② [failover]グループから、[リソースの追加]をクリックします。



- ③ [タイプ]で[EXECリソース]を選択し、[次へ]をクリックします。



- ④ [既定の依存関係に従う]のチェックを外し、ディスクリソースとfipリソースに依存するように設定します。[次へ]をクリックします。

グループのリソース定義 | failover exec

情報 → 依存関係 → 復旧動作 → 詳細

既定の依存関係に従う ☐

依存するリソース

名前	リソースタイプ
disk	ディスクリソース
fip	フローティングIPリソース

← 追加

→ 削除

利用可能なリソース

名前
利用可能なリソースはありません

◀ 戻る 次へ ▶ キャンセル

- ⑤ 復旧動作を設定し、[次へ]をクリックします。本書ではデフォルトのままにします。

グループのリソース定義 | failover exec

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* 0 回

フェイルオーバーしきい値* 1 回

最終動作* 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 設定

非活性異常検出時の復旧動作

非活性リトライしきい値* 0 回

最終動作* 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 設定

◀ 戻る 次へ ▶ キャンセル

- ⑥ [Start Script]を選択して[編集]をクリックし、後述の設定例を参考に編集します。同様に[Stop Script]も編集し、[調整]をクリックします。



start.shの設定例

```
#!/bin/sh
#*****
#*                start.sh                *
#*****

#ulimit -s unlimited

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"

        # start MySQL
        systemctl start mysqld.service
        if [ $? -ne 0 ]
        then
            echo "mysql start error"
            exit 1
        fi

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
        exit 1
    fi
fi
```



```

elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"

        # start MySQL
        systemctl start mysqld.service
        if [ $? -ne 0 ]
        then
            echo "mysql start error"
            exit 1
        fi

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
        exit 1
    fi
else
    echo "NO_CLP"
    exit 1
fi
echo "EXIT"
exit 0

```

stop.shの設定例

```

#!/bin/sh
#####
#*                stop.sh                *
#####

#ulimit -s unlimited

if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"

        # shutdown MySQL
        systemctl stop mysqld.service
        if [ $? -ne 0 ]
        then
            echo "mysql shutdown error"
            exit 1
        fi

```

```
if [ "$CLP_SERVER" = "HOME" ]
then
    echo "NORMAL2"
else
    echo "ON_OTHER1"
fi
else
    echo "ERROR_DISK from START"
    exit 1
fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"

        # shutdown MySQL
        systemctl stop mysqld.service
        if [ $? -ne 0 ]
        then
            echo "mysql shutdown error"
            exit 1
        fi

        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
        exit 1
    fi
else
    echo "NO_CLP"
    exit 1
fi
echo "EXIT"
exit 0
```

- ⑦ [タイムアウト]を設定し、[OK]をクリックします。

EXECリソース調整プロパティ

パラメータ メンテナンス

開始スクリプト

☒ 同期 タイムアウト* 1800 秒

☐ 非同期

待機系サーバで実行する ☐

タイムアウト 10 秒

終了スクリプト

☒ 同期 タイムアウト* 1800 秒

☐ 非同期

待機系サーバで実行する ☐

タイムアウト 10 秒

OK キャンセル 適用

※デフォルトのタイムアウト値は1800秒(30分)のため、起動/停止で障害発生した場合フェイルオーバーまで30分かかる可能性があります。

- ⑧ [完了]をクリックします。

グループのリソース定義 | failover exec ✕

情報 ✓ → 依存関係 ✓ → 復旧動作 ✓ → 詳細

☐ ユーザアプリケーション

☒ この製品で作成したスクリプト

編集 表示 置換

スクリプト一覧

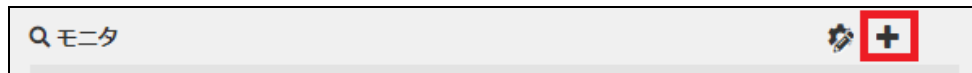
種類	名前
Start Script	start.sh
Stop Script	stop.sh

調整

◀ 戻る 完了 キャンセル

3.9.2 モニタリソースの作成

- ① [モニタ]から[モニタリソースの追加]をクリックします。



- ② [タイプ]から[MySQLモニタ]を選択し、[次へ]をクリックします。

モニタリソースの定義 mysqlw

情報 → 監視(共通) → 監視(固有) → 回復動作

タイプ* MySQLモニタ

名前* mysqlw

コメント

ライセンス情報取得

全てのタイプを表示

① モニタソースの種類を選択して名前を入力してください。

戻る 次へ キャンセル

- ③ [対象リソース]の[参照]をクリックします。

モニタリソースの定義 mysqlw

情報 ☒ → 監視(共通) → 監視(固有) → 回復動作

インターバル* 60 秒

タイムアウト* 120 秒

タイムアウト発生時に監視プロセスのダンプを採取する ☐

タイムアウト発生時にリトライしない ☐

タイムアウト発生時動作 回復動作を実行する

リトライ回数* 2 回

監視開始待ち時間* 0 秒

監視タイミング

☐ 常時

☒ 活性時

対象リソース* 参照

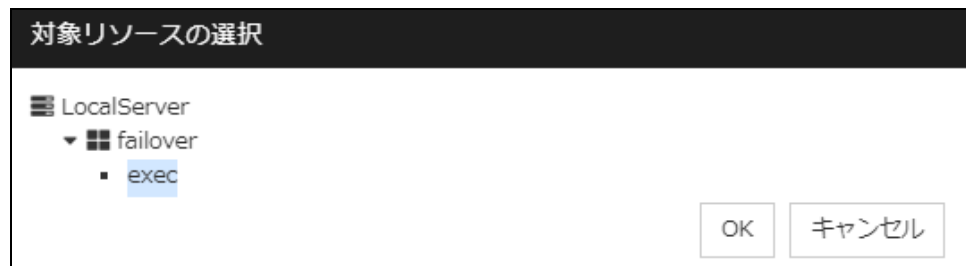
nice値 0

監視を行うサーバを選択する サーバ

監視処理時間メトリクスを送信する ☐

戻る 次へ キャンセル

- ④ [exec]リソースを選択し、[OK]をクリックします。



- ⑤ [次へ]をクリックします。本書では[対象リソース]以外の項目はデフォルトのままにします。



- ⑥ 各項目を設定します。[次へ]をクリックします。

モニタリソースの定義 mysqlw ✕

情報 ✓ → 監視(共通) ✓ → 監視(固有) → 回復動作

監視レベル* レベル2(update/selectでの監視) ▼

データベース名* watch

IPアドレス* 173.8.100.100

ポート番号* 3306

ユーザ名* mysql

パスワード* 設定

監視テーブル名* mysqlwatch

ストレージエンジン* InnoDB ▼

ライブラリパス* /usr/lib64/mysql/libmysqlclient.so.24 ▼

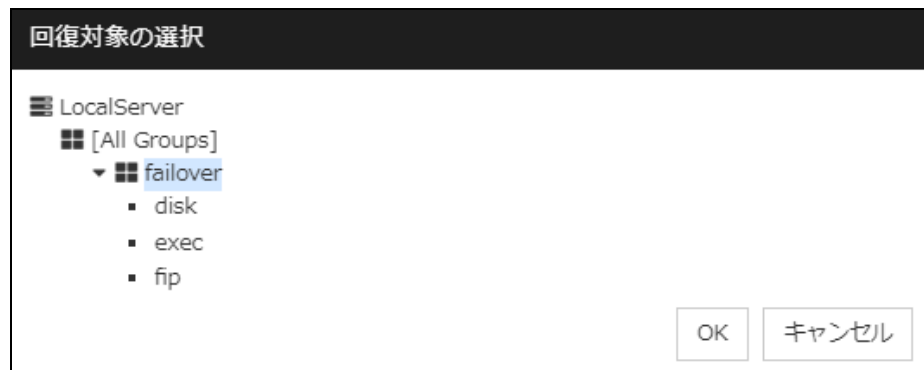
◀ 戻る 次へ ▶ キャンセル

監視レベル	任意
データベース名	3.8.3で作成したデータベース
IPアドレス	フローティングIPアドレス
ポート番号	3.8.1で指定した起動オプションportの番号
ユーザ名	3.8.3で作成したアカウントのユーザ名
パスワード	3.8.3で作成したアカウントのパスワード
監視テーブル名	3.8.3で作成したテーブル名。作成していない場合は任意
ストレージエンジン	InnoDB
ライブラリパス	3.7でインストールしたライブラリのパス。プルダウンにない場合は手動入力

- ⑦ [参照]をクリックします。



- ⑧ [failover]を選択し、[OK]をクリックします。



- ⑨ [完了]をクリックします。本書では[回復対象]以外の項目はデフォルトのままにします。

モニタリソースの定義 mysqlw

情報 ☒ → 監視(共通) ☒ → 監視(固有) ☒ → 回復動作

回復動作 カスタム設定

回復対象 failover 参照

回復スクリプト実行回数* 0 回

再活性前にスクリプトを実行する ☐

最大再活性回数* 0 回

フェイルオーバー実行前にスクリプトを実行する ☐

最大フェイルオーバー回数* 1 回

最終動作前にスクリプトを実行する ☐

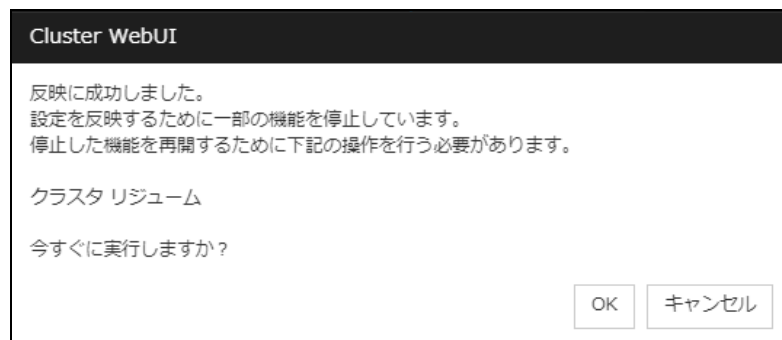
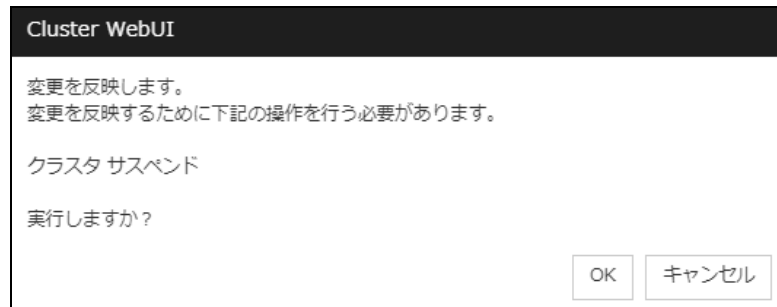
最終動作 クラスタサービス停止とOSシャットダウン

スクリプト設定

戻る 完了 キャンセル

3.9.3 設定項目の反映

- ① [設定の反映]をクリックします。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 [操作モード]に切り替えます。



- ③ グループ[failover]を停止します。確認ダイアログが表示されるので[停止]をクリックします。



- ④ グループ[failover]を起動します。確認ダイアログが表示されるので[起動]をクリックします。



- ⑤ グループ、モニタが正常に起動することを確認します。



以上で構築は完了です。