

CLUSTERPROによる MariaDBのHAクラスタ動作検証

2015.08.10

第1版

日本電気株式会社
クラウドプラットフォーム事業部
CLUSTERPROグループ

改版履歴

版数	改版日付	内容
1	2015/08/10	新規作成

© Copyright NEC Corporation 2015. All rights reserved.

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ぬ。

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。

本書記載以外のMariaDB、CLUSTERPRO、OSのバージョンの組み合わせでは動作確認を行っておりませんので、お客様側にて十分に動作確認を行ってください。

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。

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。

目次

1. はじめに.....	5
1.1. 目的	5
1.2. 構成概要	5
2. 検証構成.....	6
2.1. 構成図	6
2.2. 動作環境	7
3. 検証内容.....	7
4. 検証結果.....	8
5. まとめ.....	8
補足:構築手順.....	9
1.1. システム構成の設定.....	13
1.2. CLUSTERPROの設定	13
1.2.1. クラスタの設定.....	13
1.2.2. MariaDB のインストール	14
1.2.3. マルチインスタンスの構築(サーバ1)	15
1.2.4. マルチインスタンスの構築(サーバ2)	21
1.2.5. グループリソースの設定.....	21
1.2.6. モニタリソースの設定	21
付録 サンプルスクリプト.....	23

1.はじめに

1.1. 目的

MariaDBとCLUSTERPRO Xの連携により、MariaDBの障害を検知できること、また、待機サーバに自動的に切り替えること(フェイルオーバー)により、業務の継続ができることを確認する。

1.2. 構成概要

双方向スタンバイのミラーディスク型クラスタを構築します。

構成概要は以下の通りです。

サーバ1ではデータベース1(インスタンス1)、サーバ2ではデータベース2(インスタンス2)を運用します。クライアントはCLUSTERPROで設定するフローティングIPアドレスを使用してサーバ1、サーバ2に接続します。

サーバ1で障害が発生すると、CLUSTERPROで障害を検知し、サーバ1でフェイルオーバーが実行されます。フェイルオーバーが完了するとサーバ2で、データベース1(インスタンス1)が起動します。フェイルオーバーによってフローティングIPアドレスがサーバ2へ移行するため、クライアントはサーバが切り替わったことを意識せずに、同一のIPアドレスに再接続することで運用が可能です。

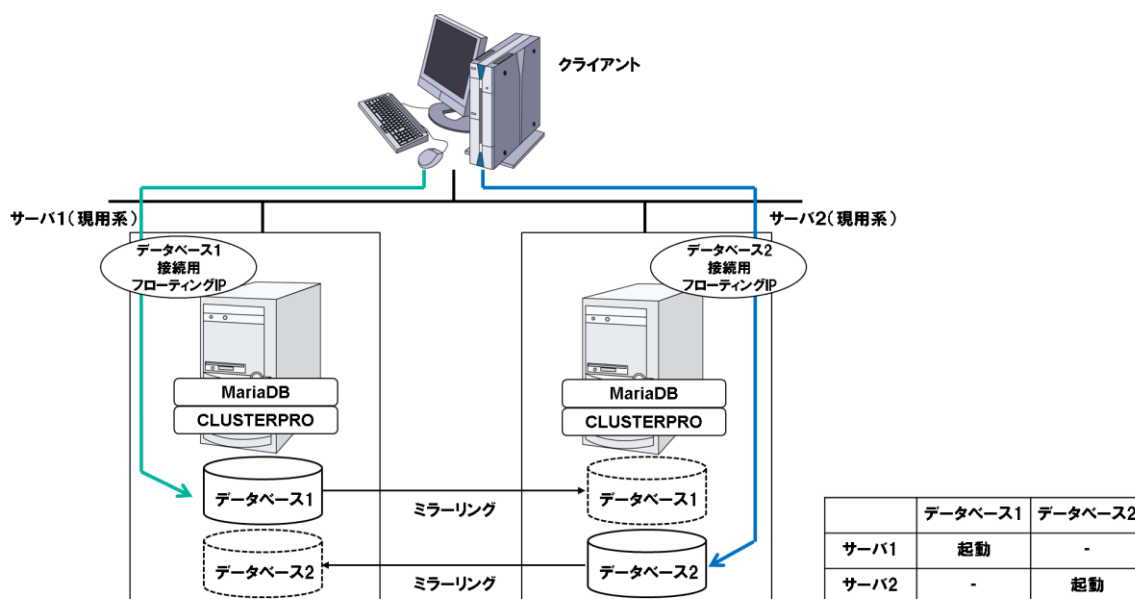


図1-1. クラスタ 概要図

1. はじめに

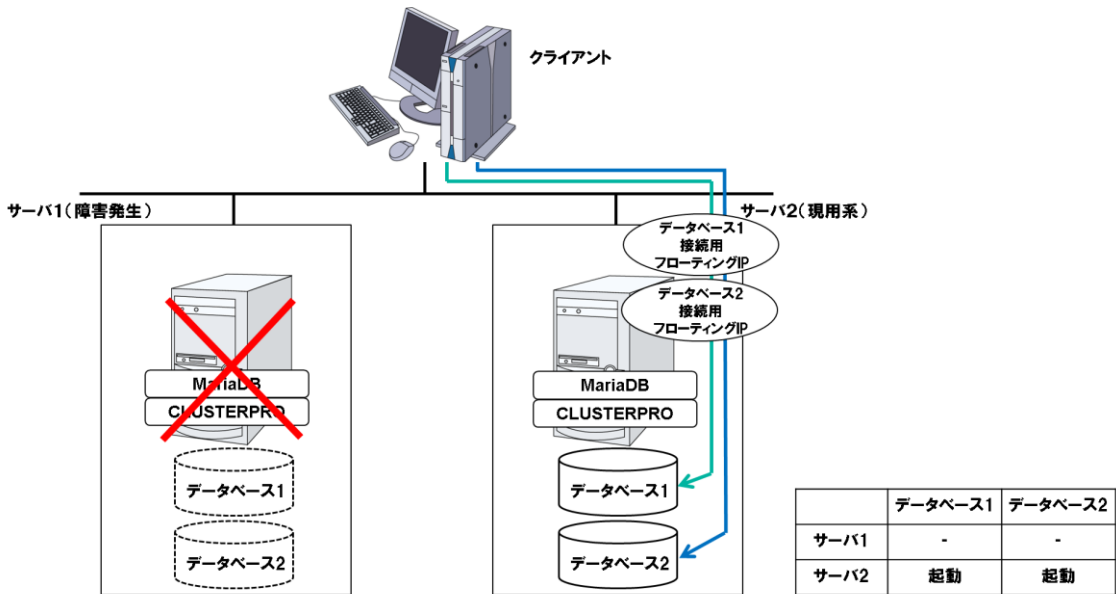


図1-2. サーバ1 障害発生時

2. 検証構成

2.1. 構成図

本検証における構成図は以下の通りです。

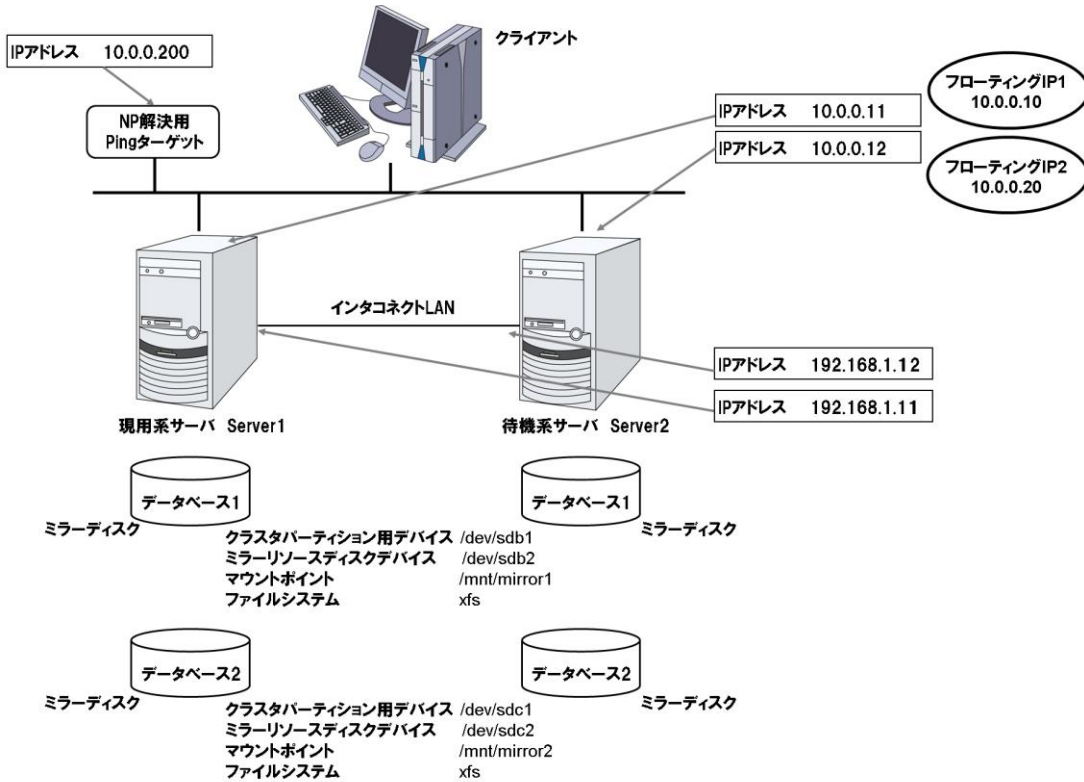


図2-1 ミラーディスク型クラスタの設定構成

2.2. 動作環境

本検証における動作環境は以下の通りです。

OS	CentOS 7.0 (x86_64)
CLUSTERPRO	CLUSTERPRO X 3.3 for Linux
MariaDB	MariaDB 10.0.17

3. 検証内容

以下の障害を契機とした可用性について検証を行います。

分類	サーバ	部位	障害
HW障害	サーバ1	サーバ	サーバダウン
		ネットワーク	ネットワークダウン (パブリックLAN抜線)
	サーバ2	サーバ	サーバダウン
		ネットワーク	ネットワークダウン (パブリックLAN抜線)
SW障害	サーバ1	OS	OS ダウン (halt コマンド実行)
		MariaDB	プロセスダウン
	サーバ2	OS	OS ダウン (halt コマンド実行)
		MariaDB	プロセスダウン

各障害について下記の観点で動作を確認します。

- CLUSTERPRO が障害を検出し MariaDB のフェイルオーバを実行すること
- クライアントからフローティングIP経由でMariaDBに接続が可能で、業務が継続可能であること。
(データの参照/更新が可能)

4. 検証結果

各障害時の期待動作と結果は以下の通りです。

分類	サーバ	部位	障害	期待動作	結果
HW障害	サーバ1	サーバ	サーバダウン	フェイルオーバー(サーバ2)	○
		ネットワーク	ネットワークダウン (パブリックLAN抜線)	フェイルオーバー(サーバ2)	○
	サーバ2	サーバ	サーバダウン	フェイルオーバー(サーバ1)	○
		ネットワーク	ネットワークダウン (パブリックLAN抜線)	フェイルオーバー(サーバ1)	○
SW障害	サーバ1	OS	OS ダウン (halt コマンド実行)	フェイルオーバー(サーバ2)	○
		MariaDB	プロセスダウン	フェイルオーバー(サーバ2)	○
	サーバ2	OS	OS ダウン (halt コマンド実行)	フェイルオーバー(サーバ1)	○
		MariaDB	プロセスダウン	フェイルオーバー(サーバ1)	○

各障害発生時にCLUSTERPROによる障害自動検知、自動フェイルオーバーが実行され、MariaDBへの接続が可能となることが確認できました。

5. まとめ

本検証において、MariaDBとCLUSTERPROの連携により、業務停止時間を短縮し、高可用性および運用効率化の実現性が確認できました。

なお、本検証では、双方向スタンバイ構成におけるMariaDBのHAクラスタの動作検証を行っておりますが、片方向スタンバイ構成における動作も可能です。

補足:構築手順

本書のCLUSTERPROの設定例を以下に記載します。
本手順は、MariaDB をクラスタ化するための注意点や設定例を参考情報として示すものであり、これらの動作保証を行うものではありません。

CLUSTERPROのインストールおよび設定に関してはマニュアルを参照ください。

クラスタの構成例は以下の通りです。

クラスタ構成例

設定対象	設定パラメータ	設定値
クラスタ構成	クラスタ名	Cluster
	サーバ数	2
	フェイルオーバーグループ数	2
	モニタリソース数	9
ハートビート リソース	カーネルモードLAN ハートビート数	2(*1)
	1 台目のサーバの情報 (マスタ サーバ)	サーバ名 パブリックの IP アドレス インタコネク트의 IP アドレス
2 台目のサーバの情報	サーバ名 パブリックの IP アドレス インタコネク트의 IP アドレス	server2 192.168.1.12 10.0.0.12
NP解決リソース	タイプ	Ping
	Ping ターゲット	10.0.0.200
	server1	使用する
	server2	使用する
1つ目のフェイルオーバー グループ	タイプ	フェイルオーバー
	グループ名	failover1
	起動サーバ	全てのサーバ
	グループ リソース数	3
1つ目のグループリ ソース	タイプ	フローティングIPリソース
	グループ リソース名	fip1
	IPアドレス	10.0.0.10

2つ目のグループリソース	タイプ グループ リソース名 マウントポイント ミラーディスクリソース クラスターパーティションドライブ名 ミラーディスクリソース データパーティションドライブ名 ファイルシステム	ミラーディスクリソース md1 /mnt/mirror1 /dev/sdb1 /dev/sdb2 xfs
3つ目のグループリソース	タイプ グループリソース名 スクリプト名	execリソース exec_MariaDB1 start.sh stop.sh
2つ目のフェイルオーバーグループ	タイプ グループ名 起動サーバ グループ リソース数	フェイルオーバー failover2 全てのサーバ 3
1つ目のグループリソース	タイプ グループ リソース名 IPアドレス	フローティングIPリソース fip2 10.0.0.20
2つ目のグループリソース	タイプ グループ リソース名 マウントポイント ミラーディスクリソース クラスターパーティションドライブ名 ミラーディスクリソース データパーティションドライブ名 ファイルシステム	ミラーディスクリソース md2 /mnt/mirror2 /dev/sdc1 /dev/sdc2 xfs
3つ目のグループリソース	タイプ グループリソース名 スクリプト	execリソース exec_MariaDB2 start.sh stop.sh
1つ目のモニタリソース 自動作成	タイプ モニタ リソース名	ユーザ空間監視 userw
2つ目のモニタリソース フローティングIPリソース(fip1)作成後自動作成	タイプ モニタリソース名 対象リソース 回復対象	フローティングIP監視 fipw1 fip1 fip1

3つ目のモニタリソース フローティングIPリソース(fip2)作成後自動作成	タイプ モニタリソース名 対象リソース 回復対象	フローティングIP監視 fipw2 fip2 fip2
4つ目のモニタリソース ミラーディスクリソース(md1)作成後自動作成	タイプ モニタリソース名 ミラーディスクリソース 回復対象 最終動作	ミラーコネクト監視 mdnw1 md1 md1 何もしない
5つ目のモニタリソース ミラーディスクリソース(md1)作成後自動作成	タイプ モニタリソース名 ミラーディスクリソース 回復対象 最終動作	ミラーディスク監視 mdw1 md1 md1 何もしない
6つ目のモニタリソース ミラーディスクリソース(md2)作成後自動作成	タイプ モニタリソース名 ミラーディスクリソース 回復対象 最終動作	ミラーコネクト監視 mdnw2 md2 md2 何もしない
7つ目のモニタリソース ミラーディスクリソース(md2)作成後自動作成	タイプ モニタリソース名 ミラーディスクリソース 回復対象 最終動作	ミラーディスク監視 mdw2 md2 md2 何もしない
8つ目のモニタリソース	タイプ モニタリソース名 インターバル タイムアウト リトライ回数 監視開始待ち時間 監視タイミング 回復動作 回復対象 最終動作	カスタム監視 genw_MariaDB1 60秒(*2) 120秒(*2) 1回(*2) 30秒(*2) 活性時 対象リソース:exec_MariaDB 回復対象に対してフェイルオーバー実行 failover1 何もしない

9つ目のモニタリソース	タイプ	カスタム監視
	モニタリソース名	genw_MariaDB2
	インターバル	60秒(*2)
	タイムアウト	120秒(*2)
	リトライ回数	1回(*2)
	監視開始待ち時間	30秒(*2)
	監視タイミグ	活性時 対象リソース:exec_MariaDB2
	回復動作	回復対象に対してフェイルオーバー 実行
	回復対象	failover2
	最終動作	何もしない

*1:パブリック用のIPアドレスをインタコネクト兼用で使用することで、ハートビートに使用しています

*2:本値は参考値です。システムの要件/特性に合わせて適切な値にチューニングしてください

MariaDB の構成例は以下の通りです。

以降は、下記設定での構築手順を記載します。

MariaDB のデータディレクトリは、データパーティション上に作成します。

MariaDB 構成例

インストールディレクトリ	/var/lib/mysql
インスタンス1の設定値	
port	3307
server_id	10
datadir	/mnt/mirror1/mysql1
socket	/mnt/mirror1/mysql1/mysql1.sock
pid-file	/mnt/mirror1/mysql1/labra.pid
tmpdir	/tmp/tmp1
インスタンス2の設定値	
port	3308
server_id	20
datadir	/mnt/mirror2/mysql2
socket	/mnt/mirror2/mysql2/mysql2.sock
pid-file	/mnt/mirror2/mysql2/labra.pid
tmpdir	/tmp/tmp2

本機能の設定は以下の手順で行います。

- 1-1. システム構成の設定
- 1-2. CLUSTERPROの設定

1.1. システム構成の設定

CLUSTERPRO のインストール前に以下の設定を実施します。
ハードウェア構成を決定し、ハードウェア、ネットワーク等の設定を行います。
システム構成の設定手順の詳細は『CLUSTERPRO X for Linux インストール&設定ガイド』の「第1章 システム構成の設定」を参照してください。

1.2. CLUSTERPROの設定

CLUSTERPRO をインストールして、ミラーディスク構成のクラスタを構築します。

以下の手順で CLUSTERPRO の設定を行います。

- 1-2-1.クラスタの設定
ミラーディスククラスタを構築します。
- 1-2-2.MariaDB のインストール
各サーバに MariaDB をインストールします。
- 1-2-3.マルチインスタンスの構築(サーバ1)
サーバ1上でMariaDB のインスタンスを作成します。
- 1-2-4.マルチインスタンスの構築(サーバ2)
サーバ2上でMariaDB のインスタンスを設定します。
- 1-2-5.グループリソースの設定
グループリソースにexecリソースを設定します。
- 1-2-6.モニタリソースの設定
モニタリソースにカスタム監視リソースを設定します。

1.2.1.クラスタの設定

CLUSTERPROのBuilder を使用して、MariaDB の運用に使用するフェイルオーバーグループを2つ作成します。各フェイルオーバーグループには以下のリソースが必要となります。

- ・ フローティングIPリソース
- ・ ミラーディスクリソース

上記のリソースを追加する手順については、『CLUSTERPRO X for Linux インストール&設定ガイド』の「クラスタ構成情報を作成する」を参照してください。
作成したフェイルオーバーグループをサーバ1で起動してください。

1.2.2.MariaDB のインストール

各サーバにMariaDBをインストールします。
ダウンロードサイトからMariaDBのRPMパッケージを取得します。

1. yumコマンドを使いMariaDBをインストールします。

```
# yum --disablerepo=* -y install MariaDB-*.rpm
```

2. インストール完了後、rpmコマンドを使い正しくインストールされていることを確認します。

```
# rpm -qa | grep -i mariadb
MariaDB-server-10.0.17-1.el7.centos.x86_64
MariaDB-devel-10.0.17-1.el7.centos.x86_64
MariaDB-client-10.0.17-1.el7.centos.x86_64
MariaDB-common-10.0.17-1.el7.centos.x86_64
MariaDB-shared-10.0.17-1.el7.centos.x86_64
MariaDB-compatible-10.0.17-1.el7.centos.x86_64
```

3. 構成パラメータを変更します。
構成パラメータファイル、「/etc/my.cnf」ファイルを利用します。
※MariaDBは/etc/my.cnf.d に構成パラメータファイルのテンプレートがあります。
利用形態にあわせ参考にしてください。

```
#[client-server]

#!includedir /etc/my.cnf.d

[client]
default-character-set = utf8
socket=/var/lib/mysql/mysql.sock

[mysqld]
character-set-server=utf8
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
log_bin=mysql-bin
innodb_flush_method=O_DIRECT
innodb_buffer_pool_size=128M
innodb_data_file_path=ibdata1:12M:autoextend
sync_binlog=1
innodb_flush_log_at_trx_commit=1

[mysql]
default-character-set = utf8
```

4. MariaDBを起動します。
systemctl start mysqlで起動します。

```
# systemctl start mysql
```

5. ユーザ設定を行います。
mysql_secure_installationを使い、不要なアカウントの削除とrootユーザのパスワード設定を行います。各ガイドメッセージにしたがって、適宜、変更してください。

```
# mysql_secure_installation
```

6. 変更したパスワードを利用して、MariaDBにログインできることを確認します。

```
# mysql -uroot -p
Enter password:

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.0.17-MariaDB MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

MariaDB [(none)]>
```

1.2.3. マルチインスタンスの構築(サーバ1)

サーバ1上で2つのインスタンスを作成します。

1. データベースを初期化します。
mysql_install_db を使用し、MariaDBのデータベースの初期化とシステムテーブルを作成します。
複数のインスタンスを立ち上げるために、データディレクトリは別々のパス(各ディスクリソースで設定したパス)を指定します。

```
# mysql_install_db --user=mysql --datadir=/mnt/mirror1/mysql1
# mysql_install_db --user=mysql --datadir=/mnt/mirror2/mysql2
```

2. 構成ファイルを変更します。
マルチインスタンスの場合、リソースの競合を回避する必要があります。
リソースの各パラメータは網掛け部分を参考にして、/etc/my.cnfを編集してください。

※インスタンスの起動、停止にはmysqld_multiを利用します。

mysqld_multiを利用すると、共通の環境、各マルチインスタンス環境を設定することができます。個別のインスタンスを起動する場合、[mysqld1],[mysqld2]、というように、[mysqld#]と、mysqldの後に起動インスタンスの数字をつけたセクション名を利用します。共通の設定は[mysqld]セクションに設定します。[mysqld_multi]セクションには、mysqld_multiが利用する値を設定します。

```

[client]
default-character-set = utf8
socket=/var/lib/mysql/mysql.sock

# mysqld_multi
[mysqld_multi]
mysqld      = /usr/bin/mysqld_safe
mysqladmin  = /usr/bin/mysqladmin

[mysqld]
character-set-server=utf8
log_bin=mysql-bin
innodb_flush_method=O_DIRECT
innodb_flush_log_at_trx_commit=1
sync_binlog=1

[mysqld1]
port=3307
server_id =10
datadir=/mnt/mirror1/mysql1
socket=/mnt/mirror1/mysql1/mysql1.sock
pid-file=/mnt/mirror1/mysql1/labra.pid
tmpdir=/tmp/tmp1 ←
innodb_buffer_pool_size=128M

[mysqld2]
port=3308
server_id =20
datadir=/mnt/mirror2/mysql2
socket=/mnt/mirror2/mysql2/mysql2.sock
pid-file=/mnt/mirror2/mysql2/labra.pid
tmpdir=/tmp/tmp2 ←
innodb_buffer_pool_size=128M

```

指定したディレクトリが存在しない場合は、事前に作成する必要があります。

3. 単一のインスタンスの動作確認をします。
mysqld_multiを使用して各インスタンスが起動、停止することを確認します。

● インスタンス1(mysqld1) の起動。

```

# mysqld_multi start 1
# mysqld_multi report
Reporting MySQL servers
MySQL server from group: mysqld1 is running
MySQL server from group: mysqld2 is not running

```

● インスタンス1(mysqld1) の停止

```

# mysqld_multi stop 1
# mysqld_multi report
Reporting MySQL servers
MySQL server from group: mysqld1 is not running
MySQL server from group: mysqld2 is not running

```


- インスタンス2(mysql2) の起動

```
# mysqld_multi start 2
# mysqld_multi report
Reporting MySQL servers
MySQL server from group: mysql1 is not running
MySQL server from group: mysql2 is running
```

- インスタンス2(mysql2) の停止

```
# mysqld_multi stop 2
# mysqld_multi report
Reporting MySQL servers
MySQL server from group: mysql1 is not running
MySQL server from group: mysql2 is not running
```

4. 複数のインスタンスの動作確認をします。
mysqld_multiを使用して複数のインスタンスが同時に起動、停止することを確認します。

- 複数のインスタンスの起動

```
# mysqld_multi start
# mysqld_multi report
Reporting MySQL servers
MySQL server from group: mysql1 is running
MySQL server from group: mysql2 is running
```

- 複数のインスタンスの停止

```
# mysqld_multi stop
# mysqld_multi report
Reporting MySQL servers
MySQL server from group: mysql1 is not running
MySQL server from group: mysql2 is not running
```

5. 各インスタンスへのアクセスを確認します。
事前に各インスタンスが起動状態であることを確認してください。

- インスタンス1 (mysql1) へのアクセス確認

```
$ mysql --port=3307 --host=127.0.0.1 -uroot
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.0.17-MariaDB-log MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

MariaDB [(none)]>
```

- インスタンス2 (mysqld2) へのアクセス確認

```
$ mysql --port=3308 --host=127.0.0.1 -uroot
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.0.17-MariaDB-log MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

MariaDB [(none)]>
```

6. パスワードを設定します。
各インスタンスにログイン後、パスワードを設定します。

```
MariaDB [(none)]> set password for 'root'@'127.0.0.1' =password('pass');
MariaDB [(none)]> set password for 'root'@'localhost' =password('pass');
```

※上記パスワードはサンプルです。設計ポリシーにしたがってパスワードは変更してください。

7. インスタンスへのアクセス確認をします。
パスワードを使用して、インスタンスへのアクセス確認をします。
defaults-extra-fileオプションを使用して、パスワードを指定します。事前に接続情報を記載した構成ファイルを作成してください。

以下の例では、各インスタンスに対して構成ファイル「var/lib/mysql/mysqld1_sample.cnf」、
「var/lib/mysql/mysqld2_sample.cnf」を作成します。ファイルのパーミッションは600で作成します。

- 構成ファイル(var/lib/mysql/mysqld1_sample.cnf)

```
[client]
port = 3307
user = root
password = pass
host = 127.0.0.1
```

- 構成ファイル(var/lib/mysql/mysqld2_sample.cnf)

```
[client]
port = 3308
user = root
password = pass
host = 127.0.0.1
```

- インスタンス1 (mysqld1) へのアクセス確認

```
$ mysql --defaults-extra-file=/var/lib/mysql/mysqld2_sample.cnf
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.0.17-MariaDB-log MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

MariaDB [(none)]>
```

- インスタンス2 (mysqld2) へのアクセス確認

```
$ mysql --defaults-extra-file=/var/lib/mysql/mysqld2_sample.cnf
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.0.17-MariaDB-log MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

MariaDB [(none)]>
```

8. 各インスタンスヘータベースを作成します。

- インスタンス1 (mysqld1) ヘータベース(db1)を作成する

```
$ mysql --defaults-extra-file=/var/lib/mysql/mysqld1_sample.cnf -e
"show databases;"
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
| performance_schema |
+-----+
$ mysql --defaults-extra-file=/var/lib/mysql/mysqld1_sample.cnf -e
"create database db1"
$ mysql --defaults-extra-file=/var/lib/mysql/mysqld1_sample.cnf -e
"show databases;"
+-----+
| Database          |
+-----+
| db1               |
| information_schema |
| mysql             |
| performance_schema |
+-----+
```

- インスタンス2 (mysqld2) ヘデータベース(db2)を作成する

```

$ mysql --defaults-extra-file=/var/lib/mysql/mysqld1_sample2.cnf -e
"show databases;"
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
| performance_schema |
+-----+
$ mysql --defaults-extra-file=/var/lib/mysql/mysqld1_sample2.cnf -e
"create database db2"
$ mysql --defaults-extra-file=/var/lib/mysql/mysqld1_sample2.cnf -e
"show databases;"
+-----+
| Database          |
+-----+
| db2               |
| information_schema |
| mysql             |
| performance_schema |
+-----+

```

9. ユーザ作成、テーブル作成。

各インスタンスのユーザやテーブルは設計ポリシーにしたがって適宜作成してください。
作成したユーザで、接続先にフローティングIPを指定して、インスタンスへの接続確認をしてください。

10. インスタンスを停止します。

パスワードを使用して、インスタンスへの停止をします。
defaults-extra-fileオプションを使用して、パスワードを指定します。事前にパスワードを記載した構成ファイルを作成してください。

以下の例では、「7.インスタンスへのアクセス確認をします。」で作成した構成ファイル「var/lib/mysql/mysqld1_sample.cnf」、「var/lib/mysql/mysqld2_sample.cnf」に対して、パスワード情報を追記します。

- 構成ファイル(var/lib/mysql/mysqld1_sample.cnf)

```

[mysqld_multi]
password=pass

```

- 構成ファイル(var/lib/mysql/mysqld2_sample.cnf)

```

[mysqld_multi]
password=pass

```

- インスタンス1 (mysqld1) の停止

```

# mysqld_multi --defaults-extra-file=/var/lib/mysql/mysqld1_sample.cnf
stop 1

```

- インスタンス2 (mysqld2) の停止

```
# mysqld_multi --defaults-extra-file=/var/lib/mysql/mysqld2_sample.cnf
stop 2
```

1.2.4. マルチインスタンスの構築(サーバ2)

サーバ2上で、2つのインスタンスの設定をします。

事前に、サーバ1で各インスタンスが停止していることを確認し、各フェイルオーバーグループをサーバ2で起動してください。

サーバ1で作成したインスタンスをサーバ2上で使用するため、構成ファイルを変更します。「4-2-3. マルチインスタンスの構築(サーバ1)」の以下の手順(構成ファイルの編集)をサーバ2でも同様に実施してください。

- ・手順2 (/etc/my.cnfの編集)
- ・手順7 (/var/lib/mysqld/mysqld1_sample.cnf, mysqld2_sample.cnfの編集)
- ・手順10 (/var/lib/mysqld/mysqld1_sample.cnf, mysqld2_sample.cnfの編集)

構成ファイルの編集後、インスタンスの動作確認、また、データベースへアクセス確認を実施してください。

1.2.5. グループリソースの設定

以下のグループリソースを設定します。

execリソース

フェイルオーバーグループ(failover1)にインスタンス1の起動、停止を実行するexecリソースを追加します。

1. [グループリソース一覧] で [追加] をクリックします。
2. [グループ (failover1) のリソース定義] 画面が開きます。
[タイプ] ボックスでグループリソースのタイプ (execute resource) を選択して、[名前] ボックスにグループリソース名 (exec_MariaDB1) を入力します。[次へ] をクリックします。
3. 依存関係設定のページが表示されます。[次へ] をクリックします。
4. [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。
5. 詳細設定のページが表示されます。[この製品で作成したスクリプト] をチェックして、「Start Script(start.sh)」と「Stop Script(stop.sh)」を作成します。
(作成するサンプルスクリプトは付録を参照してください)
[完了] をクリックして設定を終了します。

同様にフェイルオーバーグループ(failover2)にインスタンス2の起動、停止を実行するスクリプトリソースを追加します。

なお、本構成では、MariaDBのサービスは常時起動を前提としています。

1.2.6. モニタリソースの設定

指定した対象を監視するモニタリソースをクラスタに追加します。

モニタリソースの回復動作は、設計ポリシーにしたがって適宜設定してください。

以下のモニタリソースを設定します。

カスタム監視リソース

インスタンス1を監視するカスタム監視リソースを追加します。

1. [グループ一覧] で [追加] をクリックします。
2. [モニタリソース一覧] が表示されます。[追加] をクリックします。[タイプ] ボックスでモニタリソースのタイプ (custom monitor) を選択し、[名前] ボックスにモニタリソース名 (genw_MariaDB1) を入力します。
[次へ] をクリックします。
3. 監視設定を入力します。監視タイミングに [活性時]、対象リソースに[exec_MariaDB1]を設定し、[次へ] をクリックします。
4. この製品で作成したスクリプトを選択します。
ファイルを編集し、インスタンス1の監視スクリプトを作成します。
(作成するサンプルスクリプトは付録を参照してください)
5. [回復動作] ボックスで [回復対象に対してフェイルオーバー実行] を選択します。
また、[回復対象] に[failover1] を設定します。
[完了] をクリックして設定を終了します。

同様にインスタンス2を監視するカスタム監視リソースを追加します。

サンプルスクリプトでは、各インスタンス上のデータベースに作成した監視テーブルに対する接続、更新確認を行います。

サンプルスクリプトを使用する場合は、事前に監視テーブルを作成してください。

例. データベース(watchdb)を作成し、データベース(watchdb)上に監視テーブル(watchtable)を作成する

```
MariaDB [(none)]> create database watchdb;
MariaDB [(none)]> use watchdb;
MariaDB [watchdb]> create table watchtable(num int);
MariaDB [watchdb]> insert into watchtable value(0);
MariaDB [watchdb]> commit;
```

付録 サンプルスクリプト

本スクリプトで設定するファイルは以下の通りです。
サンプルスクリプトとなりますので、環境に合わせて適宜修正の上、使用します。

- | | |
|--------------------------|---|
| start_MariaDB1_sample.sh | スクリプトリソース(exec_MariaDB1)に設定するインスタンス1を起動するサンプルスクリプト |
| stop_MariaDB1_sample.sh | スクリプトリソース(exec_MariaDB1)に設定するインスタンス1を停止するサンプルスクリプト |
| start_MariaDB2_sample.sh | スクリプトリソース(exec_MariaDB2)に設定するインスタンス2を起動するサンプルスクリプト |
| stop_MariaDB2_sample.sh | スクリプトリソース(exec_MariaDB2)に設定するインスタンス2を停止するサンプルスクリプト |
| genw_MariaDB1_sample.sh | カスタム監視リソース(genw_MariaDB1)に設定するインスタンス1を監視するサンプルスクリプト |
| genw_MariaDB2_sample.sh | カスタム監視リソース(genw_MariaDB2)に設定するインスタンス2を監視するサンプルスクリプト |

•start_MariaDB1_sample.sh

インスタンス1を起動するサンプルスクリプト

環境に合わせて、網掛け部分のパラメータを設定してください。

```
#!/bin/sh
#*****
#* Copyright (c) NEC Corporation 2015. All rights reserved.*
#* start_MariaDB1_sample.sh                                     *
#* title   : start_MariaDB1_sample                             *
#* date    : 2015/05/15                                        *
#*****

if [ "$CLP_EVENT" = "START" -o "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    #Start Instance
    mysqld_multi start 1
  else
    echo "ERROR_DISK from START"
    exit 1
  fi
else
  echo "NO_CLP"
  exit 1
fi
exit 0
```

•stop_MariaDB1_sample.sh

インスタンス1を停止するサンプルスクリプト

環境に合わせて、網掛け部分のパラメータを設定してください。

本スクリプトでは、インスタンスの停止の際にパスワードを記載した構成ファイル
(/var/lib/mysql/mysqlld1_sample.cnf)を指定しています。

```
#!/bin/sh
#*****
#* Copyright (c) NEC Corporation 2015. All rights reserved.*
#* stop_MariaDB1_sample.sh *
#* title : stop_MariaDB1_sample *
#* date : 2015/05/15 *
#*****

if [ "$CLP_EVENT" = "START" -o "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        #Stop Instance
        mysqlld_multi --defaults-extra-file=/var/lib/mysql/mysqlld1_sample.cnf stop 1
    else
        echo "ERROR_DISK from START"
        exit 1
    fi
else
    echo "NO_CLP"
    exit 1
fi
exit 0
```

•start_MariaDB2_sample.sh
インスタンス2を起動するサンプルスクリプト
環境に合わせて、網掛け部分のパラメータを設定してください。

```
#!/bin/sh
#*****
#* Copyright (c) NEC Corporation 2015. All rights reserved.*
#* start_MariaDB2_sample.sh *
#* title : start_MariaDB2_sample *
#* date : 2015/05/15 *
#*****

if [ "$CLP_EVENT" = "START" -o "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    #Start Instance
    mysqld_multi start 2
  else
    echo "ERROR_DISK from START"
    exit 1
  fi
else
  echo "NO_CLP"
  exit 1
fi
exit 0
```

•stop_MariaDB2_sample.sh

インスタンス2を停止するサンプルスクリプト

環境に合わせて、網掛け部分のパラメータを設定してください。

本スクリプトでは、インスタンスの停止の際にパスワードを記載した構成ファイル
(/var/lib/mysql/mysqlld2_sample.cnf)を指定しています。

```
#!/bin/sh
#*****
#* Copyright (c) NEC Corporation 2015. All rights reserved.*
#* stop_MariaDB2_sample.sh *
#* title : stop_MariaDB2_sample *
#* date : 2015/05/15 *
#*****

if [ "$CLP_EVENT" = "START" -o "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        #Stop Instance
        mysqlld_multi --defaults-extra-file=/var/lib/mysql/mysqlld2_sample.cnf stop 2
    else
        echo "ERROR_DISK from START"
        exit 1
    fi
else
    echo "NO_CLP"
    exit 1
fi
exit 0
```

・genw_MariaDB1_sample.sh

カスタム監視リソース(genw_MariaDB1)に設定するインスタンス1を監視するサンプルスクリプト

本スクリプトでは、インスタンスへの接続の際に接続情報を記載した構成ファイル

(/var/lib/mysql/mysqlld1_sample.cnf)を指定しています。

本サンプルスクリプトを使用する場合は、事前に監視用テーブル、構成ファイルを作成してください。

環境に合わせて、網掛け部分のパラメータを設定してください。

```
#!/bin/sh
#*****
#* Copyright (c) NEC Corporation 2015. All rights reserved.*
#* genw_MariaDB1_sample.sh *
#* title : genw_MariaDB1_sample *
#* date : 2015/05/15 *
#*****

watch_db=watchdb
watch_table=watchtable

mysql --defaults-extra-file=/var/lib/mysql/mysqlld1_sample.cnf -e "use $watch_db;
insert into $watch_table value(1);"

if [ $? -eq 1 ]
then
    echo "INSERT FAILED"
    exit 1
fi

mysql --defaults-extra-file=/var/lib/mysql/mysqlld1_sample.cnf -e "use $watch_db;
delete from $watch_table where num=1"

if [ $? -eq 1 ]
then
    echo "DELETE FAILED"
    exit 1
fi

exit 0
```

•genw_MariaDB2_sample.sh

カスタム監視リソース(genw_MariaDB2)に設定するインスタンス2を監視するサンプルスクリプト

本スクリプトでは、インスタンスへの接続の際に接続情報を記載した構成ファイル

(/var/lib/mysql/mysqlid2_sample.cnf)を指定しています。

本サンプルスクリプトを使用する場合は、事前に監視用テーブル、構成ファイルを作成してください。

環境に合わせて、網掛け部分のパラメータを設定してください。

```
#!/bin/sh
#*****
#* Copyright (c) NEC Corporation 2015. All rights reserved.*
#* genw_MariaDB2_sample.sh *
#* title : genw_MariaDB2_sample *
#* date : 2015/05/15 *
#*****

watch_db=watchdb
watch_table=watchtable

mysql --defaults-extra-file=/var/lib/mysql/mysqlid2_sample.cnf -e "use $watch_db;
insert into $watch_table value(1);"

if [ $? -eq 1 ]
then
    echo "INSERT FAILED"
    exit 1
fi

mysql --defaults-extra-file=/var/lib/mysql/mysqlid2_sample.cnf -e "use $watch_db;
delete from $watch_table where num=1"

if [ $? -eq 1 ]
then
    echo "DELETE FAILED"
    exit 1
fi

exit 0
```
