

# **DRBD、Heartbeat、Pacemaker による Zabbix サーバの HA クラスタ構築**

2013 年 1 月 7 日 初版  
2013 年 1 月 21 日 1.1 版

株式会社サードウェア  
ミラクル・リナックス株式会社

# 目次

<b>1 はじめに</b> .....	<b>1</b>
1.1 Zabbix について.....	1
1.2 Linux-HA クラススタックについて.....	1
1.3 本ドキュメントの目的.....	2
1.4 対象読者ならびに前提事項.....	2
<b>2 サーバ構成</b> .....	<b>3</b>
<b>3 CentOS 6.3 のインストール</b> .....	<b>4</b>
3.1 想定するネットワークパラメータ.....	4
3.2 ハードディスクのパーティションの設計.....	4
3.3 初期インストールウィザード .....	4
<b>4 OS 基本設定</b> .....	<b>5</b>
4.1 SSH 鍵生成と交換.....	5
4.2 SELinux の無効化.....	5
4.3 ファイアウォールの無効化.....	5
4.4 ネットワーク設定ファイルの修正.....	5
4.5 /etc/hosts の調整.....	5
4.6 DRBD レプリケーション領域の作成.....	6
4.7 NTP の設定.....	6
4.8 Postfix の設定.....	6
4.9 システムファイルの更新と再起動.....	6
<b>5 パッケージの追加</b> .....	<b>6</b>
5.1 Zabbix の追加インストール.....	6
5.2 Linux-HA クラススタックのインストール.....	6
5.2.1 CentOS 用リポジトリ定義ファイルの修正.....	7
5.2.2 DRBD のインストール.....	7
5.2.3 Heartbeat、Pacemaker のインストール.....	7
5.3 ブート時の起動サービスの調整 .....	7
<b>6 DRBD の設定</b> .....	<b>7</b>
6.1 DRBD の基本設定.....	8
6.1.1 /etc/drbd.d/global_common.conf.....	8
6.1.2 /etc/drbd.d/r0.res.....	8
6.2 DRBD の初期化.....	9
<b>7 MySQL サーバの初期設定</b> .....	<b>9</b>
<b>8 Web サーバの設定</b> .....	<b>10</b>
<b>9 Zabbix サーバの初期設定</b> .....	<b>11</b>
9.1 zabbix_server.conf の設定.....	11
9.2 起動スクリプトの修正.....	11
9.3 Zabbix サーバの初期設定.....	12
<b>10 Heartbeat の設定</b> .....	<b>13</b>
10.1 /etc/ha.d/ha.cf .....	13
10.2 /etc/ha.d/authkeys .....	13
10.3 Heartbeat の起動.....	13
<b>11 Pacemaker の設定</b> .....	<b>13</b>

11.1	crm コマンドの開始.....	14
11.2	クラスタの動作オプションの設定.....	14
11.3	DRBD 関連のリソースの定義.....	14
11.4	その他のリソースの登録.....	15
<b>12</b>	<b>Zabbix クラスタの動作確認と運用.....</b>	<b>16</b>
12.1	コールドブート.....	16
12.2	スイッチオーバ.....	16
12.3	ハード障害.....	17
12.4	リソースレベルの障害.....	17
12.5	Zabbix クラスタの運用.....	17
<b>13</b>	<b>最後に.....</b>	<b>18</b>



## 1 はじめに

---

このドキュメントは、Linux-HA クラスタスタック(DRBD、Heartbeat および Pacemaker)を使って Zabbix サーバの HA クラスタ(以下「Zabbix クラスタ」)を構築するための留意点と手順を説明したホワイトペーパーである。

### 1.1 Zabbix について

---

Zabbix はサーバ、ネットワーク、アプリケーションを監視するための統合監視ソフトウェアで、GNU General Public License (GPL) v2 ライセンスにもとづいて配布されているオープンソース・ソフトウェアである。このため、すべての機能が無償で利用できるが、Zabbix Japan およびそのパートナー各社がさまざまな商用サポートを提供している。

Zabbix は、以下のような特徴を持っている。

- マルチプラットフォーム対応
- エージェントレス監視とエージェントを使った監視
- 多数の監視機能を標準で提供し、さらにカスタムスクリプトで監視機能を拡張できる
- Web インタフェースと API の提供
- 収集した監視データを RDBMS に長期保存
- 各種グラフ作成機能
- メールなどによる障害通知

すべての Zabbix 監視システムには Zabbix サーバが必要で、大規模システムやミッションクリティカルなシステムでは Zabbix サーバの可用性向上が大きな課題となっている。

参照 URL:

<http://www.zabbix.jp/>

<http://www.zabbix.com/jp/>

### 1.2 Linux-HA クラスタスタックについて

---

Linux-HA クラスタスタックは、サーバシステムの可用性を向上したり、ストレージ故障によってデータを喪失することを防止することを目的としたソフトウェア群である。次の4つのソフトウェアおよび関連パッケージによって構成されている。

#### DRBD (Distributed Replicated Block Device)

2 台のサーバのストレージ(パーティションまたは論理ボリューム)にリアルタイムに同一データを書き込む(リアルタイムレプリケーション)。このため、片方のサーバやストレージ自体が故障した場合にもデータを喪失せず、正常なサーバ側でサービスを継続できるようになる。

#### Heartbeat

## DRBD、Heartbeat、Pacemaker による Zabbix サーバの高可用クラスタ構築

HA クラスタを構成するサーバの正常稼働を相互に監視するソフトウェア。

### Pacemaker

HA クラスタが提供するサービスを監視し、サービスを提供していたサーバがダウンした場合に他のサーバでサービスを継続させるよう、クラスタ全体の可用性を維持管理する。

### リソースエージェント

HA クラスタがサービスを提供するのに必要な起動/停止/監視スクリプトを集めたライブラリ。データベース、ファイルサーバなど、約 60 種類のスクリプトが用意されている。これらはすべて GNU General Public License にもとづいて配布されている。このため、すべての機能が無償で利用できるが、主要な開発メンバーである LINBIT 社(オーストリア)は商用サポートを提供している。日本では、サードウェアが LINBIT の総代理店になっており、サードウェアとそのパートナー各社がさまざまな商用サポートを提供している。

参照 URL:

<http://www.drbd.org/ja/>  
[http://www.linux-ha.org/wiki/Main\\_Page/ja](http://www.linux-ha.org/wiki/Main_Page/ja)  
<http://clusterlabs.org/>  
<http://linux-ha.example.com/>

## 1.3 本ドキュメントの目的

---

このドキュメントの目的は、Zabbix サーバの可用性を高めるために HA クラスタを構築する手順を説明することである。このドキュメントにもとづいて構成した HA クラスタは、以下のような機能を提供する。

- HA クラスタは 2 台のサーバシステムで構成される。それぞれのサーバは物理サーバであっても仮想サーバであってもかまわない。
- つねにどちらか 1 台のサーバ上で Zabbix サーバが稼働し、サーバ等の監視、Web インタフェース、障害通知などの全機能を提供する。
- Zabbix サーバの監視設定や収集した監視データは MySQL データベースに格納される。このデータは、常時 2 台のサーバのディスクにリアルタイムにレプリケートされる。
- サービスを提供しているサーバがダウンすると、約 1 分前後で他のサーバ上で Zabbix サーバが起動し、監視等のサービスが引き継がれる。このとき、データベースのデータは失われない。

## 1.4 対象読者ならびに前提事項

---

このドキュメントは、以下の前提条件を満たす読者を想定している。

- Linux ベースの MySQL および Apache サーバを構築・運用できる
- Zabbix の基礎知識を有している
- Heartbeat および Pacemaker を使った HA クラスタシステムの基礎知識および構築のノウハウを知っている

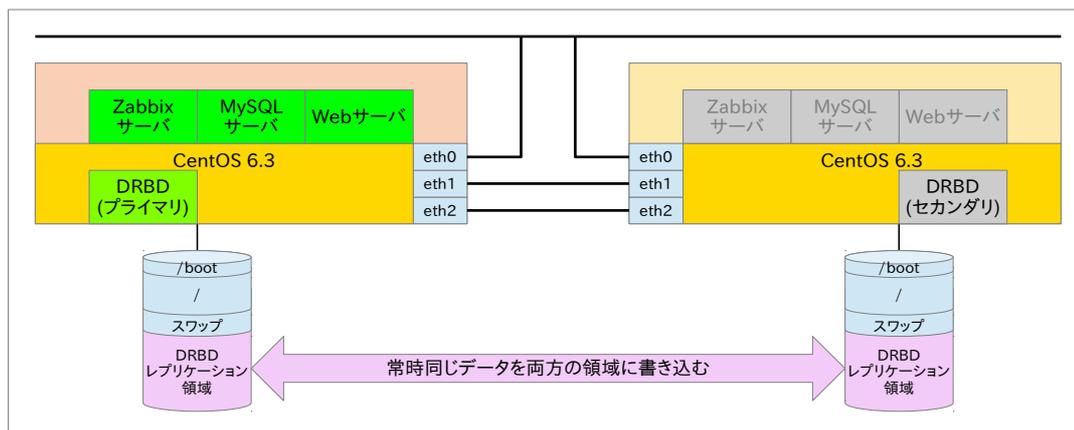
## DRBD、Heartbeat、Pacemaker による Zabbix サーバの HA クラスタ構築

このドキュメントは、以下のような仕様の仮想サーバを使って、Zabbix クラスタを構築する手順を説明している。

- 仮想サーバは KVM の virtio を採用している。このため、ディスクデバイス名は /dev/vda などとなっている(物理サーバや他の仮想基盤では /dev/sda などに読み替えてほしい)。
- OS は CentOS 6.3 (64 ビット) を採用する。
- 仮想ディスクサイズは 8GB になっている。実運用にあたっては、データベースに必要なサイズを考慮してディスクサイズを決定する必要がある。
- 各サーバはそれぞれ 3 つのネットワークインタフェースを持ち、そのうちの 2 組はサーバ間を直結するよう接続してある(後述のシステム構成図を参照)。

## 2 サーバ構成

Zabbix クラスタを構成する 2 台のサーバのディスクやネットワーク構成は、下図のようになる。



### 推奨するネットワーク構成

安定した HA クラスタを実現する上で、eth0～eth2 の 3 つのネットワークインタフェースを用意して、eth1 と eth2 をサーバ間で直結するのが望ましい。

### ハードディスクの構成

上図のようにディスクドライブが 1 つだけの場合、OS インストール時に DRBD がレプリケートする領域を空けておく必要がある。2 つのディスクドライブが利用できる場合には、片方に OS をインストールし、他方を DRBD レプリケーション領域に使う。

### クラスタの動作状態

上図では、左側のサーバで Zabbix サーバなどのサービスが起動している(アクティブ機)。右側のサーバは、左側のサーバのダウンに備えて待機している(スタンバイ機)。Zabbix が書き込んだ監視データは、DRBD レプリケーション領域に書き込まれるが、DRBD はスタンバイ機にもリアルタイムで同じデータを書き込む(レプリケーション)。

## 3 CentOS 6.3 のインストール

### 3.1 想定するネットワークパラメータ

通常、各ネットワークインタフェースはそれぞれ固有の IP アドレスを持つが、Zabbix サーバがその他のホストと通信するための追加の IP アドレス(仮想 IP アドレス)が必要になる。仮想 IP アドレスは、アクティブ機側に割り当てることになる。

このことも考慮して、以下の説明では、次のようなネットワークパラメータを想定する。

	Zabbix1	Zabbix2
ホスト名	zabbix1.example.com	zabbix2.example.com
仮想 IP アドレス	10.30.101.70/16	
eth0	10.30.101.71/16	10.30.101.72/16
eth1	172.16.2.1/28	172.16.2.2/28
eth2	172.16.3.1/28	172.16.3.2/28
ゲートウェイ	10.30.0.253	
DNS サーバ	10.30.0.9	

### 3.2 ハードディスクのパーティションの設計

想定する各サーバには 8GB のハードディスクが 1 つだけ接続されている。MySQL のデータを DRBD でレプリケートできるように、次のようにパーティションを分割することとする。これは、2 台とも同一に揃える必要がある。

パーティション	サイズ	用途
/dev/vda1	200MB	/boot パーティション
/dev/vda2	3000MB	/パーティション
/dev/vda3	1000MB	スワップ領域
/dev/vda4	残り全部	DRBD レプリケーション領域(インストール時には割り当てず、インストール後に fdisk コマンドで割り当てる)

### 3.3 初期インストールウィザード

CentOS 6.3 の初期インストールウィザードでは、次のように設定する。

1. 言語、キーボードは日本語を選択する。
2. 新規インストールを選択する。
3. それぞれのホスト名および eth0~eth2 の IP アドレスを割り当てる。
4. パーティション構成は「カスタムレイアウト」を選択し、/dev/vda1~/dev/vda3 までを上述のように割り当てる。
5. インストールするソフトウェアは、Minimal を選択し、「今すぐカスタマイズ」を選

んで以下のカテゴリを追加インストールする。

- 「Web サービス」の「PHP サポート」および「Web サーバー」
- 「データベース」の「MySQL データベースサーバー」および「MySQL データベース接続クライアント」
- 「ベースシステム」の「ベース」

## 4 OS 基本設定

---

インストールが終わったら、OS の基本的な設定を実施する。なお、特に断らない限り、OS 基本設定は両方のサーバで同一に揃える必要がある。

### 4.1 SSH 鍵生成と交換

---

HA クラスタを構築する過程で、片方のサーバで作成した設定ファイルをもう一方にリモートコピーする必要がある。このため、それぞれのサーバで SSH ホスト鍵を生成して交換し、相互に root 権限でリモートログイン/リモートコピーできるようにするのが望ましい。

- ssh-keygen を実行して root ユーザの鍵を生成する。
- zabbix1.example.com と zabbix2.example.com のそれぞれの公開鍵を相互に /root/.ssh/authorized\_keys に登録する。

### 4.2 SELinux の無効化

---

SELinux が有効な状態で HA クラスタを正常に運用するには、SELinux に関するきわめて高度な知識と経験が必要になる。このため、今回は SELinux を無効に設定する。

### 4.3 ファイアウォールの無効化

---

セキュリティ対策上ファイアウォールを無効化するのは好ましくない。しかし HA クラスタの設定が完了するまでの間、ファイアウォールを無効にしておく方が望ましい。

### 4.4 ネットワーク設定ファイルの修正

---

Zabbix サーバから送信するパケットのソース IP アドレスを仮想 IP アドレスにするために、後で IPsrcaddr というリソースエージェントを使用する。このリソースエージェントが正常に動作するために、ネットワーク設定ファイル(/etc/sysconfig/network-scripts/ifcfg-eth\*)に NOZEROCONF=yes を追加する。

### 4.5 /etc/hosts の調整

---

構築するサーバのホスト名と IP アドレスを、相互に /etc/hosts に登録する。

構築中のサーバのホスト情報がすでに DNS サーバに登録されている場合には、このステップは不要である。

## 4.6 DRBD レプリケーション領域の作成

---

fdisk コマンドを使って/dev/vda4 を作成する。この時点でファイルシステムを作成する必要はない。

## 4.7 NTP の設定

---

HA クラスタの運用管理にあたって、システムクロックが正確に同期していることは、きわめて重要である。このため、可能な限り NTP プロトコルで時刻同期する。

## 4.8 Postfix の設定

---

各サーバから送信したメールが正常に中継されるよう、必要に応じて Postfix の設定を調整する。

## 4.9 システムファイルの更新と再起動

---

yum update を実行してシステムファイルを最新に更新し、サーバを再起動する。

# 5 パッケージの追加

---

Zabbix および Linux-HA クラスタスタックのパッケージをインストールする。これらのパッケージは、両方のサーバにインストールする必要がある。

## 5.1 Zabbix の追加インストール

---

ZABBIX-JP の Web サイトに紹介されている手順に従って、Zabbix の RPM ファイルをインストールする。

参照 URL:

<http://www.zabbix.jp/documents/installation/zabbix-jp-repository>

## 5.2 Linux-HA クラスタスタックのインストール

---

Linux-HA クラスタスタックの構成プログラムは、(1) コミュニティなどのリポジトリからのインストール、(2) 商用サポートユーザ向けのリポジトリからのインストール<sup>1</sup>、の 2 とおりの方法がある。ここでは、前者の方法によるインストール手順を説明する。

---

<sup>1</sup> DRBD、Heartbeat、Pacemaker など HA クラスタ構築に必要なパッケージをすべて一括してインストールできるメリットがある(詳細は <http://linux-ha.3ware.co.jp/support.html> を参照)。

### 5.2.1 CentOS 用リポジトリ定義ファイルの修正

CentOS 6.3 は Pacemaker 1.1 を提供しているが、これは Heartbeat と組み合わせて動作させることができない。このため、CentOS 用リポジトリから pacemaker をインストールしないよう、`/etc/yum.repos.d/Centos-Base.repo` の各セクションに次の行を挿入する。

```
exclude=pacemaker* heartbeat* resource-agents*
```

### 5.2.2 DRBD のインストール

CentOS 6.3 用の DRBD は ELRepo からインストールする。

```
rpm --import http://elrepo.org/RPM-GPG-KEY-elrepo.org
rpm -Uvh http://elrepo.org/elrepo-release-6-4.el6.elrepo.noarch.rpm
yum -y install drbd83 kmod-drbd83
```

### 5.2.3 Heartbeat、Pacemaker のインストール

Heartbeat、Pacemaker および関連パッケージは、Linux-HA Japan サイトからダウンロードしてインストールする。

参照 URL:

<http://sourceforge.jp/projects/linux-ha/releases/56470>

`pacemaker-1.0.12-1.2.el6.x86_64.repo.tar.gz` をダウンロードしたら、以下の手順でインストールする。

```
cd /tmp
tar xvzf <ダウンロードしたディレクトリ>/pacemaker-1.0.12-1.2.el6.x86_64.repo.tar.gz
cd pacemaker-1.0.12-1.2.el6.x86_64.repo
yum -c pacemaker.repo install pacemaker heartbeat
```

## 5.3 ブート時の起動サービスの調整

HA クラスタシステムでは、クラスタとして制御するサービスをブート時に起動するのは好ましくない。このため、Zabbix クラスタでは以下のサービスを自動起動しないように設定する。

```
chkconfig drbd off
chkconfig heartbeat off
chkconfig mysqld off
chkconfig httpd off
chkconfig zabbix-server off
```

ここまでで Zabbix クラスタ構築に必要なパッケージと基本的な設定が完了した。いよいよこれから HA クラスタの設定に入る。

## 6 DRBD の設定

クラスタ構築の最初のステップは、DRBD を使ったデータレプリケーションの準備である。

## DRBD、Heartbeat、Pacemaker による Zabbix サーバの高可用クラスタ構築

- 各サーバ上の DRBD は、レプリケーション領域(/dev/vda4)を排他的に管理する。
- DRBD 同士は eth1 (eth2 でもかまわない)を使って相互に通信する。
- 片方の DRBD は「プライマリ」という役割で動作し、ファイルシステムやアプリケーションに対して/dev/drbd0 というブロックデバイスを提供する。ファイルシステムなどは、/dev/vda4 ではなく/dev/drbd0 にアクセスする。
- プライマリ側の/dev/drbd0 に書き込まれたデータは、プライマリ側の/dev/vda4 に書き込まれると同時に、セカンダリ DRBD (もう 1 台のサーバ)に送られ、セカンダリ側/dev/vda4 にもデータが書き込まれる。

これらを実現するための設定は、以下のようになる。

### 6.1 DRBD の基本設定

DRBD 設定ファイルは/etc/drbd.conf および/etc/drbd.d/\*になる。どちらかのサーバで設定ファイルを作成し、他のサーバにリモートコピーする。

#### 6.1.1 /etc/drbd.d/global\_common.conf

デフォルトのままをかまわない。

#### 6.1.2 /etc/drbd.d/r0.res

どちらかのサーバで新規に設定ファイルを作成して、他方にリモートコピーする。

```
resource r0 {
    protocol C;
    syncer {
        rate 20M;
        use-rle;
        verify-alg md5;
    }
    disk {
        on-io-error detach;
    }
    device      /dev/drbd0;
    disk        /dev/vda4;

    on zabbix1.example.com {
        address      172.16.2.1:7788;
        meta-disk    internal;
    }
    on zabbix2.example.com {
        address      172.16.2.2:7788;
        meta-disk    internal;
    }
}
```

## 6.2 DRBD の初期化

DRBD は初期化作業が必要である。以下の手順に従って初期化する。

ステップ 1: 両サーバで DRBD を起動する

```
/etc/init.d/drbd start
```

ステップ 2: どちらか一方の DRBD を初回のプライマリに昇格させる

以下の操作はどちらで実行してもかまわないが、説明の都合上 zabbix1.example.com で実行することとする。

```
drbdadm -- --overwrite-data-of-peer primary r0
```

プライマリ側のすべてのディスクブロックのデータがセカンダリ側に送られて、両方のディスクデータを同期させる(初期フル同期)。

ステップ 3: プライマリ側でファイルシステムを作成する。

```
mkfs.ext4 /dev/drbd0
```

## 7 MySQL サーバの初期設定

Zabbix クラスタでは、MySQL データベースのデータ領域を DRBD レプリケーション領域に置く。このことを実現するために、DRBD の初期化に引き続いて次のように MySQL を初期設定する。ステップ 1 のみ両サーバで実行するが、ステップ 2 以降は下記に指示されたサーバで実行する。

ステップ 1: 両サーバでマウントポイントを作成する

このステップのみ両ノードで実行する必要がある。

```
mkdir /h
```

ステップ 2: zabbix1.example.com で DRBD 領域をマウントする

```
mount /dev/drbd0 /h
```

ステップ 3: zabbix1.example.com で MySQL データ用ディレクトリを作成する

```
mkdir /h/mysql
chown mysql:mysql /h/mysql
```

ステップ 4: zabbix1.example.com で /h/mysql を /var/lib/mysql に bind マウントする

```
mount -o bind /h/mysql /var/lib/mysql
```

ステップ 5: zabbix1.example.com で MySQL サーバを起動して初期化する

/etc/my.cnf を修正する(以下は設定例で、必ずしも最適とは言えない可能性もある)。

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
```

## DRBD、Heartbeat、Pacemaker による Zabbix サーバの高可用クラスタ構築

```
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
default-character-set=utf8
skip-character-set-client-handshake
innodb_file_per_table
innodb_buffer_pool_size=256M
innodb_log_file_size=64M
innodb_log_files_in_group=2
```

/etc/my.cnf を zabbix2.example.com にリモートコピーする。

続いて MySQL サーバを起動する。

```
/etc/init.d/mysqld start
```

root パスワードの作成など、その他の MySQL 初期設定作業を実施する(詳細は省略)。

ステップ 6: zabbix1.example.com で MySQL サーバを停止してアンマウントする

```
/etc/init.d/mysqld stop
umount /h/mysql
umount /h
```

ステップ 7: zabbix1.example.com の DRBD をセカンダリに降格する

```
drbdadm secondary r0
```

ステップ 8: zabbix2.example.com の DRBD をプライマリに昇格する

```
drbdadm primary r0
```

ステップ 9: zabbix2.example.com でデータベース領域をマウントする

```
mount /dev/drbd0 /h
mount -o bind /h/mysql /var/lib/mysql
```

ステップ 10: zabbix2.example.com で MySQL サーバを起動する

```
/etc/init.d/mysqld start
```

ステップ 5 で設定した root パスワードでログインできることなど、データベースの挙動が zabbix1.example.com と同じになっていることを確認する。

ステップ 11: zabbix2.example.com で MySQL サーバや DRBD を停止する

```
/etc/init.d/mysqld stop
umount /h/mysql
umount /h
/etc/init.d/drbd stop
```

ステップ 12: zabbix1.example.com の DRBD を停止する

```
/etc/init.d/drbd stop
```

## 8 Web サーバの設定

Zabbix の Web インタフェースとして正しく動作するよう、zabbix1.example.com 側で/etc/httpd の設定を確認し、必要に応じて設定を変更する。

なお、apache リソースエージェントが正常に動作するために、server-status を有効にする必要がある。/etc/httpd/conf/httpd.conf の 921 行目付近を以下のように修正する。

```
#
# Allow server status reports generated by mod_status,
# with the URL of http://servername/server-status
# Change the ".example.com" to match your domain to enable.
#
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from localhost
</Location>
```

設定が終わったら、zabbix2.example.com に設定ファイルをリモートコピーする。

## 9 Zabbix サーバの初期設定

### 9.1 zabbix\_server.conf の設定

必要に応じて/etc/zabbix/zabbix\_server.conf の DBHost、DBName、DBuser、DBPasswordなどを設定する。

zabbix1.example.com で編集して zabbix2.example.com にリモートコピーする。

### 9.2 起動スクリプトの修正

デフォルトでインストールされる Zabbix サーバの起動スクリプト(/etc/init.d/zabbix-server)は Zabbix サーバの親プロセスが異常終了したときの挙動に問題があるため、修正した起動スクリプト/etc/init.d/zabbix-server-for-HA を作成する。

2つのスクリプトの差分を以下に示す。

```
--- /etc/init.d/zabbix-server      2012-09-02 20:25:28.000000000 +0900
+++ /etc/init.d/zabbix-server-for-HA2013-01-05 19:23:39.374873251 +0900
@@ -43,6 +43,8 @@

    RETVAL=0

+ [ -f $PIDFILE ] && pid=$(cat $PIDFILE)
+
    case "$1" in
        start)
            echo -n "Starting zabbix server: "
@@ -77,6 +79,13 @@
        status)
            status $ZABBIX
            RETVAL=$?
+         if [ $RETVAL -eq 0 ]; then
+             if ! ps-tree -p | grep -q "zabbix_server($pid)"; then
```

```

+         RETVAL=1
+         killall -9 zabbix_server
+         echo "Error: No parent process exists."
+     fi
+ fi
+ ;;
*)
    echo "Usage: $0 {start|stop|restart|condrestart|reload|status}"

```

zabbix1.example.com でスクリプトを作成したら、zabbix2.example.com にリモートコピーする。

### 9.3 Zabbix サーバの初期設定

zabbix1.example.com 側で Zabbix サーバ自体を初期設定する。ただし DRBD、MySQL サーバ、仮想 IP アドレス、Web サーバなどを手動で起動する必要があるため、以下の手順に従う必要がある。

ステップ 1: 両サーバの DRBD を起動する

```
/etc/init.d/drbd start
```

ステップ 2: zabbix1.example.com で MySQL サーバを起動する

```
drbdadm primary r0
mount /dev/drbd0 /h
mount -o bind /h/mysql /var/lib/mysql
/etc/init.d/mysqld start
```

ステップ 3: zabbix1.example.com で仮想 IP アドレスを手動で割り当てる

```
ifconfig eth0:0 10.30.101.70 netmask 255.255.0.0 up
```

ステップ 4: zabbix1.example.com で Web サーバを起動する

```
/etc/init.d/httpd start
```

ステップ 5: zabbix1.example.com で Zabbix サーバを初期設定する

ZABBIX-JP のインストール解説ページ(<http://www.zabbix.jp/documents/installation/install-rpm>)などを参考にして、MySQL データベースと Zabbix サーバを初期設定する。

設定に伴って/etc/php.ini その他の設定ファイルを修正した場合は、すべて zabbix2.example.com にリモートコピーする。

ステップ 6: zabbix1.example.com で Web サーバ、仮想 IP アドレス、MySQL サーバを停止する

```
/etc/init.d/httpd stop
ifconfig eth0:0 down
/etc/init.d/mysqld stop
umount /h/mysql
umount /h
```

ステップ 7: 両サーバの DRBD を停止する

```
/etc/init.d/drbd stop
```

## 10 Heartbeat の設定

Heartbeat は主にクラスタを構成するノードの監視を受け持つ。同時に、Pacemaker のための基盤サービスを提供する。設定ファイルは `/etc/ha.d/ha.cf` と `/etc/ha.d/authkeys` の 2 つである。

いずれかのサーバで設定ファイルを作成して、もう一方のサーバにリモートコピーする。

### 10.1 /etc/ha.d/ha.cf

最小限の設定は以下のようになる。

```
crm yes
debugfile /var/log/ha-debug
logfile /var/log/ha-log
node zabbix1.example.com
node zabbix2.example.com
bcast eth1
bcast eth2
```

### 10.2 /etc/ha.d/authkeys

```
auth 1
1 sha1 uQ3fUrlwsoJH9so3dEsplDfUIhrEAS8r
```

このファイルを作成したら、次のコマンドを実行してオーナーおよびパーミッションを調整する。

```
chmod 600 /etc/ha.d/authkeys
chown root:root /etc/ha.d/authkeys
```

設定が終わったら、もう一方のサーバにリモートコピーする。

### 10.3 Heartbeat の起動

設定ファイルを作成したら、両方のサーバで Heartbeat を起動する。

```
/etc/init.d/heartbeat start
```

## 11 Pacemaker の設定

Heartbeat が起動して動作を開始したら、Pacemaker を設定する。

- 端末ウィンドウを新たに 1 つ開いてどちらかのサーバにログインし、`crm_mon` コマンドを実行する。クラスタの動作状態が端末に表示され、動作状態が変わったらただちに表示に反映される。
- 別の端末ウィンドウで `crm` コマンドを実行して、クラスタの設定を対話式に実行する。

以下の出力例では、ユーザがタイプ入力する文字を太字で表示する。

## 11.1 crm コマンドの開始

crm コマンドを実行したら対話式シェルのプロンプトが表示される。クラスタの設定を投入するために、configure コマンドを実行する。

```
[root@zabbix1 ~]# crm
crm(live)# configure
INFO: building help index
crm(live)configure#
```

## 11.2 クラスタの動作オプションの設定

property コマンドを使ってクラスタのグローバルな動作オプションを設定する。設定を有効にするために commit コマンドも実行する。

```
crm(live)configure# property default-resource-stickiness="200" \
> no-quorum-policy="ignore" \
> stonith-enabled="false"
crm(live)configure# commit
```

## 11.3 DRBD 関連のリソースの定義

DRBD 関連のリソースを登録して commit する。

```
crm(live)configure# primitive res_drbd_r0 ocf:linbit:drbd \
>   params drbd_resource="r0" \
>   op monitor interval="20" role="Slave" timeout="20" \
>   op monitor interval="10" role="Master" timeout="20" \
>   op start interval="0" timeout="240" \
>   op stop interval="0" timeout="100"
crm(live)configure# ms ms_drbd_r0 res_drbd_r0 \
>   meta master-max="1" master-node-max="1" clone-max="2" \
>   clone-node-max="1" notify="true"
crm(live)configure# commit
```

しばらく待つと、crm\_mon の表示が変化する。DRBD が両方のサーバで起動し、どちらか片方が Masters (プライマリ DRBD) になって他方が Slaves (セカンダリ DRBD) になっていることがわかる。

```
=====
Last updated: Sat Jan  5 19:44:12 2013
Stack: Heartbeat
Current DC: zabbix2.example.com (5244d088-c2b4-4e07-9fff-bd3b91ef08e9) - partiti on with
quorum
Version: 1.0.12-0666152e
2 Nodes configured, unknown expected votes
1 Resources configured.
=====

Online: [ zabbix1.example.com zabbix2.example.com ]

Master/Slave Set: ms_drbd_r0
```

```
Masters: [ zabbix1.example.com ]
Slaves: [ zabbix2.example.com ]
```

## 11.4 その他のリソースの登録

その他のリソースを一括して登録する(下記出力をコピー&ペーストするとよいだろう)。

```
primitive res_httpd ocf:heartbeat:apache \
    params configfile="/etc/httpd/conf/httpd.conf" \
    op start interval="0" timeout="40" \
    op stop interval="0" timeout="120" \
    op monitor interval="30" timeout="30"
primitive res_ip ocf:heartbeat:IPaddr2 \
    params ip="10.30.101.70" cidr_netmask="16" \
    op monitor interval="50" timeout="30"
primitive res_mount ocf:heartbeat:Filesystem \
    params device="/dev/drbd0" fstype="ext4" directory="/h" options="noatime" \
    op monitor interval="10" timeout="60" \
    op start interval="0" timeout="60" \
    op stop interval="0" timeout="60"
primitive res_mount_mysql ocf:heartbeat:Filesystem \
    params device="/h/mysql" fstype="ext4" directory="/var/lib/mysql" \
    options="bind,noatime" \
    op monitor interval="10" timeout="60" \
    op start interval="0" timeout="60" \
    op stop interval="0" timeout="60"
primitive res_mysql ocf:heartbeat:mysql \
    params binary="/usr/bin/mysqld_safe" pid="/var/run/mysqld/mysqld.pid" \
    op start interval="0" timeout="120" \
    op stop interval="0" timeout="120" \
    op monitor interval="30" timeout="30"
primitive res_srcaddr ocf:heartbeat:IPsrcaddr \
    params ipaddress="10.30.101.70" \
    op monitor interval="50" timeout="30"
primitive res_zabbix_server lsb:zabbix-server-for-HA \
    op start interval="0" timeout="30" \
    op stop interval="0" timeout="120" \
    op monitor interval="30" timeout="30"
group zabbix res_mount res_mount_mysql res_ip res_srcaddr res_httpd \
    res_mysql res_zabbix_server
location l_zabbix zabbix 100: zabbix1.example.com
colocation c_zabbix inf: zabbix ms_drbd_r0:Master
order o_zabbix inf: ms_drbd_r0:promote zabbix:start
```

すべてのリソースを登録したら `commit` を実行する。

クラスタとして正常に動作を開始したら、`crm_mon` の表示は次のようになる。

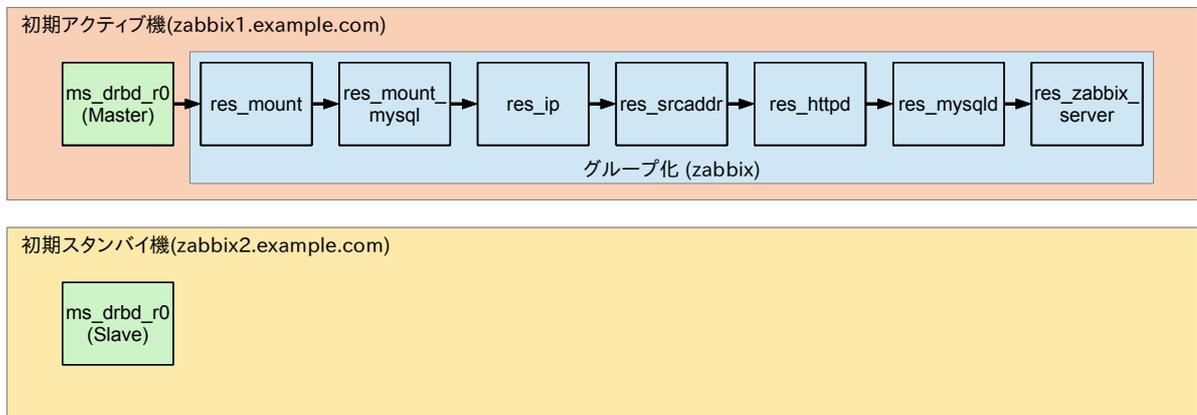
```
=====
Last updated: Sat Jan  5 20:00:58 2013
Stack: Heartbeat
Current DC: zabbix2.example.com (5244d088-c2b4-4e07-9fff-bd3b91ef08e9) - partiti on with
quorum
Version: 1.0.12-066152e
2 Nodes configured, unknown expected votes
2 Resources configured.
```

## DRBD、Heartbeat、Pacemaker による Zabbix サーバの高可用クラスタ構築

```
=====
Online: [ zabbix1.example.com zabbix2.example.com ]

Master/Slave Set: ms_drbd_r0
  Masters: [ zabbix1.example.com ]
  Slaves: [ zabbix2.example.com ]
Resource Group: zabbix
  res_mount (ocf::heartbeat:Filesystem): Started zabbix1.example.com
  res_mount_mysql (ocf::heartbeat:Filesystem): Started zabbix1.example.com
  res_ip (ocf::heartbeat:IPaddr2): Started zabbix1.example.com
  res_srcaddr (ocf::heartbeat:IPsrcaddr): Started zabbix1.example.com
  res_httpd (ocf::heartbeat:apache): Started zabbix1.example.com
  res_mysqlld (ocf::heartbeat:mysql): Started zabbix1.example.com
  res_zabbix_server (lsb::zabbix-server-for-HA): Started zabbix1.example.com
```

このクラスタシステムで制御するリソースの遷移状態を下図に示す。



## 12 Zabbix クラスタの動作確認と運用

Zabbix クラスタが動作するようになったら、さまざまな動作確認を実施する。典型的な動作確認方法をいくつか紹介する。

### 12.1 コールドブート

ステップ 1: zabbix1.example.com と zabbix2.example.com の両方をシャットダウンして再起動する。

ステップ 2: (heartbeat を自動起動しない設定にしてある場合)再起動後両サーバで heartbeat を起動する。

ステップ 3: どちらかのサーバで crm\_mon を実行し、Zabbix サーバその他のリソースが zabbix1.example.com で起動することを確認する。

### 12.2 スイッチオーバー

サーバ機器やソフトウェアメンテナンスのために、アクティブなサーバを切り替えたいことがある。これを想定して、次のような手順でリソースを移動できることを確認する。

ステップ 1: どちらかのサーバで `crm_mon` を実行しておく。

ステップ 2: どちらかのサーバで `crm` を起動して次のコマンドを実行する。

```
[root@zabbix1 ~]# crm
crm(live)# resource
crm(live)resource# move zabbix
WARNING: Creating rsc_location constraint 'cli-standby-zabbix' with a score of
-INFINITY for resource zabbix on zabbix1.example.com.
    This will prevent zabbix from running on zabbix1.example.com until the
constraint is removed using the 'crm_resource -U' command or manually with
cibadmin
    This will be the case even if zabbix1.example.com is the last node in the
cluster
    This message can be disabled with -Q
```

それまでアクティブであったサーバ側のリソースが順次停止し、もう 1 台のサーバがアクティブになることを確認する。Zabbix GUI にアクセスできることや、Zabbix サーバが正常に動作していることも確認する。

ステップ 3: 逆方向にスイッチオーバーできることを確認する。

`move zabbix` を実行するたびにリソースがサーバ間を移動する。

ステップ 4: `unmove zabbix` を実行する。

```
crm(live)resource# unmove zabbix
```

## 12.3 ハード障害

ステップ 1: アクティブサーバの電源を抜く

約 30 秒後にもう 1 台のサーバでリソースが起動することを確認する。フェールオーバーすることが確認できたら、停止したサーバを再起動し、`heartbeat` を開始しておく。

ステップ 2: スタンバイサーバの電源を抜く

リソースのフェールオーバーは起こらず、アクティブサーバのみでサービスが継続することを確認する。確認が終わったら、停止したサーバを再起動して `heartbeat` を開始しておく。

## 12.4 リソースレベルの障害

メモリ不足その他の一時的な理由でプロセスが異常終了した場合、今回設定したクラスタは、そのプロセスの再起動を試みる。この挙動を確認するために、`httpd`、`mysqld`、`zabbix_server` プロセスを `kill` コマンドで強制終了させ、約 30 秒以内に再起動することを確認する。

`zabbix_server` については、`pstree` コマンドで親プロセスと子プロセスを調べ、親プロセスと子プロセスに分けてチェックする。

## 12.5 Zabbix クラスタの運用

HA クラスタは、2 台のサーバが互いに通信してクラスタリソースの動作を制御する。このため、運用にあたっていくつかの注意点がある。

- 電気設備の点検やサーバの移動のために 2 台とも停止した状態から起動する場合、最後にスタンバイであったサーバだけを先に起動してはならない。両方の heartbeat を同時に起動するか、あるいは最後にアクティブであったサーバだけを先に起動する必要がある。
- 稼働していたクラスタを完全に停止する場合、スタンバイサーバを先にシャットダウンしてからアクティブサーバをシャットダウンするのが望ましい。
- 各種プログラムをアップデートする場合、スタンバイサーバ側をアップデートし、スイッチオーバーしてもう 1 台をアップデートする、という手法が利用できる(ローリング・アップグレード)。

## 13 最後に

Zabbix サーバの可用性を高めるために Linux-HA クラスタスタックを使う方法を説明した。その要点は以下ようになる。

- MySQL サーバのデータを DRBD でレプリケートする。両方のサーバのディスクが同時に故障しない限り、データを喪失することがなく、HA クラスタのフェールオーバーにも対応できる。
- Zabbix サーバの起動スクリプトを修正することによって、Zabbix サーバの親プロセスが異常終了したことも検出できて、正常な動作を維持できる。
- Heartbeat と Pacemaker を組み合わせて HA クラスタを構成した。クラスタとして管理するサービス(リソース)は、DRBD の役割(プライマリ/セカンダリ)の切り替え、ファイルシステムのマウント/アンマウント、仮想 IP アドレス、Web サーバ、MySQL サーバ、および Zabbix サーバである。