

冗長化インストール手順書

統合監視マネージャー

2023年5月 v1.04E04

このたびは、「統合監視マネージャー」をお買い上げいただき、まことにありがとうございます。

- このインストール手順書は、本体の取扱説明書と併せてよくお読みのうえ、正しく安全にお使いください。
- このインストール手順書は大切に保管してください。

目次

1. はじめに	5
2. 冗長化構成について.....	6
3. 冗長化インストールの流れ	8
4. パーティションの作成	9
4.1. FDISK コマンドを使用する場合	9
4.2. PARTED コマンドを使用する場合.....	11
5. ファイルシステムの作成.....	12
6. 設定ファイルの解凍および書き換え	12
7. CENTOS/RHEL7 の場合	23
7.1. DRBD インストーラの解凍、カーネル/DRBD のインストール.....	23
7.2. DRBD のリソース作成	24
7.3. マスター側の DRBD の開始	25
7.4. スレーブ側の DRBD の開始	26
7.5. DRBD フル同期の確認	27
7.6. マスター側の PACEMAKER のインストール	28
7.7. スレーブ側の PACEMAKER のインストール	28
7.8. PACEMAKER の設定	28
8. CENTOS/RHEL6 の場合	29
8.1. OS のバージョンとカーネルバージョンの確認	29
8.2. DRBD インストーラの解凍、カーネル/DRBD のインストール.....	29
8.3. DRBD のリソース作成	30
8.4. マスター側の DRBD の開始	31
8.5. スレーブ側の DRBD の開始	32
8.6. DRBD フル同期の確認	33
8.7. マスター側の PACEMAKER のインストール	34
8.8. スレーブ側の PACEMAKER のインストール	34
9. PACEMAKER の動作確認.....	36
10. 障害復旧手順	38
11. 監視マネージャーのライセンスについて	43

■ご使用にあたっての注意

Copyright © Panasonic Connect Co., Ltd. 2015

統合監視マネージャソフトウェア（以下、本ソフトウェア）は、以下のライセンスに基づいてライセンスされます。本ソフトウェアをご使用いただく場合は、以下に同意しなければなりません。

- 本ソフトウェアはコンピュータ 1 台に対してのみの使用とし、複数台のコンピュータで使用することはできません。
- サーバーに他のアプリケーションをインストールしないでください。正常に動作しなくなる場合があります。
- 本ソフトウェアを逆コンパイル、逆アセンブル、リバースエンジニアリング、またはその他の方法により、人間が認識できる形にすることはできません。
- 本ソフトウェアは、下記 OSS を使用しています。各 OSS が従うライセンスのライセンス文と著作権表示は、以下の URL にてご確認ください。

(1) apache-tomcat

<http://www.apache.org/licenses/LICENSE-2.0>

(2) commons-net

<http://www.apache.org/licenses/LICENSE-2.0>

(3) snmp4j

<http://www.snmp4j.org/LICENSE-2.0.txt>

(4) PostgreSQL JDBC Driver

<http://jdbc.postgresql.org/license/>

(5) Log4j

<http://www.apache.org/licenses/LICENSE-2.0>

(6) commons-codec

<http://www.apache.org/licenses/LICENSE-2.0>

(7) httpclient

<http://www.apache.org/licenses/LICENSE-2.0>

(8) commons-logging

<http://www.apache.org/licenses/LICENSE-2.0>

(9) PostgreSQL

<http://www.postgresql.org/about/license/>

(10) tftp-server

<http://opensource.org/licenses/bsd-license.php>

(11) commons-net

<http://www.apache.org/licenses/LICENSE-2.0>

(12) commons-httpclient

<http://www.apache.org/licenses/LICENSE-2.0>

(13)OpenJDK

<https://openjdk.java.net/>

(14)OpenSSL

<http://openjdk.org/legal/gplv2+ce.html>

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit

“Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved.”

“Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.”

(15)FreeRadius

<http://freeradius.org/releases/>

(16)DRBD

http://drbd.linbit.com/ja/home/drbd-is-gppld/?no_cache=1&sword_list%5B%5D=license

(17) pacemaker

<http://clusterlabs.org/developers.html>

Red Hat Enterprise Linux®は、米国 Red Hat 社の米国およびその他の国における登録商標です。

1. はじめに

- ① インストールする PC サーバー2 台（マスター／スレーブ）、および 2 台に同じバージョンの監視マネージャーがインストール済みのものを準備してください。

- ② インターネットに接続できる環境およびクライアント PC を準備ください。
推奨ハードウェアの要件を以下に示します。

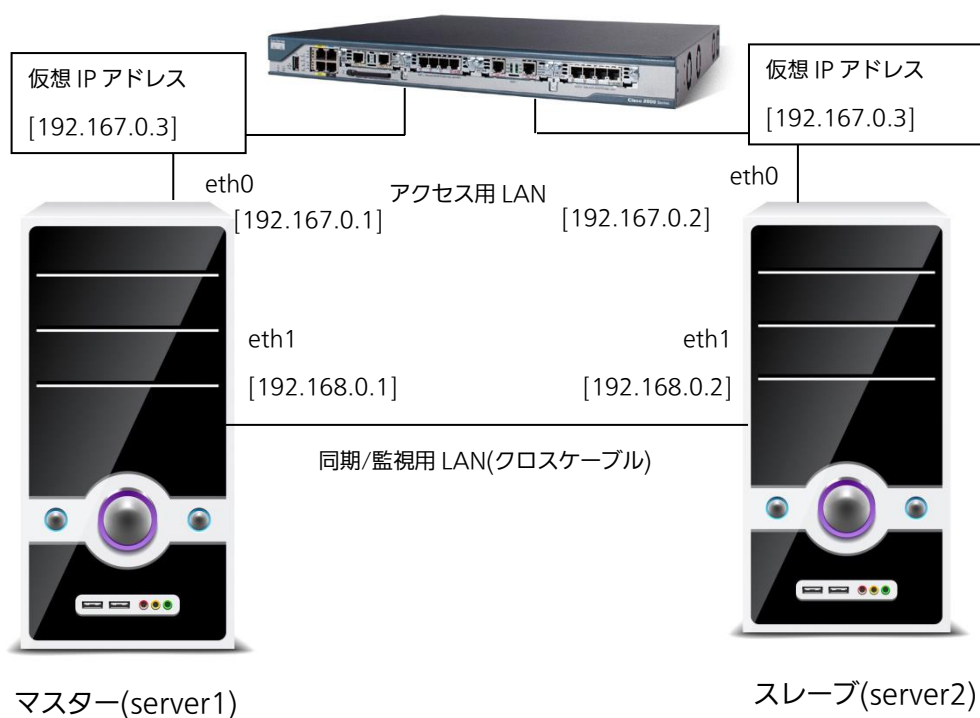
	項目	監視マネージャー	監視クライアント
1	CPU	64-bit x86 CPU 2GHz 以上	Intel 1 GHz 以上
2	メインメモリ	4 GB 以上(1000 台以下) 16GB 以上(1001 台以上)	1 GB 以上 (Windows 7, 8.1, 10) (*1)
3	HDD 容量	300 GB 以上	空き容量 200 MB 以上
4	OS	CentOS 6.9 もしくは RHEL6.9 CentOS 7.5 もしくは RHEL 7.5	Windows 7, 8.1, 10 Internet Explorer 8 ~11
5	LAN ポート	1000BASE-TX 以上 1 ポート以上 (監視マネージャーと装置が別ネットワーク で接続する場合は複数ポートを持つ必要が あります)	100BASE-TX 1 ポート

2. 冗長化構成について

以下に冗長化構成の例を示します。インストールを開始する前に、(1)または(2)のいずれかの接続方式でマスターとスレーブをネットワークに組み込んでおく必要があります。

(1) 同期/監視専用のイーサネットポートが存在する場合

イーサネットポートを2つ持ち、一方を「同期/監視用 LAN」に、もう一方を「アクセス用 LAN」に割り当てた例です。eth0 と eth1 は異なるサブネットマスクになるように、IP アドレスを決めます。

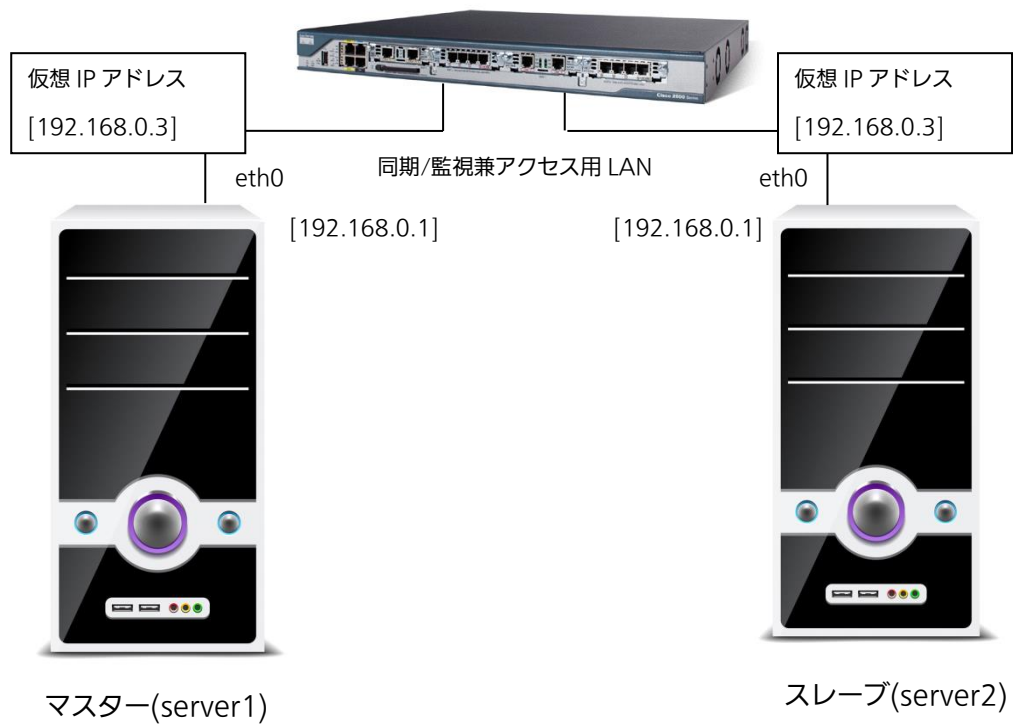


【お知らせ】

・アクセス用 LAN と同期/監視用 LAN に割り当てる IP アドレスは、異なるサブネットマスクになるよう割り当ててください。(図の例では、アクセス用 LAN が 192.167.XXX.XXX、同期/監視用 LAN が 192.168.XX.XXX にしています。)

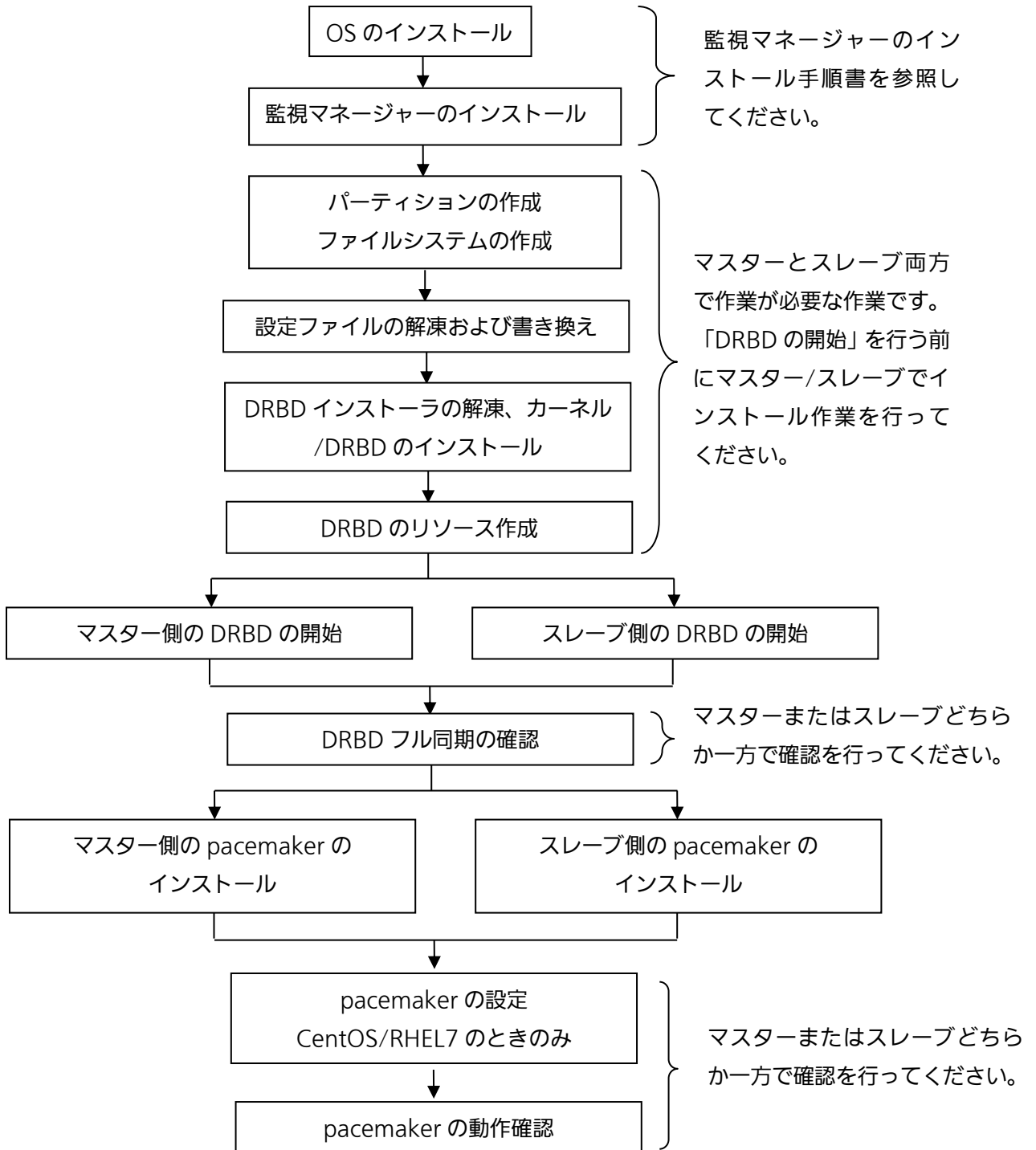
(2) イーサネットが1つしかない場合

同期/監視用の LAN とアクセス用 LAN を兼用した例です。



3. 冗長化インストールの流れ

※冗長化構成の領域割当のため、20GB 以上の未使用領域が必要になります。



4. パーティションの作成

同じバージョンの監視マネージャーがインストールされたサーバーを2台用意します。それぞれのサーバーで、使用されていない領域のパーティションを作成します。または、新しいハードディスクをご用意ください。パーティションのサイズは最低 300GB を確保してください。パーティションが作成されていない場合は、以下のコマンドで作成してください。

OS のインストーラなどで、予めパーティションが作成してある場合には、下記手順は不要です。

4.1. fdisk コマンドを使用する場合

下記のコマンドを実行して、デバイス名を確認します。

```
# fdisk -l
```

```
ディスク /dev/sda: 500.1 GB, 500107862016 バイト
ヘッド 255, セクタ 63, シリンダ 60801
Units = シリンダ数 of 16065 * 512 = 8225280 バイト
セクタサイズ (論理 / 物理): 512 バイト / 512 バイト
I/O size (minimum/optimal): 512 bytes / 512 bytes
ディスク識別子: 0x000249cc

デバイス  ブート   始点     終点     ブロック  Id システム
/dev/sda1  *         1        64       512000   83 Linux
パーティション 1 は、シリンダ境界で終わっていません。
/dev/sda2             64      34485   276480000   83 Linux
/dev/sda3          34485    34740    2048000    82 Linux スワップ
 / Solaris
```

ディスクの後に表示されている文字列「/dev/sda」がデバイス名になります。複数表示される場合は、パーティションを作成するデバイス名を使用してください。また、デバイスのところを見て、1-4 の中でどの空き番号を確認してください。上記の例では 1-3 が使用されているため、「4」が空き番号になります。

引き続き、以下のコマンドを入力して、パーティションを作成します。

```
# fdisk [デバイス名]
```

(デバイス名は、通常であれば” /dev/sda”、ハードディスクを増設した場合には” /dev/sdb” のようになります。)

「m」を入力すると、ヘルプを表示します。新たに領域を作成するには、「n」を入力します。

```
コマンド (m でヘルプ) : n          ← 「n」を入力
コマンドアクション
  e   拡張
  p   基本領域 (1-4)
```

基本領域を選択してください。「p」を入力します。領域番号が聞かれるので、1-4 までの数値を入力します。監視マネージャーと同じハードディスクの場合は、使用領域と重ならない番号を入力してください。新しいハードディスクの場合には「1」を入力します。1-3 まで既に使用されていて、「4」のみが空いている場合には、「4」が自動選択されます。

領域番号 (1-4) : 1 ← 監視マネージャーと同じハードディスクの場合には未使用領域(空き番号)の番号を入力します。

最初シリンダと終点シリンダを入力します。デフォルトで使用可能な最初シリンダになっているため、何も入力せずに、リターンキーを押します。終点シリンダはデフォルトで使用可能な終点シリンダになっているため、そのままリターンキーを押下すると、空き容量が全て割り当てられます。入力する場合は、+サイズ G または、+サイズ M、または+サイズ K で入力します。例として、20GB 分を割り当てる例を以下に示します。

最初シリンダ (1-6520, default 1) : ← 何も入力せずにリターン
Last シリンダ, +シリンダ数 or +size{K,M,G} (1-6520 default 6520) : +20G
← 20GB を指定

最後に変更内容を書き込むには、「w」を入力します。

コマンド (m でヘルプ) : w ← 「w」を入力
パーティションテーブルは変更されました！

ioctl() を呼び出して、パーティションテーブルを再読み込みします。

警告: 領域テーブルの再読み込みがエラー 16 で失敗しました: デバイスもしくはリソースがビジー状態です。
カーネルはまだ古いテーブルを使っています。新しいテーブルは
次回リブート時か、partprobe(8)または kpartx(8)を実行した後に
使えるようになるでしょう
ディスクを同期しています。

以下のコマンドを入力して、作成したパーティションができているかを確認します。

```
# fdisk -l [デバイス名]
```

下記のコマンドを実行して、OS をリブートします。

```
# reboot now
```

【お知らせ】

ファイルシステム作成は、ハードウェア環境により、数分程度かかる可能性があります。途中で強制終了などを行わないでください。

4.2. parted コマンドを使用する場合

OS 起動時に BIOS ではなく、UEFI を使用する場合は、fdisk コマンドが使用できない場合があります。その場合には、parted コマンドを使用してパーティションを作成します。

下記コマンドを入力して、parted を起動します。

```
# parted -a optimal [デバイス名]
```

(デバイス名は、通常であれば” /dev/sda”、ハードディスクを増設した場合には” /dev/sdb” のようになります。)

下記コマンドを入力して、パーティション情報を表示します。

```
(parted) print
```

以下に実行例を示します。

```
モデル: ATA MB0500GCEHE (scsi)
ディスク /dev/sda: 500GB
セクタサイズ (論理/物理): 512B/512B
パーティションテーブル: msdos

番号 開始 終了 サイズ タイプ ファイルシステム フラグ
 1 1049kB 525MB 524MB primary ext4 boot
 2 525MB 2622MB 2097MB primary linux-swaps(v1)
 3 2622MB 97.5GB 94.9GB primary ext4
```

最後の終了セクタを覚えて置きます。次のコマンドを実行して、終了セクタを開始セクタに設定し、新しいパーティションを作成します。

```
(parted) mkpart primary [開始セクタ] [終了セクタ]
```

上記の例では、開始セクタに“97.5GB”を設定します。終了セクタには、開始セクタに割り当てたい領域を足した値を設定してください。ディスク上限まで割り当てたい場合は“-1”を設定します。

以下のコマンドで、parted を終了します。

```
(parted) quit
```

以下のコマンドを入力して、作成したパーティションができているかを確認します。

```
# parted -l
```

下記のコマンドを実行して、OS をリブートします。

```
# reboot now
```

5. ファイルシステムの作成

先ほど作成したパーティションのファイルシステムを作成します。OS のインストーラなどで予めファイルシステムを作成している場合には、下記手順は不要になります。

以下のコマンドを実行します。

```
# mkfs.ext4 [パーティション名]
```

※パーティション名は通常/dev/sda1~4、新しいハードディスクの場合には、/dev/sdb1 のような名前になります。

以下のコマンドを実行して、ファイルシステムが作成されたことを確認します。

```
# parted -l
```

下記のようにファイルシステムが「ext4」と表示されていれば、ファイルシステムが作成されています。

```
# parted -l
モデル: HP LOGICAL VOLUME (scsi)
ディスク /dev/sda: 1000GB
セクタサイズ (論理/物理): 512B/512B
パーティションテーブル: msdos

番号 開始 終了 サイズ タイプ ファイルシステム フラグ
1 1049kB 525MB 524MB primary ext4 boot
2 525MB 4820MB 4295MB primary linux-swap(v1)
3 4820MB 529GB 524GB primary ext4
4 529GB 551GB 21.5GB primary ext4
```

ファイルシステムが反映されていない場合は、下記のコマンドを実行して、再起動してください。

```
# reboot now
```

6. 設定ファイルの解凍および書き換え

監視マネージャーのインストールフォルダ内で、下記コマンドを実行して、DRBD 関連ファイルを解凍して、DRBD フォルダに移動します。

```
# tar zxvf drbd_work.tar.gz
```

```
# cd drbd_work
```

下記のコマンドを実行して、設定ファイルを解凍します。

```
# tar zxvf file.tar.gz
```

解凍すると、「file」フォルダができるので、CentOS6/RHEL6 の場合は「cluster.conf」「hosts」「r0.res」ファイルを開いて、IP アドレスを環境に合わせて編集します。CentOS7/RHEL7 の場合は「corosync.conf」「corosync2.conf」「hosts」「r0.res」ファイルを開いて、IP アドレスを環境に合わせて編集します。デフォルトでは、以下の設定になっています。

```
server1 = 192.168.0.1
```

```
server2 = 192.168.0.2
```

これらの IP アドレスは、マスター/スレーブがお互いの IP アドレスを識別するのに必要な情報となります。

以下に、corosync.conf のファイルの中身を示します。corosync.conf ファイルはマスター側でのみ使用します。○で囲まれた箇所の IP アドレスをネットワーク環境に合わせて書き換えてください。(CentOS7/RHEL7 のみ)

```
# Please read the corosync.conf.5 manual page
```

```
totem {  
    version: 2  
  
    crypto_cipher: none  
    crypto_hash: none  
  
    interface {  
        member {  
            memberaddr: 192.168.0.1  
        }  
        member {  
            memberaddr: 192.168.0.2  
        }  
        ringnumber: 0  
        bindnetaddr: 192.168.0.2  
        mcastport: 5405  
        ttl: 1  
    }  
    transport: udpu  
}
```

自サーバーではなく、対向(スレーブ)の IP アドレスを設定する。

```
logging {  
    fileline: off  
    to_logfile: yes  
    to_syslog: yes  
    logfile: /var/log/cluster/corosync.log  
    debug: off  
    timestamp: on  
    logger_subsys {  
        subsys: QUORUM  
        debug: off  
    }  
}
```

```
nodelist {  
    node {  
        ring0_addr: 192.168.0.1  
        nodeid: 1  
    }  
    node {  
        ring0_addr: 192.168.0.2  
        nodeid: 2  
    }  
}
```

```
quorum {  
    # Enable and configure quorum subsystem (default: off)  
    # see also corosync.conf.5 and votequorum.5  
    provider: corosync_votequorum  
    expected_votes: 2  
}
```

スレーブ側は、corosync2.conf ファイルを使用します。以下に corosync2.conf の例を示します。
○で囲まれた箇所の IP アドレスを環境に合わせて書き換えてください。(CentOS7/RHEL7 のみ)

```
# Please read the corosync.conf.5 manual page
totem {
    version: 2

    crypto_cipher: none
    crypto_hash: none

    interface {
        member {
            memberaddr: 192.168.0.1
        }
        member {
            memberaddr: 192.168.0.2
        }
        ringnumber: 0
        bindnetaddr: 192.168.0.1
        mcastport: 5405
        ttl: 1
    }
    transport: udpu
}

logging {
    fileline: off
    to_logfile: yes
    to_syslog: yes
    logfile: /var/log/cluster/corosync.log
    debug: off
    timestamp: on
    logger_subsys {
        subsys: QUORUM
        debug: off
    }
}

nodelist {
    node {
        ring0_addr: 192.168.0.1
        nodeid: 1
    }

    node {
        ring0_addr: 192.168.0.2
        nodeid: 2
    }
}

quorum {
    # Enable and configure quorum subsystem (default: off)
    # see also corosync.conf.5 and votequorum.5
    provider: corosync_votequorum
    expected_votes: 2
}
```

自サーバーではなく、対向(マスター)の IP アドレスを設定する。

一つのサーバーで同期/監視のイーサネットがある場合は、マスター側の「corosync.conf」の代わりに「corosync_redundant.conf」を使用します。(次ページに続く)

以下に corosync_redundant.conf の例を示します。○で囲まれた IP アドレスをネットワーク環境に合わせて書き換えてください。

```
# Please read the corosync.conf.5 manual page
totem {
    version: 2
    crypto_cipher: none
    crypto_hash: none
    rrp_mode: active

    interface {
        member {
            memberaddr: 192.168.0.1
        }
        member {
            memberaddr: 192.168.0.2
        }
        ringnumber: 0
        bindnetaddr: 192.168.0.2
        mcastport: 5405
        ttl: 1
    }
    interface {
        member {
            memberaddr: 192.167.0.1
        }
        member {
            memberaddr: 192.167.0.2
        }
        ringnumber: 1
        bindnetaddr: 192.167.0.2
        mcastport: 5405
        ttl: 1
    }
    transport: udpu
}
logging {
    fileline: off
    to_logfile: yes
    to_syslog: yes
    logfile: /var/log/cluster/corosync.log
    debug: off
    timestamp: on
    logger_subsys {
        subsys: QUORUM
        debug: off
    }
}
nodelist {
    node {
        ring0_addr: 192.168.0.1
        ring1_addr: 192.167.0.1
        nodeid: 1
    }
    node {
        ring0_addr: 192.168.0.2
        ring1_addr: 192.167.0.2
        nodeid: 2
    }
}
quorum {
    # Enable and configure quorum subsystem (default: off)
    # see also corosync.conf.5 and votequorum.5
    provider: corosync_votequorum
    expected_votes: 2
}
```

同期系の IP アドレスを記載

自サーバーではなく、対向(スレーブ)の IP アドレスを設定する。

監視系の IP アドレスを記載

自サーバーではなく、対向(スレーブ)の IP アドレスを設定する。

マスター側の IP アドレス

スレーブ側の IP アドレス

一つのサーバーで同期/監視のイーサネットがある場合は、スレーブ側の「corosync2.conf」の代わりに「corosync2_redundant.conf」を使用します。以下に corosync2_redundant.conf の例を示します。○で囲まれた IP アドレスをネットワーク環境に合わせて書き換えてください。

```
# Please read the corosync.conf.5 manual page
```

```
totem {  
  version: 2  
  crypto_cipher: none  
  crypto_hash: none  
  rrp_mode: active  
  interface {
```

```
    member {  
      memberaddr: 192.168.0.1
```

同期系の IP アドレスを記載

```
    }  
    member {  
      memberaddr: 192.168.0.2
```

```
  }  
  ringnumber: 0  
  bindnetaddr: 192.168.0.1  
  mcastport: 5405  
  ttl: 1
```

自サーバーではなく、対向(マスター)の IP アドレスを設定する。

```
  }  
  interface {
```

```
    member {  
      memberaddr: 192.167.0.1
```

監視系の IP アドレスを記載

```
    }  
    member {  
      memberaddr: 192.167.0.2
```

```
  }  
  ringnumber: 0  
  bindnetaddr: 192.167.0.1  
  mcastport: 5405  
  ttl: 1
```

自サーバーではなく、対向(マスター)の IP アドレスを設定する。

```
  }  
  transport: udpu
```

```
}  
logging {
```

```
  fileline: off  
  to_logfile: yes  
  to_syslog: yes  
  logfile: /var/log/cluster/corosync.log  
  debug: off  
  timestamp: on  
  logger_subsys {  
    subsys: QUORUM  
    debug: off
```

```
}  
nodelist {
```

```
  node {  
    ring0_addr: 192.168.0.1  
    ring1_addr: 192.167.0.1  
    nodeid: 1
```

マスター側の IP アドレス

```
  }  
  node {  
    ring0_addr: 192.168.0.2  
    ring1_addr: 192.167.0.2  
    nodeid: 2
```

スレーブ側の IP アドレス

```
}  
quorum {
```

```
  # Enable and configure quorum subsystem (default: off)  
  # see also corosync.conf.5 and votequorum.5  
  provider: corosync_votequorum  
  expected_votes: 2
```

```
}
```

「corosync_redundant.conf」「corosync2_redundant.conf」を書き換えた場合には下記コマンドを使用して、corosync_redundant.conf ⇒ corosync.conf、corosync2_redundant.conf ⇒ corosync2.conf というようにファイル名を変更します。

```
# mv corosync.conf corosync.conf.bak
# mv corosync_redundant.conf corosync.conf
# mv corosync2.conf corosync2.conf.bak
# mv corosync2_redundant.conf corosync2.conf
```

以下に hosts ファイルの中身を示します。cluster.conf と同様、IP アドレスを書き換えてください。ホスト名は一つのみ設定してください。同じホスト名で複数の IP アドレスを指定した場合、最初の IP アドレスが使用されます。設定する IP アドレスは、同期/監視用 LAN の IP アドレスを設定してください。

```
127.0.0.1          localhost    localhost.localdomain    localhost4
localhost4.localdomain4
::1               localhost6  localhost6.localdomain6
192.168.0.1      server1
```

以下に r0.res ファイルの内容を示します。cluster.conf と同様に書き換えてください。ポート番号の 7789 は書き換えしないでください。IP アドレスが 2 つ以上ある場合には、同期/監視用 LAN の IP アドレスを設定してください。

```
resource r0 {
  on server1 {
    device /dev/drbd1;
    disk /dev/sda4;
    address 192.168.0.1:7789;
    meta-disk internal;
  }
  on server2 {
    device /dev/drbd1;
    disk /dev/sda4;
    address 192.168.0.2:7789;
    meta-disk internal;
  }
}
```

また、「8_pcs_config.sh」ファイルまたは「11_pcs_config.sh」ファイルを開いて、下記の仮想 IP アドレスを割り当てるイーサネットのインターフェース名および仮想 IP アドレスの値を書き換えます。下記のファイルは、マスター側のみ書き換えれば、スレーブにも同じ値が反映されます。

IP アドレスが 2 つ以上ある場合には、アクセス用 LAN のみを設定してください。同期/監視用 LAN の仮想 IP アドレスは設定する必要はありません。IP アドレスが 3 つ以上ある場合は、同期/監視用 LAN 以外の IP アドレス分の仮想 IP アドレスを登録することを推奨します。また、全ての IP アドレスは異なるサブネットになるよう、IP アドレスを設定してください。

```
# 仮想 IP アドレスの設定
pcs resource create FIP1 ocf:heartbeat:IPaddr2 nic="eno2" ip=192.168.0.3
cidr_netmask="24" op monitor interval="20s" --group rgroup
```

イーサネットのインターフェース名は「eno2」の箇所を割り当てるイーサネットのインターフェース名に書き換えます。

該当のイーサネットのインターフェース名に設定している IP アドレスの箇所を使用する仮想 IP アドレスの値に書き換えます。この IP アドレスは、マスター側のサーバーに接続できない障害が発生した場合に、スレーブ側に引き継がれます。

また、DRBD で使用するハードディスクを割り当てます。「r0.res」のファイルを開き、「4.パーティションの作成」で作成した領域を割り当てます。下記の箇所を作成したパーティションに合わせて編集してください。マスター側、スレーブ側をそれぞれ編集します。

```
disk /dev/sda4;
```

ホスト名を変更する場合は、「cluster.conf」「hosts」「r0.res」に加え、「10-01_cman_master.sh」「10-02_cman_slave.sh」を書き換えます。(CentOS/RHEL6 のときのみ)

```
<?xml version="1.0"?>
<cluster config_version="3" name="cluster1">
  <logging debug="off" syslog_facility="local5" syslog_priority="info"
to_logfile="no" to_syslog="yes"/>
  <cman expected_votes="1" two_node="1"/>
  <clusternodes>
    <clusternode name="server1" nodeid="24">
      <altname name="192.167.0.1"/>
    </clusternode>
    <clusternode name="server2" nodeid="25">
      <altname name="192.167.0.2"/>
    </clusternode>
  </clusternodes>
  <fencedevices/>
  <rm>
    <failoverdomains/>
    <resources/>
  </rm>
</cluster>
```

以下に hosts ファイルの中身を示します。cluster.conf と同様、ホスト名を書き換えてください。

```
127.0.0.1          localhost        localhost.localdomain  localhost4
localhost4.localdomain4
::1                localhost        localhost.localdomain  localhost6 localhost6.localdomain6
192.168.0.1       server1
```

以下に r0.res ファイルの内容を示します。cluster.conf と同様に書き換えてください。ポート番号の 7789 は書き換えしないでください。

```
resource r0 {
  on server1 {
    device /dev/drbd1;
    disk /dev/sda4;
    address 192.167.201.200:7789;
    meta-disk internal;
  }
  on server2 {
    device /dev/drbd1;
    disk /dev/sda4;
    address 192.167.201.201:7789;
    meta-disk internal;
  }
}
```

以下に 10-01_cman_master.sh のファイルの内容を示します。server2 のところを書き換えます。
(CentOS/RHLE6.X のときのみ)

```
#!/bin/bash

# ricci のサービス起動
passwd ricci
service ricci start 2>>err.log

# ccs による cluster.conf の同期
# server2 の ricci ユーザのパスワードを入力する
ccs -h server2 -checkconf
ccs -h server2 -sync -activate

# pacemaker サービスの起動
service pacemaker start 2>>err.log

# cman サービスの起動
# service cman start
```

以下に 10-02_cman_slave.sh のファイルの内容を示します。server1 のところを書き換えます。
(CentOS/RHEL6.X のときのみ)

```
#!/bin/bash

# ricci のサービス起動
passwd ricci
service ricci start 2>>err.log

# ccs による cluster.conf の同期
# server1 の ricci ユーザのパスワードを入力する
ccs -h server1 -checkconf
ccs -h server1 -sync -activate

# pacemaker サービスの起動
service pacemaker start 2>>err.log

# cman サービスの起動
# service cman start
```

以下に 07_hacluster7.sh の内容を示します。server1/server2 のところを書き換えます。

```
#!/bin/bash

/usr/bin/passwd hacluster -e
expect -c "
set timeout 5
spawn /usr/bin/passwd hacluster
expect \"新しいパスワード\"
send \"hacluster\n\"
expect \"新しいパスワード\"
send \"hacluster\n\"
interact
"

systemctl start pacemaker 2>>err.log
pcs cluster auth server1 server2 -u hacluster -p hacluster --force 2>>err.log
pcs cluster start --all 2>>err.log
```

本冗長化の機能では、デフォルトで障害が発生したときに、一定時間後にエラーの履歴を削除します。エラーの履歴を削除するまでの時間を変更する場合には「8_pcs_config.sh」または「11_pcs_config.sh」ファイルを開いて、下記の行を編集します。デフォルトでは、1時間(=3600秒)となっているので、下記の行を編集します。

```
pcs resource defaults failure-timeout=3600s
```

エラー履歴を自動的に削除する機能が不要な場合、値に 0 を設定します。

7. CentOS/RHEL7 の場合

以下に CentOS/RHEL7 の場合のインストール方法を示します。

7.1. DRBD インストーラの解凍、カーネル/DRBD のインストール

シェルスクリプトに実行属性を付加します。下記コマンドを実行します。

```
# chmod 755 *.sh
```

引き続き、以下のコマンドを実行し、インストールを開始します。

```
# ./install7_1.sh
```

インストールが完了すると、以下のメッセージが表示されます。

```
please reboot the server by following command
reboot now
```

以下のコマンドを入力して、リブートします。

```
# reboot now
```

7.2. DRBD のリソース作成

下記のコマンドを実行して、DRBD のリソースを作成します。

```
# drbdadm create-md r0
```

成功すると、以下のようなメッセージが表示されます。

```
# drbdadm create-md r0
Writing meta data...
initializing activity log
NOT initializing bitmap
New drbd meta data block successfully created.
```

作成したパーティションに何かのデータが書かれていた場合、コマンドが失敗することがあります。以下に失敗例を示します。

```
# drbdadm create-md r0
Writing meta data...
md_offset 8589864960
al_offset 8589832192
bm_offset 8589570048

Found ext4 filesystem
    8388544 kB data area apparently used
    8388252 kB left usable by current configuration

Device size would be truncated, which
would corrupt data and result in
'access beyond end of device' errors.
You need to either
    * use external meta data (recommended)
    * shrink that filesystem first
    * zero out the device (destroy the filesystem)
Operation refused.
```

失敗した場合は、以下のコマンドを実行して、作成したパーティションのファイルシステムを 0 埋めします。

```
# dd if=/dev/zero of=[パーティション名] bs=1M count=1
```

パーティションは、通常「/dev/sda1~4」「/dev/sdb1」のようになります。

再度、以下のコマンドを実行して、成功のメッセージが表示されたら、リソース作成の完了です。


```
# drbdadm create-md r0
```

引き続き、以下のコマンドを実行します。

```
# ./install7_2.sh
```

マスターまたはスレーブのみの実行の場合は、以下のようなメッセージが表示されます。残りのもう一方を起動するか、「yes」と入力して、リターンキーを入力してください。

```
# ./install_2.sh
.
.
Starting DRBD resources: [
    create res: r0
    prepare disk: r0
    adjust disk: r0
    adjust net: r0
]
.....
*****
***
DRBD's startup script waits for the peer node(s) to appear.
- In case this node was already a degraded cluster before the
  reboot the timeout is 0 seconds. [degr-wfc-timeout]
- If the peer was available before the reboot the timeout will
  expire after 0 seconds. [wfc-timeout]
(These values are for resource 'r0'; 0 sec -> wait forever)
```

上記の手順までは、マスター/スレーブ共通のインストールとなります。次節以降の操作を行う前に、スレーブ側での同じ操作を行って、上記までのインストールを完了させてください。

7.3. マスター側の DRBD の開始

※以下のコマンドは、マスター側のみで実行してください。

マスター側のインストールコマンドを入力します。

```
# ./install7_3_master.sh
```

7.4. スレーブ側の DRBD の開始

※以下のコマンドは、スレーブ側のみで実行してください。
スレーブ側のインストールコマンドを入力します。

```
#./install7_3_slave.sh
```

7.5. DRBD フル同期の確認

マスターとスレーブが同期して、フル同期（マスター側のデータベースの内容をスレーブにコピー）を開始します。確認のため、マスター側で下記のコマンドを入力します。

```
# cat /proc/drbd
```

以下のような表示ができれば、フル同期を開始しています。同期は、ハードディスクやネットワーク環境に依存しますが、数時間程度かかる可能性があります。画面例は、マスター側の表示になります。フル同期中は、ケーブルを抜くなどの操作をしないようにしてください。スレーブ側が正常に動作しなくなる可能性があります。

```
# cat /proc/drbd
version: 8.4.2 (api:1/proto:86-101)
GIT-hash: 7ad5f850d711223713d6dcadc3dd48860321070c build by
immpf@192.167.201.200.fis.unical.it, 2015-10-22 20:19:20

1: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent Cr--n-
ns:157312128 nr:0 dw:126160 dr:157194003 al:21 bm:9595 lo:0 pe:67
ua:26 ap:0 ep:1 wo:f oos:47576476
[======>.....] sync'ed: 00.1% (199880/199980)M
finish: 3:24:25 speed: 3,872 (4,876) K/sec
```

- ① 接続状態を示します。「SyncSource」（マスター側のフル同期中）であることを確認してください。
- ② マスター/スレーブの状態を示します。「Primary/Secondary」になっていることを確認してください。
- ③ ディスクの状態を示します。「UpToDate/Inconsistent」になっていることを確認してください。
- ④ 同期の完了率を表示します。完了率が 100%になると、フル同期が完了します。

フル同期が完了すると、以下のような表示になります。

```
# cat /proc/drbd
version: 8.4.2 (api:1/proto:86-101)
GIT-hash: 7ad5f850d711223713d6dcadc3dd48860321070c build by
immpf@192.167.201.200.fis.unical.it, 2015-10-22 20:19:20

1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate Cr-----
ns:206319800 nr:142572 dw:1708380 dr:208890542 al:60 bm:12499
lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

- ① 接続の状態を示します。「Connected」になっていることを確認してください。
- ② マスター/スレーブの状態を示します。「Primary/Secondary」になっていることを確認してください。

い。

③ディスクの状態を示します。「UpToDate/UpToDate」になっていることを確認してください。

スレーブ側で実行すると、上記コマンドの「Primary/Secondary」の部分が「Secondary/Primary」と表示されます。また、「Primary/Secondary」の部分が以下のような表示になった場合には、フル同期が開始されていません。

```
「Secondary/Secondary」
```

```
「Primary/Unknown」
```

```
「Secondary/Unknown」
```

その場合は、スレーブ側で以下のコマンドを実行してください。

```
# drbdadm down r0
```

```
# drbdadm invalidate r0
```

```
# drbdadm up r0
```

```
# drbdadm connect r0
```

引き続き、マスター側で以下のコマンドを実行します。

```
# drbdadm connect r0
```

再度、「# cat /proc/drbd」コマンドを入力して、フル同期を開始したかを確認します。

7.6. マスター側の pacemaker のインストール

DRBD フル同期中でも、pacemaker のインストールおよび動作確認は可能です。

※以下のコマンドはマスター側でのみ実行してください。

マスター側に pacemaker をインストールします。以下のコマンドを実行してください。

```
# ./install7_4.sh
```

7.7. スレーブ側の pacemaker のインストール

※以下のコマンドはスレーブ側でのみ実行してください。

スレーブ側に pacemaker をインストールします。以下のコマンドを実行してください。

```
# ./install7_4.sh
```

7.8. pacemaker の設定

※以下のコマンドはマスター側でのみ実行してください。

pacemaker の設定を行います。以下のコマンドを実行してください。

```
# ./install7_5_master.sh
```

8. CentOS/RHEL6 の場合

以下に、CentOS/RHEL6 のインストール方法を示します。

8.1. OS のバージョンとカーネルバージョンの確認

カーネルのバージョンの確認を行います。以下のファイルを開き、インストールするカーネルライブラリのバージョンを確認します。カーネルライブラリのバージョンが異なっている場合は、下記表を参考に書き換えてください。

```
# vi install_1.sh
```

OS のバージョン	カーネルライブラリのバージョン
CentOS/RHEL6.4	kernel-devel-2.6.32-358.el6.x86_64.rpm
CentOS/RHEL6.7	kernel-devel-2.6.32-573.el6.x86_64.rpm
CentOS/RHEL6.9	kernel-devel-2.6.32-696.el6.x86_64.rpm

インストールされている OS のバージョンとカーネルのバージョンが合っているかを確認します。OS のバージョンは下記のコマンドで確認することができます。

```
# cat /etc/redhat-release
```

8.2. DRBD インストーラの解凍、カーネル/DRBD のインストール

シェルスクリプトに実行属性を付加します。下記コマンドを実行します。

```
# chmod 755 *.sh
```

引き続き、以下のコマンドを実行し、インストールを開始します。

```
# ./install_1.sh
```

インストールが完了すると、以下のメッセージが表示されます。

```
please reboot the server by following command
reboot now
```

以下のコマンドを入力して、リブートします。

```
# reboot now
```

8.3. DRBD のリソース作成

下記のコマンドを実行して、DRBD のリソースを作成します。

```
# drbdadm create-md r0
```

成功すると、以下のようなメッセージが表示されます。

```
# drbdadm create-md r0
Writing meta data...
initializing activity log
NOT initializing bitmap
New drbd meta data block successfully created.
```

作成したパーティションに何かのデータが書かれていた場合、コマンドが失敗することがあります。以下に失敗例を示します。

```
# drbdadm create-md r0
Writing meta data...
md_offset 8589864960
al_offset 8589832192
bm_offset 8589570048

Found ext4 filesystem
    8388544 kB data area apparently used
    8388252 kB left usable by current configuration

Device size would be truncated, which
would corrupt data and result in
'access beyond end of device' errors.
You need to either
    * use external meta data (recommended)
    * shrink that filesystem first
    * zero out the device (destroy the filesystem)
Operation refused.
```

失敗した場合は、以下のコマンドを実行して、作成したパーティションのファイルシステムを 0 埋めします。

```
# dd if=/dev/zero of=[パーティション名] bs=1M count=1
```

パーティションは、通常「/dev/sda1~4」「/dev/sdb1」のようになります。

再度、以下のコマンドを実行して、成功のメッセージが表示されたら、リソース作成の完了です。

```
# drbdadm create-md r0
```

引き続き、以下のコマンドを実行します。

```
# ./install_2.sh
```

マスターまたはスレーブのみの実行の場合は、以下のようなメッセージが表示されます。残りのもう一方を起動するか、「yes」と入力して、リターンキーを入力してください。

```
# ./install_2.sh
.
.
Starting DRBD resources: [
    create res: r0
    prepare disk: r0
    adjust disk: r0
    adjust net: r0
]
.....
*****
***
DRBD's startup script waits for the peer node(s) to appear.
- In case this node was already a degraded cluster before the
  reboot the timeout is 0 seconds. [degr-wfc-timeout]
- If the peer was available before the reboot the timeout will
  expire after 0 seconds. [wfc-timeout]
(These values are for resource 'r0'; 0 sec -> wait forever)
```

上記の手順までは、マスター/スレーブ共通のインストールとなります。次節以降の操作を行う前に、スレーブ側での同じ操作を行って、上記までのインストールを完了させてください。

8.4. マスター側の DRBD の開始

※以下のコマンドは、マスター側のみで実行してください。

マスター側のインストールコマンドを入力します。

```
# ./install_3_master.sh
```

8.5. スレーブ側の DRBD の開始

※以下のコマンドは、スレーブ側のみで実行してください。
スレーブ側のインストールコマンドを入力します。

```
#!/install_3_slave.sh
```


8.6. DRBD フル同期の確認

マスターとスレーブが同期して、フル同期（マスター側のデータベースの内容をスレーブにコピー）を開始します。確認のため、マスター側で下記のコマンドを入力します。

```
# cat /proc/drbd
```

以下のような表示ができれば、フル同期を開始しています。同期は、ハードディスクやネットワーク環境に依存しますが、数時間程度かかる可能性があります。画面例は、マスター側の表示になります。フル同期中は、ケーブルを抜くなどの操作をしないようにしてください。スレーブ側が正常に動作しなくなる可能性があります。

```
# cat /proc/drbd
version: 8.4.2 (api:1/proto:86-101)
GIT-hash: 7ad5f850d711223713d6dcadc3dd48860321070c build by
immpf@192.167.201.200.fis.unical.it, 2015-10-22 20:19:20

1: cs: SyncSource ro: Primary/Secondary ds: UpToDate/Inconsistent C r--n-
ns:157312128 nr:0 dw:126160 dr:157194003 al:21 bm:9595 lo:0 pe:67
ua:26 ap:0 ep:1 wo:f oos:47576476
[=====>.....] sync'ed: 00.1% (199880/199980)M
finish: 3:24:25 speed: 3,872 (4,876) K/sec
```

- ①接続状態を示します。「SyncSource」（マスター側のフル同期中）であることを確認してください。
- ②マスター/スレーブの状態を示します。「Primary/Secondary」になっていることを確認してください。
- ③ディスクの状態を示します。「UpToDate/Inconsistent」になっていることを確認してください。
- ④同期の完了率を表示します。完了率が 100%になると、フル同期が完了します。

フル同期が完了すると、以下のような表示になります。

```
# cat /proc/drbd
version: 8.4.2 (api:1/proto:86-101)
GIT-hash: 7ad5f850d711223713d6dcadc3dd48860321070c build by
immpf@192.167.201.200.fis.unical.it, 2015-10-22 20:19:20

1: cs: Connected ro: Primary/Secondary ds: UpToDate/UpToDate C r----
ns:206319800 nr:142572 dw:1708380 dr:208890542 al:60 bm:12499
lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

- ①接続の状態を示します。「Connected」になっていることを確認してください。
- ②マスター/スレーブの状態を示します。「Primary/Secondary」になっていることを確認してください。

い。

③ディスクの状態を示します。「UpToDate/UpToDate」になっていることを確認してください。

スレーブ側で実行すると、上記コマンドの「Primary/Secondary」の部分が「Secondary/Primary」と表示されます。また、「Primary/Secondary」の部分が以下のような表示になった場合には、フル同期が開始されていません。

```
「Secondary/Secondary」
```

```
「Primary/Unknown」
```

```
「Secondary/Unknown」
```

その場合は、スレーブ側で以下のコマンドを実行してください。

```
# drbdadm down r0
```

```
# drbdadm invalidate r0
```

```
# drbdadm up r0
```

```
# drbdadm connect r0
```

引き続き、マスター側で以下のコマンドを実行します。

```
# drbdadm connect r0
```

再度、「# cat /proc/drbd」コマンドを入力して、フル同期を開始したかを確認します。

8.7. マスター側の pacemaker のインストール

DRBD フル同期中でも、pacemaker のインストールおよび動作確認は可能です。

※以下のコマンドはマスター側でのみ実行してください。

マスター側に pacemaker をインストールします。以下のコマンドを実行してください。

```
# ./install_4_master.sh
```

途中で、マスター側のパスワード設定の入力を求められます。また、スレーブ側に先に pacemaker をインストールしていた場合、スレーブ側のパスワード入力が必要になります。スレーブ側のパスワード入力は、次節の「スレーブ側の pacemaker のインストール」で設定するパスワードを入力してください。

8.8. スレーブ側の pacemaker のインストール

※以下のコマンドはスレーブ側でのみ実行してください。

スレーブ側に pacemaker をインストールします。以下のコマンドを実行してください。

```
# ./install_4_slave.sh
```

途中側で、スレーブ側のパスワード設定の入力が求められます。また、マスター側に先に pacemaker をインストールしていた場合、マスター側のパスワード入力が必要になります。マスター側のパスワード入力は、次節の「マスター側の pacemaker のインストール」で設定するパスワード

ードを入力してください。

9. pacemaker の動作確認

マスター側またはスレーブ側で以下のコマンドを実行します。

```
# pcs status
```

以下のような表示がされます。

```
# pcs status
Cluster name: cluster1
Last updated: Fri Nov  6 21:14:06 2015
Last change: Fri Nov  6 11:50:35 2015
Stack: cman
Current DC: server2 - partition with quorum
Version: 1.1.11-97629de
2 Nodes configured
6 Resources configured

Online: [ server1 server2 ]

Full list of resources:

Master/Slave Set: drbd-clone [drbd]
Masters: [ server1 ]
Slaves: [ server2 ] ①
Resource Group: rgroup
② FIP1 (ocf::heartbeat:IPAddr2): Started server1
cluster (ocf::heartbeat:Filesystem): Started server1
Posgres (ocf::heartbeat:pgsql): Started server1 ③
Tomcat (ocf::heartbeat:tomcat): Started server1
Failed actions: ④
cluster_start_0 on server2 'unknown error' (1): call=455, status=complete,
last-rc-change='Fri Nov  6 11:42:15 2015', queued=0ms, exec=641ms
```

以下に確認ポイントを示します。

① DRBD の同期 4

DRBD 同期の監視状態を確認します。「Masters」「Slaves」の文字が表示されていることを確認します。DRBD 同期の監視が正常に動作していない場合は、一方の文字列、または両方が「Stopped」

のような表示になります。その場合は、以下のコマンドを実行してください。

```
# pcs cluster unstandby --all
```

再度「pcs status」を実行して、DRBDの同期が正常に監視されていることを確認します。

②リソース名

リソース名を表示します。「FIP1」「cluster」「Posgres」「Tomcat」の4つが表示されていることを確認します。

③リソース起動状態

全リソースがマスター側で起動していることを確認します。全リソースで「Started server1」と表示されていれば、マスター側が正常に起動しています。スレーブ側が起動した状態だと「Started server2」という表示になります。

④Failed actions

pacemakerの監視が正常な状態の場合、このエリアは表示されなくなります。何かのリソースが正常に起動していない場合にのみ表示されます。エラーメッセージの「cluster_start_0 on server2」という箇所から、「リソース名『cluster』がserver2(スレーブ側)で起動に失敗した」ということを示します。

リソースに問題がない場合は、下記のコマンドを実行して、エラー履歴を削除します。

```
# pcs resource cleanup [リソース名]
```

上記の例では、[リソース名]のところには「cluster」を入力します。再度コマンド「pcs status」を実行して、Failed actionsの表示が消えることを確認します。

再度、リソースに異常がある場合は、リソースに何かしらの原因があることが考えられます。ネットワークやハードウェアの状態を確認してください。それでも復旧しない場合は、再度、インストール手順を確認してください。

10. 障害復旧手順

マスター、またはスレーブで障害が発生し、その後障害が回復しても、自動的に冗長構成が復旧しない場合があります。冗長構成が正常に動作していることは、以下のように確認します。

①DRBD 状態の確認

②pacemaker 状態の確認

以下に、確認手順を示します。

(1) DRBD 状態の確認

下記のコマンドを実行して、DRBD の状態を確認します。

cat /proc/drbd

```
# cat /proc/drbd
version: 8.4.2 (api:1/proto:86-101)
GIT-hash: 7ad5f850d711223713d6dcadc3dd48860321070c build by
immpf@192.167.201.200.fis.unical.it, 2015-10-22 20:19:20

1: cs(①)Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
   ns:206319800 nr:142572 dw:1708380 dr:208890542 al:60 bm:12499
   lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

①接続状態

正常に動作している場合、「Connected」と表示されます。正常に動作していない場合は、主に以下のような表示がされます。

表示文字	概要
StandAlone	マスター/スレーブが正常に認識できていない状態です。まだ接続が確立されていない、スプリットブレインにより、接続が解除されたなどの理由が考えられます。
Disconnecting	切断中の一時的な状態です。次の状態は StandAlone になります。
Unconnected	接続を試行するまえの一時的な状態です。
WFConnection	対向ノードがネットワーク上で通信できるまで、待機している状態です。
SyncSource	ローカルノードを同期元(マスター)にして、フル同期を実行している状態です。
SyncTarget	ローカルノードを同期先(スレーブ)にして、フル同期を実行している状態です。
PausedSyncS	同期元のフル同期が中断している状態です。
PausedSyncT	同期先のフル同期が中断している状態です。

WFBitMapS	同期元（マスター）が部分同期を開始しようとしている状態です。
WFBitMapT	同期先（スレーブ）が部分同期を開始しようとしている状態です。
WFSyncUUID	同期を開始しようとしている状態です。

②マスター/スレーブの状態

正常な状態の場合は「Primary/Secondary」（マスター側）、「Secondary/Primary」（スレーブ側）と表示されます。正常に動作していない場合は、以下のように表示されます。

表示文字	概要
Primary/Unknown	マスター側からスレーブ側が監視できない状態です。ケーブルの接続などを確認してください。
Secondary/Unknown	スレーブ側からマスター側を監視できていない状態です。ケーブルの接続を確認してください。
Secondary/Secondary	両方ともスレーブと認識している状態です。後述のコマンドで、どちらか一方を Primary(マスター)に昇格させる必要があります。

③ディスクの状態

正常に同期され、マスター/スレーブのデータが一致している場合は、「UpToDate/UpToDate」と表示されます。上記以外の状態では、以下のような表示がされます。

表示文字	概要
Inconsistent	データが一致していない状態です。初期フル同期の前には両方のノードがこの状態になります。フル同期中は、片方（スレーブ）がこの状態になります。
Outdated	リソースデータは一致しているが、無効な状態です。
Dunknown	ネットワークが接続されていない場合、対向ノードに表示されます。
Consistent	接続していない状態で、データが一致している状態です。接続が確立すると、UpToDate か Outdated になります。

正常に動作していない場合には、どちらのサーバーをマスター側に、どちらのサーバーをスレーブ側にするかを決めます。必要に応じて、以下のコマンドを実行してください。なお、スレーブ側に指定したデータベースの内容は全て削除され、マスター側のデータベースの内容が同期されます。マスターをスレーブに降格させたい場合には、以下のコマンドを実行します。

```
# drbdadm secondary r0
```

スレーブをマスターに昇格させたい場合には、以下のコマンドを実行します。

```
# drbdadm primary r0
```

マスター側のディスクの内容をスレーブ側にフル同期させます。スレーブ側で以下のコマンドを実行します。

```
# drbdadm down r0
```

```
# drbdadm invalidate r0
```

```
# drbdadm up r0
```

```
# drbdadm connect r0
```

引き続き、マスター側で以下のコマンドを実行します。

```
# drbdadm connect r0
```

DRBD の同期状態を確認するために、以下のコマンドを実行します。下記のような表示が出れば、DRBD のフル同期が開始されたことを示します。

```
# cat /proc/drbd
version: 8.4.2 (api:1/proto:86-101)
GIT-hash: 7ad5f850d711223713d6dcadc3dd48860321070c build by
immpf@192.167.201.200.fis.unical.it, 2015-10-22 20:19:20

1: cs:SyncSource ro Primary/Secondary ds:UpToDate/Inconsistent C r--n-
ns:157312128 nr:0 dw:126160 dr:157194003 al:21 bm:9595 lo:0 pe:67
ua:26 ap:0 ep:1 wo:f oos:47576476
[=====>.....] sync'ed: 00.1% (199880/199980)M
finish: 3:24:25 speed: 3,872 (4,876) K/sec
```

「cat /proc/drbd」を実行しても、何も表示されない場合は、DRBD が起動していない可能性があります。DRBD が正常に起動しているか調べるために、以下のコマンドを実行します。

```
# service drbd status
```

起動していない場合には、以下のコマンドを実行して起動します。

```
# service drbd start
```

【お知らせ】

次ページ以降の pcs コマンドによる設定は、フル同期終了後に行ってください。状態を確認する「pcs status」コマンドは実行しても問題ありません。

(2) pacemaker 状態の確認

Pacemaker の状態を確認するために、以下のコマンドを実行します。

```
# pcs status
```

正常な状態の場合、以下のような表示がされます。

```
# pcs status
Cluster name: cluster1
Last updated: Fri Nov  6 21:14:06 2015
Last change: Fri Nov  6 11:50:35 2015
Stack: cman
Current DC: server2 - partition with quorum
Version: 1.1.11-97629de
2 Nodes configured
6 Resources configured

Online: [ server1 server2 ]

Full list of resources:

Master/Slave Set: drbd-clone [drbd]
Masters: [ server1 ]
Slaves: [ server2 ]

Resource Group: rgroup
FIP1 (ocf::heartbeat:IPaddr2): Started server1
cluster (ocf::heartbeat:Filesystem): Started server1
Posgres (ocf::heartbeat:pgsql): Started server1
Tomcat (ocf::heartbeat:tomcat): Started server1
```

①DRBD の監視状態

正常な場合、「Masters: [server1]」「Slaves: [server2]」と表示されます。「Slaves : [Server2]」の箇所が「Stopped : [server2]」と表示される場合は、スレーブが正常に動作していない状態です。その場合は、以下のコマンドを入力します。

```
# pcs cluster unstandby --all
```

再度、「pcs status」コマンドを実行して、pacemaker の状態を確認します。

マスター側のリソース「FIP1」「cluster」「Posgres」「Tomcat」全てが「Started server1」（スレーブがマスターになっている場合は「Started server2」）と表示されます。リソースの機能が停止

している場合は「Stopped」と表示されます。その場合には下に「Failed actions」という表示も同時に表示されます。その場合、以下のコマンドを入力して、リソースのエラーをクリアしてください。

```
# pcs resource cleanup [リソース名]
```

上記コマンドを入力しなくても、エラーの原因が復旧していれば、一定時間後（デフォルトでは1時間）に復旧します。

リソースに問題がない場合は、エラーがクリアされ、正常にリソースが起動します。

また、表示が正常に表示されていた場合でも、スレーブのリソースが停止している場合があります。その場合には下部に、以下のような「Failed actions」のような表示がされます。

```
Failed actions:
cluster_start_0 on server2 'unknown error' (1): call=455, status=complete,
last-rc-change='Fri Nov 6 11:42:15 2015', queued=0ms, exec=641ms
```

上記のような場合も「pcs resource cleanup [リソース名]」（上記の場合、リソース名は「cluster」になります）のコマンドを入力し、「pcs status」で状態を確認します。

下記のように、「drbd-monitor」が表示された場合は、リソース名として「drbd-clone」を指定して、リソースのクリアを行います。

```
Failed actions:
drbd_monito_20000 on server2 'not running' (7): call=280,
status=complete, last-rc-change='Wed Nov 18 15:42:04 2015', queued=0ms,
```

マスター/スレーブを切り替えるには、以下のコマンドを実行します。

```
# pcs cluster standby [スレーブに降格させたいホスト名] (server1 または server2)
```

このコマンド実行後、30 秒程度待ってから、以下のコマンドを実行してください。または「pcs status」コマンドを実行して、マスター/スレーブが切り替わっていることを確認してください。

```
# pcs cluster unstandby --all
```

「pcs status」で確認すると、マスター/スレーブが入れ替わっていることが確認できます。

スプリットブレインからの復旧時、「pcs status」コマンドを実行して、以下のエラーメッセージが表示されることがあります。

```
Error: cluster is not currently running on this node
```

pacemaker が起動されていません。下記コマンドを実行して、pacemaker を起動してください。

```
# service pacemaker start
```

Pacemaker 起動後、再度「pcs status」コマンドを実行してください。

【お知らせ】

DRBD がフル同期中は、マスター/スレーブの切り替え操作を行わないでください。スレーブの起動処理が正常に行われません。

11. 監視マネージャーのライセンスについて

監視マネージャーのライセンスは、冗長化構成の場合、マスター側からスレーブ側のライセンスも入力することができます。マスター側のライセンスは、そのまま入力することができますが、スレーブ側のライセンスを入力する場合は、ライセンスキーに加え、スレーブ側の MAC アドレスの情報が必要になります。必要に応じて、予めメモなどに記載してください。

■使い方・お手入れ・修理などは、まずお買い求め先へご相談ください。

■その他ご不明な点は下記へご相談ください。

パナソニック システムお客様ご相談センター

電話

フリー
ダイヤル



パナハ ヨイワ
0120-878-410

携帯電話・PHS からもご利用いただけます。

受付時間：9時～17時30分（土・日・祝祭日は受付のみ）

ホームページからのお問い合わせは

<https://panasonic.biz/cns/cs/cntctus/index.html>

ご使用の回線（IP電話やひかり電話など）によっては、回線の混雑時に数分で切れる場合があります。

【ご相談窓口におけるお客様の個人情報のお取り扱いについて】

パナソニック株式会社およびグループ関係会社は、お客様の個人情報をご相談対応や修理対応などに利用させていただき、ご相談内容は録音させていただきます。また、折り返し電話をさせていただくための発信番号を通知いただいております。なお、個人情報を適切に管理し、修理業務等を委託する場合や正当な理由がある場合を除き、第三者に開示・提供いたしません。

個人情報に関するお問い合わせは、ご相談いただきました窓口にご連絡ください。

パナソニック コネクト株式会社

〒224-8539 神奈川県横浜市都筑区佐江戸町 600 番地