

drbd.conf

名前

drbd.conf — DRBD によるデバイス制御の設定ファイル

紹介

/etc/drbd.conf ファイルは drbdadm が読み込む設定ファイルである。

本ファイルの形式は、クラスタを構成する 2 つのノード間で機械的にコピーしても問題が生じないようにデザインされている。実際に、設定を管理できるためには、一方のノードで作成したファイルを他ノードにコピーすることを強く推奨する。/etc/drbd.conf は、クラスタを構成する 2 つのノードで同じ内容でなければならない。本ファイルに施した変更は、ただちに反映されるものではない。

例 1. 簡単な drbd.conf の例

```
global { usage-count yes; }
common { syncer { rate 10M; } }
resource r0 {
  protocol C;
  net {
    cram-hmac-alg sha1;
    shared-secret "FooFunFactory";
  }
  on thost1 {
    device /dev/drbd1;
    disk /dev/hda7;
    address 10.1.1.31:7789;
    meta-disk internal;
  }
  on thost2 {
    device /dev/drbd1;
    disk /dev/hda7;
    address 10.1.1.32:7789;
    meta-disk internal;
  }
}
```

上の例では、**r0** という名前の 1 つの DRBD リソースを設定している。ノード間の通信プロトコルは **C** である。ホスト *thost1* 側ではこのリソースにアクセスするために */dev/drbd1* というデバイス名を使う。このリソースのデータを格納する低レベルのデバイスは */dev/hda7* である。IP アドレスは、2 つのノード間の通信に使うネットワークインタフェースのアドレスである。再同期が行われるとき、ネットワーク帯域のうち最大 10 メガバイト/秒の帯域を使うことができる。

1 つの drbd.conf ファイル中に複数のリソースセクションを定義してもよい。 *DRBD Quick start Guide* (<http://www.linux-ha.org/DRBD/QuickStart07>) にいくつかの設定例を掲載しているので、参照のこと。

ファイル形式

本ファイルはセクションとパラメータで構成される。セクションは、キーワード、任意指定のパラメータ、開き中括弧 (“{”) で始まる。閉じ中括弧でセクションが終わる。中括弧はパラメータを囲むのにも使われる。

```
section [name] { parameter value; [...] }
```

パラメータ名と値の間にホワイトスペースが必要である。パラメータ名の後ろの文字列はパラメータに対する値と見なされる。ブールパラメータは特別なパラメータで、パラメータ名だけで構成される。パラメータの最後にセミコロン (“;”) が必要である。

いくつかのパラメータ値はデフォルトの単位を持つが、**K**、**M**、または**G**を明示的に指定することによって単位を変えられる。これらの単位はコンピュータでおなじみの方法で定義される (**K** = 2^4 = 1024、**M** = 1024 **K**、**G** = 1024 **M**)。

設定ファイルにコメントを記述できる。コメント行はハッシュ記号 (“#”) で始まらなければならない。この文字以降行末までの文字がコメントと見なされる。

セクション

skip

このセクション内のテキストは、複数行にわたってすべてコメントと見なされる。キーワード skip と開き中括弧 (“{”) の間の文字も無視される。このセクションは、定義済みの他のセクション全体を無効にするときに便利である。たとえば `resource [name] {...}` というセクション全体を skip セクションで囲むことができる。

global

いくつかの全般的なパラメータを記述する。現在 minor-count、dialog-refresh、disable-ip-verification、および usage-count パラメータが許されている。global セクションは 1 回だけ記述でき、設定ファイルの先頭を書くのが望ましい。

common

common セクションに記述したパラメータはすべてのリソースに継承される。このセクションには、startup、syncer、handlers、net および disk セクションを書くことができる。

resource name

DRBD リソースを定義するセクションである。各リソースセクションは、2つの on host セクションを持つ必要があり、必要に応じて startup、syncer、handlers、net および disk セクションを書くことができる。このセクションに指定しなければならない必須パラメータは protocol である。

on host-name

リソースを構成する DRBD デバイスを定義するためのセクションで、必須である。host-name は省略できない。また、ノードの inux ホスト名 (uname -n) でなければならない。このセクションには、device、disk、address、meta-disk および flexible-meta-disk パラメータが必須である。

disk

このセクションは、下位ストレージに対する DRBD の取り扱いをチューニングするパラメータで構成される。各パラメータの詳細は `drbdsetup(8)` を参照。指定できるパラメータは次のとおり: `on-io-error`、`size`、`fencing`、`use-bmbv`。

net

このセクションは、ネットワークに対する DRBD の取り扱いをチューニングするパラメータで構成される。各パラメータの詳細は `drbdsetup(8)` を参照。指定できるパラメータは次のとおり: `sndbuf-size`、`timeout`、`connect-int`、`ping-int`、`ping-timeout`、`max-buffers`、`max-epoch-size`、`ko-count`、`allow-two-primaries`、`cram-hmac-alg`、`shared-secret`、`after-sb-0pri`、`after-sb-1pri`、`after-sb-2pri`、`data-integrity-alg`。

startup

このセクションは DRBD の振る舞いをチューニングするパラメータを含む。各パラメータの詳細は `drbdsetup(8)` を参照。指定できるパラメータは次のとおり: `wfc-timeout`、`degr-wfc-timeout`、`wait-after-sb`、`become-primary-on`。

syncer

このセクションは同期デモンの振る舞いをチューニングするパラメータを含む。各パラメータの詳細は `drbdsetup(8)` を参照。指定できるパラメータは次のとおり: `rate`、`after`、`al-extents`、`cpu-mask`。

handlers

このセクションは、DRBD が特定のイベントに応じて実行するハンドラ (実行可能プログラム) を登録する。指定できるパラメータは次のとおり。 `pri-on-incon-degr`、`pri-lost-after-sb`、`pri-lost`、`outdate-peer`、`local-io-error`、`split-brain`。

パラメータ

`minor-count` *count*

count には 1 から 255 の値を指定できる。

DRBD カーネルモジュールをロードしなおすことなく多数のリソースを定義したい場合、*minor-count* を指定する。デフォルトでは、現在定義されているリソースの数に加えてさらに 11 個のリソースを定義できる。ただし最低でも 32 個は定義できる。

`dialog-refresh` *time*

time には 0 または正の整数を指定する。

ダイアログを再描画する時間間隔を *time* 秒に設定する。ゼロを指定すると再描画は行われない。デフォルト値は 1 である。

`disable-ip-verification`

何らかの理由で `drbdadm` が `ip` または `ifconfig` コマンドを使って IP アドレスの正常さを検証できない場合、`disable-ip-verification` を指定する。このオプションを指定することによって、IP アドレスを検証しないようにできる。

`usage-count val`

DRBD の利用者統計 *DRBD's online usage counter* (<http://usage.drbd.org>) に参加するには、このオプションに `yes` を指定する。指定できる値は、`yes`、`no`、または `ask` である。

`protocol prot-id`

TCP/IP の通信の際に使われる DRBD のプロトコルを指定する。`prot-id` には A、B または C を指定する。

プロトコル A: ローカルディスクとローカル TCP 送信バッファにデータを書き込んだら ディスクへの書き込みが完了したと判断する。

プロトコル B: ローカルディスクとリモートバッファキャッシュにデータを書き込んだら ディスクへの書き込みが完了したと判断する。

プロトコル C: ローカルディスクとリモートディスクの両方にデータを書き込んだら ディスクへの書き込みが完了したと判断する。

`pri-on-incon-degr-cmd command`

損傷モード (`degr-wfc-timeout` フラグがセットされている) のノードが起動して、かつそのローカルディスクのデータが不整合状態になっている場合、このオプションに指定した `command` が実行される。コマンドが正常に終了すると、(heartbeat の) `drbddisk` はこの DRBD デバイスをプライマリモードにしようとする。

`device name`

定義している DRBD リソースに対応するブロックデバイス名を指定する。アプリケーション (通常はファイルシステム) では、ここで指定する名前をデバイス名として指定する。逆に、`disk` に指定したデバイス名を決して指定してはならない。

すべてのデバイスのメジャー番号は、DRBD ドライバに対応した値を持っていないなければならない。現バージョンのメジャー番号は 147 で、対応するデバイス名は通常 `/dev/drbd0`、`/dev/drbd1` などである (drbd-0.7.1 以前でのメジャー番号は 43、デバイス名は `/dev/nb*` であった)。

DRBD パッケージのインストールスクリプトは、`/dev/drbd0` から `/dev/drbd8` があらかじめ定義されていることを要求する。登録されていることは、たとえば `ls /dev/drbd*` を実行することで確認できる。

`disk name`

ブロックデバイス名を指定する。DRBD はこのデバイスを対象にデータを読み書きする。DRBD 動作中は、このデバイスを絶対に別の方法でアクセスしてはならない。この禁止事項には、`dumpe2fs(8)` やその他の類似コマンドも含まれる。

```
address IP:port
```

各リソースごとに、デバイスに対応した IP アドレスを指定する。DRBD は、この IP アドレスで他ノードのデバイスとの通信を行う。

さらに各 TCP ポート番号も指定する必要がある。複数の DRBD リソースで同一の IP:port を共用することはできない。

```
meta-disk internal
flexible-meta-disk internal
meta-disk device [index]
flexible-meta-disk device
```

内部 (**internal**) を指定すると、下位デバイスの最後の部分にメタデータが作られる。内部メタデータの場合、`[index]` は指定できない。meta-disk、flexible-meta-disk のどちらを指定した場合も、デバイスの残りの部分に見合った必要なサイズの内部メタデータ領域が確保される。

`device` を指定すると、そのデバイスに DRBD デバイスに対応したメタデータがそのデバイスに書き込まれる。たとえば、2つのリソースに対して、それぞれ `meta-disk /dev/hde6[0]`、および `meta-disk /dev/hde6[1]` を指定できる。この例の場合、少なくとも 256MB 以上のメタディスク領域が必要である。

flexible-meta-disk キーワードにブロックデバイス名を指定できる。このキーワードは通常 LVM と組み合わせて多数の異なるサイズのブロックデバイスを扱うときに指定する。メタディスクに必要なサイズは、 $36\text{kB} + \text{Backing-Storage-size} / 32\text{k}$ となり、実際のサイズはこの値を 4 キロバイト単位で切り上げたものになる。おおまかな目安を得るには、ストレージサイズ 1 ギガバイトごとにに対して 32 キロバイト、これをメガバイト単位で切り上げる。

```
on-io-error handler
```

下位デバイスが入出力エラーを返したときに実行するハンドラを指定する。

`handler` に指定できる値は、`pass_on`、`call-local-io-error`、または `detach` のどれかである。

`pass_on`: 入出力エラーを上位レイヤに報告する。プライマリ側ではファイルシステムにエラーを伝える。セカンダリ側では何も報告しない。

`call-local-io-error`: ハンドラスクリプト `local-io-error` を呼び出して実行する。

`detach`: 低レベルデバイスを切り離して、ディスクレスモードで処理を続行する。

```
fencing fencing_policy
```

フェンシングは、2つのノードがともにプライマリになること (スプリットブレイン) を防止し、どちらかを切り離す防御手段のことである。

次のフェンシングポリシーを指定できる。

```
dont-care
```

デフォルトの設定値で、フェンシングのためのアクションを実行しない。

`resource-only`

ノードが切り離されたプライマリ状態になると、DRBD は、DRBD は `outdate-peer` ハンドラを実行して他ノードを期限切れ状態に変えようとする。`outdate` ハンドラは、他ノード上にネットワーク経由で接続し、`'drbdadm outdate res'` を実行しようとする。

`resource-and-stonith`

ノードが切り離されたプライマリ状態になると、DRBD はすべてのディスク I/O を停止して `outdate-peer` ハンドラを呼び出す。`outdate` ハンドラは、他ノード上にネットワーク経由で接続し、`'drbdadm outdate res'` を実行しようとする。ハンドラが他ノードに到達できない場合、DRBD は STONITH 機能を使って他ノードを強制排除する。これらが完了したら、ディスク I/O を再開する。ハンドラの実行が失敗した場合、`drbdsetup` の `resume-io` コマンドを使ってディスク I/O を復旧させることができる。

`use-bmbv`

下位レベルのストレージドライバが `merge_bvec_fn()` 関数を備えている場合、DRBD は 4 キロバイトを越えない単位でのディスク I/O リクエストだけを処理する。本マニュアル執筆時点では、この機能を備えているドライバは、`md` (ソフトウェア RAID)、`dm` (デバイスマップ - LVM)、および DRBD 自身だけである。

ソフトウェア RAID やその他の `merge_bvec_fn()` 関数を持つドライバの上位で DRBD を使う場合であって、全部のクラスタ構成ノードで同関数が同様に振る舞うことがわかっている場合 (すなわちソフトウェア RAID などを構成する物理的なディスクが同タイプである場合)、最高のパフォーマンスを得るにはこのパラメータを設定すべきである。このオプションは、何をしているのかを理解した上でのみ使うように。

`sndbuf-size size`

TCP ソケットの送信バッファサイズを `size` に指定する。デフォルト値は 128K である。これより小さい値も大きい値も指定できる。プロトコル A と高速ネットワークを使う場合に大きな値を指定すると書き込みスループットを向上できる。しかし極端に大きな値、たとえば 1M を指定すると、問題が生じる可能性がある。また、32K より小さい値は現実的ではない。

`timeout time`

対向ノードからの応答パケットが $1/10$ 秒の `time` 倍の時間以内に返ってこない場合、対向ノードが死んだと判断して、TCP/IP コネクションを切断する。この値は `connect-int` 値および `ping-int` よりも小さくなければならない。デフォルト値は 60 で、これは 6 秒に相当する。すなわちこのパラメータの単位は 0.1 秒である。

`connect-int time`

対向ノードにただちに接続できない場合、DRBD は接続を繰り返し試行する。このパラメータは試行間隔を指定する。デフォルト値は 10 で、このパラメータの単位は秒である。

ping-int *time*

2つの DRBD ドライバ間の接続が確立していて、*time* 秒の間に何も通信が行われなかった場合、DRBD は対向ノードが生きているか確認するためのパケットを送出する。デフォルト値は 10 で、このパラメータの単位は秒である。

ping-timeout *time*

ping-int パケットに対して対向ノードはこのパラメータに指定した時間以内に応答しなければならない。応答パケットが返ってこない場合、その対向ノードは死んだと判断される。デフォルト値は 500ms で、100ms 単位で指定する。

max-buffers *number*

DRBD に割り当てるバッファの最大数を指定する。単位は PAGE_SIZE で、ほとんどのシステムで 4kB である。最小値は 32 (=128 KB) にハードコードされている。良好な環境の場合、このパラメータを大きく設定することによって良好な結果が期待できる。このバッファは、ディスクに書き込み中のデータブロックを保持するのに使われる。

ko-count *number*

セカンダリノードが書き込みリクエストを *count* 回以上失敗した場合、そのセカンダリノードはクラスタから排除され、プライマリノードは StandAlone モードに遷移する。デフォルト値は 0 で、これは本機能が無効になることを意味する。

max-epoch-size *number*

書き込みバリア間に処理するデータブロックの最大数を指定する。10 未満の値を指定するとパフォーマンスが低下することがある。

allow-two-primaries

このオプションを指定すると、両ノードがともにプライマリになれる。このオプションは、DRBD の上位に分散ファイルシステムを使っている場合のみ指定すべきである。本マニュアル執筆時点でサポートされるファイルシステムは OCFS2 と GFS のみである。他のファイルシステム利用時にこのオプションを指定すると、ノードとデータを破壊する。

unplug-watermark *number*

セカンダリノード上に書き込まれていない書き込みリクエスト数がこの値を上回ると、下位レベルのデバイスに対して書き込みリクエストを送る。ストレージによっては小さい値でも良好な結果が得られるが、多くのデバイスでは max-buffers と同じ値を指定するときに最良の結果が得られる。デフォルト値は 128 で、指定できる最小値は 16、最大値は 131072 である。

cram-hmac-alg

対向ノードの認証を行いたい場合、HMAC アルゴリズムを指定する。対向ノードの認証は行うべきである。チャレンジ-レスポンス方式で対向ノードを認証するのに、HMAC アルゴリズムが使われる。/proc/crypto に記録されている任意のダイジェストアルゴリズムを指定できる。

shared-secret

64 文字までの共有秘密鍵を指定する。

after-sb-0pri policy

スプリットブレイン状態が生じて両ノードが同時にプライマリになった後で、両ノードの接続が回復した場合に行われる修復方法を指定する。次のポリシーを指定できる。

disconnect

自動再同期を行わず接続を切断する。

discard-younger-primary

スプリットブレイン発生前にプライマリであったノードからの再同期を自動的に実行する。

discard-older-primary

スプリットブレイン発生時にプライマリになったノードからの再同期を自動的に実行する。

discard-zero-changes

スプリットブレイン発生後 どちらか一方のノードに書き込みがまったく行われなかったことが明白な場合、書き込みが行われたノードから行われなかったノードに対する再同期が実行される。どちらも書き込まれなかった場合は、DRBDはランダムな判断によって0ブロックの再同期を実行する。両ノードに書き込みが行われた場合、このポリシーはノードの接続を切断する。

discard-least-changes

スプリットブレイン発生後、より多くのブロックを書き込んだノードから他方に対する再同期を実行する。

discard-node-NODENAME

指定した名前のノードに対する再同期を実行する。

after-sb-1pri policy

スプリットブレイン発生後どちらか一方のノードがプライマリになっている場合、このパラメータのポリシーにしたがって修復が行われる。次のポリシーを指定できる。

disconnect

自動再同期を行わず接続を切断する。

consensus

after-sb-0pri アルゴリズムの結果が現在のセカンダリノードのデータを壊すことになる場合、セカンダリノードのデータを捨てる。そうではない場合は接続を切断する。

violently-as0p

プライマリ側のデータに対する間違った変更を施すことになるとしても、`after-sb-0pri` アルゴリズムの判断をつねに採用する。このポリシーは、**allow-two-primaries** オプションを指定した上で **OCFS2** および **GFS** 以外の通常のファイルシステムを利用した場合、さらにこのポリシーを指定していることを明確に意識している場合のみ有用である。このポリシーは危険である。プライマリノード側でファイルシステムをマウントしている場合、マシンがクラッシュする可能性がある。

discard-secondary

セカンダリ側のデータを捨てる。

call-pri-lost-after-sb

`after-sb-0pri` アルゴリズムの判断をつねに採用する。セカンダリ側のデータが正しいと判断された場合には、現在のプライマリ側で `pri-lost-after-sb` ハンドラが呼び出される。

after-sb-2pri policy

スプリットブレイン発生後に両ノードがプライマリになってしまった場合の処理を指定する。次のポリシーを指定できる。

disconnect

自動再同期を行わず接続を切断する。

violently-as0p

プライマリ側のデータに対する間違った変更を施すことになるとしても、`after-sb-0pri` アルゴリズムの判断をつねに採用する。このポリシーは、**allow-two-primaries** オプションを指定した上で **OCFS2** および **GFS** 以外の通常のファイルシステムを利用した場合、さらにこのポリシーを指定していることを明確に意識している場合のみ有用である。このポリシーは危険である。プライマリノード側でファイルシステムをマウントしている場合、マシンがクラッシュする可能性がある。

call-pri-lost-after-sb

どちらか一方のマシンで `pri-lost-after-sb` ハンドラを呼び出す。このプログラムには、マシンをリポートし、そのマシンをセカンダリにするような機能が要求される。

always-asbp

3番目のノードが存在しないことが現在の **UUID** 値から明らかな場合、通常のスプリットブレイン発生後の修復ポリシーだけが適用される。

このオプションを指定すると、両ノードのデータに関連性があると認められる場合のみ通常のスプリットブレイン発生後のポリシーが適用される。**UUID** の分析により3番目のノード

の存在が疑われる場合や、なんらかの別の原因によって間違っただ UID セットで判断してしまっただ場合には、フル同期が行われるかもしれない。

`rr-conflict policy`

再同期に関する判断と現在のクラスタ状態における役割が食い違っただ場合のポリシーを指定する。

`disconnect`

自動再同期を行わず接続を切断する。

`violently`

どちらか一方のノードのデータは安定であるという前提を無視して、プライマリノードへの同期を許可する。このオプションは危険で、使ってはいけない。

`call-pri-lost`

`pri-lost` ハンドラを呼び出す。このプログラムには、マシンをリブートしてそのマシンをセカンダリにするような機能が要求される。

`data-integrity-alg alg`

ネットワーク経由で受け渡されるデータの整合性を担保するために、DRBD はハッシュ値を比較する機能を備えている。通常は、TCP/IP パケット自体のヘッダに含まれる 16 ビットチェックサムで保証されるが、ギガビット NIC を備えた TCP/IP オフロードエンジンの中にはチェックサムを壊すものがあることがわかっている。したがってこのオプションをテスト中に指定し、十分にテストが終わった後は CPU 負荷を軽減するためにこのオプションを無効にするとよい。オプション値には、カーネルがサポートする任意のダイジェストアルゴリズムを指定できる。一般的なカーネルの場合、少なくとも md5、sha1 または crc32c のどれかが利用できる。デフォルトでは、この機能は無効である。

`wfc-timeout time`

接続確立までの待機時間にタイムアウト値を設定する。drbd(8) は、DRBD リソースの接続が確立するまで待ち続ける。後に起動されるクラスタマネージャは、通常はリソース内のスプリットブレイン状態まではチェックしない。待ち時間を制限したい場合には、このパラメータ値を設定する。デフォルト値は 0 で、タイムアウトせずに待ち続ける。単位は秒である。

`degr-wfc-timeout time`

縮退運転中のクラスタにおける接続タイムアウトを指定する。縮退とは、片方のノードしか動作していない状態を表す。このようなクラスタをリブートした場合には、対向ノードが一定時間内に起動する可能性が低いいため、wfc-timeout ではなくこのパラメータ値が使われる。デフォルト値は 60 秒で、秒単位で指定する。0 を指定すると、タイムアウトしなくなる。

`wait-after-sb`

このパラメータを指定すると、スプリットブレイン状態にあって接続自体が拒否されるような状態でも、起動スクリプトは接続まで待ち続けるようになる。

`become-primary-on node-name`

起動時にプライマリになるべきノードの名前を指定する。 `node-name` にはホスト名またはキーワード `both` を指定する。このパラメータが指定されていない場合は、両ノードともセカンダリ状態で起動する。通常は、役割の決定をクラスタマネージャ(`heartbeat` など)に委ねることが多い。

`rate rate`

DRBD の上位レイヤの円滑な実行のために、バックグラウンドの同期作業が利用するバンド幅を制限できる。デフォルト値は 250KB/秒。デフォルトの単位は KB/秒だが、K、M、G の接尾語を補って単位を変更できる。

`after res-name`

デフォルトでは、DRBD は複数リソースの再同期を並行して実行する。このパラメータを指定すると、`res-name` が接続完了してから、すなわち再同期が実行された場合にはそれが終了してから、他のリソースの再同期を実行する。

`al-extents extents`

DRBD はホットエリア(アクティブセット)を自動的に検出できる。このパラメータを指定すると、ホットエリアの大きさを制御できる。各エクステントは、低レベルデバイスの 4 メガバイトの領域になる。予定外の事情によってプライマリノードがクラスタから切り離されると、そのときのホットエリアのデータは、次回接続したときの再同期の対象になる。このデータ構造は、メタデータ領域に書き込まれる。したがって、アクティブセットの状態更新は、メタデータデバイスへの書き込みを引き起こす。エクステント値を大きくすると、再同期所要時間が長くなるが、メタデータの更新頻度を減らすことができる。`extents` のデフォルト値は 127 で、指定できる最小値は 7、最大値は 3843 である。

`verify-alg hash-alg`

`drbdsetup` コマンドの `verify` サブコマンドでディスク内容をオンライン検証できる。ビット単位の比較の代わりに、ブロックごとのハッシュ値による検証が行われる。検証に利用するハッシュアルゴリズムは、このパラメータで指定する。オプション値には、カーネルがサポートする任意のダイジェストアルゴリズムを指定できる。一般的なカーネルの場合、少なくとも `md5`、`sha1` または `crc32c` のどれかが利用できる。デフォルトでは、この機能は無効である。オンライン検証を有効にするには、このパラメータを明示的に設定する必要がある。

`cpu-mask cpu-mask`

DRBD カーネルスレッドに対する CPU アフィニティマスクを指定する。`cpu-mask` のデフォルト値は 0 で、DRBD カーネルスレッドは利用可能なすべての CPU にまたがって動作することを表す。

`pri-on-incon-degr cmd`

このパラメータに指定したハンドラは、ノードがプライマリで、セカンダリとの接続が切れていて、しかもローカルコピーに不整合がある場合に実行される。

pri-lost-after-sb *cmd*

このパラメータに指定したハンドラは、ノードがプライマリで、スプリットブレイン後の修復プロセスが失敗した場合に実行される。

pri-lost *cmd*

このパラメータに指定したハンドラは、ノード状態はプライマリであるにもかかわらず、DRBDはこのノードが同期先になるべきだと判断したときに実行される。このような状態になった場合、このノードはプライマリであることをやめるべきである。

outdate-peer *cmd*

このパラメータは、フェンシングメカニズムの一部を構成する。このパラメータに指定したハンドラは、対向ノードを「時代遅れ」状態にする必要が生じたときに実行される。呼び出されたコマンドは、DRBDが使っている通信経路とは別の経路を使うべきである。

local-io-error *cmd*

このパラメータに指定したハンドラは、ローカルディスク I/O サブシステムが I/O エラーを報告したときに実行される。

split-brain *cmd*

このパラメータに指定したハンドラは、スプリットブレイン状態が検出されたときに実行される。修復のための手作業が必要なので、このハンドラは、誰かにこのことを通知するのが望ましい。

バージョン

このドキュメントは DRBD バージョン 8.0 向けに書かれている。

著者

Philipp Reisner <philipp.reisner@linbit.com>、Lars Ellenberg <lars.ellenberg@linbit.com>

バグ報告方法

バグについては、<drbd-user@lists.linbit.com>宛のメールで報告してほしい。

著作権

Copyright 2001-2008 LINBIT Information Technologies, Philipp Reisner, Lars Ellenberg. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

本マニュアルの日本語版の翻訳著作権は、株式会社サードウェアが保有しています。

参照

drbd(8), drbddisk(8), drbdsetup(8) drbdadm(8) *DRBD* ホームページ (英語)
(<http://www.drbd.org/>) *DRBD* ホームページ (日本語) (<http://www.drbd.jp/>)