

Pacemakerで かんたんクラスタリング 体験してみよう！

～仮想環境で構築デモしますよ編～

2011年6月11日 OSC2011 Hokkaido

Linux-HA Japan プロジェクト

田中 崇幸



自己紹介

- 名前：田中崇幸 (Takayuki Tanaka)
 - Twitter: @tanakacchi21
- 所属：Linux-HA Japanプロジェクト
 - コミュニティ旗揚時のメンバー
 - Pacemaker普及促進のため、OSCでの講演で全国行脚中
- 趣味：マラソン・野球観戦・サッカー観戦
 - 念願のサブスリーを達成したばかりの市民マラソンランナー
 - もちろん、「北海道日本ハムファイターズ」と「コンサドーレ札幌」の大ファン



本日のお話

- ① Linux-HA Japanについて
- ② 本日のPacemakerデモ環境
- ③ インストール・設定をデモします！
- ④ フェイルオーバ・系切り替えをデモします！

①

Linux-HA Japanについて



Linux-HA Japanの経緯

『Heartbeat(ハートビート)』の日本における更なる普及展開を目的として、2007年10月5日「Linux-HA (Heartbeat) 日本語サイト」を設立しました。

その後、日本でのLinux-HAコミュニティ活動として、Heartbeat2のrpmバイナリと、オリジナルのHeartbeat機能追加用パッケージを提供してきました。

Linux-HA Japan URL

<http://linux-ha.sourceforge.jp/>

(一般向け)

<http://sourceforge.jp/projects/linux-ha/> (開発者向け)



Pacemaker情報の公開用として
新しい一般向けウェブサイトが
2010/6/25にオープンしました。

本日の資料もこのサイトから
公開予定です！

Linux-HA Japanメーリングリスト

日本におけるHAクラスタについての活発な意見交換の場として「Linux-HA Japan日本語メーリングリスト」も開設しています。

Linux-HA-Japan MLでは、Pacemaker、Heartbeat3、Corosync DRBDなど、HAクラスタに関連する話題は歓迎！

• ML登録用URL

<http://linux-ha.sourceforge.jp/>
の「メーリングリスト」をクリック



• MLアドレス

linux-ha-japan@lists.sourceforge.jp

※スパム防止のために、登録者以外の投稿は許可制です

■ gihyo.jp に連載しました！

『Pacemakerでかんたんクラスタリング
体験してみよう！』



<http://gihyo.jp/admin/serial/01/pacemaker>

合計5回の連載で、Pacemakerの概要説明から構築方法、保守運用にいたるまで紹介しています。

②

本日のPacemakerデモ環境

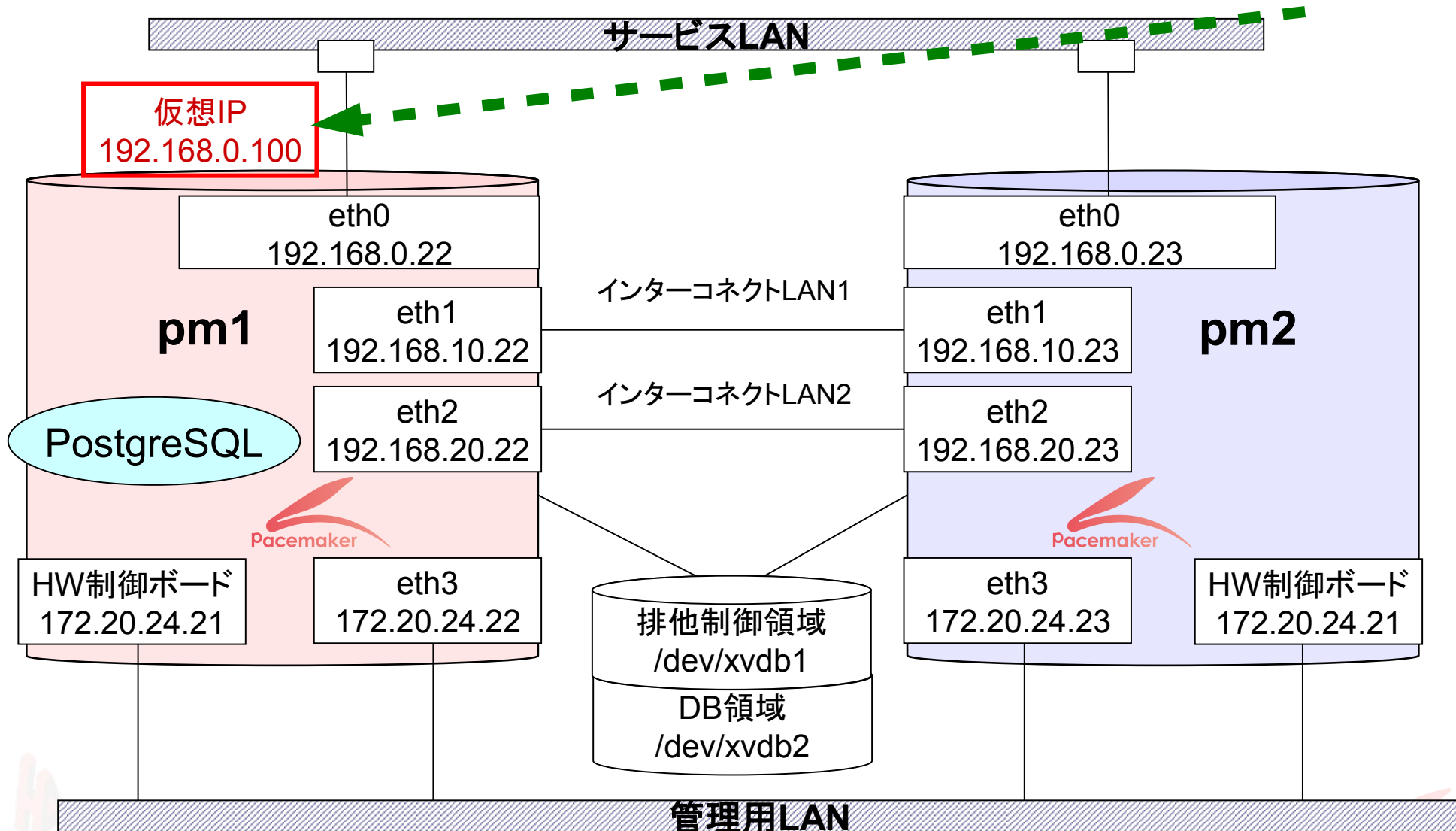



本日のPacemakerデモ環境

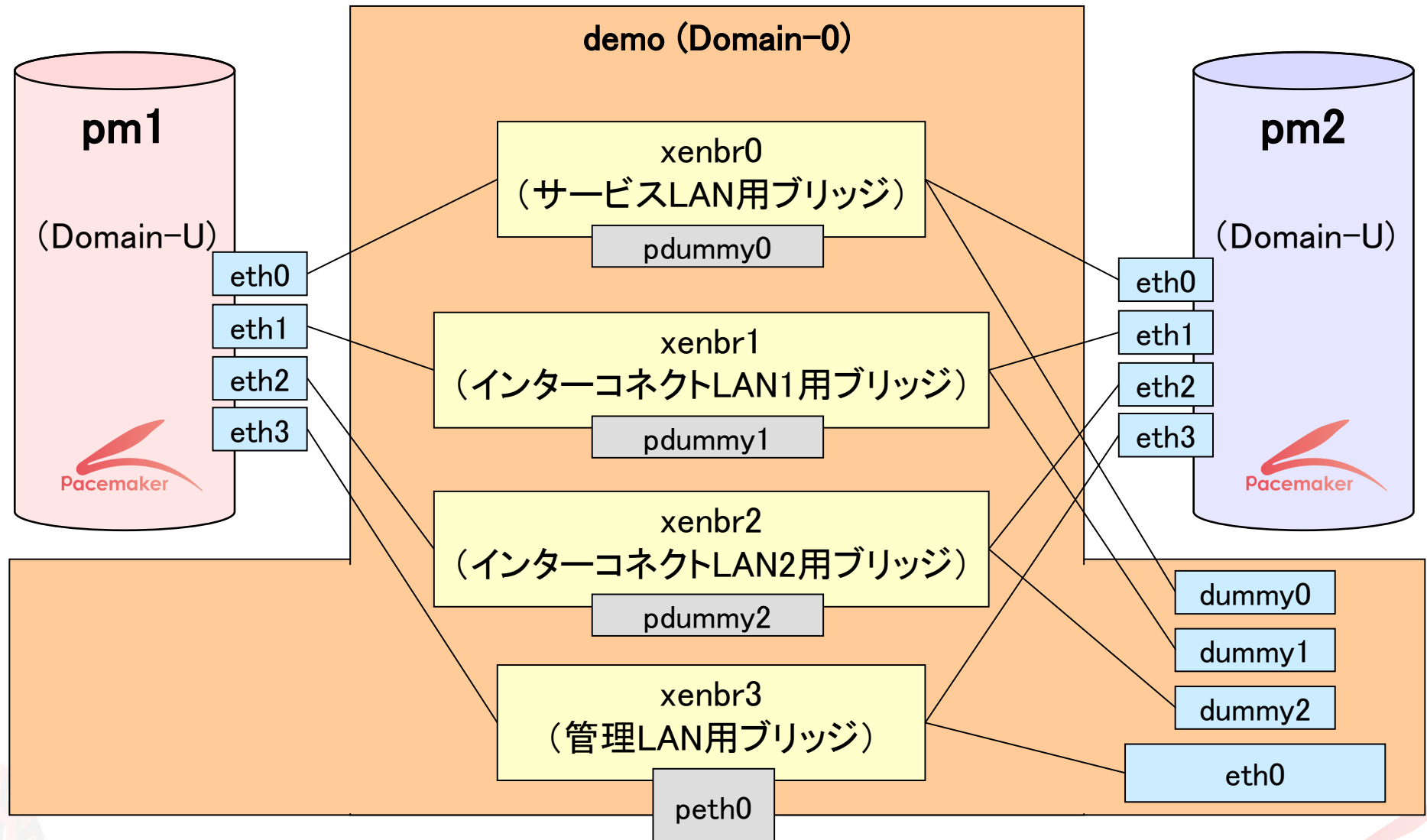
- ハードウェア
 - ノートPC (Core2Duo 2.26MHz、メモリ 2G)
- OS
 - CentOS 5.6 x86_64
- HAクラスタ
 - Pacemaker-1.0.10 (インストールのデモを行います)
- クラスタ化するアプリケーション
 - PostgreSQL 9.0.4 (インストール済み)
- 仮想環境
 - Xen (CentOS 5.6同梱版)
 - Domain-Uは2ドメインで構成
 - 各ドメインには、CPU×1・メモリ480M を割り当て

Pacemakerデモ構成

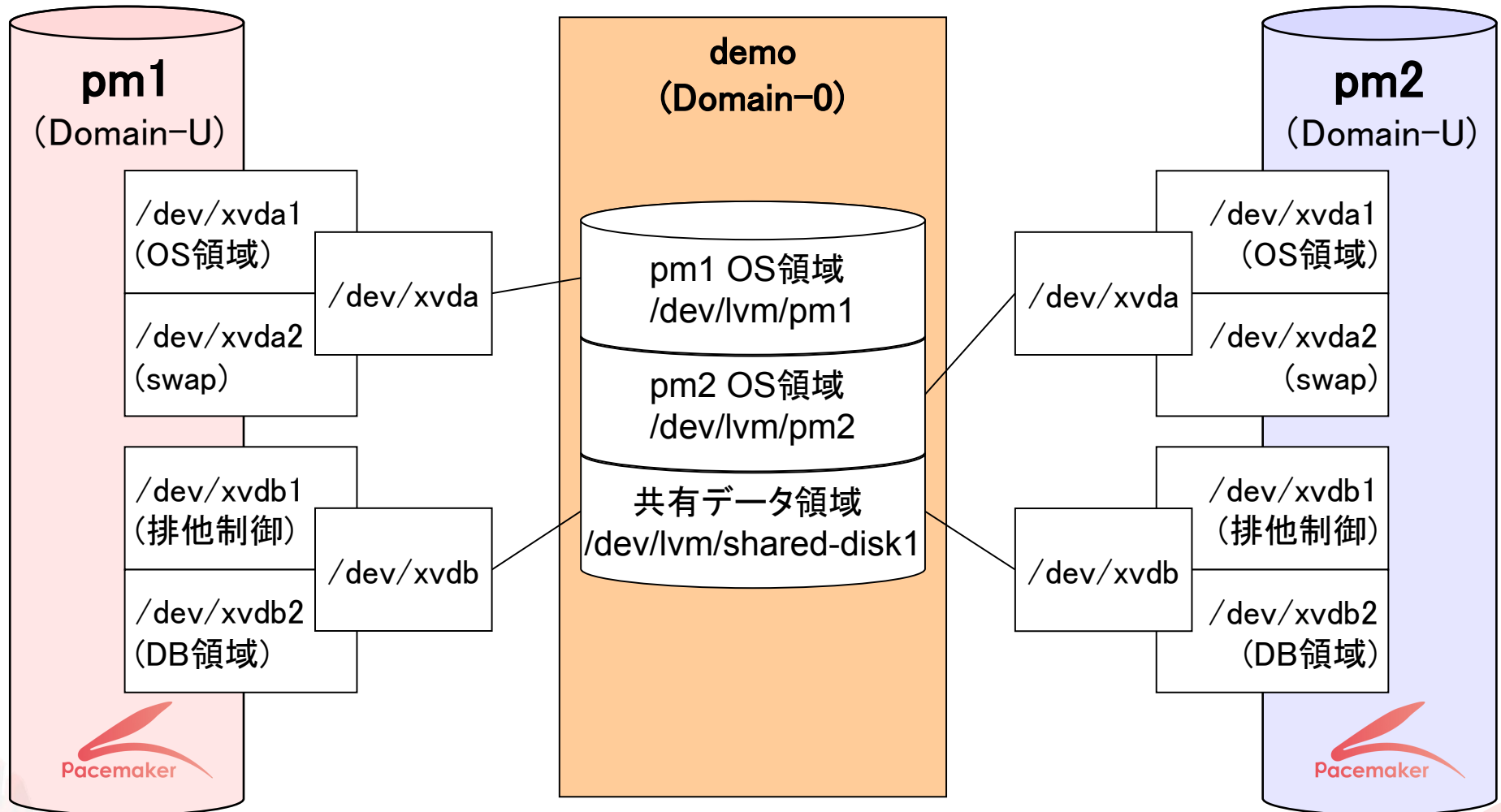
demo
(Domain-0)



Pacemakerデモ機構成 (Xen仮想NW)



Pacemakerデモ機構成 (Xen仮想ディスク)



Pacemakerデモ

リソース構成

これら4つの
リソースは
グループ設定します

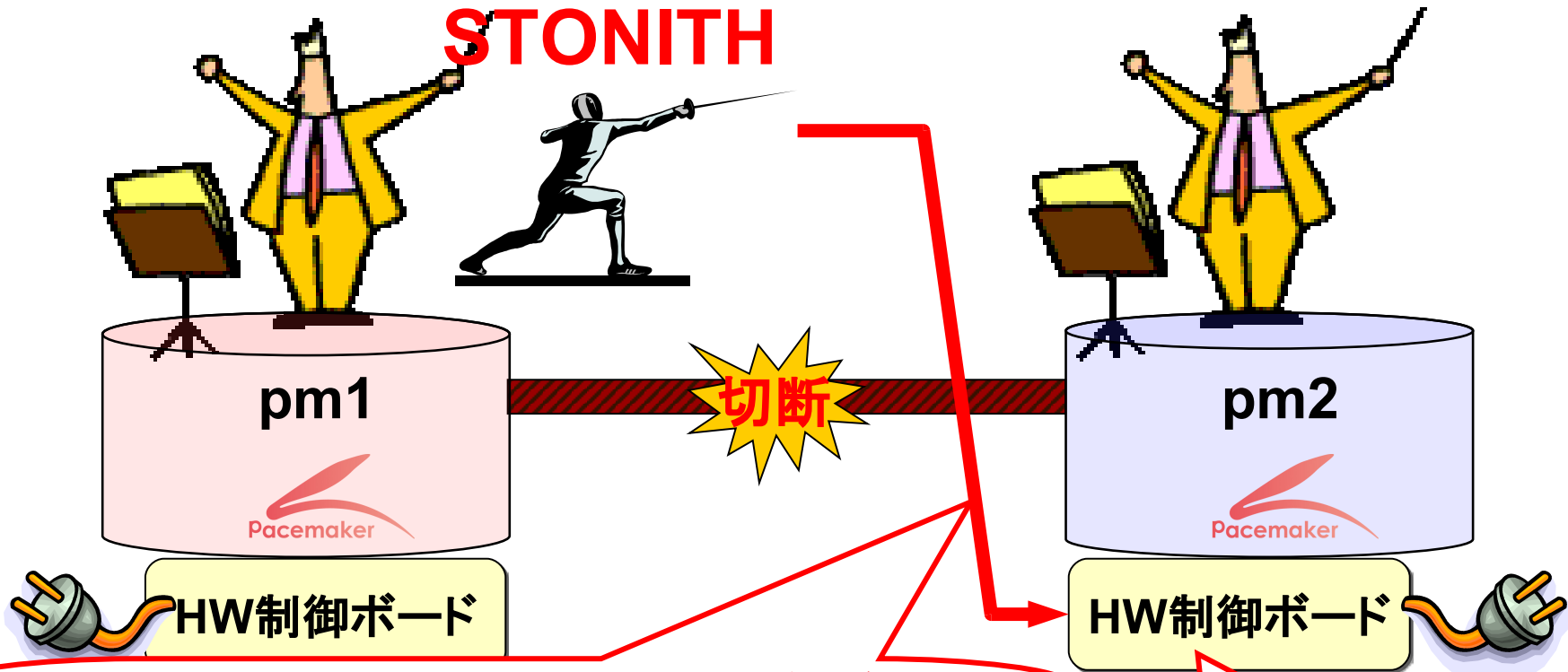
- ディスク排他制御 (sfex)
 - 共有ディスクの排他制御を行います
- DBデータ領域マウント (Filesystem)
 - 共有ディスクにあるDBデータ領域のマウント制御を行います
- 仮想IP割り当て (IPaddr2)
 - サービス提供用の仮想IPを割り当てます
- PostgreSQL制御 (pgsql)
 - PostgreSQL 9.0.4 の制御を行います

今日はSTONITH
のデモも行います

- STONITH (stonith-helper, xen0, meatclient)
 - STONITHは「Shoot The Other Node In The Head」の略で、監視対象サーバの異常を検出したときに、強制的にそのサーバをダウンさせるノードフェンシングを行います。
- ネットワーク監視 (pingd)
 - 指定したIPアドレスに ping送信し、ネットワーク疎通があるかどうかの監視を行います。
- ディスク監視 (diskd)
 - 指定したディスクデバイスにアクセスし、ディスクの正常性確認を行います。

STONITH実行例

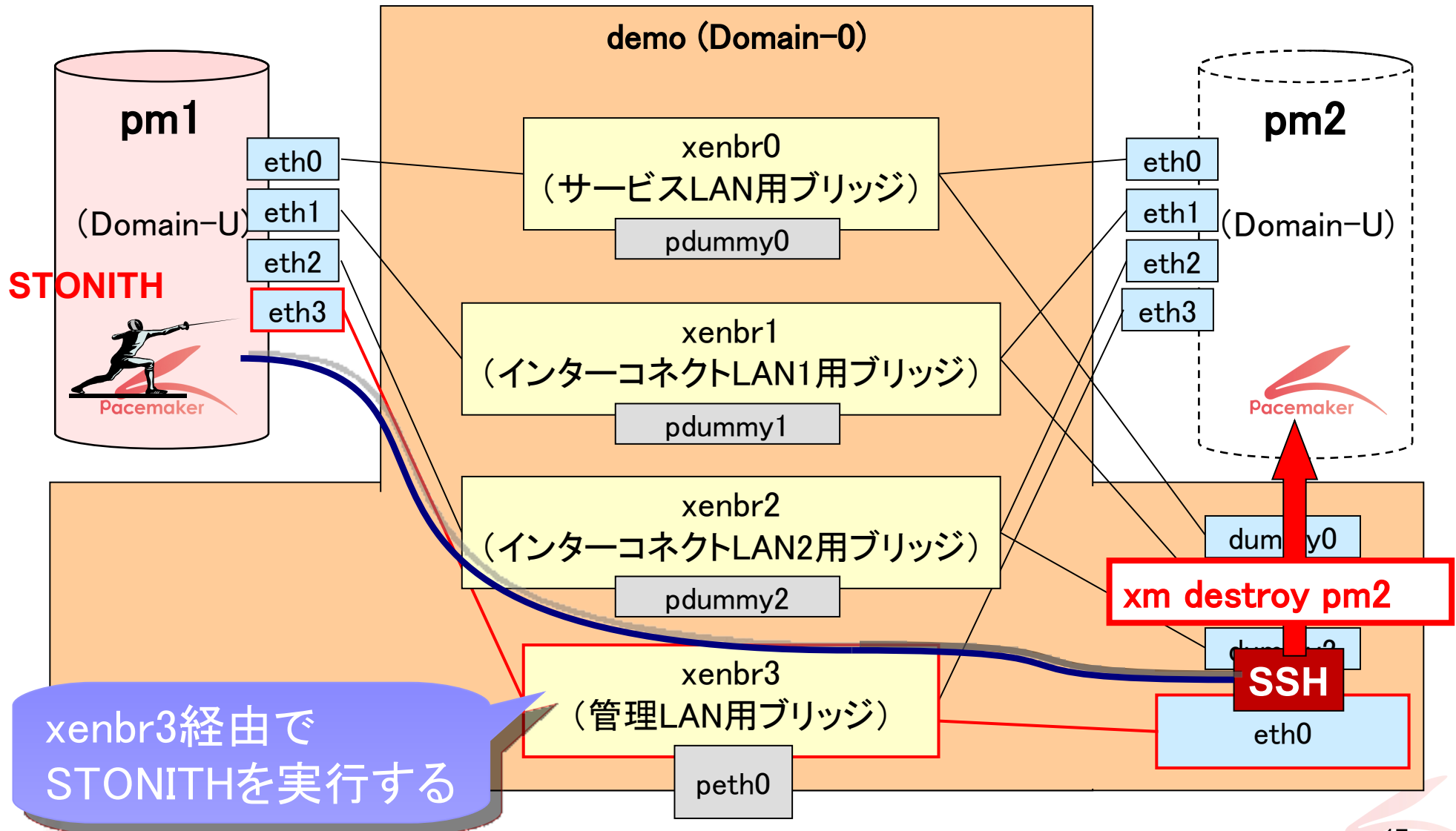
STONITH



インターコネクトとは別の通信経路で
HW制御ボードに対しリセットを実行

OSと連動しないHW制御ボードから強制電源断

Pacemakerデモ機フェンシング (STONITH) 構成



③

インストール・設定を
デモします！



インストール方法の種類

1. yum を使ってネットワークインストール
 - Pacemaker本家(clusterlabs) の yumのリポジトリを使用
 - サーバにインターネット接続必須
2. ローカルリポジトリ + yum を使ってインストール
 - Linux-HA Japan 提供のリポジトリパッケージを使用
 - Linux-HA Japan オリジナルパッケージも含まれる
3. rpm を手動でインストール
 - 沢山のrpmを個別にダウンロードする必要あり
4. ソースからインストール
 - 最新の機能をいち早く試せる
 - コンポーネントが多いので、コンパイルは面倒

本日は「2」の
構築デモを行
います

～ ローカルリポジトリ + yum を使ってインストール ～

(サーバにインターネット接続環境がなくてもOK!)

■ 1. Pacemakerリポジトリパッケージをダウンロード

Linux-HA Japan 提供の Pacemakerリポジトリパッケージを sourceforge.jp からダウンロードしておきます。

pacemaker-1.0.10-1.4.2.el5.x86_64.repo.tar.gz
をダウンロード

SourceForge.jp > ソフトウェアを探す > Linux-HA Japan > 概要

Linux-HA Japan

本ページはLinux-HA Japan 開発者向けサイトです。プロジェクトのメインサイトはこちらです <http://linux-ha.sourceforge.jp/>
Linux-HA Japanプロジェクトは、Linux上で高可用クラスシステムを構築するための部品として、オープンソースの、クラスターソースマネージャ、クラスター通信レイヤ、ブロックデバイス複製、その他、さまざまなアプリケーションに対応するための数多くのリソースエージェント、などを、日本国内向けに維持管理、支援等を行っているプロジェクトです。
主な製品として、Pacemaker、Heartbeat、Corosync、DRBD等を取り扱っています。

[Linux-HA Japanの詳細情報へ](#)

[Linux-HA Japanのインストール方法](#)

[Linux-HA Japanの使い方](#)

最終更新日: 2010-08-23 12:44

開発メンバー: [ksk](#), [t-matsuo](#), [takayukitanaka](#), [b-oka](#), [bellche](#), [hideoyamauchi](#), [iidayuus](#), [ikedaj](#), [inouekazu](#), [jzujlura](#), [kmi](#), [kitateishi](#), 他6名 [一覧]

[その他の情報](#)

Pacemaker-1.0.10-1.4.2 版は
6/6リリース

ダウンロード

最終更新日: 2010-06-18 07:21

■ 2. yumでインストール！

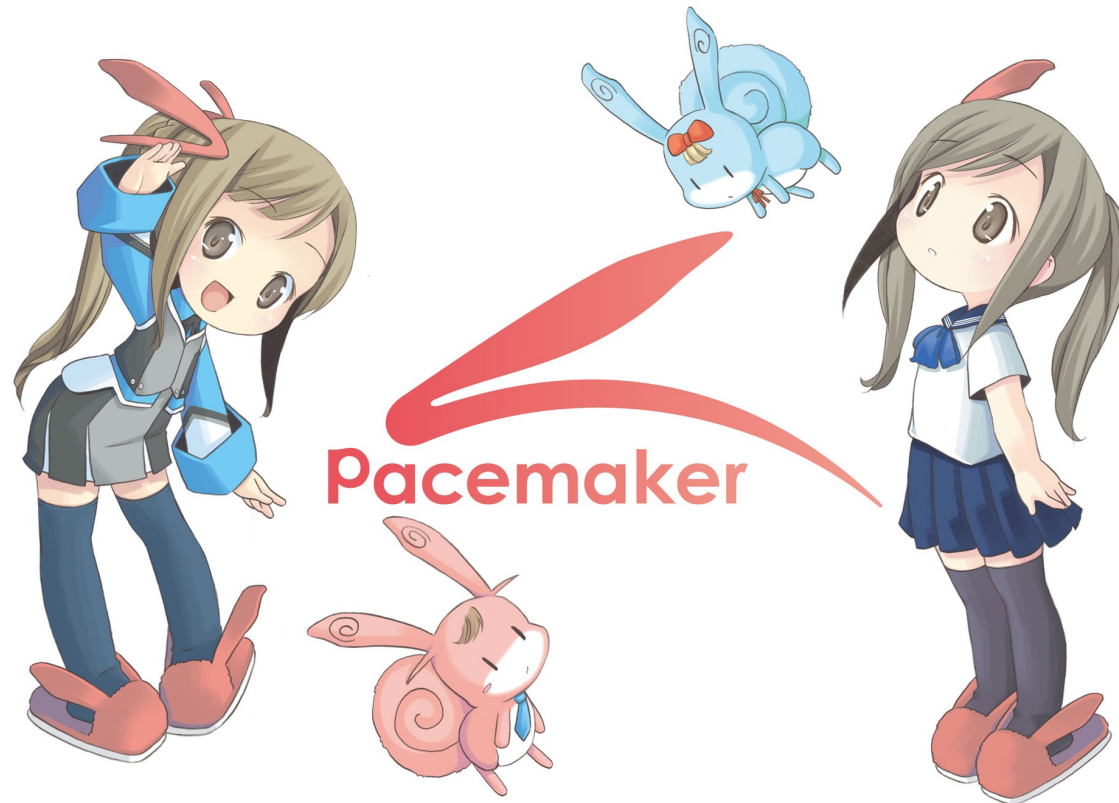
/tmp で展開し、yumコマンドでインストールします。

```
# cd /tmp
# tar zxvf pacemaker-1.0.10-1.4.2.el5.x86_64.repo.tar.gz
# cd /tmp/pacemaker-1.0.10-1.4.2.el5.x86_64.repo/
# yum -c pacemaker.repo install pacemaker pm_crmgen pm_diskd
pm_logconv-hb pm_extras
```

- pm_crmgen-1.1-1.el5.noarch.rpm … crm用設定ファイル編集ツール
- pm_diskd-1.0-1.el5.x86_64.rpm … ディスク監視アプリとRA
- pm_logconv-hb-1.1-1.el5.noarch.rpm … ログ変換ツール
- pm_extras-1.1-1.el5.x86_64.rpm … その他オリジナルRA 等

ぜひぜひ使ってみてください！

ここでやっつと Pacemakerインストールを デモします！



クラスタ制御部基本設定

/etc/ha.d/ha.cf

- クラスタ制御部の基本設定ファイル
- クラスタ内の全サーバに同じ内容のファイルを設置

```
pacemaker on  
  
debug 0  
udpport 694  
keepalive 2  
warntime 7  
deadtime 10  
initdead 48  
logfacility local1  
  
bcast eth1  
bcast eth2  
  
node pm1  
node pm2  
  
watchdog /dev/watchdog  
respawn root /usr/lib64/heartbeat/ifcheckd
```

pm_extrasをインストールし、
この ifcheckd の設定を追加
すればインターコネクトLAN
の接続状況も確認可能です

/etc/ha.d/authkeys

- サーバ間の「認証キー」を設定するファイル
- クラスタ内の全サーバに、同じ内容のファイルを配置
- 所有ユーザ/グループ・パーミッションは root/root ・ rw---- に設定

```
auth 1  
1 sha1 hogehoge
```

これも基本的に
Heartbeat2 と
設定は同じです

認証キー: 任意の文字列

認証キーの計算方法: sha1, md5, crcを指定可

/etc/syslog.conf

- 必須の設定ではないが、多くのログが /var/log/messages に出力されるため出力先を個別のファイルに変更するのがお勧め

local1.info を使用し、/var/log/ha-log へ出力する場合の例

```
*.info;mail.none;authpriv.none;cron.none;local1.none  
/var/log/messages
```

⋮
(省略)

local1.info

/var/log/ha-log

ha.cf で設定したlogfacility 名

ここまでいけば、
Pacemakerが起動できます！

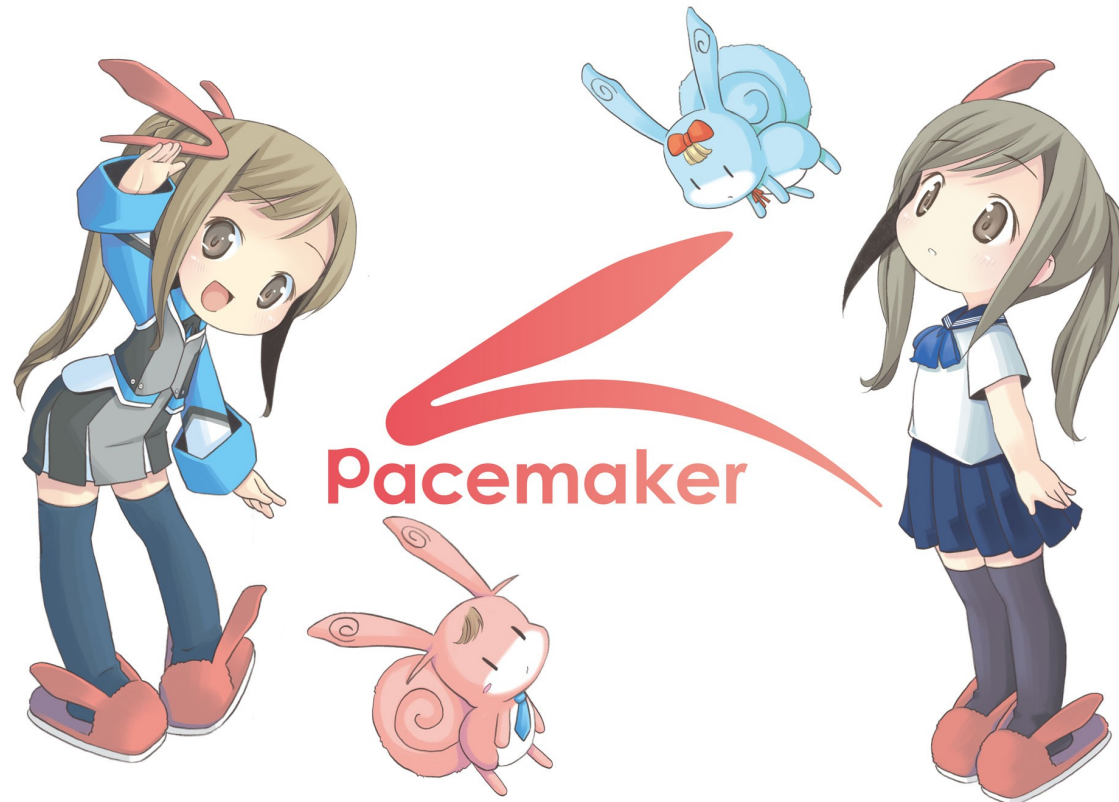
```
# /etc/init.d/heartbeat start
```

← 2サーバで実行

```
Starting High-Availability services:
```

[OK]

ということで、
Pacemakerを起動してみます！



起動確認

Pacemakerの状態表示コマンドである
crm_monコマンドを利用します。

```
# crm_mon
```

```
=====
```

```
Last updated: Wed Nov 10 14:28:55 2010
```

```
Stack: Heartbeat
```

```
Current DC: pm2 (a59a9306-d6e7-4357-bb0c-a5aea0615e61) - partition  
with quorum
```

```
Version: 1.0.10-da7075976b5ff0bee71074385f8fd02f
```

```
2 Nodes configured, unknown expected votes
```

```
0 Resources configured.
```

```
=====
```

```
Online: [ pm2 pm1 ]
```

クラスタに組み込まれている
サーバ名が表示されます

-fA オプションを付与すると、インターコネクト LANの接続状況も確認可能です。

```
# crm_mon -fA
```

```
=====  
~ 省略 ~  
=====
```

```
Online: [ pm2 pm1 ]
```

```
Node Attributes:
```

```
* Node pm2:
```

```
+ pm1-eth1           : up
```

```
+ pm1-eth2           : up
```

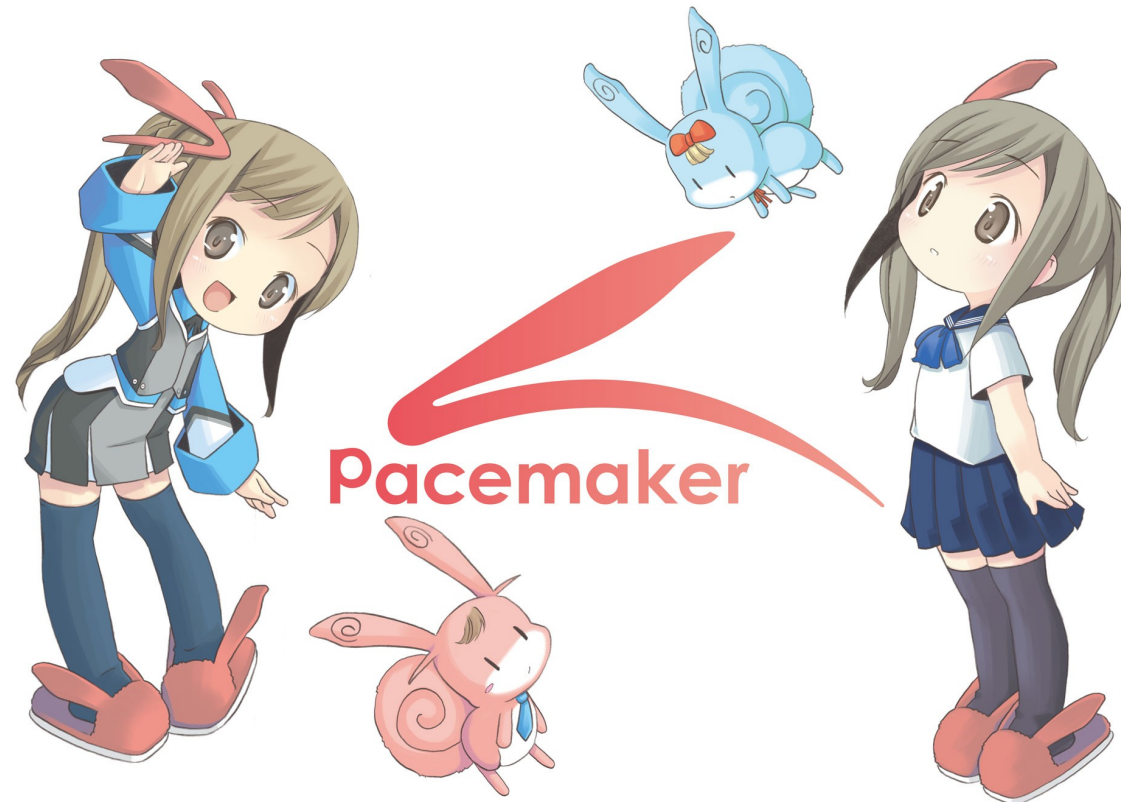
```
* Node pm1:
```

```
+ pm2-eth1           : up
```

```
+ pm2-eth2           : up
```

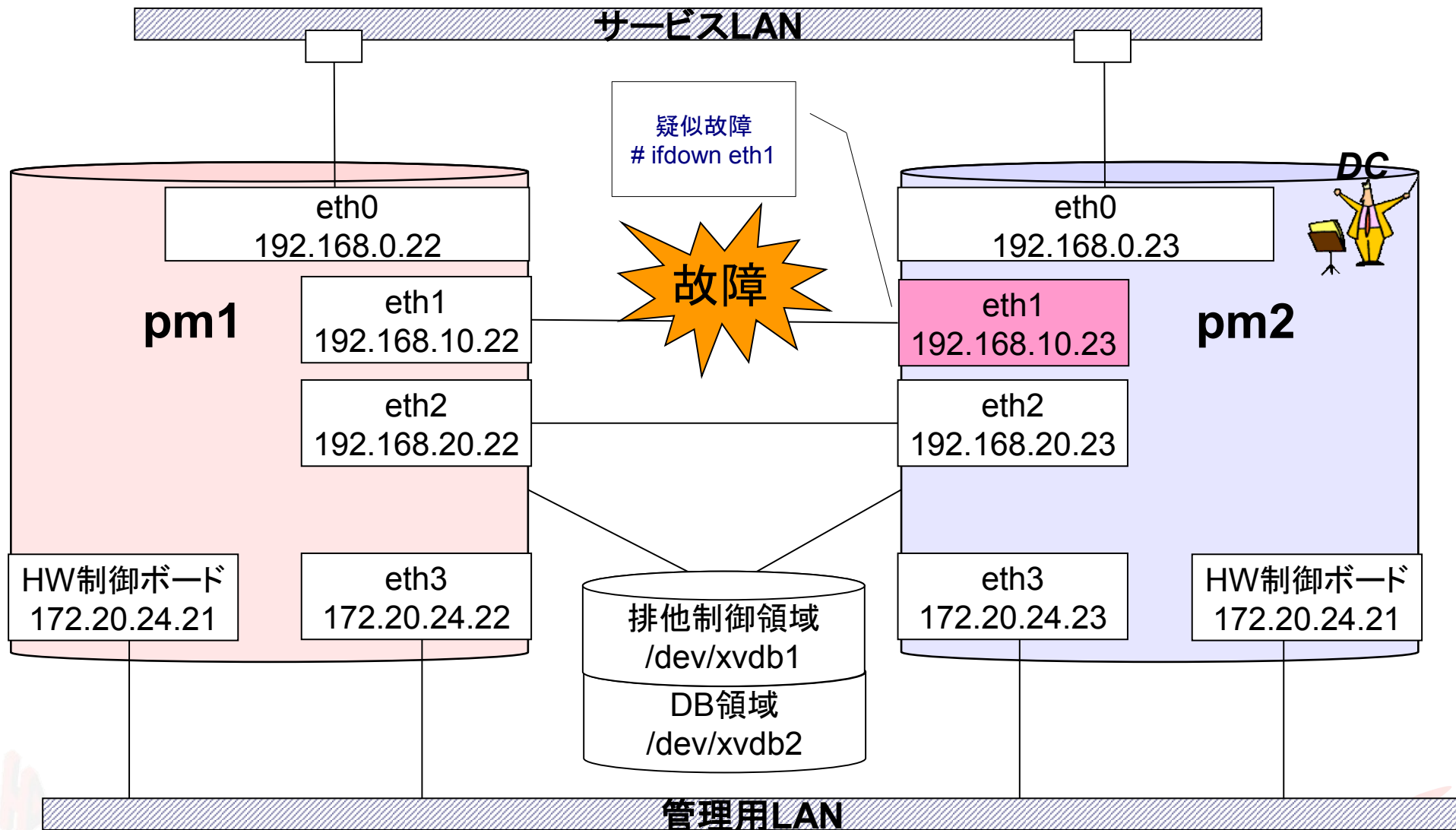
インターコネクトがUPされている
のが確認可能

ここで、Pacemaker状態表示と インターコネクトLAN故障を デモします！



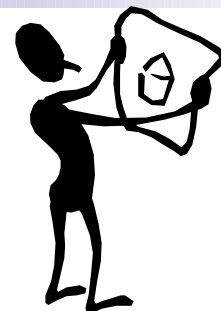
デモ例は
次ページ

インターコネクトLAN1を故障させてみる…



これだけでは、
リソース設定が無いので
なーんにも
アプリケーションは
起動していません…





リソース計画

■ リソース制御するには事前に計画が必要

□ リソースの選択

Apache、PostgreSQL、NW監視など、何を使用するか？
リソースエージェント(RA)がなければ、予め自作してみるか？

□ リソースの動作の定義

リソースの監視(monitor)間隔は何秒にするか？タイムアウトは？
故障時はどのように動作させるか？
リソースエージェント(RA)に与えるパラメータは？

□ リソース配置・連携の定義

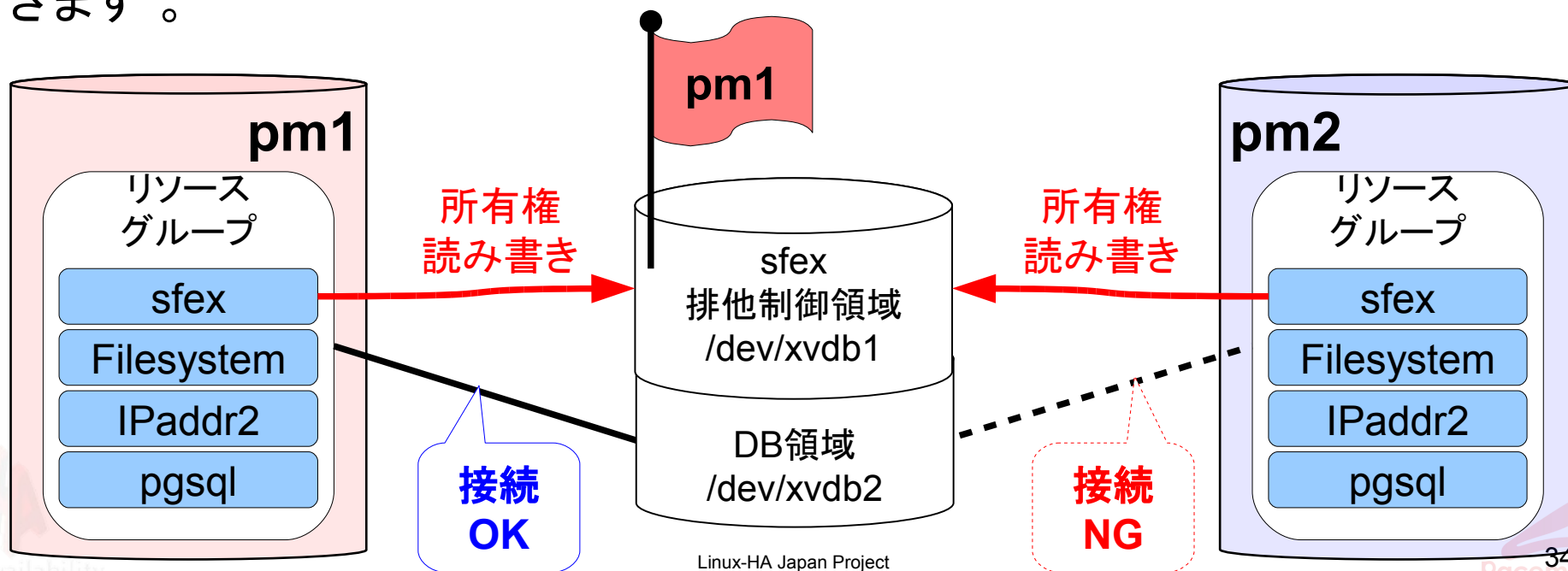
リソースをどのサーバで起動させるか？
リソースの起動順番は？



共有ディスク排他制御機能

sfex (Shared Disk File EXclusiveness Control Program)

sfexは共有ディスクの所有権を制御するリソースです。
共有ディスク上のデータパーティションを使うリソースと一緒にリソースグループを作ります。
所有権を持ったサーバのリソースのみがデータパーティションにアクセスできます。



排他制御領域の初期化

共有ディスク排他制御機能(sfex)を使用するためには、排他制御領域を初期化する必要性があります。

```
# sfex_init -n 1 /dev/xvdb1
```

排他制御領域のデバイス名を指定する

※ ext3などのファイルシステムを作成する必要性はありません。

リソース設定方法

■ 主に2通り

- cib.xml ファイルにXML形式で設定を記述
 - 従来のHeartbeat 2での方法
 - XMLを手で書く必要があり面倒
- crmコマンドで設定
 - Pacemakerからの新機能
 - crmファイル編集ツールは、Linux-HA Japanより提供

本日はcrmファイル編集ツールを使用して構築デモを行います



crmファイル編集ツール pm_crmgen

6/6 に pm_crmgen 1.1版を
リリース

Linux-HA Japanで
crmファイル編集ツールを提供中！

Excelのテンプレートファイルから簡単に
crm用設定ファイルを生成してくれるツールです。

リポジトリパッケージに含まれていますし、
個別にダウンロードも可能です。

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



- ・どのサーバが優先的にActive？
 - ・NW監視は？
 - ・NWが壊れた時の挙動は？
 - ・STONITHの設定は？
- など細かい挙動の設定も
可能です！



設定イメージ

1) Excelのテンプレートファイルにリソース定義を記載

`/usr/share/pacemaker/pm_crmgen/pm_crmgen_env.xls` ファイルを

Excel が使用できるPCにコピーします。
 テンプレートは青枠の中に値を
 記入していきます。

本日の仮想デモ環境
 は、このExcelの設定
 例シートでほとんど構
 築が可能です！

#表 5-3 クラスタ設定 ... Primitiveリソース (id=prmlp)

PRIMITIVE					
P	id	class	provider	type	
#	リソースID	class	provider	type	
	prmlp	ocf	heartbeat	IPaddr2	
A	type	name	value		
#	パラメータ種別	項目	設定内容		
	params	ip	192.168.0.100		
		nic	eth0		
		cidr_netmask	24		
O	type	timeout	interval	on-fail	start-delay
#	オペレーション	タイムアウト値	監視間隔	on_fail (障害時の動作)	起動前処理
	start	60s	0s	restart	
	monitor	60s	10s	restart	
	stop	60s	0s	fence	

監視間隔やタイムアウト値、
 故障時の動作などを入力

どのサーバをActiveにするかといった
リソース配置制約の設定も、サーバ名を記述
するだけで可能です。

270 #表 6-1 クラスタ設定 ... リソース配置制約

271 LOCATION

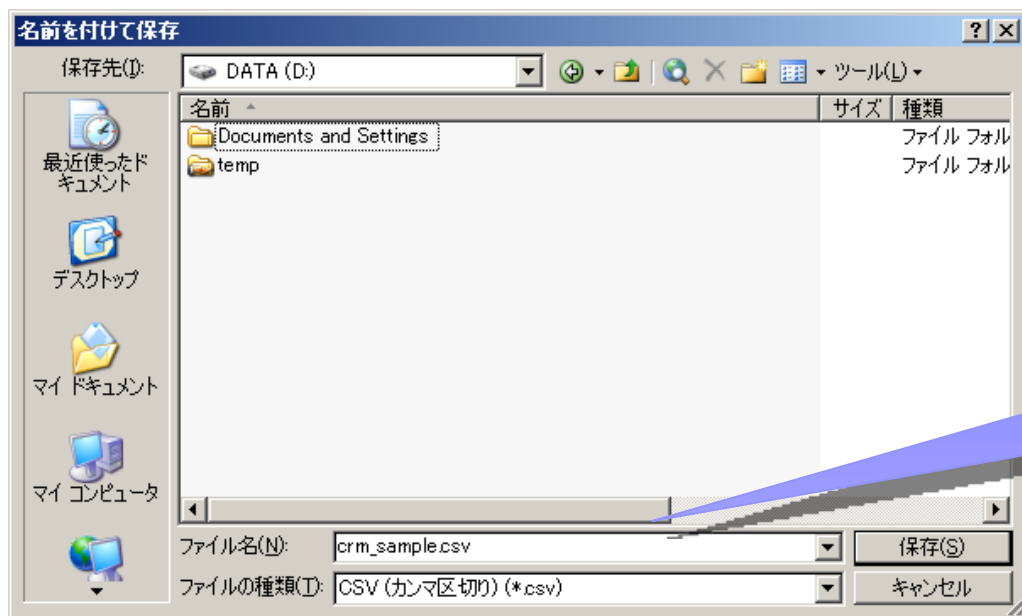
272	rsc	score:200	score:100	score:-inf
273 #	リソースID	Activeノード	Standbyノード	非稼働ノード
274	erpPe	pm1	pm2	
275	erpStonith1			pm1
276	erpSto			pm2

リソースID

ActiveとStandbyサーバを指定

crm用設定ファイルに変換

2) CSV形式でファイルを保存



3) CSVファイルをサーバへ転送

CSVファイル保存後、SCPやFTP等でpm_crmgenがインストールされたサーバへ転送

crm用設定ファイルに変換

4) pm_crmdgenコマンドでcrmファイルを生成

```
# pm_crmdgen -o crm_sample.crm crm_sample.csv
```

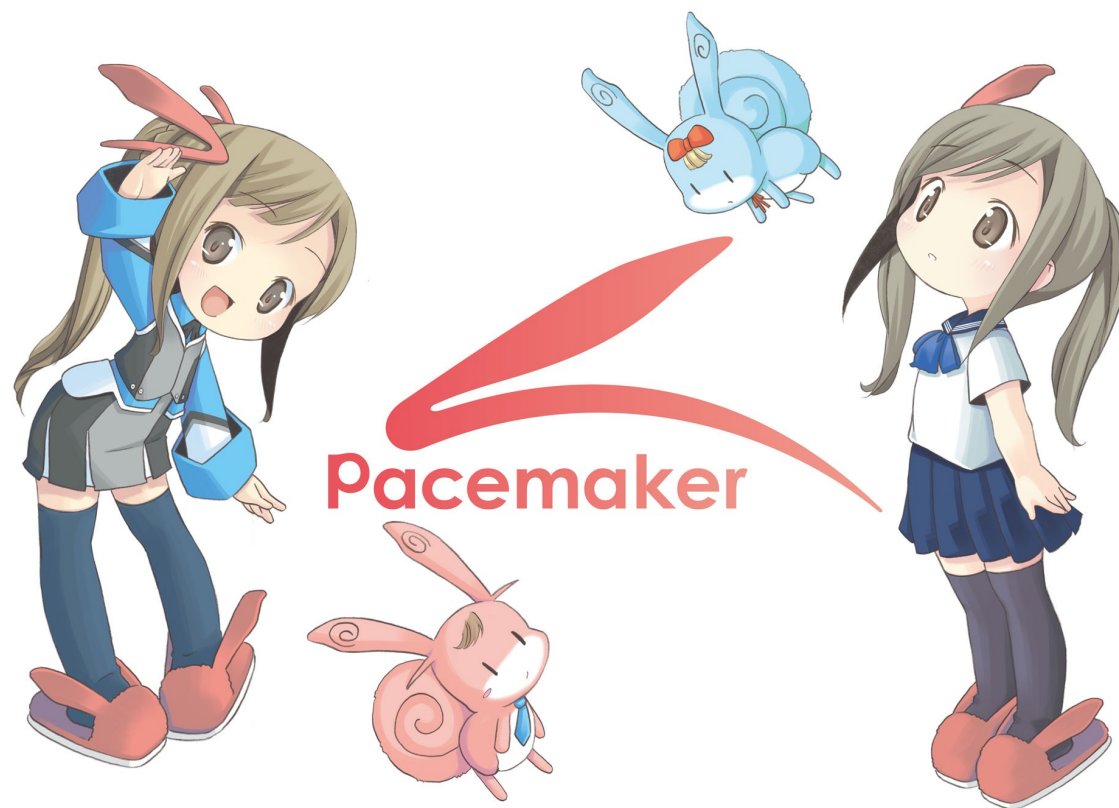
生成する設定ファイル名

3)で転送した
CSVファイル

5) crmコマンドを実行してリソース設定を反映

```
# crm configure load update crm_sample.crm
```

リソース設定をして サービスの起動と、本当にサービス が起動しているかデモします！

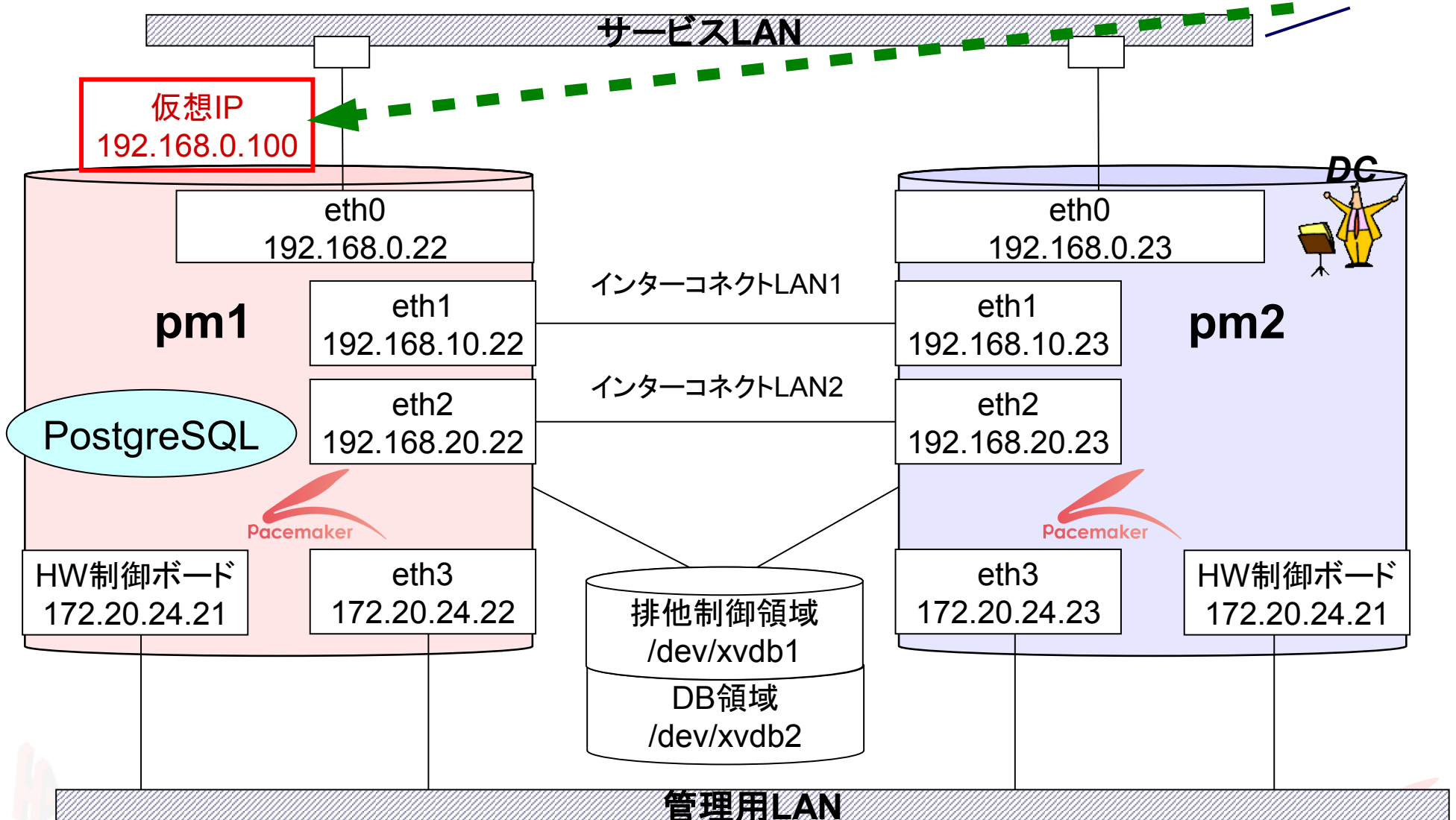


デモ例は
次ページ

PostgreSQLに接続してみる...

```
demo# psql -U postgres -h 192.168.0.100 -l
```

demo
(Domain-0)



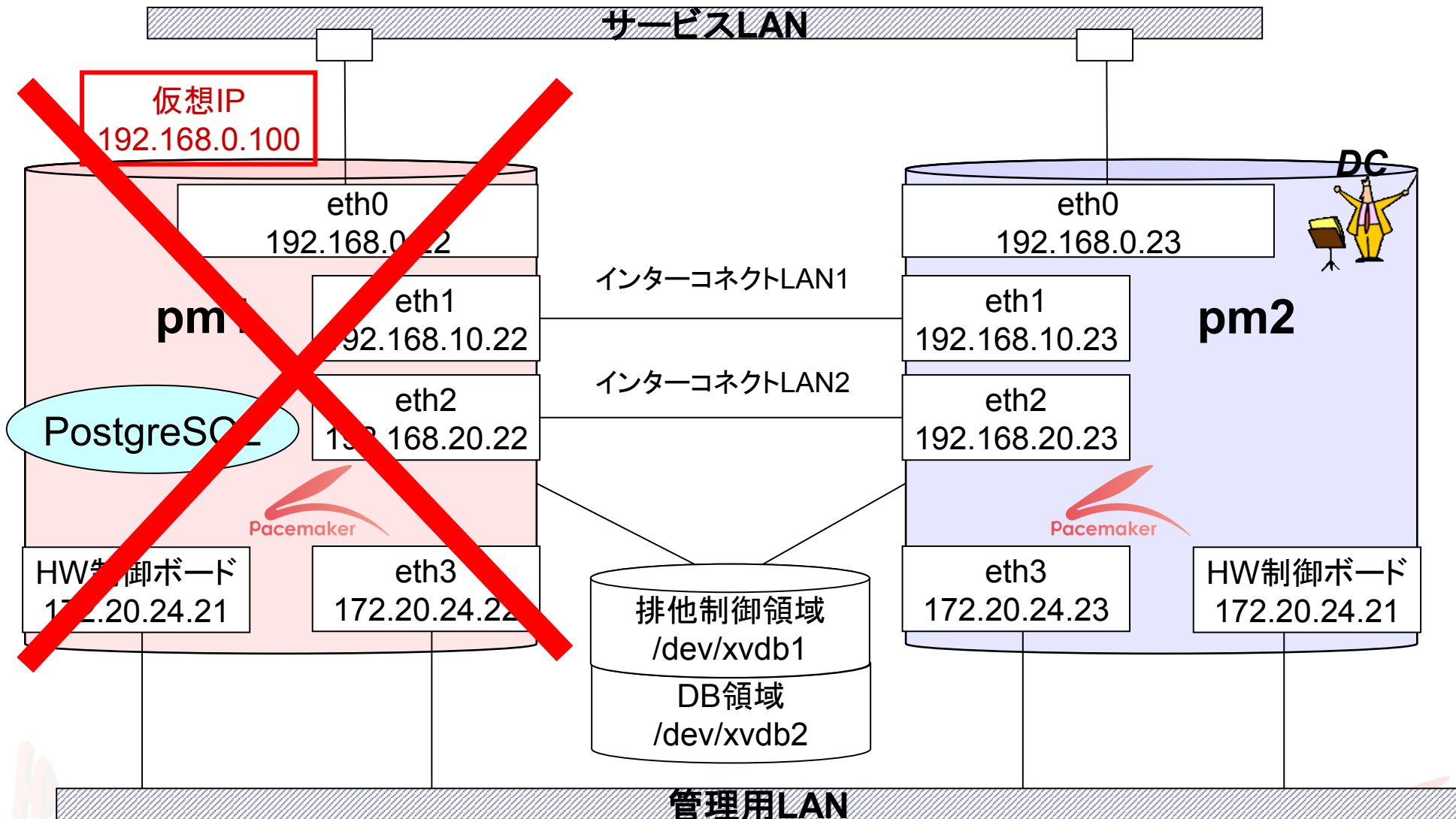
③

フェイルーバ・系切り替えを
デモします！



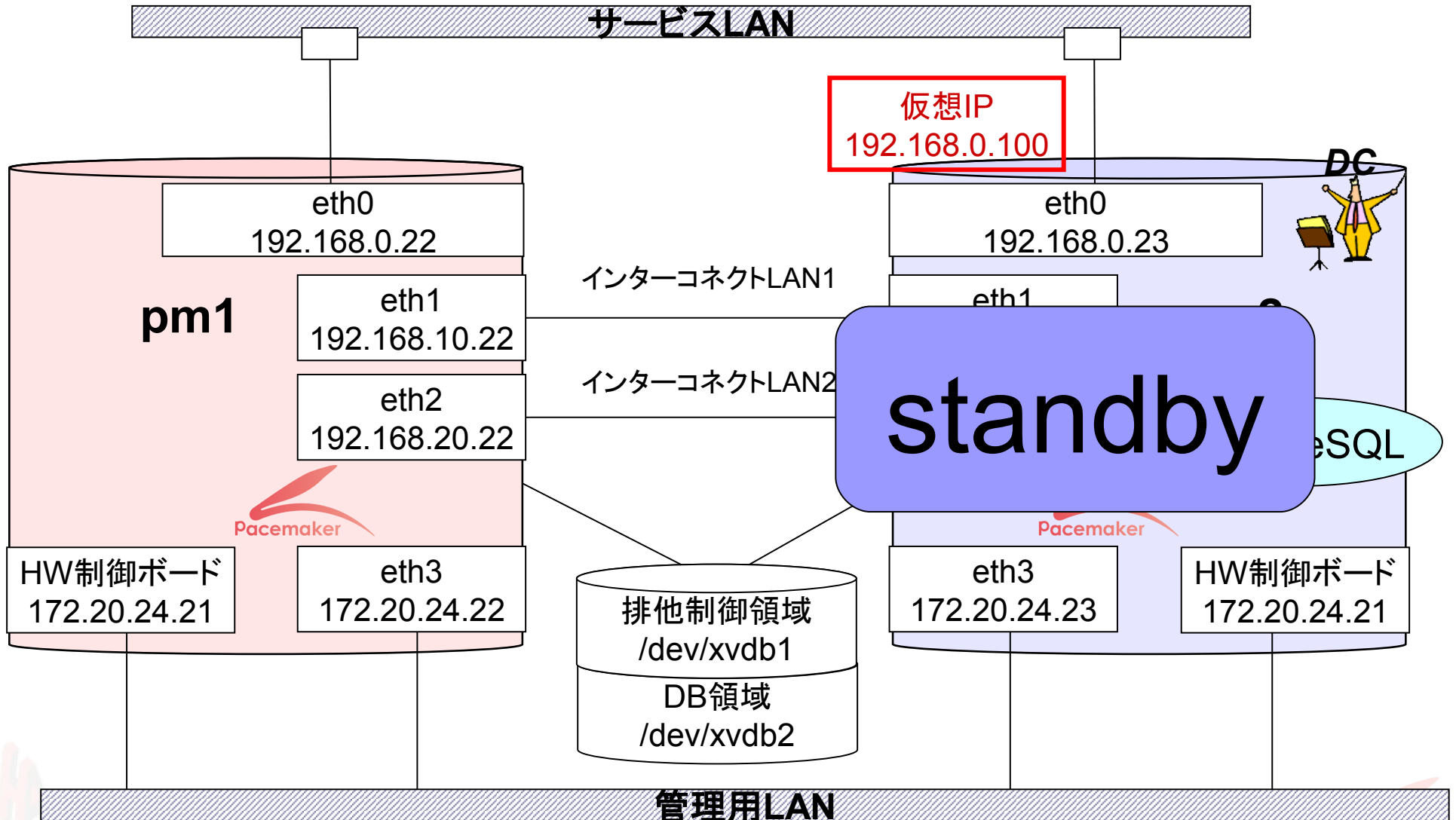
pm1を強制停止してみる…

demo# xm destroy pm1



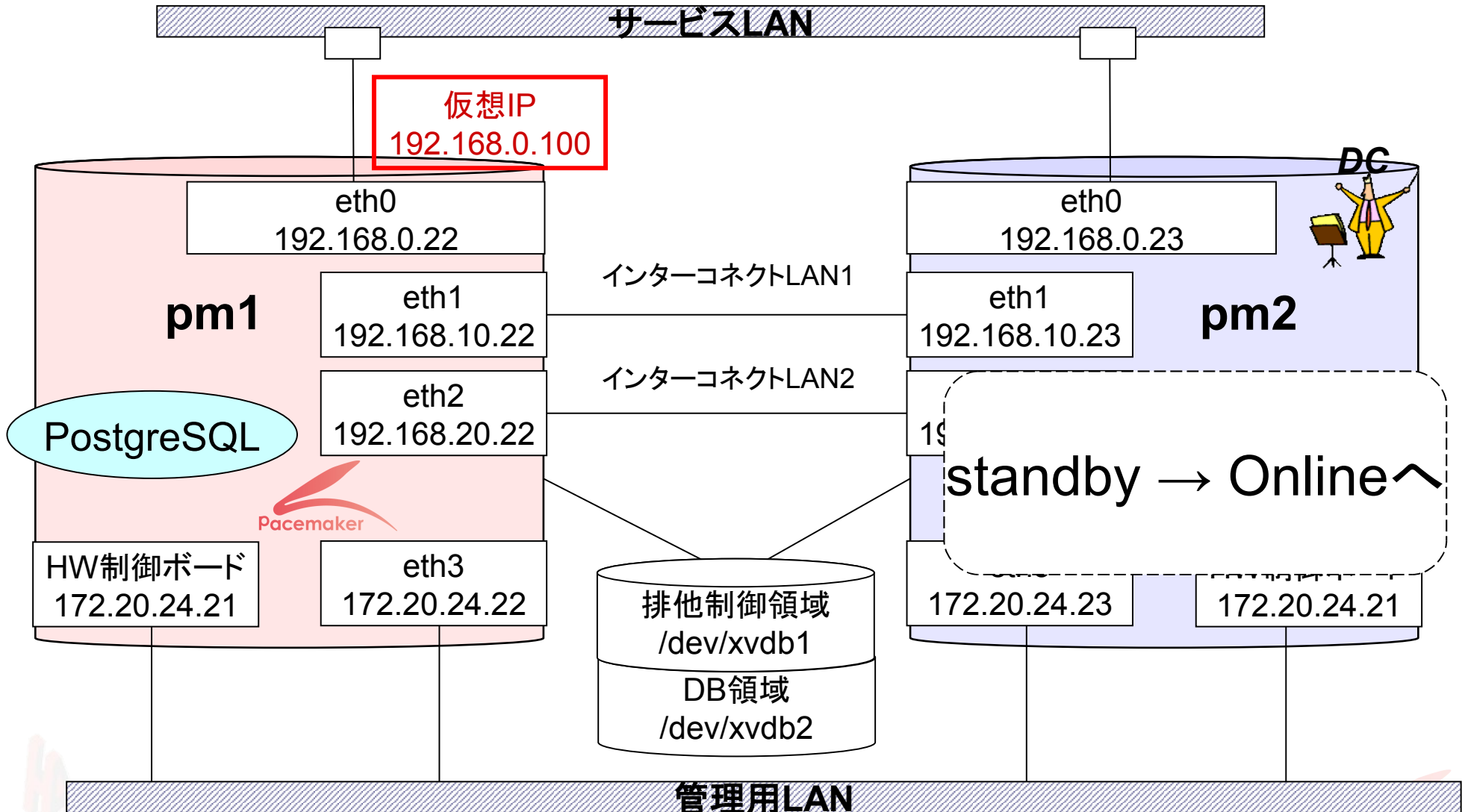
pm2をスタンバイ化してみる...

crm node standby pm2

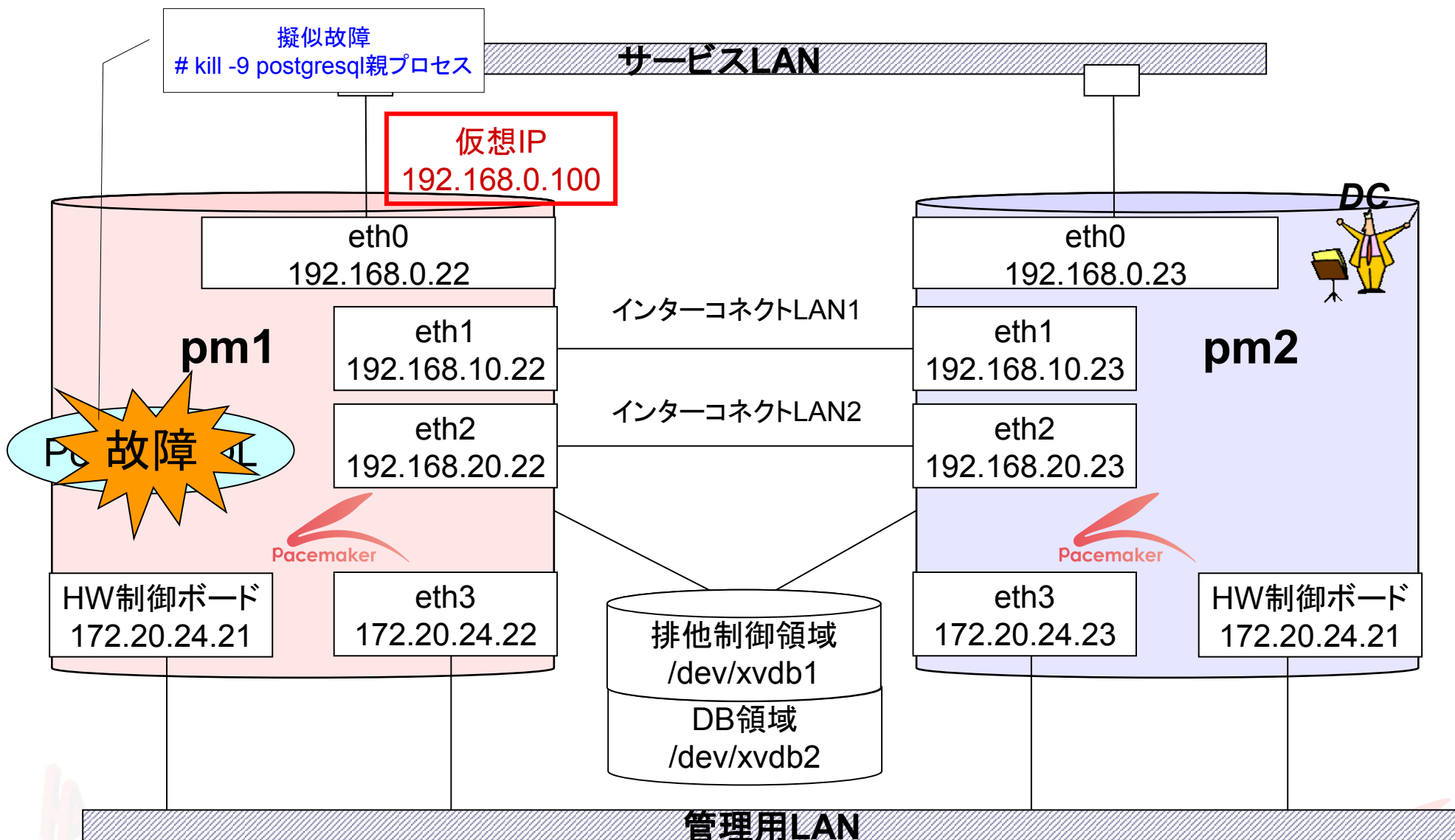


pm2をスタンバイ解除してみる...

crm node online pm2

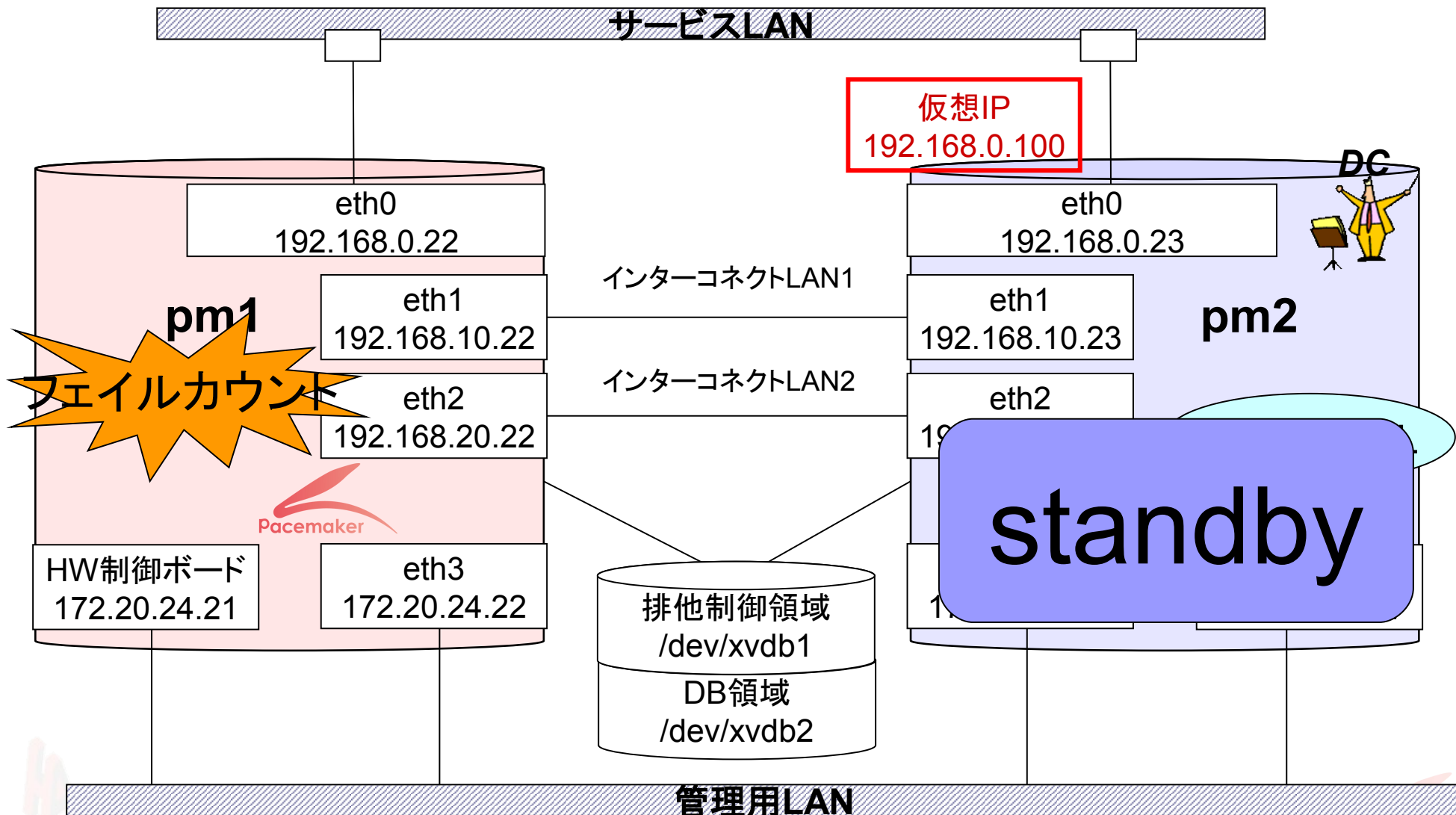


リソース故障させてみる…

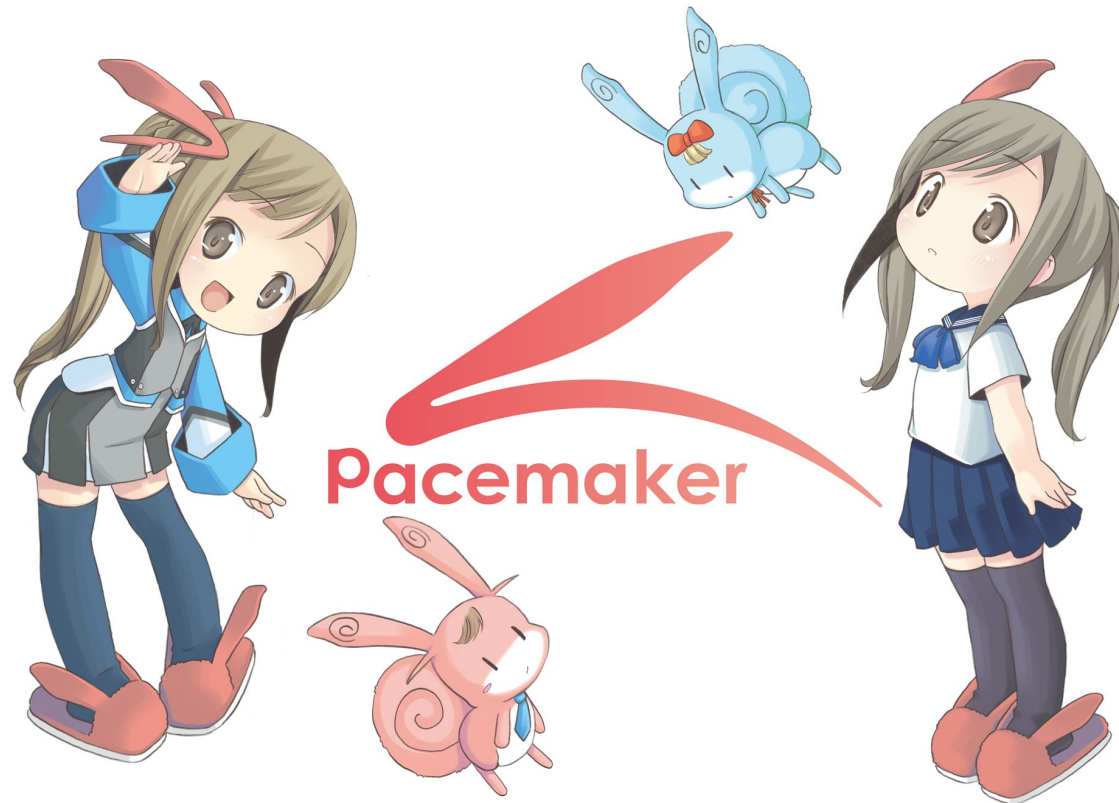


この状態でpm2をスタンバイしてみる…

```
# crm node standby pm2
```



切り替わらないのは ミスではありません！



フェイルカウントがカウントアップされているため、
クリアしなければ切り替わりません。

```
# crm_mon -fA
```

```
=====
```

```
～ 省略 ～
```

```
=====
```

Migration summary:

* Node pm1:

prnPg: migration-threshold=1 fail-count=1

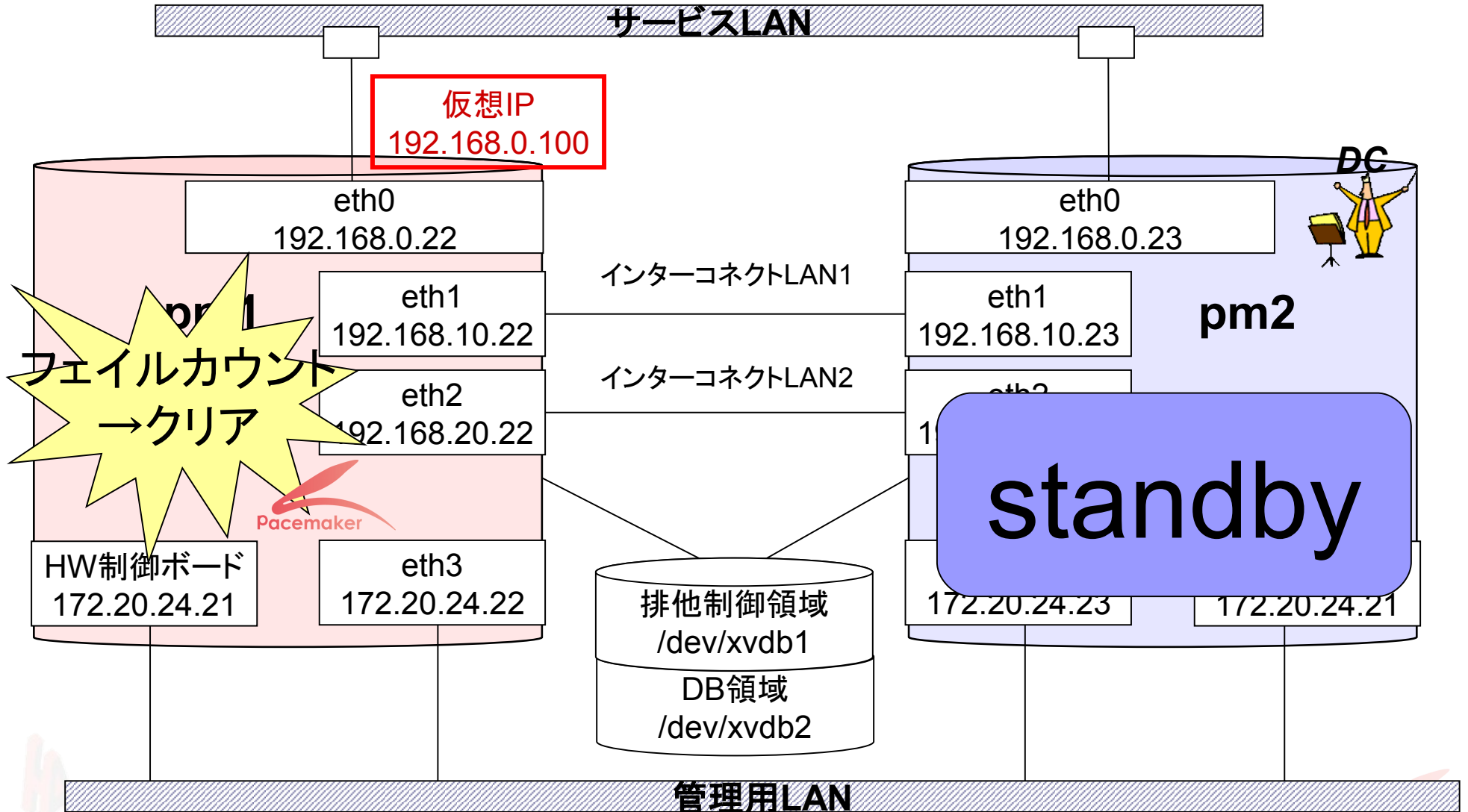
* Node pm2:

Failed actions:

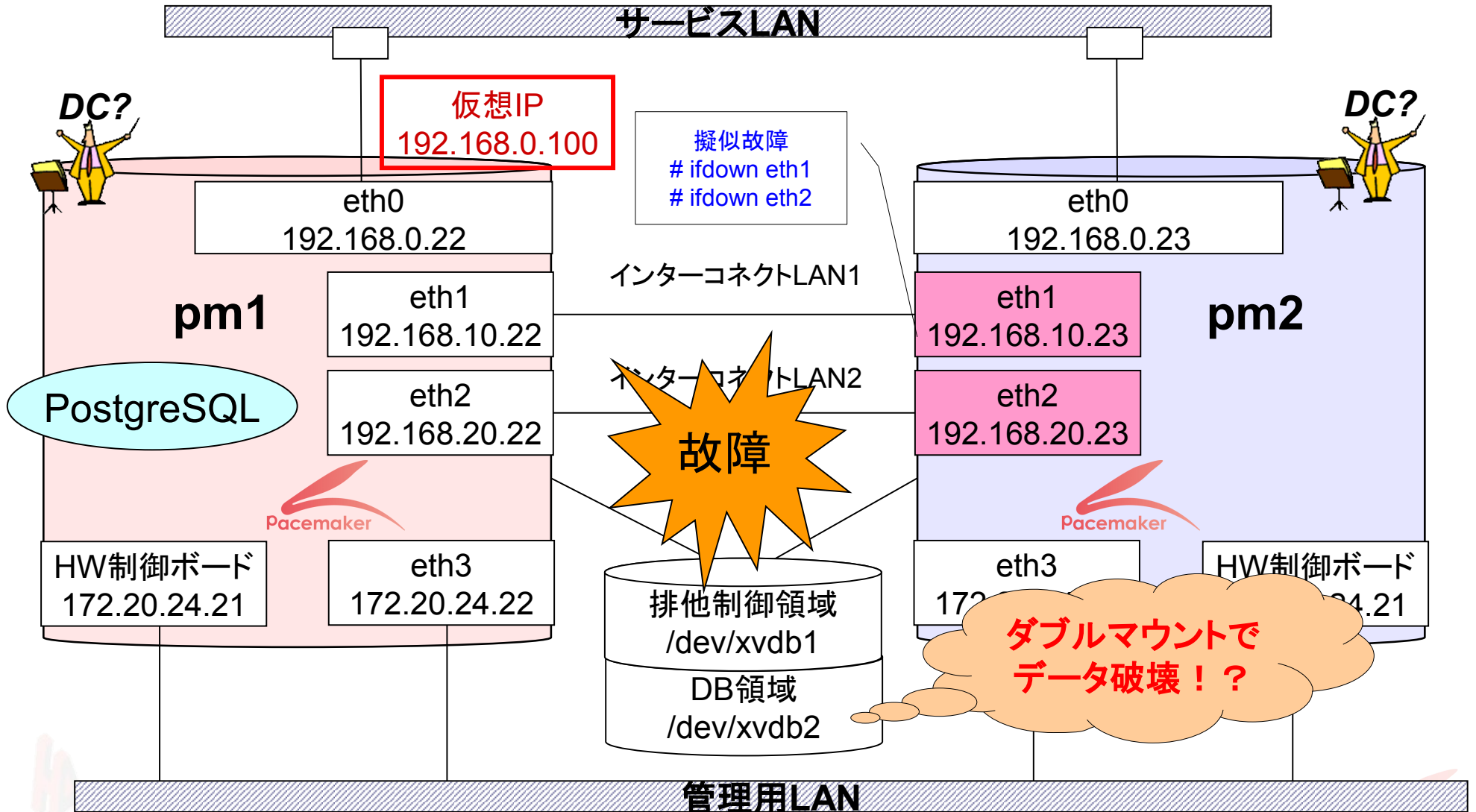
prnPg_monitor_10000 (node=pm1, call=34, rc=7, status=complete):
not running

フェイルカウトをクリアしてみる…

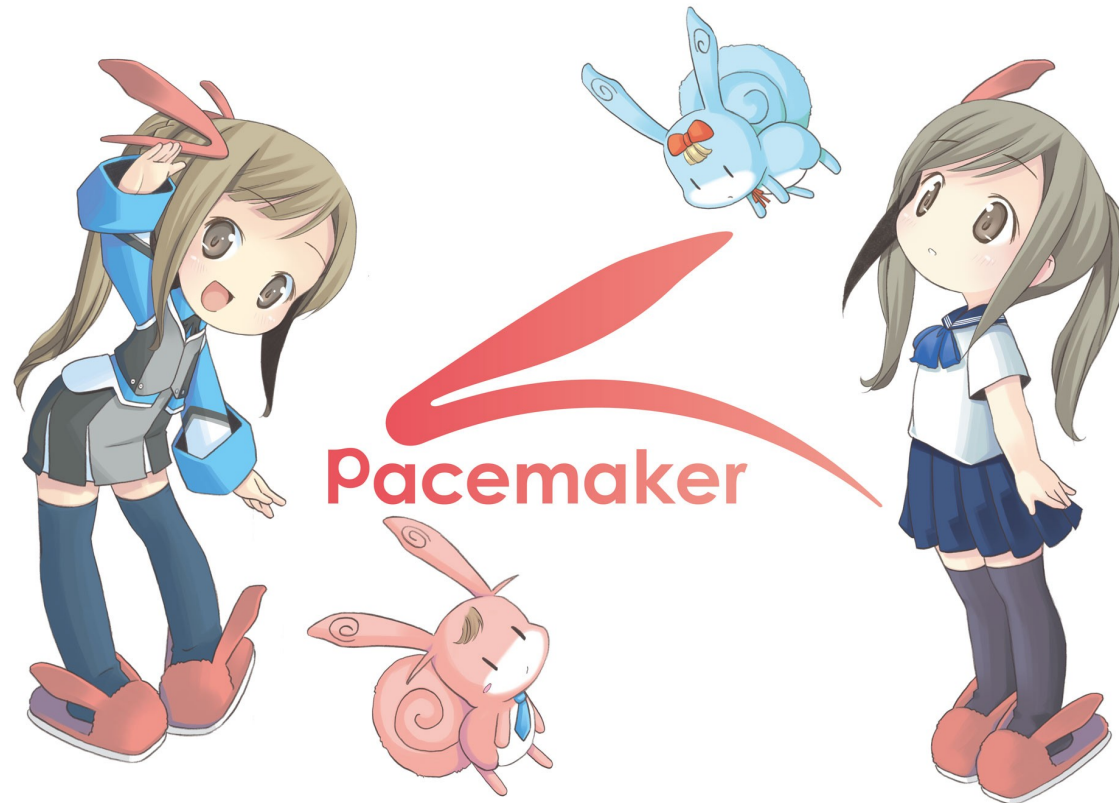
```
# crm resource cleanup prmPg pm1
```



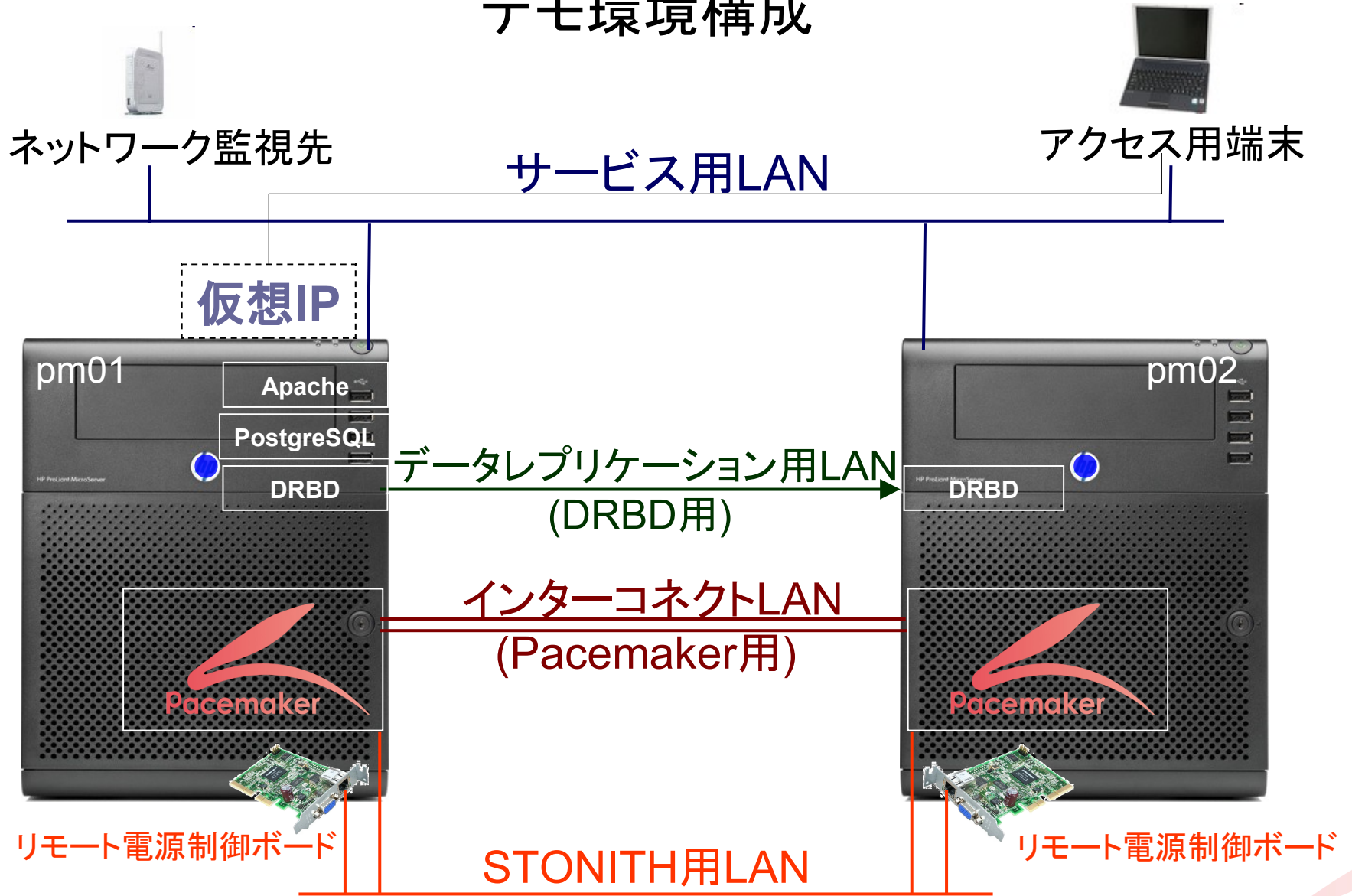
pm2 Online後に、スプリットブレインにしてみる…



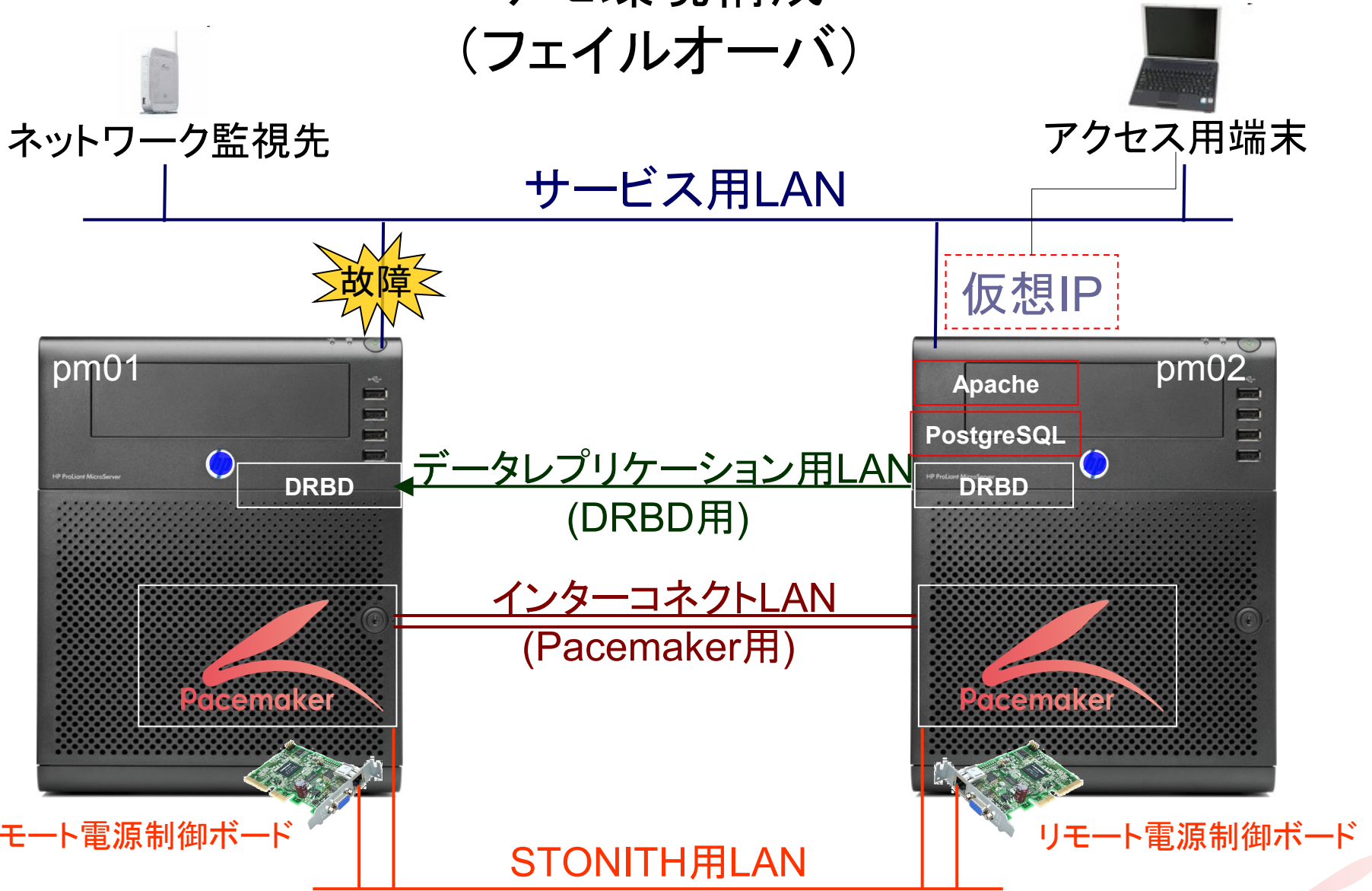
本日の展示会場ではこんな構成で Pacemakerのデモしています！



デモ環境構成



デモ環境構成 (フェイルオーバー)



デモ環境構成 (スプリットブレイン ⇒ STONITH)

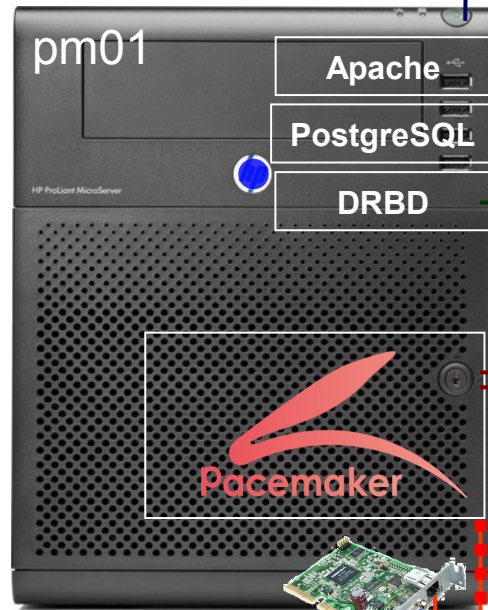


ネットワーク監視先

アクセス用端末

サービス用LAN

仮想IP



データレプリケーション用LAN
(DRBD用)

故障 インターコネクトLAN
(Pacemaker用)



リモート電源制御ボード

電源断

リモート電源制御ボード

STONITH用LAN

Linux-HA Japan Project