第3回 Linux-HA Japan勉強会

まずは、Pacemakerを使ってみよう！

2011年7月1日　@パソナグループ本部ビル

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、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
  - 道産子なので「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 本日のPacemakerデモ環境
② インストール・設定をデモします！
③ フェイルオーバ・系切り替えをデモします！

①

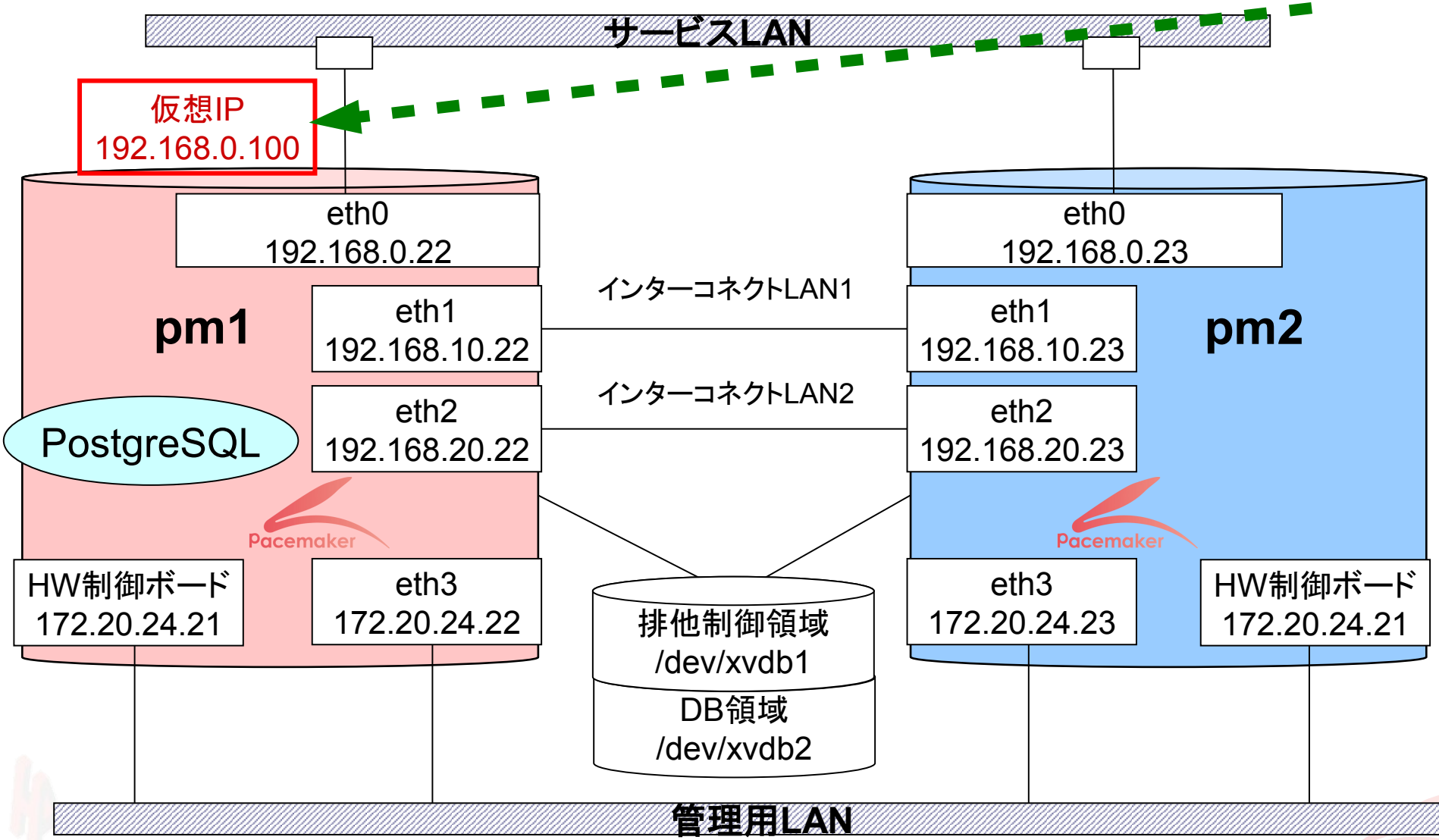
本日のPacemakerデモ環境

本日のPacemakerデモ環境

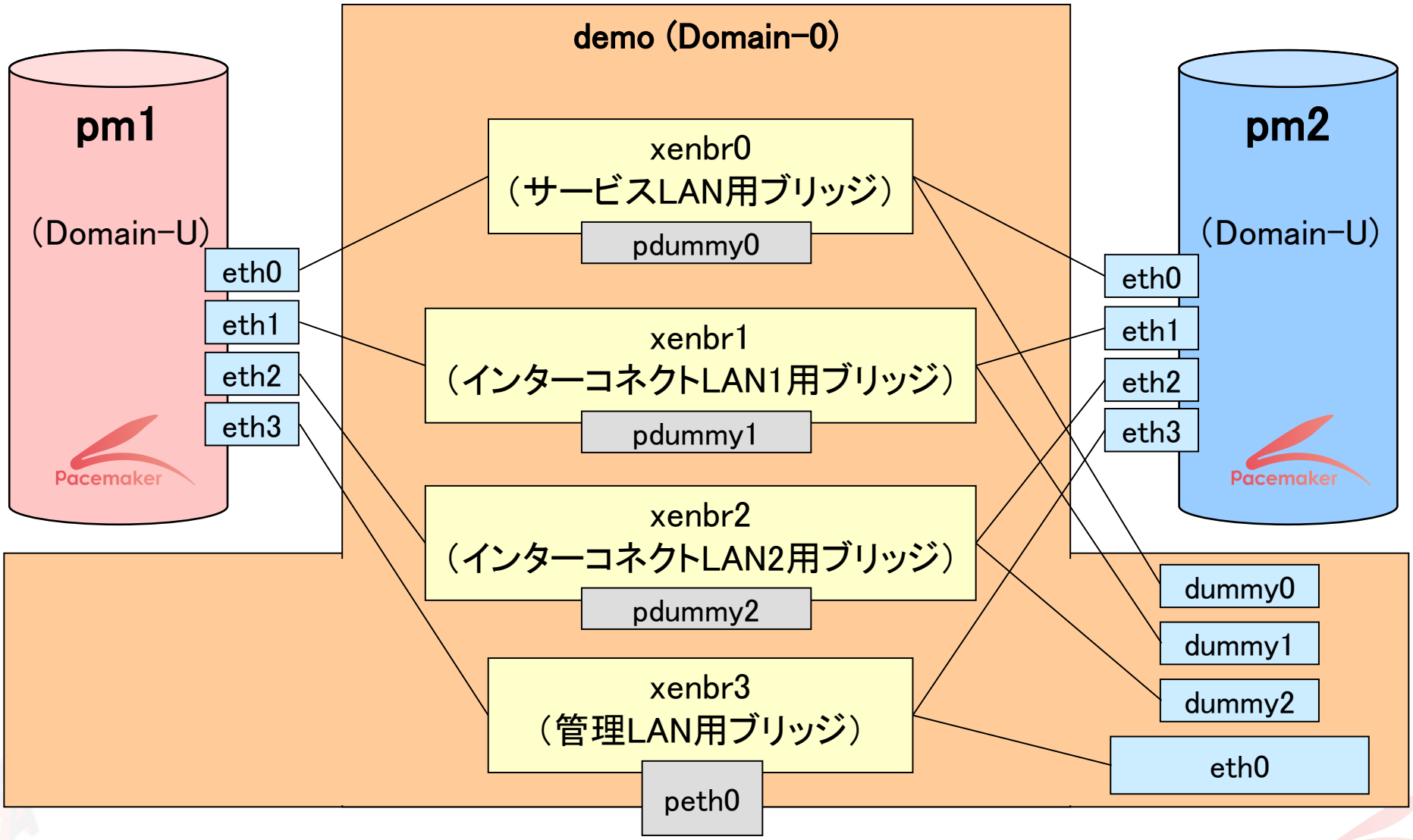
- ハードウェア
  - ノートPC（Core2Duo 2.26MHz、メモリ 2G）
- OS
  - CentOS 5.6 x86_64
- HAクラスタ
  - Pacemaker-1.0.11（インストールのデモを行います）
- クラスタ化するアプリケーション
  - PostgreSQL 9.0.4（インストール済み）
- 仮想環境
  - Xen（CentOS 5.6同梱版）
  - Domain-Uは2ドメインで構成
  - 各ドメインには、CPU×1・メモリ480M を割り当て

Pacemakerデモ構成

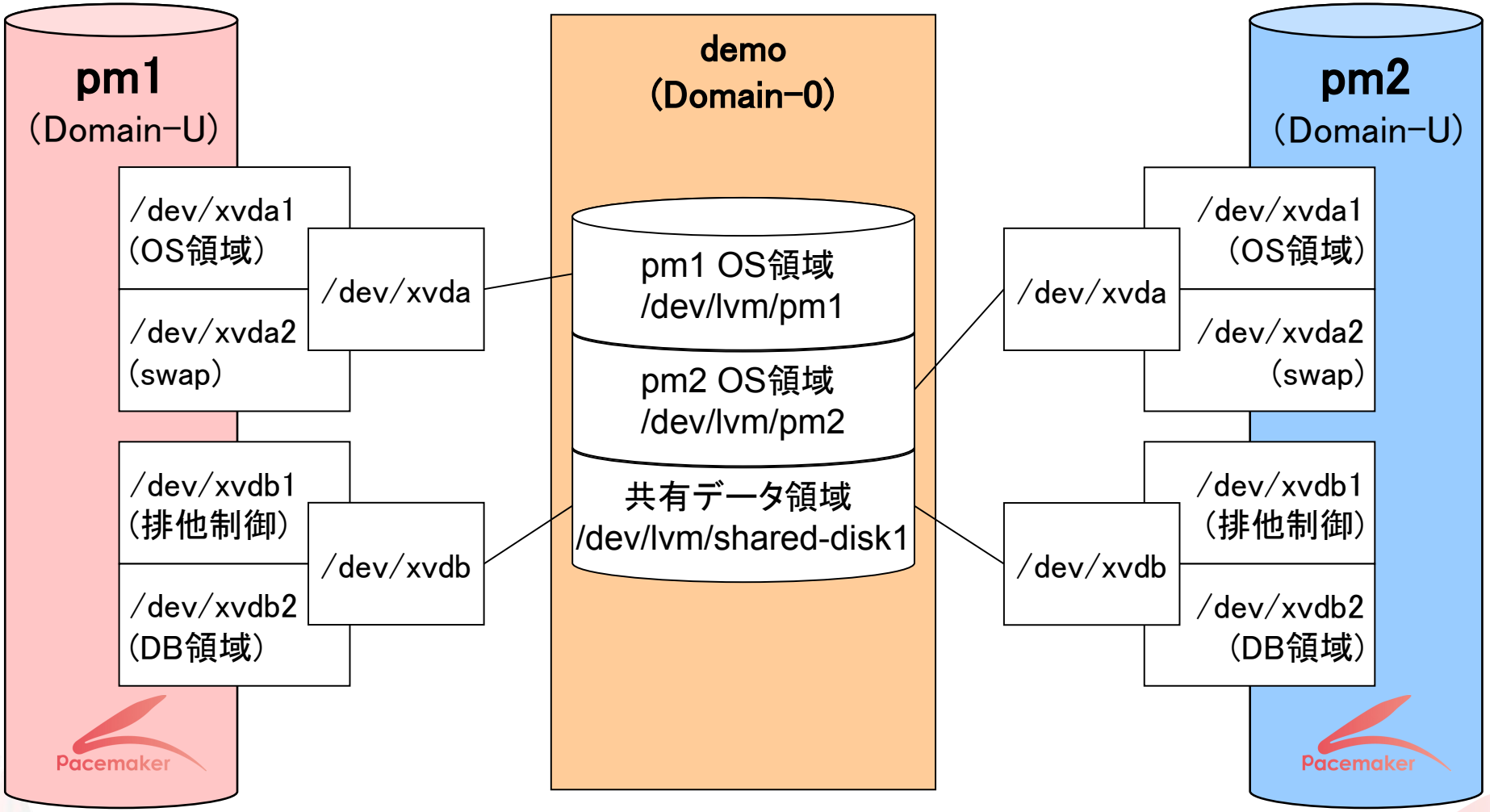
demo
(Domain-0)



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

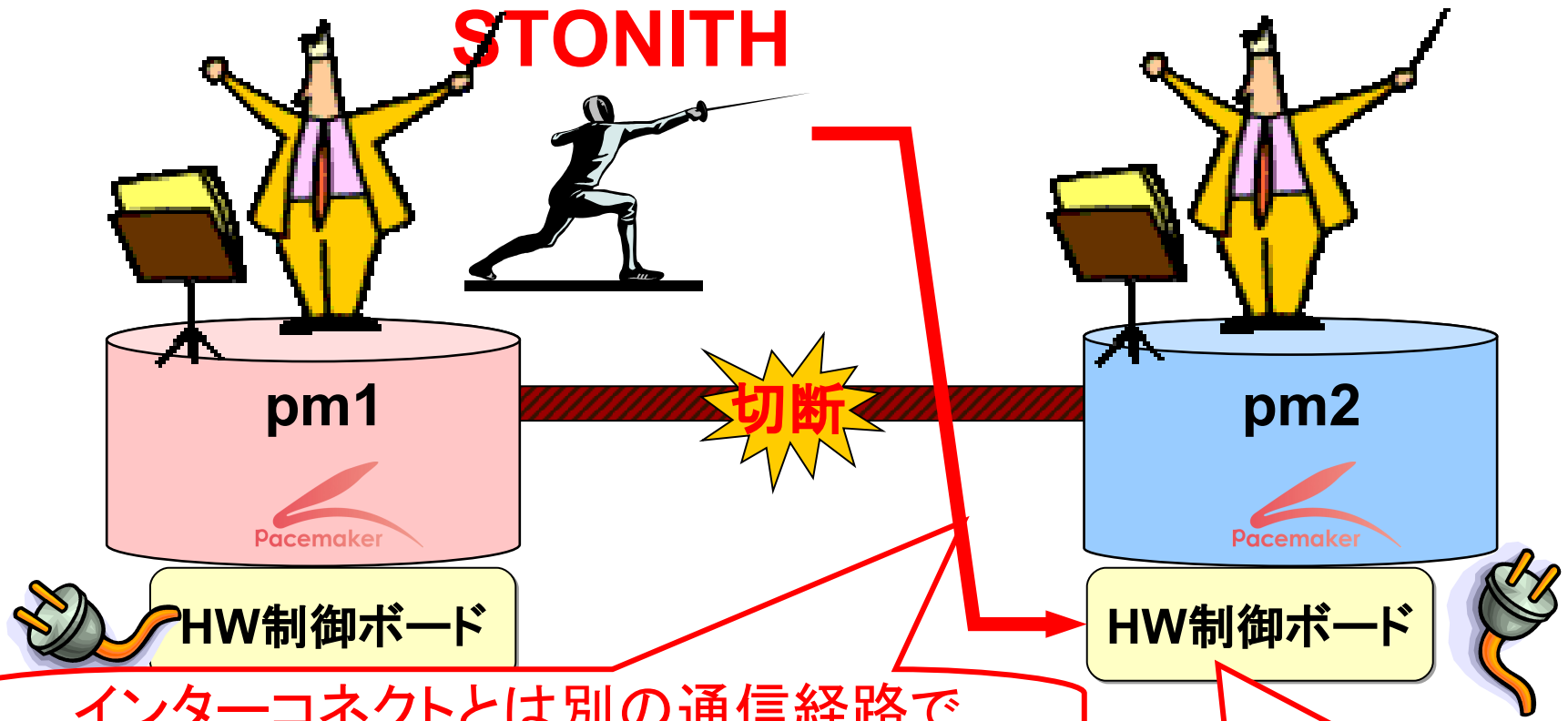
これら4つの
リソースは
グループ設定します

- 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

本日は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。

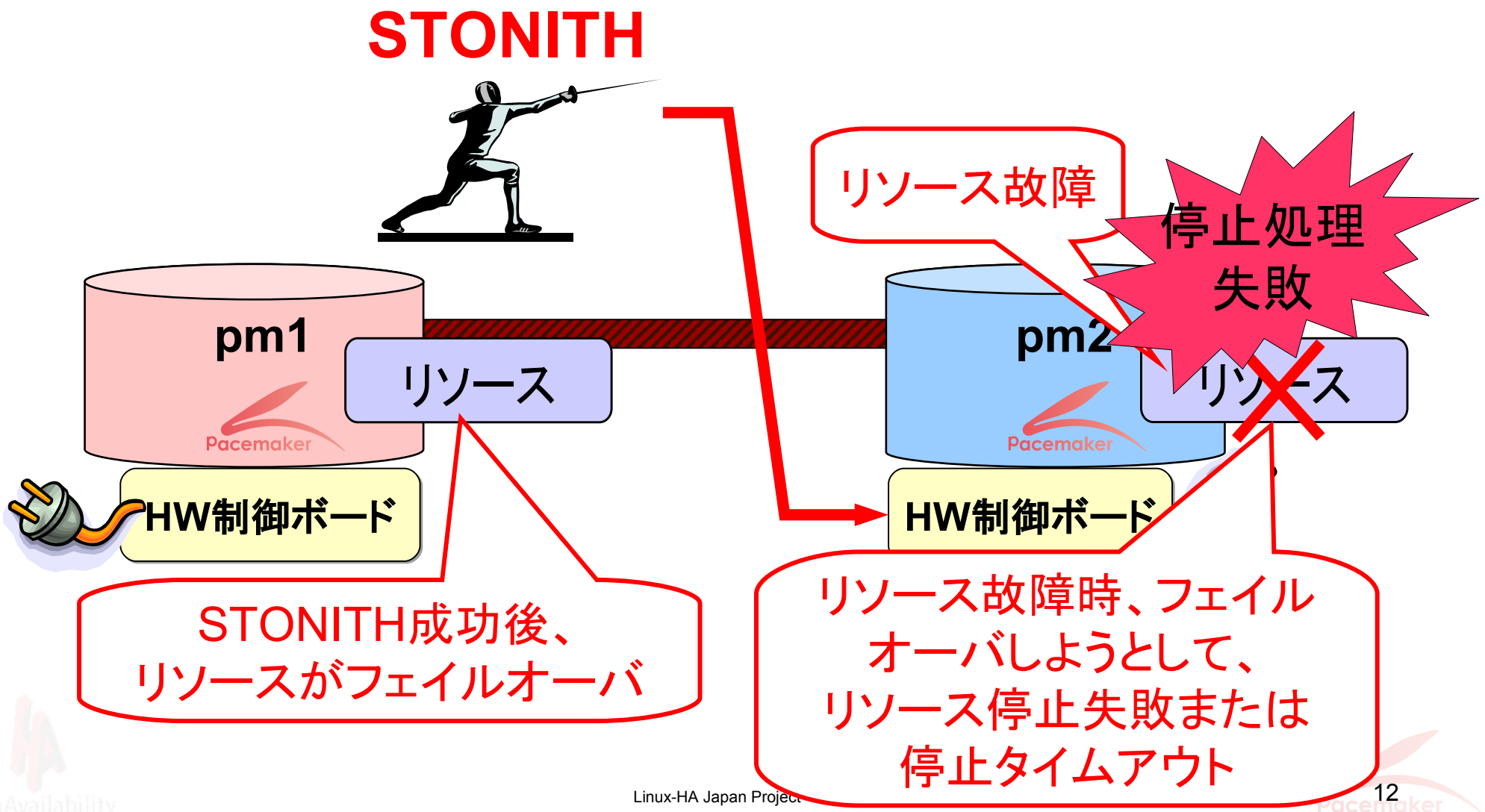
STONITH実行例（スプリットブレイン）



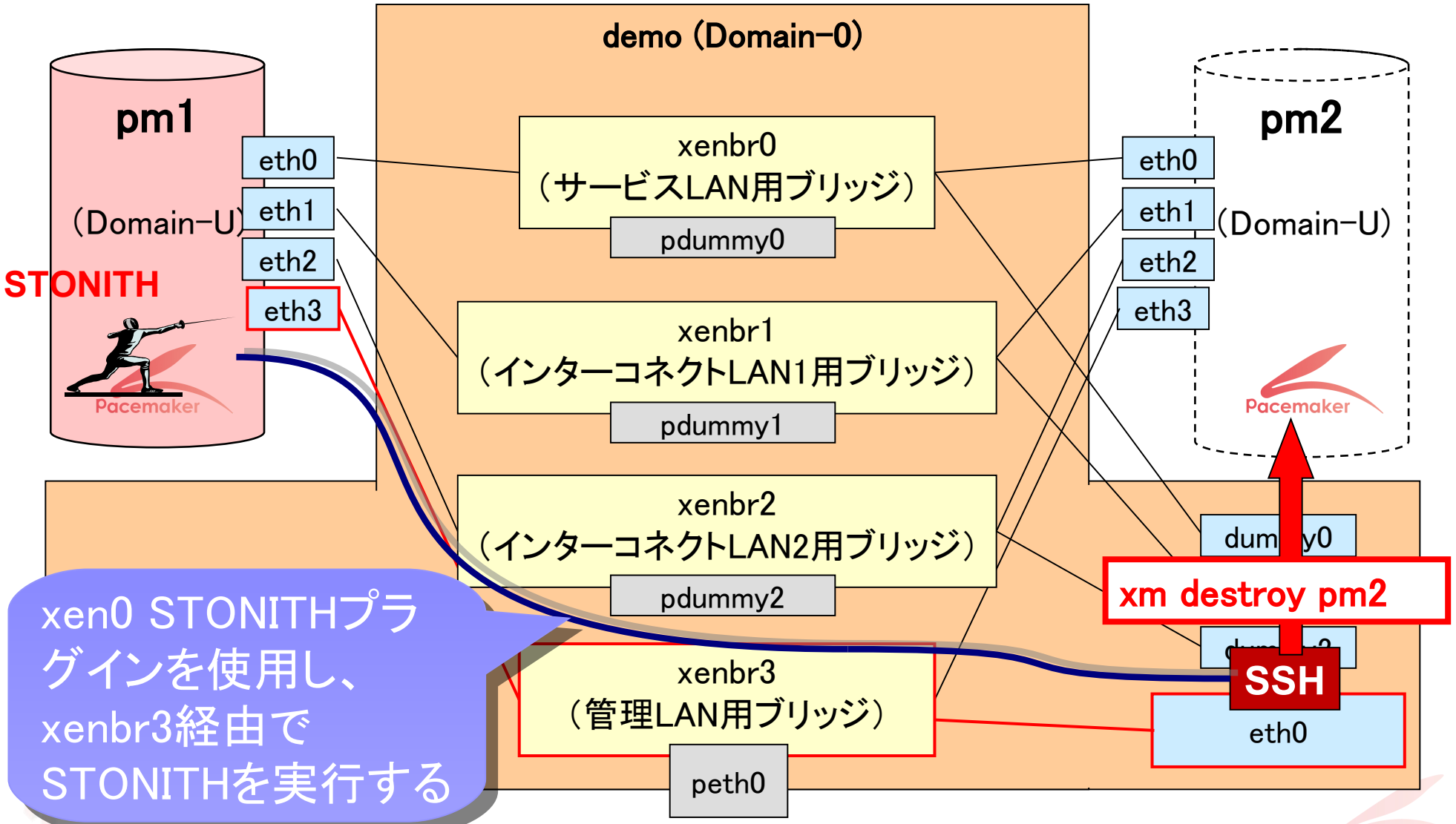
インターコネクトとは別の通信経路で
HW制御ボードに対しリセットを実行

OSと連動しないHW制御ボードから強制電源断

STONITH実行例(リソース停止失敗)



Pacemakerデモ機フェンシング（STONITH）構成



②

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-1.0.11-1.2.1.el5.x86_64.repo.tar.gz
をダウンロード

Pacemaker-1.0.11-1.2.1 版は 7月上旬リリース予定

2. yumでインストール！

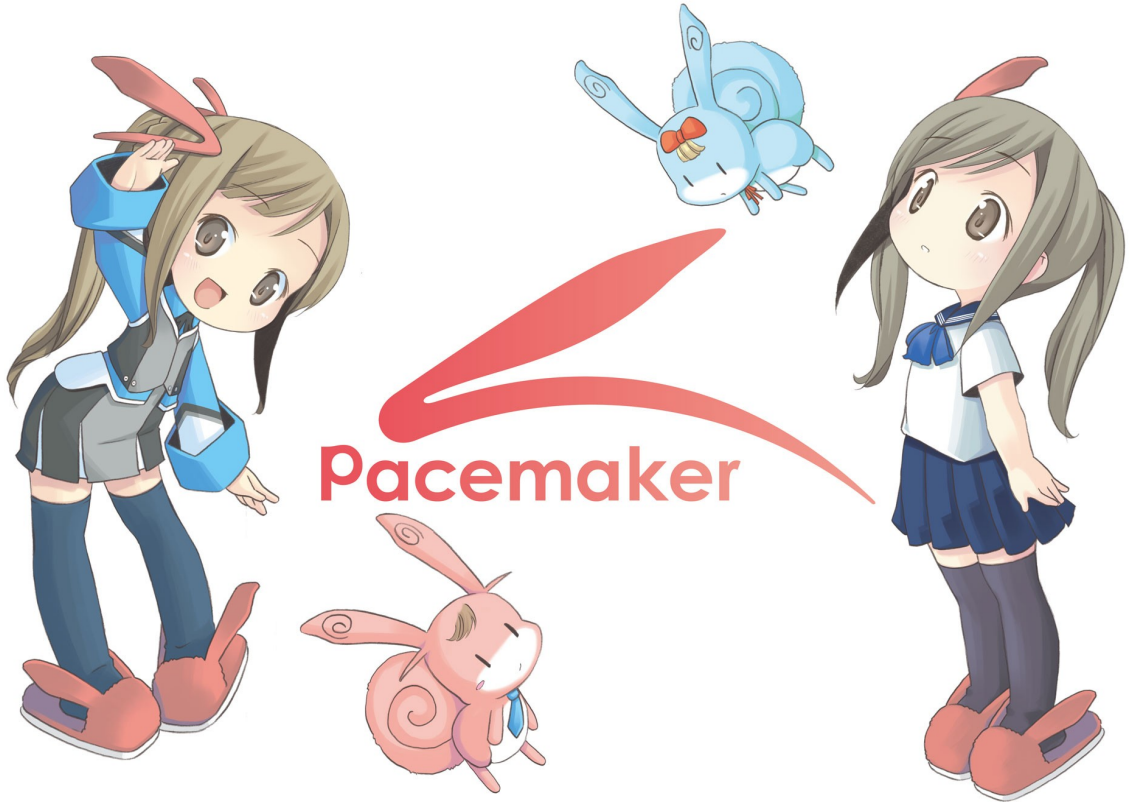
/tmp で展開し、yumコマンドでインストールします。

```
# cd /tmp
# tar zxvf pacemaker-1.0.11-1.2.1.el5.x86_64.repo.tar.gz
# cd /tmp/pacemaker-1.0.11-1.2.1.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！

ここでやっと
Pacemakerインストールを
デモします！



/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

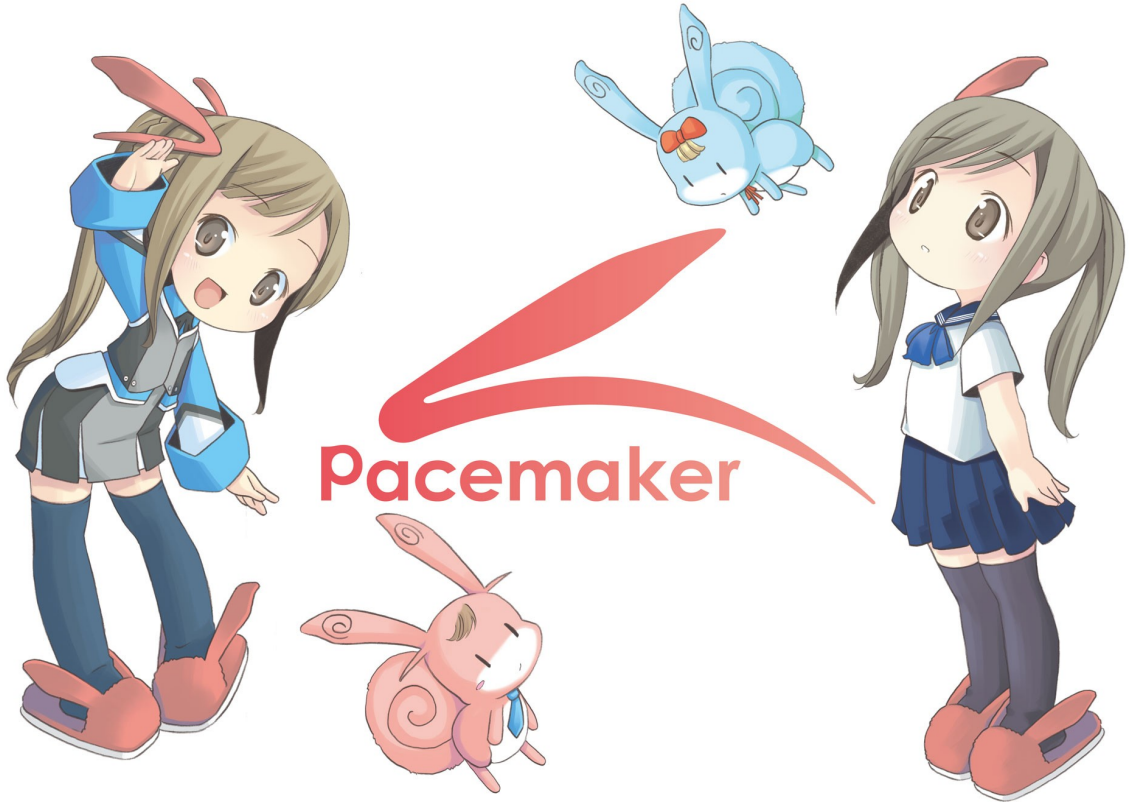
```
*.info;mail.none;authpriv.none;cron.none;local1.none
/var/log/messages
                :
          （省略）
                :
local1.info                                    /var/log/ha-log
```

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: Fri Jun 24 11:51:30 2011
Stack: Heartbeat
Current DC: pm2 (2d9dccc6-e3db-486c-b028-15fea6bc9567) - partition with
quorum
Version: 1.0.11-1554a83db0d3c3e546cfd3aaff6af1184f79ee87
2 Nodes configured, unknown expected votes
0 Resources configured.
============

Online: [ pm1 pm2 ]
```

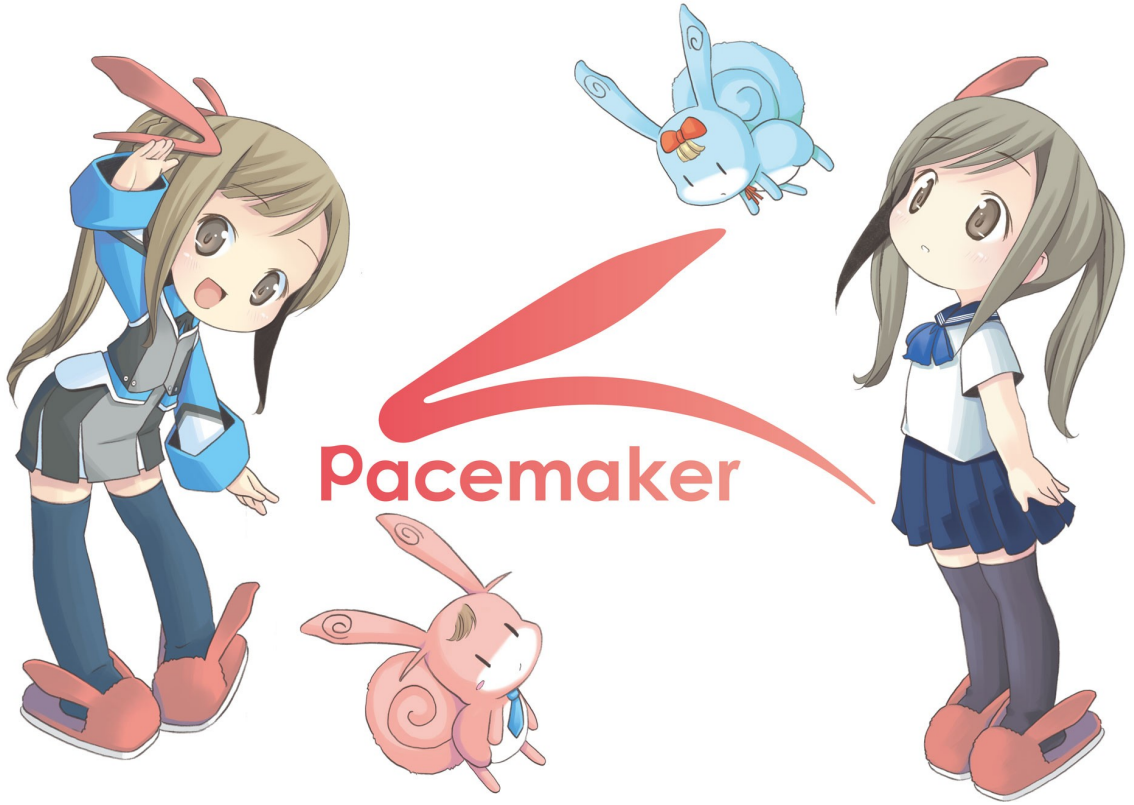
クラスタに組み込まれている
サーバ名が表示されます

-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: [ pm1 pm2 ]

Node Attributes:
* Node pm1:
    + pm2-eth1                  : up
    + pm2-eth2                  : up
* Node pm2:
    + pm1-eth1                  : up
    + pm1-eth2                  : up
```

インターコネクトがUPされているのが確認可能

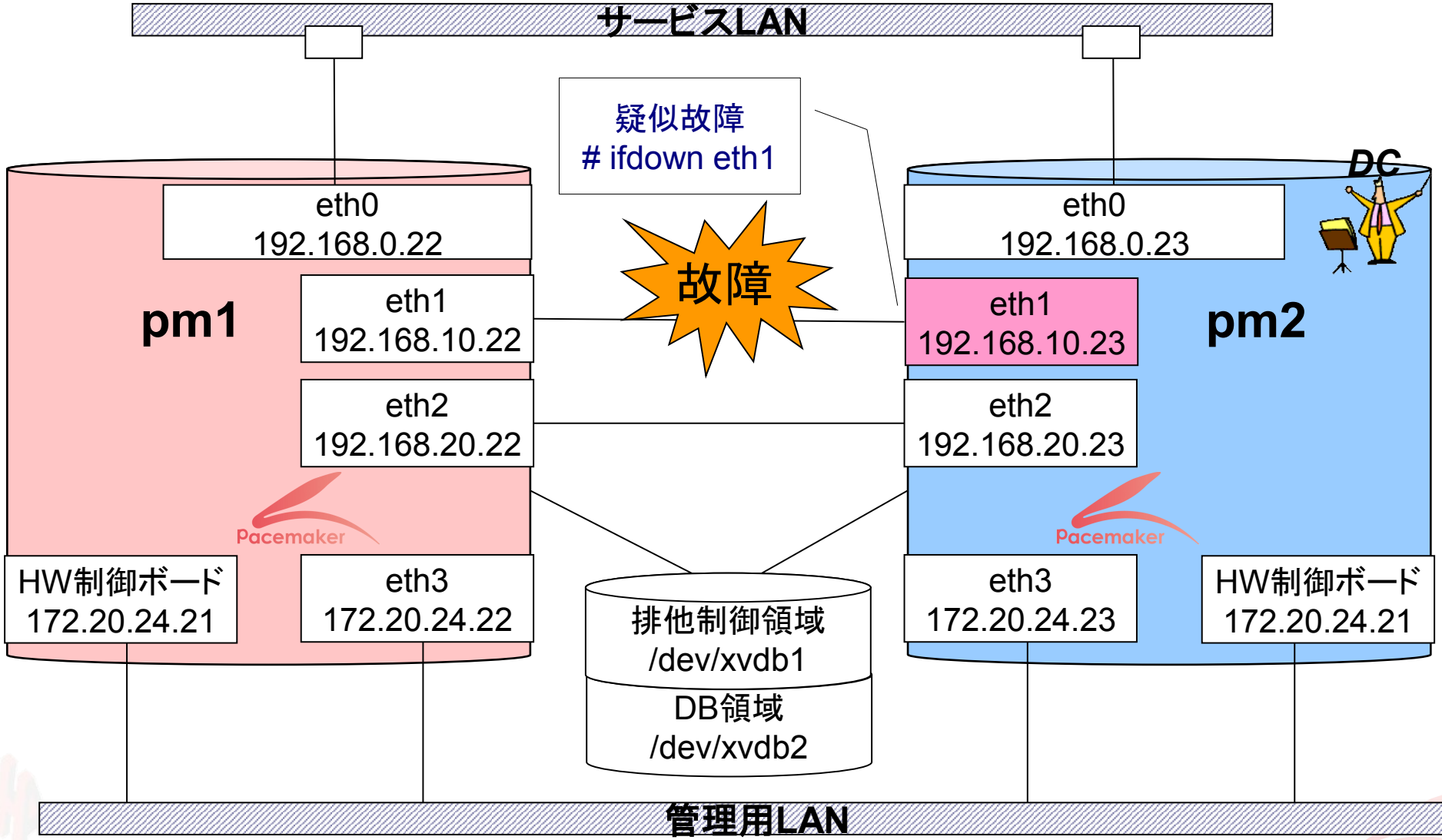
ここで、Pacemaker状態表示とインターコネクトLAN故障をデモします！



デモ例は
次ページ

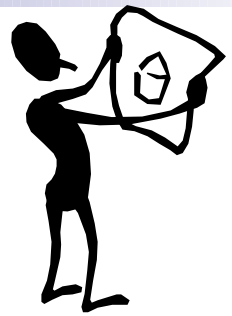
デモ1

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？

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。

83	#表 5-3 クラスタ設定　...　Primitiveリソース (id=prmIp)					
84	PRIMITIVE					
85	P	id	class	provider	type	
86	#	リソースID	class	provider	type	
87		prmIp	ocf	heartbeat	IPaddr2	
88	A	type	name	value		
89	#	パラメータ種別	項目	設定内容		
90		params	ip	192.168.0.100		
91			nic	eth0		
92			cidr_netmask	24		
93	O	type	timeout	interval	on-fail	start-delay
94	#	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
95		start	60s	0s	restart	
96		monitor	60s	10s	restart	
97		stop	60s	0s	fence	

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！

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。

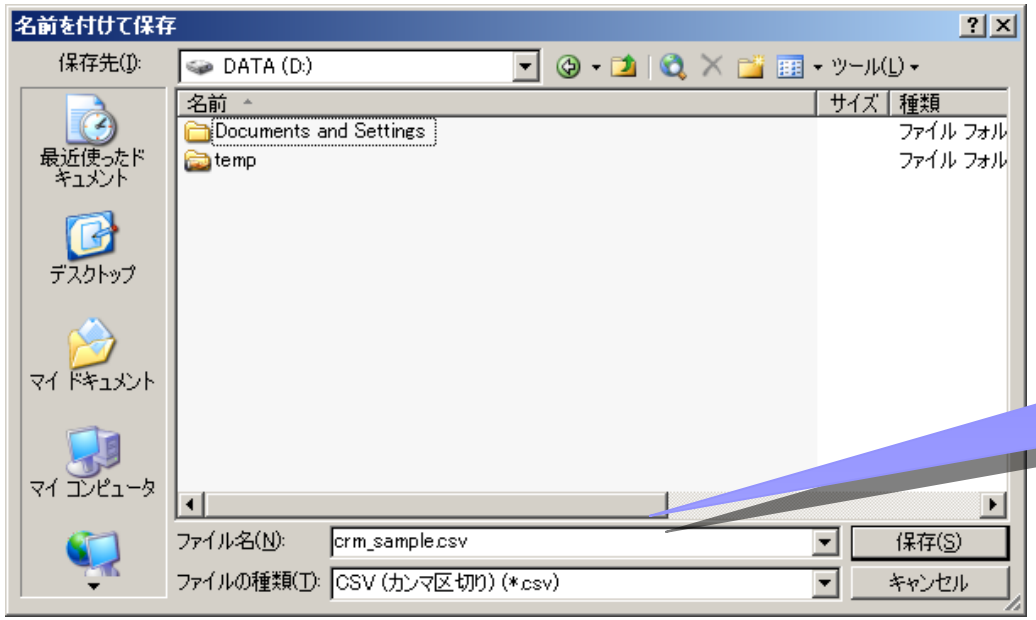
```
270 #表 6-1 クラスタ設定　...　リソース配置制約
271   LOCATION
272     rsc          score:200   score:100   score:-inf
273 #   リソースID   Activeノード  Standbyノー  非稼働ノード
274     grpPg        pm1         pm2
275     grpStonith1                          pm1
276     grpSt                                pm2
```

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



「crm_sample.csv」などとしてCSV形式で保存

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_crmgenコマンドでcrmファイルを生成

```
# pm_crmgen –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

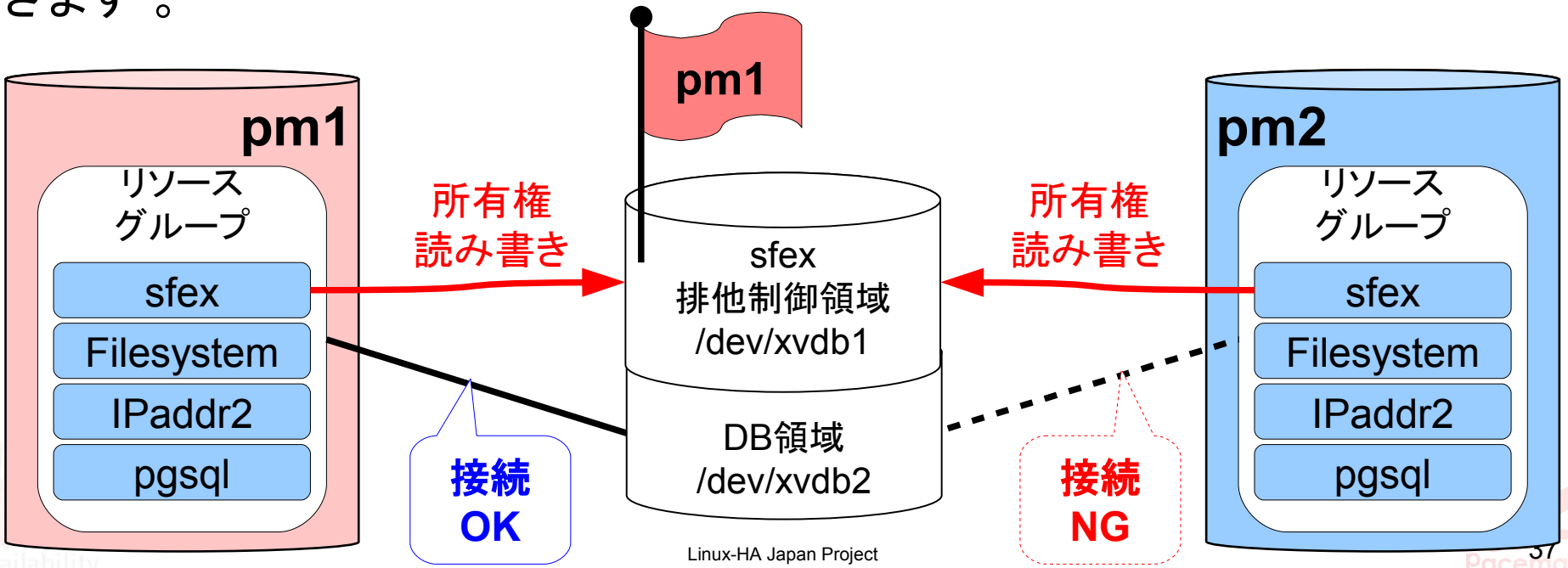
5）crmコマンドを実行してリソース設定を反映

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

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

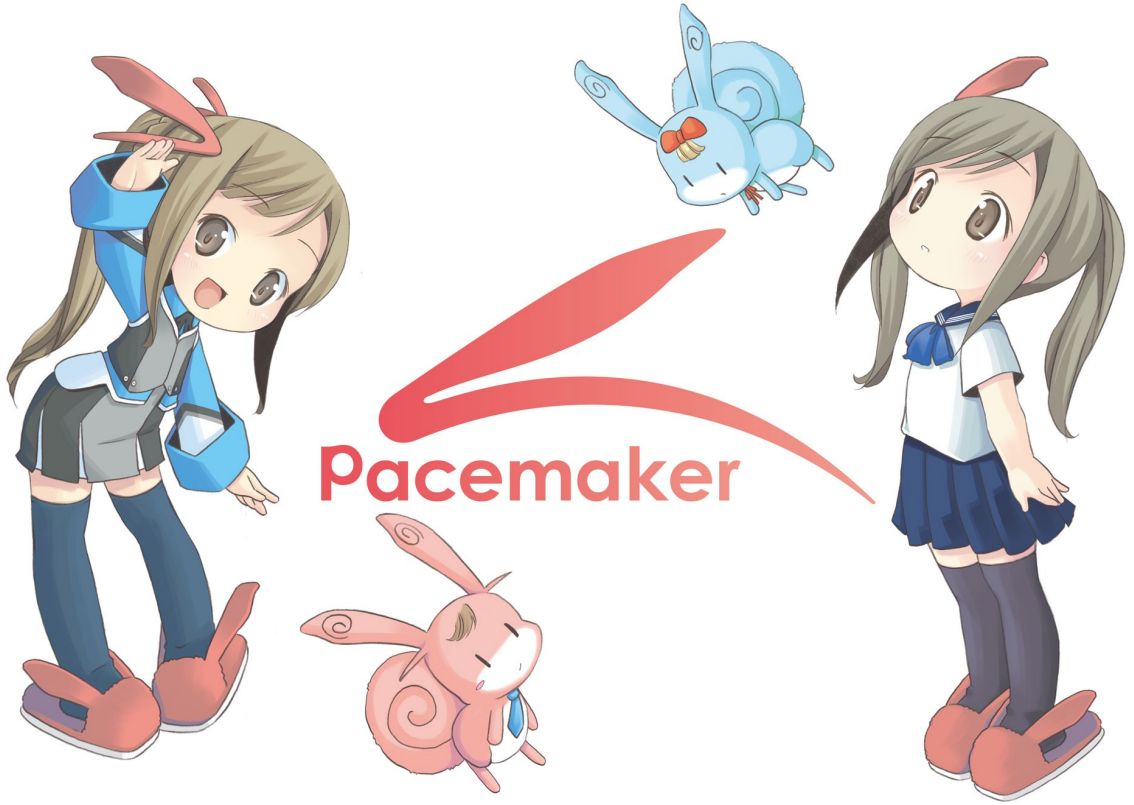
共有ディスク排他制御機能(sfex)を使用するためには、排他制御領域を初期化する必要性があります。

```
# 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！

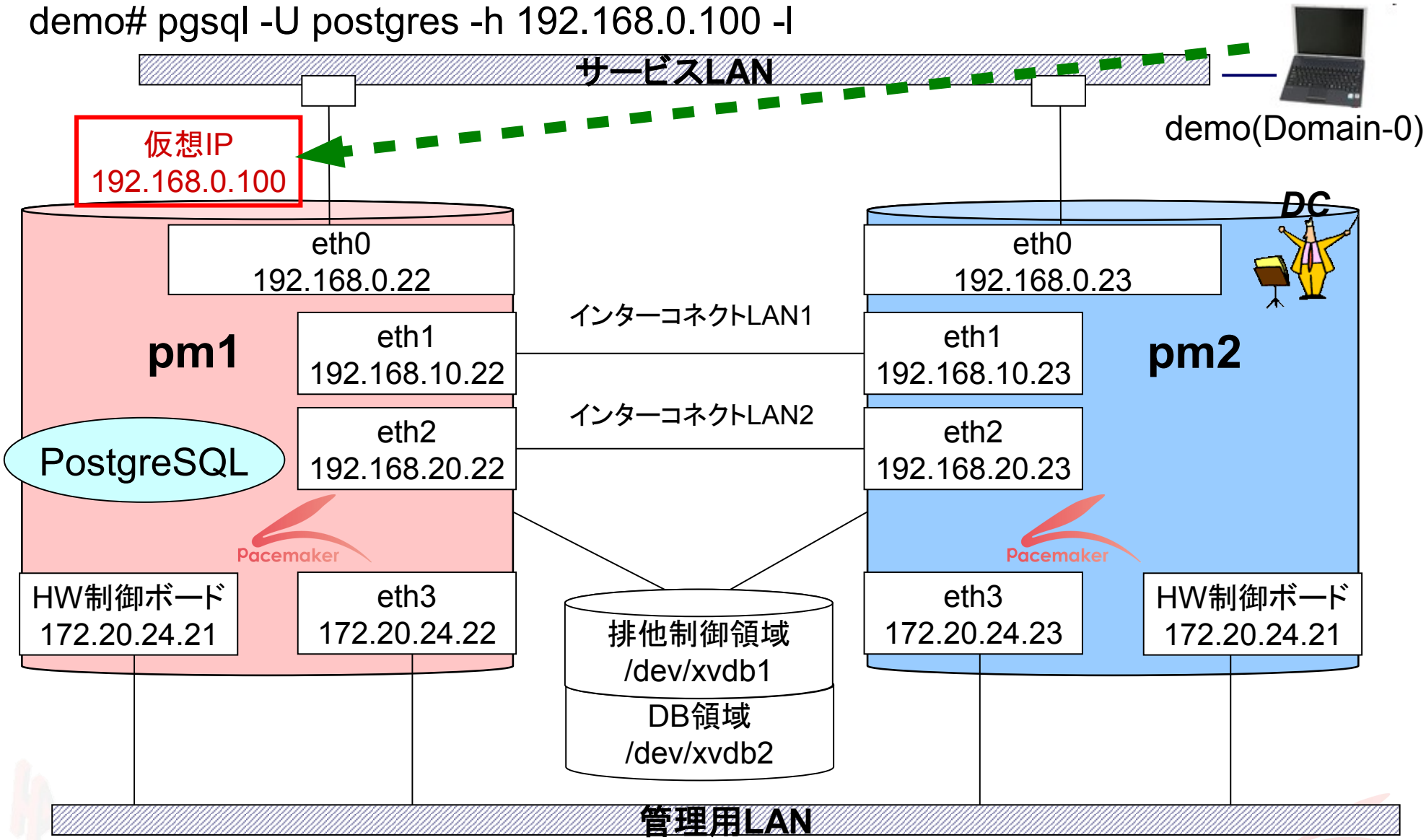


デモ例は
次ページ

PostgreSQLに接続してみる…

デモ2

demo# pgsql -U postgres -h 192.168.0.100 -l

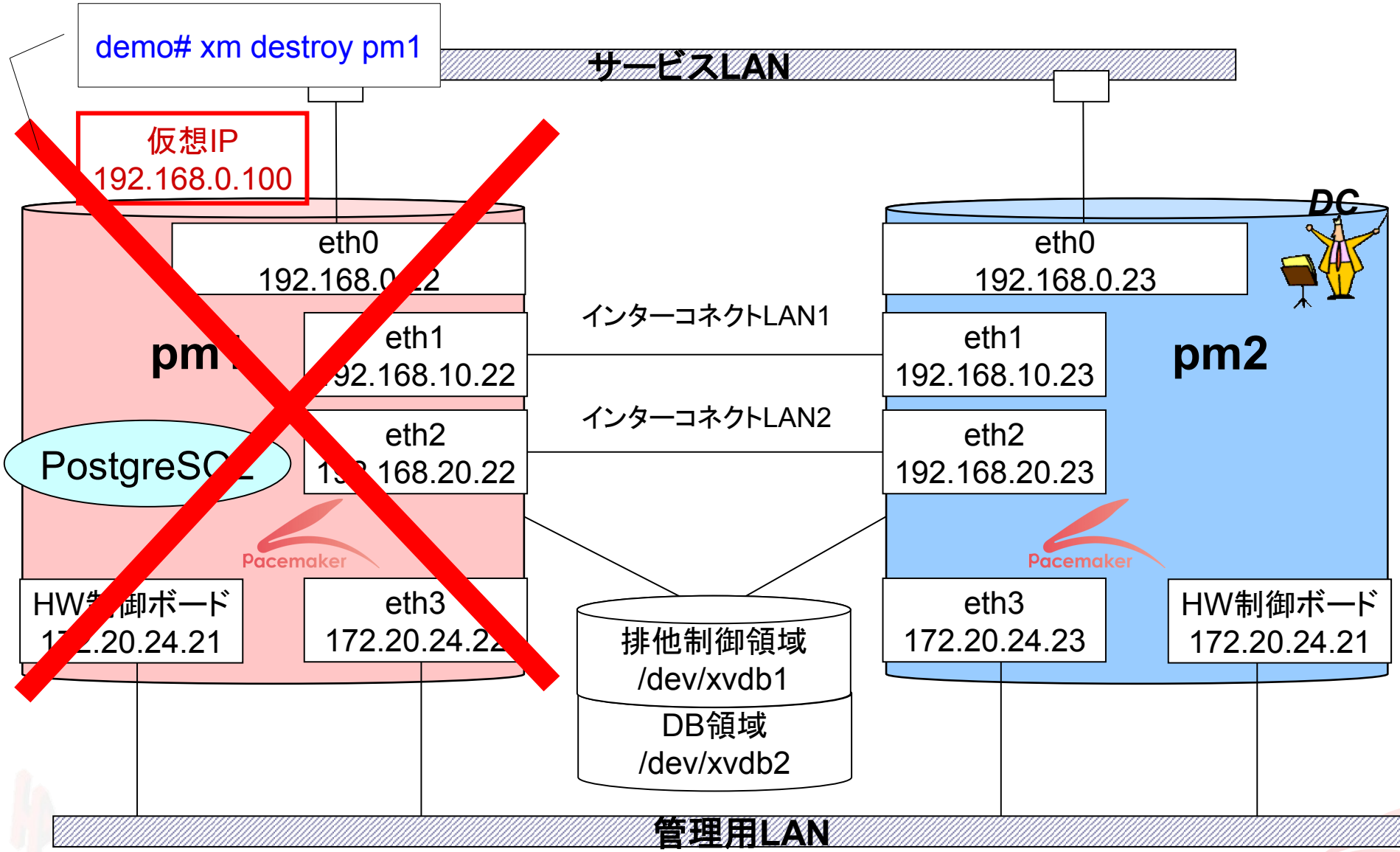


③

フェイルオーバ・系切り替えをデモします！

pm1を強制停止してみる…

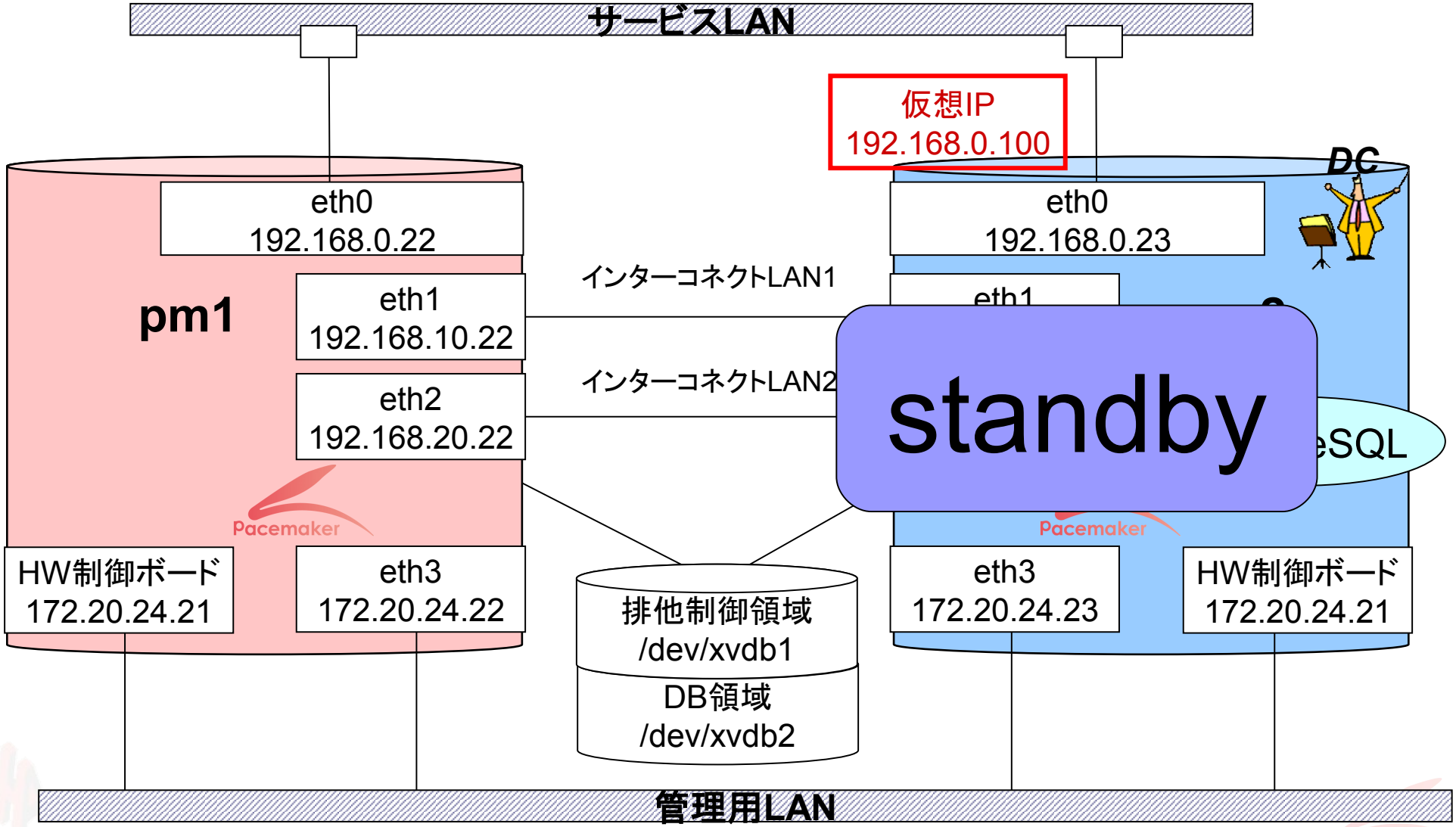
デモ3



pm2をスタンバイ化してみる…

デモ4

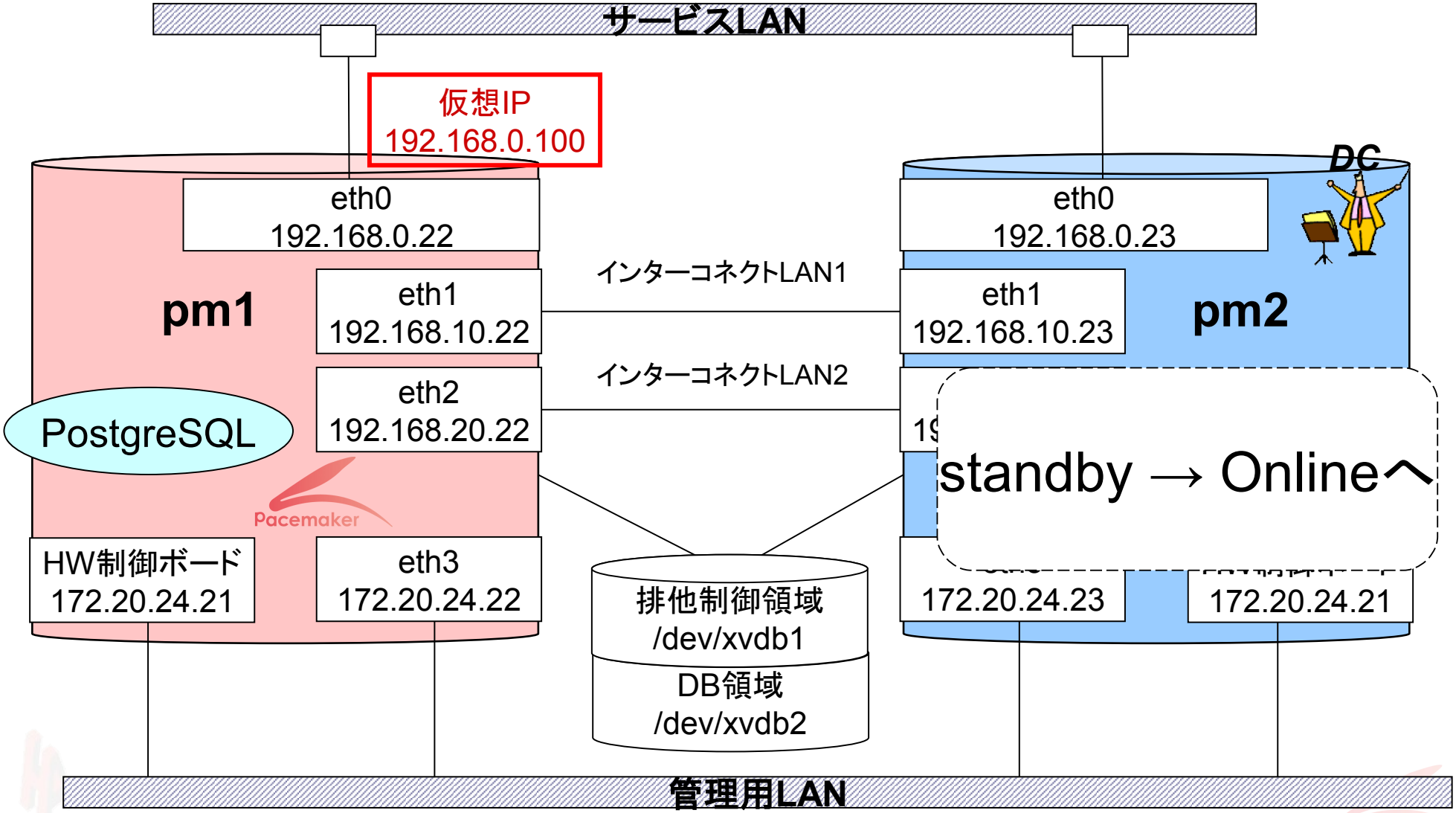
```
# crm node standby pm2
```



pm2をスタンバイ解除してみる…

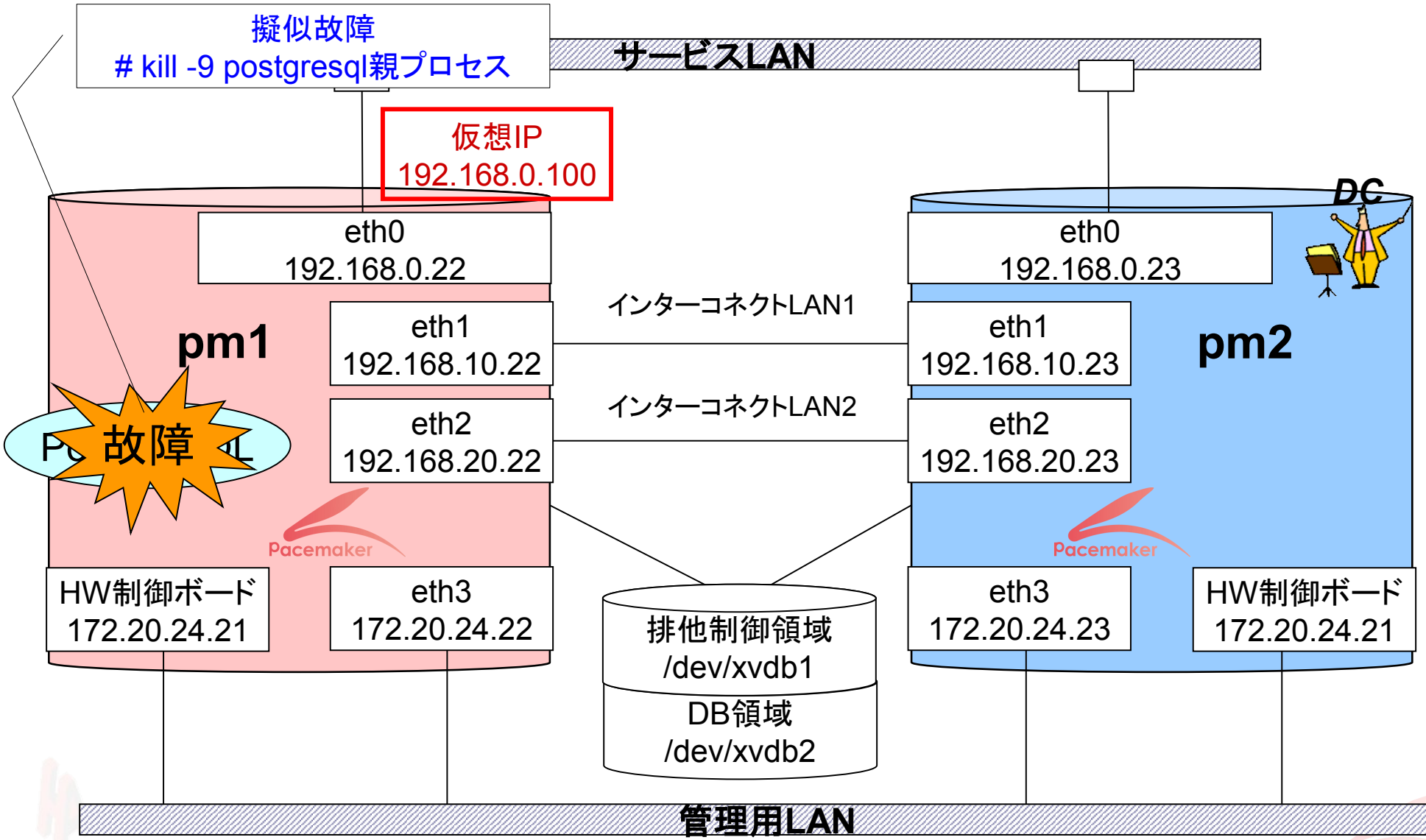
デモ5

```
# crm node online pm2
```



リソース故障させてみる…

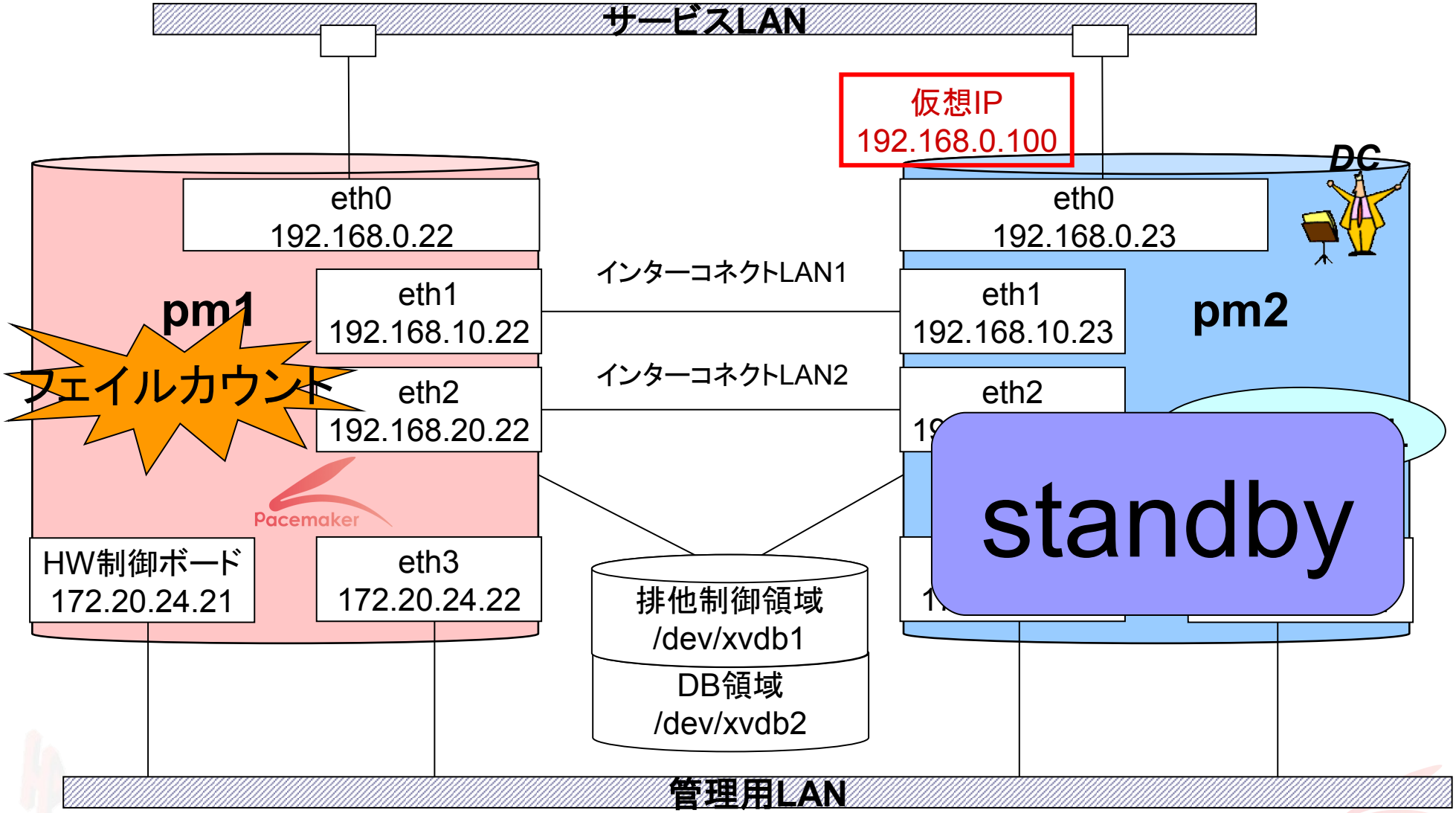
デモ6



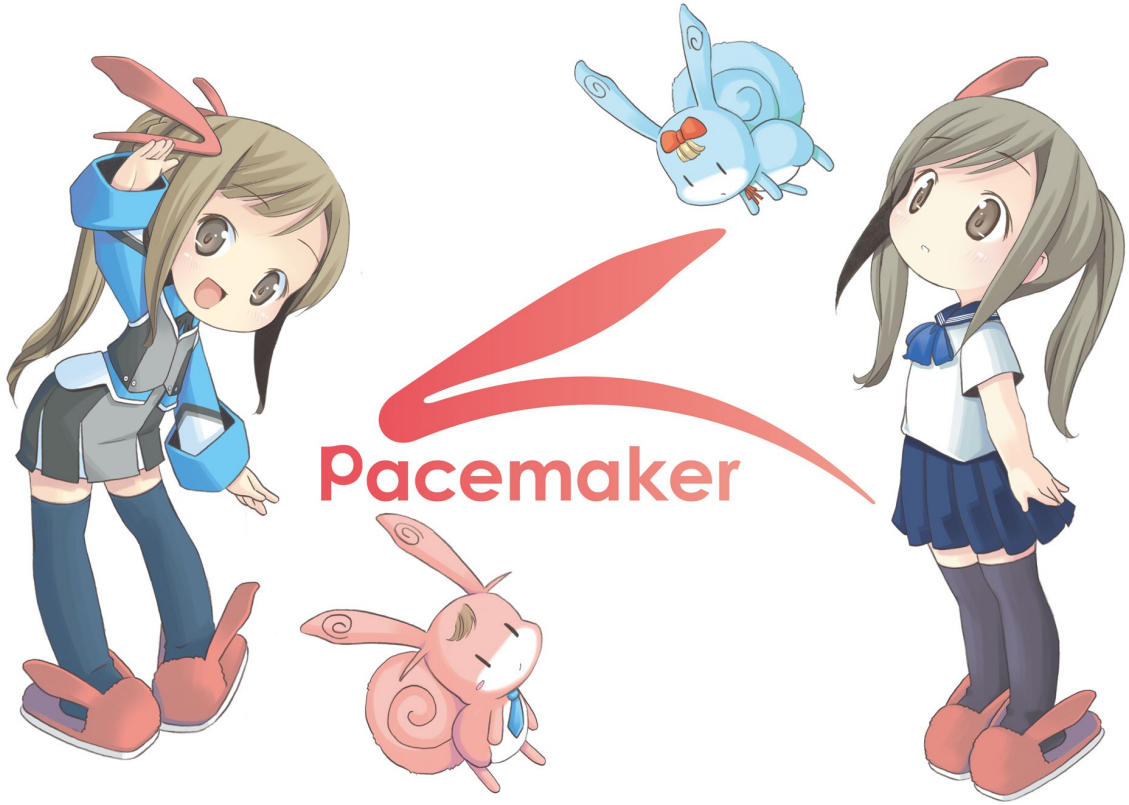
この状態でpm2をスタンバイしてみる…

デモ7

```
# 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。

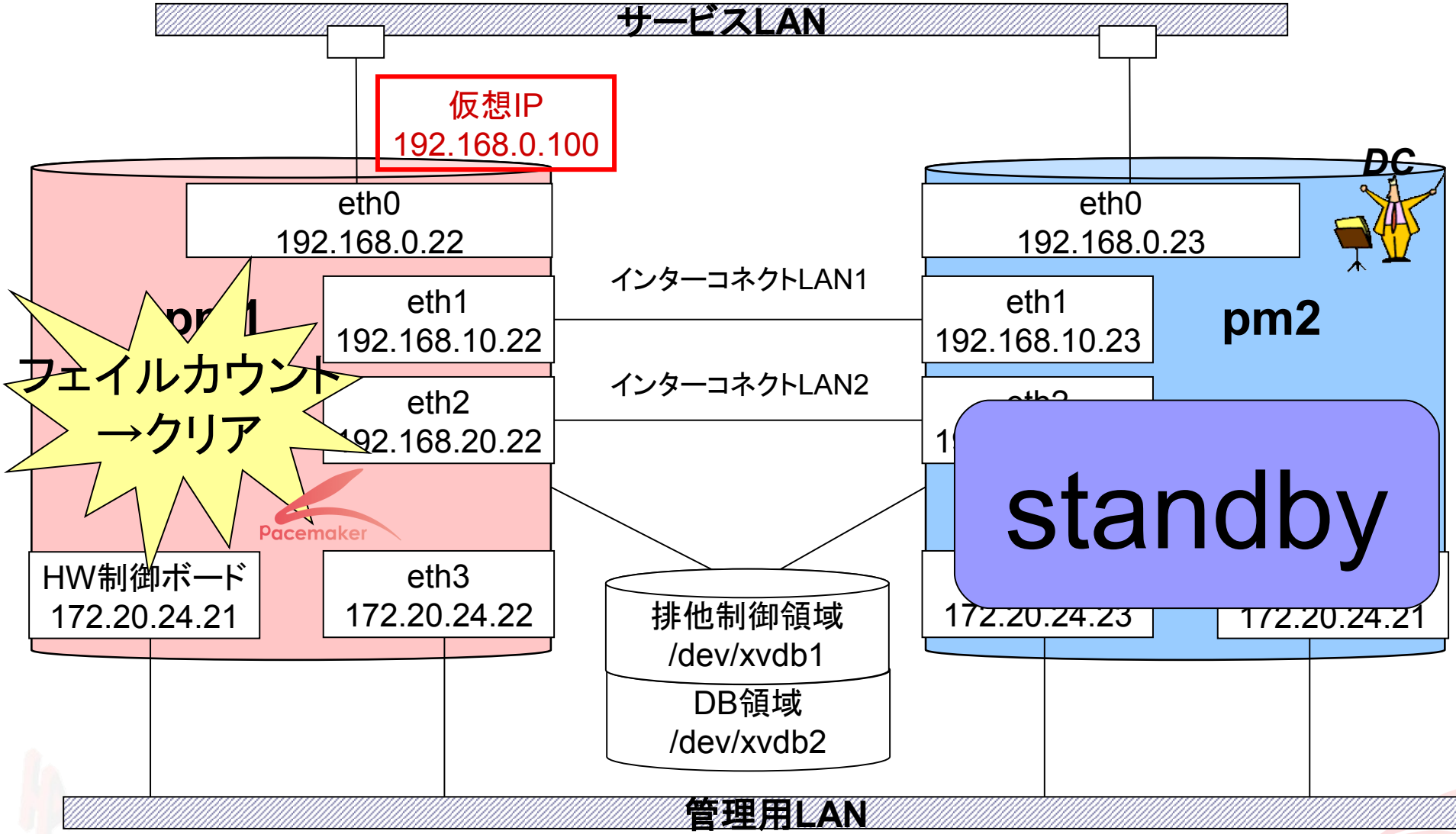
```
# crm_mon -fA
```

```
============
～ 省略 ～
============

Migration summary:
* Node pm1:
   prmPg: migration-threshold=1 fail-count=1
* Node pm2:

Failed actions:
    prmPg_monitor_10000 (node=pm1, call=34, rc=7, status=complete):
not running
```

デモ8

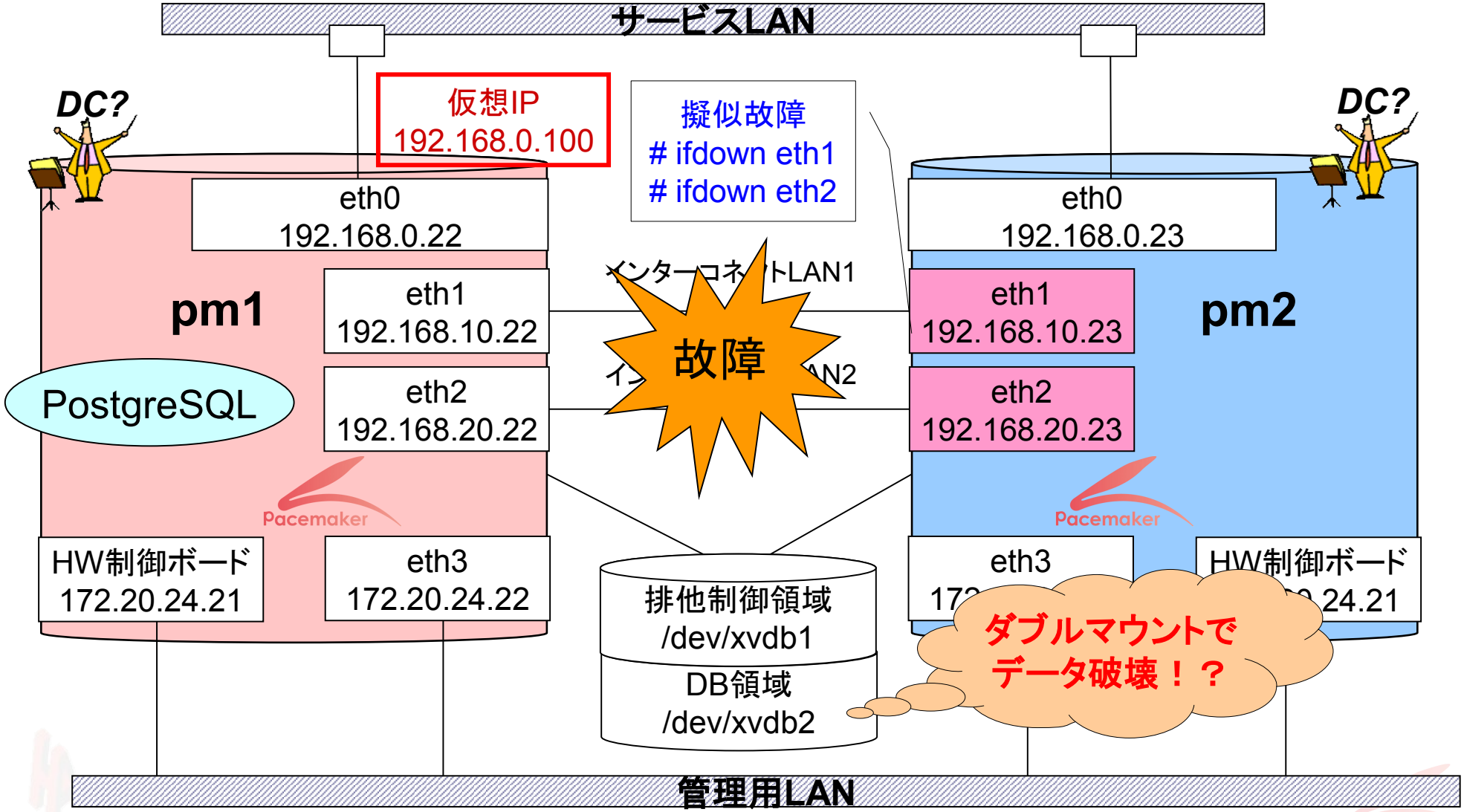
フェイルカウントをクリアしてみる…

crm resource cleanup prmPg pm1



デモ9

pm2 Online後に、スプリットブレイン…



リソース故障時、停止タイムアウト…

デモ10