実演！ Pacemakerで 楽々クラスタリング

2011年3月5日 OSC2011 Tokyo/Spring

Linux-HA Japan
田中 崇幸

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・設定方法を実演します！
④ フェイルオーバ・系切り替えを実演します！

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。

Linux-HA Japan勉強会情報など
随時情報を更新しています！

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

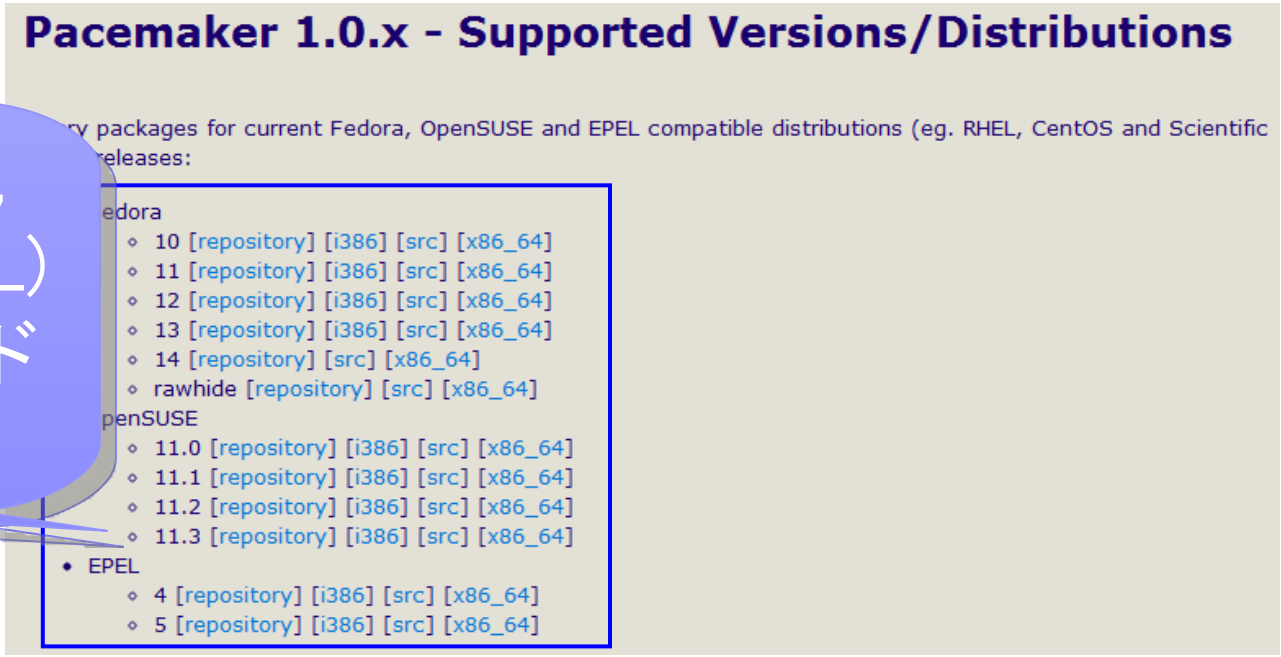
linux-ha-japan@lists.sourceforge.jp

※スパム防止のために、登録者以外の投稿は許可制です

本家Pacemakerサイト

http://clusterlabs.org/

Fedora, openSUSE, EPEL(CentOS/RHEL)のrpmがダウンロード可能です。



http://clusterlabs.org/rpm

ところで、
昨年12月まで
実は本家の
Pacemakerのロゴはこれでした

しかし

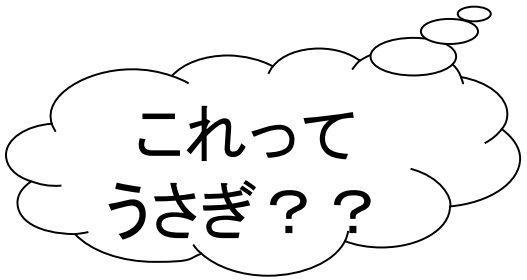
これ 

では、

いかにも医療機器っぽいので…

Pacemakerロゴ

Linux-HA Japan では、
Pacemakerのロゴ・バナーを独自に作成



本家Pacemakerロゴに勝負を挑みました！

http://theclusterguy.clusterlabs.org/post/1551578523/new-logo

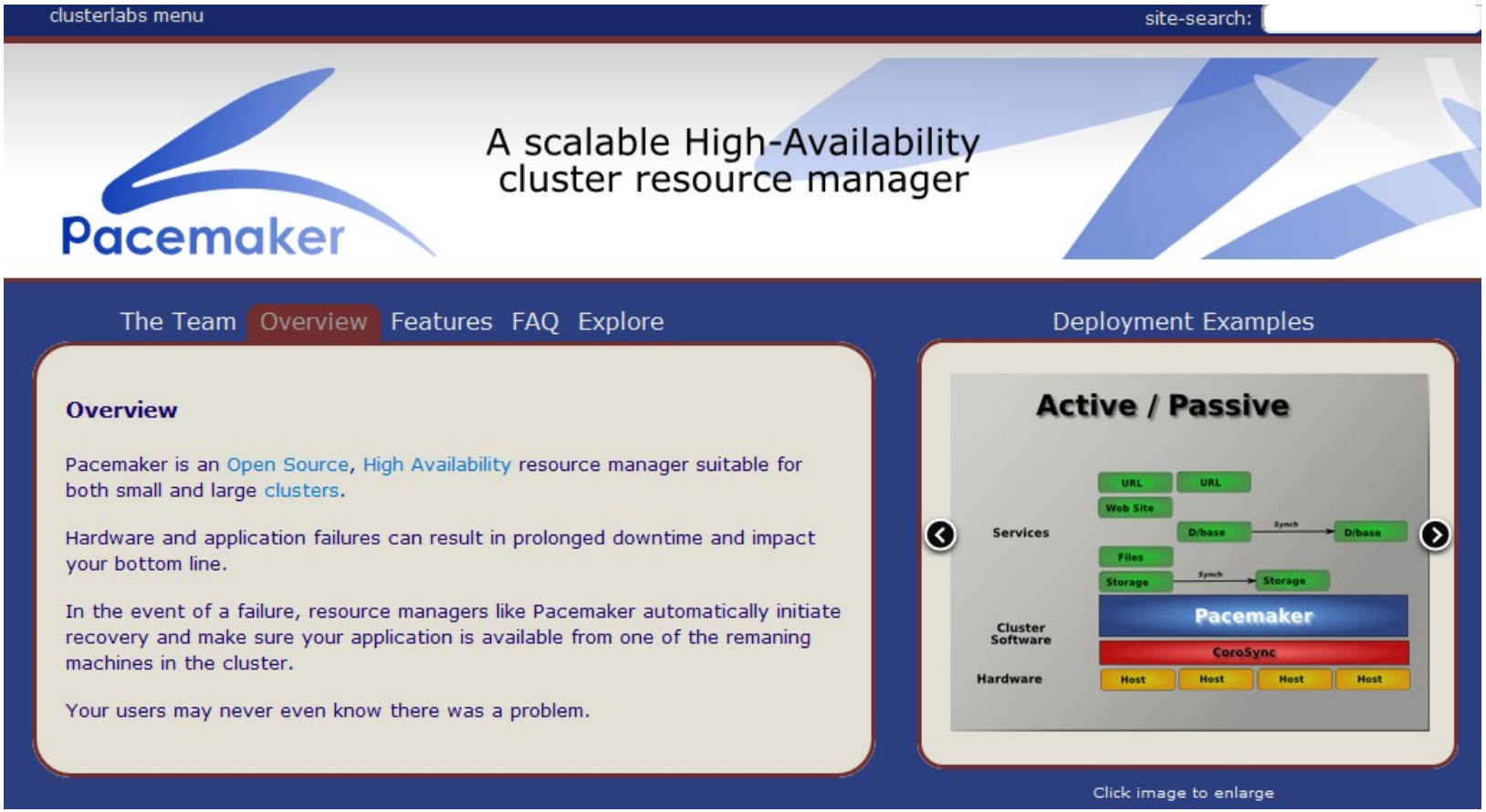
Cluster Guy New logo　検索

VS

すると新ロゴが
圧倒的リードで勝利したのです！

しかし！？

本家Pacemaker新ロゴは、青になるというオチに…



めげずに配布するクリアファイルも
ピンクから青に更新してみました…。

ほしい方は
ぜひブースに
来てください！

Linux-HA Japan は、あくまでロゴは、ピンクでいきます！

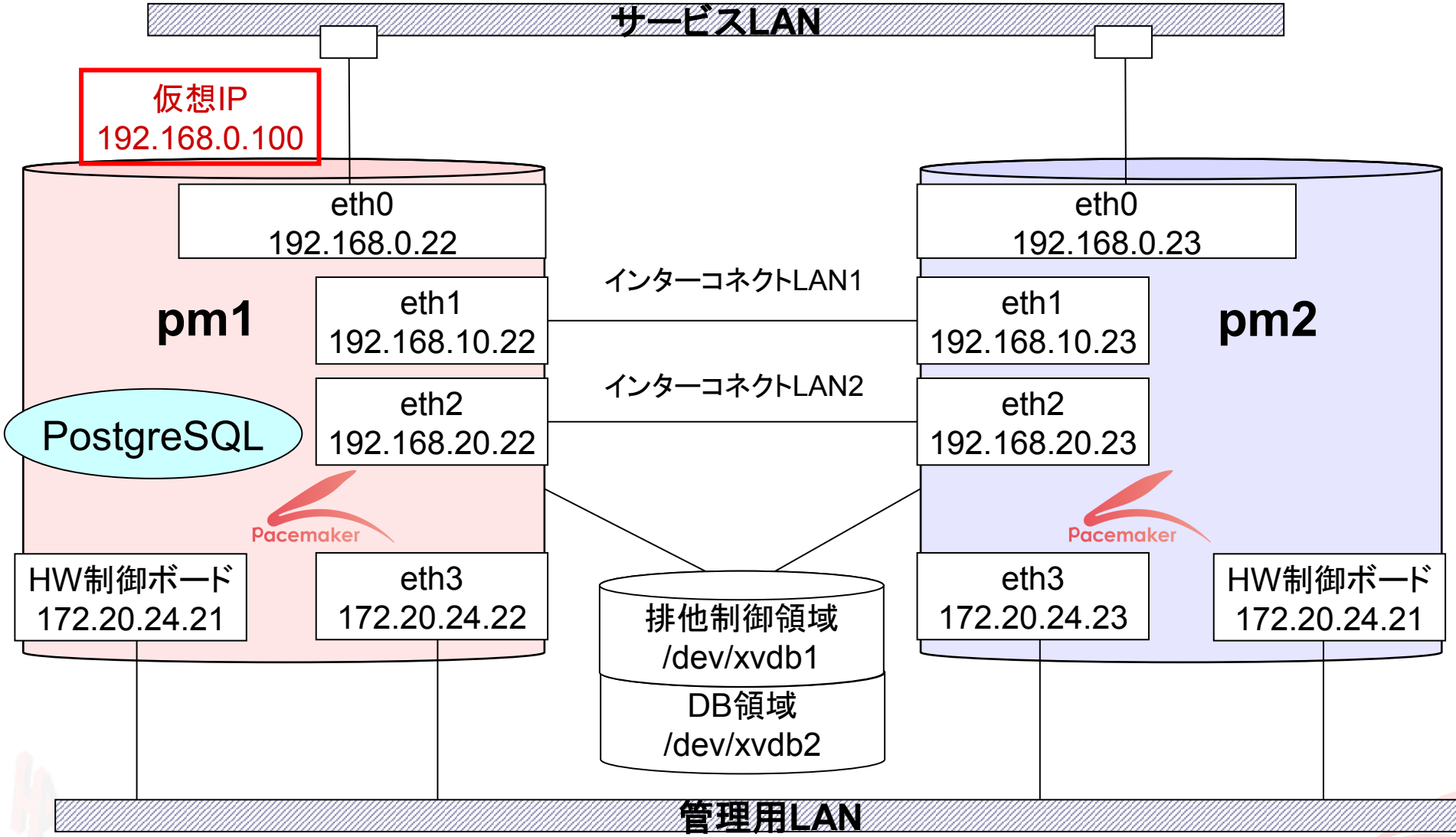
②

本日のPacemakerデモ環境

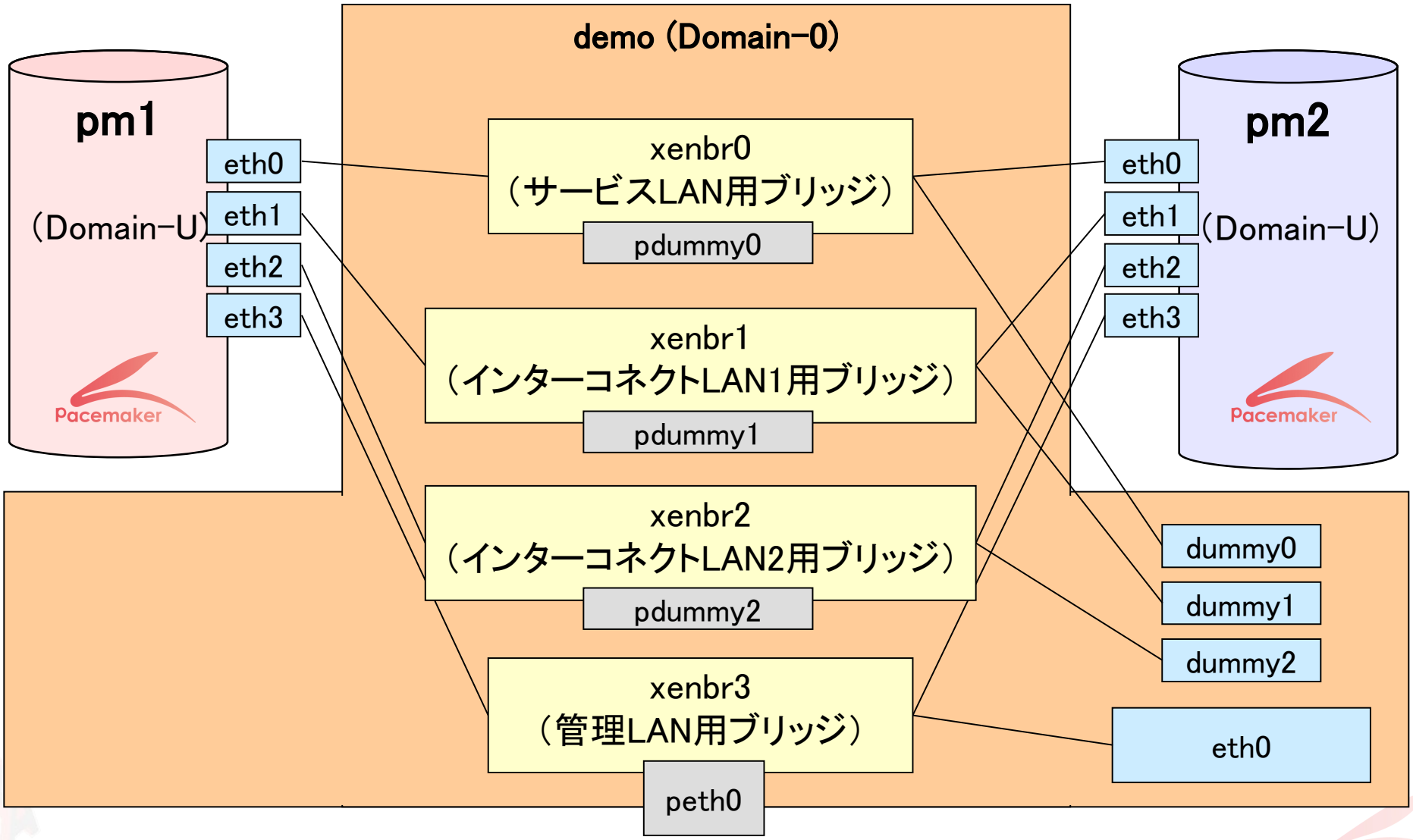
本日のPacemakerデモ環境

- ハードウェア
  - ノートPC（Core2Duo 2.26MHz、メモリ 2G）
- OS
  - CentOS 5.5 x86_64
- HAクラスタ
  - Pacemaker-1.0.10（インストールの実演を行います）
- クラスタ化するアプリケーション
  - PostgreSQL 9.0.3（インストール済み）
- 仮想環境
  - Xen（CentOS 5.5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

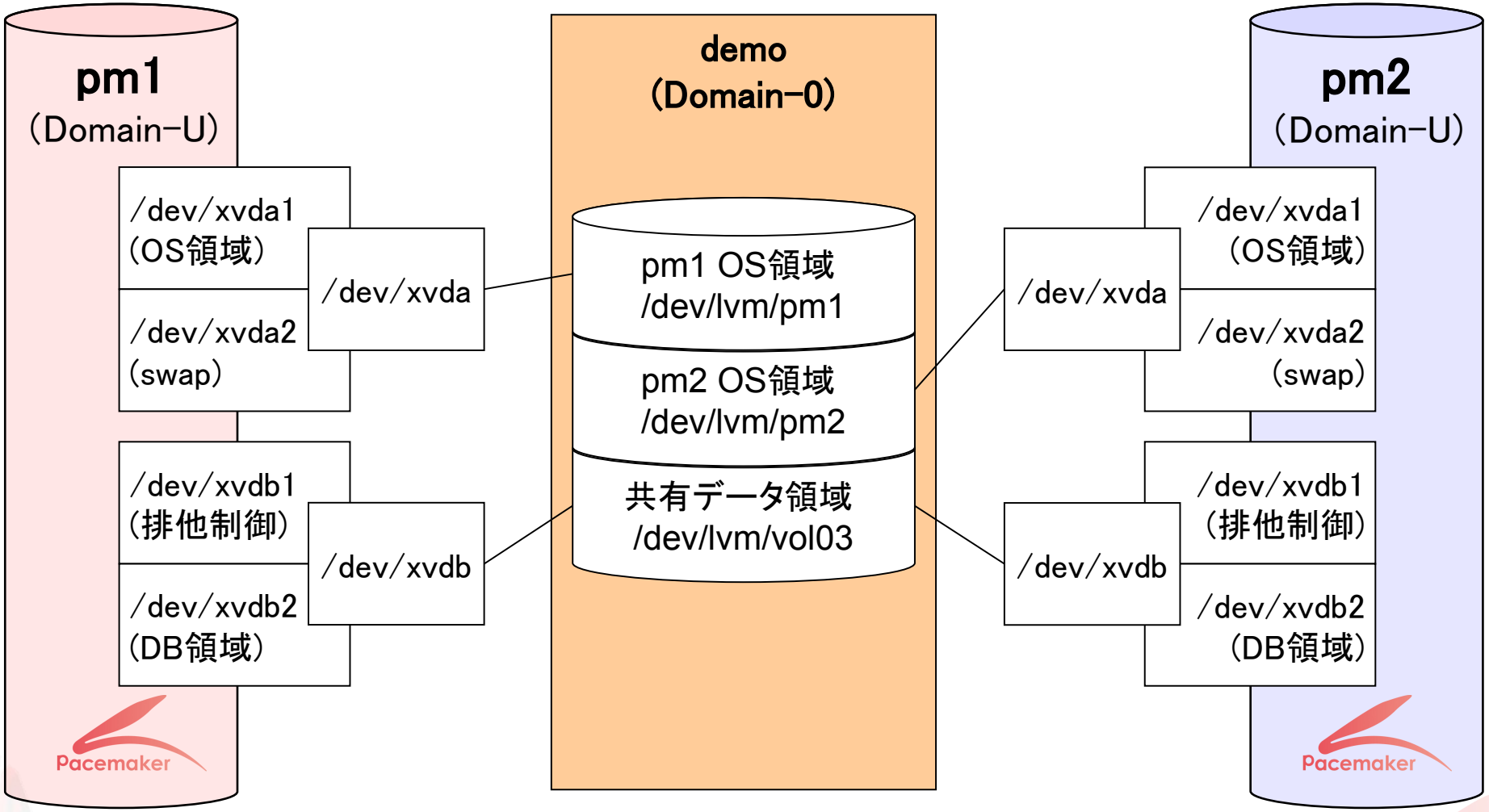
Pacemakerデモ構成



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

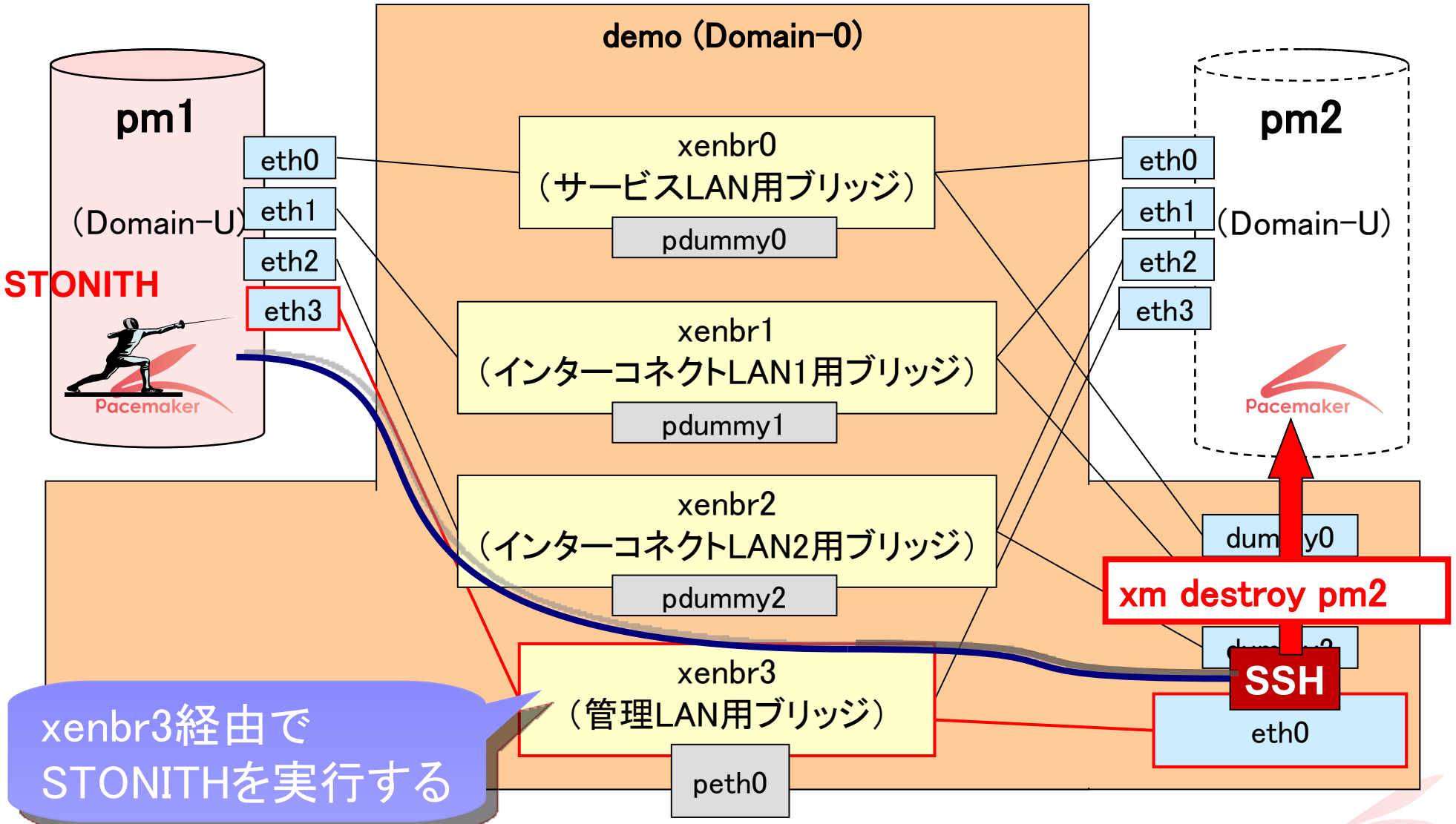
これら4つの
リソースは
グループ設定します

- 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 9.0.3 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、監視対象ノードの異常を検出したときに、強制的にそのノード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。

Pacemakerデモ機フェンシング（STONITH）構成



③

インストール・設定方法を実演します！

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。

2. Pacemaker リポジトリパッケージを展開

sourceforge.jp からダウンロードしたリポジトリパッケージを
/tmp 等のディレクトリで展開します。

```
# cd /tmp
# tar zxvf pacemaker-1.0.10-1.4.1.el5.x86_64.repo.tar.gz
                :
pacemaker-1.0.10-1.4.1.el5.x86_64.repo/rpm/pacemaker-1.0.10-1.4.1.el5.x86_64.rpm
pacemaker-1.0.10-1.4.1.el5.x86_64.repo/pacemaker.repo
pacemaker-1.0.10-1.4.1.el5.x86_64.repo/repodata/
pacemaker-1.0.10-1.4.1.el5.x86_64.repo/repodata/primary.xml.gz
pacemaker-1.0.10-1.4.1.el5.x86_64.repo/repodata/other.xml.gz
pacemaker-1.0.10-1.4.1.el5.x86_64.repo/repodata/filelists.xml.gz
pacemaker-1.0.10-1.4.1.el5.x86_64.repo/repodata/repomd.xml
```

インストールするRPMファイルと
repoファイル等が展開されます

3. ローカルyumリポジトリを設定

展開したrepoファイルをローカルyumリポジトリとして設定します

```
# cd /tmp/pacemaker-1.0.10-1.4.1.el5.x86_64.repo/
# vi pacemaker.repo
```

```
[pacemaker]
name=pacemaker
baseurl=file:///tmp/pacemaker-1.0.10-1.4.1.el5.x86_64.repo/
gpgcheck=0
enabled=1
```

パッケージを展開したディレクトリを指定
（デフォルトは /tmp なので、/tmpに tar.gzファイルを展開したのならば修正不要）

4. yumでインストール！

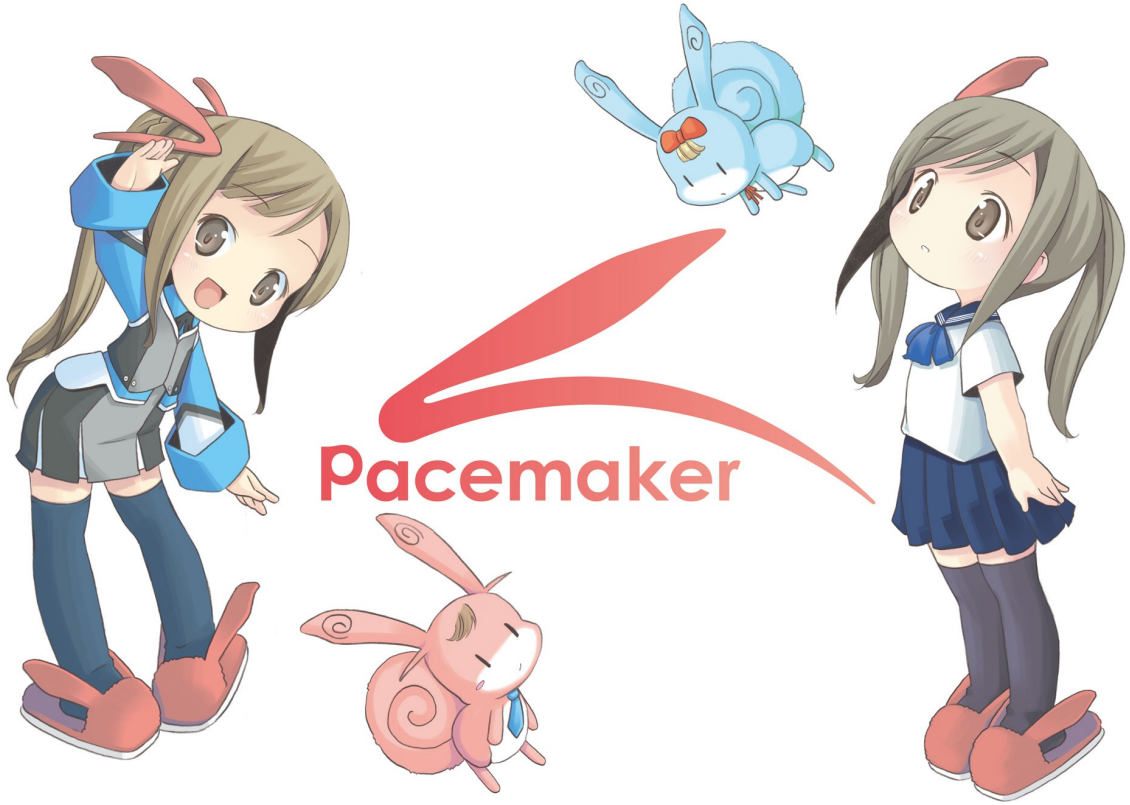
Linux-HA Japanオリジナルパッケージも同時にインストールします。

```
# yum –c pacemaker.repo install pacemaker pm_crmgen pm_diskd
pm_logconv-hb pm_extras
```

- pm_crmgen-1.0-1.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-1.0-1.el5.noarch.rpm ・・・ ログ変換ツール
- pm_extras-1.0-1.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 実演します！



/etc/ha.d/ha.cf

- クラスタ制御部の基本設定ファイル
- クラスタ内の全ノード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

- ノード間の「認証キー」を設定するファイル
- クラスタ内の全ノードに、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
- 以下は /var/log/ha-log への出力例
- 設定変更後は、syslogの再起動が必要

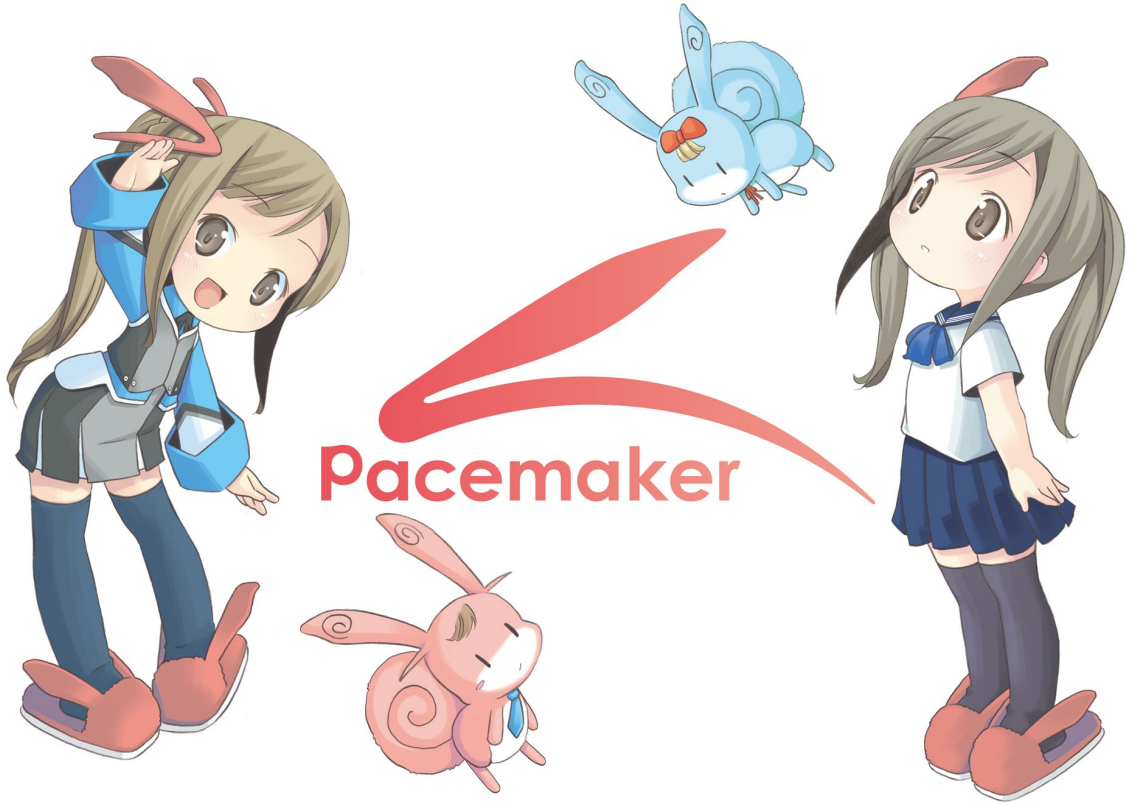
```
*.info;mail.none;authpriv.none;cron.none;local1.none
/var/log/messages
                :
          （省略）
                :
local1.info                                    /var/log/ha-log
```

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ノードで実行
Starting High-Availability services:                    [  OK  ]
```

ということで、
クラスタ設定とPacemaker起動を
実演します！



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
ノード名(ホスト名)が表示されます

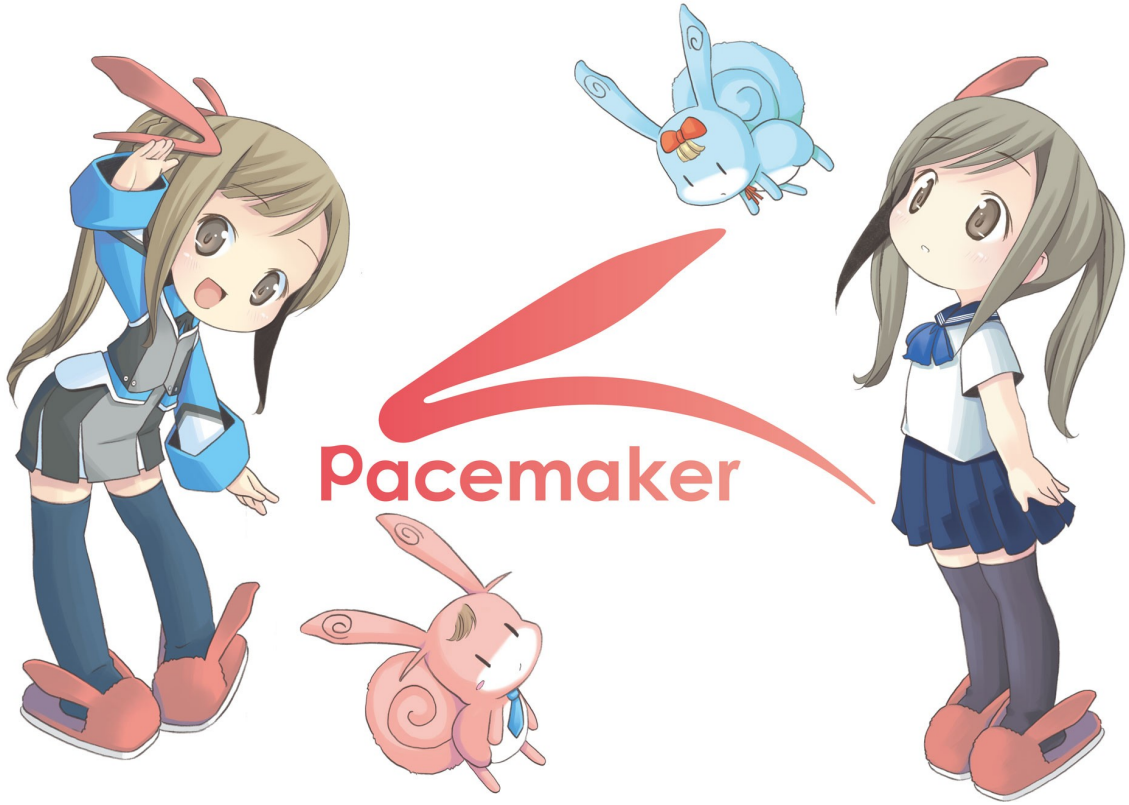
-fA オプションを付与すると、インターコネクトLANの接続状況も確認可能です。

```
# 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

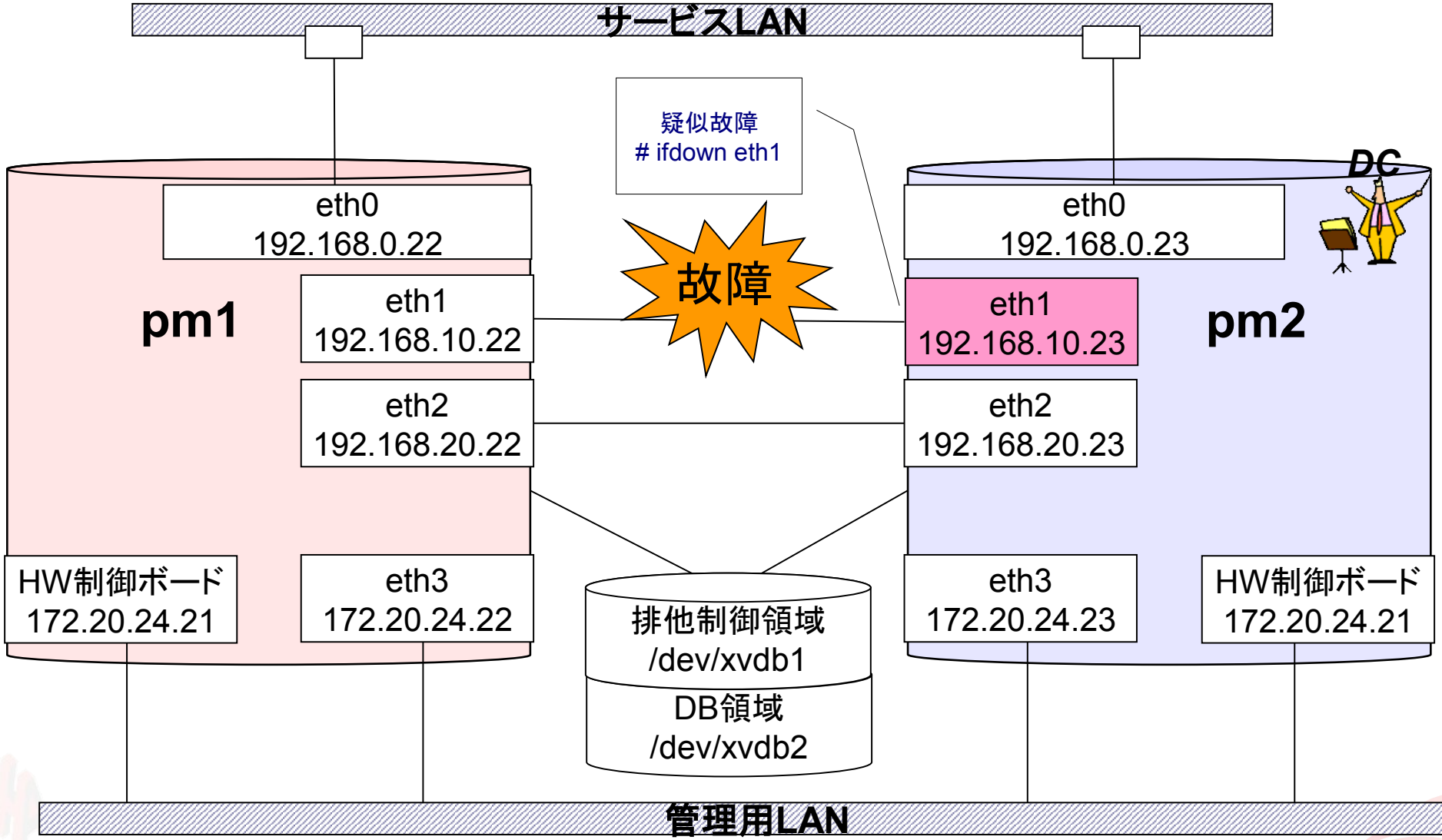
Linux-HA Japan Project

ここで、Pacemaker状態表示とインターコネクトLAN故障を実演します！



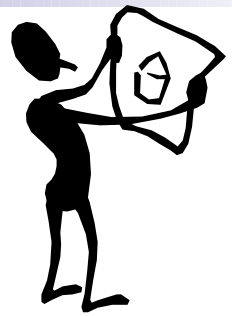
デモ例は
次ページ

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

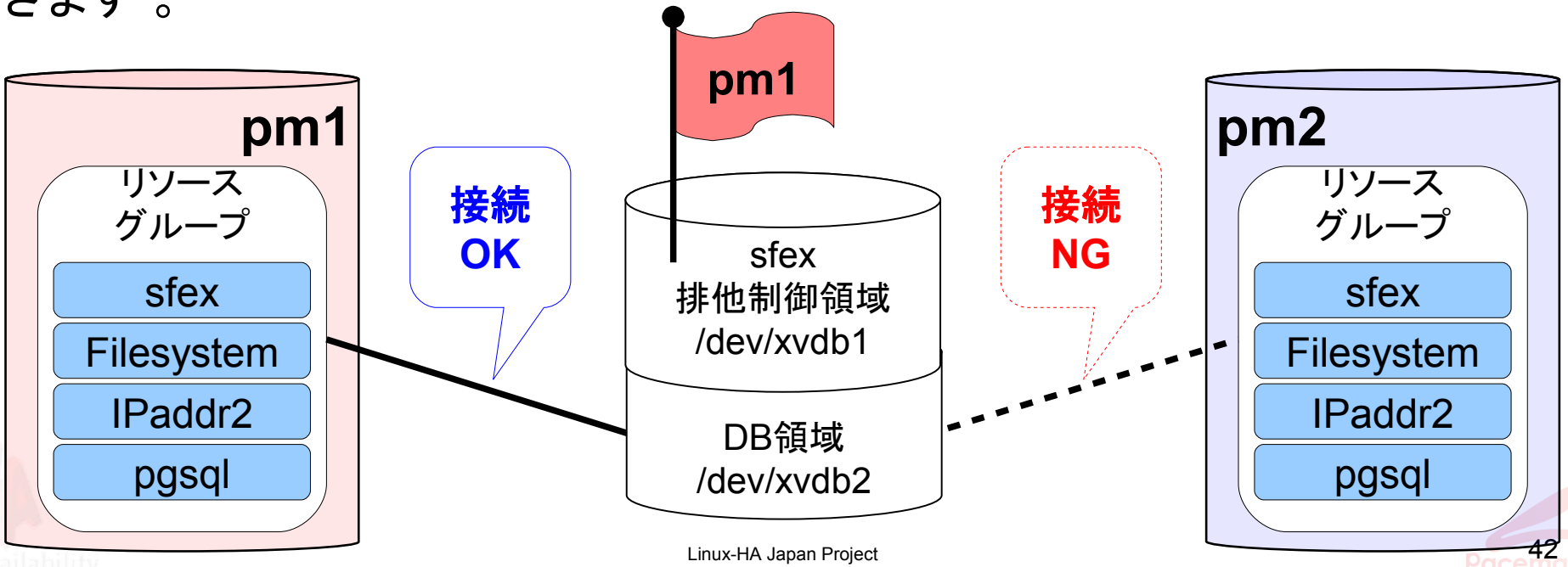
    リソースの監視（monitor）間隔は何秒にするか？タイムアウトは？
    故障時はどのように動作させるか？
    リソースエージェント(RA)に与えるパラメータは？
  - リソース配置・連携の定義

    リソースをどのノードで起動させるか？
    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。
所有権を持ったノード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

11/26 に pm_crmgen 1.0版をリリース

Linux-HA Japanで
crmファイル編集ツールを開発！

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/

- どのノードが優先的に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

どのノードをActiveにするかといった
リソース配置制約の設定も、ノード名を記述
するだけで可能です。

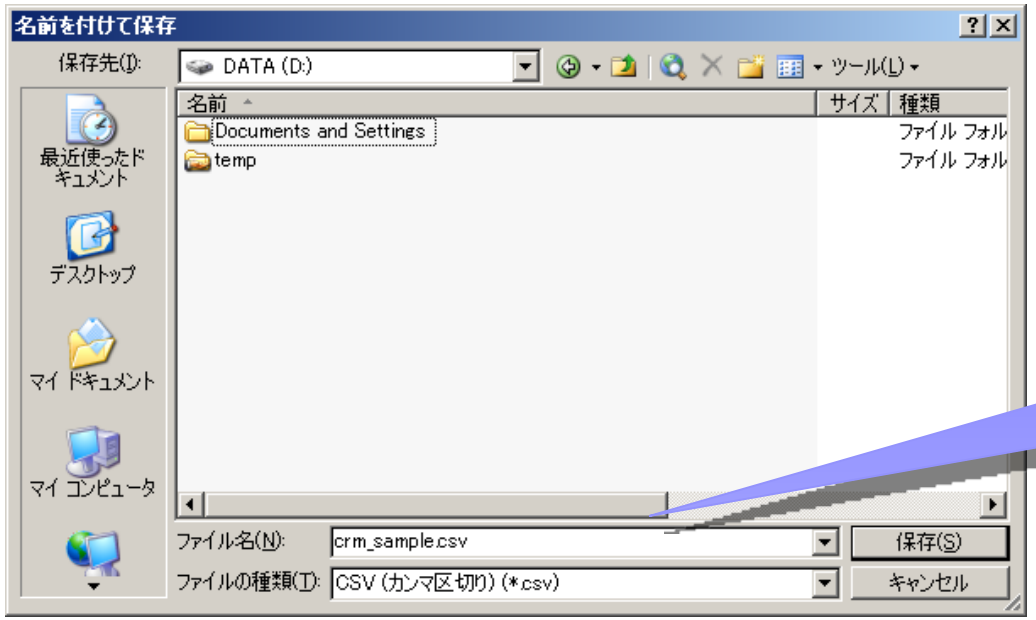
```
270 #表 6-1 クラスタ設定　...　リソース配置制約
271     LOCATION
272         rsc                 score:200   score:100   score:-inf
273 #       リソースID          Activeノード Standbyノー 非稼働ノード
274         grpPg               pm1         pm2
275         grpStonith1                                 pm1
276         grpSto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

リソースID

ActiveとStandbyノードを指定

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ファイルをノードへ転送

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
```

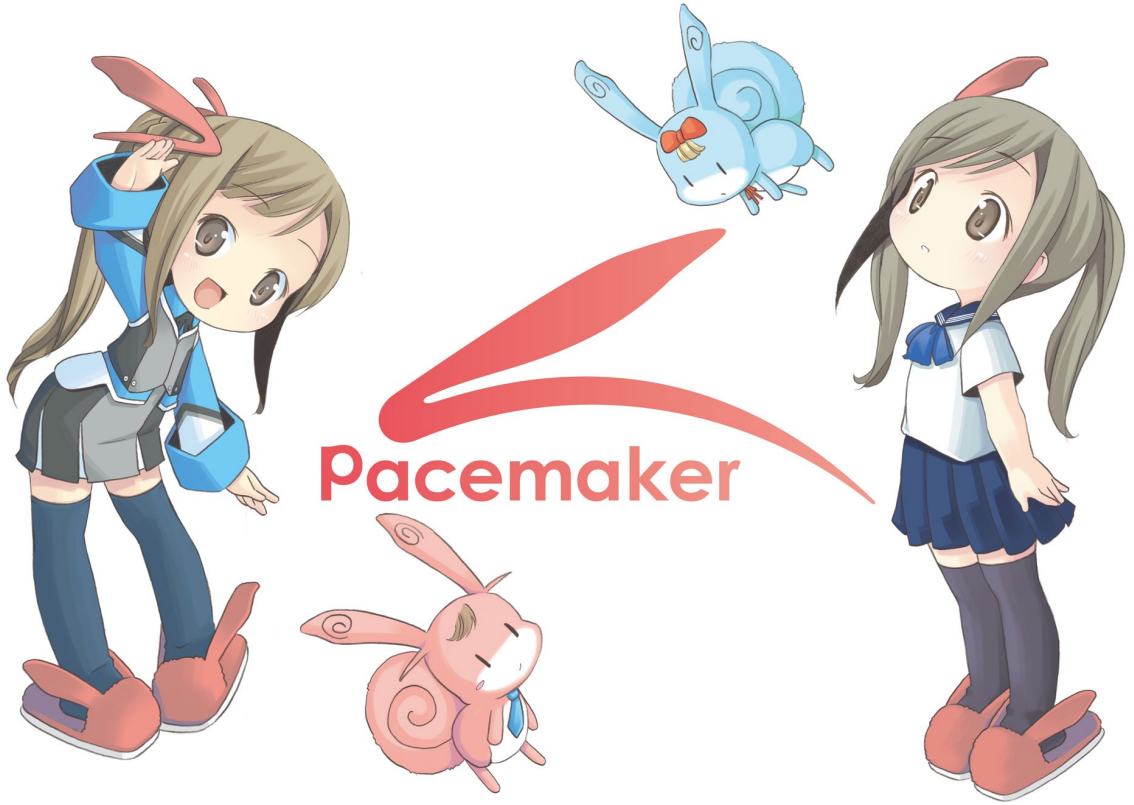
生成する設定ファイル名

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
```

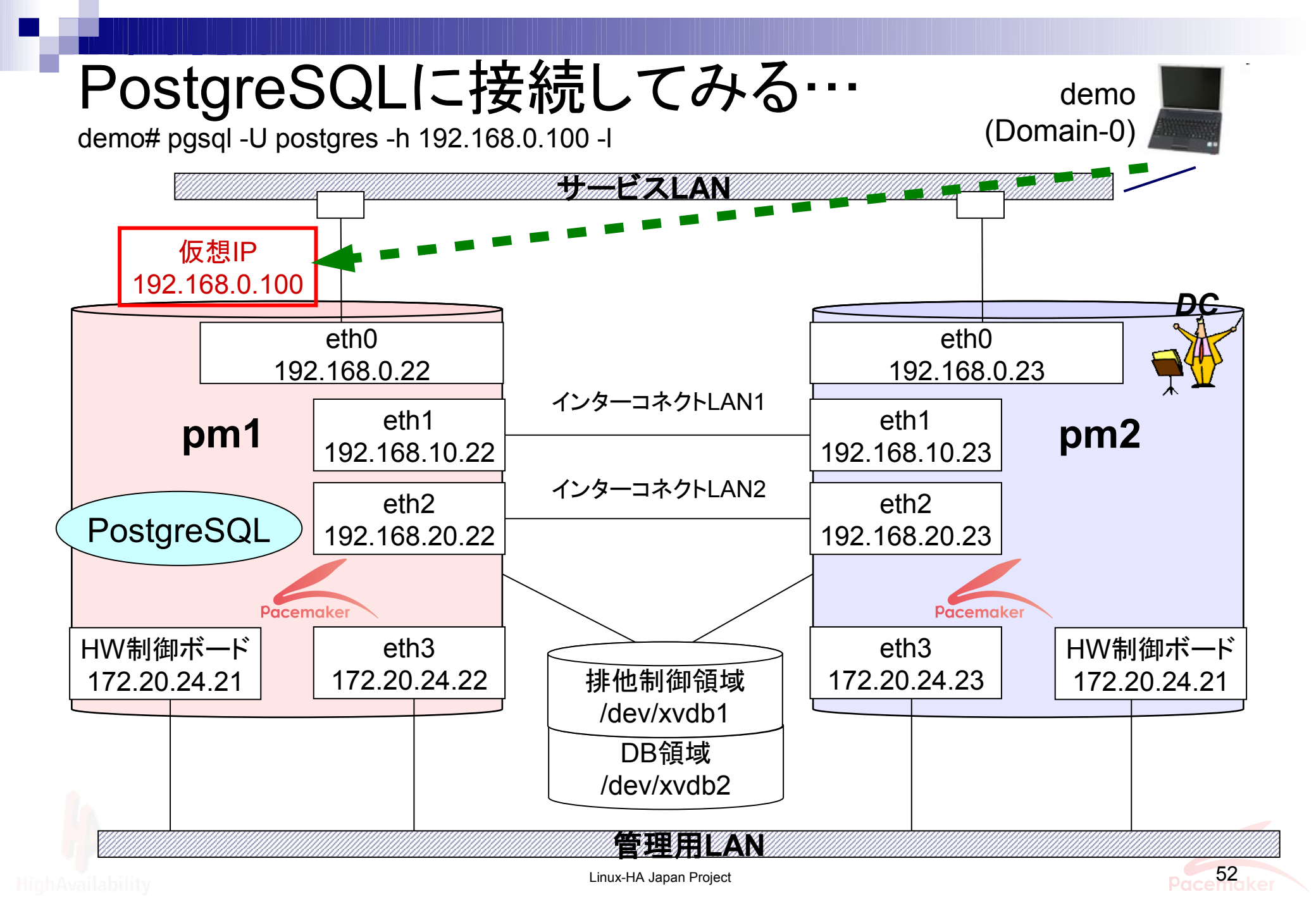
リソース設定をして
サービスの起動と、本当にサービス
が起動しているか実演します！



デモ例は
次ページ

PostgreSQLに接続してみる…

demo# pgsql -U postgres -h 192.168.0.100 -l

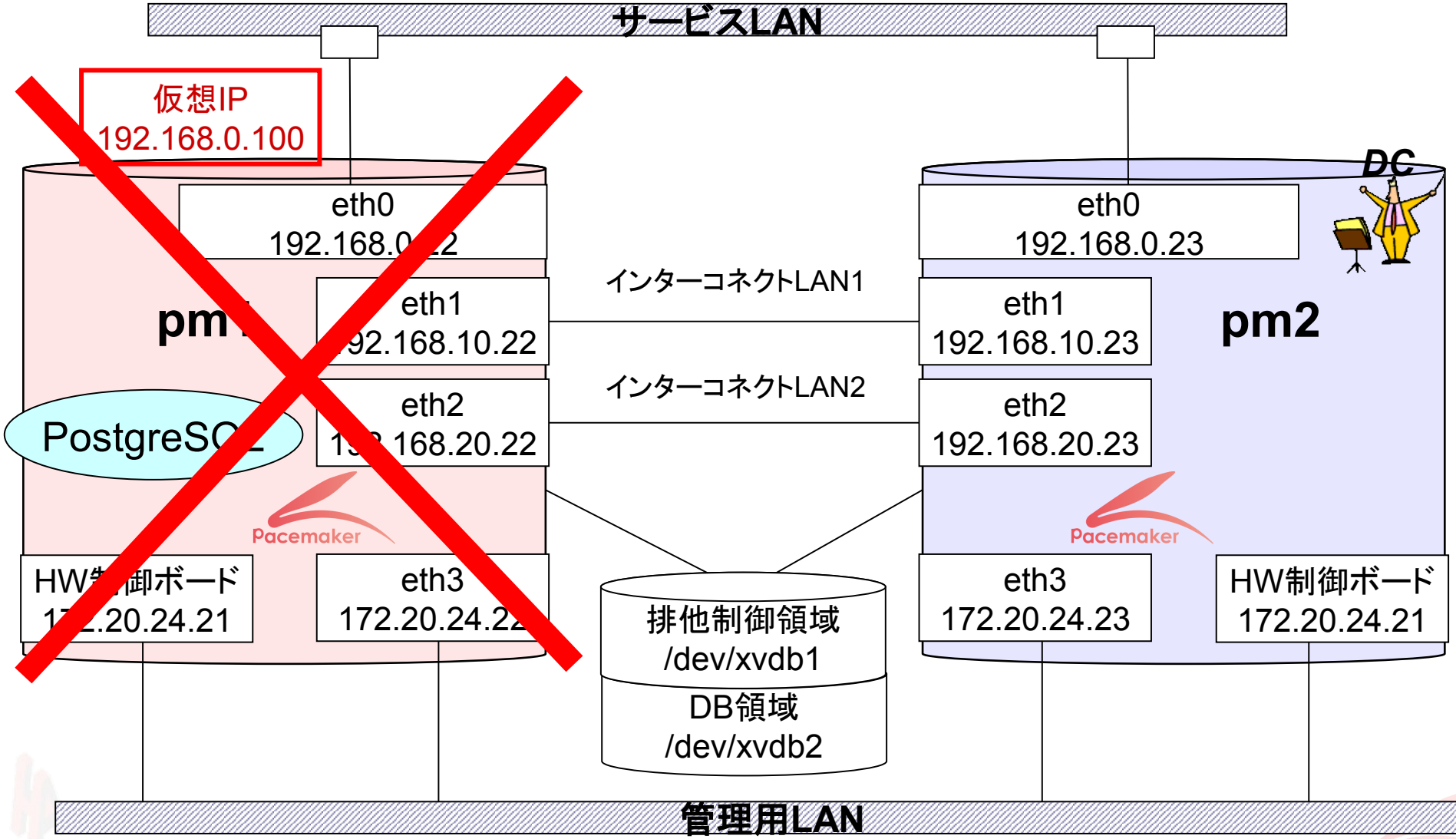


③

フェイルーバ・系切り替えを実演します！

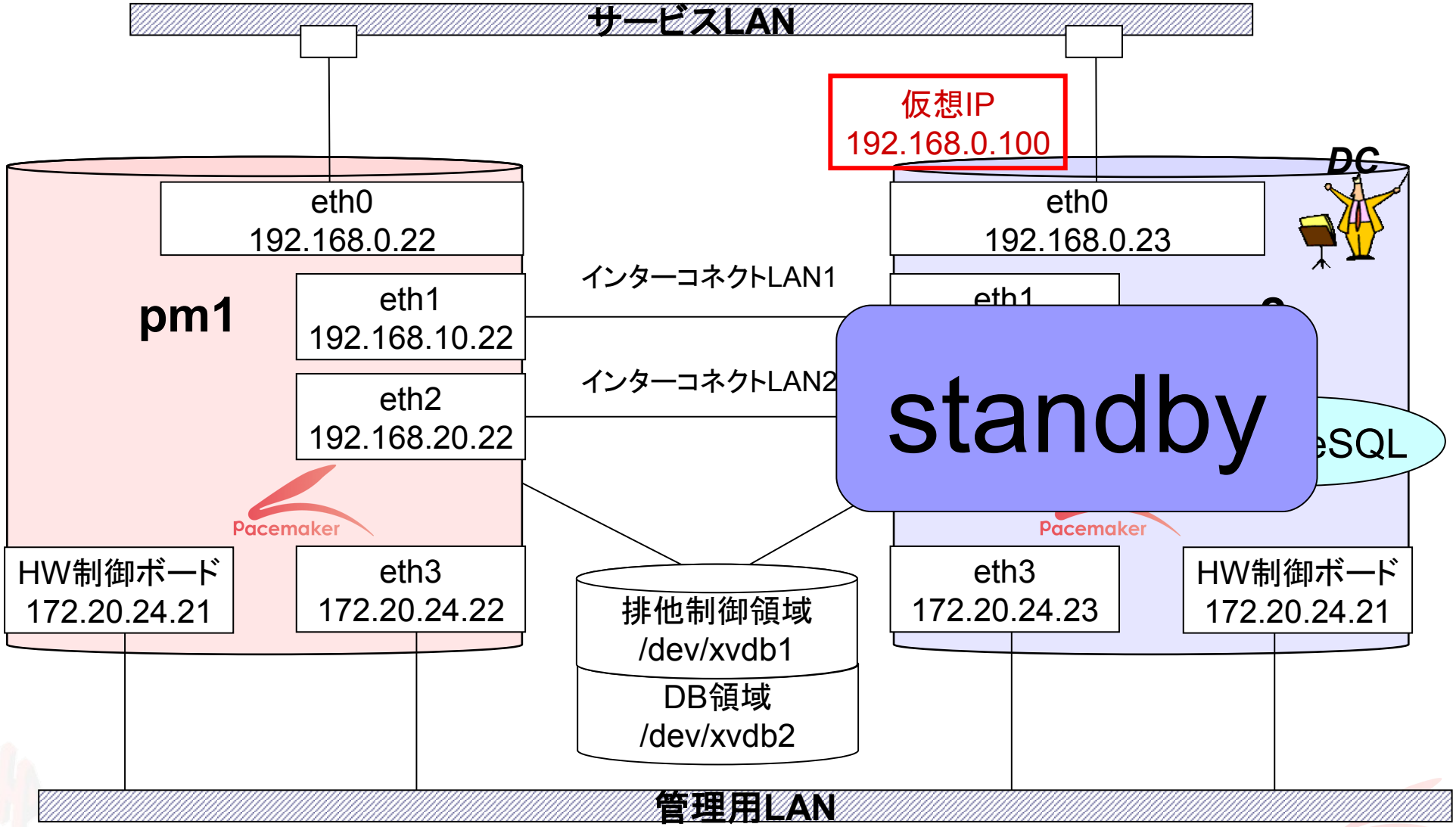
pm1を強制停止してみる…

demo# xm destroy pm1



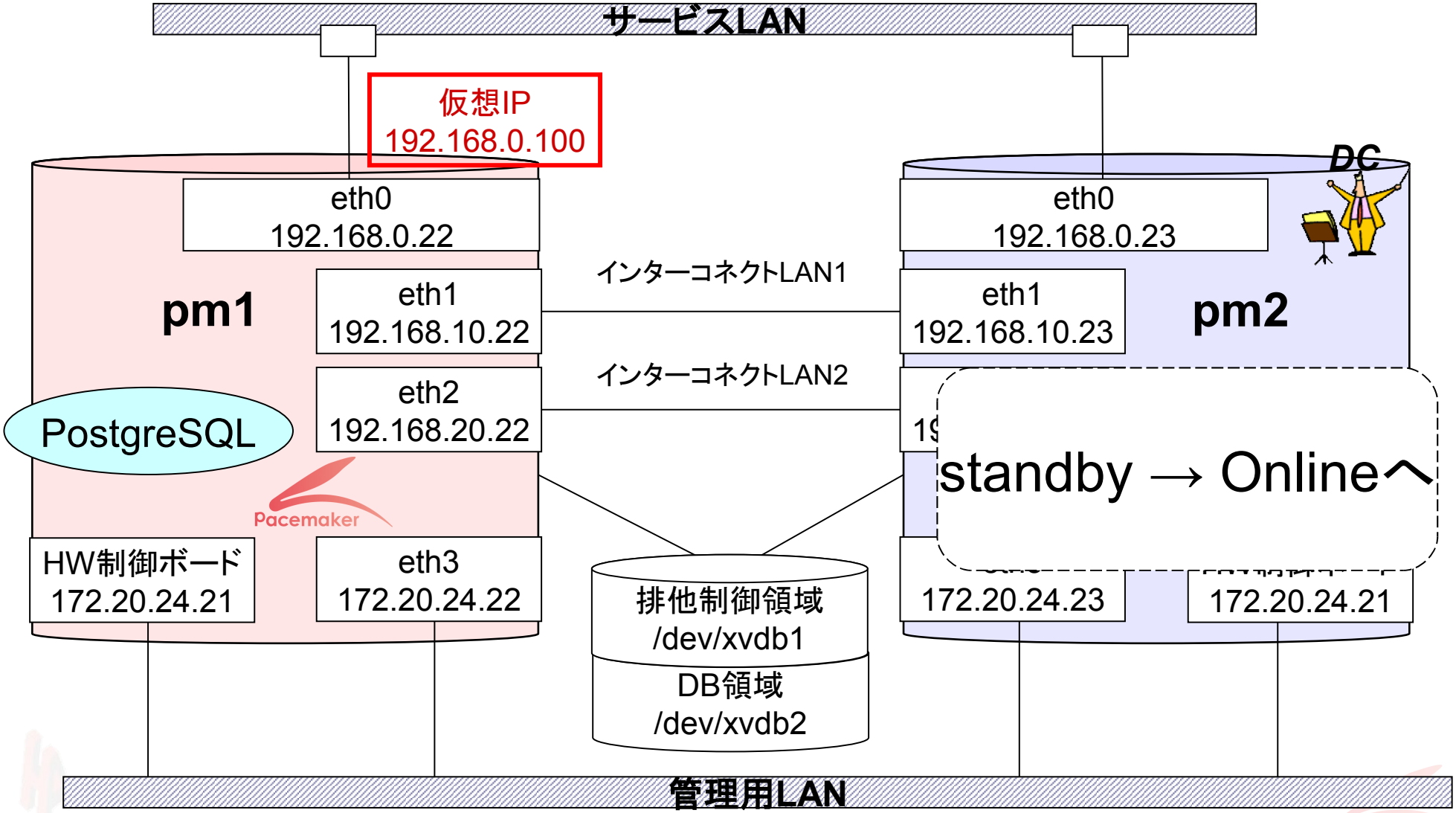
pm2をスタンバイ化してみる…

crm_standby –U pm2 –v on

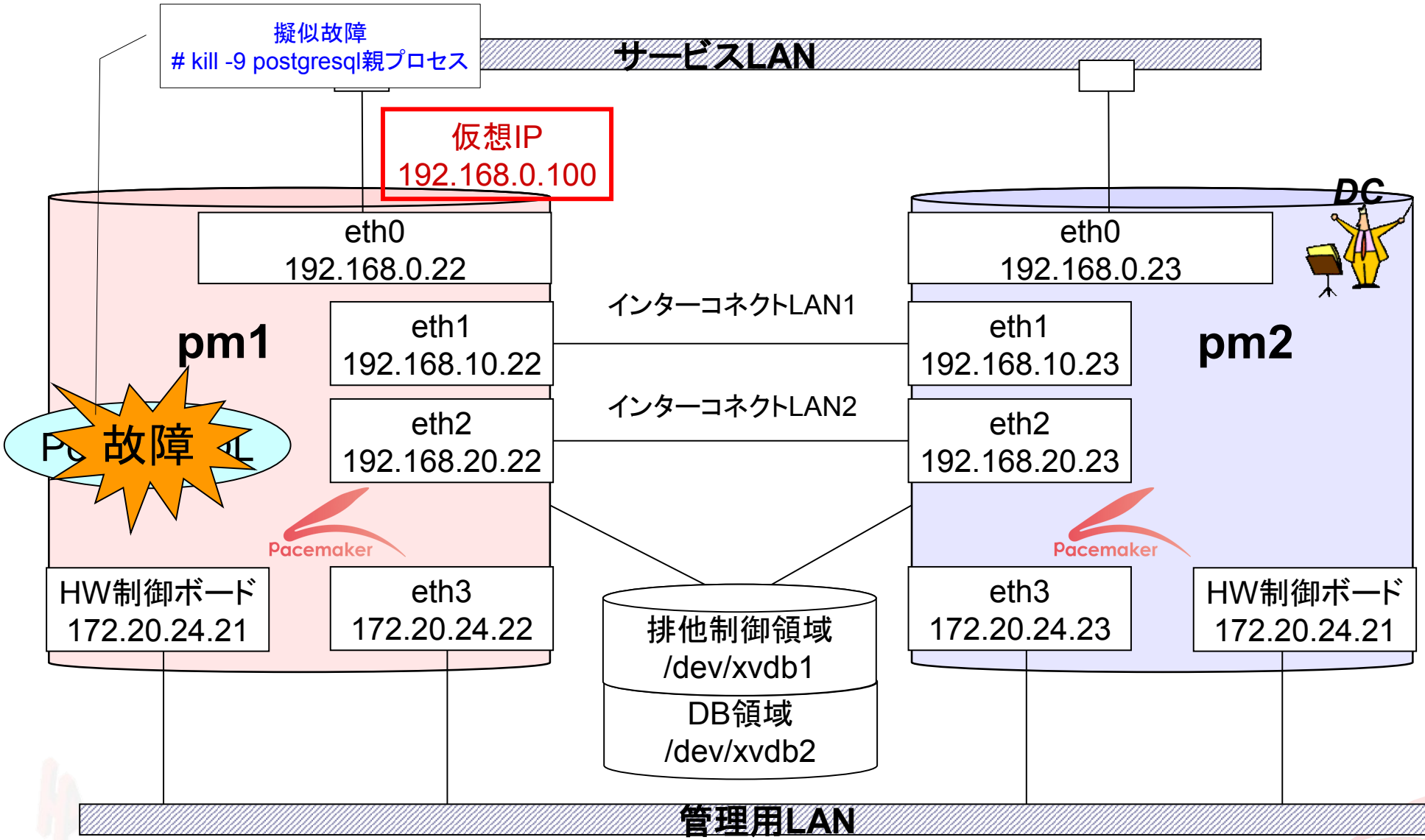


pm2をスタンバイ解除してみる…

crm_standby –U pm2 –v off

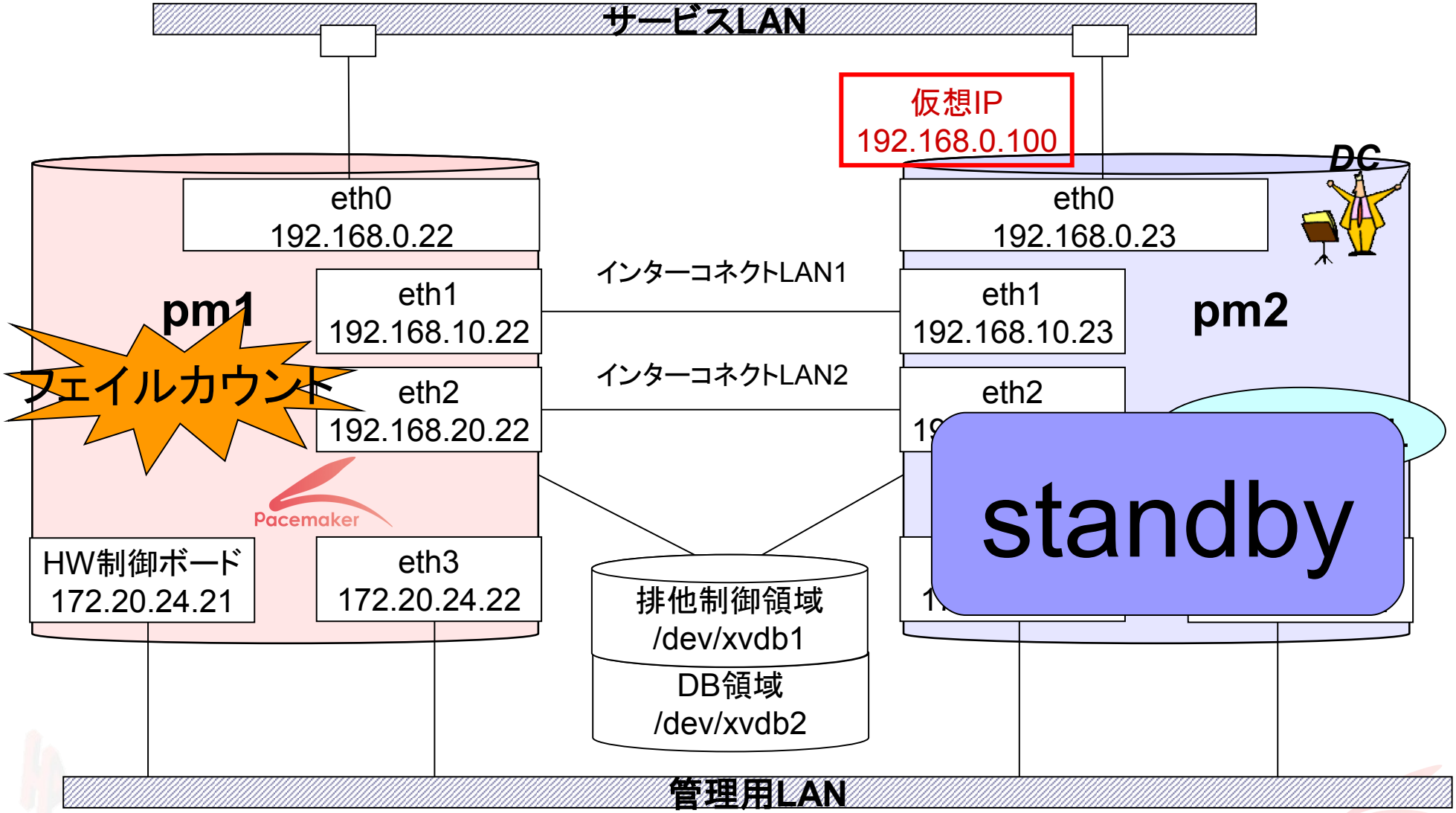


リソース故障させてみる…



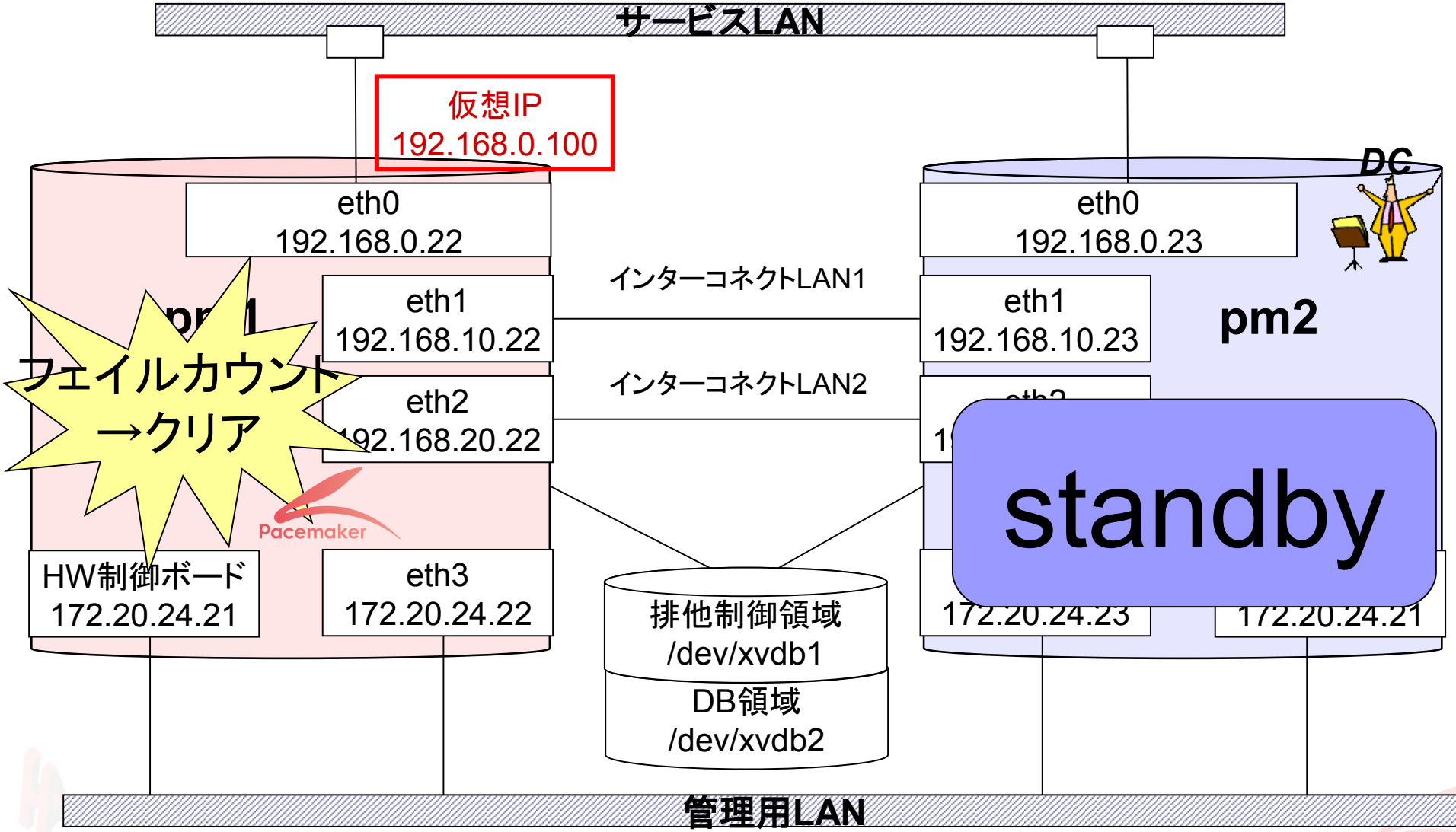
この状態でpm2をスタンバイしてみる…

crm_standby –U pm2 –v on

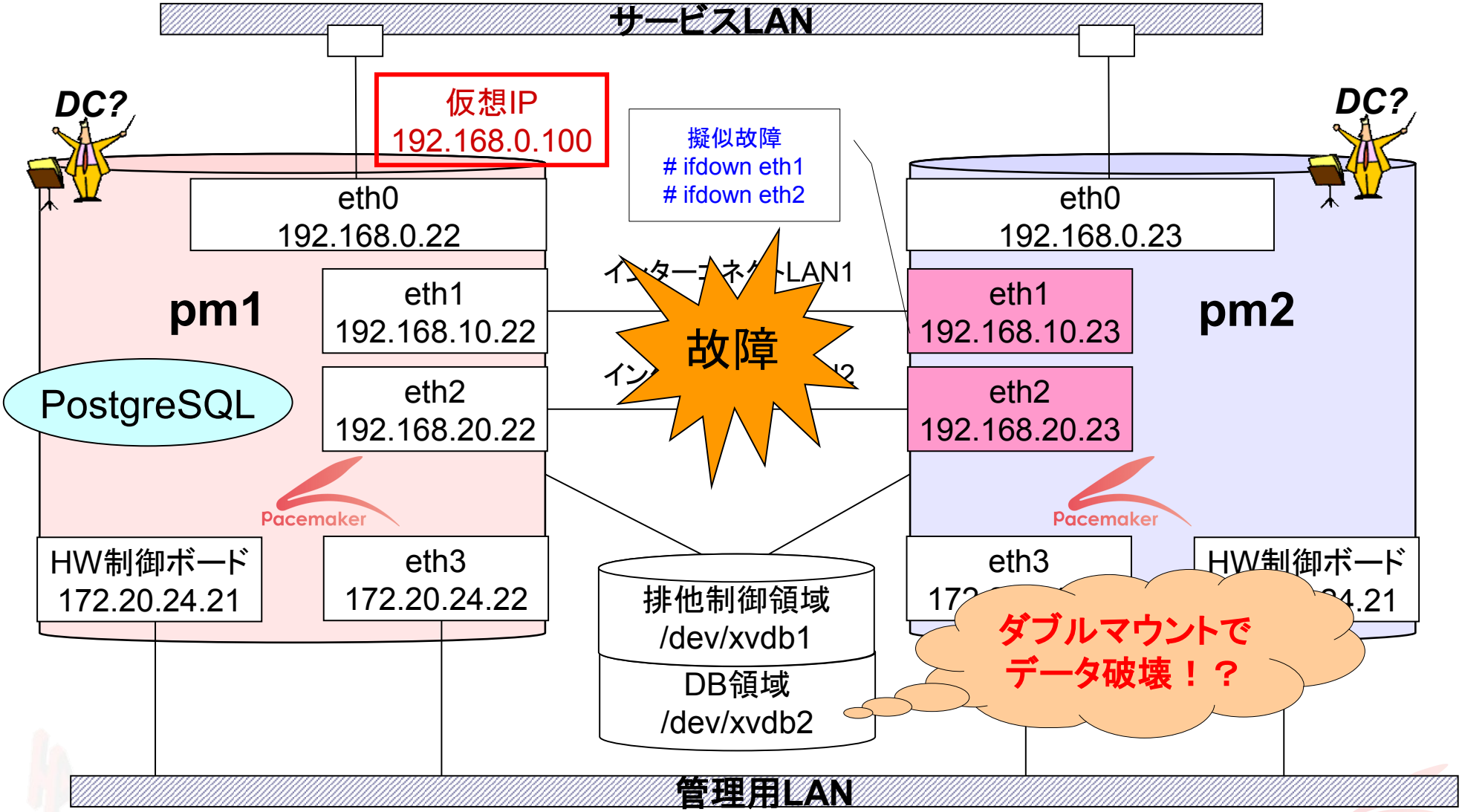


フェイルカウントをクリアしてみる…

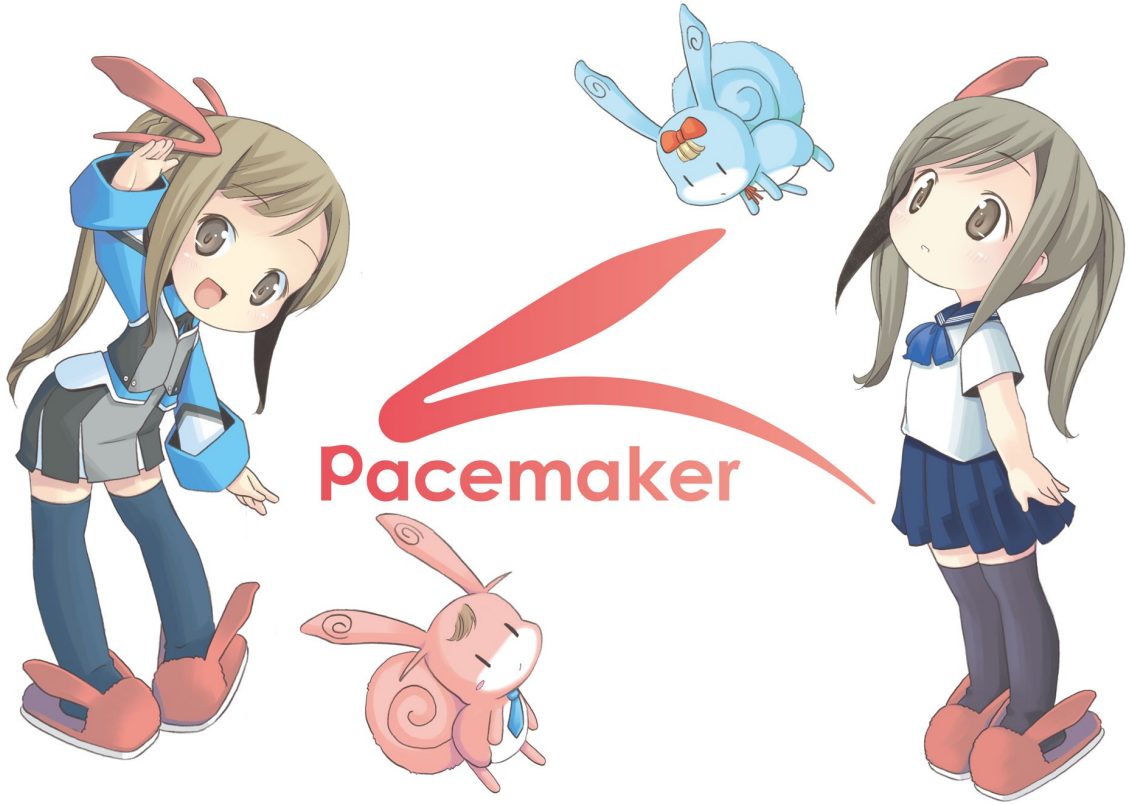
crm_resource –C –r prmPg –N pm1



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のデモしてます！



デモ環境構成

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

pm01

Apache

PostgreSQL

DRBD

データレプリケーション用LAN
(DRBD用)

pm02

DRBD

インターコネクトLAN
(Pacemaker用)

Pacemaker

Pacemaker

リモート電源制御ボード

リモート電源制御ボード

STONITH用LAN

デモ環境構成
（フェイルオーバ）

ネットワーク監視先

アクセス用端末

サービス用LAN

故障

仮想IP

pm01

pm02

Apache

PostgreSQL

DRBD

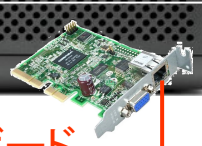
DRBD

データレプリケーション用LAN
(DRBD用)

インターコネクトLAN
(Pacemaker用)

Pacemaker

Pacemaker



リモート電源制御ボード

リモート電源制御ボード

STONITH用LAN

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

pm01

Apache

PostgreSQL

DRBD

pm02

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)

Pacemaker

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