

LPICレベル3 304技術解説セミナー

2014/6/22

有限会社ナレッジデザイン

取締役

市来 秀男



- LPIの紹介
- 自己紹介
- 問題と解説
- デモンストレーション
 - Pacemaker/Corosync
- 終わりに
- ブースのご案内



- LPIC(エルピック)はNPO法人/Linux技術者認定機関「LPI」(本部：カナダ)が実施している全世界共通のLinux技術者認定制度です。

LPI-Japan <http://www.lpi.or.jp>

LPI本部 <http://www.lpi.org>

LPICにはLPIC-1,LPIC-2,LPIC-3の3段階のレベルがあり、順次ステップアップしていく認定構成になっています。



- Linux、Java、Perl、MySQLの講師しています
- Linuxでは主にLPIC対策のトレーニングコースを担当しています

有限会社ナレッジデザイン

URL: <http://kwd-corp.com>

E-mail(受付): encke@kwd-corp.com



■ LPI304試験はLPIC-3の3つの試験(300,303,304)のうちの一つで次のカテゴリから出題されます。

- 仮想化
- ロードバランシング(負荷分散)
- クラスタ管理
- クラスタストレージ



■ オペレーティングシステム

- SUSE Linux Enterprise Server 11 SP3

■ アドオン

- SUSE Linux Enterprise High Availability Extension 11 SP3

本資料で扱っているCorosync, Pacemaker, OCFS2と関連ツールはこの製品に含まれています。

- haproxy-1.5-17.1.i586.rpm for SLES11

本資料で扱っているhaproxyは <http://www.rpmseek.com>からダウンロードしたものです。



■ corosyncあるいはheartbeatが提供するクラスタの機能はどれですか？ 2つ選択して下さい。

A. メンバーシップ

B. メッセージング

C. リソース管理

D. STONITH



■ corosyncあるいはheartbeatが提供するクラスタの機能はどれですか？ 2つ選択して下さい。

A. メンバーシップ

B. メッセージング

C. リソース管理

D. STONITH

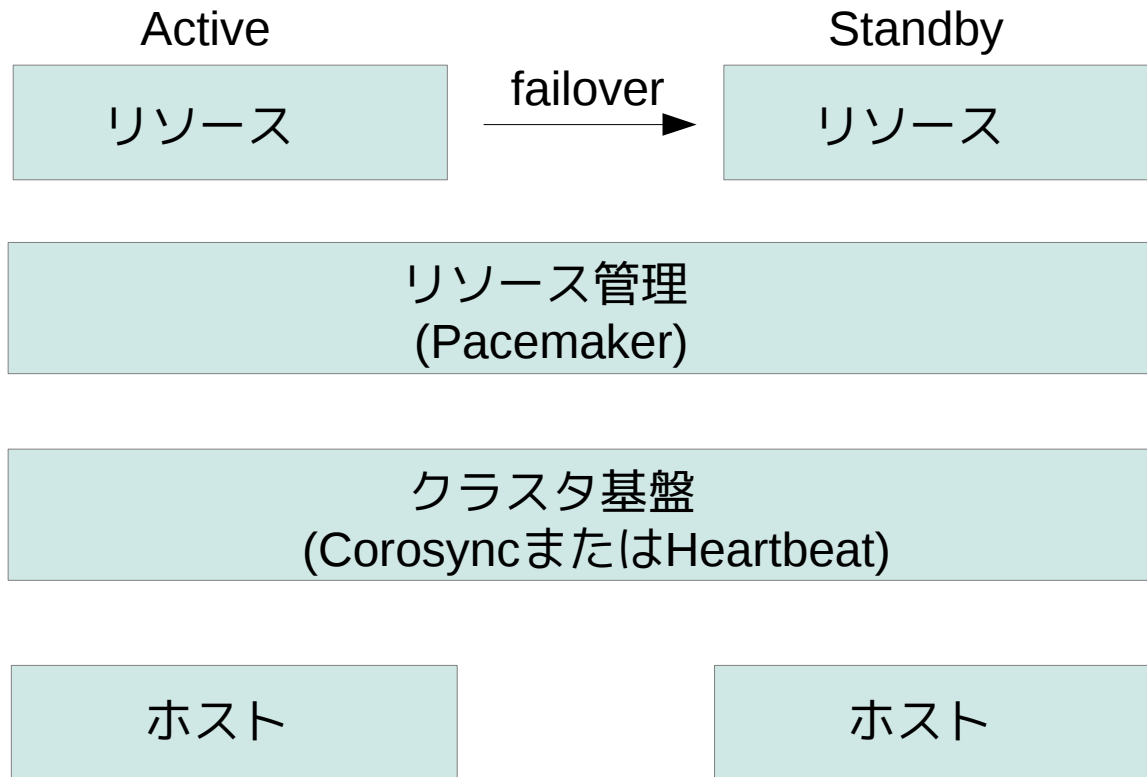


■ クラスタの構成

- クラスタは複数のノード(ホスト)、クラスタ基盤(Cluster Infrastructure)、クラスタリソース管理(CRM: Cluster Resource Management)、リソースなどから構成される
- クラスタ基盤サービスとはクラスタのメンバーシップ(どのノードがクラスタに参加しているか)とメッセージング(クラスタに参加しているノード間の通信)である
- CorosyncあるいはHeartbeatはクラスタ基盤サービスを提供する
- Corosyncは OpenAIS で開発され、現在はそこから派生した Corosync Cluster Engine プロジェクトで開発されている
- HeartbeatはLinux-HA プロジェクトで開発されている
- クラスタリソース管理を行うソフトウェアとしてPacemakerがある



■ 2ノードクラスタのActive/Standby構成例



フェイルオーバー：障害などの原因でActive側のリソースあるいはホストが停止するとStandby側のホストに切り替わる



■ Corosync

- 各ノードのcorosyncデーモンは設定ファイル/etc/corosync/corosync.confを参照してクラスタ基盤サービスを提供

■ /etc/corosync/corosync.confの記述例 (抜粋)

```
totem {  
  interface {  
    #Network Address to be bind for this interface setting  
    bindnetaddr:    172.16.0.0  
    #The multicast address to be used  
    mcastaddr:     239.95.240.82  
    #The multicast port to be used  
    mcastport:     5405  
  }  
}  
logging {  
  to_logfile:      no  
  #Log to syslog  
  #Log to syslog  
  to_syslog:       yes  
}
```



■ Pacemakerのコンポーネントでないものは次のうちどれですか？
1つ選択して下さい。

A. cib

B. crmd

C. clvmd

D. pengine

E. stonithd



■ Pacemakerのコンポーネントでないものは次のうちどれですか？
1つ選択して下さい。

A. cib

B. crmd

C. clvmd

D. pengine

E. stonithd



■ Pacemaker

- Pacemaker はクラスタリソースマネージャー(Cluster Resource Manager)
- クラスタ基盤サービスである Corosync あるいは Heartbeat によるメッセージングとメンバーシップ機能を利用してリソースの障害検知とフェイルオーバーを行う
- Heartbeat のクラスタリソースマネージャーとして Andrew Beekhof (アンドリュー・ビーコフ)氏によって開発され、現在は Heartbeat から独立し、Andrew Beekhof 氏をリーダーとする Pacemaker プロジェクトによって開発されている



■ Pacemakerの内部コンポーネント

Pacemaker は次の4つの主要な内部コンポーネントから構成されている。

- cib(Cluster Information Base)
クラスタ構成とクラスタリソース状態を管理するデーモン
CIB の情報はクラスタの全ノードに対して同期される
- crmd(Cluster Resource Management Daemon)
各ノードの crmd の中からマスターとして選定された crmd がクラスタの意思決定を行う
- pengine(Policy Engine)
クラスタの最適な状態と、そのために必要な処理を計算
- stonithd(Shoot The Other Node In The Head Daemon)
孤立して同期の取れなくなったノードがデータにアクセスすることのないようにデバイスを利用して当該ノードをオフラインにするデーモン



■ Pacemakerの内部コンポーネント(続き)

この他にも2つの内部コンポーネント(デーモン)がある。

- lrmd(Local Resource Management Daemon)

個々のノードのローカルリソースを管理する

- attrd(Attribute Daemon)

CIB のノード属性を管理する

■ 関連用語

- フェンシング(fencing)

他から孤立したノードによる共有リソースへのアクセスを防ぐこと

- スプリットブレイン(split brain)

障害により、ノードが互に認識できない複数のグループに分かれること

- クォーラム(quorum)

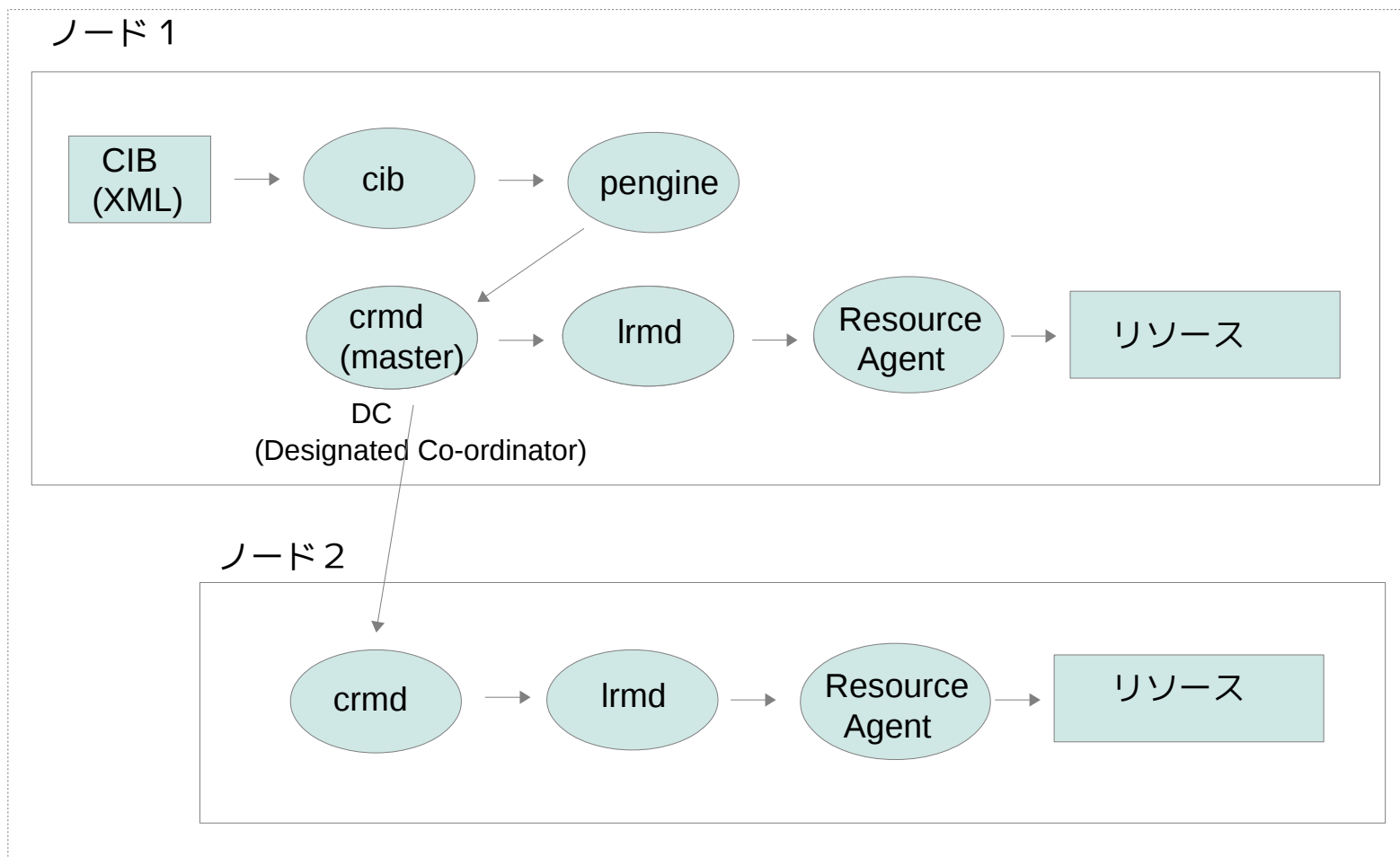
スプリットブレインが発生した場合、過半数のノードが参加したグループが持つクラスタの決定権のこと。

2ノード構成クラスタの場合、1つのノードだけでは過半数とならないので、no-quorum-policy="ignore" と設定することにより、処理を続行する



■ 内部コンポーネントによる処理シーケンス図

クラスタ





■ 内部コンポーネントによる処理シーケンス

Pacemakerの内部コンポーネントは次のような処理を行う。

- CIBはXML形式で記述された以下の2つのデータを保持する。
 - ・ クラスタの設定情報
 - ・ クラスタ内の全リソースの状態
- CIBのデータは同期した状態でクラスタの全ノードに保持される。PEngineはCIBのデータを使って最適な状態を計算し、そこへ到達するための処理を行う。
- Pacemaker はどれか 1 つの crmdデーモンをマスタに選出して DC (Designated Co- ordinator) とする。
(DC になった crmdがフェイルすると、ただちに新しいcrmdがDC に選出される)
- PEngine の計算結果は 一連の処理命令としてDCに渡される。
DCは処理命令を順番にLRMdと、他ノードのCRMdにクラスタメッセージ基盤を使って送る。他ノードのCRMdはそれをLRMdに送る。
- 各ノードは期待された結果と実際の結果に基づいて処理結果をDCに返し、DCからのさらなる処理の指示あるいはアボートの指示を待つ。期待した結果が得られなかった時はPEngineに最適状態の再計算を依頼する。



■ Pacemakerのコンポーネントでないものは次のうちどれですか？
1つ選択して下さい。

A. cib

B. crmd

C. clvmd

D. pengine

E. Stonithd

* clvmd(Cluster LVM Daemon)は共有ストレージ上のLVMへの複数ノードからのアクセスを制御するデーモンです。
Pacemakerのコンポーネントではありません。



■ 仮想IPとApache Webサーバをリソースとして管理する場合に必要なPacemakerのリソースエージェントは次のうちどれですか？ 2つ選択して下さい。

A. dhcpd

B. IPaddr2

C. httpd

D. apache



■ 仮想IPとApache Webサーバをリソースとして管理する場合に必要なPacemakerのリソースエージェントは次のうちどれですか？ 2つ選択して下さい。

A. dhcpd

B. IPaddr2

C. httpd

D. apache



■ リソース

- Pacemakerはリソースを監視、管理する
- リソースの例:
仮想IP、Apache Webサーバ、OCFS2(Oracle Cluster FileSystem 2)

■ リソースエージェント(RA: Resource Agent)

- リソースエージェントはリソースの起動、停止、監視を行うプログラム
- ほとんどのリソースエージェントはシェルスクリプト
- C, Python, Perl など、他の言語で記述することもできる
- RAは仕様、提供元によりクラス(class)、プロバイダ(provider)に分けられている
- コマンドラインでのRAの指定は、「クラス:プロバイダ:RA名」
クラス:lsb(Linux Standard Base), ocf(Open Cluster Framework), ...
プロバイダ: heartbeat, linbit, pacemaker, ...



■ crmコマンドでリソースエージェントの情報を表示

クラスの一覧表示

```
# crm ra classes  
lsb  
ocf / heartbeat linbit lvm2 ocfs2 pacemaker  
service  
stonith
```

クラスocfのリソースエージェントの一覧表示

```
# crm ra list ocf
```

クラスocf, プロバイダheartbeatのリソースエージェントの一覧表示

```
# crm ra list ocf heartbeat
```



■ crmコマンドでリソースエージェントの情報を表示(続き)

例) 仮想IPのリソースエージェント IPAddr2の情報を表示

```
# crm ra info IPAddr2 (抜粋表示)
```

```
Manages virtual IPv4 addresses (Linux specific version) (ocf:heartbeat:IPAddr2)
```

```
Parameters (* denotes required, [] the default):
```

```
ip* (string): IPv4 address
```

```
The IPv4 address to be configured in dotted quad notation, for example  
"192.168.1.1".
```

リソースエージェント IPAddr2のファイルのパスと種類(SLSE11 HAの例)

```
# file /usr/lib/ocf/resource.d/heartbeat/IPAddr2
```

```
/usr/lib/ocf/resource.d/heartbeat/IPAddr2: POSIX shell script text
```

例) Apache Webサーバのリソースエージェント apacheの情報を表示

```
# crm ra info apache (抜粋表示)
```

```
Manages an Apache Web server instance (ocf:heartbeat:apache)
```

```
Parameters (* denotes required, [] the default):
```

```
configfile (string, [/etc/apache2/httpd.conf]): configuration file path
```

```
httpd (string, [/usr/sbin/httpd]): httpd binary path
```

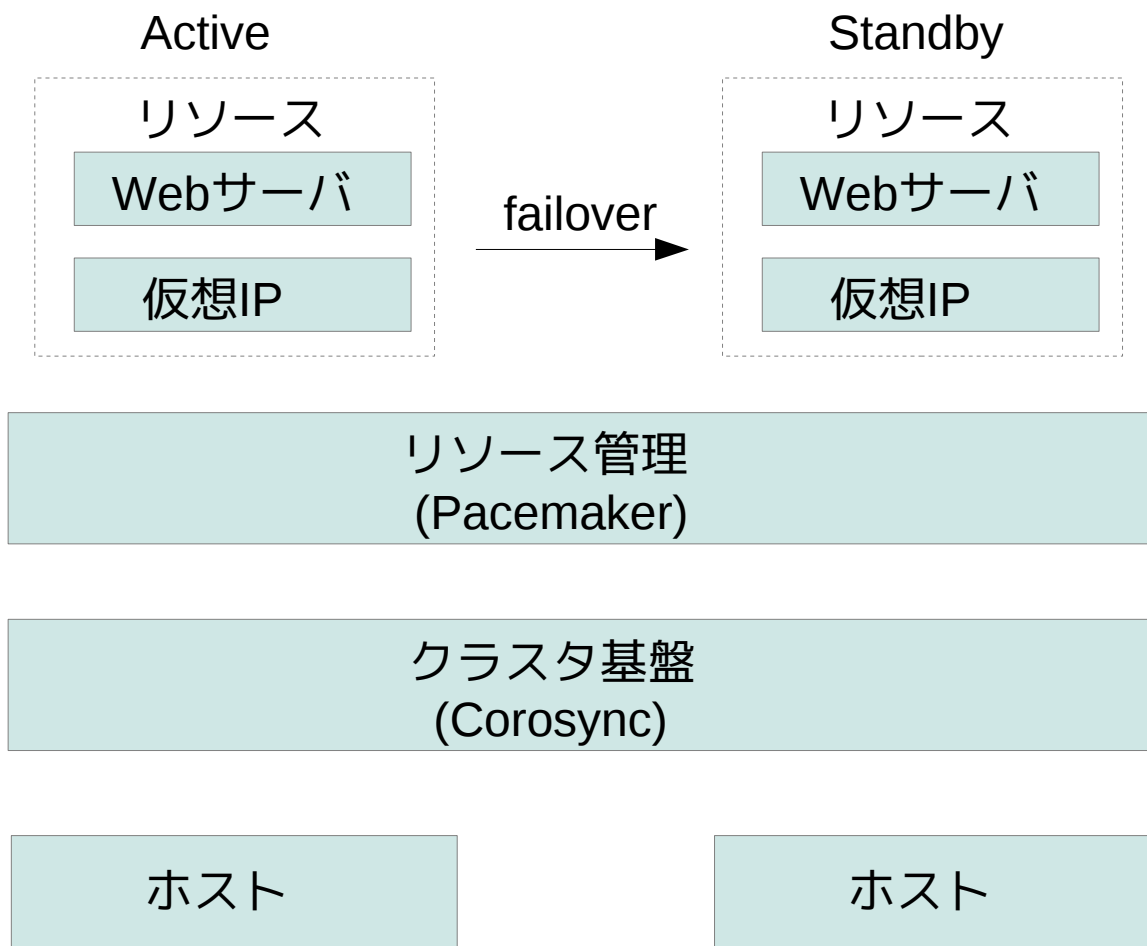
リソースエージェント apacheのファイルのパスと種類(SLSE11 HAの例)

```
# file /usr/lib/ocf/resource.d/heartbeat/apache
```

```
/usr/lib/ocf/resource.d/heartbeat/IPAddr2: POSIX shell script text
```

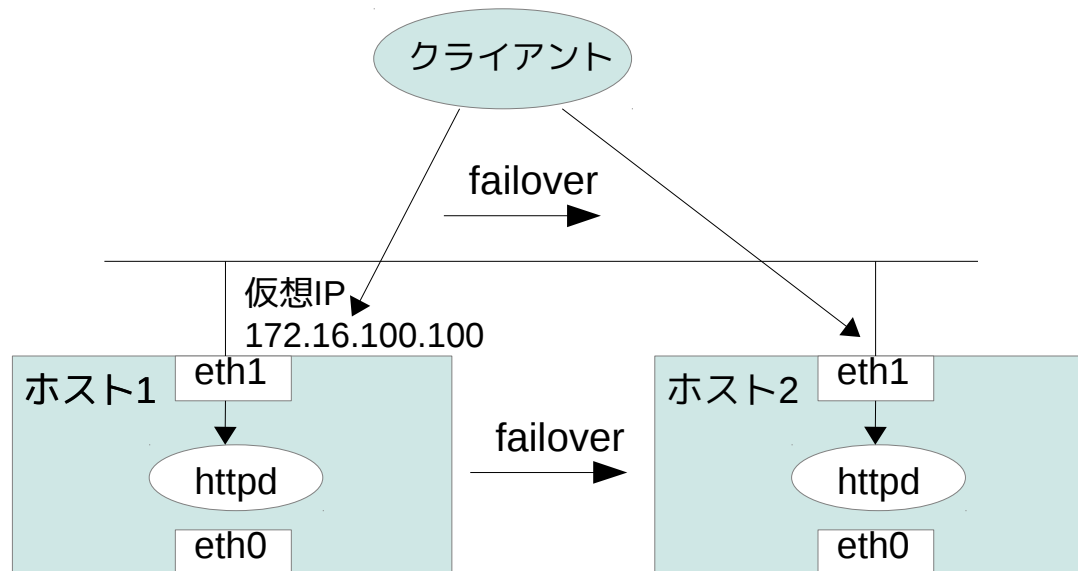



■ 「仮想IP+Apache Webサーバ」のActive/Standby構成例





■ 「仮想IP+Apache Webサーバ」のActive/Standby構成例(続き)





■ crmコマンドによる設定例

[リソースエージェントIPAddr2とapacheの登録]

```
lx02-sles-kvm-1:~ # crm configure primitive vip_172.16.100.100 ocf:heartbeat:IPAddr2 \  
  params ip="172.16.100.100" cidr_netmask="16" nic="eth1 op monitor interval="5s"  
lx02-sles-kvm-1:~ # crm configure op_defaults timeout=70s  
lx02-sles-kvm-1:~ # crm configure primitive apache ocf:heartbeat:apache op monitor interval="5s"  
lx02-sles-kvm-1:~ # crm configure group vip_apache vip_172.16.100.100 apache
```

```
lx02-sles-kvm-1:~ # crm configure show (抜粋表示)  
node lx02-sles-kvm-1 \  
  attributes standby="off"  
node lx02-sles-kvm-2 \  
  attributes standby="off"  
primitive apache ocf:heartbeat:apache \  
  op monitor interval="5s"  
primitive vip_172.16.100.100 ocf:heartbeat:IPAddr2 \  
  params ip="172.16.100.100" cidr_netmask="16" nic="eth1"  
  op monitor interval="5s"  
group vip_apache vip_172.16.100.100 apache  
property $id="cib-bootstrap-options" \  
  expected-quorum-votes="2" \  
  no-quorum-policy="ignore" \  
  stonith-enabled="false"  
rsc_defaults $id="rsc-options" \  
  migration-threshold="1"  
op_defaults $id="op-options" \  
  timeout="70s"
```



■ crmコマンドによる設定例(続き)

[動作確認]

```
lx02-sles-kvm-1:~ # crm status (抜粋表示)
2 Nodes configured, 2 expected votes
2 Resources configured.
Online: [ lx02-sles-kvm-1 lx02-sles-kvm-2 ]
Resource Group: vip_apache
  vip_172.16.100.100 (ocf::heartbeat:IPaddr2): Started lx02-sles-kvm-1
  apache (ocf::heartbeat:apache): Started lx02-sles-kvm-1
```

```
[ichiki@lx01 ~]$ wget -O - http://172.16.100.100 (抜粋表示)
0% [          ] 0      --.-K/s   This is kvm-1.
100%[=====>] 15      --.-K/s 時間 0s
```

(ホストlx02-sles-kvm-1を停止)

```
lx02-sles-kvm-2:~ # crm status (抜粋表示)
Online: [ lx02-sles-kvm-2 ]
OFFLINE: [ lx02-sles-kvm-1 ]
Resource Group: vip_apache
  vip_172.16.100.100 (ocf::heartbeat:IPaddr2): Started lx02-sles-kvm-2
  apache (ocf::heartbeat:apache): Started lx02-sles-kvm-2
```

```
[ichiki@lx01 ~]$ wget -O - http://172.16.100.100 (抜粋表示)
0% [          ] 0      --.-K/s   This is kvm-2.
100%[=====>] 15      --.-K/s 時間 0s
```



■ crmコマンドによる設定例(続き)

(ホストlx02-sles-kvm-1を起動)

```
lx02-sles-kvm-2:~ # crm status
```

```
Last updated: Mon Jun 16 18:40:25 2014
```

```
Last change: Mon Jun 16 17:52:10 2014 by root via cibadmin on lx02-sles-kvm-1
```

```
Stack: classic openais (with plugin)
```

```
Current DC: lx02-sles-kvm-2 - partition with quorum
```

```
Version: 1.1.9-2db99f1
```

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

```
2 Resources configured.
```

```
Online: [ lx02-sles-kvm-1 lx02-sles-kvm-2 ]
```

```
Resource Group: vip_apache
```

```
vip_172.16.100.100 (ocf::heartbeat:IPaddr2): Started lx02-sles-kvm-2
```

```
apache (ocf::heartbeat:apache): Started lx02-sles-kvm-2
```

リソースapacheの起動と停止

```
lx02-sles-kvm-1:~ # crm resource start apache
```

```
lx02-sles-kvm-1:~ # crm resource stop apache
```

リソースapacheの削除

```
lx02-sles-kvm-1:~ # crm configure delete apache
```



■ crmコマンドによる設定例(続き)

ノード lx02-sles-kvm-2 のstandby属性の管理

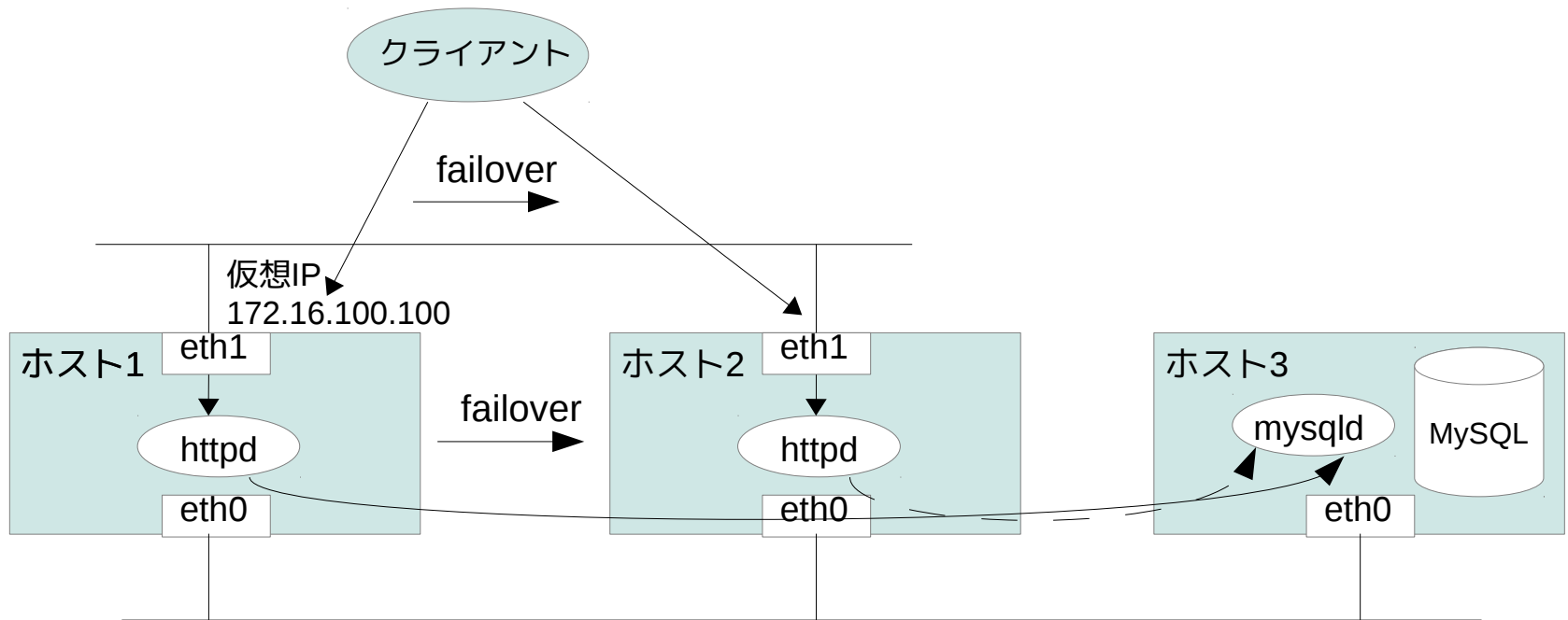
```
lx02-sles-kvm-1:~ # crm node online lx02-sles-kvm-2  
lx02-sles-kvm-1:~ # crm node standby lx02-sles-kvm-2
```

ノード lx02-sles-kvm-2 の削除

```
lx02-sles-kvm-1:~ # crm node delete lx02-sles-kvm-2
```



■ Pacemaker+Corosyncによる「仮想IP+Apache Webサーバ」のActive/Standby構成例





■ デモ1 確認 --Pacemaker編--

1. Webサーバー1のプロセスを止める（ソフトウェアの障害が起きたと仮定）
`/etc/init.d/apache2 stop`

2. PacemakerがWebサーバー1の状態を検知して、
Webサーバー2をactiveにすることを確認

注) 「`op monitor interval="5s"`」による監視でapacheの停止を検知します。

```
primitive apache ocf:heartbeat:apache \  
    op monitor interval="5s"
```

3. WebブラウザでWebサーバー2で動作することを確認

■ デモ2 確認 --Corosync編--

1. Webサーバー1の仮想マシンを止める（ハードウェアの障害が起きたと仮定）
`halt`、あるいは`init 0`

2. PacemakerがWebサーバー1の状態を検知して、
Webサーバー2をactiveにすることを確認

3. WebブラウザでWebサーバー2で動作することを確認



■ ホスト1→稼働、ホスト2→稼働

The screenshot shows a web browser window on the left and two virtual machine windows on the right. The browser displays a web page titled 'Web Server 1' with a form for course information. The form fields are as follows:

course_id	KDC-J02
course_title	Java programming 2
topic	Advanced topic of Java programming such as GUI.
day_length	3
price	90000
level_id	2
category	JAVA

The two virtual machines on the right are 'sles11-kvm-g1 仮想マシン' and 'sles11-kvm-g2 仮想マシン'. Both have terminal windows open, showing the root prompt and the current directory as ~/Desktop.



■ ホスト1→停止、ホスト2→稼働

The screenshot shows a web browser window on the left displaying a course details page for 'single course'. The page includes a table with the following data:

course_id	KDC-J02
course_title	Java programing 2
topic	Advanced topic of Java programing such as GUI.
day_length	3
price	90000
level_id	2
category	JAVA

Below the table is a '送信' (Send) button. To the right, two virtual machine windows are visible. The top window is titled 'sles11-kvm-g2 仮想マシン' and shows a terminal window with the prompt 'lx02-sles-kvm-2:~/Desktop #'. The bottom window is titled 'sles11-kvm-g1 仮想マシン' and shows a message at the bottom: 'ゲストは稼働していません' (Guest is not running).



■ OCFS2をリソースとして管理する場合に必要なPacemakerのコンポーネントは次のうちどれですか？
1つ選択して下さい。

- A. controld
- B. o2cb
- C. nfsserver
- D. Filesystem



■ OCFS2をリソースとして管理する場合にPacemakerのコンポーネントとして必要でないものは次のうちどれですか？
1つ選択して下さい。

A. controld

B. o2cb

C. nfssserver

D. Filesystem



■ OCFS2

- OCFS2(Oracle Cluster File System バージョン 2)はクラスタの全ノードから同時にアクセスされる共有ストレージ上で利用するクラスタファイルシステム
- カーネルモジュール `ocfs2.ko` として実装されている
- OCFS の後継のファイルシステムとしてオラクル社が開発



■ OCFS2に必要なリソースエージェント

OCFS2をPacemakerのリソースとして利用するには、以下の3つのリソースエージェントが必要

- ocf:pacemaker:controld
DLM(Distributed Lock Manager)
- ocf:ocfs2:o2cb
OCFS2クラスタ
- ocf:heartbeat:Filesystem
汎用ファイルシステム

OCFS2のように、複数のノードが特定のリソースを同時に実行するActive/Active構成のクラスタの場合は、リソースをクローンとして設定する必要がある。

- * nfsserver(ocf:heartbeat:nfsserver)はNFSサーバを管理するリソースエージェントです。
OCFS2を管理するリソースエージェントではありません。



- HAProxyの設定が以下の時、urlパスが「/images」以外で始まるリクエストをすべてbackendセクションで定義されたdefault-serverに送るために空欄に入れるキーワードはどれですか？ 1つ選択して下さい。

```
frontend http-front
  bind *:80
  acl url_images path_beg /images
  use_backend image-server if url_images
  _____ default-server
backend image-server
  server server1 172.16.210.62:80
backend default-server
  server server2 172.16.210.67:80
```

- A. use_backend
- B. use_backend_default
- C. default_backend
- D. default_server



- HAProxyの設定が以下の時、urlパスが「/images」以外で始まるリクエストをすべてbackendセクションで定義されたdefault-serverに送るために空欄に入れるキーワードはどれですか？ 1つ選択して下さい。

```
frontend http-front
  bind *:80
  acl url_images path_beg /images
  use_backend image-server if url_images
  _____ default-server
backend image-server
  server server1 172.16.210.62:80
backend default-server
  server server2 172.16.210.67:80
```

- A. use_backend
- B. use_backend_default
- C. default_backend
- D. default_server

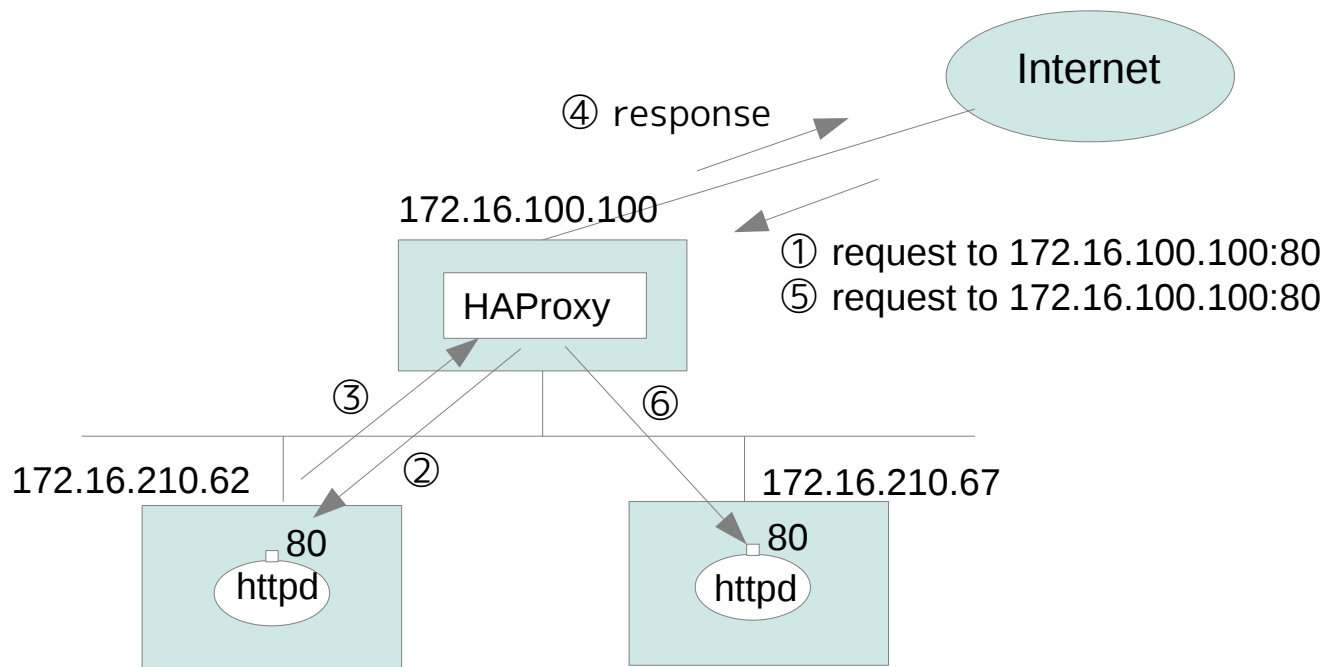


■ HAProxy

- HTTP/TCP のリバースプロキシであり、Layer7(アプリケーション層)のロードバランサー
- リクエストされた URL や HTTP ヘッダによってバックエンドサーバを振り分けることができる
- Layer4(トランスポート層)のロードバランサーとしても動作する
- Willy Tarreau 氏によって開発された



■ロードバランサーの例





■ L4ロードバランサー設定例

/etc/haproxy/haproxy.conf

```
global
  daemon
  maxconn 256

defaults
  mode http
  timeout connect 5000ms
  timeout client 50000ms
  timeout server 50000ms

frontend http-in
  bind *:80
  default_backend servers

backend servers
  server kvm-1 172.16.210.62:80
  server kvm-2 172.16.210.67:80
```



■ L4ロードバランサーの動作確認

kvm-1 (バックエンドサーバ1)

```
lx02-sles-kvm-1:~ # /etc/init.d/openais stop
lx02-sles-kvm-1:~ # cat /srv/www/htdocs/index.html
This is kvm-1.
lx02-sles-kvm-1:~ # /etc/init.d/apache2 start
```

kvm-2 (バックエンドサーバ2)

```
lx02-sles-kvm-2:~ # /etc/init.d/openais stop
lx02-sles-kvm-2:~ # cat /srv/www/htdocs/index.html
This is kvm-2.
lx02-sles-kvm-2:~ # /etc/init.d/apache2 start
```

lx02 (HAProxy ホスト)

```
lx02:~ # /etc/init.d/haproxy restart
```

lx01 (クライアント)

```
[ichiki@lx01 ~]$ wget -O - http://lx02 (抜粋表示)
0% [          ] 0    --.-K/s    This is kvm-1.
100%[=====>] 15    --.-K/s 時間 0s
[ichiki@lx01 ~]$ wget -O - http://lx02 (抜粋表示)
0% [          ] 0    --.-K/s    This is kvm-2.
100%[=====>] 15    --.-K/s 時間 0s
```



■ L7ロードバランサー設定例

/etc/haproxy/haproxy.conf

```
global
  daemon
  maxconn 256
defaults
  mode http
  timeout connect 5000ms
  timeout client 50000ms
  timeout server 50000ms
frontend http-in
  bind *:80
  acl url_images path_beg /images
  use_backend image-server if url_images
  default_backend default-server
backend image-server
  server kvm-1 172.16.210.62:80
backend default-server
  server kvm-2 172.16.210.67:80
```



■ L7ロードバランサーの動作確認

kvm-1 (バックエンドサーバ1)

```
lx02-sles-kvm-1:~ # /etc/init.d/openais stop
lx02-sles-kvm-1:~ # cat /srv/www/htdocs/images/index.html
This is kvm-1/images.
lx02-sles-kvm-1:~ # /etc/init.d/apache2 start
```

kvm-2 (バックエンドサーバ2)

```
lx02-sles-kvm-2:~ # /etc/init.d/openais stop
lx02-sles-kvm-2:~ # cat /srv/www/htdocs/index.html
This is kvm-2.
lx02-sles-kvm-2:~ # /etc/init.d/apache2 start
```

lx02 (HAProxy ホスト)

```
lx02:~ # /etc/init.d/haproxy restart
```

lx01 (クライアント)

```
[ichiki@lx01 ~]$ wget -O - - http://lx02/images (抜粋表示)
0% [          ] 0    --.-K/s    This is kvm-1/images.
100%[=====>] 22    --.-K/s 時間 0s
[ichiki@lx01 ~]$ wget -O - - http://lx02 (抜粋表示)
0% [          ] 0    --.-K/s    This is kvm-2.
100%[=====>] 15    --.-K/s 時間 0s
```



- HAパッケージのインストール
GUIツール「yast2 sw_single」を実行
- クラスタの初期化(最初のノードの自動設定)
シェルスクリプト「sleha-init」を実行
対話的に必要な情報を入力することで設定ができる
- クラスタへのノードの追加
シェルスクリプト「sleha-join」を実行
- GUIツール、Webインターフェイスによる管理
crmコマンドの他にGUIツール「crm_gui」あるいは
Webインターフェイス「Hawk」でもPacemakerの管理ができる
- 技術ドキュメント

https://www.suse.com/ja-jp/documentation/sle_ha/book_sleha/data/book_sleha.html



- GUIツール「crm_gui」

The screenshot shows the Pacemaker GUI interface. The window title is "Pacemaker GUI". The menu bar includes "接続(C)", "ビュー(V)", "シャドウ(S)", "ツール(T)", and "ヘルプ(H)". The toolbar contains various icons for navigation and actions.

ライブ

- 構成
 - CRMの構成
 - リソースのデフォルト
 - 操作のデフォルト
- ノード
- リソース
- 制約
- ACLs
- Fencing Topology

保守

名前	状態	詳細
クラスタ	● QUORUM有	Openais & Pacemaker
ix02-sles-kvm-1	● オンライン (dc)	
ix02-sles-kvm-2	● オンライン	
リソース	●	
vip_apache	● グループ	
vip_172.16.100.100	● 起動中 on [ix02-sles-kvm-1]	ocf::heartbeat:IPAddr2
apache	● 起動中 on [ix02-sles-kvm-1]	ocf::heartbeat:apache

Validate With: pacemaker-1.2
Epoch: 97
Num Updates: 9
CRM Feature Set: 3.0.7
Have Quorum: 1
DC UUID: ix02-sles-kvm-1
CIB Last Written: Tue Jun 17 21:56:38 2014
Update Origin: ix02-sles-kvm-1
Update Client: cibadmin
Update User: root

接続中...hacluster@127.0.0.1 (簡単モード)



- LPIレベル3 304では、クラウドとその高可用性に関する技術項目（仮想化、ロードバランシング、クラスタ管理、クラスタストレージ）を広範に取り上げている
- LPIは、Linuxのタイムリーな技術を提供し、技術力の底上げに貢献



■ LPIブースにてLPIC模擬問題を行っています。

- 各カテゴリ毎の模擬試験(5問5分程度)やっています！！
 - ・ ネットワーク
 - ・ セキュリティ
 - ・ 仮想化・クラウド



- 全問正解でLPIグッズプレゼントしています！！