試して覚える Pacemaker-2.0入門 『構築・リソース設定』

2020/10/23 OSC2020 Online/Fall Linux-HA Japan プロジェクト 三浦 貴紀



自己紹介

三浦 貴紀(みうら たかのり)



OSCがオンライン開催になってから初参加です



普段はNTT OSSセンタという所で 高信頼技術をやってます。



目次

①Pacemakerの概要



②Pacemaker-2.0で変わったこと



③構築・リソース設定の紹介



目次

1 Pacemakerの概要



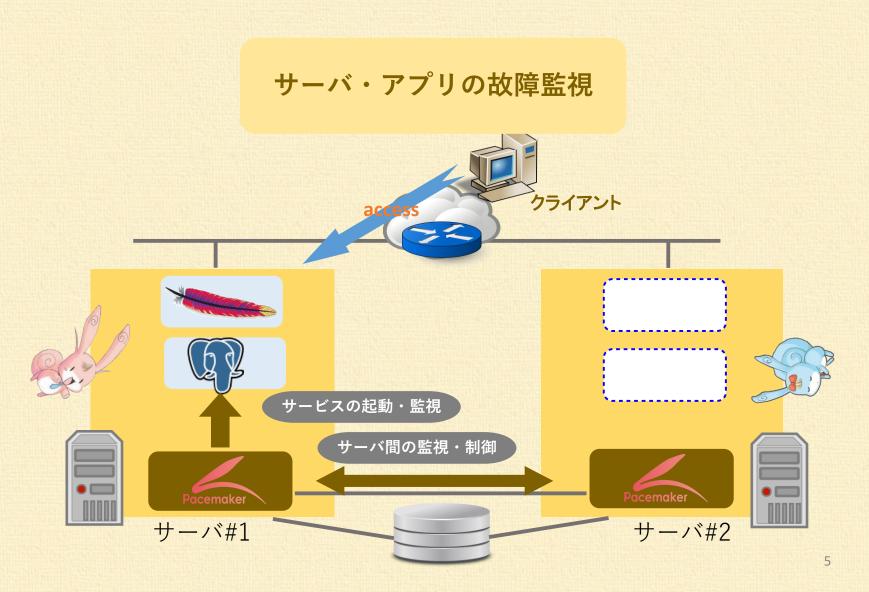
②Pacemaker-2.0で変わったこと



③構築・リソース設定の紹介

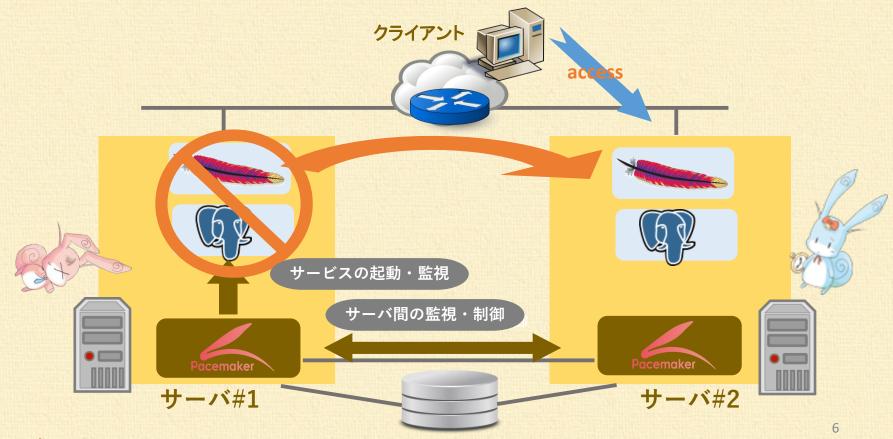


Pacemakerの概要(1/2)



Pacemakerの概要(2/2)

故障検知時に自動的に フェイルオーバ ダウンタイムの 最小化 STONITH※による データの安全性確保



もっと知りたい方はこちらまで

http://linux-ha.osdn.jp/



目次

①Pacemakerの概要



②Pacemaker-2.0で変わったこと



③構築・リソース設定の紹介



Pacemaker-2.0では特にここが変わりました

✓パッケージ提供

これまで(Pacemaker-1.1)はLinux-HA Japanからパッケージを提供していましたが、Pacemaker-2.0からはRHEL/CentOS 8 High Availability Add-On (以下、HA Add-On)の利用を推奨します。

▶利用手順・設定もRed Hatの方針に近くなります

✓構築・リソース設定

Pacemaker-1.1ではクラスタ管理ツールにcrmの利用を推奨していましたが、 Pacemaker-2.0からはpcsに変更します。

- ▶構築・リソース設定が割と変わります
- ▶ただできるだけ今まで通りに使えるようLinux-HA Japanから役立つツールを提供しています!

✓STONITH必須

Pacemaker-2.0ではSTONITH有構成のみを推奨します。 ▶共有ディスクを介した代替STONITH方式等も利用できます。

Pacemaker-2.0では特にここが変わりました

✓パッケージ提供

これまで(Pacemaker-1.1)はLinux-HA Japanからパッケージを提供していましたが、Pacemaker-2.0からはRHEL/CentOS 8 High Availability Add-On (以下、HA Add-On)の利用を推奨します。

▶利用手順・設定もRed Hatの方針に近くなります

✓構築・リソース設定

Pacemaker-1.1ではクラスタ管理ツールにcrmの利用を推奨していましたが、 Pacemaker-2.0からはpcsに変更します。

- ▶構築・リソース設定が割と変わります
- ▶ただできるだけ今まで通りに使えるようLinux-HA Japanから役立つツールを提供しています!

✓STONITH必須

Pacemaker-2.0ではSTONITH有構成のみを推ってこを中心に話します ▶共有ディスクを介した代替STONITH方式等も利か。

ここからは

目次

①Pacemakerの概要



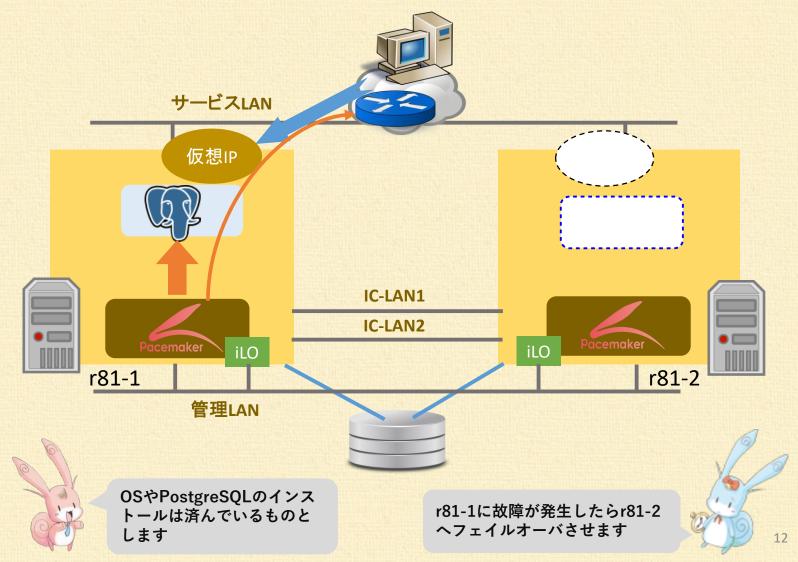
②Pacemaker-2.0で変わったこと



③構築・リソース設定の紹介



今日はこんな感じのクラスタを 作る手順を紹介します



目次

①Pacemakerの概要



②Pacemaker-2.0で変わったこと



③構築・リソース設定の紹介





先ずは構築から

構築手順(1/2)

- 1. インストール ■両系で
 - # dnf install pcs pacemaker fence-agents-all -enablerepo=HighAvailability
- 2. クラスタが利用するポートの許可 ■両系で
 - # firewall-cmd -permanent -add-service=high-availability # firewall-cmd -add-service=high-availability
- **3.** pcsdサービスの起動 ■両系で
 - # systemctl start pcsd.service
 # systemctl enable pcsd.service
- 4. hacluster ユーザのパスワード設定 ■両系で
 - # passwd hacluster

Changing password for user hacluster.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

構築手順(2/2)

5. クラスタノードの認証設定 ★いずれか一方のノードで

pcs host auth r81-1 addr= <r81-1の管理LAN> r81-2 addr= <r81-2の管理LAN>

Username: hacluster

Password:

r81-1: Authorized r81-2: Authorized

6. クラスタの作成 ●5を実行したノードで

pcs cluster setup < クラスタ名(任意)> r81-1 addr=<r81-1のIC-LAN1> addr=<r81-1のIC-LAN2> r81-2 addr=<r81-2のIC-LAN1> addr=<r81-2のIC-LAN2>



今まで" corosync-keygen -l"で作った認証ファイルを手動で各 ノードへ配置していたのが「4. クラスタノードの認証設定」に変わりました

今まで"corosync.conf"を手動で作成していたのが「5. クラスタの作成」に代わりました



目次

①Pacemakerの概要



②Pacemaker-2.0で変わったこと



③構築・リソース設定の紹介

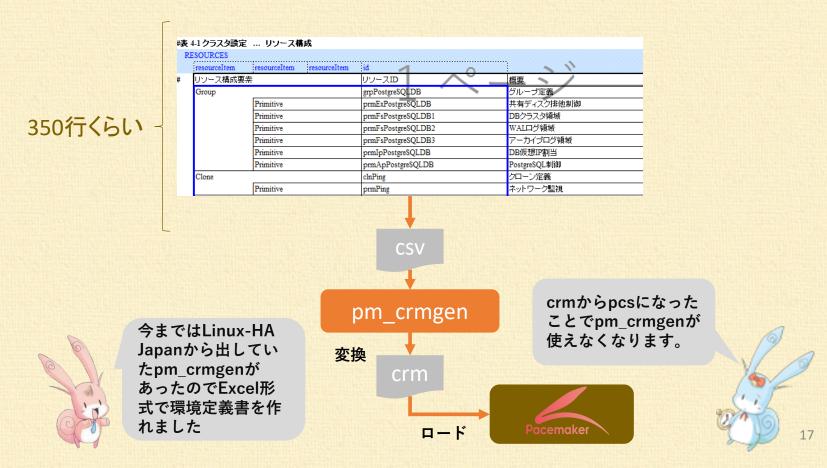


次はこっち



リソース設定の流れ(1/3)

これまで(Pacemaker-1.1系)リソース設定はExcel形式の環境定義書を変換してPacemakerへロードしていました。



リソース設定の流れ(2/3)

Red Hatのお作法に従いpcsで構築する際には、これらのpcsコマンドを一から作成しなければなりません。

pcs resource defaults resource-stickiness=200

pcs resource defaults migration-threshold=1

pcs resource create filesystem1 ocf:heartbeat:Filesystem device="/dev/mapper/mpatha2" directory="/dbfp/pgdata" fstype="xfs" force_unmount="safe" op start timeout=60s on-fail=restart monitor timeout=60s interval=10s on-fail=restart stop timeout=60s on-fail=fence

pcs resource create filesystem2 ocf:heartbeat:Filesystem device="/dev/mapper/mpatha3" directory="/dbfp/pgwal" fstype="xfs" force_unmount="safe" op start timeout=60s on-fail=restart monitor timeout=60s interval=10s on-fail=restart stop timeout=60s on-fail=fence

pcs resource create filesystem3 ocf:heartbeat:Filesystem device="/dev/mapper/mpatha4" directory="/dbfp/pgarch" fstype="xfs" force_unmount="safe" op start timeout=60s on-fail=restart monitor timeout=60s interval=10s on-fail=restart stop timeout=60s on-fail=fence

pcs resource create ipaddr ocf:heartbeat:lPaddr2 ip="192.168.1.87" nic="ens4" cidr_netmask="24" op start timeout=60s on-fail=restart monitor timeout=60s interval=10s on-fail=restart stop timeout=60s on-fail=fence

pcs resource create pgsql ocf:linuxhajp:pgsql pgctl="/usr/pgsql-11/bin/pg_ctl" psql="/usr/pgsql-11/bin/psql" pgdata="/dbfp/pgdata/data" pgdba="postgres" pgport="5432" pgdb="template1" op start timeout=300s on-fail=restart monitor timeout=60s interval=10s on-fail=restart stop timeout=300s on-fail=fence # pcs resource group add pgsql-group filesystem1 filesystem2 filesystem3 ipaddr pgsql

pcs resource create ping ocf:pacemaker:ping name="ping-status" host_list="192.168.1.1" attempts="2" timeout="2" debug="true" op start timeout=60s on-fail=restart monitor timeout=60s interval=10s on-fail=restart stop timeout=60s on-fail=fence

pcs resource clone ping

pcs stonith create fence1-ipmilan fence_ipmilan delay="10" pcmk_host_list="r81-1" ip="192.168.2.85" username="root" password="XXXXXX" lanplus="1" ipmitool_path="/usr/bin/ipmitool_stub" op start timeout=60s on-fail=restart monitor timeout=60s interval=3600s on-fail=restart stop timeout=60s on-fail=ignore # pcs stonith create fence2-ipmilan fence_ipmilan delay="0" pcmk_host_list="r81-2" ip="192.168.2.86" username="root" password="XXXXXX" lanplus="1" ipmitool_path="/usr/bin/ipmitool_stub" op start timeout=60s on-fail=restart monitor timeout=60s interval=3600s on-fail=restart stop timeout=60s on-fail=ignore # pcs constraint location pgsql-group prefers r81-1=200

pcs constraint location pgsql-group prefers r81-2=100

pcs constraint location fence1-ipmilan avoids r81-1

pcs constraint location fence2-ipmilan avoids r81-2

pcs constraint location pgsql-group rule score=-INFINITY ping-status It 1 or not defined ping-status

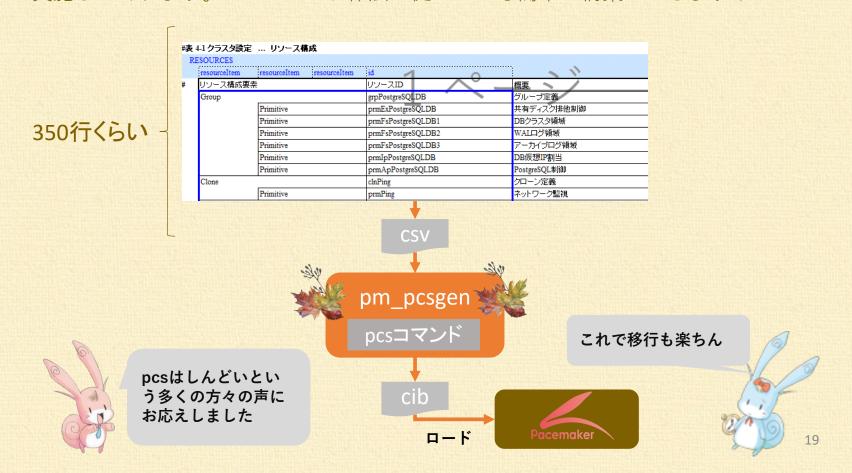
pcs constraint colocation add pgsql-group with ping-clone score=INFINITY

pcs constraint order ping-clone then pgsql-group symmetrical=false

正直しんどい!なので

リソース設定の流れ(3/3)

Excel形式の環境定義書からpcsコマンドを生成するpm_pcsgenというツールを作りました!作成したpcsコマンドからクラスター設定ファイル(cib)を作成するところも自動で実施してくれます。Red Hatのお作法に従いつつも簡単に構築ができます!



pm_extra_toolsについて(1/2)

pm_pcsgenはpm_extra_toolsというパッケージに 詰めてここに置いてます

http://linux-ha.osdn.jp/wp/archives/4963



LINUX-HA JAPAN High-Availability Clustering on Linux

HOME

ダウンロード&インストール マニュアル メーリングリスト デスクトップテーマ・壁紙等

握り出し物

ニュース リリース情報

イベント情報

読み物 WEBラジオ

その他

pm extra_tools-1.0-1 リリースノート»

pm extra tools-1.1-1 リリースノート

BY KSK, ON 9月 25TH, 2020



〒 ● サツイート



pm_extra_tools パッケージは Pacemaker-2.0系用の追加ツールをパッケージしたものです。 RHEL 8 High Availability Add-On(以下 HA Add-On) / CentOS 8 に同梱されている Pacemaker-2.0系 と組み合わせて利用します。

pm_extra_toolsについて(2/2)

pm_extra_toolsにはpm_pcsgenの他にRA(リソースエージェントも入ってます)

1. pgsql RA

HA Add-onにも含まれていますがそちらはPG-REXに対応していません。 PG-REXやPostgreSQL(コミュニティ版)を考えているならこちらを使用してください。

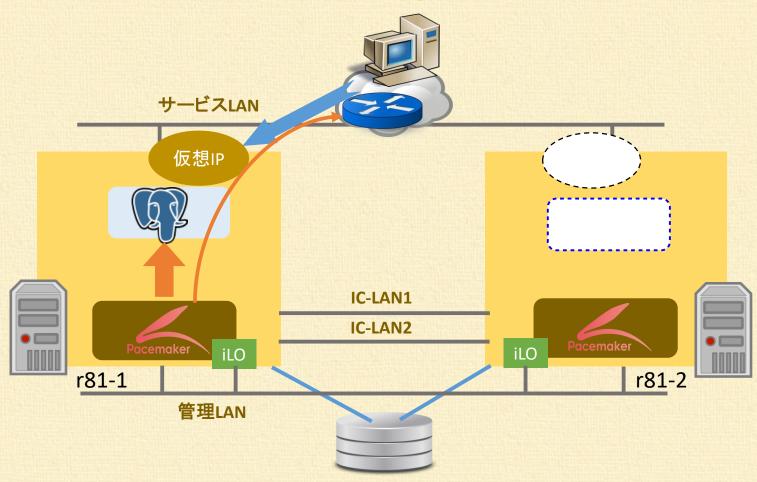
2. hulft RA

セゾン情報システムズ社のデータ連携ソフトウェアであるHULFT用の RAです。

ではリソース設定の話に戻ります

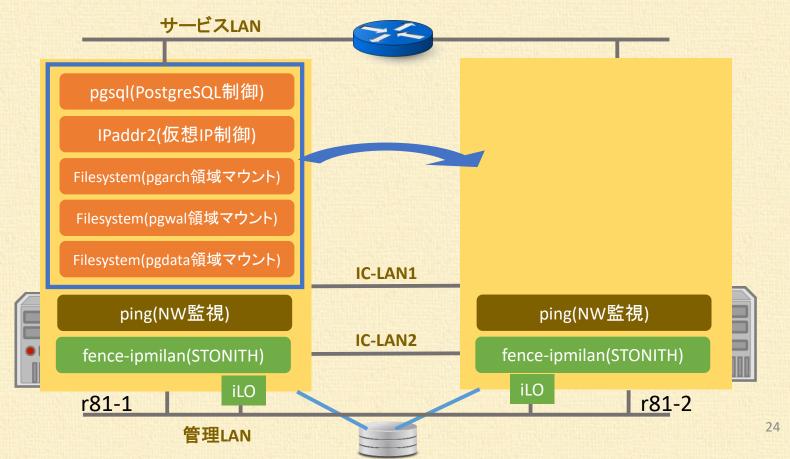
今まで通り、Excelでリソース設定ができると古参のユーザが安心できたところで、初回の方でもリソース設定ができるよう、ここからリソース設定の話に戻ります。

改めて今日はこんな感じの クラスタを作ります



リソース構成としてはこうします

- 共有ディスクマウント/仮想IP/PostgreSQLをどちらか一方(r81-1)で動作させます。これらは常にセットで動作させる必要があるため、グループ化して故障した際にはまとめてF/Oさせます。
- 両系でゲートウェイへのNW監視を行い、監視が失敗しているノードはActにさせません。
- IC-LANが全て切断された場合には、スプリットブレイン抑止のため、STONITHで対向ノードを電源断します。



リソース設定~サンプル置き場

pm_extra_toolsをインストールすると下記にExcel形式の環境定義書のサンプルが配置されます。前頁のリソース構成を実現するリソース設定をこのサンプルを元に解説します。サンプルにはいくつか表がありますが、今回の構成に関わる表1~11を順に解説します。

/usr/share/pm_extra_tools/pm_pcsgen_sample.xlsx

ちなみにサンプルの全体はこのぐらいあります 見えませんねる

# pm_poign 環境定義者 ファイル形式パーション: I.il	このファイルに含まれる文章は、クリエイティブコモン	ンズ 表示 - 柳州 4.0 國際 (CC BY-SA 4.0) によってライセンスされています。	1 PRIMITIV	E				#表 8-1 クラスタ影	ま STONITHリソ	-2			
	https://creativecommons.org/licenses/by-sa/4.0/				provider type			STONITH					
1	Copyright (C) 2020 NEPPON TELEGRAPH AND TELE	EPHONE CORPORATION	# UV-	ZID class ten2 ocf	provider type hearbest Filesy	ಪ್ರಕ			-ZID type			85	
『表 - クラスタ設定 クラスタ・ノード属性			A type	Interes poer	Ivalue Pricky	stem		fencel-ipmila		realise			
NODE				−夕種別 項目	IRotophik	数で		A lype //ラメータ科	HARD OF THE	Richt-Print		and the second	
4 ノード名 パラメータ種別 項目	I IPSTYN	68	option	s device	/dev/mappen/mpatha/t /dlnfp/pg/wal			entions	delay	10 10		ex o	
		-		fitype	x fe			1	penk_host_list	81-1			
				force unmount	safe				ip usemane	192.168.28.13 maccrask cr			
#表 2-1 クラスタ設定 クラスタ・プロバティ			O type	timecout ーションタイムアウト値	interval on-fail 転換開隔 障害時の動作	role start-delay 役別 起動的物	院時間 俗名		nsswed	innings l			
			a NAL	-51394A7514	To SCIRCING IN INCIDENT CONTROL	(VA) (2:40A) 471	(0.010) (0.95)		lanplus	ř.			
PROPERTY value			monito	r 60s	10x restart			O type	timeout ンタイムアウト値	SE SERVER	(ce-fail 障実時の動作	68	
a 項目 原治内容	al o	協考	stop	60x	fence			WAD-5	60s	To SCIONS	restart	10.5	
			PRIMITIV	F				monitor	60x	3600x	restart		
					provider type			stop	60x		ignore		
<u> </u>			4 119-		provider type	板等		STONITH					
#表 3-1 クラスタ設定 リソース・デフォルト			filesys	ion3 ocf	heatheat Filesy	stem							
RSC_DEFAULTS Traine Traine			# /(5x	- 夕種別 項目	設計内容	数で		# STONITH'J': fence2-ipmilia	-ZID type fence ip			数で	
a 項目 設定内容	概要	備考	option	s device	/dev/mappen/mpaths4			A type	1101102	value		_	
resource-stickiness 200	リソース割当て リソース故障可能回数			fiture	/dhfp/pgarch			4 バラメータ目	9 48	IRV27598		60	
migration-threshold	ラグース的障可能知識			force unmount	safe			options	penk_host_list	81.2			
4表 3-2 クラスタ設定 オペレーション・デフォルト			O type	-ションタイムアウト値	interval on-fail 監視開隔 障害時の動作	role start-delay 役制 起動前後	統時間 備考		ip	192.168.28.14			
OP DEFAULTS			a NAL	一ションタイムアウト船	製作的の機 国際実施の動作	(U/R) 20.50(R) (O	RUTIN SES		usemane	pacerrak er			
a 項目 Partys	野茶	協名	monito	r 60s	10s estart				password lumbus	ipmipass1			
· MCI MELIYM	als.	10.5	stop	60s	fence			O type	tincout	interval	on-fail		
			PR INSTITU					4 X/V-5:	ンタイムアウト値	監視関係	障害時の動作	领考	
#表 4-I クラスタ設定 リソース構成					provider (type			start	60x	1600s	restart		
RESOURCES US-XMI			4 119-		provider type freatheat Padd	数罗		stop	60x		ignose		
			ipald:	eef	restreat Paid	2		(T.) (DETAIL	定 リソース配置	Man () (v)			
# リソース構成要素 リソースID	概要	俗号	# /C53	- 夕種別 項目	接近内容	数要							
Croup pgsql-group Primitive filesystemi			option	s ip	192.168.201.130			60	professis	roeds node	Score		
Primitive filesystem2				cide network	24			# UV-ZID pgsql-group	prefessis	noids J-F	スコア	報考(※roleを設定	「したい場合は、表9-2を使用」
Primitive filesystem3			O type	limecoul	interval on-fail	role start-delay		bised-incab	presess	181-2	100		
Primitive ipaddr Primitive pasel			4 AVE	ーションタイムアウト値	監視管隔 隠害時の動作	役割 起動前待	接時間 備考	fencel-ipmila		r81-1			
Clone pina clone		CloneのDには「-Cloneが象リシースのD>-clone」を推定すること	start menits	r 60s	10s restart			Crocc2-ipmilia	avoids	81-2			
Primitive ping			stop	60x	fence			#表 9-2 クラスタ新	定 リソース配面 類	知 (ルール)			
Storith Encel ipmilia Storith Storith			***************************************					LOCATION RU	E		Total Name		
Montals Kncc2-sprinter			Post		Securitar true			4 UV-ZD			2.4 S		68
#表 5-1 クラスタ設定 リソース・パラメータ (meta)			# UV-		psovider type	数セ		pgsql-group	-INFINI	TY or ping-status	i f	-	
RSC_ATTRIBUTES	,		pgsql	ocf	linuxhaip pgsql			29122		ping-status	not_defined	$\overline{}$	
# リソースID 項目	Value IPは内容 M等		# 1/2×	- 夕神9 項目	IP:d::/hole	400							
77 700	10.00		option	s pgcd	/use/pgsql-11/bin/pg_cdl				9定 リソース同葉				
				psql	/use/pg.sql-11/bin/psql			COLOCATION					
#表 6-1 クラスタ設定 STONITHの実行順序				pycas	Allofo/pgdata/data postares			4 制約開降リン	-スm 制約対1	象リソースID スコア (質	(3+(引:ナ) 制約関連リソースの	役割 制約対象リソースの	役割 領考
STONITH LEVEL				pypod	3432			pgsql-group	ping-cle	one INFINITY			
node id	level			pydh	template1	into the second						+	
# STONITH対象ノード 実行するSTONITHリソース	D 実行順序 備考		4 AVT	ーションタイムアウト値	監視関隔 障害時の動作	役割 起動前待	施時間 協考						
15			start	300x	restart				9定 リソース起動				
			monito	r 60s	10s restart			ORDER					
#表 7-1 クラスタ設定 Primitiveリソース			in aup	Look	pama			# 先に記動する	リソースID 後に配	動するリソース! Optional/N	landstory/Sor 先起動リソースのア	クショ後起動けつスのアクシ	ション 総動と逆順に停止 (値
			PRIMITIV	E				ping-clone	pgsql-g	юшр			false
P id class per	rider type		a 199-		provider type	新 帝							
# UV-ZID class prov			ping	ocf etas	pacousker ping	94.90		91117					
Blesystemi per hear	theat Filesystem		A type	many	value	-		HUMBER MI					
a パラメータ種別項目 設定内容	概要		# /(5)	- 夕禄別 清日	19x4x4xW	E(0)							
options device (devinappents			option	host_list	192.168.201.254			22 22 23 27 27					
directory /dhfp/pgdsta				stlempts	á.								
fitype x fit				timeout	2								
O type tincout interval	on-fail sole	start-delay	O tree	(imaxwa)	Sinterval con-fail	Stole Start-delin							
# オペレーションタイムアウト値 監視関係	障害時の動作 役割	起影的传统時間 磁号	# 3/KL	ーションタイムアウト値	監視管隔 障害時の動作	役割 起動前待	総時間 勝考						
start 60x	restart		start	60x	ecstart								
tion 60s 10s	force		stee	e 60s	10x restart								
V-			100										

リソース設定解説(1/8) Excelサンプル 表 1~3

ここにはクラスタ全体に関わる設定を記述します。 基本的には常にサンプル通りでよいです。

#表 1	-1 クラスタ設定 クラスタ	タ・ノード属性										
NO	DDE											
	uname	type	name	value								
#	ノード名	パラメータ種別	項目	設定内容	備考							
#表 2	g 2-1 クラスタ設定 クラスタ・プロバティ											
PR	PROPERTY											
	name	value										
#	項目	設定内容	概	要	備考							
		_										
	-1 クラスタ設定 リソース	ス・デフォルト										
RS	C_DEFAULTS											
	name	value										
#	項目	設定内容	概		備考							
	resource-stickiness	200		ソース割当て								
	migration-threshold	1	リ	ソース故障可能回数								
	-2 クラスタ設定 オペレー	-ション・デフォルト										
OI	_DEFAULTS	.,										
	name	value										
#	項目	設定内容	概	要	備考							
				<u>'</u>		·						



今回は1回でも故障を検知したらF/Oさせます。「N回故障したらF/O」としたい場合は表3-1の"migration-threshold"にNを設定してください。

リソース設定解説(2/8) Excelサンプル 表 4

設定するリソースの種別を定義します。

- ・各リソースに任意の名前を付けリソースIDに指定します。リソースIDとRAの 紐づけは表7で行います
- 基本的にリソースは全て"Primitive"として定義します。その上で"Group"に組み込んで1セットにしたり、"Clone"に組み込んで両系で動作させたりします。
- STONITHリソースは"Stonith"で定義します。

#表 4	-1 クラスタ設定	リソース構成			
RE	SOURCES				
	resourceItem	resourceItem resourceItem	id		
#	リソース構成要	素	リソースID	概要	備考
	Group		pgsql-group		
		Primitive	filesystem1		
		Primitive	filesystem2		
		Primitive	filesystem3		
		Primitive	ip addr		
		Primitive	pgsql		
	Clone		ping-clone		CloneのIDには「 <clone対象リソースのid>-clone」を指定すること</clone対象リソースのid>
		Primitive	ping		
	Stonith		fence1-ip milan		
	Stonith		fence2-ip milan		



今回はpgsql-groupという名前の Groupにfilesystemからpgsqlまで のリソースを含めています。 備考にも書いていますが Cloneの名前の付け方には注 意です。Pacemaker-2.0で変 わったところです。



リソース設定解説(3/8) Excelサンプル 表 5~6

ここはもうサンプル通りです。すっ飛ばします。

#表 5	-1 クラスタ設定 リソース	・パラメータ(meta)									
RS	C_ATTRIBUTES										
)		value								
#	リソースID	項目	設定内容	備考							
#表 6	表 6-1 クラスタ設定 STONITHの実行順序										
ST	ONITH_LEVEL										
	≀ .		level								
#	STONITH対象ノード	実行するSTONITHリソースID	実行順序	備考							



Pacemaker-1.1の時はSTONITHリソース を複数使っていたので表6-1のSTONITHの 実行順序を使っていました。2.0からは1種 類になりましたので何も設定しません。

リソース設定解説(4/8) Excelサンプル 表 7 – サービスリソース

各リソースの動作を制御するパラメータを指定します。

- 以下はpgsql RAでPostgreSQL 11を制御する場合の例です。サンプルには他の設定例も記載されています。
- 各リソースにどのようなパラメータが指定できるかを調べたい時はHA Add-onナレッジの以下が参考になります。

▶10.2. リソース固有のパラメーターの表示

id リソースID pgsq1	class	provider					
		provider					
PESQI	ocf	linuxhajp	pgsql		概要		
	name	value					
パラメータ種別	項目	設定内容	設定内容				
options	pgct1	/usr/pgsql-11/bin/pg_ctl	/usr/pgsql-11/bin/pg_ctl				
	psq1	/usr/pgsq1-11/bin/psq1					
	pgdata	/dbfp/pgdata/data					
	pgdba	postgres	postgres				
	pgport	5432	5432				
	pgdb	template1	template1				
type	timeout	interval	on-fail	role		start-delay	
オペレーション	タイムアウト値	監視間隔	障害時の動作	役割		起動前待機時間	備考
start	300s		restart				
monitor	60s	10s	10s restart				

providerにlinuxhajpを指定すると pm_extra_toolsに含まれるpgsql RAを使用することになります。 コミュニティ版の PostgreSQLやPG-REXを使 う場合はpm_extra_tools版を 使ってください



リソース設定解説(5/8) Excelサンプル 表 7 – NW監視

NW監視のパラメータを指定します。

NW監視は常に両系で動作させ、サービスに必要なネットワーク経路の正常性を監視します。 host_listに監視先のIPを指定します。

	PRI	IMITIVE		,								
	P	id		class		provider		type				
#		リソースID		class		provider	provider type			概要		
		ping		ocf		pacemaker ping						
		type	name		value	ie						
#		パラメータ種別	項目		設定内容	内容			概要			
		options	name		ping-status							
			host_list		68.1.1							
			attempts		2							
			timeout		2							
			debug		true							
		type	timeout		interval		on-fail		role		start-delay	
#		オペレーション	シ タイムアウト値 監視間		監視間隔		障害時の動	协作	役割		起動前待機時間	備考
		start	60s			restart						
		monitor	60s		10s		restart					
		stop	60s				fence					



通常、ゲートウェイのIPをhost_listに指定 します。

リソース設定解説(6/8) Excelサンプル 表 8

STONITHリソースに渡すパラメータを指定します。

- 今回使用するfence_ipmilanはiLO経由で強制電源断を行うリソースです。そのため指定するパラメータはiLOのIPやユーザIDです。
- 下記はr81-1の電源断を行うSTONITHリソースです。このリソースは常に対向のr81-2で動作させる必要があります。これは次の表9で設定します。

	ST	ONITH								
	P	id		type						
#		STONITHリソー	·スID	type						
		fence1-ipmilan		fence_ip milan			L			
	A	type	name		value					
#		パラメータ種別	項目		設定内容		#			
		options	delay		10					
			pcmk_host_list		r81-1					
			ip		192.168.2.85					
			username		root					
			password		nttossc					
			lanplus		1					
			ipmitool_path		/usr/bin/ipmitool_stub		L			
		71	timeout		interval	on-fail				
#		オペレーション	タイムアウト	値	監視間隔	障害時の動作	#			
		start	60s			restart				
		monitor	60s		3600s	restart				
		stop	60s			ignore	L			
			101							

	ST	ONITH				
	P	id		type		
#		STONITHリソー	·スID	type		
		fence2-ip milan		fence_ip milan		
	A	type	name		value	
#		パラメータ種別	項目		設定内容	
		options	delay		0	
			pcmk_host_list		r81-2	
			ip		192.168.2.86	
			username		root	
			password		nttossc	
			lanp lus		1	
			ipmitool_path		/usr/bin/ipmitool_stub	
	0	type	timeout		interval	on-fail
#		オペレーション	タイムアウト	直	監視間隔	障害時の動作
		start	60s			restart
		monitor	60s		3600s	restart
		stop	60s			ignore



スプリットブレインになった際、相撃ちして両系が停止しないようr81-1をSTONITHする↑の設定ではdeleyというSTONITH発動の待機時間10秒を指定しています

r81-2側にはdelayに0秒 を指定しています



リソース設定解説(7/8) Excelサンプル 表 9

リソースを配置するノード/ルールを設定します。

- ・ リソース配置制約(ノード)では下記の2つの制約を課しています。
 - ▶ pgsql-groupはr81-1で優先的に動作させる(スコアがr81-1の方が高い)
 - ➤ fence1_ipmilan、 fence2_ipmilanはそれぞれr81-1、r81-2では動作させない(必ず対向をSTONITHさせる)

#表	9-1 クラスタ設定 リソース	配置制約(ノード)			
I	OCATION_NODE				
	rsc	prefers/avoids	• •	score	
#	リソースID	prefers/avoids	ノード	スコア	備考 (※roleを設定したい場合は、表9-2 を使用してください)
	pgsql-group	prefers	r81-1	200	
			r81-2	100	
	fence1-ipmilan	avoids	r81-1		
	fence2-ipmilan	avoids	r81-2		

• リソース配置制約(ルール)ではpgsql-groupはNW監視が異常(ping-statusが1未満)もしくはNW 監視が定義されていない環境では動作させないという制約を課しています。

#表 9-2 クラスタ設定 ... リソース配置制約(ルール)

	CATION_RULE	CATION_RULE										
	rsc			attribute	1	value	role					
#	リソースID	スコア	and/or	条件属性名	条件	条件値	役割	備考				
	pgsql-group	-INFINITY	or	ping-status	lt	1						
				ping-status	not_defined							
ě												



Pacemaker-1.1ではルール/ノードをまとめて一つの表にしていましたが、 Pacemaker-2.0から分割しました

リソース設定解説(8/8) Excelサンプル 表 10~11

リソース間の同居/起動順序制約を設定します。

• リソース同居制約ではpgsql-groupをping-cloneが動作しているノードで動作させるというリソース間の同居に関する制約を課しています。

#表 10-1 クラスタ設定 ... リソース同居制約

CO	DLOCATION					
	rsc				with-rsc-role	
#	制約関連リソースID	制約対象リソースID	スコア(重み付け)	制約関連リソースの役割	制約対象リソースの役割	備考
	pgsql-group	ping-clone	INFINITY			

• リソース順序制約ではping-cloneが起動した後にpgsql-statusを起動させるというリソース間の起動順序に関する制約を課しています。

#表 11-1 クラスタ設定 ... リソース起動順序制約

OI	ORDER										
	first-rsc	then-rsc		first-action	then-action	sy mmetrical					
#	先に起動するリソースID	後に起動するリソースID	Optional/Mandatory/Serial	先起動リソースのアクション	後起動リソースのアクション	起動と逆順に停止(y	備考				
	ping-clone	pgsql-group				false					
		_	_	_		_					



サンプル解説はここまでです。 次のページではリソース設定を反映させ る手順を紹介します

リソース設定反映手順

- 1. Excel環境定義書をCSVに変換し、いずれか一方のノードへ格納 ▶ここでは"sample.csv"とします。
- 2. xmlファイルへの変換 ●CSVを配置したノードで # pm_pcsgen sample.csv sample.xml (CIB), sample.sh (PCS) を出力しました。
- 3. Pacemakerの起動 ★いずれか一方のノードで # pcs cluster start --all
- 4. xmlファイルのロード ●CSVを配置したノードで # pcs cluster cib-push sample.xml



○○.csvをpm_pcsgenで変換 すると○○.xmlと○○.shが作 成されます。 sample.sh(PCS)の中には pcsでリソース設定を行う場 合のコマンドが羅列されてい ます。



構築・リソース設定が上手くいくと こんな感じになります♪

pcs status -full

```
•••省略•••
Node List:
 * Online: [ r81-1 (1) r81-2 (2) ]
Full List of Resources:
 * Resource Group: pgsql-group:
  * filesystem1 (ocf::heartbeat:Filesystem): Started r81-1 (ocf::heartbeat:Filesystem): Started r81-1
  * filesystem3 (ocf::heartbeat:Filesystem): Started r81-1
  * ipaddr (ocf::heartbeat:IPaddr2): Started r81-1
  * pgsql (ocf::linuxhajp:pgsql): Started r81-1
 * Clone Set: ping-clone [ping]:
  * ping (ocf::pacemaker:ping): Started r81-2
  * ping (ocf::pacemaker:ping): Started r81-1
  * Started: [ r81-1 r81-2 ]
 * fence1-ipmilan (stonith:fence ipmilan):
                                                  Started r81-2
 * fence2-ipmilan (stonith:fence ipmilan):
                                                  Started r81-1
Node Attributes:
 * Node: r81-1 (1):
  * ping-status
                             : 1
 * Node: r81-2 (2):
  * ping-status
```

Started ●●は●●で対象の リソースが起動していること を示しています



PostgreSQL故障が発生して F/Oするとこうなります

Node List:

* Online: [r81-1 (1) r81-2 (2)]



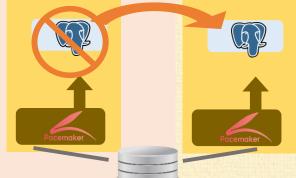
r81-2でリソースが起動してい ることが分かります

Full List of Resources:

- * Resource Group: pgsql-group:
- * filesystem1 (ocf::heartbeat:Filesystem): Started r81-2
- * filesystem2 (ocf::heartbeat:Filesystem): Started r81-2
- * filesystem3 (ocf::heartbeat:Filesystem): Started r81-2
- * ipaddr (ocf::heartbeat:IPaddr2): Started r81-2
- * pgsql (ocf::linuxhajp:pgsql): Started r81-2
- * Clone Set: ping-clone [ping]:
- * ping (ocf::pacemaker:ping): Started r81-2
- * ping (ocf::pacemaker:ping): Started r81-1
- * Started: [r81-1 r81-2]
- * fence1-ipmilan (stonith:fence ipmilan): Started r81-2
- * fence2-ipmilan (stonith:fence ipmilan): Started r81-1

Node Attributes:

- * Node: r81-1 (1):
- * ping-status : 1
- * Node: r81-2 (2):
- * ping-status : 1



↓のように故障情報が残ります

Failed Resource Actions:

* pgsql monitor 10000 on r81-1 'not running' (7): call=48, status='complete', exitreason='No process state file found', last-rc-change='2020-10-06 03:34:49 -04:00', queued=0ms, exec=0ms



という感じでPacemaker-1.1と ほとんど変わりません!

Pacemaker-1.1の環境定義書についての詳細は↓の付録Aで解説してますので興味がある方は見てください

<u>試して覚えるPacemaker入門 リソース設定編 ~ Pacemakerで</u> <u>ノードやサービスを手玉に取ろう! ~</u>

http://linux-ha.osdn.jp/wp/archives/4532

それでも分からないことがあればMLへ

http://linux-ha.osdn.jp/wp/ml





今後もPacemakerを よろしくお願いします



