

PacemakerでDockerコンテナを クラスタリング

2017年 3月 11日
OSC2017 Tokyo/Spring

Linux-HA Japan
竹下 雄大



本日の内容

- Pacemakerってなに？
- 最新版Pacemaker-1.1.15-1.1のご紹介
- PacemakerでDockerクラスタリング！

Pacemakerってなに？

Pacemakerはオープンソースの
HAクラスタソフトです

High **A**availability = 高可用性

つまり

サービス継続性

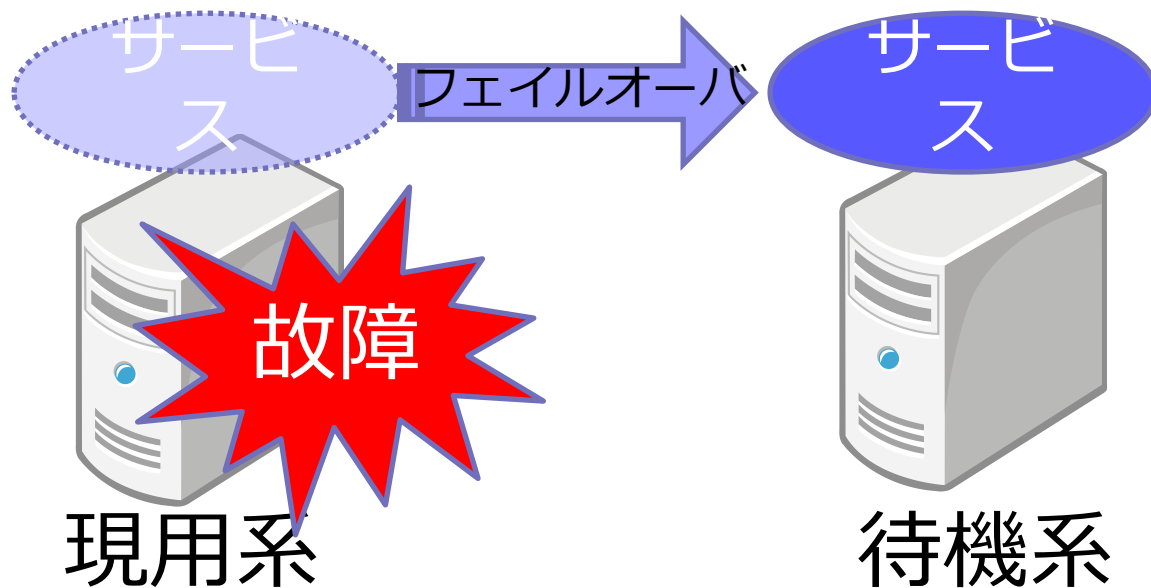
一台のコンピュータでは得られない高い信頼性を得るために、複数のコンピュータを結合(クラスタ化)し、ひとまとまりとする...

ためのソフトウェアです

Pacemakerってなに？

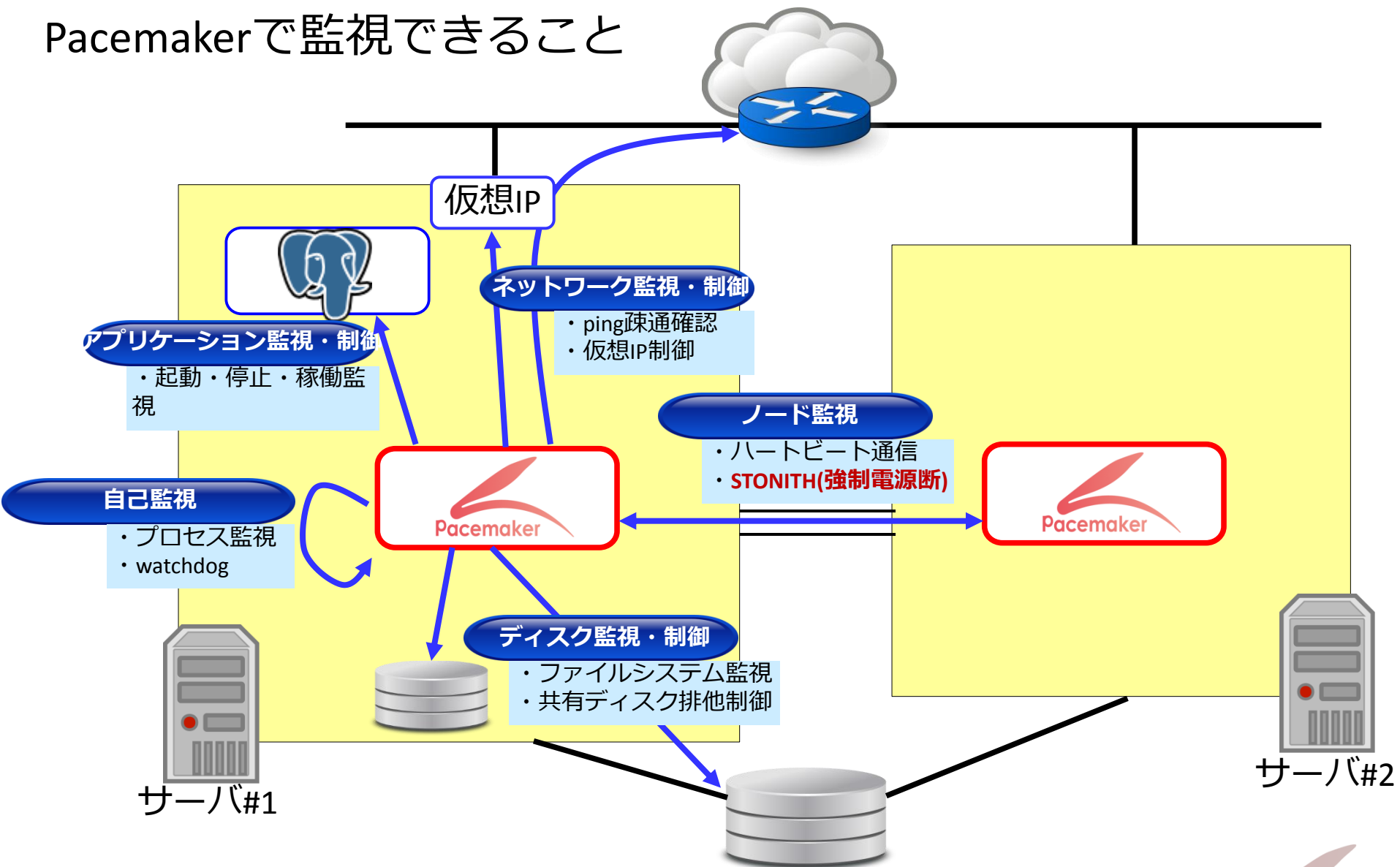
HAクラスタを導入すると、
故障で現用系でサービスが運用できなくなったときに、
自動で待機系でサービスを起動させます

→このことを「**フェイルオーバー**」と言います



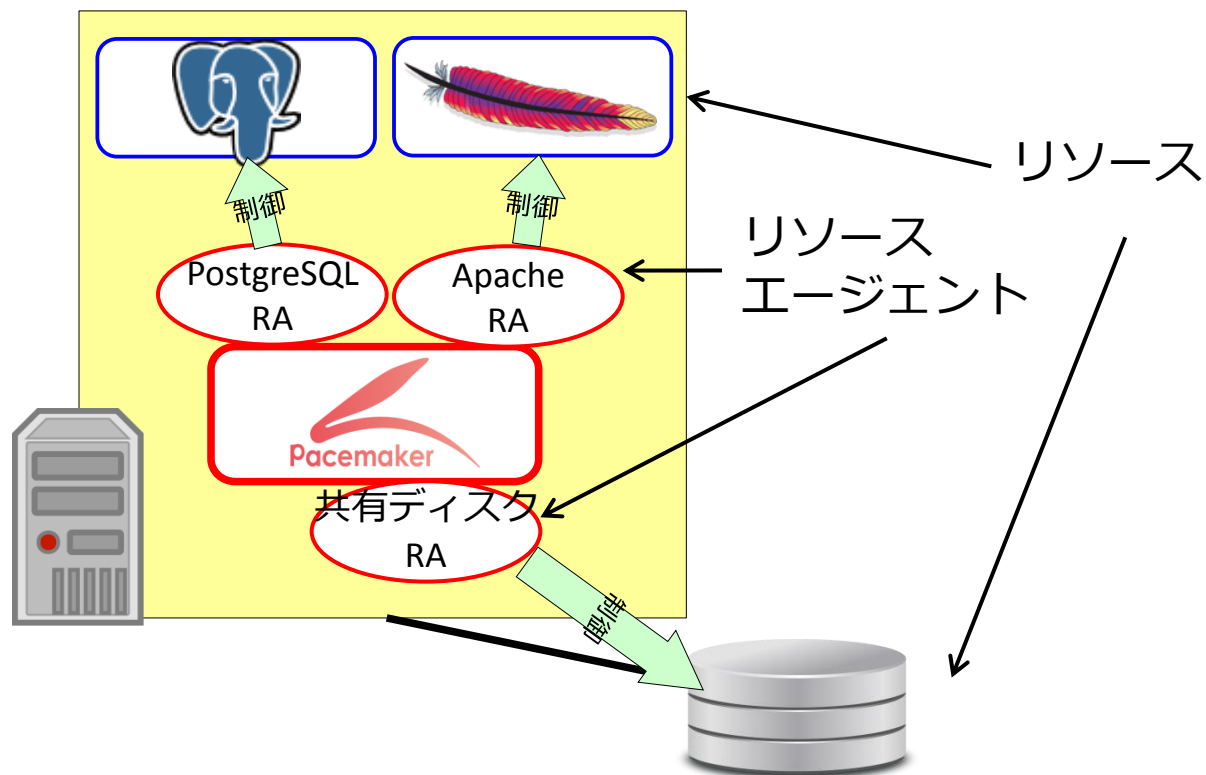
Pacemakerってなに？

Pacemakerで監視できること



Pacemakerってなに？

- Pacemakerが起動/停止/監視を制御する対象を**リソース**と呼ぶ
 - 例：Apache、PostgreSQL、共有ディスク、仮想IPアドレス...
- リソースの制御は**リソースエージェント(RA)**を介して行う
 - RAが各リソースの操作方法の違いをラップし、Pacemakerで制御できるようにしている
 - 多くはシェルスクリプト



最新版Pacemaker-1.1.15-1.1の ご紹介

Pacemaker-1.1.15-1.1の変更点

- 最新版Pacemaker-1.1.15-1.1が2017/1/6にリリースされました
- 主な変更点
 - SNMP対応
 - ログメッセージの簡易化
 - その他バグフィックス

□ SNMP Trap通知機能の(再)実装

- Pacemaker-1.1.14-1.1以前でも使うことはできましたが**非推奨**
 - 「既に送信されたSNMP Trapが再送される場合がある」 不具合
 - Pacemaker-1.1.15-1.1以降で方式・設定方法などが変更になる

□ 運用監視ツールとの親和性が向上

- これまではログ監視によってアラートを通知
 - ログ監視の正規表現の作成
 - Pacemakerのログが変更された場合の追従
- 最近よく変更されます
- } 過不足なく行うのは
困難な作業



SNMP対応

- ❑ Corosync層とPacemaker層でそれぞれ異なる役割・方式

❑ Corosync層

- ❑ ノード状態(クラスタ参加・離脱、クォーラム変化、IC-LAN状態など)を通知
- ❑ **corosync-notifydプロセス**によりトラップされる
 - ❑ corosync-notifydの起動が必要

❑ Pacemaker層

- ❑ リソース状態、STONITH実行などを通知
 - ❑ ノード状態も一部通知できるが、クォーラムやIC-LAN状態などがPacemakerからは分からないため、ノード状態通知はcorosync-notifydを利用
- ❑ **トラップ用スクリプト**を実行してトラップ
 - ❑ トラップ用スクリプトのサンプルは同梱
 - ❑ /usr/share/pacemaker/alerts/alert_snmp.sh.sample
 - ❑ **crm設定で指定**(次ページでご紹介)

SNMPのcrm設定

- Linux-HA Japanのリポジトリパッケージ同梱のpm_crmgenで設定可能
- <https://ja.osdn.net/projects/linux-ha/releases/66936>

#表 13-1 クラスタ設定 ... ALERT設定

ALERT			
#	P path		
	スクリプトを指定		概要
	/usr/share/pacemaker/alerts/alert_snmp.sh		
#	A type	name	value
	パラメータ種別 (attributes/meta)	項目	設定内容
	attributes	trap_add_hires_timestamp_oid	false
		trap_resource_tasks	start,stop,monitor,promote,demote
#	R recipient		
	受信者を指定		概要
	192.168.28.189		

- 複数の受信者にTrapを送信する場合、上記をコピーして設定する



Linux-HA Japan ブースで絶賛実演中！

□ ログメッセージの簡易化

- Pacemaker-1.1.15では、以下の通りログメッセージの簡易化が図られました
 - 自然な英文ライクな形式で出力されるようになりました
 - 内部ステータスの表示が抑止されました

□ Pacemaker-1.1.14

□ crmd[XXXX]: notice: **Operation** prmApPostgreSQLDB_start_0: **unknown error** (node=srv2, call=100, rc=1, cib-update=50, confirmed=true)

□ Pacemaker-1.1.15

□ crmd[XXXX]: notice: Result of **start** operation for prmApPostgreSQLDB on srv2: **1 (unknown error)**

- 併せて、pm_logconvでも、エラーコードの意味を出力するように変更しました

□ Pacemaker-1.1.14

□ info: Resource prmExPostgreSQLDB started. (rc=0)
□ error: Resource prmExPostgreSQLDB does not work. (rc=7)

□ Pacemaker-1.1.15

□ info: Resource prmExPostgreSQLDB started. (rc=0) **ok**
□ error: Resource prmExPostgreSQLDB does not work. (rc=7) **not running**

監視メッセージには影響なし(※)

□ (※) STONITH関連メッセージのみ変更あり

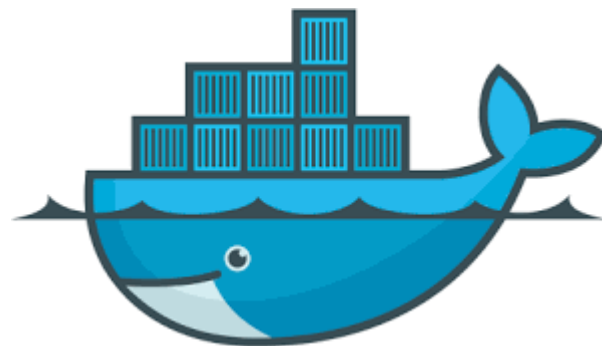
その他バグフィックス

- 詳細は以下参照

- <http://linux-ha.osdn.jp/wp/archives/4591>

- <https://github.com/ClusterLabs/pacemaker/blob/Pacemaker-1.1.15/ChangeLog>

PacemakerでDockerクラスタリング！



話さないこと / 話すこと

□ 話さないこと

- コンテナとは？

- Dockerとは？

- Dockerを使うと何がうれしいの？

- Dockerの使い方

□ 話すこと

- Pacemaker + Dockerの方法論

- Pacemakerでクラスタリングするメリット

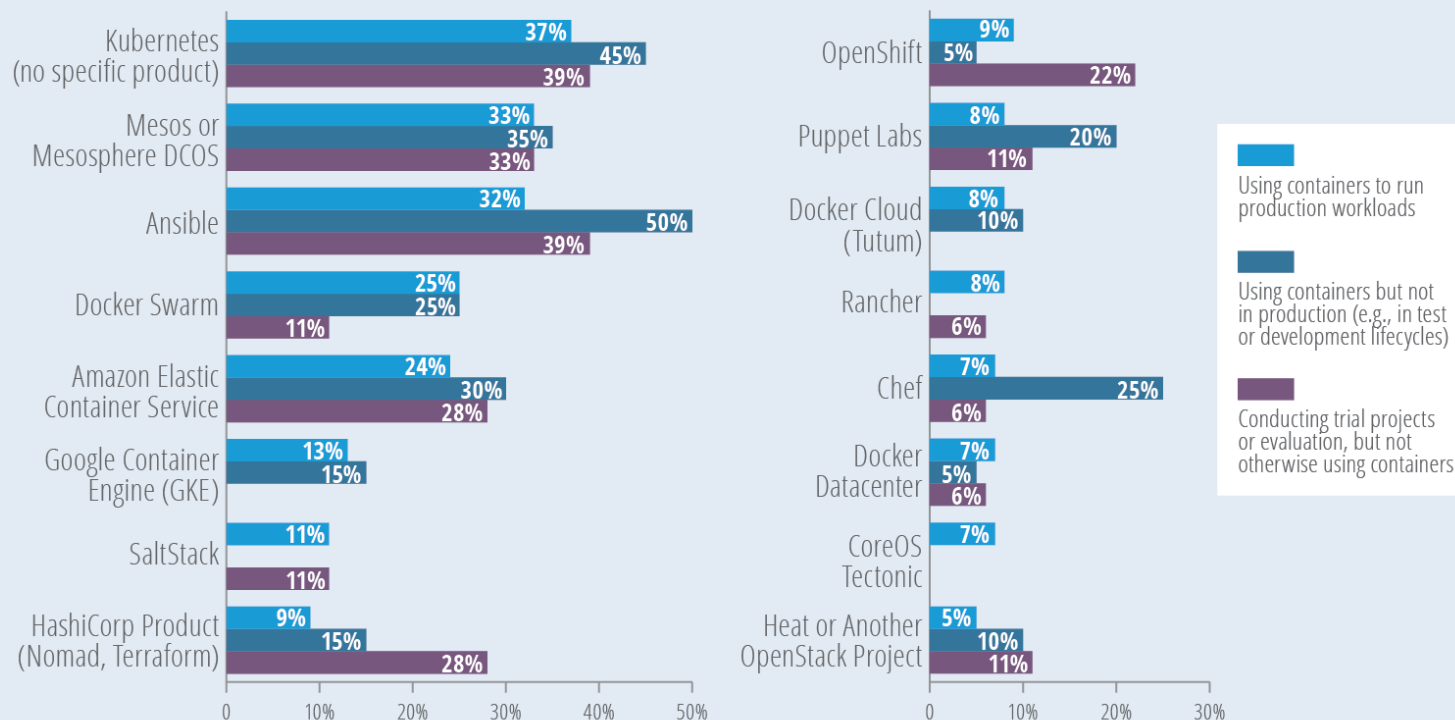
DockerコンテナのHA

- ❑ サービスを商用環境で運用する際にはHAが非常に重要
 - ❑ Dockerコンテナでも同様
- ❑ DockerコンテナのHAはオーケストレーションツールの利用が主流
 - ❑ Kubernetes
 - ❑ Docker Swarmモード / Docker Swarm
 - ❑ Apache Mesos
 -
 -
 -

オーケストレーションツールの利用率

2016/6のデータ

Top Orchestration Products Based on Expected Usage Within Next Year: Differences Based on Implementing Status



Source: The New Stack Survey, March 2016. Within the next year, what are the top three products or services you expect to utilize to manage or orchestrate containers? Select all that apply. Using in production, n=76; Using, but not in production, n=20; Conducting trials/evaluation, n=18. Choices with less than five responses are not shown.

THE NEW STACK

<https://thenewstack.io/ansible-leading-chef-puppet/>

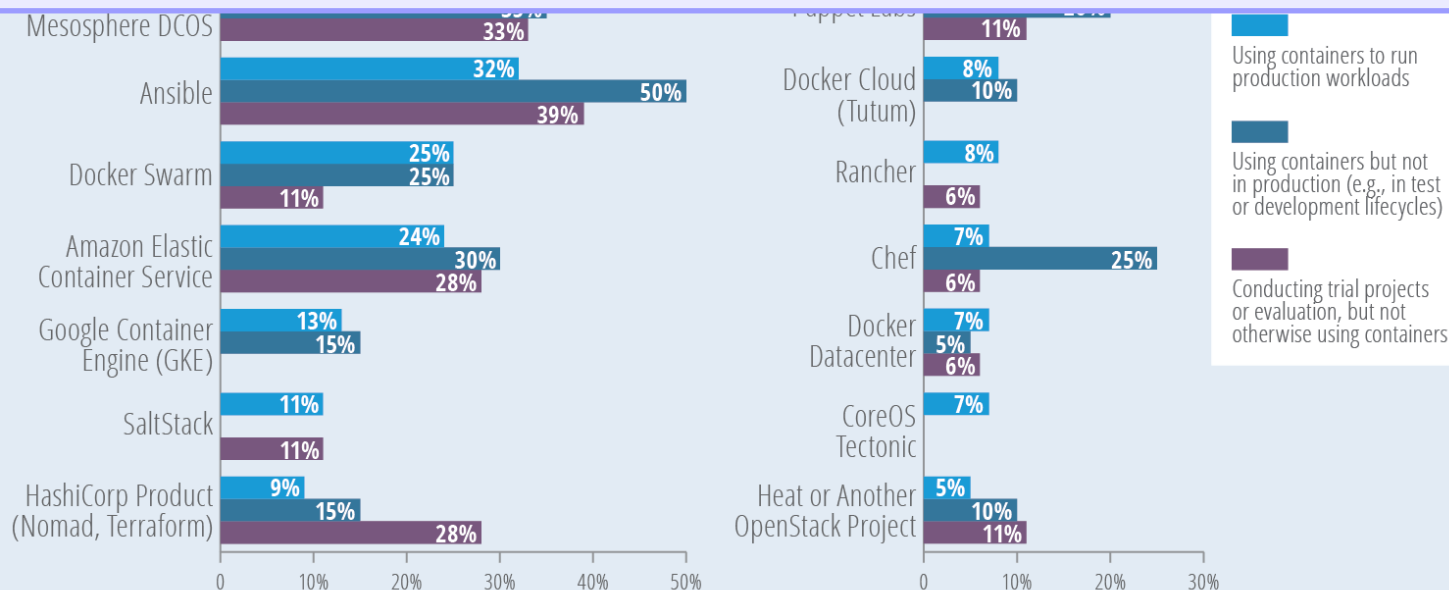
Linux-HA Japan Project

オーケストレーションツールの利用率

□ 2016/6のデータ

Top Orchestration Products Based on Expected Usage Within Next Year: Differences Based on Implementing Status

「ペ」の字もない！



Source: The New Stack Survey, March 2016. Within the next year, what are the top three products or services you expect to utilize to manage or orchestrate containers? Select all that apply. Using in production, n=76; Using, but not in production, n=20; Conducting trials/evaluation, n=18. Choices with less than five responses are not shown.

THE NEW STACK

<https://thenewstack.io/ansible-leading-chef-puppet/>

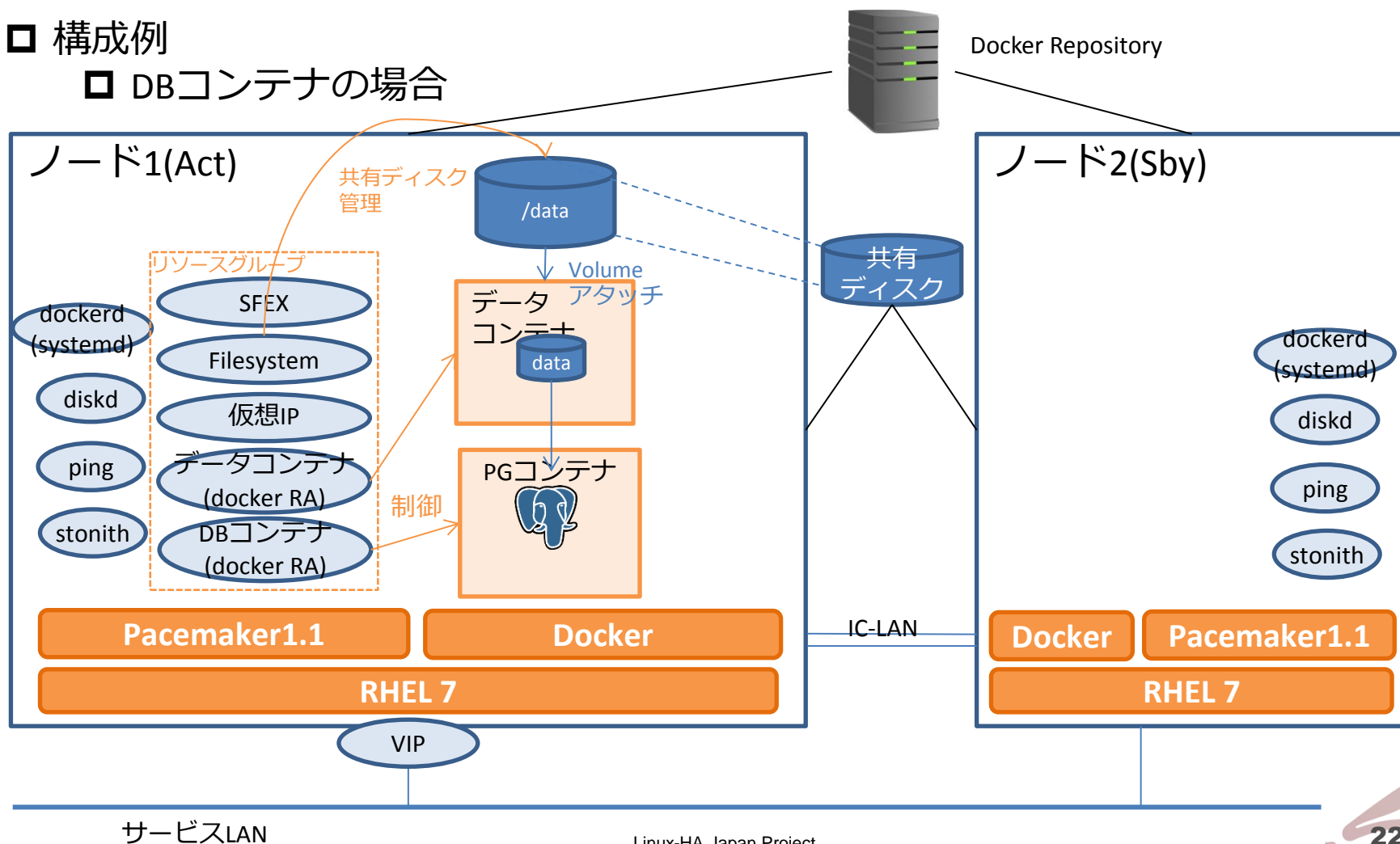
Linux-HA Japan Project

PacemakerでDockerコンテナをクラスタ化

- PacemakerでもDockerコンテナをクラスタリングすることが可能！
 - Docker RAの利用

□ 構成例

□ DBコンテナの場合



Docker RA設定例

```
primitive prmDBContainer ocf:heartbeat:docker ¥
  params ¥
    allow_pull="true" ¥
    image="postgres:latest" ¥
    name="test_db" ¥
    run_opts="--volumes-from data -p 5432:5432 ¥
    reuse="false" ¥
    run_cmd=" /bin/sh -c /entrypoint.sh" ¥
  op start interval="0s" timeout="60s" on-fail="restart" ¥
  op monitor interval="10s" timeout="60s" on-fail="restart" ¥
  op stop interval="0s" timeout="60s" on-fail="fence"
```

- ❑ 現在のDocker RAはimage名に制約あり
 - ❑ ポート番号付のimage名を設定不可
 - ❑ 修正パッチ pull request 予定
- ❑ 暫定対処
 - ❑ 両系で予めimageをpullし、allow_pullを"false"に設定する
 - ❑ 80番ポートでlistenする
 - ❑ 80番ポートは省略可能

Docker RAパラメーター一覧

パラメータ	設定値	デフォルト	必須
image	Docker image名	なし	○
name	コンテナ名	リソース名	×
allow_pull	Docker imageがローカルに存在しない場合pullするか	false	×
run_opts	docker run実行時のオプション (-d --nameを除く)	-d --name	×
run_cmd	docker run実行時にコンテナで実行するコマンド	なし	×
monitor_cmd	コンテナ内アプリケーションの監視コマンド	なし	×
force_kill	コンテナ停止時にdocker killを利用するか	false	×
reuse	リソースの再起動時にコンテナを再利用するか	false	×

□ 注意点

- image名は制約有(前述)
- run_optsには「-d --name」をのぞいたオプションを設定
- monitor_cmdは非推奨
 - Docker ImageのHEALTHCHECK CMDで実行すべき
- アプリケーションコンテナではreuseはfalseを設定すべき
 - PID 1問題の回避

余談：アプリケーションコンテナとPID 1問題

- アプリケーションコンテナ内で複数プロセスを動作させる場合、ゾンビプロセスが残存する可能性がある

- この状態で運用を続けると・・・

増加したゾンビプロセスによりプロセステーブルがひっ迫し、**新規プロセスの生成ができなくなる**恐れがある

余談：アプリケーションコンテナとPID 1問題

□ アプリケーションコンテナ

- PID 1で一つのアプリケーションが動作するコンテナ

□ PID 1

- Unix/Linuxで一番最初に起動するプロセスで全てのプロセスの親プロセス
- 通常はinit相当のプロセス
 - init(相当のプロセス)により、全てのゾンビプロセスが適切に処理される

```
# ps ax | sort | head -n 1  
1 ?    Ss    0:00 /usr/lib/systemd/systemd --switched-root --system --deserialize 24
```

- アプリケーションコンテナでは・・・？
 - init(相当のプロセス)ではなくアプリケーション

```
# ps ax | sort | head -n 1  
1 ?    Ss+   0:00 postgres
```

Docker RAを使うといいことあるの？

- その前に・・・

- Docker Swarmモードの検証結果を少しご紹介

Docker Swarmモード検証構成

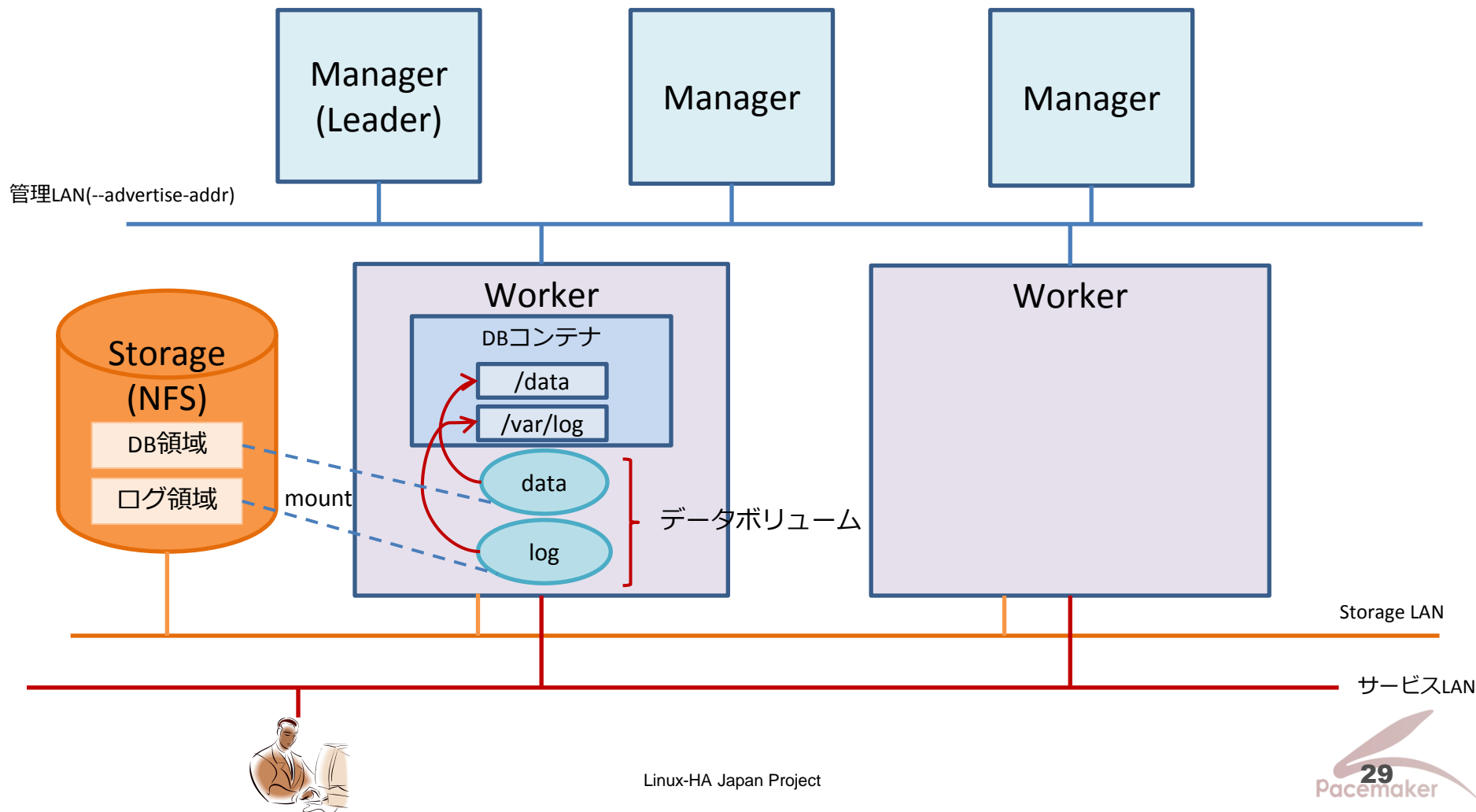
- Docker SwarmモードでDBコンテナ(PostgreSQL)をオーケストレーションしてみた
 - 10月ごろなので、最新のデータではありません
- 以下の環境をKVMゲストマシンで5ノード分用意
 - Manager : 3ノード
 - Worker : 2ノード

種別	バージョン
OS	Red Hat Enterprise Linux 7.2
CPU	1コア
メモリ	2GB
Docker	1.12.2, build bb80604
コンテナイメージ	PostgreSQL 9.5

- コンテナイメージのチューニング
 - HEALTHCHECK CMD : pg_isready

Docker Swarmモード検証構成

- ❑ 管理LANとStorageLAN(NFS)は分離
- ❑ Manager上ではコンテナは起動させない(Drain)
- ❑ DB領域とログ領域はNFSで永続化する



検証結果概略

□ ほとんどのケースで、可用性を確保できる

- Managr故障
- Worker故障

□ 一部、可用性を確保できないケース有り

1. Worker – Storage間NW故障

- HEALTHCHECK CMDが失敗し、故障検知する
- しかし、STOP SIGNALおよびSIGKILLでも停止できずF/O不可

2. Manager – Worker(コンテナ稼働系)間NW故障

- 別のWorkerへF/O
- しかし、稼働中のコンテナは停止されないため、NW復旧時に一瞬スプリットブレイン状態(Storageの二重マウント)が発生する可能性あり
- Docker 1.13でも事象確認

3. サービスLAN故障

- Managerの管理対象外なので故障検知しない
- サービスは稼働しているが、クライアントはアクセス不可

□ その他の懸念点

- F/O後、古いコンテナが削除されないため手動削除が必要

検証結果概略

- ほとんどのケースで、可用性を確保できる

- Managr故障
- Worker故障

- 一部、可用性を確保できないケース有り

- 1. Worker - Storage間NW故障

ステートフルコンテナではデータ破壊が発生する可能性あり！

コンテナのストレージ状態(Storageの二重マウント)が元々ある可能性あり

- Docker 1.13でも事象確認

- 3. サービスLAN故障

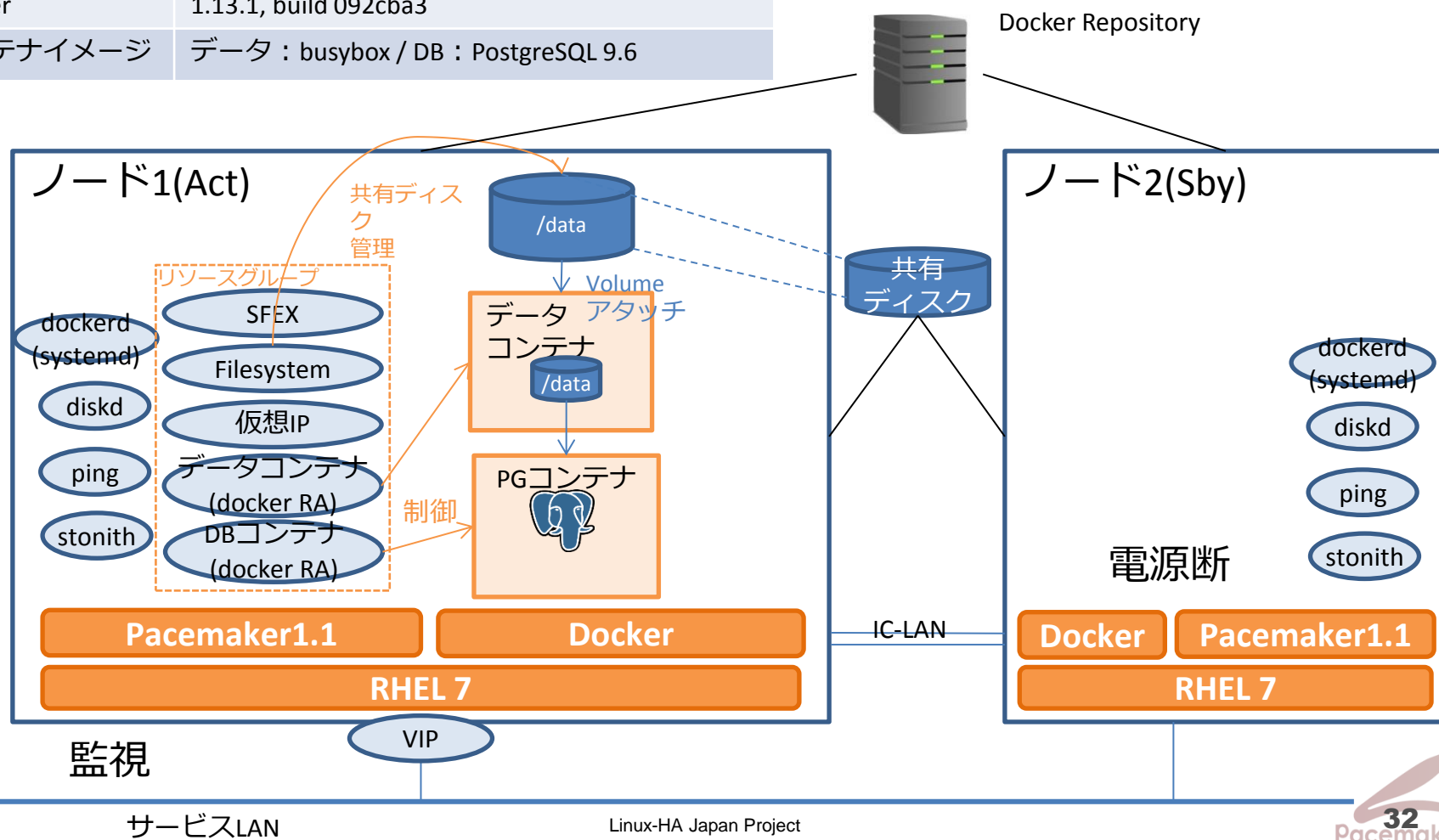
- Managerの管理対象外なので故障検知しない
 - サービスは稼働しているが、クライアントはアクセス不可

- その他の懸念点

- F/O後、古いコンテナが削除されないため手動削除が必要

Docker RA検証構成

種別	バージョン
OS	Red Hat Enterprise Linux 7.2
Pacemaker	1.1.15-1.1
Docker RA	PM-1.1.15-1.1同梱 (image の不具合修正済み)
Docker	1.13.1, build 092cba3
コンテナイメージ	データ : busybox / DB : PostgreSQL 9.6



Docker RA検証結果概略

故障種別	操作	サービス継続	備考
ノード故障	稼働系ノード電源断	○	STONITH
スプリットブレイン	IC-LAN切断	○	STONITH /データ破壊なし
dockerd故障	dockerdプロセス強制停止	○	
データコンテナ故障	コンテナ強制停止	○	
DBコンテナ故障	コンテナ強制停止	○	
ストレージ故障	ストレージ接続断	○	STONITHの場合あり
サービスLAN故障	サービスLAN切断	○	

Docker RA検証結果概略

故障種別	操作	サービス継続	備考
ノード故障	稼働系ノード電源断	○	STONITH

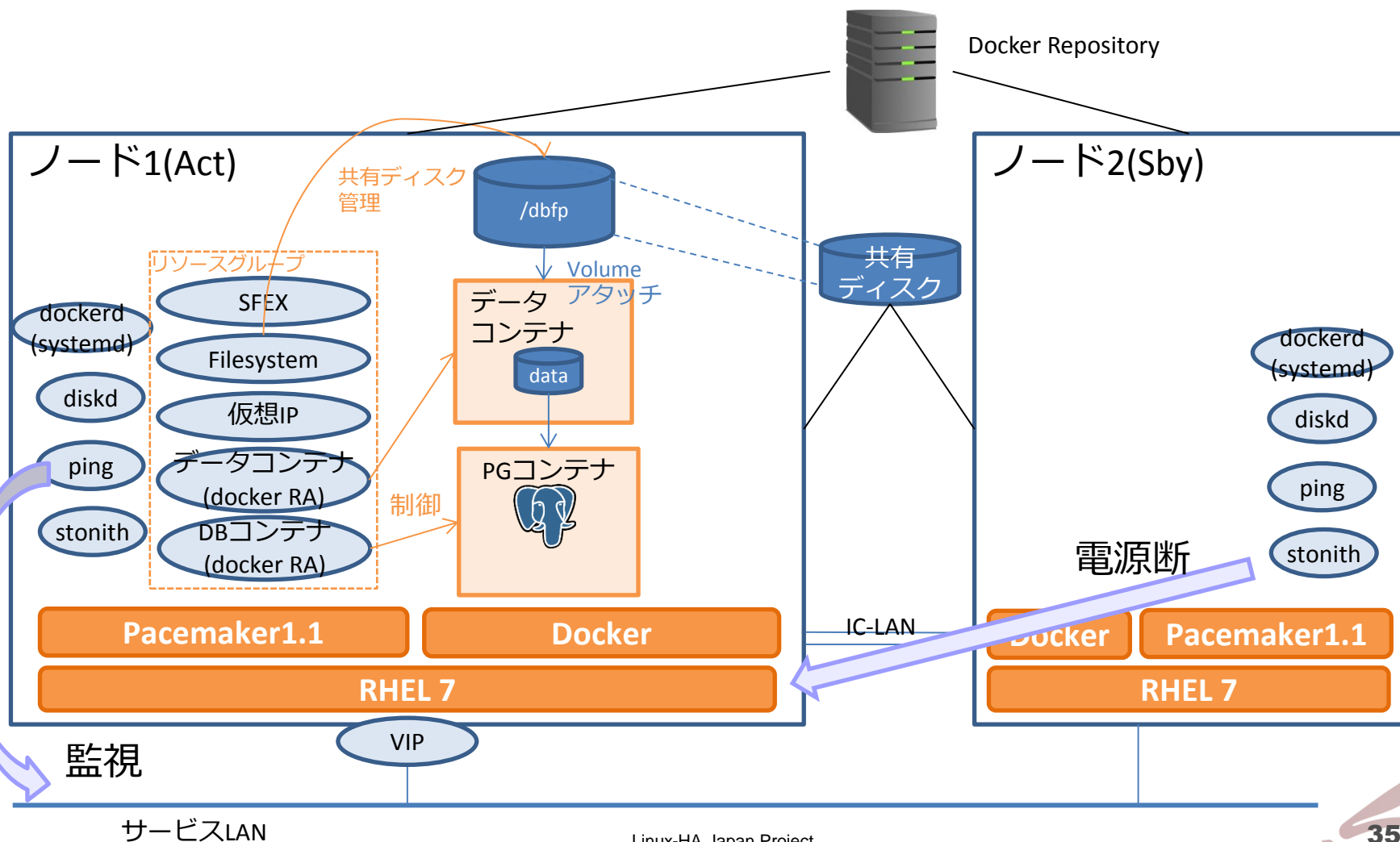
全てのパターンでサービス継続！
データ破壊なし！

コンテナ故障			
DBコンテナ故障	docker kill <コンテナ>	○	
ストレージ故障	ストレージ接続断	○	STONITHの場合あり
サービスLAN故障	サービスLAN切断	○	


Docker RAを使うといいことあるの？

□ Docker Swarm モードより高い可用性・信頼性

- STONITHによる**対向ノード強制電源断** → **スプリットブレインの防止**
- pingリソースによる**サービスLAN監視**



Docker RAを使うといいことあるの？

- ❑ コンテナ故障時、古いコンテナを削除してくれる
 - ❑ reuse="false" の場合
 - ❑ PID 1問題の回避
 - ❑ 利用しているコマンドがDocker Engineの基本的なコマンドのみ
 - ❑ docker pull
 - ❑ docker run
 - ❑ docker start
 - ❑ docker stop
 - ❑ docker kill
 - ❑ docker image
 - ❑ docker inspect
 - ❑ docker rm
- 
- Docker Swarmモードに比べて
バージョンによる影響を受けにくい

Docker RAによるクラスタリングの弱点

- ❑ 圧倒的に少ないユーザ・・・
 - ❑ バグがあるかも
- ❑ スケーラビリティに弱い
 - ❑ Docker SwarmモードやKubernetesのようにコマンド1つで簡単スケールとはいかない
 - ❑ Pacemaker 1.1系の(実用的な)最大ノード数は16ノード (Linux-HA Japan調べ)
 - ❑ KubernetesやDocker Swarmモードは数十万以上
 - ❑ とはいえ、Pacemaker Remote機能を使えば1000ぐらいはいけるか？
- ❑ 設計の難しさ
 - ❑ リソースの依存関係、リソースの配置先などユーザが設計する
 - ❑ オーケストレーションツールのように、ツールにお任せとはいかない
 - ❑ 管理するコンテナが増えれば増えるほど、crmファイル(Pacemakerのリソース設定ファイル)が複雑に・・・
 - ❑ 1コンテナにつき1設定
 - ❑ Apache、Tomcat、PostgreSQLを管理する場合、3つのDocker RAの設定が必要

オーケストレーションツールとPacemakerを 組み合わせることも可能

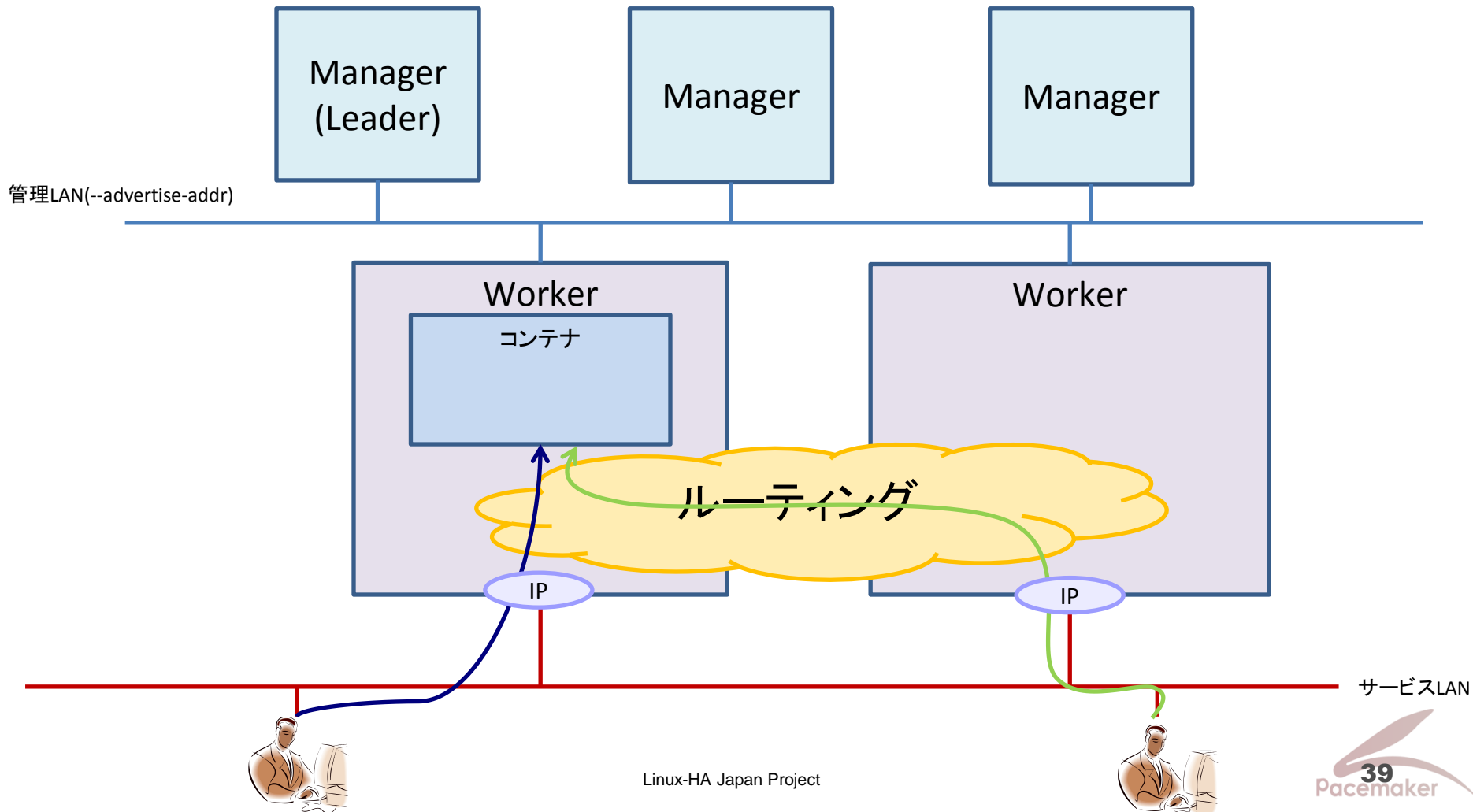
□ それぞれの弱点を補完して、より強固なHA構成を！

- Docker Swarmモードの弱点
 - サービス継続不可の故障パターンが存在
 - データ破壊が発生する故障パターンが存在
- Pacemakerの弱点
 - スケーラビリティ
 - 複雑な設計

弱点を補完し、
いいところを目指す

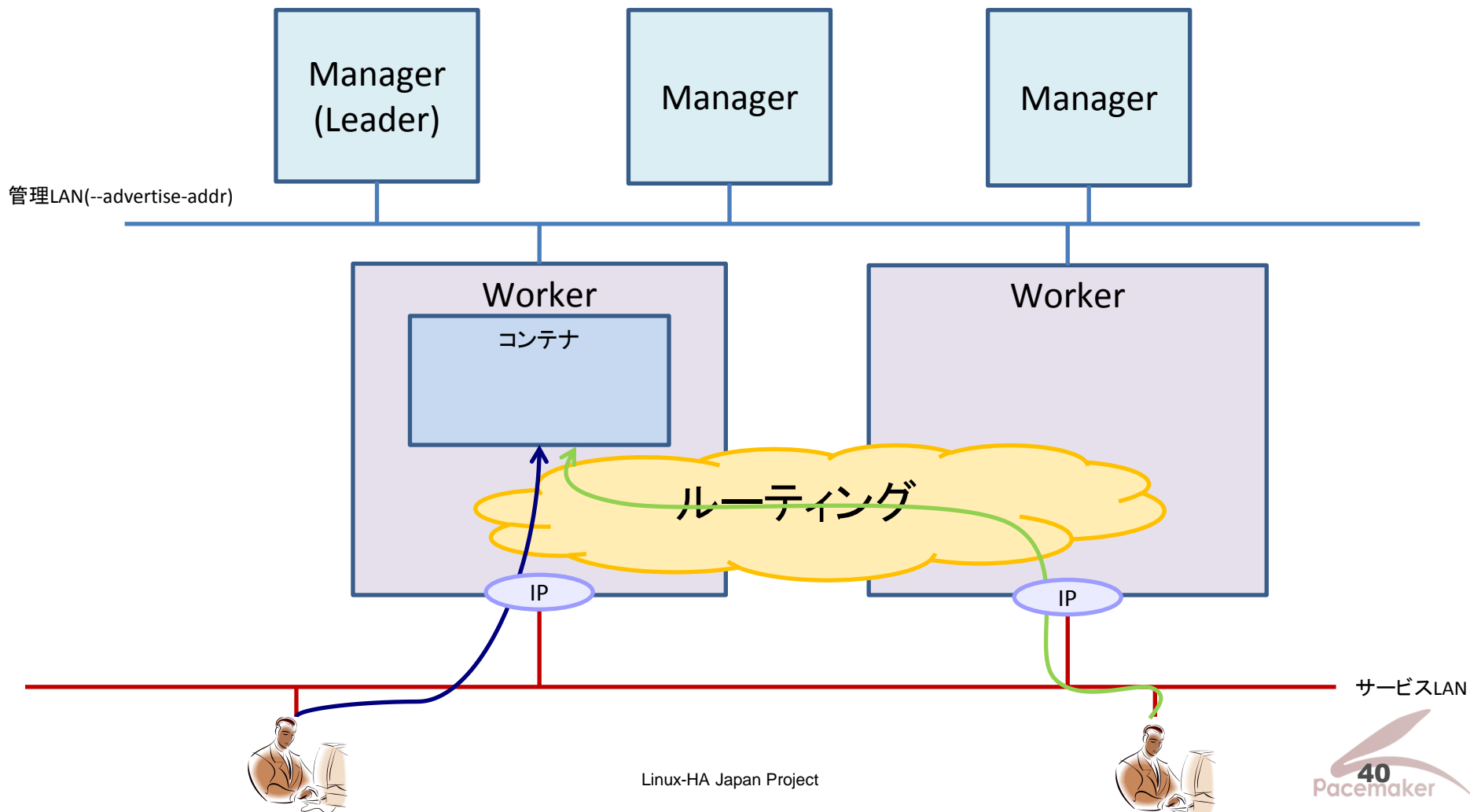
Docker Swarmモード + Pacemaker

- ❑ Docker SwarmモードではクライアントはどのWorkerにアクセスしてもコンテナへ到達できる
 - ❑ iptables + ipvsでルーティング



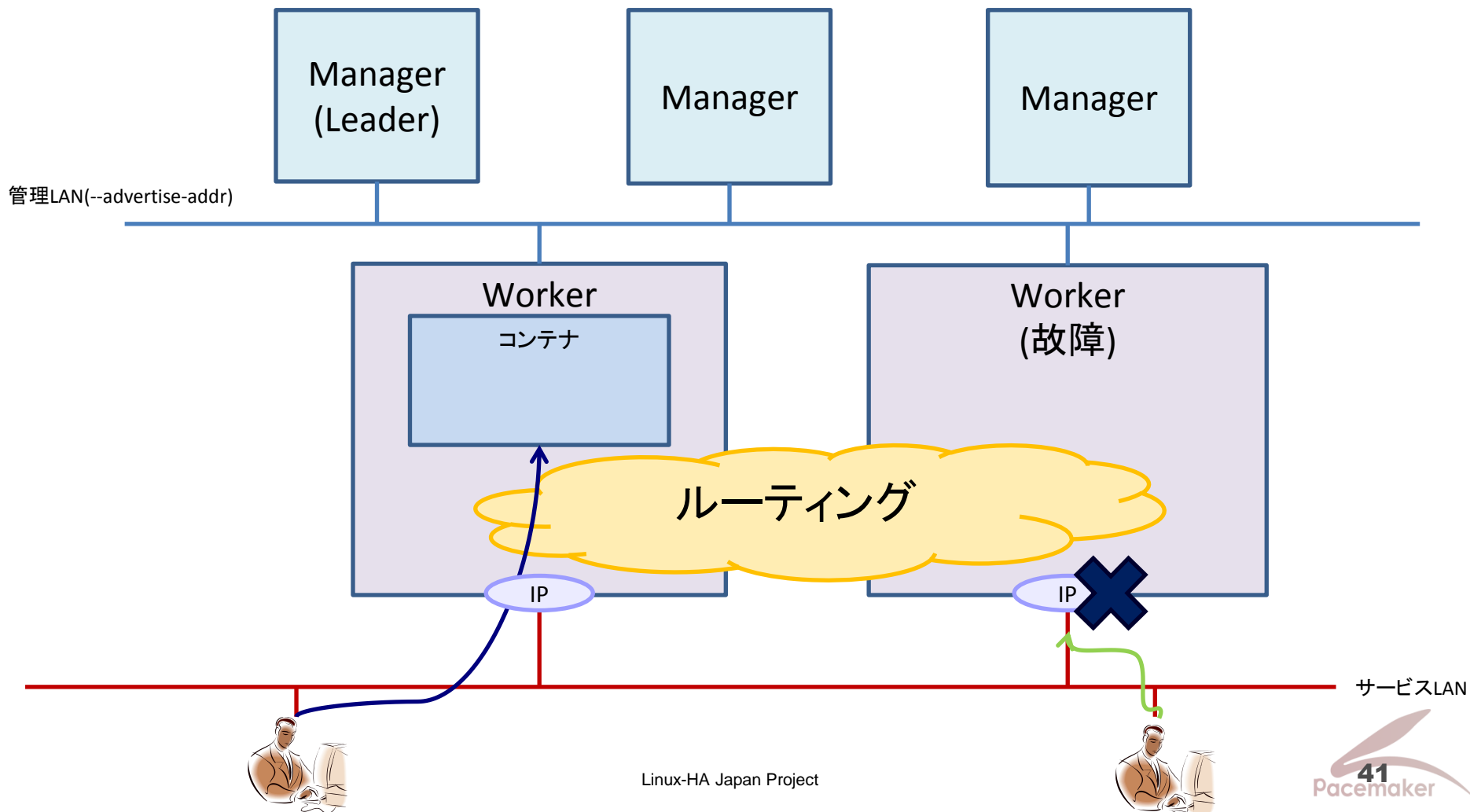
Docker Swarmモード + Pacemaker

- 故障時にアクセスできるクライアントとアクセスできないクライアント
 - dockerd故障
 - NW故障
- 故障時にもすべてのクライアントからアクセスできるようにしたい



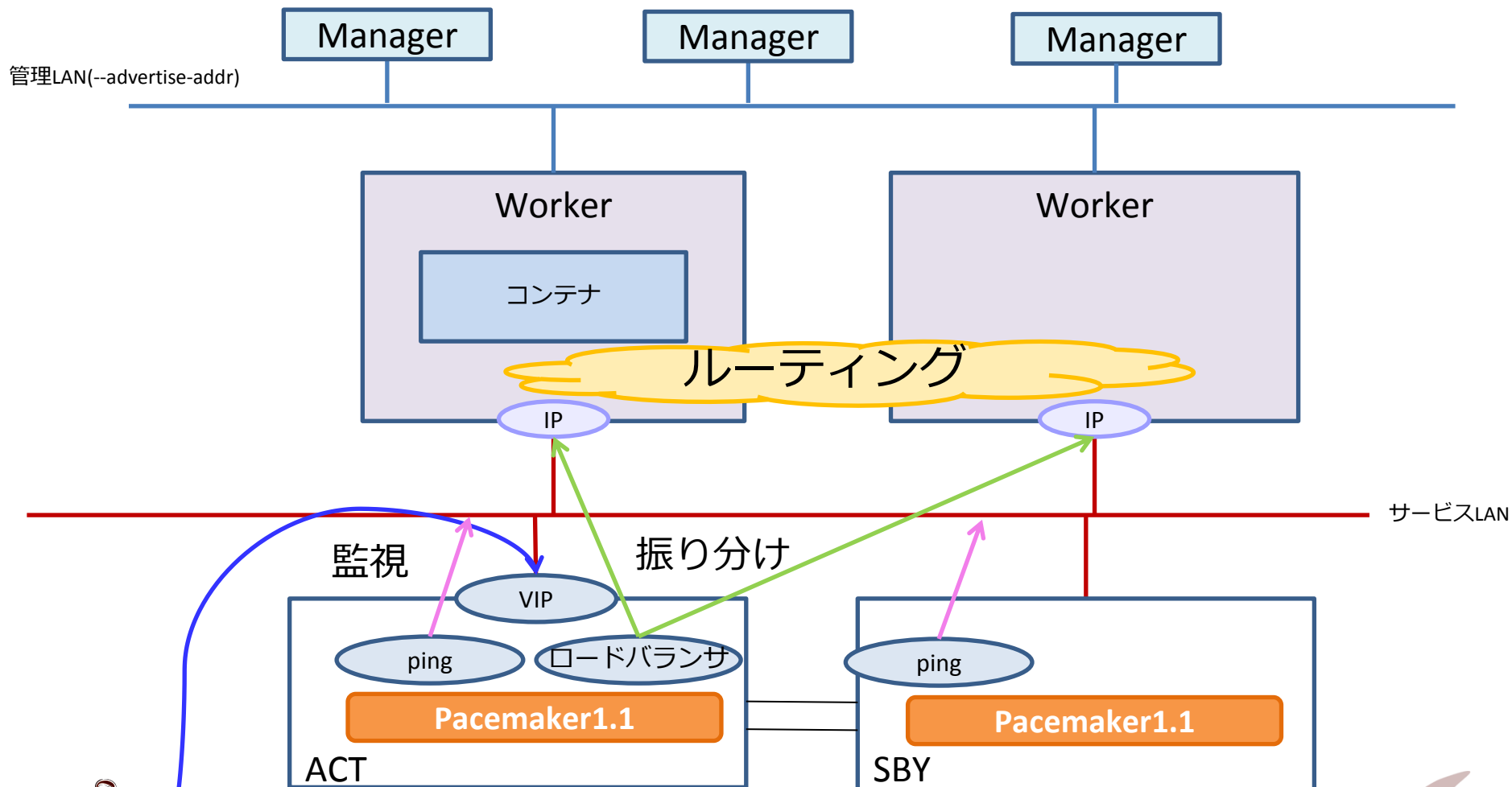
Docker Swarmモード + Pacemaker

- 故障時にアクセスできるクライアントとアクセスできないクライアント
 - dockerd故障
 - NW故障
- 故障時にもすべてのクライアントからアクセスできるようにしたい



Docker Swarmモード + Pacemaker

- ❑ Pacemaker + ロードバランサ(UltraMonkeyなど)で解決
 - ❑ クライアントはVIPだけ知っていればよい
 - ❑ ロードバランサが疎通可能なWorkerに適宜振り分け
 - ❑ Docker Swarmモード単体ではできないサービスLAN監視も可能
- 全てのクライアントが
コンテナへアクセス可能



まとめ

- ❑ PacemakerでもDockerコンテナのクラスタリングは可能
 - ❑ DBなどのステートフルコンテナではDocker RAの方が安全
 - ❑ スプリットブレイン対策
- ❑ オーケストレーションツールの方が有利な場面もある
 - ❑ スケール性能
 - ❑ お手軽な運用
 - ❑ 圧倒的なユーザ数
- ❑ 目的・用途に応じて使い分けることが大事
 - ❑ Pacemaker、オーケストレーションツールそれぞれに長短がある
 - ❑ 組み合わせによる利用も可能

おまけ：今後のPacemaker + Dockerに怪しい動き！？

- ❑ 最近、**Andrew Beekhof氏**(Pacemakerの一番偉い人)の個人リポジトリでコンテナ制御関連の実装が・・・！？
 - ❑ <https://github.com/beekhof>
- ❑ RAではなく本体機能(engine)
- ❑ **bucket**というコンテナ専用(?)のリソース種別が・・・？

beekhof / pacemaker
forked from ClusterLabs/pacemaker

Watch 4 Star 15 Fork 179

Code Pull requests 1 Projects 0 Pulse Graphs

History for pacemaker / engine

Commits on Mar 6, 2017

- PE: Containers: Default to a short-form print output
beekhof committed a day ago

Commits on Mar 3, 2017

- PE: Update tests for simplified clone allocation function
beekhof committed 4 days ago
- PE: Clone: Simplified allocation function
beekhof committed 4 days ago
- PE: Containers: Better checks when assuming a container will start
beekhof committed 4 days ago
- PE: Preferred nodes are only accepted if their scores are equal to th...
beekhof committed 4 days ago
- PE: Containers: Prevent invalid recovery graphs
beekhof committed 4 days ago

Commits on Mar 1, 2017

- PE: Container: Add support for non-default docker networks and supply...
beekhof committed 6 days ago

Commits on Feb 28, 2017

- PE: Containers: Allow specifying the maximum number of peers per host
beekhof committed 7 days ago

Commits on Feb 26, 2017

おまけ：今後のPacemaker + Dockerに怪しい動き！？

- ❑ 最近、**Andrew Beekhof氏**(Pacemakerの一番偉い人)の個人リポジトリでコンテナ制御関連の実装が・・・！？
 - ❑ <https://github.com/beekhof>
- ❑ RAではなく本体機能(engine)
- ❑ **bucket**というコンテナ専用(?)のリソース種別が・・・？



さいごに

Linux-HA Japan URL

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

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



The screenshot shows the Linux-HA Japan Project website. At the top is the logo with the text "LINUX-HA JAPAN High-Availability Clustering on Linux". Below the logo is a navigation bar with links: HOME, メーリングリスト, ダウンロード&インストール, マニュアル, デスクトップテーマ・壁纸等, コミュニティ概要. Below the navigation bar is a section titled "Linux-HA Japan プロジェクト" with a "Check" button. The main content area lists various resources: "Linux-HA Japan 成果物ダウンロード" (RHEL/CentOS向けPacemaker RPMパッケージ, yumのリポジトリ形式や設定ファイル(crm)作成支援ツール, ディスク監視機能などをダウンロードできます。), "マニュアル" (本家コミュニティ提供の公式マニュアルやLinux-HA Japan提供の翻訳マニュアル。), "メーリングリスト" (インストール方法や設定方法等の質問はMLまで。), "イベント情報" (カンファレンスへの出席や講演、勉強会開催情報、講演時のスライド公開など。), and "開発者向けサイト" (Linux-HA Japan開発者向けサイトです。). At the bottom, there is a Twitter link and a footer note about the site's maintenance.

Pacemaker関連の最新情報を 日本語で発信

Pacemakerのダウンロードも こちらからどうぞ (インストールが楽なリポジトリパッ ケージを公開しています)

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.osdn.jp/>
の「メーリングリスト」をクリック



- ・ MLアドレス

linux-ha-japan@lists.osdn.me

※スパム防止のために、登録者以外の投稿は許可制です

ご清聴ありがとうございました。
May the Pacemaker be with you !



Linux-HA Japan

検索

