

Internet Week 2015

# OpenStack実践講座

吉瀬 淳一 (@jyoshise)

# 自己紹介

## お仕事

- 1999年 日本ヒューレット・パッカード(株)入社
- 2014年 クラウド関連事業の強化に伴い、Helion OpenStack Professional Servicesに異動
  - Asia Pacific & Japan Region最大の拠点として、OpenStackの変態^h^hエキスパートが東京に集結

## 得技

- ギター、スキー、料理
- プログラミング(でも仕事にはほとんど生かせていない)

## 趣味

- キャンプ、映画鑑賞、音楽、ボルダリング、電子工作、カメラ、チェス、読書、などなど
- 要するに多趣味ゆえの無趣味



**Hewlett Packard  
Enterprise**



**HPE Helion**

---

# 本日の内容

## OpenStackで幸せになるために

### –企画/要件定義編

- そもそもなんでOpenStackが使いたいの？
- アンチパターンから見る勘所
- OpenStackだけでは足りないものとは？
- OSS / ディストリビューション / アプライアンス

### –設計/構築編

- ノード配置
- HA
- ネットワーク
- ストレージ
- バックアップ
- DR
- ロギング/監視



# 企画/要件定義編

---

## そもそもなんでOpenStackなの？

- 増え続けるシステム/アプリケーションを効率的に管理したい
- DevOpsやPaaSを活用した迅速なアプリケーション開発と実行管理
- XaaSを実現するための基盤
- ベンダーロックイン回避
- 共通基盤化によるコスト削減
- 開発部門はAWS使いたがってるんだけど
- なんか流行ってるから

# アンチパターンから見る勘所

TechTarget Japan 連載：エンタープライズのためのOpenStack検討ガイドより

【第5回】OpenStack導入で先が思いやられる「アンチパターン」ベスト5

<http://techtarget.itmedia.co.jp/tt/news/1506/19/news09.html>

- なんとなく仮想化基盤の更改先とする
- インフラ担当者だけで進める
- ペット感覚が抜けない
- いきなり壮大な計画を立てる
- 枝葉の技術にこだわる

新着ホワイトペーパー：IT導入の課題を検討するヒント

総合トップ ニュース ビジネス 経営 企業とIT システム導入 Webマーケ スマホ eBook パソコン デジカメ AV家電 ヘルスクエア 物ネタ 動画 ブログ 電力

TechTarget Japan

電源設計のスキル向上を目指す方へ 特集 電源設計 EDN Japan

ITインフラ セキュリティ 業務アプリ システム開発 IT経営 医療・教育 製品比較 導入事例 ホワイトペーパー 会員サービス

Pick up > 標準型攻撃 NEW モバイル NEW ファイルサーバ SSD 認証システム Computer Weekly プレミアムPDF

TechTarget ジャパン > クラウド > OpenStack導入で先が思いやられる「アンチパターン」ベスト5 (1/2)

2015年06月25日 08時00分 UPDATE

## エンタープライズのためのOpenStack検討ガイド【第5回】

### OpenStack導入で先が思いやられる「アンチパターン」ベスト5 (1/2)

OpenStack導入で最も重要なのは初期コンセプトを固めることだ。OpenStackを仮想化の延長と考えると失敗する。多くのOpenStack案件に関わってきた筆者が5つのアンチパターンを紹介する。

[真壁 徹、伊藤雅典、Hewlett-Packard]

進化するIT基盤構築、運用の最前線では何が起きている？

印刷/PDF ツイート 42 いいね! 96 2 0 Pocket 20 メールで送信

類似記事の掲載をメールで通知 連載「エンタープライズのためのOpenStack検討...」の更新をメールで通知

関連キーワード

OSS SDN (Software Defined Networking) ネットワーク オープンソース クラウドコンピューティング OpenStack プライベートクラウド

本連載「エンタープライズのためのOpenStack検討ガイド」

ベンダーコンテンツ -PR-

- 仮想化、ストレージ課題を一掃する「ハイパーコンバージド」の何がすごい？ (2015/6/4)
- オンプレミスと同じでいいの？ クラウドにはクラウドのセキュリティ対策を (2015/5/13)
- 「vSphere互換クラウド」の「vCloud Air」が本稼働、国内ユーザーはこう評価した (2015/4/20)
- 「Windows Server 2003」移行でクラウド化を促進、今必要とされる攻めのIT投資 (2015/3/31)
- クラウドの知識が無くてもできる

---

過去のしがらみを捨てることも必要。

- アプリのアーキテクチャを変えたくない
- 現行の運用を変えたくない
- 現行の基盤で実現できている機能は全て実現しなければならない

□ 「うちのやり方に合わせろ」

こういった考えに固執するとアーキテクチャに歪みが生じます (OpenStackに限った話ではありませんけど)。結果的にトラブルを量産します。



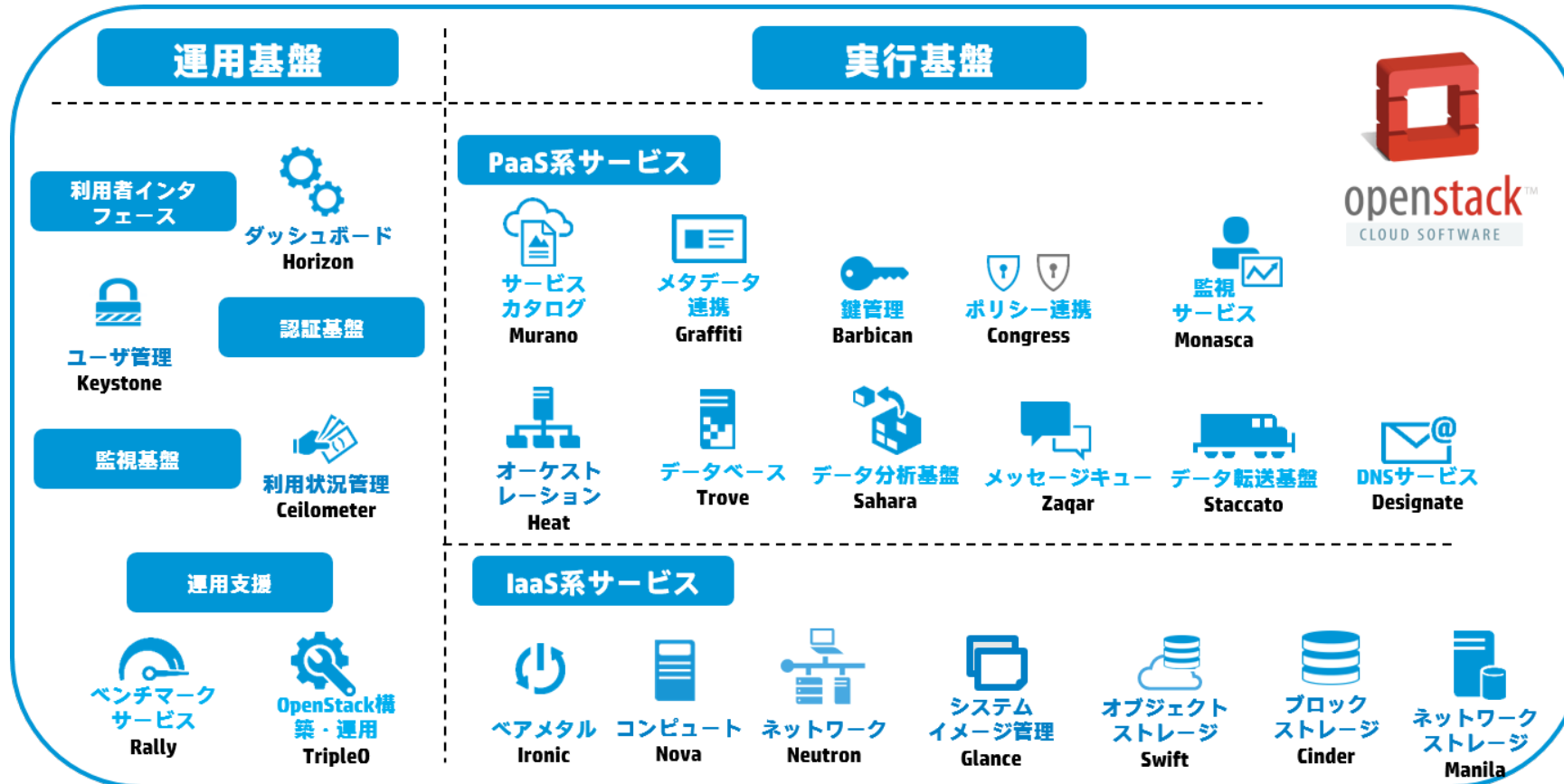
# OpenStack だけでは足りないものとは？

それでも  
足りない  
ものが...

## □ OpenStack を構成するプロジェクトの全体像

### □ 公式に登録されているものだけで、44プロジェクト

□ <http://governance.openstack.org/reference/projects/>



(注)図は一部HP独自のプロジェクトを含みます。



---

# OpenStack だけでは足りないものとは？

## □ 案件対応上、よく課題になる機能 (1/2)

### □ 課金系

- ✓ 課金モデルはユーザごとにまちまち
- ✓ OpenStack は課金のための Metering までにフォーカス

### □ ポータル

- ✓ OpenStack Horizon はAPIをほぼそのまま見せる発想。一方、カタログ化やワークフロー化の需要も強い。
- ✓ 既存ポータルとの連続性。自社要員のスキル(Java)等

### □ 運用管理系

- ✓ 優れた運用監視ツールは、OSSにも商用にも多数存在
- ✓ 通常、ユーザ個別に使い慣れた既存運管ツールが存在

---

# OpenStack だけでは足りないものとは？

## □ 案件対応上、よく課題になる機能 (2/2)

### □ オーケストレータ

- ✓ OpenStack にもHeat が存在。しかし、Heatと商用ジョブ管理ツールでできる範囲にはギャップがある

### □ 構成管理

- ✓ OSSにもPuppet/Chef/Ansibleがあるが、勢いは時代によって変化

# OSS / ディストリビューション / アプライアンス



## 自力で組む！

- なんでもできる

- HAなど、実運用に耐える環境を作り上げるのは相当大変
- すべては自己責任



## ディストリビューションを活用

- ディストリビューターにより検証された構成
- 入れれば動く

- ディストリビューターのサポートを受けるためには各種制約を受け入れる必要がある
- お手軽であるがゆえに、機能をよく理解しないまま導入してしまう



## アプライアンス製品を利用

- 迅速な導入
- HWも含めたベンダーのサポートが得られる

- 左記に加えて
- アプライアンスとして事前構成済みであることによる制約(カスタマイズの制限)
- 仮想化による制約



# 設計/構築編



---

# ノード配置

## – 考慮すべき点

### □ 可用性

- OpenStackコントロールノード(Core API / Network Service / DB / MQ)
- オブジェクトストレージ(Swift Proxy / Account / Container / Object)
- ブロックストレージ
- 運用系機能(Monitoring / Metering / Logging)
- インスタンス ……?

### □ インスタンスの実行環境(Hypervisor / Container / BareMetal)

### □ パフォーマンス

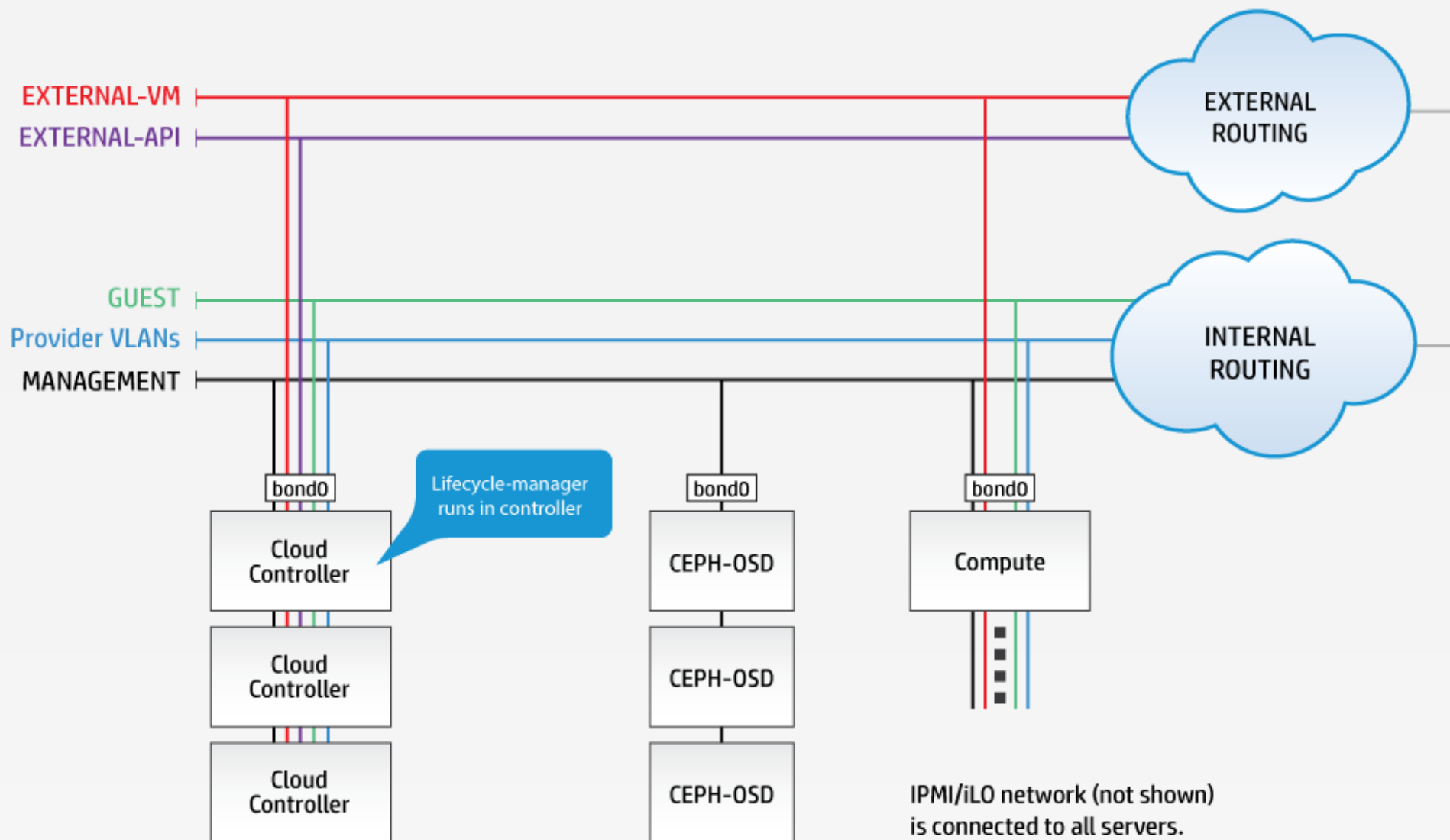
- (特に)ネットワーク

### □ スケーラビリティ

– 次ページ以降、Helion OpenStack 2.0のExample Configuration  
[http://docs.hpcloud.com/helion/example\\_configurations.html](http://docs.hpcloud.com/helion/example_configurations.html)

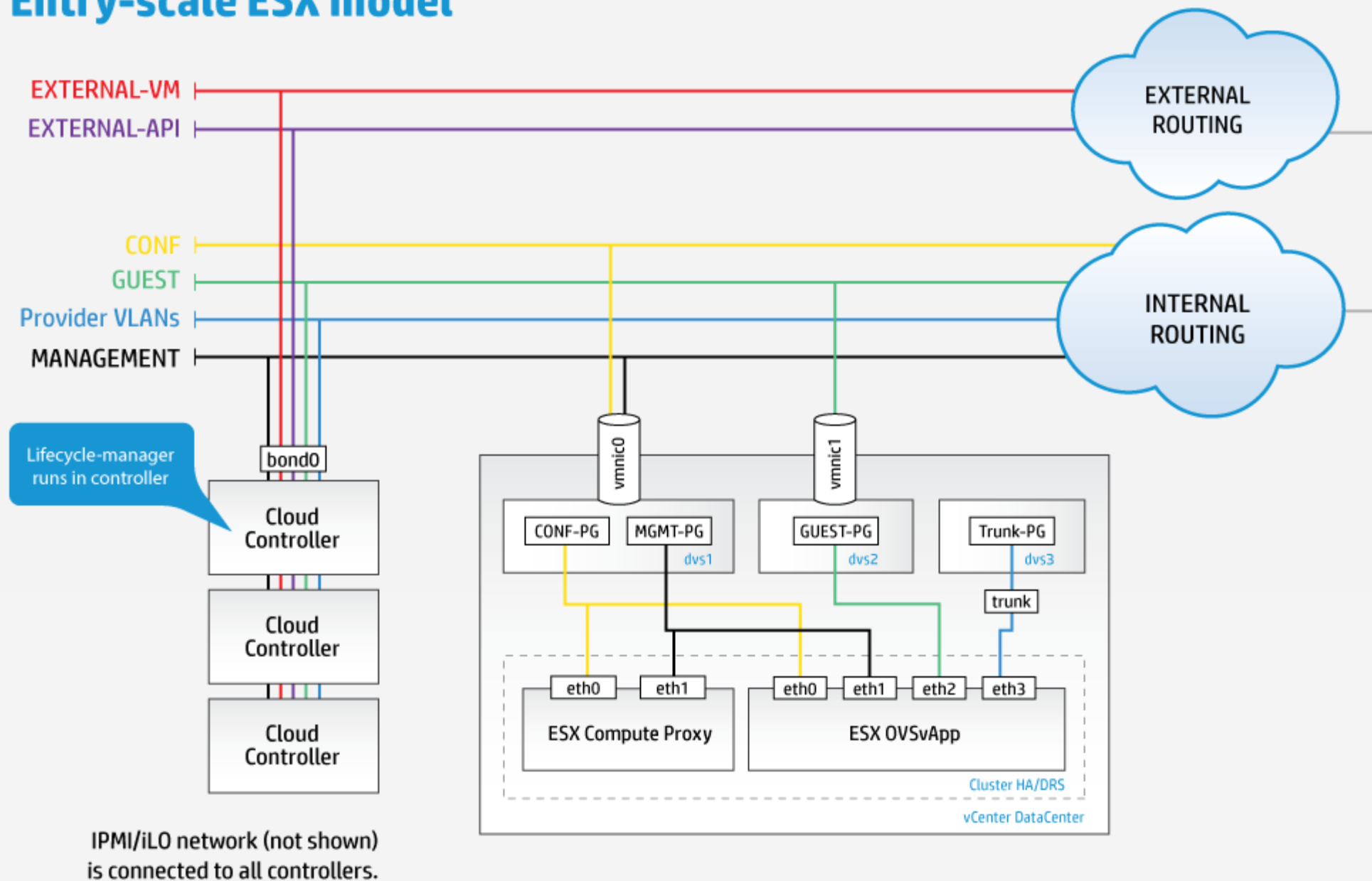
# ノード配置 例①

## Entry-scale KVM with Ceph model



# ノード配置 例②

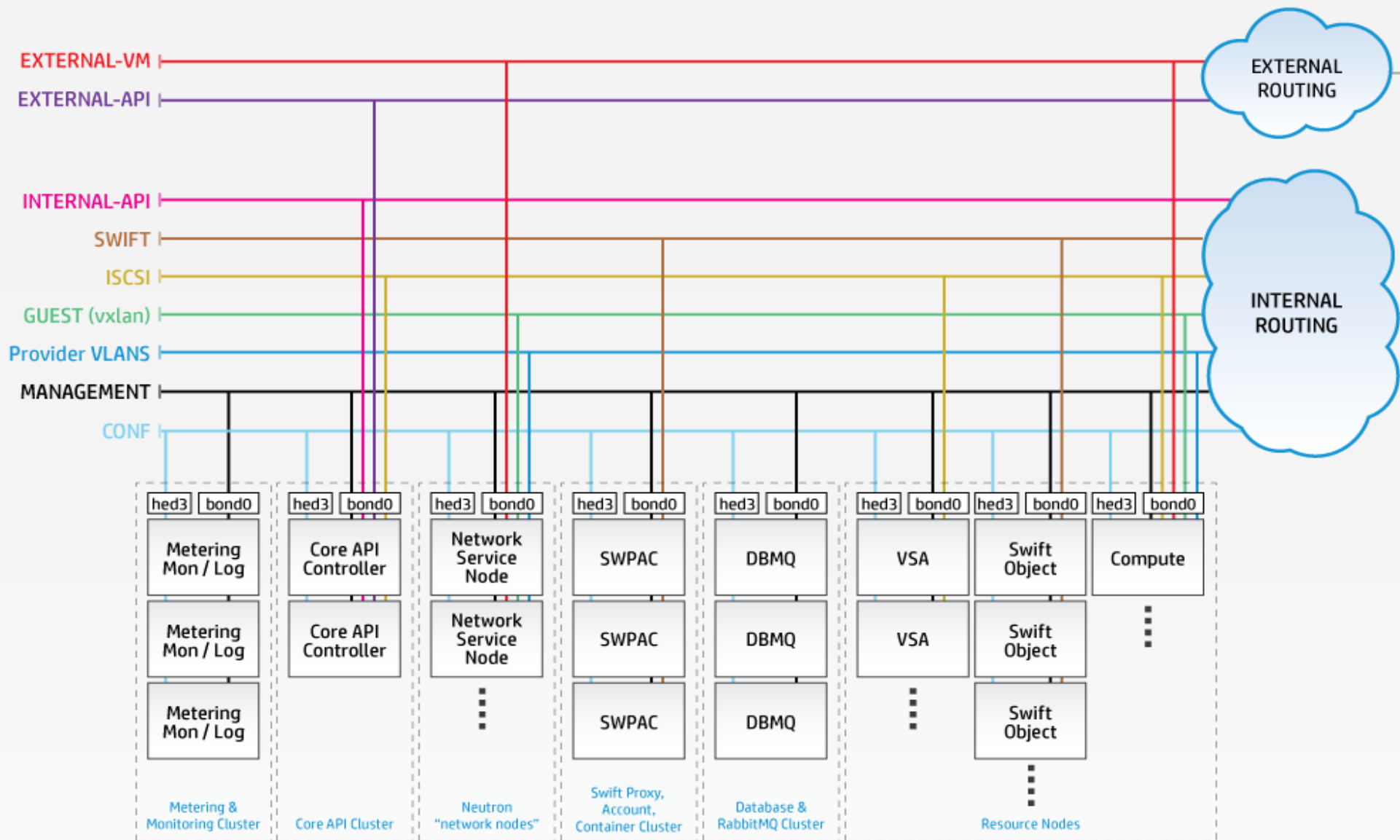
## Entry-scale ESX model





# ノード配置 例③

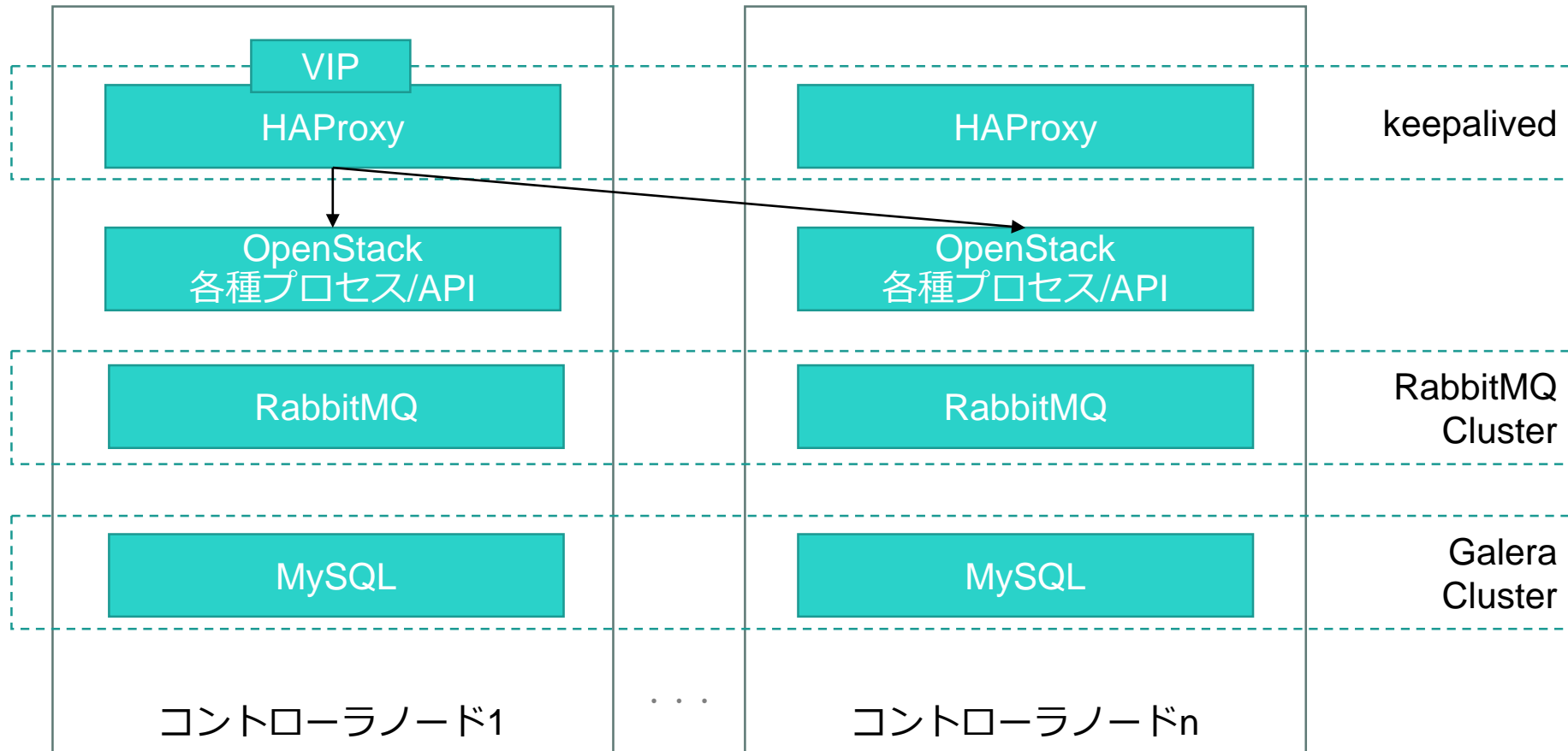
## Mid-scale KVM with VSA model



IPMI/iLO network (not shown) is connected to all controllers.

# コントロールプレーンをHA化するのはけっこう大変。

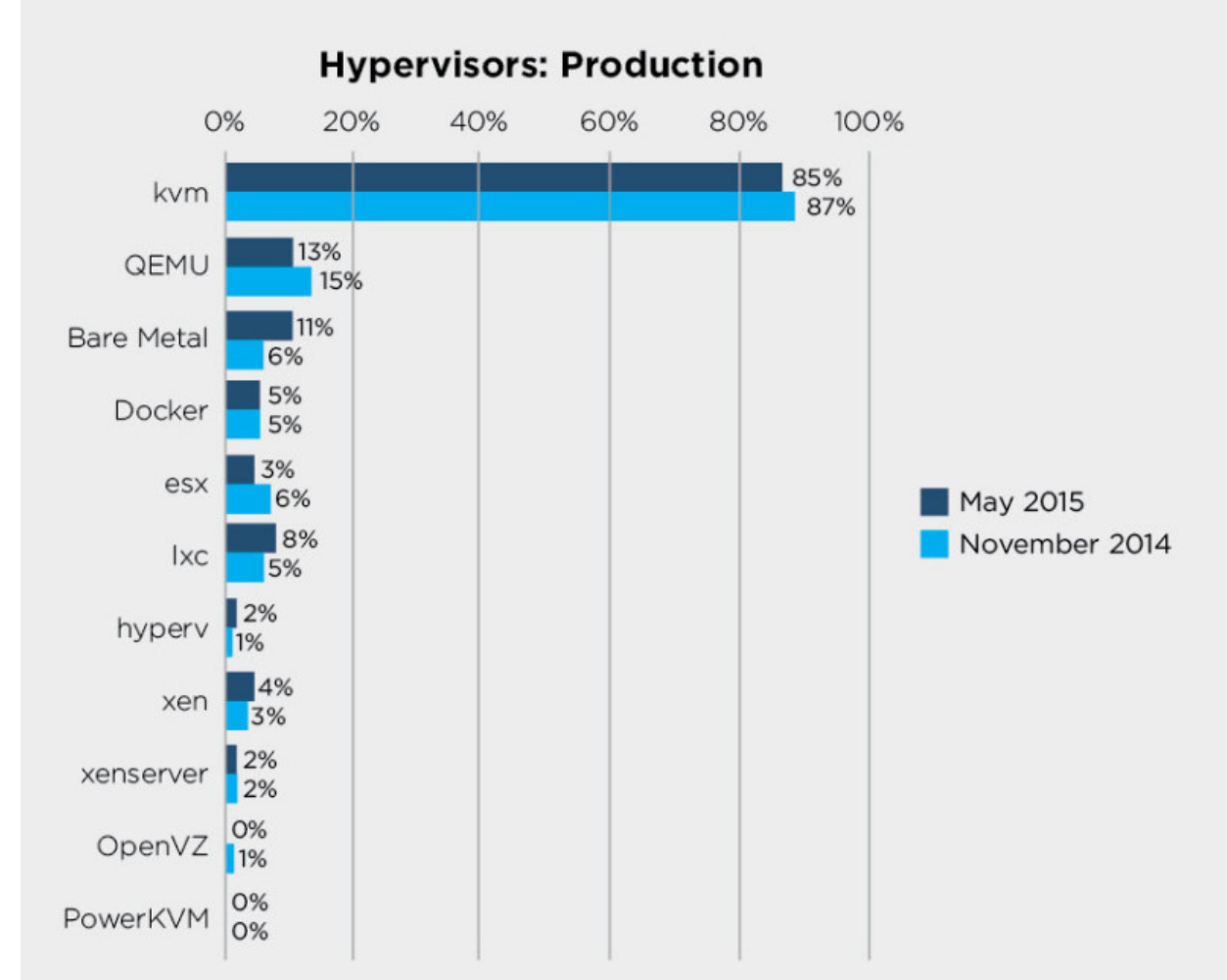
これだけ見ると簡単そうに見えるけど・・・



# コンピュート (Nova)

## - 考慮すべき点

- Hypervisorの選択
  - 基本はKVM
  - マルチHypervisor構成の場合はネットワーク/ストレージの制約に注意
  - BareMetalは今後に期待
- ノード追加/削除のオペレーション

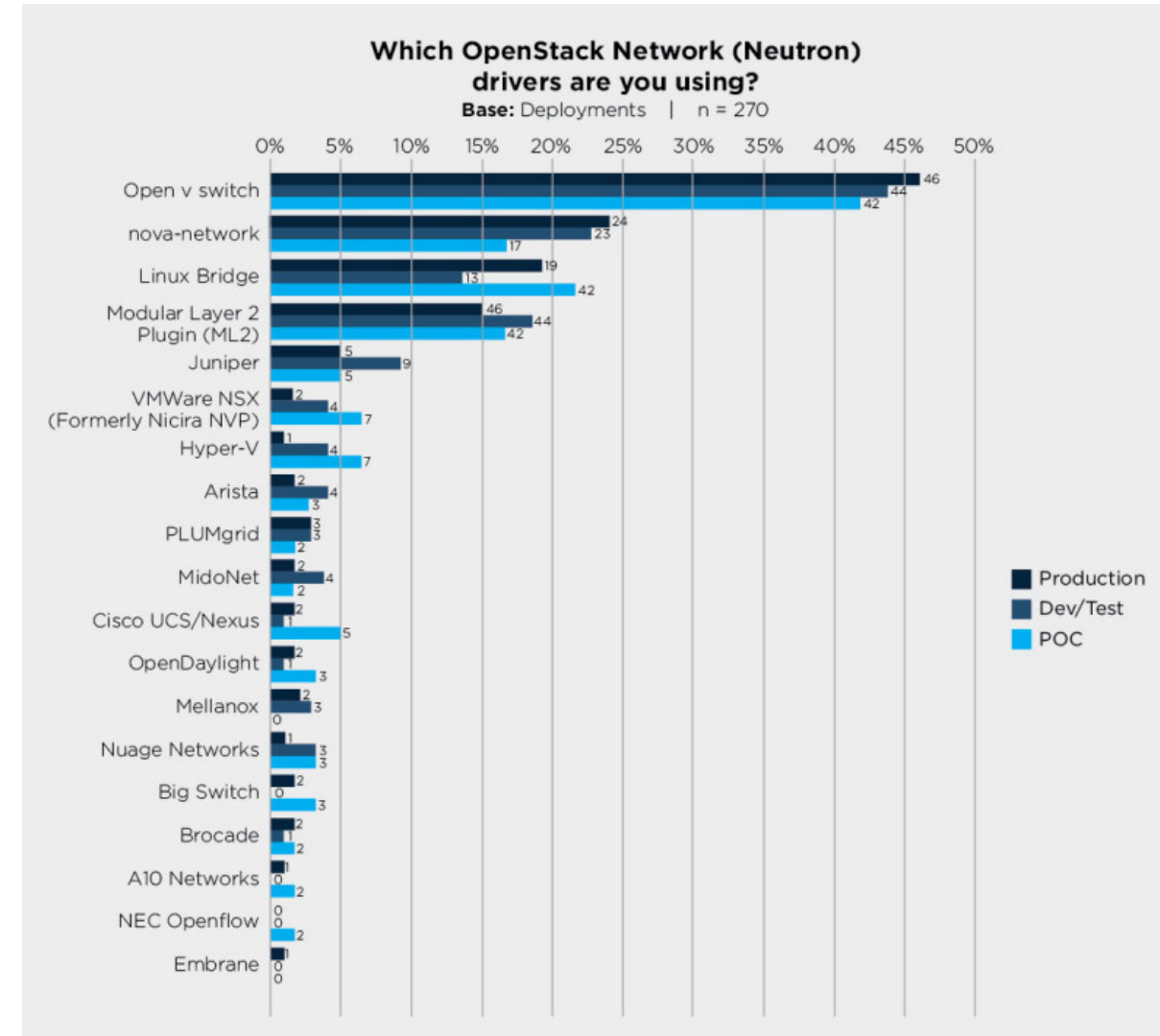


<http://superuser.openstack.org/articles/openstack-users-share-how-their-deployments-stack-up>

# ネットワーク (Neutron)

## - 考慮すべき点

- 基本はML2+OVS
- 商用SDN製品を使うべきか？
- 外部からのアクセス経路(External Network)
- テナントネットワークの種別(フラット / VLAN / VXLAN)
- Provider Networkの要否
- 仮想routerの負荷(Neutron DVRを使う?)
- 各種ネットワークサービスの要否(LBaaS, FWaaS, VPNaaS)



<http://superuser.openstack.org/articles/openstack-users-share-how-their-deployments-stack-up>

# ストレージ

## – 考慮すべき点

### □ オブジェクトストレージ (Swift)

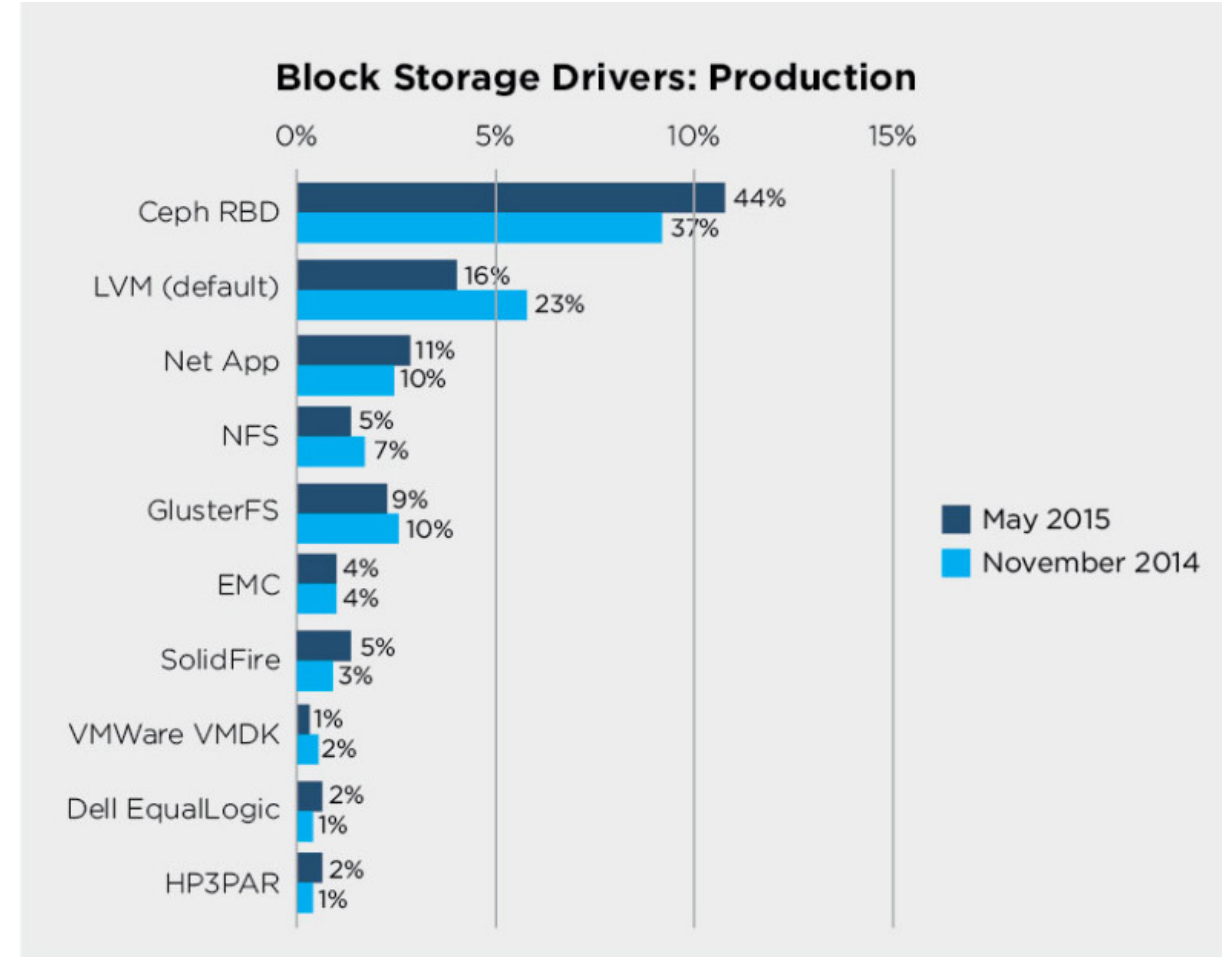
- PACの構成
- レプリケーション
- Glanceのストアとして使う？

### □ ブロックストレージ (Cinder)

- Ceph人気
- 箱モノ使う場合はFCかiSCSIか
- スナップショットなど、ストレージ側の機能にどこまで期待するか
- なんだかんだ、サポートの問題な気がする

### □ Nova/Glanceとの絡み

- Boot from Volumeとか



<http://superuser.openstack.org/articles/openstack-users-share-how-their-deployments-stack-up>

---

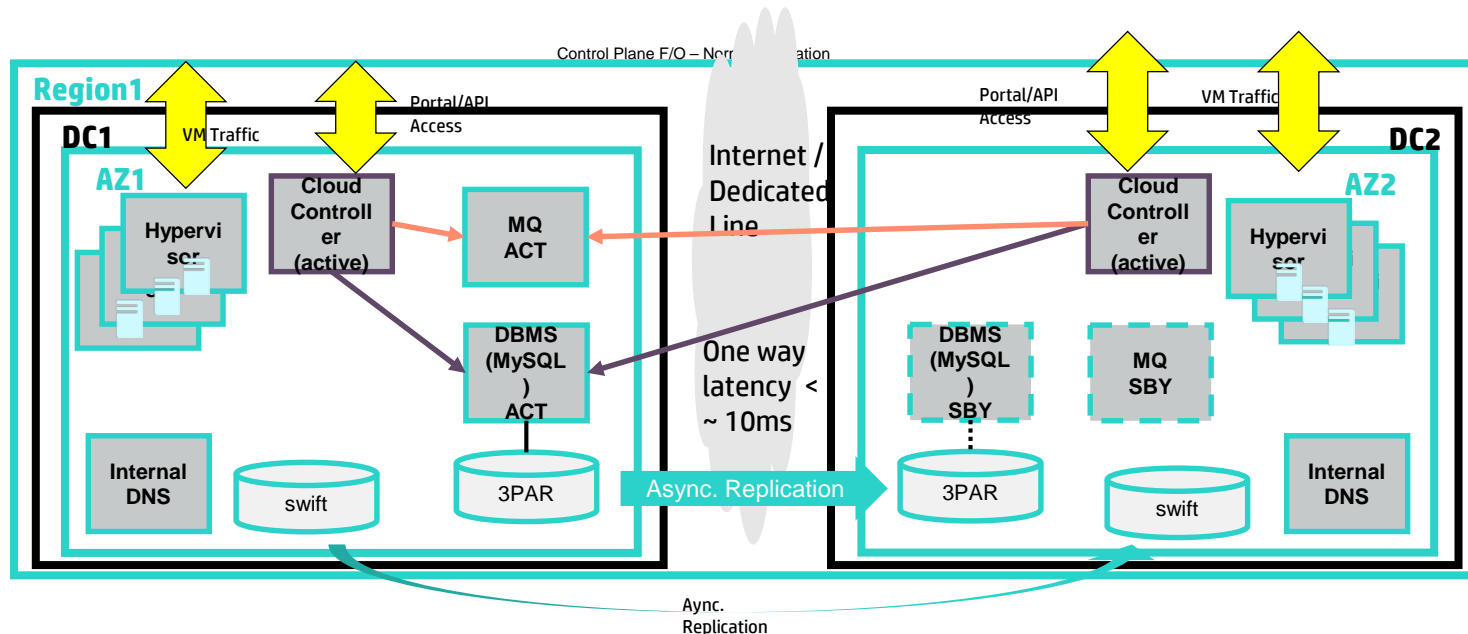
# バックアップ/リカバリ

- OpenStackのバックアップって？
  - DB
  - 各種定義情報 (Nova, Glance, Keystone, Cinder, Swift等)
  
- 「OpenStackでは仮想マシンのシステムバックアップはどうなるのかね」
  - (知らんがな。。。)
  - クラウドネイティブなアーキテクチャであれば、インスタンスは再生成すればよい(Glanceイメージが残っていればよい)
  - ユーザデータのバックアップはテナント側で取ってください、が基本
  - とは言えインフラ側にそこまで求められる場合は
    - インスタンスのImage Snapshot
    - ブロックストレージ上のデータはVolume Snapshot とか
    - エージェント型のバックアップツールを使ってBackup as a Service的な

# DR

## □ このぐらいはできる

- 引き継ぐのは管理情報(ユーザアカウント、Project構成情報など)、Swiftのデータ、Glanceイメージ
- 手動でのサイト切り替え
- インスタンスはユーザが再立ち上げ
- Cinderボリュームのデータはバックアップからのリストア(バックアップはSwiftで同期)



## □ ここまで求められるとハードル高い

- 自動サイト切り替え
- 落ちたインスタンスが同じ姿で自動的に立ち上がってくる

## □ そもそもクラウドネイティブなアプリであれば

- クラウド自体がDRしなくても、別のクラウドにアプリをデプロイすればよいのでは？



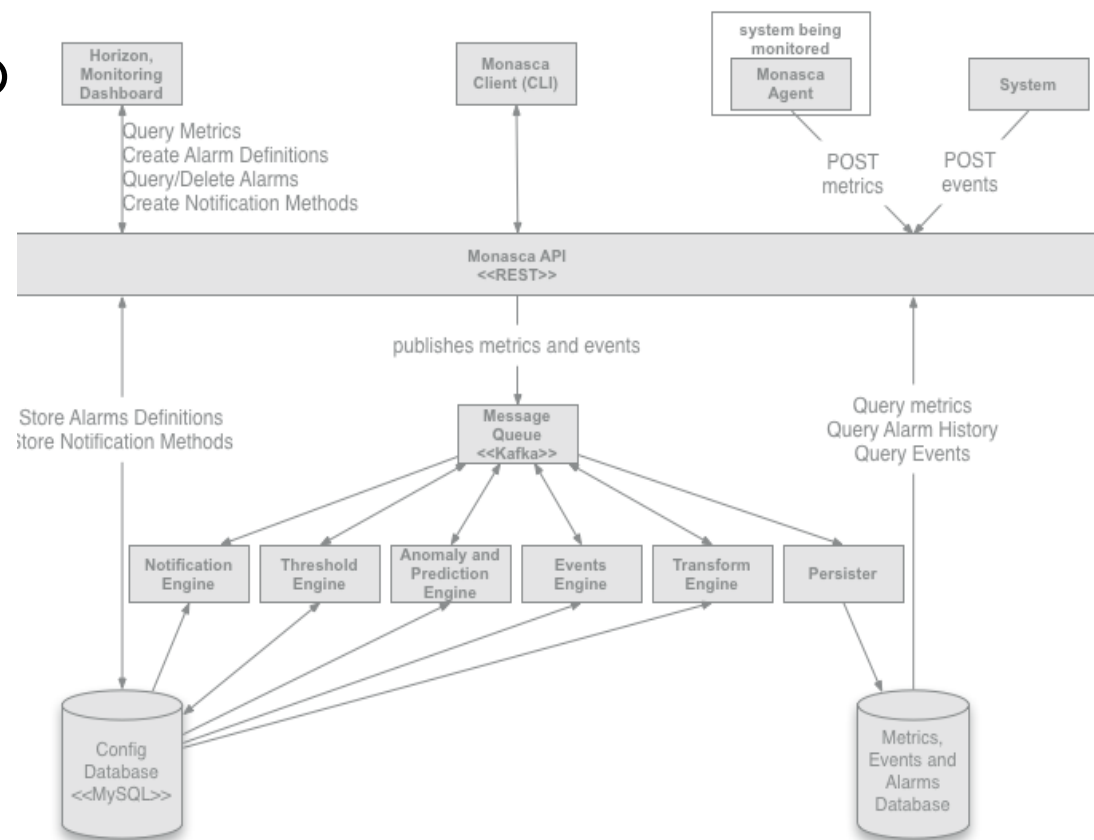
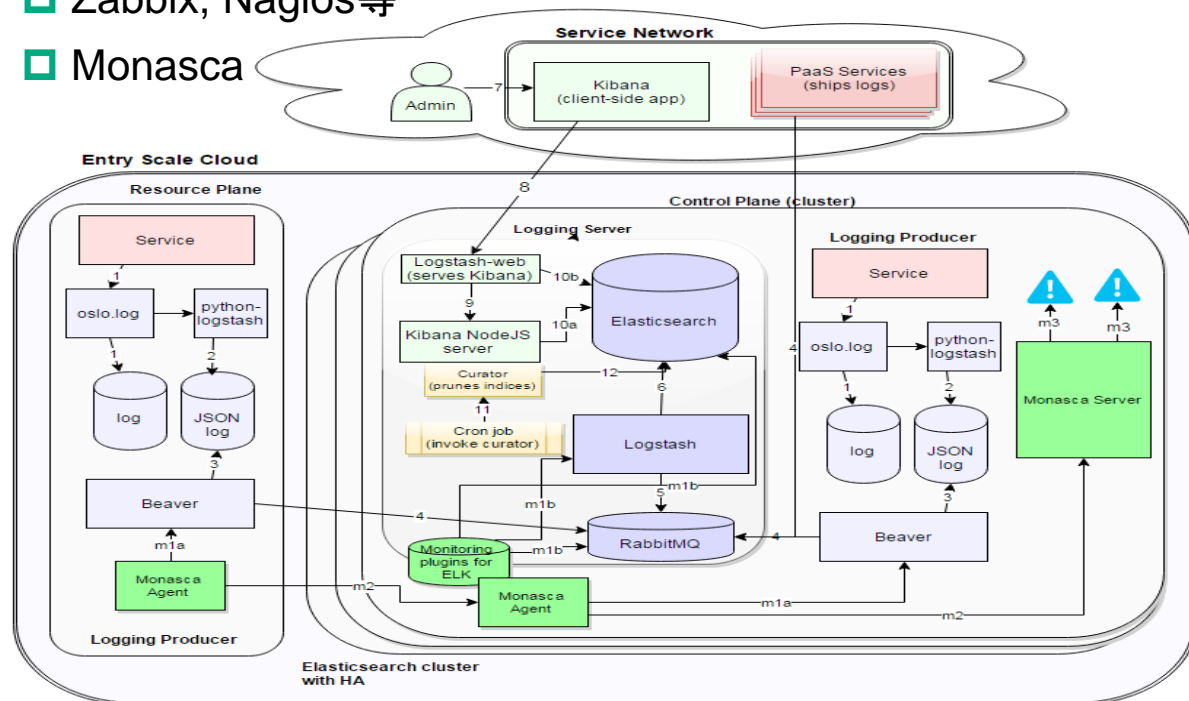
# ロギング/監視

## □ ログ:集中管理の仕組みが必要

- rsyslog
- Logstash / Elasticserch / Kibana

## □ 監視:クラウド管理者のための監視とテナント管理者のための監視

- Zabbix, Nagios等
- Monasca



# まとめ





---

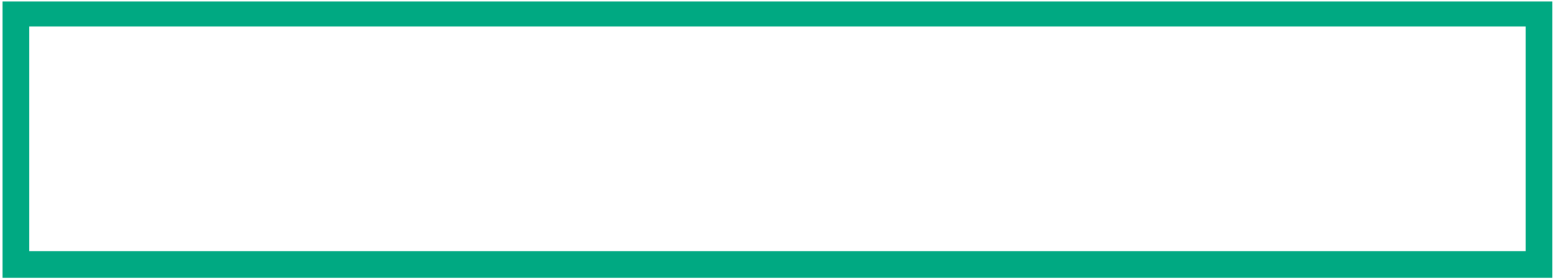
## □ 「基本思想」が重要

- 基盤もアプリもクラウドネイティブ化を志向する
- シンプルな仕様・構造からはじめ、スケールできるようにする

## □ 構成/設計の選択肢は無限大

- しっかり理解して使いましょう。それが強いサービスを生み出す現場力の源泉となります。
- 悩んだら、ディストリビュータのリファレンスアーキテクチャに素直に乗っかるという手もあります。

## □ まずは触ってみる！



**Thank You**