



Openstack 導入事例

ヤフー株式会社

システム統括本部基盤システム開発本部 インフラ技術1部

伊藤 拓矢

2013年12月9日(月)

- ・ 自己紹介
- ・ OpenStack導入に至る経緯
- ・ 導入システム構成
- ・ OpenStackを選択した事で困った事/苦勞した事
- ・ 今後



自己紹介

- ヤフー株式会社 (2008年新卒入社)

入社以来インフラ、主にネットワークを担当

2010年から IaaS開発に片足を突っ込む

2013年には IaaS開発に両足を突っ込む

- 現在の業務内容

検証、設計、開発、運用、障害対応、部外交渉、予算管理、

教育、物理から仮想までなんでもやる



OpenStack導入に至る経緯

- ・ 2008、2009年ころ

VMの利用はこの頃から開始

VMは次Qに利用する数をサービス担当者が申請

申請数に応じて予算化し物理サーバを取得

CLI管理ツールからVMを管理

といった感じで、VMの提供まで3か月ほど掛かる

開発環境のみ 数百VM程度の運用



OpenStack導入に至る経緯

- ・ 2010、2011年ころ

VMを管理しきれなくなる

WebUIからサービス担当者がVMを作成できるシステムの
開発着手

CLI管理ツールをWebでwrapした感じの実装

物理サーバと同じインストールプロセスを踏む

仮想用イメージの作成をしなくても済む

開発環境のみ 数千VMの運用

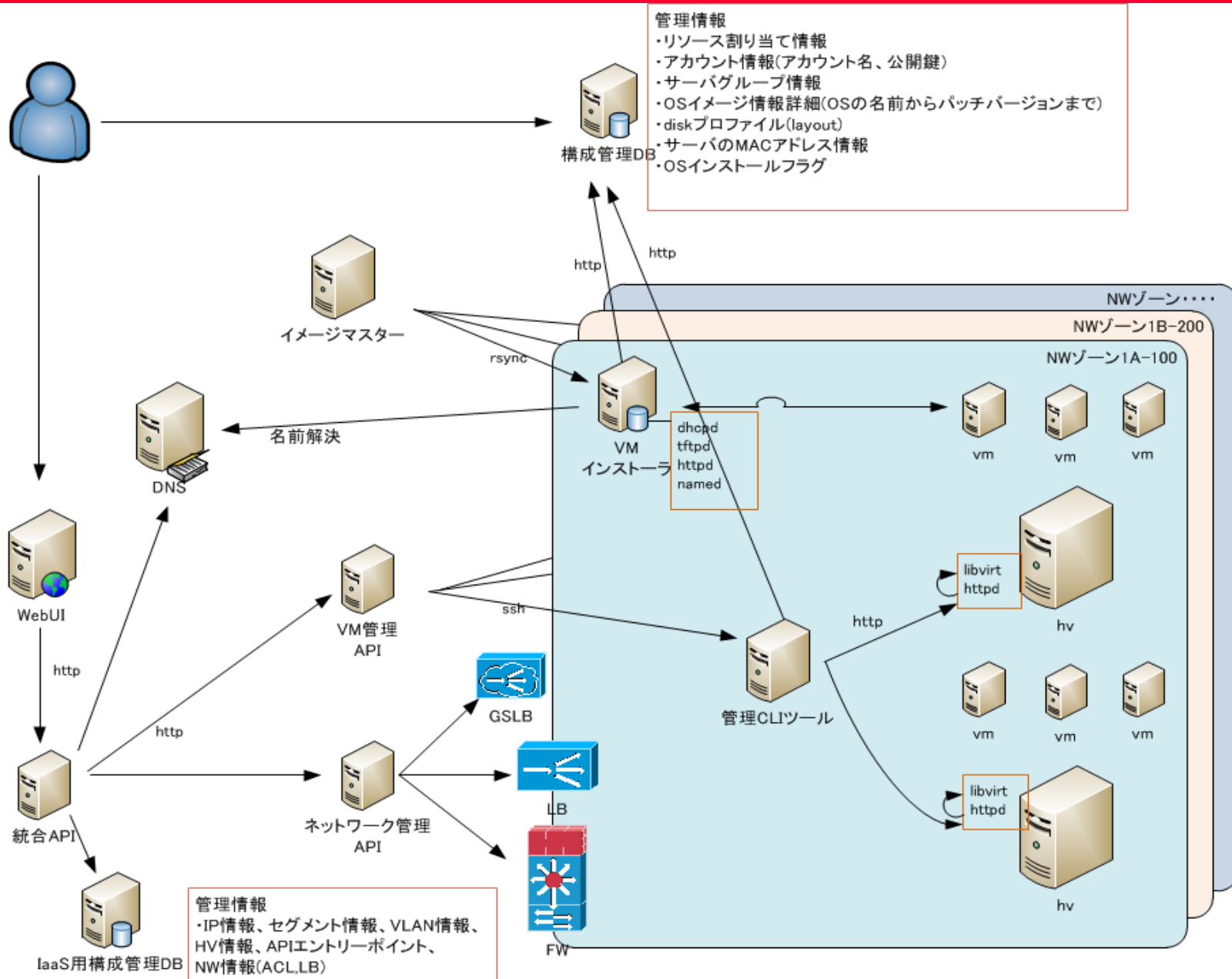


OpenStack導入に至る経緯

- ・ 2010、2011年ころ WebUIによる下記の機能
 - リソースを事前取得してVMを作成(申請してリソースを得る)
 - ロードバランサの設定(VIP、LBaaS)
 - GSLBの設定
 - Ciscoスイッチのコンフィグ変更(ACL、FWaaS)
 - Volumeストレージの設定(動的attach、dettach)
 - DNSの自動設定と任意の設定
 - リソース情報の提供
 - WAF、IDSからセキュリティ情報の提供
 - 社内システムとの連携(構成管理、アカウント管理)

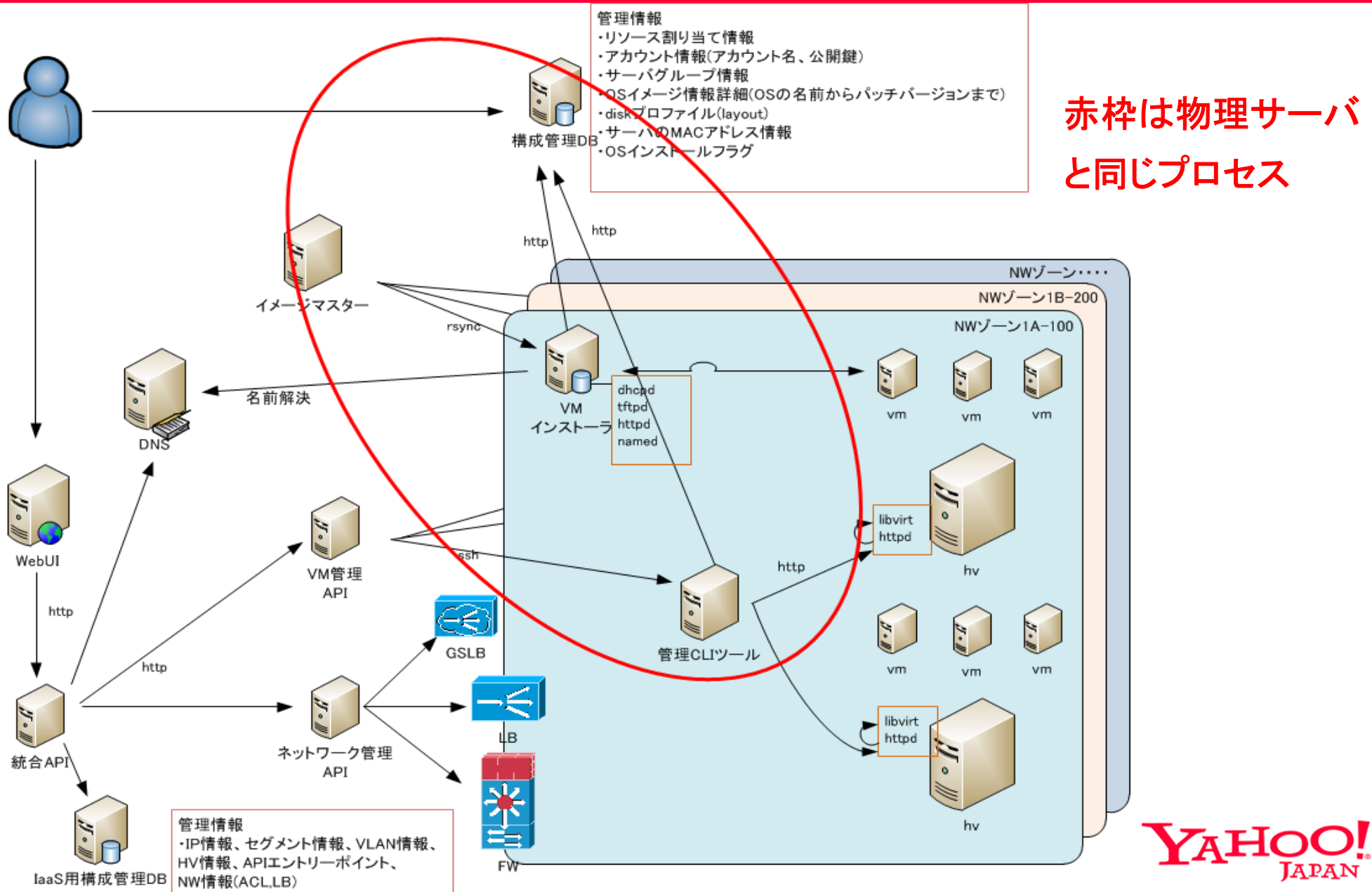


OpenStack導入に至る経緯





OpenStack導入に至る経緯





OpenStack導入に至る経緯

- ・ 2012年

- 技術的に古くなりパフォーマンスも良くない

- 好きな時に好きなだけVMを作りたいニーズも高まる

- APIは独自のため、OSSとの連携が悪い

- コンポーネントの依存が激しく、VMのbootまで時間がかかる

- 社内は爆速がキャッチフレーズに

開発環境とプロダクションでの運用 数万VMの運用



OpenStack導入に至る経緯

・ 2013年

Openstack、CloudStackの検証を始める

社内システムとの連携が必須のため、改良し易いOpenstackを選択

旧来の機能を求めるも、無いため機能を割り切る

ベーシックな機能は問題なく、すぐに導入を決定

APIもサービス担当者は利用可能になり、Jenkinsなどとも連携

夏頃、開発環境で提供を開始

最近、プロダクション環境でも提供を開始

1Qで数千VMほど利用して頂いている状況

10 900テナント、4000ユーザ



導入システム構成

- ・ 利用しているOpenstackのコンポーネント
Keystone, Nova, Quantum, Cinder, Glance, Horizon
- ・ Openstackを動かすために
mysql, qpid, rabbitmq, kvm, linuxbridge, CentOS6.4
手を加えた qemu (L3DSR動かないので)
- ・ その他
config driveによる社内システム情報の挿入(アカウント、構成管理)
(ホスト名、アカウント、公開鍵、IPアドレス、GW、ntp.conf、
resolv.conf、sudoers、group)



導入システム構成

- ・ 構成のポイント

 - 安定して稼働できる

 - パフォーマンスが出る(CPU, IO, Network)

 - 内部統制を実現してしっかり守る仕組み

 - 誰がいつ作成したのか、作成後、即座にポリシーの強制を開始

 - 運用できること(必要なロックは掛ける)

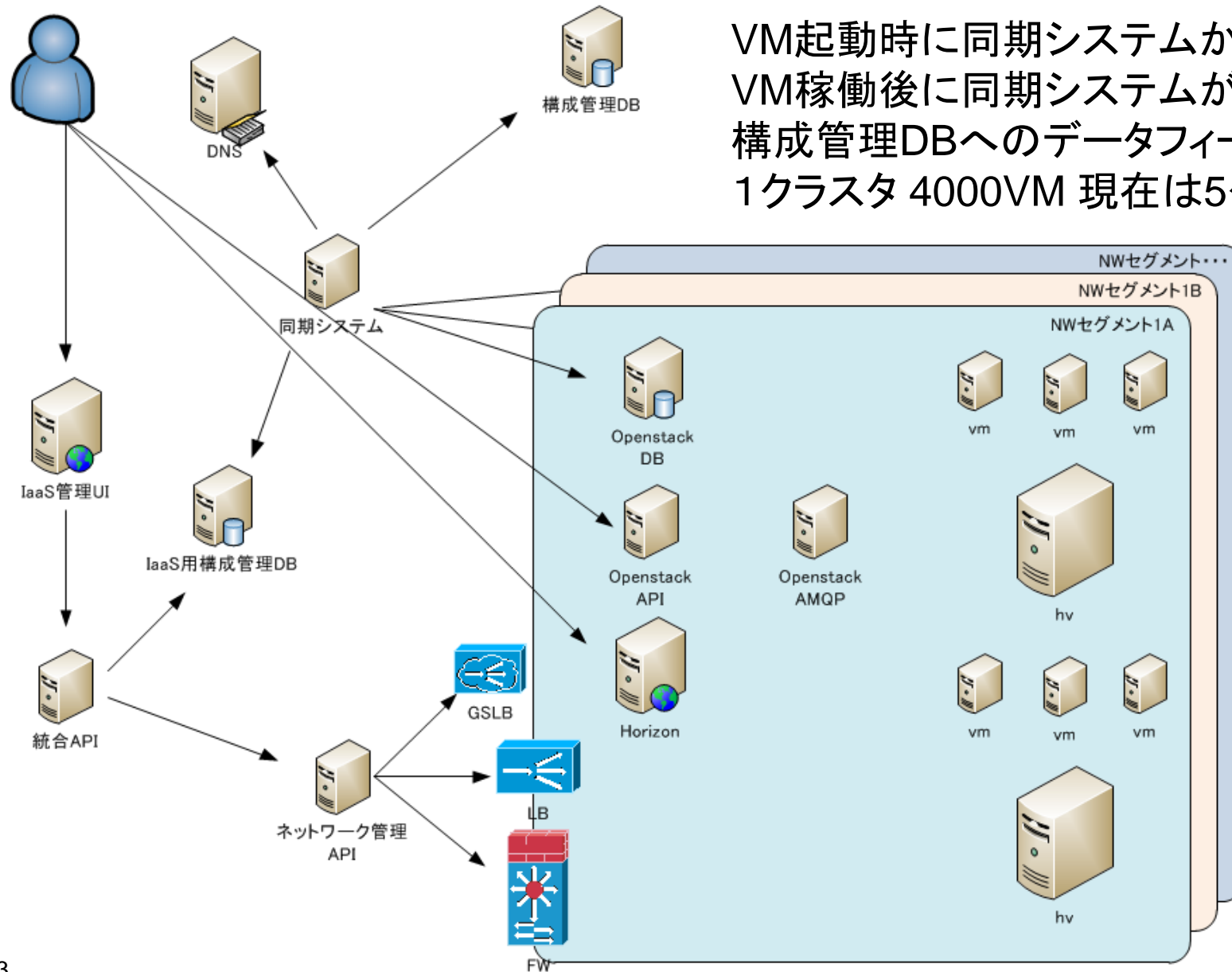
- ・ 使っていない機能

 - Openstackに登録した公開鍵の配布機構

 - FloatingIP



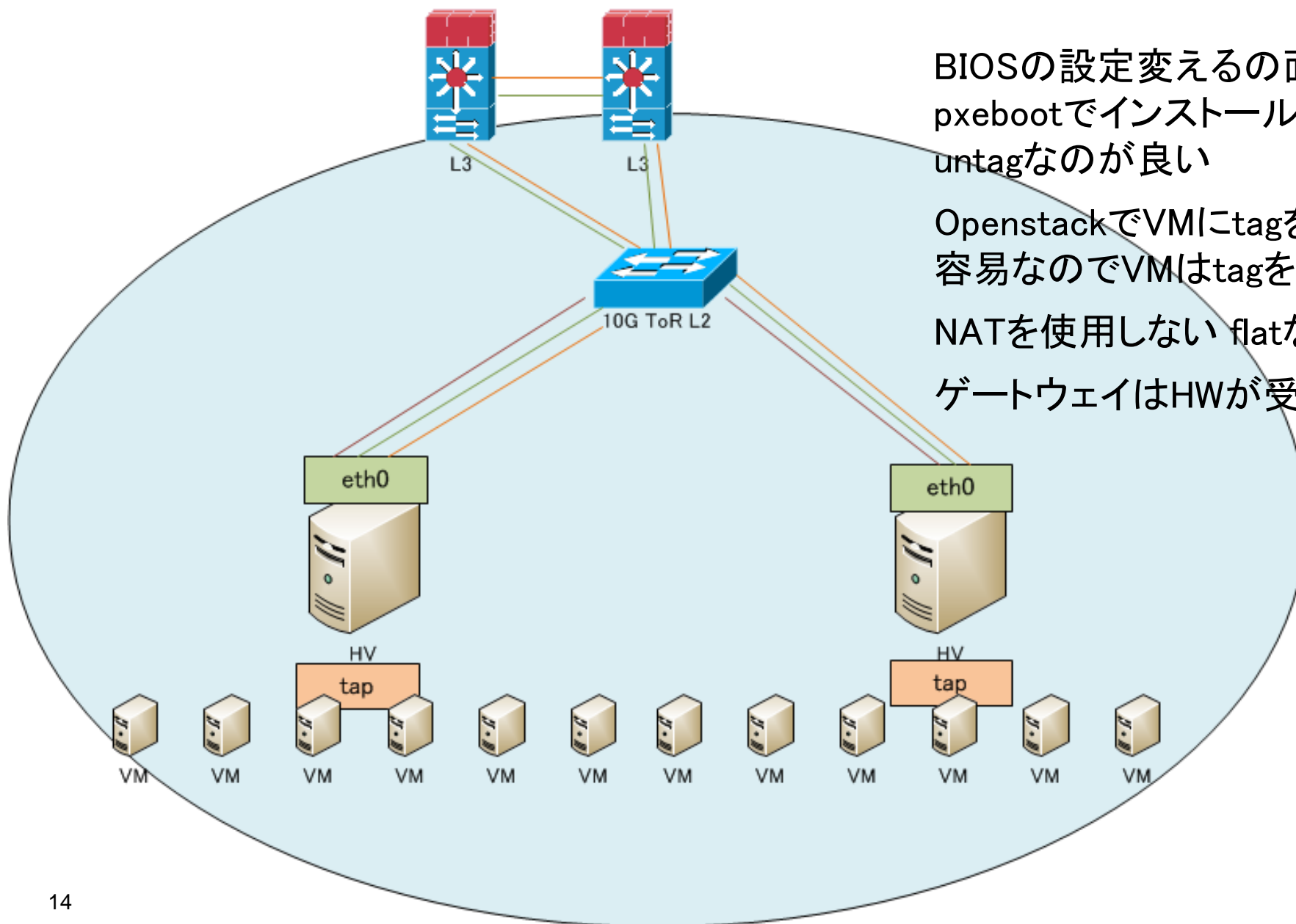
導入システム構成図 全体



VM起動時に同期システムから構成情報を渡す
VM稼働後に同期システムがDNS紐付けや
構成管理DBへのデータフィードを行う
1クラスター 4000VM 現在は5クラスターほど運用中



導入システム構成図 VM net



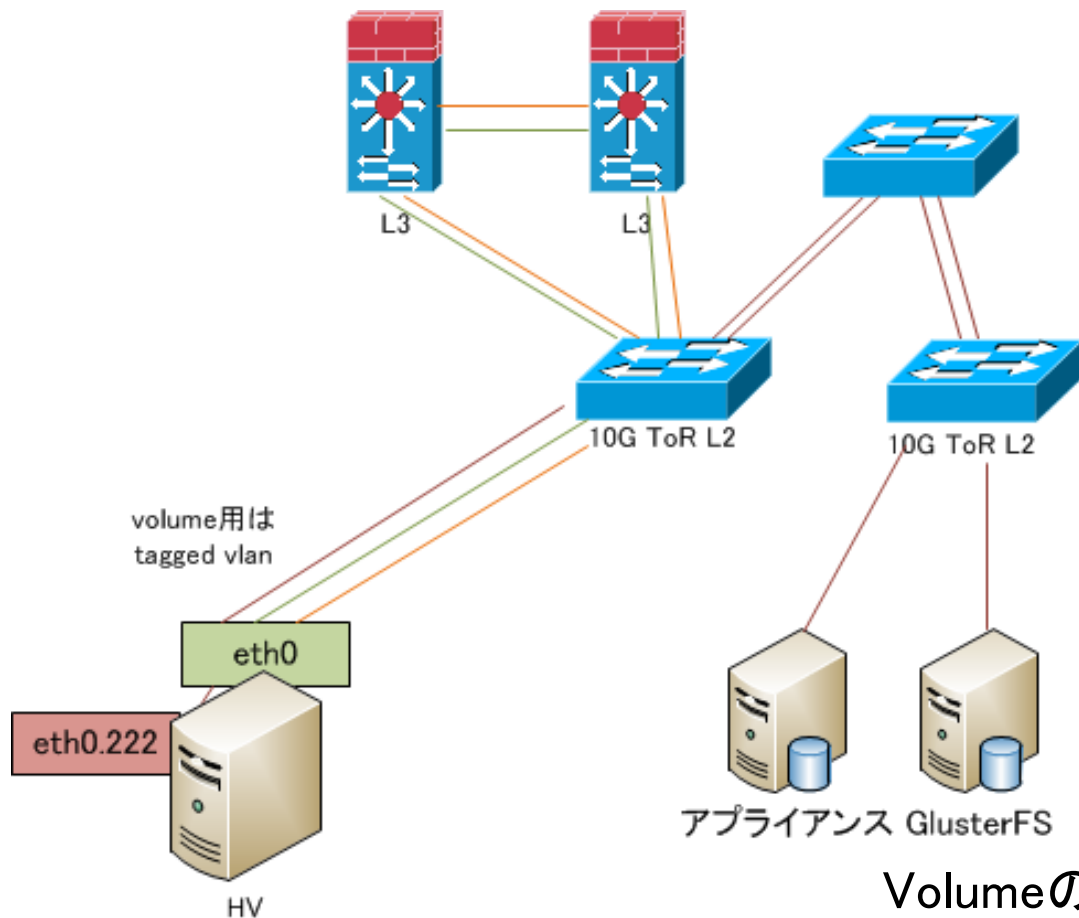
BIOSの設定変えるの面倒なので、
pxebootでインストールされるHVは
untagなのが良い

OpenstackでVMにtagを付けるのは
容易なのでVMはtagを付ける

NATを使用しない flatなnetwork構成
ゲートウェイはHWが受け持つ



導入システム構成図 Volume



Volumeの提供はToRSWから
枝分かれしたネットワークで行う

1G時代はToRを2つ置き、物理配線を分けていた
ので昔の名残という側面も



導入システム構成 config_drive

- ・ 肝となるシステム

cloud-init と独自のinitスクリプトで拡張まわりの実装

VMのイメージには、ランチャーのみ配置

ランチャーはconfig_driveに付属するスクリプトを実行する

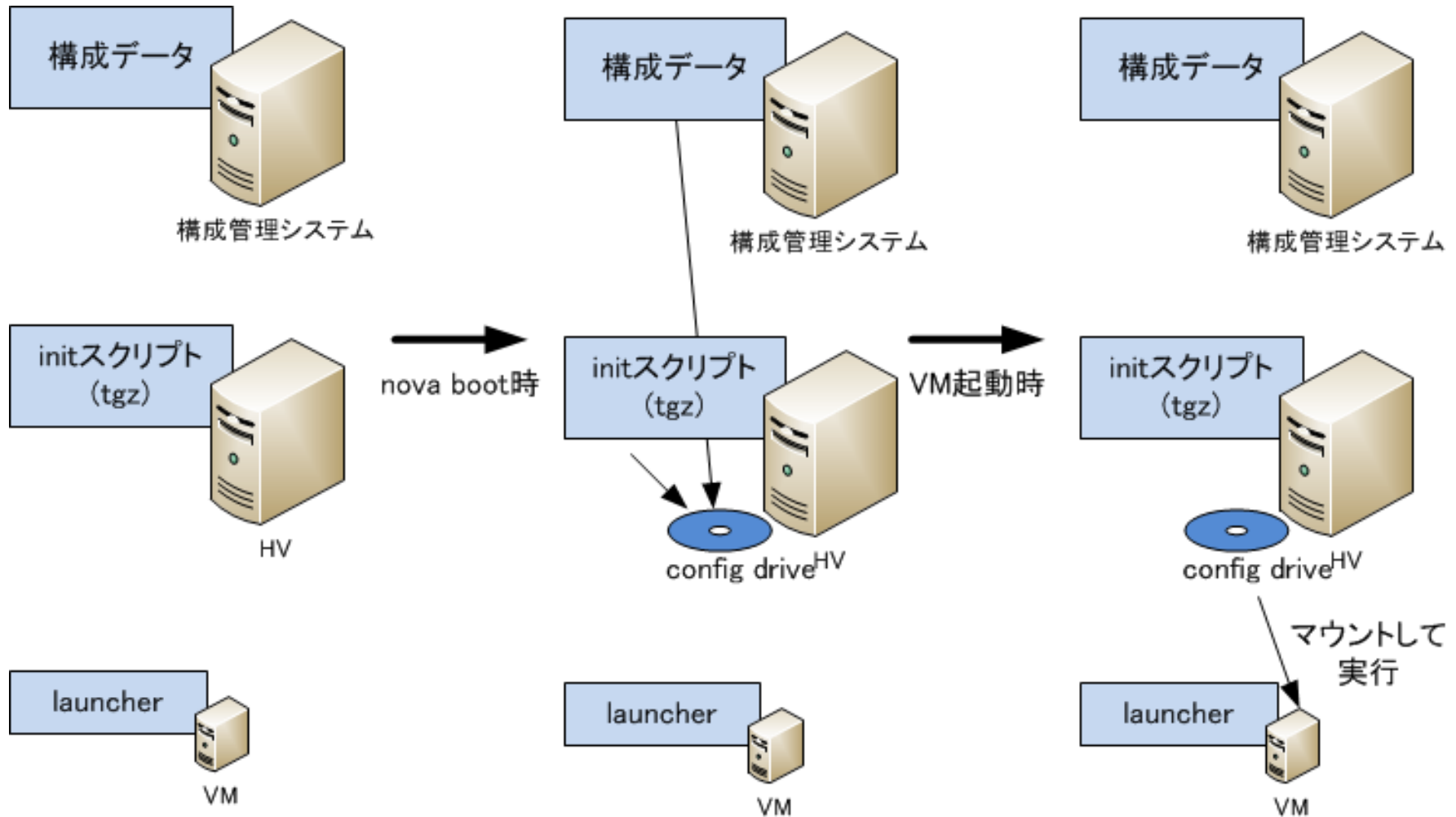
スクリプトは構成管理データを読み込んで、ファイルの配置変更を行う

頻繁に更新が入るために、スクリプトの実体はHVに配置

VM起動時点で既に、構成管理システムと接続された状態になる



導入システム構成 config_drive





Openstackの開発と管理

- ・ 社内のgithubとJenkinsを利用してUpstreamと独自コードのマージ
- ・ Jenkinsによる自動テスト、自動ビルド
- ・ QA環境に持って行って、展開、テスト
- ・ Openstack+LBの動作テストといったアプライアンスとのテストはまだまだ手でやる
- ・ 今のところは構築、運用に工数を取られる状況が続き、仕様変更をここで知る・・・



OpenStackを選択した事で困った事 / 苦労した事

- ・ ロードバランサなど、IaaS用途で使いたい機能面の弱さ、設計の甘さ
- ・ amqpの高い稼働率、起こる不具合原因はほぼここ
- ・ Upstreamを追いかけるのが大変・・・
- ・ 仕様がよく変わりますね



- Neutronまわりもっと検証
ベンダーAPIとの連携
オーバーレイNW(パフォーマンスと管理性の問題解決)
様々な社内要求NWへの対応
- baremetal改め、Ironic導入
VMの柔軟性や近年のオーバヘッドの極小化を加味しても、物理サーバの利点は大きい。TripleO
- OSS連携の強化
開発の効率化、テストの自動化、本番環境Deployの高速化



今後

- nova-cells の検証、実装
本番環境で利用したい
- 社内ニーズに応じた実装
サービスごと望む機能が違う
同じサービスでも開発者、運用者で目線が違う
ただ、様々なニーズを実現できるOpenstackは良い

