

# アドテクに必要なSDN/NFV (前編)

@SDN Japan 2016/1/15

サイバーエージェント  
アドテク本部

山本 孔明

# Agenda

1. はじめに ~会社紹介と今回のテーマ~
2. ネットワーク自動化(?)をやってみた話
3. REST APIのお話
4. 苦労したポイントとわかったこと
5. おわりに

# サイバーエージェントについて

1998年の創業以来、インターネットを軸に事業を展開し、現在では代表的なサービスである「Ameba」をはじめ、スマートフォン向けに多数のコミュニティサービスやゲームを提供しています。

## アドテク本部とは

インターネット広告において、広告配信の最適化やメディアの収益最大化という観点からアドテクノロジーの重要度が高まっています。

サイバーエージェントではアドテクノロジー分野におけるこれらのサービスについて各子会社を通じ開発しておりましたが、各サービスの開発部門を横断して組織化する専門部署としてアドテク本部が設立されました。

# 自己紹介

## プロフィール @komeinw

アドテク本部のインフラエンジニア。  
主にオンプレのネットワークを担当。

- 「ネットワークを監視するZabbixの活用事例」  
Mellanox@Interop Tokyo 2015

<https://speakerdeck.com/komeiy/it-infra-summit-2014-network-zabbix>

# 今回のテーマについて

サイバーエージェントではアドテクサービス向けのクラウド基盤としてOpenStackを採用しています。

スイッチ、ルータといったネットワーク機能はもちろん、ロードバランサーなどネットワークサービスもAPI経由でユーザーに提供しています。

本講演では変化の激しいサービス基盤を少人数のチームで運用するため、ネットワークも含めたクラウドをどのように構築し効率的に運用しているのかについてお話します。

また、後半ではOpenStack LibertyとMidoNetを使って構築している環境についてご紹介します。

# 一般的なネットワークのお仕事

- VLAN追加/変更
- ルーティング追加/変更
- ロードバランサー設定追加
- ACL追加
- トラフィック確認
- VPNユーザ追加/変更

- 増強などの構成変更
- 新技術の検証
- トラブルシューティング

ここに注力したい

- 構成管理
- etc...

# ネットワークエンジニアの悩み

「ACL追加してほしいんだけど。今日」

「VPNユーザって今誰が登録されてましたっけ？」

「トラフィックっていまどのくらい出てます？」

**面倒に思ったり後まわりにしたいと思いつながら作業した経験はありませんか？**

```
169
170
171 var bool,
172     docElem = doc.documentElement,
173     refNode = docElem.firstElementChild || docElem.firstChild,
174     // fakeBody required for all cross-browser (including Opera < 12.0)
175     fakeBody = doc.createElement( "body" ),
176     div = doc.createElement( "div" );
177
178 div.id = "mq-test-1";
179 div.style.cssText = "position: absolute; top: -999px; width: 0px; height: 0px; border-left: 1px solid black; border-right: 1px solid black; border-top: 1px solid black; border-bottom: 1px solid black; margin: 0 auto; padding: 5px 0; text-align: center; font-size: 1em; font-weight: bold;">
180 fakeBody.appendChild( div );
181
182 return function( q ){
183
184     div.innerHTML = "<div style='width: 1px; height: 1px; border: 1px solid black; margin: 0 auto; padding: 0 5px;'></div>";
185
186     docElem.insertBefore( fakeBody, refNode );
187     bool = div.offsetWidth < 4;
188     docElem.removeChild( fakeBody );
189
190     return {
191         matches: bool,
192         media: q
193     };
194 };
195
196 }( document );
197
198 /*
199 * jquery.requestAnimationFrame
200 * https://github.com/greentree/jQuery-RequestAnimationFrame
201 * Requires jQuery 1.3+
202 *
203 * Copyright (c) 2012 Dave DeSantis
204 * Licensed under the MIT license
```

申請するくらいならその時間で設定しよう

ということで、作ってみました。

# コンセプト

- 自動化というよりセルフサービス化
- AWS CLI や gsutil みたいな感じ

## なんで？（おさらい含む）

- 少人数で運用してるから
- 新規サービスの立ち上げスピードを損なわないため
- つまらない作業は極力やりたくない

# ダメ！！

作業ミス、リリース日への影響、エンジニアのモチベーション低下

# 操作イメージと効果

```
PC$ axc nodelist
```

ADDRESS	NAME	SESSION	DESCRIPTION
10.1.1.1	web01	enabled	
10.2.1.1	db01	enabled	

こんな感じでのルータ・スイッチ・ロードバランサー、Neutronの設定ができる。

Regionの概念も持たせています

- 学習コストが低い（アプリケーションの開発者が固有のコマンド・GUI操作を覚える必要がない）
- 簡単にJOBに組み込んだりすることが可能
- 思い立ったときにすぐインフラの操作ができる



# サービスメニュー

- 新規サービスが立ち上がるごとに増えるものや操作するものを基本に構成
- 基幹系の作業は一部のユーザの権限のみでしか触れないように制御
- サービスメニューは右記以外にも複数存在
- DC内のローカルレポジトリから配布

## axc

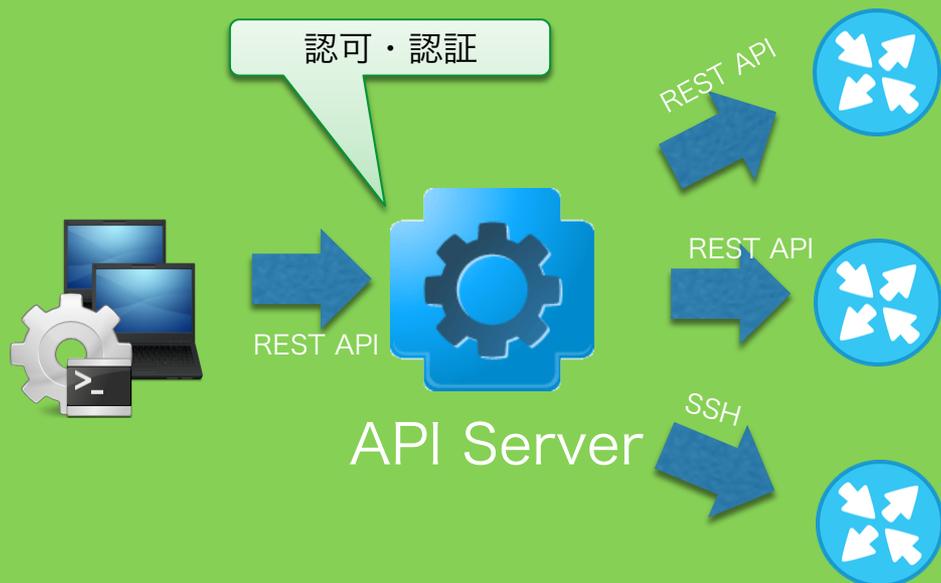
This package provides a unified command line interface to Adtech x Cloud.

### Table of Contents

- Installation
- Bash Completion
- Options
- Subcommand details
  - axc config
  - axc nodelist
  - axc nodecreate
  - axc nodestate
  - axc nodedelete
  - axc poollist
  - axc poolcreate
  - axc poolmember
  - axc poolmodify
  - axc pooldelete
  - axc monitorlist
  - axc vslist
  - axc vscreate
  - axc vsdelete
  - axc jenkins
  - axc chatwork
  - axc slack
  - axc zabbix
  - axc neutron
  - axc nova
  - axc httpd
  - axc vpnuser

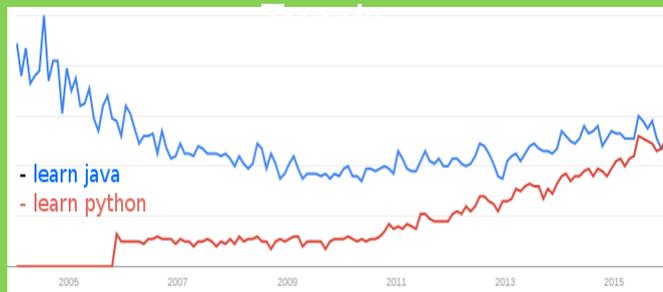


# 構成イメージと使用した技術



- 言語  
Python  
Shell Script
- ソフトウェア  
Flask  
Nginx  
Gunicorn  
その他

"Learn Python" Overtakes "Learn Java" on Google



出展 : <https://dzone.com/articles/learn-python-overtakes-learn-java>

- 構成管理  
Ansible

# その他

- 一般的なWEBサーバと同様にヘルスチェック用のURLを使って分散/冗長
- 操作履歴はAPI Server側で簡易取得
- VLAN変更などはAnsibleとかChefでまとめたい（障壁あり）
- CI的な観点では、Develop用ブランチでつけたTagを監視してAct/Stbをローリングアップデート。
- テストをした上で本番へ組み込む。

The screenshot shows a 'Build Pipeline' interface with three stages: #26 apigateway.repo.develop-sync (1.8s), #24 apigateway.repo.master-merge (4.5s), and #5 apigateway.production.lbdisc (0.53s). Below the pipeline is a table of releases:

Releases	Tags
6 days ago	<b>v0.7.9</b> -> d37f9af zip tar.gz
6 days ago	<b>v0.7.8</b> -> 2b35322 zip tar.gz
8 days ago	<b>v0.7.7</b> -> 389919a zip tar.gz

# Chatも活用した方が良い

komei 11:52 AM

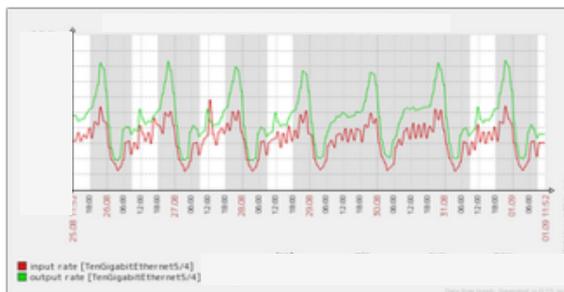
platform\_bot nwgraph

platform\_bot BOT 11:52 AM

Jenkinsのjob(slack.upload.nw.graph)をbuildするよ！

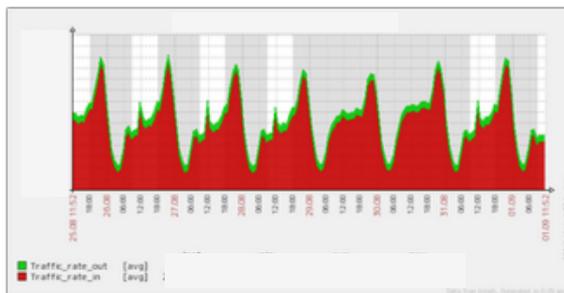
platform-jenkins 11:52 AM

uploaded an image:



platform-jenkins 11:52 AM

uploaded an image:



platform-jenkins 11:52 AM

- グラフィカルな部分はChatで完結できるようにしておく意外と便利。
- “頼まれる側”も “頼む側” に取ってもストレスフリー
- 他の可視化や簡単なデプロイ用途で活用

# REST APIの話

アプリケーションプログラミングインタフェース (API、英: Application Programming Interface) とは、ソフトウェアコンポーネントが互いにやりとりするのに使用するインタフェースの仕様である。

APIには、サブルーチン、データ構造、オブジェクトクラス、変数などの仕様が含まれる。

出展 <https://ja.wikipedia.org/wiki/アプリケーションプログラミングインタフェース>

- REST APIってエラー処理が楽
- ヒューマンエラーさようなら（に近づく）
- ネットワークエンジニア以外でも操作しやすい
- 監視装置は情報の宝庫
- 構成管理は苦手
- 適材適所で使いましょう

# 苦労したポイント

- APIなしの機器を操作する場合、expectで対応せざるを得ない  
→処理が途中で止まるとか・・・  
負担にならない程度のエラー処理を。
- 問題があった時のDebugをどうするか  
→ Debugオプションを実装。Client側で取れるように。  
APIのエラーコードとメッセージをクライアント側へ出力
- REST APIの設計  
→ 素人がやったので汚い。苦労した方が良かったポイント。  
後悔。

# わかったこと

- 権限周りは自分で作ろう  
→ 各機器の機能差を吸収する必要あり
- 商用に頼りすぎてベンダーロックインにならないよう注意  
→ コントローラー乱立もちょっと・・・
- でも自作で俗人化にも注意。エンジニアに一定のレベルを。  
→ チーム開発できる環境を

NWエンジニアの世界は狭いかもしれない。

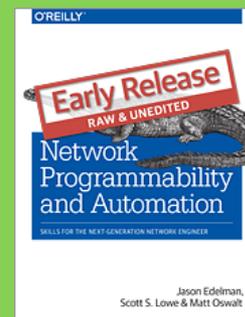
こうやって学んだことは色々応用できるので  
スキルアップの観点でもGood!

# おわりに

- REST APIの設計はメーカーにより設計の質が異なる
- REST APIもAnsibleなどの構成管理も適材適所
- SDN / NFVは別のモチベーションがあってアプローチ。Overlay !

## 5分の手作業より15分でコードを書きましょう

Automation is the new skillset that network engineers need to pick up. Much like sysadmins have had to learn how to use new tools like Chef and Puppet, network engineers are learning that they just can't do things manually anymore. With examples in each chapter, this practical book provides you with baseline skills in network programmability and automation, using a range of technologies including Linux, Python, JSON, and XML. No previous knowledge of software development, programming, automation, or DevOps is required.



# アドテクに必要なSDN/NFV (後編)

OpenStack Liberty + Midonet 5.0.0

@SDN Japan 2016/1/15

サイバーエージェント  
アドテク本部



長谷川 誠

# 自己紹介



Makoto Hasegawa

アドテク本部にて  
データセンター運用や  
Private Cloud(OpenStack)を  
構築・運用したりしてます

Twitter : @makocchi

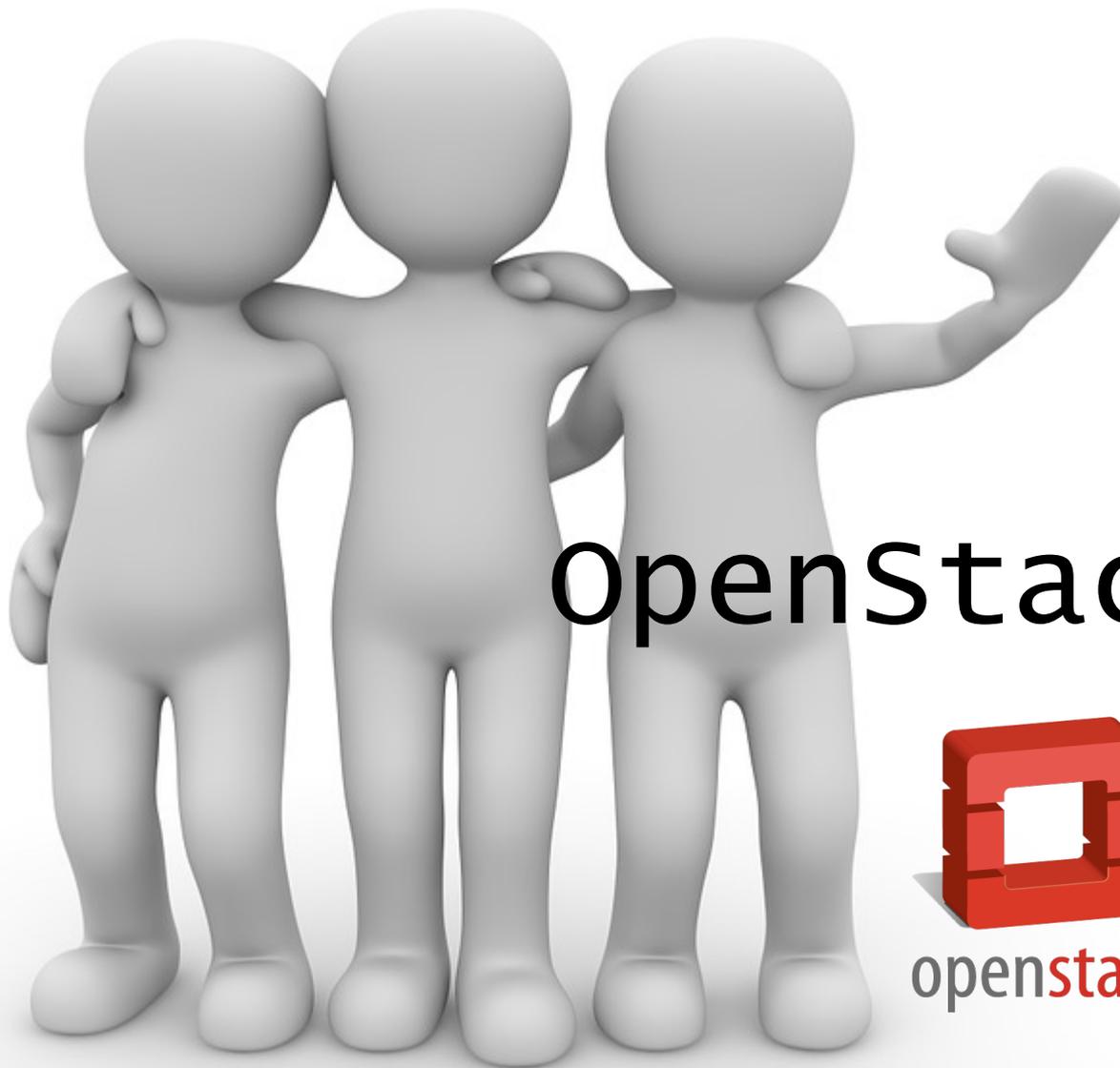


OpenStack  
Professional



# Agenda

1. OpenStack とは
2. Midonet とは
3. OpenStack + Midonet
4. 導入にあたり
5. 今後について

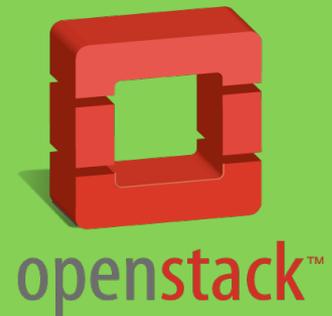


# OpenStackとは



# OpenStack とは

汎用的なPCサーバやネットワーク機器などの標準的なハードウェアを用いて IaaS (Infrastructure as a Service) 型のクラウドコンピューティング環境を構築可能なOSS



簡単に言うと・・・

プライベートクラウドに構築できる AWS のようなもの  
→ もともと AWS を参考に開発が始まった



# Midonetとは



# Midonetとは



Midokura 社(<http://www.midokura.com/>)により開発されている  
L2 から L4 をカバーするネットワーク仮想化ソフトウェア

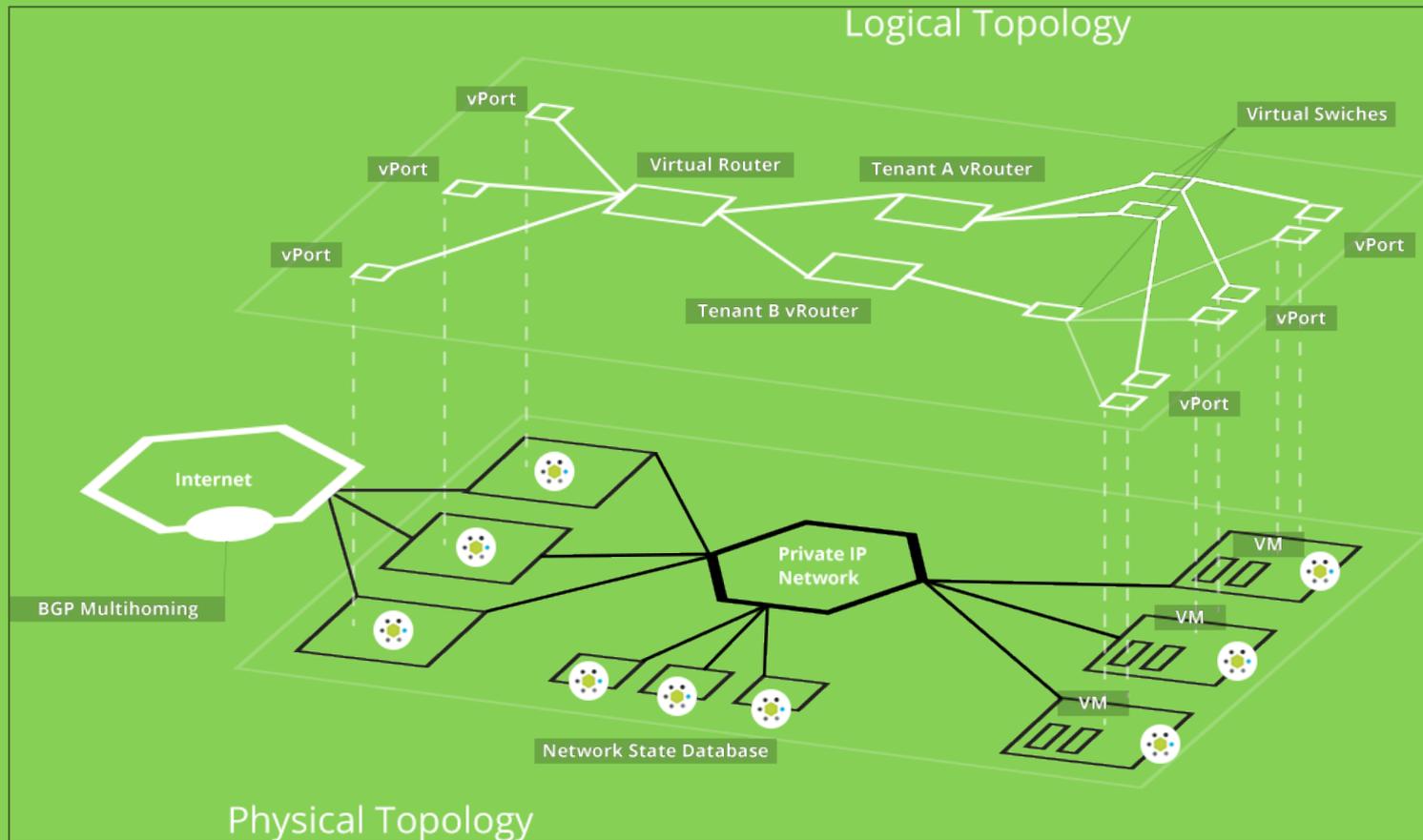
OpenStack のネットワーク機能 Neutron の  
Plugin として稼働させることができる

SPOF の無い分散アーキテクチャを採用

日本法人(ミドクラジャパン)があり日本語のサポートがあることが  
地味にうれしい

2014年11月にオープンソース化 (Community Edition)

# MidonetのOverlay



<https://www.midonet.org/> より引用



# OpenStack + Midonet

# OpenStack + Midonet

従来の運用は OpenStack のネットワークの構成(Neutron) はネットワーク管理者によって決められていた

つまりこういうこと

- ・ この Tenant(project) はこの vlan 使ってね
- ・ 新しい Tenant 増えたから vlan 増やさなきゃ・・・
- ・ この vlan どこの Tenant が使ってたっけ？
- ・ あらかじめ用意していた vlan がいつの間にか枯渇していた・・・
- ・ 新規プロジェクト「vlan 50個下さい」「え・・・」

# OpenStack + Midonet

OpenStack 管理者は Tenant を作成したりするタイミングで  
Neutron 経由で Network を作成し、Tenant に紐付ける必要がある

ネットワーク管理者は vlan を払い出したり新規作成したりする

そんなめんどくさい運用から開放されたい

そこで・・・

ネットワークを仮想化することで解決できるのではないか？

と思い立ちいろいろ SDN を調べていた

# OpenStack + Midonet

## Midonet との出会い

とある OpenStack のセミナーにて Midonet の存在を知ること

- ・ SPOFが無かったりスケールしやすそうだし、良さそう
- ・ しかも日本のエンジニアがいる
- ・ しかし当時はまだオープンソースではなかった為導入には至らず

しかし 2014年11月に・・・

# ミドクラ、OpenStack向けネットワーク仮想化ソフトウェアをオープンソース化

November 3, 2014

“MidoNet”オープンソースプロジェクト、パートナーの支持を受けてスタート

従来の商用版の機能全てを本日からオープンソースに  
コミュニティサイトも立ち上げ <http://www.midonet.org/>

<http://www.midokura.jp/press-releases/midokura-open-sources-complete-iaas-network-virtualization-solution-openstack-community/>



## この流れに乗るしかない

# OpenStack + Midonet

オープンソースが発表されてから様々な検討を重ね・・・

この度社内のエンジニア用の開発環境用の OpenStack において  
Midonet を採用決定、構築開始

Neutron の Plugin に Midonet を採用することで OpenStack 利用者は自由にネットワーク構成を構築することが可能に  
(例えば開発環境でも本番環境と同じ ip アドレスを使うことが可能)





導入にあたり . .

# 導入にあたり . . .

## 苦労したポイント

今回チャレンジしたのは OpenStack Liberty (最新) と Midonet 5.0.0 (最新) という 超 cutting edge

- ・正直動くかどうか分からない

当然 Document 等は揃っていない

- ・ Midonet の Document は Kilo + 5.0.0 しかまだない
- ・ OpenStack の Document は信用できない
- ・そこは過去のノウハウでなんとかカバー



今後について

# 今後について

OpenStack + Midonet の環境は開発環境と相性がよい

- ・ 利用者側が自由にネットワークを設計できる

現在アドテク本部では個人用の開発環境 OpenStack  
として活躍中

運用上のノウハウを貯めつつ実際のサービスを行う環境へ  
持っていくことが目標

- ・ VXLAN の offload 等考慮しなければならないこともある
- ・ モニタリングをどうするか

# 今後について

L3 DSR できるようになったらいいな

- ・ 現状はいろいろ手をいれないと実現することができない

OpenStack と コンテナ技術との親和性に期待

- ・ Midokura 社が積極的に関わっている Kuryr
- ・ flannel や libnetwork と Neutron を連携させることで OpenStack とコンテナ技術の融合の架け橋に

# アドテクに必要なSDN/NFV

@SDN Japan 2016/1/15

ご清聴ありがとうございました