

クラウド事業者側の 仮想環境ネットワーク運用の 自動化について

GMOインターネット株式会社
オープンネットワークキングチーム

村上 勉

tsutomu-murakami@gmo.jp

自己紹介



名前：村上 勉

所属：GMOインターネット

仕事：ネットワーク？サーバ？ハードウェア？

基本なんでも屋さん。隙間産業

趣味：料理とお酒、あとゲームとか。

最近はウォーキング

経歴的なもの：

プロバイダとかSIとかNetworkとかサーバとか

前職は北海道にデータセンターもってる会社さん

で約6年半ほど色々やってきました

(サーバ、ネットワーク、OS、ミドルウェア、
運用モロモロ)

事業者の悩み

- 設定などの日々のオペレーションの数々
→ スイッチやアプライアンス機器設定などの自動化が必要
 - 不正通信などの検知・対応
→ どう検知する？そしてどう対応する？
 - 大量通信ユーザに対する対応
→ どう検知する？そしてどう対応する？
 - 営業や顧客などの要求に対する機能拡張
→ 無い機能はどうする？作る？買う？
- その他ネットワーク以外の部分を追加すると膨大な量・・・
(バグ対応のコード修正・検証・テスト等・・・)

自動化しないとやってられない！

設定の自動化・運用の自動化の方針

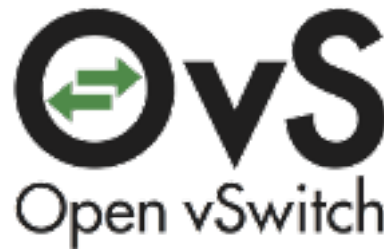
- ・ 現状Openstackを利用している為Openstackベースで考える。
- ・ 但しOpenstackで出来るネットワークの自動化も限られるし、Bugも多い。
- ・ Openstackのバージョンによっては使いたい機能も使えない
その場合は新規機能開発でなんとか回避する
- ・ ユーザーサイドで設定出来るところはOpenstack準拠
(API公開とかもあるので)
- ・ システム側で完結出来るところはOpenstackにこだわらない



ケース1：帯域制限とポートブロック対応

- ・ Openstack neutron + openvswitchで実装
- ・ QoSの仕組みを変更してrate limitとポートブロックにも対応した仕組みをつくってみました。
- ・ 実際はOpenstack側からOVSに追加でflowを入れているだけ
Flowを入れる仕組み自体は標準で備わっているのを拡張しました。

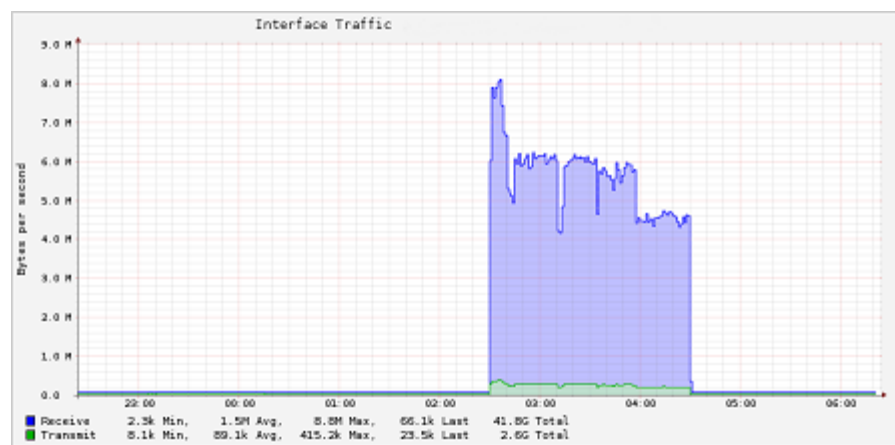
余り重い処理にしないでポートの追加時や作成時に追加の処理をいれるだけにしています。



ケース2：高トラフィックユーザーに対する制御

- ・ Openstack ceilometer + ケース2の帯域制御の組み合わせ
- ・ ceilometer側で利用トラフィックを取得出来る為
1日数十GBとかの高トラフィックを吐き出すユーザーの制限は出来る。Ceilometerから情報を取得してオーバーしてる場合に制限
- ・ ただ仕組みとしてはいれてあるが現状は一部のユーザーにしか上記を適用していない。制限されると以下の感じ。24Hで解除

※openstack ceilometerとはメータリング機能となります。トラフィック量もメータリング出来ます。



ケース3：特定アプリケーションのブロック

- ・ flow collectorで情報収集を行い。問題のある通信をブロック
- ・ openvswitchからflowをcollectorに飛ばして情報収集
- ・ VMから問題がある通信が発生。
- ・ collector側から現在の状態の確認を行う
- ・ 閾値を越えてるものがあれば内部APIで制御を行う。
- ・ 主に大量のメール配信とかのSPAM対策の為に対応出来るようにしました。（自動化しているのはここの部分のみ）

ケース4：その他モロモロ

- スイッチなどの設定自動ツール（Cisco Nexus）
- Openstackの自社作成LBaaSドライバ(Brocade ADX)
- Openstackの自社作成LBaaSドライバ(LVS-DSR)

とかとか・・・

大体独自の仕様に併せた形で設定したい為
メーカー標準を利用してません。
カスタマイズって大事です。

～まとめ～

- 既存の有りもの機能を利用してつくってます。
- 足りない部分は拡張してカバー
- パブリック・クラウドは色々な人が使うので制限系の仕組みは多いほうが良い。
- 少人数での開発や運用の為、API化して運用ツール化すると楽
- 内部API化すると自動化しやすくなる
- 運用の自動化は情報の取得と設定のコントロールが肝心
- 設定される側にやっぱりAPIあるとすご〜く楽

やっぱり楽したいですね！

～最後に～

今回の内容の詳細につきましては、今後勉強会とか開きつつご説明出来ればと。

ご清聴ありがとうございました！