



## パネルディスカッション

# 運用自動化で実現しなければならないこと、 そのために必要なもの

2017/7/21

5-A3-10 DevOps/自動化セッション  
OpenStack Days Tokyo 2017

後日メモ(ディスカッション後に当日の会話をまとめたものです)

# DevOps/自動化セッションについて

- ❑ DevOps/自動化セッションでは、最新の自動化技術の基礎からDevOpsへの応用、そして実践までを取り上げ、運用にまつわるオペレーションの課題と解決について掘り下げていきます。
  
- ❑ プログラム
  - ❑ Day1
    - ❑ [4-B1-2] 進化を続ける運用自動化ツール
    - ❑ [4-B1-3] CI/CDパイプラインを支えるツールと組織
    - ❑ [4-B1-4] 実環境での運用自動化とその管理方法
    - ❑ [5-A1-5] Googleのインフラ技術に見る基盤標準化と DevOpsの真実
  - ❑ Day2
    - ❑ [5-A3-10] パネル:運用自動化で実現しなければならないこと、そのために必要なもの



# 目次

1. 登壇者紹介
2. テーマ
  - a. 自動化の成功談、失敗談
  - b. 今の手順をそのまま自動するのはありかなしか？
  - c. 自動化するとスキルは落ちるのか？
3. まとめ



# 登壇者

- パネリスト
  - 真壁 徹／日本マイクロソフト株式会社 @tmak\_tw
  - 曾我部 崇／ヤフー株式会社 @rev4t
  
- モデレーター
  - 中島 倫明／レッドハット株式会社 @irix\_jp



# パネリスト自己紹介 (1-1)

- ❑ 真壁 徹(Toru Makabe)
- ❑ 今の仕事
  - ❑ Azureのソリューションアーキテクト
  - ❑ クラウド活用に関する技術支援
- ❑ 経歴
  - ❑ シンクタンク系Slerでパッケージサービス(今でいうSaaS)の開発と運用
  - ❑ ハードウェアに強いITメーカーでアプリ開発からインフラのプリセールスまで
  - ❑ OpenStackやってみました！

# 自動化は必要か? (真壁)

- ❑ 必要
  - ❑ そもそも人間がめんどくさいと思っていることをやるのがコンピュータの仕事
  - ❑ ただし条件付き
    - ❑ 恩恵を受ける人が主導する
    - ❑ ビジネス的な妥当性を説明できる
    - ❑ 自動化「だけ」では予算はつかない



# パネリスト自己紹介 (2-1)

- ❑ 曾我部 崇(Takashi Sogabe)
- ❑ 今の仕事
  - ❑ 法人向けクラウド基盤の開発・運用 (主務)
  - ❑ 社内向けクラウド基盤の開発(兼務)
- ❑ 経歴
  - ❑ パブリッククラウドにて クラウドサポートエンジニア
  - ❑ ISPにて ネットワーク機器・アプリ開発等



## パネリスト自己紹介 (2-2)

- ❑ 法人向けクラウド基盤
  - ❑ 「素の」OpenStack
    - ❑ Floating-IP, DHCP, metadata service, DVR, VXLAN overlay, etc.
  - ❑ Juju / MAAS を用いたデプロイの自動化
  - ❑ Ansible を用いた運用作業の自動化
    - ❑ Juju だけで、全ての運用を自動化できるわけではない
      - ❑ Juju / MAAS 自体の backup ワークフロー等
    - ❑ Ansible の Juju dynamic inventory 対応をしたら、連携しやすくなった





# 自動化は必要か? (曾我部1)

- ❑ 必要
  - ❑ ただし、費用対効果の高いものを優先する
  - ❑ 繰り返し実施しない作業は、必ずしも自動化は必要ない

# 自動化は必要か? (曾我部 2)

- ❑ 何のためにするのか?
  - ❑ お客様、エンドユーザにお届けできる価値
    - ❑ 早さ(Agility)
    - ❑ 品質
    - ❑ コスト
  - ❑ 自動化は課題を解決するための手段
  - ❑ 解決したい課題が見える化(メトリック化)すれば、目標を設定しやすい
  - ❑ コストが課題の場合は、自動化の前に取り組むべきことがあるかもしれない



テーマ1

# 自動化の成功談、失敗談

# 自動化に失敗した経験 (真壁)

- ❑ 恐怖！コンフィグ汚れ
  - ❑ 自動化して作ったリソースの設定を直接変更される
    - ❑ VMにSSHで入ってパッケージ入れちゃう、とか
  - ❑ その内容は自動化ツール側ではわからない
  - ❑ 再現不能
  - ❑ 「じゃあ今後はVMのクローン機能で」
    - ❑ 地獄の始まり
    - ❑ 誰もあるべき姿を定義、把握できない

# 自動化に失敗した経験 (真壁)

- ❑ 受益者不在問題
  - ❑ 自動化してうれしい人が現場にいない
  - ❑ 自動化して評価される人が現場にいない
  - ❑ そりゃ続かない
    - ❑ VMにSSHしてやっちゃえ！ (繰り返し...)
  - ❑ うまくいってるケースは
    - ❑ 小さな成功体験があり、かつ評価される人が主体的に、熱く進める



# 自動化に成功した経験 (真壁)

- ❑ 基盤の持つ標準機能を使う
  - ❑ Azureは標準機能、無償枠ででかなりのことができる (デプロイやアラートからの自動アクションなど)
  - ❑ OpenStack Heatとか、最近どうですか
  - ❑ 有償ツールだとツール選定の議論に終始しがち
  - ❑ 手を動かす意思のあるお客様にはテンプレートをお渡しします



# 自動化に失敗した経験 (曾我部)

- ❑ 中途半端に自動化すると、事故になりやすい
  - ❑ 事例1: Chef cookbook の server へのデプロイを手動で実施していた場合に、誤って古いバージョンのデプロイをした
  - ❑ 事例2: バージョン管理をしていない デプロイscript を用いていたところ、誤って古いバージョンのデプロイをした



# 自動化に成功した経験 (曾我部)

- ❑ cron による自動化はお手軽だけれど強力
  - ❑ 殆どの Linux /Unix系OSで標準提供されているので、すぐに始められる
  - ❑ 通知の仕掛け(Email等)をキッチリ作っておけば、安心して運用できる





# QA

ボトムアップが有効(小さな成功体験)というのが二人の共通意見だが、自動化は大きな範囲で適用したほうが効果が高いはず。そのためにはボトムアップよりはトップダウンの方が有効ではないか？

→

もちろんトップダウンが必要だが、いきなりのトップダウンは失敗し易い(受益者の理解不足やモチベーション)。まずは実際の受益者が小さな成功を積み重ねた上で、その成果を持ってトップとの協力の元に展開するのが良い。



## テーマ2

今の手順をそのまま自動するのは  
ありかなしかな？

# 今の手順をそのまま自動化するのはありか？ (真壁)

- ❑ そもそも「今の手順」が整理、合意されているのか、という疑問
  - ❑ 「手順」を利用者がリクエストしてから結果を得るまでのプロセスとすると
  - ❑ 多くは断片的、組織横断な作業の組み合わせで、一気通貫した手順になっていないのでは？
  - ❑ 全ての作業の意味、妥当性を一貫して説明できる人がいるか？
- ❑ 整理されており、かつ妥当性があれば、あり
- ❑ 整理されていなければ、ツールのリファレンスやベストプラクティスに合わせたほうが楽

# 今の手順をそのまま自動化するのはありか？ (曾我部1)

- ❑ システム次第
- ❑ シンプルなシステム(不具合の影響範囲が小さい)
  - ❑ 自動化することで生産性の向上が期待できる
  - ❑ 自動化の効果が得られそうな箇所を優先して取り組むのが良い
- ❑ 複雑なシステム(モノリシック、不具合の影響範囲が大きい)
  - ❑ サーバ構成やリリースするパッケージ等を小さい単位にまとめて、不具合の影響範囲を小さくすることが先決
  - ❑ システム構成を変更せずに自動化を進めると、大事故を引き起こしやすい



# 今の手順をそのまま自動化するのはありますか？ (曾我部2)

- ❑ アラート対応の自動化
  - ❑ 自動的に修復できるようになれば、オンコール対応に要する時間を削減できる
  - ❑ 不具合切り分け等が不要かつ繰り返し発生する作業は手作業の自動化を検討したい
    - ❑ nova-compute 故障による evacuation 等



## テーマ3

自動化するとスキルは落ちるのか？

# 自動化するとスキルは落ちるのか？ (真壁)

- ❑ 落ちない (一`d一`)キリッ
- ❑ 抽象化によるブラックボックス化が質問の背景だと想像
- ❑ 実際に手を動かさせばわかる
  - ❑ リソースをAPIを通じて操作するには、リソースのデータモデルや属性、リソース間の依存関係を理解する必要がある
  - ❑ 分かってない状態で動かしても期待した状態にならない
    - ❑ 期待やあるべき状態の合意すらない、というのは論外
- ❑ 少数精鋭化が進む可能性はある

# 自動化するとスキルは落ちるのか? (曾我部)

- ❑ スキルの定義次第だが、エンジニアのスキルは上がるはず
- ❑ 自動化を進める結果、起こること
  - ❑ 開発エンジニア: インフラ(運用環境)への理解が深まる
  - ❑ インフラエンジニア: 実装への理解が深まる
  - ❑ 単純作業から開放されるので、もっと付加価値の高いことにチャレンジできるようになる
- ❑ 付加価値の高い仕事の例
  - ❑ 障害原因の解析および対策・改善案の検討
  - ❑ 障害を未然に防ぐ仕掛けの実装
  - ❑ ビジネスに役立つメトリックをシステムから抽出





## まとめ

自動化で成し遂げないといけないことは？

そのために必要なものとは？



ご参加ありがとうございました