

AWSを使った CI&IaC
サービスの高速改善を行うためのクラウド構築事例

株式会社リクルートコミュニケーションズ



- リクルートコミュニケーションズのビジネス概要
- 第一部：AWSを利用したインフラ×アプリの連携事例
 - アドテク製品を世に送り出すまでの歩みとAWS 活用法
 - 事例紹介：OpsWorks中心のインフラ x アプリ
 - Blue-Green Deployment
 - ゾーンまるごと交換手法
- 第二部：インフラ×アプリの連携による
エンジニア協業体制の変化

株式会社リクルートコミュニケーションズ
ICTソリューション局
アドテクノロジーサービス開発部
クラウドアーキテクト

小林 智則 (こばやし ともり)

- 21世紀はじめ 元々はプログラマで C/C++ でシステム系プログラミング
- サーバエンジニアとして2007年くらいに今で言うプライベートクラウド的なものを構築(当時でも200近いVM)
- 2011年～ インフラエンジニアとして RCOへ、AWS を使い始める

リクルートコミュニケーションズの ビジネス概要



リクルートコミュニケーションズは、リクルートグループのクライアントの集客ソリューションから、制作・流通・宣伝、カスタマーサポート、デジタルマーケティングのサポートを担当。



サポート

リクルートコミュニケーションズ

メディア
制作

流通・宣伝

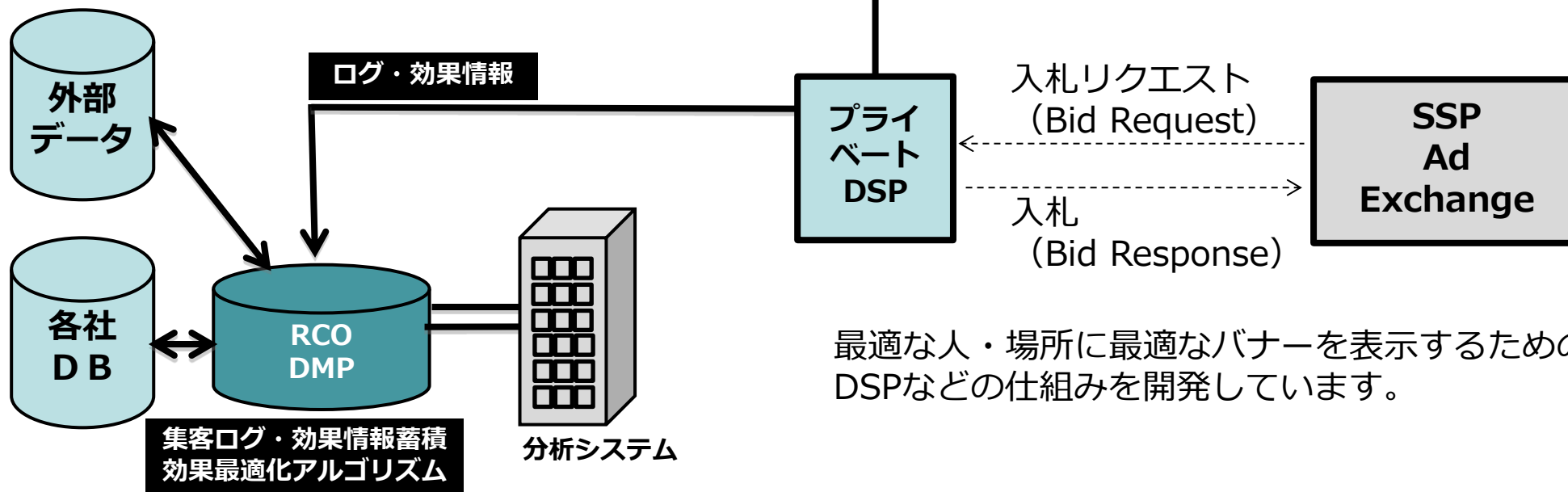
カスタマー
サポート

デジタル
マーケティング

- 独自アルゴリズム
- 高度な自動化
- 新規性

▶効果の最大化と競争優位性を実現

※ネット集客の世界はコストをかけずに効果（アクション）を最大化する世界

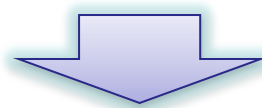


最適な人・場所に最適なバナーを表示するためのDSPなどの仕組みを開発しています。


アドテク製品を世に送り出すまでの歩み

 **2013年 4月** : アドテック製品の開発開始


- 同時に継続的デリバリーな環境構築、体制での開発 (DevOps)
- AWS のみでアドテックをやれるのでは? と挑戦



 **2013年 6月** : OpsWorks をインフラxアプリの中心に

 **2013年 7月**：最初のアドテック製品リリース
- インフラ構築もアプリのデプロイも全部 OpsWorks



 **~2015年**：8つ程度のアドテック製品をリリース
- やはり OpsWorks が中心に

アドテク製品を世に送り出すまでの歩み

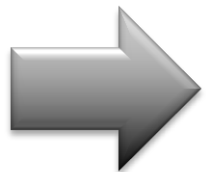
～意識して変えてきたこと～

- 提供するサービスの開発において、エンジニアが積極的に仮説検証していく
 - ニーズに応じた機能追加などはエンジニアの方が理解が高い
- しかもその課程を高速なPDCAサイクルで実現できれば、市場優位性や、継続的改善により、**継続的なイノベーション**を起こせるはず

• サービスを高速に改善できる PDCAサイクルの実現

- 継続的なサービス開発
- 高速リリースによるサービス改善

- ビジネスプランの迅速な検証
- 高速リリースによる継続的なフィードバック
- 市場の変化に対する競争優位性の確保/維持



継続的デリバリ可能な環境の実現

インフラ x アプリで

クラウドファーストを軸に

AWS のフル活用



RCO の

アドテク



アドテクを支える基盤としてのクラウド



継続的デリバリ


インフラコスト
自由度
可用性, etc...

Chefなどの構成管理ツール →
Jenkins で CI してみる →



← AWSの機能を使いこなす
← より高度な機能の実装

AWS には魅力的なサービスがいっぱい
ちょうど2013年に  OpsWorks もリリース

-  OpsWorks
 - 自前で Chef環境などを用意しなくてもいい
- Infrastructure の Code化がより簡単に
 - VPC設定やその他の設定には CloudFormation やそれをラッピングするツールも登場 (codenize.toolsなど)
- Serverspec などのベストプラクティス的なテストツールの登場



インフラの自動構築、テストに
アプリ的手法が取り入れ可能に

アドテクサービス開発を AWSで

継続的デリバ리를 OpsWorks が**さらに支える**イメージへ



• OpsWorks を活用

AWS

- アプリデプロイと Infrastructure as Code の一体化
- CloudWatch/NewRelic などの便利サービス
- コスト見える化

• DevOps (アプリ x インフラ)

- アプリ開発とインフラ構築・運用の垣根を無くす
- 組織のフラット化
- エンジニアの役割として、インフラ寄りかアプリ寄りかぐらいの概念

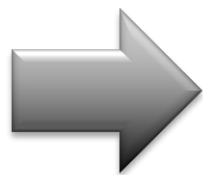
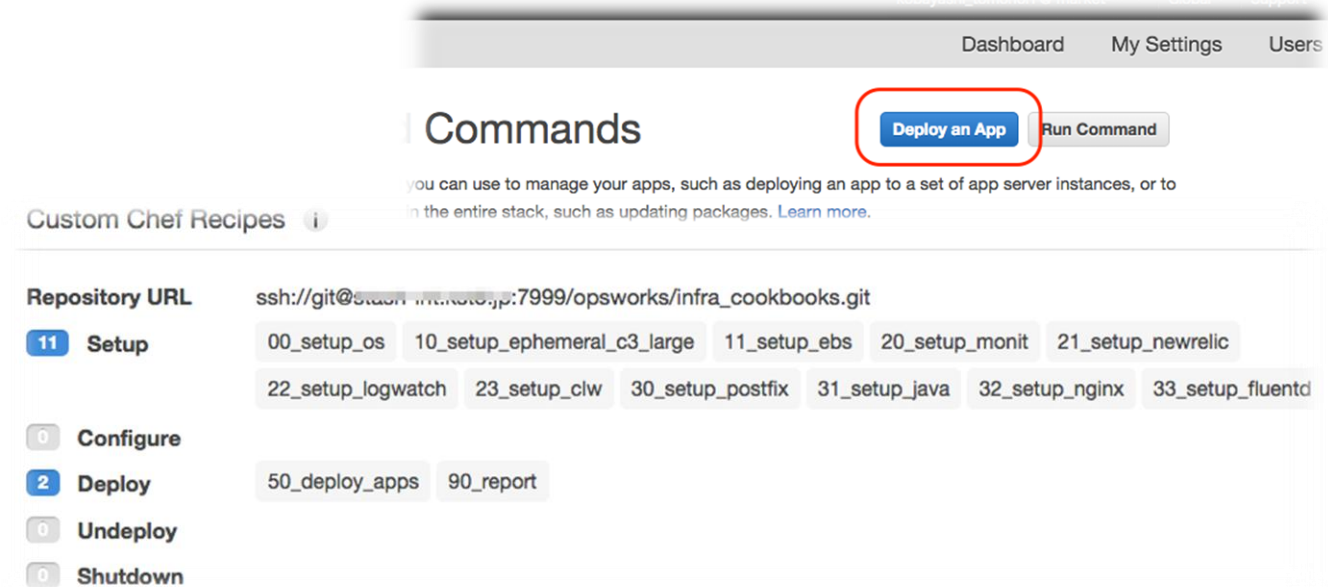


高頻度での
サービス改善

• 継続的デリバリ

- インフラもアプリも CI
- Infrastructure as Code を単なる自動化と考えない

- インフラ構築、その後の運用と アプリデプロイを全部 OpsWorks で実現できる

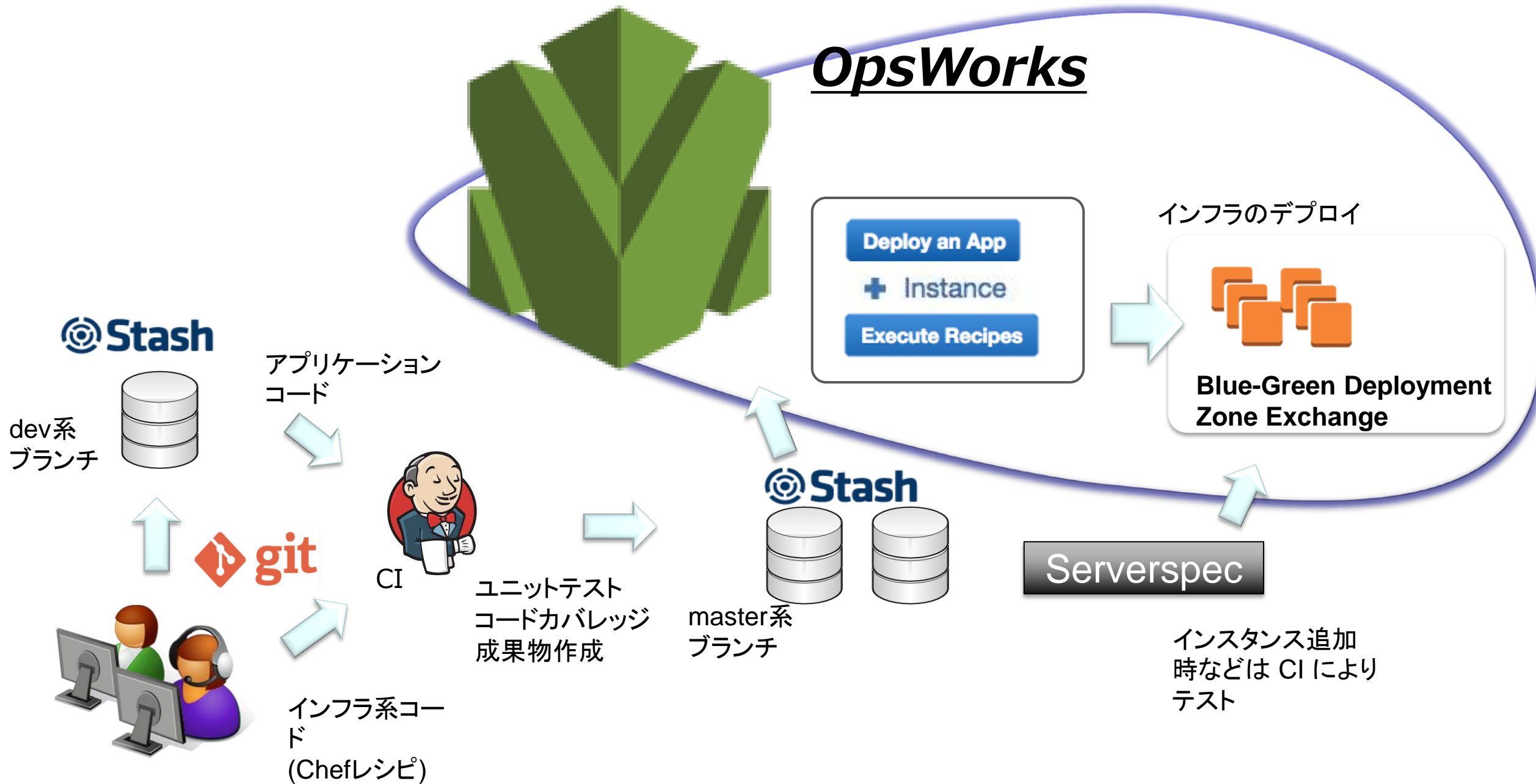


ベースは AWSサービスで、
その上にもっと高度な AWS利用法を構築

今回ご紹介する事例

事例紹介





• **誰でもデプロイできる**

- OpsWorks の Deploy an App ボタン
- ChatOps による hubot への指示

• **安全かつ高速なデプロイ**

- Git (Stash)
- Blue-Green Deployment

安全かつ高速な
デリバリー実現への取り組み

• **可用性対策**

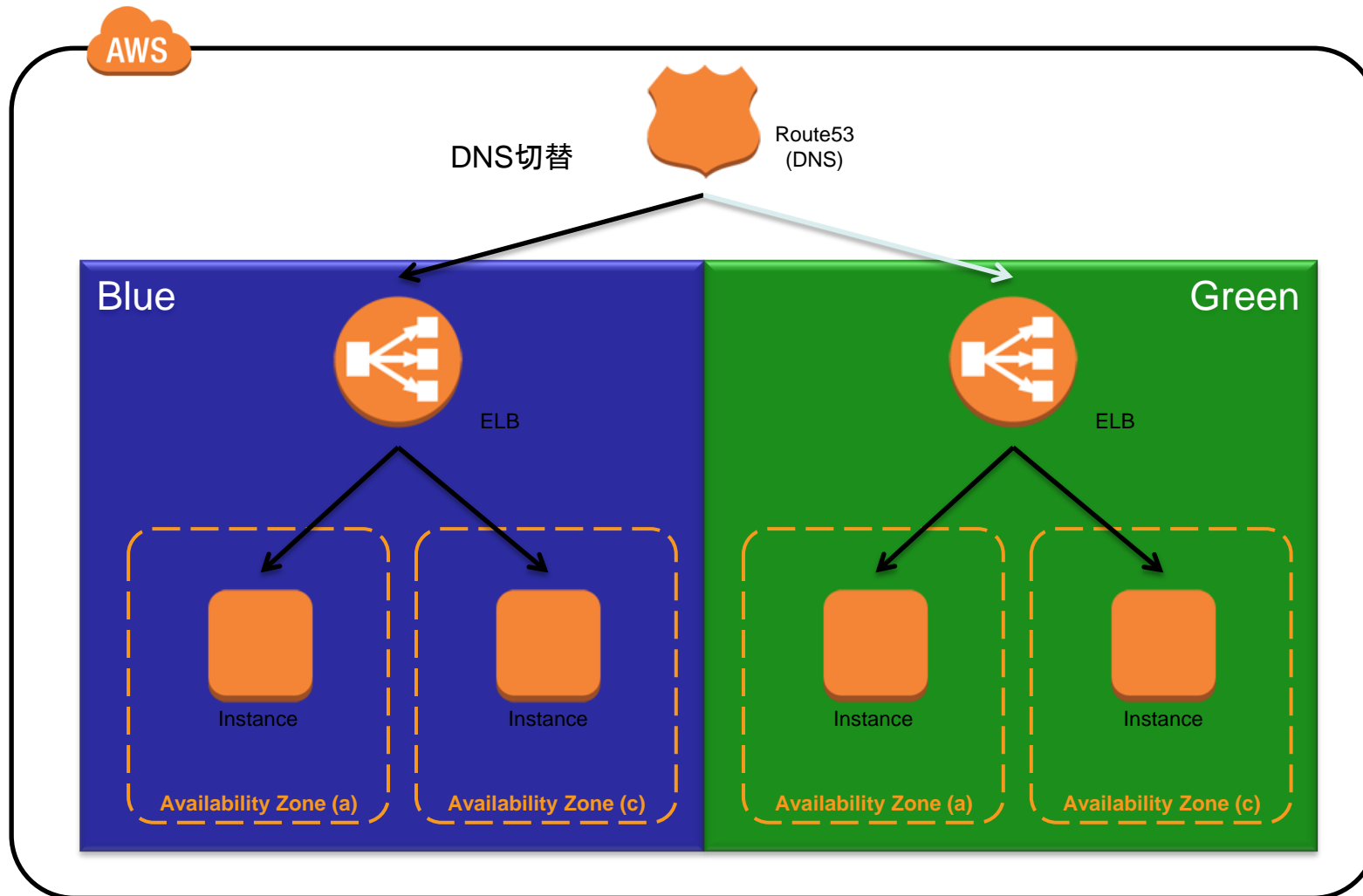
- AutoScaling や Multi-AZ の積極利用
- Availability Zone のまるごと交換手法

• **安全なデリバリ**

- アプリCI
- Serverspec、インフラCI

Blue-Green Deployment





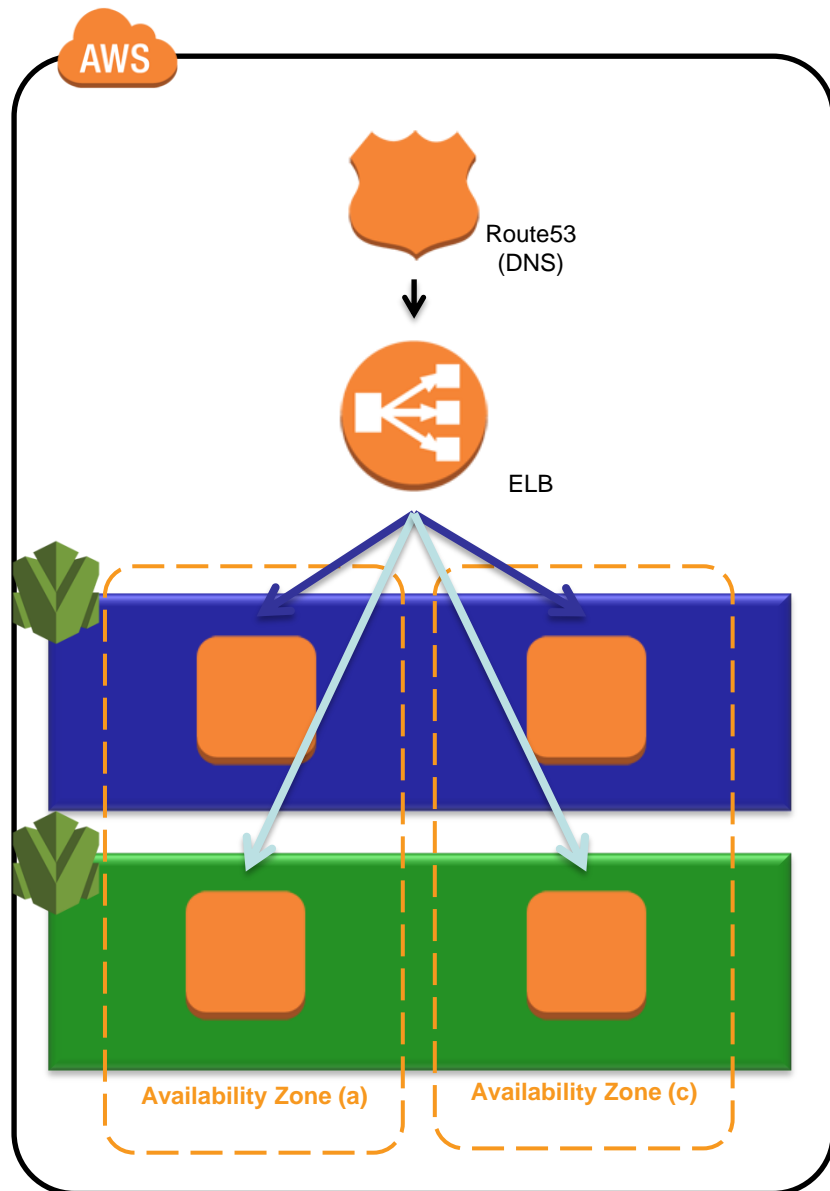
- AWSであれば割と構築は簡単
 - 手作業
 - CloudFormation
 - Ansible

最近ですと、Docker などのコンテナを使って Blue-Green Deployment する方もいらっしゃると思います。

- DNS の切替に時間がかかって、クライアントからのアクセスが一斉に切り替わらない。。。
- いつ、クライアントからのアクセスが切り替わるか制御できない

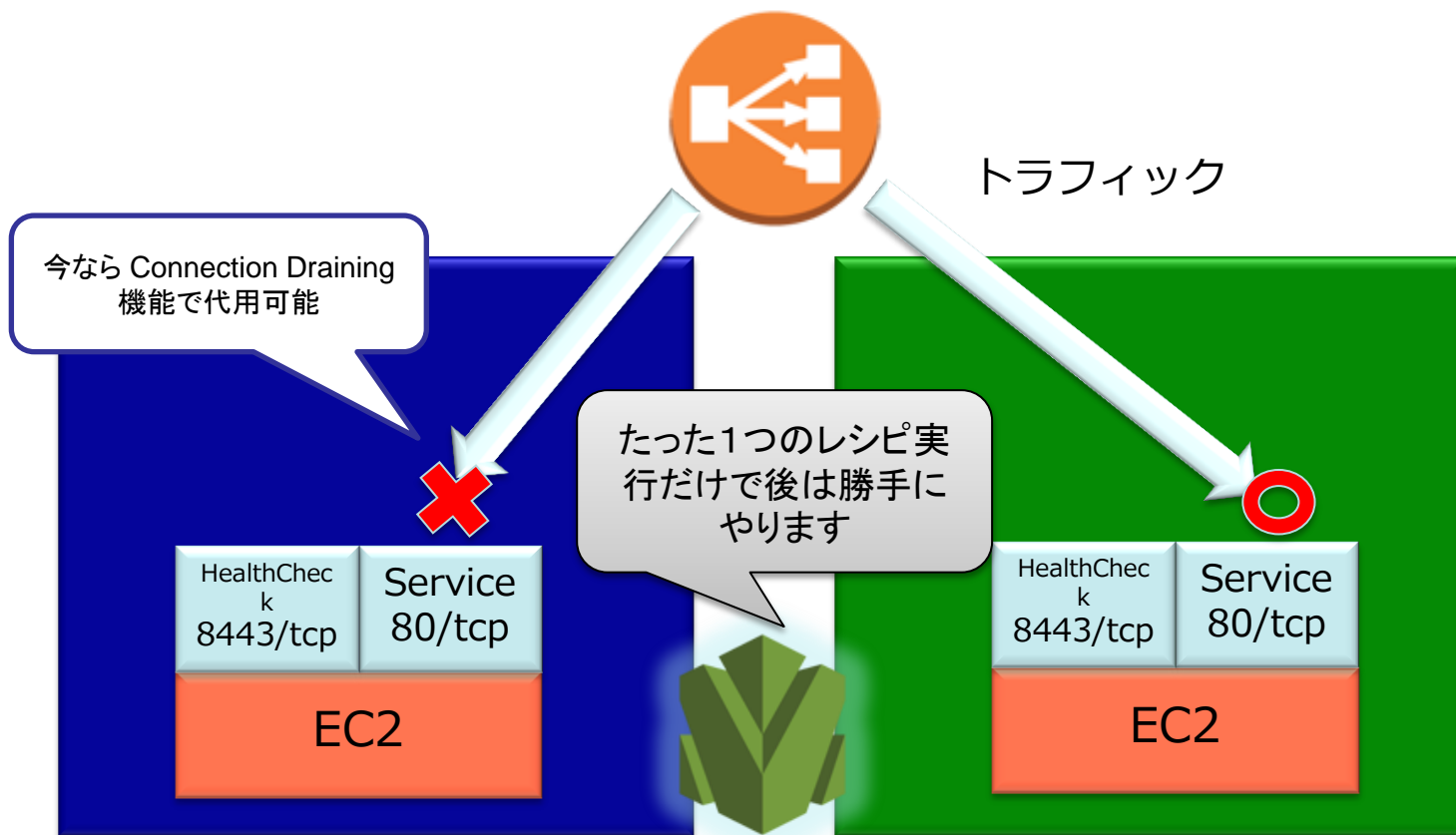
OpsWorks をベースに開発した Blue-Green Deployment

～ELB はそのまま～



- ポイント
 - ELB は同じ (DNS書き換え不要)
 - 配下の EC2を瞬時に Blue側からGreen側に切り替える
- メリット
 - ELB を2台用意して、DNS を変更する方式に対して、瞬間で切り替わるので、サービスダウンタイムがない
- 自動化
 - OpsWorks によりインフラもコード化されているので、アプリケーションのデプロイを含めて、命令ひとつで実行可能

- EC2上のサービス提供用ポートは 80/tcp
- ELB からの HealCheck 用のポートは 8443/tcp
 - アプリケーションヘリバーブプロキシ（アプリの動作チェック用パスがあり、動作OKで 200 OK を返す仕様とする）
 - ELB から見てその EC2が使用可能かどうかの判定には 8443/tcp で 200 となること



1. ELB に Green側 EC2を全部追加
2. 全てのEC2が InService になるのを API経由で監視
3. Blue側の 80/tcp を iptables によって SYN パケットのみ接続不可にする (OpsWorks API によりレシピ実行)
 - ELB の配下においてもトラフィックが流れない
4. 少し待ってから Blue側の EC2 を ELB から抜いていく
5. 一応 iptables をフラッシュ

- ✓ ELB は変わらないので、Blue側、Green側毎に暖気運転の申請不要
- ✓ ELB は変わらないので DNS に起因する問題が起きない
- ✓ ユーザアクセスは必ず切り替わるので、切り替わった側をいつまでも維持する必要はない
- ✓ 全部コード化されている

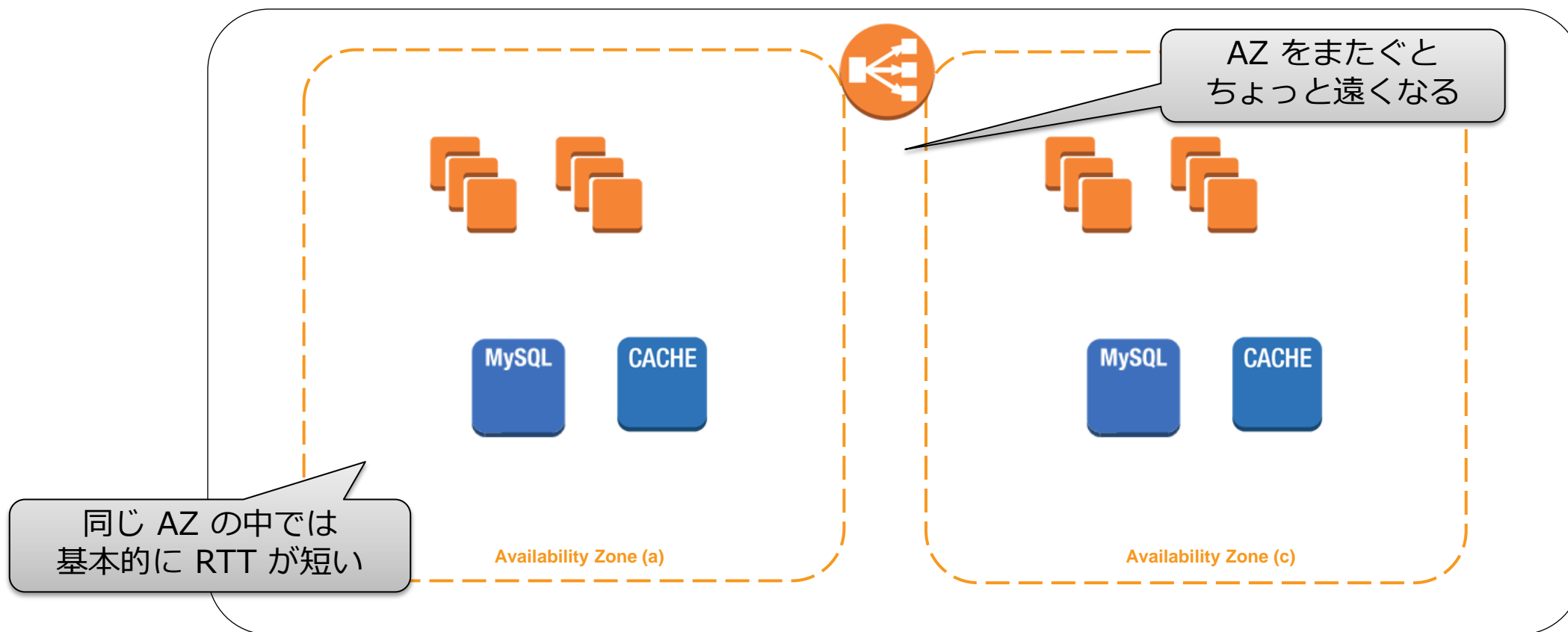
Zone Exchange

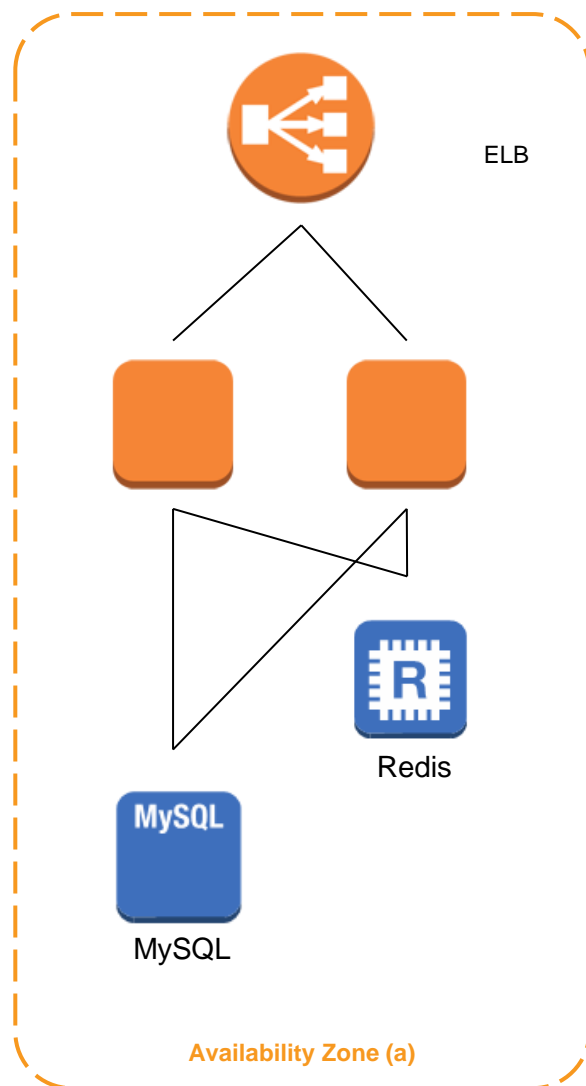
～ゾーンまるごと障害に備えて～

Multi-AZ 配置にはしない
でも、枕は高くして寝たい

そんな悩みはインフラエンジニア
だけじゃないはず

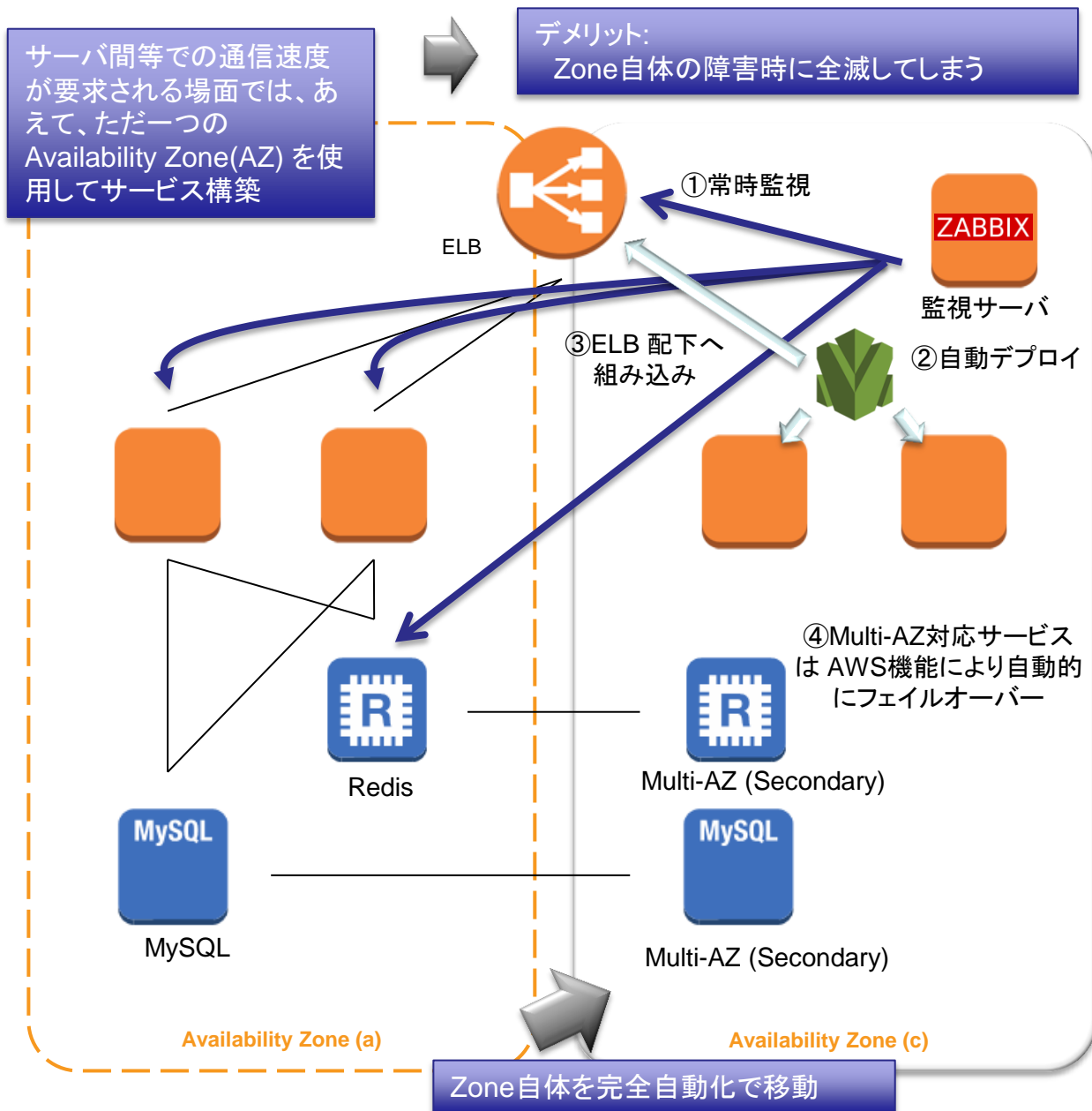
- 同一のリージョン内において、2つ以上のAvailability Zone (AZ)に EC2 や RDS などの AWSサービスを配置すること。





- AZ が異なると確実に 2ms くらいは RTT が増える
 - 単一ゾーンでシステム構築
- RTT が重要なサービス
 - DSP (Real-time bidding)
 - 各サーバ間のデータ同期
- ゾーン丸ごと障害時に全滅する . . .

ゾーン交換手法 (ゾーン障害時の完全自動切替)



- ① サービス提供Zone側(a)の EC2 や Redis、その他サーバを常時監視
- ② 監視サーバZABBIX により、(a)が全体障害を起こしていると判断された場合には、自動的に(c)側の EC2 をOpsWorks により起動、最新の環境へ自動デプロイ
- ③ EC2インスタンスを ELB に組み込む
- ④ 同時に、Redis や MySQL と言った元々 Multi-AZ 対応サービスについては、自動的に(c)側に変更される



ZABBIX からキックされる rubyスクリプトの中で OpsWorks API をコールしてレイヤーIDに基づいて ELB配下の EC2 を切り替えます。

- ✓ ELB はゾーン毎に存在するので、影響を受けない
- ✓ ELB は変わらないので DNS に起因する問題が起きない
- ✓ AWSが提供している Multi-AZ配置可能なサービスはそのまま使う
- ✓ ZABBIX と OpsWorks とのコラボ
- ✓ 全部コード化されている

- 高頻度なサービス改善を行う環境は AWSだと簡単に構築できる
- 継続的デリバリ x DevOps が新たに進化していく
- OpsWorks をその中心におくことで、インフラもアプリも同じ文脈で捉えることができ、できることへの可能性が広がっていく

インフラ×アプリの連携による エンジニア協業体制の変化

株式会社リクルートコミュニケーションズ
アドテクノロジー開発部

阿部直之



2015年06月03日

株式会社リクルートコミュニケーションズ

ICTソリューション局 アドテクノロジー開発部 アーキG

マネージャ / テクニカルリード



阿部直之 (あべ なおゆき)

- 独立系SIerにてインフラ～開発～運用など幅広い業務を担当
- RCOに入社後、エンジニアとしてアドテクノロジーサービスの立ち上げ
 - 効果計測タグシステムや運用管理システム等の開発を担当
- 開発プロセス改善やエンジニア支援環境の構築も担当
- エンジニア組織づくりというテーマで最近マネージャに ← イマココ

本日の内容を一言で

- 継続的デリバリの推進→分業から協働へ
 - **分業**：各々の専門領域を住み分けてタスクを遂行
 - タスクに特化したスキルに習熟し生産性を高めていく
 - **協働**：各々の専門を主担当としつつも周辺領域にもコミット
 - 相互作用によって**新しい価値を創造**する → コラボレーション



⇒ どのような変化が起こったのかを事例も含めてご紹介します

- RCOアドテク部のご紹介
- 継続的デリバリ推進前の風景
- クラウド（AWS）の到来
- そして継続的デリバリがやってきた
- 継続的デリバリがもたらしたものの
- さらに副次的な効果が。。。。

RCOアドテク部という組織

• RCOアドテクノロジー開発部

- アドテク製品やWebサービス開発を行う**エンジニア中心**の組織



アプリエンジニア



インフラエンジニア



UXデザイナー



ITプランナー

機械学習系エンジニア
も含まれます



データサイエンティスト



マネージメント

構成メンバーの人数比イメージ

● RCOエンジニアの特徴

– 激しく技術志向

- 採用はコーディング試験の突破が前提
- 良いコードが書ける・良い技術を持ってるやつがカッコイイ！

– プロダクト・**サービス**志向

- システム開発のその先を見据え圧倒的な当事者意識で開発を推進
- ビジネスや企画に対しても技術観点でコミット

– **職人**基質だけどちょっと地味

- でも、技術ドリブンで新たな価値を作って行くメンバー
- いろいろと面白い人が多いです



RECRUIT リクルートコミュニケーションズ エンジニア中途・新卒採用

> 新卒エントリー

> 中途エントリー
(コーディング試験を受ける)

ミッション

開発環境・技術向上支援

キャリアパス

インタビュー

募集要項



「テクノロジー×人を動かすコミュニケーション」でできることの未来を創る

<https://www.rco.recruit.co.jp/career/engineer/>

または「RCO エンジニア 中途採用」で検索！

グループインタビュー

リクルートコミュニケーションズでのエンジニアの働き方、考え方はどのようなものなのか。アドテクノロジー（アドテク）領域でコア開発を行っているエンジニア3名と彼らを束ねる大石部長の話をたっぷり聞きました。

※掲載情報は2014年9月取材時点のものです



知的好奇心を刺激しながら、
技術開発に挑戦できる環境がここに

継続的デリバリ推進前の風景

- 分業による協力体制で高い生産性を目指す組織
 - **アプリエンジニア**：システムを素早く・高品質に作っていく人
 - プログラミング技術を中心にシステム開発スキルを高めていく
 - **インフラエンジニア**：インフラを構築して運用をしていく人
 - サーバ・NW技術を中心にサービスの安定運用スキルを高めていく



- (特に) インフラ業務の周辺業務が膨大
 - ハードウェアの取り扱いに関するスキル
 - 各種機器の検討・購買の調整
 - ハードウェア組み立て、ラッキング、、、etc
 - ネットワークに関する諸々のスキル
 - ルーティング、OSPF, BGP、、、etc
 - 配線、ケーブリング、、、etc



プログラミングとは
直接リンクしづらい
業務がとても多い



イ、インフラ業務は
お願いしたほうがいいかな、、、

クラウド (AWS) の到来

- クラウドコンピューティング（AWS）
 - ハードウェアに関する業務が軽減
 - アプリケーションのようにインフラを利用可能（IaaS）

⇒ インフラ周辺業務のハードルが低くなった

- サーバ・NW技術に触れやすい環境が身近に
 - 冗長化等の可用性なども含めてすぐにアプリケーションを試せる
 - プライベートでのWebサービス公開などの機会も増大

イ、インフラにも
手を出しやすくなったような？



AWS使ってみようかな。。。

そして継続的デリバリーがやってきた

● 継続的デリバリ

– サービス価値を向上するためにソフトウェアを継続的にデリバリ

- 小さく高頻度なリリースによるPDCAサイクルの高速化

– リリース速度・FB対応力を高めるためのDevOps

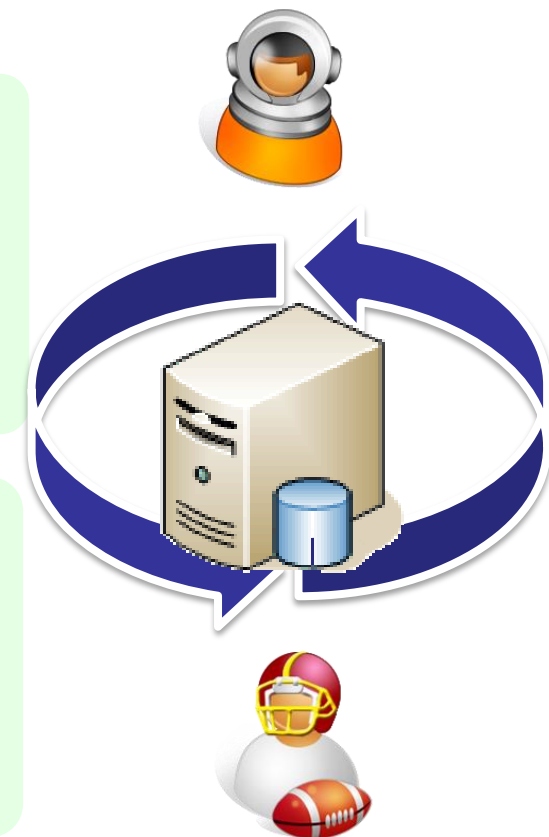
- アプリ（開発）とインフラ（運用）の協力体制の変化

⇒ 分業によるリードタイムの低下

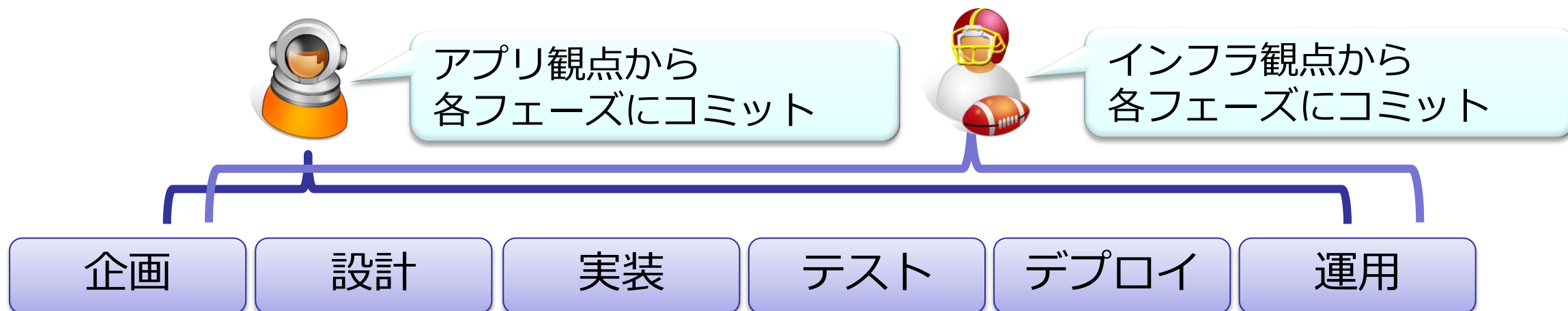
– IaCやCIによる暗黙知の形式知化

- 作業の自動化やコード化による手続きの共通言語化

⇒ アプリとインフラの共通言語も増えた！



- 各々がコミットすべき領域が拡大
 - **アプリエンジニア**：インフラ領域への主体的な関与
 - インフラ構成・運用・監視も念頭に置いたアプリ設計・開発
 - 障害の初期検知からの参加等、対応レベルが向上
 - **インフラエンジニア**：企画、開発工程への主体的な関与
 - OpsWorks導入などによるアプリ設計への参加
 - 各種自動化ツールの導入支援



継続的デリバリがもたらしたもの

- 協働によるサービス価値の創造的な向上
 - **アプリエンジニア**：開発を主軸にサービスを作る人
 - 機能開発や効果最適化を起点にサービス価値の向上にコミット
 - **インフラエンジニア**：インフラを主軸にサービスを作る人
 - 高度な安定運用やサーバ技術を起点にサービス価値の向上にコミット

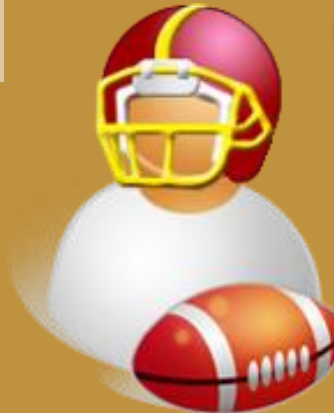
ディフェンスを意識しつつ
更にオフェンスに！！



アプリエンジニア

より攻め攻めに
技術志向で突っ走って価値を創造

協働



ディフェンスしつつも
オフェンスも！！

インフラエンジニア

無茶を許容する安定感を持ちつつ
新たに攻めて価値を創造

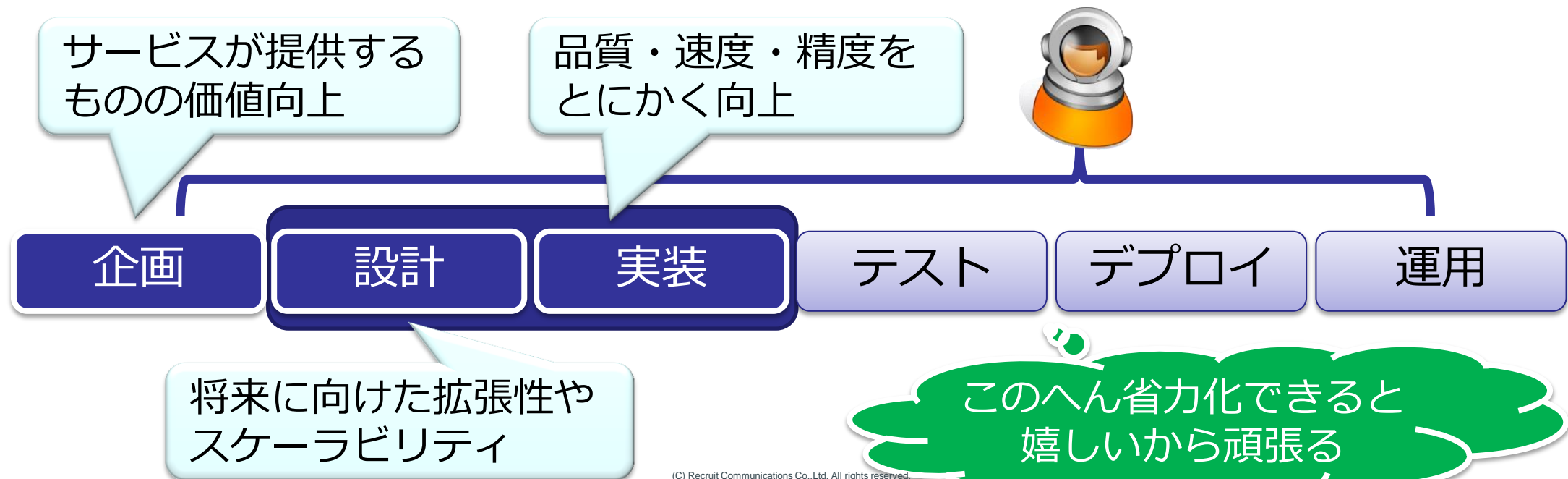
- サービス価値における**コアタスク**の明確化

- サービスの価値向上に一番影響のあるタスクを明確化

- 全てのフェーズが**サービス価値向上**につながる

- エンジニアが頑張れば頑張るほど**価値・成果につながるポイント**はどこ？

- それ以外のフェーズについては自動化等により省力化



- 開発プロセス内におけるDB構造変更の共有が必須
 - ある環境ではRDSとRedshiftのETLによる同期が必要



ETLによるデータの同期

Extract, Transform, Load



DB構造変更
によるETL失敗

同期が失敗するのでDB構造のズレはなくしたい

コンフリクトが発生!

DB構造も含めてアプリのアップデートは高頻度で行いたい



- 担当インフラエンジニアの選択

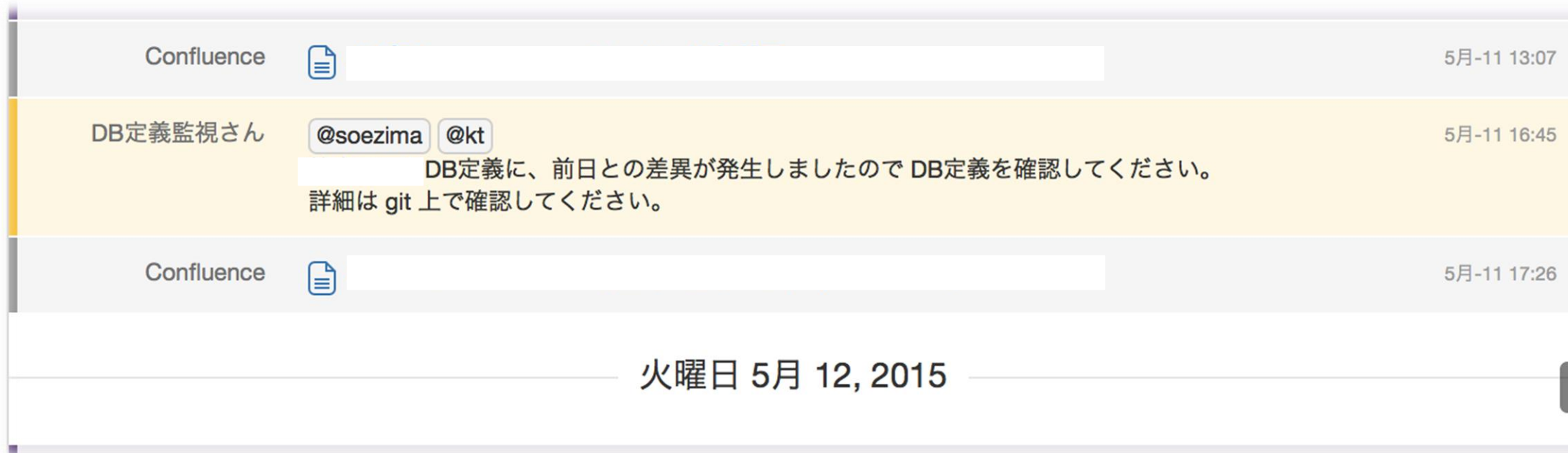
✖ DB構造の変更の確認手続きを導入して。。。

◎ マイグレートが行われたら自動的に検知して通知！



- サービス価値は高速なPDCAサイクルで向上する
 - ⇒ 開発・リリース速度を落とす選択肢はそもそもありえない
 - ⇒ 構造差分の検知は自動化が可能！！
 - ⇒ よし、やるか！！！！

- RDSとRedShiftのスキーマ構造を比較してChatへ通知



DB構造もアップデートし放題だぜ！！



さらに副次的な効果が。。。。

- 継続的デリバリ推進のための各種技術の転用
 - 自動化やChatOpsなどが業務以外にも応用
 - 部内イベント：コードバトル
 - 自動ゲーム実行環境によるアルゴリズム対決（海戦ゲーム）
 - Docker等の技術によりゲーム単位で環境を新規構築
 - 機械学習大会
 - エンジニア有志が自発的にはじめた機械学習のスキル向上イベント
 - 某「ファミコン時代の爆弾で戦うあのゲーム」をアルゴリズムで対戦



エンジニアのスキル向上が自発的に加速！
環境構築はもちろんAWS!!



- インフラ×アプリの連携における継続的デリバリ
 - AWSを利用したインフラ×アプリの連携事例
 - RCOにおける継続的デリバリ推進のための事例紹介
 - ブルーグリーンデプロイメント
 - ゾーンエクステンション
 - インフラ×アプリの連携によるエンジニア協業体制の変化
 - 継続的デリバリによりエンジニアの協働が加速
 - サービス価値を高めるメンバーの意識の変化
 - 各種技術が転用され組織的にプラスの効果が！！

ミッション

開発環境・技術向上支援

キャリアパス

インタビュー

募集要項



ご清聴ありがとうございました！

「テクノロジー×人を動かすコミュニケーション」でできることの未来を創る
ミッション

アドテクの進化のスピードに、エンジニア集団が100%以上の力を出し切って挑む

グループインタビュー

リクルートコミュニケーションズでのエンジニアの働き方、考え方はどのようなものなのか「RCO アドテクノロジー（アドテク）領域でコア開発を行っているエンジニア3名と彼らを束ねる大石部長の話をつぶりに聞きました。

※掲載情報は2014年9月取材時点のものです

<https://www.rco.recruit.co.jp/career/engineer/>



エンジニア 中途採用」で検索！

知的好奇心を刺激しながら、
技術開発に挑戦できる環境がここに