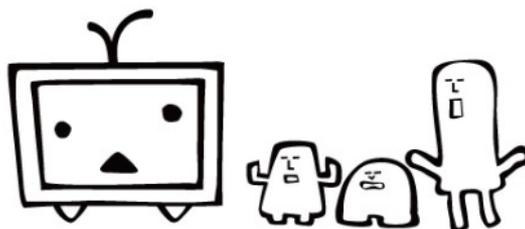


～ドワンゴがAWSを使ってみた～

AWS Summit Tokyo 2015



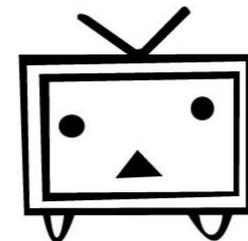
株式会社ドワンゴ プラットフォーム事業本部
共通基盤開発部 関 剛

わたしは誰？

わたしは誰?



- 名前：関 剛(せき たけし)
- 所属：株式会社 ドワンゴ
ニコニコ事業統括本部
共通基盤開発部
共通基盤インフラセクション
- 役目：インフラエンジニア(AWS初心者)
- 陣営：緑(A16); これから始める方は緑をおすすめします



なぜ今ここに立っているのか？

なぜ今ここに立っているのか?



- AWS を使ってサービス構築してみた
- ほぼ初めて AWS 使ってみた
- AWS ビギナーの視点から AWS 導入を振り返る
- AWS を使ってみて変わったこと



Live Dwango Reader

Live Dwango Reader



- Livedoor Reader サービス終了

A screenshot of an email notification. The header shows the date '1 Oct' and the subject line '【重要】 livedoor Reader サービス終了のお知らせ'. Below the subject is the date 'October 01, 2014' and a folder icon with the text 'お知らせ'. The main body of the email is a large blue rectangle with the 'livedoor Reader' logo in white. At the bottom, there is a paragraph of Japanese text.

1
Oct

【重要】 livedoor Reader サービス終了のお知らせ

October 01, 2014 | お知らせ

livedoor Reader

いつも livedoor Reader をご利用いただきましてありがとうございます。
2006年4月より提供してまいりました livedoor Reader ですが、このたび 2014年12月をもちまして、サービスを終了いたします。

Confidential

Live Dwango Reader



- Livedoor Reader サービス終了“撤回”

15
Oct

【重要】 サービス終了撤回のお知らせ

2014/10/15 | お知らせ



いつも livedoor Reader をご利用いただき、ありがとうございます。

先日、サービス終了をお知らせした livedoor Reader ですが、みなさまのご要望を受けこれを撤回し、**サービス継続の道を検討中です。**

Confidential

Live Dwango Reader



- Livedoor Reader サービス継続決定→“**運営譲渡**”

24
Oct

【重要】サービスの継続決定と運営譲渡のお知らせ

2014/10/24 | お知らせ

The logo for livedoor Reader, featuring the text 'livedoor Reader' in a white serif font and a white bird-like icon to the right, all centered on a green rectangular background.

livedoor Reader

いつも livedoor Reader をご利用いただき、ありがとうございます。

先日、サービス終了の撤回をお知らせした livedoor Reader ですが、このたびサービスの継続と株式会社ドワンゴ様への運営譲渡が決定いたしました。

Confidential

Live Dwango Reader



- Livedoor Reader から Live **Dwango** Reader へ

1
Dec

【重要】livedoor Reader 運営元変更のお知らせ

2014/12/01 | お知らせ

The logo for Live Dwango Reader, featuring a red circle with a white 'LD' monogram inside, followed by the text 'Live Dwango Reader' in a dark grey sans-serif font.

いつも livedoor Reader(LDR) ご利用いただき、ありがとうございます。

先日からお伝えしておりました通り、「livedoor Reader」は、本日12月1日よりサービス運営元がLINE株式会社から株式会社ドワンゴへと変更になりました。

Confidential

Live Dwango Reader



- 派生スマートフォン版 LDR Pocket リリース

2
Apr

LDR Pocket リリースのお知らせ

2015/04/02 | 📁 お知らせ



いつも Live Dwango Reader(LDR) をご利用いただき、ありがとうございます。

このたび、Live Dwango Reader の派生サービスとしまして、「LDR Pocket」をリリースいたしました。



LiveDwangoReaderを AWS で作る

LiveDwangoReaderを AWS で作る



- まとめると
 - 2014-10-01 Livedoor Reader サービス終了発表
 - 2014-10-15 Livedoor Reader サービス終了撤回
 - 2014-10-24 Livedoor Reader 運営譲渡発表
 - 2014-12-01 運営譲渡 & Live Dwango Readerへ
 - 2015-04-02 派生スマートフォン版 LDR Pocket
- 5ヶ月の猶予はあった(**5ヶ月しかなかった**)
 - →**リリース直前の3日間**ほどでAWSでインフラ構築

LiveDwangoReaderを AWS で作る



- エンジニアとして「イヤな予感」がする
 - Livedoor Reader のサービスはそのまま継続する
 - 8年間近く使われてきた枯れたサービス
 - 元Livedoor社は現LINE社になっている
 - LDR Pocket のリリース日は決まっている

LiveDwangoReaderを AWS で作る



- 「**そのまま**」を維持したまま移設する
 - ソースコードは**そのまま**
 - ミドルウェアは**そのまま**
 - データ(DB, キャッシュ)は**そのまま**
 - サーバのホスト名は**そのまま**
 - IP アドレスは**そのまま**
- サーバの負荷見積
 - 現行サーバのスペックはわかるが構成は**そのまま**
→サーバリソースの余剰が発生する可能性がある

LiveDwangoReaderを AWS で作る



● そこで AWS!



● 「そのまま」 どうする?

- ソースコードはそのまま→OK!
- ミドルウェアはそのまま→OK!
- データ(DB, キャッシュ)はそのまま→OK!
- サーバのホスト名はそのまま→OK!
- IP アドレスはそのまま→OK!

● サーバの負荷見積

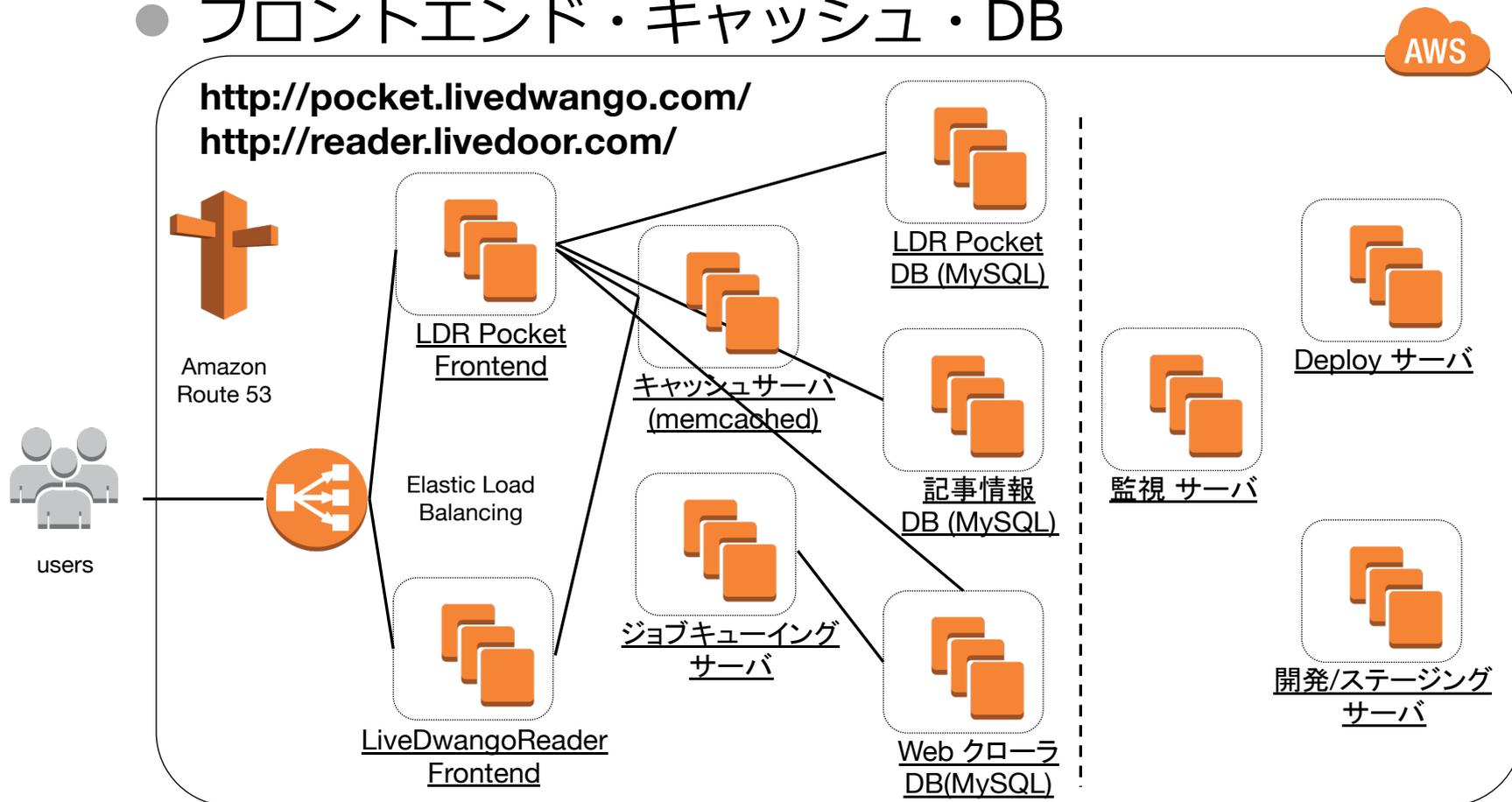
- 現行サーバのスペックはわかるが構成はそのまま
→**インスタンスタイプを変更すればよい!**

むしろ状況に合わせて変更できる!

AWS 上の Live Dwango Reader 構成

AWS 上の Live Dwango Reader 構成 Live Dwango Reader

- シンプル構成(もともとシンプルだった)
 - フロントエンド・キャッシュ・DB



AWS 使ってみた感想(ビギナー視点)

AWS 使ってみた感想



- EC2 / ELB / Route53 だけ使った
 - 旧環境と構成を合わせるため
- Webコンソールポチポチ→数分でインフラができる
 - オンプレミス環境だと機器の手配で数週間
 - 圧倒的なプロビジョニングスピード
- 懸念だったこと→結果的に杞憂だった
 - 仮想化のオーバヘッドは感じていない
 - ネットワーク、ストレージにボトルネック出てない
 - いわゆる「ELB暖機運転」→実施無しで問題なし

AWSで変わったこと(オンプレミス環境と比較)

AWSで変わったこと(オンプレミス比) Live Dwango Reader

- インフラエンジニア・アプリエンジニアの関係に**変化**
 - アプリ開発エンジニアがインフラ構築・運用できる
 - アプリ開発エンジニアがサービス開発に集中できる
 - **綿密なサイジング不要で調整コストがすっきり**
 - インフラ・アプリ開発業務のクロスオーバー
- インフラ環境の設計・構築のスピードアップ
 - 綿密なアクセス予測・負荷見積が**不要**→まず動かす
 - **仮にサイジングミスってもすぐに増減可能**
 - アプリエンジニアが直接見積するようになった

相互業務をオーバーラップして、柔軟にサービスに取り組めるようになった

AWSで変わったこと(オンプレミス比) Live Dwango Reader

- コンポーネントごとにコストが見えるようになった
 - フロントエンド, アプリ, DBそれぞれのコストを把握
 - エンジニア担当サービスが**コスト意識**を持つように
 - 固定費の変動費化 (将来的は差異分析, 評価したい)
- AWS でプロダクト担当者がデータセンタ持てる
 - プロダクト担当者のオーナー意識が向上する
 - アプリケーションから帯域、IOPSまで一貫意識
 - サービス障害発生時の対応に関する責任感

開発者自らがリソース調達を実施し、プロダクト=事業という視点が生まれ
プロダクトオーナーという意識を今までよりも持つようになった

さらに変えていくために（ドワンゴ視点）

さらに変えていくために



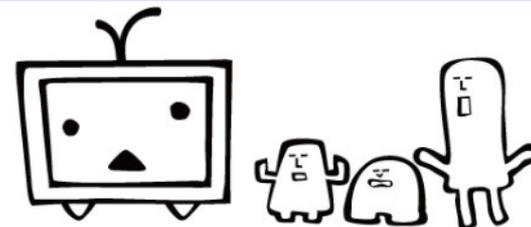
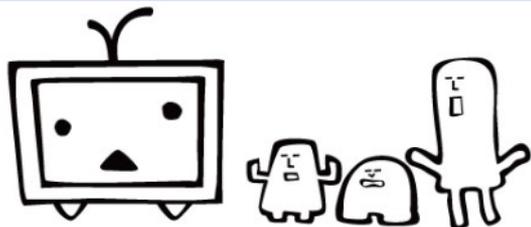
- オンプレミス(既存のデータセンタ)とAWSの使い分け
 - 「まずやってみる環境」：機会費用の最小化
 - 気軽なトライ & エラーの影響コスト最小化
- オンプレ→AWS / AWS→オンプレ両方向あり得る
 - 適材適所は必ずある、適材適所は変化する
 - 中期・長期(固定資産の耐用年数)でのTCO比較
- 最適なインスタンスタイプと課金メニューの選択
 - リザーブドインスタンスを利用→低コスト化
 - 一時的環境としてスポットインスタンスの利用
 - 検証環境・開発環境など
 - 短期間で大量のコンピューティングリソース

さらに変えていくために



- より良いサービス・スピードを生み出すために
 - オンプレ環境とAWS環境のシームレス接続
 - VPC VPN / Direct Connect
- Web コンソールポチポチからの脱却
 - API 利用と既存オンプレ環境との操作性の統一
 - 手順のコード化・自動化
- EC2 以外のサービスの利用
 - 「スイスアーミーナイフ」を使いこなす
 - AWSサービスをもっと試す・知る
 - AWSサービスの使いどころを判断する

Born in the NET, Connected by the NET.



ありがとうございました!