



# データ活用のための権限管理

2016/09/26

BigData-JAWS 勉強会 #2

Hokuto Hoshi

[hokuto@cookpad.com](mailto:hokuto@cookpad.com)

# 星 北斗 (ほし ほくと) / @kani\_b

---

- クックパッド株式会社  
インフラストラクチャー部 部長
- 2013年新卒入社
- Web {インフラ, セキュリティ} エンジニア
  - 設計/構築/運用 (最近だとログ処理基盤とか)
  - 脆弱性診断, セキュリティ設計,  
IDS, WAF, ISMS 運用 etc
- AWS 認定 SA, DevOps エンジニア (Professional)
- 好きな AWS サービス: EC2, Lambda, Kinesis\*, IAM, CloudTrail



# ところで...

---

- ・ データは好きですか？？？ (雑)

# 過去の発表

Speaker Deck

Search...

Sign Up

Sign In

Speaker Deck

Published on Jun 3, 2016


 cookpad

秒間数万のログを  
いい感じにするアーキテクチャ

Hokuto Hoshi @ Cookpad Inc.  
2016/06/03 AWS Summit Tokyo 2016

◀ ▶

share

 **Hokuto Hoshi**  
1 Presentation

★

Star this Talk

54 Stars

📅

Published in

Technology

📊

Stats

20,428 Views

Share

🐦

Twitter, Facebook

</>

Embed

🔗

Direct Link

📄

Download PDF

秒間数万のログをいい感じにするアーキテクチャ  
by Hokuto Hoshi  
Published June 3, 2016 in Technology

AWS Summit Tokyo 2016 Developer Conference (2016/06/03)



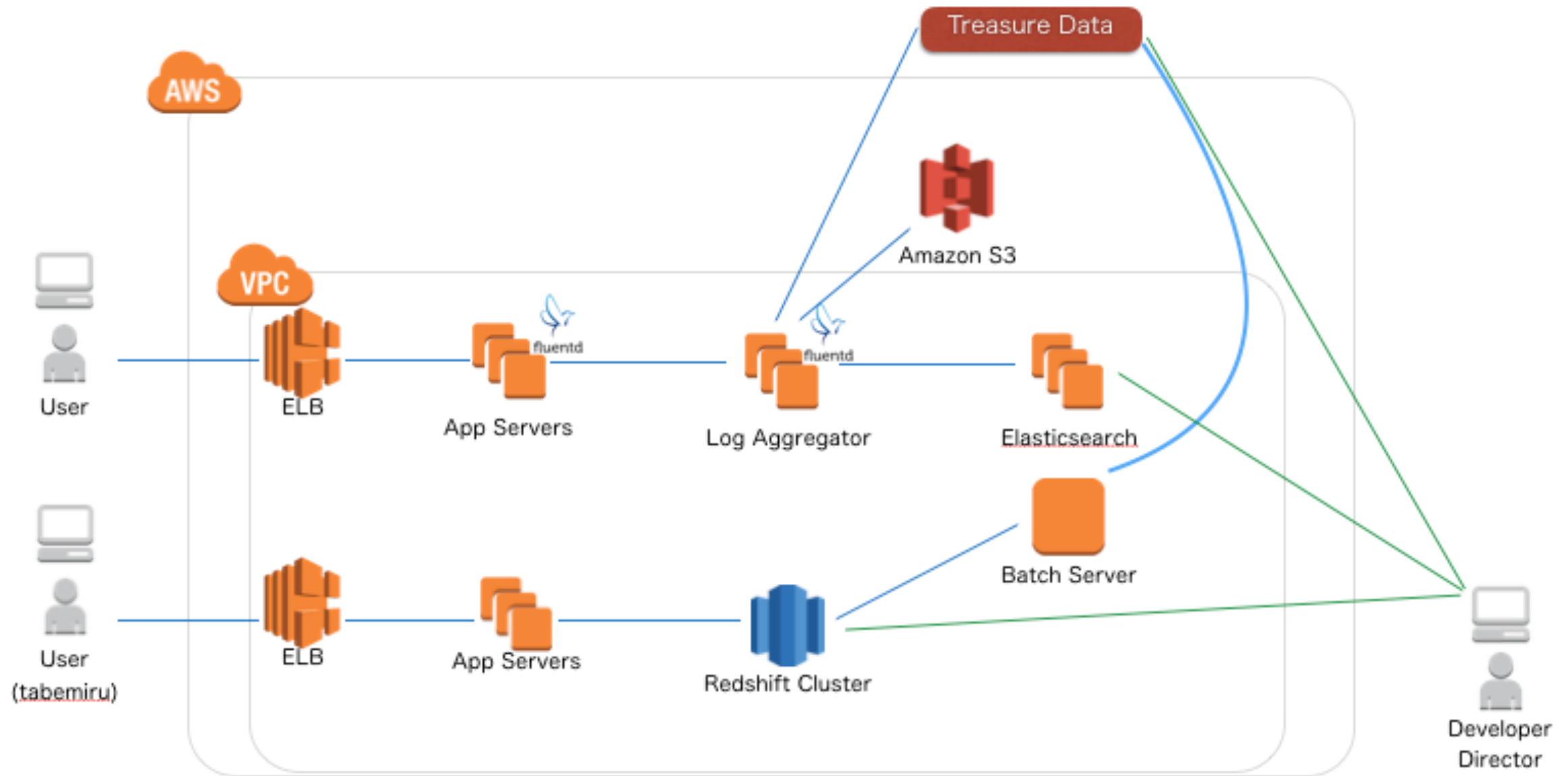
# 現在の規模

---

- Fluentd に流れているログ (概算)
  - データ総量: 500~700GB / 日
  - レコード数: 10億レコード以上 / 日
  - 秒間 10,000 ~ 30,000 レコードくらい



# ログ収集アーキテクチャ



# データはどこにあるのか

---

- Redshift や Treasure Data
  - ログデータが多い
  - サービスデータも一部転記されている
- MySQL
  - サービスデータが存在する
- Elasticsearch
  - 一部ログデータが存在する (ほぼ Kibana 用)
- S3
  - AWS 関連のログやシステムログ

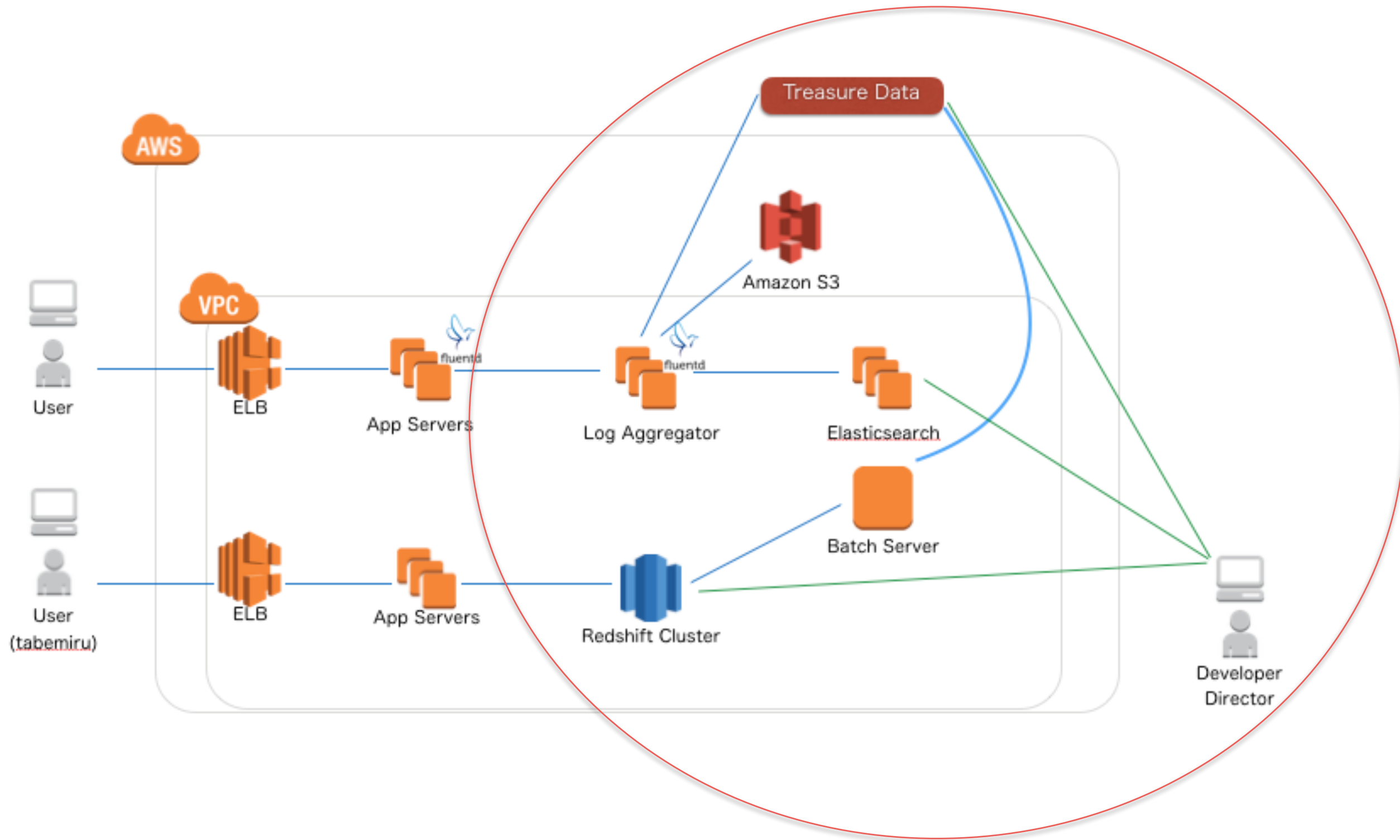
# どんなデータがあるのか

---

- サービスのログデータ
  - 行動、検索、監査、etc
- サービスそのもののデータ
  - レシピ、献立、ユーザ、etc
- システムのログデータ
- 使われなければ意味がない!!!



# ログ収集アーキテクチャ



# 誰がデータを使っているのか

- ・ サービスに関わる全ての人
  - ・ エンジニアはもちろんディレクターも



クックパッド開発者ブログ  
COOKPAD Engineers' Blog

2016-07-06

## ディレクターがSQLを使えてよかった話

こんにちは。ディレクターの川原田です。クックパッドでお気に入りレシピを保存する「MYフォルダ」のサービス開発や、保存・記録に関する新規サービスの検討・開発を担当しています。

ディレクターの仕事は様々ありますが、今回は私が身につけたことで仕事領域が広がった！と感じているSQLについてお話ししたいと思います。

いきなりですが、SQLが使えてよかった点をまとめると以下です。

### よかったこと

- ・ 数値抽出から分析まで自己完結
- ・ エンジニアとのコミュニケーションがスムーズに
- ・ 仕事が増えていそうで実は効率アップ
- ・ 周囲の知的好奇心を刺激

それぞれ具体例を交えてお話します。

クックパッドでは  
エンジニア・デザイナー  
を募集中です

[詳しくはこちら](#)

クックパッド スタッフによる  
開発者向け発表資料

[詳しくはこちら](#)



検索

ブログ内検索 🔍

### 最近の投稿

- ・ 仮説検証とサンプルサイズの基礎
- ・ ユーザーをムフムフさせるための

# 湧くかもしれない疑問

---

- ・ 権限管理どうしてるの？
  - ・ 担当外のサービスデータは？
  - ・ 個人情報情報は？？？
- ・ ところが今回のメインテーマ
  - ・ なんですけど思ったより短いと思います

# 基本的な方針

---

- ・ いわゆる個人情報は何よりも保護する
- ・ 各サービスのデータは誰でも使えるように
  - ・ サービスを越えた分析ができるように

# 個人情報の扱い

---

- ユーザ ID そのものと具体的な個人情報 (e-mail, 住所, 電話番号 etc) は分離されている
- 個人情報が含まれるテーブルは特定の prefix を付加
  - 機械的に分離しやすくするため
  - 設計ルールとして定義されレビュー時にチェック
- (法的な) 個人情報には該当しないが  
プライバシー上の問題がありうるデータ
  - データの性質によるため収集前に個別対応

# どう分離されるか

---

- いずれかの手段でデータは本番のみに残す
  - レプリケーションしない
  - 上書きしてマスクする
- アプリケーションログとしても残らない
- DWH, staging などが対象

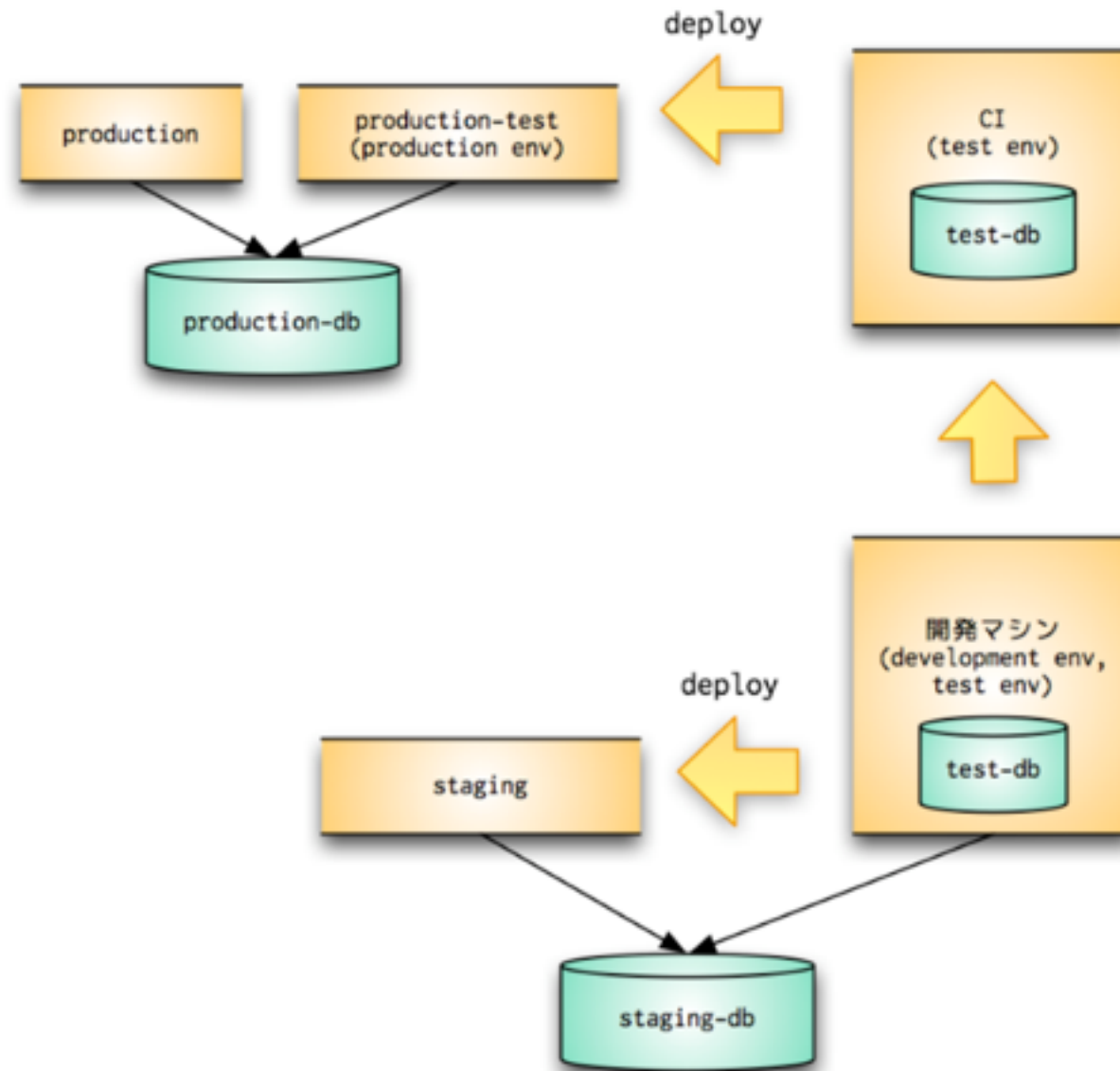


# 開発環境のデータをできるだけ本番に近づける

こんにちは。技術部の吉川です。今回はクックパッドの開発環境構成、特に開発用データベースの構成についてご紹介します。

## 開発環境の構成

クックパッドのシステム環境は以下のようなフェイズに分かれています。



※ これはcookpad.comの構成で、サブシステムや個別のサービスはその規模や特性に応じて

クックパッドでは  
エンジニア・デザイナー  
を募集中です

[詳しくはこちら](#)

クックパッド スタッフによる  
開発者向け発表資料

[詳しくはこちら](#)



### 検索

ブログ内検索

### 最近の投稿

- 仮説検証とサンプルサイズの基礎
- ユーザーをムフムフさせるための「お料理アルバム」デザインリニューアル
- エンジニア全体ミーティング Tech MTGのすゝめ
- ECS を利用したオフラインジョブの実行環境
- インターンシップ「サービス開発演習」の舞台裏
- クックパッド サマーインターンシップ2016の資料を公開します
- 新規アプリのデザインで心がけた5つのこと
- Ruby on Rails アプリケーションにおけるモンキーパッチの当て方
- 新サービス立ち上げ時の重要指標のデザイン

# 個人情報が含まれるテーブルを参照したいケース

---

- ・ エンジニアの調査
- ・ ユーザへのコンタクト
  - ・ etc...
- ・ 個別のユーザアカウントを発行して権限付与
  - ・ 例外なし

# 日常的な DB 参照

---

- ・ センシティブデータへのアクセスがないもの
- ・ 読み取り専用のユーザを利用
  - ・ バッチ用などには使わず人間のみ
  - ・ BI 用の GUI を別途使っている人もいる
- ・ DWH などとは全て個人ごとのユーザを利用
  - ・ ワークロード管理のため

# データソースの権限管理（どうやるか）

---

- 信頼と実績のスプレッドシート管理
  - 証跡残しにくい、履歴管理むずい
  - 消耗しかない
- コードを使った管理に変更した
  - MySQL, PostgreSQL, Redshift

# コードによる権限管理

---

- winebarrel/gratan (<https://github.com/winebarrel/gratan>)
- winebarrel/posgra (<https://github.com/winebarrel/posgra>)

```
require 'other/grantfile'

user "scott", "%" do
  on " *.* " do
    grant "USAGE"
  end

  on "test.*", expired: '2014/10/08', identified: "PASSWORD '*ABCDEF'" do
    grant "SELECT"
    grant "INSERT"
  end

  on /^foo\.prefix_/ do
    grant "SELECT"
    grant "INSERT"
  end
end

user "scott", ["localhost", "192.168.%"], expired: '2014/10/10' do
  on " *.* ", with: 'GRANT OPTION' do
    grant "ALL PRIVILEGES"
  end
end
```

# Git, GHE

---

- コード (テキスト) であることによるメリット
  - VCS が使える
- GHE を使った管理スタイルに変更



# GHE を使った権限管理の流れ

---

- 開発者が必要な権限を編集して Pull-Request
- 管理者が権限のレビューをしてマージ & 適用
- 作業理由は Pull-Request に残る
- 履歴はそもそも Git により管理される
- ツールにより冪等性が担保されている
  - 棚卸し漏れなどの可能性を軽減できる

# コードによって管理されているもの

---

- MySQL, PostgreSQL, Redshift の権限
  - IAM の権限 (User, Group, Role 全て)
  - データベースのスキーマ
  - Route53 の設定
  - etc...
- 
- <https://codenize.tools/>

# その他のアクセス経路

---

- 管理アプリケーションやバッチなど
  - アプリケーション側で認可処理を行う
- re:dash など
  - ユーザ管理ができるようになったので導入
- 認証は基本的に Google Apps 経由での認証
  - 自前で作りこむより遥かに良い

# コードによる権限管理になっていないもの

---

- Treasure Data
  - 個々のアカウントで権限分離して運用
  - ほぼ自動化されている
- Elasticsearch / Kibana
  - 今のところ必要としていない
  - サービス開発上の BI は re:dash に移っている

# まとめ

---

- データの活用を前提にした権限の管理についてお話しました
- 会社規模によって必要になる対策は異なりそう
  - より細かい権限管理など
- 本当に守りたい情報以外は自由に参照できるままにしたい
  - モニタリングの仕組みを用意している  
(audit plugin など)

# Thank you!

毎日の料理を楽しみに

