

# サーバレスアーキテクチャで実現した 『M-1グランプリ』敗者復活戦投票システム



朝日放送株式会社 技術局開発部 小南英司





# 自己紹介



🌀 **小南 英司** (こみなみ えいじ)

🌀 朝日放送株式会社 技術局開発部

🌀 **サーバサイドの構築** から **アプリの実装** まで

▶ **プリキュア応援アプリ**の開発/実装

▶ **高校野球速報アプリ**の開発/実装

▶ **ライブ動画配信用制作システムの構築**

 @eijikominami

...など



# Agenda

## M-1グランプリ投票システムの概要

- M-1グランプリの裏側を支える投票システムとは -

## AWS/サーバレスを選択した理由

- なぜAWSのサーバレスアーキテクチャを採用したか -

## 番組との連動に耐えうるシステムの設計

- 信頼性の高いシステムの構築 -



# M-1グランプリ投票システムの概要



# M-1敗者復活戦投票システムの概要

## 視聴者投票の実施と集計を行うシステム

- ▶ AWSのサーバレスアーキテクチャの活用
- ▶ 自社社員2名のみでシステム構築から運用までを担当
- ▶ 生放送や番組連動でも確実に運用可能な高可用性の実現

❖ 視聴者投票で最も得票を得たコンビが、決勝戦の舞台へ進出

AWS Webページ

投票

AWS 集計処理

投票

ON AIR 結果の表示

1 尼神インター	11 尼神インター
2 亡ろサーモン	12 モンスターエンジン
3 ナイツ	13 相席スタート
4 チーモンチョー	14 POISON GIRL BAND
5 東京ダイナマイト	15 がまいたち
6 拳天即	16 ダイタク
7 さらば舞音の光	17 セルライトスバ
8 ダイアン	18 田舎将棋
9 笑撃戦隊	19 アインシュタイン
10 天竺鼠	20 ニッポンの社長

結果



# M-1敗者復活戦投票システムの概要

AWS

Amazon Web Services

Webコンテンツの提供

膨大な負荷への対策

システム管理

Amazon CloudFront

Elastic Load Balancing

Amazon EC2

Amazon CloudWatch

Webコンテンツ

視聴者

2016年～  
データ放送からも  
投票が可能に！

インターネット

投票データの受付

データの抽出/判定

データの蓄積

結果表示ページ

Amazon Kinesis

Kinesis Enabled App

Amazon DynamoDB

Amazon EC2

投票データ処理

安定した集計処理の実現

表示

管理

番組制作者  
システム運用者



# 集計結果の表示とエクスポート



## 投票データはリアルタイムに集計される

- ▶ 番組関係者**専用Webページ**による結果の表示
- ▶ CG送出等にも利用可能な**JSON形式**による**データエクスポート**機能

敗者復活戦のリアルタイム投票総数を提供しています。結果は2秒に1回更新されます。

201862

## 2. 投票結果

敗者復活戦のリアルタイム速報を提供しています。結果は2秒に1回更新されます。

エントリーNo.	コンビ名	得票
2311	トレンディエンジェル	39753
753	とろサーモン	23916



# AWS/サーバレスを選択した理由

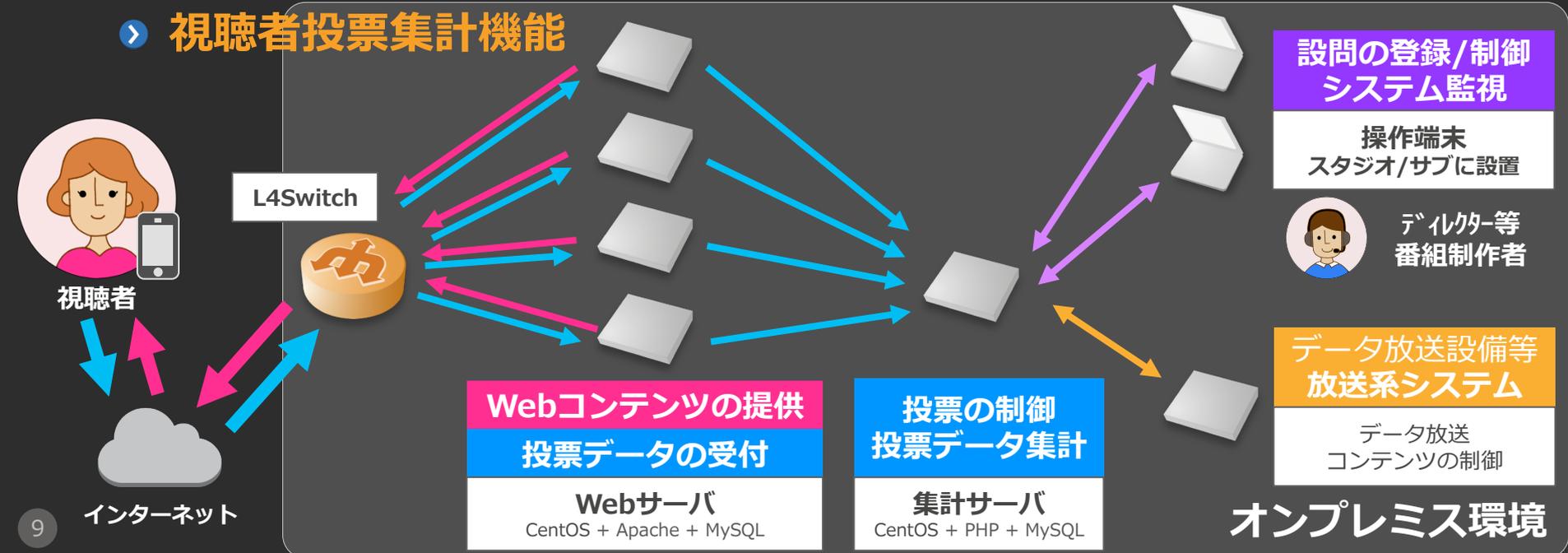


# 実はオンプレにも投票システムがあります



## ABC放送通信連携システム (2012年～)

- ▶ データ放送コンテンツ制御
- ▶ SNS連携機能
- ▶ **視聴者投票集計機能**





# 実はオンプレにも投票システムがあります

## ABC放送通信連携システム（2012年～）

- ▶ 全国ネット番組からローカル番組まで多くの実績

ABC 制作の全国ネット番組

スーパーベースボール



ABC 制作の関西ローカル番組



選挙 STATION 2016



忙しい夕方に やさしいニュース



月・金 4時58分～7時





# 既存の投票システムを使わなかった理由

## アクセスの集中↔限りあるサーバリソース

- ▶ 5年ぶりに復活/年に1度のお笑いイベント
  - 2008年決勝戦視聴率：35.0%（関西地区）
- ▶ 敗者復活戦出場コンビの運命を決める投票企画

❖ オンプレミス環境では投票データを処理しきれない可能性

## M-1グランプリ専用に機能の作り込みが必要

- ▶ ...なのに限られた非常にタイトな開発スケジュール

開発期間：わずか43日

仕様決定  
開発開始

10/13



開発完了

11/25



検証期間

11/26-12/5

9日間



M-1グランプリ当日

12/6





# AWS/サーバレスを選択した理由 当たり前ですが...

## 🌀 アクセス集中による高負荷対策がラク

▶ 「マネージドサービス」を利用することで簡単に負荷対策等が可能に

❖ サーバの増設, 設定値のチューニングなどの手間を省力化

❖ 短い開発期間（1ヶ月強）でシステム構築が可能

## 🌀 ドキュメントやツールが豊富

▶ SDKの組み込みにより、プログラムの実装を簡素化

❖ 少ない学習コストで実現したいシステムを構築

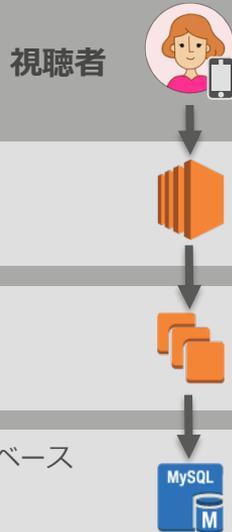
❖ 少ない作業コストでシステム運用が可能



# AWS/サーバレスを選択した理由



## 従来の構成



ロードバランサ

従来型

複数台の仮想サーバ

従来型

リレーショナルデータベース

従来型

負荷対策やバックアップ等  
設定作業が必要



## 本システム（サーバレスアーキテクチャの採用）



Amazon Kinesis

サーバレス

大量データをリアルタイム処理  
バッファ機能による負荷の平準化

Kinesis Enabled App

自社開発プログラム

徹底した作り込みが可能に！

Amazon DynamoDB

サーバレス

高い拡張性、自動バックアップ機能

負荷対策やバックアップはサービス側におまかせ  
⇒ プログラムの開発に人的リソースを集中



# AWS/サーバレスを選択した理由



## 従来の構成

視聴者



ロードバランサ

従来型



複数台の仮想サーバ

従来型



リレーショナルデータベース

従来型



負荷対策やバックアップ等  
設定作業が必要



## 本システム（サーバレスアーキテクチャの採用）



視聴者



Amazon Kinesis

サーバレス

放送通信連動企画で、  
本サービスの利用は日本初！



Kinesis Enabled App

自社開発プログラム

徹底した作り込みが可能に！



Amazon DynamoDB

サーバレス

高い拡張性、自動バックアップ機能

負荷対策やバックアップはサービス側におまかせ  
⇒ プログラムの開発に人的リソースを集中



# Kinesisはバースト対策に最適



## 番組連動：バーストトラフィックが発生

- ▶ 番組内の告知によって**莫大なトラフィック**が発生
- ▶ 全投票数のうち**告知後1分間の投票が2～6割を占める**



## Amazon Kinesis：水量調整機能を担うダム

- ▶ 指定したシャード数のトラフィックは**きちんと受けてくれる**
- ▶ Kinesis Enabled Appの**処理能力を自分で決められる**

❖ 莫大なアクセスが来ても自分で負荷をコントロールできる





# DynamoDBで簡単集計処理

## テーブル構成

- 投票データテーブル
- 得票数テーブル

## Atomic Counter

- 投票データ受信毎にどのコンビであるかを判定し、**updateItemメソッド**によって各コンビの得票数を増減





# API Gatewayを用いた投票データ受付

## 🔧 テレビからの投票に対応（2016年～）

- ▶ データ放送コンテンツ
  - ・ 特殊なマークアップ言語（**BML**）で記述されている
  - ・ スクリプト上で使える関数が限られている
- ▶ 通信を行う上での制約
  - ・ HTTP(S) POSTのみ送信可能
  - ・ POSTで送信可能なパラメータは1つだけ（Denbun）

❖ テレビ受像機からKinesisには直接データ投入できない





# 番組との連動に耐えうるシステムの設計



# 番組との連動に耐えうるシステムの設計

番組内での投票呼びかけによって短時間に莫大なアクセスが集中

**安定した集計処理**



# 負荷試験に基づいたシステムの最適化

## 番組告知時に莫大な負荷が集中

- ▶ 高負荷に耐えうるシステムであるか何度もテスト
- ▶ 負荷試験に基づいてシステム構成や設定値を決定

### AWS 負荷生成システム



300万件/20分  
最大 **1万件/毎秒**



### AWS M-1投票システム



想定**の最大10倍程**の  
投票リクエストを受信



# 集計処理プログラム (Kinesis Enabled App)



## 高速処理

- ▶ 複数スレッドによる**並列処理**と**パラメータの最適化**



## 不正データの排除

- ▶ 厳格なフォーマットチェック



## 高い安定性

- ▶ 強力な例外処理やエラー処理を実装



## 高い汎用性

- ▶ パラメータを詳細に指定することが可能
  - **全国ネットから深夜帯ローカル番組まで**様々な番組で使用できる

```
# java -jar kinesis-app-m1gp.jar -streamname m1gp_mob -region ap-northeast-1 -maxrecords 1000 -deadline 1449394200000
```

Kinesis



データ取得

Kinesis Enabled App



書き込み

DynamoDB





# 番組との連動に耐えうるシステムの設計

番組に与える影響が甚大  
1秒もシステムを止められない

システムの多重化



# バージニア州にバックアップシステムを構築

## 本番システム

東京



AWS

AWS

❖ 東京・バージニア間で  
受信データバックアップ転送

❖ 通信障害やデータセンタ障害時も  
バックアップシステムで処理を継続

## バックアップシステム



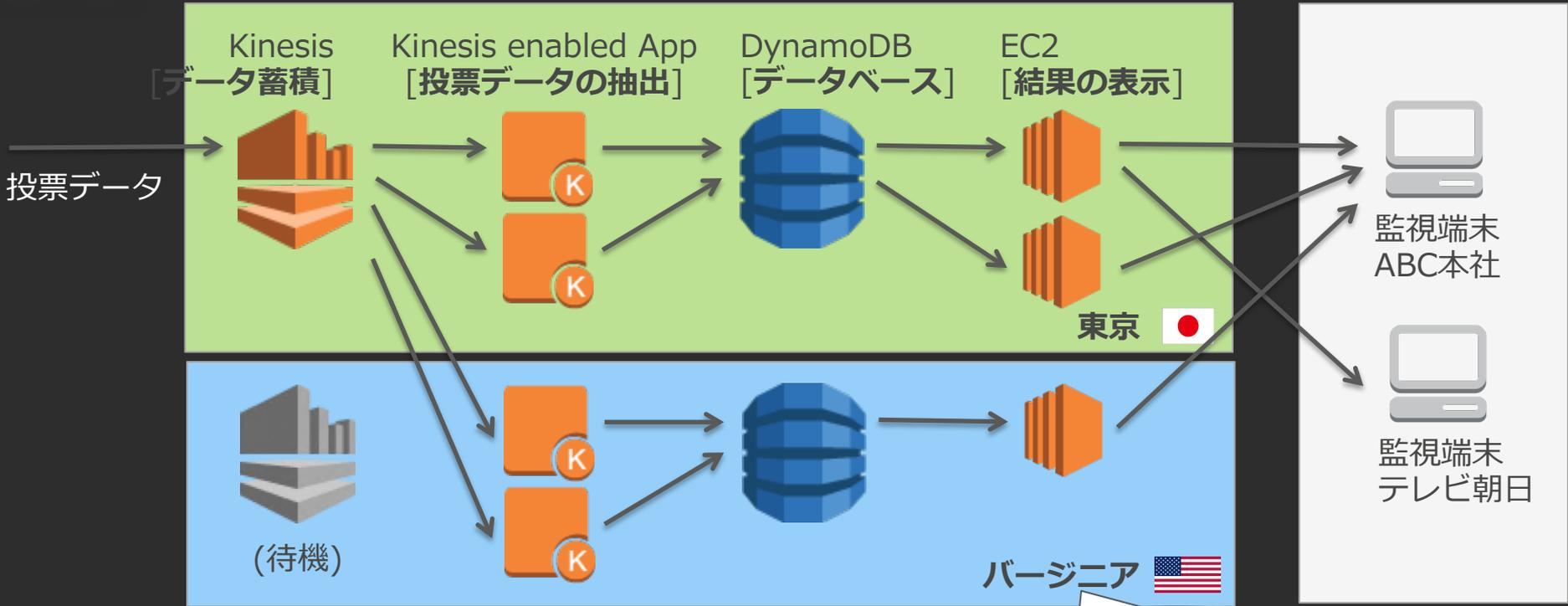
US・バージニア州



日米で**同一システム**を構築  
日米**並行して集計処理**を実施



# バージニア州にバックアップシステムを構築



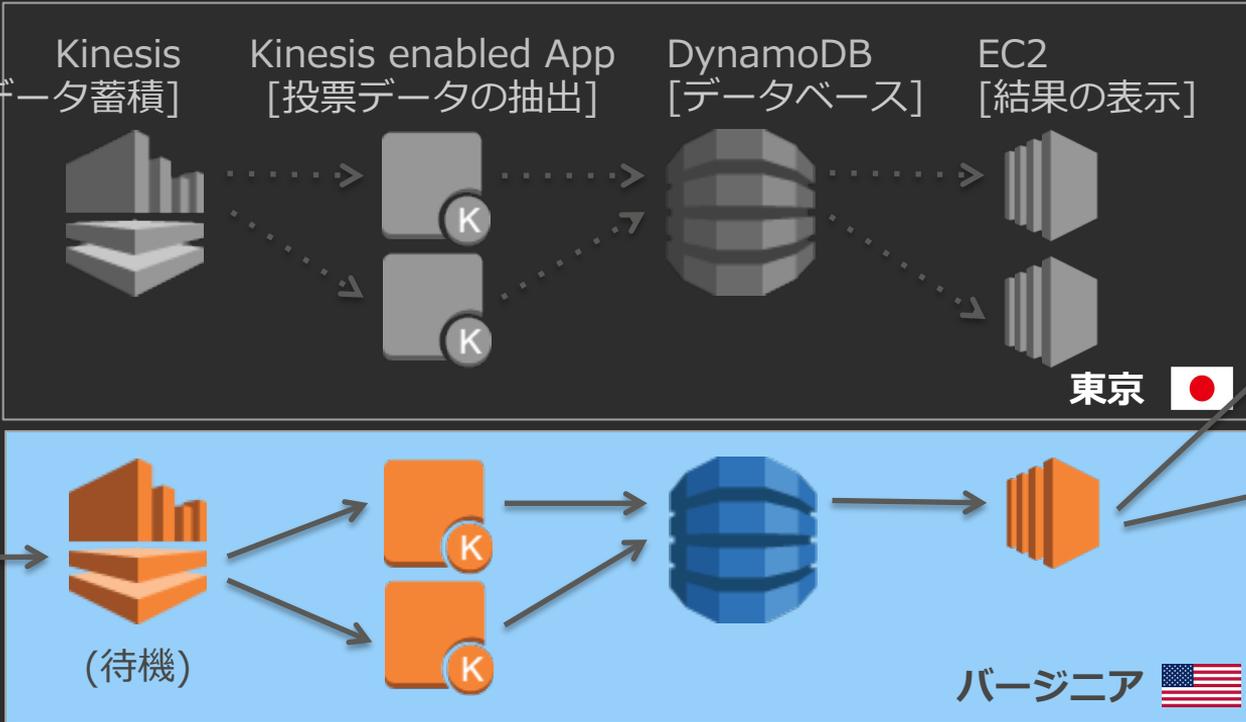
AWS創業の地  
リソースが豊富

## 通常時

❖ 投票データを共有し、並行して集計処理を実施



# バージニア州にバックアップシステムを構築



## 障害発生時

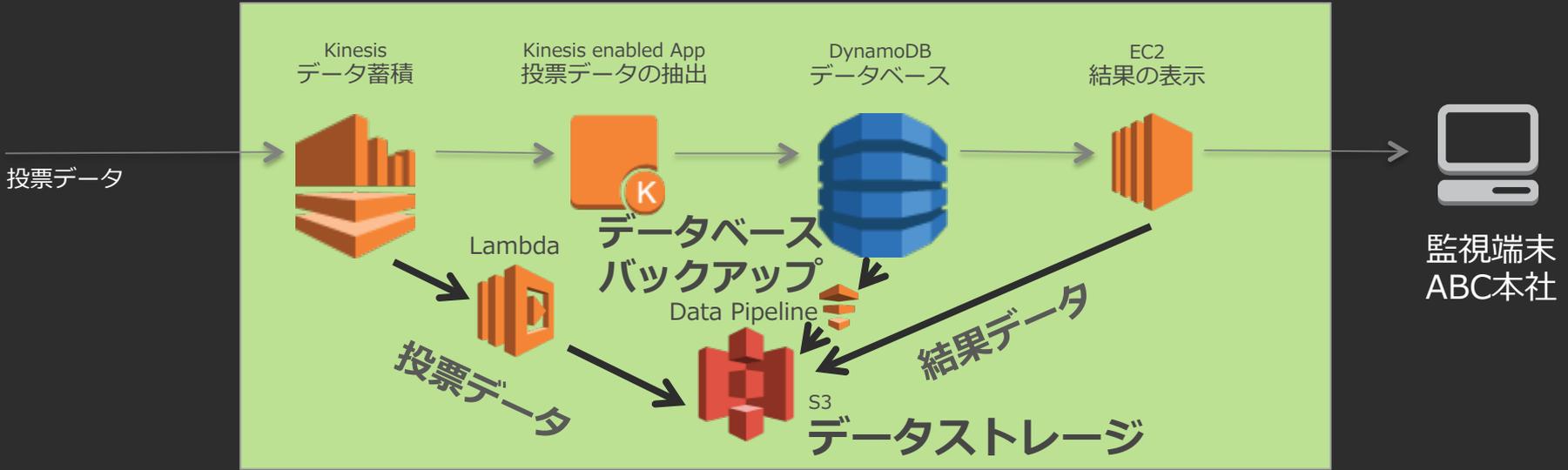
❖ バックアップシステムで集計を継続する



# データストレージへの自動バックアップ

各システムが持つデータ/ログをストレージにバックアップ

- サービス障害が発生してもある程度復旧可能



## 通常時

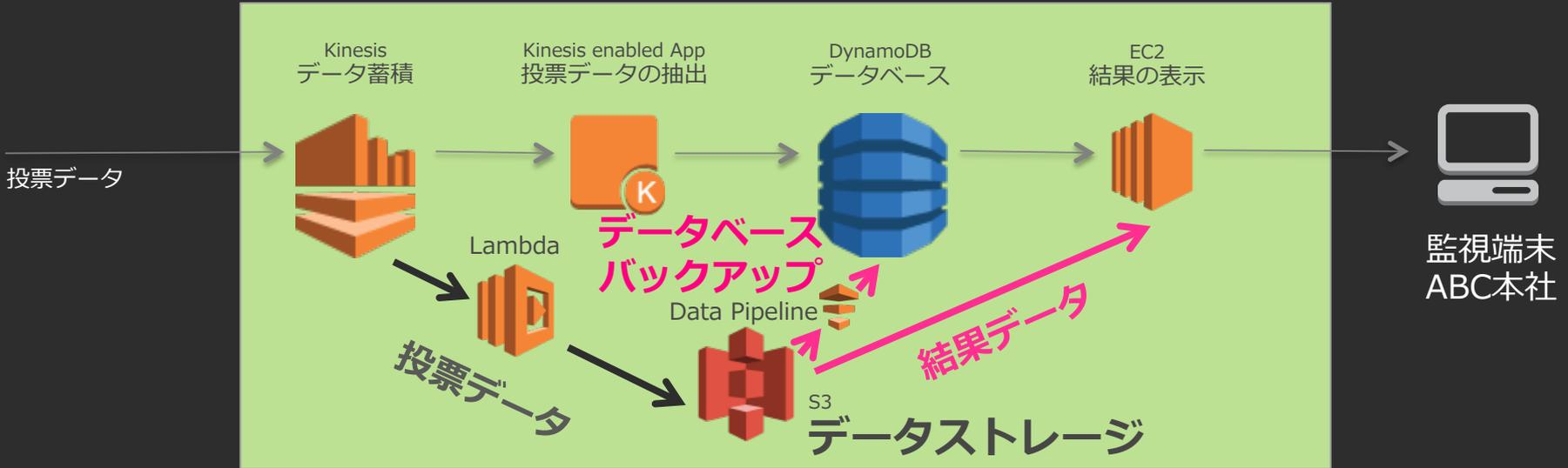
各システムのデータ/ログを定期的にバックアップ



# データストレージへの自動バックアップ

各システムが持つデータ/ログをストレージにバックアップ

- サービス障害が発生してもある程度復旧可能



## 障害発生時

❖ バックアップデータから投票データを復元



# 運用負荷低減の試み

## システム設定用スクリプトを作成

- 各システムの初期化や設定変更を行う対話式スクリプト
- システムの再構築や初期化を**簡単**に行うことが可能

```
# sh /home/ec2-user/initialize-dynamodb/initialize-dynamodb.sh  
Read/WriteCapacityUnits : 1000
```



システム運用者

少人数でも  
運用可能に



操作端末

### 自社作成スクリプト

- Kinesis設定スクリプト
- データベース設定スクリプト
- ログ取得設定スクリプト
- 集計システム設定スクリプト
- ストレージ設定スクリプト

初期化  
設定変更





# クラウドの機能を**最大限**に活用したシステム

## サーバレスアーキテクチャの活用

- ▶ **自社社員2名のみ**でシステム構築から運用までを行う
- ▶ プログラムの**開発実装**にリソースを集中
  - わずか1ヶ月強でシステムを内製
  - 安定性を重視したシステムを設計



❖ システムの構築

高い信頼性を実現

❖ 当日の運用

少ない作業負荷

今後も改修/機能追加を予定...

放送通信連携の**コア**となるシステムへ



さいごに...

他の事例のご紹介



# 動画配信システム基盤におけるAWSの活用

## 動画配信コンテンツ管理システム (技術協力: 株式会社 デジアサ)

- ▶ 動画配信プラットフォームに素材やメタデータを配信



## 動画プレイヤーログ解析

- ▶ **サーバレス & 完全自社開発** でシステム運用中





# 放送システムとAWSとの連携

## CG送出機の制御と連動したWebシステム

▶ 若手社員から成る開発チームによる自社開発





朝日放送株式会社