



3-P2-1-12

Startup Tech Talks

Data Lake と分析事例集

深澤 雄太, 株式会社AppBrew 代表取締役

礪 暁毅, Bulbit株式会社 TOP Platform Architect

小林 寛和, 株式会社primeNumber 執行役員 エンジニア

塚田 朗弘, アマゾン ウェブ サービス ジャパン株式会社 シニアソリューションアーキテクト

Data Lake を構築、活用していますか？

- AWS スタートアップチームは、日々多くの企業を技術的に支援
- その中で最もよくいただくご相談の一つが「ログの分析方法」
- 今日はモデルケースとなる3社にご登壇いただき実践的な知見を共有していただきます



本日の発表テーマとご登壇いただく皆様

1. ベンチャーでも出来る開発基盤

- 株式会社AppBrew 代表取締役 深澤 雄大 氏

2. UNICORN

- Bulbit 株式会社 TOP Platform Architect 礒 暁毅 氏

3. スタートアップ企業が100億レコード/月捌くために データ基盤作った話

- 株式会社 primeNumber 執行役員 エンジニア 小林 寛和 氏

株式会社 AppBrew
代表取締役

深澤 雄太 氏

ベンチャーでも出来る開発基盤

AWS Summit Tokyo 2018

@yfuka86



LIPS

深澤 雄太 / Fukazawa Yuta / @yfuka86

株式会社AppBrew
代表取締役 兼 エンジニア
スタートアップ特有のジェネラリスト
実はまだ学部生



株式会社AppBrew

<http://appbrew.io>

2016年 設立

2017年 コスメのコミュニティアプリ LIPS をリリース

累計6.5億調達



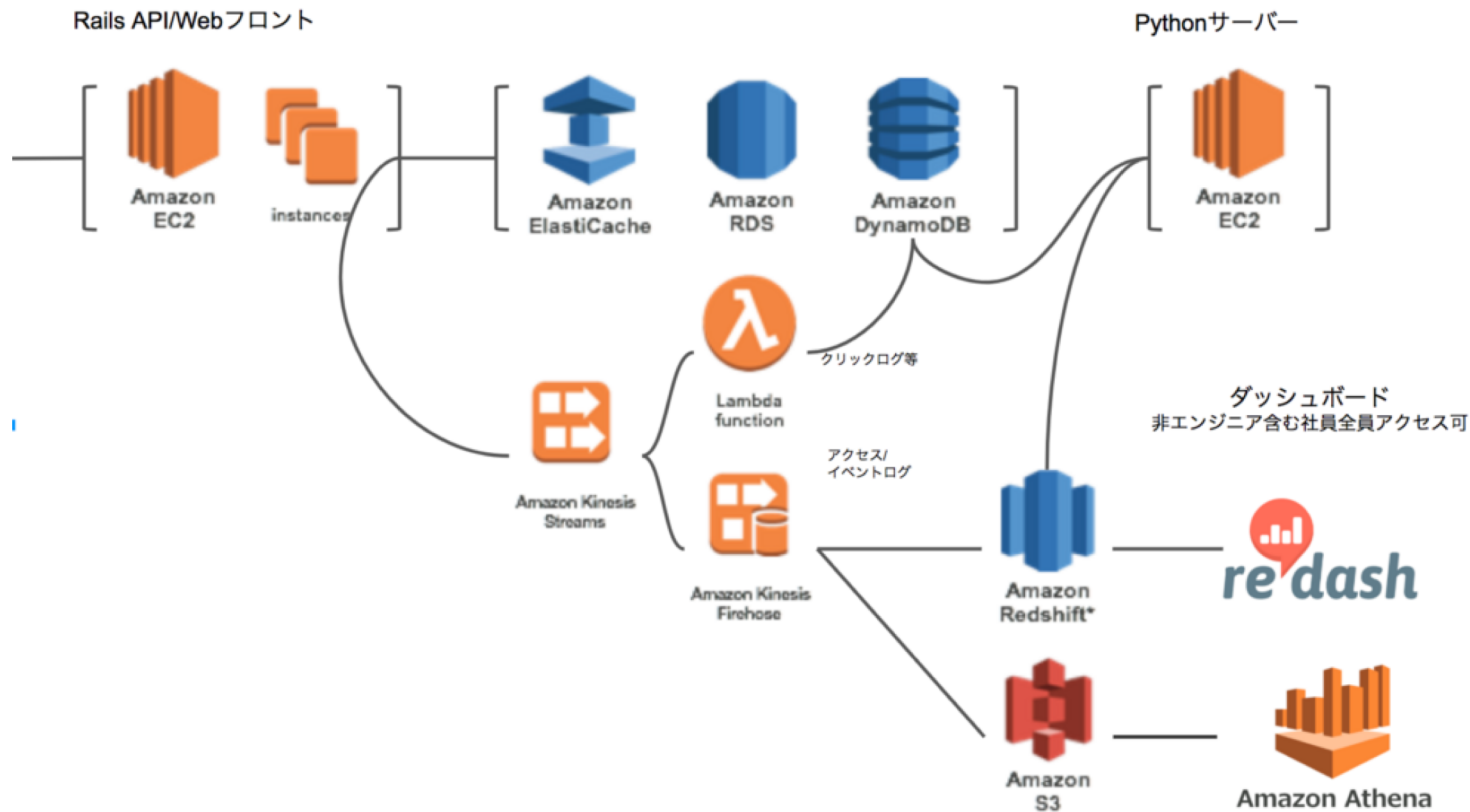
ゴール

- 何もない（メンバー2人・サーバ1台）の状態からでも、徐々にデータと向き合ってきたプロセスを共有したい
- リソースが無い中でデータに投資してきたからこそ、
現在（100万DL/口コミ40万件以上）まで来られた部分はあると思う
- スタートアップも創業当初から真面目にデータと付き合うといいことがあるはず

前提

- メインのプロダクトがアプリなので、ログの集約・分析からの改善フローが命
- 技術は凝りすぎない程度に
 - メンテが楽で、でも負債化しない程度
- データを活かす文化の醸成
 - 分析専門の人がいないと、皆の協力がないと成りたたない

全体の構成



Amazon Redshift

構成の中で最初期に導入。以下選定の際のポイント

メリット

- ・サーバーを立てる形式なので、最初期のスタートアップには安くないが価格が見積もりやすく、皆が触れるBIバックエンドとしては安心感（クエリ課金は怖い）

- ・ standardSQLが使えるので、敷居が低い

デメリット

- ・ スキーマ・インデックス等の作成・メンテナンスが必要なので、リソース不足の中ではつらい

→レコード数の増加に伴って整備してきた

Redash

メンバーが増えたタイミングでRedshiftのフロントとして導入
狙いは「皆でデータを扱う文化を作ること」

活用法

- ・皆でログ仕込む・クエリ書く（社員全員）
- ・slackのあらゆるチャンネルに垂れ流す（接点を増やす）
- ・githubのissueにもredashのクエリ貼る（定性的な議論も裏付けを）
- ・分析するだけでなくプロダクトの開発・仮説検証・振り返りに活かすための仕組みづくり

Reminder: <https://app.redash.io/appbrew/queries/73619#127760>



redash APP 4:26 AM

Masashi Okabe

直近24hのDAU (29 kB) ▾

直近24hのDAU (先週比)

DAU (先週比)

© 2018-05-28 18:46



LIPS

Kinesis Data Streams/Data Firehose

サービスが成熟してきたタイミングで導入

各インスタンスのfluentdのpluginから直接S3/Redshiftに送っていた

→ 各インスタンスのfluentdからkinesis Data Streamsに集約して、利用性を高める・透過的に扱えるように

DynamoDB

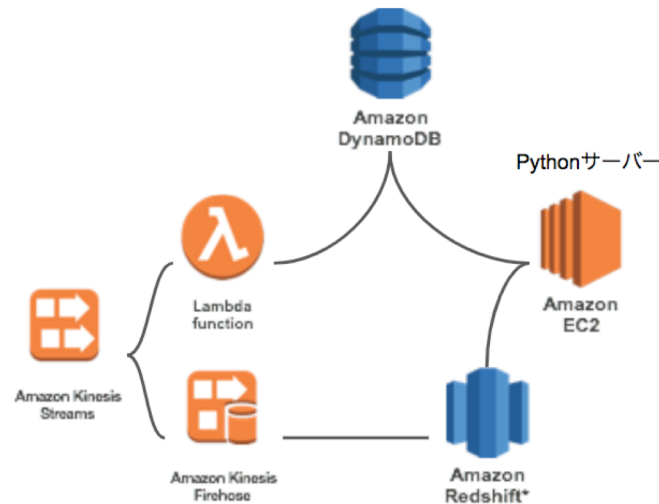
機械学習用に導入

機械学習（レコメンドなど）に、impやclickなどの**件数の多いログ(10M~/day)**をユーザーごとに保存して、計算時に即時利用するニーズ

→ Kinesis Data StreamsからLambdaを発火させ、DynamoDBのList型にAppendする形で保存

フルマネージド・スケーラビリティ・（ほぼ）即時反映などのメリット

学習結果（レコメンドするコンテンツ・ユーザー特徴量のベクタ）などもユーザーごとにDynamoDBに保存し、学習用に再利用・APIサーバーから利用



We're hiring !

データドリブンで、プロダクトに「再現性」をもったプロセスとチーム

代表をはじめエンジニア中心の組織文化

給与オープン / 全社員SO付与 / 裁量労働など、オープン・フラットを実践する組織



Bulbit 株式会社
TOP Platform Architect
塚 暁毅 氏



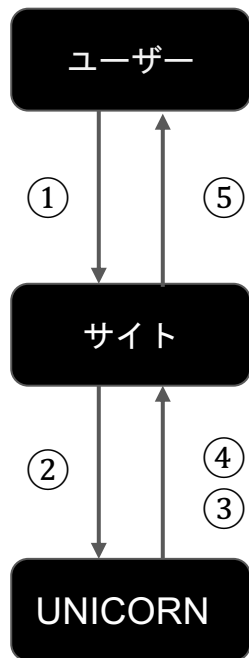
UNICORN

REDEFINE DIGITAL MARKETING

UNICORNとは？

**アプリデベロッパー向けの
Automated Marketing Platformです**

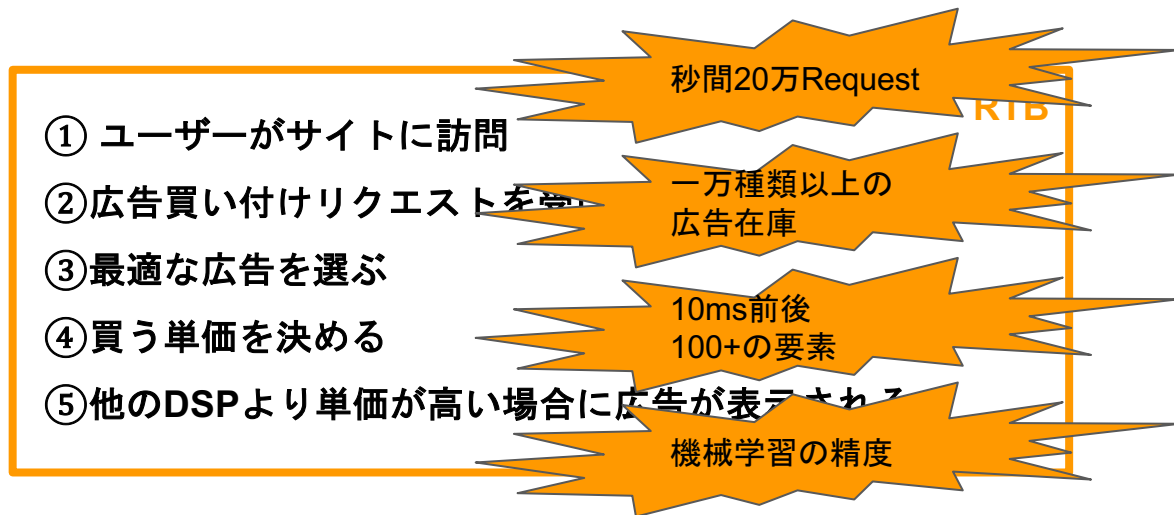
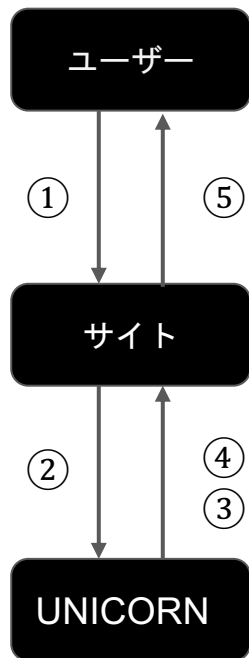
UNICORNはRTBでの広告買い付けに対応しています。



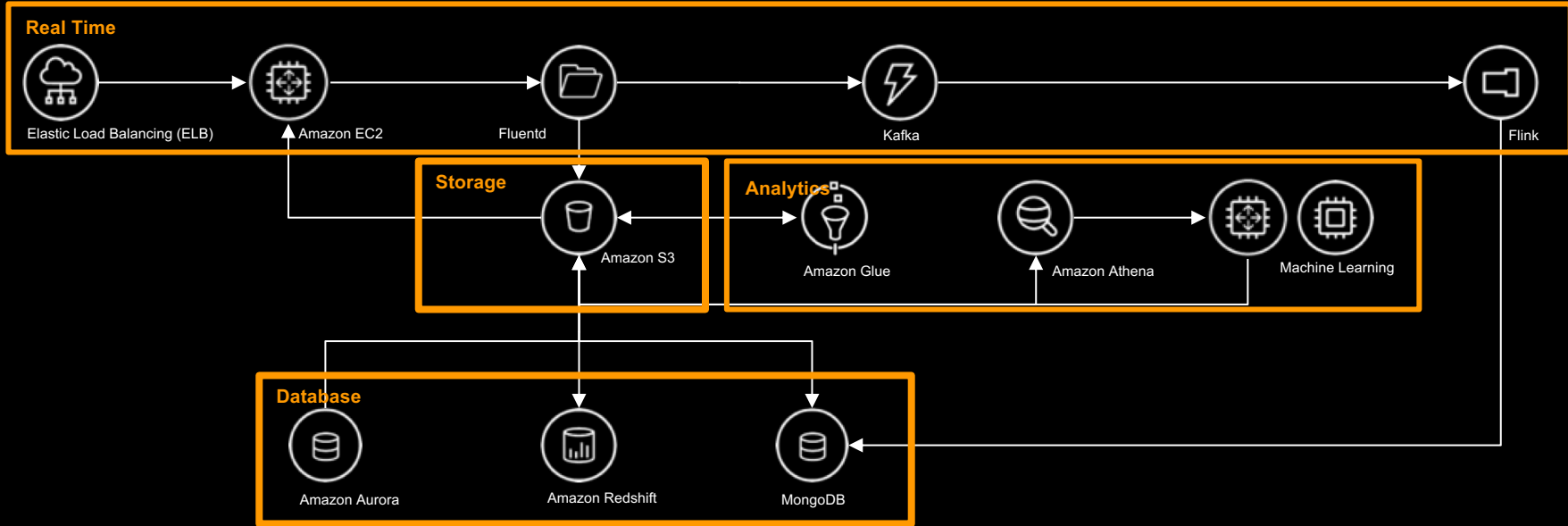
- ① ユーザーがサイトに訪問
- ② 広告買い付けリクエストを受け
- ③ 最適な広告を選ぶ
- ④ 買う単価を決める
- ⑤ 他のDSPより単価が高い場合に広告が表示される

RTB

UNICORNはRTBでの広告買い付けに対応しています。



Architecture



Architecture

Real Time

Max 200,000 QPS

Analytics

Fast and Cheaper

Storage

PB Size Data

Database

Variety

Architecture

Real Time

Max 200,000 QPS

Analytics

Fast and Cheaper

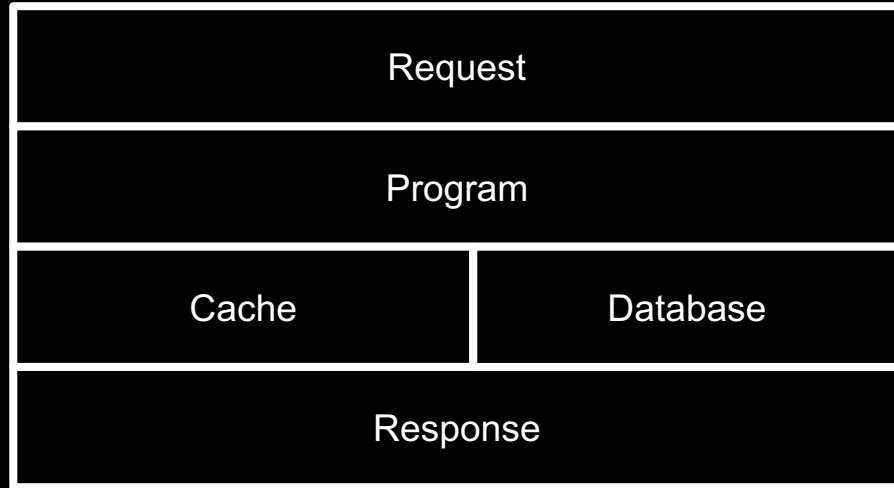
Storage

PB Size Data

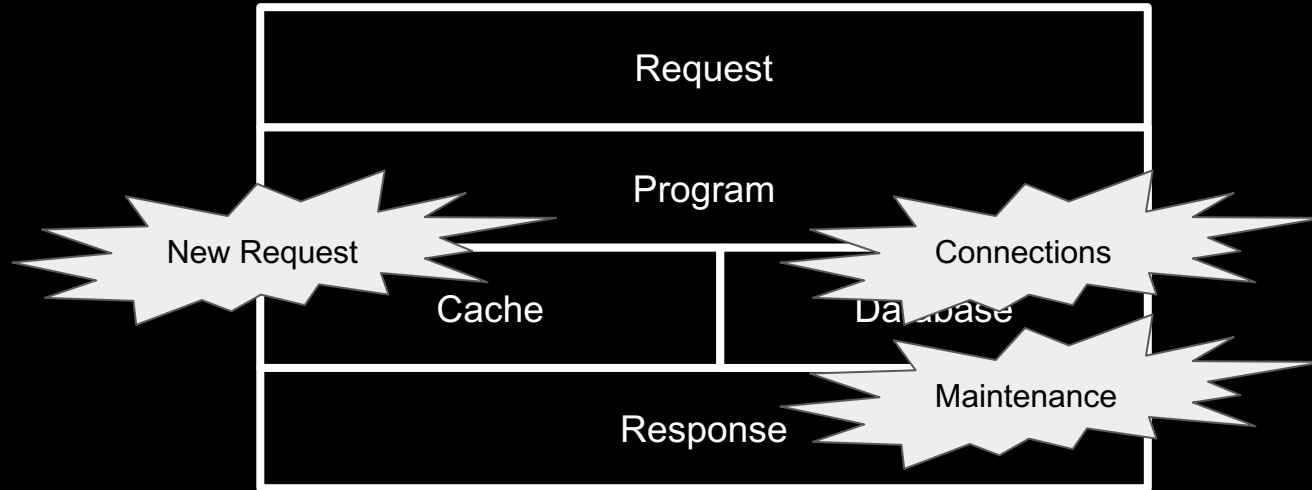
Architecture

**S3を活用してHttpRequestの処理スピードを
10ms台まで短縮しました！**

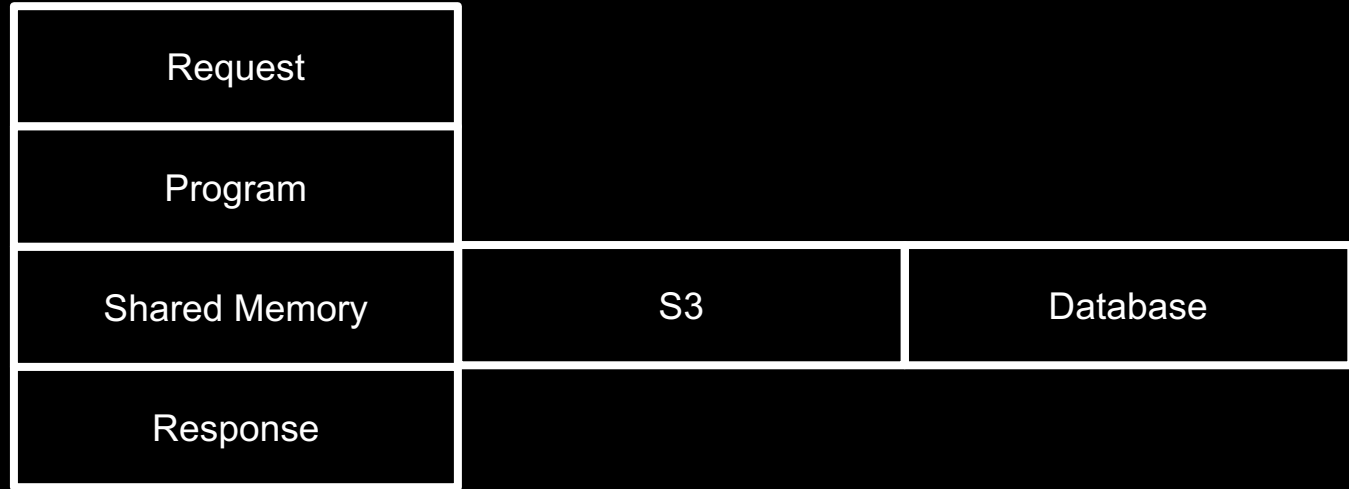
Architecture - Real Time



Architecture - Real Time

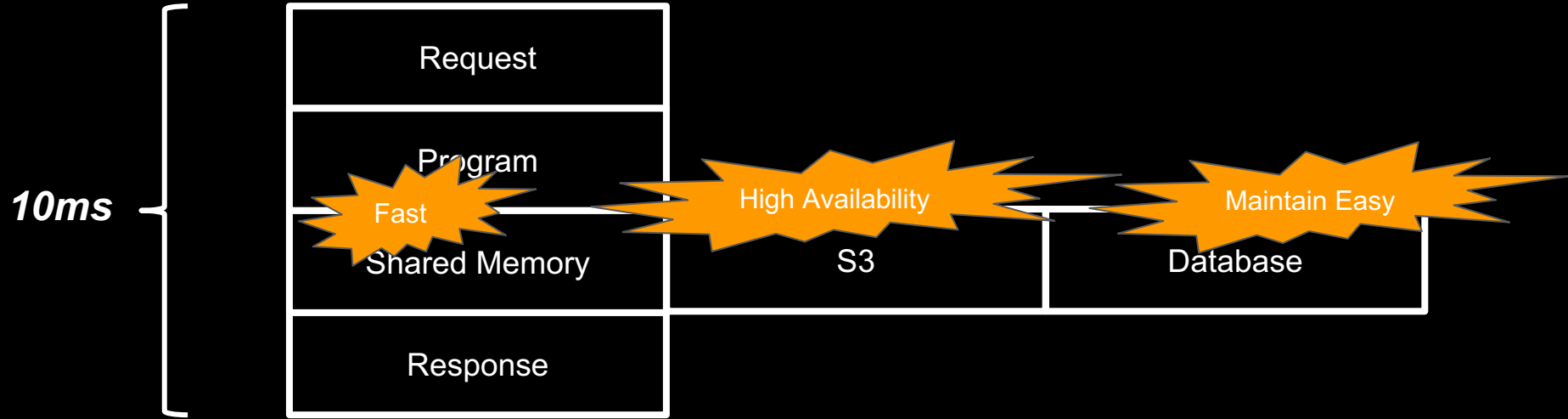


Architecture - Real Time



100ms → 10ms !

Architecture - Real Time



100ms → 10ms !

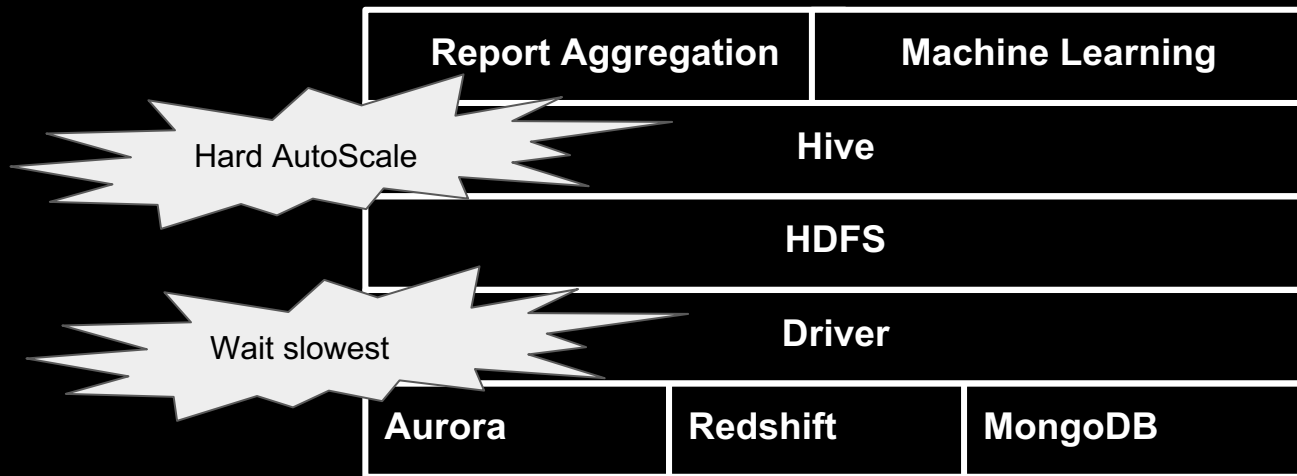
Architecture

**S3とAthenaの利用で
機械学習の高速化に成功！！！！**

Architecture - Storage

Report Aggregation		Machine Learning	
Hive			
HDFS			
Driver			
Aurora	Redshift	MongoDB	

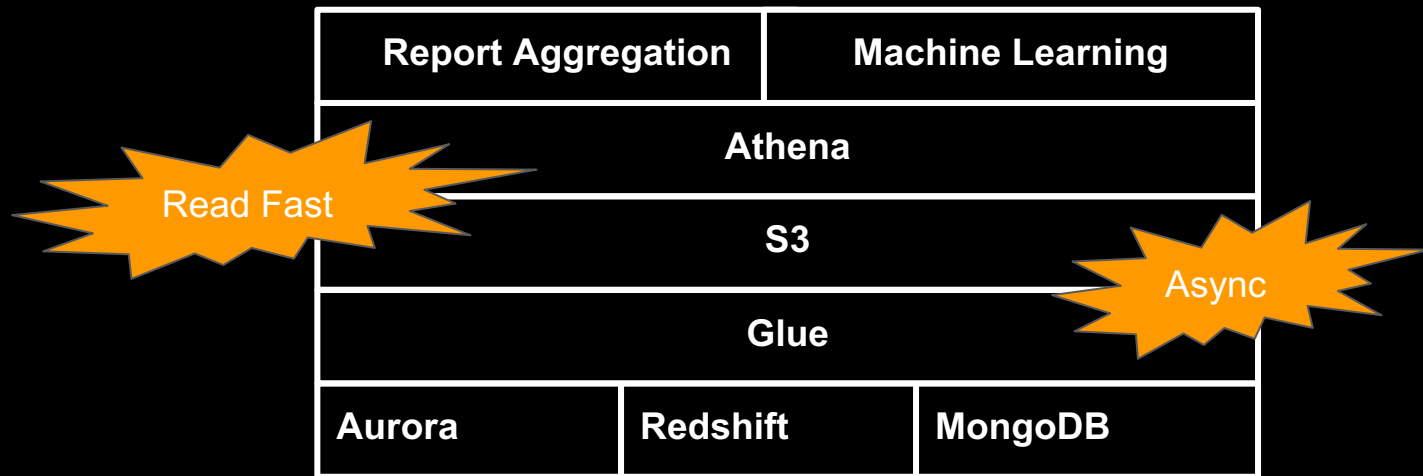
Architecture - Storage



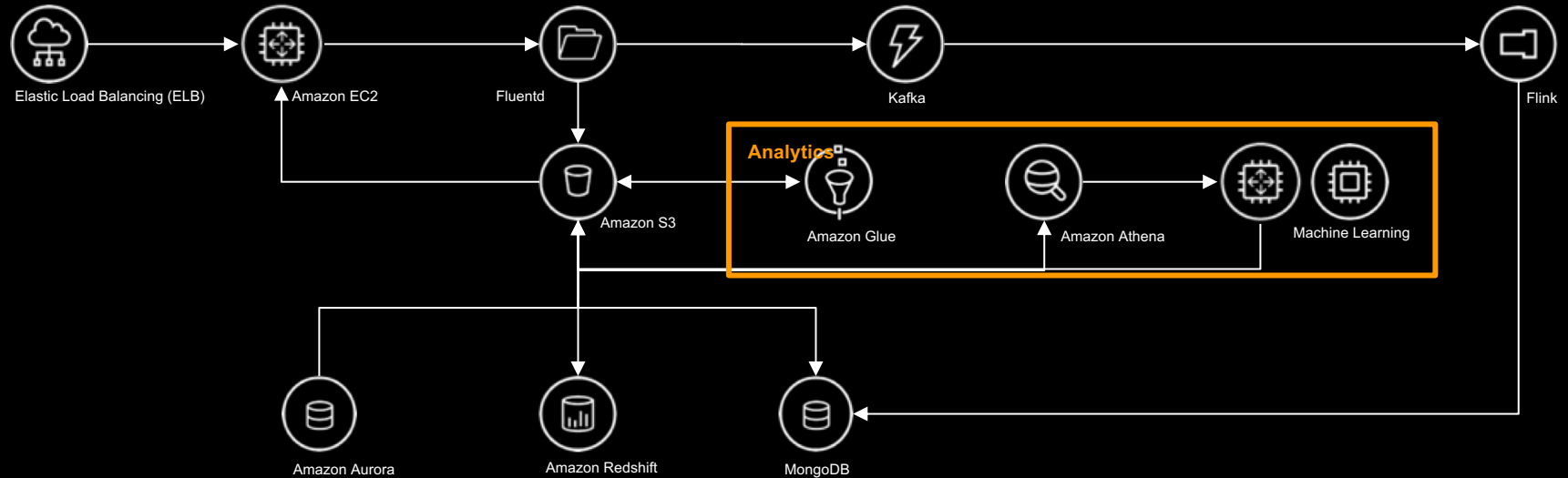
Architecture - Storage

Report Aggregation		Machine Learning	
Athena			
S3			
Glue			
Aurora	Redshift	MongoDB	

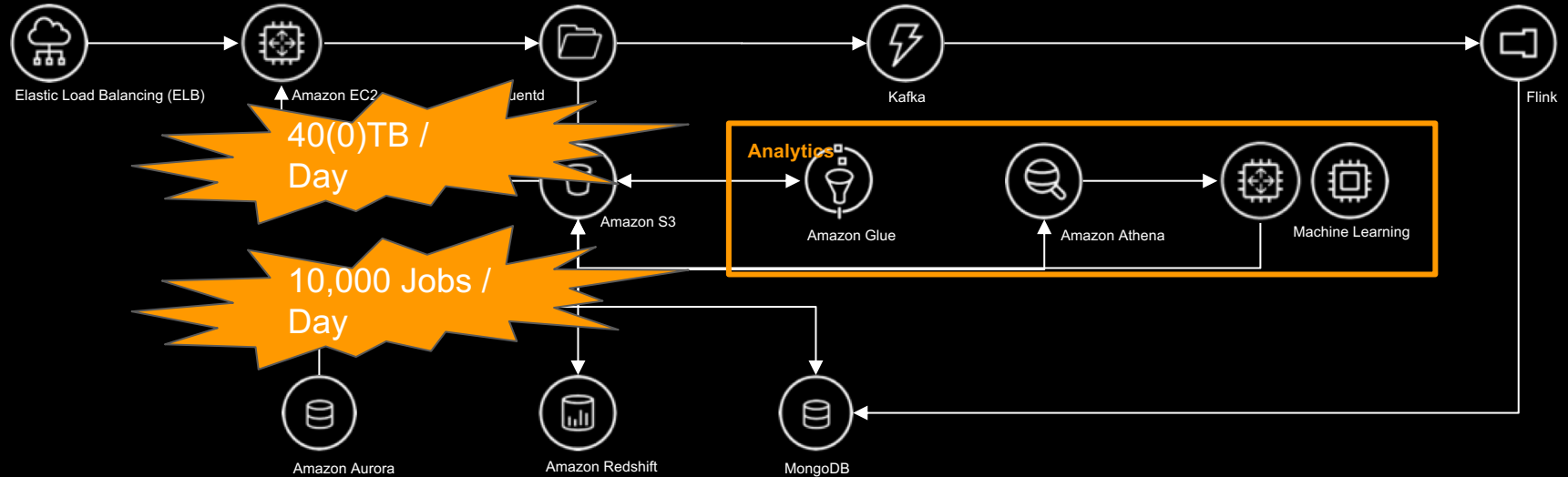
Architecture - Storage



Architecture - Analytics



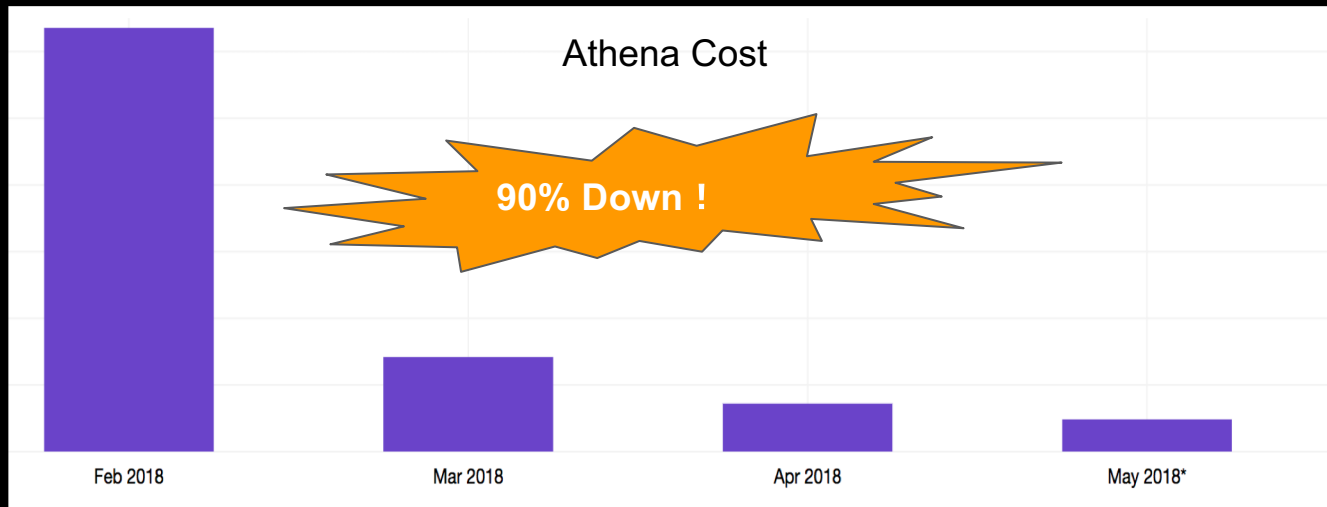
Architecture - Analytics



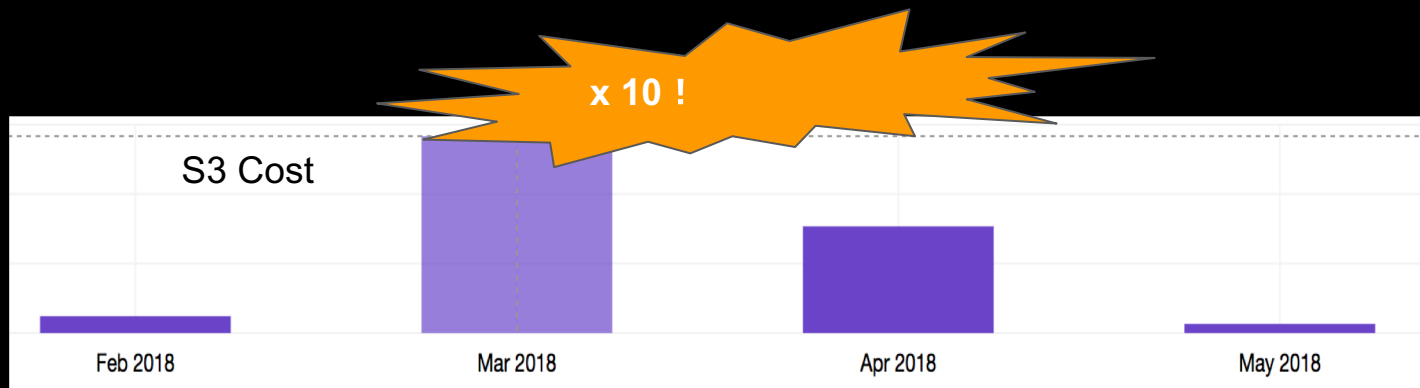
Architecture - Analytics

Parquet

Architecture - Analytics



Architecture - Analytics



JSON And Parquet

対象5ファイルに対してAthenaのSelectを実行します。

```
SELECT col_1 FROM table
```

JSONの場合

Athena Scan Data: $5\text{GB} * 5 = 25\text{GB}$

S3 Request Count: 5

Parquetの場合

Athena Scan Data: $5\text{GB} * 5 / 10 = 2.5\text{GB}$

S3 Request Count: 5

Parquet使うと
Athena Costが1/10に!

※ 5GB / file, 10 columns / row

JSON And Parquet

対象5ファイルに対してAthenaのSelectを実行します。

```
SELECT col_1 ... col_10 FROM table
```

JSONの場合

Athena Scan Data: $5\text{GB} * 5 = 25\text{GB}$

S3 Request Count: 5

Parquetの場合

Athena Scan Data: $5\text{GB} * 5 = 25\text{GB}$

S3 Request Count: $5 * 10 = 50$

Parquetのファイル数を
考慮しないとS3コスト増！

※ **5GB / file, 10 columns / row**

JSON And Parquet

**Parquetを使用する時にS3のアクセス回数も
考慮する必要があります！！！！**

Cost = Athena Scan Data + S3 Requests Count

An orange starburst shape with multiple points, containing text.

解決策：
Parquet生成する時に、
1パーティション1ファイル
に固める

チーム構成:

Engineer 5人

BizDev 3人

AWS EC2 400+台

ほぼ全日本のモバイル端末からの
広告トラフィックを処理できています。

これからGlobal展開していきます。
QPSまだまだ上がります！

ご清聴ありがとうございました

株式会社primeNumber
執行役員 エンジニア
小林 寛和 氏

スタートアップ企業が
月間 100 億レコードを捌く
データ基盤作った話

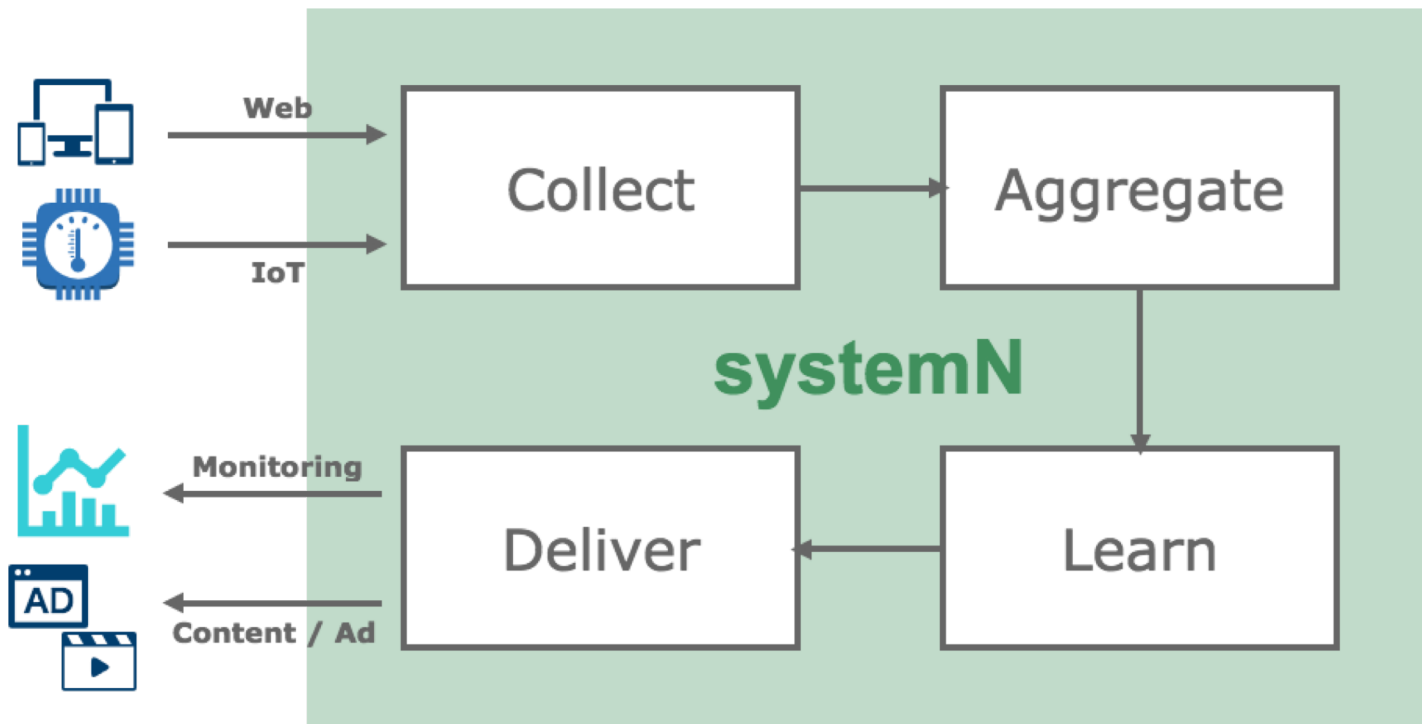
2018-06-01@AWS Summit Tokyo

自己紹介

- 小林寛和
 - @hiro-koba (GitHub)
 - @hiro_koba_jp (Twitter / Qiita)
- 株式会社primeNumber 執行役員
 - 中目黒のエンジニアリングカンパニー
 - エンジニア人数: 社員4人
 - 自社サービス「systemN」を提供



データエンジニアリングPaaS「systemN」



導入事例: 総合マーケティング企業のDMP

- 用途

- Webブラウザ・スマホアプリ上の行動ログ収集
- セグメンテーション・広告利用

- 規模感

- ログ処理量: 月間100億レコード超
- ETLジョブ実行回数: 1日500回
- Redshiftノード数: 20 nodes (dc2.large)

スタートアップ×データエンジニアリング基盤

テーマ: ETL & DWH

- ビッグデータ処理に不可欠なETLとDWH
- 準リアルタイムな分析基盤を作る想定
- スタートアップ視点での技術選定
- 本番運用した実際の声

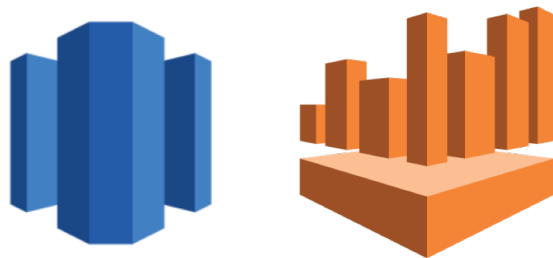
ETLで使われる技術 × AWS

- 並列分散処理FW
 - Hadoop, Sparkなど
- AWSだと
 - EMR
 - AWS Glue
 - EC2上に自前構築



DWHで使われる技術 × AWS

- MPPやカラムナを採用し、SQLによる高速集計処理
- AWSだと
 - Redshift
 - Athena / Redshift Spectrum
 - Presto on EMR



スタートアップ企業がAWSでどう実現したか

意識したこと



スモールスタート

- 小さく・すぐに・手軽な価格から始められる



スケーラビリティ

- データ量に対し、容易に・無限にスケール可能

意識したこと



低コスト

- 安い、運用・学習コストが低い
- スケールしても低コスト



できるだけ新しい技術を使おう

ETL基盤の選定

- AWS Glueを採用

最低限の機能

サーバレス

- EMRも候補に上がるが

🤔 全体的にToo much

🤔 EC2/クラスタの管理を
(まだ)したくない

	Glue	EMR	自前構築
スモールスタート		🤔	
スケーラビリティ			
運用コスト		🤔	
学習コスト			

DWHの選定

	Redshift	Redshift Spectrum	Athena	Presto on EMR
クエリ速度(~中規模)		🤔	🤔	🤔
クエリ速度(大規模)				
スモールスタート				
スケーラビリティ				🤔
学習・運用コスト	🤔	🤔		
クエリエンジン	Redshift (PostgreSQL互換)	Redshift (PostgreSQL互換)	Presto	Presto
課金形態	ノード	クエリ (+ノード)	クエリ	インスタンス+EMR

DWHの選定

- 基盤技術はRedshiftを採用

クエリ実行速度

フルマネージド

ノード課金でコストが見積もりやすい

- Redshift Spectrumとのハイブリッド構成



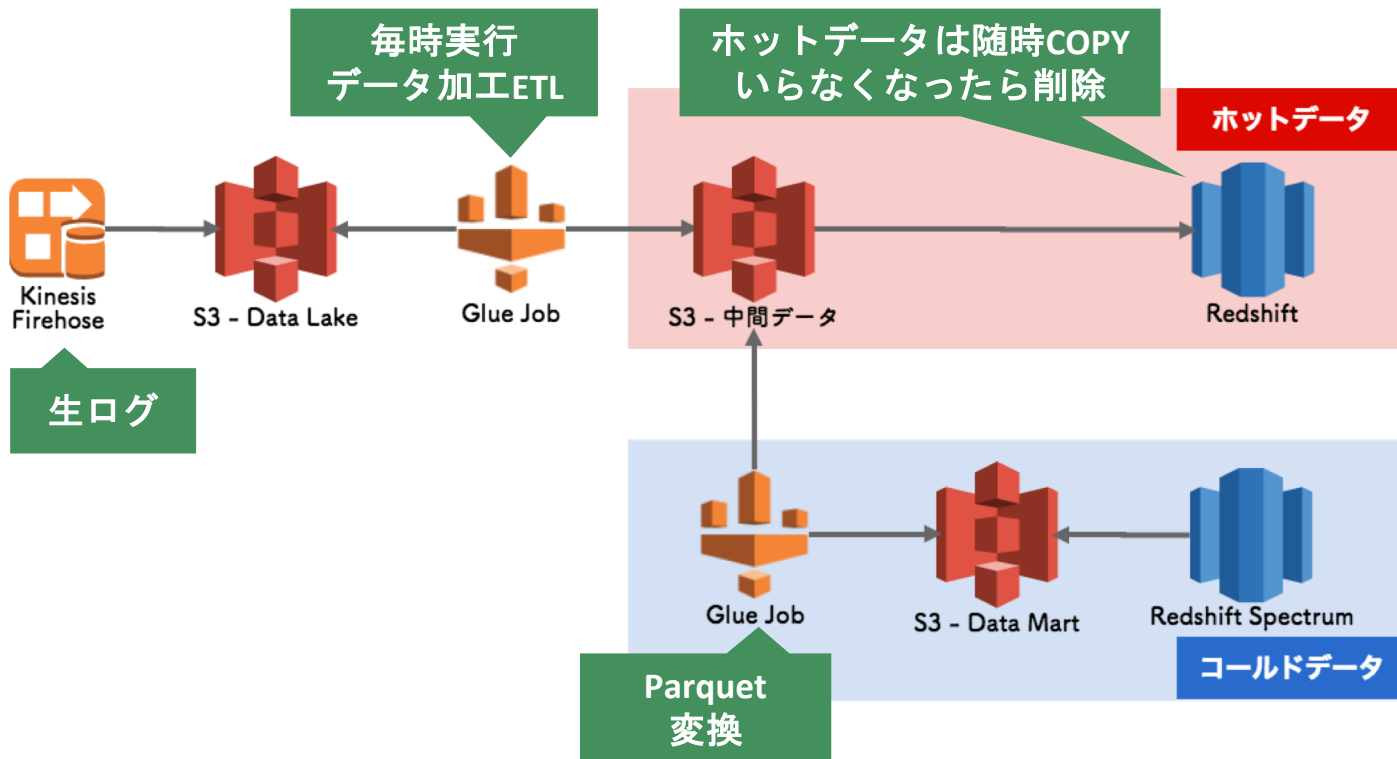
スケーラビリティと低コストの両立

Redshiftと同じI/Fで利用可

ホットデータ → Redshift

コールドデータ → S3 + Glue Data Catalog + Redshift Spectrum

基盤の全体像



実際に本番運用してみて

スモールスタートできたか

- 開発期間 = 1ヶ月
 - 2017年9月1日、リプレイスPJT始動
 - 2017年9月前半、アーキテクチャ設計
 - 2017年9月後半、実装
 - 2017年9月末日、本番リリース・完全切替
- 運用開始時のコストも、かなり抑えられた



低コストなのか

- Glue

サーバレス = 運用が楽

やれること少ない = 学習コスト低い

🤔 EMRに比べると料金は高い

- Redshift & Redshift Spectrum

Spectrumの恩恵で、料金はかなり抑えられた

🤔 安定稼働までは運用・学習コスト高い

スケーラビリティ

- どうやってスケールするか
 - ETL→GlueのDPUを上げるだけ(自動化済)
 - DWH→Redshiftのノード数を増やすだけ
- どこまでスケールできるか
 - 月間100億レコードくらいなら
余裕で捌けている



まとめ

AWS GlueとRedshiftで作るデータ基盤



GlueはSparkのコードを書くだけ



DPUやノード数を調整するだけで、容易にスケール可能



管理しているインスタンス数 = 0台



スケーラビリティと低コストの両立



より詳しい内容を、Speaker Deckで公開しています

https://speakerdeck.com/hiro_koba_jp/aws-etlji-ri-aws-gluehuo-yong-shi-li-at-primenumber



We are hiring.

primeNumberではデータエンジニアリング基盤を
一緒に育ててくれる仲間を募集中です

ご参加ありがとうございました