

AWS Summits 2014

Amazon Kinesis Deep Dive

堀 剛, ディベロップメントマネージャー, Kinesis
Amazon Web Services, Inc

大谷 晋平, 部長/ソリューションアーキテクト,
アマゾン データ サービス ジャパン株式会社

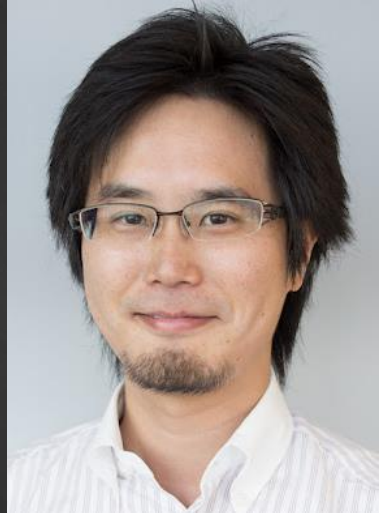


July 18, 2014



自己紹介

- 名前：大谷 晋平
 - ohitani@amazon.co.jp
 - @shot6
- 仕事：
 - お客様のクラウド活用アーキテクチャ支援
 - エマージングソリューション部
- 担当技術分野
 - ビッグデータ、データベース、分散システム



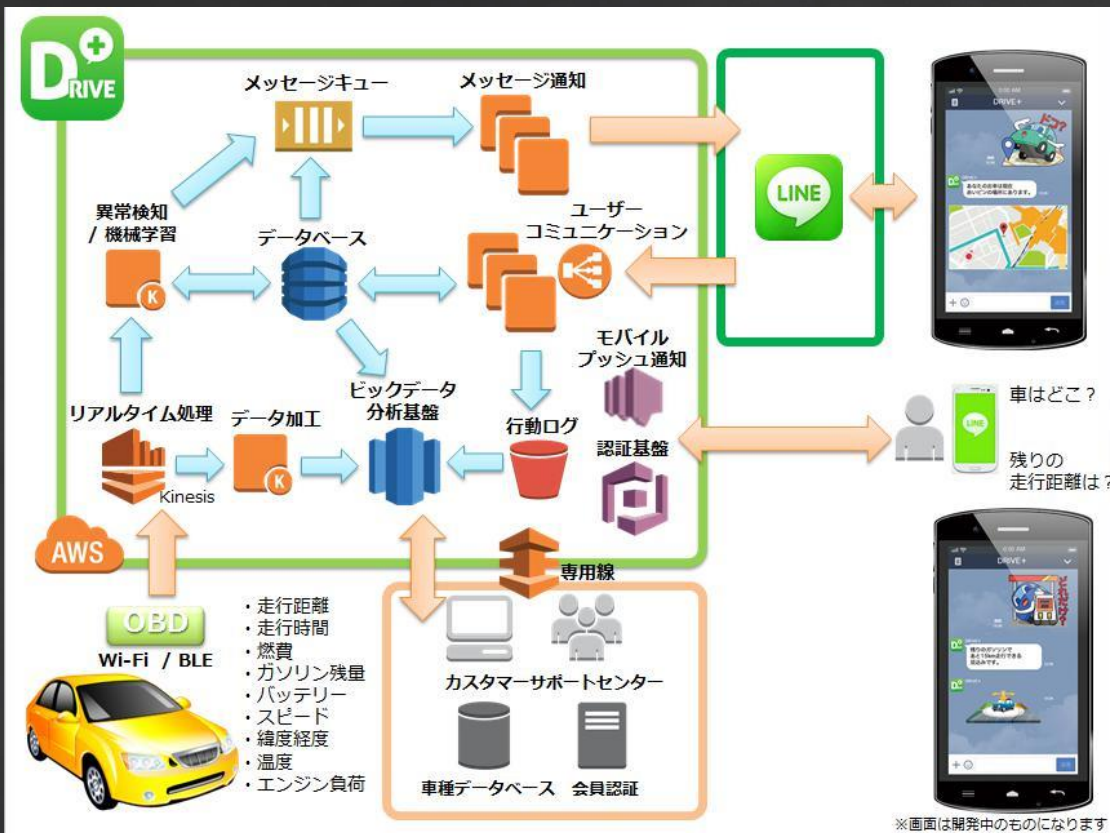
Kinesisの新規ロウンチ

- 東京リージョンロウンチ！
- Fluentdプラグイン
 - <https://github.com/aws-labs/aws-fluent-plugin-kinesis>
- MQTTアダプター
 - <https://github.com/aws-labs/mqtt-kinesis-bridge>



事例紹介

日本の事例(ガリバー様Drive+)



※画面は開発中のものになります



日本の事例(Pencil様)

- Kinesisを利用したリアルタイムユーザーモニター
- Kinesisで構想が現実に。開発にかかった時間は1/4以下
 - <http://www.pencil.co.jp/newsrelease/2014070200001.html>

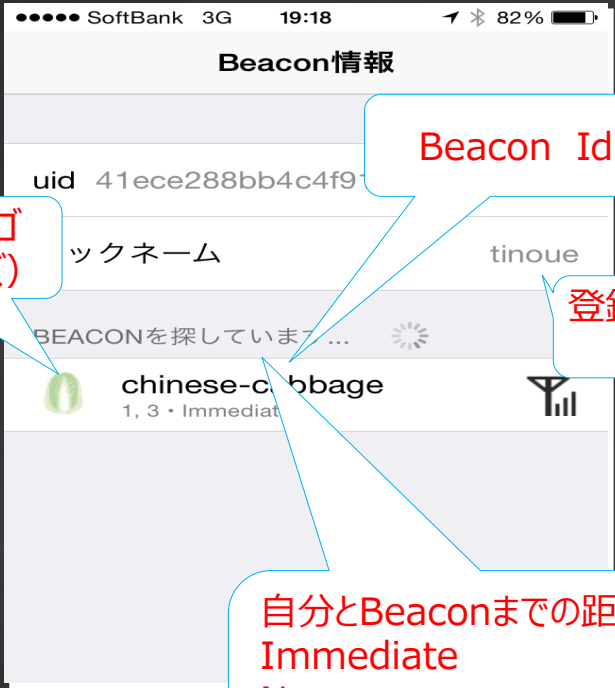


SmartInsight様事例

設置するBeacon (MyBeacon Pro MB004 Aplix社製)



Beaconのロゴ
(野菜シリーズ)



Beacon Id

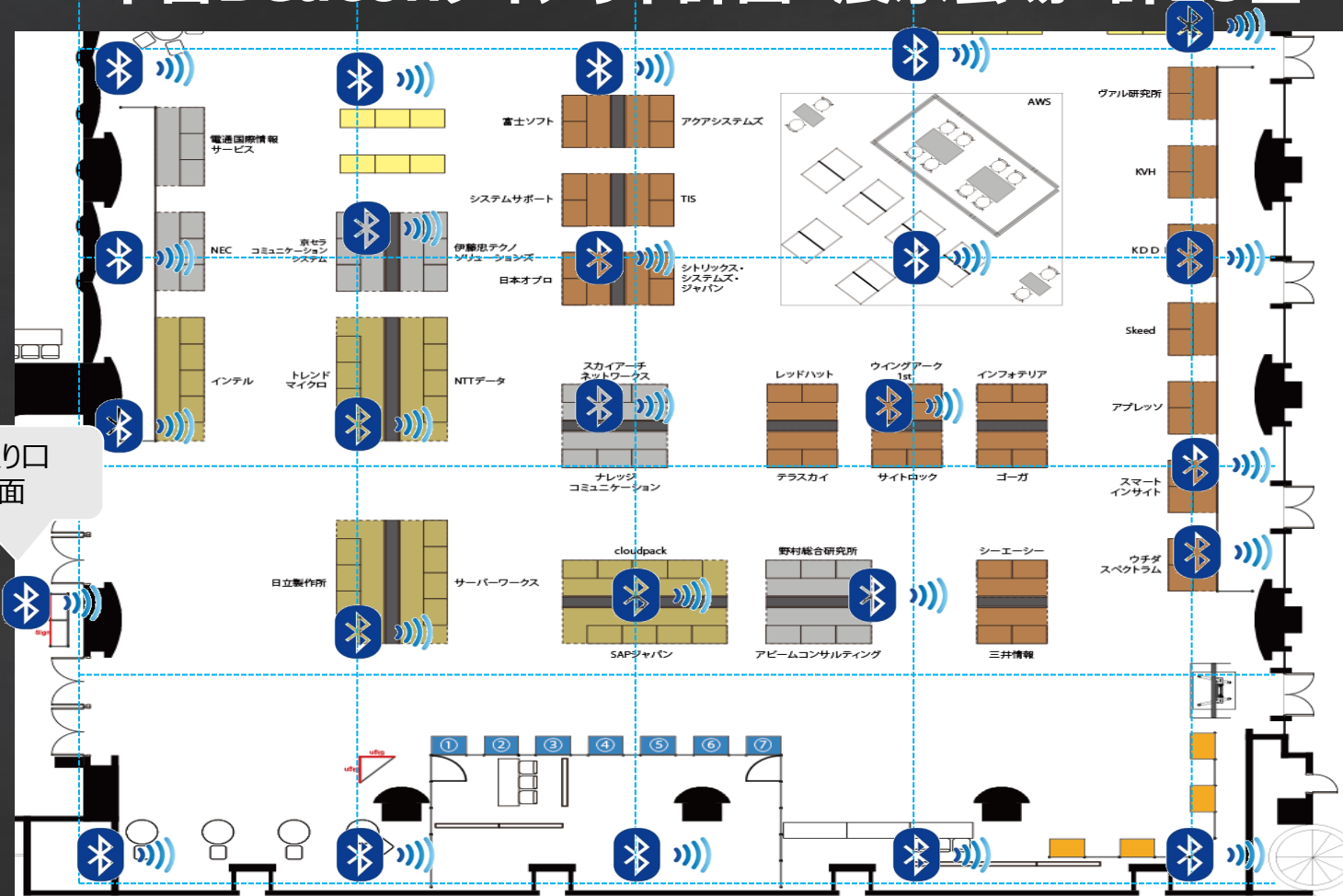
登録時に指定した
ニックネーム

自分とBeaconまでの距離

| | |
|-----------|-------|
| Immediate | <50cm |
| Near | <3m |
| Far | <10m |
| Unknown | >それ以上 |



本番Beaconレイアウト計画 展示会場 計25基



展示会場入り口
案内版裏面



会場内を歩き回ると

SMART/InSight - Internet Explorer

http://localhost:8080/InsightG2_20140701/user/ SMART/InSight

insightadmin | Administration | Log out

Beacon

Logger: Ponno1e Expand

DrillDown Navigator

Beacon(30)

Show Legend

More >> All

ニックネーム(13)

Hide Legend

ootomo
140711
iphone5
lime
yellow
Others

More >> All

Beacon距離(4)

Hide Legend

Far
Unknown
Immediate
Near

HeatMapTp3

Search Option

-- Select condition --

文書更新日時(年月日)-20140714(Match...
major-1(Match all words)

SoftBank 3G 18:45 87%

Grid style Search Result(details)

Showing 1 of 10 pages. Total results: 9,075

| ニックネーム | Beacon | Beacon距離 | インデックス作成日時 |
|--------|------------|----------|----------------------|
| ootomo | Eggplant | Far | 2014-07-14T15:36:59Z |
| ootomo | Red Pepper | Far | 2014-07-14T15:36:59Z |
| ootomo | Grape | Far | 2014-07-14T15:36:59Z |
| ootomo | Onion | Far | 2014-07-14T15:36:59Z |
| ootomo | Asparagus | Far | 2014-07-14T15:36:59Z |
| ootomo | Banana | Far | 2014-07-14T15:36:59Z |
| ootomo | Kiwi | Far | 2014-07-14T15:36:59Z |

SMART/InSight

スマートサイト株式会社が平成26年7月17日(木)~18日(金) 両日開催されるAWS Summit Tokyo 2014 展示会場にて出展する「SMART/InSight Cloud RealtimeM2M Analytics(仮称)」専用スマートフォンアプリケーションです。

本アプリケーションにより会場に設置されたApple Beacon対応BLE (Bluetooth Low Energy)ベースのBeaconモジュールからの電波をiPhoneが検知し、各Beaconモジュールからのユーザの位置状況や個別Beacon位置での滞在時間等の情報をリアルタイムにクラウド基盤に送信、「SMART/InSight Cloud Realtime M2M Analytics」で可視化分析する事が出来ます。

本アプリケーションのご利用にはメールアドレス、ニックネームの登録が必要です。

Beacon情報

radish
1, 2 • Near

turnip
1, 6 • Near

chinese-cabbage
1, 3 • Near

cucumber
1, 7 • Near

carrot
1, 10 • Near

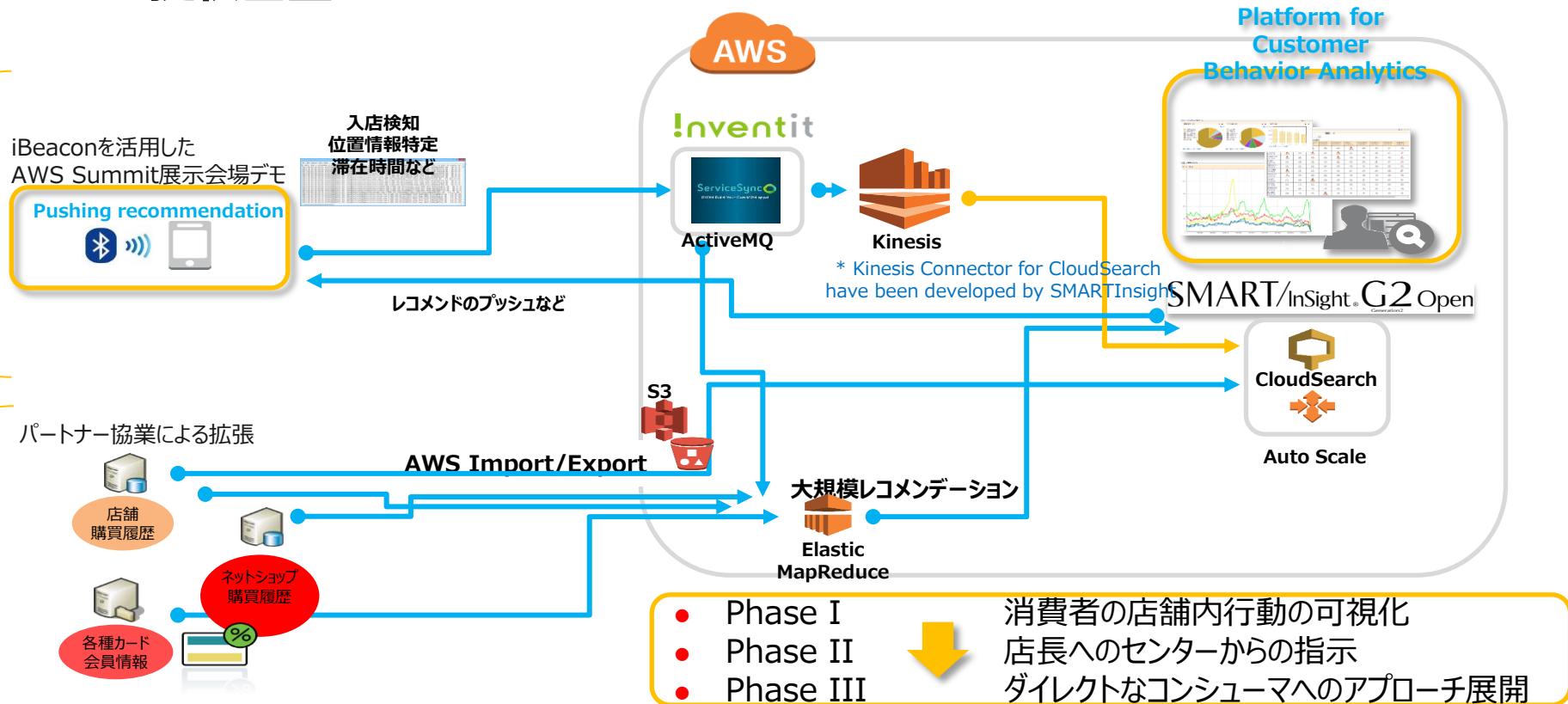
eggplant
1, 8 • Near

chinese-cabbage
1, 4 • Near

pumpkin
4, 5 • Near

AWS Summit展示会場デモシステム構成

～単純な動向可視化からLocation&Context Awareなサービス提供基盤へ～



導入事例



Amazon Kinesisにゲームエンジンサーバーから送られる何百ものゲームデータをリアルタイムで送り、ビジネスに不可欠な分析やダッシュボードアプリケーションを遅れることなく利用



データパイプラインとレポートのインフラだった、既存のバッチに依存したデータ取得および収集のメカニズムをAmazon Kinesisに置き換えることにより、システム運用のリソースが削減できた



自己紹介

- 名前：堀 剛
- 仕事：
 - Amazon Kinesisの開発マネージャー
 - 前職は測定サービス開発マネージャー
- 1番好きなサービスは。。。：???



Amazon Kinesis

ストリーミングデータを リアルタイムで収集し処理するマネージドサービス

- なぜリアルタイム？
- Kinesis概要
- データ入力
- データ取得と処理
- 拡張性



なぜリアルタイム？

測定(Metering) サービス

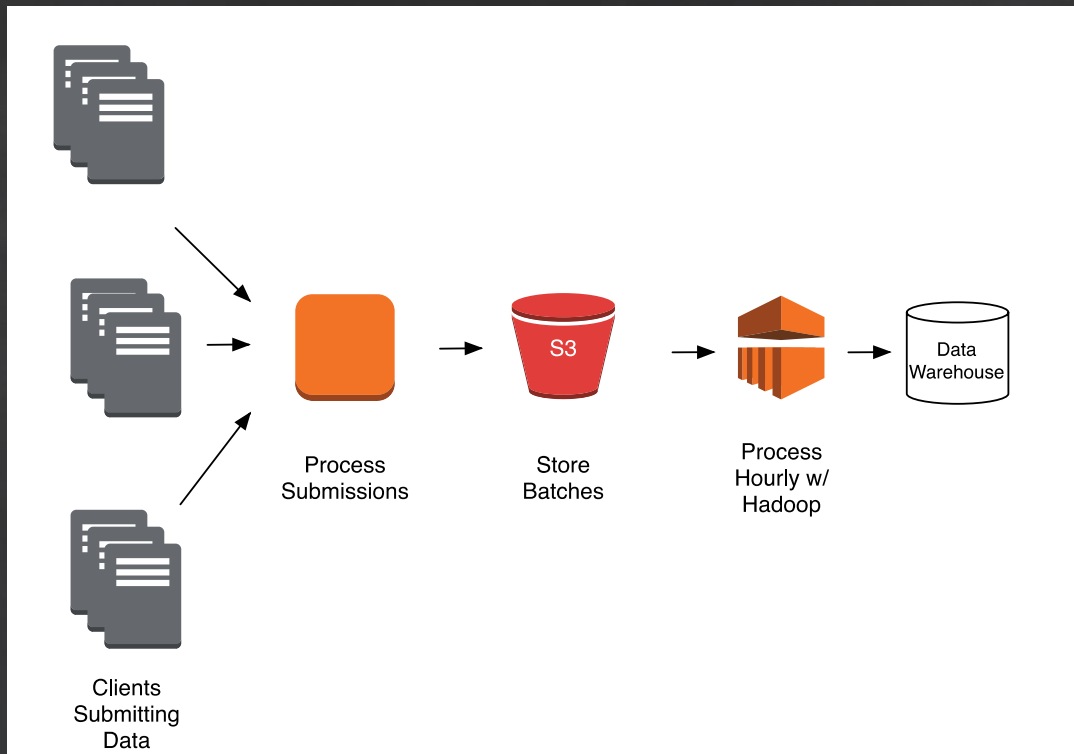


AWSでのデータ量

- 測定サービス
 - 毎秒数千万レコード
 - 毎時数テラバイト
 - 数十万のデータソース
 - 月末にはオーディターでの100%の正確性が必要
- データウェアハウス
 - 毎時数百万のファイル
 - 毎日100以上のextract-transform-load (ETL) jobs
 - 毎日数百のユーザーによる数百のクエリ



AWS 測定サービス



- ワークロード
 - 数千万レコード / 秒
 - 数テラバイト / 時間
 - 何十万ものデータソース
 - 100%の正確性
- 課題
 - スケールの問題
 - リアルタイムの要望
 - 高い運用コスト



要求の変化

従来の要求

- 毎時か毎日の大量なデータ処理

新しい要求

- リアルタイム, 早い意思決定
- “keep everything”
- エラスティックな拡張性
- 複数の目的に応じて同じデータを並行処理



Amazon Kinesis – An Overview

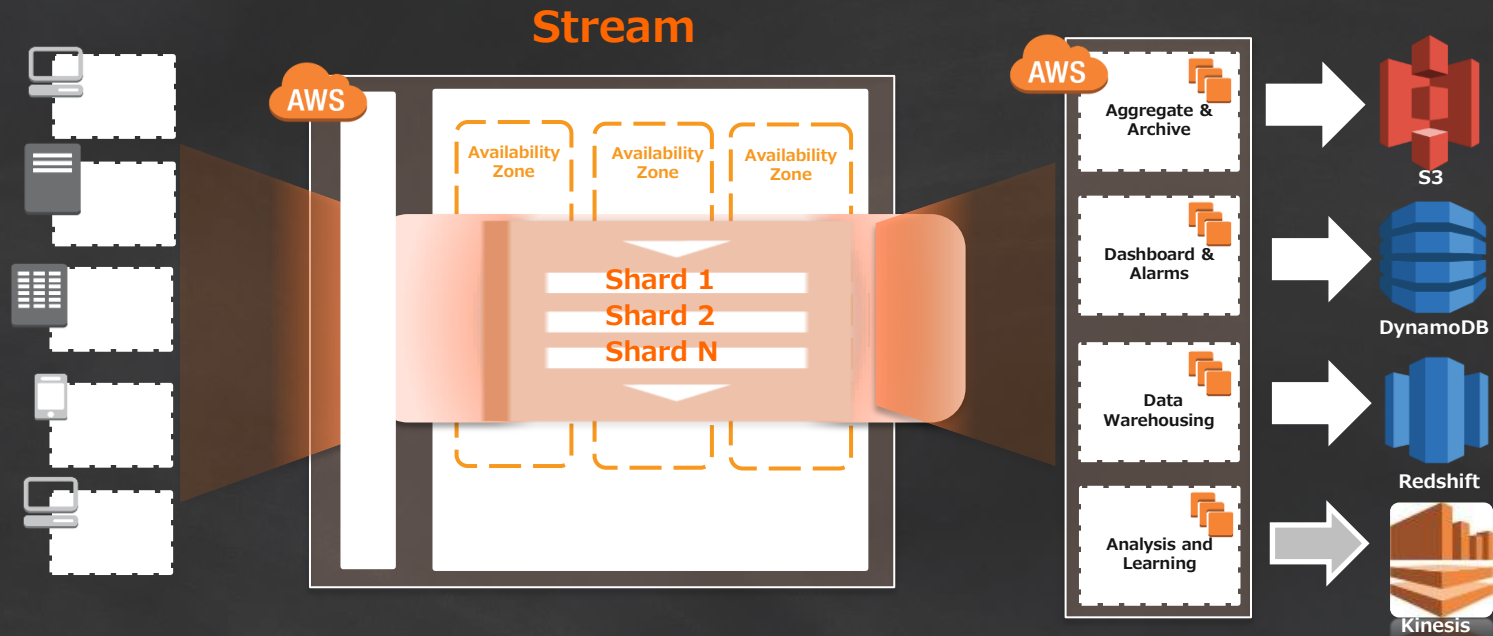


Amazon Kinesisとは？

- 大規模なストリーミングデータをリアルタイムで収集し処理するフルマネージド型サービス
- 1時間あたり数十万のソースから数百TBのデータを収集、処理することができる
- データは複数のアベイラビリティゾーンに格納され高い信頼性と耐久性をもつ



Kinesis構成内容



- 用途単位で**Stream**を作成し、Streamは1つ以上の**Shard**で構成される
- Shardは、入力側 秒間1MB, 1000TPS、データ処理側 秒間2 MB, 5TPSのキャパシティを持つ
- 入力されたデータは、複数のAZに24時間保管される
- Shardの増減によってスケールの制御が可能



データ入力、収集、処理



データ入力側

データ入力側

データ取得と処理

HTTPS Post



AWS SDK



Fluentd



Flume



LOG4J



Get* APIs

Kinesis Client Library
+
Connector Library



Apache
Storm

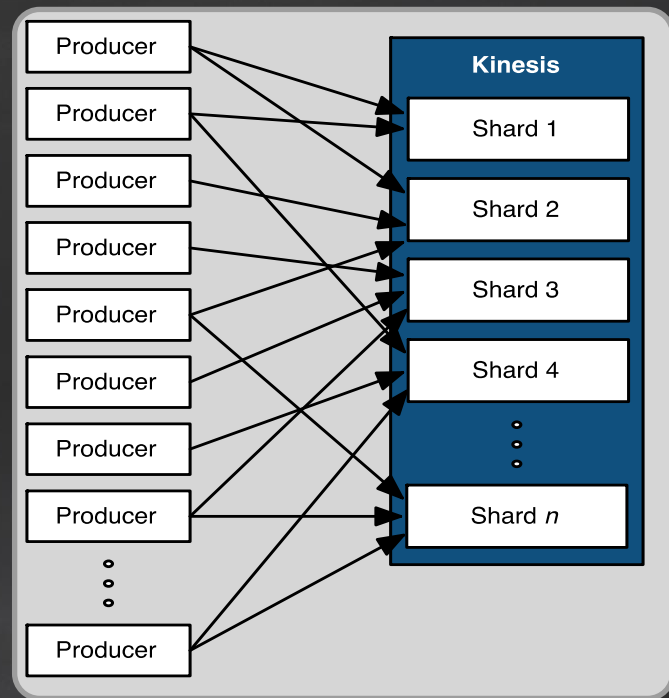


Amazon Elastic
MapReduce



データ入力方法

- プロデューサーは**PutRecord**を使ってストリームにデータを入力する
- **PutRecord** (base64-encoded-data, StreamName, PartitionKey)
- **パーティションキー**を基に**shard**に分配
- 成功すると**シーケンス番号**と**shard番号**が返される
- データサイズは最大50KB



PutRecord

AWS CLI を利用した例 :

```
$ aws kinesism put-record \  
--stream-name StreamName --data 'foo' --partition-key $RANDOM
```

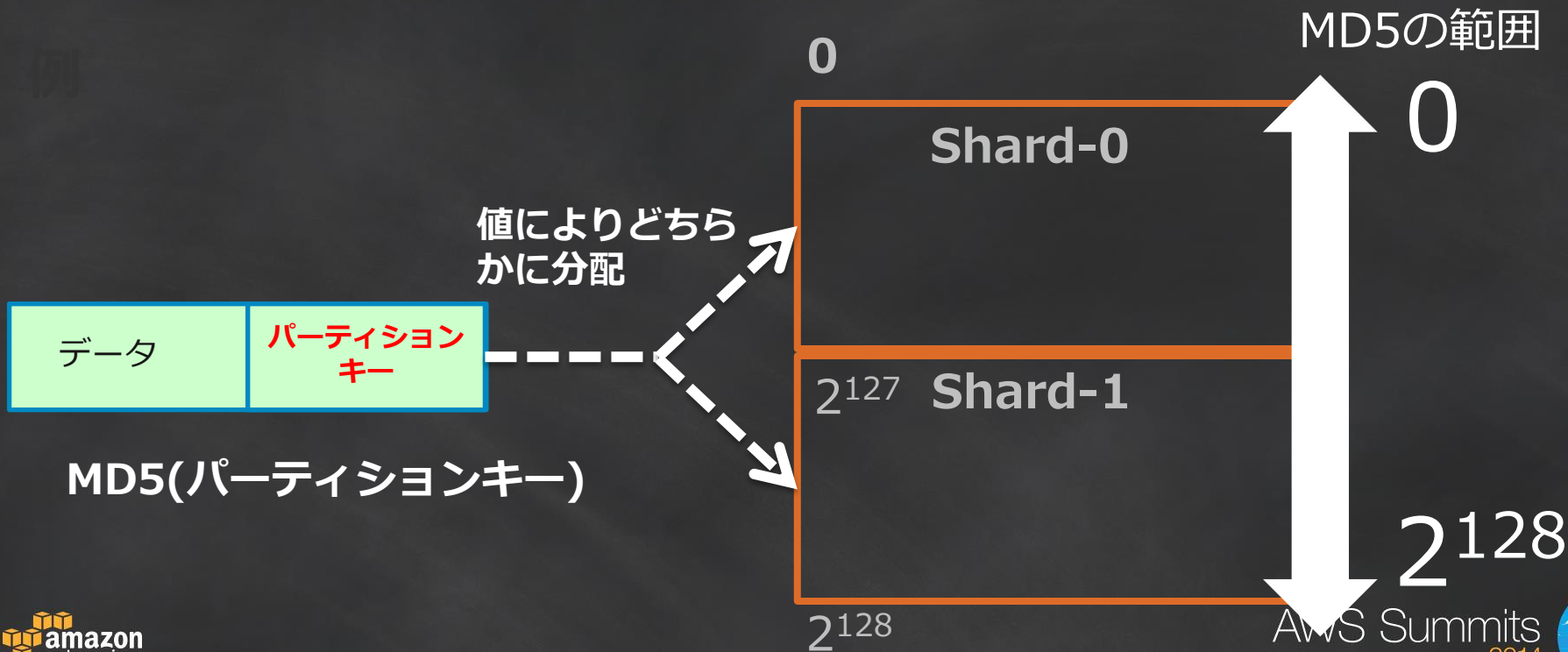
レスポンス

```
{  
  "ShardId": "shardId-0000000000013",  
  "SequenceNumber":  
  "49541296383533603670305612607160966548683674396982771921"  
}
```



Shardへの分配ロジック

Shardは担当するレンジを持ち、パーティションキーをMD5でハッシュ化した値によって該当のShardに分配される



パーティションキーについてのTips

- パーティションキーを基にshardに分配される
- Shardにはキャパシティがある

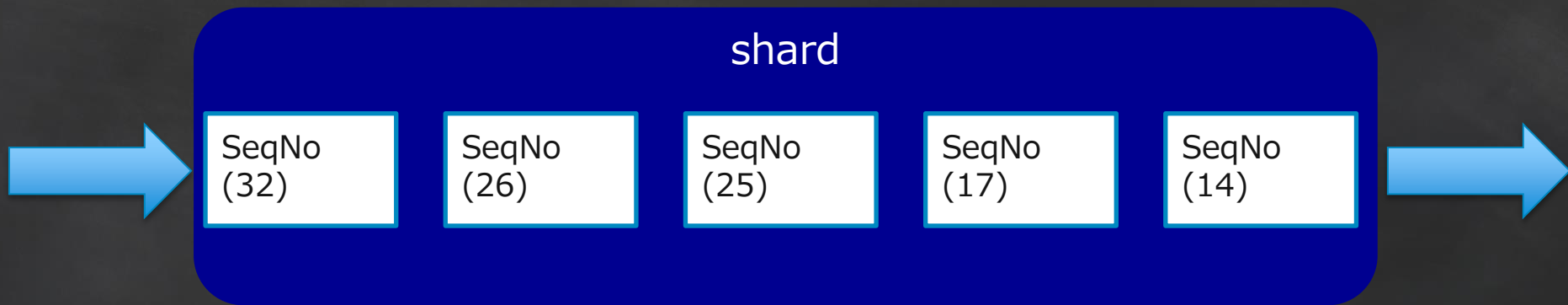


- パーティションキーの数 > shardの数
- カーディナリティーの高いパーティションキーを使う



シーケンス番号

- KinesisがStream内でユニークなシーケンス番号を付与
- データもシーケンス番号も不変
- シーケンス番号でデータが何回でも取得できる（24時間以内）
- 何度取得してもシーケンス番号の順番はかわらない



データの取得と処理

データ入力側

データ取得と処理

HTTPS Post



AWS SDK



Fluentd



Flume



LOG4J



Get* APIs

Kinesis Client
Library

+

Connector Library

Apache
Storm



Amazon Elastic
MapReduce



AWS CLIによるデータの取得方法

AWS CLI を利用した例 :

```
$ aws kinesis get-shard-iterator --stream-name StreamName \  
--shard-id shardId-000000000013 --shard-iterator-type AT_SEQUENCE_NUMBER \  
--starting-sequence-number 49541296383533603670305612607160966548683674396982771921  
{  
  "ShardIterator": "Fakelterator"  
}  
$ aws kinesis get-records --shard-iterator Fakelterator --limit 1  
{  
  "Records": [  
    {  
      "PartitionKey": "16772",  
      "Data": "Zm9v",  
      "SequenceNumber": "49541296383533603670305612607160966548683674396982771921"  
    }  
  ],  
  "NextShardIterator": "YetAnotherFakelterator"  
}
```

get-shard-iterator

get-records



どうやって信頼性と拡張性の
あるアプリケーションをつくるの？

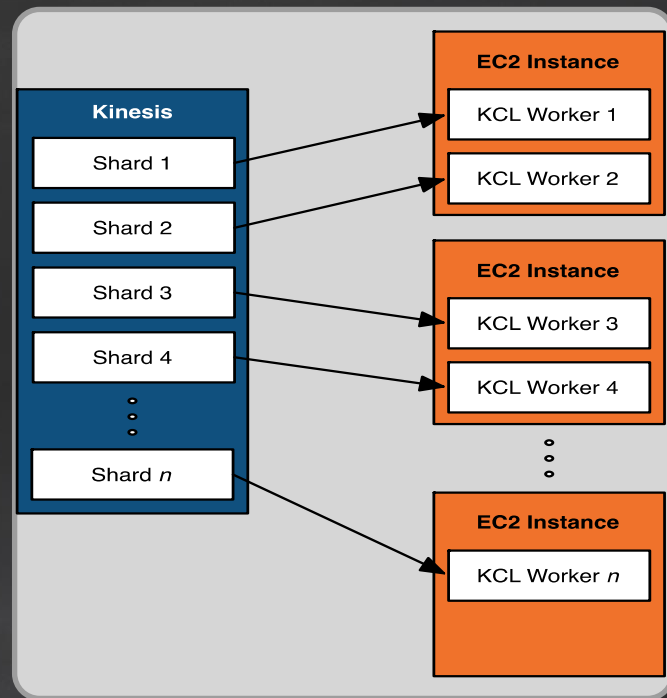


Kinesis Client Library (KCL)

Client library for fault-tolerant, at least-once, Continuous Processing

- Shardと同じ数のWorker
- Workerを均等にロードバランシング
- 障害感知と新しいWorkerの立ち上げ
- シャードの数に応じてworkerが動作する
- AutoScalingでエラスティック
- チェックポイントとAt least once処理

これらの煩雑な処理を意識することなく
ビジネスロジックに集中することができる。



Sample RecordProcessor

```
public class SampleRecordProcessor implements IRecordProcessor {
    @Override
    public void initialize(String shardId) {
        LOG.info("Initializing record processor for shard: " + shardId);
        // initialize to start processing records.
        initializeLocalState(shardId);
    }

    @Override
    public void processRecords(List<Record> records, IRecordProcessorCheckpointter checkpointer) {
        LOG.info("Processing " + records.size() + " records for kinesisShardId " + kinesisShardId);

        // Process records and perform all exception handling.
        processRecordsWithRetries(records);

        // Checkpoint once every checkpoint interval.
        if (System.currentTimeMillis() > nextCheckpointTimeInMillis) {
            checkpoint(checkpointer);
            nextCheckpointTimeInMillis = System.currentTimeMillis() + CHECKPOINT_INTERVAL_MILLIS;
        }
    }
}
```

重複データについてのTips

- ネットワーク障害や500レベルエラーでプロデューサーはリトライする
- リカバリーやロードバランシングでデータが最後のチェックポイントからリプレイされる



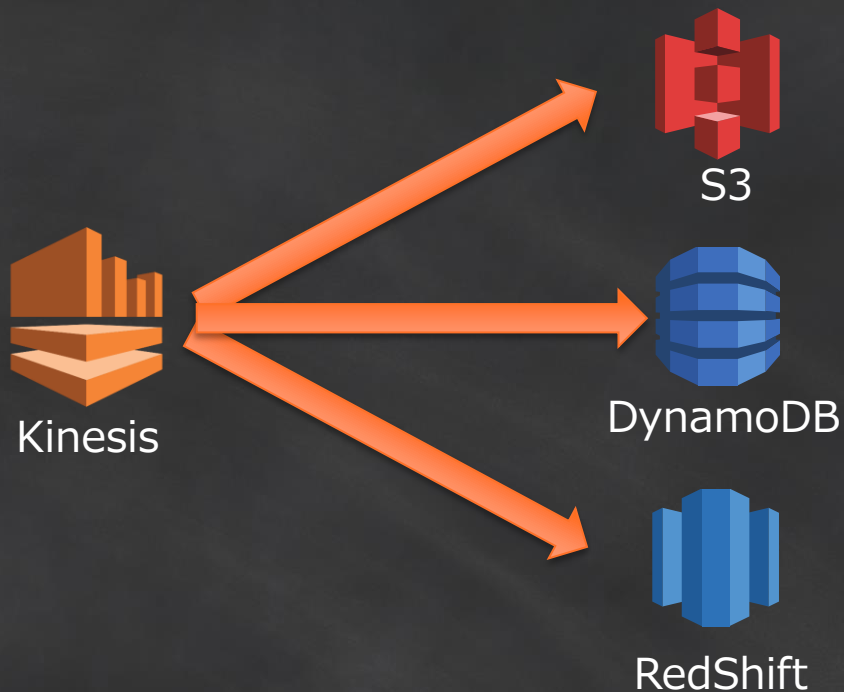
ユースケースを理解する

- べき等なアプリをつくる
- 重複を許容する



Kinesis Connector Library

S3, DynamoDB、Redshiftといった他のサービスとのインテグレーションが容易



他のデータ処理のために保存

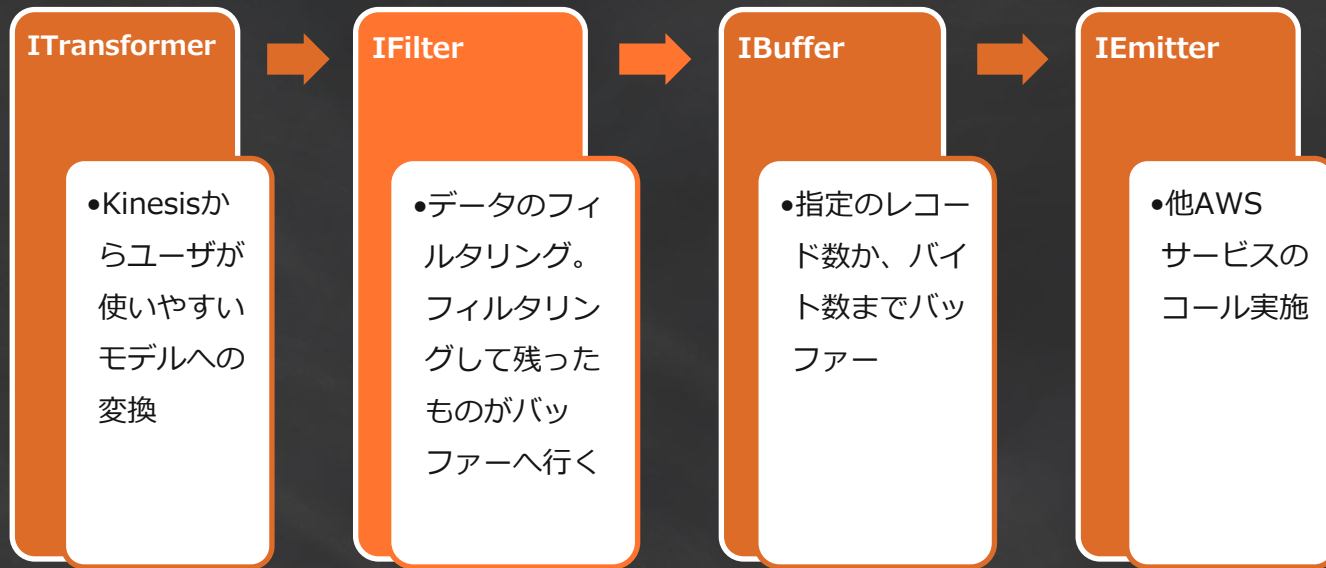
リアルタイムダッシュボード
やランキングなどに利用

蓄積された過去のデータを用
いた多角的な分析に利用



Kinesis Connector Library

- データの取得、変換、フィルタ、バッファ、書き出しを簡単に実装できる。



S3



DynamoDB



Redshift

アプリの拡張性についてのTips

- 一つのデータをいろいろなシステムで使いたい
- プロデューサーの負荷を軽減したい
- データの一貫性を保ちたい



- 📦 Kinesisにすべてのデータを1回入力する
- 📦 必要に応じて新しいアプリを追加していく
- 📦 Agilityを高める



Right Tools for the Right Job

データ入力側

データ取得と処理

HTTPS Post



AWS SDK



Fluentd



Flume



LOG4J



Get* APIs

Kinesis Client Library



+ Connector Library

Apache Storm



Amazon Elastic MapReduce

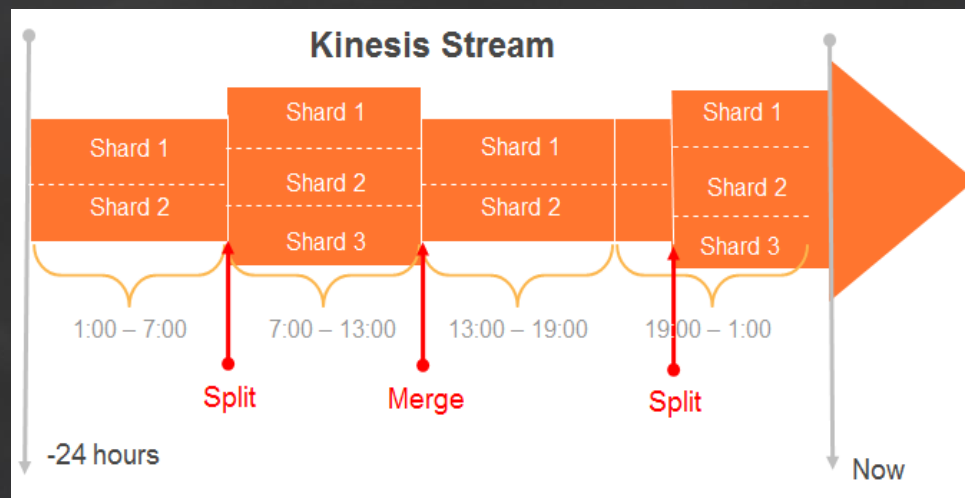


エラスティックな拡張性



エラスティックな拡張性

- Streamは1つ以上の**Shard**で構成
- Shardが**キャパシティの単位**
- Shardを分割して**拡張**
- Shardをマージして**コストの最適化**



SplitShard API

SplitShardでは、担当するハッシュキーのスタートの値を指定する。

AWS CLI を利用した例 : Shardを半分に分割

```
$ aws kinesis split-shard \  
--stream-name StreamName \  
--shard-to-split shardId-00000000000001 \  
--new-starting-hash-key  
255211775190703847597530955573826158592
```

ハッシュキーのスタート値



キャパシティについてのTips

- shardにはキャパシティがある
- アプリケーションが走っているEC2もキャパシティがある



📦 ProvisionedThroughputExceededException

📦 EC2インスタンスをモニターする (AutoScaling)



Kinesisコスト

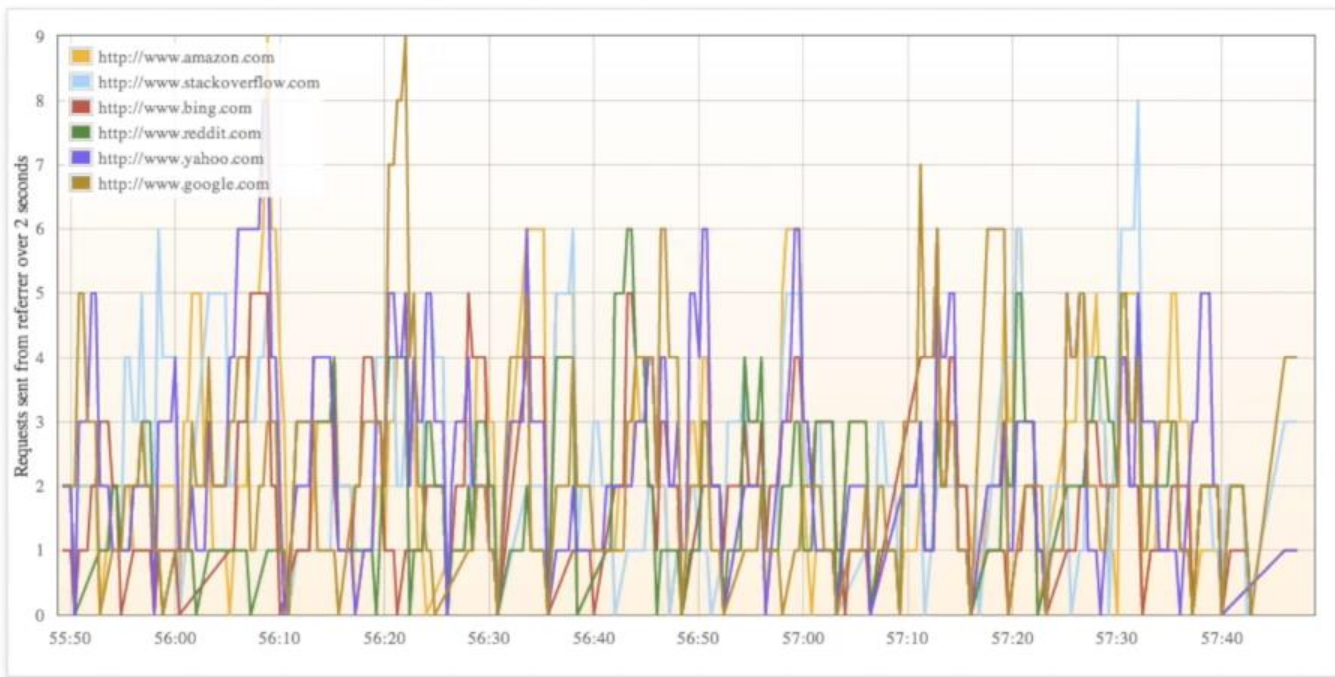
従量課金 & 初期費用不要

| 課金項目 | 単価 |
|-------------|-------------------|
| シャード利用料 | \$0.0195/shard/時間 |
| Putトランザクション | \$0.043/100万Put |

- シャード1つで、一ヶ月約\$14
- Getトランザクションは無料
- インバウンドのデータ転送料は無料
- アプリケーションが走るEC2は通常の料金がかかります



This graph displays the last 120 seconds of counts as calculated by the Amazon Kinesis demo application.



Data last updated by: ec2-54-237-105-252.compute-1.amazonaws.com

Top 3 referrers by counts (Updated every 2000ms):

| | |
|---|-----|
| http://www.yahoo.com | 627 |
| http://www.amazon.com | 582 |
| http://www.stackoverflow.com | 576 |

Try out Amazon Kinesis

- Try out Amazon Kinesis
 - <http://aws.amazon.com/jp/kinesis/>
- Thumb through the Developer Guide
 - <http://aws.amazon.com/jp/documentation/kinesis/>
- Visit, and Post on Kinesis Forum
 - <https://forums.aws.amazon.com/forum.jspa?forumID=169#>