

AWS未経験の新卒がAWS Glue使ってみた

DMM.com Labo ビックデータ部 DREチーム 山崎 隼也

自己紹介

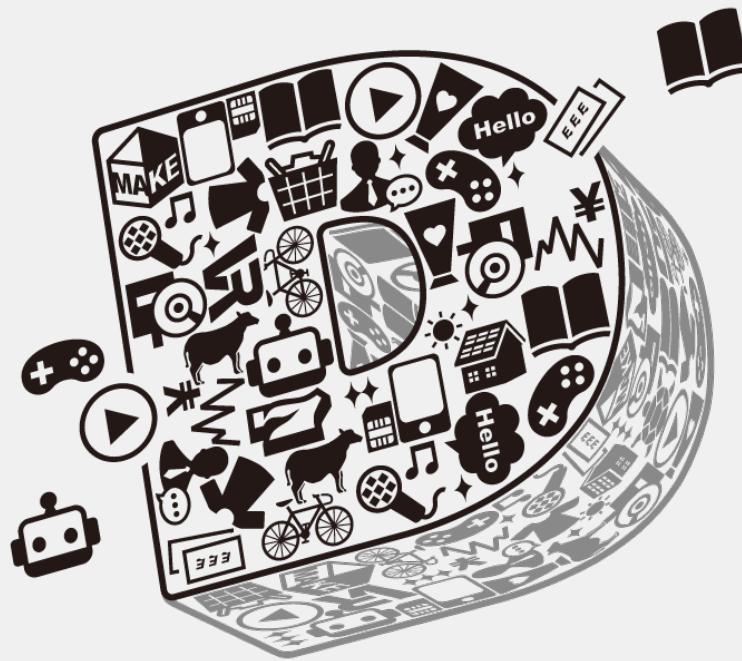


山崎 隼也

DMM.com Labo システム本部 ビッグデータ部 DREチーム

- ✓ Hadoop/日次バッチ運用
- ✓ AWS を利用した分析基盤の開発

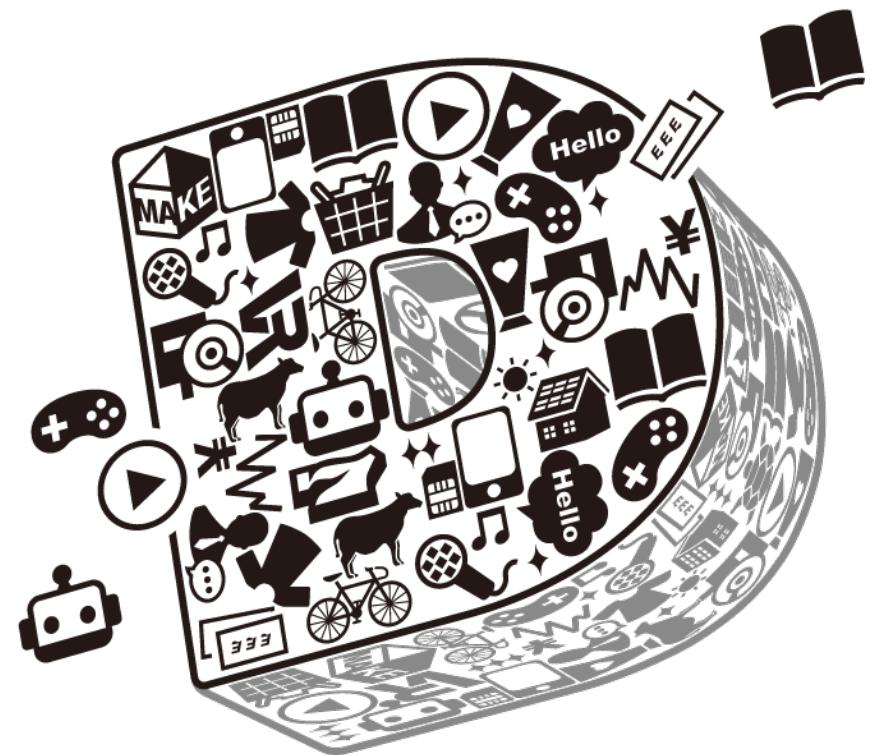
＼17新卒(配属4ヶ月)／



本日のアジェンダ

アジェンダ

- ✓ AWSプロジェクト概要
- ✓ Glue概要と導入背景
- ✓ Glue運用構想
- ✓ Glue検証
- ✓ 実際に導入して
- ✓ まとめ



AWSプロジェクト概要

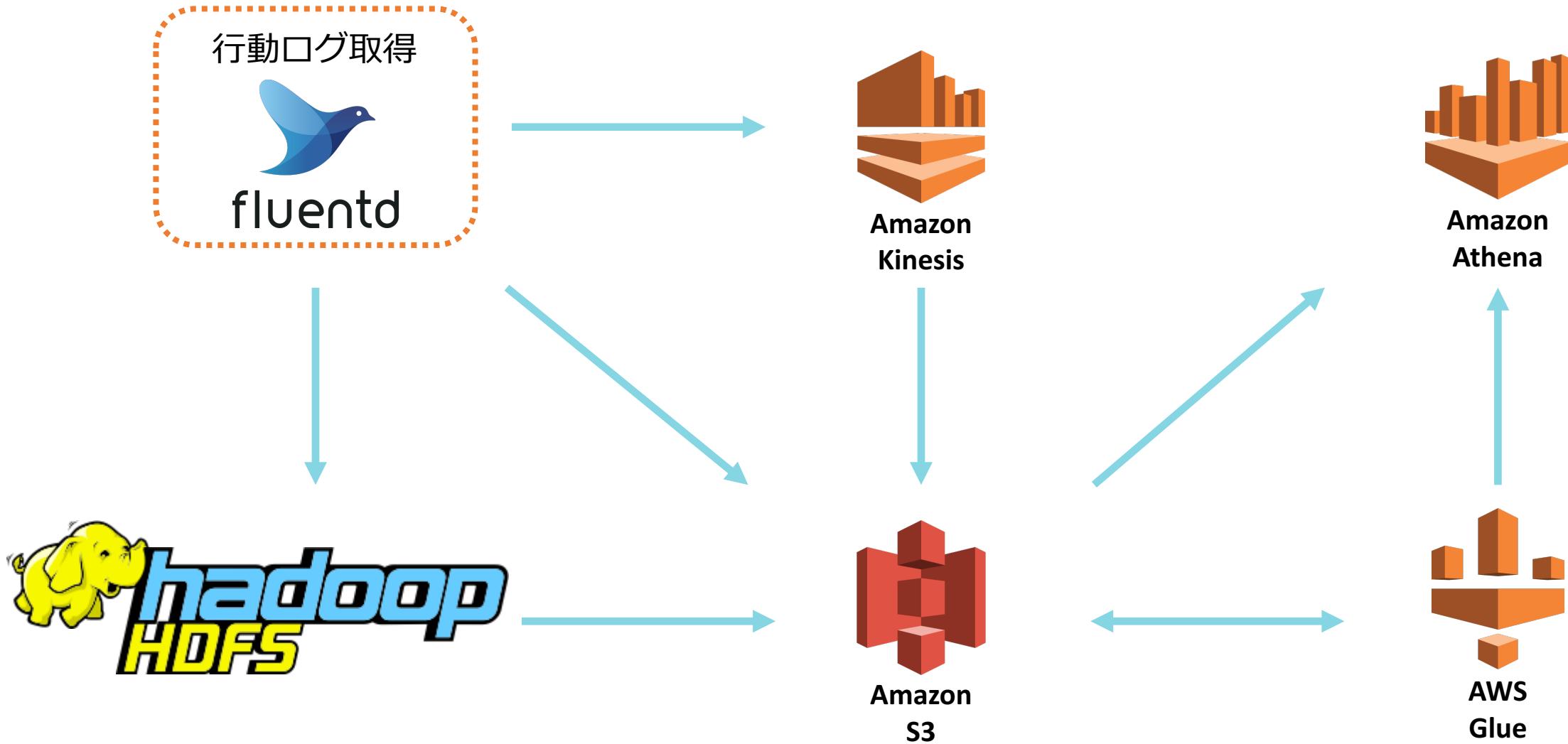


AWS推進 背景

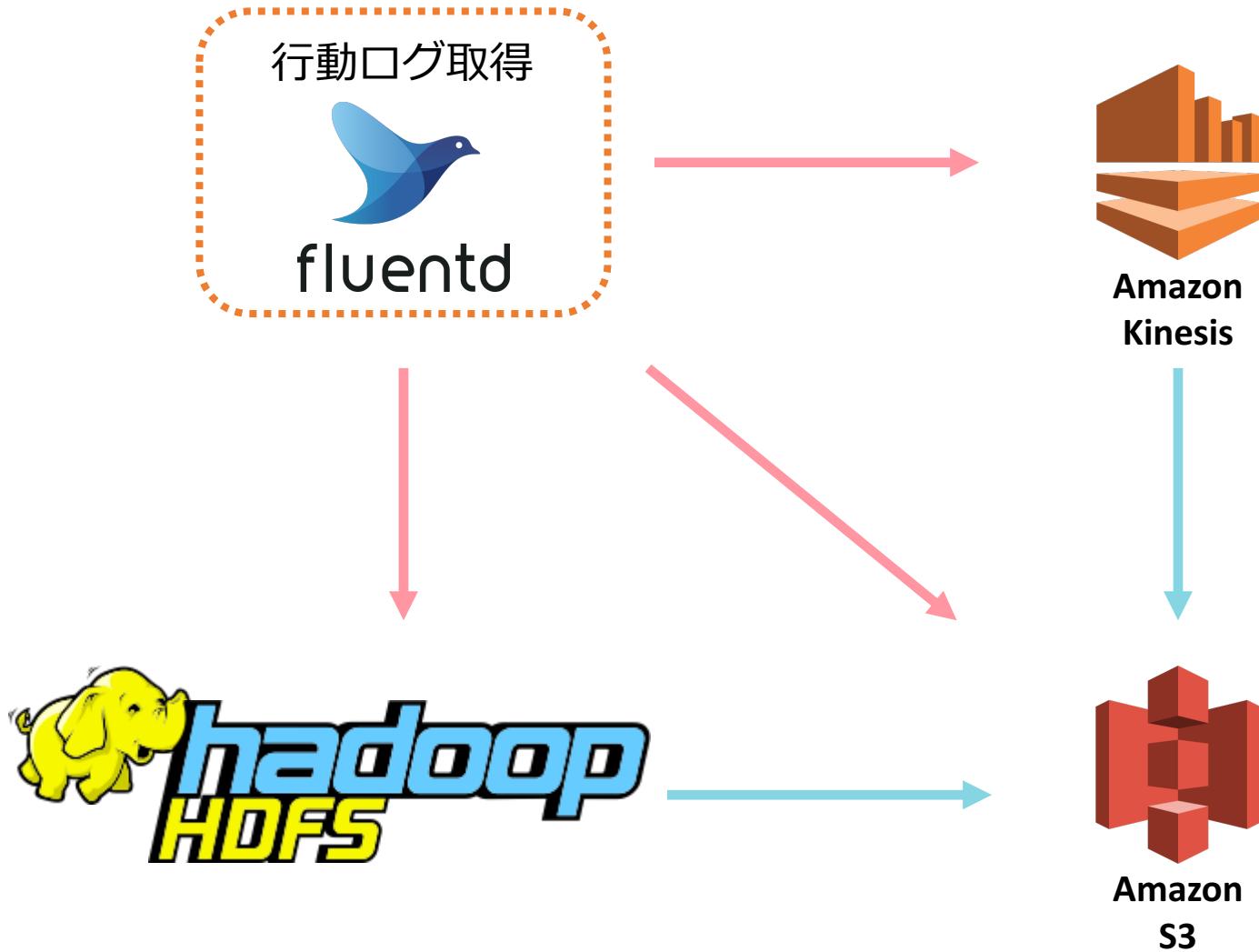
- ▶ オンプレ上で動いているHadoop基盤上でjobが動いている際
新たな作業を行うにはリソースが不足している

＼AWSを活用して解決／

AWSシステム全体図～構想～



AWSシステム全体図～構想～



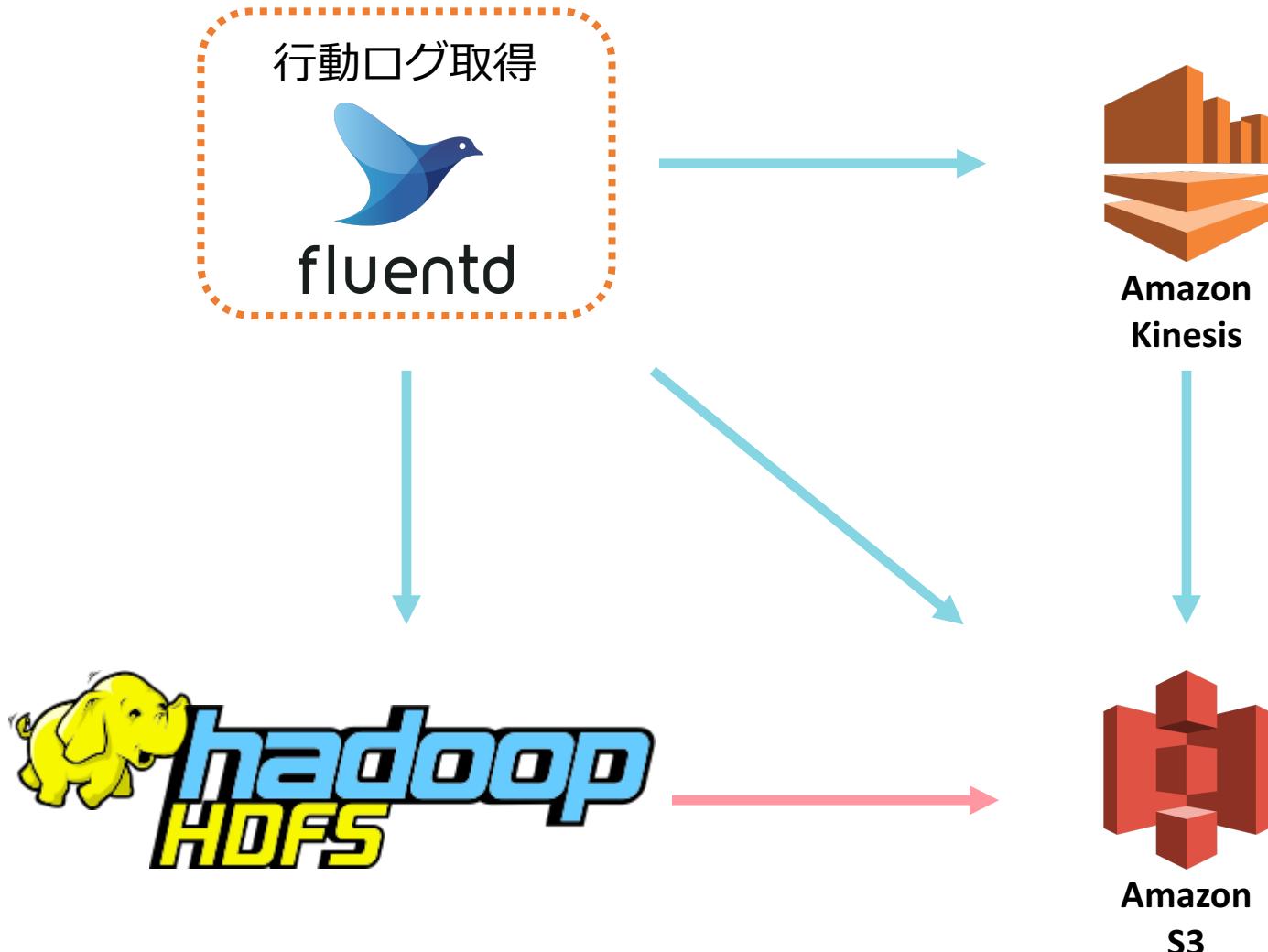
行動ログの送信

fluentdから

- HDFS
- Amazon kinesis
- Amazon S3

に対して行動ログを送信

AWSシステム全体図～構想～



行動ログの送信

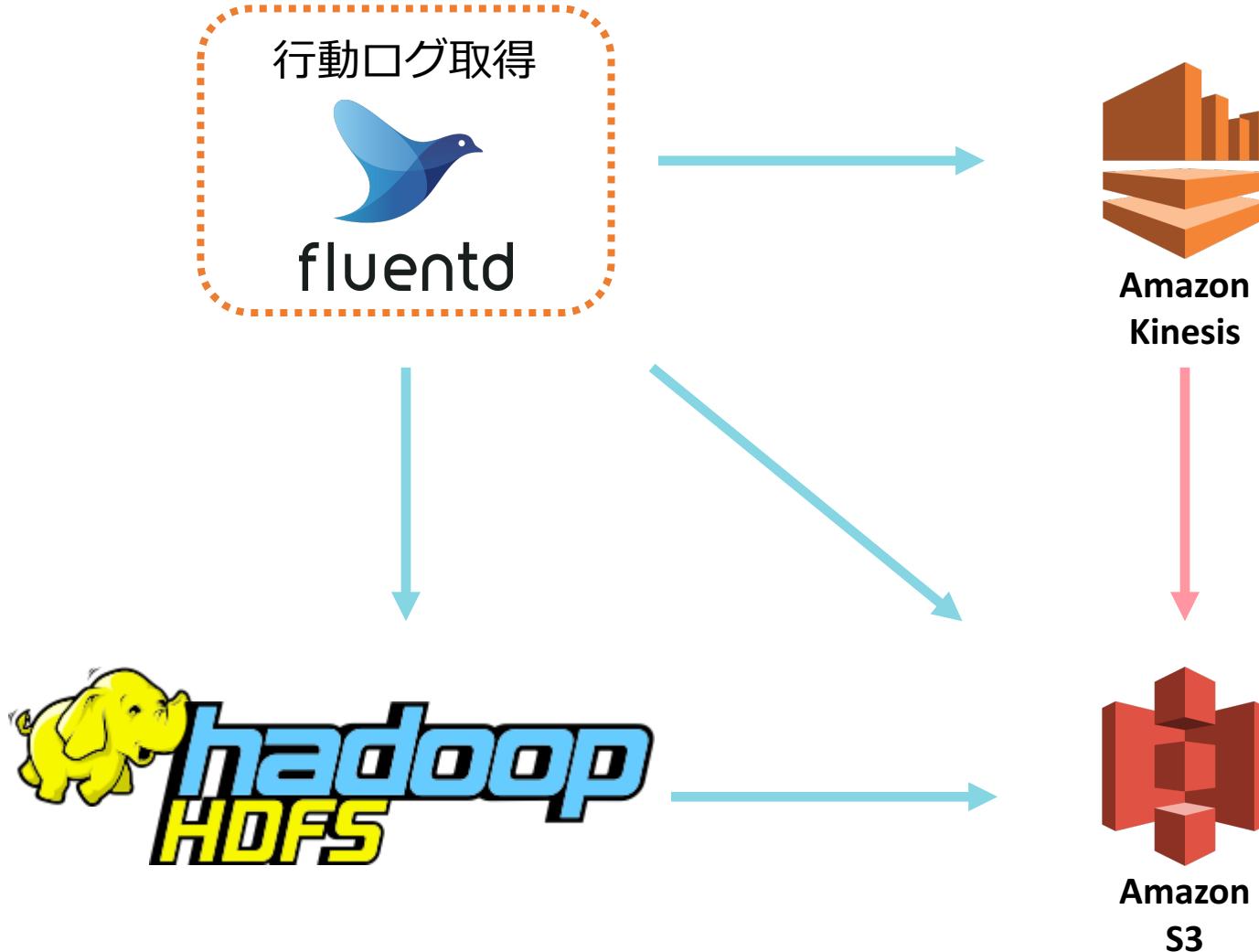
fluentdから

- HDFS
- Amazon kinesis
- Amazon S3

に対して行動ログを送信

HDFSでデータクレンジングされた
行動ログをAmazon S3に対して
Distcpしている

AWSシステム全体図～構想～



行動ログの送信

fluentdから

- HDFS
- Amazon kinesis
- Amazon S3

に対して行動ログを送信

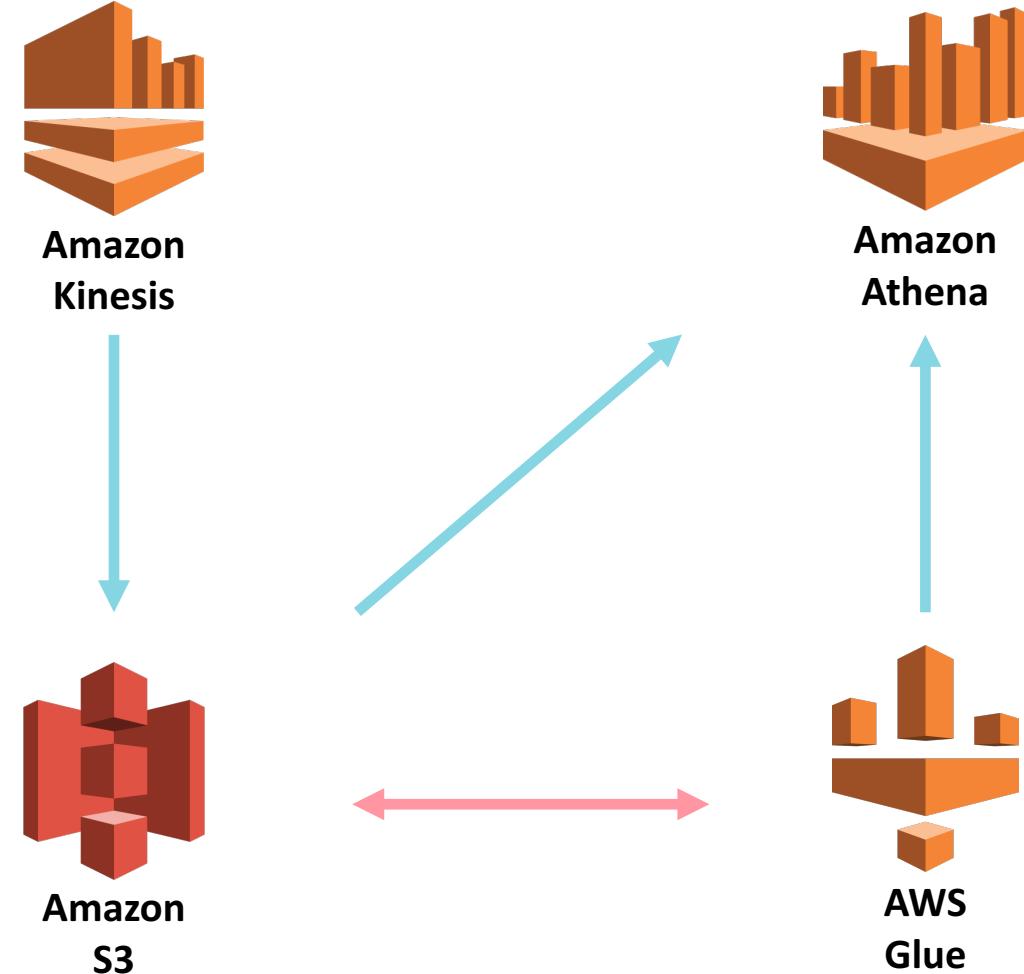
HDFSでデータクレンジングされた
行動ログをAmazon S3に対して
Distcpしている

Amazon kinesisでは別用途で
streaming処理を行った行動ログを
S3に送っている

AWSシステム全体図～構想～

データ変換

S3にある行動ログのフォーマットを
Athenaで効率的に利用するために
AWS Glueにてparquetに変換
さらにスキーマ定義などを記載する
Glue Data Catalog作成

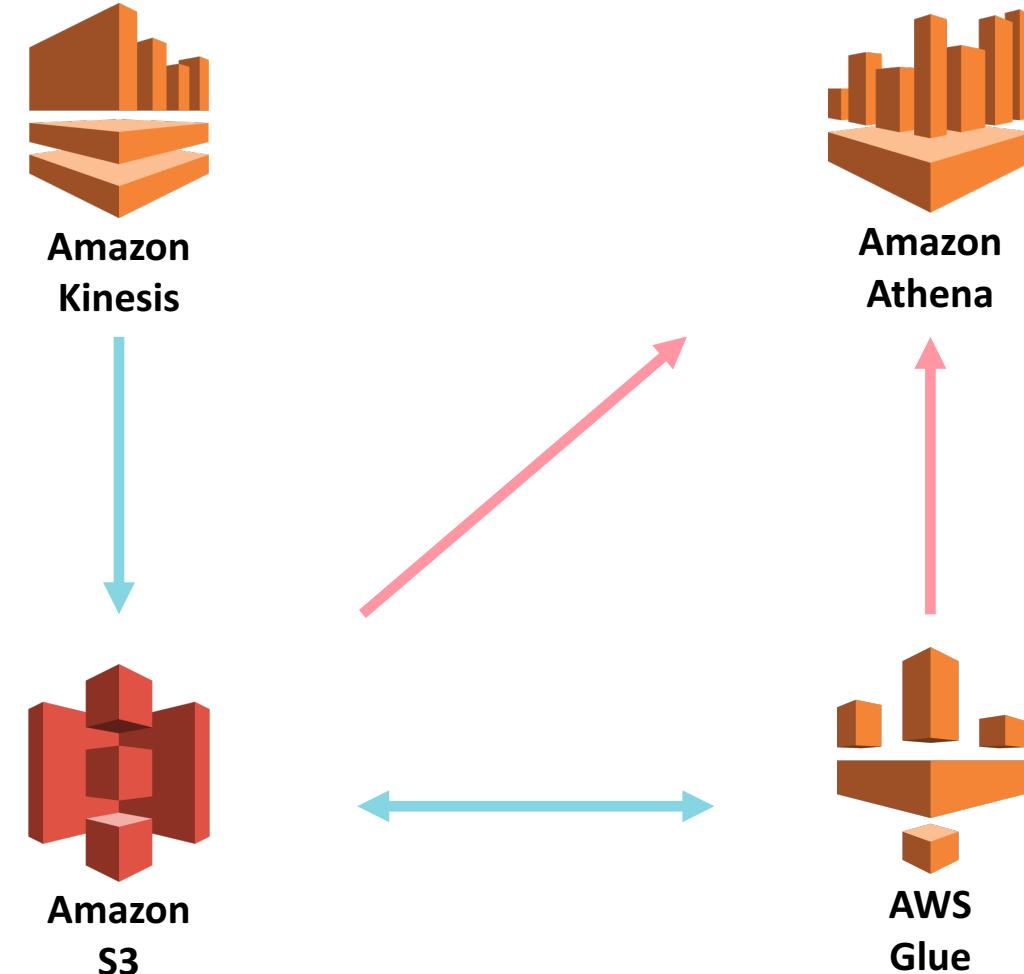


AWSシステム全体図～構想～

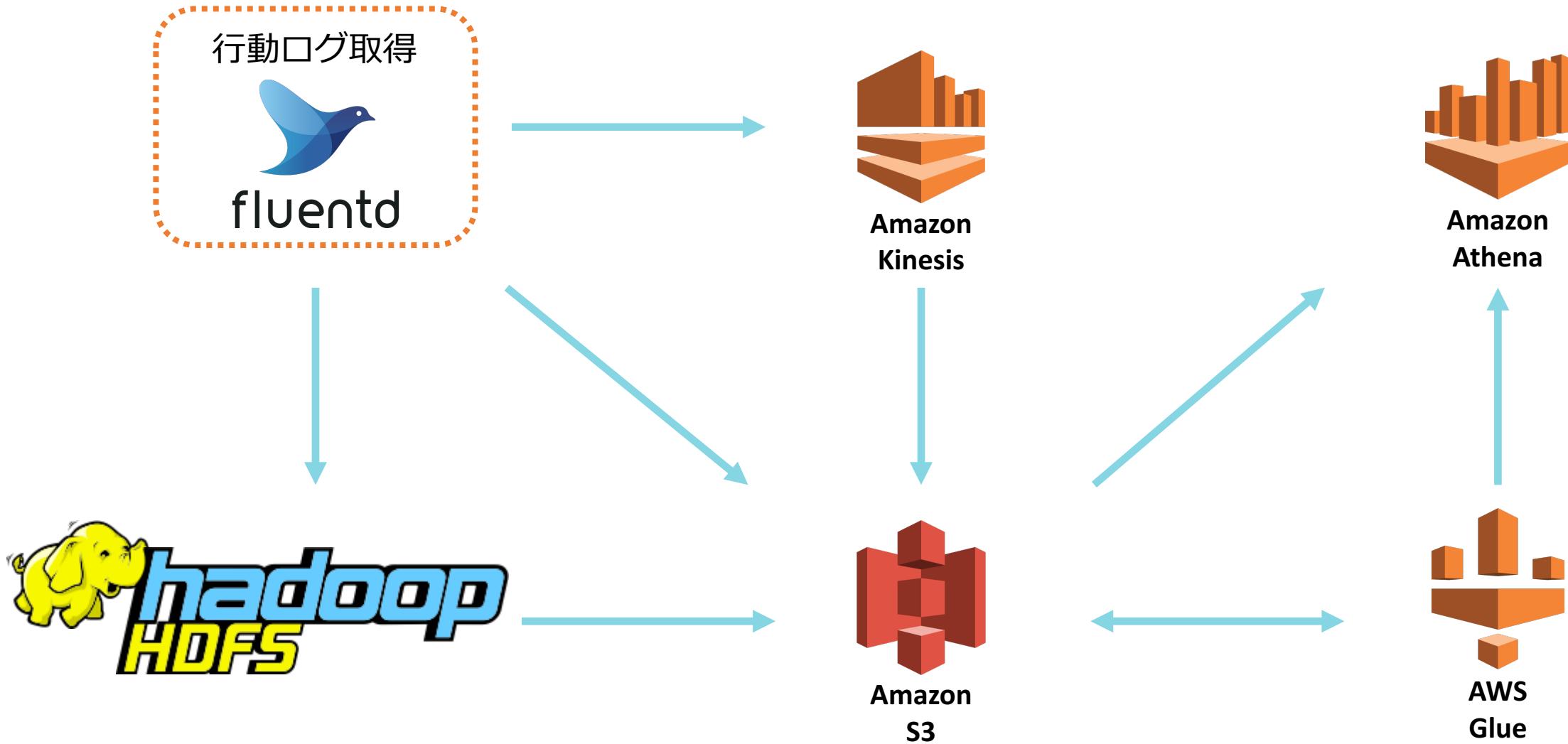
データ変換

S3にある行動ログのフォーマットを
Athenaで効率的に利用するために
AWS Glueにてparquetに変換
さらにスキーマ定義などを記載する
Glue Data Catalog作成

Amazon Athena上でparquet形式の
行動ログとGlue Data Catalogを用いて
分析環境を整える

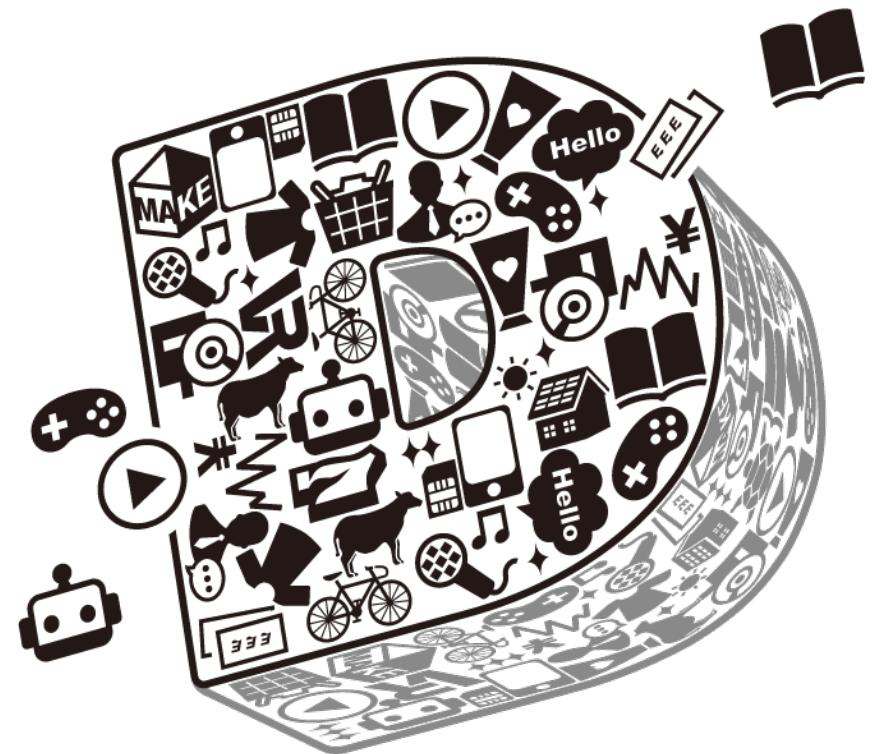


AWSシステム全体図～構想～



アジェンダ

- ✓ AWSプロジェクト概要
- ✓ Glue概要と導入背景
- ✓ Glue運用構想
- ✓ Glue検証
- ✓ 実際に導入して
- ✓ まとめ

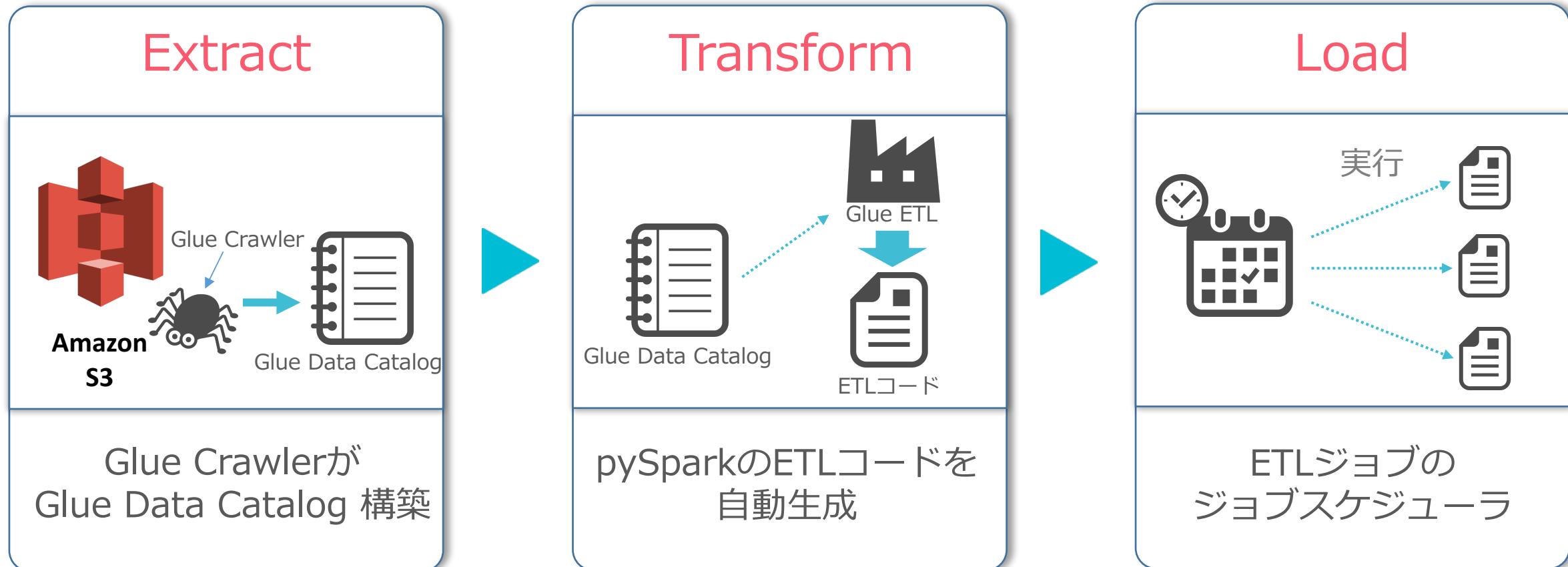


Glue概要と導入背景



AWS Glueとは

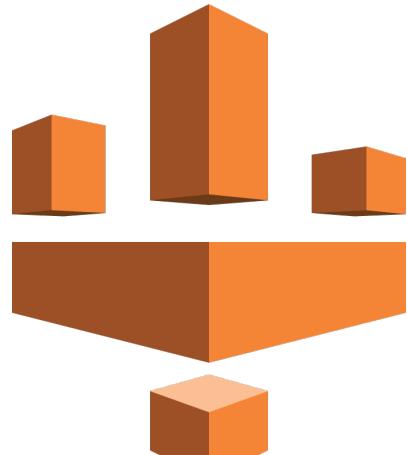
✓ フルマネージド且つサーバレスのETLサービス



AWS Glueの選定理由

✓ 日次バッチで行動ログのフォーマットを変換(text→parquet)

- ✓ ワークフローツールの選定、運用、保守
- ✓ スクリプト実行用サーバ
- ✓ Athenaの内部カタログにメタデータを別途追記

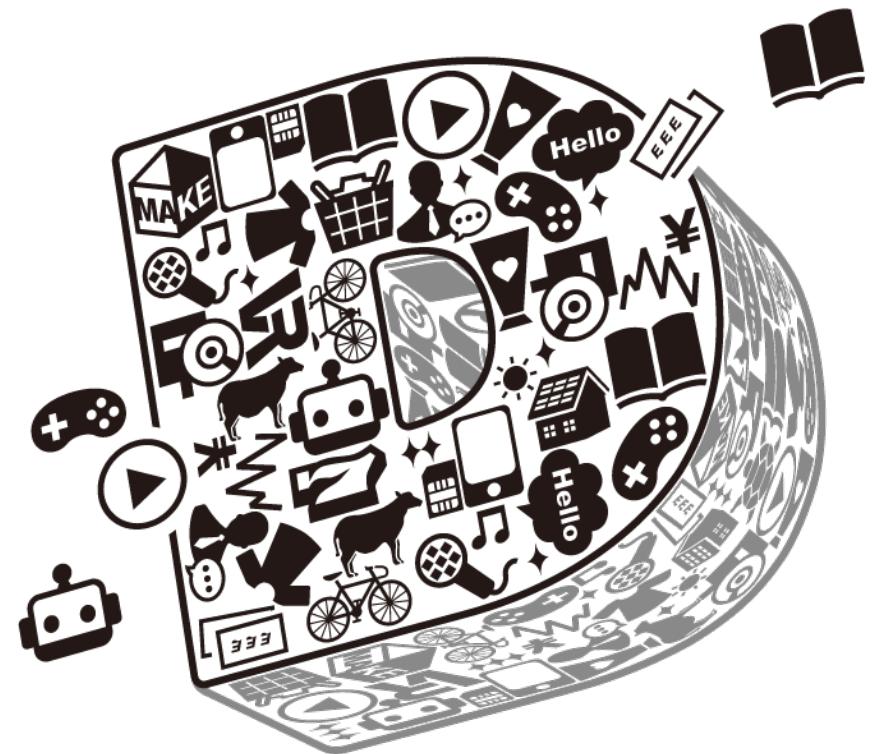


AWS Glue

AWS Glueだけができる！

アジェンダ

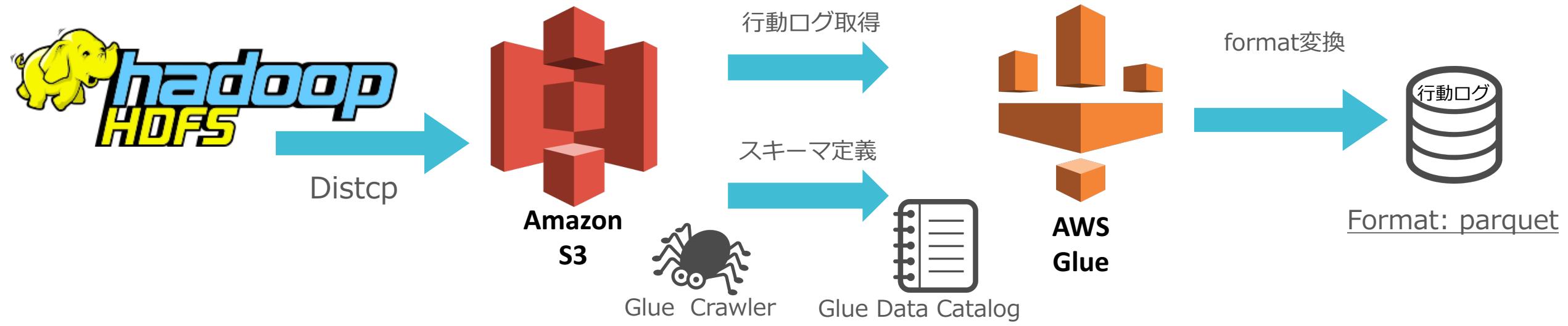
- ✓ AWSプロジェクト概要
- ✓ Glue概要と導入背景
- ✓ Glue運用構想
- ✓ Glue検証
- ✓ 実際に導入して
- ✓ まとめ



Glue運用構想

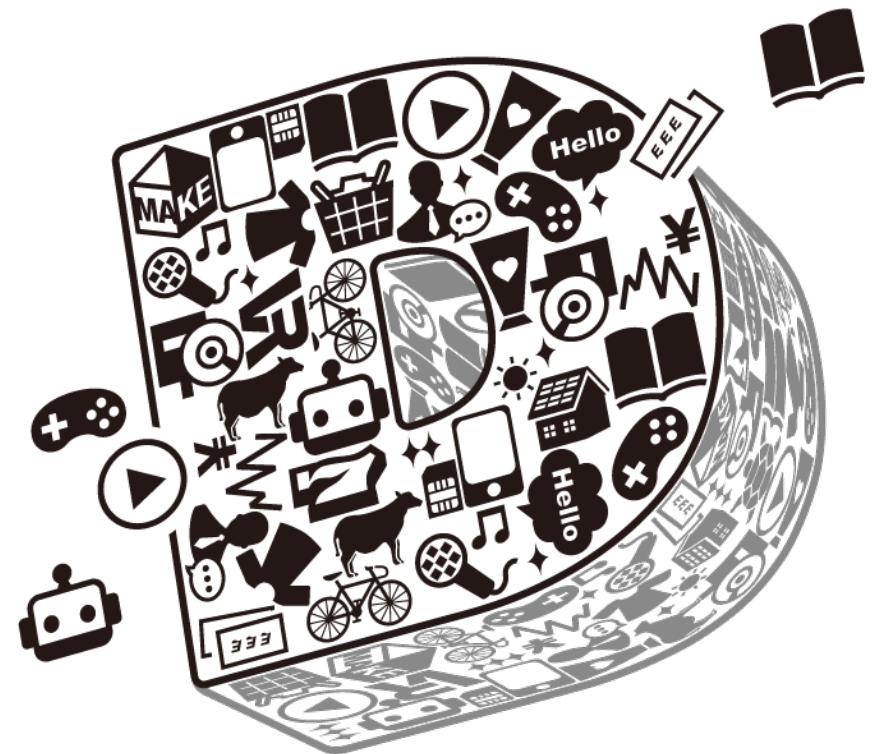


Glue運用構想



アジェンダ

- ✓ AWSプロジェクト概要
- ✓ Glue概要と導入背景
- ✓ Glue運用構想
- ✓ Glue検証
- ✓ 実際に導入して
- ✓ まとめ



Glue検証



検証内容

1



クローラの検証

行動ログのスキーマ定義をクローリング



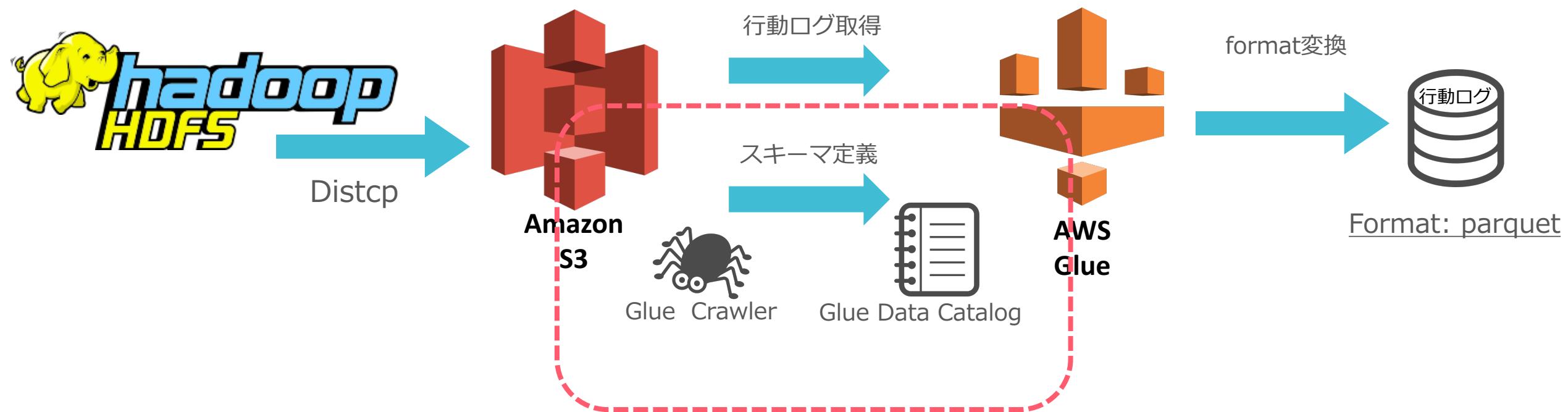
2



ジョブの検証

Glue Data Catalogを用いてジョブ生成

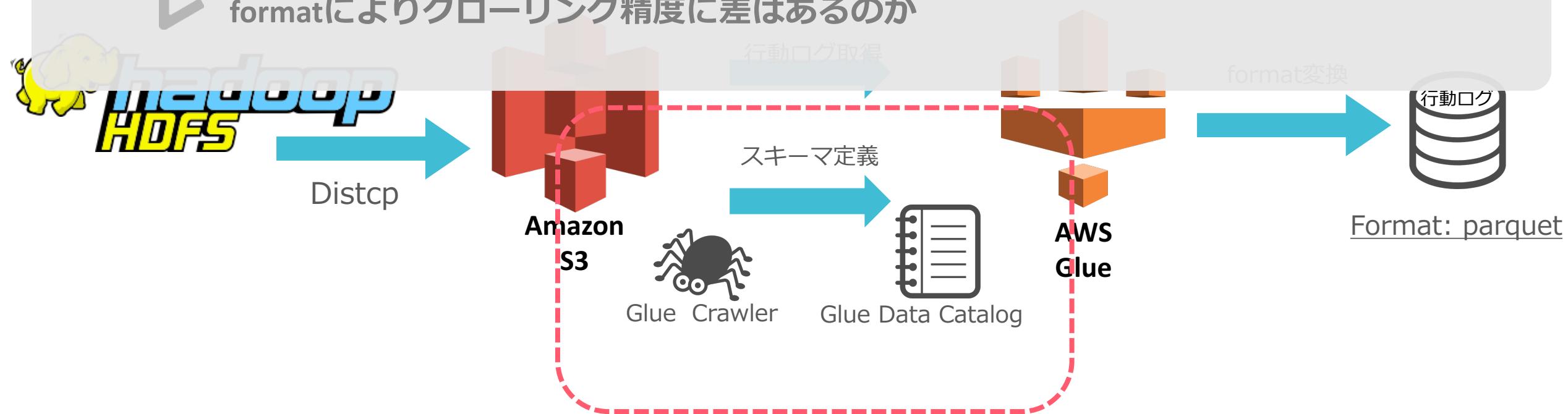
クローラの検証



クローラの検証

▶ 検証ポイント

- ✓ 正確にスキーマ定義を抽出できているか
- ✓ formatによりクローリング精度に差はあるのか



ハマったところ



- クローラが走るごとにカラム名消える問題
- csv形式で読み込むとarrayがstring問題

ハマったところ



クローラが走るごとにカラム名消える問題

カラム名を手作業で書いた後にクローラを走らせるとカラム名が消えてしまう

Grok構文のため
技術的コスト高し

Glue Crawler に対してClassifierを記述し
カラム名が消えないようにする

ハマったところ



クローラが走るごとにカラム名消える問題

Glue Crawler の設定でSchema changeを

`ignore the change, do not modify the table in the data catalog` に変更

ハマったところ

Configure the crawler's output

Database i
yamazaki_junya

[Add database](#)

Prefix added to tables (optional) i
Type a prefix added to table names

▼ Schema changes (optional)

During the crawler run, all schema changes are logged.

How should we handle updated schemas in the data store?

Update the table in the data catalog
 Ignore the change, do not modify the table in the data catalog

How should we handle deleted objects in the data store?

Delete the table from the data catalog
 Ignore the change, do not modify the table in the data catalog
 Mark the table deprecated in the data catalog i

[Back](#) [Next](#)

ハマったところ



クローラ走るごとにカラム名消える問題

Grok構文のため
技術的コスト高し

Glue Crawler に対してClassifierを記述し
カラム名が消えないようにする



Glue Crawler の設定でSchema changeを
“ignore the change, do not modify the table in the data catalog” に変更

ハマったところ



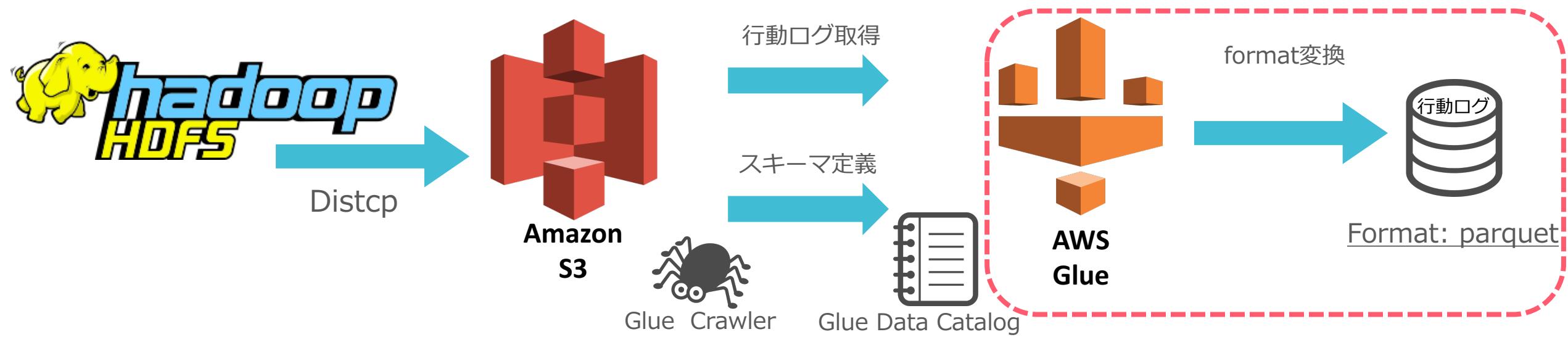
csv形式で読み込むとarrayがstring問題

csv形式でセミコロン区切りのarrayがstringとして読み込まれる

\ jsonやorcなどのformatではarray対応できた！ /

ダブルクオートを区切り文字とした場合
Table Propertyを
“multiLine”: true
に設定すると読み込める

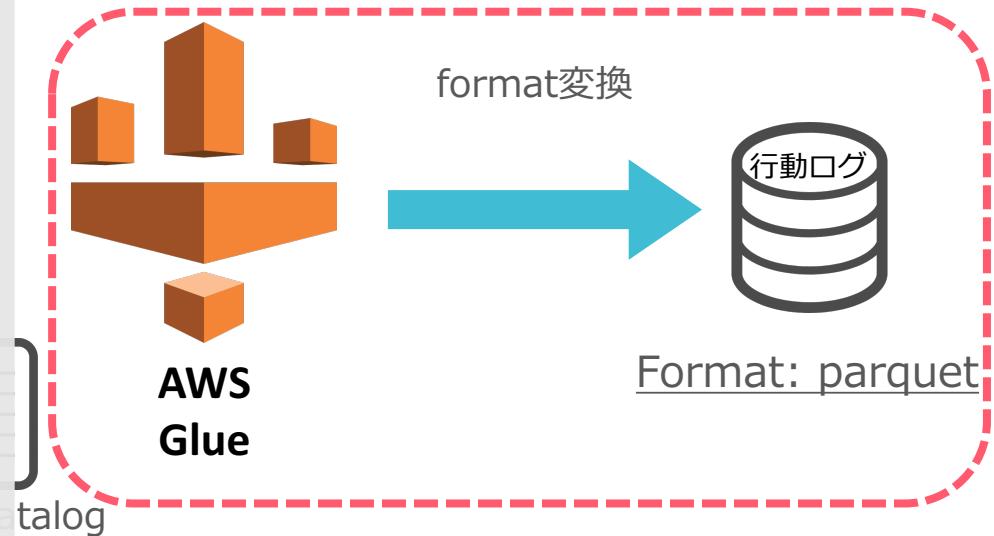
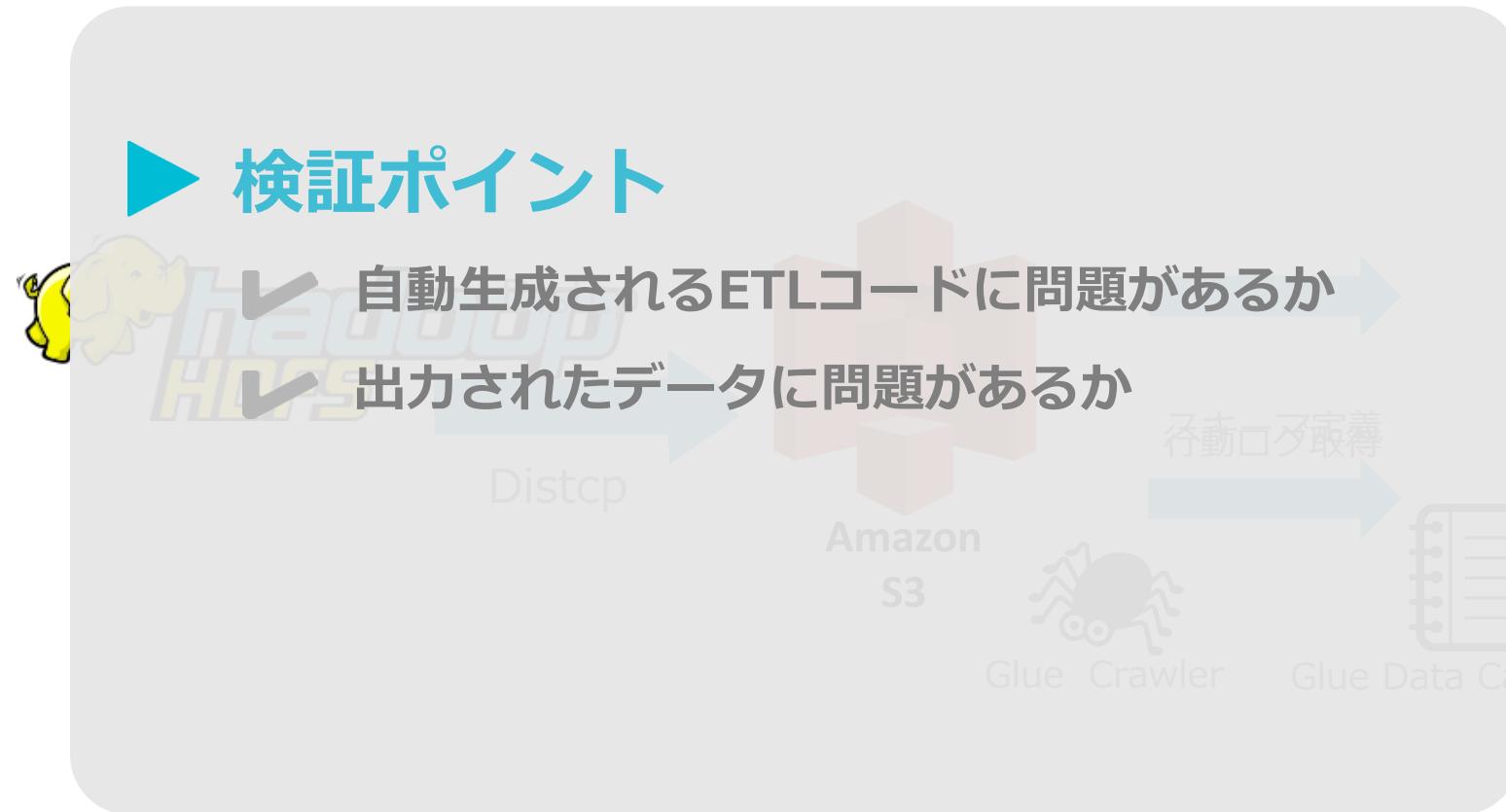
ジョブの検証



ジョブの検証

▶ 検証ポイント

- ✓ 自動生成されるETLコードに問題があるか
- ✓ 出力されたデータに問題があるか



ハマったところ



読み込むパーティションが指定できない問題

変換でデータ件数が減る問題

ハマったところ



読み込むパーティションが指定できない問題

jobではクローラが認識したpartitionを指定できない

<https://github.com/awslabs/aws-glue-samples> の

3.a How can I use SQL queries with Dynamic Frames?
で紹介されている手法を用いることでSQLが書ける!!



＼パーティションを読み込むことが可能に！／

ハマったところ



読み込むパーティションが指定できない問題

<https://github.com/awslabs/aws-glue-samples>

```
...  
df = my_dynamic_frame.toDF()  
df.createOrReplaceTempView("temptable")  
sql_df = spark.sql("SELECT * FROM temptable")  
new_dynamic_frame = DynamicFrame.fromDF(sql_df, glueContext, "new_dynamic_frame")  
...
```

ハマったところ



読み込むパーティションが指定できない問題

<https://github.com/awslabs/aws-glue-samples>

```
...  
df = my_dynamic_frame.toDF() ← DataFrameに変換  
df.createOrReplaceTempView("temptable")  
sql_df = spark.sql("SELECT * FROM temptable")  
new_dynamic_frame = DynamicFrame.fromDF(sql_df, glueContext, "new_dynamic_frame")  
...
```

ハマったところ



読み込むパーティションが指定できない問題

<https://github.com/awslabs/aws-glue-samples>

```
...  
df = my_dynamic_frame.toDF()  
df.createOrReplaceTempView("temptable")  
sql_df = spark.sql("SELECT * FROM temptable")  
new_dynamic_frame = DynamicFrame.fromDF(sql_df, glueContext, "new_dynamic_frame")  
...
```

仮のテーブルを作成



ハマったところ



読み込むパーティションが指定できない問題

<https://github.com/awslabs/aws-glue-samples>

```
...  
df = my_dynamic_frame.toDF()  
df.createOrReplaceTempView("temptable")  
sql_df = spark.sql("SELECT * FROM temptable")  
new_dynamic_frame = DynamicFrame.fromDF(sql_df, glueContext, "new_dynamic_frame")  
...
```

ここでSQLが書けるので取りたい情報だけ抜く
where句にパーティションを指定



ハマったところ



読み込むパーティションが指定できない問題

<https://github.com/awslabs/aws-glue-samples>

```
...  
df = my_dynamic_frame.toDF()  
df.createOrReplaceTempView("temptable")  
sql_df = spark.sql("SELECT * FROM temptable")  
new_dynamic_frame = DynamicFrame.fromDF(sql_df, glueContext, "new_dynamic_frame")  
...
```

Dynamicframeに戻す



出力されたデータソースに向けてクロークを走らせると
パーティションを指定することができた

ハマったところ

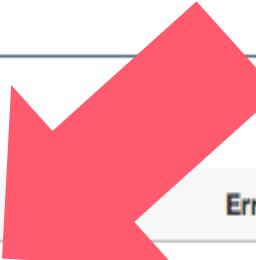


変換でデータ件数が減る問題

\ jsonからparquetに変換するジョブで出力先のデータ件数が減っていた /

ハマったところ

Job実行画面



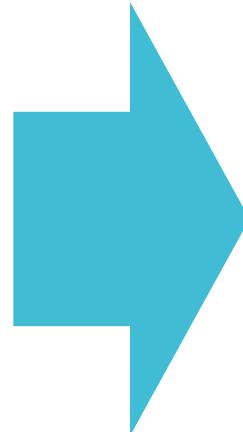
Run ID	Retry attempt	Run status	Error	Logs	Error logs	Duration
	-	<u>Succeeded</u>		Logs		13 mins
	-	<u>Succeeded</u>		Logs		5 mins

ハマったところ

Amazon Athenaでデータ件数を確認

行動ログ変換前(json)

201,767,796件



行動ログ変換後(parquet)

5,550,038件

ハマったところ

Amazon Athenaでデータ件数を確認

行) \ 一部(ほぼ)消えた!! / uet)

201,767,796件



5,550,038件

ハマったところ

Job実行画面

\ エラーが握りつぶされている /

Run ID	Retry attempt	Run status	Error	Logs	Error logs	Duration
	-	Succeeded		Logs		13 mins
	-	Succeeded		Logs		5 mins

ハマったところ



変換でデータ件数が減る問題

同じjsonからparquetに変換する処理に関して

＼(EMR上の)HIVEに変換を実行させると欠損しない／

ハマったところ



変換でデータ件数が減る問題

jsonからparquetに変換するジョブで出力先のデータ件数が減っていた

結果

＼ binary文字が混ざっていた!! ／

現在は \$ nkf -w 変換前ファイル でUTF-8に変更してから
ジョブにかけることで件数が減らないことを確認済み

検証内容

1



クローラの検証

行動ログのスキーマ定義をクローリング

2



ジョブの検証

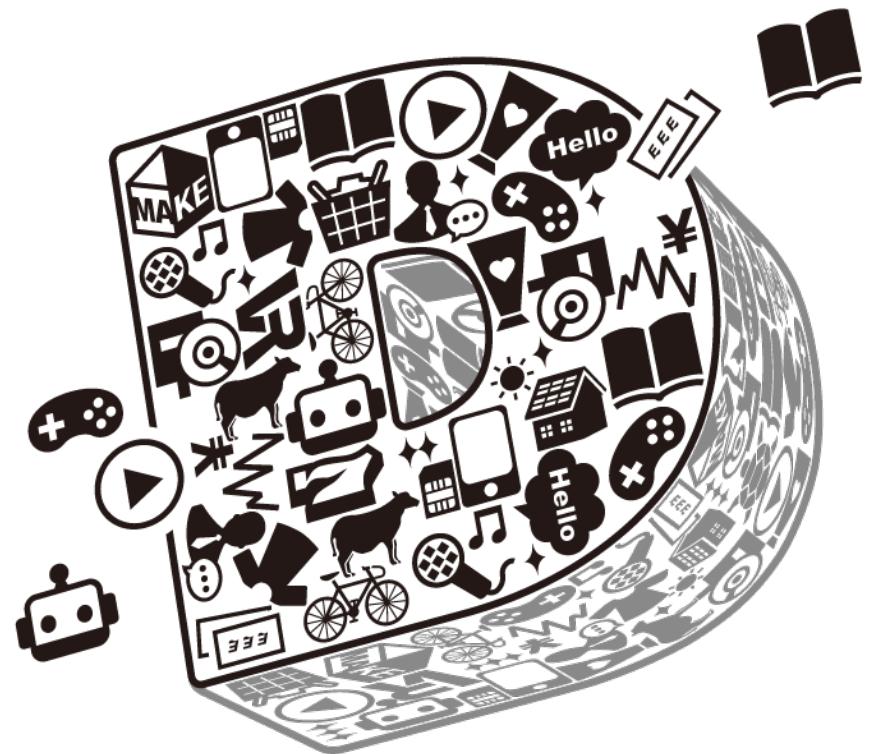
Glue Data Catalogを用いてジョブ生成



自分たちのデータとは相性が悪そう？？？

アジェンダ

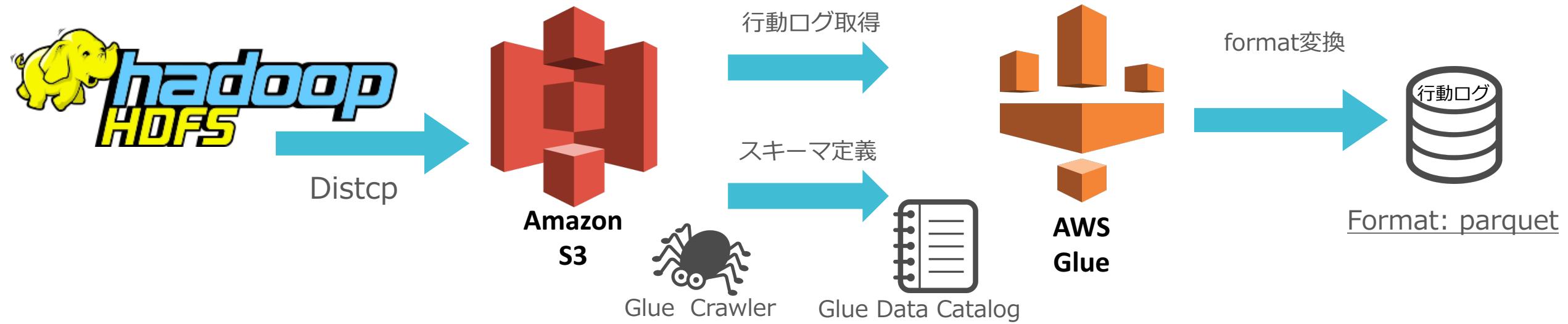
- ✓ AWSプロジェクト概要
- ✓ Glue概要と導入背景
- ✓ Glue運用構想
- ✓ Glue検証
- ✓ 実際に導入して
- ✓ まとめ



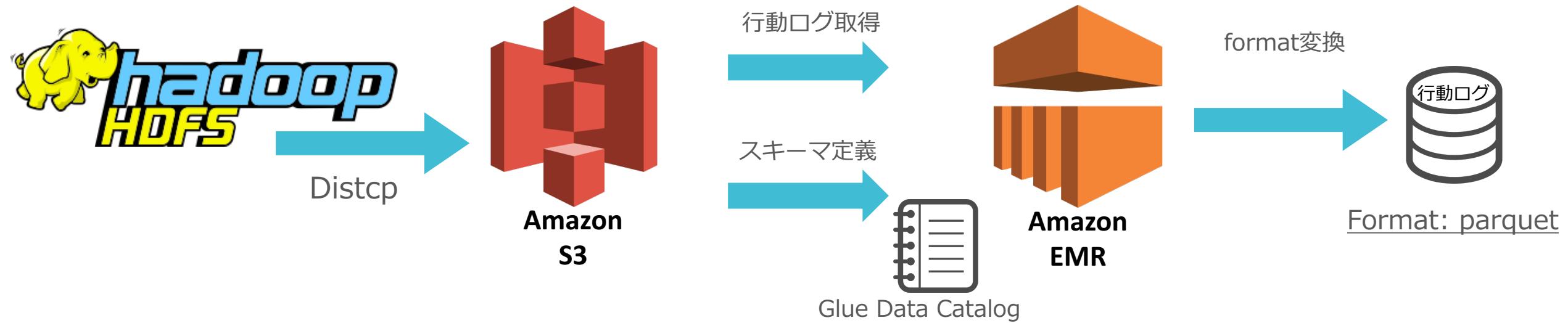
実際に導入して



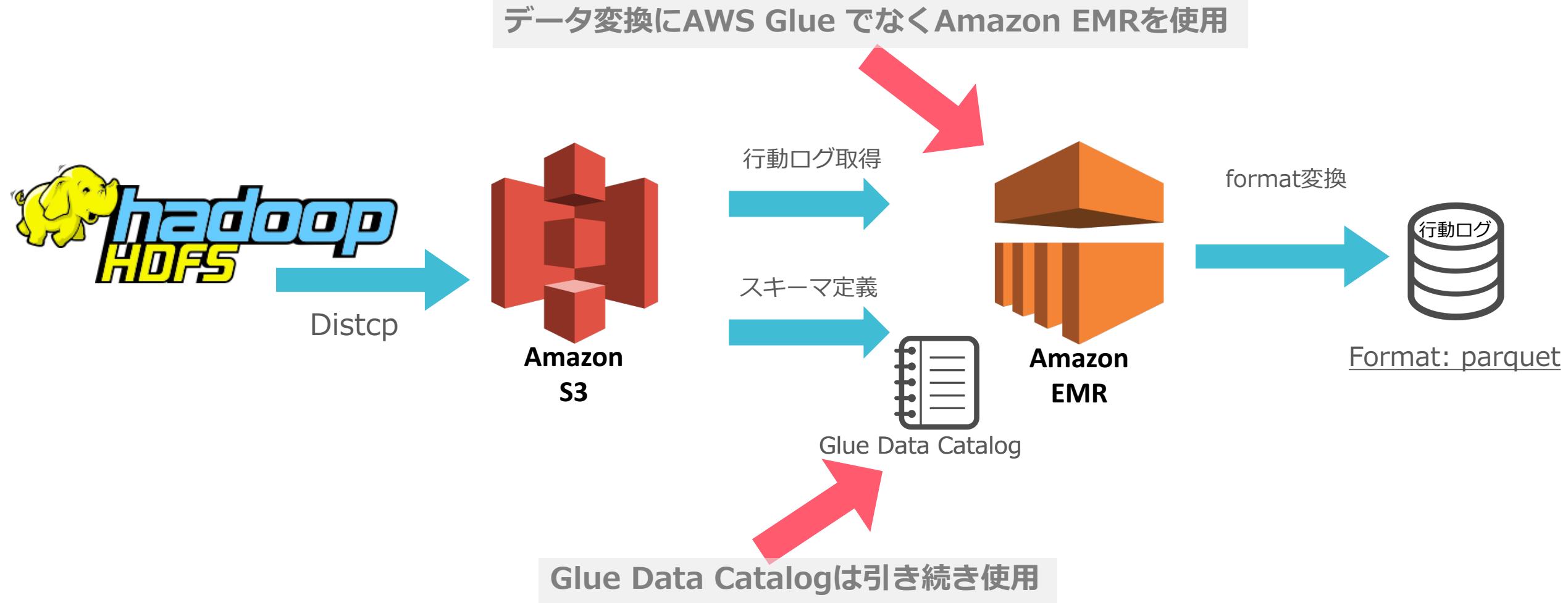
Glue運用構想



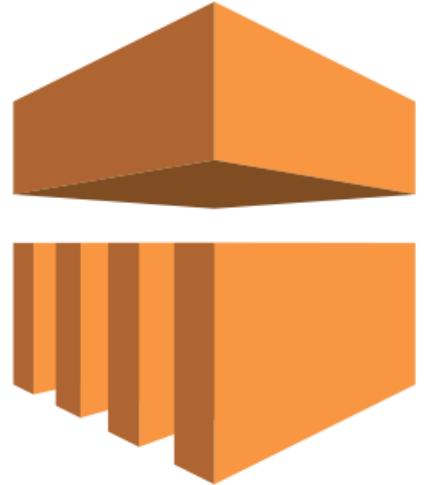
Glue運用



Glue運用



EMR 運用



Amazon EMR

EMRでGlue Data Catalog 利用

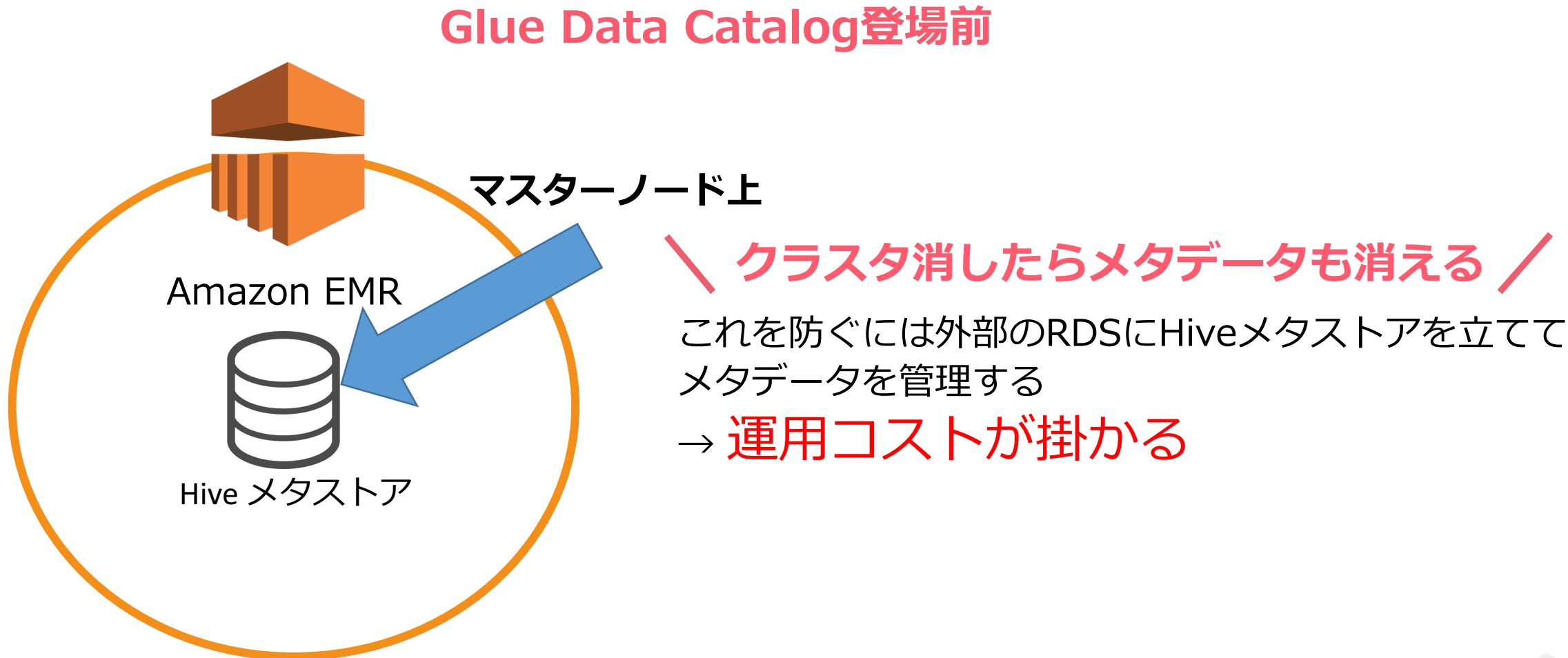
- ✓ configurationで利用設定可
 - EMR5.8.0以降

Glue Data Catalogの選定理由

- ✓ EMRでのmetadataの運用コストが抑えられる
- ✓ Glueと連携できる
- ✓ Athenaでも使える

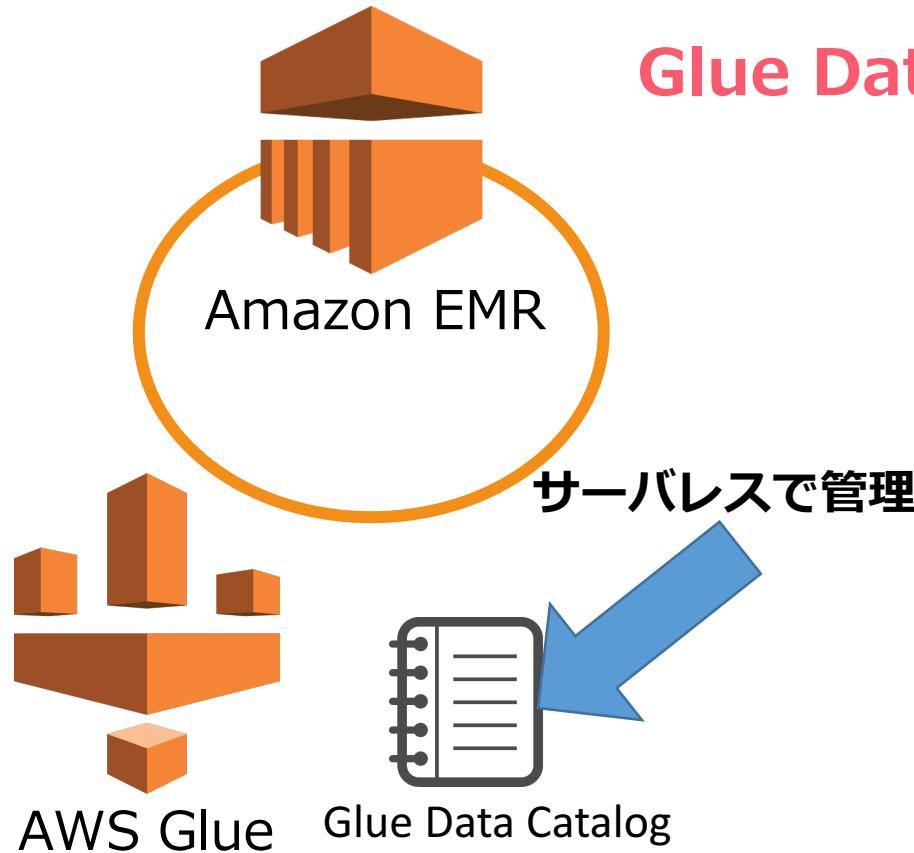
Glue Data Catalogの選定理由

- ✓ EMRでのmetadataの運用コストが抑えられる



Glue Data Catalogの選定理由

- ✓ EMRでのmetadataの運用コストが抑えられる



＼ クラスタ消してもメタデータが消えない／

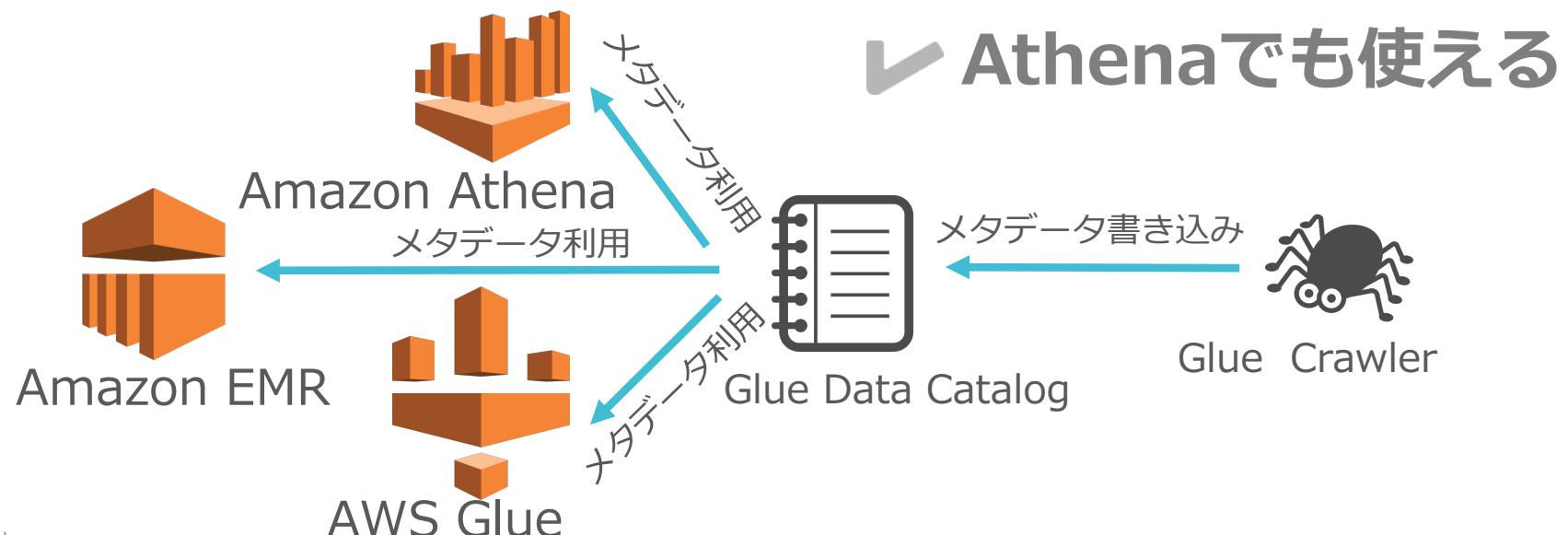
メタデータはGlue Data Catalog上でサーバレスに
管理されている
→ 運用コストがかからない

Glue Data Catalogの選定理由

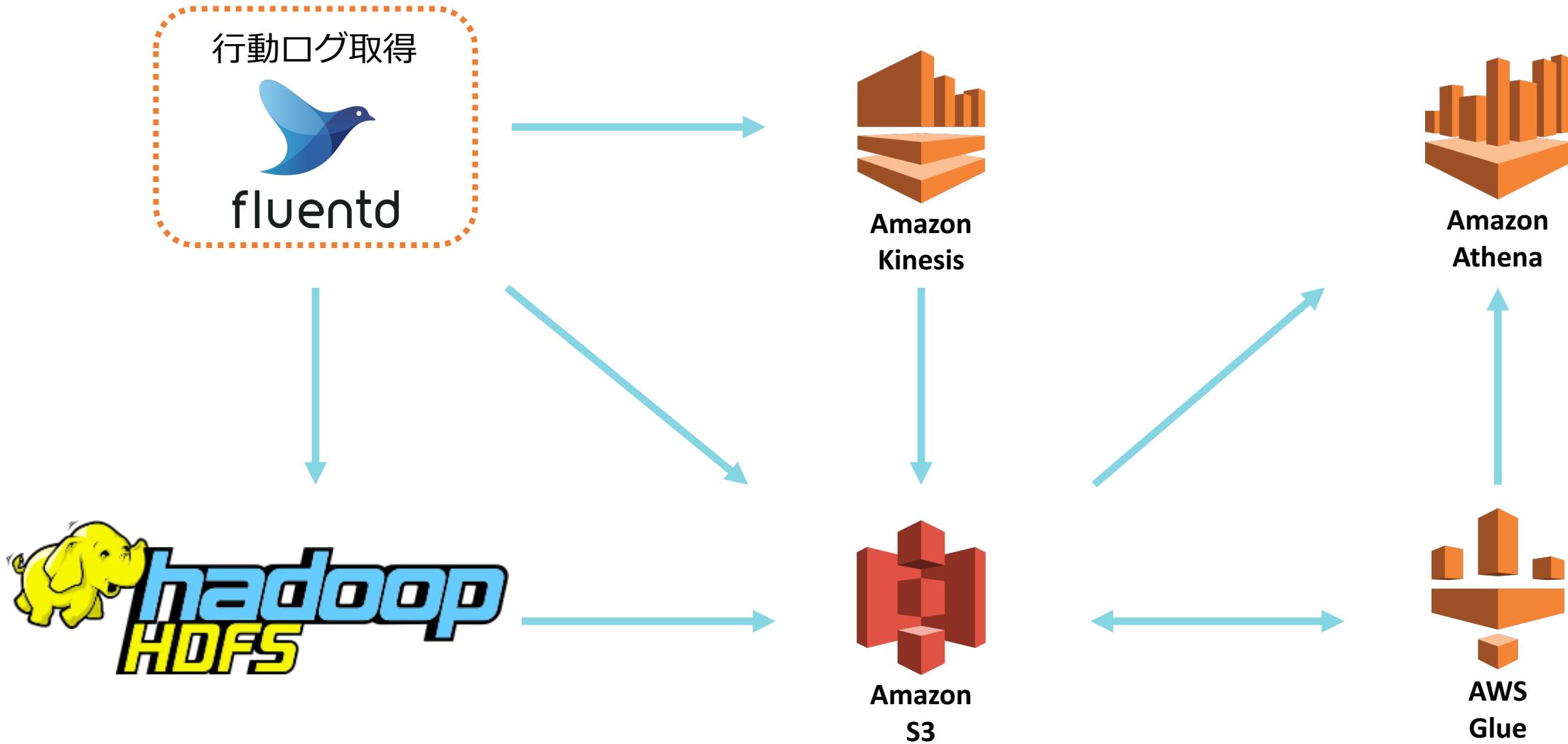
✓ Glueと連携できる



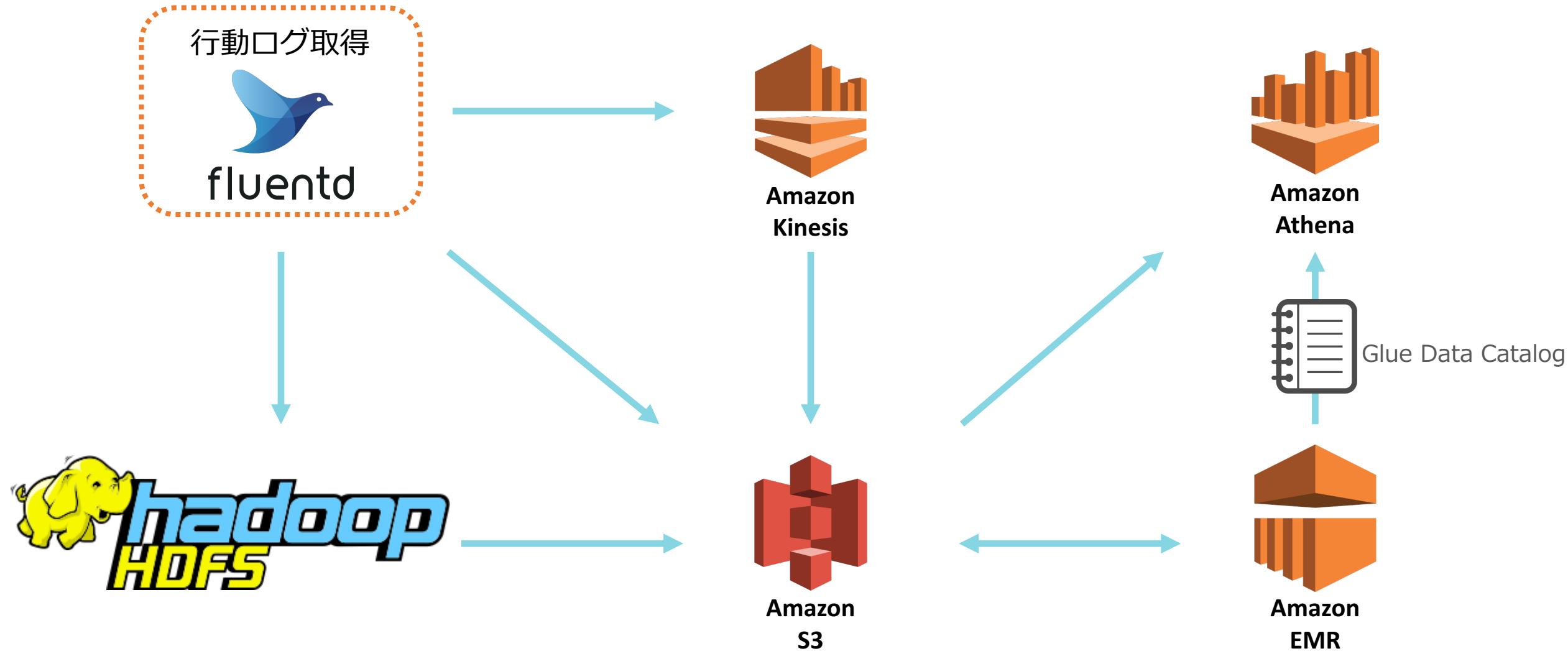
＼ Glue Crawlerが使える／



AWSシステム全体図～構想～

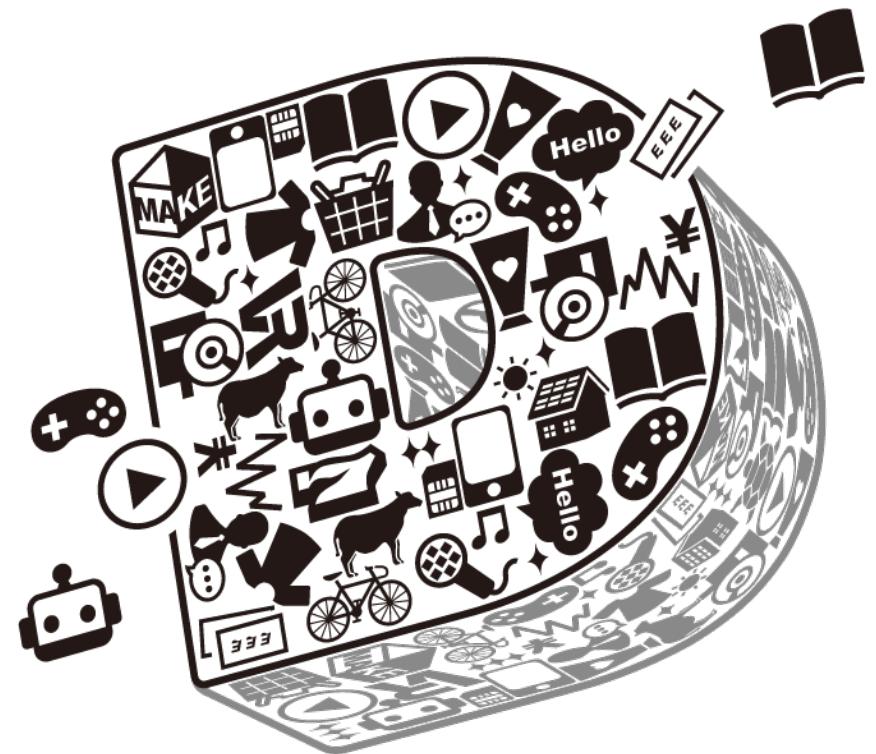


AWSシステム全体図



アジェンダ

- ✓ AWSプロジェクト概要
- ✓ Glue概要と導入背景
- ✓ Glue運用構想
- ✓ Glue検証
- ✓ 実際に導入して
- ✓ まとめ



まとめ



まとめ

Glueのメリット

- ✓ サーバレスでETLができる
- ✓ Glue Data Catalogは非常に便利

Glueのデメリット

- ✓ GAから日が浅いのでナレッジが少ない
- ✓ 足りない要素がまだまだありそう(ジョブの監視etc..)

※あくまで個人の所感です

まとめ

✓ 日次バッチで行動ログのフォーマットを変換(text→parquet)

頑張れば

AWS Glueだけでできる！

はず

現状は

AWS GlueとAmazon EMRでできた！

