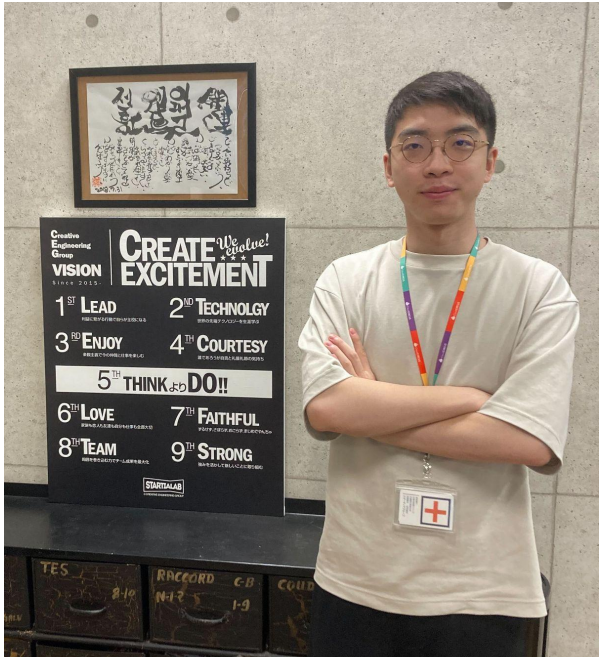
ClickHouse導入によるCloudFrontログ分析の高速化とコスト削減

クラウドサーカス インフラエンジニア 朱 九霖

自己紹介



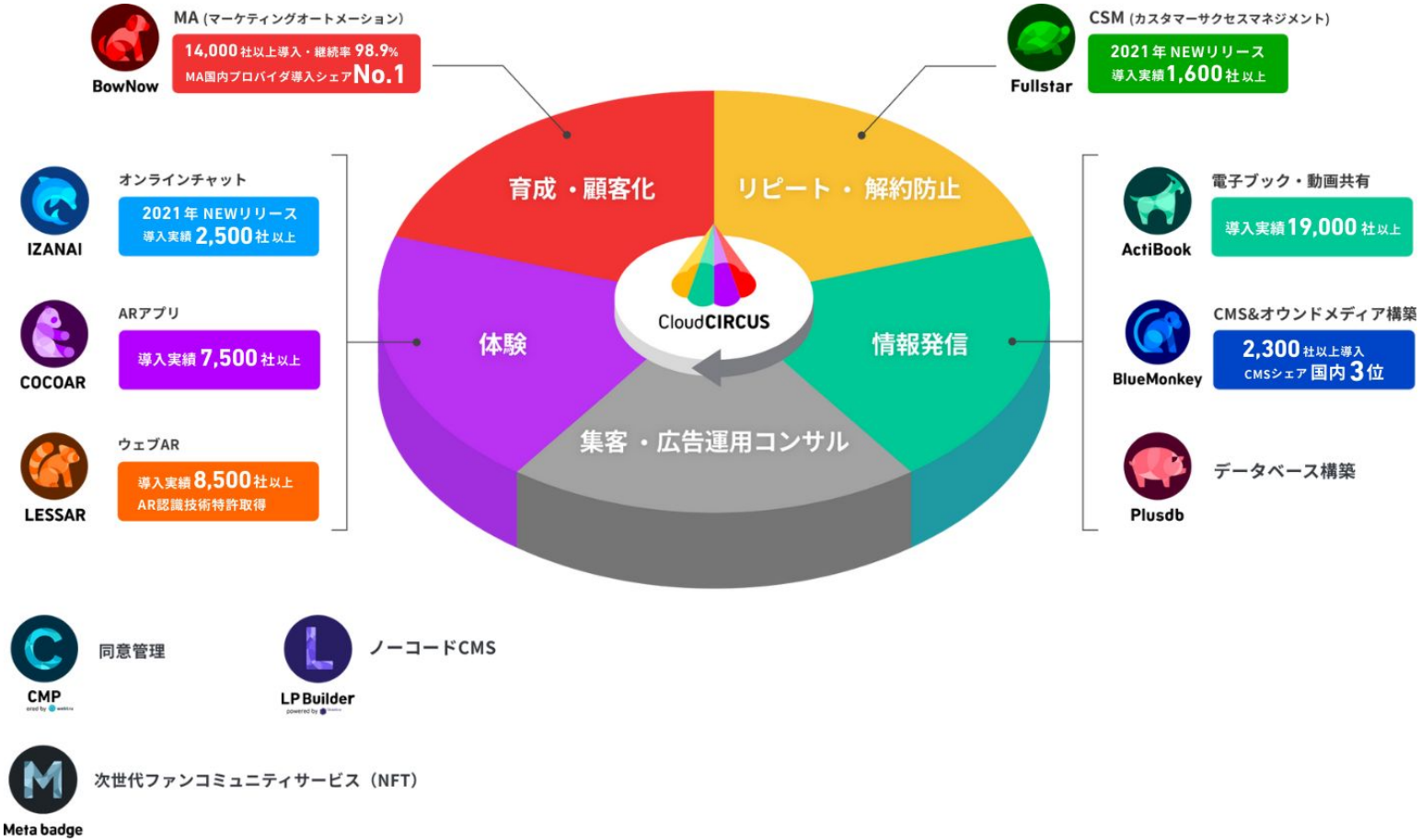
名前：朱 九霖（Shu Kyurin）

所属：クラウドサーカス株式会社
　　インフラエンジニア

経歴：
- WebARサービスの開発
- AWSのインフラ環境構築・運用/開発

好きなサービス：
AWS CDK、ClickHouse

CloudCIRCUS



高品質なウェブサイトを制作・運営できるCMS

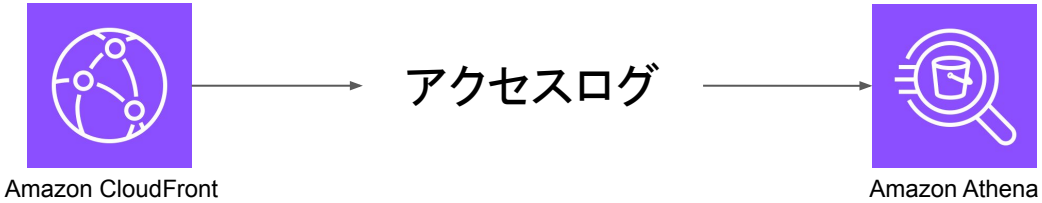
- **2000** 以上のサイトをAWS環境上でホスティング
- 各サイトで**CloudFront** (CDN)を利用(1 サイト = 1 CloudFront)
- ビジネス的な分析やシステム上で発生したエラーの解析のために CloudFrontのアクセスログを集計分析

背景

- 2000以上のCloudFrontを利用し、毎日合計で**数千万レコード**のアクセスログが発生
- CloudFrontログの集計分析は**Athena**を利用

クエリごとに課金

全ログのレコード量が膨大すぎて、**時間**と**コスト**がかかりすぎる



ClickHouseの導入

Athenaによるログの集計分析では、**時間**と**コスト**がかかりすぎる

改善したい

- 実行速度が速い
- コストを抑えられる

解決できそう

ClickHouse

集計分析に特化したDB（OSS）

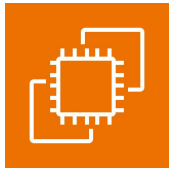
ClickHouseの導入

1. 環境構築
2. テーブル構成
3. ログのインポート
4. コストパフォーマンスの比較

構築方法

構成

- EC2 1台構成
- Amazon Linux 2023
- 4 vCPU、32 GiBメモリ
- server/clientが同一サーバ内



Amazon EC2

インストール手順

https://clickhouse.com/docs/en/install#from-rpm-packages（公式ドキュメント）

From RPM Packages

It is recommended to use official pre-compiled `rpm` packages for CentOS, RedHat, and all other rpm-based Linux distributions.

Setup the RPM repository

First, you need to add the official repository:

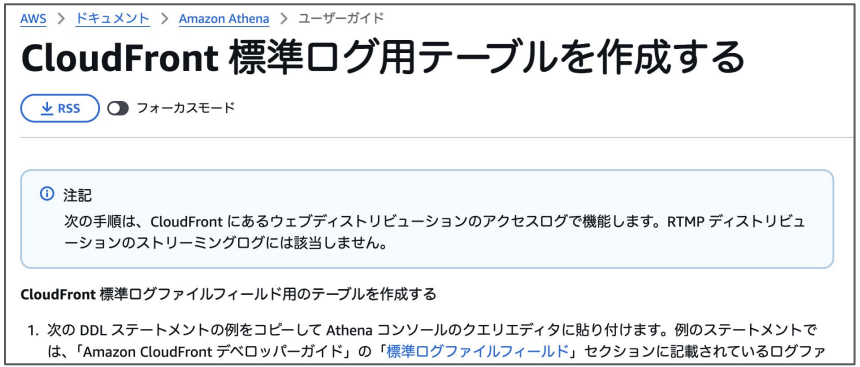
```
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://packages.clickhouse.com/rpm/clickhouse.repo
```

テーブル構成

```
CREATE TABLE cloudfront_logs
(
    `date`          Date,
    `time`          String,
    `request_ip`    String,
    `method`        String,
    `uri`           String,
    `status`        String,
    `host header`   String,
    ...
)
ENGINE = MergeTree
PARTITION BY toYYYYMMDD(date)
ORDER BY (host_header, date)
```

（AWS Athenaの公式ドキュメント）

https://docs.aws.amazon.com/ja_jp/athena/latest/ug/create-cloudfront-table-standard-logs.html



補足

`status: Int -> String`

（'000'の値が存在するため）

テーブル構成

```
CREATE TABLE cloudfront_logs
(
    `date`        Date,
    `time`        String,
    `request_ip`  String,
    `method`      String,
    `uri`         String,
    `status`      String,
    `host header` String,
    ...
)
ENGINE = MergeTree
PARTITION BY toYYYYMMDD(date)
ORDER BY (host_header, date)
```

一定期間のみ残すため

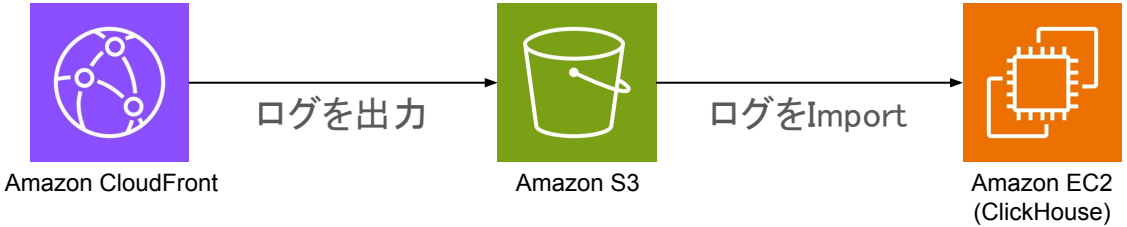
（古いものから削除する）

host_header: ドメイン名

date: 日付(YYYY-MM-DD)

各サイト(ドメイン)に関する
日付ごとの集計クエリが多いため

CloudFrontログの仕様



- CloudFrontログはS3上に保存される（常時追加）
- S3にてドメイン名ごとにフォルダ分けしている

ログのインポート

```
INSERT INTO cloudfront_logs
SELECT *
FROM
s3('https://cloudfront-logs-example-bucket.s3
.ap-northeast-1.amazonaws.com/abc.example.com
/*2024-08-*.gz', 'TabSeparated')
SETTINGS
input_format_tsv_skip_first_lines = 2,
format_tsv_null_representation = '-'
```

名前	タイプ
E1HT1DOENJW4Z3.2024-08-01-00.15fa4bbe.gz	gz
E1HT1DOENJW4Z3.2024-08-01-00.58c68e7e.gz	gz
E1HT1DOENJW4Z3.2024-08-01-00.6f4fd6b4.gz	gz
E1HT1DOENJW4Z3.2024-08-01-00.790dd555.gz	gz
E1HT1DOENJW4Z3.2024-08-01-00.98d026d4.gz	gz

abc.example.com/*2024-08-*.gz

ドメイン：abc.example.com

集計期間：2024年8月

ログのインポート

```
INSERT INTO cloudfront_logs
SELECT *
FROM
s3('https://cloudfront-logs-example-bucket.s3
.ap-northeast-1.amazonaws.com/abc.example.com
/*2024-08-*.gz', 'TabSeparated')
SETTINGS
input_format_tsv_skip_first_lines = 2,
format_tsv_null_representation = '-'
```

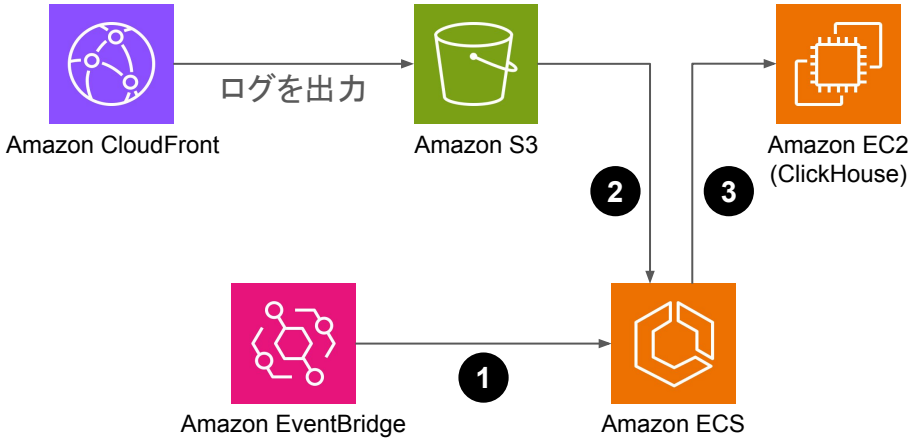
`input_format_tsv_skip_first_lines=2`

=> 頭2行(ヘッダー)を除外

`format_tsv_null_representation='-'`

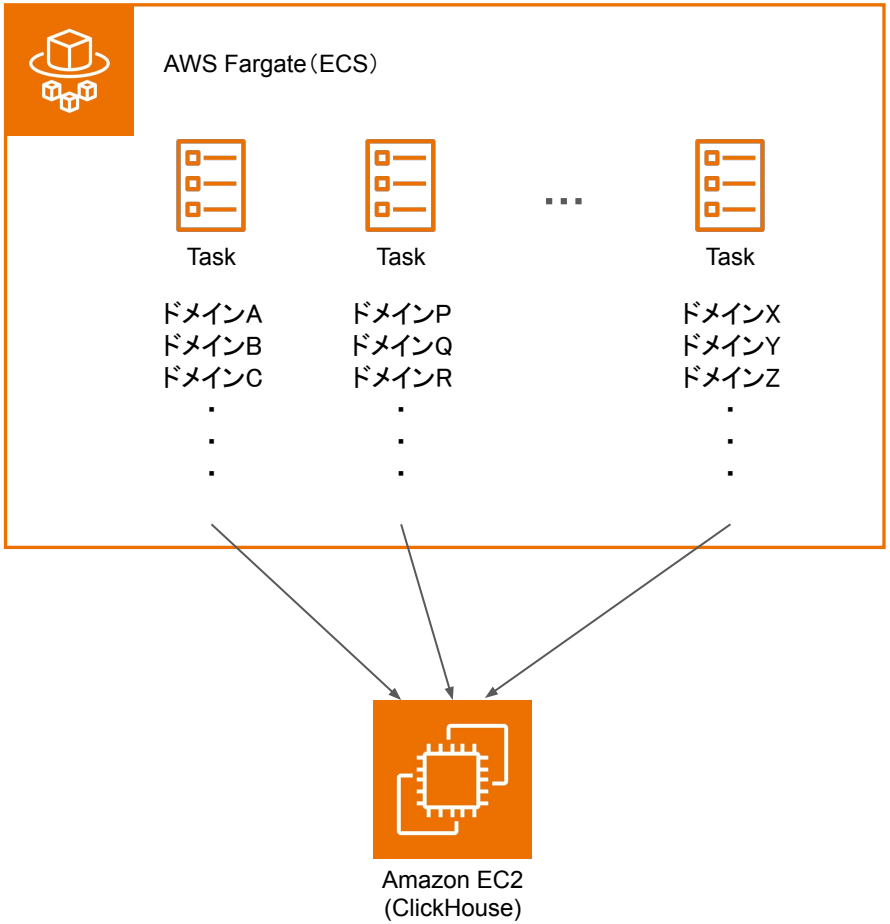
=> 文字列'-'を Null に変換

インポートの自動化



1. EventBridge SchedulerでECSのTaskを定期的に実行
2. S3からフォルダ（ドメイン）一覧を取得
3. 取得してきたドメインごとに、対象期間のログを取得・挿入

インポートの並列化



全サイトのログを一気に挿入すると**数時間**かかる

ドメインを複数のTaskに分散し、
Taskごとにログの挿入を並列実行

30分以内で完了できるように！

（20 Task、32GBメモリ使用率 80%↑）

構成の比較

ClickHouse

- 1 テーブル
- 半年分のログのみ保存
- 30億レコード
- 300GB
- ローカルストレージ（EBS）

Athena

- 2000 テーブル（全サイト分）
- AWS S3がデータソース
- 600GB以上

パフォーマンスの比較

例）「ar-go.jp の月間の日毎アクセス数」を集計するクエリ

host_header	date	status	count()
ar-go.jp	2024-08-01	200	71935
ar-go.jp	2024-08-02	200	69630
ar-go.jp	2024-08-03	200	47014
ar-go.jp	2024-08-04	200	66253
ar-go.jp	2024-08-05	200	77899
ar-go.jp	2024-08-06	200	71604
ar-go.jp	2024-08-07	200	75262
ar-go.jp	2024-08-08	200	61293
ar-go.jp	2024-08-09	200	50838
ar-go.jp	2024-08-10	200	32480

```
SELECT
    host_header,
    date,
    status,
    count()
FROM
    cloudfront_logs
WHERE
    date >= DATE '2024-08-01'
    AND date < DATE '2024-09-01'
    AND status = '200'
    AND host_header = 'ar-go.jp'
GROUP BY
    date, host_header, status
ORDER BY
    date
```

パフォーマンスの比較

ClickHouse

```
SELECT
    host_header,
    date,
    status,
    count()
FROM
    cloudfront_logs
WHERE
    date >= DATE '2024-08-01'
    AND date < DATE '2024-09-01'
    AND status = '200'
    AND host_header = 'ar-go.jp'
GROUP BY
    date, host_header, status
ORDER BY
    date
```

Athena

```
SELECT
    host_header,
    date,
    status,
    count()
FROM
    ar-go_jp_cf_logs
WHERE
    date >= DATE '2024-08-01'
    AND date < DATE '2024-09-01'
    AND status = '200'
GROUP BY
    date, host_header, status
ORDER BY
    date
```

パフォーマンスの比較

ClickHouse

```
SELECT
    host_header,
    date,
    status,
    count()
FROM
    cloudfront_logs
WHERE
    date >= DATE '2024-08-01'
    AND date < DATE '2024-09-01'
    AND status = '200'
    AND host_header = 'ar-go.jp'
GROUP BY
    date, host_header, status
ORDER BY
```

Elapsed: 0.043 sec.
Processed 1.78 million rows, 57.22 MB

Athena

```
SELECT
    host_header,
    date,
    status,
    count()
FROM
    ar-go_jp_cf_logs
WHERE
    date >= DATE '2024-08-01'
    AND date < DATE '2024-09-01'
    AND status = '200'
GROUP BY
    date, host_header, status
ORDER BY
    date
```

実行時間: 15.987 sec
スキャンしたデータ: 2.11 GB

全サイトを集計するクエリだと・・・？

ClickHouse

```
SELECT
    host_header,
    date,
    status,
    count()
FROM
    cloudfront_logs
WHERE
    date >= DATE '2024-08-01'
    AND date < DATE '2024-09-01'
    AND status = '200'
GROUP BY
    date, host_header, status
ORDER BY
    date
```

Athena

```
SELECT
    host_header,
    date,
    status,
    count()
FROM
    xxxxx_cf_logs
WHERE
    date >= DATE '2024-08-01'
    AND date < DATE '2024-09-01'
    AND status = '200'
GROUP BY
    date, host_header, status
ORDER BY
    date
```

（すべてのテーブル！）

全サイトを集計するクエリだと・・・？

ClickHouse

```
SELECT
    host_header,
    date,
    status,
    count()
FROM
    cloudfront_logs
WHERE
    date >= DATE '2024-08-01'
    AND date < DATE '2024-09-01'
    AND status = '200'
GROUP BY
    date, host_header, status
ORDER BY
    date
```

Elapsed: 8.973 sec.
Processed 470.62 million rows, 19.25 GB

Athena

```
SELECT
    host_header,
    date,
    status,
    count()
FROM
    xxxxx_cf_logs
WHERE
    date >= DATE '2024-08-01'
    AND date < DATE '2024-09-01'
    AND status = '200'
GROUP BY
    date, host_header, status
ORDER BY
    date
```

実行時間: 15.987 sec
スキャンしたデータ: 2.11 GB

× 2000 クエリ？

コストの比較

ClickHouse

- 4 vCPU、32 GiBメモリの EC2インスタンス
- 500 GB EBS（ストレージ）
- 常時稼働

281.02 USD / 月

- 固定費
- AWSの割引プランによるコスト削減可能

Athena

- 1クエリにつき
  5.00 USD per TB of data scanned
- 平均 10 GB データスキャン
- 月 2000 クエリ（=全サイト）

例

97.66 USD / 月

- クエリ数によるコスト増減がある

まとめ

- CloudFrontログのClickHouseへの導入を
  AWSリソースによる仕組み化を行った

- ログ分析をAthenaからClickHouseに移行したことで、
  時間とコストの削減を実現することができた
  - クエリ実行の度にお金の心配をしなくて良くなった