



ドローン運行監視

—Amazon AthenaのBucketingとAmazon S3 Selectの活用—

2023/04/21 AWS Summit Tokyo 2023
株式会社ナイルワークス
山田直行

山田 直行 *Naoyuki Yamada*

株式会社ナイルワークス
シニアエンジニア





社名 株式会社ナイルワークス

設立 2015年1月7日

所在地 本社 : 東京都千代田区
研究所 : 埼玉県さいたま市
エリア拠点 : 北海道

代表 代表取締役社長 小嶋 康弘

従業員 34名

資本金 9,782万円

株主 住友商事株式会社
ヤマハ発動機株式会社
三井住友ファイナンス&リース株式会社
ダイハツ工業株式会社
住友化学株式会社
クミアイ化学工業株式会社
未来創生2号ファンド(運営者:スパークス・グループ株式会社)
全国農業協同組合連合会
Drone Fund 2号
アグリビジネス投資育成株式会社
他5名

製品・サービス

Drone

農業用ドローンの開発、製造、販売



Digital

デジタル農業の技術開発、サービス提供



Matching

農作業マッチングサービス



改良の歴史

GO号機 液剤散布



1号機完成
ドローン飛行制御システム 完成
(フライトコントローラ)

G4号機 液剤散布



完全自動飛行制御システム 完成
(自動飛行、自動薬剤散布ポンプON・OFF)

G5号機 粒剤散布



粒剤散布機を開発

2015
会社設立

研究開発

2016

2017

2018

販売

2019

2020

Nile-T18
液剤散布・生育診断



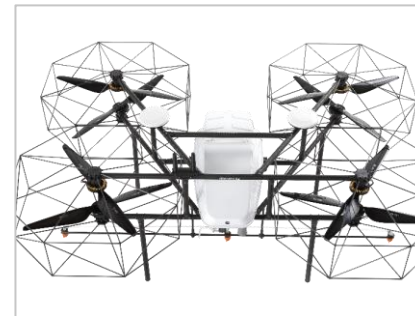
タブレット操縦

Nile-T19
液剤散布・生育診断機



日本初のLTE通信搭載機体を実用化

Nile-T20
液剤散布・生育診断機



改良モデル

飛行イメージ



圃場の形にあわせて、
自動で飛行



作物上空30~50cmから、
株元へ散布



操作は簡単で、
特別なスキルは不要

Nile-JZ

空からの農業
誰でも、適切なタイミングに、肥料・農薬を散布

2023年 発売予定



本製品は、国立研究開発法人農業・食品産業技術総合研究機構(農研機構/NARO)が事業実施主体である国際競争力強化技術開発プロジェクト「安全安心な農業用ハイスペックドローン及び利用技術の開発」を受託(2021年6月)したハイスペックドローン開発コンソーシアム(※)の事業で開発された機体をベースにしております。

※ハイスペックドローン開発コンソーシアム

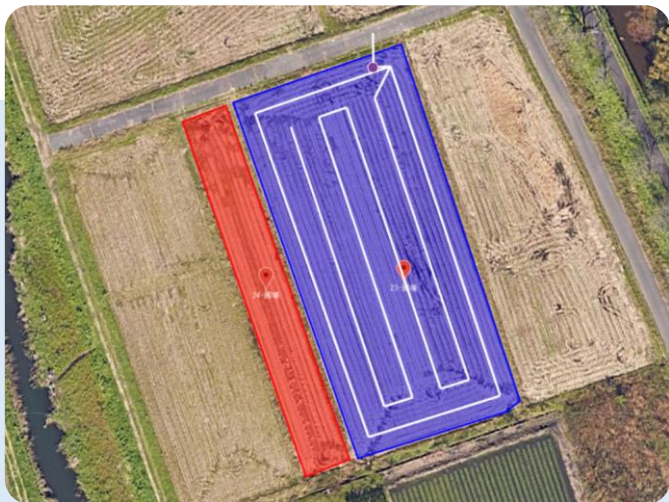
代表機関 : ヤマハ発動機株式会社

共同研究機関 : 〈企業〉 株式会社ザクティ、株式会社ザクティエンジニアリングサービス、株式会社ナイルワークス、ヤンマーアグリ株式会社

〈研究機関〉 農研機構、大分県農林水産研究指導センター、鹿児島県農業開発総合センター、佐賀県農業試験研究センター、長崎県農林技術開発センター

(C) Nileworks Inc.,

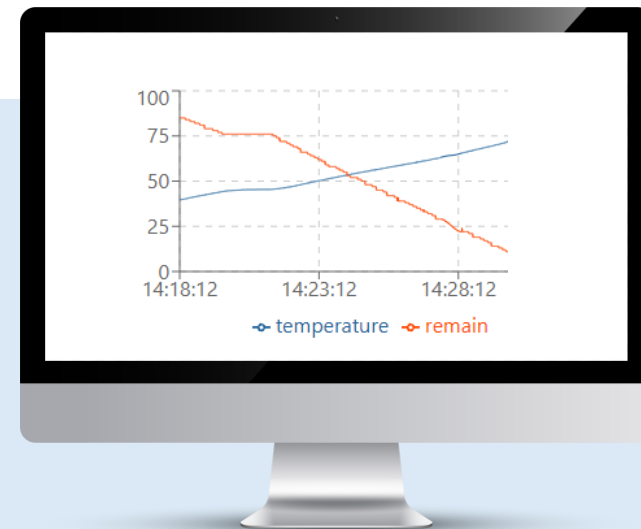
LTEの活用



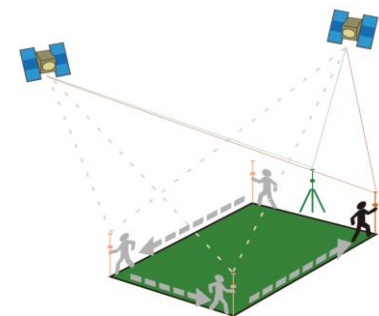
飛行経路は、自動生成



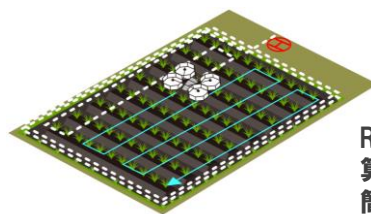
アプリ・プロポで、簡単操作



リアルタイムで、飛行監視



圃場・障害物を測量し、データ化。
オンデマンドで、
すぐに飛行経路が自動生成。

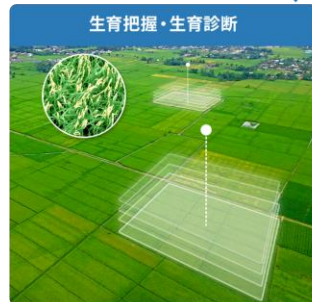
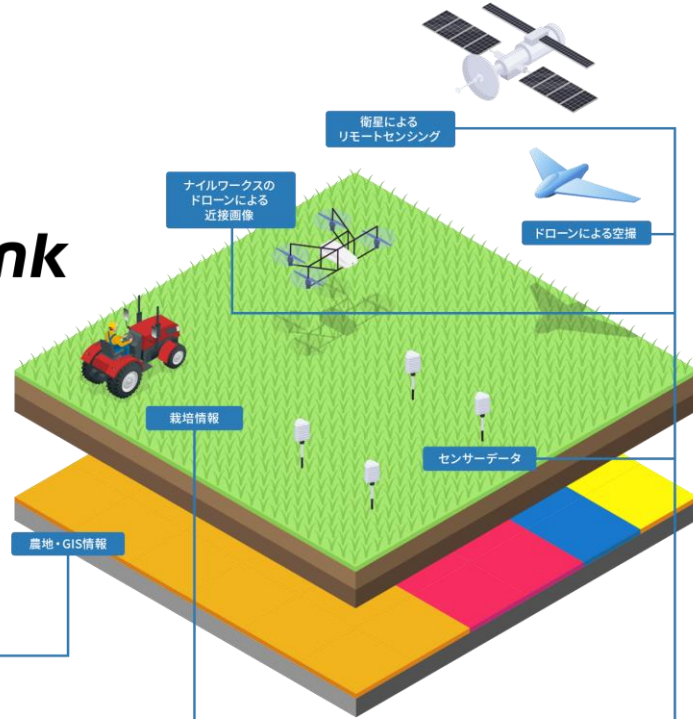


RTK-GNSSや複数センサーから
算出される位置・方位情報で、正確に飛行。
簡単な操作で、ドローンは自動飛行。



ドローンにはLTEを搭載し、
飛行情報を監視サーバへ送信。
サポートに活用。

Digital transformation



ブリッジブレスト
シンプルな操作で、防除作業をマッチング

防除を**依頼**する人 防除を**受託**する人

依頼者：適期に防除を実現 嬉しいポイント 受託者：業務拡大のチャンス！

- 省力化** 防除を任せる相手が簡単に見つかる
- コスト削減** 自身でのドローン導入・操作習得が不要
- 安心の品質** 適期に防除が可能なので安心
- 収入アップ** 新規顧客の発掘、防除エリア拡大、時間の有効活用
- 効率化** スケジュール・作業管理が見える化
- 適正な報酬** 価格交渉の基準となる推奨価格があるので納得



2023年4月 サービス開始

Amazon AthenaのBucketingと Amazon S3 Selectを活用した ドローンの運行監視システム

過去の通信ログを安価にクエリする工夫



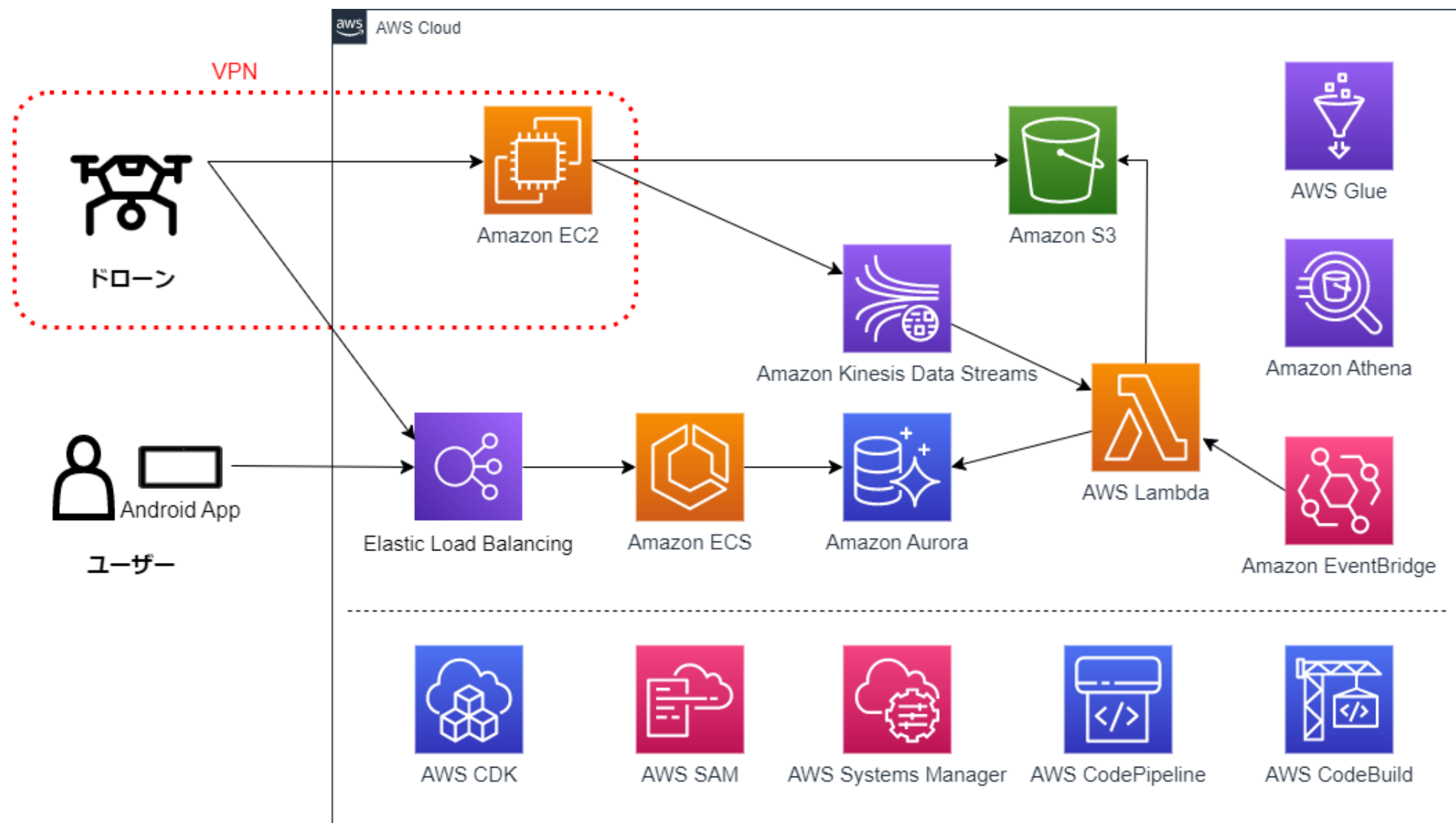


ナイルワークスの農業用ドローン「Nile-T20」では、リアルタイムおよび過去の飛行を監視・閲覧・分析する「運行監視システム」を開発・運用しています。

すべての通信ログ・飛行ログをリアルタイムかつ永続的にAWSに保存していつでもフロントエンドのウェブアプリケーション (Next.js) から参照できるようにしています。

Amazon AthenaのBucketingとAmazon S3 Selectを活用し、それを安価に実現する工夫をご紹介します。

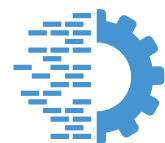
AWS活用



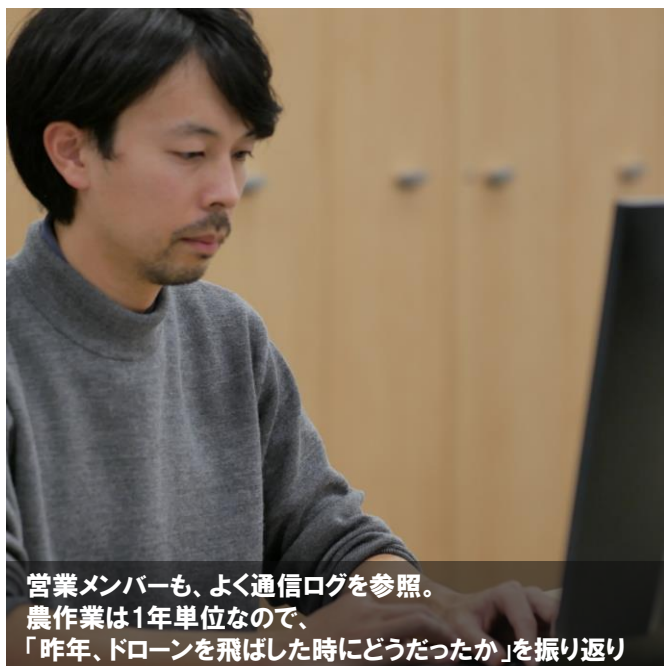
ドローンの通信ログ



ドローンには、LTEを搭載
インターネット+VPN経由で、クラウドと通信



ドローンは起動中、
mavlinkというドローン用の軽量なプロトコルで
ログを常時クラウドに送信



営業メンバーも、よく通信ログを参照。
農作業は1年単位なので、
「昨年、ドローンを飛ばした時にどうだったか」を振り返り

ユーザーサポート

異常検知

事故分析

開発に活用

実績の振り返り

課題

過去5年の数万件におよぶ詳細な飛行ログを
どのようにAWSに蓄積し、
アプリケーションから利用可能にするか。

100GBを超えているので、
今後を考えると
全てをAmazon Auroraに載せ続けるのは
ためらうデータ量です。



社内ツール



機材管理 (本番環境)

AC200095のMAVLink解析

日時: 2021/08/03 14:33:33 ~ 14:45:56
AC200095 (event) (log)
TB200025 (event) (log)
BS200025 (event) (log)
CA200095 (event)
ユーザー: [redacted]
組織名: [redacted] (FieldViewer)
圃場名: 4-圃場 (FieldViewer)
測量結果: 12802.3m² (DB)
飛行経路: 波割散布飛行 800m10a飛行幅4m (FieldViewer)
バッテリー: 8677(MPA03006) (サイクル数:1 最大温度:38)
バッテリー: 8678(MPA03007) (サイクル数:1 最大温度:38)

Px4log解析

詳細を表示 グラフを拡大

温度・残量(バッテリー0)

温度・残量(バッテリー1)

電圧

スロットル

[2021-08-03 14:46:06] [753.1sec] 高さ:0.0m | Ac位置 | (Tb位置情報なし) (Bs位置情報なし) | 圃場名非表示

0 376.5sec 753.1sec

機体ステータスメッセージ

日時	種別	メッセージ
2021/08/03 14:45:59	STATUSTEXT	[レベル: 6] Landing detected
2021/08/03 14:45:59	STATUSTEXT	[レベル: 6] [blackbox] stopped (0 drops)
2021/08/03 14:45:57	MISSION_STATUS	6:LANDING → 0:LANDED に更新
2021/08/03 14:45:57	STATUSTEXT	[レベル: 6] [cmd] DISARMED by arm/disarm component command

データ設計における 機能要件



ドローンは、日中のみ飛行するため、
日付をまたいだデータは存在しない前提



ドローンごとのログなので、
「日付」「ドローンの機体番号」で、パーティショニング可能



時間(〇時〇分～〇時〇分)の部分は、基本的には飛行中の時間をクエリできればよい
ただし、離陸前や着陸後の状態を知りたい場合もあるので、時間は可変である必要がある



ウェブアプリケーション(Next.js)から、
GraphQLで、日時とドローンの機体番号を引数にクエリしたい



非機能要件



インフラコストを
なるべく安く実現したい



長期間運用しても、
パフォーマンスが悪化しないようにしたい



過去の飛行ログを見たいというニーズは、
ロングテールなもの



直近のデータほどよく参照される
ただし、参照されいログは、ほぼ参照されない



深夜に飛行ログをみるニーズは、
ほぼゼロ



常時稼働しているデータベースに
すべての通信ログを入れておくと、
データ量とパフォーマンスが問題になる



なるべくサーバーレス寄りの
ソリューションを使いたい



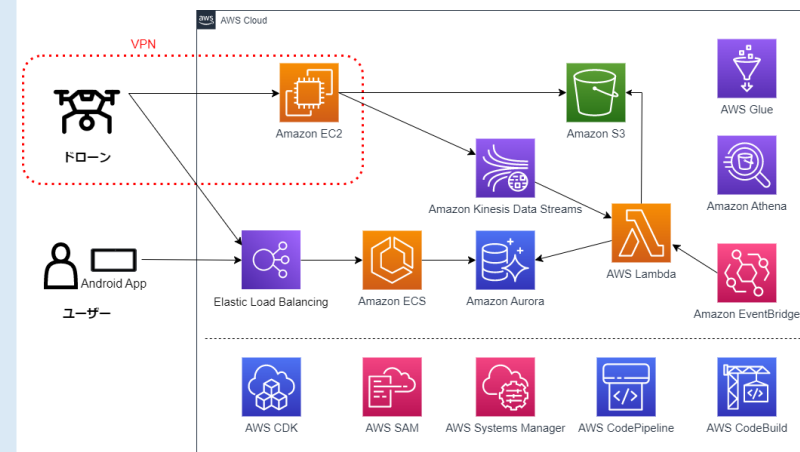
採用した方法

リアルタイムで表示する画面は、
Amazon Auroraに
Amazon Kinesis Data Streams
+ AWS Lambda経由で
INSERTし、
GraphQLのSubscriptionを利用

Amazon AuroraのPostgreSQLには、
当日分のみデータを保持
1日1回深夜に過去のデータを消去

過去の飛行を表示する画面は、
GraphQL Queryで
Amazon S3 Selectを
内部的に使ったResolverを
呼び出す

Amazon AthenaのBucketingを利用し
year/month/day/drone ID で
1ファイルになるようにまとめておき、
Amazon S3 Selectできる状態にしておく



Amazon AthenaのBucketingとは

Amazon Athena のパフォーマンスチューニング Tips トップ 10

<https://aws.amazon.com/jp/blogs/news/top-10-performance-tuning-tips-for-amazon-athena/>

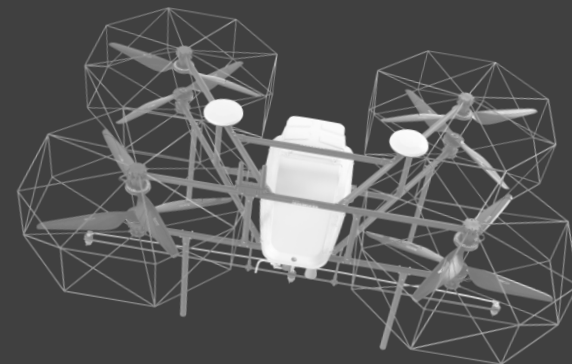
Partitioning and bucketing in Athena – Amazon Athena

https://docs.aws.amazon.com/ja_jp/athena/latest/ug/ctas-partitioning-and-bucketing.html



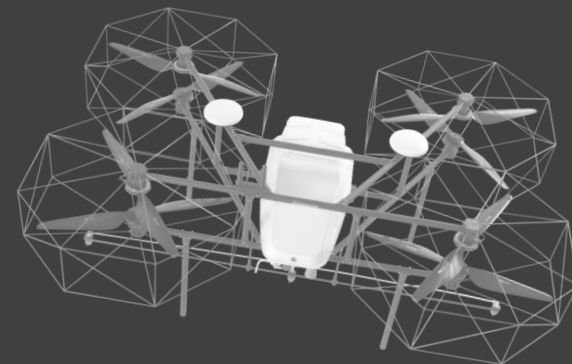
**Amazon Athenaを使い、
Amazon S3のファイルを別の単位で配置しなおす方法**

Create Table As (CTAS) 1日1回深夜にバッチ実行



```
CREATE TABLE xxxxxxxxxxxx WITH ( format='JSON',  
    external_location='s3:// {mavlink_bucket_name} /xxxxxxxx/year= {year} /month= {month} /day= {day} /',  
    partitioned_by = ARRAY ['drone_id'],  
    bucketed_by = ARRAY ['mavpackettype'],  
    bucket_count = 1 ) AS  
SELECT received_at,  
    mavpackettype,  
  
    (中略)  
FROM "default"."mavlink_parsed"  
WHERE received_at LIKE ' {year} - {month} - {day} %'  
ORDER BY received_at AS
```

Amazon S3 Selectを呼び出す単位で ファイルを1つにまとめる



Amazon S3 > バケット > [redacted] > [redacted] > year=2021/ > month=08/ > day=01/ > [redacted] /

[redacted]

オブジェクト | プロパティ

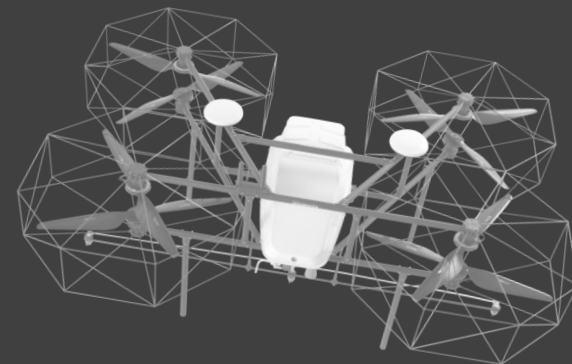
オブジェクト (1)

オブジェクトは、Amazon S3 に保存された基本的なエンティティです。Amazon S3 インベントリ [\[リンク\]](#)を使用して、バケット内のすべてのオブジェクトのリストを取得できます。他のユーザーが自分のオブジェクト的にアクセス権限を付与する必要があります。詳細は [こちら \[リンク\]](#)

🔍 プレフィックスでオブジェクトを検索

<input type="checkbox"/>	名前 ▲	タイプ ▼	最終更新日時 ▼	サイズ ▼
<input type="checkbox"/>	📄 20211206_064410_00019_qt8wr_bucket-00000.gz	gz	2021/12/07 08:54:23 AM JST	49.1 MB

GraphQLのResolver 側



```
s3_select_sql = ""SELECT
(中略)
FROM s3object s
WHERE s.received_at > '{received_datetime_gt}' AND s.received_at < '{received_datetime_lte}'
AND s.mavpackettype IN ({"".join(",".join([f"{x}" for x in message_type]))})""
response = s3_client.select_object_content (
    Bucket=get_s3_mavlink_bucket_name (),
    Key=s3_key,
    Expression=s3_select_sql,
    ExpressionType="SQL",
    InputSerialization= {"CompressionType": "GZIP", "JSON": {"Type": "LINES"}},
    OutputSerialization= {"JSON": {"RecordDelimiter": "\n"}} ,)
```

結果と評価

平均3秒程度

最大10秒程度で、結果を返せる

ウェブアプリから、過去データを閲覧する要件としては、許容範囲
本格的な分析用途ではAmazon Athenaを利用

安価

Amazon S3のファイル取得と同料金

セキュア

Amazon S3 : GetObjectの権限だけで、OK

スケーラブル

Amazon Athenaは同時実行数に制限があるので、
ウェブアプリケーションから直接呼び出すのには不適



まとめ

データをS3に持っておくと、Amazon Athena + AWS Glueの組み合わせで、面白い使い方ができます。パーティションが分かりやすく区切ることのできるタイプのデータであれば、Amazon S3単体でも十分にデータベースとして機能してくれます。

RDSやDynamoDBに入れてそこからクエリするのが王道だと思いますが、時系列のデータを安価に保持・クエリする方法として、参考になれば幸いです。

今後は、Amazon Timestreamを試している途中であり、Kinesis Data Analyticsがリアルタイム分析用途に使えないかを模索していきたいと思っています。