アカウント周りの処理とかについて

みずあめ(@mizuameisgod)

@JAWS-UG 茨城 #1 Education-JAWSコラボ回
2025/03/15

About Me

名前: 佐藤 良

年齢: 18歳

所属: 筑波大学情報学群情報科学類1年

ついったー: @mizuameisgod

web: https://mizuame.works/

やってること: ゲームクライアント、インフラ、機械学習、言語モデル?、セキュリティ、web、ネットワーク etc… 興味があることは割と何でもかじっています

gametreeというwebアプリをリリースした

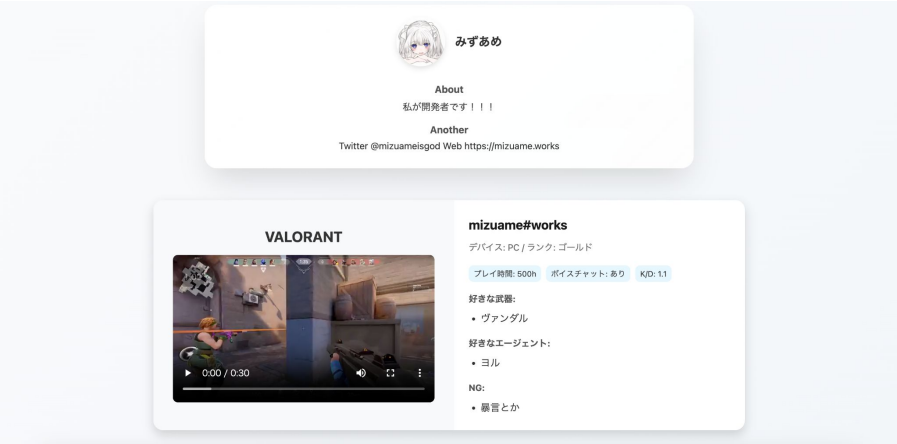
← ポスト

みずあめ
@mizuameisgod

[宣伝]
個人開発していたgametreeというwebサービスをリリースしました！
ゲーマー向けのプロフィール作成サービスになっています！！！
良ければ登録お願いします。

gametree @gametree_info · 1月12日
#gametree というゲーマー向けのプロフィール作成webアプリを開発しています！
現在βテスト中で #APEX #VALORANT #ow2 #Fortnite にしか対応していませんが、今後増やしていく予定です！
良ければ登録お願い致します。(リンクはリプにあります)
このスレッドを表示

https://gametree.info/@mizuame



gametreeとは

GUIでぽちぽちやるだけで、

ゲーマー向けのプロフィールサイトを作成できる。

動画も設定可能。

ゲーマー版Linktreeみたいな

構成

バックエンド: AWS , Cloudflare

フロント: ただのMPA(Pages)

Cloudflare
Pages

CLOUDFLARE
Workers

SNSにおけるユニークIDと通常ID

まずcognitoからはsubというUUIDが振られる。

が、

gametree.info/@{username}

ルーティングしたい。

つまり、

sub(アカウント作成時にcognitoから自動割り当て、ユニーク、変更不可)
user_name(アカウント作成時に開発者側で割り当てる、ユニーク、変更可)
display_name(アカウント作成時に開発者側で割り当てる、Notユニーク、変更可)

DynamoDBのテーブル設計

自分もデータベースはそこまで詳しくないので以下の説明は大きく間違っている可能性があります。

DynamoDBは通常のSQLなどと大きく異なり、パーティションキー、ソートキーに対してしか基本クエリがかけられない。これらの組み合わせでプライマリキーを構成。

→逆に言うと、これらの組み合わせはユニークであると保証されている

scanでAttributesクエリをかけることもできるが、テーブルを総なめすることになるため、ダメ絶対。

Global secondary indexesでインデックスをはり、Attributesに対してもクエリはかけられるが、整合性が保証されていない。

基本一つのテーブルにするのが推奨されている。

CognitoとDynamoDBの連携

DynamoDBのテーブル設計:

	PK	SK	displayname	item_type	profile	sub	username	username_lower
1	USER#{sub}	PROFILE	みずあめ		{"about_profiles":{...	{sub}	mizuame	mizuame
2	USERNAME#mizuame	METADATA		Username		{sub}	mizuame	mizuame

CognitoのLambdaトリガー

Cognitoの拡張機能から、Lambdaトリガーを設定することができる。

確認後に設定することで、ユーザーがemail認証をする or Googleアカウントでサインアップした時にLmabdaを動かすことが出来る。

DynamoDBのトランザクション処理

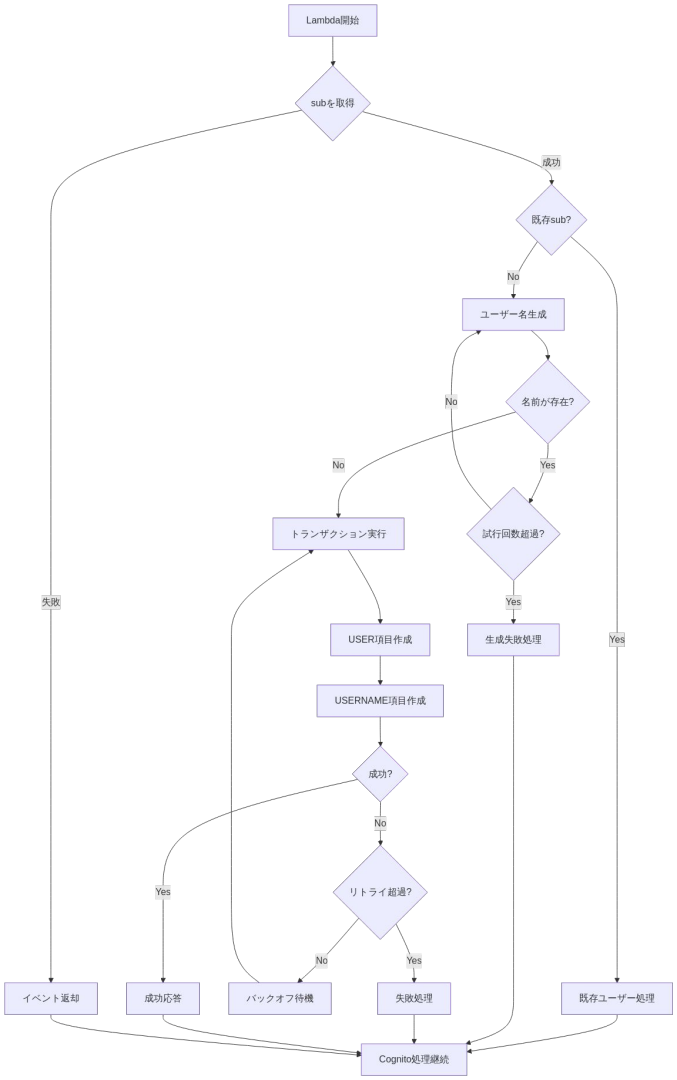
DynamoDBは楽観的ロックを採用しており、トランザクション処理時にテーブルロックは行わない。

transact_write_itemsで

USER#{sub}と
USERNAME#{username_lower}

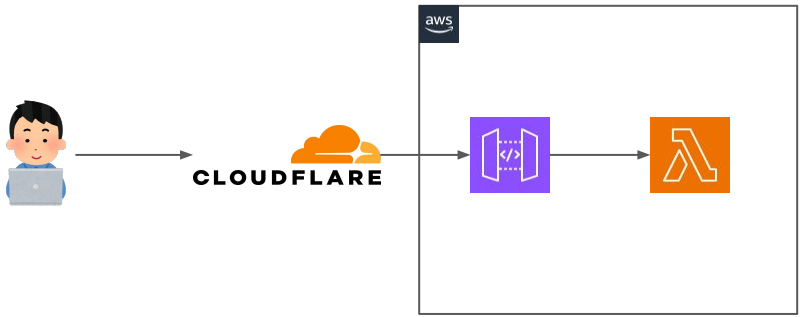
のアイテム両方の作成が成功しないと、全てがキャンセルされる。

attribute_not_exists(SK)で無いことを確認した上で、書き込みを実行。



REST API 設計

API Gatewayを用いて
RESTAPIを作成した。



ステージ

- v1
  - /
    - /users/username
    - /users/{username}/clip/{game-name}
    - /users/{username}/display-name
    - /users/{username}/icon
    - /users/{username}/profile

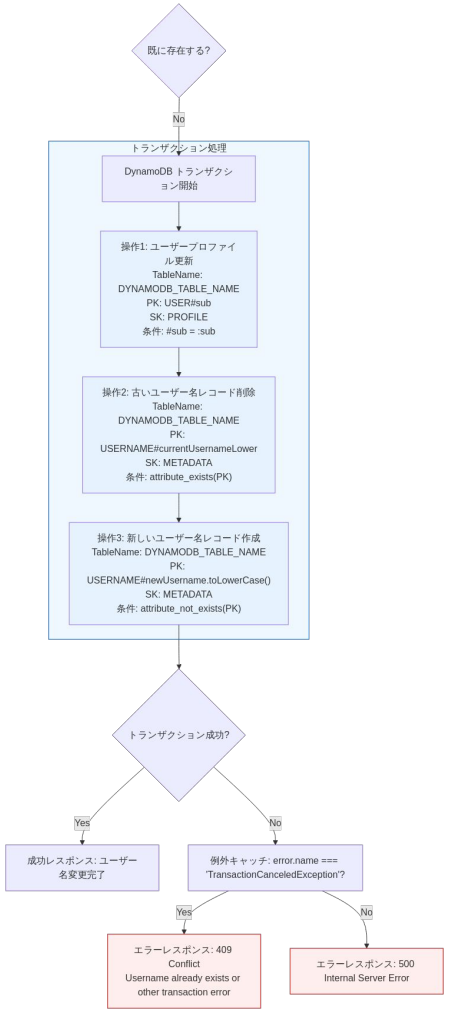
user_name変更処理

まずクライアントからJWTを受け取り、整合性を確認した後、サニタイズを行う。

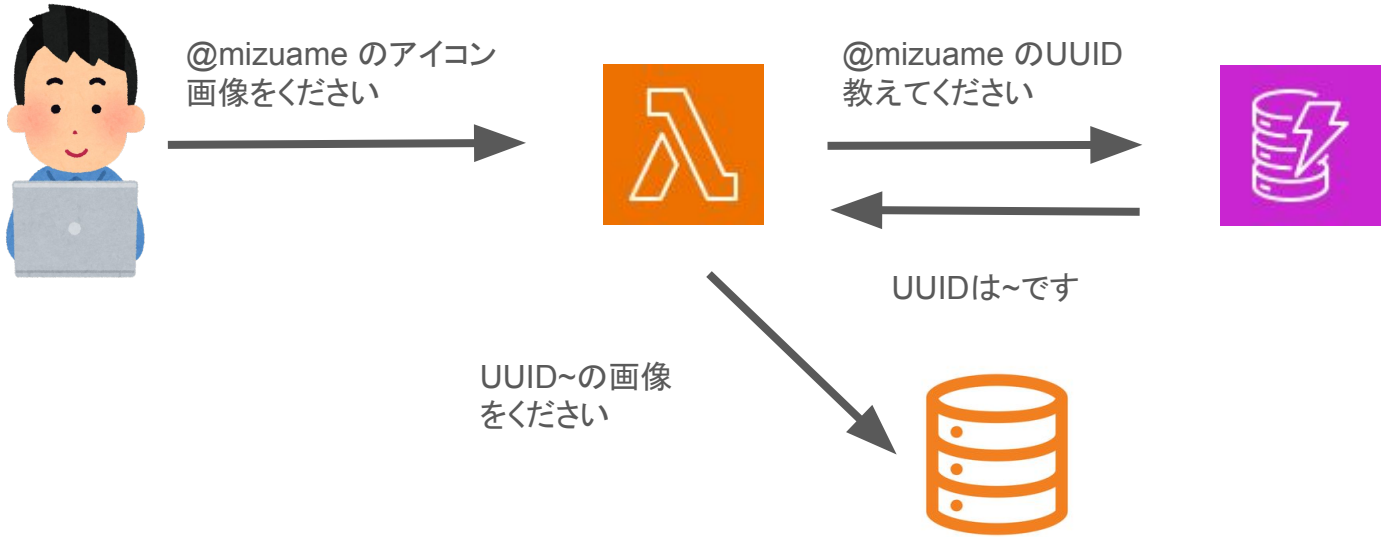
成功したら次は同じuser_nameが居ないか確認し、

成功したらトランザクション処理を開始する。

既存のレコードをコピーし、新しい名前に変えた後、古いレコードを削除し、その後新しいレコードを挿入する。どれかが失敗した時点で全ての操作が無かったことになる。



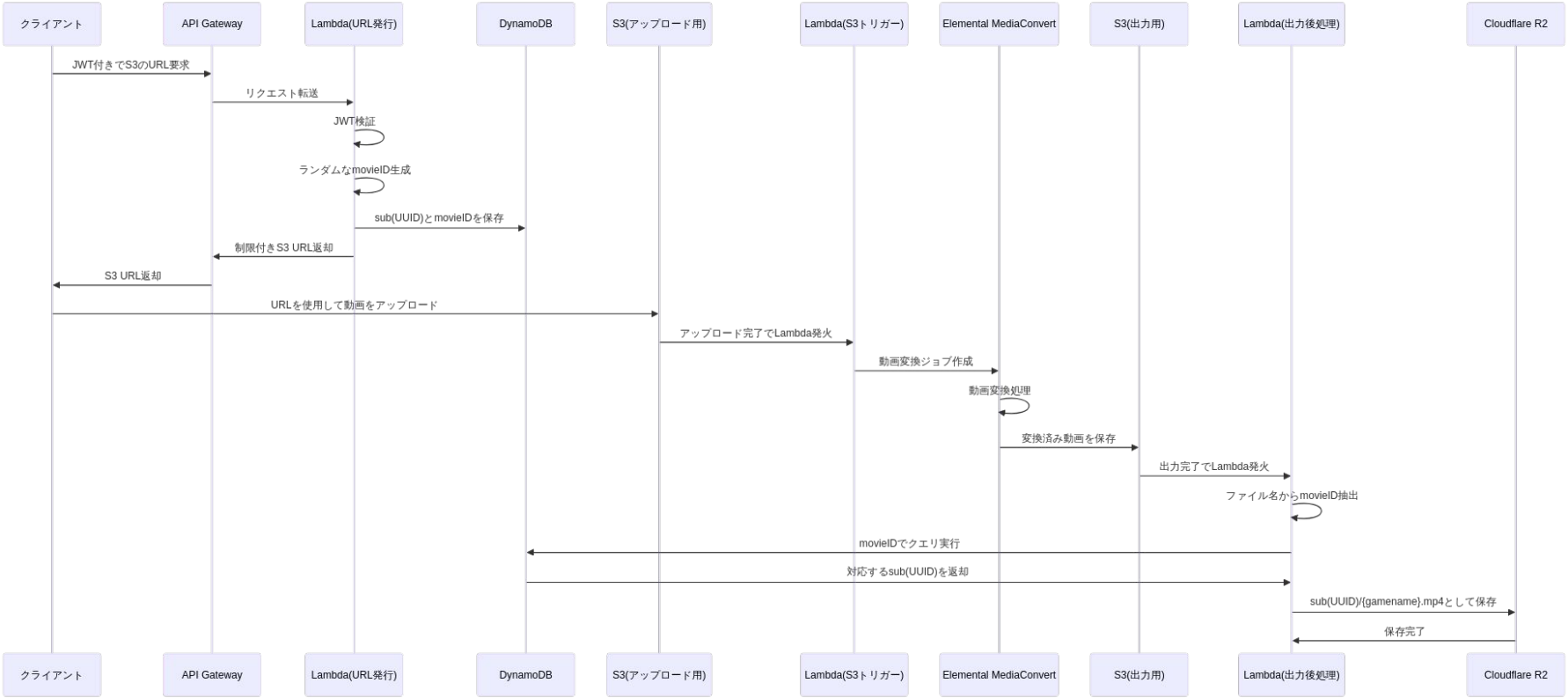
user_nameトラバーサル



動画アップロード

ゲームのクリップをアップロード出来るわけだが、そのまま配信するととんでもない脆弱性になる可能性があるので、一回変換をかける必要がある。

S3、elemental mediaconvert、R2による感動の変換



DynamoDBへの一時記録

まず、S3への署名付きアップロードを作成する際、DynamoDBに一時的にIDとsubのペアを記録する。

変換後に動画のアップロードユーザーを識別するために使用する。

又TTLが設定できる為、1時間後に自動削除される。

```
const putParams = {
  TableName: 'gametree_MoviePair',
  Item: {
    MovieID: fields.key, // PK
    Sub: sub,
    GameName: normalizedGameName,
    CreatedAt: new Date().toISOString(),
    unixtime: unixtime // TTL
  }
};
```

S3の制限付きアップロードURLの生成

制限付きURLの生成の際に、Key、有効期限、最大容量を制限したものを発行する。

Keyは

upload_abcdef123456.mp4
のようになる。

```
const presignedPost = await createPresignedPost(s3Client, {
  Bucket: bucketName,
  Key: objectKey,
  Expires: 1800, // 有効期限
  Conditions: [['content-length-range', 0, 1000000000]] // 容量制限
});

return {
  uploadUrl: presignedPost.url,
  fields: presignedPost.fields
};
```

https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPOST.html

elemental mediaconvertに入れる

S3にアップロードされ次第、自動でLambda
が発火し、設定通りにelemental
mediaconvertに突っ込まれ、変換される。

ドキュメント見ながらやったが設定がこけたの
で、lemental mediaconvertのUIから設定を
生成した。

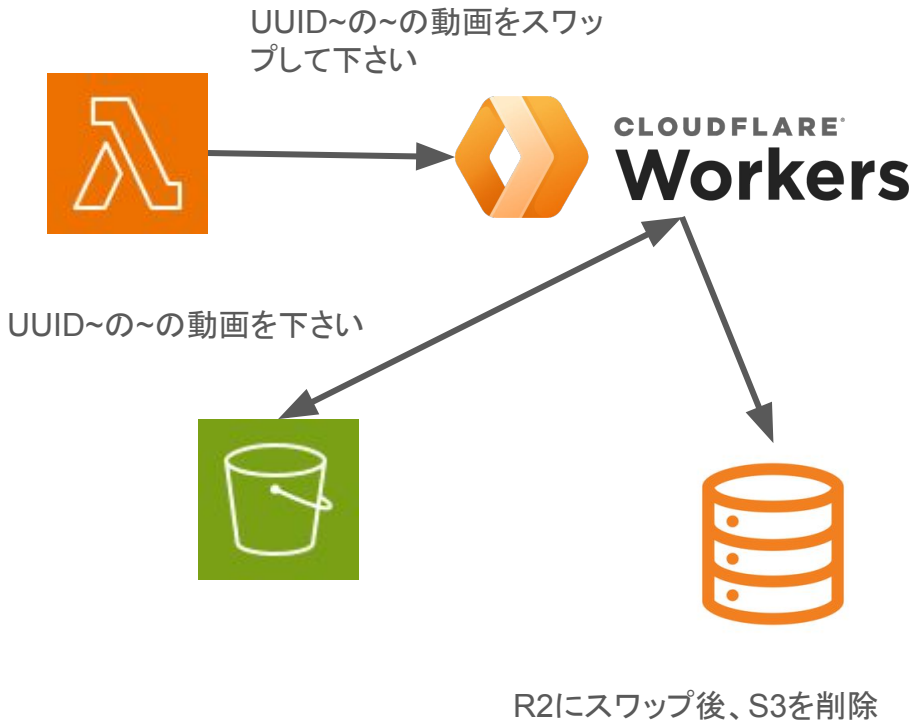
```
job_settings = {
    "TimecodeConfig": {
        "Source": "ZEROBASED"
    },
    "OutputGroups": [
        {
            "Name": "File Group",
            "OutputGroupSettings": {
                "Type": "FILE_GROUP_SETTINGS",
                "FileGroupSettings": {
                    "Destination": f"s3://{destination_bucket}/",
                    "DestinationSettings": {
                        "S3Settings": {
                            "StorageClass": "STANDARD"
                        }
                    }
                },
                "Outputs": [
                    {
                        "ContainerSettings": {
                            "Container": "MP4",
                            "Mp4Settings": {
                                "MoovPlacement": "PROGRESSIVE_DOWNLOAD"
                            }
                        },
                        "VideoDescription": {
                            "ColorMetadata": "INSERT",
                            "CodecSettings": {
                                "Codec": "H_264",
                                "H264Settings": {
                                    "MaxBitrate": 30000000,
                                    "RateControlMode": "QVBR",
                                    "QvbrSettings": {
                                        "QvbrQualityLevel": 7
                                    },
                                    "SceneChangeDetect": "TRANSITION_DETECTION"
                                }
                            },
                            "Width": 854,
                            "Height": 480
                        },
                        "AudioDescriptions": [
                            {
                                "AudioSourceName": "Audio Selector 1",
                                "CodecSettings": {
                                    "Codec": "AAC",
                                    "AacSettings": {
                                        "Bitrate": 96000,
                                        "CodingMode": "CODING_MODE_2_0",
                                        "SampleRate": 48000
                                    }
                                }
                            }
                        ]
                    }
                ]
            }
        ],
        "Inputs": [
            {
                "FileInput": f"s3://{source_bucket}/{source_key}",
                "AudioSelectors": {
                    "Audio Selector 1": {
                        "DefaultSelection": "DEFAULT"
                    }
                },
                "VideoSelector": {
                    "ColorSpace": "REC_709",
                    "ColorSpaceUsage": "FORCE",
                    "SampleRange": "LIMITED_RANGE"
                },
                "InputClippings": [
                    {
                        "StartTimecode": "00:00:00:00",
                        "EndTimecode": "00:00:30:00"
                    }
                ],
                "TimecodeSource": "ZEROBASED"
            }
        ]
    }
```

workersにスワップさせる

AWS Lambdaは実行時間で課金されるのに加え、実行数に制限がある為、実際の cloudflare R2へのスワップはcloudflare workersが行う。

LambdaはworkersのエンドポイントにUUID~の~の動画をスワップして下さいリクエストを投げた後、DyamoDBから一時記憶を消すだけ。

workersはスワップが完了したらS3のオブジェクトを消す。



ご清聴ありがとうございました。

Cloudflareとの感動のマルチベンダーバックエンドについてはこちらをご覧ください。

https://www.docswell.com/s/mizuame/ZXEGY3-2025-02-01-UNTIL