

 この記事は最終更新日から1年以上が経過しています。

 @myokit (Keita Miyano)

S3内のファイルやフォルダをPythonで一覧取得する

Python AWS S3 boto3

最終更新日 2022年03月23日 投稿日 2022年03月13日

環境

念のためですが、動作環境は以下です

```
macOS Catalina
Python 3.8
boto3 1.21.16
```

結論

以下のls関数を用いれば、指定したバケットのパスに対して再帰的、もしくはその階層のみのファイル・フォルダ一覧を取得できます。

```
import boto3
from typing import List

def ls(bucket: str, prefix: str, recursive: bool = False) -> List[str]:
    """S3上のファイルリスト取得

    Args:
        bucket (str): バケット名
```

```

        prefix (str): バケット以降のパス
        recursive (bool): 再帰的にパスを取得するかどうか

"""

paths: List[str] = []
paths = __get_all_keys(
    bucket, prefix, recursive=recursive)
return paths

def __get_all_keys(bucket: str, prefix: str, keys: List = None, marker: str = None):
    """指定した prefix のすべての key の配列を返す

    Args:
        bucket (str): バケット名
        prefix (str): バケット以降のパス
        keys (List): 全パス取得用に用いる
        marker (str): 全パス取得用に用いる
        recursive (bool): 再帰的にパスを取得するかどうか

    """

    s3 = boto3.client('s3')
    if recursive:
        response = s3.list_objects(
            Bucket=bucket, Prefix=prefix, Marker=marker)
    else:
        response = s3.list_objects(
            Bucket=bucket, Prefix=prefix, Marker=marker, Delimiter='/')

    # keyがNoneのときは初期化
    if keys is None:
        keys = []

    if 'CommonPrefixes' in response:
        # Delimiterが'/'のときはフォルダがkeyに含まれない
        keys.extend([content['Prefix']
                      for content in response['CommonPrefixes']])
    if 'Contents' in response: # 該当する key がないと response に 'Content:
        keys.extend([content['Key'] for content in response['Contents']])
    if 'IsTruncated' in response:
        return __get_all_keys(bucket=bucket, prefix=prefix, keys=keys,
                               marker=response['NextMarker'])
    return keys

if __name__ == '__main__':
    keys = ls('bucket1', 'folder1/', recursive=False)
    print(keys) # ['folder1/', 'folder1/sample.txt', 'folder1/html/']

```

S3内のファイル一覧取得方法

~/`.aws/credentials` のdefaultプロファイルに、S3へのアクセス権限 (s3:ListBucket)のあるアクセスキーが入力してあれば、例えば以下のコードを実行すると以下のようなリストが返ってきます。

```
import boto3

client = boto3.client('s3')
obj = client.list_objects(Bucket='bucket1', Prefix='folder1/')
[content['Key'] for content in obj['Contents']]
# ['folder1/', 'folder1/sample.txt', 'folder1/html/', 'folder1/html/sampl
```

これはつまり、Prefixに指定したパス以下のファイルやフォルダを再帰的に取得していることになります。

再帰的ではなく、指定したPrefixの階層のみのファイルリストを取得したい場合は `Delimiter='/'` を指定します。

```
import boto3

client = boto3.client('s3')
obj = client.list_objects(Bucket='bucket1', Prefix='folder1/', Delimiter=
[content['Key'] for content in obj['Contents']]
# ['folder1/', 'folder1/sample.txt']
```

これで深い階層のファイルを取得せずに済みましたが、同じ階層にあるはずの'folder1/html/'が取得できてません。Delimiterを指定した場合、フォルダの情報は `obj['Contents']` ではなく `obj['CommonPrefixes']` に格納されるようです。(公式ドキュメント参照)

そのためフォルダも含めて指定したPrefixの階層のみファイル・フォルダを取得する場合は次のようにします

```
import boto3

client = boto3.client('s3')
obj = client.list_objects(Bucket='bucket1', Prefix='folder1/', Delimiter=
- [content['Key'] for content in obj['Contents']]
+ keys = []
```



3



4



```
+ if 'Contents' in obj.keys():
+     keys.extend([content['Key'] for content in obj['Contents']])
+ if 'CommonPrefixes' in obj.keys():
+     keys.extend([content['Prefix'] for content in obj['CommonPrefixes']]
+ keys # ['folder1/', 'folder1/sample.txt', 'folder1/html/']
```

これで同階層のフォルダを取得できました。

ファイル数が1000件を超える場合

list_objectsで一度に取得できるファイル数は1000件のみのため、1000件を超える場合はpaginationのような処理が必要になります。list_objectsを使う場合はMarkerを用いて無理やり全取得させることができます。

```
obj = client.list_objects(Bucket='bucket1', Prefix='folder1/')
```

でobjを取得したとき、もし全取得できていない場合は

obj['IsTruncated'] の値がTrueとなります。またlist_objectsのMarker引数にファイル名を記載すればそのファイルの次のファイルからリストを取得してくれます。

これを踏まえると以下の関数で全取得できるようになります。

```
import boto3

def get_all_keys(bucket, prefix, keys = None, marker = '', recursive = False):
    s3 = boto3.client('s3')
    if recursive:
        response = s3.list_objects(
            Bucket=bucket, Prefix=prefix, Marker=marker)
    else:
        response = s3.list_objects(
            Bucket=bucket, Prefix=prefix, Marker=marker, Delimiter='/')

    if keys is None:
        keys = []

    if 'CommonPrefixes' in response:
        keys.extend([content['Prefix']
                     for content in response['CommonPrefixes']])
```

```
if 'Contents' in response:
    keys.extend([content['Key'] for content in response['Contents']])
if 'IsTruncated' in response:
    return __get_all_keys(bucket=bucket, prefix=prefix, keys=keys,
    return keys

get_all_keys('bucket1', 'folder1/', recursive=False) # ['folder1/', 'foldr
```



新規登録して、もっと便利にQiitaを使ってみよう

1. あなたにマッチした記事をお届けします
2. 便利な情報をあとで効率的に読み返せます

[ログインすると使える機能について](#)

新規登録

ログイン



コメント

この記事にコメントはありません。

あなたもコメントしてみませんか :)

新規登録

すでにアカウントを持っている方は[ログイン](#)

Qiita

How developers code is here.

© 2011-2023 Qiita Inc.

ガイドとヘルプ

About

利用規約

プライバシーポリシー

ガイドライン

デザインガイドライン

ご意見

ヘルプ

広告掲載

コンテンツ

リリースノート

公式イベント

公式コラム

募集

アドベントカレンダー

Qiita 表彰プログラム

API

SNS

✕ Qiita（キータ）公式

✕ Qiita マイルストーン

✕ Qiita 人気の投稿

📘 Qiita（キータ）公式

Qiita 関連サービス

Qiita Team

Qiita Jobs

Qiita Zine

Qiita 公式ショップ

運営

[運営会社](#)

[採用情報](#)

[Qiita Blog](#)