Disposable Email Domains Data FeedをAWS S3にインポートするには

2023年10月3日

本文書では、AWS Lambdaを活用して、WhoisXML APIが提供するDisposable Email Domain Data FeedをAWS S3バケットにダウンロードする基本的な方法を紹介します。AWS Lambdaは、サーバーをプロビジョニングしたり管理したりすることなくコードを書いて実行できる、サーバーレスのコンピューティングサービスです。AWS S3は、ファイルを保存および取得するためのオブジェクトストレージサービスです。ここでは、AWS LambdaとAWS S3バケットの両方を構成するプロセスを説明します。

以下は本文書の対象外です：

- Lambda関数のスケジューリング
- ETLパイプライン
- PythonのRequestsモジュールのインポート

前提条件

事前に以下を用意する必要があります：

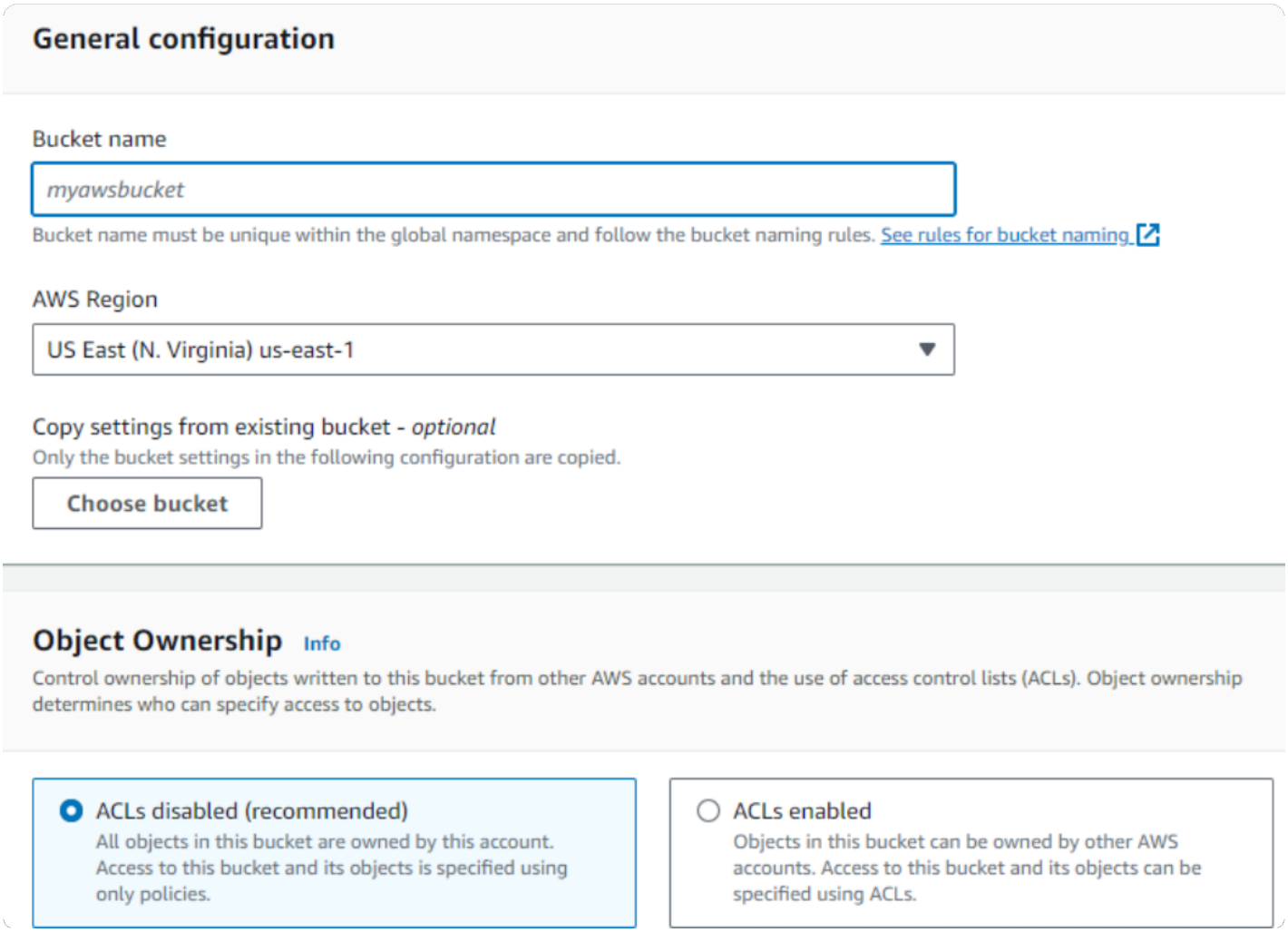
- AWSアカウント
- AWSサービス、特にAWS LambdaとS3に関する基礎～中級程度の知識

- Lambda関数で使われるPythonの知識
- WHOIS API Disposable Email Domainsデータフィードへのアクセス。APIキーが必要です。詳細につきましては、sales@whoisxmlapi.com にお問い合わせください。このデータフィードの仕様は、こちらでご確認いただけます。

ステップ1：AWS S3バケットの作成

最初のステップは、Disposable Email Domainsファイルを書き込むS3バケットの作成です。

- AWS Management Consoleで、S3サービスに移動します。
- 「Create Bucket」をクリックします。
- バケットにユニークな名前をつけ、適切な地域を選択します。

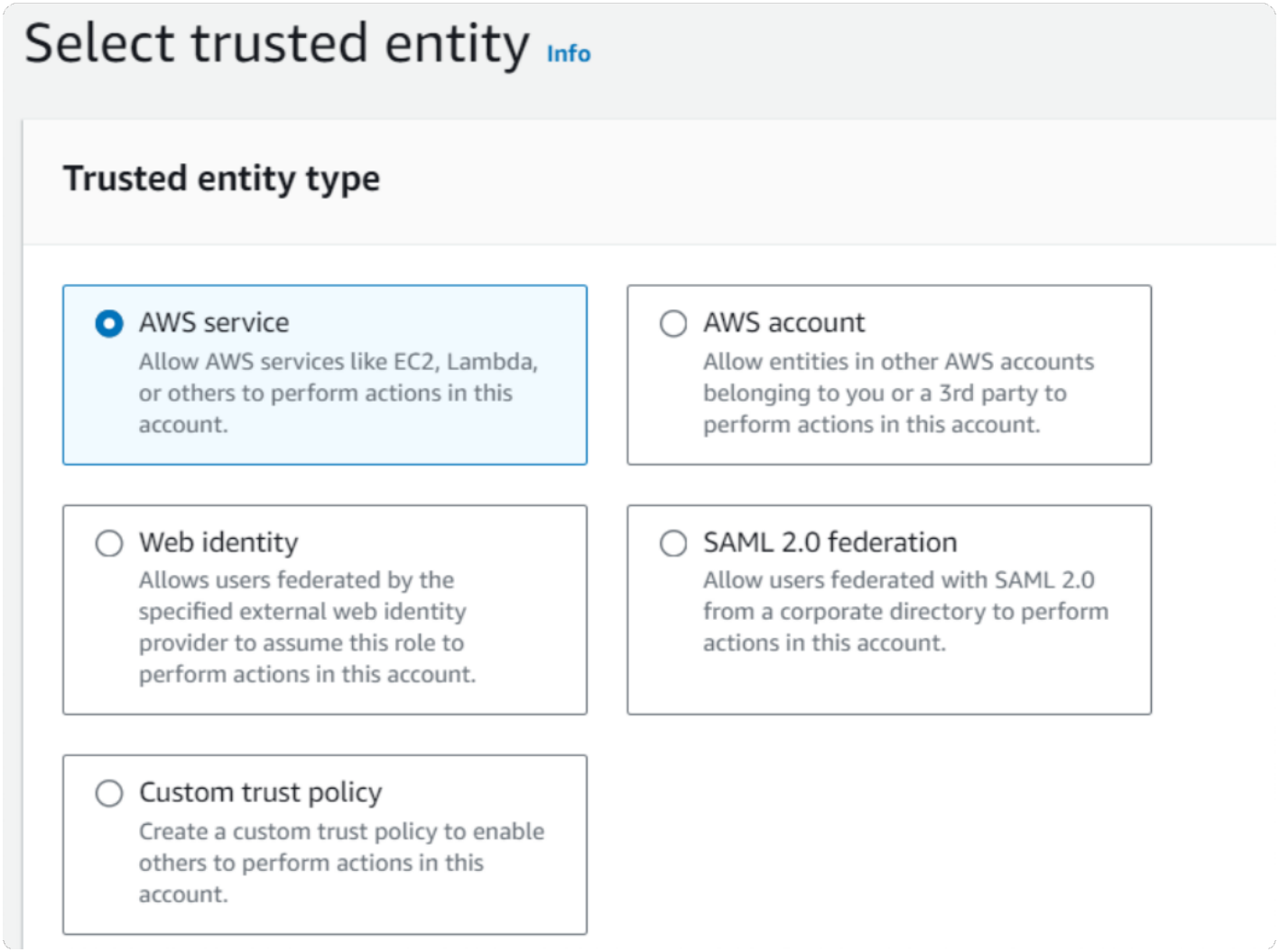


- ここでは、デフォルト設定のまま「Create Bucket」をクリックします。

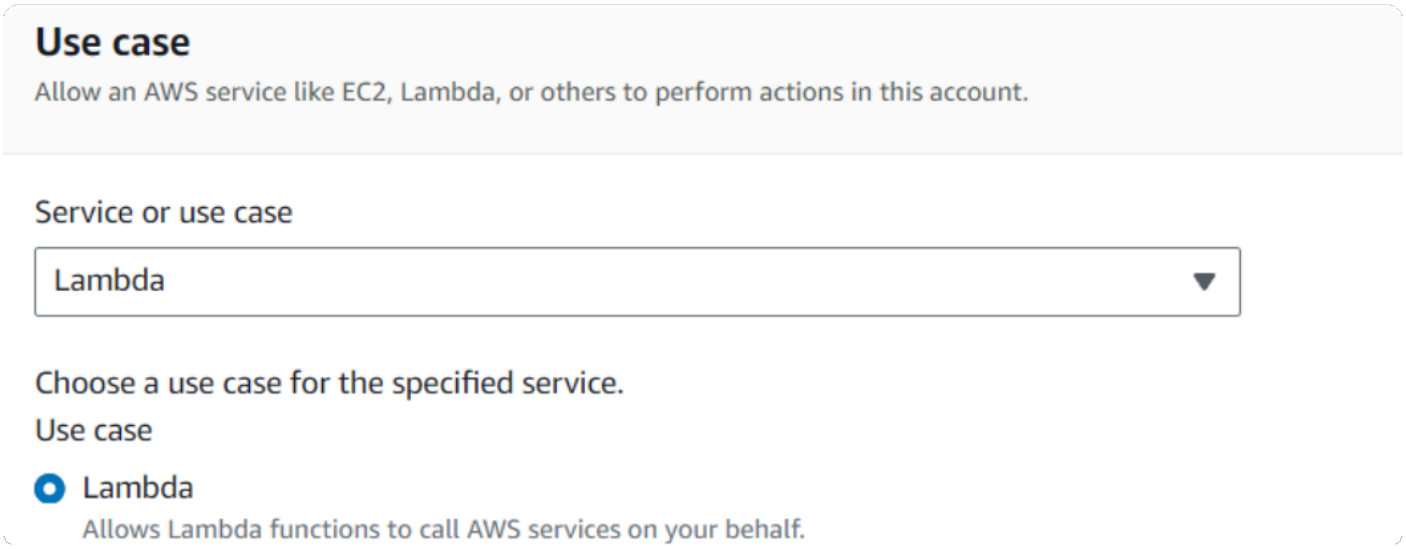
ステップ2： IAMロールの作成

AWS Lambdaでは、S3バケットの読み書きに必要な権限を持つIAMロールが必須となります。以下の手順でIAMロールを作成してください：

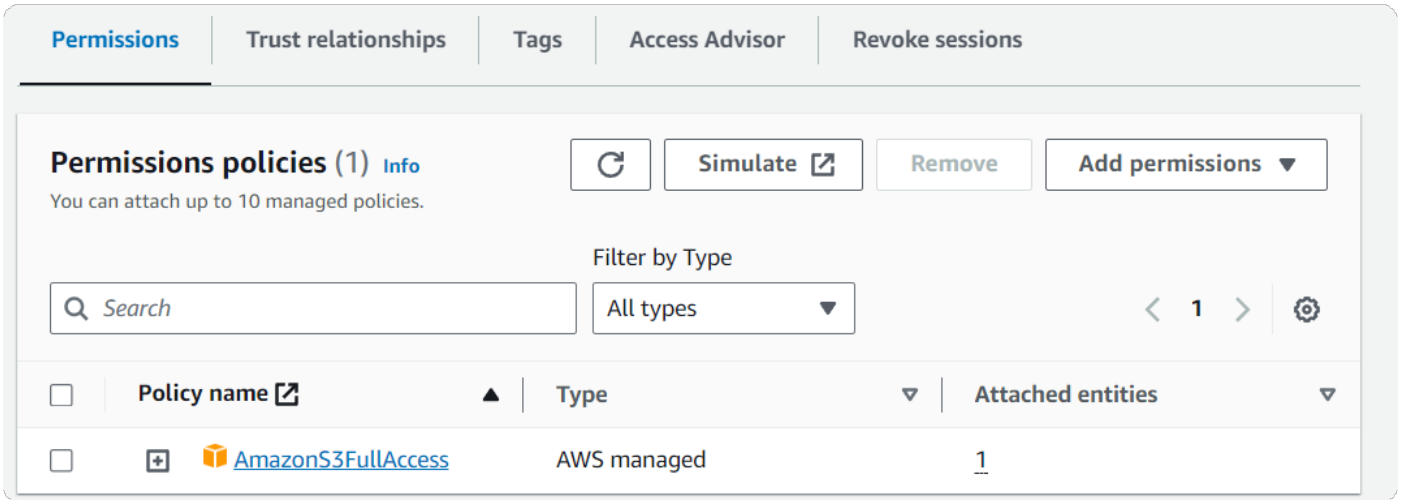
- AWSマネジメントコンソールでIAMサービスに移動します。



- 「Roles」をクリックし、次に「Create Role」をクリックします。
- このロールのサービスとして「Lambda」を選択し、「Next: Permissions」をクリックします。



- 検索バーに「S3」と入力し、「AWSS3FullAccess」、そして「Next: Tags」を選択します。



- タグは任意です。次に「Next: Review」をクリックします。
- あなたのロール、名前、簡単な説明を入力し、「Create Role」をクリックします。

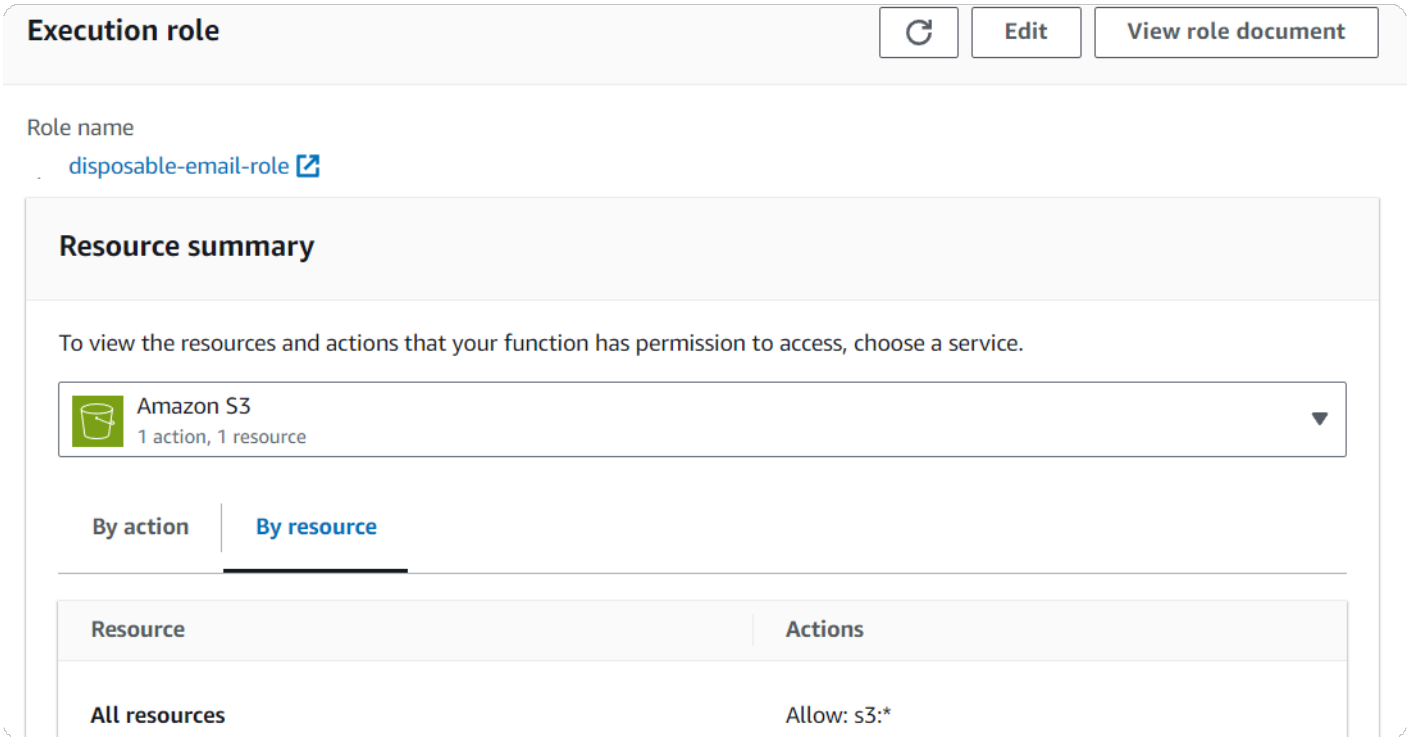
ステップ3：Lambda関数の作成

Lambda関数の作成は楽しく、簡単です。その方法は以下の通りです：

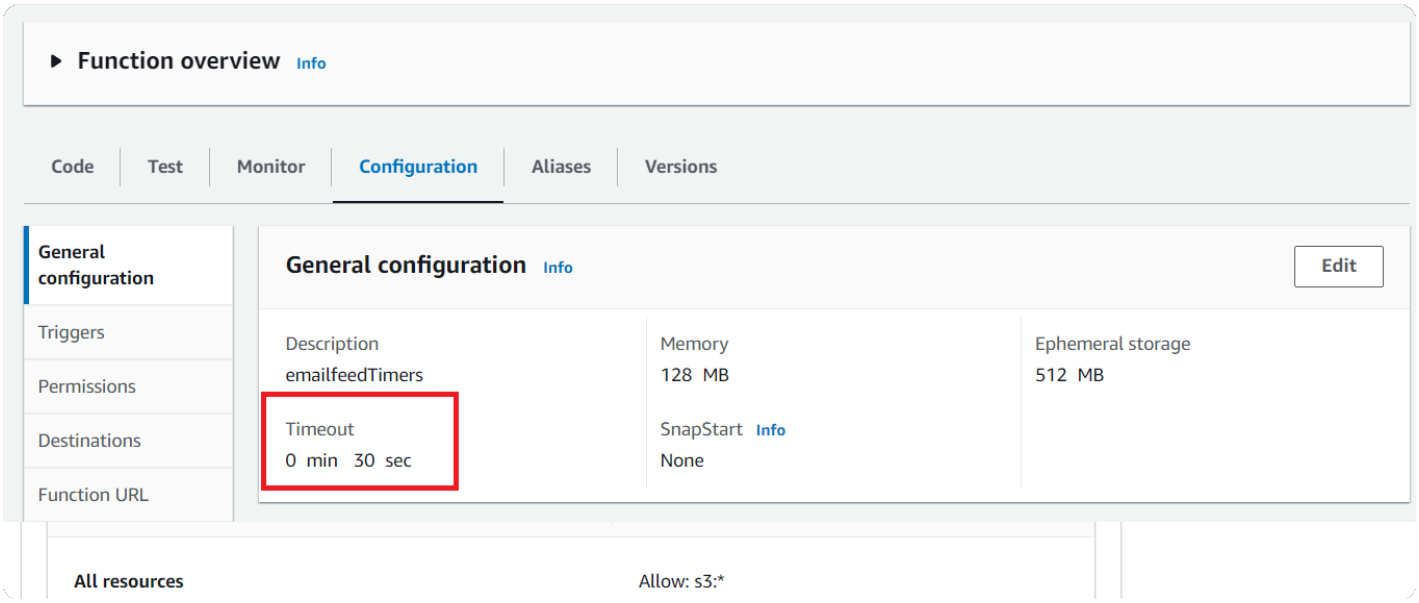
- AWSマネジメントコンソールでLambdaサービスに移動します。
- 「Create Function」をクリックします。
- 関数にわかりやすい名前を付け、ランタイムとしてPythonを選択します。そして、上記のステップ2で作成したIAMロールを選択します。
- 「Create function」をクリックします。

注：

Execution roleの設定：



Lambda関数のタイムアウト値を設定します。今回は30秒に設定しています。



ステップ4：Disposable Email DomainリストをS3にインポートするLambda関数を記述

この例ではpython requestsモジュールを使用していますが、Boto3の一部ではなくなったため、インポートする必要があるかもしれません。この方法に関するAWSのドキュメントは曖昧ですが、インターネット上で様々な技術記事を見つけることができます。

コードの例：

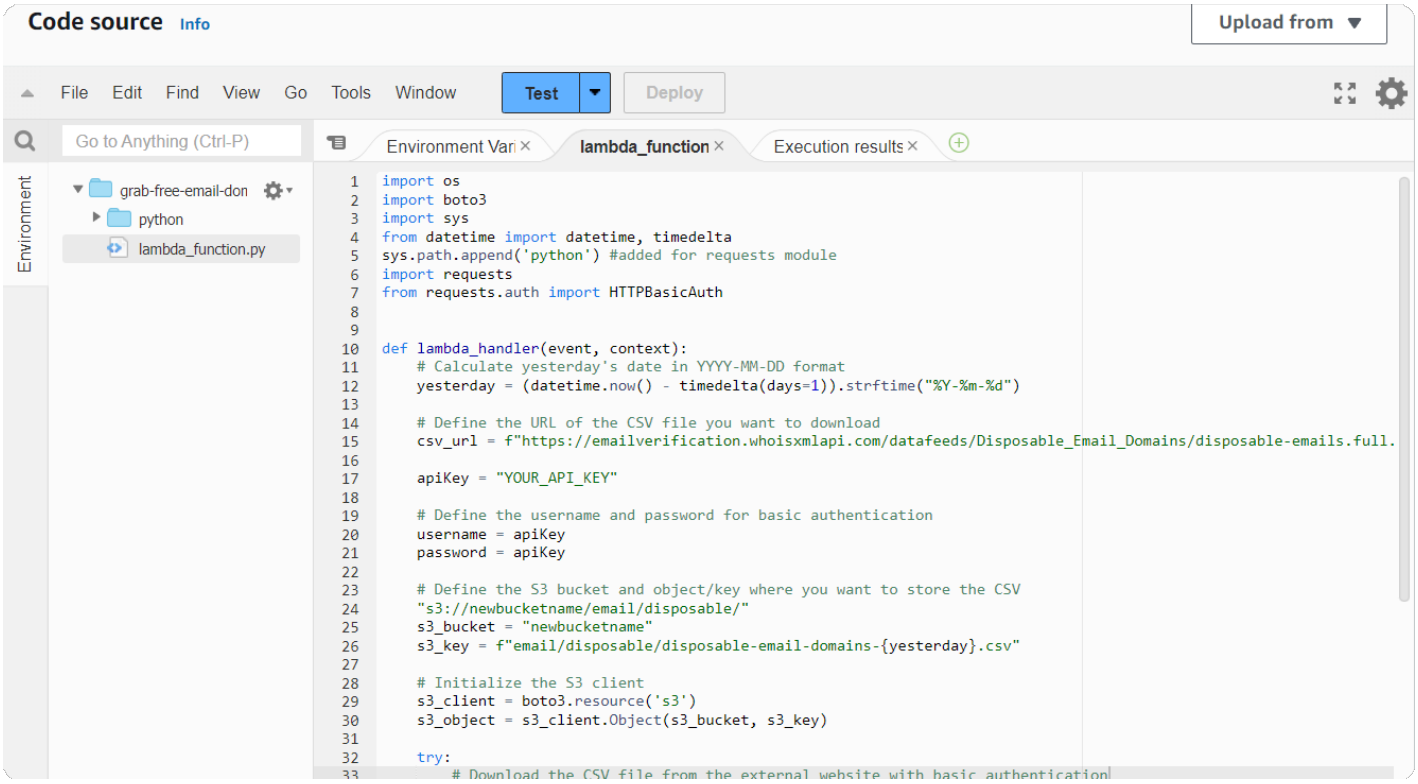
以下のPythonコード（GitHubでも入手できます）は、lambda_handlerのエントリーポイントです：

```
import os
import boto3
import sys
from datetime import datetime, timedelta
sys.path.append('python') #added for requests module
```

```
import requests
from requests.auth import HTTPBasicAuth


def lambda_handler(event, context):
    # Calculate yesterday's date in YYYY-MM-DD format
    yesterday = (datetime.now() - timedelta(days=1)).strftime("%Y-%m-%d")

    # Define the URL of the CSV file you want to download
    csv_url = f"https://emailverification.whoisxmlapi.com/datafeeds/Disposa

    apiKey = "YOUR_API_KEY"

    # Define the username and password for basic authentication
    username = apiKey
    password = apiKey

    # Define the S3 bucket and object/key where you want to store the CSV
    "s3://newbucketname/email/disposable/"
    s3_bucket = "newbucketname"
    s3_key = f"email/disposable/disposable-email-domains-{yesterday}.csv"

    # Initialize the S3 client
    s3_client = boto3.resource('s3')
    s3_object = s3_client.Object(s3_bucket, s3_key)

    try:
        # Download the CSV file from the external website with basic authen
        response = requests.get(csv_url, auth=HTTPBasicAuth(username, passw

        if response.status_code == 200:
            # Upload the CSV file to S3
            print(f"Uploading file to ", s3_bucket, s3_key)
            s3_object.put(Body=response.content)
            return {
                'statusCode': 200,
                'body': 'CSV file successfully downloaded and uploaded to S
            }
        else:
            bodyStr = f"Failed to download {csv_url}"
```

```
            return {
                'statusCode': response.status_code,
                'body': bodyStr
            }
    except Exception as e:
        return {
            'statusCode': 500,
            'body': str(e)
        }
```

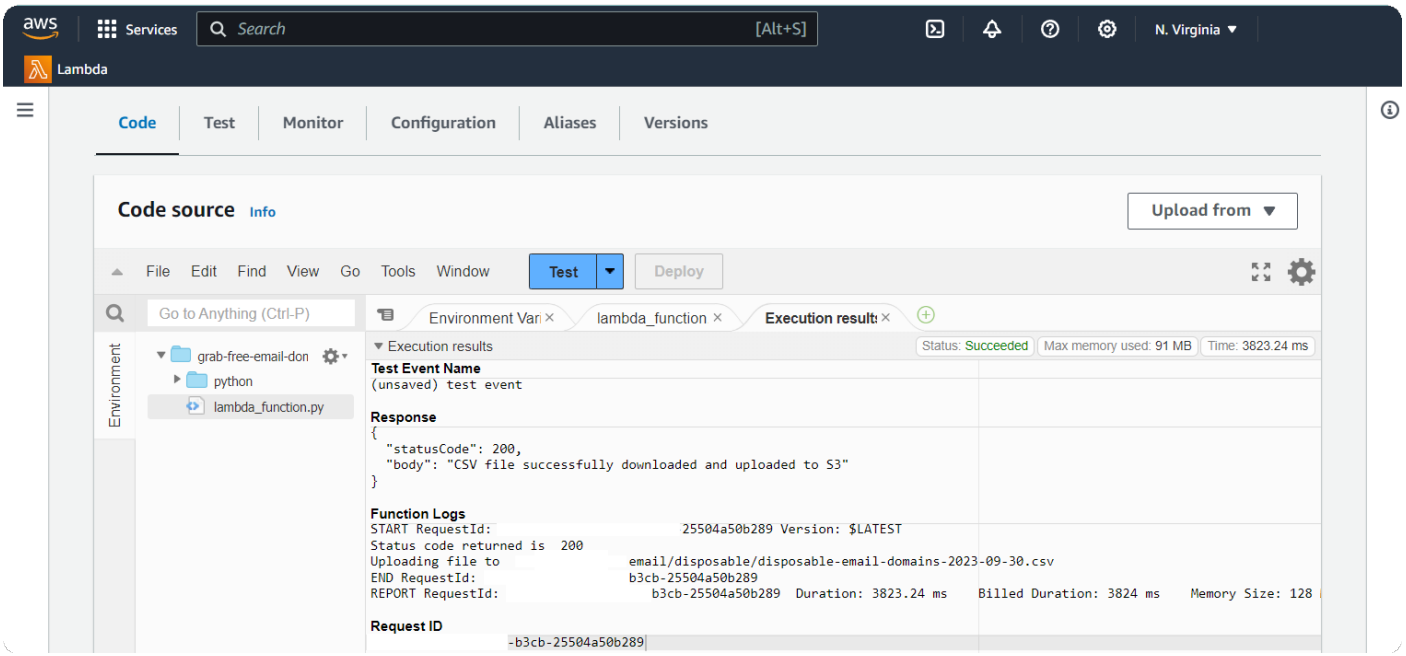
完成すると、このようなものができるはずです：



ステップ5：作成したLambda関数をテストする

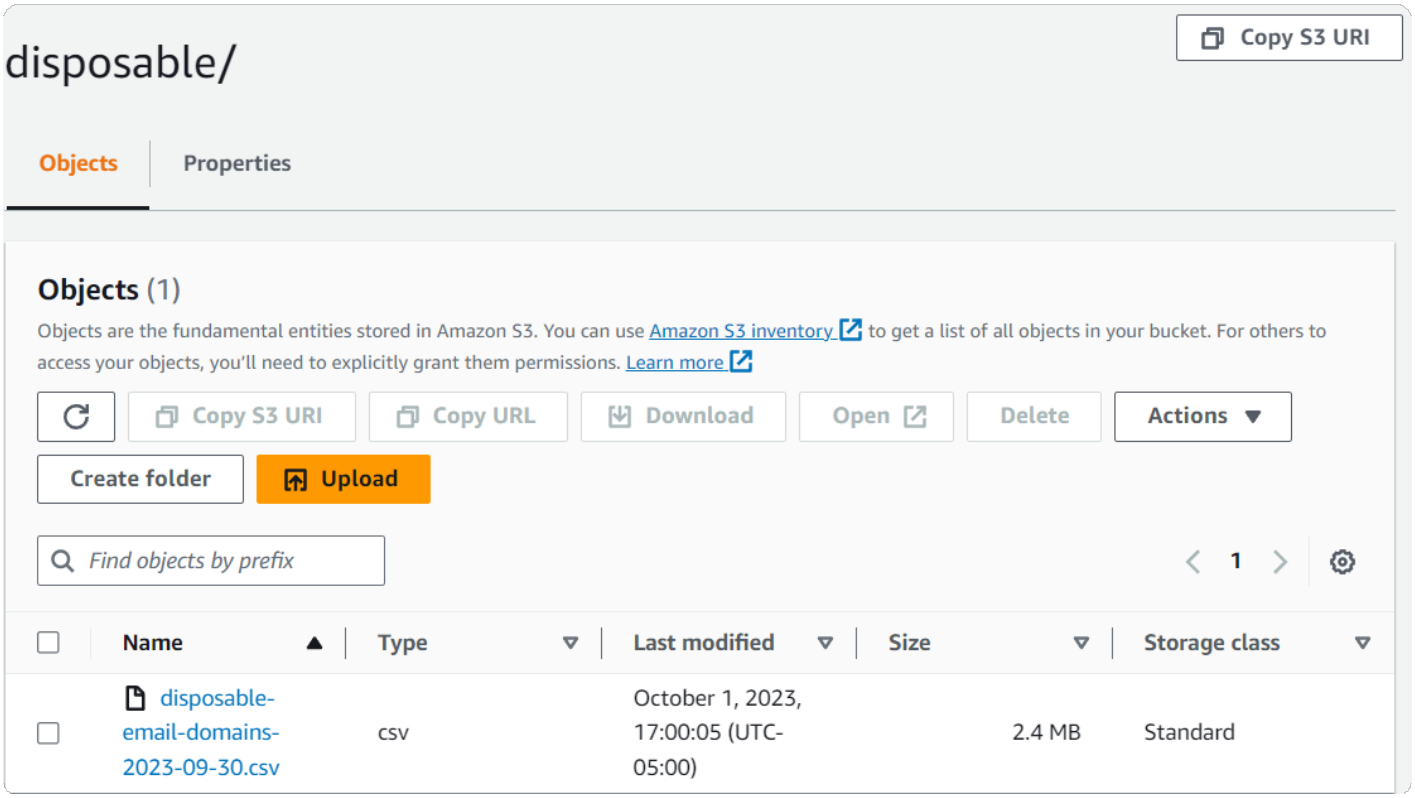
最後のステップとして、作成したLambda関数をテストし、a) Disposable Email Domainファイルを正常に取得できること、b) S3バケットに書き込めることを確認します：

- ページ上部の「Test」をクリックすると、以下のようなものが表示されるはずです。



- 「”requests" module not found 」というメッセージを受け取った場合は、python requests ライブラリを正しく設定する必要があります（本文書の範囲外）。

Lambda関数が正しく設定されていれば、関数はファイルを取得し、S3バケットに書き込みます。S3バケットに移動してファイルの存在を確認できます。



まとめ

S3バケットにアクセスできるAWS Lambdaの設定は、クラウドエンジニアにとってはごく一般的なタスクです。このプロセスの後に踏む次のステップは、Athena、Postgres、MySQLデータベースへのインポートなど、このデータで何をするかを決めることです。ETL用のAWS Glueをご存じない方は、そちらもチェックしてみてください。