AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈構築①〉）

2021.11.12　2021.10.29

【前回】AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈概要〉）
【次回】AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈構築②〉）
【簡易版】AmazonConnectによる自動電話通知（まとめ）

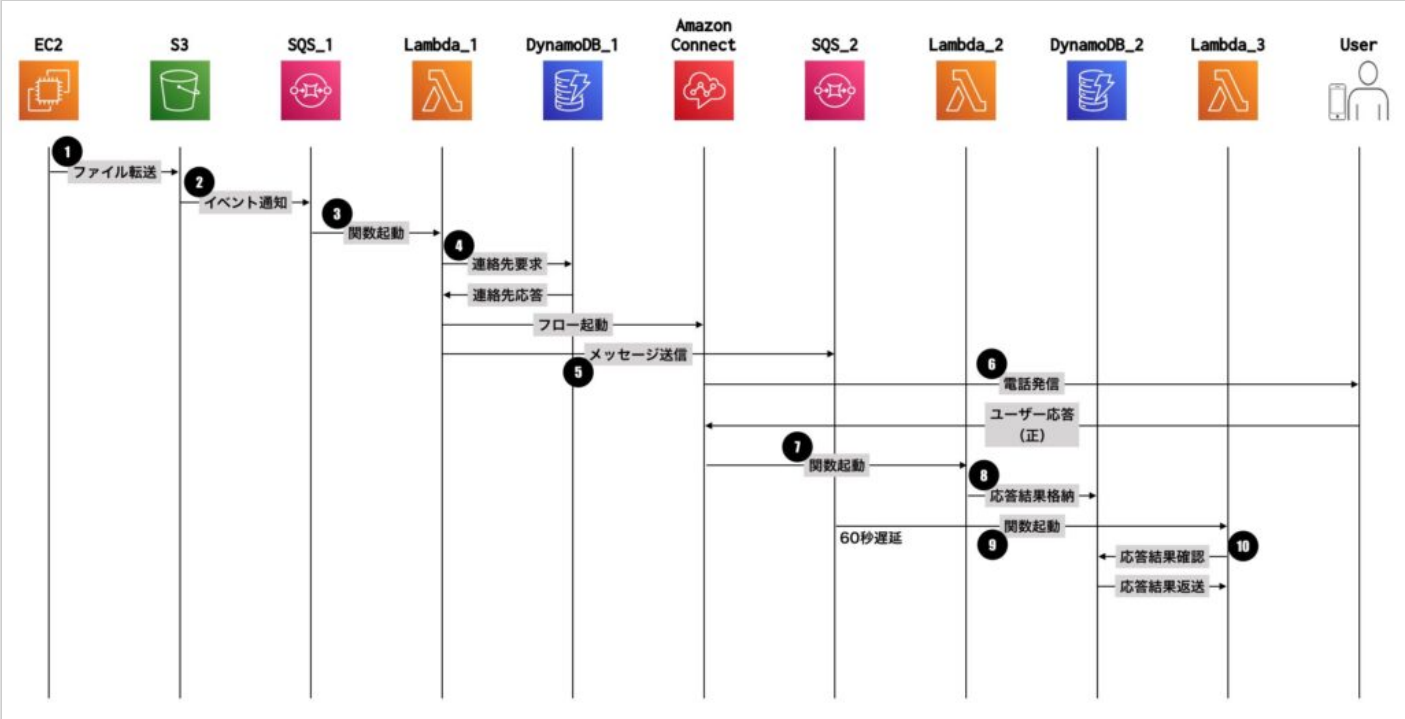
監視サーバーで障害を検知した際に、自動で電話通知できるようにしていきます。ネットワークエンジニアも利用することの多い監視サーバー(Zabbix)で障害検知し、AWS上のAmazonConnectを利用し自動電話を発信します。

今回は下記の条件を満たせるようにAWSの各サービスを利用して自動電話通知の仕組みを導入します。

- 複数の通知先を登録した連絡先リストを持たせる。
- 連絡先リストに優先度(通知順)を設定する。
- 優先度が高い人に最初に電話する。
- 応答が無かった場合、次の優先度の人に順番に電話する。
- 連絡先リストの最後まで電話しても応答が無かった場合、最初に戻って継続する。

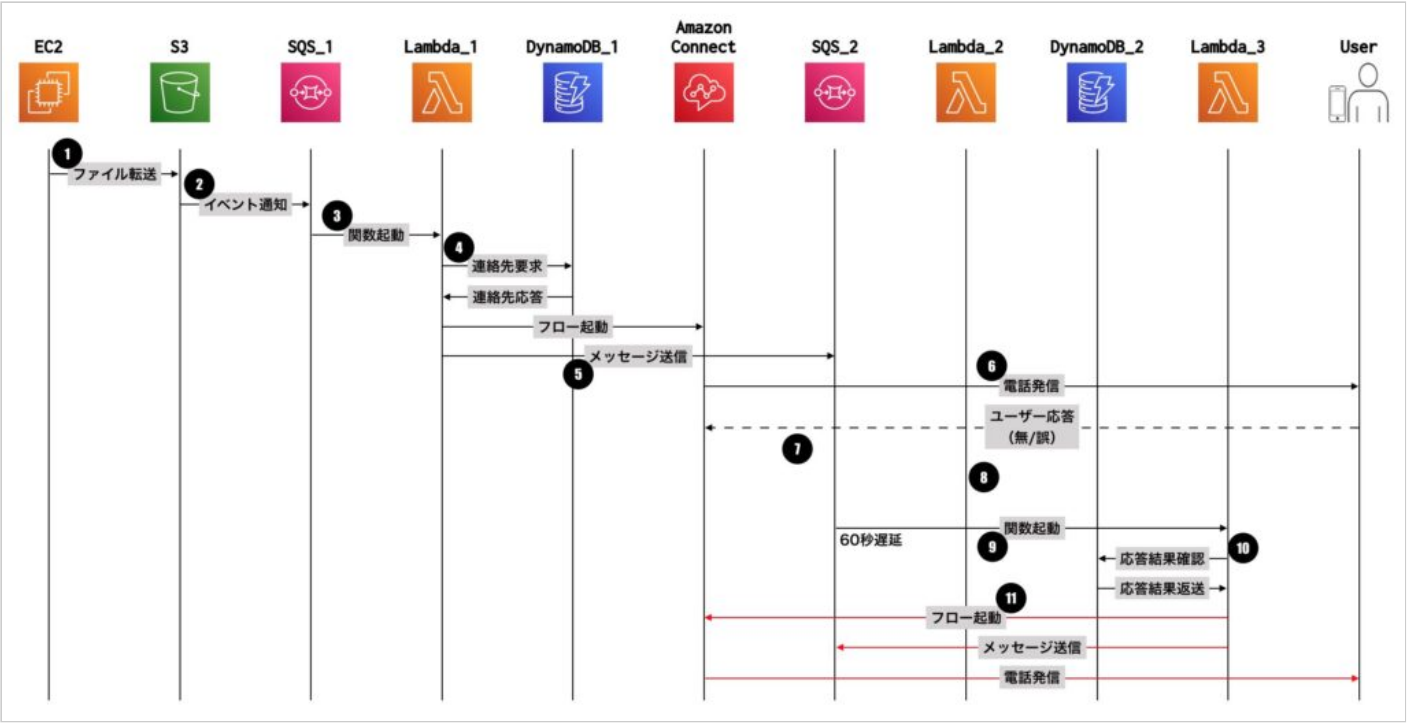
自動電話通知フロー

電話に応答した場合のフロー



1. EC2上の監視サーバーで障害を検知し、S3へトリガーファイルを格納
2. S3のイベント通知機能で、SQS_1にメッセージを送信
3. SQS_1をトリガーとして、Lambda_1を起動
4. Lambda_1がDynamoDB_1から連絡先を取得し、AmazonConnectを起動
5. Lambda_1がAmazonConnectを起動すると同時に、SQS_2へメッセージを送信
6. AmazonConnectがユーザーへ自動電話通知を実施
7. ユーザが正常応答し、AmazonConnectがLambda_2を起動
8. Lambda_2が応答結果をDynamoDB_2に保存(応答OK)
9. 60秒後にSQS_2をトリガーとしてLambda_3を起動
10. Lambda_3がDynamoDB_2の応答結果を確認(正常応答しているため、何もせずに処理完了)

電話に応答しなかった場合のフロー



1. EC2上の監視サーバーで障害を検知し、S3へトリガーファイルを格納
2. S3のイベント通知機能で、SQS_1にメッセージを送信
3. SQS_1をトリガーとして、Lambda_1を起動
4. Lambda_1がDynamoDB_1から連絡先を取得し、AmazonConnectを起動
5. Lambda_1がAmazonConnectを起動すると同時に、SQS_2へメッセージを送信
6. AmazonConnectがユーザーへ自動電話通知を実施
7. ユーザが正常応答せず、AmazonConnectがLambda_2を起動
8. Lambda_2が応答結果をDynamoDB_2に保存(応答NG)
9. 60秒後にSQS_2をトリガーとしてLambda_3を起動
10. Lambda_3がDynamoDB_2の応答結果を確認
11. 正常応答していないため、再度AmazonConnectを起動(以降、5から繰り返し)

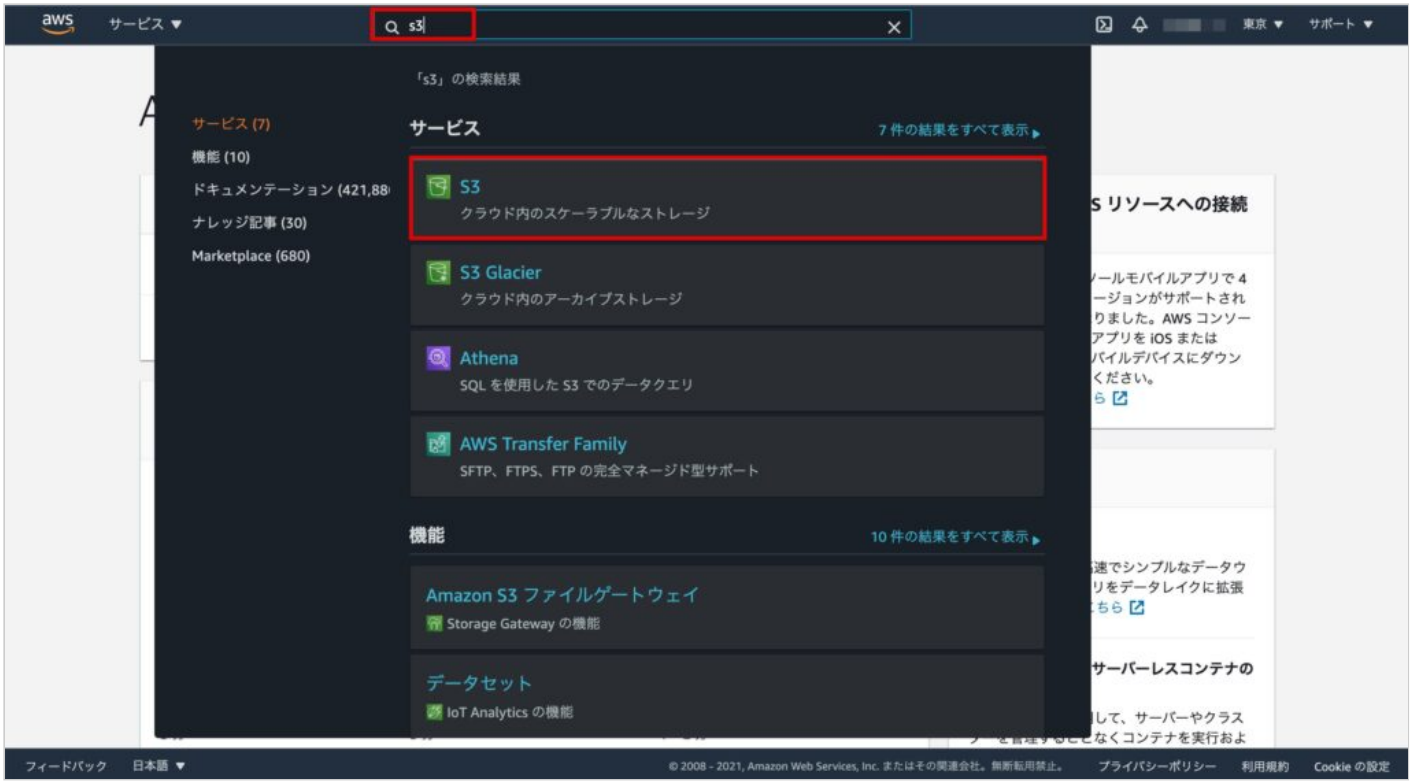
EC2上で監視サーバーを構築

EC2上でのZabbixの構築はこちらを参照してください。

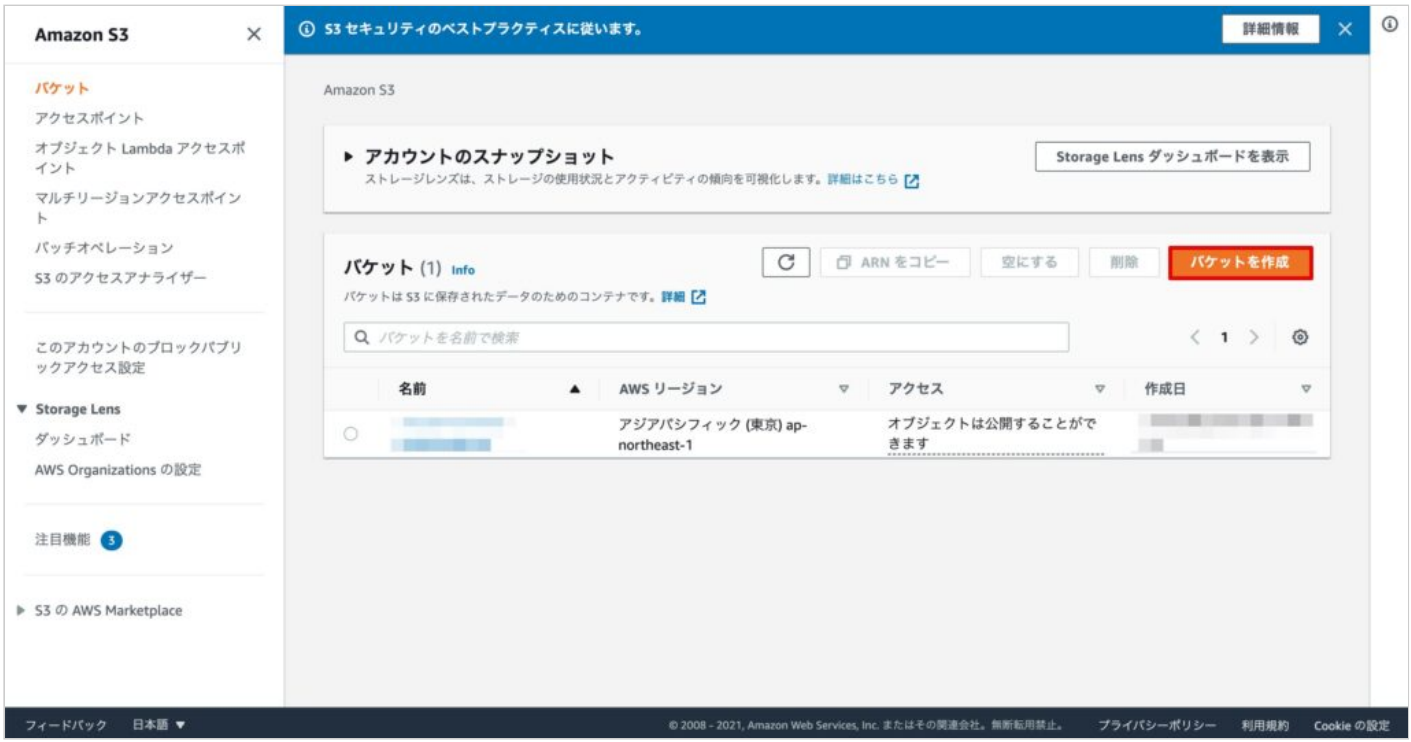
AWS上での監視サーバー(Zabbix)構築

S3のバケット作成

AWSマネジメントコンソール上で、「s3」を検索します。



「バケットの作成」をクリックします。



バケット名を入力し、それ以外はデフォルトのままで、「バケットを作成」をクリックします。

一般的な設定

バケット名

amazonconnect-alert-notification-bucket

バケット名は一意である必要があり、スペース、または大文字を含めることはできません。**バケットの命名規則をご参照ください**

AWS リージョン

アジアパシフィック (東京) ap-northeast-1

既存のバケットから設定をコピー - *オプション*

次の設定のバケット設定のみがコピーされます。

バケットを選択する

このバケットのブロックパブリックアクセス設定

パブリックアクセスは、アクセスコントロールリスト (ACL、Access Control List)、バケットポリシー、アクセスポイントポリシー、またはそのすべてを介してバケットとオブジェクトに許可されます。このバケットとそのオブジェクトへの公開アクセスが確実にブロックされるようにするには、[パブリックアクセスをすべてブロック] を有効にします。これらの設定はこのバケットとそのアクセスポイントにのみ適用されます。AWS では [パブリックアクセスをすべてブロック] を有効にすることをお勧めしますが、これらの設定を適用する前に、アプリケーションが公開アクセスなしで正しく機能することをご確認ください。このバケットやオブジェクトへのある程度の公開アクセスが必要な場合は、各ストレージユースケースに合わせて以下にある個々の設定をカスタマイズできます。**詳細**

☑ **パブリックアクセスをすべてブロック**

この設定をオンにすることは、以下の 4 つの設定をすべてオンにすることと同じです。次の各設定は互いに独立しています。

- ☑ *新しいアクセスコントロールリスト (ACL) を介して付与されたバケットとオブジェクトへのパブリックアクセスをブロックする*
  S3 は、新しく追加されたバケットまたはオブジェクトに適用されたパブリックアクセス許可をブロックし、既存のバケットおよびオブジェクトに対する新しいパブリックアクセス ACL が作成されないようにします。この設定では、ACL を使用して S3 リソースへのパブリックアクセスを許可する既存のアクセス許可は変更されません。
- ☑ *任意のアクセスコントロールリスト (ACL) を介して付与されたバケットとオブジェクトへのパブリックアクセスをブロックする*
  S3 はバケットとオブジェクトへのパブリックアクセスを付与するすべての ACL を無視します。
- ☑ *新しいパブリックバケットポリシーまたはアクセスポイントポリシーを介して付与されたバケットとオブジェクトへのパブリックアクセスをブロックする*
  S3 は、バケットとオブジェクトへのパブリックアクセスを許可する新しいバケットポリシーおよびアクセスポイントポリシーをブロックします。この設定は、S3 リソースへのパブリックアクセスを許可する既存のポリシーを変更しません。
- ☑ *任意のパブリックバケットポリシーまたはアクセスポイントポリシーを介したバケットとオブジェクトへのパブリックアクセスとクロスアカウントアクセスをブロックする*
  S3 は、バケットとオブジェクトへのパブリックアクセスを付与するポリシーを使用したバケットまたはアクセスポイントへのパブリックアクセスとクロスアカウントアクセスを無視します。

バケットのバージョニング

バージョニングは、オブジェクトの複数のバリアントを同じバケット内に保持する手段です。バージョニングを使用すると、Amazon S3 バケットに格納されているすべてのオブジェクトのすべてのバージョンを保存、取得、復元できます。バージョニングを使用すると、意図しないユーザーアクションと意図しないアプリケーション障害の両方から簡単に復旧できます。**詳細**

バケットのバージョニング

- ◉ 無効にする
- ○ 有効にする

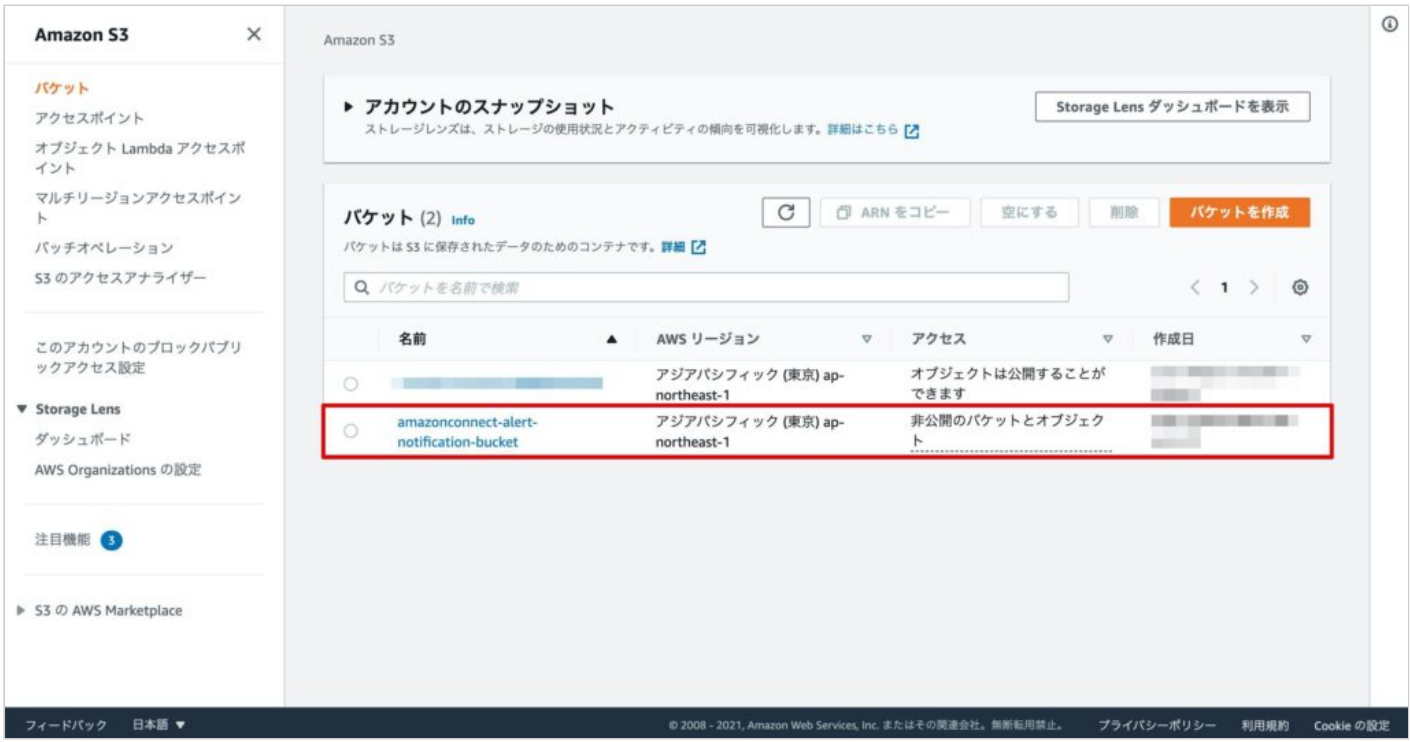
タグ (0) - *オプション*

バケットにタグ付けすることで、ストレージコストやその他の基準を追跡します。**詳細**

このバケットに関連付けられたタグはありません。

タグの追加

バケットが作成されたことを確認します。

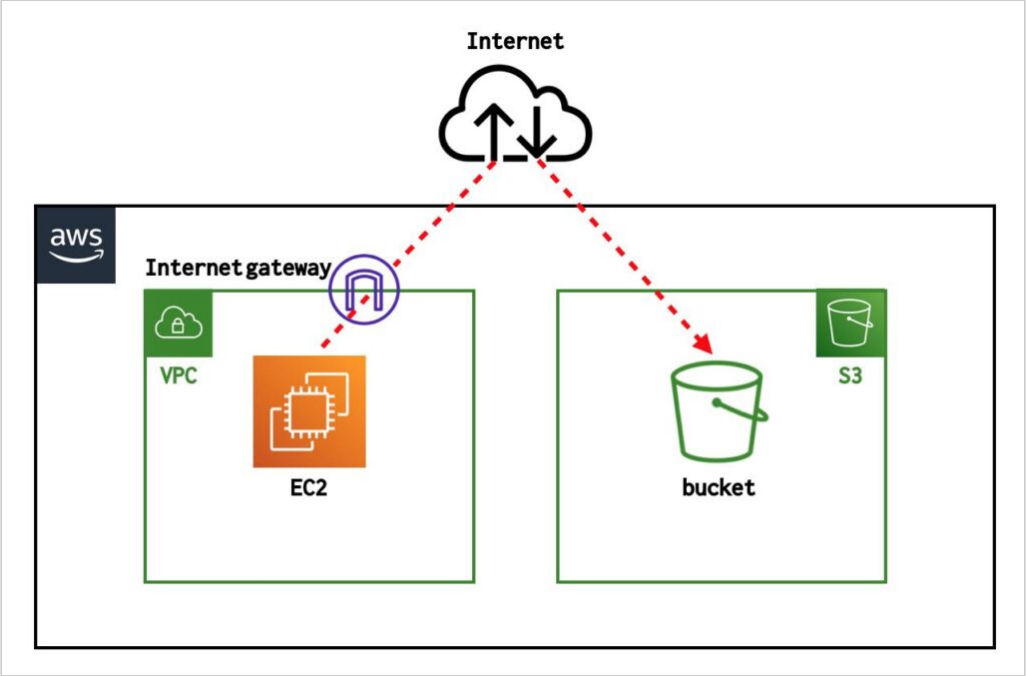


S3エンドポイント作成

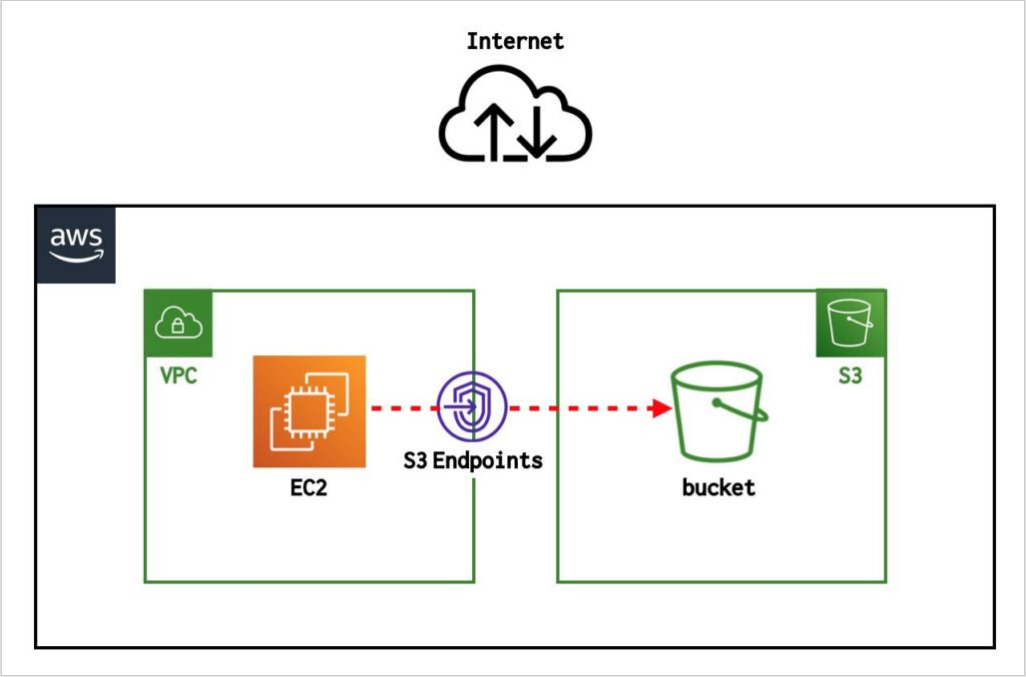
エンドポイントとは

EC2からS3へトリガーファイルをアップロードする際、デフォルトではインターネットを介した通信となります。S3エンドポイントを作成し、VPCのルートテーブルに追加することでAWS内部での通信が可能となり、セキュリティが向上します。

デフォルトの通信経路（EC2→S3）

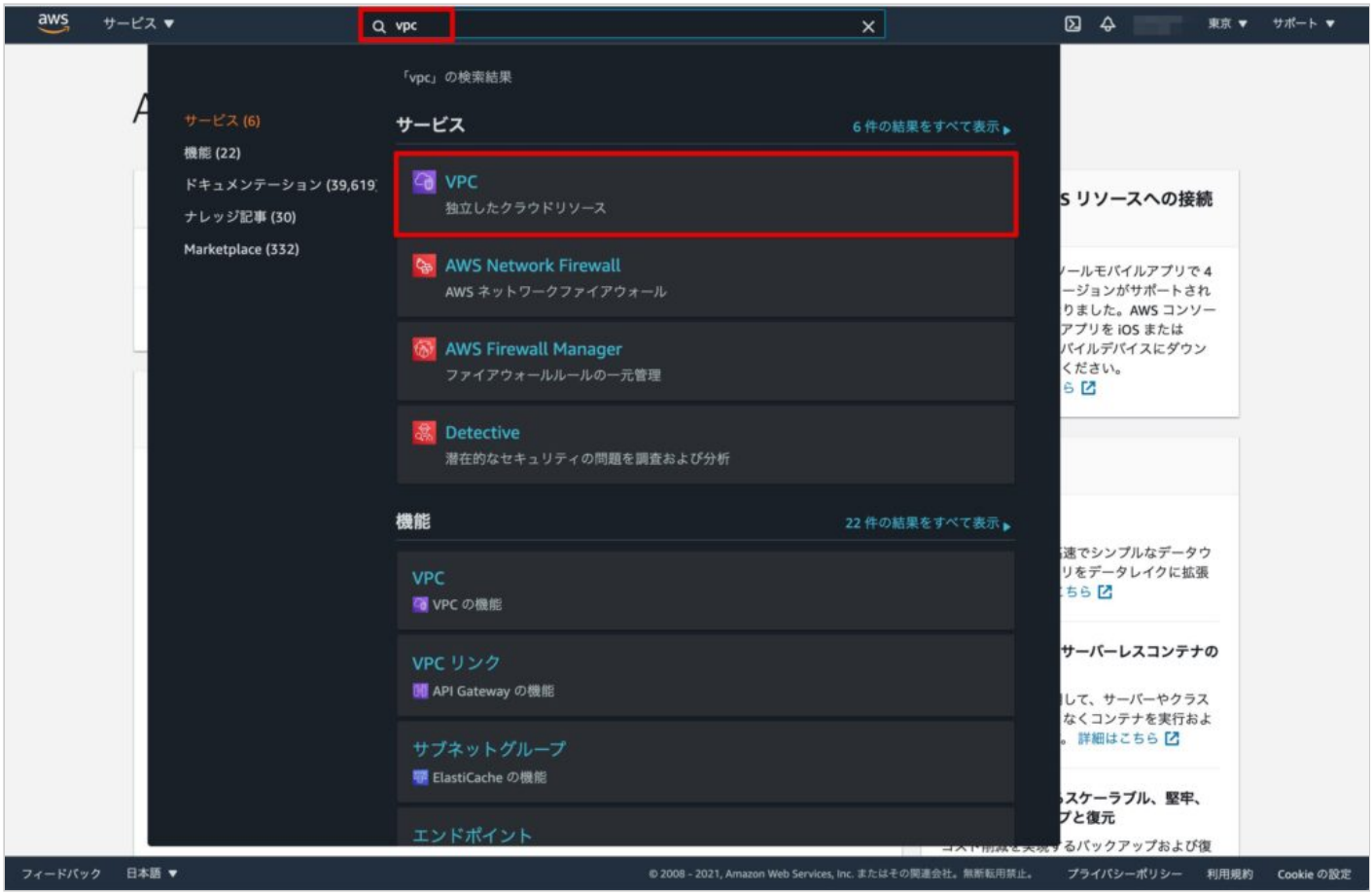


エンドポイントを利用した通信経路（EC2→S3）



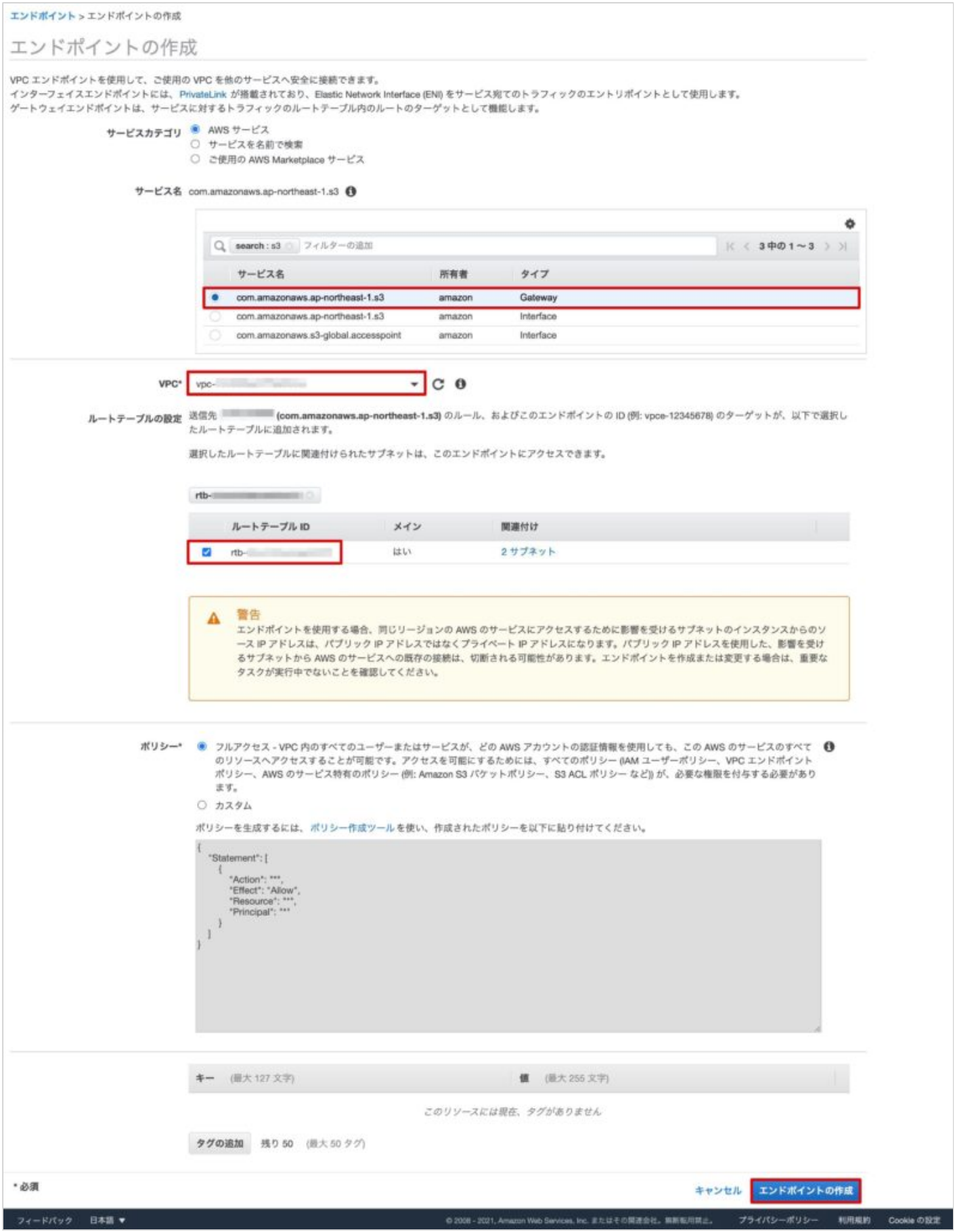
S3エンドポイントの作成

AWSマネジメントコンソール上で、「vpc」を検索します。

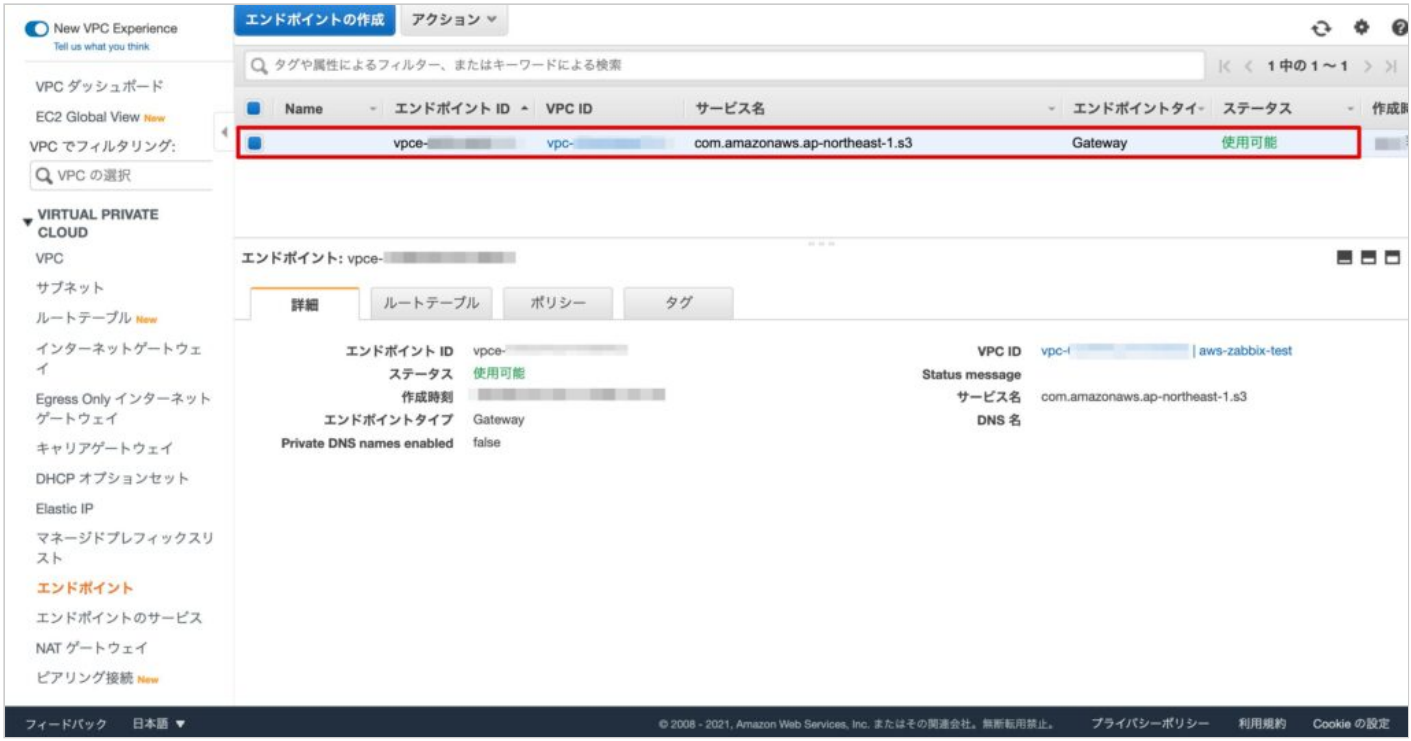


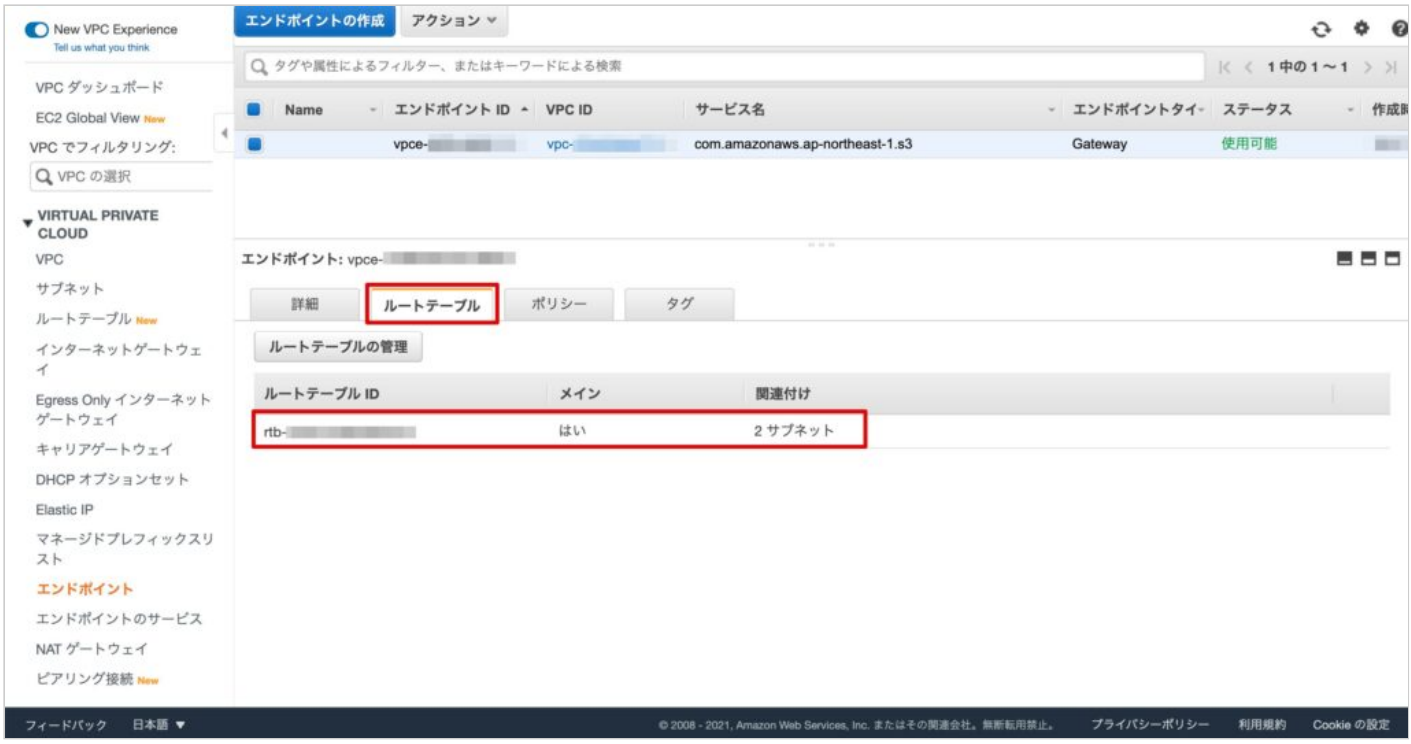
下記の通り選択・入力し、「エンドポイントの作成」をクリックします。

サービスカテゴリ：「AWSサービス」を選択
サービス名：s3で検索し、「com.amazonaws.ap-northeast-1.s3」を選択
VPC：EC2を設置しているVPCを選択
ルートテーブルの設定：チェック
ポリシー：「フルアクセス」を選択



S3のエンドポイントが作成され、ルートテーブルに紐づいていることを確認します。





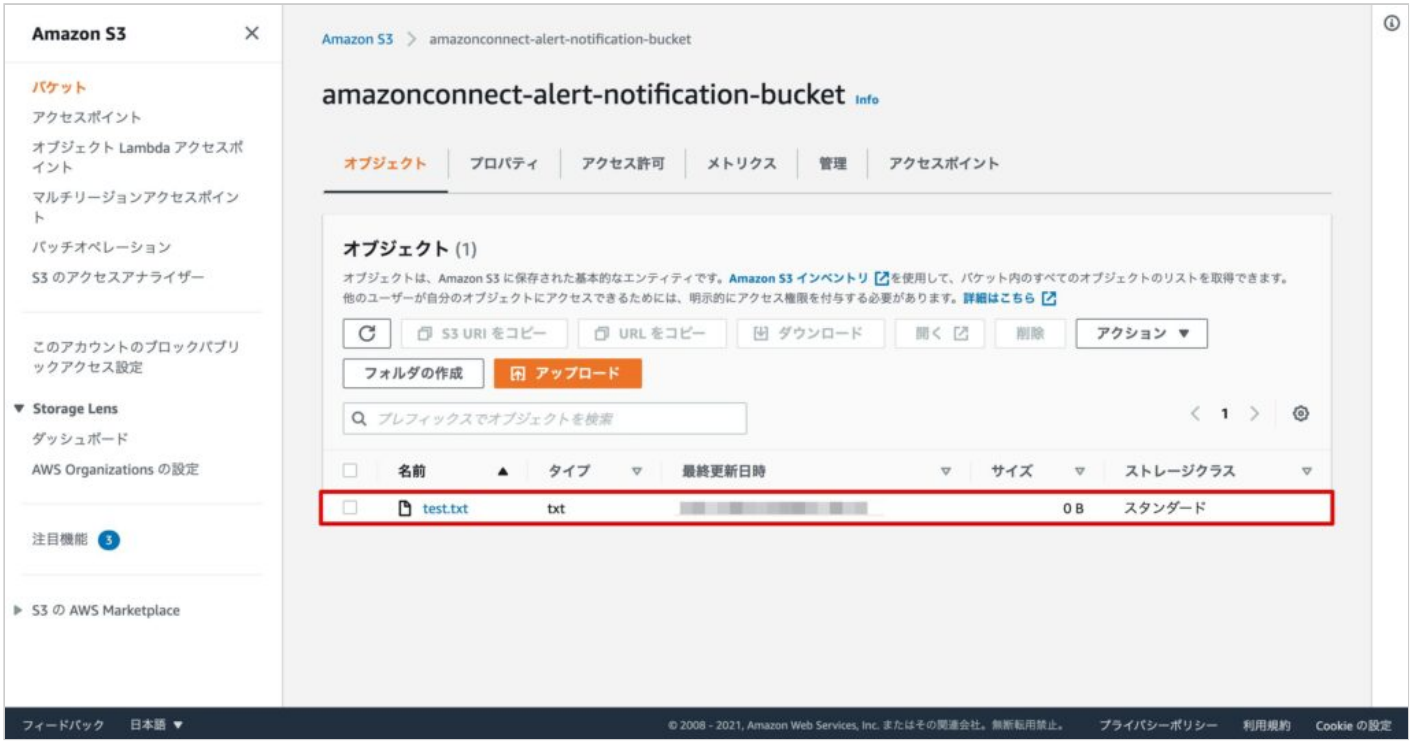
EC2からS3バケットへのファイル格納確認

EC2から下記のコマンドを実施し、S3へファイルをアップロードします。※黄色アンダーライン箇所は、作成したS3バケット名を指定してください。

```
touch /tmp/test.txt
aws s3 cp /tmp/test.txt s3://amazonconnect-alert-notification-bucket
```

```
[ec2-user@ip-10-0-0-100 ~]$ touch /tmp/test.txt
[ec2-user@ip-10-0-0-100 ~]$
[ec2-user@ip-10-0-0-100 ~]$ aws s3 cp /tmp/test.txt s3://amazonconnect-alert-notification-bucket
upload: ../../tmp/test.txt to s3://amazonconnect-alert-notification-bucket/test.txt
[ec2-user@ip-10-0-0-100 ~]$
```

S3側でファイルがアップロードされたことを確認します。



以上で、AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈構築①〉）の説明は完了です。