手順とAPIの例

StorageGRID solutions and resources

NetApp
November 21, 2025

目次

手順とAPIの例

StorageGRID でS3暗号化オプションをテストして実証

アロンクライン著

StorageGRID とS3 APIには、保存データを暗号化するためのさまざまな方法が用意されています。詳細については、を参照してください "StorageGRID の暗号化方式を確認します"。

このガイドでは、S3 APIの暗号化メソッドについて説明します。

サーバー側の暗号化（SSE）

SSEを使用すると、クライアントがオブジェクトを格納し、StorageGRID で管理される一意のキーで暗号化できます。オブジェクトが要求されると、StorageGRID に格納されたキーによってオブジェクトが復号化されます。

SSEの例

- SSEを持つオブジェクトを配置します

```
aws s3api put-object --bucket <bucket>  --key <file> --body "<file>"
--server-side-encryption AES256 --endpoint-url https://s3.example.com
```

- オブジェクトのヘッダーで暗号化を確認します

```
aws s3api head-object --bucket <bucket>  --key <file>   --endpoint-url
https://s3.example.com
```

```
{
    "AcceptRanges": "bytes",
    "LastModified": "2022-05-02T19:03:03+00:00",
    "ContentLength": 47,
    "ETag": "\"82e8bfb872e778a4687a26e6c0b36bc1\"",
    "ContentType": "text/plain",
    "ServerSideEncryption": "AES256",
    "Metadata": {}
}
```

- オブジェクトを取得します

```
aws s3api get-object --bucket <bucket>  --key <file> <file> --endpoint
-url https://s3.example.com
```

ユーザ指定のキーによるサーバ側の暗号化（SSE-C）

SSEを使用すると、クライアントがオブジェクトを格納し、クライアントがオブジェクトで提供する一意のキーでオブジェクトを暗号化できます。オブジェクトが要求されたときに、オブジェクトを復号化して返すために同じキーを指定する必要があります。

SSE-Cの例

- テストまたはデモ目的で暗号化キーを作成できます
  - 暗号化キーを作成します

```
openssl enc -aes-128-cbc -pass pass:secret -P`
```

```
salt=E9DBB6603C7B3D2A
key=23832BAC16516152E560F933F261BF03
iv =71E87C0F6EC3C45921C2754BA131A315
```

- 生成されたキーを持つオブジェクトを配置します

```
aws s3api put-object --bucket <bucket> --key <file> --body "file" --sse
-customer-algorithm AES256 --sse-customer-key
23832BAC16516152E560F933F261BF03 --endpoint-url https://s3.example.com
```

- オブジェクトの先頭に追加します

```
aws s3api head-object --bucket <bucket> --key <file> --sse-customer
-algorithm AES256 --sse-customer-key 23832BAC16516152E560F933F261BF03
--endpoint-url https://s3.example.com
```

```
{
    "AcceptRanges": "bytes",
    "LastModified": "2022-05-02T19:20:02+00:00",
    "ContentLength": 47,
    "ETag": "\"f92ef20ab87e0e13951d9bee862e9f9a\"",
    "ContentType": "binary/octet-stream",
    "Metadata": {},
    "SSECustomerAlgorithm": "AES256",
    "SSECustomerKeyMD5": "rjGuMdjLpPV1eRuotNaPMQ=="
}
```

暗号化キーを指定しないと、「The error occurred (404) when calling the HeadObject operation: not found"」(ヘッダオブジェクト操作:見つかりません)というエラーが表示されます。

- オブジェクトを取得します

```
aws s3api get-object --bucket <bucket> --key <file> <file> --sse
-customer-algorithm AES256 --sse-customer-key
23832BAC16516152E560F933F261BF03 --endpoint-url https://s3.example.com
```

暗号化キーを指定しないと、「An error occurred（InvalidRequest）when calling the GetObject operation：the object was stored using a form of Server Side Encryption」というエラーが表示されます。オブジェクトを読み出すには、正しいパラメータを指定する必要があります。"

バケットサーバ側の暗号化（SSE-C）

SSE-Cを使用すると、バケットに格納されているすべてのオブジェクトのデフォルトの暗号化動作をクライアントで定義できます。オブジェクトはStorageGRID で管理される一意のキーで暗号化されます。オブジェクトが要求されると、StorageGRID に格納されているキーによってオブジェクトが復号化されます。

バケット**SSE-C**の例

- 新しいバケットを作成し、デフォルトの暗号化ポリシーを設定
  - 新しいバケットを作成する

```
aws s3api create-bucket --bucket <bucket> --region us-east-1
--endpoint-url https://s3.example.com
```

  - PUT Bucket encryptionの設定

```
aws s3api put-bucket-encryption --bucket <bucket> --server-side
-encryption-configuration '{"Rules":
[{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm":
"AES256"}}]}' --endpoint-url https://s3.example.com
```

- オブジェクトをバケットに配置します

```
aws s3api put-object --bucket <bucket> --key <file> --body "file"
--endpoint-url https://s3.example.com
```

- オブジェクトの先頭に追加します

```
aws s3api head-object --bucket <bucket> --key <file> --endpoint-url
https://s3.example.com
```

```
{
"AcceptRanges": "bytes",
"LastModified": "2022-05-02T20:16:23+00:00",
"ContentLength": 47,
"ETag": "\"82e8bfb872e778a4687a26e6c0b36bc1\"",
"ContentType": "binary/octet-stream",
"ServerSideEncryption": "AES256",
"Metadata": {}
}
```

- オブジェクトを取得します

```
aws s3api get-object --bucket <bucket>  --key <file> <file> --endpoint
-url https://s3.example.com
```

StorageGRID でS3オブジェクトロックをテストして実証

アロンクライン著

Object Lockは、オブジェクトが削除または上書きされないようにWORMモデルを提供します。StorageGRIDによるオブジェクトロックの実装では、規制要件を満たし、オブジェクト保持のリーガルホールドとコンプライアンスモードをサポートし、バケットのデフォルト保持ポリシーをサポートするように、Cohassetが評価されます。

このガイドでは、S3オブジェクトロックAPIについて説明します。

リーガルホールド

- オブジェクトロックリーガルホールドは、オブジェクトに適用される単純なオン/オフステータスです。

```
aws s3api put-object-legal-hold --bucket <bucket> --key <file> --legal
-hold Status=ON --endpoint-url https://s3.company.com
```

- GET処理で検証します。

```
aws s3api get-object-legal-hold --bucket <bucket> --key <file>
--endpoint-url https://s3.company.com
```

```
{
    "LegalHold": {
        "Status": "ON"
    }
}
```

- リーガルホールドをオフにします

```
aws s3api put-object-legal-hold --bucket <bucket> --key <file> --legal
-hold Status=OFF --endpoint-url https://s3.company.com
```

- GET処理で検証します。

```
aws s3api get-object-legal-hold --bucket <bucket> --key <file>
--endpoint-url https://s3.company.com
```

```
{
    "LegalHold": {
        "Status": "OFF"
    }
}
```

Complianceモード

- オブジェクトの保持には、タイムスタンプがretain untilを使用します。

```
aws s3api put-object-retention --bucket <bucket> --key <file>
--retention '{"Mode":"COMPLIANCE", "RetainUntilDate": "2025-06-
10T16:00:00"}' --endpoint-url https://s3.company.com
```

- 保持ステータスを確認

```
aws s3api get-object-retention --bucket <bucket> --key <file> --endpoint
-url https://s3.company.com
+
```

```
{
    "Retention": {
        "Mode": "COMPLIANCE",
        "RetainUntilDate": "2025-06-10T16:00:00+00:00"
    }
}
```

デフォルトの保持

- オブジェクト単位のAPIで定義された保持期限を日数と年数で設定します。

```
aws s3api put-object-lock-configuration --bucket <bucket> --object-lock
-configuration '{"ObjectLockEnabled": "Enabled", "Rule": {
"DefaultRetention": { "Mode": "COMPLIANCE", "Days": 10 }}}' --endpoint
-url https://s3.company.com
```

- 保持ステータスを確認

```
aws s3api get-object-lock-configuration --bucket <bucket> --endpoint-url
https://s3.company.com
```

```
{
    "ObjectLockConfiguration": {
        "ObjectLockEnabled": "Enabled",
        "Rule": {
            "DefaultRetention": {
                "Mode": "COMPLIANCE",
                "Days": 10
            }
        }
    }
}
```

- オブジェクトをバケットに配置します

```
aws s3api put-object --bucket <bucket> --key <file> --body "file"
--endpoint-url https://s3.example.com
```

- バケットで設定された保持期間がオブジェクトの保持タイムスタンプに変換されます。

```
aws s3api get-object-retention --bucket <bucket> --key <file> --endpoint
-url https://s3.company.com
```

```
{
    "Retention": {
        "Mode": "COMPLIANCE",
        "RetainUntilDate": "2022-03-02T15:22:47.202000+00:00"
    }
}
```

保持期間が定義されているオブジェクトの削除をテストします

オブジェクトロックは、バージョン管理の上に構築されます。保持期間はオブジェクトのバージョンで定義されます。保持が定義されているオブジェクトを削除しようとしたときに、バージョンが指定されていない場合は、削除マーカーがオブジェクトの現在のバージョンとして作成されます。

- 保持期間が定義されたオブジェクトを削除します

```
aws s3api delete-object --bucket <bucket> --key <file> --endpoint-url
https://s3.example.com
```

- バケット内のオブジェクトをリストします

```
aws s3api list-objects --bucket <bucket> --endpoint-url
https://s3.example.com
```

◦ オブジェクトがリストされていないことに注意してください。

- 削除マーカーとロックされた元のバージョンを表示するには、バージョンをリストします

```
aws s3api list-object-versions --bucket <bucket> --prefix <file>
--endpoint-url https://s3.example.com
```

```
{
    "Versions": [
        {
            "ETag": "\"82e8bfb872e778a4687a26e6c0b36bc1\"",
            "Size": 47,
            "StorageClass": "STANDARD",
            "Key": "file.txt",
            "VersionId":
"RDVDMjYwMTQtQkNDQS0xMUVDLThGOEUtNjQ3NTAwQzAxQTk1",
            "IsLatest": false,
            "LastModified": "2022-04-15T14:46:29.734000+00:00",
            "Owner": {
                "DisplayName": "Tenant01",
                "ID": "56622399308951294926"
            }
        }
    ],
    "DeleteMarkers": [
        {
            "Owner": {
                "DisplayName": "Tenant01",
                "ID": "56622399308951294926"
            },
            "Key": "file01.txt",
            "VersionId":
"QjVDQzgzOTAtQ0FGNi0xMUVDLThFMzgtQ0RGMjAwQjk0MjM1",
            "IsLatest": true,
            "LastModified": "2022-05-03T15:35:50.248000+00:00"
        }
    ]
}
```

- ロックされているオブジェクトのバージョンを削除します

```
aws s3api delete-object  --bucket <bucket> --key <file> --version-id
"<VersionId>" --endpoint-url https://s3.example.com
```

```
An error occurred (AccessDenied) when calling the DeleteObject
operation: Access Denied
```

StorageGRIDのポリシーと権限

ここでは、StorageGRID S3のポリシーと権限の例を示します。

ポリシーの構造

StorageGRIDでは、グループポリシーはAWSユーザ（IAM）のS3サービスポリシーと同じです。

StorageGRIDではグループポリシーが必要です。S3アクセスキーを持っていてユーザグループに割り当てられていないユーザや、一部の権限を許可するポリシーが設定されていないグループに割り当てられているユーザは、データにアクセスできません。

バケットポリシーとグループポリシーは、ほとんどの要素を共有します。ポリシーはJSON形式で作成され、"AWSポリシージェネレータ"

すべてのポリシーで、効果、アクション、リソースが定義されます。バケットポリシーではプリンシパルも定義されます。

Effect*はリクエストを許可するか拒否するかのどちらかになります。

プリンシパル

- バケットポリシーにのみ適用されます。
- プリンシパルは、権限を付与または拒否するアカウント/ユーザです。
- 次のように定義できます。
  - ワイルドカード"+*"

```
"Principal":"*"
```

```
"Principal":{"AWS":"*"}
```

  - テナント内のすべてのユーザのテナントID（AWSアカウントに相当）

```
"Principal": { "AWS": "27233906934684427525" }
```

- ユーザ（バケットが存在するテナント内からのローカルまたはフェデレーテッド、またはグリッド内の別のテナント）

```
"Principal": { "AWS":
"arn:aws:iam::76233906934699427431:user/tenant1user1" }
```

```
"Principal": { "AWS": "arn:aws:iam::27233906934684427525:federated-
user/tenant2user1" }
```

- グループ（バケットが配置されているテナント内からのローカルまたはフェデレーテッド、またはグリッド内の別のテナント）。

```
"Principal": { "AWS":
"arn:aws:iam::76233906934699427431:group/DevOps"  }
```

```
"Principal": { "AWS": "arn:aws:iam::27233906934684427525:federated-
group/Managers"  }
```

Action *は、ユーザに許可または拒否される一連のS3処理です。

グループポリシーの場合、S3操作を実行するには、許可されているs3：ListBucket操作が必要です。

Resource *は、プリンシパルに対してアクションの実行を許可または拒否するバケットです。必要に応じて、ポリシーアクションが有効な場合の*条件*を指定できます。

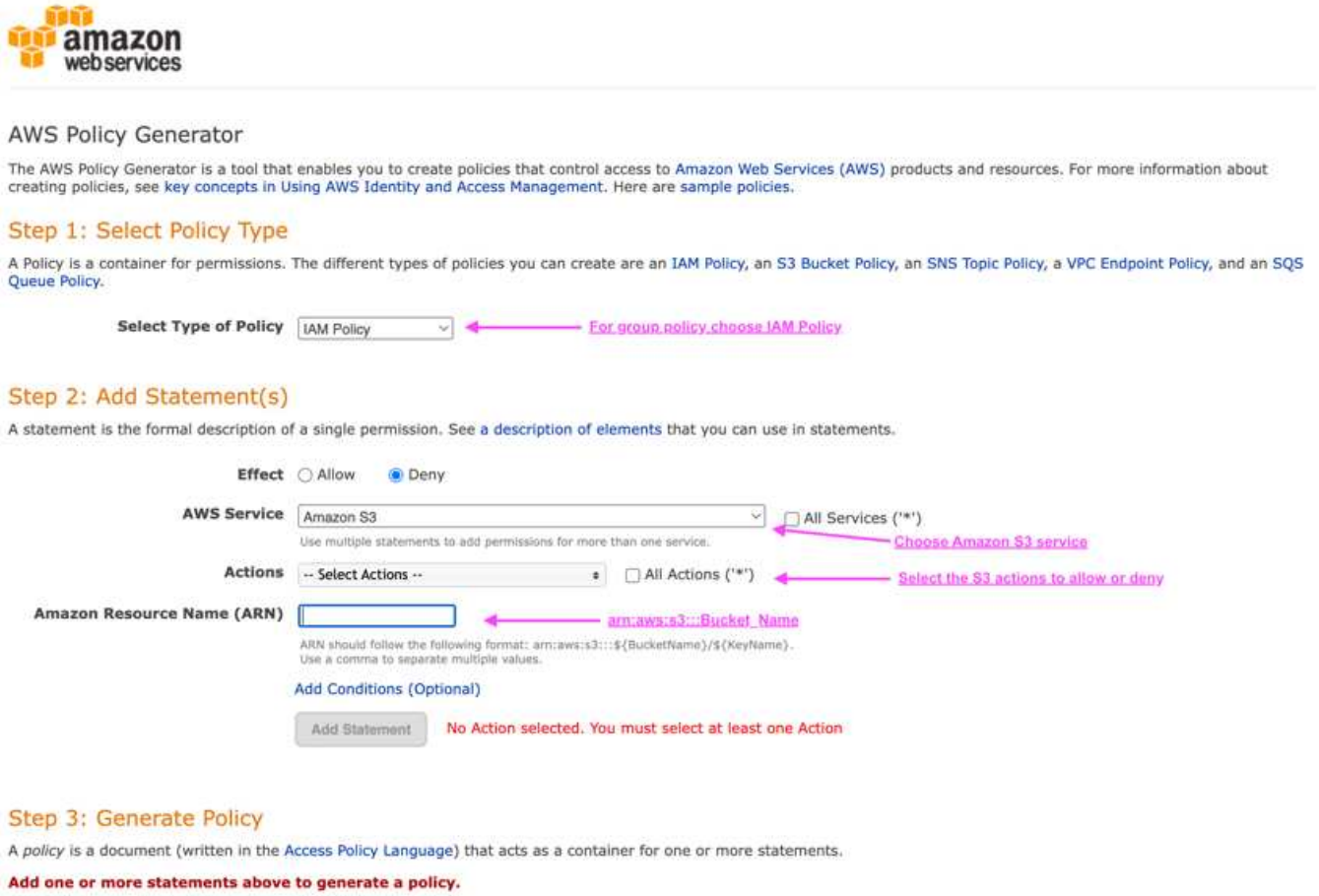
JSONポリシーの形式は次のようになります。

```
{
  "Statement": [
    {
      "Sid": "Custom name for this permission",
      "Effect": "Allow or Deny",
      "Principal": {
        "AWS": [
          "arn:aws:iam::tenant_ID::user/User_Name",
          "arn:aws:iam::tenant_ID::federated-user/User_Name",
          "arn:aws:iam::tenant_ID:group/Group_Name",
          "arn:aws:iam::tenant_ID:federated-group/Group_Name",
          "tenant_ID"
        ]
      },
      "Action": [
        "s3:ListBucket",
        "s3:Other_Action"
      ],
      "Resource": [
        "arn:aws:s3:::Example_Bucket",
        "arn:aws:s3:::Example_Bucket/*"
      ],
    }
  ]
}
```

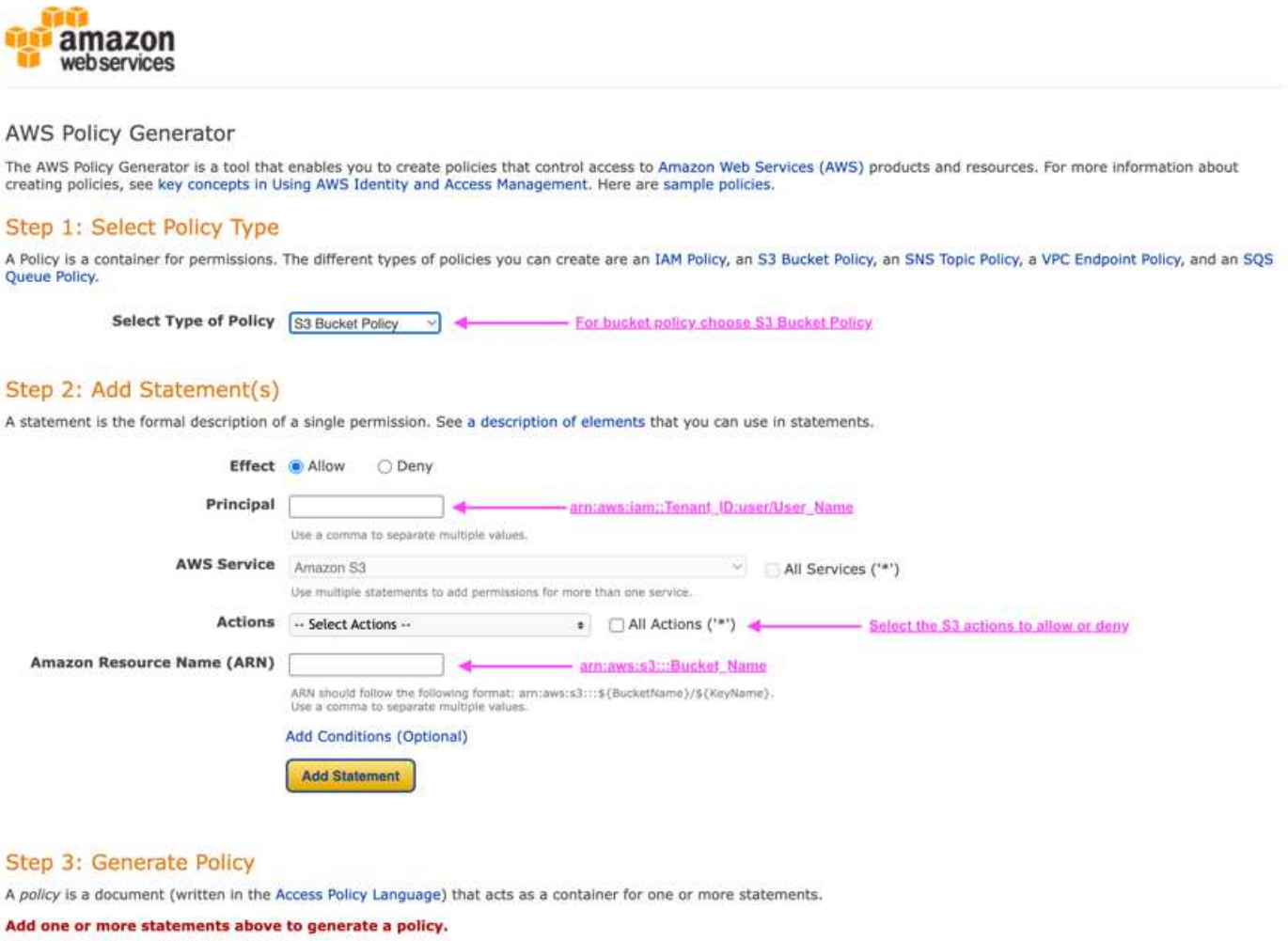
AWSポリシージェネレータの使用

AWSポリシージェネレータは、実装しようとしている正しい形式と情報でJSONコードを取得するのに役立つ優れたツールです。

StorageGRIDグループポリシーの権限を生成するには、次の手順を実行します。ポリシーのタイプに応じた**IAM**ポリシーを選択します。*希望する効果のボタンを選択します。「許可」または「拒否」です。ポリシーで**deny**権限を指定して開始し、アクションドロップダウンに**Allow**権限*を追加することを推奨します。この権限または**[**すべての操作**]**ボックスに含める**S3**操作の横にあるボックスをクリックします。[Amazon Resource Name（ARN）]ボックスにバケットパスを入力します。バケット名の前に「arn：aws：s3：：：」を含めます。ex."arn：aws：s3：：：example_bucket"

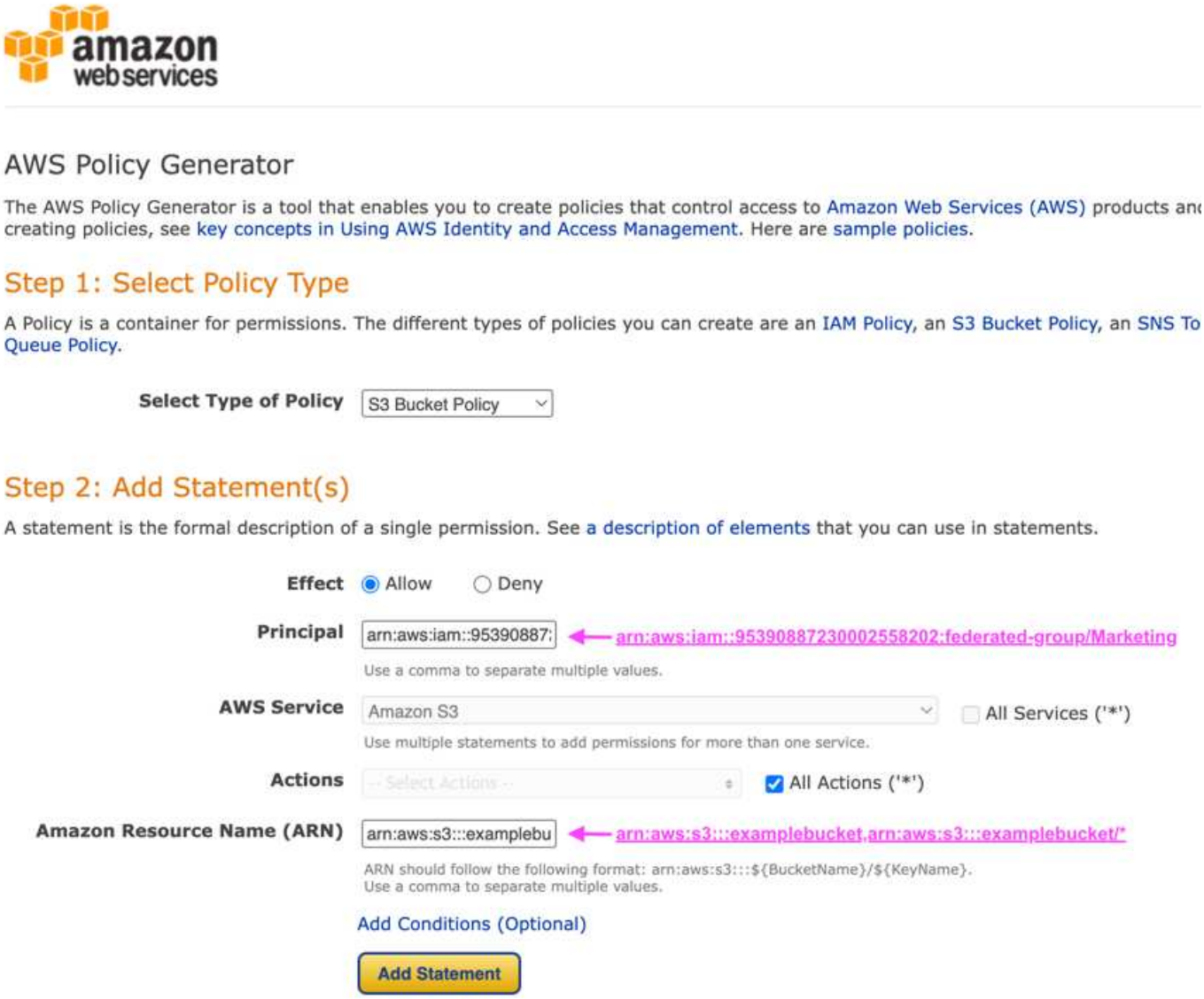


バケットポリシーの権限を生成するには：ポリシーのタイプに**[S3 Bucket Policy]**を選択します。*希望する効果のボタンを選択します。「許可」または「拒否」です。ポリシーを**deny**権限で開始し、**Principal**のユーザまたはグループ情報に**Allow permissions * Type**を追加することを推奨します。[Actions]ドロップダウンで、この権限または[All Actions]ボックスに含めるS3アクションの横にあるボックスをクリックします。*[Amazon Resource Name（ARN）]ボックスにバケットパスを入力します。バケット名の前に「arn：aws：s3：：：」を含めます。ex."arn：aws：s3：：：example_bucket"



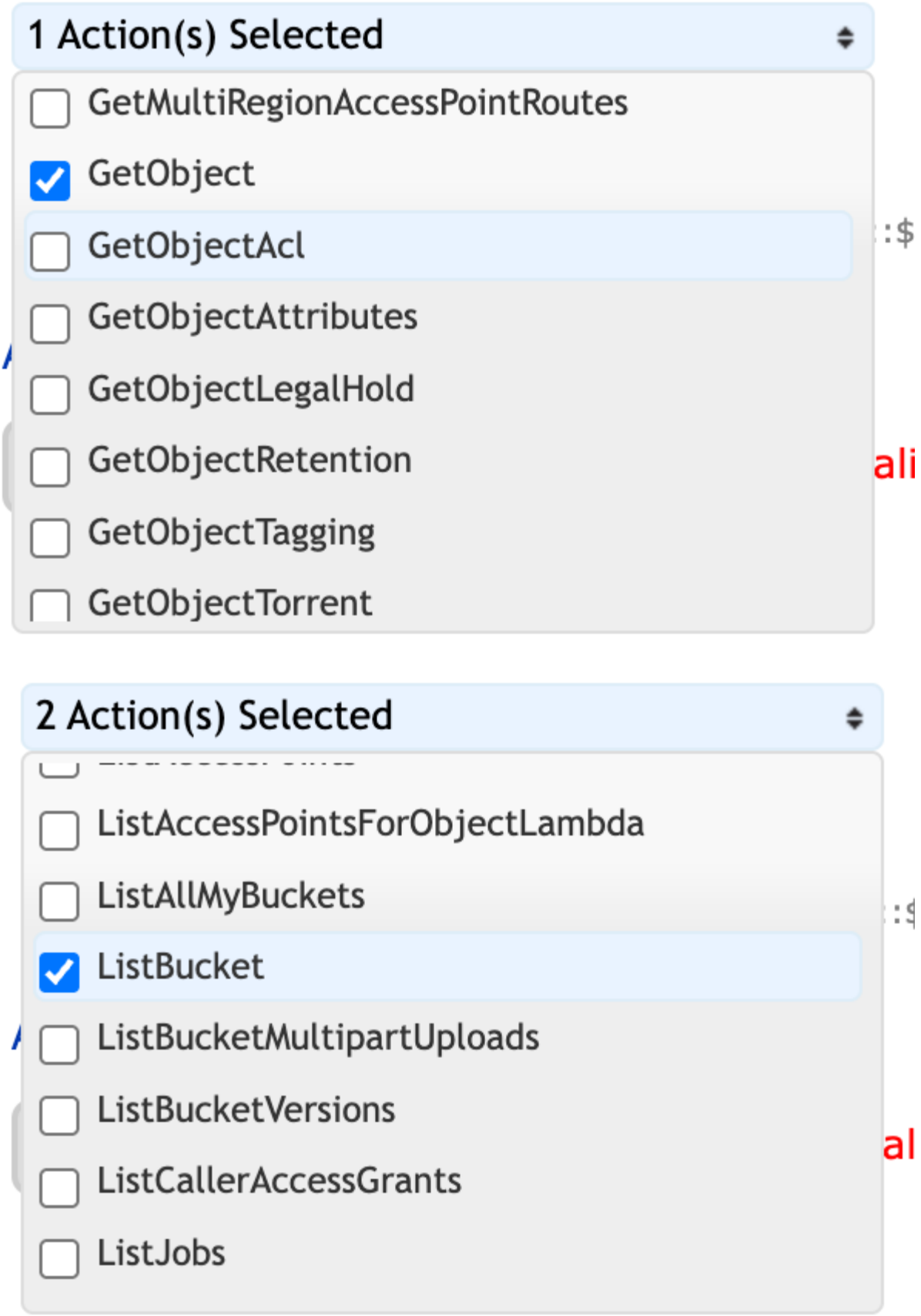
たとえば、すべてのユーザにバケット内のすべてのオブジェクトに対するGetObject処理の実行を許可するバケットポリシーを生成し、指定したアカウントの「Marketing」グループに属するユーザにのみフルアクセスを許可するとします。

- ポリシータイプとして[S3][Bucket Policy]を選択します。
- 「許可」エフェクトを選択します。
- マーケティンググループの情報を入力します。arn：aws：iam：：95390887230002558202：federated-group/Marketing
- [すべてのアクション]のボックスをクリックします。
- バケット情報を入力します。arn：aws：s3：：：example_bucket、arn：aws：s3：：：example_bucket/*



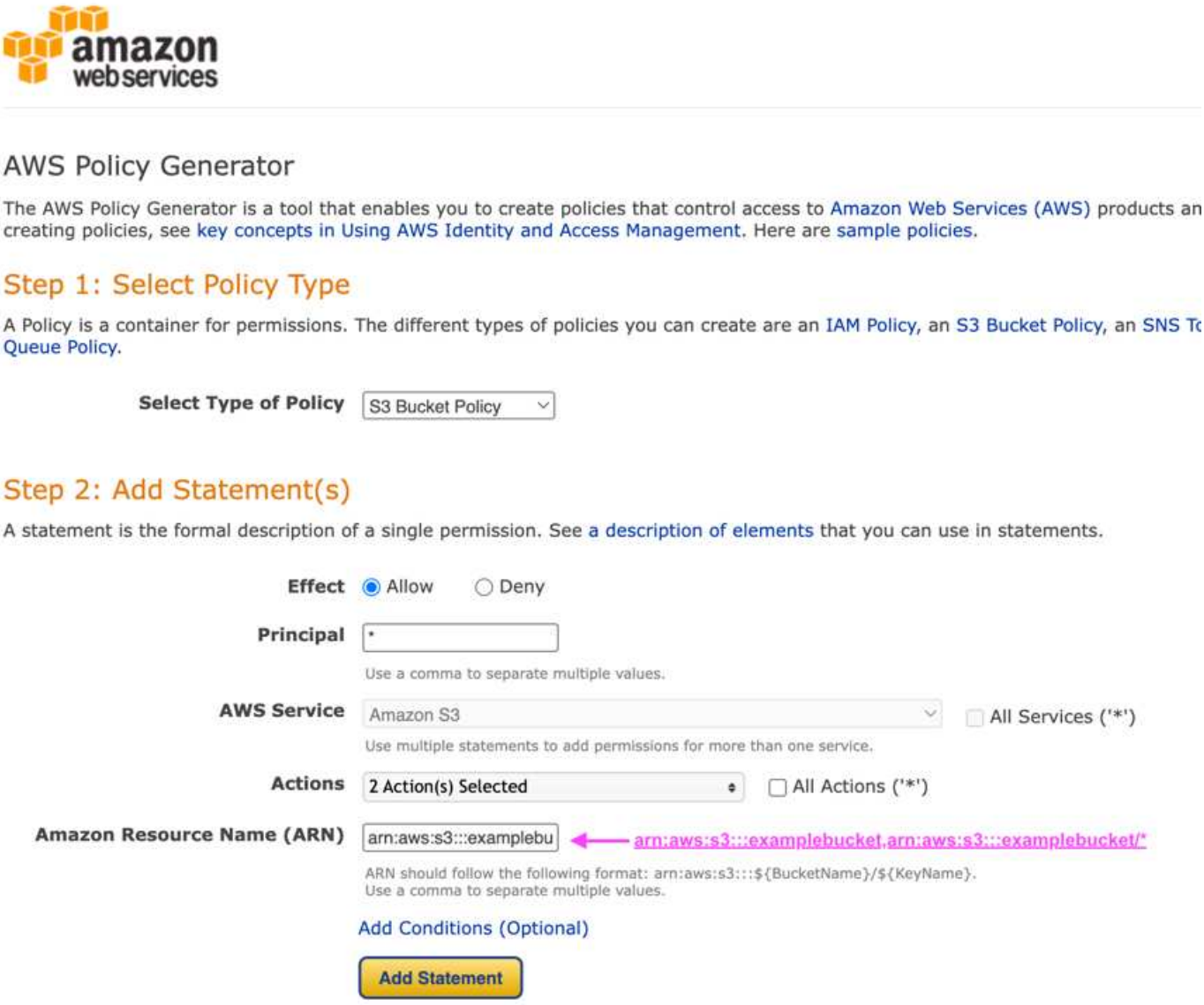
- [Add Statement]ボタンをクリックします。

- 「許可」エフェクトを選択します。
- すべてのユーザのアスタリスク「*」を入力します。
- [GetObject actions]と[ListBucket actions]の横にあるボックスをクリックします。



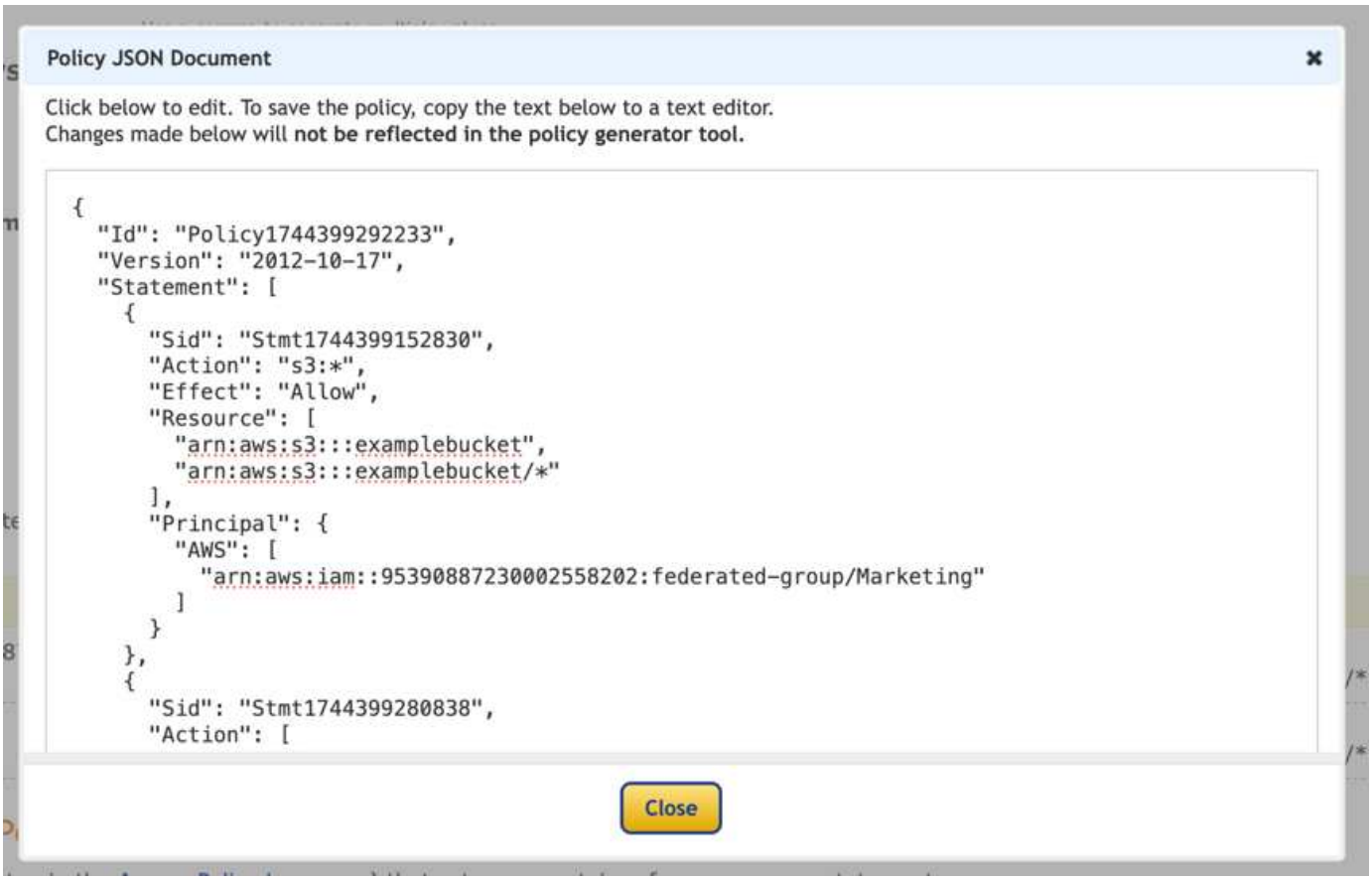
- バケット情報を入力します。arn：aws：s3：：：example_bucket、arn：aws：s3：：：example_bucket

/*



- [Add Statement]ボタンをクリックします。

- 「ポリシーの生成」ボタンをクリックすると、生成されたポリシーを含むポップアップウィンドウが表示されます。



- 次のような完全なJSONテキストをコピーします。

```
{
  "Id": "Policy1744399292233",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1744399152830",
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::example_bucket",
        "arn:aws:s3:::example_bucket/*"
      ],
      "Principal": {
        "AWS": [
          "arn:aws:iam::95390887230002558202:federated-group/Marketing"
        ]
      }
    },
    {
      "Sid": "Stmt1744399280838",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::example_bucket",
        "arn:aws:s3:::example_bucket/*"
      ],
      "Principal": "*"
    }
  ]
}
```

このJSONはそのまま使用することも、"Statement"行の上にあるIDとバージョンの行を削除することもできます。また、アクセス許可ごとに、より意味のあるタイトルでSIDをカスタマイズしたり、削除したりすることもできます。

例：

```
{
  "Statement": [
    {
      "Sid": "MarketingAllowFull",
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::example_bucket",
        "arn:aws:s3:::example_bucket/*"
      ],
      "Principal": {
        "AWS": [
          "arn:aws:iam::95390887230002558202:federated-group/Marketing"
        ]
      }
    },
    {
      "Sid": "EveryoneReadOnly",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::example_bucket",
        "arn:aws:s3:::example_bucket/*"
      ],
      "Principal": "*"
    }
  ]
}
```

グループポリシー（IAM）

ホームディレクトリ形式のバケットアクセス

このグループポリシーでは、users usernameという名前のバケット内のオブジェクトへのアクセスのみがユーザに許可されます。

```
{
"Statement": [
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::home",
      "Condition": {
        "StringLike": {
          "s3:prefix": "${aws:username}/*"
        }
      }
    },
    {
      "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
      "Effect": "Allow",
      "Action": "s3:*Object",
      "Resource": "arn:aws:s3:::home/?/?/${aws:username}/*"
    }

  ]
}
```

オブジェクトロックバケットの作成を拒否します

このグループポリシーでは、ユーザがバケットを作成してそのバケットでオブジェクトロックを有効にすることはできません。

> このポリシーはStorageGRID UIでは適用されず、S3 APIでのみ適用されます。

```
{
    "Statement": [
        {
            "Action": "s3:*",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Action": [
                "s3:PutBucketObjectLockConfiguration",
                "s3:PutBucketVersioning"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```

オブジェクトロックの保持制限

このバケットポリシーでは、Object-Lockの保持期間が10日以下に制限されます

```
{
 "Version":"2012-10-17",
 "Id":"CustSetRetentionLimits",
 "Statement": [
   {
    "Sid":"CustSetRetentionPeriod",
    "Effect":"Deny",
    "Principal":"*",
    "Action": [
      "s3:PutObjectRetention"
    ],
    "Resource":"arn:aws:s3:::testlock-01/*",
    "Condition": {
      "NumericGreaterThan": {
        "s3:object-lock-remaining-retention-days":"10"
      }
    }
   }
  ]
}
```

ユーザーによるオブジェクトの削除を**versionId**で制限します

このグループポリシーは、versionIdでバージョン管理オブジェクトを削除することをユーザに制限します

```
{
    "Statement": [
        {
            "Action": [
                "s3:DeleteObjectVersion"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Action": "s3:*",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```

グループを読み取り専用アクセスで単一のサブディレクトリ（プレフィックス）に制限する

このポリシーでは、グループのメンバーにバケット内のサブディレクトリ（プレフィックス）への読み取り専用アクセスを許可します。バケット名は「study」、サブディレクトリは「study01」です。

```
{
    "Statement": [
        {
            "Sid": "AllowUserToSeeBucketListInTheConsole",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Sid": "AllowRootAndstudyListingOfBucket",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
```

```
                "arn:aws:s3::: study"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:prefix": [
                        "",
                        "study01/"
                    ],
                    "s3:delimiter": [
                        "/"
                    ]
                }
            }
        },
        {
            "Sid": "AllowListingOfstudy01",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::study"
            ],
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "study01/*"
                    ]
                }
            }
        },
        {
            "Sid": "AllowAllS3ActionsInstudy01Folder",
            "Effect": "Allow",
            "Action": [
                "s3:Getobject"
            ],
            "Resource": [
                "arn:aws:s3:::study/study01/*"
            ]
        }
    ]
}
```

バケットポリシー

バケットを読み取り専用アクセス権を持つ単一ユーザに制限します

このポリシーでは、1人のユーザにバケットへの読み取り専用アクセスを許可し、他のすべてのユーザへのアクセスを明示的に拒否します。評価を迅速に行うには、ポリシーの先頭にDenyステートメントをグループ化することを推奨します。

```
{
    "Statement": [
        {
            "Sid": "Deny non user1",
            "Effect": "Deny",
            "NotPrincipal": {
                "AWS": "arn:aws:iam::349215141330028336 65:user/user1"
            },
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::bucket1",
                "arn:aws:s3:::bucket1/*"
            ]
        },
        {
            "Sid": "Allow user1 read access to bucket bucket1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::349215141330028336 65:user/user1"
            },
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket1",
                "arn:aws:s3:::bucket1/*"
            ]
        }
    ]
}
```

バケットを読み取り専用アクセス権を持つ少数のユーザに制限する。

```
{
    "Statement": [
      {
        "Sid": "Deny all S3 actions to employees 002-005",
        "Effect": "deny",
        "Principal": {
          "AWS": [
            "arn:aws:iam::465215141330027038882:user/employee-002",
            "arn:aws:iam::465215141330027038882:user/employee-003",
            "arn:aws:iam::465215141330027038882:user/employee-004",
            "arn:aws:iam::465215141330027038882:user/employee-005"
          ]
        },
        "Action": "*",
        "Resource": [
          "arn:aws:s3:::databucket1",
          "arn:aws:s3:::databucket1/*"
        ]
      },
      {
        "Sid": "Allow read-only access for employees 002-005",
        "Effect": "Allow",
        "Principal": {
          "AWS": [
            "arn:aws:iam::465215141330027038882:user/employee-002",
            "arn:aws:iam::465215141330027038882:user/employee-003",
            "arn:aws:iam::465215141330027038882:user/employee-004",
            "arn:aws:iam::465215141330027038882:user/employee-005"
          ]
        },
        "Action": [
          "s3:GetObject",
          "s3:GetObjectTagging",
          "s3:GetObjectVersion"
        ],
        "Resource": [
          "arn:aws:s3:::databucket1",
          "arn:aws:s3:::databucket1/*"
        ]
      }
    ]
}
```

バケット内のバージョン管理オブジェクトのユーザによる削除を制限する

このバケットポリシーは、ユーザ（ユーザID「56622399308951294926」で識別）がversionIdでバージョン管理オブジェクトを削除することを制限します

```
{
  "Statement": [
    {
      "Action": [
        "s3:DeleteObjectVersion"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:s3:::verdeny/*",
      "Principal": {
        "AWS": [
          "56622399308951294926"
        ]
      }
    },
    {
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::verdeny/*",
      "Principal": {
        "AWS": [
          "56622399308951294926"
        ]
      }
    }
  ]
}
```

StorageGRIDのバケットライフサイクル

S3 ライフサイクル設定を作成して、特定のオブジェクトが StorageGRID システムから削除されるタイミングを制御できます。

ライフサイクル構成とは

ライフサイクル設定は、特定の S3 バケット内のオブジェクトに適用される一連のルールです。各ルールは、影響を受けるオブジェクトと、それらのオブジェクトの有効期限（特定の日付または日数後）を指定します。

各オブジェクトは、S3バケットライフサイクルまたはILMポリシーの保持設定に従います。S3バケットライフサイクルが設定されている場合は、バケットライフサイクルフィルタに一致するオブジェクトのILMポリシーがライフサイクル有効期限のアクションで上書きされます。バケットライフサイクルフィルタに一致しないオブジェクトには、ILMポリシーの保持設定が使用されます。オブジェクトがバケットライフサイクルフィル

タに一致し、有効期限の操作が明示的に指定されていない場合、ILMポリシーの保持設定は使用されず、オブジェクトのバージョンが無期限に保持されることが暗黙的に示されます。

そのため、 ILM ルールの配置手順がオブジェクトに引き続き適用されていても、オブジェクトがグリッドから削除されることがあります。あるいは、オブジェクトに対するILM配置指示が失効した後でも、オブジェクトがグリッド上に保持される可能性がある。

StorageGRID では、 1 つのライフサイクル設定で最大 1 、 000 個のライフサイクルルールがサポートされます。各ルールには、次の XML 要素を含めることができます。

- Expiration ：指定した日付に達した場合、またはオブジェクトが取り込まれたときから指定した日数に達した場合にオブジェクトを削除します。
- NoncurrentVersionExpiration ：指定した日数に達したオブジェクトを削除します。これは、オブジェクトが最新でなくなったときからです。
- フィルタ（プレフィックス、タグ）
- ステータス *ID

StorageGRID では、次のバケット処理を使用してライフサイクル設定を管理できます。

- DeleteBucketLifecycle
- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration

ライフサイクルポリシーの構造

ライフサイクル設定を作成するための最初の手順として、 1 つ以上のルールを含む JSON ファイルを作成します。たとえば、この JSON ファイルには次の 3 つのルールが含まれています。

1. *ルール1*は、プレフィックス「category1/」に一致し、key2の値が「tag2」であるオブジェクトにのみ適用されます。Expirationパラメータは、フィルターに一致するオブジェクトが2020年8月22日の午前0時に期限切れになることを指定します。
2. *ルール2*は、プレフィックス「category2/」に一致するオブジェクトにのみ適用されます。Expirationパラメータは、フィルターに一致するオブジェクトが取り込まれてから100日後に有効期限切れになることを指定します。

   日数を指定するルールは、オブジェクトが取り込まれた時点を基準とした相対的なルールです。現在の日付が取り込み日と日数を超えている場合は、ライフサイクル設定の適用後すぐに一部のオブジェクトがバケットから削除される可能性があります。

3. *ルール3*は、プレフィックス「category3/」に一致するオブジェクトにのみ適用されます。Expirationパラメータは、一致するオブジェクトの非現行バージョンが、非現行バージョンになってから50日後に期限切れになることを指定します。

```
{
    "Rules": [
        {
            "ID": "rule1",
            "Filter": {
                "And": {
                    "Prefix": "category1/",
                    "Tags": [
                        {
                            "Key": "key2",
                            "Value": "tag2"
                        }
                    ]
                }
            },
            "Expiration": {
                "Date": "2020-08-22T00:00:00Z"
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule2",
            "Filter": {
                "Prefix": "category2/"
            },
            "Expiration": {
                "Days": 100
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule3",
            "Filter": {
                "Prefix": "category3/"
            },
            "NoncurrentVersionExpiration": {
                "NoncurrentDays": 50
            },
            "Status": "Enabled"
        }
    ]
}
```

バケットにライフサイクル設定を適用

ライフサイクル設定ファイルを作成したら、PutBucketLifecycleConfiguration要求を発行してバケットに適用します。

この要求は、サンプルファイルのライフサイクル設定をという名前のバケット内のオブジェクトに適用し`testbucket`ます。

```
aws s3api --endpoint-url <StorageGRID endpoint> put-bucket-lifecycle-
configuration
--bucket testbucket --lifecycle-configuration file://bktjson.json
```

ライフサイクル設定がバケットに正常に適用されたことを確認するには、GetBucketLifecycleConfiguration要求を問題します。例えば：

```
aws s3api --endpoint-url <StorageGRID endpoint> get-bucket-lifecycle-
configuration
 --bucket testbucket
```

標準（バージョン管理されていない）バケットのライフサイクル ポリシーの例

90日後にオブジェクトを削除する

ユースケース：このポリシーは、一時ファイル、ログ、中間処理データなど、限られた期間のみ関連するデータの管理に最適です。メリット：ストレージコストを削減し、バケットを整理整頓できます。

```
{
    "Rules": [
      {
        "ID": "Delete after 90 day rule",
        "Filter": {},
        "Status": "Enabled",
          "Expiration": {
              "Days": 90
        }
      }
    ]
}
```

バージョン管理されたバケットのライフサイクル ポリシーの例

10日後に非最新版を削除する

ユースケース：このポリシーは、時間の経過とともに蓄積され、大量のスペースを消費する可能性のある、最

新バージョンではないオブジェクトのストレージ管理に役立ちます。メリット：最新バージョンのみを保持することで、ストレージ使用量を最適化します。

```
{
    "Rules": [
            {
        "ID": "NoncurrentVersionExpiration 10 day rule",
        "Filter": {},
        "Status": "Enabled",
          "NoncurrentVersionExpiration": {
              "NoncurrentDays": 10
        }
    }
    ]
}
```

5つの非最新バージョンを保持する

使用例: 回復または監査の目的で、限られた数の以前のバージョンを保持する場合に便利です。利点: 十分な履歴と回復ポイントを確保するために、十分な数の非最新バージョンを保持します。

```
{
    "Rules": [
      {
        "ID": "NewerNoncurrentVersions 5 version rule",
        "Filter": {},
        "Status": "Enabled",
        "NoncurrentVersionExpiration": {
            "NewerNoncurrentVersions": 5
        }
    }
    ]
}
```

他のバージョンが存在しない場合は削除マーカーを削除します

ユースケース：このポリシーは、すべての非最新バージョンを削除した後に残る削除マーカーを管理するのに役立ちます。これらのマーカーは時間の経過とともに蓄積される可能性があります。メリット：不要な混乱を軽減します。

```
{
    "Rules": [
    {
        "ID": "Delete marker cleanup rule",
        "Filter": {},
        "Status": "Enabled",
        "Expiration": {
        "ExpiredObjectDeleteMarker": true
        }
    }
    ]
}
```

現在のバージョンは **30** 日後に削除され、現在のバージョン以外のバージョンは **60** 日後に削除され、他のバージョンが存在しなくなったら現在のバージョンの削除によって作成された削除マーカーが削除されます。

ユースケース：削除マーカーを含む、現在のバージョンと非現在のバージョンの完全なライフサイクルを提供します。メリット：十分なリカバリポイントと履歴を保持しながら、ストレージコストを削減し、バケットを整理された状態に保ちます。

```
{
  "Rules": [
    {
      "ID": "Delete current version",
      "Filter": {},
      "Status": "Enabled",
      "Expiration": {
        "Days": 30
      }
    },
    {
      "ID": "noncurrent version retention",
      "Filter": {},
      "Status": "Enabled",
      "NoncurrentVersionExpiration": {
        "NoncurrentDays": 60
      }
    },
    {
      "ID": "Markers",
      "Filter": {},
      "Status": "Enabled",
      "Expiration": {
        "ExpiredObjectDeleteMarker": true
      }
    }
  ]
}
```

他のバージョンがない削除マーカーを削除し、「**accounts_** プレフィックス」を持つオブジェクトについては **4** つの非最新バージョンと少なくとも **30** 日分の履歴を保持し、他のすべてのオブジェクト バージョンについては **2** つのバージョンと少なくとも **10** 日分の履歴を保持します。

ユースケース：特定のオブジェクトと他のオブジェクトに固有のルールを適用し、削除マーカーを含む現在のバージョンと非現在のバージョンのライフサイクル全体を管理します。メリット：ストレージコストを削減し、バケットを整理しながら、十分なリカバリポイントと履歴を保持することで、多様なクライアント要件に対応できます。

```
{
  "Rules": [
    {
      "ID": "Markers",
      "Filter": {},
      "Status": "Enabled",
      "Expiration": {
        "ExpiredObjectDeleteMarker": true
      }
    },
    {
      "ID": "accounts version retention",
      "Filter": {"Prefix":"account_"},
      "Status": "Enabled",
      "NoncurrentVersionExpiration": {
        "NewerNoncurrentVersions": 4,
        "NoncurrentDays": 30
      }
    },
    {
      "ID": "noncurrent version retention",
      "Filter": {},
      "Status": "Enabled",
      "NoncurrentVersionExpiration": {
        "NewerNoncurrentVersions": 2,
        "NoncurrentDays": 10
      }
    }
  ]
}
```

まとめ

- ライフサイクル ポリシーを定期的に確認および更新し、ILM およびデータ管理の目標に合わせて調整します。
- ポリシーを広範囲に適用する前に、非本番環境またはバケットでテストして、意図したとおりに機能することを確認します。
- ロジック構造が複雑になる可能性があるため、ルールをより直感的にするために説明的なIDを使用します。
- これらのバケット ライフサイクル ポリシーがストレージの使用状況とパフォーマンスに与える影響を監視し、必要な調整を行います。

著作権に関する情報

商標に関する情報