过程和API示例

StorageGRID solutions and resources

NetApp
November 21, 2025

This PDF was generated from https://docs.netapp.com/zh-cn/storagegrid-enable/examples/test-demonstrate-S3-encryption.html on November 21, 2025. Always check docs.netapp.com for the latest.

目录

过程和API示例 . . . 1
在StorageGRID 上测试和演示S3加密选项 . . . 1
服务器端加密(SSR) . . . 1
使用客户提供的密钥(SSI-C)进行服务器端加密 . . . 2
存储分段服务器端加密(SSI-S3) . . . 3
测试并演示StorageGRID 上的S3对象锁定 . . . 4
合法保留 . . . 4
合规模式 . . . 5
默认保留 . . . 6
测试删除已定义保留的对象 . . . 7
StorageGRID中的策略和权限 . . . 9
策略的结构 . . . 9
使用AWS策略生成器 . . . 11
组策略(IAM) . . . 18
存储分段策略 . . . 23
StorageGRID 中的存储桶生命周期 . . . 25
什么是生命周期配置 . . . 25
生命周期策略的结构 . . . 26
将生命周期配置应用于存储分段 . . . 28
标准（非版本化）存储桶的生命周期策略示例 . . . 28
版本控制存储桶的生命周期策略示例 . . . 28
结论 . . . 32

过程和API示例

在StorageGRID 上测试和演示S3加密选项

作者：*Aron Klein*

StorageGRID 和S3 API提供了多种不同的方法来加密空闲数据。要了解更多信息，请参见 "查看 StorageGRID 加密方法"。

本指南将演示S3 API加密方法。

服务器端加密(SSR)

使用SSE、客户端可以存储对象、并使用由StorageGRID 管理的唯一密钥对其进行加密。请求对象时、该对象将通过存储在StorageGRID 中的密钥进行解密。

SS一 示例

- 使用SSR放置对象

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

- 对对象执行HEAD以验证加密

```
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": {}
}
```

- 获取对象

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

使用客户提供的密钥(SSI-C)进行服务器端加密

通过"SSE "、客户端可以存储对象、并使用客户端随对象提供的唯一密钥对其进行加密。请求对象时、必须提供相同的密钥才能解密并返回对象。

SSI-C示例

- 出于测试或演示目的、您可以创建加密密钥
  - 创建加密密钥

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

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

- 放置具有生成密钥的对象

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

- 为对象设置头

```
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=="
            }
```

如果不提供加密密钥、则会收到错误"An error occurred (404) when calling the HeadObject operation：not found"(调用HeadObject操作时出错(404)：未找到)

- 获取对象

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

如果不提供加密密钥、则在调用GetObject操作时将收到错误"An error occurred (InvalidRequest)：The object was stored using a form of Server side Encryption"。要检索对象、必须提供正确的参数。"

存储分段服务器端加密(SSI-S3)

SSI-S3允许客户端为存储在存储在存储分段中的所有对象定义默认加密行为。这些对象使用由StorageGRID 管理的唯一密钥进行加密。请求对象时、该对象将通过存储在StorageGRID 中的密钥进行解密。

存储分段**SSI-S3**示例

- 创建新存储分段并设置默认加密策略
  - 创建新存储分段

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

  - 放入存储分段加密

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

- 将对象放入存储分段中

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

- 为对象设置头

```
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": {}
}
```

- 获取对象

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

测试并演示StorageGRID 上的S3对象锁定

作者：*Aron Klein*

对象锁定提供了一个WORM模型、用于防止删除或覆盖对象。对StorageGRID 对象锁定实施情况进行了评估、以帮助满足法规要求、支持对象保留的合法保留和合规模式以及默认存储分段保留策略。

本指南将演示S3对象锁定API。

合法保留

- 对象锁定合法保持是应用于对象的简单开/关状态。

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

- 使用GET操作对其进行验证。

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

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

- 关闭合法保留

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

- 使用GET操作对其进行验证。

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

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

合规模式

- 对象保留是使用"保留到"时间戳完成的。

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

- 验证保留状态

```
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"
    }
}
```

默认保留

- 将保留期限设置为天数和年数以及使用每个对象API定义的保留截止日期。

```
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
```

- 验证保留状态

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

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

- 将对象放入存储分段中

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

- 存储分段上设置的保留持续时间将转换为对象上的保留时间戳。

```
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"
    }
}
```

测试删除已定义保留的对象

对象锁定基于版本控制构建。保留是在对象的某个版本上定义的。如果尝试删除定义了保留的对象、但未指定版本、则会创建一个删除标记作为对象的当前版本。

- 删除定义了保留的对象

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

- 列出存储分段中的对象

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

  - 请注意、此对象未列出。

- 列出可查看删除标记的版本以及原始锁定版本

```
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": "566223993089512949 26"
            }
        }
    ],
    "DeleteMarkers": [
        {
            "Owner": {
                "DisplayName": "Tenant01",
                "ID": "566223993089512949 26"
            },
            "Key": "file01.txt",
            "VersionId":
"QjVDQzgzOTAtQ0FGNi0xMUVDLThFMzgtQ0RGMjAwQjk0MjM1",
            "IsLatest": true,
            "LastModified": "2022-05-03T15:35:50.248000+00:00"
        }
    ]
}
```

- 删除对象的锁定版本

```
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中的策略和权限

下面是StorageGRID S3中的策略和权限示例。

策略的结构

在StorageGRID中、组策略与AWS用户(IAM) S3服务策略相同。

StorageGRID中需要使用组策略。具有S3访问密钥但未分配给用户组或分配给组但策略未授予某些权限的用户将无法访问任何数据。

存储分段和组策略共享大多数相同的元素。策略以json格式构建、可使用生成 "AWS策略生成器"

所有策略都将定义效果、操作和资源。存储分段策略还将定义主体。

其*效果*是允许或拒绝请求。

负责人

- 仅适用于存储分段策略。
- 主体是指被授予或拒绝权限的帐户/用户。
- 可以定义为：
  - 通配符"*"

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

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

  - 租户中所有用户的租户ID (相当于AWS帐户)

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

  - 用户(存储分段所在租户的本地或联合用户、或网格中的其他租户)

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

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

  - 组(存储分段所在租户的本地或联合组、或网格中的其他租户)。

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

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

“操作”是授予或拒绝给用户的一组S3操作。

对于组策略、用户需要执行S3：ListBucket"操作才能执行任何S3操作。

资源*是指被授予或拒绝对其执行操作的主体的一个或多个分段。(可选)策略操作的有效时间可以是一个*条件*。

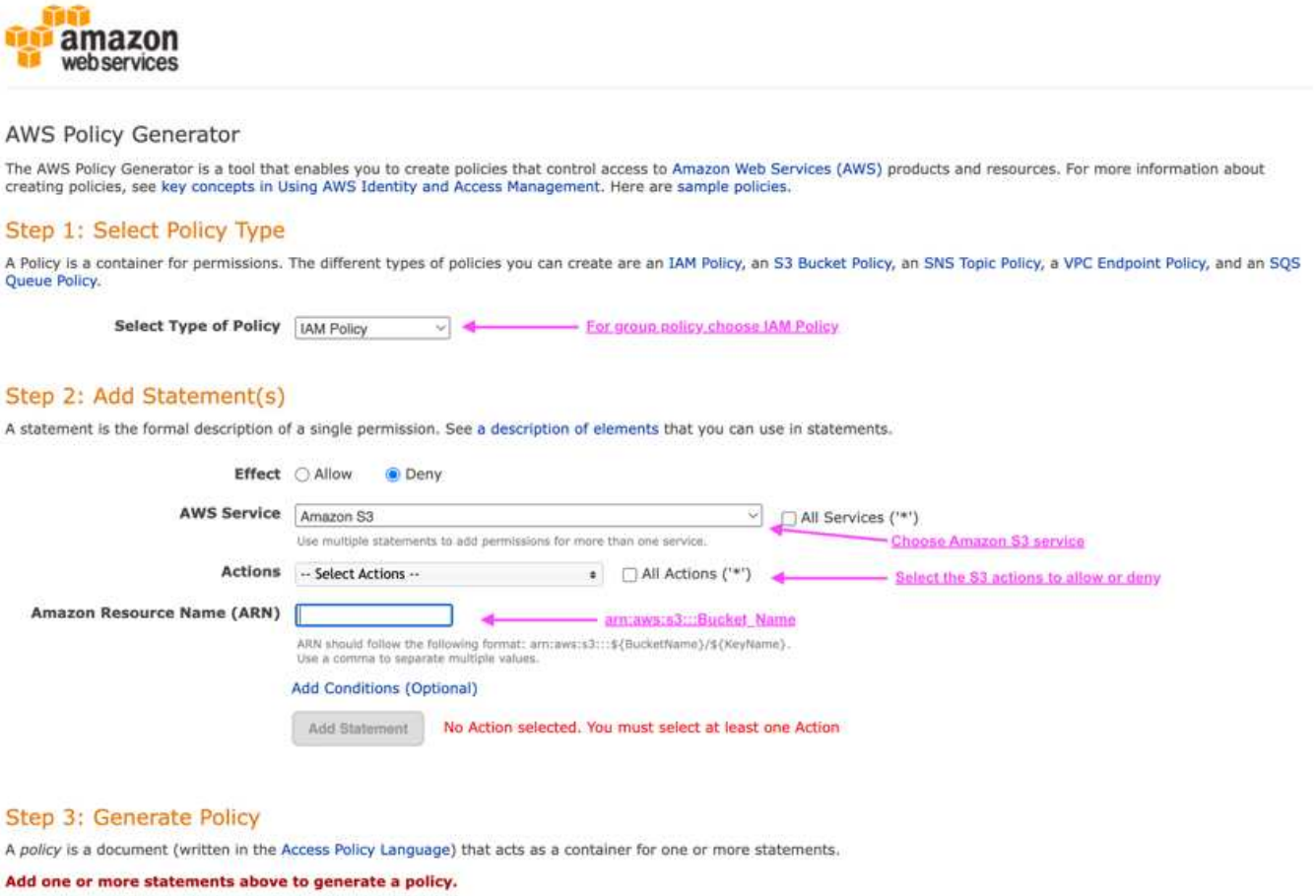
JSON策略的格式如下所示：

```
{
  "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/*"
      ],
    }
  ]
}
```

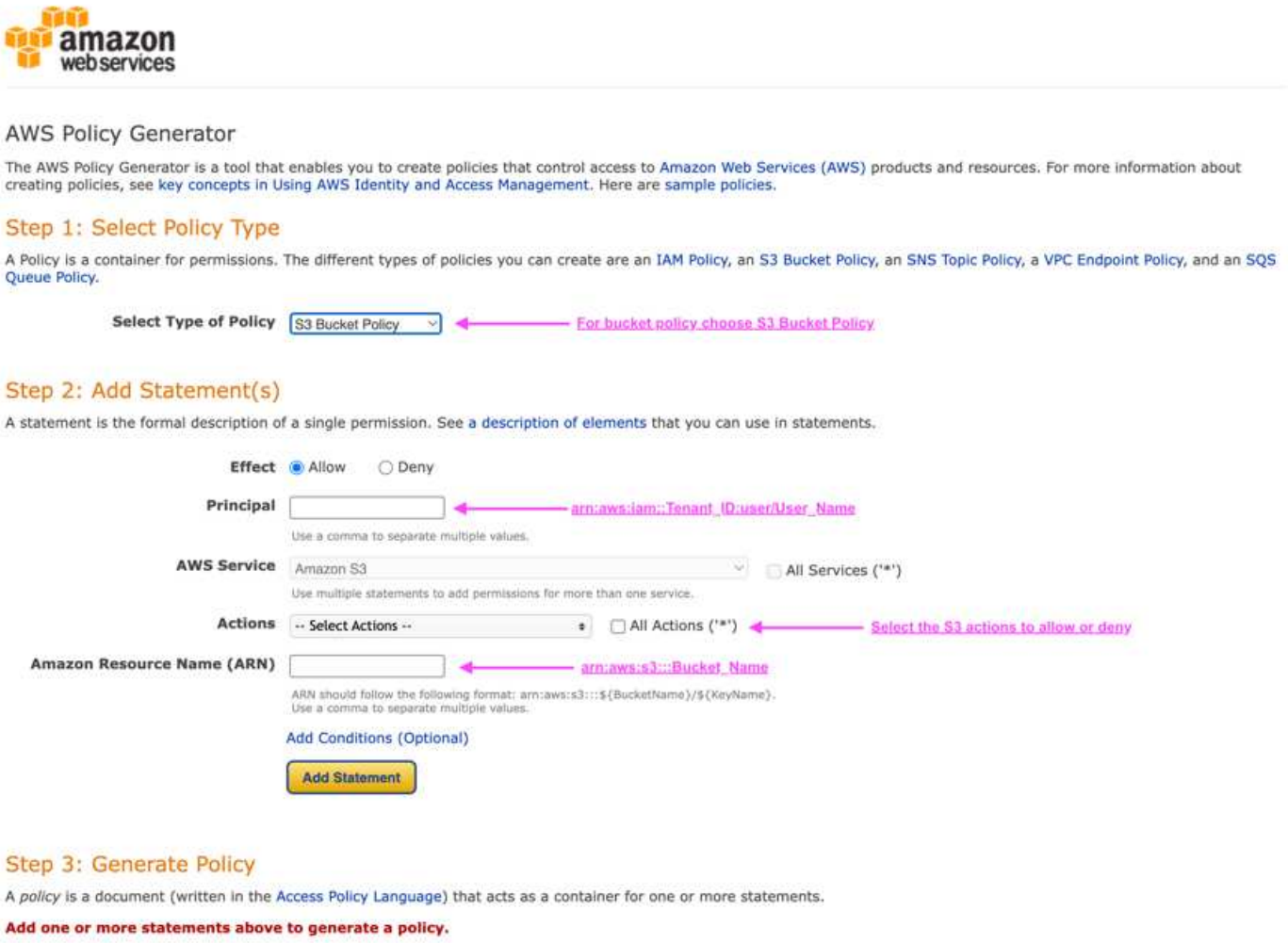
使用AWS策略生成器

AWS策略生成器是一个很好的工具、可帮助您获取具有正确格式的json代码以及您尝试实施的信息。

要为StorageGRID组策略生成权限、请执行以下操作：*为策略类型选择IAM策略。*选择所需效果的按钮-允许或拒绝。最好先使用拒绝权限启动策略、然后在操作下拉列表中添加允许权限*单击要包含在此权限中的尽可能多的S3操作旁边的框或"所有操作"框。*在Amazon资源名称(ARN)框中键入存储分段路径。在存储分段名称前包括"arn：aws：s3：：：："。例如、"ARN：AWS：S3：：：exple_bket"

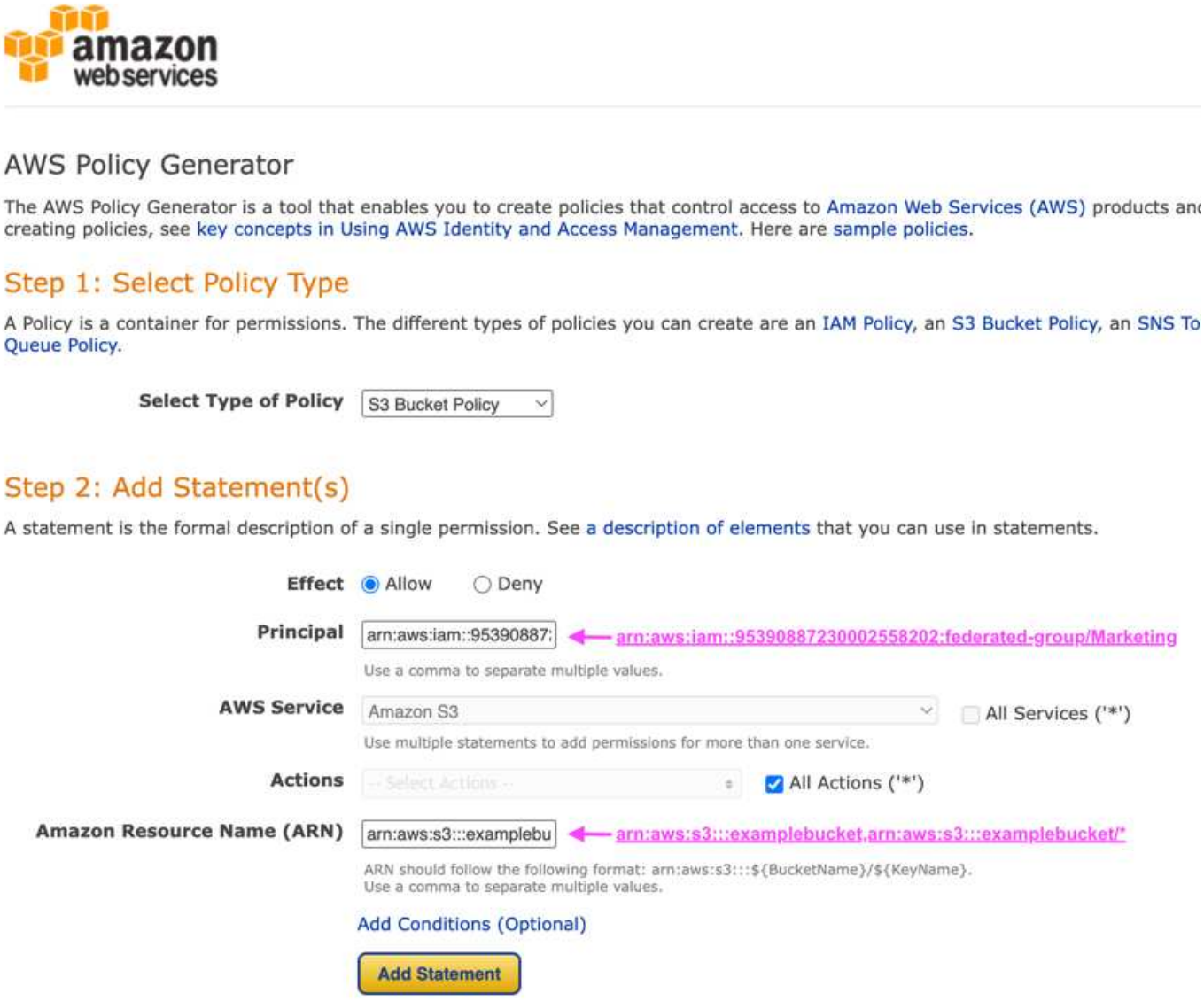


要为存储分段策略生成权限、请执行以下操作：*为策略类型选择S3存储分段策略。*选择所需效果的按钮-允许或拒绝。最好先使用拒绝权限启动策略、然后在主体的用户或组信息中添加允许权限*键入。*在操作下拉列表中、单击要包含在此权限中的S3操作旁边的框或"所有操作"框。*在Amazon资源名称(ARN)框中键入存储分段路径。在存储分段名称前包括"arn：aws：s3：：：："。例如、"ARN：AWS：S3：：：exple_bket"



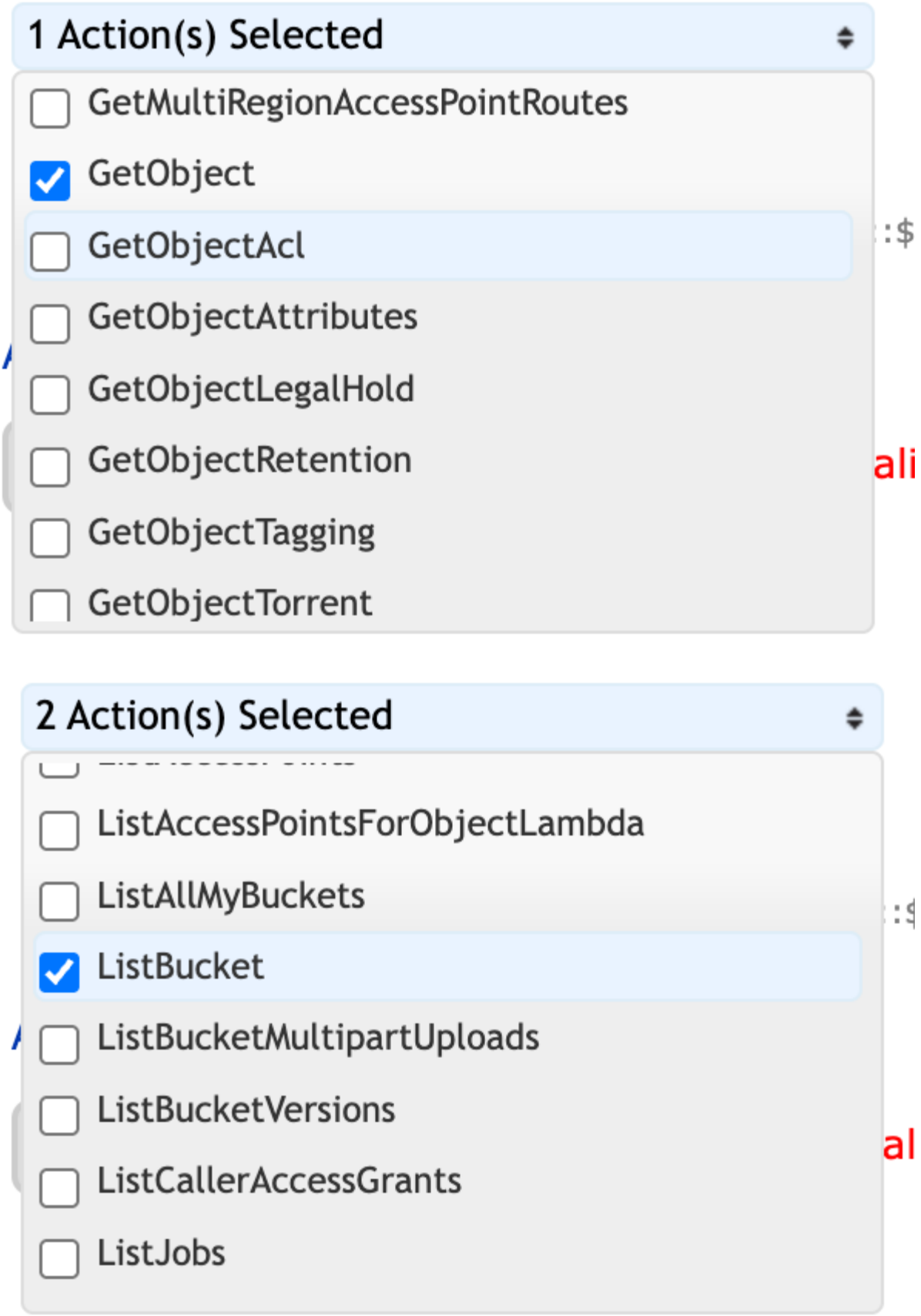
例如、如果要生成存储分段策略、以允许所有用户对存储分段中的所有对象执行GetObject操作、而仅允许指定帐户中属于组"Marketing"的用户进行完全访问。

- 选择S3存储分段策略作为策略类型。
- 选择允许效果
- 输入Marketing组信息：arn：aws：iam：：95390887230002558202：联合组/营销
- 单击"所有操作"框
- 输入存储分段信息—arn：aws：s3：：：：exple_bket,arn：aws：s3：：：：exple_bket/*

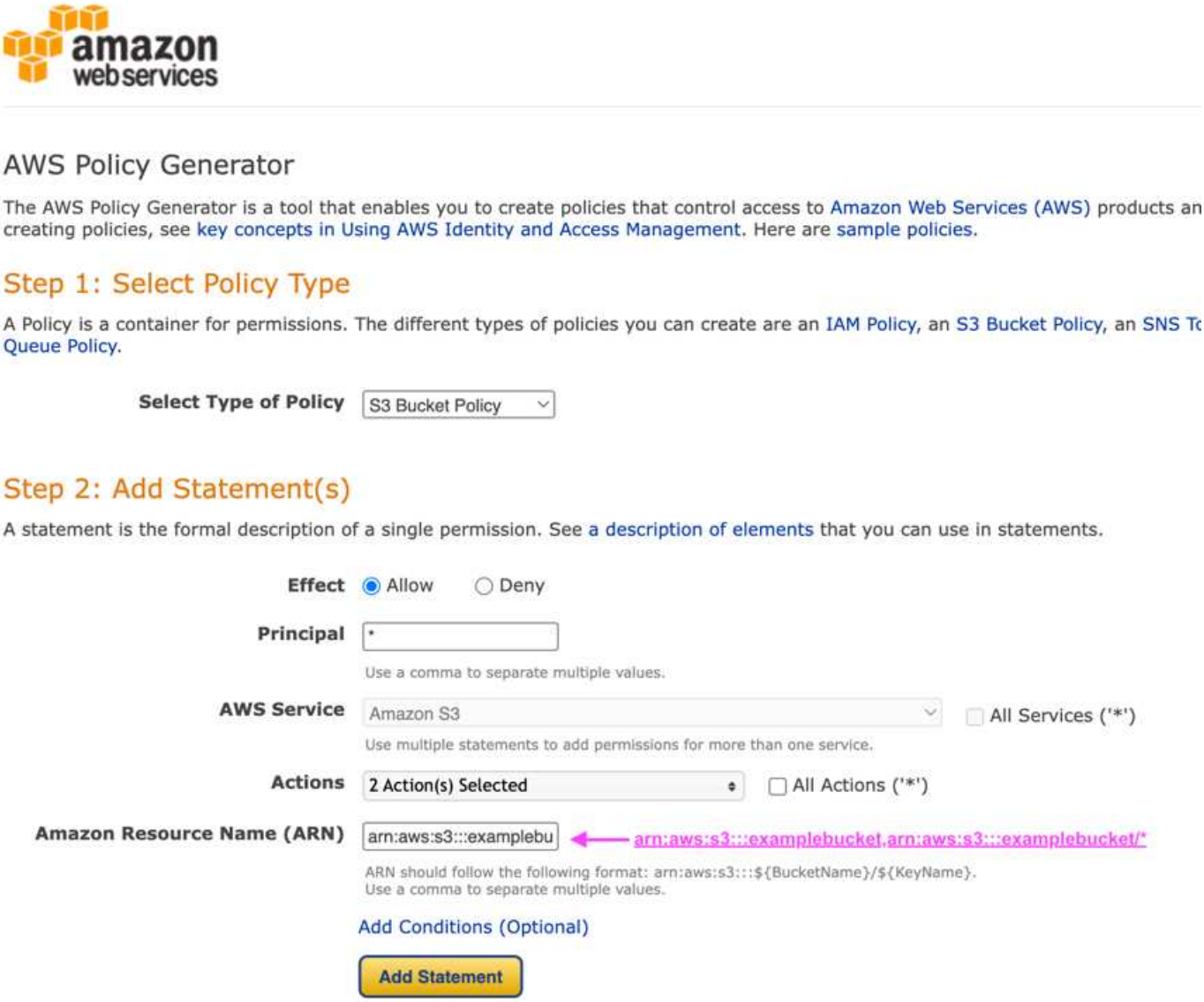


- 单击"Add Statement"(添加诊断代码)按钮

- 选择允许效果
- 为每个人输入星号*
- 单击GetObject和ListBucket"旁边的框

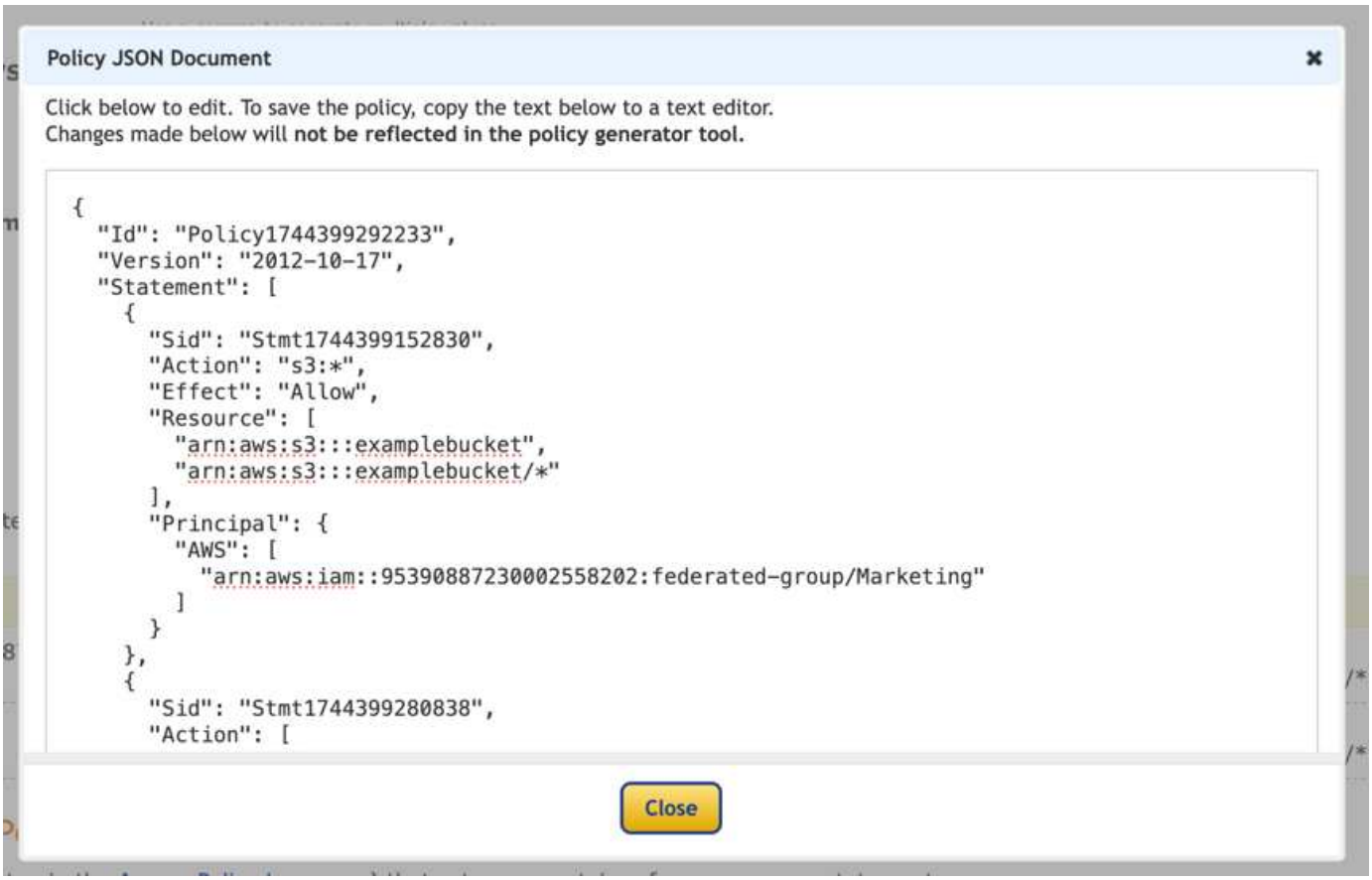


- 输入存储分段信息—arn：aws：s3：：：：exple_bket,arn：aws：s3：：：：exple_bket/*



- 单击"Add Statement"(添加诊断代码)按钮

- 单击"生成策略"按钮、此时将显示一个弹出窗口、其中会显示您生成的策略。



- 复制完整的json文本、如下所示：

```
{
  "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": "*"
    }
  ]
}
```

此json可以按原样使用、也可以删除"Statement"行上方的ID和版本行、您可以自定义每个权限的Sid、并为每个权限指定更有意义的标题、也可以删除这些内容。

例如：

```
{
  "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": "*"
    }
  ]
}
```

组策略(IAM)

主目录模式的存储分段访问

此组策略仅允许用户访问名为Users username的分段中的对象。

```
{
"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}/*"
    }

  ]
}
```

拒绝创建对象锁定分段

此组策略将限制用户创建在存储分段上启用了对象锁定的存储分段。

此策略不会在StorageGRID UI中强制实施、而是仅通过S3 API强制实施。

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

对象锁定保留限制

此存储分段策略会将对象锁定保留期限限制为10天或更短

```
{
 "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"
      }
    }
   }
  ]
}
```

按版本ID限制用户删除对象

此组策略将限制用户按版本ID删除受版本控制的对象

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

将组限制为具有只读访问权限的单个子目录(前缀)

此策略允许组成员对分段中的子目录(前缀)具有只读访问权限。分段名称为"study"、子目录为"study01"。

```
{
    "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/*"
            ]
        }
    ]
}
```

存储分段策略

将存储分段限制为具有只读访问权限的单个用户

此策略允许单个用户对某个存储分段拥有只读访问权限、并明确授予所有其他用户的访问权限。将deny语句分组在策略顶部是一种较好的做法、可以加快评估速度。

```
{
    "Statement": [
        {
            "Sid": "Deny non user1",
            "Effect": "Deny",
            "NotPrincipal": {
                "AWS": "arn:aws:iam::349215141330028336665: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::349215141330028336665:user/user1"
            },
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket1",
                "arn:aws:s3:::bucket1/*"
            ]
        }
    ]
}
```

将存储分段限制为少数具有只读访问权限的用户。

```
{
    "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/*"
        ]
      }
    ]
}
```

限制用户删除分段中受版本控制的对象

此存储分段策略将限制用户(由用户ID "56622399308951294926"标识)按版本ID删除版本控制的对象

```
{
  "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 中的存储桶生命周期

您可以创建 S3 生命周期配置，以控制何时从 StorageGRID 系统中删除特定对象。

什么是生命周期配置

生命周期配置是一组应用于特定 S3 分段中的对象的规则。每个规则都指定受影响的对象以及这些对象的到期时间（在特定日期或一定天数后）。

每个对象都遵循S3存储分段生命周期或ILM策略的保留设置。配置S3存储分段生命周期后、对于与存储分段生命周期筛选器匹配的对象、生命周期到期操作将覆盖ILM策略。与存储分段生命周期筛选器不匹配的对象将使用ILM策略的保留设置。如果某个对象与存储分段生命周期筛选器匹配、并且未明确指定到期操作、则不会使用ILM策略的保留设置、这意味着对象版本将永久保留。

因此，即使 ILM 规则中的放置说明仍适用于某个对象，该对象也可能会从网格中删除。或者，即使对象的任何

ILM 放置指令已经失效，对象仍可能保留在网格上

StorageGRID 在一个生命周期配置中最多支持 1 ， 000 条生命周期规则。每个规则可以包含以下 XML 元素：

- 到期日期：从对象载入开始，在达到指定日期或达到指定天数时删除对象。
- NoncurrentVersionExpiration ：从对象变为非最新状态开始，在达到指定天数时删除对象。
- 筛选器（前缀，标记）
- 状态 *ID

StorageGRID 支持使用以下存储分段操作来管理生命周期配置：

- DeleteBucketLifecycle
- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration

生命周期策略的结构

作为创建生命周期配置的第一步，您需要创建一个包含一个或多个规则的 JSON 文件。例如，此 JSON 文件包含三个规则，如下所示：

1. 规则 **1** 仅适用于与前缀 category1/ 匹配且 key2 值为 tag2 的对象。Expiration参数指定与该筛选条件匹配的对象将于 2020 年 8 月 22 日午夜到期。
2. 规则 **2** 仅适用于与前缀 category2/ 匹配的对象。Expiration参数指定与过滤器匹配的对象将在被提取后 100 天过期。

   > 指定天数的规则与对象的载入时间相关。如果当前日期超过载入日期加上天数，则在应用生命周期配置后，可能会立即从存储分段中删除某些对象。

3. 规则 **3** 仅适用于与前缀 category3/ 匹配的对象。Expiration参数指定匹配对象的任何非当前版本将在变为非当前版本后 50 天过期。

```
{
    "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"
        }
    ]
}
```

将生命周期配置应用于存储分段

创建生命周期配置文件后、您可以通过发出PutBucketLifecycleConfiguration请求将其应用于存储分段。

此请求将示例文件中的生命周期配置应用于名为的分段中的对象 `testbucket`。

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

要验证生命周期配置是否已成功应用于存储分段、请发送问题描述a GetBucketLifecycleConfiguration请求。例如：

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

标准（非版本化）存储桶的生命周期策略示例

90 天后删除对象

用例：此策略非常适合管理仅在有限时间内相关的数据，例如临时文件、日志或中间处理数据。优势：降低存储成本并确保存储桶整洁有序。

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

版本控制存储桶的生命周期策略示例

10 天后删除非当前版本

用例：此策略有助于管理非当前版本对象的存储，这些对象会随着时间的推移而累积并占用大量空间。优势：通过仅保留最新版本来优化存储空间利用率。

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

保留 **5** 个非当前版本

用例：当您想保留有限数量的先前版本以用于恢复或审计目的时很有用。好处：保留足够的非当前版本以确保足够的历史记录和恢复点。

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

当不存在其他版本时移除删除标记

用例：此策略有助于管理所有非当前版本删除后留下的删除标记，这些标记会随着时间的推移而累积。好处：减少不必要的混乱。

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

30 天后删除当前版本，**60** 天后删除非当前版本，一旦不存在其他版本，则删除当前版本创建的删除标记。

用例：为当前版本和非当前版本提供完整的生命周期，包括删除标记。优势：降低存储成本，确保存储桶整洁有序，同时保留足够的恢复点和历史记录。

```
{
  "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
      }
    }
  ]
}
```

删除没有其他版本的删除标记，为带有“**accounts_** 前缀”的对象保留 **4** 个非当前版本和至少 **30** 天的历史记录，为所有其他对象版本保留 **2** 个版本和至少 **10** 天的历史记录。

用例：为特定对象以及其他对象提供独特的规则，以管理当前版本和非当前版本（包括删除标记）的完整生命周期。优势：降低存储成本，确保存储桶整洁有序，同时保留足够的恢复点和历史记录，以满足各种客户需求。

```
{
  "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
      }
    }
  ]
}
```

结论

- 定期审查和更新生命周期政策，并使其与 ILM 和数据管理目标保持一致。
- 在广泛应用策略之前，先在非生产环境或存储桶中测试策略，以确保其按预期工作
- 使用描述性 ID 作为规则使其更直观，因为逻辑结构可能变得复杂
- 监控这些存储桶生命周期策略对存储使用情况和性能的影响，以做出必要的调整。

版权信息

版权所有 © 2025 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。 美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 http://www.netapp.com/TM 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。