Procédures et exemples d'API

StorageGRID solutions and resources

NetApp
November 21, 2025

Sommaire

Procédures et exemples d'API

Tester et démontrer les options de cryptage S3 sur StorageGRID

Par Aron Klein

StorageGRID et l'API S3 proposent plusieurs façons de chiffrer vos données au repos. Pour en savoir plus, voir "Étudiez les méthodes de cryptage StorageGRID".

Ce guide présente les méthodes de chiffrement de l'API S3.

Chiffrement côté serveur (SSE)

SSE permet au client de stocker un objet et de le chiffrer à l'aide d'une clé unique gérée par StorageGRID. Lorsque l'objet est demandé, il est décrypté par la clé stockée dans StorageGRID.

Exemple SSE

- PLACER un objet avec SSE

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

- DIRIGEZ l'objet pour vérifier le chiffrement

```
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": {}
}
```

- OBTENIR l'objet

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

Chiffrement côté serveur avec clés fournies par le client (SSE-C)

SSE permet au client de stocker un objet et de le chiffrer à l'aide d'une clé unique fournie par le client avec l'objet. Lorsque l'objet est demandé, la même clé doit être fournie pour décrypter et renvoyer l'objet.

Exemple SSE-C.

- Vous pouvez créer une clé de chiffrement à des fins de test ou de démonstration
  - Créez une clé de chiffrement

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

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

- Placer un objet avec la clé générée

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

- Dirigez l'objet

```
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=="
}
```

Si vous ne fournissez pas la clé de cryptage, vous recevrez une erreur « une erreur s'est produite (404) lors de l'appel de l'opération HeadObject : introuvable ».

- Obtenir l'objet

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

Si vous ne fournissez pas la clé de cryptage, vous recevrez une erreur "une erreur s'est produite (InvalidRequest) lors de l'appel de l'opération GetObject: L'objet a été stocké à l'aide d'une forme de chiffrement côté serveur. Les paramètres corrects doivent être fournis pour récupérer l'objet. »

Chiffrement côté serveur godet (SSE-S3)

SSE-S3 permet au client de définir un comportement de cryptage par défaut pour tous les objets stockés dans un compartiment. Les objets sont chiffrés avec une clé unique gérée par StorageGRID. À la demande de l'objet, celui-ci est décrypté par la clé stockée dans StorageGRID.

Exemple de godet SSE-S3

- Créez un compartiment et définissez une règle de chiffrement par défaut
  - Créer un nouveau compartiment

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

  - Put bucket Encryption

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

- Placer un objet dans le godet

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

- Dirigez l'objet

```
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": {}
}
```

- OBTENIR l'objet

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

Testez et faites une démonstration du verrouillage d'objet S3 sur StorageGRID

Par Aron Klein

Le verrouillage d'objet fournit un modèle WORM pour éviter que les objets ne soient supprimés ou remplacés. L'implémentation StorageGRID du verrouillage d'objet est une fonctionnalité qui est évaluée afin de respecter les exigences réglementaires, et qui prend en charge le mode de conservation légale et de conformité pour la conservation des objets et les règles de conservation des compartiments par défaut.

Ce guide présente l'API de verrouillage d'objet S3.

Obligation légale

- La mise en attente légale de verrouillage d'objet est un état activé/désactivé simple appliqué à un objet.

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

- Vérifiez-le avec une opération GET.

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

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

- Désactiver la mise en attente légale

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

- Vérifiez-le avec une opération GET.

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

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

Mode de conformité

- La conservation de l'objet s'effectue avec une conservation jusqu'à l'horodatage.

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

- Vérifiez l'état de la rétention

```
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"
    }
}
```

Conservation par défaut

- Définissez la période de conservation en jours et années par rapport à une date de conservation définie avec l'api par objet.

```
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
```

- Vérifiez l'état de la rétention

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

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

- Placer un objet dans le godet

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

- La durée de conservation définie dans le compartiment est convertie en horodatage de conservation sur l'objet.

```
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"
    }
}
```

Test de la suppression d'un objet avec une rétention définie

Le verrouillage d'objet est basé sur la gestion des versions. La conservation est définie sur une version de l'objet. Si une tentative de suppression d'un objet avec une rétention définie et qu'aucune version n'est spécifiée, un marqueur de suppression est créé comme version actuelle de l'objet.

- Supprimez l'objet dont la conservation est définie

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

- Lister les objets dans le compartiment

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

  - Notez que l'objet n'est pas répertorié.
- Répertorier les versions pour voir le marqueur de suppression et la version verrouillée d'origine

```
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"
        }
    ]
}
```

- Supprimer la version verrouillée de l'objet

```
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
```

Stratégies et autorisations dans StorageGRID

Voici des exemples de règles et d'autorisations dans StorageGRID S3.

Structure d'une politique

Dans StorageGRID, les règles de groupe sont identiques aux règles de service S3 de l'utilisateur AWS (IAM).

Les stratégies de groupe sont requises dans StorageGRID. Un utilisateur avec des clés d'accès S3, mais qui n'est pas affecté à un groupe d'utilisateurs, ou affecté à un groupe sans règle lui accordant certaines autorisations, ne pourra accéder à aucune donnée.

Les règles de compartiment et de groupe partagent la plupart des mêmes éléments. Les stratégies sont créées au format json et peuvent être générées à l'aide de "Générateur de règles AWS"

Toutes les règles définissent l'effet, les actions et les ressources. Les règles de compartiment définiront également un principal.

Effet sera soit permettre ou refuser la demande.

Le **principal**

- S'applique uniquement aux politiques de compartiment.
- L'entité de sécurité est le(s) compte(s)/utilisateur(s) auquel(s) les autorisations ont été accordées ou refusées.
- Peut être défini comme :
  - Un caractère générique "+*"

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

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

  - ID de locataire pour tous les utilisateurs d'un locataire (équivalent au compte AWS)

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

  - Utilisateur (local ou fédéré depuis le locataire où réside le compartiment ou un autre locataire de la grille)

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

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

- Un groupe (local ou fédéré depuis le locataire où réside le compartiment ou un autre locataire de la grille).

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

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

Action est l'ensemble des opérations S3 accordées ou refusées aux utilisateurs.

> Pour les stratégies de groupe, l'action s3:ListBucket autorisée est requise pour que les utilisateurs puissent exécuter n'importe quelle action S3.

La ressource **Resource** est le compartiment ou les compartiments auxquels les principaux ont été accordés ou refusés la capacité d'exécuter les actions sur. En option, il peut y avoir une **condition** lorsque l'action de stratégie est valide.

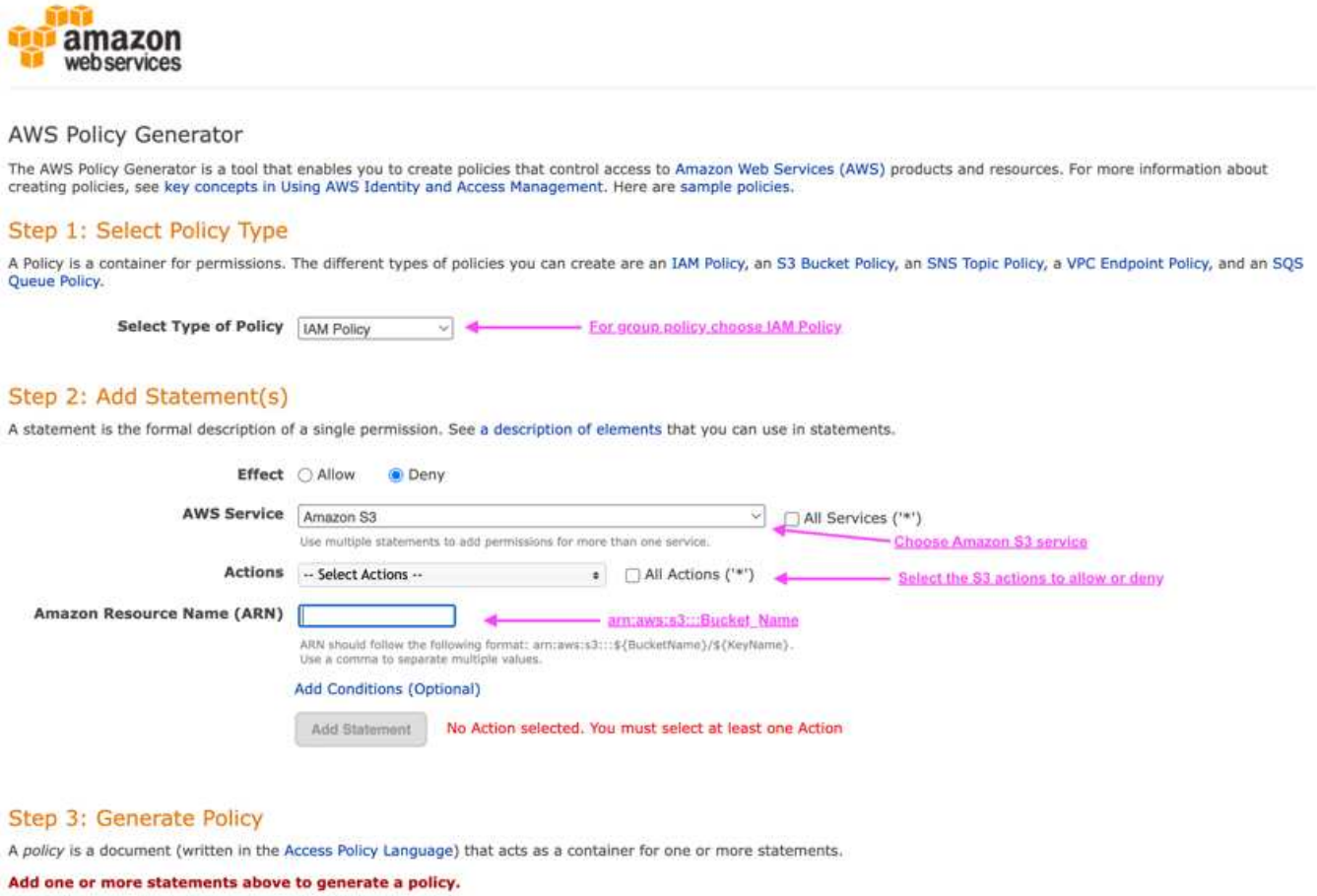
Le format de la politique JSON se présente comme suit :

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

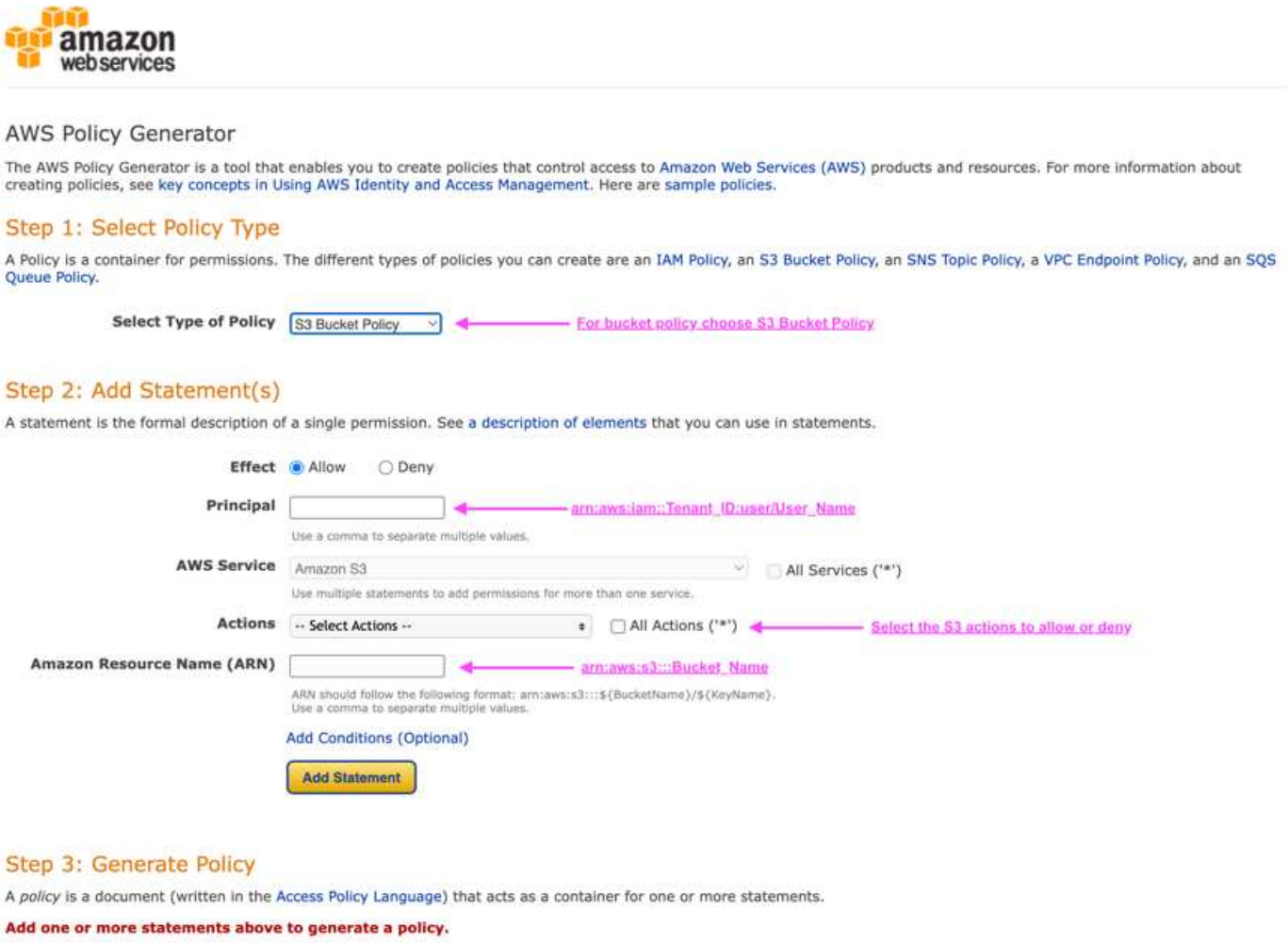
À l'aide du générateur de règles AWS

Le générateur de règles AWS est un excellent outil pour vous aider à obtenir le code json avec le format et les informations que vous essayez d'implémenter.

Pour générer les autorisations d'une stratégie de groupe StorageGRID : * Choisissez la stratégie IAM pour le type de stratégie. * Sélectionnez le bouton pour l'effet désiré - Autoriser ou refuser. Il est recommandé de démarrer vos stratégies avec les autorisations de refus, puis d'ajouter les autorisations d'autorisation * dans la liste déroulante actions, cliquez sur la case en regard du nombre d'actions S3 que vous souhaitez inclure dans cette autorisation ou dans la zone « toutes les actions ». * Tapez les chemins de compartiment dans la zone Amazon Resource Name (ARN). Incluez "arn:aws:s3:::" avant le nom du compartiment. Ex. « arn:aws:s3:::example_bucket »

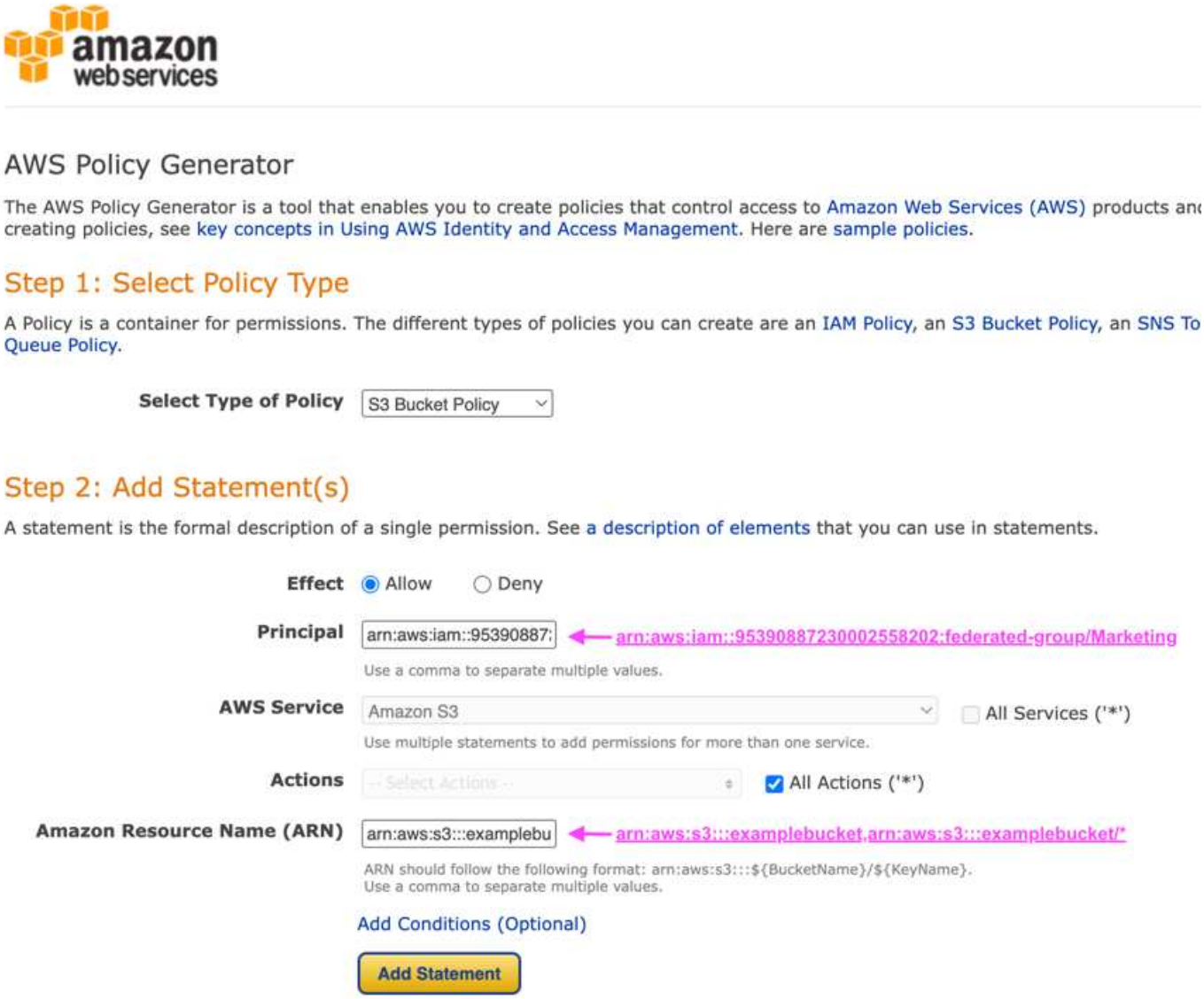


Pour générer les autorisations d'une règle de compartiment : * Choisissez la règle de compartiment S3 pour le type de règle. * Sélectionnez le bouton pour l'effet désiré - Autoriser ou refuser. Il est recommandé de démarrer vos stratégies avec les autorisations refuser, puis d'ajouter le type Autoriser les autorisations * dans les informations sur l'utilisateur ou le groupe pour le principal. * Dans la liste déroulante actions, cliquez sur la case en regard du nombre d'actions S3 que vous souhaitez inclure dans cette autorisation ou de la case « toutes les actions ». * Tapez les chemins de compartiment dans la zone Amazon Resource Name (ARN). Incluez "arn:aws:s3:::" avant le nom du compartiment. Ex. « arn:aws:s3:::example_bucket »



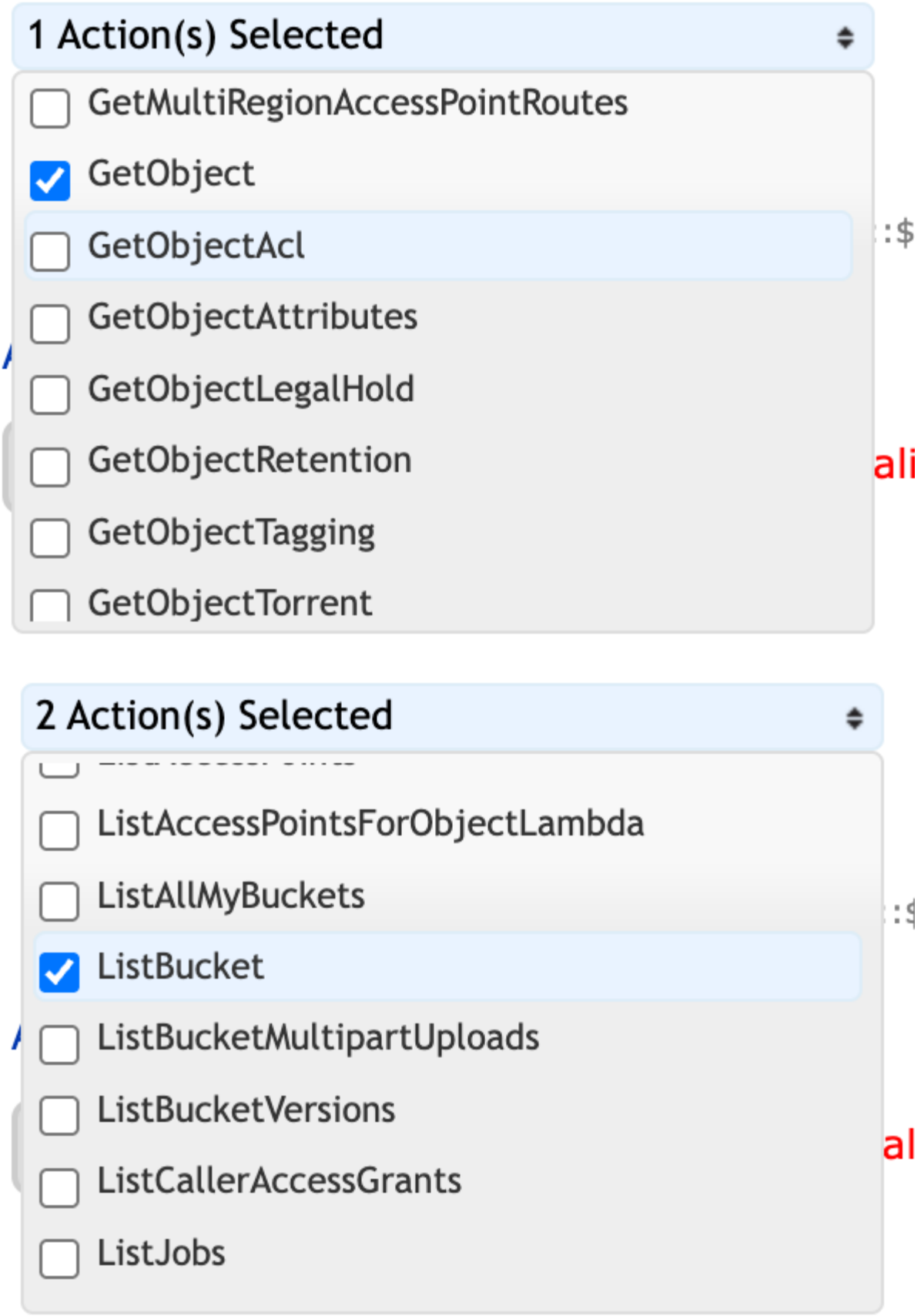
Par exemple, si vous souhaitez générer une stratégie de compartiment pour permettre à tous les utilisateurs d'effectuer des opérations GetObject sur tous les objets du compartiment, alors que seuls les utilisateurs appartenant au groupe « Marketing » du compte spécifié disposent d'un accès complet.

- Sélectionnez S3 Bucket Policy comme type de règle.
- Choisissez l'effet d'autorisation
- Entrez les informations du groupe Marketing - arn:aws:iam::95390887230002558202:Federated-group/Marketing
- Cliquez sur la case « toutes les actions ».
- Entrez les informations relatives au compartiment - arn:aws:s3:::example_bucket,arn:aws:s3:::example_bucket/*



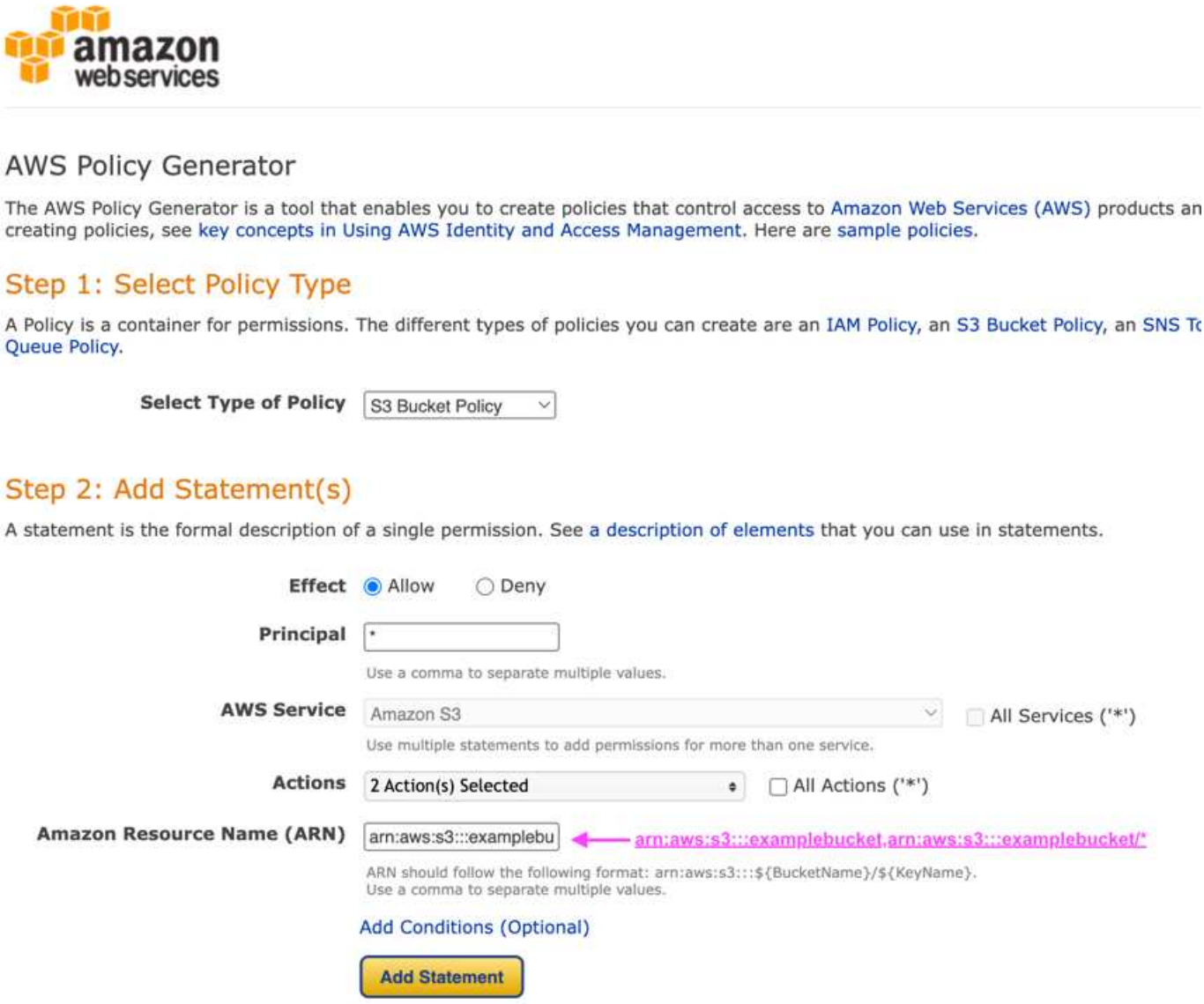
- Cliquez sur le bouton « Ajouter une déclaration »

- Choisissez l'effet d'autorisation
- Entrez l'astérisque +* pour tout le monde
- Cliquez sur la case en regard des actions GetObject et ListBucket »



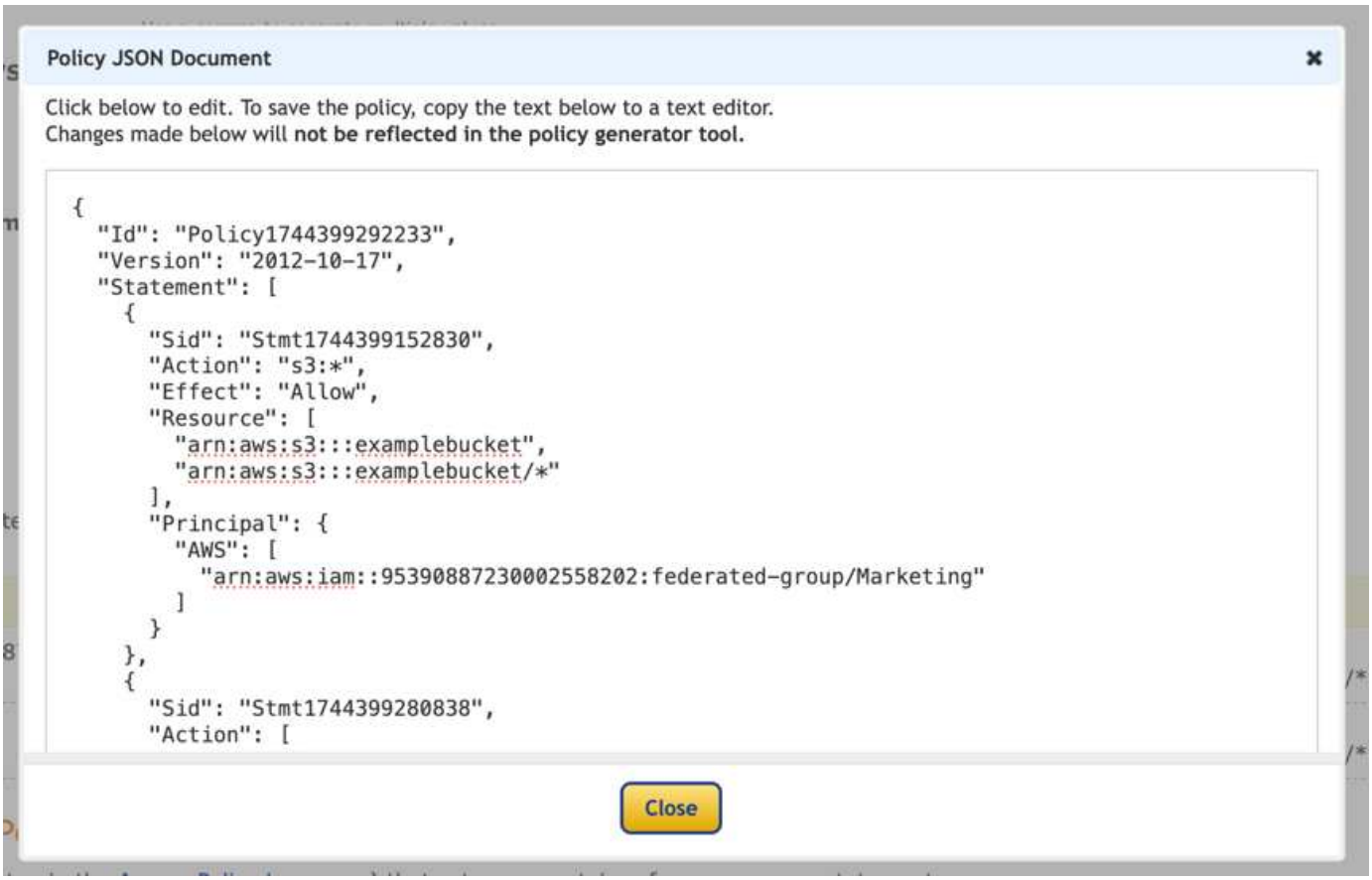
- Entrez les informations relatives au compartiment -

arn:aws:s3:::example_bucket,arn:aws:s3:::example_bucket/*



- Cliquez sur le bouton « Ajouter une déclaration »

- Cliquez sur le bouton « générer une politique » et une fenêtre contextuelle s'affiche avec votre police générée.



- Copiez le texte Json complet qui devrait ressembler à ceci :

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

Ce Json peut être utilisé tel quelle, ou vous pouvez supprimer les lignes ID et version au-dessus de la ligne « Statement » et vous pouvez personnaliser l'ID pour chaque autorisation avec un titre plus significatif pour chaque autorisation ou elles peuvent également être supprimées.

Par exemple :

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

Stratégies de groupe (IAM)

Accès au compartiment de style Home Directory

Cette stratégie de groupe autorise uniquement les utilisateurs à accéder aux objets du compartiment nommé nom d'utilisateur utilisateurs.

```
{
"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}/*"
    }

  ]
}
```

Refuser la création de compartiments de verrouillage d'objet

Cette stratégie de groupe empêche les utilisateurs de créer un compartiment avec le verrouillage d'objet activé sur le compartiment.

> Cette règle n'est pas appliquée dans l'interface utilisateur de StorageGRID et elle n'est appliquée que par l'API S3.

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

Limite de conservation du verrouillage des objets

Cette stratégie de compartiment limite la durée de conservation du verrouillage de l'objet à 10 jours ou moins

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

Empêcher les utilisateurs de supprimer des objets par ID de version

Cette stratégie de groupe empêche les utilisateurs de supprimer des objets multiversion par ID de version

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

Limiter un groupe à un sous-répertoire unique (préfixe) avec accès en lecture seule

Cette règle permet aux membres du groupe d'accéder en lecture seule à un sous-répertoire (préfixe) au sein d'un compartiment. Le nom du compartiment est « Study » et le sous-répertoire est « 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/*"
            ]
        }
    ]
}
```

Règles de compartiment

Restriction du compartiment à un seul utilisateur avec un accès en lecture seule

Cette stratégie permet à un seul utilisateur de disposer d'un accès en lecture seule à un compartiment et d'accéder explicitement à tous les autres utilisateurs. Le regroupement des déclarations de refus en haut de la politique est une bonne pratique pour une évaluation plus rapide.

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

limitez un compartiment à quelques utilisateurs disposant d'un accès en lecture seule.

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

Limitez les suppressions d'objets multiversion par l'utilisateur dans un compartiment

Cette stratégie de compartiment empêche un utilisateur (identifié par l'ID utilisateur « 56622399308951294926 ») de supprimer des objets multiversion par l'ID de version

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

Cycle de vie du bucket dans StorageGRID

Vous pouvez créer une configuration de cycle de vie S3 afin de contrôler la suppression d'objets spécifiques du système StorageGRID.

Qu'est-ce qu'une configuration de cycle de vie

La configuration du cycle de vie est un ensemble de règles appliquées aux objets dans des compartiments S3 spécifiques. Chaque règle indique quels objets sont affectés et quand ces objets vont expirer (à une date spécifique ou après un certain nombre de jours).

Chaque objet respecte les paramètres de conservation du cycle de vie d'un compartiment S3 ou une règle ILM. Lorsqu'un cycle de vie d'un compartiment S3 est configuré, les actions d'expiration du cycle de vie remplacent la règle ILM pour les objets correspondant au filtre de cycle de vie du compartiment. Les objets qui ne correspondent pas au filtre de cycle de vie des compartiments utilisent les paramètres de conservation de

la règle ILM. Si un objet correspond à un filtre de cycle de vie de compartiment et qu'aucune action d'expiration n'est explicitement spécifiée, les paramètres de conservation de la règle ILM ne sont pas utilisés et les versions d'objet sont conservées indéfiniment.

Par conséquent, il est possible de supprimer un objet de la grille, même si les instructions de placement d'une règle ILM s'appliquent toujours à l'objet. Ou bien, un objet peut être conservé sur la grille même après l'expiration des instructions de placement ILM pour l'objet.

StorageGRID prend en charge jusqu'à 1,000 règles de cycle de vie dans une configuration cycle de vie. Chaque règle peut inclure les éléments XML suivants :

- Expiration : supprimez un objet lorsqu'une date spécifiée est atteinte ou lorsqu'un nombre de jours spécifié est atteint, à partir de l'ingestion de l'objet.
- NonactuelVersionExexpiration : supprimez un objet lorsque le nombre de jours spécifié est atteint, à partir de quand l'objet est devenu non courant.
- Filtre (préfixe, étiquette)
- Statut *ID

StorageGRID prend en charge les opérations suivantes des compartiments pour gérer les configurations du cycle de vie :

- DeleteBuckeLifecycle
- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration

Structure d'une politique de cycle de vie

Comme première étape de la création de la configuration du cycle de vie, vous créez un fichier JSON qui inclut une ou plusieurs règles. Par exemple, ce fichier JSON contient trois règles, comme suit :

1. La **Règle 1** s'applique uniquement aux objets correspondant au préfixe category1/ et dont la valeur key2 est tag2. Le paramètre Expiration spécifie que les objets correspondant au filtre expireront à minuit le 22 août 2020.
2. La **Règle 2** s'applique uniquement aux objets correspondant au préfixe category2/. Le paramètre Expiration spécifie que les objets correspondant au filtre expireront 100 jours après leur ingestion.

   > Les règles spécifiant un nombre de jours sont relatives à l'ingestion de l'objet. Si la date actuelle dépasse la date d'ingestion et le nombre de jours, certains objets peuvent être supprimés du compartiment dès que la configuration de cycle de vie est appliquée.

3. La **Règle 3** s'applique uniquement aux objets correspondant au préfixe category3/. Le paramètre Expiration spécifie que toute version obsolète des objets correspondants expirera 50 jours après sa date d'expiration.

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

Appliquez la configuration du cycle de vie au compartiment

Après avoir créé le fichier de configuration du cycle de vie, vous l'appliquez à un compartiment en émettant une demande PutBucketLifecycleConfiguration.

Cette requête applique la configuration de cycle de vie du fichier d'exemple aux objets d'un compartiment nommé `testbucket`.

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

Pour vérifier qu'une configuration de cycle de vie a été correctement appliquée au compartiment, exécutez une demande GetBucketLifecycleConfiguration. Par exemple :

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

Exemples de politiques de cycle de vie pour les buckets standard (non versionnés)

Supprimer les objets après 90 jours

Cas d'utilisation : Cette stratégie est idéale pour gérer les données pertinentes pendant une durée limitée, telles que les fichiers temporaires, les journaux ou les données de traitement intermédiaire. Avantage : Réduisez les coûts de stockage et assurez-vous que le bucket est épuré.

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

Exemples de politiques de cycle de vie pour les buckets versionnés

Supprimer les versions non actuelles après 10 jours

Cas d'utilisation : Cette stratégie permet de gérer le stockage des objets de version obsolète, qui peuvent s'accumuler au fil du temps et consommer un espace important. Avantage : Optimisez l'utilisation du stockage

en conservant uniquement la version la plus récente.

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

Conserver 5 versions non actuelles

Cas d'utilisation : utile lorsque vous souhaitez conserver un nombre limité de versions précédentes à des fins de récupération ou d'audit. Avantage : conservez suffisamment de versions non actuelles pour garantir un historique et des points de récupération suffisants.

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

Supprimer les marqueurs de suppression lorsqu'aucune autre version n'existe

Cas d'utilisation : Cette politique permet de gérer les marqueurs de suppression restants après la suppression de toutes les versions obsolètes, qui peuvent s'accumuler au fil du temps. Avantage : Réduit l'encombrement inutile.

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

Supprimez les versions actuelles après 30 jours, supprimez les versions non actuelles après 60 jours et supprimez les marqueurs de suppression créés par la suppression de la version actuelle une fois qu'aucune autre version n'existe.

Cas d'utilisation : Fournir un cycle de vie complet pour les versions actuelles et obsolètes, y compris les marqueurs de suppression. Avantage : Réduire les coûts de stockage et garantir un bucket épuré tout en conservant suffisamment de points de récupération et d'historique.

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

supprimez les marqueurs de suppression qui n'ont pas d'autres versions, conservez 4 versions non actuelles et au moins 30 jours d'historique pour les objets avec le préfixe « accounts_ » et conservez 2 versions et au moins 10 jours d'historique pour toutes les autres versions d'objet.

Cas d'utilisation : Fournissez des règles uniques pour des objets spécifiques, en plus d'autres objets, afin de gérer l'intégralité du cycle de vie des versions actuelles et obsolètes, y compris les marqueurs de suppression. Avantage : Réduisez les coûts de stockage et assurez-vous que le bucket est épuré, tout en conservant suffisamment de points de récupération et d'historique pour répondre aux différents besoins des clients.

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

Conclusion

- Examinez et mettez à jour régulièrement les politiques de cycle de vie et alignez-les sur les objectifs de gestion ILM et de gestion des données.
- Testez les politiques dans un environnement ou un compartiment hors production avant de les appliquer à grande échelle pour vous assurer qu'elles fonctionnent comme prévu
- Utilisez des identifiants descriptifs pour les règles afin de les rendre plus intuitives, car la structure logique peut devenir complexe
- Surveillez l'impact de ces politiques de cycle de vie de bucket sur l'utilisation et les performances du stockage pour effectuer les ajustements nécessaires.

Informations sur le copyright

Informations sur les marques commerciales