



Procedures and API examples

How to enable StorageGRID in your environment

NetApp
April 26, 2024

Table of Contents

- Procedures and API examples 1
 - Test and demonstrate S3 encryption options on StorageGRID 1
 - Test and demonstrate S3 object lock on StorageGRID 4
 - Example bucket and Group(IAM) policies 9

Procedures and API examples

Test and demonstrate S3 encryption options on StorageGRID

StorageGRID and the S3 API offer a number of different ways to encrypt your data at rest. To learn more, see [Review StorageGRID encryption methods](#).

This guide will demonstrate the S3 API encryption methods.

Server Side Encryption (SSE)

SSE allows the client to store an object and encrypt it with a unique key that is managed by StorageGRID. When the object is requested, the object is decrypted by the key stored in storageGRID.

SSE Example

- PUT an object with SSE

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

- HEAD the object to verify encryption

```
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": {}  
}
```

- GET the object

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

Server Side Encryption with Customer provided keys (SSE-C)

SSE allows the client to store an object and encrypt it with a unique key that is provided by the client with the object. When the object is requested, the same key must be provided in order to decrypt and return the object.

SSE-C Example

- For testing or demonstration purposes you can create an encryption key
 - Create an encryption key

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

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

- Put an object with the generated key

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

- Head the object

```
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=="  
}
```



If you do not provide the encryption key, you will receive an error "An error occurred (404) when calling the HeadObject operation: Not Found"

- Get the object

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



If you do not provide the encryption key, you will receive an error "An error occurred (InvalidRequest) when calling the GetObject operation: The object was stored using a form of Server Side Encryption. The correct parameters must be provided to retrieve the object."

Bucket Server Side Encryption (SSE-S3)

SSE-S3 allows the client to define a default encryption behavior for all objects stored in a bucket. The objects are encrypted with a unique key that is managed by StorageGRID. When the object is requested, the object is decrypted by the key stored in storageGRID.

Bucket SSE-S3 Example

- Create a new bucket and set a default encryption policy
 - Create new bucket

```
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
```

- Put an object in the bucket

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

- Head the object

```
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": {}
}
```

- GET the object

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

By Aron Klein

Test and demonstrate S3 object lock on StorageGRID

Object Lock provides a WORM model to prevent objects from being deleted or overwritten. StorageGRID implementation of object lock is Cohasset assessed to help meet regulatory requirements, supporting legal hold and compliance mode for object retention, and default bucket retention policies.

This guide will demonstrate the S3 Object Lock API.

Legal hold

- Object Lock legal hold is a simple on/off status applied to an object.

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

- Verify it with a GET operation.

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

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

- Turn legal hold off

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

- Verify it with a GET operation.

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

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

Compliance mode

- The object retention is done with a retain until timestamp.

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

- Verify the retention status

```
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"
  }
}
```

Default retention

- Set the retention period in days and years verses a retain until date defined with the per object api.

```
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
```

- Verify the retention status

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

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

- Put an object in the bucket

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

- The retention duration set on the bucket is converted to a retention timestamp on the object.


```
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 deleting an object with a defined retention

Object Lock is built on top of versioning. The retention is defined on a version of the object. If an attempt is made to delete an object with a retention defined, and no version is specified, a delete marker is created as the current version of the object.

- Delete the object with retention defined

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

- List the objects in the bucket

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

- Notice the object is not listed.
- List versions to see the delete marker, and the original locked version

```
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":
"RDVDMjYwMTQtQkNDQS0xMUVDLThGOEUtNjQ3NTAwQzAxQTkl",
      "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"
    }
  ]
}
```

- Delete the locked version of the object

```
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

By Aron Klein

Example bucket and Group(IAM) policies

Here are examples of bucket policies and group policies(IAM Policies).

Group Policies (IAM)

Home Directory style bucket access

This group policy will only allow users to access objects in the bucket named the users username.

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

Deny object lock bucket creation

This group policy will restrict users from creating a bucket with object lock enabled on the bucket.



This policy is not enforced in the StorageGRID UI, it is only enforced by S3 API.

```

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

```

Object lock retention limit

This Bucket policy will restrict Object-Lock retention duration to 10 days or less

```

{
  "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"
        }
      }
    }
  ]
}

```

Restrict users from deleting objects by versionID

This group policy will restrict users from deleting versioned objects by versionID

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

This bucket policy will restrict a user(identified by userID "56622399308951294926") from deleting versioned objects by versionID

```

{
  "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"
        ]
      }
    }
  ]
}

```

Restrict bucket to single user with read-only access

This policy allows a single user to have read-only access to a bucket and explicitly denies access to all other users. Grouping the Deny statements at the top of the policy is a good practice for faster evaluation.

```

{
  "Statement": [
    {
      "Sid": "Deny non user1",
      "Effect": "Deny",
      "NotPrincipal": {
        "AWS":
"urn:sgws:identity::34921514133002833665:user/user1"
      },
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "urn:sgws:s3::bucket1",
        "urn:sgws:s3::bucket1/*"
      ]
    },
    {
      "Sid": "Allow user1 read access to bucket bucket1",
      "Effect": "Allow",
      "Principal": {
        "AWS":
"urn:sgws:identity::34921514133002833665:user/user1"
      },
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "urn:sgws:s3::bucket1",
        "urn:sgws:s3::bucket1/*"
      ]
    }
  ]
}

```

Restrict a group to single subdirectory (prefix) with read-only access

This policy allows members of the group to have read-only access to a subdirectory (prefix) within a bucket. The bucket name is "study" and the subdirectory is "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/*"
    ]
  }
}
```

Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.