You can also create EventBridge rules to match the CloudTrail events for these API calls. For more information about CloudTrail events, see Enable logging for objects in a bucket using the console. For more information about EventBridge events, see Events from AWS services.

You can use CloudTrail logs for object-level Amazon S3 actions to track PUT and POST requests to Amazon S3. You can use these actions to verify whether default encryption is being used to encrypt objects when incoming PUT requests don't have encryption headers.

When Amazon S3 encrypts an object by using the default encryption settings, the log includes one of the following fields as the name-value pair: `"SSEApplied":"Default_SSE_S3"`, `"SSEApplied":"Default_SSE_KMS"`, or `"SSEApplied":"Default_DSSE_KMS"`.

When Amazon S3 encrypts an object by using the PUT encryption headers, the log includes one of the following fields as the name-value pair: `"SSEApplied":"SSE_S3"`, `"SSEApplied":"SSE_KMS"`, `"SSEApplied":"DSSE_KMS"`, or `"SSEApplied":"SSE_C"`.

For multipart uploads, this information is included in your `InitiateMultipartUpload` API operation requests. For more information about using CloudTrail and CloudWatch, see Monitoring Amazon S3.

Working with Mountpoint for Amazon S3

Mountpoint for Amazon S3 is a high-throughput open source file client for mounting an Amazon S3 bucket as a local file system. With Mountpoint, your applications can access objects stored in Amazon S3 through file system operations, such as open and read. Mountpoint automatically translates these operations into S3 object API calls, giving your applications access to the elastic storage and throughput of Amazon S3 through a file interface.

Mountpoint for Amazon S3 is generally available for production use on your large-scale read-heavy applications: data lakes, machine learning training, image rendering, autonomous vehicle simulation, extract, transform, and load (ETL), and more.

Mountpoint supports basic file system operations, and can read files up to 5 TB in size. It can list and read existing files, and it can create new ones. It cannot modify existing files or delete directories, and it does not support symbolic links or file locking. Mountpoint is ideal for applications that do not need all of the features of a shared file system and POSIX-style permissions but require Amazon S3's elastic throughput to read and write large S3 datasets. For details, see Mountpoint file system behavior on GitHub. For workloads that require full POSIX support, we recommend Amazon FSx for Lustre and its support for linking S3 buckets.

Mountpoint for Amazon S3 is available only for Linux operating systems. You can use Mountpoint to access S3 objects in all storage classes except S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Intelligent-Tiering Archive Access Tier, and S3 Intelligent-Tiering Deep Archive Access Tier.

Topics

- Installing Mountpoint
- Configuring and using Mountpoint

Installing Mountpoint

You can download and install prebuilt packages of Mountpoint for Amazon S3 by using the command line. The instructions for downloading and installing Mountpoint vary, depending on which Linux operating system that you're using.

Topics

- RPM-based distributions (Amazon Linux, Fedora, CentOS, RHEL)
- DEB-based distributions (Debian, Ubuntu)
- Other Linux distributions
- Verifying the signature of the Mountpoint for Amazon S3 package

RPM-based distributions (Amazon Linux, Fedora, CentOS, RHEL)

1. Copy the following download URL for your architecture.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm
   ```

   ARM64 (Graviton):

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.rpm
   ```

2. Download the Mountpoint for Amazon S3 package. Replace *`download-link`* with the appropriate download URL from the preceding step.

```
wget download-link
```

3. (Optional) Verify the authenticity and integrity of the downloaded file. First, copy the appropriate signature URL for your architecture.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm.asc
   ```

   ARM64 (Graviton):

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.rpm.asc
   ```

   Next, see Verifying the signature of the Mountpoint for Amazon S3 package.

4. Install the package by using the following command:

   ```
   sudo yum install ./mount-s3.rpm
   ```

5. Verify that Mountpoint is successfully installed by entering the following command:

   ```
   mount-s3 --version
   ```

   You should see output similar to the following:

   ```
   mount-s3 1.3.1
   ```

DEB-based distributions (Debian, Ubuntu)

1. Copy the download URL for your architecture.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.deb
   ```

   ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.deb
```

2. Download the Mountpoint for Amazon S3 package. Replace `download-link` with the appropriate download URL from the preceding step.

```
wget download-link
```

3. (Optional) Verify the authenticity and integrity of the downloaded file. First, copy the signature URL for your architecture.

   x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.deb.asc
```

   ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.deb.asc
```

   Next, see Verifying the signature of the Mountpoint for Amazon S3 package.

4. Install the package by using the following command:

```
sudo apt-get install ./mount-s3.deb
```

5. Verify that Mountpoint for Amazon S3 is successfully installed by running the following command:

```
mount-s3 --version
```

   You should see output similar to the following:

```
mount-s3 1.3.1
```

Other Linux distributions

1. Consult your operating system documentation to install the `FUSE` and `libfuse2` packages, which are required.

2. Copy the download URL for your architecture.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.tar.gz
   ```

   ARM64 (Graviton):

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.tar.gz
   ```

3. Download the Mountpoint for Amazon S3 package. Replace *`download-link`* with the appropriate download URL from the preceding step.

   ```
   wget download-link
   ```

4. (Optional) Verify the authenticity and integrity of the downloaded file. First, copy the signature URL for your architecture.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.tar.gz.asc
   ```

   ARM64 (Graviton):

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.tar.gz.asc
   ```

   Next, see Verifying the signature of the Mountpoint for Amazon S3 package.

5. Install the package by using the following command:

   ```
   sudo mkdir -p /opt/aws/mountpoint-s3 && sudo tar -C /opt/aws/mountpoint-s3 -xzf ./
   mount-s3.tar.gz
   ```

6. Add the `mount-s3` binary to your PATH environment variable. In your `$HOME/.profile` file, append the following line:

   ```
   export PATH=$PATH:/opt/aws/mountpoint-s3/bin
   ```

   Save the `.profile` file, and run the following command:

```
source $HOME/.profile
```

7. Verify that Mountpoint for Amazon S3 is successfully installed by running the following command:

   ```
   mount-s3 --version
   ```

   You should see output similar to the following:

   ```
   mount-s3 1.3.1
   ```

Verifying the signature of the Mountpoint for Amazon S3 package

1. Install GnuPG (the gpg command). It is required to verify the authenticity and integrity of a downloaded Mountpoint for Amazon S3 package. GnuPG is installed by default on Amazon Linux Amazon Machine Images (AMIs). After you installGnuPG, proceed to step 2.
2. Download the Mountpoint public key by running the following command:

   ```
   wget https://s3.amazonaws.com/mountpoint-s3-release/public_keys/KEYS
   ```

3. Import the Mountpoint public key into your keyring by running the following command:

   ```
   gpg --import KEYS
   ```

4. Verify the fingerprint of the Mountpoint public key by running the following command:

   ```
   gpg --fingerprint mountpoint-s3@amazon.com
   ```

   Confirm that the displayed fingerprint string matches the following:

   ```
   673F E406 1506 BB46 9A0E  F857 BE39 7A52 B086 DA5A
   ```

   If the fingerprint string doesn't match, do not finish installing Mountpoint, and contact AWS Support.

5. Download the package signature file. Replace *`signature-link`* with the appropriate signature link from the preceding sections.

```
wget signature-link
```

6. Verify the signature of the downloaded package by running the following command. Replace *signature-filename* with the file name from the previous step.

   ```
   gpg --verify signature-filename
   ```

   For example, on RPM-based distributions, including Amazon Linux, enter the following command:

   ```
   gpg --verify mount-s3.rpm.asc
   ```

7. The output should include the phrase `Good signature`. If the output includes the phrase `BAD signature`, redownload the Mountpoint package file and repeat these steps. If the issue persists, do not finish installing Mountpoint, and contact AWS Support.

   The output may include a warning about a trusted signature. This does not indicate a problem. It only means that you have not independently verified the Mountpoint public key.

Configuring and using Mountpoint

To use Mountpoint for Amazon S3, your host needs valid AWS credentials with access to the bucket or buckets that you would like to mount. For different ways to authenticate, see Mountpoint AWS Credentials on GitHub.

For example, you can create a new AWS Identity and Access Management (IAM) user and role for this purpose. Make sure that this role has access to the bucket or buckets that you would like to mount. You can pass the IAM role to your Amazon EC2 instance with an instance profile.

Using Mountpoint for Amazon S3

Use Mountpoint for Amazon S3 to do the following:

1. Mount buckets with the `mount-s3` command.

   In the following example, replace *DOC-EXAMPLE-BUCKET* with the name of your S3 bucket, and replace *~/mnt* with the directory on your host where you want your S3 bucket to be mounted.

```
mkdir ~/mnt
mount-s3 DOC-EXAMPLE-BUCKET ~/mnt
```

Because the Mountpoint client runs in the background by default, the `~/mnt` directory now gives you access to the objects in your S3 bucket.

2. Access the objects in your bucket through Mountpoint.

   After you mount your bucket locally, you can use common Linux commands, such as `cat` or `ls`, to work with your S3 objects. Mountpoint for Amazon S3 interprets keys in your S3 bucket as file system paths by splitting them on the forward slash (/) character. For example, if you have the object key `Data/2023-01-01.csv` in your bucket, you will have a directory named `Data` in your Mountpoint file system, with a file named `2023-01-01.csv` inside it.

   Mountpoint for Amazon S3 intentionally does not implement the full POSIX standard specification for file systems. Mountpoint is optimized for workloads that need high-throughput read and write access to data stored in Amazon S3 through a file system interface, but that otherwise do not rely on file system features. For more information, see Mountpoint for Amazon S3 file system behavior on GitHub. Customers that need richer file system semantics should consider other AWS file services, such as Amazon Elastic File System (Amazon EFS) or Amazon FSx.

3. Unmount your bucket by using the `umount` command. This command unmounts your S3 bucket and exits Mountpoint.

   To use the following example command, replace `~/mnt` with the directory on your host where your S3 bucket is mounted.

   ```
   umount ~/mnt
   ```

   Note

   To get a list of options for this command, run `umount --help`.

For additional Mountpoint configuration details, see S3 bucket configuration, and file system configuration on GitHub.

Configuring caching in Mountpoint

When you use Mountpoint for Amazon S3, you can configure it to cache the most recently accessed data from your S3 buckets on Amazon EC2 instance storage or an attached Amazon EBS volume. Caching this data can help to accelerate performance and reduce the cost of repeated data access. Caching in Mountpoint is ideal for use cases where you repeatedly read the same data that doesn't change during the multiple reads. For example, you can use caching with machine learning training jobs that need to read a training dataset multiple times to improve model accuracy.

When you mount an S3 bucket, you can optionally enable caching through flags. You can configure the location and size of the data cache and the amount of time metadata is retained in the cache. When you mount a bucket and caching is enabled, Mountpoint creates an empty sub-directory at the configured cache location, if that sub-directory doesn't already exist. When you first mount a bucket and when you unmount, Mountpoint deletes the contents of the cache location. For more information about configuring and using caching in Mountpoint, see Mountpoint for Amazon S3 Caching configuration on GitHub.

When you mount an S3 bucket, you can enable caching with the `--cache` *CACHE_PATH* flag. In the following example, replace *CACHE_PATH* with the filepath to the directory that you want to cache your data in. Replace *DOC-EXAMPLE-BUCKET* with the name of your S3 bucket, and replace *~/mnt* with the directory on your host where you want your S3 bucket to be mounted.

```
mkdir ~/mnt
mount-s3 --cache CACHE_PATH DOC-EXAMPLE-BUCKET ~/mnt
```

> ⚠ **Important**
>
> If you enable caching, Mountpoint will persist unencrypted object content from your S3 bucket at the caching location configured at mount. In order to protect your data, we recommend that you restrict access to the data cache location.

Troubleshooting Mountpoint

Mountpoint for Amazon S3 is backed by AWS Support. If you need assistance, contact the AWS Support Center.

You can also review and submit Mountpoint Issues on GitHub.