Client Data Sharing Specification

03/31/2026 5:12 am EDT

Exchanging data with ID5

Our main delivery mechanism is to exchange data directly to an AWS S3 bucket or an S3-compatible object storage (e.g., GCS) controlled by the client.

There are several different ways to share data:

1. Client-Hosted Bucket (push method)

The preferred method is for clients to host their own AWS S3 or S3-compatible storage. To help with this, ID5 will share our AWS account ID.

The required permissions and examples for select cloud providers are detailed below.

2. ID5-Hosted Buckets (pull method)

For clients unable to host their own S3 compatible storage or some specificuse cases where pull is better, we offer an alternative where ID5 hosts the bucket for data sharing. In this scenario, we support two methods for client authentication and access which require the client to have an AWS account:

Direct Cross-Account Access via Bucket Policy

We attach a bucket policy to the hosted bucket that explicitly grants read (and/or write) permissions to principals in the client's AWS account. Once this is in place, the clients are free to manage access on your side — for example, they can delegate permissions to specific IAM users, roles, or applications within their account through your own IAM policies. This approach is straightforward and works well when their team already has established IAM governance.

IAM Role-Based Access (AssumeRole)

We create a dedicated IAM role in our AWS account with the appropriate permissions on the bucket and configure its trust policy to allow the client's account to assume it. Client's applications then call `sts:AssumeRole` to obtain short-lived, scoped credentials. This is the recommended approach for programmatic or automated workloads, as it avoids the need for long-lived credentials and provides a clear audit trail through CloudTrail.

In the case of either method, we require the client's AWS account ID(s) or canonical user ID(s). The choice between these options will depend on the client's security preferences and existing AWS setup.

For security reasons, ID5 **does not** support creating buckets or providing client access via Access and Secret keys.

Data Retention

For all storage hosted by ID5, our data retention policy expires data after 90 days.

Supported Data Formats

We prefer to deliver and receive data in Parquet format with zstd compression, however we are also able to support CSV and JSON, compressed with gzip or zstd as well as snappy for Parquet.

Permissions Details

AWS S3 Bucket Policy Requirements

- Our account ARN is: **arn:aws:iam::243105029713:root**
- We can handle either an entire bucket or a specific prefix.

Permissions needed

- For verification purposes:
  - s3:ListBucket
  - s3:GetObject

- For uploading:
  - s3:PutObject
  - s3:DeleteObject

Policy Example

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::243105029713:root"
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name/some/prefix/*"
      ]
    },
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::243105029713:root"
      },
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name"
      ],
      "Condition": {
        "StringLike": {
          "s3:prefix": [
            "some/prefix/*"
          ]
        }
      }
    }
  ]
}
```

Other S3-compatible services

- For S3-compatible services, we require that the client provide us with:
  - The endpoint URL
  - An Access Key ID
  - A Secret Access Key
- The bucket should have a policy equivalent to that granted to an AWS S3 Bucket.
  E.g. s3:ListBucket, s3:GetObject, s3:PutObject, s3:DeleteObject, etc.

GCP

- For GCP integration, we require one of the following:
  - HMAC keys with matching access to the storage. This leverages the interoperability layer offered by GCP
  - a JSON key for a service account with permissions matching those described above

Azure

- Azure does not offer first-party S3 compatibility, but several open-source services re-expose the API as S3-compatible.