Client Data Sharing Specification

09/09/2025 11:40 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

The preferred method is for clients to host their own 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 (Alternative Option)

For clients unable to host their own S3-compatible storage, we offer an alternative where ID5 hosts the bucket for data sharing.

In this setup, we support two methods for client authentication and access.

Direct IAM User Access

- We can grant permissions directly to the client's AWS account.
- The client can delegate access as required

IAM Role-Based Access

- We provide an IAM role to the client.
- Client applications can then assume the role, gaining the correct permissions to access the bucket.

For either method, we need the client's AWS account ID(s) or canonical user ID(s). The decision between these options will depend on the client's security preferences and existing AWS setup.

⚠ For security reasons, ID5 **does not** natively 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 receive data in Parquet format; however, we also support CSV and JSON, compressed with gzip or ZSTD.

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 the client to generate HMAC keys with matching access to the storage.
- This leverages the interoperability layer offered by GCP.

Azure

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